Combinatorial Semantics and Idiomatic Expressions in Head-Driven Phrase Structure Grammar

Manfred Sailer

Philosophische Dissertation
angenommen von der Neuphilologischen Fakultät
der Universität Tübingen
am 15. November 2000

Tübingen
2003

Gedruckt mit Genehmigung der Neuphilologischen Fakultät
der Universität Tübingen

Hauptberichterstatter:	Prof. Dr. Erhard W. Hinrichs
1. Mitberichterstatter:	Priv. Doz. Dr. Friedrich Hamm
2. Mitberichterstatter:	Prof. Dr. Uwe Mönnich
3. Mitberichterstatter:	Prof. Dr. Jürgen Pafel
Dekan	Prof. Dr. Tilman Berger

Contents

Acknowledgments

It is a great pleasure to thank my teachers, colleagues and friends, all the people who have helped me with their comments, questions and grammaticality judgments, and the family who has supported me throughout my work on this thesis.

Far too much time has passed between the completion of this thesis and its publication. Therefore, the sources consulted are dated prior to 2001 despite the considerable attention paid to combinatorial semantics and idiomatic expressions within the last two years. However, since I am still working in the field, I hope to be able to do justice to this research in other publications.

I was first introduced to the HPSG formalism in Paul King's classes. The feedback and confidence I received through the exchange of ideas with Thilo Götz, Tilmann Höhle, Paul King, Gerald Penn, Carl Pollard, Adam Przepiórkowski, Frank Richter and Kiril Simov encouraged many of the points which I developped in my thesis.

I am especially grateful for the fruitful environment in which discussions on the usage of this formalism for writing grammars could flourish among all of us in the HPSG group in Tübingen: Mike Calcagno, Dale Gerdeman, Erhard Hinrichs, Tilmann Höhle, Valia Kordoni, Anna Kupść, Kordula de Kuthy, Detmar Meurers, Guido Minnen, Paola Monachesi, Gerald Penn, Adam Przepiórkowski, Lucia Reinhard, Sabine Reinhard, and Frank Richter.

The incorporation of standard semantics into HPSG was nurtured by the advise and inspirational climate in which many ideas of my thesis could mature. I thank the following people for their contribution: Ralf Albrecht, Sigrid Beck, Fritz Hamm, Herman Hendriks, Erhard Hinrichs, Tilmann Höhle, Tibor Kiss, Wim Klooster, Valia Kordoni, Uwe Mönnich, Adam Przepiórkowski, Satoshi Tomioka. I would also like mention Klaus Schulz for his excelent treatment of Montague grammar in his introductory semantics class.

Thanks also to Tilmann Höhle for awakening my interest in idiomatic expressions and to Anna Siewierska in whose class on *Functional Grammar* I could explore the collocational perspective which ripened into the theory presented in Part II.

I would like to express my appreciation to the people who enriched my understanding of related topics: Michaela Banzhaf, Joana Błaszczak, Mily Crevels, Francis Corblin, Agnès Gobbo, Paweł Karnowski, Inge Pflugfelder, Danuta Przepiórkowska, Beata Trawiński, Ruben van der Vijfer.

A special thank goes to the brave people that agreed to read drafts of this thesis or helped to improve my English: Fritz Hamm, Erhard Hinrichs, Laura Kallmeyer, Kordula de Kuthy, Detmar Meurers, Gerald Penn, Guergana Popova, Adam Przepiórkowski, Janina Radó, Christian Richard, Frank Richter, Jesse Tseng. Carmella Payne, finally, proofread the entire thesis.

For technical and *tex*-nical support, I am grateful to Michael Betsch, Detmar Meurers, Gerald Penn and Jochen Saile.

All these people have directly contributed to the genesis of this work. Just as importantly, I would like to thank everyone; Your presence made the writing of this thesis enjoyable.

I would also like to express my gratitude to my reviewers, Erhard Hinrichs, Fritz Hamm, Uwe Mönnich and Jürgen Pafel, whose comments encourage me on to further research.

Special thanks also to the members of the Special Research Program *Linguistic Data Structures* (SFB 441) for accomodating the continuation of my work on collocations and in particular to the people who have worked with me on the project so far: Carmella Payne, Frank Richter and Jan-Philipp Soehn.

Enjoying that occasional cup of coffee helped bring balance to my life. I am grateful to those who managed to make me forget about the thesis from time to time: Chedva and Shimshon Bar-Yehuda, Martin Engelhard, Claudia Ferber, Michael Kaschek, Heidi, Joschka and Sofia Müller, Lars Optehostert, Inge Pflugfelder, Susanne, Christian, Karla and Bernhard Traulsen.

A very special thanks goes to my parents and to my in-laws for being there whenever I needed help. The only reward that I can offer to them is to try and do my best to be as supportive to my son as they have been to my wife and me.

Finally, I am grateful to my wife Anne and to our son Ruben. They put up with me through all my enthusiasm, frustration and procrastination in writing this thesis.

Introduction

One of the most striking and, at the same time, most intriguing properties of language is the harmonious interplay of regularity and irregularity. While in language these two are as inseparable as two sides of a coin, different grammatical traditions emphasize either the regularities of language or its irregularities. In either case, the "other side of the coin" provides a rich source of counterexamples to the claims made by the respective theory. Since it is obvious that both phenomena are part of natural language, any architecture of grammar that ignores one of them is doomed to be empirically inadequate. To do justice to regularity, the grammar must minimize the areas of irregularity and express as many generalizations as possible. To acknowledge the irregularity of language, the tools used to account for irregularity should fit in a natural way into the overall architecture. In this thesis, we develop the framework of *Head-Driven Phrase Structure Grammar* (HPSG) further in the direction of a theory that meets these two requirements.

In this thesis, we are concerned with the interaction of regularity and irregularity in combinatorial semantics, i.e., in the way complex syntactic structures are interpreted. The empirical domain of our study is the varying degree of syntactic and semantic regularity attested in the continuum between free combinations, (more or less flexible) idiomatic expressions, and fully fixed expressions. We will explore in this introduction and, detail in later chapters of this thesis that the sentences in (1) contain VPs which exhibit different degrees of regularity and irregularities.

(1) a. John saw Mary.
b. Pat spilled the beans.
c. Pat tripped the light fantastic.

In this introduction, we give an intuitive outline of the problems and of the solution that we are concerned with throughout this work.

Within formal grammar, it is assumed that the meaning of a syntactically complex entity can be computed from the meaning of its parts and from knowledge about how to interpret their specific way of combination. This underlying assumption is also called the *Principle of Compositionality* or *Frege's Principle*, which we quote in (2).[1]

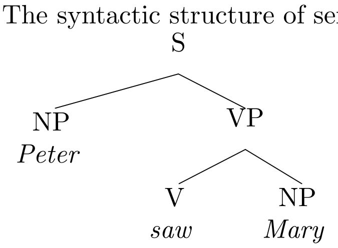
[1]While it is uncontroversial that this principle is referred to as *Frege's Principle* (Partee et al., 1993; Gamut, 1991a), Janssen 1997 (pp. 420f.) argues that it was not explicitly stated in Frege's work. Janssen even claims that

> "The conclusion is that Frege rejected the principle of compositionality in the period in which he wrote *Grundlagen der Mathematik* [= *Grundlagen der Arithmetik*, M.S.], but may have accepted the principle later on in his life. It seems that nowhere in his published works does he mention compositionality as a principle. It is, therefore, inaccurate to speak of 'Frege's principle'. Compositionality is not Frege's, but it might be called 'Fregean' because it is in the spirit of his later writings." (Janssen, 1997, p. 421)

(2) The *Principle of Compositionality* (Janssen, 1997, p. 419)
The meaning of a compound expression is a function of the meanings of its parts.

According to the principle of compositionality, the meaning of the simple sentence in (1a) can be computed given the syntactic structure in (3) and knowledge of the meaning of the non-compound expressions that occur in this sentence, i.e., the words *Peter*, *saw*, and *Mary*.

(3) The syntactic structure of sentence (1a):



In this example, the non-compound expressions are the terminal nodes in the syntactic tree, which correspond to *words*. The non-terminal nodes in the tree, the *phrases*, are the compound expressions. Adopting the architecture of HPSG, we assume that a word is licensed by some *lexical entry*. A phrase, on the other hand is licensed by some general rule of syntactic combination, called *Immediate Dominance Schema*. There is a clear correspondence between the syntactic and the semantic side of the sentence: it is precisely for words that we must give an explicit interpretation, and it is precisely for phrases that we can compute the interpretation according to a given rule.

HPSG is a sign-based framework, which means that words and phrases are considered *linguistic signs*. A sign contains syntactic and semantic information. Thus, the lexical entry of a word does not only contain its syntactic specification, but also some specification of the meaning of the word. The meaning of a phrase is determined by some general principle of the grammar. This division of labor is a direct incorporation of the principle of compositionality.

The principle of compositionality contrasts with another principle, which, according to Janssen 1997 (p. 420), was explicitly adopted by Frege: the *Principle of Contextuality*.

(4) The *Principle of Contextuality*:
One should ask for the meaning of a word only in the context of a sentence, and not in isolation.

To illustrate the kind of contextuality that we are concerned with in this thesis, consider the following sentence.

(5) Pat spilled the beans.

Sentence (5) is ambiguous. In one reading, it expresses that Pat caused some vegetables to fall out of a container, in a second reading, Pat disclosed a secret. For both readings, we can give an interpretation to the words *spill* and *beans* such that the meaning of the sentence is a function of the meaning of its parts. We must, however interpret *beans* as a secret exactly if we interpret *spill* as a telling. While in (5), the interpretation of the word *beans* depends on the interpretation of the word *spill*, the meaning *secret* is not available in the following context, i.e., the sentences in (6) are not synonymous.

(6) a. Pat didn't like beans.
b. Pat didn't like secrets.

What we illustrated so far is that the meaning of a phrase can be determined on the basis of the meaning of its parts. In contrast to this, the meaning of words is not predictable in this sense and, furthermore, may depend in an idiosyncratic way on the meaning of the overall sentence.

The system that results under such an interpretation of compositionality and contextuality distinguishes clearly between the *regular* aspects of interpretation, attested for phrases, and the *irregular* aspects, attributed to words. The regular assignment of meaning is done by some function at the level of phrases, the irregular assignment of meaning is done at the level of words in the lexicon. The irregularity, thereby, comes in two kinds: first, it is an idiosyncratic property of a word *what* its meaning is, and second, it is another idiosyncratic property of a word *when* it may occur with a particular meaning, i.e., in which contexts.

We can show, however, that this architecture of grammar cannot be maintained: there are *irregular phrases* as well as *regular* words. Thus, if we assume that all phrases are regular, we cannot account for the data, and if we assume that all words are irregular, we risk missing generalization. We will first show the existence of irregular phrases. For this purpose, consider sentence (7).

(7) Pat tripped the light fantastic. (= Pat danced.)

It is generally assumed (Wasow et al., 1983) that the string *tripped the light fantastic* is syntactically complex, i.e., a phrase. However, it is not built according to the syntactic rules of English. In addition, the meaning of this phrase cannot be computed from that of its parts.

This indicates that neither the syntactic properties nor the meaning of the phrase *trip the light fantastic* are licensed by some rule. In this respect, it is similar to the words that we considered so far. Since the phrase must be licensed somehow by the grammar and since this cannot be done by the regular principles of syntax and semantics, we must assume that it is licensed by some *lexical entry*. To achieve this, we must extend our notion of lexical elements to include irregular phrases.

Furthermore, there are also words with regular syntactic and semantic properties. This can be illustrated with passive participles. In many HPSG analyses, including that of Pollard and Sag 1994 (pp. 121f.) and the one assumed in this thesis, a passive participle such as *seen* in (8a) is considered to be derived from an active participle as it occurs in (8b). The relation between the active and the passive participle is expressed by some *Lexical Rule*.

(8) a. Mary was seen.
b. John had seen Mary.

There is no doubt that the passive participle is a word. On the other hand, its syntactic properties are fully predictable on the basis of those of the active participle and the knowledge which Lexical Rule is used to relate these two participles. The same is true of the semantic properties of the passive participle. Given the meaning of the active participle and the information that the word *seen* in (8a) is passive, we can predict the meaning of the passive participle.

With *trip the light fantastic* we have found a phrase which is irregular. Passive participles, on the other hand, are words that are regular. Therefore, we can no longer maintain the assumption that the regular/irregular-distinction can be reduced to the phrase/word dichotomy. In Table 0.1 we summarize this result, pointing to examples from above that show that words and phrases can be cross-classified with respect to the criterion of regularity.

In this thesis, we propose some architectural changes to the theory of HPSG which will enable us to account for the fact that all four slots in Table 0.1 are filled. These changes will be conservative with respect to the presentation of HPSG in Pollard and Sag 1994, because we adopt the same word/phrase distinction made there, and the same distinction between

TABLE 0.1. The cross-classification of regularity/irregularity and word/phrase:

	word		phrase	
irregular	*saw*	in (1a)	*tripped the light fantastic*	in (7)
	spilled	in (5)		
regular	*seen*	in (8a)	*saw Mary*	in (1a)

derived and non-derived words. In addition, the system that we will present preserves the *lexicalism* of Pollard and Sag 1994 in the sense that we locate all irregularity in the lexicon. As we want to do justice to the empirical facts, we will introduce lexical entries not only for non-derived words, but also for irregular phrases.

This means that the words and the phrase in the upper slots of Table 0.1 are licensed by a lexical entry. We will, therefore, call them *lexical elements*. On the other hand, the VP *saw Mary* as it occurs in (1a) is licensed by some syntactic rule, an ID-Schema, just as all phrases are in Pollard and Sag 1994. Finally, the passivized verb *seen* in the last line in the table is licensed by the output specification of some *Lexical Rule* as defined in Meurers 2000, which we will call *Derivational Rule* throughout this thesis.

With the introduction of the notion of a lexical element, we can re-state the principles of compositionality and contextuality to cover the cases of irregular phrases and regular words as well. For the computation of meaning, we assume a version of compositionality as given above, but we speak of non-lexical elements instead of compound expressions. This leads to the following reformulation of the principle:

(9) The *Principle of Compositionality* (adapted version):
The meaning of a non-lexical element is a function of the meaning of its parts.[2]

As pointed out before, in HPSG, syntactic and semantic properties are simultaneously present in each sign. Thus, we cannot say that a word has different meanings, depending on the context. Instead, we must say that a word may or may not occur in a certain context. This perspective implies that it is not the context that restricts the interpretation of a word, but the word that restricts the contexts in which it may occur. This leads us to a re-formulation of the principle of contextuality.

(10) The *Principle of Contextuality* (adapted version):
A lexical element can impose idiosyncratic restrictions on the linguistic contexts in which it may occur.

In the re-formulation of the principle of contextuality in (10) we have also changed the domain of the principle from "words" to "lexical elements". Towards the end of Part II of this thesis, we will present evidence that irregular phrases impose contextual restrictions, just as is done by non-derived words.

The principles of compositionality and contextuality apply to different domains: The first is relevant to non-lexical elements, which are treated as fully predictable on the basis of their parts and the general rules of the grammar. The second applies to lexical elements. They have the freedom to have arbitrary syntactic and semantic properties and, furthermore, they can impose constraints on the linguistic contexts in which they occur. With this architecture of grammar, we enable HPSG to express the interaction of regularity and irregularity in a natural way. Yet, we preserve the original lexicalism of HPSG, as we locate the source of all idiosyncrasies and irregularities in the lexical elements.

[2]In the case of a derived word, such as the passive participle *loved* in sentence (8a), we assume that the base word, i.e., the active verb in this example, is the "part" of the non-lexical element.

In the first part of this thesis, we will focus on those aspects of HPSG that are concerned with regularity in combinatorial semantics, i.e., we will ignore irregular phrases and the contextual restrictions on lexical elements. It is necessary to address the treatment of regularity first, because it is only possible to identify what counts as "irregular" once we have a precise notion of what is understood to be "regular".

In addition, it is necessary to establish a clear understanding of the formal foundations of HPSG and of the way meaning is represented in HPSG. While the formal foundations of HPSG, as we adopt them in this work, have already been presented in Richter 2000, based on King 1999, the relation between the HPSG analysis of a sentence and its truth-conditional meaning has not yet been the topic of extensive research. In Part I we will discuss this issue in detail. As a result of our exploration we will show that it is unproblematic to assume standard semantic representations within an HPSG grammar and to interpret them in the usual way. This result is highly desirable, because it allows HPSG grammars to incorporate insights and analyses of formal semantics directly. Furthermore, without a clear notion of how the words and phrases that are licensed by an HPSG grammar can be interpreted semantically, the predictions of an HPSG grammar are hardly testable.[3]

The architecture for semantics that we will present is the following: we assume that every word or phrase in the language has a *logical form*, which is a representation of its meaning. For lexical elements, the logical form is given in the lexical entry; for non-lexical elements, the logical form is a function on the logical forms of their parts.

The logical forms used in this thesis are terms of a well-studied semantic representation language, Ty2 (Gallin, 1975). Using such terms as logical forms allows us to combine the tradition of syntactic Logical Forms as assumed in the Chomskyan paradigm (May, 1977, 1985; von Stechow, 1993), with the tradition of direct semantic interpretation (Montague, 1974a; Cooper, 1983; Lappin, 1991). As the logical form of a word or a phrase is a term, it is some syntactic representation that we can impose constraints on. In this sense, it is similar to the LF proposals. On the other hand, the logical forms assumed in this thesis are terms of a semantic representation language. Thus, they can be assigned a model-theoretic interpretation in the standard way. This makes it possible to express constraints that rely on the meaning of an expression directly.

Our semantic fragment is based on the semantic analysis in PTQ (Montague, 1974b). The semantic fragment is combined with a syntactic analysis which is based on the grammar in Pollard and Sag 1994. As both choices are relatively conservative within the respective fields, formal semantics and HPSG, we put the emphasis on the presentation of the integration of these two fragments, not on the motivation of the particular analysis. Furthermore, it should be clear that by combining these two fragments, we do not resolve any of their fundamental problems: In the case of the semantic fragment these include the lack of a treatment of dynamic effects (Kamp, 1981; Heim, 1982; Groenendijk and Stokhof, 1991) and potential conceptual inadequacies of the possible-world approach to intensionality (Lappin and Pollard, 1999). In the case of the syntactic fragment, we should mention at least the lack of worked-out theories of phonology (Höhle, 1999) and morphology (Reinhard, 2001), and word order (Reape, 1990, 1992, 1994; Kathol, 1995; Richter and Sailer, 1995; Richter, 1997; Penn, 1999a,b,c). It is nonetheless useful to start with these particular fragments, since they are relatively well-studied and, as the references indicate, proposals of how to overcome their shortcomings have been developed.

The fragment that results from the combination of the PTQ semantic analysis and the Pollard and Sag 1994 syntactic analysis is defined in the chapters of Part I. It introduces the

[3]In this case, HPSG runs the risk of falling back to the mere representationalism of early generative semantics such as Katz and Fodor 1964. There, the meaning of an expression was given as a collection of semantic features, but it was not indicated how the features can be interpreted.

necessary syntactic and semantic structures to enable us to give a formally precise account of the irregularity phenomena studied in Part II.

The structure of this study is the following. Besides the introduction and a conclusion, there are two main parts. As mentioned above, the first part is devoted to the definition of a fragment which accounts for the syntax and semantics of words and regular phrases. It consists of four chapters. In Chapter 1 we will give an overview of the first part of this thesis and introduce the semantic fragment used throughout the later chapters. The grammar formalism and our syntactic fragment are presented in Chapter 2. In Chapter 3, we show that it is possible to use the terms of a semantic representation language, Ty2, as parts of the objects in the denotation of an HPSG grammar. With this result, in Chapter 4, we extend the syntactic fragment of Chapter 2 with the semantic fragment of Chapter 1.

Part II addresses the treatment of idiomatic expressions within formal grammar in general and within the present fragment in particular. It starts with a short introduction (Chapter 5) to the connection between idiomatic expressions and the two aspects of irregularity that we are concerned with in this thesis, i.e., the irregularity of meaning assignment and the irregularity of distribution. In Chapter 6 we present the data that we will use to motivate our own analysis and to evaluate alternative proposals in Chapter 7. There we consider the analysis of idiomatic expressions found in *Generalized Phrase Structure Grammar* (Gazdar et al., 1985), in *Tree Adjoining Grammar* (Abeillé, 1995), and in a constructional approach to HPSG (Riehemann, 1997). In Chapter 8 we show how the fragment of Part I can be extended to handle the irregularity attested in the domain of idiomatic expressions. Chapter 9 contains a comparison between our own approach and those discussed in Chapter 7 and some remarks on applications of the Principle of Contextuality within the grammar of Pollard and Sag 1994.

We close this work with a summary in which we reconsider the architecture of grammar that emerges as a result of this study. We show that the new architecture for semantics developed in Part I has considerable advantages over that of Pollard and Sag 1994, and that the treatment of idiomatic expressions in Part II meets our requirement for a harmonious and integrated account of regularity and irregularity in natural language.

I

A Logical Form for HPSG

CHAPTER 1

CONTENT, Logical Form and Lexicalized Flexible Ty2

The present study is carried out in the framework of HPSG, starting with the version of the theory presented in Pollard and Sag 1994. HPSG is a sign-based theory in a broader sense, i.e., all properties of a linguistic sign are present in the sign itself: ranging from its phonology, its syntactic category, its potentially complex syntactic structure, to its meaning and its context of usage. This is witnessed by attributes which bear names such as PHONOLOGY, CATEGORY, DAUGHTERS, CONTENT and CONTEXT. Pollard and Sag 1994 is mainly concerned with syntax, leaving aside most other aspects of linguistic signs. In Chapter 8, however, we learn more about the authors' assumptions on semantics. Just as in Pollard and Sag 1987, Pollard and Sag 1994 assumes a version of *situation semantics* as the underlying semantic theory. The authors do admit, though that

> "[i]t will doubtlessly be a disappointment to some readers — and a relief to others — that we will not attempt here to formulate in precise terms the principles that relate our linguistic descriptions to any one version of situation semantic analysis [...] Our goal here will be rather to offer examples of how certain familiar analytical insights of a semantic nature can be integrated with the syntactic treatments of the earlier chapters."
> (Pollard and Sag, 1994, p. 318)

Our approach departs from this quote in two respects. First, we will characterize how the CONTENT value of a sign is related to the sign's meaning. Second, we do not assume a situation semantic background, but rather we adopt the semantic tradition of Montague. We agree, however, with Pollard and Sag 1994 in the assumption that whatever theory of semantics is being proposed for HPSG, it should clearly be possible to account for *certain familiar analytical insights of semantic nature* and be fully integrated with the other parts of the theory, i.e., syntax and, if present, phonology.

Due to the different choice of the underlying semantic framework, the analytical insights are necessarily different. The interaction of semantic and syntactic analysis will not play a major role in this work, but it should become obvious how pertinent constraints can be formulated. What will be central is the interaction between phrasal semantics and semantic specification in the lexicon, i.e., how the particular choice of lexical items restricts the possible interpretations of a sentence.

In this part of the thesis, we will examine how an HPSG grammar can be equipped with a standard semantic analysis. This apparently simple program does, however, require some explanation. First, we will have to show what kind of semantic treatment we envision. We will adopt a Montague-style semantic analysis, but we will use the two-sorted extensional language Ty2 (Gallin, 1975) as semantic representation language, instead of Montague's *Intensional Logic*. In addition, we will assume an adaptation of the *flexible type shifting* system of Hendriks 1993 to account for scope ambiguities. The resulting semantic system will be presented in Section 1.3. Second, we will indicate what we understand under an HPSG grammar. We assume the formal foundations of HPSG as presented in Richter 2000, which are based on the work of King 1989, 1994, 1999 and allow us to formalize the grammar presented in Pollard and Sag 1994. In Section 1.1 we will address some basic properties of what we assume to be an HPSG grammar and come back to this in more detail in Chapter 2. Finally, we must provide the synthesis of these two, i.e., integrate the semantic analysis explicitly into the HPSG grammar. This will be the issue of Chapter 3, where we will present an HPSG encoding of the semantic representation language Ty2, and of Chapter 4, where we will integrate the flexible semantic system of Section 1.3 into an HPSG grammar for some fragment of English.

In the sections of this introductory chapter, we will give some conceptual motivation for the semantic treatment conceived in this thesis. In Section 1.1 we will give a simple HPSG grammar and indicate some of the formal assumptions about HPSG adopted in this thesis. In Section 1.2 we will locate the approach that is going to be taken in this thesis with respect to assumptions about logical form and semantic interpretation made within the generative tradition. Finally, in Section 1.3 we will present the system of *Lexicalized Flexible Ty2* which is a combination of the efforts of Gallin 1975, Groenendijk and Stokhof 1982 and Zimmermann 1989 to provide a technically simpler semantic representation language than Montague's *Intensional Logic* on the one side, and of the attempt of Hendriks 1993 to simplify the way scope ambiguities are handled within Montague Grammar on the other.

1.1. What Is an HPSG Grammar?

We must distinguish two possible interpretations of this question. First, we can characterize HPSG as a formal system. In this sense, an HPSG grammar is whatever grammar is written within this formal system. Second, we can consider a particular linguistic analysis which is expressed as a grammar in this formal system. In this thesis, we address both issues. In the present section and in Section 2.1, we will examine HPSG as a formal framework for writing grammars. There, we will choose an example grammar to illustrate the definitions and properties of the formalism which is only loosely connected to a linguistic analysis. In Section 2.3, we will consider a concrete syntactic analysis for a fragment of English, which is a mildly modified version of the analysis in Pollard and Sag 1994.

For our purpose here it is important to provide a general description of both aspects of HPSG, i.e., the formalism of HPSG and the concrete linguistic analysis. The first is needed because, in Chapter 3, we will use the HPSG formalism, to write the "grammar" of the semantic representation language Ty2. We can show that this grammar is an adequate encoding of the semantic representation language. The understanding of a particular syntactic analysis for English is also necessary, because in Chapter 4, we will enrich such a syntactic analysis that with a semantic analysis based on the representation language of Chapter 3. In Part II, the syntactic analysis will be augmented to allow for an account of idiomatic expressions. In the present section, we will focus on the HPSG formalism and give a first impression of the concepts that underlie it.

Our understanding of what the formalism of HPSG is, is primarily based on the assumptions made in King 1999, which provides a formal characterization of what it means for an HPSG grammar, based on the paradigm of Pollard and Sag 1994, to be *true* of a natural language. According to King, the idea of an HPSG grammar is to *describe* a particular language, say English. In fact, for an HPSG grammar to be an adequate description of English, the English language must be a *model* of that grammar. There are many conceptual and formal details related to this idea, some of which we will address in Chapter 2. There we will present the formal language that we will use to throughout this thesis.[1] The

[1]The position of King 1999 differs in some respect from that taken in Pollard and Sag 1994 and Pollard 1999. While these differences are not crucial for our purpose, the terminology that we are using in this thesis is follows that of King rather than that of the other papers. For a detailed consideration of these issues, see Richter 2000.

formal language is called *Relational Speciate Re-entrant Language* (RSRL), which, as the name suggests, is an extension of Paul King's *Speciate Re-entrant Logic* (SRL) (King, 1989, 1999). RSRL extends the set of possible descriptions by providing relations and quantification as part of the description language. The underlying model-theoretical assumptions are basically the same. In this section, we will give a simple grammar which will serve to illustrate the basic properties of RSRL. In Chapter 2 we will expand this grammar to make it similar to the grammar given in Pollard and Sag 1994.

An HPSG grammar consists of two parts: a *signature* and a *theory*. In the signature, the basic ontological assumptions about (the) language are expressed. In the theory, more specific properties of the language are formulated.

Let us briefly turn to the signature. One of the basic ontological assumptions, made in this thesis is that linguistic signs have a phonology and syntactic and semantic properties. In addition, we assume that a linguistic sign is either a word or phrase. Moreover, phrases have daughters. These assumptions are collected in the signature. We will usually express these assumptions in graphs such as the following. The tree in (11) expresses what we call technically the *sort hierarchy* and the *appropriateness conditions* below the sort *sign*.

(11) The sort hierarchy and appropriateness conditions below *sign*:

sign
PHONOLOGY *list*
SYNSEM *synsem*

word *phrase*
DAUGHTERS *constituent-structure*

In an HPSG perspective, we are interested in *interpretations* of such signatures. An empirically adequate grammar of a natural language will be such, that the natural language is, *inter alia*, an interpretation of the signature of the grammar. An interpretation contains a set of linguistic objects, which is usually called the (linguistic) *universe*. Each of these objects is assigned a sort from the signature.

Let us consider the verb *walks* as a linguistic object with the label *word*. This linguistic object is related via an attribute to another linguistic object which is the phonology of the word. In this thesis, we are not concerned with phonology. We will, therefore, assume an orthographic representation instead of a real phonological representation. In the case of the word *walks*, we take its PHON value to be a list which contains the string *walks*.[2]

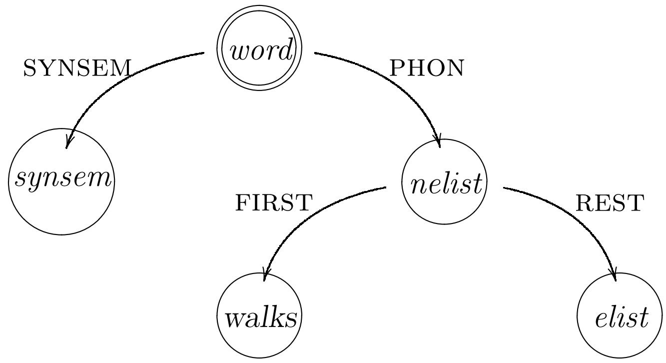
In the sort hierarchy given in (11), it can be seen that the sort *word* appears below the sort *sign*. This means that the sort *word* is a *subsort* of the sort *sign*. Parallel to this, we assume that the sort *list* has two subsorts, *empty-list* (*elist*) and *non-empty-list* (*nelist*). In an interpretation of a signature, the objects in the universe are always assigned sorts that do not have subsorts, i.e., there will be objects that are assigned the sort *word*, but there are no objects to which the sorts *sign* or *list* are assigned.

For the sort *sign*, we have declared two *attributes* as being appropriate: PHONOLOGY and SYNSEM. This declaration has its reflex in the interpretations of the signature. In the case of (11) this means that for every sign, there is some object with the label *elist* or *nelist* which is related to this sign by the attribute interpretation function for the attribute PHONOLOGY. In our simple example of the word *walks*, this list would be $\langle$walks$\rangle$.

The part of the signature depicted in (11) also introduces a second subsort of *sign*: the sort *phrase*. An example of a phrase would be the sentence *Mary walks*. By virtue of being a subsort of *sign*, every *phrase* must have a phonology and some syntactic and semantic

[2]Höhle 1999 considers in detail what can and should be the value of the PHONOLOGY attribute within the formal framework assumed in this thesis.

FIGURE 1.1. The word *walks*:



properties. In addition, the attribute DAUGHTERS is declared appropriate for the sort *phrase*. This means that every phrase is related to some object of the sort *constituent-structure* by the attribute interpretation function for the attribute DAUGHTERS.

In this thesis, we assume that the declarations made in the signature lead to a total specification of the shape of the linguistic objects in an interpretation. This means that all linguistic objects have a label of a maximally specific sort, i.e., there are no objects with label *sign* in the denotation of the grammar, but objects with the labels *word* or *phrase* may exist. In addition, the attribute interpretation function is such that if an attribute is appropriate for a given sort, then all objects that are of this sort or of a subsort thereof, are related to some other linguistic object by the attribute interpretation function for this attribute. In more intuitive terms, this means that declaring the attribute PHONOLOGY appropriate for the sort *sign* has the consequence that every *word* object and every *phrase* object is related to some list by the attribute interpretation function for PHONOLOGY. On the other hand, the fact that the attribute DAUGHTERS is appropriate for the sort *phrase* in (11), but not for the sort *sign* or *word* means that the attribute interpretation function for DAUGHTERS is undefined on every object of sort *word*.

In Figure 1.1 we see parts of an interpretation of our signature. Each bubble in the figure depicts an object, the label inside the bubble states its sort. The arrows show which attributes map which object to which object. In the figure, we have marked the object of sort *word* with a double-lined bubble, because every other bubble can be reached via an arrow from this bubble. In addition, the figure contains all objects that can be reached by an attribute from the double-lined bubble. We also speak of parts of an interpretation that have these two properties as a *configuration of objects under an object*. In our case, the double-lined bubble is used for the object o, such that the figure shows a configuration of objects under o. For ease of reference, we will often call this object o the *matrix object*. All objects in the configuration are *components* of the matrix object.

It should also be noted, that we use an encoding of lists such as the one proposed in Pollard and Sag 1994. As mentioned above, we assume that the *list* has two subsorts, *elist* and *nelist*. For the latter sort, the attributes FIRST and REST are appropriate. This can also be seen in Figure 1.1. There, the matrix object is connected with a *nelist* object via the attribute PHON. From this *nelist* object, there are two arrows departing: one with the name FIRST which leads to an object labelled *walks*, and one with the name REST which leads to an object labelled *elist*.

In graphs as the one in (11) we indicate (parts of) the sort hierarchy and the appropriateness function. In addition to these, we assume that a signature also provides names for relations and an arity function that indicates the arity for the relation symbols. Usual relation symbols used in HPSG are for example "`append`" with arity three, or "`member`" with arity two. As the interpretations of the signature must reflect all the properties of the signature, they also contain the interpretation of the relation symbols. For each relation symbol ρ with arity i, the interpretation contains a set of i-tuples that are built from the objects in the universe.[3]

As interpretations contain the interpretation of the relation symbols, and as configurations are parts of an interpretation, they must contain parts of the interpretation of the relation symbols as well. In fact, we will assume that every configuration contains all the n-tuples that are in the interpretation of a relation symbol and that are built only from components of the matrix object. When we give configurations in figures such as Figure 1.1, we will not, however, indicate those aspects that relate to relations.

Given the symbols for sorts, attributes and relations as provided by the signature, we can write *descriptions*. Descriptions are normally given in the form of an AVM (*Attribute Value Matrix*). In (12) we give a description of the word *walks*. In later AVMs, we will use the usual list notation, i.e., we write ⟨*walks*⟩ as the description of the PHON value in (12).

(12) $$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \begin{bmatrix} \text{FIRST} & \textit{walks} \\ \text{REST} & \textit{elist} \end{bmatrix} \end{bmatrix}$$

This AVM describes every object o that is of sort *word* and has the attribute PHON defined on o such that the attribute interpretation function maps this object to some other object o' such that the attribute FIRST is defined on o' and maps o' to an object labelled *walks*. Furthermore, the attribute REST is defined on o' and maps it to an object labelled *elist*. The matrix object in Figure 1.1 is described by this description. This shows that if we want to verify whether an object is described by a given description, we must consider the configuration under this object in the given interpretation.

In contrast to configurations, which are always total, descriptions of objects may contain supersorts and need not mention all attributes defined on the objects it describes. Given the sort hierarchy in (11), we know that the attribute interpretation function for SYNSEM is defined on every word. Therefore, we know that it is also defined on every element described by the AVM in (12).

Descriptions can be a lot more complex than the AVM given in (12). In (13) we give a description of the sentence *Every man walks*. In this description, we use the attributes H-DTR and N-DTR which we have not yet introduced. We assume that these attributes are defined on the sort *constituent-structure* and that the sort *sign* is appropriate for them.

(13) $$\begin{bmatrix} \textit{phrase} & \\ \text{PHON} & \langle \textit{every, man, walks} \rangle \\ \text{DTRS} & \begin{bmatrix} \text{H-DTR} & \begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{walks} \rangle \end{bmatrix} \\ \text{N-DTR} & \begin{bmatrix} \textit{phrase} & \\ \text{PHON} & \langle \textit{every, man} \rangle \\ \text{DTRS} & \begin{bmatrix} \text{H-DTR} & \begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{man} \rangle \end{bmatrix} \\ \text{N-DTR} & \begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{every} \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

[3]As will be made precise in Chapter 2, these are i-tuples of objects of the universe or sequences of objects of the universe. In the present section, however, it is enough to assume that they are i-tuples of objects of the universe.

Figure 1.2. The phrase *every man*:

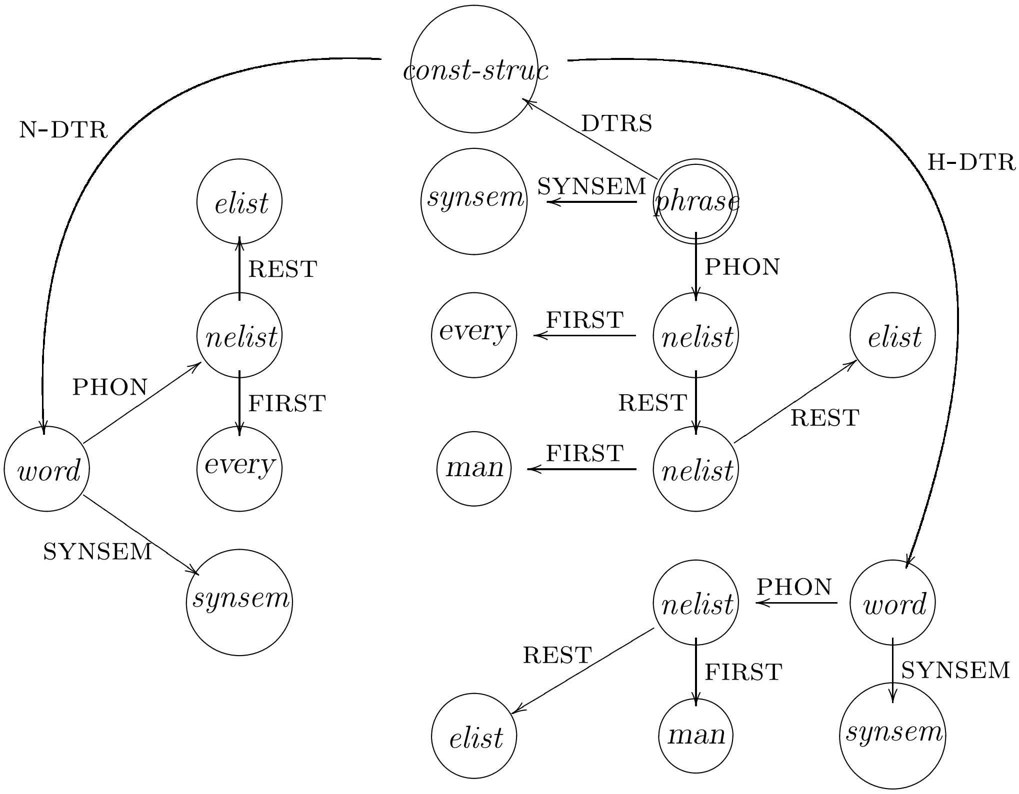


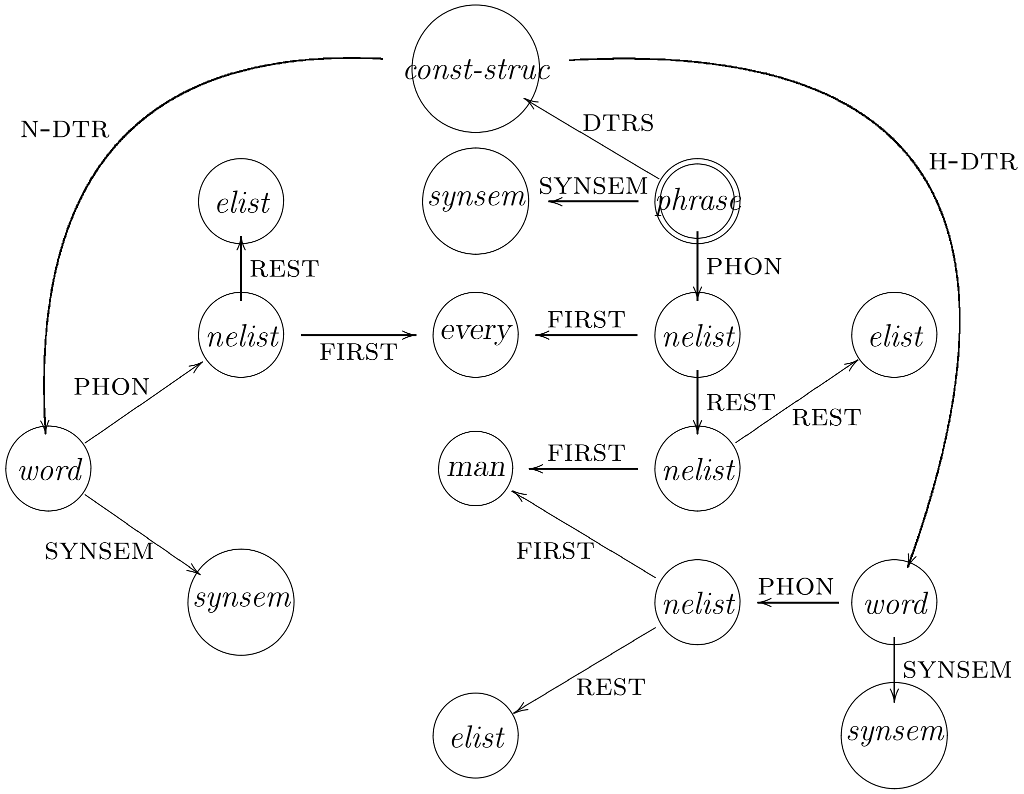
Figure 1.2 shows a configuration of objects under a *phrase* object. The matrix object is described by the description of the nonhead daughter in the AVM in (13).

So far, we have only seen AVMs containing sort names and attribute names. In addition to this, we will also use *variables*. These variables are usually called *tags* in HPSG and are written as boxed integers ($\boxed{1}$, $\boxed{2}$, ...). Alternatively, we will sometimes use lower case letters (a, b, $\ldots x$, y, z) for variables. Tags are used in HPSG to indicate *identities*. We can modify the description of the sentence *Every man walks* in (13) so that we only describe objects which are the matrix objects of configurations that do not contain more than one object with label *every*, *man* or *walks*. Such a description is given in (14).

(14)
$$\begin{bmatrix} \textit{phrase} & \\ \text{PHON} & \langle \boxed{1}, \boxed{2}, \boxed{3} \rangle \\ \text{DTRS} & \begin{bmatrix} \text{H-DTR} & \begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \boxed{3}\ \textit{walks} \rangle \end{bmatrix} \\ \text{N-DTR} & \begin{bmatrix} \textit{phrase} & \\ \text{PHON} & \langle \boxed{1}, \boxed{2} \rangle \\ \text{DTRS} & \begin{bmatrix} \text{H-DTR} & \begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \boxed{2}\ \textit{man} \rangle \end{bmatrix} \\ \text{N-DTR} & \begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \boxed{1}\ \textit{every} \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

We can verify that the description of the nonhead daughter in (14) does not describe the matrix object in Figure 1.2 (page 16): There, starting from the matrix object, the paths PHON FIRST and DTRS N-DTR PHON FIRST lead to different objects. Consider, however, the configuration in Figure 1.3 (page 17). The matrix object of this configuration satisfies the

FIGURE 1.3. The phrase *every man* (as a slightly different configuration):



description of the nonhead daughter in the AVM in (13), just as was the case of the matrix object of the configuration in Figure 1.2. In contrast to the first configuration, however, the matrix object of the second configuration also satisfiesthe description of the nonhead daughter given in (14). As can be seen, the configuration in Figure 1.3 contains only one object with label *every* and only one object with label *man*. For each of these objects, there are two arrows pointing to it: Starting from the matrix object of the configuration, the paths PHON FIRST and DTRS N-DTR PHON FIRST both lead to the single *every* object; and the paths PHON REST FIRST and DTRS H-DTR PHON FIRST both lead to the single *man* object in the configuration.

The use of the tags in AVMs indicates that the AVM describes an object o if the paths that lead to the same tag in the AVM, denote the same linguistic object when applied to o. As we treat tags formally as variables, we say that there is a variable assignment function such that the paths that lead to some variable $\boxed{i}$, map the described object into the linguistic object denoted by applying the variable assignment to $\boxed{i}$.

Under this interpretation of tags, the AVM in (14) contains three free variables, $\boxed{1}$, $\boxed{2}$, and $\boxed{3}$. In order to bind variables, we can use *quantifiers*. Our description language provides two quantifiers, an existential quantifier, written as **E**, and a universal quantifier, written as **A**. In RSRL, we use a restricted quantification, in the sense that a quantifier does not range over the entire universe. Instead its range is determined by the objects in the configuration under the described object.[4] In (15) we quantify existentially over the variables used in (14).

(15) $\mathbf{E}\boxed{1}\mathbf{E}\boxed{2}\mathbf{E}\boxed{3}$

$$\begin{bmatrix} \textit{phrase} & \\ \text{PHON} & \langle \boxed{1}, \boxed{2}, \boxed{3} \rangle \\ \text{DTRS} & \begin{bmatrix} \text{H-DTR} & \begin{bmatrix}\textit{word} & \\ \text{PHON} & \langle \boxed{3}\ \textit{walks}\rangle\end{bmatrix} \\ \text{N-DTR} & \begin{bmatrix} \textit{phrase} & \\ \text{PHON} & \langle\boxed{1},\boxed{2}\rangle \\ \text{DTRS} & \begin{bmatrix} \text{H-DTR} & \begin{bmatrix}\textit{word}&\\ \text{PHON} & \langle\boxed{2}\ \textit{man}\rangle\end{bmatrix} \\ \text{N-DTR} & \begin{bmatrix}\textit{word}&\\ \text{PHON}&\langle\boxed{1}\ \textit{every}\rangle\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

This description describes an object exactly if this object has three components o_1, o_2 and o_3 such that interpreting the variables $\boxed{1}$, $\boxed{2}$ and $\boxed{3}$ as these components, the object is described by the rest of the description. In our example configuration in Figure 1.3, the variable $\boxed{1}$ must be interpreted as the *every* object, variable $\boxed{2}$ as the *man* object. In the case of the earlier configuration in Figure 1.2, no such variable assignment is possible, because the elements on the PHON list of the nonhead daughter are distinct from those on the PHON lists of the words.

As said before, the signature also contains names for relations such as `append` and `member`. We can give a description of the sentence *Every man walks* that contains relations. The PHON value of the phrases in the sentence can be characterized as the concatenations of the PHON values of their daughters. The following description makes this explicit.

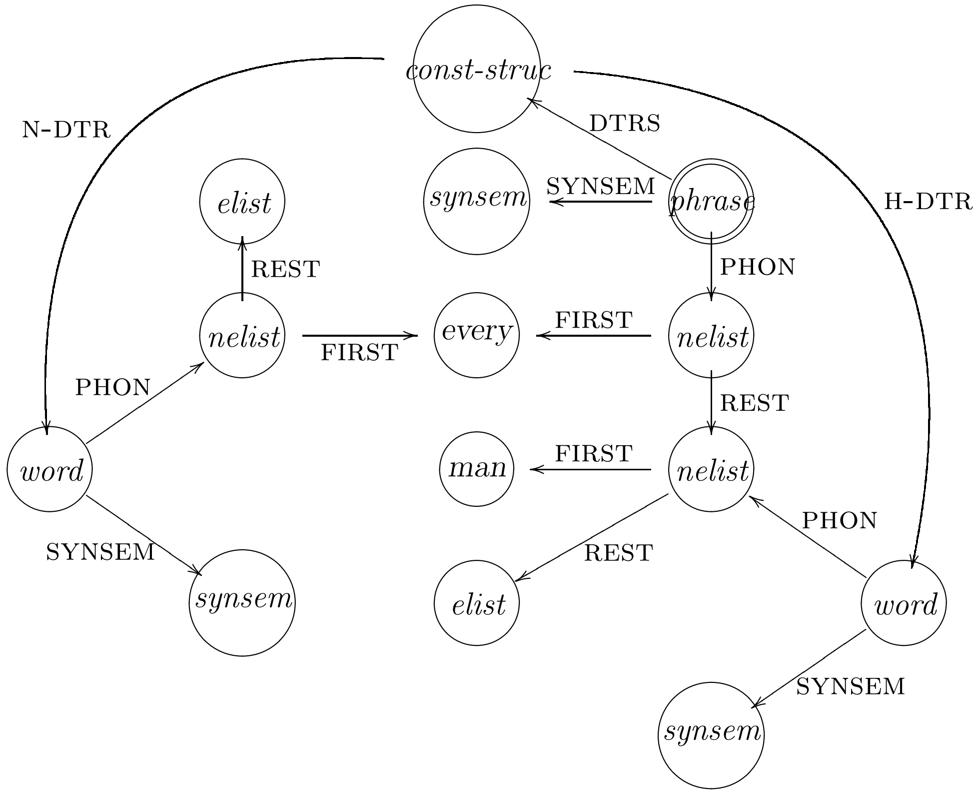
(16) $\mathbf{E}\boxed{1}\mathbf{E}\boxed{2}\mathbf{E}\boxed{3}\mathbf{E}\boxed{4}\mathbf{E}\boxed{5}$

$$\begin{bmatrix} \textit{phrase} & \\ \text{PHON} & \boxed{5} \\ \text{DTRS} & \begin{bmatrix} \text{H-DTR} & \begin{bmatrix}\textit{word} & \\ \text{PHON} & \boxed{3}\ \langle \textit{walks}\rangle\end{bmatrix} \\ \text{N-DTR} & \begin{bmatrix} \textit{phrase} & \\ \text{PHON} & \boxed{4} \\ \text{DTRS} & \begin{bmatrix} \text{H-DTR} & \begin{bmatrix}\textit{word}&\\ \text{PHON} & \boxed{2}\ \langle\textit{man}\rangle\end{bmatrix} \\ \text{N-DTR} & \begin{bmatrix}\textit{word}&\\ \text{PHON}&\boxed{1}\ \langle\textit{every}\rangle\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix} \begin{array}{ll} \textbf{and} & \texttt{append}(\boxed{1},\boxed{2},\boxed{4}) \\ \textbf{and} & \texttt{append}(\boxed{4},\boxed{3},\boxed{5}) \end{array}$$

The description in (16) contains the logical connective **and**, which expresses conjunction. In RSRL we will use more connectives such as **or** for disjunction, **not** for negation, ⇒ for implication. The description in (16) describes an object if and only if (*iff*) there are five components of this object, referred to by the variables $\boxed{1}$, $\boxed{2}$, $\boxed{3}$, $\boxed{4}$, and $\boxed{5}$, such that the object is described by the description in the big square brackets and, in addition, the objects referred to by the variables $\boxed{1}$, $\boxed{2}$ and $\boxed{4}$ are in the `append` relation, and the objects referred to by the variables $\boxed{4}$, $\boxed{3}$ and $\boxed{5}$ are also in the `append` relation. In (17) we give an informal characterization of the intended interpretation of the relation `append`. Such a characterization means that we only want to consider interpretations of the signature in which the triples that are in the relation `append` have the indicated properties.

[4] In Chapter 2 we will make this range more precise. We will define it in such a way that it contains the components of the described object and all finite sequences thereof. For the purpose of the intuitive characterization in this section, it is enough to consider the set of components as the range of the quantifiers.

FIGURE 1.4. The phrase *every man* (as yet another configuration):



(17) The relation `append` has in its interpretation all triples of lists l_1, l_2, and l_3 such that
  (i) if l_1 is an empty list, then l_2 and l_3 are identical, and
  (ii) else,
    the first element of l_1 is also the first element of l_3, and
    the rest list of l_1, the list l_2 and the rest list of l_3 are in the interpretation of the relation `append`.

Given this specification of the relation `append`, the matrix object of the configuration in Figure 1.3 is not described by the description of the nonhead daughter together with the relation call `append`(1,2,4) in (16). In this description, the relation `append` is required to hold between the three lists that are the PHON value of the word *every*, the PHON value of the word *man* and the PHON value of the phrase *every man*. Given our specification of the relation `append`, for this to be the case, the first element of the PHON value of the word *every* must be the first element of the PHON value of the phrase *every man*. So far, everything is unproblematic, but, the remaining PHON value of the word *every*, i.e., the relevant *elist* object in the configuration, must stand in the relation `append` with the PHON value of the word *man* and the PHON REST value of the matrix object of the configuration. As the first of these three lists is empty, our specification of the relation `append` requires the other two lists to be identical. It can be verified in the configuration in Figure 1.3 that the paths PHON REST and DTRS H-DTR PHON lead to different objects. Therefore, the matrix object of this configuration does not satisfy the description of the nonhead daughter and the first call of the relation `append` in (16).

In Figure 1.4 (page 19) we give a third configuration for a phrase with phonology *every man.* This configuration differs slightly from the one given in Figure 1.3. In the new configuration, the paths PHON REST and DTRS H-DTR PHON actually lead to the same object. As for the rest, the two configurations are alike, the matrix object of the configuration in Figure 1.4 is described by the description of the nonhead daughter and the first use of the relation `append` in (16).

With the description in (16) we have given examples of all parts of the description language used in this thesis. In Chapter 2, the formal definitions will be given. It should be noted, though, that the signature not only specifies the shape of the linguistic objects, it also provides the non-logical symbols (i.e., the names of sorts, attributes, and relations) that we can use to write descriptions.

As we have seen, objects in an interpretation can be described by some description, or can fail to be described by a certain description. There is one particularly interesting case of objects: there may be objects o and o' in an interpretation such that they are described by exactly the same descriptions. It has been shown (King, 1999) (pp. 342ff.) that this is the case exactly if the configurations under the objects are of the same shape, i.e., they have the same sort label, the same paths are defined on them and lead, again, to objects with the same sort label. In addition, whenever two paths have identical values in the object o, these two paths also have identical values in the object o'.[5] Such objects (respectively the configurations under them) are called *indiscernible* as they cannot be told apart by any description. Alternatively, such objects are called *congruent*, as the configurations under them are of exactly the same shape. When we talk in an intuitive way about indiscernible/congruent (configurations of) objects, we also say that they are *copies* of each other.

For illustration, consider the configuration under the word *walks* as given in Figure 1.1 (page 14). The way we have set up the signature, all *word* objects with the phonology ⟨*walks*⟩ are indiscernible/congruent. Independent of this, we can easily assume an interpretation of the signature which contains several such configurations of objects.

In the case of phrases, it is a different matter. Here we have seen that the three configurations that we have given for the phrase *every man* in Figures 1.2, 1.3, and 1.4 are the same with respect to the paths that are defined on them and with respect to the labels of the objects reached by these paths. The three configurations are, however, different with respect to those paths that lead to the same objects, and with respect to the triples that stand in the relation `append`. This configurational difference is reflected by the fact that the matrix objects of these three configurations behaved differently with respect to the descriptions in (13), (14), and (16). It was only the third configuration whose matrix object met the description of the nonhead daughter in these cases, and it was only the first configuration that failed to meet the description of the nonhead daughter in the AVM in (14).

Now that we have seen what a description is, we can address the second part of an RSRL grammar, the *theory.* The theory is a set of descriptions which only use symbols provided by the signature, i.e., the symbols introduced there for sorts and relation names, and some "built-in" symbols, such as logical connectors, quantifiers and variables. Just as the signature is meant to specify the general shape of every possible linguistic object, the theory is meant to describe every object of the language. In (18) we give an example of a description that could be part of a theory of natural language.

---

[5]As configurations in RSRL also contain relations, we must require that the correspondence also extends to the relational part of the configurations. This extended notion of congruence is defined in Richter 2000 (p 184). In Section 3.2 we will address it in more detail.

$$\text{(18)}\quad \textit{phrase} \Rightarrow \left(\begin{array}{l} \mathbf{E}\boxed{1}\ \mathbf{E}\boxed{2}\ \mathbf{E}\boxed{3} \\ \begin{bmatrix} \text{PHON} & \boxed{3} \\ \text{DTRS} & \begin{bmatrix}\text{H-DTR PHON} & \boxed{1} \\ \text{N-DTR PHON} & \boxed{2}\end{bmatrix}\end{bmatrix} \\ \textbf{and} \begin{pmatrix} \texttt{append}(\boxed{1},\boxed{2},\boxed{3}) \\ \textbf{or}\ \texttt{append}(\boxed{2},\boxed{1},\boxed{3})\end{pmatrix}\end{array}\right)$$

This description is met by every linguistic object that is either not a *phrase* at all, or else, its PHON value ($\boxed{3}$) is required to be the concatenation of the PHON values of its daughters ($\boxed{1}$ and $\boxed{2}$). Clearly any phrase described by (16) satisfies this description, and all its components do so as well. In particular, the matrix object of the configuration in Figure 1.4 is described by (18), but, as we have seen, the matrix objects of the configurations in Figures 1.2 and 1.3 are not. As the matrix object of the configuration in Figure 1.1 is a *word*, it trivially satisfies the description in (18).

In descriptions which are part of the theory, we collect our specific generalizations about the language. In order to relate our grammar to the language we need to interpret the grammar with respect to an interpretation of its signature. To do this, we require that an interpretation of the signature is a *model* of the grammar, if and only if every object is described by every description in the theory.

Turning again to the concrete objects that we have considered so far, we can illustrate the kind of model that will be used in this thesis. All the objects that occurred in the configurations shown in Figures 1.1–1.4 are built according to the signature. Therefore, we can assume that there is an interpretation of the signature which contains exactly those configurations. However, if we assume that description (18) is part of the theory, then this interpretation is not a model of the grammar, because the matrix objects of the configurations in Figures 1.2 and 1.3 fail to satisfy this description.

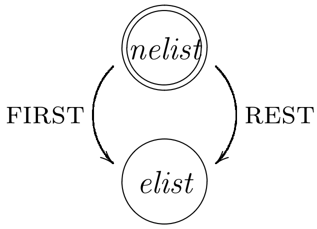
We can, of course, consider the interpretation that contains only the configurations in Figures 1.1 and 1.4. In this case, every object in the universe of the interpretation is described by the description in (18). This interpretation is, then, a model of our grammar. Let us call this particular model M.

As a further condition, we are only interested in certain kinds of models, called *exhaustive models*. Exhaustive models are maximal, i.e., they contain at least one copy of each configuration which can appear in a model of the grammar. Intuitively this means that, if we have an exhaustive model of the grammar, then we are sure that we have collected instances of all configurations that are licensed by the grammar. To consider a simple example, the adequacy of a grammar for a given language should depend, *inter alia*, on the question whether there is a configuration which has as its matrix object a *word* object with the phonology ⟨*walks*⟩ in the model of the grammar or not. However, it is independent of the question how many copies of such a configuration there are.[6]

As we have seen, M as characterized above is a model of our grammar, but it is not an exhaustive model. To see this, we add the configuration in Figure 1.5 to the model M to arrive at some model M'. The configuration in Figure 1.5 clearly adheres to the signature: all its objects are labelled with maximally specific sorts, *nelist* and *elist*. To the sort *nelist*, two attributes are appropriate, which are both present in the configuration. The values of these attributes are objects whose sorts follow the signature. Furthermore, all the objects in the configuration trivially satisfy the description (18) in the theory, as they are not phrases. Finally, we have implicitly added those triples of objects to the configuration that occur in the configuration and stand in the relation `append` according to our intuitive characterization. Thus, M' is a model of the grammar, as is M. The new configuration

[6]As elaborated in Richter 2000, this is the position taken in King 1999. In contrast to this, in Pollard 1999 copies are eliminated from the denotation of a grammar.

FIGURE 1.5. One more configuration:



that we have added to form M' is, however, not congruent with any of the configurations already present in M. Therefore, M cannot be an exhaustive model of our grammar.

To take a more linguistic example, the model M does not contain a configuration whose matrix object is of sort *phrase* and whose phonology is the list ⟨*every, man, walks*⟩. Such a configuration would clearly be compatible with the grammar. Therefore, it will occur in an exhaustive model of our grammar.

Using the notion of an exhaustive model, we can show that the small grammar of this section is not a grammar of English. Assume that I is an exhaustive model of the grammar. If our grammar is a grammar of English, then the English language would be an exhaustive model of the grammar as well. But this means that for each configuration in I, we must be able to find a congruent configuration in English, and for each configuration in English, there must be a congruent configuration in I. We can show that the first of these requirements is not satisfied:

In addition to configurations whose matrix objects are phrases with the phonology *every man* or *every man walks*, any exhaustive model of the grammar will contain configurations whose matrix objects are phrases with phonologies such as *man every*, *walks walks walks*, etc, or any other permutation of the phonologies assumed for words. Clearly, there are no configurations in English that are congruent to such configurations. We say that our little grammar of this section *overlicenses*.

If the second requirement is not met, we can speak of *underlicensing*. In that case, English contains configurations for which we cannot find congruent configurations in I. To construct an example for underlicensing, let us add a constraint to the theory that requires that the phonology of every phrase contains no more than two elements. English contains a configuration that corresponds to the sentence *every man walks*, but this configuration has a phonology with more than two elements. If we now consider an exhaustive model I of this new grammar, then we cannot find such a configuration in I.

In this section, we have given a first characterization of our view of an HPSG grammar. In the next chapter, we will provide the underlying definitions that we have glossed over in this section. In the final section of Chapter 2 we will also indicate the syntactic analyses that we base our semantic specification upon.

1.2. CONTENT AND LF

The central point of Part I of this thesis is to provide the necessary architecture to furnish syntactic analyses made within HPSG with a model-theoretic semantic analysis. In Pollard and Sag 1994, every linguistic sign is built in such a way that a path SYNSEM LOCAL CONTENT is defined on it. Thus, for every *sign* object, there is another object that the sign is related to by the interpretation of this path. Put differently, every sign has a component which specifies its content. What then, is this component? In Pollard and Sag 1987, we are given the example of the linguistic sign *cookie*. Its content, or the *signifié*, is characterized as either "a semantic representation, the psychological concept of cookiehood" (p. 2) or the "thing described: the property of being a cookie" (p. 5). These different interpretations are labeled *conceptualist* vs. *realist* respectively. Later in the text, the authors state:[7]

> "In the interest of practicality, though, we will simply ignore such questions, and leave it to the reader to interpret the symbolism of SEMANTICS values in a way that is consistent with his or her philosophical scruples. As a matter of expository convenience, we will usually talk about meaning in a realistic way; but the reader is urged to adopt a healthy scepticism and remain alert to the possibility of linguistic evidence which might help to resolve the conflict between realism and conceptualism one way or the other." (Pollard and Sag, 1987, p. 82)

This quote indicates that the authors do not give a clear indication as to how we should interpret the semantic part of a sign. They suggest, however, that as far as they can see, both perspectives should be compatible with their analysis.

In Pollard and Sag 1994, the authors are even more vague about the role of CONTENT:

> "CONTENT, on the other hand, is concerned principally with linguistic information that bears directly on semantic interpretation (and is therefore most closely analogous to GB's level of LF (logical form))." (Pollard and Sag, 1994, p. 3)

GB's level of LF is a syntactic level of representation, therefore, by definition, it can only be a description of a situation or a concept and not be this concept itself. Towards the end of this section, we will come back to this question, considering an example.

In order to be able to locate our own approach, it is first worth looking at several alternatives that have been proposed within the GB tradition. We will confine ourselves to three major alternatives. First, there is the mainly syntactically oriented conception of LF as developed in May 1985, 1989, 1991. Second, von Stechow 1993 proposes a view of LF which is called *transparent Logical Form* (TLF). This TLF is closer to semantics than May's LF. Finally, Lappin 1991 proposes an approach which makes LF, as a separate level of representation, obsolete. Instead, he proposes to interpret S-structures directly. We will summarize these three approaches and give a rough characterization of the similarities and differences of our own approach in contrast to these.

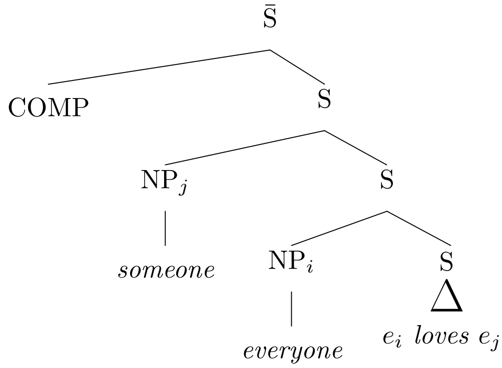
For May 1991 (p. 334), the syntactic level of LF is distinct from a semantic *logical form* (lf). The latter is "the representation of the *form of the logical terms* of the language". This lf is considered a syntactic level, but it represents the syntax of logical terms. May assumes (p. 347) that the interpretation of, at least, the logical connectives and quantifiers is universal. In contrast to this, LF is characterized as a component of the grammar of natural languages.[8] As LF is conceived as part of the derivation of a sentence — just like its D-structure, S-structure and Phonological Form (PF) — LF has the form of a syntactic tree which has words at its leaves, syntactic category labels at its nodes, etc. It is, then, at the syntax-semantics interface that, for syntax, LF looks like a tree, but for semantics, the LF is seen as the representation of a (set of) lf(s) which can, then, be interpreted.

[7]Note that in Pollard and Sag 1987 semantic information of a sign is indicated as the value of the attribute SEMANTICS, not CONTENT as in Pollard and Sag 1994.

[8]Confusingly enough, though, May 1991 (p. 334) says that

> "the grammars of natural languages have, as one of their components, a level of LF, which represents the form of the logical terms"

FIGURE 1.6. The Logical Form of the sentence (19a) according to May 1989:



In order to substantiate his claim for the existence of LF, May shows that LF complies principles which hold for other syntactic levels as well. Among these, there are the ECP, the Theta Criterion and Binding Theory. In addition to these general principles, he argues that LF differs from the other levels in that some movement operations are triggered by semantic considerations. The prototypical type of movement at LF is *Quantifier Raising* (QR). May 1991 shows that QR is necessary to determine the scope relations in a sentence, i.e., to arrive eventually at the right lf. On the other hand, QR is restricted by Binding Theory, just as movement from D-structure to S-structure.

As a concrete example, consider the following sentence from May 1989. According to May, sentence (19a) has two readings, which are given in (19b) and (19c). In Figure 1.6, we indicate the Logical Form for the sentence in (19a).

(19) a. Everyone loves someone.
b. $\forall x \exists y \mathsf{love}'(x, y)$
c. $\exists y \forall x \mathsf{love}'(x, y)$

In May's view of Logical Form as a syntactic level of representation, both readings of the sentence have the same Logical Form. May's motivation for having a single LF for two distinct readings is twofold: syntactic and semantic.

Let us first address the syntactic argument. May wants the principles of grammar that hold for other levels of representation to be valid at the level of LF as well. In a LF in which the NP *everyone* is higher in the tree than the NP *someone* at LF, the paths that result from quantifier raising would cross each other. Such a crossing is, however, excluded for *wh*-phrases in English. This is illustrated in May 1985 (p. 32) with the following example. In (20) and (21), we give the relevant example sentence in (a) and indicate the LF representation in (b).

(20) a. Who admired what?
b. [what$_j$ [who$_i$[e_i admired e_j]]]

(21) a. * What did who admire?
b. [who$_i$ [what$_j$ [did e_i admire e_j]]]

In the LF for sentence (20), the object *wh*-phrase *what* is moved in such a way that its path contains that of the subject *wh*-phrase. Sentence (21) is ungrammatical. This is derived by a ban on crossing paths at the LF representation. The explanation of the ungrammaticality of sentence (21) makes essential use of the fact that movement at LF behaves just as movement in S-structure. As May also wants quantifier raising to be parallel to *wh*-movement, he is forced to have a single LF representation for the sentence in (19a).

In May 1985, 1989, there is also a semantic argument given in favor of a single LF for sentences with multiple quantifiers. May points out that there are in general various possible interpretations for sentences with multiple quantifiers. May assumes that all quantifiers that adjoin to the same S node must be considered a unit whose interpretation is determined independently. For illustration, consider the following ambiguous sentence:

(22) a. Nobody loves nobody.
b. $\forall x \neg \forall y \neg [\mathsf{love}'(x, y)]$
c. $\forall x \forall y \neg [\mathsf{love}'(x, y)]$

The sentence can have the readings indicated in (b) and (c). Under the (b)-reading, everyone actually loves someone, whereas under the (c)-reading, no-one loves anyone. According to May, the LF of sentence (22) contains two negative quantifiers adjoined to the S node. The interpretation, then, has the possibility to interpret either quantifier independently, which results in the (b)-reading, or to collapse the two negative quantifiers into a single negative quantifier which binds two variables.[9] With examples like these, May tries to show that the different interpretations of sentences with multiple quantifiers are a semantic property of such sequences of quantifiers and not due to structural differences.

From this short exposition of May's concept of LF, we can extract the following points: First, LF is needed as a syntactic level of representation, as it is needed to exclude sentences as (21). Second, the LF of a sentence does not necessarily express a unique reading, it only indicates the scope domain of a quantifier, i.e., within which clause the quantifier takes its scope, but it is neutral with respect to the relative scope of quantifiers which are adjoined to the same clause.

So far, we have seen that there are some arguments in favor of a syntactic level of LF, i.e., a level that is quite close to semantics but still is the same kind of entity as the other syntactic representations (D-structure, S-structure and PF). The theory of *transparent LF* as presented in von Stechow 1993 shares exactly this perspective. There, however, it is assumed that any transparent LF can be directly used as an lf, i.e., it can be directly interpreted semantically.

> "Wir verstehen *hier* unter den logischen Formen eines Satzes die Repräsentationen, die seine Lesarten eindeutig festlegen, d.h. ein n-deutiger Satz soll n logische Formen haben. [...] Eine plausible Annahme ist es, daß die Ebene der transparenten LF für diesen Fall genau n Strukturen zuläßt, die diese Bedeutungen jeweils eindeutig kodieren."[10]
> (von Stechow, 1993, p. 54)

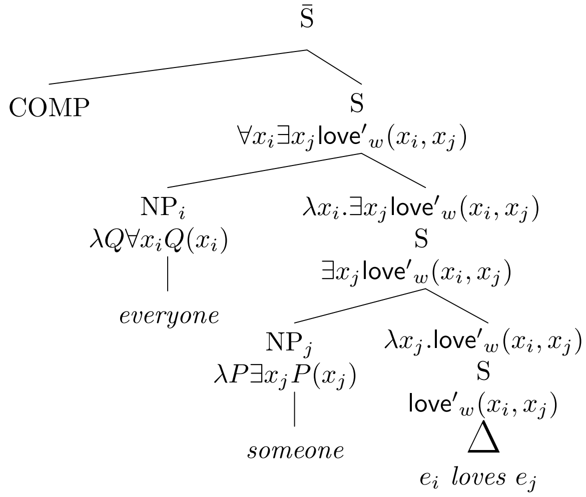
Consequently, Stechow excludes cases of too few LFs or LFs which are not entirely motivated by the needs of semantics. The first situation arises with LFs which allows for several interpretations such as the May-style LF given in Figure 1.6 for sentence (19a). The second situation arises when there are certain movement operations assumed at LF which do not have an impact on the semantic interpretation. We will not encounter this type of an example in this thesis.[11]

[9] In the semantic representation, we should, then, give the (c)-reading as $\forall x, y \neg [\mathsf{love}'(x, y)]$.

[10] "*Here* the logical forms of a sentence are understood as those representations which uniquely determine its readings, i.e., an n-times ambiguous sentence should have n logical forms. [...] It is a plausible assumption that in this case, the level of transparent LF admits exactly n structures which unambiguously encode these meanings." (our translation)

[11] The theory of Progovac 1988 might provide such an example. Progovac assumes that some negative polarity items in Serbo-Croatian move to a negation in an upper clause. By doing so, however, they do not extend their own scope. Thus, this movement is not semantically driven and, the theory is not compatible with a transparent LF.

FIGURE 1.7. The transparent LF for sentence (19a) under reading (19b):



The operations that von Stechow 1993 assumes at the level of LF are QR but also reconstruction, i.e., some of the S-structure movement must be undone to achieve the right LF. LFs also contain material that is not needed elsewhere in syntax, such as some implicit arguments (a world index, a temporal index, an event variable, to name just a few) and also some extra operations at phrasal nodes that influence the interpretation (lambda abstraction and existential quantification over free variables). Let us consider again the ambiguous sentence in (19a). For Stechow, this sentence has two transparent LFs, which correspond to their two readings. In Figure 1.7 we give the transparent LF for the reading in which the subject quantifier has wide scope over the direct object quantifier.[12]

The tree in Figure 1.7 reveals some of the above mentioned properties of Stechow's approach. Firstly, in the semantic representation we use the variable w, which is a world index. It is an implicit argument which does not correspond to any overt syntactic argument in the sentence. Secondly, the traces that Quantifier Raising effectuates, are translated as variables x_i and x_j respectively. Thirdly, when we interpret a tree, we are allowed to execute some semantic operations to the semantics of the daughters in order to make them compatible with each other. In Figure 1.7 this is indicated by giving the semantic representation of a node below its syntactic category, and giving the semantic representation used further in the interpretation above the syntactic category label. As an example, see the translation of the lowest S node is $\mathsf{love'}_w(x_i, x_j)$. Before we can combine this formula with the logical form of the raised quantifier *someone* in the next local tree, we must first abstract over the variable x_j. The semantic representation with which the lower S node is associated in the next local tree is $\lambda x_j.\mathsf{love'}_w(x_i, x_j)$. Analogously, for the next S node, the translation is $\exists x_j.\mathsf{love'}_w(x_i, x_j)$. Before this formula can be combined with the translation of *everybody*, we perform λ-abstraction over the remaining free variable x_i. The translation of the overall sentence is, then, the term given in (19b).

In his concluding remarks, von Stechow 1993 (pp. 83f.) admits that, even though he considers LF a level of syntactic representation, its main mechanisms, reconstruction and abstractions, are not readily found in syntax proper. In particular, reconstruction, being downward movement, is not compatible with standard assumptions about movement. Thus, even though all elements of a transparent LF are motivated by semantics, it seems to be less certain whether they can be equally well motivated by syntax. This, of course,

[12]For the other reading, a different transparent LF would be assumed, in which the NP *someone* is attached higher in the tree than the NP *everyone*.

questions whether the tree structure representation is really appropriate. Under the working hypothesis that syntactic trees are the only data type available, which seems to be shared among the researchers within the Chomskian paradigm, this question does, however, not arise, i.e., if some kind of LF is needed, it can only be a syntactic tree.

Beside the primarily syntactically motivated LF of May, and the primarily semantically motivated transparent LF of von Stechow, there are also researchers claiming that an additional level of representation such as LF is unnecessary. To pick out one proposal in this direction, consider Lappin 1991. Lappin's position is concisely summarized in the following quote.

> "...I propose that rules of model theoretic interpretation apply directly to the S-structure of a sentence in order to yield its interpretation. [...] On this approach, S-structure is the input to semantic rules which define the truth conditions (or appropriate analogue) of a sentence, and there is no additional level of representation defined by QR (and *wh*-raising) which stands between S-structure and the application of these rules. I will refer to this view as "S-structure Interpretivism"."
>
> (Lappin, 1991, p. 310)

Lappin shows for a number of phenomena that an analysis can be given that makes no use of a separate level of representation. Instead, he states interpretation rules for S-structure representations directly.

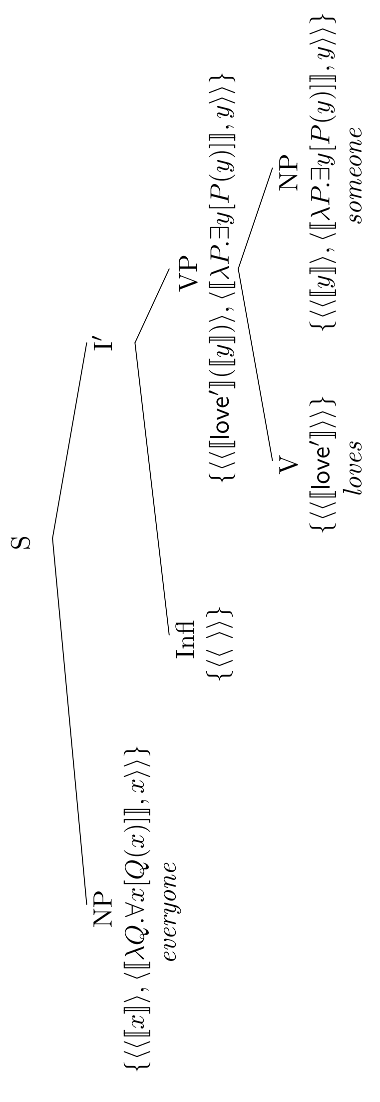
Under the assumption of S-structure Interpretivism, a sentence must be interpreted as a set of meanings, i.e., for a semantically ambiguous sentence such as (19a), there is a single structure, S-structure, which can be interpreted. The interpretation of this structure is the set of all readings of the sentence.

In order to handle scope ambiguities, Lappin 1991 assumes a storage mechanism along the lines of Cooper 1983. In this approach, an NP is interpreted as a free variable together with a store. A store, on the other hand, is a pair consisting of the denotation of the NP and an index. Consequently, using such a storage mechanism introduces a further complexity in the objects that are denoted by a sentence: an S-structure is interpreted as a set of sequences of sequences of meanings (Cooper, 1975, p. 160).

This can best be illustrated with an example. In Figure 1.8 we give the S-structure of sentence (19a). Working within the framework of GB, Lappin assumes an IP to be present in the structure of simple sentences. Below the terminal nodes in the tree, we indicate the translation of the words. The word *loves* is simply translated as the singleton set with the sequence which contains the sequence which contains the denotation of the constant love' as its element. The use of sequences of meanings becomes clearer in the case of the interpretation of the quantified NPs *everyone* and *someone*. In the case of the direct object NP *someone*, the interpretation of the word is a singleton set which contains the sequence consisting of $\langle [\![y]\!] \rangle$ and the store $\langle [\![\lambda P.\exists y[P(y)]]\!], y\rangle$. When these two meanings combine to give the interpretation of the VP, the result is a singleton set whose first element is the sequence containing the functional application of $[\![\mathsf{love}']\!]$ and $[\![y]\!]$, i.e., the functional application of the elements in the first element of the sequences denoted by the daughters. In addition, the sequence in the denotation of the VP also contains the store of the direct object, $\langle [\![\lambda P.\exists y[P(y)]]\!], y\rangle$.

For the node Infl, we can simply assume an empty semantics. Combining this with the interpretation of the VP makes the interpretation of the I$'$ node identical to that of the VP. The interpretation of the S node is more complicated. As the sentence is ambiguous, it does not contain a single sequence, but several distinct sequences. In (23) we give the sequences that appear as elements in the interpretation of the S node.

FIGURE 1.8. The structure of sentence (19a):



(23) a. No storage retrieval:
$\langle\langle[\![\mathsf{love}']\!]([\![y]\!])([\![x]\!])\rangle, \langle[\![\lambda P.\exists y[P(y)]]\!], y\rangle, \langle[\![\lambda Q.\forall x[Q(x)]]\!], x\rangle\rangle$

b. *everyone* retrieved:
$\langle\langle[\![\lambda Q.\forall x[Q(x)]]\!](\{x|[\![\mathsf{love}']\!]([\![y]\!])([\![x]\!]) = 1\})\rangle, \langle[\![\lambda P.\exists y[P(y)]]\!], y\rangle\rangle$

c. *someone* retrieved:
$\langle\langle[\![\lambda P.\exists y[P(y)]]\!](\{y|[\![\mathsf{love}']\!]([\![y]\!])([\![x]\!]) = 1\})\rangle, \langle[\![\lambda Q.\forall x[Q(x)]]\!], x\rangle\rangle$

d. the ∀∃-reading:
$\langle\langle[\![\lambda Q.\forall x[Q(x)]]\!](\{x|[\![\lambda P.\exists y[P(y)]]\!](\{y|[\![\mathsf{love}']\!]([\![y]\!])([\![x]\!]) = 1\}) = 1\})\rangle\rangle$

e. the ∃∀-reading:
$\langle\langle[\![\lambda P.\exists y[P(y)]]\!](\{y|[\![\lambda Q.\forall x[Q(x)]]\!](\{x|[\![\mathsf{love}']\!]([\![y]\!])([\![x]\!]) = 1\}) = 1\})\rangle\rangle$

The sequence in (a) is obtained by combining the first elements of the sequences of the daughters via functional application, i.e., $[\![\mathsf{love}']\!]([\![y]\!])$ is applied to $[\![x]\!]$. In addition, the sequence contains the store $\langle[\![\lambda P.\exists y[P(y)]]\!], y\rangle$ of the VP and the store $\langle[\![\lambda Q.\forall x[Q(x)]]\!], x\rangle$ of the subject NP.

The following sequences are all shorter than the sequence in (a). This is achieved by so-called *retrieval*. Intuitively, in the case of retrieval, a non-initial element of a sequence is removed and integrated into the initial element of the sequence. Starting from a sequence such as (a), retrieval can be characterized schematically as in (24).

(24) $\langle\langle\phi\rangle, s_1, \ldots, s_{i-1}, \langle\psi, x_i\rangle, s_{i+1} \ldots s_n\rangle$
$\mapsto \langle\langle\psi(\{x_i|\phi = 1\})\rangle, s_1 \ldots, s_{i-1}, s_{i+1}, \ldots s_n\rangle$

We can illustrate this with the relation between the sequences (a) and (b) in (23). ϕ is $[\![\mathsf{love}']\!]([\![y]\!])([\![x]\!])$, ψ is $[\![\lambda P.\exists y[P(y)]]\!]$, x_i is simply the variable x, and there is just one more store, $\langle[\![\lambda P.\exists y[P(y)]]\!], y\rangle$. In the sequence in (b), the first store of the (a) sequence is missing. Instead, $[\![\lambda P.\exists y[P(y)]]\!]$ appears as part of the first element of the (b) sequence, just as indicated in the characterization of retrieval in (24).

The (c) sequence is derived just like the (b) sequence, but retrieving the quantifier introduced by the direct object. The last two sequences contain a single element, i.e., there are no quantifiers stored anymore. These sequences correspond to the two readings of the sentence. In (d), the ∀∃-reading is expressed. To derive it, we start from the sequence in (a) and, first, retrieve the existential quantifier, as indicated in (c). Then, the universal quantifier is retrieved. For the ∃∀-reading, the inverse order of retrieval is required.

The interpretation of the S node in Figure 1.8 is the set that contains the two readings and those interpretations which still have some unretrieved quantifier stores.

The advantage of S-structure Interpretivism is clear: we can avoid a further level of representation such as LF. The question, however, arises whether all the constraints formulated in terms of LF can be expressed within such a framework as well. Cooper 1983 (pp. 134ff.) addresses some so called *Island Constraints*. Cooper quotes the following sentences from Rodman 1976, which indicate that complex NPs form an island for scope.[13]

(25) a. Guinevere has $[_{NP}$ a bone in every corner of the house].
b. Guinevere has $[_{NP}$ a bone $[_{S}$ which is in every corner of the house]].

In sentence (25a) the most natural interpretation is such that *every corner of the house* has wide scope over *a bone*, i.e., in every corner of the house there is a bone. The other reading would require the same bone to be simultaneously in all corners of the house. Given the normal size of bones and houses and the laws of physics, this reading is quite absurd. Still, it is impossible to get the more natural reading in the case of (25b). The reason for this restriction, it seems, is the fact that the quantifier *every corner of the house* is embedded in a relative clause in (25b), but not in (25a). Cooper 1983 proposes to account for such data by requiring that at certain nodes the interpretation set may not contain sequences with certain elements. In our case, we would require that the set of interpretations of a relative clause contain only sequences with a single element. This automatically forces the quantifier *every corner of the house* to take scope inside the relative clause in (25b), while it does not impose any restriction on the relative scopes of the quantifiers in the case of (25a).

This brief illustration indicates that some constraints on scope relations can be captured easily with a storage technique. Lappin 1991 also addresses other cases of constraints which have been used to support the existence of a level of representation such as LF. In (26) we repeat the sentences given in (20) and (21).

[13]Notice that the fact that the *Complex NP Constraint* holds for quantifier scope as well as for *wh*-movement has been one of the arguments for LF as a syntactic level of representation. In (i) and (ii) we give the correspondent examples from Cooper 1983 (p. 135) with a *wh*-word.

(i) Which hat do you believe that Otto is wearing t?
(ii) * Which hat do you believe $[_{NP}$ the claim that Otto is wearing t]?

(26) a. (= (20)) Who t admired what?
b. (= (21)) * What did who admire t?

Lappin 1991 (pp. 324f.) argues that the contrast can be reduced to a constraint on S-structures and need not be considered an LF phenomenon. The concrete explanation of the data given in Lappin 1991 goes back to the *Connectedness Condition* of Kayne 1983. The basic effect of this explanation is that a non-moved subject blocks the connection between the specifier of CP and material inside the IP. As such, Kayne intends to substitute the ECP by some other principle which applies at S-structure rather than at LF.[14]

As could be seen in the discussion of the examples in (25), the storage mechanism provides a useful tool to express scope islands. It has been illustrated furthermore with the examples in (26) that for many alleged LF phenomena, an S-structure explanation might be possible. When we consider S-structure Interpretivism, we must differentiate between two aspects: First, the claim that S-structures can be interpreted directly, and second the particular choice of the interpretation mechanism via Cooper storage.

Our own analysis will differ from Lappin 1991 in both respects: We will show in the rest of this section, that the grammar of Pollard and Sag 1994, while using a similar interpretation mechanism, assumes a translation into some logical form instead of a direct interpretation. Our own analysis will follow Pollard and Sag 1994 in this respect. In the next section, we will present the framework of *Lexicalized Flexible Ty2* which is based on Hendriks 1993. We will show that this system can do without the storage mechanism.

Having considered S-structure Interpretivism, we have reached a natural point to come back to the treatment of the CONTENT value in Pollard and Sag 1994, where a Cooper store mechanism is used for the determination of quantifier scope as well. Additionally, the syntactic trees along which the semantics of a sentence is being combined resemble S-structures more than any other level of representation assumed in GB.

In the introduction to this section, we have given some quotations from Pollard and Sag 1987 and Pollard and Sag 1994. The quotes from Pollard and Sag 1987 left it relatively open whether the authors assume that some part of a linguistic sign is a semantic representation or an interpretation, i.e., whether they take a conceptualist or a realistic view. Pollard and Sag 1994 make a direct link to GB's level of LF. In the following, we will show that the semantic system of Pollard and Sag 1994, while being technically similar to Lappin's system, assumes that CONTENT values are *representations of meaning*, not meanings themselves. We will show that a non-ambiguous sentence is assigned two necessarily distinct CONTENT values by the grammar of Pollard and Sag 1994.

Consider sentence (27a). Structurally this sentence is very close to example (19a). In (b) and (c),we give two terms of our semantic representation language which are parallel to the two readings given for example (19a), i.e., in (b) the quantifier that is contributed by the subject has scope over the one contributed by the direct object. In (c) the inverse scope relation holds. It is, however, a consequence of the interpretation of formulae of predicate logic that the terms in (b) and (c) are necessarily equivalent.

(27) a. Everyone loves everyone.
b. $\forall x \forall y \mathsf{love}'(x, y)$
c. $\forall y \forall x \mathsf{love}'(x, y)$

It follows from the equivalence of the terms in (b) and (c) that under the S-structure Interpretivism of Lappin 1991, this sentence has exactly one reading with an empty store.

[14] An alternative S-structure explanation of the contrast in (26) could be based on the assumption that the *wh*-phrase in subject position works as a potential binder of the trace in direct object position.

However, this reading can be derived through two different orderings of quantifier retrieval, which correspond to the scope relations expressed in the terms in (27b) and (27c).

Let us consider how sentence (27a) would be analyzed in the grammar of Pollard and Sag 1994. We do not want to discuss the full semantic analysis of Pollard and Sag 1994, but informally consider the parts that are necessary for this particular sentence. In (28) we give the relevant parts of the lexical entries of the words *everyone* and *loves* as needed for the analysis of sentence (27a).

(28) a. The relevant parts of the lexical entry of *everyone*:

$$\begin{bmatrix} \textit{word} \\ \text{PHON } \langle \textit{everyone} \rangle \\ \text{SYNS LOC} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{SUBCAT} & \langle\rangle \end{bmatrix} \\ \text{CONT } \boxed{2} \begin{bmatrix} \textit{nominal-object} \\ \text{INDEX } \boxed{1} \\ \text{RESTR } \left\{ \begin{bmatrix} \textit{person} \\ \text{INST } \boxed{1} \end{bmatrix} \right\} \end{bmatrix} \end{bmatrix} \\ \text{QSTORE } \left\{ \begin{bmatrix} \textit{quantifier} \\ \text{DET } \textit{forall} \\ \text{RESTIND } \boxed{2} \end{bmatrix} \right\} \\ \text{RETRIEVED } \langle\rangle \end{bmatrix}$$

b. The relevant parts of the lexical entry of *loves*:

$$\begin{bmatrix} \textit{word} \\ \text{PHON } \langle \textit{loves} \rangle \\ \text{SYNS LOC} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{SUBCAT} & \left\langle \text{NP}_{\boxed{1}}, \text{NP}_{\boxed{2}} \right\rangle \end{bmatrix} \\ \text{CONT} \begin{bmatrix} \textit{psoa} \\ \text{QUANTS } \langle\rangle \\ \text{NUCL} \begin{bmatrix} \textit{love} \\ \text{LOVER } \boxed{1} \\ \text{LOVED } \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{QSTORE } \{\} \\ \text{RETRIEVED } \langle\rangle \end{bmatrix}$$

The Cooper store mechanism is implemented in Pollard and Sag 1994 using several different attributes. First, the CONTENT value contains what we treated above as the first element in a sequence. Second, the QSTORE value is a set which contains the stored NPs, i.e., the non-initial members of the sequences considered above. The INDEX value, i.e., the object referred to by the tag $\boxed{1}$ in (28a), is shared between the CONTENT value and the single element in the QSTORE set of *everyone*.

In (28a) the CONTENT value of the quantifier contains the index $\boxed{1}$ and the restriction to humans, expressed by the *person* object. The QSTORE value contains a *quantifier* object. In addition to the attributes CONTENT and QSTORE, the attribute RETRIEVED is part of the Cooper store mechanism as well: the RETRIEVED value of a sign contains those quantifiers that are retrieved at the sign. In the case of the word *everyone*, no such retrieval occurs.

Let us turn to the lexical entry of the verb *loves* as given in (28b). The lexical entry specifies the word as being a verb with two elements on its SUBCAT list. These elements are NPs.[15] The verb *loves* has an empty QSTORE and an empty RETRIEVED value. This means

---

[15]We adopt the abbreviatory conventions of Pollard and Sag 1994 (p. 28) and write $\text{NP}_{\boxed{1}}$ as short form of the description in (i).

(i) $\text{NP}_{\boxed{1}}$: $\begin{bmatrix} \textit{synsem} \\ \text{LOC} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{SUBCAT} & \langle\rangle \end{bmatrix} \\ \text{CONT} \begin{bmatrix} \text{INDEX } \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix}$

that it neither contains any stored quantifiers nor retrieves any quantifiers. Remember from Lappin's analysis, that the interpretation of the verb was a sequence which contained a single element, and no stored quantifier. The CONTENT value of the verb is more interesting. It is of sort *psoa*. It has two attributes, QUANT(IFIER)S and NUCL(EUS). The QUANTS value is a list of quantifiers, the empty list in the case of the verb. The NUCL value indicates some semantic relation. The verb *loves* expresses the relation *love* which takes the indices of its syntactic complements as semantic arguments, i.e., $\boxed{1}$ and $\boxed{2}$.

In Figure 1.9 (page 33), we show the syntactic structure assumed for sentence (27a) in Pollard and Sag 1994. In this figure, we do not describe the CONTENT values at the verbal projections. These are constrained by the SEMANTICS PRINCIPLE (Pollard and Sag, 1994, pp.401f.), from which we quote the relevant parts in (29).

(29) The SEMANTICS PRINCIPLE of Pollard and Sag 1994:
- (a) In a headed phrase, the RETRIEVED value is a list whose set of elements is disjoint from the QSTORE value set, and the union of those two sets is the union of the QSTORE values of the daughters.
- (b) If the semantic head's SYNSEM LOCAL CONTENT value is of sort *psoa*, then the SYNSEM LOCAL CONTENT NUCLEUS value is token-identical with that of the semantic head, and the SYNSEM LOCAL CONTENT QUANTS value is the concatenation of the RETRIEVED value and the semantic head's SYNSEM LOCAL CONTENT QUANTS value; [...]

For the structure in Figure 1.9, the semantic head is always identical to the syntactic head. Furthermore, Pollard and Sag 1994 assume that all verbal projections have a CONTENT value of sort *psoa*. As a consequence of part (b) of the SEMANTICS PRINCIPLE, the NUCL values of the VP and the S node are identical to that of the verb *loves*.

Let us, next, consider the QUANTS, QSTORE and RETRIEVED values at the VP and the S node. The union of the QSTORE values of the daughters of the VP is a set that contains exactly the quantifier contributed by the direct object NP. Let us assume that the RETRIEVED value of the VP is the empty list. In this case, part (a) of the SEMANTICS PRINCIPLE requires the QSTORE value of the VP to be the union of the daughters' QSTORE values. In our case, this is the set that contains the quantifier contributed by the direct object. As the RETRIEVED value is assumed to be empty, the concatenation of the RETRIEVED list to the QUANTS list of the verb results in the empty list. In (30) we give a description of the VP in the tree in Figure 1.9.[16]

(30)
$$\begin{bmatrix} \textit{phrase} & \\ \text{SYNS LOC} & \begin{bmatrix} \text{CAT SUBCAT} \left\langle \boxed{1} \right\rangle & \\ \text{CONT} & \begin{bmatrix} \textit{psoa} & \\ \text{QUANTS} \ \langle\rangle & \\ \text{NUCL} & \begin{bmatrix} \textit{love} \\ \text{ARG1} \ \boxed{3} \\ \text{ARG2} \ \boxed{4} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{QSTORE} & \left\{ \begin{bmatrix} \text{DET } \textit{forall} \\ \text{RESTIND } \boxed{6} \end{bmatrix} \right\} \\ \text{RETRIEVED} \ \langle\rangle & \end{bmatrix}$$

At the S node, again, the CONTENT value is of sort *psoa*, and, thus, the NUCL value is identical to that of the VP. For the S node, we want to assume that the QSTORE value is

[16]Alternatively, we could, of course, retrieve the direct object quantifier at the VP. In this case, the QSTORE value of the VP would be the empty set and the RETRIEVED and the QUANTS value would both be a list that contains exactly the direct object quantifier. If we used such a VP to construct the rest of the sentence, the overall sentence would be described by the AVM in (31a).

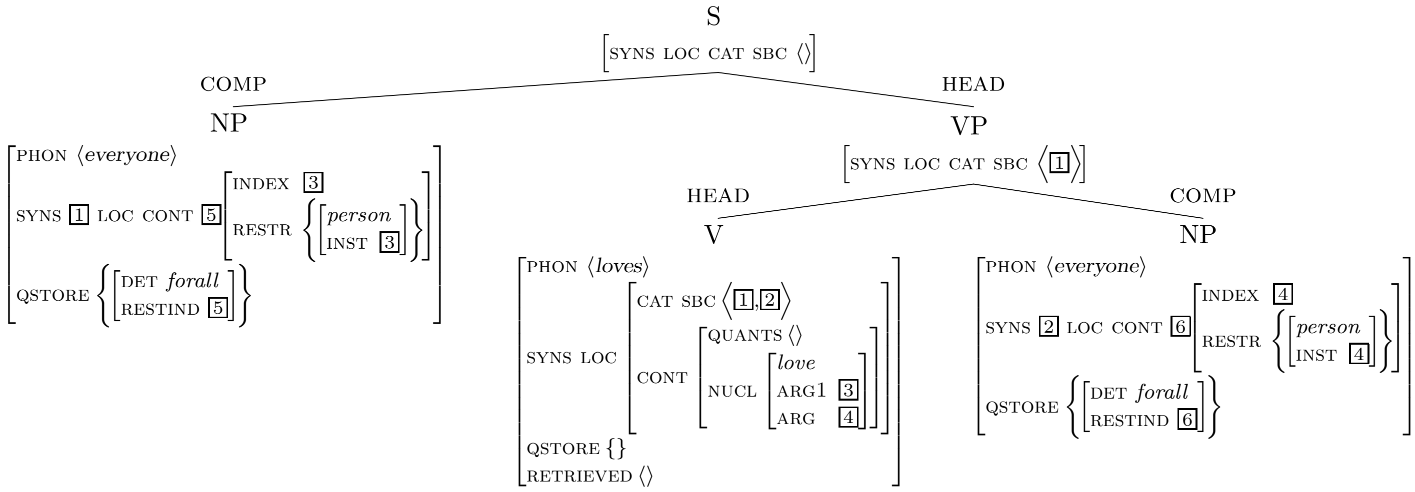


FIGURE 1.9. The structure of the sentence *Everyone loves everyone* according to Pollard and Sag 1994:

empty. Since the union of the QSTORE values of the daughters contains the quantifier contributed by the subject and the quantifier contributed by the direct object, both quantifiers must appear in the RETRIEVED list in order to satisfy part (a) of the SEMANTICS PRINCIPLE. As the RETRIEVED value is a list, there are two possible orderings of these quantifiers: one where the subject quantifier precedes the object quantifier, and one where the object quantifier precedes the subject quantifier. According to part (b) of the SEMANTICS PRINCIPLE, the QUANTS list of the sentence is the concatenation of its RETRIEVED list and the QUANTS list of its head daughter, the VP as described in (30) in our case. In (30) we have assumed that the QUANTS value is empty, but, as there are two possible RETRIEVED values for the sentence, the VP given above can occur as the head daughter in two different kinds of sentences. In (31) we give descriptions of both sentences, depending on the order of the elements in the RETRIEVED list.

(31) a.
$$\begin{bmatrix} \textit{phrase} & \\ \text{SYNS LOC} & \begin{bmatrix} \text{CAT SUBCAT } \langle\rangle & \\ \text{CONT} & \begin{bmatrix} \textit{psoa} & \\ \text{QUANTS} & \boxed{7} \\ \text{NUCL} & \begin{bmatrix} \textit{love} & \\ \text{ARG1} & \boxed{3} \\ \text{ARG2} & \boxed{4} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{QSTORE } \{\} & \\ \text{RETRIEVED } \boxed{7} \left\langle \begin{bmatrix} \text{DET } \textit{forall} \\ \text{RESTIND } \boxed{5} \end{bmatrix}, \begin{bmatrix} \text{DET } \textit{forall} \\ \text{RESTIND } \boxed{6} \end{bmatrix} \right\rangle & \end{bmatrix}$$

b.
$$\begin{bmatrix} \textit{phrase} & \\ \text{SYNS LOC} & \begin{bmatrix} \text{CAT SUBCAT } \langle\rangle & \\ \text{CONT} & \begin{bmatrix} \textit{psoa} & \\ \text{QUANTS} & \boxed{7} \\ \text{NUCL} & \begin{bmatrix} \textit{love} & \\ \text{ARG1} & \boxed{3} \\ \text{ARG2} & \boxed{4} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{QSTORE } \{\} & \\ \text{RETRIEVED } \boxed{7} \left\langle \begin{bmatrix} \text{DET } \textit{forall} \\ \text{RESTIND } \boxed{6} \end{bmatrix}, \begin{bmatrix} \text{DET } \textit{forall} \\ \text{RESTIND } \boxed{5} \end{bmatrix} \right\rangle & \end{bmatrix}$$

What is crucial for our purpose is that the CONTENT values described in (31a) and (31b) **must** be distinct objects in any model of the grammar. To show that they are distinct, it is enough to find a single path identity which occurs in one but not in the other object. For every *psoa* object that is described by description of the CONTENT value in (31a), the QUANTS FIRST RESTIND INDEX value is identical to the NUCL ARG1 value. This identity is expressed by the tag $\boxed{3}$. In the case of the *psoa* objects that meet the description given for the CONTENT value in (31b), on the other hand, the QUANTS FIRST RESTIND INDEX value is identical to the NUCL ARG2 value (indicated by tag $\boxed{4}$).

This shows that there are at least two configurationally distinct *psoa* objects that occur as the CONTENT value of a sign with phonology *Everyone loves everyone.* In Pollard and Sag 1994, if one quantifier precedes another on the QUANTS list of some sign, this is interpreted as the first quantifier having scope over the second quantifier. Therefore, the two descriptions given for the S node in (31) encode the two scoping relations indicated in (27b) and (27c). At the level of semantic interpretation, however, these two scopings lead to identical denotations. This shows that the CONTENT value of a sign in Pollard and Sag 1994 cannot be taken as the *interpretation* of the sign. It can only be seen as a *representation* of this interpretation, i.e., as a *logical form.*

In this thesis, we want to stay within the overall architecture of grammar outlined in Pollard and Sag 1994, but we will give a precise characterization of what the CONTENT value is supposed to be. It will, in fact, share properties of all the accounts just summarized. We will assume that the CONTENT value of a linguistic sign is a configuration of linguistic entities that stand in a one to one correspondence to terms of a semantic representation language. Thus, by its very shape, the CONTENT value is a configuration of objects, i.e., it uses the same data structure that is used in HPSG to encode all linguistic information, from phonology to constituent structure. In this respect, it resembles May's LFs. But, as we assume a direct correspondence between CONTENT values and semantic representations, there obviously are distinct configurations for every reading of a given sentence, thus it is actually closer to the transparent LFs of von Stechow 1993.

On the other hand, the configurations of entities can be understood as logical terms, i.e, as an lf instead of an LF. Consequently, we can incorporate many of the analytic and conceptual insights of S-structure Interpretivism. In particular, there is no claim being made that the restrictions that hold on CONTENT values be of the same kind as those found for movement or other syntactic processes.

To summarize, the general architecture of HPSG as presented in Pollard and Sag 1987 as well as in Pollard and Sag 1994 suggests that the authors assume some semantic representation to be part of linguistic signs. Among the various potential candidates for such a representation, terms of a semantic representation language are clearly the closest to actual semantics, i.e., the closest to an S-structure Interpretivism. We, therefore, assume that the CONTENT value is a logical form. Still, we are going to impose constraints on this lf which are inspired by those used in the GB framework to constrain LFs.

1.3. Introduction to Lexicalized Flexible Ty2

1.3.1. The Choice of the Semantic Representation Language. As a consequence of our decision to use logical forms as values of the attribute CONTENT, it is necessary to address the question of the particular semantic representation language that provides these logical forms. In Section 1.3.2, we will present an extensional typed language, Ty2 (Gallin, 1975), as the relevant semantic representation language. Obviously, this is not the only possible choice. In this section, we will briefly comment on alternatives.

We have decided to adopt an HPSG-encoding of Ty2 because, as is with Montague's *Intensional Logic* (IL), Ty2 is a lingua franca among both semanticists and syntacticians who are concerned with the syntax-semantics interface. Due to its technical advantages shown in Zimmermann 1989 and its successful application to interrogatives in Groenendijk and Stokhof 1982, Ty2 has gained ground among researchers. In contrast to IL, Ty2 makes it possible to treat the world index and other implicit arguments as regular types and, therefore, is more transparent than IL. Still, all the work done in IL can be directly translated. By its inclusion of possible worlds and other implicit arguments, Ty2 allows researchers to propose logical forms for all sorts of semantic problems, including tense, aspect and modality. We, therefore, hope to have chosen a language that can be fruitfully used for the study of many phenomena. More importantly, the choice of Ty2 enables us to integrate directly into HPSG most of the standard techniques and insights that have been developed within formal semantics over the last decades.

It is well known that there are problems with a language such as Ty2. As criticized in Lappin and Pollard 1999, Ty2 is an extensional language and, therefore, all treatments of intensionality have their limitations. As an alternative, they propose a so called *hyperintensional logic* which is built on category theoretic rather than on set theoretic intuitions. In their framework, two predicates can be told apart even if their denotation is identical with respect to all indices.

In this thesis, we do not make any attempt to address the question of what treatment of hyperintensionality (in the Lappin-Pollard form or in another) should ultimately be adopted to solve this problem of extensional languages. This is, in part, legitimate, as the main body of this thesis addresses the question of possible logical forms of a certain linguistic expression, and, thus, only indirectly touches upon the semantic denotation of an expression. If, for example, it should turn out that the representation language proposed in Lappin and Pollard 1999 is empirically more adequate for natural language semantics than Ty2, then it is possible to apply the techniques used here to encode Ty2 within an HPSG grammar for a similar HPSG-encoding of the new language. This shift will be further facilitated by the fact that the syntax of the language of Lappin and Pollard 1999 is very similar to the one used here. Therefore, only the semantic denotation of the terms needs to be modified; the terms themselves, i.e., the logical forms of a given expression, can be adopted largely unchanged.[17]

A second justified objection is that any sentence-based determination of logical form ignores dynamic effects in meaning. Such effects have been extensively studied in frameworks such as Discourse Representation Theory (DRT) (Kamp, 1981; Kamp and Reyle, 1993), file change semantics (Heim, 1982), or dynamic predicate logic (DPL) (Groenendijk and Stokhof, 1991). While we will not address dynamic phenomena in the following, it is shown in Dekker 1993 that the system of *Flexible Montague Grammar* (FMG) of Hendriks 1993 can easily be extended to capture dynamic effects as well by adding another type shifting rule. As the system of Lexicalized Flexible Ty2 that we are going to present in this section is based on Hendriks' FMG, the dynamic extension proposed in Dekker 1993 can be integrated into our system as well.

It would, of course, require far more effort to change from Ty2 to a DRT-style representation. There are some proposals of how to represent DRSs (or UDRSs) within HPSG (Frank and Reyle, 1995; Eberle, 1997). The present proposal shares the assumption with DRT that there should be a syntactic level of semantic representation. Furthermore, the semantic representations used in DRT also contain quantifiers and logical constants as well as variables for explicit and implicit arguments. Again, to a certain extent our choice of Ty2 rather than DRT is motivated not so much by empirical considerations but by the fact that our own approach is strongly influenced by research within the Montegovian tradition.

A further alternative to a truth-value based, possible-worlds semantics is situation semantics (Barwise, 1989). In situation semantics, situations and relations are treated as primitives, whereas truth is only a derived term. In Pollard and Sag 1987, 1994, the authors intend to use a version of situation semantics. This program is further worked out in Ginzburg and Sag 2000 for interrogatives and in Przepiórkowski and Kupść 1999 and Przepiórkowski 1999b for negation. Lappin and Pollard 1999 (p. 2) call situation semantics "the principle competitor to possible-world semantics as a mathematical theory of linguistic meaning". Within HPSG, however, there is not much work done within possible-world semantics. As far as we can see, Nerbonne 1992 was a single attempt to furnish HPSG with terms of predicate logic as CONTENT values. But, in that paper, issues of intensionality are not addressed. We, therefore, consider it an important gap to be filled to show how a possible-world semantics approach can be integrated into HPSG.[18] Thus, the decision for a particular kind of syntactic framework, be it GB or HPSG, should not automatically force the linguist to abandon his or her favorite semantic framework.

To conclude, there are at present many promising semantic representation languages whose adequacy is being tested on wide ranges of semantic phenomena. The particular

[17]This is due to the fact that Lappin and Pollard 1999 also use an explicit negation operator, a similar representation of generalized quantifiers and some implicit arguments such as an event argument.

[18]Richter and Sailer 1996a sketch an HPSG integration of Montague's *Intensional Logic*, and Richter and Sailer 1999a,b,c,d, 2001 either sketch or presuppose an integration of Ty2.

choice taken in this thesis represents just one of many standard representation languages within the study of natural language semantics. Finally, if this represenation language proves to be empirically inferior to some other language, it is very likely that the technique of HPSG-encoding a semantic representation language can be adopted.

1.3.2. Ty2. In this section, we define the semantic representation language Ty2, which Gallin 1975 (pp. 58f.) first presented. We will first define the syntax of Ty2 and then give the semantics. We assume that the reader is familiar with Montague's *Intensional Logic* (IL) and will point out some of the differences between Ty2 and IL. Furthermore our definition of Ty2 departs slightly from the usual definitions as given in Gallin 1975 or Zimmermann 1989. We will indicate the places where these differences occur.[19]

In Definition 1.1 we define the set of semantic types, $Type$. In Ty2 we use the atomic types e, t and s. The set of semantic types is closed under pairs. The crucial difference between the set of semantic types of Ty2 and IL is that in Ty2, s is an atomic type just like t and e.

DEFINITION 1.1 *(Type)*

Let Type be the smallest set such that

$$e, t, s \in Type, \text{ and}$$
$$\text{for each } \tau, \tau' \in Type, \langle \tau, \tau' \rangle \in Type.$$

We call each element of the set $Type$ a *(semantic) type.* The set $Type$ as defined above contains the semantic types used in Montague Grammar such as $\langle e, t \rangle$ for intransitive verbs or $\langle s, t \rangle$ for propositions. In addition, it also contains types such as s, $\langle e, s \rangle$ etc. which have no direct counterpart in IL.

When we give semantic types, we usually use some common abbreviatory notation: for complex types, we simply write $\tau\tau'$ or $(\tau\tau')$. Consequently, we write et instead of $\langle e, t \rangle$, $(se)t$ instead of $\langle \langle s, e \rangle, t \rangle$, and $s((se)t)$ instead of $\langle s, \langle \langle s, e \rangle, t \rangle \rangle$.

In Definition 1.2 we define the set Var as a countably infinite set of variables.

DEFINITION 1.2 *(Var)*

Let Var be the smallest set such that

$$\text{for each } \tau \in Type \text{ and for each } i \in \mathbb{N}, v_{\tau,i} \in Var$$

We will call each element of the set Var a *variable.* Usually, we will distinguish variables by giving them distinct numbers or by using distinct letters. As will become clear in the following, the variable $v_{s,0}$ plays a special role in Lexicalized Flexible Ty2. We will introduce the special symbol "@" for this variable.

In addition to variables, we also assume that there are constants in our semantic representation language. This is made explicit in the next definition.

DEFINITION 1.3 *(Const)*

Let Const be a finite set of symbols.

We call each element of the set $Const$ a *constant.* In the standard definitions of Ty2, the set of constants is defined parallel to the set of variables, i.e., for each semantic type τ and

[19]The HPSG encoding of Ty2 given in Richter 2000 (Section 5.4) is more faithful to the standard definitions.

for each natural number i, there is a constant $c_{\tau,i}$. In contrast, we assume that there are only finitely many constants. The reason lies in the particular HPSG encoding of Ty2 that we assume. There, we will introduce a linguistic sort for each semantic constant, and more crucially, we will introduce a relation for each semantic constant. Our HPSG description language requires the set of relations to be finite. Therefore, we cannot use an infinite number of constants under this encoding.[20]

We assume that each constant is associated with a semantic type. This requirement is expressed in the following definition.

DEFINITION 1.4 *(Type assignment to constants)*

Let $\mathcal{C}$ be a total function from $Const$ to $Type$.

The function $\mathcal{C}$ assigns each constant a semantic type. As the set $Type$ is non-empty, there is such a function for each set $Const$. For example, the constant walk' will be assigned the semantic type $s(et)$, the constant $\mathsf{believe}'$ will be assigned the type $s((st)(et))$.

The semantic types assigned to these constants is slightly different from the types that they bear in IL. In fact, when, in IL, a constant is of type et, it will typically be of type $s(et)$ in Ty2. This is the case, because in IL all terms are interpreted with respect to some world index. In Ty2, this world index is an explicit part of the terms. In fact, we use the variable $v_{s,0}$ (written as @) for this index. This explicit occurrence of the world index results in the requirement that all constants whose denotations vary with the world index have one semantic argument more than their corresponding IL constants.

The only constants for which we do not assume such a dependence on the index are the individual constants, i.e., constants of type e. They appear in the translations of proper names such as j_e and m_e which are used for the logical form of *John* and *Mary* respectively.

In the next definition, we define the set of terms of Ty2. We write Ty2 for this set. Our definition of the set Ty2 differs in two respects from the standard definition. First, as we assume a finite set of constants with a particular type assignment function, the set of Ty2 terms is parameterized for the function $\mathcal{C}$. For this reason, we can only give the definition of the Ty2 terms under a particular function $\mathcal{C}$, i.e., we define the set $\text{Ty2}_\mathcal{C}$ for a concrete $\mathcal{C}$. In the following, we will assume a fixed set $Const$ and a fixed function $\mathcal{C}$ and, in most cases, ignore the relativization to $\mathcal{C}$ and simply write Ty2 instead of $\text{Ty2}_\mathcal{C}$.

The second difference between the following definition and the usual definitions of Ty2 lies in the fact that we assume the semantic type to appear explicitly as part of each term of Ty2. In the standard definition, the set Ty2 is defined as the smallest family of sets Ty2_τ for each semantic type τ. Again, the difference has to do with the way we encode terms of Ty2 in HPSG. In our HPSG encoding, each linguistic object that corresponds to some term of Ty2 will be explicitly marked for its semantic type. For this reason, we require the semantic type to be a part of each term of Ty2 as well.

In Definition 1.5 we give the resulting definition of Ty2 terms.

DEFINITION 1.5 *($Ty2_\mathcal{C}$)*

Given a finite set $Const$ and a function $\mathcal{C}$ from $Const$ to $Type$, $Ty2_\mathcal{C}$ is the smallest set such that,

$Var \subset Ty2_\mathcal{C}$,

for each $c \in Const$, $c_{\mathcal{C}(c)} \in Ty2_\mathcal{C}$,

[20]It is, of course possible to give an encoding of an infinite set of constants. This is done, for example in Richter 2000 (Section 5.4), where constants are encoded the same way we encode variables.

for each $\phi_{\tau\tau'}, \psi_\tau \in Ty2_\mathcal{C}$,
$(\phi_{\tau\tau'}\psi_\tau)_{\tau'} \in Ty2_\mathcal{C}$,
for each $v_{\tau,i} \in Var$, and for each $\phi_{\tau'} \in Ty2_\mathcal{C}$,
$(\lambda v_{\tau,i}.\phi_{\tau'})_{\tau\tau'} \in Ty2_\mathcal{C}$
for each $\tau \in Type$, and for each $\phi_\tau, \psi_\tau \in Ty2_\mathcal{C}$
$(\phi_\tau = \psi_\tau)_t \in Ty2_\mathcal{C}$

Constants and variables are atomic terms. Complex terms are formed by functional application, λ-abstraction or equation. Even though, according to our definition, each term of Ty2 has its semantic type as a subscript, we will usually not write the semantic types of complex expressions, because it can be computed from the semantic types of its atomic subterms. In addition, we will deliberately alternate between curried and uncurried notation. In (32) we indicate some terms in explicit and in shorthand notation.

(32)

explicit notation	shorthand
$((\mathsf{walk}'_{s(et)} v_{s,0})_{et} m_e)_t$	$\mathsf{walk}'(m, v_{s,0})$
$(((\mathsf{love}'_{s(e(et))} v_{s,0})_{e(et)} m_e) j_e)_t$	$\mathsf{love}'(j, m, v_{s,0})$

With Definition 1.5 we have introduced the syntax of the semantic representation language Ty2. We can then proceed to give its semantics. Based on the semantic types, a domain of individuals E, and some set of possible worlds W, we can define a *frame* F.

DEFINITION 1.6 *(Frame) Let E be a set of individuals, and W be a set of possible worlds, then $F = \bigcup_{\tau \in Type} D_{E,W,\tau}$ is a frame where,*

$D_{E,W,t} = \{1, 0\}$,
$D_{E,W,e} = E$, *and*
$D_{E,W,s} = W$, *and*
for each $\tau, \tau' \in Type$,
$D_{E,W,\tau\tau'} = D_{E,W,\tau'}^{D_{E,W,\tau}}$.

The definition of a *frame* in Definition 1.6 provides us with the necessary objects that populate the denotation of Ty2 terms. Given a frame, we can say what a model of Ty2 is.

DEFINITION 1.7 *(Model)*

Given a set of constants $Const$, a type assignment function to constants $\mathcal{C}$, a set of individuals E and a set of possible worlds W,

a Ty2 model is a pair $M = \langle F, int\rangle$, such that
F is a frame, and
int is a function from the set of constants to F such that
for each $c \in Const$,
$int(c) \in D_{E,W,\mathcal{C}(c)}$.

A model consists of a frame F and an interpretation function *int* for constants. The function *int* is chosen in a way that respects the semantic type of the constants, i.e., if a constant is of type τ, then its interpretation is an element of $D_{E,W,\tau}$.

While the function *int* is an interpretation function for the constants of $\text{Ty2}_\mathcal{C}$, we still need to give a definition of what a variable assignment is. This definition is given as follows.

DEFINITION 1.8 *(Variable assignment)*

Ass is a subset of F^{Var} *such that,*

$$Ass = \left\{ a \in F^{Var} \,\middle|\, \begin{array}{l} \textit{for each } i \in \mathbb{N}, \\ \textit{for each } \tau \in Type, \\ a(v_{\tau,i}) \in D_{E,W,\tau} \end{array} \right\}$$

We call each element of the set Ass a *variable assignment*. The way Ass is defined, it is guaranteed that each variable assignment respects the type of each variable, i.e., if a variable is of type τ, then each variable assignment will map it into an element in $D_{E,W,\tau}$.

We can now give the semantics of Ty2 terms. In Definition 1.9, we define the denotation function $[\![\ ]\!]$ with respect to a model and a variable assignment.

DEFINITION 1.9 *(The semantics of* $Ty2_{\mathcal{C}}$*)*

For each term $\phi_\tau \in Ty2$*, for each model* M *and for each variable assignment* $a \in Ass$*,*

> $[\![\phi_\tau]\!]^{M,a}$, *the extension of a term* ϕ_τ *in a model* $M = \langle F, int\rangle$ *under a variable assignment* $a \in Ass$*, is defined as follows:*
> *for each* $c \in Const$,
> $[\![c_{\mathcal{C}(c)}]\!]^{M,a} = int(c)$,
> *for each* $\tau \in Type$, *and for each* $i \in \mathbb{N}$,
> $[\![v_{i,\tau}]\!]^{M,a} = a(v_{i,\tau})$ *for variables* $v_{i,\tau}$
> *for each* $\phi_{\tau\tau'} \in Ty2_{\mathcal{C}}$, *and for each* $\psi_\tau \in Ty2_{\mathcal{C}}$,
> $[\![(\phi_{\tau\tau'}\psi_\tau)_{\tau'}]\!]^{M,a} = [\![\phi_{\tau\tau'}]\!]^{M,a}([\![\psi_\tau]\!]^{M,a})$
> *for each* $v_{\tau,i} \in Var$ *and for each* $\phi_\tau \in Ty2_{\mathcal{C}}$,
> $[\![(\lambda v_{\tau,i}.\phi_{\tau'})_{\tau\tau'}]\!]^{M,a} = f \in D_{E,W,\tau'}^{D_{E,W,\tau}}$ *such that*
> *for each* $d \in D_{E,W,\tau}$: $f(d) = [\![\phi_{\tau'}]\!]^{M,a[v_{\tau,i}/d]}$,
> *for each* $\phi_\tau, \psi_\tau \in Ty2_{\mathcal{C}}$,
> $[\![(\phi_\tau = \psi_\tau)_t]\!]^{M,a} = 1$ *if* $[\![\phi_\tau]\!]^{M,a} = [\![\psi_\tau]\!]^{M,a}$, *else* 0.

The interpretation of atomic terms depends on the interpretation function int for constants and on the variable assignment a for variables. Complex terms are interpreted in the standard way: Applications are interpreted as functional application. λ-abstraction is interpreted as a function from the domain determined by the type of the variable to the domain determined by the type of the second term. An identity denotes 1 in case both terms have identical denotation and otherwise it denotes 0.

In the definitions given so far, we have only included three kinds of complex terms in Ty2. With the help of these, it is possible to define further constructs such as true, false, negation, conjunction, disjunction, implication, existential and universal quantification as syntactic sugar. The necessary definitions are given in (33), following Gallin 1975 (p. 15).

(33) a. true: $[\lambda x_t.x_t = \lambda x_t.x_t]$
b. false: $[\lambda x_t.x_t = \lambda x_t.\mathsf{true}]$
c. $\neg\phi_t$: $[\mathsf{false} = \phi_t]$
d. $\phi_t \wedge \psi_t$: $[\lambda f_{tt}[f\phi = \psi] = \lambda f_{tt}[f\mathsf{true}]]$
e. $\phi_t \rightarrow \psi_t$: $[[\phi \wedge \psi] = \phi]$
f. $\phi_t \vee \psi_t$: $[\neg\phi \rightarrow \psi]$
g. $\forall x_\tau\phi_t$: $[\lambda x_\tau\phi = \lambda x_\tau\mathsf{true}]$
h. $\exists x_\tau\phi_t$: $\neg\forall x_\tau\neg\phi$

To see that this shorthand has the desired meaning, consider the case of true and false. Independent of the model and the variable assignment, the term in (33a) denotes 1, because the terms on either side of the equation symbol are identical. The term in (33b), on the other hand, denotes 0 independent of the model and the variable assignment. This is the case, because the subterm $\lambda x_t.x_t$ is the identity function on the set $\{0,1\}$, i.e., $\lambda x_t.x_t$ maps 0 into 0 and 1 into 1. The subterm $\lambda x_t.\mathsf{true}$, however, is the constant function that maps each element of $\{0,1\}$ into 1. Clearly these two functions differ, and therefore, the equation in (33b) is always interpreted as 0.

Using the syntactic sugar introduced in (33), we also receive the standard interpretation for the logical connectives and the quantifiers: Negation reverses the truth value of a term, i.e., the negation of a term denotes 1 if and only if the term denotes 0.

Conjunction denotes 1 if and only if both conjuncts do so. To see that this is the case, we must consider the term in (33d) in more detail. In (34), the first column gives all the functions from t to t. In the second column, we indicate the value of the function $\lambda f_{tt}.f(\mathsf{true})$ on the function in the first column. Similarly, in the next four columns, we give the value of the function $\lambda f_{tt}.[f(\phi) = \psi]$ for the functions in the first column, depending on the interpretation of ϕ and ψ.

(34)

		$(\lambda f_{tt}.[f(\phi) = \psi])(f_{tt})$			
f_{tt}	$(\lambda f_{tt}.f(\mathsf{true}))(f_{tt})$	$\phi = 1$, $\psi = 1$	$\phi = 1$, $\psi = 0$	$\phi = 0$, $\psi = 1$	$\phi = 0$, $\psi = 0$
$\begin{bmatrix} 1 \mapsto 1 \\ 0 \mapsto 1 \end{bmatrix}$	1	1	0	1	0
$\begin{bmatrix} 1 \mapsto 1 \\ 0 \mapsto 0 \end{bmatrix}$	1	1	0	0	1
$\begin{bmatrix} 1 \mapsto 0 \\ 0 \mapsto 1 \end{bmatrix}$	0	0	1	1	0
$\begin{bmatrix} 1 \mapsto 0 \\ 0 \mapsto 0 \end{bmatrix}$	0	0	1	0	1

As the table shows, the two functions $\lambda f_{tt}.f(\mathsf{true})$ and $\lambda f_{tt}.[f(\phi) = \psi]$ are only identical if ϕ and ψ are both interpreted as 1, in all other cases, the two functions differ.

The definition of conjunction is used in the definition of implication: Implication denotes 1 if the conjunction of the antecedent and the consequent has the same denotation as the antecedent alone. To define disjunction, we use implication and negation: a disjunction denotes 1 if the negation of the first disjunct implies the second disjunct. To see that this is the case iff at least one of the disjuncts denotes 1, consider the following two cases: First, assume that ϕ denotes 1. In this case $\neg\phi$ denotes 0, and $\neg\phi \rightarrow \psi$ denotes 1. Second, assume that ϕ denotes 0 and ψ denotes 1. In this case $\neg\phi$ denotes 1 and $\neg\phi \rightarrow \psi$ also denotes 1.

Universal quantification denotes 1 if and only if for each entity d in the frame which is of the appropriate type, the term in the scope of the quantifier denotes 1 if the variable bound by the quantifier is interpreted as d. This is expressed by the λ-abstraction in (33g).

An existential quantification denotes 1 if and only if there is an entity d in the frame such that the term in the scope of the quantifier denotes 1 if the variable bound by the quantifier is interpreted as d. In (33h) this is expressed by a negation on the universal quantifier.

With the definitions given so far, we have defined the language Ty2$_{\mathcal{C}}$ for which we are going to provide and prove an HPSG encoding in Chapter 3. Before we comment further on the precise semantic representation language that we are going to assume for natural language semantics, we will consider some terms of Ty2 as they occur in the fragment of English that we will treat in the next subsection.

Our fragment will be based on that of PTQ (Montague, 1974b) in the Ty2-rendered form following the method given in Gallin 1975 (p. 61) and Groenendijk and Stokhof 1982 (p. 188). In (35) we give the definition of the function "*" which maps each term of IL into a translation in Ty2. The definition is adapted from Gallin 1975 (p. 61) and ignores the parameterization with respect to the function $\mathcal{C}$ which is needed for Ty2$_{\mathcal{C}}$. In contrast to Gallin, but following Groenendijk and Stokhof 1982, we translate IL constants of type e as Ty2 constants of the same type, whereas, we choose an intensional type $(s\tau)$ for the translate of other constants. With that we express the view that the interpretation of individual constants does not vary with the world index.[21]

(35) For each term ϕ_τ of IL we define ϕ^*_τ, the translate of ϕ_τ in Ty2, as follows:

$$v^*_{\tau,i} = v_{\tau,i}$$
$$c^*_e = c^*_e$$
$$c^*_\tau = (c_{s\tau} v_{s,0})_\tau, \text{ for } \tau \neq e,$$
$$(\phi_{\tau\tau'}\psi_\tau)^*_{\tau'} = (\phi^*_{\tau\tau'}\psi^*_\tau)_{\tau'}$$
$$(\lambda x_{\tau,i}.\phi_{\tau'})^*_{\tau\tau'} = (\lambda x_{s\tau,i}.\phi^*_{\tau'})_{\tau\tau'}$$
$$(\phi_\tau = \psi_\tau)^*_t = (\phi^*_\tau = \psi^*_\tau)_t$$
$$\hat{}\phi_\tau = (\lambda v_{s,0}.\phi^*_\tau)_{s\tau}$$
$$\check{}\phi_{s\tau} = (\phi^*_{s\tau} v_{s,0})_\tau$$

In the translate of formulae of the form $\hat{}\phi$ and $\check{}\phi$, we see a variable, $v_{s,0}$, of type s. In IL, the interpretation of a formula depends on the world index, i.e., it is given as an extension with respect to a model, a variable assignment function and a world index. In Ty2, world indices are part of the representation language. Therefore, if we want to express that the interpretation of a formula may vary with some index, we can use a formula in which a variable of type s is free. In this case, the interpretation of the formula depends on the variable assignment and, in particular, on the assignment to the free variable of type s. In the semantic analysis, we assume that the first variable of type s, i.e., $v_{s,0}$, is used in the way, the "current" index is used in IL. For this reason, we translate the IL term $\mathsf{walk}'(j)$ as $\mathsf{walk}'(v_{s,0})(j)$. The result of the occurrence of the free variable $v_{s,0}$ in this Ty2 term is that the extension of this term depends on the variable assignment on the variable $v_{s,0}$, i.e., it depends on the current world index.

Because of the frequent occurrence of this particular variable, we introduce some short-hand. We usually write @ instead of $v_{s,0}$. In addition, we write $\phi_@$ instead of $\phi(@)$. In (36) we indicate some example terms where we use this notation. The terms in (36) correspond to the IL terms $\mathsf{walk}'(\check{}x)$, $\mathsf{love}'(\check{}x, \check{}y)$, and $\lambda P.\check{}P(\hat{}m)$ respectively.

(36)

explicit notation	short hand
$((\mathsf{walk}'_{s(et)} v_{s,0})_{et}(v_{se,0}(v_{s,0}))_e)_t$	$\mathsf{walk}'_@(x_@)$
$(((\mathsf{love}'_{s(e(et))} v_{s,0})_{e(et)}(v_{se,1}(v_{s,0}))_e)_{et}(v_{se,0}(v_{s,0}))_e)_t$	$\mathsf{love}'_@(x_@, y_@)$
$\lambda v_{s((se)t),0}.((v_{s((se)t),0}(v_{s,0}))_{(se)t}(\lambda v_{s,0}.m_e))_t$	$\lambda P.P_@(\lambda @.m)$

The semantic representation language Ty2 as defined in this subsection is expressive enough to capture effects of intensionality in the way developed in Montague Grammar. We will use the terms of Ty2 as the logical form of words or phrases in a grammar of a fragment of English. We will sometimes refer to the logical forms of a sign as its *translation*. The translations that we will propose in this thesis are close to the semantic analysis in PTQ. In the following subsections, we will show how the right translations are associated with the linguistic signs.

[21]Note that we follow Gallin 1975 (p. 61) in calling ϕ^* the *translate* of ϕ, not the *translation*.

For some semantic phenomena, it is convenient if the semantic description language contains *generalized quantifiers*.[22] As an example, consider the quantifier *most* in (37).

(37) Most students like wine.

This sentence is true if and only if the number of students that like wine is greater than the number of students that do not like wine. While it is impossible to express this meaning in first order predicate logic, Ty2 is powerful enough.

For the quantifiers $\forall$ and $\exists$, the truth of a term $\forall x_\tau\, \phi$ or $\exists x_\tau\, \phi$ depends on the cardinality of the set $\lambda x_\tau.\phi$. In the first case, it is required to be the entire domain $D_{E,W,\tau}$, while in the latter case, it is required to be a nonempty set. A generalized quantifier such as introduced by the word *most* can express a more complex relation on the cardinality of two (or even more) sets. In the example in (37), we are interested in two sets: the set of all students and the set of wine-likers. The sentence is true if the cardinality of the intersection of these two sets is greater than the cardinality of the first set minus the second set. This means that there is a quantifier most which expresses a relation between two sets A and B such that $|A \cap B|$ is greater than $|A \setminus B|$.

In our linguistic examples in the second part of this thesis, we will use description language which contains such quantifiers explicitly. For this purpose, we introduce some more abbreviatory syntax to the semantic representation language: we use symbols such as most, few, etc. In (38), we give the necessary definition for *most*.

(38)
$$
\begin{array}{ll}
(\mathsf{most}\ x_\tau : \phi_t)[\psi_t]_t = & \\
\exists S_{\tau t}\ \exists T_{\tau t}\ \exists R_{((\tau t)((\tau t)t))} & \\
\quad [S = \lambda x_\tau.[\phi \wedge \psi] \wedge T = \lambda x_\tau.[\phi \wedge \neg\psi] & \text{(i)} \\
\quad \wedge \forall x & \\
\qquad [\exists y R(x,y)] \leftrightarrow T(x) & \text{(ii)} \\
\qquad \wedge \forall y \forall z[[R(x,y) \wedge R(x,z)] \rightarrow y = z] & \text{(iii)} \\
\qquad \wedge \forall y[R(x,y) \rightarrow S(y)] & \text{(iv)} \\
\qquad \wedge \forall z[\exists y[R(x,y) \wedge R(z,y)] \rightarrow x = z] & \text{(v)} \\
\quad \wedge \exists y[S(y) \wedge \neg\exists x R(x,y)]] & \text{(vi)}
\end{array}
$$

In line (i) of (38), the variable S is defined as exactly that element of $\mathrm{D}_{E,W,t}{}^{D_{E,W,\tau}}$ that denotes the intersection of the two sets provided in the restriction (ϕ) and the nucleus (ψ) of the most-formula. Analogously, T denotes exactly the complement of these two sets. The remaining lines express that the intersection of these two sets (S) is bigger than the complement (T). To express this cardinality requirement, we state that R is a function from T to S (lines (ii)–(iv)), which is injective (line (v)) but not surjective (line (vi)).

In this subsection, we have introduced the semantic representation language Ty2 as we are going to use it in the rest of this thesis. We have shown that each term of IL has a direct counterpart in Ty2. Therefore, we can use a PTQ-style semantic analysis, while working with Ty2 instead of IL. In the next subsection, we will show how words and phrases are assigned some term of Ty2 as their translation.

In Chapter 3 we will make the link between the definitions of Ty2 in this subsection and the grammar formalism for HPSG, i.e., the language RSRL. We will give an RSRL grammar $\mathcal{TY}2$. We will prove that this grammar is an adequate account of the language Ty2, because the language Ty2 is an exhaustive model of this grammar. As a result, we will be able to use terms of Ty2 as CONTENT values of linguistic signs.

[22] See Barwise and Cooper 1981, Westerståhl 1989, Gamut 1991b (Section 7.2), or Keenan and Westerståhl 1997 for some introduction.

1.3.3. Lexicalized Flexible Ty2. In the preceding subsection, we introduced the semantic representation language that we will use throughout this thesis. The purpose of this subsection is to show how we intend to address the assignment of semantic representations to words and phrases. This semantic framework, called *Lexicalized Flexible Ty2* (LF-Ty2), is based on the *Flexible Montague Grammar* (FMG) of Hendriks 1993. In this framework, we assume that every word is assigned a *basic translation.* In order to arrive at the translation of phrases, we assume that the translation of a phrase is the intensional functional application of the translations of its daughters. In addition to these two fundamental specifications of the translation, it is the special property of *flexible* systems such as FMG and LF-Ty2 that we can also apply a number of so-called *shifting* rules to translations. These shifting rules will allow us to dispense with syntactic movement such as QR or semantic storage mechanisms such as Cooper stores. In this subsection, we will present the basic translations of the words of a small fragment of English. Then, we will show how intensional functional application can be used to compute the translation of a phrase. Finally we will define two shifting operations and show how they account for the different readings of ambiguous sentences.

TABLE 1.1. Basic translations

word	Ty2 term	IL term (as used in FMG)
man	$\lambda x_{se}.\mathsf{man}'_{@}(x_{@})$	$\lambda x.\mathsf{man}'({}^{\vee}x)$
woman	$\lambda x_{se}.\mathsf{woman}'_{@}(x_{@})$	$\lambda x.\mathsf{woman}'({}^{\vee}x)$
walks	$\lambda x_{se}.\mathsf{walk}'_{@}(x_{@})$	$\lambda x.\mathsf{walk}'({}^{\vee}x)$
Mary	m_e	m_e
every	$\lambda P\lambda Q.\forall x_{se}[P(@)(x) \rightarrow Q(@)(x)]$	$\lambda P\lambda Q.\forall x_{se}[{}^{\vee}P(x) \rightarrow {}^{\vee}Q(x)]$
some	$\lambda P\lambda Q.\exists x_{se}[P(@)(x) \wedge Q(@)(x)]$	$\lambda P\lambda Q.\exists x_{se}[{}^{\vee}P(x) \wedge {}^{\vee}Q(x)]$
believes	$\lambda p_{st}\lambda x_{se}.\mathsf{believe}'_{@}(p)(x_{@})$	$\lambda p\lambda x.\mathsf{believe}'(p)({}^{\vee}x)$
that	$\lambda p_{st}.p_{@}$	$\lambda p.{}^{\vee}p$

In Table 1.1 we indicated the basic translation of some words of our fragment. The first column shows the word whose translation is given. The Ty2 term in the second column is the assumed translation. In order to make the semantic type of the term clear, we write the type of the variables at their first occurrence in a term. In the last column, we state an IL term. This IL term is the basic translation of the word as put forth in the framework of FMG in Hendriks 1993.

We will use the symbol "$\leadsto$" for the basic translation of a word. To express that the function "$\leadsto$" maps the word *walks* into the term $\lambda x.\mathsf{walk}'_{@}(x_{@})$, we write *walks* $\leadsto \lambda x.\mathsf{walk}'_{@}(x_{@})$.

The semantic types of the translations in Table 1.1 are not those of PTQ, but those assumed in FMG of Hendriks 1993. In the flexible system of Hendriks 1993 the basic translation of a proper name such as *Mary* is just a constant of type e. In contrast to this, in PTQ, the name would translate into the IL term $\lambda P_{s((se)t)}.{}^{\vee}P({}^{\wedge}m_e)$, which corresponds to the Ty2 term $\lambda P_{s((se)t)}.P(@)(\lambda @.m_e)$. We have also given a basic translation for the complementizer *that.* In Montague 1974b and Hendriks 1993 the phonology of the complementizer is part of the terminal symbol used for verbs that take *that*-complements. Because we will treat the complementizer as an independent word in syntax, it is necessary to assign it some basic translation. We will see later that the particular assignment that we have chosen results in an identity function under intensional functional application. For the other examples given in Table 1.1, the basic translation of the words is the same as in PTQ.

It should be noted that the variable "@" occurs free in every basic translation in Table 1.1. In fact, as we shall see later, this variable occurs free in every translation of a sign. As pointed out below (35) this free variable accounts for the fact that Montague assumes that the denotation of a formula depends on the world index at which the formula is interpreted.

Having introduced the basic translation of words, we can consider the translation of syntactically complex entities, i.e., phrases. In PTQ, (intensional) functional application is the basic semantic operation for constructing the translation of some syntactically complex structure. The intensional functional application of two terms, $\phi_{(s\tau)\tau'}$ and ψ_τ, is the term $\phi(\hat{}\psi)$ — or its Ty2 translate $\phi(\lambda @.\psi)$. This operation will play an important role in Lexicalized Flexible Ty2, because we assume that the translation of the mother node in a local tree is always the intensional functional application of the translation of the daughters.

The fragment that we consider in Part I of this thesis, is relatively small, and we will concentrate on scope ambiguities such as those illustrated in the sentences in (39). In (39), we give the relevant sentences together with the possible semantic representations that we assume for these sentences.

(39) a. Every man walks.
$\forall x_{se}[\mathsf{man}'_@(x_@) \rightarrow \mathsf{walk}'_@(x_@)]$
b. Every man loves some woman.
$\forall\exists$-reading:
$\forall x_{se}[\mathsf{man}'_@(x_@) \rightarrow \exists y_{se}[\mathsf{woman}'_@(x_@) \wedge \mathsf{love}'_@(x_@, y_@)]]$
$\exists\forall$-reading:
$\exists y_{se}[\mathsf{woman}'_@(x_@) \wedge \forall x_{se}[\mathsf{man}'_@(x_@) \rightarrow \mathsf{love}'_@(x_@, y_@)]]$
c. Every man believes that some woman walks.
de dicto reading:
$\forall x_{se}[\mathsf{man}'_@(x_@) \rightarrow \mathsf{believe}'_@(x_@, \lambda @.\exists y_{se}[\mathsf{woman}'_@(y_@) \wedge \mathsf{walk}'_@(y_@)])]$
$\forall\exists$-*de re* reading:
$\forall x_{se}[\mathsf{man}'_@(x_@) \rightarrow \exists y_{se}[\mathsf{woman}'_@(y_@) \wedge \mathsf{believe}'_@(x_@, \lambda @.\mathsf{walk}'_@(y_@))]]$
$\exists\forall$-*de re* reading:
$\exists y_{se}[\mathsf{woman}'_@(y_@) \wedge \forall x_{se}[\mathsf{man}'_@(x_@) \rightarrow \mathsf{believe}'_@(x_@, \lambda @.\mathsf{walk}'_@(y_@))]]$

As can be seen from these examples, we assume that sentences with more than one quantifier can, in principle, be ambiguous. In (39b) the two quantifiers are clause mates. In this case, we assume that either order of the quantifiers is possible in the translation of the sentence. We refer to the two readings by the order of the quantifiers, i.e., we talk about the ∀∃-reading in case the universal quantifier outscopes the existential quantifier, and about the ∃∀-reading, if the inverse scope relation holds.

In (39c) the existential quantifier is part of the embedded clause. The first reading given for this sentence will be referred to as the *de dicto* reading. Under this reading, the existential quantifier is in the scope of the intensionality operator, $\lambda @$, which corresponds to Montague's "ˆ" operator. Sentences like (39c) are, however, assumed to have more readings: The existential quantifier is able to take its scope outside the clause that it occurs in. The second and the third readings given in (39c) are both so-called *de re* readings, i.e., the existential quantifier is not in the scope of the intensionality operator, $\lambda @$. In the representation of readings where the existential quantifier outscopes the intensionality operator, it can either take wide or narrow scope with respect to the universal quantifier. To differentiate between the two *de re* readings, we add the relative scope of the universal and the existential quantifier, i.e., we write ∀∃-*de re* for the second reading given for sentence (39c), and ∃∀-*de re* for the third reading.

In the following, we will derive the given semantic representations for these sentences in a step by step manner, introducing thus the essential parts of the framework of Lexicalized Flexible Ty2 (LF-Ty2).

For LF-Ty2, we assume the following overall architecture: The *basic translation* of each word is given by the function $\leadsto$, which maps each word to a term of Ty2. In Table 1.1 we have given some values of this function. In addition to the basic translation, there are *derived translations*. We assume that there is a small set of *type shifting operations* which can be recursively applied to the basic translation of a word. In this thesis, we will only be concerned with two such operations, *argument raising* (AR) and *value raising* (VR).[23] Thus, in LF-Ty2, we assume that the semantic representation of a word is its basic translation or the result of a finite number of applications of the operations AR and VR to it. For phrases, we simply assume that the semantic representation of the mother is the intensional functional application of the semantic representations of the daughters.

It is the availability of type shifting operations that marks the major difference between Hendriks' flexible system and original Montague Grammar. As far as our examples in (39) are concerned, the derivation of some readings does not involve any type shifting operation, whereas we need the combination of AR and VR for others. The first is the case if we want to derive the right representation for sentence (39a) and the *de dicto* reading of sentence (39c). AR will be needed to derive the readings of example (39b). For the two *de re* readings of example (39c), the operation VR is also required, in combination with AR.

This characterization explains why we call the resulting system Lexicalized Flexible Ty2: Ty2 is the semantic representation language we are using. The translation of words is flexible, due to the possibility of applying shifting operations, AR and VR in our case. Finally, the flexibility is lexicalized, as type shifting operations may only be applied to the semantic representation of words.

Having said all this, we can look at the derivation of sentence (39a) as given in Figure 1.10 (page 47). For the terminal nodes in the tree, we give the basic translation as provided by the function $\leadsto$, and as indicated in Table 1.1. At the phrasal nodes, we state the term that represents the intensional functional application of the semantic representations of the daughters and also the term that is derived from it by λ-conversion (β-reduction).[24]

Consider first the NP *every man*. The basic translation of the determiner is as given in Table 1.1, $\lambda P_{s((se)t)} \lambda Q_{s((se)t)}.\forall x_{se}[P_{@}(x) \rightarrow Q_{@}(x)]$. The basic translation of the noun *man* is $\lambda x_{se}.\mathsf{man}'_{@}(x_{@})$. To compute the translation of the NP, we combine these two basic translations by intensional functional application. To do this, we use the translation of the determiner as the functor, and prefix the translation of the noun with the intensionality operator $\lambda @$. The intensional functional application, thus results in the term $\lambda P_{s((se)t)} \lambda Q_{s((se)t)}.\forall x_{se}[P_{@}(x) \rightarrow Q_{@}(x)](\lambda @ \lambda x_{se}.\mathsf{man}'_{@}(x_{@}))$. This term is given as the lower term at the NP node in the tree in Figure 1.10. To simplify this term, we can execute a series of λ-conversions (also called β-reductions) to arrive at the upper Ty2 term given for the NP node. We indicate the (recursive) application of λ-conversion in figures by the symbol "$\uparrow \lambda$". In (40) we give the step by step application of λ-conversion needed to arrive at this upper term.[25]

[23]In Hendriks 1993, a third operation, *argument lowering* (AL) is given to account for the *de re* readings with intensional verbs, i.e., for the *de re* reading of sentences such as (i) (see Hendriks 1993 (pp. 83f.) for further examples).

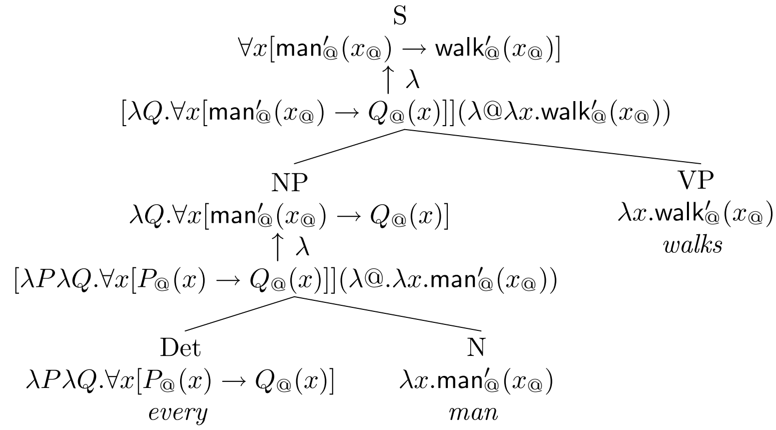
(i) A man is missing. $\exists x_{se}[\mathsf{man}'_{@}(x_{@}) \wedge \mathsf{be\text{-}missing}'_{@}(\lambda @ \lambda P_{s((se)t)}.P_{@}(x))]$

In Dekker 1993 (p. 91), a further operation, *devision of the i-th argument*, is introduced to account for dynamic effects, i.e., for the extension of scope across sentences.

[24]In the trees in this thesis, we use the label V for a verb or a verbal projection that needs a subject and at least one further complement. The label VP is used for a verb or a verbal projection that is missing the subject. S is the label used for saturated verbs or verbal projections. $\bar{\text{S}}$ is used for saturated verbal projections that contain a complementizer. This means that the labels indicate the degree of syntactic saturation, not whether the node dominates a word or a phrase.

[25]Remember from (36) that we use a subscript notation for the variable @, i.e., the term $P_{@}(x)$ is short for $P(@)(x)$.

FIGURE 1.10. The structure of the sentence *Every man walks.*:



(40) $[\lambda P_{s((se)t)}\lambda Q_{s((se)t)}.\forall x_{se}[P(@)(x) \rightarrow Q(@)(x)]](\lambda @\lambda x_{se}.\mathsf{man}'_{@}(x_{@}))$
$= \lambda Q_{s((se)t)}.\forall x_{se}[(\lambda @\lambda x_{se}.\mathsf{man}'_{@}(x_{@}))(@)(x) \rightarrow Q(@)(x)]$
$= \lambda Q_{s((se)t)}.\forall x_{se}[(\lambda x_{se}.\mathsf{man}'_{@}(x_{@}))(x) \rightarrow Q_{@}(x)]$
$= \lambda Q_{s((se)t)}.\forall x_{se}[\mathsf{man}'_{@}(x_{@}) \rightarrow Q(@)(x)]$

To arrive at the translation of the S node, we must combine the basic translation of the verb *walk* with the translation of the subject NP. The basic translation of the verb is a term of type $(se)t$. As such, it is of the appropriate type to function as the argument to the translation of the subject in functional application. In (41), we show the step by step λ-conversion abbreviated by the symbol "$\uparrow \lambda$ " in Figure 1.10.

(41) $[\lambda Q_{s((se)t)}.\forall x_{se}[\mathsf{man}'_{@}(x_{@}) \rightarrow Q(@)(x)]](\lambda @\lambda x_{se}.\mathsf{walk}'_{@}(x_{@}))$
$= \forall x_{se}[\mathsf{man}'_{@}(x_{@}) \rightarrow (\lambda @\lambda x_{se}.\mathsf{walk}'_{@}(x_{@}))(@)(x)]$
$= \forall x_{se}[\mathsf{man}'_{@}(x_{@}) \rightarrow (\lambda x_{se}.\mathsf{walk}'_{@}(x_{@}))(x)]$
$= \forall x_{se}[\mathsf{man}'_{@}(x_{@}) \rightarrow \mathsf{walk}'_{@}(x_{@}))]$

We can also derive the *de dicto* reading of sentence (39c) in a similarly straightforward way. We assume that the basic translation of the complementizer *that* is the term $\lambda p_{st}.p(@)$. In Figure 1.11 on page 48 we show the derivation of this reading. In the figure, we have abbreviated the derivation of the NP *every man*, as it is just like given in Figure 1.10.

The translation of the embedded S node is parallel to that of sentence (39a) as given in Figure 1.10: The determiner *some* combines with the noun *woman* via intensional functional application, and this NP combines with the basic translation of the verb *walk* via intensional functional application. To compute the translation of the $\bar{\text{S}}$ node, we apply intensional functional application to the basic translation of the complementizer and the translation of the embedded S node. As indicated in the figure, the basic translation of the complementizer is chosen in such a way that it does not make any proper contribution to the translation, rather, it expresses an identity function with respect to functional application.

The translation of the embedded clause is an expression of type t. Thus, it can serve as an argument for the basic translation of the matrix *believe*. What is crucial in the translation of the VP is that the intensionality operator which is introduced by intensional functional application is conserved in the translation of the VP, i.e., the semantic argument of the constant $\mathsf{believe}'_{@}$ that corresponds to the complement clause is of type st.

To obtain the translation of the overall sentence, the translation of the VP combines with that of the subject NP *every man* in exactly the way we have seen in Figure 1.10.

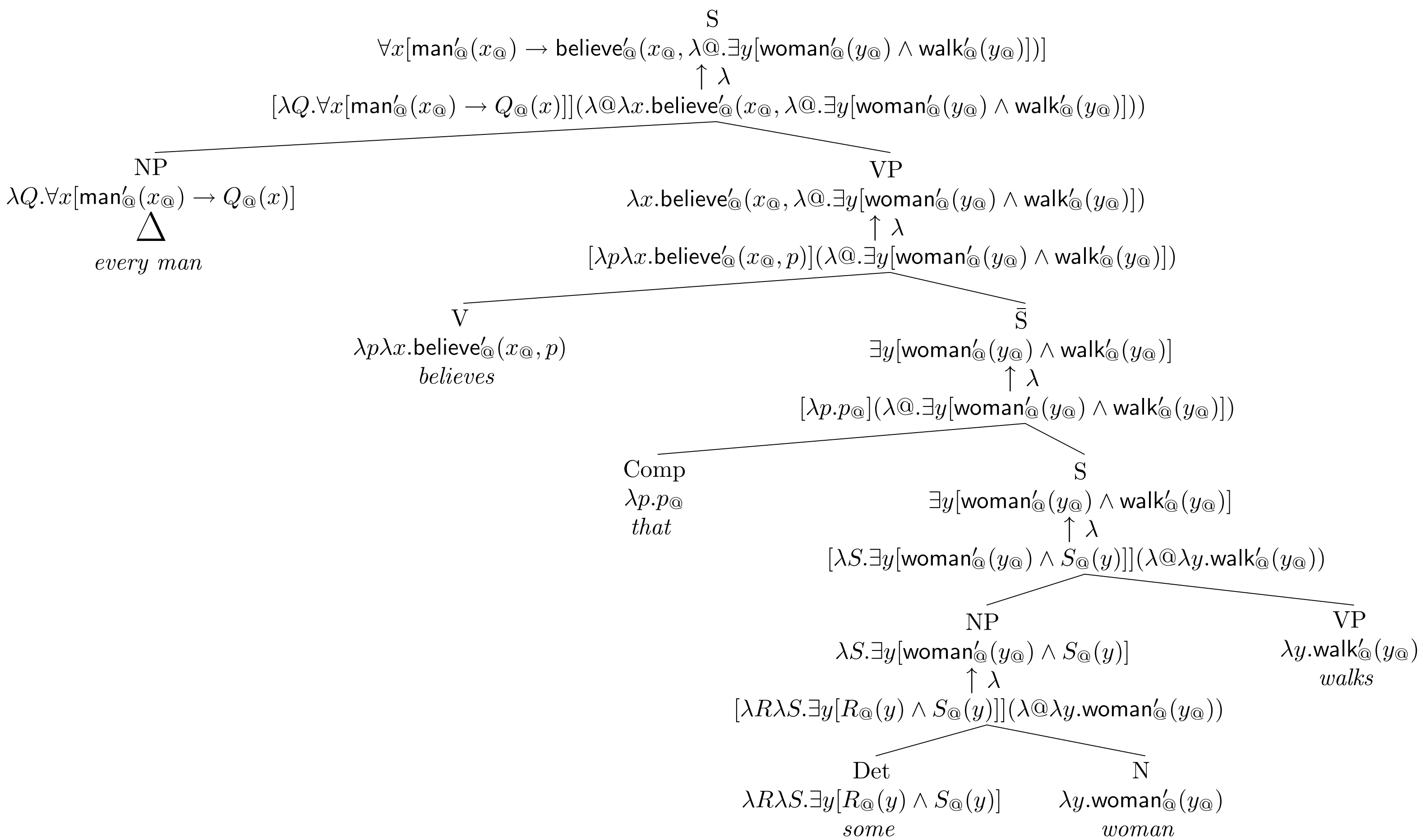


FIGURE 1.11. The *de dicto* reading of the sentence *Every man believes that some woman walks.*:

For the second example, given in (39b), we assume the basic translation of the words as in Table 1.1. The syntactic structure of the sentence is indicated by the bracketing in (42).

(42) $[_S\ [_{NP}$ Every man$]\ [_{VP}$ loves $[_{NP}$ some woman$]]]$.

It is clear that we cannot derive any interpretation of this sentence by the basic translations of the words and intensional functional application alone: The NP *some woman* is of type $(s((se)t))t$ just as the NP *every man* in Figure 1.10. The basic translation of the transitive verb, however, is of type $(se)((se)t)$. These terms cannot be combined by intensional functional application, because if the NP were the functor, the verb should be of type $(se)t$ — which it is not; and if the verb were the functor, the NP should be of type se. This indicates that the basic translation of at least one of the words in the sentence needs to be type-shifted. In fact, we will apply *argument raising* (AR) to the basic translation of the verb to give it the representation $\lambda Y_{(s((se)t))t}\lambda x_{se}.Y_{@}(\lambda @\lambda y_{se}.\mathsf{love}'_{@}(x_{@}, y_{@}))$. As indicated by the type of the variable Y, this term can be used as a functor which takes the translation of the NP *some woman* as its argument.

In Definition 1.10 we define the operation *argument raising* after Hendriks 1993 (p. 75). As for predicates which have more than one semantic argument, any of the arguments can be raised, we define an operation AR_i which raises the ith argument.

DEFINITION 1.10 *(ARi)*

For each $i \in \mathbb{N}$, AR_i *is a relation between two terms* α *and* β *such that*

if α *is of some type* $(a_1(\ldots((sa_i)(\ldots(a_n b)\ldots)))))$
then β *is some term*
$\lambda x_{a_1,1}\ldots\lambda X_{s((s((sa_i)b))b),i}\ldots\lambda x_{a_n,n}.X(@)(\lambda @\lambda x_{sa_i,i}.\alpha(x_1)\ldots(x_i)\ldots(x_n))$

In the following, we use operational terminology. If a pair $\langle\alpha,\beta\rangle$ is in the relation AR_i, then we say that we *obtain* β *from* α *by applying* AR_i. In this case, we write $\alpha \longrightarrow_{AR_i} \beta$.

In the text before Definition 1.10 we have said that the basic translation of the verb *loves* is mapped into the term $\lambda Y_{(s((se)t))t}\lambda x_{se}.Y_{@}(\lambda @\lambda y_{se}.\mathsf{love}'_{@}(x_{@}, y_{@}))$ by the operation AR. To see that this is the case, we must take $i = 1$, i.e., we have an instance of AR_1. The type a_i is the type e, b the type t, and $n = 2$, where a_n is the type se. As the basic translation of the verb *walk* is of type $(se)((se)t)$ $(= (sa_i)a_n)$, the operation AR_1 relates the basic translation of the verb to some other term. In (43) we give the basic translation of the verb in (a) and show the term obtained by applying AR_1 to it in (b). As the obtained term can be further reduced by λ-conversion, we give the conversion steps as well.

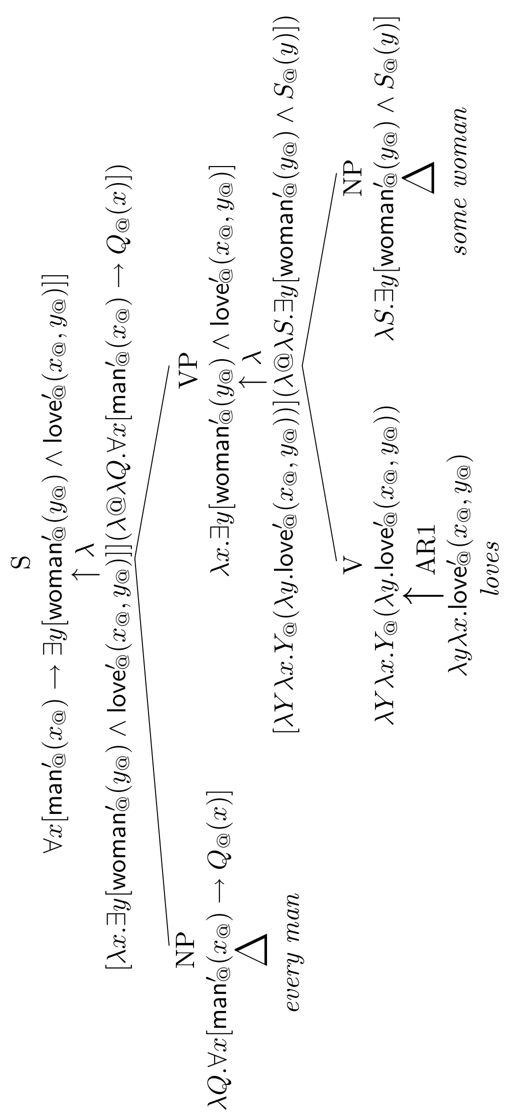
(43) a. $\alpha = \lambda y_{se}\lambda x_{se}.\mathsf{love}'_{@}(x_{@}, y_{@})$
b. $\longrightarrow_{AR1} \lambda Z_{s((s((se)t))t)}\lambda u_{se}.Z(@)(\lambda @\lambda z_{se}.\alpha(z)(u))$
$= \lambda Z_{s((s((se)t))t)}\lambda u_{se}.Z(@)(\lambda @\lambda z_{se}.[\lambda y_{se}\lambda x_{se}.\mathsf{love}'_{@}(x_{@}, y_{@})](z)(u))$
$= \lambda Z_{s((s((se)t))t)}\lambda u_{se}.Z(@)(\lambda @\lambda z_{se}.\mathsf{love}'_{@}(u_{@}, z_{@}))$

Having defined the operation AR, we can account for the different readings of example (39b), repeated below.

(44) Every man loves some woman.
∀∃-reading: $\forall x_{se}[\mathsf{man}'_{@}(x_{@}) \rightarrow \exists y_{se}[\mathsf{woman}'_{@}(y_{@}) \wedge \mathsf{love}'_{@}(x_{@}, y_{@})]]$
∃∀-reading: $\exists y_{se}[\mathsf{woman}'_{@}(y_{@}) \wedge \forall x_{se}[\mathsf{man}'_{@}(x_{@}) \rightarrow \mathsf{love}'_{@}(x_{@}, y_{@})]]$

In Figure 1.12 (page 50), we show how the narrow scope reading of the existential quantifier is derived. For convenience, we include the application of type shifting operations in the figure. The result of applying AR to the first semantic argument of the basic translation of the verb is a term which is able to combine with the translation of the NP *some woman*.

FIGURE 1.12. The ∀∃ reading of the sentence *Every man loves some woman*:



We indicate the application of AR to the first semantic argument by the annotation "↑AR1 " in the tree. The semantic annotation at the phrasal nodes, just as in Figure 1.10, consists of two lines: the lower line is the intensional functional application of the semantic representations of the daughters; the upper line is the fully β-reduced equivalent of this term.

The two NPs *every man* and *some woman* are translated just as we have seen in the previous examples. The verb *love* is, however, type shifted in order to become compatible with the semantic type of its direct argument. In (45) we show how the semantic representation of the VP node is derived by intensional functional application of the derived translation of the verb and the translation of the NP *some woman*.

(45) $[\lambda Y \lambda x.Y_{@}(\lambda @ \lambda y.\mathsf{love}'_{@}(x_{@}, y_{@}))](\lambda @ \lambda S.\exists y[\mathsf{woman}'_{@}(y_{@}) \wedge S_{@}(y)])$
$= \lambda x.(\lambda @ \lambda S.\exists y[\mathsf{woman}'_{@}(y_{@}) \wedge S_{@}(y)])(@)(\lambda @ \lambda y.\mathsf{love}'_{@}(x_{@}, y_{@}))$

$= \lambda x.(\lambda S.\exists y[\mathsf{woman}'_@(y_@) \wedge S_@(y)])(\lambda @\lambda y.\mathsf{love}'_@(x_@, y_@))$
$= \lambda x.\exists y[\mathsf{woman}'_@(y_@) \wedge (\lambda @\lambda y.\mathsf{love}'_@(x_@, y_@))(@)(y)]$
$= \lambda x.\exists y[\mathsf{woman}'_@(y_@) \wedge (\lambda y.\mathsf{love}'_@(x_@, y_@))(y)]$
$= \lambda x.\exists y[\mathsf{woman}'_@(y_@) \wedge \mathsf{love}'_@(x_@, y_@)]$

Now that the translation of the VP is calculated, the translation of the entire sentence is achieved by intensional functional application, taking the translation of the subject NP as the functor. This example has illustrated that via the operation AR, we can make the verb a functor that takes a quantified NP as its argument, i.e., we raise the type of an argument of a verb from *se* (the intension of an individual) to the type $s((s((se)t))((s((se)t))t))$, the intension of a quantified NP.

The same operation can be used to derive the other reading of sentence (39b), i.e., the wide scope reading of the existential quantifier. In this case, we must apply AR to both semantic arguments of the verb. This is indicated in Figure 1.13 (page 52).

In the derivation of this reading we apply AR first to the second semantic argument of the verb and then to the first argument. As can be seen, the order of application of AR determines which quantifier takes wide scope.[26] To make the derivation more transparent, we give the β-reduction steps for the VP node and the S node in (46).

(46) a. The VP node:
$[\lambda Y \lambda X.Y_@(\lambda @\lambda y.X_@(\lambda @\lambda x.\mathsf{love}'_@(x_@, y_@)))]$
$\quad (\lambda @\lambda S.\exists y[\mathsf{woman}'_@(y_@) \wedge S_@(y)])$
$= \lambda X.(\lambda @\lambda S.\exists y[\mathsf{woman}'_@(y_@) \wedge S_@(y)])(@)$
$\quad (\lambda @\lambda y.X_@(\lambda @\lambda x.\mathsf{love}'_@(x_@, y_@)))$
$= \lambda X.(\lambda S.\exists y[\mathsf{woman}'_@(y_@) \wedge S_@(y)])(\lambda @\lambda y.X_@(\lambda @\lambda x.\mathsf{love}'_@(x_@, y_@)))$
$= \lambda X.\exists y[\mathsf{woman}'_@(y_@) \wedge (\lambda @\lambda y.X_@(\lambda @\lambda x.\mathsf{love}'_@(x_@, y_@)))(@)(y)]$
$= \lambda X.\exists y[\mathsf{woman}'_@(y_@) \wedge (\lambda y.X_@(\lambda @\lambda x.\mathsf{love}'_@(x_@, y_@)))(y)]$
$= \lambda X.\exists y[\mathsf{woman}'_@(y_@) \wedge X_@(\lambda @\lambda x.\mathsf{love}'_@(x_@, y_@))]$

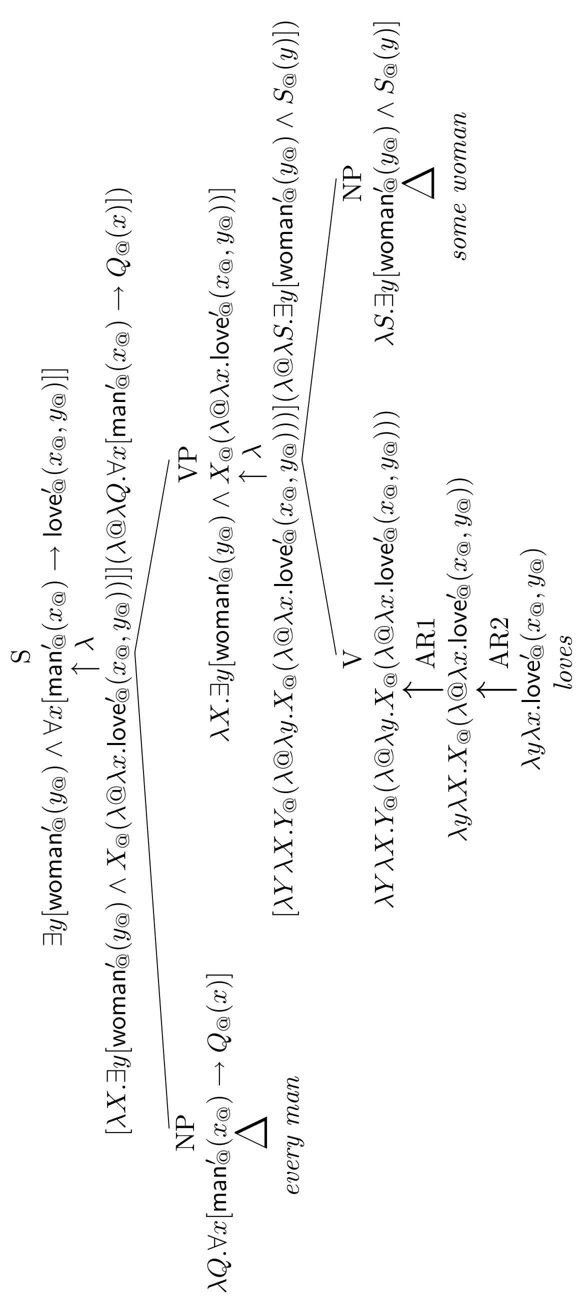
b. The S node:
$[\lambda X.\exists y[\mathsf{woman}'_@(y_@) \wedge X_@(\lambda @\lambda x.\mathsf{love}'_@(x_@, y_@))]]$
$\quad (\lambda @\lambda Q.\forall x[\mathsf{man}'_@(x_@) \rightarrow Q_@(x)])$
$= \exists y[\mathsf{woman}'_@(y_@) \wedge (\lambda @\lambda Q.\forall x[\mathsf{man}'_@(x_@) \rightarrow Q_@(x)])(@)$
$\quad (\lambda @\lambda x.\mathsf{love}'_@(x_@, y_@))]$
$= \exists y[\mathsf{woman}'_@(y_@) \wedge (\lambda Q.\forall x[\mathsf{man}'_@(x_@) \rightarrow Q_@(x)])$
$\quad (\lambda @\lambda x.\mathsf{love}'_@(x_@, y_@))]$
$= \exists y[\mathsf{woman}'_@(y_@) \wedge \forall x[\mathsf{man}'_@(x_@) \rightarrow (\lambda @\lambda x.\mathsf{love}'_@(x_@, y_@))(@)(x)]]$
$= \exists y[\mathsf{woman}'_@(y_@) \wedge \forall x[\mathsf{man}'_@(x_@) \rightarrow (\lambda x.\mathsf{love}'_@(x_@, y_@))(x)]]$
$= \exists y[\mathsf{woman}'_@(y_@) \wedge \forall x[\mathsf{man}'_@(x_@) \rightarrow \mathsf{love}'_@(x_@, y_@)]]$

The semantic representation of the VP is very illuminating: it can be seen that the variable X occurs in the scope of the existential quantifier. As X is of the type of the intension of a quantified NP, once the VP combines with such an NP, the quantifier of this NP ends up in the scope of the existential quantifier. This is actually the case in the translation of the S node, as indicated in (46b).

These examples should suffice to indicate that the operation AR allows us to derive all hypothetical scope relations that exist between semantic co-arguments of a predicate. There are, however, not only scopal interactions among co-arguments of the same predicate. An example of this more complicated case was given in (39c). In (47) we repeat example (39c), together with the two *de re* readings.

[26] In that sense, AR can be said to have the same effect we find for QR in the framework of transparent LF.

FIGURE 1.13. The ∃∀ reading of the sentence *Every man loves some woman*:



(47) Every man believes that some woman walks.

$\forall x_{se}[\mathsf{man}'_{@}(x_{@}) \rightarrow \exists y_{se}[\mathsf{woman}'_{@}(y_{@}) \wedge \mathsf{believe}'_{@}(x_{@}, \lambda @.\mathsf{walk}'_{@}(y_{@})]$

$\exists y_{se}[\mathsf{woman}'_{@}(y_{@}) \wedge \forall x_{se}[\mathsf{man}'_{@}(x_{@}) \rightarrow \mathsf{believe}'_{@}(x_{@}, \lambda @.\mathsf{walk}'_{@}(y_{@})]$

In these readings, the existential quantifer which originates in the embedded clause interacts scopally with some argument of the matrix predicate. As these readings are available, it is necessary to provide some means of allowing scope interaction between non-co-arguments as well. In Hendriks 1993 the operation *value raising* (VR) is given for this purpose. We can derive the *de re* readings by an interaction of VR and AR.

Before we can turn to the derivation of these readings, we define the operation VR. Definition 1.11 is modelled after the definition given in Hendriks 1993 (p. 75).

DEFINITION 1.11 *(VR)*

For each type $d \in Type$,

VR_d is a relation between two terms α and β such that

> *if α is of some type $a_1(\ldots(a_n b)\ldots)$.*
> *then β is some term $\lambda x_{a_1,1}\ldots\lambda x_{a_n,n}\lambda u_{s((sb)d)}.u(@)(\lambda @.\alpha(x_1)\ldots(x_n))$.*

Whereas AR only changes the semantic type of one argument of a predicate, VR adds another argument. The simplest instance of VR is the mapping of an individual constant such as m_e to the term $\lambda P_{s((se)t)}.P(@)(\lambda @.m_e)$, which is the Ty2 equivalent of the PTQ translation of the name *Mary* ($\lambda P.\check{}P(\hat{}m)$). For this mapping, there is no x_i, b is the type e, and d is the type t. Thus, from the term m_e, we obtain the term $\lambda P.P_{@}(\lambda @.m)$ by VR$_t$. Following our conventions for shifting operations, we write $m_e \longrightarrow_{VR} \lambda P.P_{@}(\lambda @.m)$.

As indicated in Figure 1.14 (page 54), we must perform type shifting operations to the basic translation of the verbs *walks*, *believes*, and of the complementizer *that*. The application of VR to the basic translation of the verb *walks* creates a new semantic argument slot for this verb. The new argument is of the type of a predicate that takes a sentential complement. In effect, this new argument marks the place where the constant $\mathsf{believe}'_{@}$ is going to appear. Now that we made the matrix predicate a semantic co-argument of the embedded subject, we have created an escape hatch for the subject of the embedded clause. By applying AR to the first semantic argument of the clause, we ensure that the matrix predicate will end up in the scope of the embedded subject. This becomes clear in the semantic representation of the embedded S node: there, the variable u occurs in the scope of the existential quantifier.

The semantic representation of the embedded S node is of the type $(s((st)t))t$. The basic translation the complementizer is $\lambda p_{st}.p(@)$, i.e., a term of type $(st)t$. Intuitively, we want the complementizer *that* to denote the identity function. To achieve this effect in our architecture, the complementizer must undergo exactly the same type shifting operations, which we performed on the main verb of the clause, i.e., we must first apply VR and then raise the first semantic argument, just as shown in Figure 1.14.[27] In (48) we indicate how the translation of the $\bar{\text{S}}$ node is achieved in a step by step manner, by using the derived translation of the complementizer as the functor.

(48) $[\lambda\mathcal{P}\lambda u.\mathcal{P}_{@}(\lambda @\lambda p.u_{@}(\lambda @.p_{@}))](\lambda @\lambda u.\exists y[\mathsf{woman}'_{@}(y_{@}) \wedge u_{@}(\lambda @.\mathsf{walk}'_{@}(y_{@}))])$
$= \lambda u.(\lambda @\lambda u.\exists y[\mathsf{woman}'_{@}(y_{@}) \wedge u_{@}(\lambda @.\mathsf{walk}'_{@}(y_{@}))])(@)(\lambda @\lambda p.u_{@}(\lambda @.p_{@}))$
$= \lambda u.(\lambda u.\exists y[\mathsf{woman}'_{@}(y_{@}) \wedge u_{@}(\lambda @.\mathsf{walk}'_{@}(y_{@}))])(\lambda @\lambda p.u_{@}(\lambda @.p_{@}))$

[27] Alternatively, we could apply VR twice to the verb *walks*. In this case, the translation of the S node would be the functor that takes the complementizer as its argument.

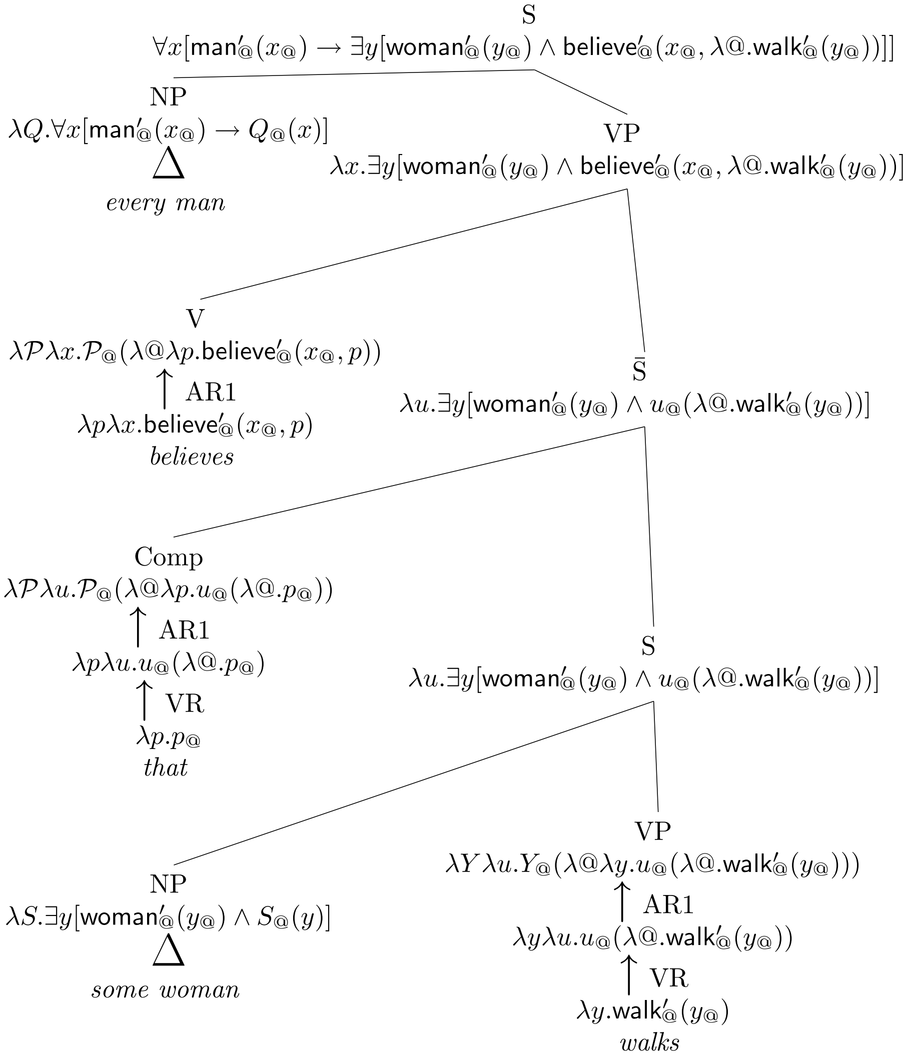
(i) *walks* $\rightsquigarrow$ $\lambda y.\mathsf{walk}'_{@}(y_{@})$
$\longrightarrow_{VR}$ $\lambda y\lambda u'.u'_{@}(\lambda @.\mathsf{walk}'_{@}(y_{@}))$
$\longrightarrow_{VR}$ $\lambda y\lambda u'\lambda u.u_{@}(\lambda @.u'_{@}(\lambda @.\mathsf{walk}'_{@}(y_{@})))$
$\longrightarrow_{AR1}$ $\lambda Y\lambda u'\lambda u.Y_{@}(\lambda @.u_{@}(\lambda @.u'_{@}(\lambda @.\mathsf{walk}'_{@}(y_{@}))))$

Combining this with the embedded subject results in:

(ii) $\lambda u'\lambda u.\exists y[\mathsf{woman}'_{@}(y_{@}) \wedge u_{@}(\lambda @.u'_{@}(\lambda @.\mathsf{walk}'_{@}(y_{@})))]$

This can take the basic translation of the complementizer as its argument and yields the term that we gave as the translation of the $\bar{\text{S}}$ node in Figure 1.14.

FIGURE 1.14. The ∀∃-*de re* reading of the sentence *Every man believes that some woman walks.*:



$$
\begin{aligned}
&= \lambda u.\exists y[\mathsf{woman}'_@(y_@) \wedge (\lambda @ \lambda p.u_@(\lambda @.p_@))(@)(\lambda @.\mathsf{walk}'_@(y_@))] \\
&= \lambda u.\exists y[\mathsf{woman}'_@(y_@) \wedge (\lambda p.u_@(\lambda @.p_@))(\lambda @.\mathsf{walk}'_@(y_@))] \\
&= \lambda u.\exists y[\mathsf{woman}'_@(y_@) \wedge u_@(\lambda @.(\lambda @.\mathsf{walk}'_@(y_@))(@))] \\
&= \lambda u.\exists y[\mathsf{woman}'_@(y_@) \wedge u_@(\lambda @.\mathsf{walk}'_@(y_@))]
\end{aligned}
$$

The β-reduced term in (48) is identical (up to the names of bound variables) to the translation of the S node. In this translation, the existential quantifier has scope over an occurrence of the variable u. As this variable corresponds to the matrix predicate, we have derived a *de re* semantic representation of the embedded clause.

The basic translation of the verb *believe* asks for a semantic argument of type st. However, the translation of the complement clause is, of a higher type. Therefore, we must apply AR1 once more, now to the basic translation of *believe*. In the translation of the matrix VP, the variable u from the embedded clause is finally replaced by the matrix predicate and it can be seen that the existential quantifier has wide scope over the propositional argument of $\mathsf{believe}'$. The translation of the VP combines with that of the matrix subject NP by intensional functional application to give the translation of the entire sentence.

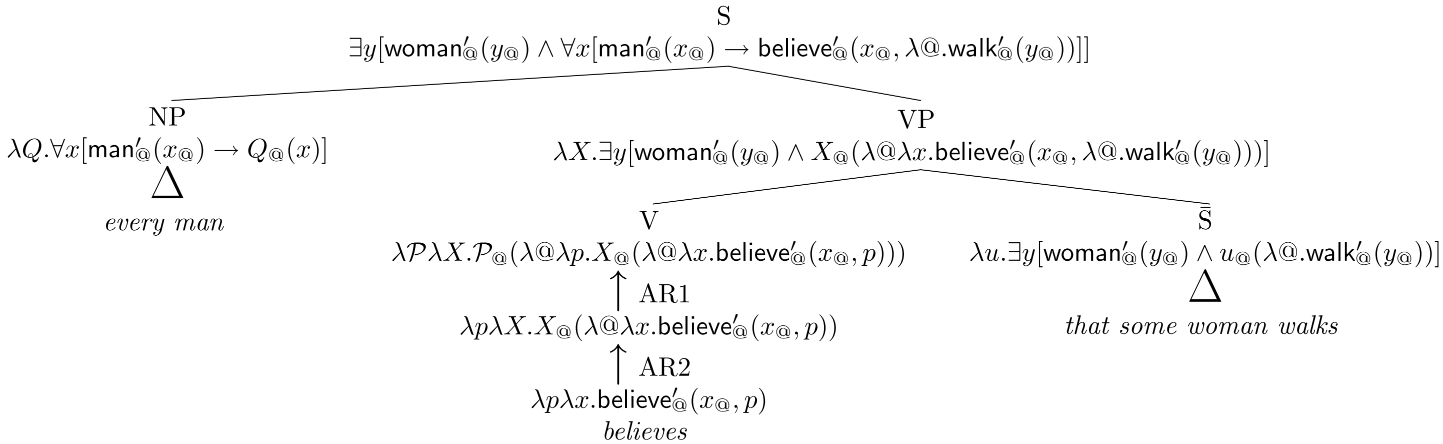
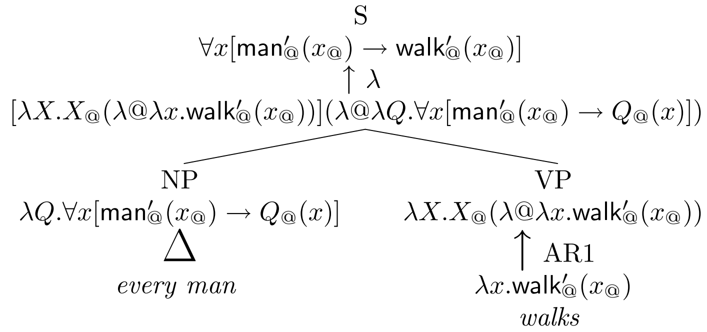


FIGURE 1.15. The $\exists\forall$-*de re* reading of the sentence *Every man believes that some woman walks.*:

FIGURE 1.16. Alternative structure of the sentence *Every man walks.*:



In (47) we have seen that there are in fact two *de re* readings for the sentence. In Figure 1.14 we have given the derivation of the reading in which the existential quantifier is in the scope of the universal quantifier contributed by the matrix subject. The other *de re* reading is such that the existential quantifier has wide scope over the universal quantifier as well. In Figure 1.15 we give the derivation of this reading. We left out the structure of the embedded clause in the figure, because the translation of the complement clause is the same for both readings.

The derivations differ only with respect to the translation of the matrix verb. In the case of the $\forall\exists$ reading, the second semantic argument of the verb was not raised. To derive the $\exists\forall$ reading, we must raise the second semantic argument first and, then, raise the first semantic argument, just as in the case of the $\exists\forall$ reading of sentence (39b) in Figure 1.13.

With this brief consideration of the $\exists\forall$ reading of sentence (39c), our presentation of the mechanisms of LF-Ty2 comes to an end. We have confined our attention to scope variation and it was shown how all the hypothetically possible readings can be derived. In our presentation, we have left aside other type shifting operations such as *argument lowering* (Hendriks, 1993) and operations needed to cover trans-sentential phenomena (Dekker, 1993). Furthermore, we have not addressed coordination, which plays a central role in Hendriks 1993. For all these cases, it is, however, straightforward to extend the present system to integrate the additional operations and the required basic translations for the coordinating particles. In the remainder of this section, we will address the question of how we can impose constraints in LF-Ty2. Within this framework, a constraint may either reduce the number of derivations of a certain logical form, or it may reduce the number of distinct logical forms that may be derived. In the latter case we will speak of a reduction of the number of readings.[28]

Now that we have illustrated some derivations, we can address a property of flexible systems such as FMG or LF-Ty2 which might be undesirable for many HPSGians: The system LF-Ty2, just as FMG, generates a number of different derivations which lead to the same readings. To see a simple example, let's review sentence (39a), repeated in (49).

(49) Every man walks.

In the derivation of the translation of this sentence in Figure 1.10 (page 47), we have simply combined the basic translations of the words. Instead, we could of course, have applied AR to the semantic argument of the verb. The derived translation of the verb would be $\lambda X_{s(s((se)t))}.X_@(\lambda @ \lambda x_{se}.\mathsf{walk}'_@(x_@))$. This derived translation of the verb can

[28]We use "reading" as synonymous for logical form. As noted in the context of example (27), a sentence may have several distinct logical forms which have the same denotation.

combine with the translation of the subject NP. In Figure 1.16 the tree for this derivation is given. The difference to the derivation that we have given in Figure 1.10 is that, now, the verb is the functor, and the NP the argument. The resulting translation is of course the same. In (50) the step-by-step λ-conversion is given for the type-shifted verb.

(50) $[\lambda X_{s(s((se)t))}.X_@(\lambda @ \lambda x_{se}.\mathsf{walk}'_@(x_@))]$
$(\lambda @ \lambda Q_{s((se)t)}.\forall x[\mathsf{man}'_@(x_@) \rightarrow Q_@(x)])$
$= (\lambda @ \lambda Q_{s((se)t)}.\forall x[\mathsf{man}'_@(x_@) \rightarrow Q_@(x)])(@)(\lambda @ \lambda x_{se}.\mathsf{walk}'_@(x_@))$
$= (\lambda Q_{s((se)t)}.\forall x[\mathsf{man}'_@(x_@) \rightarrow Q_@(x)])(\lambda @ \lambda x_{se}.\mathsf{walk}'_@(x_@))$
$= \forall x[\mathsf{man}'_@(x_@) \rightarrow (\lambda @ \lambda x_{se}.\mathsf{walk}'_@(x_@))(@)(x)]$
$= \forall x[\mathsf{man}'_@(x_@) \rightarrow (\lambda x_{se}.\mathsf{walk}'_@(x_@))(x)]$
$= \forall x[\mathsf{man}'_@(x_@) \rightarrow \mathsf{walk}'_@(x_@)]$

As can be seen in the final line of (50), the translation of the overall sentence is the same, but the derivation of this translation is different. This is just one instance of what Bouma 1994 calls *spurious derivations*, i.e., derivations that do not differ in the translation that they ultimately lead to.

In much work within HPSG, an effort is made to avoid "spurious ambiguities". An example that is clearly related to the present one is the contrast between the analyses of scope ambiguity in Pollard and Yoo 1998 and Przepiórkowski 1998. Pollard and Yoo 1998 provide an analysis for quantifier scope within HPSG. They use a storage mechanism, just as done in Pollard and Sag 1994, but solve some of the empirical problems of the particular analysis of Pollard and Sag 1994.[29] The analysis of Pollard and Yoo 1998 introduces a large amount of spurious ambiguity. Przepiórkowski 1998 optimizes this account: He introduces severe restrictions on the nodes at which quantifier retrieval is possible (they must be words whose CONTENT value is not identical with that of one of their complements) and, thus, he eliminates the spurious ambiguities. These restrictions do not lead to an exclusion of some readings, but simply reduce the number of distinct analyses for a sentence.

Similarly, it can be shown that we can, at least, reduce the number of possible derivations in the flexible system considerably. Building on results from Hendriks 1993 (pp. 118–128), Bouma 1994 (p. 34) imposes a number of restrictions on the applicability of type shifting operations. These restrictions reduce the number of derivations, while leaving the number of derivable readings untouched. In fact, in Bouma's system there is only a finite number of derivations for each sentence. Bouma had some independent reason for trying to reduce the number of derivations, because he proposed a computer implementation that used FMG as the underlying semantic analysis. For this particular application, it is vital that all readings of a sentence can be found via a finite number of derivations.

One of Bouma's restrictions is that the shifting operations may only be applied to lexical items. We adopt this restriction, which makes our overall system *lexicalized.* Thus, lexicalization only leads to a reduction of the number of derivations, not to a loss of readings.

While we do not consider the existence of spurious derivations to be a severe problem, the result of Bouma 1994 is promising, as it indicates that the elimination of (some) spurious derivations is possible. We leave it to further research to formulate the constraints that would be needed to avoid spurious derivations for a particular grammar.

[29] The semantic analysis of Pollard and Sag 1994 fails to account for the *de dicto*-reading of sentence (i) as given in (ia). The reason for this is that the semantic contribution of the quantified NP *a unicorn* is introduced into the sentence by the nominative NP. Therefore, the quantifier must have scope over the matrix predicate according to this analysis. Empirically, however, the sentence as both readings given in (i).

(i) A unicorn appears to be approaching.
 a. *de dicto*: $\mathsf{appear}'_@(\lambda @.\exists x[\mathsf{unicorn}'_@(x_@) \wedge \mathsf{approach}'_@(x_@)])$
 b. *de re*: $\exists x[\mathsf{unicorn}'_@(x_@) \wedge \mathsf{appear}'_@(\lambda @.\mathsf{approach}'_@(x_@))]$

It was noted in Hendriks 1993 that his own system, FMG, licenses too many readings. The same is, of course, true for our system as it stands. While the problem of "spurious derivations" seems to be relevant only for practical applications, but not from a theoretical point of view, the existence of too many readings must be excluded by the grammar. In the following, we will discuss one example of a possible constraint on the numbers of readings. As this thesis is more concerned with idiosyncratic constraints on the occurrence of lexical items (see Part II), we will not discuss general constraints on possible logical forms in much detail. Thus, this one example should be considered as an illustration of how such constraints can be incorporated in principle.

It has been noted, among others, in Zimmermann 1993 and von Stechow 1993 that the scope properties of expressions as *every N* differ considerably from that of expressions of the form *some N*. This can be illustrated with the following example. We put the symbol "$" in front of readings that are not available for a given sentence.

(51) Some man believes that every woman walks.

a. *de dicto* reading:
$\exists x[\mathsf{man}'_@(x_@) \wedge \mathsf{believe}'_@(x_@, \lambda @.\forall y[\mathsf{woman}'_@(y_@) \rightarrow \mathsf{walk}'_@(y_@)])]$

b. $\exists\forall$-*de re* reading:
\$ $\exists x[\mathsf{man}'_@(x_@) \wedge \forall y[\mathsf{woman}'_@(y_@) \rightarrow \mathsf{believe}'_@(x_@, \lambda @.\mathsf{walk}'_@(y_@))]]$

c. $\forall\exists$-*de re* reading:
\$ $\forall y[\mathsf{woman}'_@(y_@) \rightarrow \exists x[\mathsf{man}'_@(x_@) \wedge \mathsf{believe}'_@(x_@, \lambda @.\mathsf{walk}'_@(y_@))]]$

Structurally, sentence (51) is similar to example (39c). Our mechanism of shifting allows us to derive the three readings indicated in (a)–(c), which are parallel to the readings attested for sentence (39c). In (51a) we have a *de dicto* reading. In (b) and (c) we give the logical forms of *de re* readings; in (51b), the universal quantifier has scope over the intensionality operator $\lambda @$, but is in the scope of the existential quantifier contributed by the matrix subject. In (51c) the universal quantifier contributed by the embedded subject has wide scope over all other operators in the logical form. The derivations of these terms are parallel to those indicated in Figure 1.11, Figure 1.14 and Figure 1.15 respectively.

The problem with the *de re* derivations is that these readings do not exist. In (51), this is indicated by the symbol "$" in front of the term. The contrast between the possible readings of (39c) and those of (51) is just one instance of the complicated interactions of quantifiers. For the purpose of this section, we will make the simplifying assumption that a universal quantifier cannot take scope outside the clause in which it is introduced, whereas an existential quantifier is not subject to this constraint.

In the derivations of the *de re* readings of sentence (39c), we have shown that in the logical form of the embedded clause, there is some variable u which corresponds to the semantic contribution of the matrix predicate. In (52a) we repeat the logical form of the S node in the derivation of the *de re* readings of sentence (39c). In (52b) we replace the existential quantifier of (52a) by a universal quantifier as it appears in the *de re* derivations for sentence (51).

(52) a. logical form for the *de re* reading of *some woman walks*:
$\lambda u.\exists y[\mathsf{woman}'_@(y_@) \wedge u_@(\lambda @.\mathsf{walk}'_@(y_@))]$

b. putative logical form for the *de re* derivation of *every woman walks*:
\$ $\lambda u.\forall y[\mathsf{woman}'_@(y_@) \wedge u_@(\lambda @.\mathsf{walk}'_@(y_@))]$

It is our task to exclude a logical form as (52b), while allowing logical forms such as (52a). As we have mentioned in the discussion of Figure 1.14, the *de re* reading is made possible by the application of VR which introduces a new semantic argument, u in this case, and, the subsequent application of AR. This combination of shifting rules results in a logical form in which a quantifier contributed by a complement of the embedded verb has in its scope a free occurrence of the variable u. To exclude *de re* readings, it is, therefore, enough to exclude free variables in the scope of a universal quantifier. In (53) we formulate a simple constraint on logical forms that has this effect.

(53) A simple lf constraint:
For each node of category S,
the logical form of S does not contain a subterm of the form $\lambda x.\psi$ such that ψ has a subterm $\forall v\phi$ which has a free occurrence of the variable x.

We can show that the logical form of the embedded S node of sentence (51) does not satisfy this constraint in the case of the *de re* readings. The relevant logical form is given in (52b). This term has a subterm $\lambda u.\psi$ such that ψ contains a universal quantifier, $\forall y$. In the scope of this quantifier, there is a free occurrence of the variable u. Thus, the logical form violates the lf constraint in (53).

The logical form given in (52a) satisfies the constraint, as this term does not contain any universal quantifier. Similarly, the logical form of the embedded clause of sentence (51) satisfies the constraint under the *de dicto* reading. In this case, the logical form of the embedded clause is as given in (54).

(54) logical form for the *de dicto* reading of *every woman walks*:
$\forall y[\mathsf{woman}'_{@}(y_{@}) \wedge \mathsf{walk}'_{@}(y_{@})]$

In this logical form, there is a universal quantifier. In addition, the variable @ does occur freely in the scope of the universal quantifier, but it is not bound by a λ-abstractor in the term. Therefore, the constraint is not violated.

The constraint in (53) is just one example of the kind of lf constraints that we expect in the present approach. The constraint links some syntactic properties of a sign, its syntactic category in this case, to some property of its logical form. Similarly, other constraints might express restrictions between the PHONOLOGY value of a sign and its logical form. Such constraints would be used to account for the influence of word order on the availability of certain readings.

The constraint in (53) can be expressed under the assumption of a Ty2 term as the logical form of a linguistic sign. It is, however, less obvious how such a constraint can be expressed within the frameworks presented in Section 1.2. Let us briefly consider why this is the case. The basic reason is that the constraint in (53) restricts the scope potential of the universal quantifier, while it leaves that of the existential quantifier untouched.

In a framework that assumes a level of LF, the scope of a quantifier is (at least partially) determined by QR. As we have seen, in May 1985 it is assumed that a quantified NP attaches to the S node of the clause which determines its scope domain. May assumes a single LF for clauses with multiple quantifiers independent of the relative scope of the quantifiers. However, in the case of the three readings of (39c), we would still need one LF for the *de dicto* reading and one LF for the two *de re* readings, because in the latter case, it is the matrix clause that constitutes the scope domain of the quantifier contributed by the subject of the embedded clause. This means that the operation of QR must allow quantifiers to raise over a clause boundary. If such a long QR is allowed, it remains uncertain how long QR can be parametrized to allow it for an existential quantifier, but disallow it for a universal quantifier. In order to express such parametrization, syntax must be able to have access to the difference between *some* and *every*. Whereas such a difference is obvious if we operate with terms of Ty2, it must be stipulated within an approach that uses syntactic categories as the basic entities of LFs.

A similar argument could be made against the transparent LFs of von Stechow 1993. There the issue of the differences in the scopal potential of quantifiers is addressed directly (pp. 63–65). Stechow claims that QR is always clause bound. He admits, though, that so far this restriction must be stipulated and cannot be derived from any assumption made within syntax proper. This, of course, weakens the argument for the existence of a level of LF which is a syntactic tree considerably. Once long QR is excluded, it is not obvious how the *de re* readings of sentences with existential quantifiers can be accounted for. In von Stechow 1993 the *de re* readings are related to the literature on indefinites, in particular to Fodor and Sag 1982. Stechow speculates that indefinite NPs can be interpreted as names that point to some specific individual. The difference between ordinary names and indefinites, then, is that in the latter case, it is the speaker that is responsible for assigning such an individual to the indefinite term. From such an account, it is predicted that the indefinite term must have a wide scope with respect to other quantifiers in the sentence. Thus, Stechow's account would predict the *de dicto* reading of sentence (39c), in which the indefinite is QR-ed inside the embedded clause, and the $\exists\forall$-*de re* reading, under which the indefinite is interpreted as a name. The approach does not predict the intermediate scope of the existential quantifier as given by the $\forall\exists$-*de re* reading of sentence (39c).

Stechow's considerations correctly point out two things. First, we must distinguish indefinites from elements that are always quantificational. Second, the explanation of the contrast between the possible readings of (39c) and (51) will not be found in a syntactic tree configuration LF, but in the semantics of the determiners.

Finally, let us consider the S-structure Interpretivism of Lappin 1991. We have shown that it is possible to express the *Complex NP Constraint* as a constraint on the semantic type of an NP. The CNPC was formulated so that the denotation of an NP does not contain any stored quantifiers. The number of stored quantifiers is available from the denotation of a node in a tree, as it can be read off from the length of the denoted sequence of meanings. If we want to derive the constraint in (53), it is not enough to consider the length of the sequence in the denotation of an S node. Remember that we want to allow an existential quantifier to take scope outside the clause in which it is introduced. This means that we must allow the translation of some S nodes to contain stored quantifiers. For the derivation of the *de re* readings of sentence (39c), we must allow the denotation of the existential quantifier as the second element in the denotation of the embedded S node. To exclude a *de re* reading of sentence (51) we must, however, disallow the denotation of a universal quantifier in this position.

Under S-structure Interpretivism we cannot access the difference between the two quantifiers directly, but only via the difference in their denotation. In the case of our examples, it is the difference between the denotation of the term $\lambda P.\exists y[\mathsf{woman}'_{@}(y_{@}) \wedge P_{@}(y)]$ and $\lambda P.\forall y[\mathsf{woman}'_{@}(y_{@}) \wedge P_{@}(y)]$. It is unclear what this difference could be.[30] This indicates that expressing the constraint in (53) as a constraint on possible denotations is not without its complications. But a purely denotational formulation is exactly what is needed under S-structure Interpretivism.

On the other hand, it is very likely that the constraint in (53) will disappear in the present form, once we find an adequate analysis of indefinites. In this case, it might be possible to differentiate between the NPs *some woman* and *every woman* purely on the grounds of their denotation, for example by referring to their semantic types.

[30]Note that *some woman* and *every woman* do not differ in monotonicity. As the examples in (i) and (ii) show, both NPs are upward monoton, i.e., if a set A is in the denotation of the NP, then so is every superset of A.

(i) a. Some woman walks and talks $\rightarrow$ Some woman walks.
b. Some woman walks or talks $\not\rightarrow$ Some woman walks.

(ii) a. Every woman walks and talks $\rightarrow$ Every woman walks.
b. Every woman walks or talks $\not\rightarrow$ Every woman walks.

To conclude this discussion, let us summarize that assuming a logical form as a semantic term allows us to differentiate between an existential quantifier and a universal quantifier and, thus, to impose constraints on possible logical forms that are based on this distinction. The distinction is, however, not syntactic in nature, as both kinds of quantifiers appear as part of NPs, i.e., inside elements of the same syntactic category. It is, furthermore, not obvious how a universally quantified NP and an existentially quantified NP can be told apart purely on the basis of their denotation, as they are both interpreted as objects of the same semantic type, at least under the present analysis.

Clearly a refined system of syntactic categories, or, more adequately, a refined semantic analysis, can put the LF proposals or the lf-less proposal in the situation to express the effect of the constraint in (53). Still, it is worth noting that the assumption of an lf allows us to treat the reported differences even within the simple semantic analysis of this section.

1.4. Summary

In this chapter, we have characterized an HPSG grammar as a theory of linguistic objects. The grammar denotes a collection of linguistic objects. Some of these linguistic objects are signs. These signs have components which represent their phonology, their syntactic properties and, most importantly for us, a compontent which represents their semantics. We call this particular component of a sign its *logical form*. In Section 1.2, we summarized various concepts of grammar which differ in their assumptions on whether and how a semantic representation should be part of the structures described by a grammar. We came to the conclusion, that Pollard and Sag 1994 assume the existence of a semantic representation as a component of every sign. Deviating from the further assumptions of Pollard and Sag 1987, 1994, we decided to take as the semantic representation language the language Ty2 (Gallin, 1975), in order to be able to link HPSG semantics to the Montegovian tradition of formal semantics.

Based on the assumption that every linguistic sign has a component which is a term of Ty2, we wanted to avoid the introduction of further components to linguistic signs. This can be done if we adapt a flexible system in the spirit of Hendriks 1993 for the derivation of the semantic representation of signs. Such a system is able to account for scope ambiguities without necessitating the existence of a storage mechanism, as used for example in Pollard and Sag 1994. Therefore, the adaptation of LF-Ty2 will enable us to give a standard semantic representation to linguistic signs, but avoids the introduction of otherwise unmotivated auxiliary compontents such as the QSTORE and RETRIEVED values in the case of the grammar in Pollard and Sag 1994.

In the following chapters of this part of the thesis, we will give a precise formalization of the system of semantic representation sketched above. In Chapter 2, we will define the description language that we use to write HPSG grammars. In Chapter 3 we will present an HPSG grammar, $\mathcal{TY}2$, which will be shown to denote exactly the set of Ty2 terms of Section 1.3.2. As a consequence, terms of Ty2 can be used in HPSG grammars. In Chapter 4 we integrate the grammar $\mathcal{TY}2$ into an HPSG fragment of English such that each sign has a Ty2 term as its logical form. In addition, we provide the formal architecture for the use of type shifting operations needed in LF-Ty2. With the integration of Ty2 into an HPSG grammar, we can also formulate constraints between the syntactic properties of a sign and its logical form as regular principles of grammar.

CHAPTER 2

RSRL

Throughout this thesis, the same underlying formal language will be assumed. The language is called RSRL (*Relational Speciate Re-entrant Language*). RSRL was developed as a formal language that fulfills the requirements of current HPSG theories. As a test case, Richter 2000 shows that RSRL provides exactly the functionality that is needed to formalize the grammar given in the appendix of Pollard and Sag 1994. RSRL has been assumed as the formal foundation in a number of recent HPSG accounts of a variety of linguistic phenomena, including scope (Przepiórkowski, 1997, 1998), case assignment and the complement/adjunct dichotomy (Przepiórkowski, 1999a), linearization (Penn, 1999a,b,c), phonology (Höhle, 1999), morphology (Reinhard, 2001), German clause structure (Richter, 1997; Richter and Sailer, 2001), *tough* movement (Calcagno, 1999), underspecified semantics (Richter and Sailer, 1999d), and negation (Przepiórkowski, 1999b; Richter and Sailer, 1999a,c). In this thesis, we will add to this an integrated treatment of combinatorial semantics, based on the system of Lexicalized Flexible Ty2 (Chapter 4), and an analysis of idiomatic expressions (Chapter 8).

In the present chapter, we will give all the definitions of RSRL and illustrate them first with the small example grammar introduced in Section 1.1. We will, then, provide the syntactic fragment that we can use in Chapter 4 to build our HPSG version of LF-Ty2 on. Even though some understanding of the formal properties of RSRL is necessary to see why and how our encoding of LF-Ty2 behaves in the intended way, we feel this thesis is not the place to discuss the motivation behind RSRL and its place within competing attempts to formalize HPSG. Fortunately, there are many presentations of RSRL which focus on exactly those questions. Richter et al. 1999 explain the main motivation behind those aspects of RSRL that differ from its predecessor *Speciate Re-entrant Logic* (SRL) (King, 1989, 1994, 1999)) . In Richter 1999, the basic definitions are given and an AVM language and some useful notational conventions are defined which provide a reader-friendly description language. Parts of this will be integrated in Section 2.2. Additionally, as SRL is a proper sublanguage of RSRL and as RSRL has adopted SRL's model theory, King 1999 should be consulted for an extensive discussion of the relation between an (R)SRL grammar and natural language. It is in this paper that the importance of *exhaustive models* is emphasized. We have already pointed out in Section 1.1 that exhaustive models are used to determine the denotation of a grammar.[1] Finally, Richter 2000 subsumes all of the above mentioned aspects and includes a comparative survey of the most prominent proposals for formalisms for HPSG from Pollard and Sag 1987 to RSRL.

The present chapter consists of three sections. In Section 2.1, the formal language RSRL is defined. To illustrate the definitions, we will exemplify them with the toy grammar that we have already used in Section 1.1. Section 2.2 gives some simple examples of how the notation used in the definitions of RSRL can be related to the notation which is more

[1]One might also consult Pollard and Sag 1994 and Pollard 1999 for different positions regarding the meaning of HPSG grammars. According to King 1999, the denotation of a grammar is the set of *tokens* of the language. In Pollard and Sag 1994 and Pollard 1999, on the other hand, the notion of *types*, or at least some abstractions over tokens, is central to the interpretation of a grammar. In this thesis, we adopt King's position. As shown in Richter 2000, Pollard's position is equally compatible with RSRL.

commonly used within the HPSG literature: the AVM notation which we have already used in Section 1.1 and in our discussion of the semantic framework used in Pollard and Sag 1994 in Section 1.2. In Section 2.3 we will present the syntactic analysis which we assume throughout the rest of this thesis. This analysis is a slightly modified form of the grammar given in the appendix of Pollard and Sag 1994.

2.1. Definitions

In this section, we repeat the definitions of RSRL, as they are given in the main presentations of RSRL (Richter et al., 1999; Richter, 1999, 2000). See these papers for detailed explanations of the definitions and their underlying motivation.

DEFINITION 2.1 *Σ is a signature iff*

Σ is a septuple $\langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$,
$\langle \mathcal{G}, \sqsubseteq \rangle$ is a partial order,
$$\mathcal{S} = \left\{ \sigma \in \mathcal{G} \;\middle|\; \begin{array}{l} \text{for each } \sigma' \in \mathcal{G}, \\ \text{if } \sigma' \sqsubseteq \sigma \text{ then } \sigma = \sigma' \end{array} \right\},$$
$\mathcal{A}$ is a set,
$\mathcal{F}$ is a partial function from the Cartesian product of $\mathcal{G}$ and $\mathcal{A}$ to $\mathcal{G}$,
for each $\sigma_1 \in \mathcal{G}$, for each $\sigma_2 \in \mathcal{G}$ and for each $\alpha \in \mathcal{A}$,
if $\mathcal{F}\langle \sigma_1, \alpha \rangle$ is defined and $\sigma_2 \sqsubseteq \sigma_1$
then $\mathcal{F}\langle \sigma_2, \alpha \rangle$ is defined and $\mathcal{F}\langle \sigma_2, \alpha \rangle \sqsubseteq \mathcal{F}\langle \sigma_1, \alpha \rangle$,
$\mathcal{R}$ is a finite set, and
$\mathcal{AR}$ is a total function from $\mathcal{R}$ to $\mathbb{N}^+$.

Definition 2.1 defines what an RSRL signature is. The set $\mathcal{G}$ is the set of sorts. $\mathcal{G}$ has a special subset, $\mathcal{S}$ of maximally specific sorts, also called *species*. The pair $\langle \mathcal{G}, \sqsubseteq \rangle$ is the sort hierarchy. In addition to sorts, the signature also provides a set of attributes $\mathcal{A}$, and the appropriateness conditions, which are encoded in the function $\mathcal{F}$. So far, an RSRL signature does not differ significantly from an SRL signature as defined in King 1999 or Richter and Sailer 1995.[2] However, RSRL further includes relational symbols $\mathcal{R}$ in the signature, combined with an explicit statement of the arity of the relations given by the function $\mathcal{AR}$.

In the following, we will illustrate the definitions with a grammar such as the one used in Section 1.1. There, we have assumed a very small signature that differentiated between words and phrases, and uses a list-valued representation of phonology. In (55) we give the signature for this grammar.

(55) The signature of the grammar assumed in Section 1.1:

$$\text{a. } \mathcal{G} = \left\{ \begin{array}{l} \textit{top}, \\ \textit{sign}, \textit{word}, \textit{phrase}, \\ \textit{const-struc}, \\ \textit{synsem}, \\ \textit{list}, \textit{elist}, \textit{nelist}, \\ \textit{phonstring}, \text{Mary}, \text{every}, \text{man}, \text{walks} \end{array} \right\}$$

[2] The definition of SRL given in Richter and Sailer 1995 differs from that in King 1999 in that it explicitly includes the sort hierarchy in the signature. In addition, Richter and Sailer 1995 assume a finite set of sorts.

$$\text{b. } \sqsubseteq = \left\{ \begin{array}{l} \langle \textit{top}, \textit{top} \rangle, \\ \langle \textit{sign}, \textit{sign} \rangle, \langle \textit{word}, \textit{word} \rangle, \langle \textit{phrase}, \textit{phrase} \rangle, \\ \langle \textit{sign}, \textit{top} \rangle, \langle \textit{word}, \textit{top} \rangle, \langle \textit{phrase}, \textit{top} \rangle, \\ \langle \textit{word}, \textit{sign} \rangle, \langle \textit{phrase}, \textit{sign} \rangle, \\ \langle \textit{const-struc}, \textit{const-struc} \rangle, \langle \textit{const-struc}, \textit{top} \rangle, \\ \langle \textit{synsem}, \textit{synsem} \rangle, \langle \textit{synsem}, \textit{top} \rangle, \\ \langle \textit{list}, \textit{list} \rangle, \langle \textit{elist}, \textit{elist} \rangle, \langle \textit{nelist}, \textit{nelist} \rangle, \\ \langle \textit{list}, \textit{top} \rangle, \langle \textit{elist}, \textit{top} \rangle, \langle \textit{nelist}, \textit{top} \rangle, \langle \textit{elist}, \textit{list} \rangle, \langle \textit{nelist}, \textit{list} \rangle, \\ \langle \textit{phonstring}, \textit{phonstring} \rangle, \langle \text{Mary}, \text{Mary} \rangle, \langle \text{every}, \text{every} \rangle, \\ \langle \text{man}, \text{man} \rangle, \langle \text{walks}, \text{walks} \rangle, \\ \langle \textit{phonstring}, \textit{top} \rangle, \langle \text{Mary}, \textit{top} \rangle, \langle \text{every}, \textit{top} \rangle, \\ \langle \text{man}, \textit{top} \rangle, \langle \text{walks}, \textit{top} \rangle, \\ \langle \text{Mary}, \textit{phonstring} \rangle, \langle \text{every}, \textit{phonstring} \rangle, \\ \langle \text{man}, \textit{phonstring} \rangle, \langle \text{walks}, \textit{phonstring} \rangle \end{array} \right\}$$

$$\text{c. } \mathcal{S} = \left\{ \begin{array}{l} \textit{word}, \textit{phrase}, \\ \textit{const-struc}, \\ \textit{synsem}, \\ \textit{elist}, \textit{nelist}, \\ \textit{Mary}, \textit{every}, \textit{man}, \textit{walks} \end{array} \right\}$$

$$\text{d. } \mathcal{A} = \left\{ \begin{array}{l} \text{PHON}, \text{SYNSEM}, \text{DTRS}, \\ \text{H-DTR}, \text{N-DTR}, \\ \text{FIRST}, \text{REST} \end{array} \right\}$$

$$\text{e. } \mathcal{F} = \left\{ \begin{array}{l} \langle \langle \textit{sign}, \text{PHON} \rangle, \textit{list} \rangle, \langle \langle \textit{sign}, \text{SYNSEM} \rangle, \textit{synsem} \rangle, \\ \langle \langle \textit{word}, \text{PHON} \rangle, \textit{list} \rangle, \langle \langle \textit{word}, \text{SYNSEM} \rangle, \textit{synsem} \rangle, \\ \langle \langle \textit{phrase}, \text{PHON} \rangle, \textit{list} \rangle, \langle \langle \textit{phrase}, \text{SYNSEM} \rangle, \textit{synsem} \rangle, \\ \langle \langle \textit{phrase}, \text{DTRS} \rangle, \textit{const-struc} \rangle, \\ \langle \langle \textit{const-struc}, \text{H-DTR} \rangle, \textit{sign} \rangle, \langle \langle \textit{const-struc}, \text{N-DTR} \rangle, \textit{sign} \rangle, \\ \langle \langle \textit{nelist}, \text{FIRST} \rangle, \textit{top} \rangle, \langle \langle \textit{nelist}, \text{REST} \rangle, \textit{list} \rangle \end{array} \right\}$$

f. $\mathcal{R} = \{\texttt{append}\}$

g. $\mathcal{AR} = \{\langle \texttt{append}, 3 \rangle\}$

The signature as given in (55) includes the sort hierarchy and the appropriateness conditions as we have expressed them in the graph in (11) (page 13). In addition, we have now introduced explicitly those sorts that were left implicit in Section 1.1, such as the top sort *top* and the encoding of lists via the two sorts *e(mpty-)list* and *n(on-)e(mpty-)list*. It can be seen in (55b) that the sort *word* is a subsort of the sort *sign*. As the sort *word* is not a supersort of any other sort, it appear in the set of species, $\mathcal{S}$. In (55e) we see that the attributes defined on the sort *word* are exactly those defined on the sort *sign*, whereas the sort *phrase* has a further attribute, DTRS (DAUGHTERS), defined on it. Finally, we use exactly one relation, `append`, which takes three arguments, as stated in (55g).

To keep the following definitions readable, we introduce some notational conventions in Convention 2.2.

CONVENTION 2.2 *For each set* S,

(1) *for each natural number* n,
  S^n *is the set of* n*-tuples over elements of a set* S,
(2) S^* *is the set of finite sequences or strings of over elements of a set* S,
(3) S^+ *is the set of nonempty finite sequences or strings of elements of a set* S,
(4) $\overline{S}$ *is an abbreviation for* $S \cup S^*$.

A signature structures the empirical domain. This is expressed in the following definition of a signature interpretation.

DEFINITION 2.3 *For each signature* Σ, I *is a* Σ *interpretation iff*

> I *is a quadruple* $\langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$,
> U *is a set,*
> S *is a total function from* U *to* $\mathcal{S}$,
>
> A *is a total function from* $\mathcal{A}$ *to the set of partial functions from* U *to* U,
> *for each* $\alpha \in \mathcal{A}$, *for each* $u \in \mathsf{U}$,
> *if* $\mathsf{A}(\alpha)(u)$ *is defined*
> *then* $\mathcal{F}\langle \mathsf{S}(u), \alpha \rangle$ *is defined, and* $\mathsf{S}(\mathsf{A}(\alpha)(u)) \sqsubseteq \mathcal{F}\langle \mathsf{S}(u), \alpha \rangle$,
> *for each* $\alpha \in \mathcal{A}$, *for each* $u \in \mathsf{U}$,
> *if* $\mathcal{F}\langle \mathsf{S}(u), \alpha \rangle$ *is defined then* $\mathsf{A}(\alpha)(u)$ *is defined,*
> R *is a total function from* $\mathcal{R}$ *to the power set of* $\bigcup_{n \in \mathbb{N}} (\overline{\mathsf{U}})^n$,
> *and for each* $\rho \in \mathcal{R}$, $\mathsf{R}(\rho) \subseteq (\overline{\mathsf{U}})^{\mathcal{AR}(\rho)}$.

A signature interpretation has a number of properties. There is a universe U, which contains all the objects of the domain. Each object in the universe is assigned exactly one species by the function S. The attribute interpretation function A respects the appropriateness conditions as specified in $\mathcal{F}$.

The way we have defined the signature interpretation, the objects in the interpretation of a signature are the way we have characterized them in Section 1.1, i.e., all objects have some label, which is taken from the maximally specific sorts. Furthermore, all attributes declared appropriate for a sort are defined on objects of that sort and take values which are of an appropriate sort.

Finally, the relation symbols are interpreted by the function R. A relation symbol `relation` with arity n denotes a set of n-tuples. In contrast to what one might expect, these n-tuples are not necessarily n-tuples of objects, but n-tuples of elements of $\overline{\mathsf{U}}$, i.e., by convention 2.2, n-tuples of elements of $\mathsf{U} \cup \mathsf{U}^*$.

With these formal definitions, we can see that the configurations of objects given in Section 1.1 (Figures 1.1–1.5) all obey the signature in (55). If we take the collection of objects given in these configurations as our universe U, then the function S is the function that assigns each of the objects its label. The function A is given by the arrows used in the configurations. What is missing in the configurations is the relation interpretation. In (17) on page 19 we have given an informal characterization of what should be in the interpretation of the relation `append`. A set $\mathsf{R}(\texttt{append})$ that meets this characterization is compatiple with the requirements expressed in Definition 2.3, as such a set is a subset of U^3, which is a subset of $\overline{\mathsf{U}}^3$, and $\mathcal{AR}(\texttt{append}) = 3$.

In RSRL, elements of the set U^* are of a particular importance. We refer to them as *chains*. As chains are elements of U^*, they are finite sequences of objects in an interpretation. While each element in such a sequence appears in the universe U of an interpretation, the chain itself does not. In this sense chains are "virtual" objects. Chains are needed in an RSRL grammar, whenever complex dependencies between components of an object must be expressed. In Section 4.2 we will show that λ-conversion in Ty2 is such a relation: to express λ-conversion in RSRL, we must either extend the ontology of our grammar, or use these virtual entities instead. In the simple grammar that we will present in this section, we will state a constraint in the theory that enforces that for each object in the denotation of the grammar, the configuration under the object consists only of a finite number of objects. There is no known method to impose such a constraint using the signature in (55) and not allowing for chains. If we use chains, the formulation of such a constraint is strainghtforward. The reason for this is that chains are always finite and acyclic, by virtue of being elements of U^*. In contrast to this, the configuration under a linguistic object can be infinite or cyclic in principle.

Coming back to the interpretation of relations in RSRL, we, thus, can say that a relation denotes an n-tuple of objects or chains.

The following definition gives us some special syntax to work with chains.

DEFINITION 2.4 *For each signature* $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, *we define*

$$\begin{aligned}
\widehat{\mathcal{G}} &= \mathcal{G} \cup \{\textit{chain}, \textit{echain}, \textit{nechain}, \textit{metatop}\}, \\
\widehat{\sqsubseteq} &= \sqsubseteq \\
&\cup \left\{ \begin{array}{l} \langle \textit{echain}, \textit{echain} \rangle, \langle \textit{nechain}, \textit{nechain} \rangle, \\ \langle \textit{chain}, \textit{chain} \rangle, \langle \textit{echain}, \textit{chain} \rangle, \langle \textit{nechain}, \textit{chain} \rangle \end{array} \right\} \\
&\cup \left\{ \langle \sigma, \textit{metatop} \rangle \,\middle|\, \sigma \in \widehat{\mathcal{G}} \right\}, \\
\widehat{\mathcal{S}} &= \mathcal{S} \cup \{\textit{echain}, \textit{nechain}\}, \text{ and} \\
\widehat{\mathcal{A}} &= \mathcal{A} \cup \{\dagger, \triangleright\}.
\end{aligned}$$

Independent of the signature, RSRL provides the special symbols introduced in Definition 2.4. These symbols augment the signature specified by a linguist. The augmentation introduces an extension of the sorts ($\widehat{\mathcal{G}}$), including the species ($\widehat{\mathcal{S}}$) and the sort hierarchy ($\left\langle \widehat{\mathcal{G}}, \widehat{\sqsubseteq} \right\rangle$). Note that $\left\langle \widehat{\mathcal{G}}, \widehat{\sqsubseteq} \right\rangle$ is a partial order. We call the new symbols *chain*, *echain* and *nechain* quasi-sorts.

In addition, there is also an augmented set of attributes $\widehat{\mathcal{A}}$. The symbols that are contained in $\widehat{\mathcal{A}}$ but not in $\mathcal{A}$, i.e., $\dagger$ and $\triangleright$, are called quasi-attributes.

Definition 2.4 gives us an augmented signature. In order to be able to interpret the newly introduced symbols, we must, next, define an interpretation for them. This is done in the following definition.

DEFINITION 2.5 *For each signature* $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, *for each* Σ *interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$,

$\widehat{\mathsf{S}}$ *is the total function from* $\overline{\mathsf{U}}$ *to* $\widehat{\mathcal{S}}$ *such that*
for each $u \in \mathsf{U}$, $\widehat{\mathsf{S}}(u) = \mathsf{S}(u)$, *for each* $u_1 \in \mathsf{U}$, ..., *for each* $u_n \in \mathsf{U}$,
$\widehat{\mathsf{S}}(\langle u_1, \ldots, u_n \rangle) = \begin{cases} \textit{echain} & \textit{if } n = 0, \\ \textit{nechain} & \textit{if } n > 0 \end{cases}$, *and*

$\widehat{\mathsf{A}}$ *is the total function from* $\widehat{\mathcal{A}}$ *to the set of partial functions from* $\overline{\mathsf{U}}$ *to* $\overline{\mathsf{U}}$ *such that*
for each $\alpha \in \mathcal{A}$, $\widehat{\mathsf{A}}(\alpha) = \mathsf{A}(\alpha)$, *and*
$\widehat{\mathsf{A}}(\dagger)$ *is the total function from* U^+ *to* U *such that*
for each $\langle u_0, \ldots, u_n \rangle \in \mathsf{U}^+$, $\widehat{\mathsf{A}}(\dagger)(\langle u_0, \ldots, u_n \rangle) = u_0$, *and*
$\widehat{\mathsf{A}}(\triangleright)$ *is the total function from* U^+ *to* U^* *such that*
for each $\langle u_0, \ldots, u_n \rangle \in \mathsf{U}^+$, $\widehat{\mathsf{A}}(\triangleright)(\langle u_0, \ldots, u_n \rangle) = \langle u_1, \ldots, u_n \rangle$.

Definition 2.5 introduces an augmented species interpretation function $\widehat{\mathsf{S}}$. Regular objects, i.e., elements of the universe U are interpreted just as under S. For chains, i.e., elements of U^*, a special treatment is introduced. An empty chain is assigned the symbol *echain* by $\widehat{\mathsf{S}}$, a non-empty chain is assigned the symbol *nechain*.

Similarly, the attribute interpretation is augmented. For ordinary attributes, the augmented attribute interpretation function $\widehat{\mathsf{A}}$ behaves just like A. For (non-empty) chains, the symbol † is used to relate a non-empty chain to its first element via $\widehat{\mathsf{A}}$. The symbol ▷ relates a non-empty chain c to a chain c', which is just like c with the first element removed.

From these definitions, we can see that chains have a list-like structure. If we have an RSRL grammar such as that of Pollard and Sag 1994, or the toy grammar of Section 1.1, both of which assume a sort *list*, then there is an intuitive correspondence between the sort *elist* and the symbol *echain*. Similarly for the sort *nelist* and the symbol *nechain*, the attribute FIRST and the symbol †, and, finally, the attribute REST and the symbol ▷.

As shown in Richter et al. 1999 and in more detail in Richter 2000, chains have been used implicitly throughout HPSG grammars. They are needed to encode principles of Pollard and Sag 1994, such as the (a) clause of the SEMANTIC PRINCIPLE (see (29)), and many other works in HPSG, such as linearization grammars which use `shuffle` (Kathol and Pollard, 1995). In (63b) we will augment our toy grammar by a principle that makes essential use of chains to impose a finiteness restriction on linguistic objects.

Before we can turn to complex syntactic entities in RSRL, we must introduce yet another set of basic symbols:

DEFINITION 2.6 *$\mathcal{VAR}$ is a countably infinite set of symbols.*

We call each element of $\mathcal{VAR}$ a *variable*. As we have seen in Section 1.1, we will use either lower case letters (a, b, c, ...) or boxed integers ($\boxed{1}$, $\boxed{2}$, ...) for variables. Furthermore, in order to capture common HPSG terminology, we will often refer to variables as *tags*.

Now that we have defined the basic symbols that we want to use and their interpretation, we can define larger syntactic units. We will start with *terms*.

DEFINITION 2.7 *For each signature Σ, $\mathcal{T}^{\Sigma}$ is the smallest set such that*

$: \in \mathcal{T}^{\Sigma}$,
for each $v \in \mathcal{VAR}$, $v \in \mathcal{T}^{\Sigma}$,
for each $\alpha \in \widehat{\mathcal{A}}$ *and each* $\tau \in \mathcal{T}^{\Sigma}$, $\tau\alpha \in \mathcal{T}^{\Sigma}$.

The set $\mathcal{T}^{\Sigma}$ is called the set of Σ terms. A term starts with the symbol ": " or a variable, followed by a finite sequence of elements for $\widehat{\mathcal{A}}$, i.e., attributes or quasi-attributes. Using our toy signature in (55), we can give some terms.

(56) a. : PHON
b. : PHON SYNSEM
c. $\boxed{2}$DTRS H-DTR SYNSEM
d. a†SYNSEM

In (a) and (b), we have terms that start with the special symbol ": ", followed by some attributes. In (c) and (d), the terms start with variables, a tag ($\boxed{2}$) in (56c), and a lower case letter (a) in (56d). In (a)–(c), we use attributes from the signature, in (d), the quasi-attribute "†" from the augmented set of attributes occurs.

As terms contain variables, we will introduce a *variable assignment function*, before interpreting terms.

DEFINITION 2.8 *For each signature Σ, for each Σ interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$,

$\mathsf{Ass_I} = \overline{\mathsf{U}}^{\mathcal{VAR}}$ *is the set of variable assignments in* I.

Each element of $\mathsf{Ass_I}$ is called a *variable assignment.* A variable assignment maps each variable either into an element of the universe U or into a chain, i.e., an element of U^*. We can now state how terms are interpreted.

DEFINITION 2.9 *For each signature* $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, *for each* Σ *interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$, *for each* $ass \in \mathsf{Ass_I}$,

> T_{I}^{ass} *is the total function from* $\mathcal{T}^\Sigma$ *to the set of partial functions from* U *to* $\overline{\mathsf{U}}$, *to the power set of* U *such that, for each* $u \in \mathsf{U}$,
> $T_{\mathsf{I}}^{ass}(:)(u)$ *is defined and* $T_{\mathsf{I}}^{ass}(:)(u) = u$,
> *for each* $v \in \mathcal{VAR}$,
> $T_{\mathsf{I}}^{ass}(v)(u)$ *is defined and* $T_{\mathsf{I}}^{ass}(v)(u) = ass(v)$.
> *for each* $\tau \in \mathcal{T}^\Sigma$, *for each* $\alpha \in \widehat{\mathcal{A}}$,
> $T_{\mathsf{I}}^{ass}(\tau\alpha)(u)$ *is defined iff*
> $T_{\mathsf{I}}^{ass}(\tau)(u)$ *is defined and* $\widehat{\mathsf{A}}(\alpha)(T_{\mathsf{I}}^{ass}(\tau)(u))$ *is defined, and*
> *if* $T_{\mathsf{I}}^{ass}(\tau\alpha)(u)$ *is defined then* $T_{\mathsf{I}}^{ass}(\tau\alpha)(u) = \widehat{\mathsf{A}}(\alpha)(T_{\mathsf{I}}^{ass}(\tau)(u))$.

T_{I}^{ass} is called a *term interpretation function.* It maps every Σ term into a partial function from U to $\overline{\mathsf{U}}$. The symbol ": " is mapped into the identity function. Variables are mapped into constant functions. The variable assignment *ass* determines the particular choice of the constant function. The interpretation of complex terms of the form $\tau\alpha$ is a partial function, because it is only defined if the attribute α is defined on the object $T_{\mathsf{I}}^{ass}(\tau)(u)$.

Now that we know how terms are interpreted, we can re-consider the terms in (56). The term given in (56a) is defined on every object in the universe U on which the attribute PHON is defined. In our toy signature, the only species for which this attribute is appropriate are the sorts *word* and *phrase*. Thus, if we take an object o with label *word*, i.e., $\widehat{\mathsf{S}}(o) = word$, then the term interpretation function is defined on the term : PHON and on o and has as value the phonology of the word o, i.e., the list that occurs as the the value of $\widehat{\mathsf{A}}(o)$(PHON). In the configuration under the word *walks* in Figure 1.1 (page 14) the term interpretation of the term : PHON is only defined on the matrix object, and yields the *nelist* object that occurs in the configuration.

If we consider the term given in (56b), we can see that the term interpretation is not defined for this term: Let us take again some object o which is of sort *word.* Then, as we have seen above, the term interpretation is defined on that object and the term : PHON, and it yields an object of sort *elist* or *nelist*, as the appropriateness function $\mathcal{F}$ in (55e) contains $\langle\langle$*word*, PHON$\rangle$, *list*$\rangle$. The attribute SYNSEM, however, is not appropriate to either of these sorts. This is indicated in the matrix object of the configuration in Figure 1.1: there is no path that starts from the matrix object of this configuration and goes first via an arrow with the name PHON and then via an arrow with name SYNSEM.

With the same reasoning, we can show that the term interpretation function on the term :PHON SYNSEM is not defined on an object with label *phrase.* For objects with labels other than *word* or *phrase*, the term interpretation function on the term :PHON SYNSEM is not defined either, because the attribute PHON is only appropriate to these two species.

For the term interpretation of the terms in (56c) and (56d), we must know the values of the variable assignment function on the variables [2] and a respectively. Let us assume that the term interpretation is defined on these terms. Note, however, that the first attribut in the term in (56d) is taken from the augmented attribute set, i.e., it is a quasi-attribute. According to the way the augmented attribute interpretation function is defined, this means that the variable assigment must be chosen in such a way that the variable a is interpreted as a chain, i.e., $ass(a) \in \mathsf{U}^*$.

We use terms to build larger syntactic units, *descriptions*:[3]

DEFINITION 2.10 *For each signature* Σ, $\mathcal{D}^\Sigma$ *is the smallest set such that*

$$\begin{aligned}
&\textit{for each } \sigma \in \widehat{\mathcal{G}}, \textit{ for each } \tau \in \mathcal{T}^\Sigma, \tau{\sim}\sigma \in \mathcal{D}^\Sigma,\\
&\quad \textit{for each } \tau_1 \in \mathcal{T}^\Sigma, \textit{ for each } \tau_2 \in \mathcal{T}^\Sigma, \tau_1{\approx}\tau_2 \in \mathcal{D}^\Sigma,\\
&\quad \textit{for each } \rho \in \mathcal{R}, \textit{ for each } x_1 \in \mathcal{VAR}, \ldots, \textit{ for each } x_{\mathcal{AR}(\rho)} \in \mathcal{VAR},\\
&\qquad \rho(x_1, \ldots, x_{\mathcal{AR}(\rho)}) \in \mathcal{D}^\Sigma,\\
&\quad \textit{for each } x \in \mathcal{VAR}, \textit{ for each } \delta \in \mathcal{D}^\Sigma, \mathbf{E}x\ \delta \in \mathcal{D}^\Sigma,\\
&\quad \textit{for each } x \in \mathcal{VAR}, \textit{ for each } \delta \in \mathcal{D}^\Sigma, \mathbf{A}x\ \delta \in \mathcal{D}^\Sigma,\\
&\quad \textit{for each } \delta \in \mathcal{D}^\Sigma, \mathbf{not}\ \ \delta \in \mathcal{D}^\Sigma,\\
&\quad \textit{for each } \delta_1 \in \mathcal{D}^\Sigma, \textit{ for each } \delta_2 \in \mathcal{D}^\Sigma, [\delta_1 \mathbf{and}\ \delta_2] \in \mathcal{D}^\Sigma,\\
&\quad \textit{for each } \delta_1 \in \mathcal{D}^\Sigma, \textit{ for each } \delta_2 \in \mathcal{D}^\Sigma, [\delta_1 \mathbf{or}\ \delta_2] \in \mathcal{D}^\Sigma,\\
&\quad \textit{for each } \delta_1 \in \mathcal{D}^\Sigma, \textit{ for each } \delta_2 \in \mathcal{D}^\Sigma, [\delta_1 {\Rightarrow} \delta_2] \in \mathcal{D}^\Sigma,\\
&\quad \textit{for each } \delta_1 \in \mathcal{D}^\Sigma, \textit{ for each } \delta_2 \in \mathcal{D}^\Sigma, [\delta_1 {\Leftrightarrow} \delta_2] \in \mathcal{D}^\Sigma.
\end{aligned}$$

Σ descriptions come in several forms. The following names will be helpful for further reference. They are given in the order of the lines in Definition 2.10: sort assignments, i.e., descriptions of the form $\tau{\sim}\sigma$, identities, relation calls, existentially quantified descriptions, universally quantified descriptions, negated descriptions, conjunctions, disjunctions, implications and equivalences.

One remark should be made about the choice of symbols in Definition 2.10. In other presentations of RSRL such as Richter 1999, 2000 or Richter et al. 1999, the more standard symbols $\exists$, $\forall$, $\neg$, $\wedge$, $\vee$, $\rightarrow$ and $\leftrightarrow$ are used for logical constants instead of **E**, **A**, **not**, **and**, **or**, $\Rightarrow$ and $\Leftrightarrow$. As the former symbols are part of our semantic representation language Ty2 as introduced in Section 1.3.2, we use different symbols for RSRL to avoid confusion.

In (57) we find the formal equivalent of the AVM notations used in Section 1.1 and indicate which AVM corresponds to which description. In Section 2.2 we will show how AVMs as those used in Section 1.1 can be transformed into the format of descriptions defined in Definition 2.10.

(57) a. = (12) on page 15:
: ~*word*
and : PHON FIRST~*walks*
and : PHON REST~*elist*

b. = (14) on page 16:
: ~*phrase* **and** : PHON FIRST~*every*
and : PHON REST FIRST~*man*
and : PHON REST REST FIRST~*walks*
and : PHON REST REST REST~*elist*

and : DTRS H-DTR~*word*
and : DTRS H-DTR PHON FIRST ≈ : PHON REST REST FIRST
and : DTRS H-DTR PHON REST~*elist*

and : DTRS N-DTR~*phrase*
and : DTRS N-DTR PHON FIRST ≈ : PHON FIRST
and : DTRS N-DTR PHON REST FIRST ≈ : PHON REST FIRST
and : DTRS N-DTR PHON REST REST~*elist*

and : DTRS N-DTR DTRS H-DTR~*word*
and : DTRS N-DTR DTRS H-DTR PHON FIRST ≈ : PHON REST FIRST

[3]In Richter 2000 (pp. 165f.), the term *descriptions* is confined to elements of $\mathcal{D}^\Sigma$ which do not contain free variables. The set $\mathcal{D}^\Sigma$ is called the set of *formulae*. In the present work, we do not follow this terminology.

and : DTRS N-DTR DTRS H-DTR PHON REST∼*elist*

and : DTRS N-DTR DTRS N-DTR∼*word*
and : DTRS N-DTR DTRS N-DTR PHON FIRST ≈ : PHON FIRST
and : DTRS N-DTR DTRS N-DTR PHON REST∼*elist*

c. = (15) on page 18:

$\mathbf{E}\boxed{1}\mathbf{E}\boxed{2}\mathbf{E}\boxed{3}$

$\boxed{1}$∼*every*
and $\boxed{2}$∼*man*
and $\boxed{3}$∼*walks*

and : ∼*phrase*
and : PHON FIRST ≈ $\boxed{1}$
and : PHON REST FIRST ≈ $\boxed{2}$
and : PHON REST REST FIRST ≈ $\boxed{3}$
and : PHON REST REST REST∼*elist*

and : DTRS H-DTR∼*word*
and : DTRS H-DTR PHON FIRST ≈ $\boxed{3}$
and : DTRS H-DTR PHON REST∼*elist*

and : DTRS N-DTR∼*phrase*
and : DTRS N-DTR PHON FIRST ≈ $\boxed{1}$
and : DTRS N-DTR PHON REST FIRST ≈ $\boxed{2}$
and : DTRS N-DTR PHON REST REST∼*elist*

and : DTRS N-DTR DTRS H-DTR∼*word*
and : DTRS N-DTR DTRS H-DTR PHON FIRST ≈ $\boxed{2}$
and : DTRS N-DTR DTRS H-DTR PHON REST∼*elist*

and : DTRS N-DTR DTRS N-DTR∼*word*
and : DTRS N-DTR DTRS N-DTR PHON FIRST ≈ $\boxed{1}$
and : DTRS N-DTR DTRS N-DTR PHON REST∼*elist*

d. = (16) on page 18:

$\mathbf{E}\boxed{1}\mathbf{E}\boxed{2}\mathbf{E}\boxed{3}\mathbf{E}\boxed{4}\mathbf{E}\boxed{5}$

$\boxed{1}$FIRST∼*every*
and $\boxed{1}$REST∼*elist*

and $\boxed{2}$FIRST∼*man*
and $\boxed{2}$REST∼*elist*

and $\boxed{3}$FIRST∼*walks*
and $\boxed{3}$REST∼*elist*

and : ∼*phrase*
and : PHON ≈ $\boxed{5}$

and : DTRS H-DTR∼*word*
and : DTRS H-DTR PHON ≈ $\boxed{3}$

and : DTRS N-DTR∼*phrase*
and : DTRS N-DTR PHON ≈ $\boxed{4}$

and : DTRS N-DTR DTRS H-DTR∼*word*
and : DTRS N-DTR DTRS H-DTR PHON ≈ $\boxed{2}$

and : DTRS N-DTR DTRS N-DTR∼*word*
and : DTRS N-DTR DTRS N-DTR PHON ≈ $\boxed{1}$

and append($\boxed{1}$,$\boxed{2}$,$\boxed{4}$)
and append($\boxed{4}$,$\boxed{3}$,$\boxed{5}$)

The descriptions in (a) only contains conjunctions and sort assignments. The description in (b) furthermore also contains identities. Note that in the corresponding AVM in (14) we have used tags to express these identities. This is due to the fact that the AVM language does not provide means to express identities, except for variable identities. In (c) we have added variables and quantifiers, and, finally, in (d) we also added relation calls.

We need two auxiliary definitions before we can interpret descriptions. These additional definitions are, actually, only necessary for the interpretation of quantified descriptions. Quantification and relation calls are, however, not independent of each other. As we will see later, descriptions in a theory will never contain free variables. By definition 2.10, relation calls must contain variables. Therefore, in a linguistic theory, all calls must occur inside quantified descriptions.

DEFINITION 2.11 *For each signature* Σ*, for each* Σ *interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$*, for each* $ass \in \mathsf{Ass}_\mathsf{I}$*, for each* $v \in \mathcal{VAR}$*, for each* $w \in \mathcal{VAR}$*, for each* $u \in \overline{\mathsf{U}}$*,*

$$ass\tfrac{u}{v}(w) = \begin{cases} u & \text{if } v = w \\ ass(w) & \text{otherwise.} \end{cases}$$

Definition 2.11 defines what it means to change a variable assignment at one place. The following definition introduces the notion of a *component*.

DEFINITION 2.12 *For each signature* $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$*, for each* Σ *interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$*, for each* $u \in \mathsf{U}$*,*

$$\mathsf{Co}_\mathsf{I}^u = \left\{ u' \in \mathsf{U} \;\middle|\; \begin{array}{l} \text{for some } ass \in \mathsf{Ass}_\mathsf{I}, \\ \text{for some } \pi \in \mathcal{A}^*, \\ \quad T_\mathsf{I}^{ass}(:\pi)(u) \text{ is defined, and} \\ \quad u' = T_\mathsf{I}^{ass}(:\pi)(u) \end{array} \right\}.$$

In Definition 2.12 the set of components is defined for objects, i.e., elements of U. A component of an object u is an object u' that is the value of the term interpretation of some path $:\pi$, applied to u. Intuitively speaking, the set of components of an object contains all objects that can be reached from this object by some sequence of attributes.

In the case of the configurations of objects given in Section 1.1, all (and only those) objects that occurred in a configuration under a given matrix object are the components of the matrix object. Thus, in the case of Figure 1.1, all objects in this figure are components of the *word* object. It can be seen that each of these objects can be reached via a finite series of arrows, starting from the matrix object of the configuration.

The next definition provides us with the interpretation of Σ descriptions as they were given in Definition 2.10.

DEFINITION 2.13 *For each signature* $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$*, for each* Σ *interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$*, for each* $ass \in \mathsf{Ass}_\mathsf{I}$*,*

D_I^{ass} *is the total function from* $\mathcal{D}^\Sigma$ *to the power set of* U *such that, for each* $u \in \mathsf{U}$*, for each* $\tau \in \mathcal{T}^\Sigma$*, for each* $\sigma \in \widehat{\mathcal{G}}$*,*

$$D_\mathsf{I}^{ass}(\tau \sim \sigma) = \left\{ u \in \mathsf{U} \;\middle|\; \begin{array}{l} T_\mathsf{I}^{ass}(\tau)(u) \text{ is defined, and} \\ \widehat{\mathsf{S}}(T_\mathsf{I}^{ass}(\tau)(u)) \mathrel{\widehat{\sqsubseteq}} \sigma \end{array} \right\},$$

for each $\tau_1 \in \mathcal{T}^\Sigma$, *for each* $\tau_2 \in \mathcal{T}^\Sigma$,

$$D_\mathsf{I}^{ass}(\tau_1 \approx \tau_2) = \left\{ u \in \mathsf{U} \left| \begin{array}{l} T_\mathsf{I}^{ass}(\tau_1)(u) \textit{ is defined,} \\ T_\mathsf{I}^{ass}(\tau_2)(u) \textit{ is defined, and} \\ T_\mathsf{I}^{ass}(\tau_1)(u) = T_\mathsf{I}^{ass}(\tau_2)(u) \end{array} \right. \right\},$$

for each $\rho \in \mathcal{R}$, *for each* $x_1 \in \mathcal{VAR}$, ..., *for each* $x_{\mathcal{AR}(\rho)} \in \mathcal{VAR}$,

$$D_\mathsf{I}^{ass}(\rho(x_1, \ldots, x_{\mathcal{AR}(\rho)})) =$$
$$\left\{ u \in \mathsf{U} \;\middle|\; \left\langle ass(x_1), \ldots, ass(x_{\mathcal{AR}(\rho)}) \right\rangle \in \mathsf{R}(\rho) \right\},$$

for each $v \in \mathcal{VAR}$, *for each* $\delta \in \mathcal{D}^\Sigma$,

$$D_\mathsf{I}^{ass}(\mathbf{E}v\,\delta) = \left\{ u \in \mathsf{U} \left| \begin{array}{l} \textit{for some } u' \in \overline{\mathsf{Co}_\mathsf{I}^u}, \\ u \in D_\mathsf{I}^{ass\frac{u'}{v}}(\delta) \end{array} \right. \right\},$$

for each $v \in \mathcal{VAR}$, *for each* $\delta \in \mathcal{D}^\Sigma$,

$$D_\mathsf{I}^{ass}(\mathbf{A}v\,\delta) = \left\{ u \in \mathsf{U} \left| \begin{array}{l} \textit{for each } u' \in \overline{\mathsf{Co}_\mathsf{I}^u}, \\ u \in D_\mathsf{I}^{ass\frac{u'}{v}}(\delta) \end{array} \right. \right\},$$

for each $\delta \in \mathcal{D}^\Sigma$,

$$D_\mathsf{I}^{ass}(\mathbf{not}\ \delta) = \mathsf{U} \backslash D_\mathsf{I}^{ass}(\delta),$$

for each $\delta_1 \in \mathcal{D}^\Sigma$, *for each* $\delta_2 \in \mathcal{D}^\Sigma$,

$$D_\mathsf{I}^{ass}([\delta_1 \mathbf{and}\ \delta_2]) = D_\mathsf{I}^{ass}(\delta_1) \cap D_\mathsf{I}^{ass}(\delta_2),$$

for each $\delta_1 \in \mathcal{D}^\Sigma$, *for each* $\delta_2 \in \mathcal{D}^\Sigma$,

$$D_\mathsf{I}^{ass}([\delta_1 \mathbf{or}\ \delta_2]) = D_\mathsf{I}^{ass}(\delta_1) \cup D_\mathsf{I}^{ass}(\delta_2).$$

for each $\delta_1 \in \mathcal{D}^\Sigma$, *for each* $\delta_2 \in \mathcal{D}^\Sigma$,

$$D_\mathsf{I}^{ass}([\delta_1 \Rightarrow \delta_2]) = (\mathsf{U} \setminus D_\mathsf{I}^{ass}(\delta_1)) \cup D_\mathsf{I}^{ass}(\delta_2).$$

for each $\delta_1 \in \mathcal{D}^\Sigma$, *for each* $\delta_2 \in \mathcal{D}^\Sigma$,

$$D_\mathsf{I}^{ass}([\delta_1 \Leftrightarrow \delta_2]) =$$
$$((\mathsf{U} \setminus D_\mathsf{I}^{ass}(\delta_1)) \cap (\mathsf{U} \setminus D_\mathsf{I}^{ass}(\delta_2))) \cup (D_\mathsf{I}^{ass}(\delta_1) \cap D_\mathsf{I}^{ass}(\delta_2)).$$

The interpretation function for Σ descriptions (D_I^{ass}) is defined according to the recursive structure of Σ descriptions. A description denotes a set of objects, or, put differently, it describes a set of objects. For example, a sort assignment $\tau \sim \sigma$ describes an object, if the term τ is defined on that object and the interpretation of τ on that object is an object that is of a subsort of σ. Similarly for identities, the two terms must be defined on the object and their interpretation on that object must be the same.

Negation, conjunction, disjunction, implication and equivalence are interpreted in the classical way as set complement, set union, intersection, etc. Also the interpretation of relation calls appears innocuous: a relation call describes an object iff the interpretation of the variables yield a tuple which is in the interpretation of the relation symbol. There is, however, an important caveat to this. As the term interpretation of a variable is a constant function, it is independent of the particular object at which the variable is interpreted. Therefore, for a given variable assignment, a call will always describe either all objects, i.e., the entire universe, or no object at all.

The interpretation of quantified expressions is formulated in term of the Definitions 2.11 and 2.12. In RSRL, a quantifier may only range over components or chains of components of an object. This means, an existentially quantified description $\exists v \delta$ describes an object u iff, there is some u' which is either a component of u or a chain which consists of components of u, such that δ describes u if the variable assignment is modified in such a way that the variable v is mapped to u'. For a universally quantified description, all components and all chains of components of an object must fulfill the corresponding requirement. It can be seen that the interpretation of quantified descriptions is standard with the exception that (i) it is not quantification over the entire universe U, and (ii) the range of the quantifier contains components and chains of components.

Richter et al. 1999 and Richter 2000 argue that this special kind of restricted quantification is exactly what is needed to formalize Pollard and Sag 1994 and other work in HPSG. To prove this strong claim, Richter 2000 gives an RSRL formalization of every principle contained in the appendix of Pollard and Sag 1994 and a detailed explanation about why a particular formalization is chosen. On a more general level, Richter et al. 1999 claims that quantification over the entire universe contradicts the expectation that it should be possible to determine the grammaticality of an object simply by considering the object itself and its parts (i.e., the object and the configuration under it). If we restricted the range of the RSRL quantifiers to components only (instead of allowing chains of components as well), we would not be able to formalize linguistic principles in the way they were stated originally. In particular, the possibility of using relation calls would be too limited. Below, we will use chains to express a constraint that all configurations under a linguistic object contain only a finite number of objects. Furthermore, in Section 4.2.2 we will use chains to encode λ-conversion on terms of Ty2.

After these general remarks, we can turn again to our examples from Section 1.1. In that section, we have already shown informally how the matrix objects of the given configurations either match or don't match (parts of) the descriptions given there in AVM form. For the purpose of illustration, let us re-consider the description in (57a) and the matrix object of the configuration in Figure 1.1. The object is described by the first line of the description, because the term ": " is defined on it (just as on every object) and maps the object to itself. Furthermore, the matrix object is labelled *word*. Similarly, the matrix object also meets the second line of the description: the path : PHON FIRST is defined on it and the object arrived at via this path has the label *walks*. Finally, the third line is met, because the path : PHON REST is also defined on the object, and leads to an object with label *elist*.

Let us also consider a simple example of a quantified expression, such as the description in (57c). Let us assume that we consider some object o whose : DTRS N-DTR value is the object which serves as the matrix object in Figure 1.4 (page 19). In this case, the object o can only satisfy the description if there is some variable assignment ass such that $ass(\boxed{1})$ is the object reached via the sequence of arrows named PHON FIRST from the matrix object in the configuration, and such that $ass(\boxed{2})$ is the object reached via the sequence of arrows named PHON REST FIRST. In addition, these very same objects must also be reachable via the paths DTRS N-DTR PHON FIRST and DTRS H-DTR PHON FIRST respectively.

In linguistic theories, we want to confine ourselves to descriptions whose interpretation is independent of particular variable assignments. This is done by allowing only those descriptions as elements of a linguistic theory which do not contain any unbound occurrences of variables. We write $\mathcal{D}_0{}^{\Sigma}$ for the set of Σ descriptions that satisfy this syntactic requirement.

DEFINITION 2.14 *For each signature* Σ, *for each* Σ *interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$,

D_I *is the total function from* $\mathcal{D}_0^{\Sigma}$ *to the power set of* U, *such that for each* $\delta \in \mathcal{D}_0^{\Sigma}$,

$$D_\mathsf{I}(\delta) = \left\{ u \in \mathsf{U} \;\middle|\; \begin{array}{l} \textit{for some } ass \in \mathsf{Ass}_\mathsf{I}, \\ u \in D_\mathsf{I}^{ass}(\delta) \end{array} \right\}.$$

Definition 2.14 works, because the denotation of a description which does not contain free variables is independent of the choice of the variable assignment.

A Σ *theory* is nothing but a set of descriptions from $\mathcal{D}_0^{\Sigma}$. In its explicit formalization, our toy grammar of Section 1.1 contains three descriptions in its theory. The first description constrains the possible *word* objects. It requires that the phonology of every word be either empty, or contain exactly one *phonstring* object. The second constrains phrases and is simply a formally precise version of the description given in (18). The third element in the theory constrains the interpretation of the relation `append` in the way indicated in (17).

(58) Theory of the grammar assumed in Section 1.1:

a. Constraint on the sort *word*:

$$: \sim word \Rightarrow \left(\begin{array}{l} (:\ \text{PHON}\sim elist) \\ \textbf{or} \\ \left(\begin{array}{l} :\ \text{PHON FIRST}\sim phonstring \\ \textbf{and}\ :\ \text{PHON REST}\sim elist \end{array} \right) \end{array} \right)$$

b. Constraint on the sort *phrase* (= (18) on page 20):

$$: \sim phrase \Rightarrow \left(\begin{array}{l} \mathbf{E}\boxed{1}\ \mathbf{E}\boxed{2}\ \mathbf{E}\boxed{3} \\ :\ \text{PHON} \approx \boxed{3} \\ \textbf{and}\ :\ \text{DTRS H-DTR PHON} \approx \boxed{1} \\ \textbf{and}\ :\ \text{DTRS H-DTR PHON} \approx \boxed{2} \\ \textbf{and}\ (\texttt{append}(\boxed{1},\boxed{2},\boxed{3})\ \textbf{or}\ \texttt{append}(\boxed{2},\boxed{1},\boxed{3})) \end{array} \right)$$

c. Constraint for the relation `append`:

$$\mathbf{A}\boxed{1}\mathbf{A}\boxed{2}\mathbf{A}\boxed{3}$$

$$\left(\begin{array}{l} \texttt{append}(\boxed{1},\boxed{2},\boxed{3}) \Leftrightarrow \\ \left(\left(\begin{array}{l} \boxed{1}\sim elist \\ \textbf{and}\ \boxed{2} \approx \boxed{3} \end{array} \right) \textbf{or} \left(\begin{array}{l} \mathbf{E}\boxed{4}\ \mathbf{E}\boxed{5} \\ \boxed{1}\text{FIRST} \approx \boxed{3}\text{FIRST} \\ \textbf{and}\ \boxed{1}\text{REST} \approx \boxed{4} \\ \textbf{and}\ \boxed{3}\text{REST} \approx \boxed{5} \\ \textbf{and}\ \texttt{append}(\boxed{4},\boxed{2},\boxed{5}) \end{array} \right) \right) \end{array} \right)$$

Notice that none of these descriptions contains a free variable. They are elements of $\mathcal{D}_0^\Sigma$ and, therefore, may be elements of a theory. Using the function D_I, we can define the denotation of a theory.

DEFINITION 2.15 *For each signature* Σ, *for each* Σ *interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$,

Θ_I *is the total function from the power set of* $\mathcal{D}_0^\Sigma$ *to the power set of* U *such that for each* $\theta \subseteq \mathcal{D}_0^\Sigma$,

$$\Theta_\mathsf{I}(\theta) = \left\{ u \in \mathsf{U} \,\middle|\, \begin{array}{l} \text{for each } \delta \in \theta, \\ u \in D_\mathsf{I}(\delta) \end{array} \right\}.$$

The denotation of a theory is defined as the intersection of the denotation of all the descriptions in the theory.

As we have mentioned in Section 1.1, a grammar consists of a signature and a theory. This is reflected in the following definition of what an RSRL *grammar* is.

DEFINITION 2.16 Γ *is a grammar iff*

Γ *is a pair* $\langle \Sigma, \theta \rangle$,
Σ *is a signature and*
$\theta \subseteq \mathcal{D}_0^\Sigma$.

In our example, the grammar consists of the signature in (55) and the theory in (58).

Given a grammar, a *model* of that grammar is taken to be an interpretation with the special property given in Definition 2.17: all elements of the universe must be described by all elements of the theory of the grammar.

DEFINITION 2.17 *For each signature* Σ, *for each* $\theta \subseteq \mathcal{D}_0^{\Sigma}$, *for each* Σ *interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$,

I *is a* $\langle \Sigma, \theta \rangle$ *model iff* $\Theta_{\mathsf{I}}(\theta) = \mathsf{U}$.

Among the descriptions in (58), it is only the third that requires comment. Though it is not an implication, it is of the form that we usually will find for the "definitions" of a relation. In fact, the constraint in (58c) determines which triples appear in the interpretation of the relation `append`. The description is met by each object such that for each triple of components $\langle o_1, o_2, o_3 \rangle$, the triple is in the interpretation of `append` iff (i) o_1 is labelled *elist* and o_2 and o_3 are identical, or (ii) there are objects o_4 and o_5 such that the objects reached by the interpretation of the attribute FIRST on o_1 and o_3 are identical, o_4 and o_5 are the objects reached by interpreting the attribute REST on o_1 and o_3 respectively, and o_4, o_2 and o_5 are in the interpretation of the relation `append`.

Through this summary of the description in (58c), it is clear that it has roughly the effect of the informal characterization given in (17). There is, however, an important difference: whereas in (17), we just talked about lists, these lists are, now, required to be *components* of the described object.

Consider a trivial example to make the difference clear. Let us assume that we have an interpretation that contains the objects of the configurations in Figure 1.1 and Figure 1.5. In this case, we have two *elist* objects in the universe, one is a component of the *word* object, and the other is a component of the matrix object in Figure 1.5. Let us call these two *elist* objects o_1 and o_2 respectively. In (59), all possible triples are listed that consist only of the objects o_1 and o_2.

(59) $\langle o_1, o_1, o_1 \rangle$, $\langle o_1, o_1, o_2 \rangle$, $\langle o_1, o_2, o_1 \rangle$, $\langle o_1, o_2, o_2 \rangle$,
$\langle o_2, o_1, o_1 \rangle$, $\langle o_2, o_1, o_2 \rangle$, $\langle o_2, o_2, o_1 \rangle$, $\langle o_2, o_2, o_2 \rangle$

Under the characterization of the relation `append`, we would expect that all the triples of (59) are part of the interpretation of the relation where the second and the third element of the triple are identical, i.e., the triples given in (60).

(60) $\langle o_1, o_1, o_1 \rangle$, $\langle o_1, o_2, o_2 \rangle$,
$\langle o_2, o_1, o_1 \rangle$, $\langle o_2, o_2, o_2 \rangle$

The constraint on the relation `append` in (58c), however, confirms that those lists are in the relation `append` in the interpretation of the grammar that are all components a particular object. As such, including the constraint in (58c) in our theory ensures that the interpretation of the relation `append` contains the following triples:

(61) $\langle o_1, o_1, o_1 \rangle$,
$\langle o_2, o_2, o_2 \rangle$

Let us extend the interpretation of the grammar to also contain the objects that occur in Figure 1.4, and assume that this interpretation is a model of our grammar. The matrix object of this configuration has two components of sort *elist*, one being reached by the path DTRS N-DTR PHON REST, and one being reached by the path PHON REST REST. As both these *elist* objects are components of the matrix object of this configuration, the relation `append` must hold between them. Let us call them o_3 and o_4. In other words, the constraint in (58c) ensures among other things that the following triples are part of the interpretation of the relation `append`.

(62) $\langle o_3, o_3, o_3 \rangle$, $\langle o_3, o_4, o_4 \rangle$,
$\langle o_4, o_3, o_3 \rangle$, $\langle o_4, o_4, o_4 \rangle$

What all of this shows is that by the way in which we define a relation via a constraint in the theory, we can assure that the intended objects are in the relation, given that the objects are components of a larger object in the universe. This limitation is what is needed for the purpose of a grammar writer who works under the assumption that the grammaticality of a linguistic object depends on this object and its components alone. This assumption is also built into the requirement that the descriptions in a theory may not contain free variables.

So far, we have not included chains in our example theory. Let us next consider a slightly modified theory which makes use of chains. For this purpose, we assume a theory which contains the descriptions (58a) and (58b). Instead of the description (58c), we assume the constraint in (63a). In addition, we add a further constraint in (63b).

(63) a. Extension of the constraint for the relation `append` to include chains:

$$\mathbf{A}\boxed{1}\mathbf{A}\boxed{2}\mathbf{A}\boxed{3}\left(\mathtt{append}(\boxed{1},\boxed{2},\boxed{3}) \Leftrightarrow \left(\begin{array}{l}\left(\begin{array}{l}(\boxed{1}\sim \mathit{elist}\ \mathbf{or}\ \boxed{1}\sim \mathit{echain})\\ \mathbf{and}\ \boxed{2}\approx\boxed{3}\end{array}\right)\\ \mathbf{or}\\ \left(\begin{array}{l}\mathbf{E}\boxed{4}\ \mathbf{E}\boxed{5}\\ (\boxed{1}\text{FIRST}\approx\boxed{3}\text{FIRST}\ \mathbf{or}\ \boxed{1}\dagger\approx\boxed{3}\dagger)\\ \mathbf{and}\ (\boxed{1}\text{REST}\approx\boxed{4}\ \mathbf{or}\ \boxed{1}\triangleright\approx\boxed{4})\\ \mathbf{and}\ (\boxed{3}\text{REST}\approx\boxed{5}\ \mathbf{or}\ \boxed{3}\triangleright\approx\boxed{5})\\ \mathbf{and}\ \mathtt{append}(\boxed{4},\boxed{2},\boxed{5})\end{array}\right)\end{array}\right)\right)$$

b. Finiteness constraint:

$$\mathbf{E}a\ \mathbf{A}\boxed{1}\left(\begin{array}{l}a\sim \mathit{chain}\\ \mathbf{and}\ \boxed{1}\sim \mathit{top}\Rightarrow\left(\begin{array}{l}\mathbf{E}b\ \mathbf{E}c\\ b\dagger\approx\boxed{1}\\ \mathbf{and}\ \mathtt{append}(c,b,a)\end{array}\right)\end{array}\right)$$

When we introduced the quasi-sorts and quasi-attributes for chains in Definition 2.4, we pointed out that we intend to use chains parallel to lists. This becomes obvious in the way we have modified the constraint for `append` in (63a). There, we added that the relation `append` should not only hold between components of the described object, but also between the chains of components of the described object.

For illustration, we will consider the use of this definition of the relation `append` in the description in (63b). If we add this constraint to our grammar, it has the effect of enforcing that in each model of our grammar, each linguistic object only has a finite number of components. In (63b), we have used a boxed integer ($\boxed{1}$) for a variable that is assigned an object of the universe, and lower-case letters (a, b, c) for variables that are assigned chains. The constraint describes an object iff there is some chain a formed of components of this object and for each component $\boxed{1}$ of this object, there are chains b and c such that $\boxed{1}$ is the first element on the chain b and c, b and a stand in the relation `append`.

Let us consider whether the matrix object of the configuration in Figure 1.1 matches this description. The configuration consists of five objects. Let us refer to the matrix object as m, the *synsem* object as s, the *nelist* object as n, the *elist* object as e and the *walks* object as w. The quantification, then, is over the objects m, s, n, e and w as well as over all the finite sequences that consist only of these objects. Let us assume that the chain a is given as $\langle m,s,n,e,w\rangle$. Then, the object m is described by (63b) iff the consequent of the implication is satisfied for each $\boxed{1}$, where $\boxed{1}$ ranges over the components of m.

Let us assume an assignment ass such that $ass(\boxed{1}) = m$. In this case, we chose an assigment ass' which is identical to ass, but assigns the variables b and c the chains $\langle m,s,n,e,w\rangle$ and $\langle\rangle$ respectively. Then, the triple $\langle c,b,a\rangle$, i.e., $\langle\langle\rangle,\langle m,s,n,e,w\rangle,\langle m,s,n,e,w\rangle\rangle$, is in the interpretation of the relation `append` by virtue of the constraint in (63a).

Let us, next, assume an assignment ass such that $ass(\boxed{1}) = s$. In this case, we chose an assigments ass' which again is like ass, but assigns the variables b and c the chains $\langle s, n, e, w\rangle$ and $\langle m\rangle$ respectively. Again, it is easy to see that the first element on b is $\boxed{1}$ and that c, b and a stand in the relation `append`. We can show this for the other components of the matrix object m in a similar way.

The constraint given in (63b) is writen to ensure that the chain a contains at least one instance of every component of the described object. As chains are elements of U^*, they are required to be finite. This means that if there is a chain that contains all components of some object, then this object must have a finite number of components.

Without a constraint such as (63b), our grammar would allow for infinite objects. Consider for example a *phrase* object with the following property: all the paths DTRS H-DTR, DTRS H-DTR DTRS H-DTR , . . . lead to distinct objects, all of which are labelled *phrase*. Such an object has an infinite number of components, therefore there exists no chain containing every component of this object. Thus, it is not described by the constraint in (63b).

After these illustrations of the way a theory is interpreted, and this example of the use of chains in a theory, we can address a more general issue, the question of what should be considered the denotation of a grammar.

King 1999 considers the relation between a grammar and the phenomenon it is intended to capture, i.e., language. According to King, this relation should be one of denotation. In particular, the language should be a model of the grammar a linguist writes. For a grammar to be adequate, the language must be a particular kind of model, an *exhaustive model* as defined in 2.18.

DEFINITION 2.18 *For each signature* Σ, *for each* $\theta \subseteq \mathcal{D}_0^\Sigma$, *for each* Σ *interpretation* I,

I *is an exhaustive* $\langle\Sigma,\theta\rangle$ *model iff*
I *is a* $\langle\Sigma,\theta\rangle$ *model, and*
for each $\theta' \subseteq \mathcal{D}_0^\Sigma$, *for each* Σ *interpretation* I',
if I' *is a* $\langle\Sigma,\theta\rangle$ *model and* $\Theta_{\mathsf{I}'}(\theta') \neq \emptyset$ *then* $\Theta_{\mathsf{I}}(\theta') \neq \emptyset$.

The notion of exhaustive model is taken over from SRL. Richter 2000 contains the proof, due to Richter and King 1997, that a non-empty exhaustive model of the grammar exists for every RSRL grammar whose theory has an non-empty model.

The definition of an exhaustive model implies that given a grammar Γ, every exhaustive model of Γ contains at least one copy of every configuration that is licensed by Γ. For linguistic purposes this means that the explored language should be an exhaustive model of the proposed grammar. We consider a grammar with this property *observationally adequate*.

We will briefly show how the definition of an exhaustive model captures the basic idea of observational adequacy. Assume a grammar Γ which has the English language as an exhaustive model. Then, because English is a model, the grammar licenses all grammatical sentences of English, i.e., it does not underlicense. We then have to show that Γ does not overlicense either. For this purpose, take a description δ which describes only ungrammatical sentences. Let us assume that Γ also licenses sentences described by δ, then there must be a model M of Γ which contains some of the ungrammatical sentences described by δ, i.e., the theory interpretation of $\{\delta\}$ would be a non-empty set in M. By Definition 2.18, this means that δ must also describe some objects in the exhaustive model of Γ, English in our example. But, as English does not contain ungrammatical sentences, we derived a contradiction. Therefore, if English is an exhaustive model of Γ, then Γ does not overlicense.

Using the notion of an exhaustive model, it is easy to show that our toy grammar as given by the signature in (55) and the theory in (58) is not an adequate grammar of English.

First its ontology is not adequate, as there is no way to have objects that would correspond to words of English other than *every*, *man*, *Mary*, and *walks*. Second, the grammar also heavily overgenerates. Consider the configuration of objects given in Figure 1.4 and assume a configuration which is just like that, but where the phonstrings are used in the reverse order, i.e., the phonology of the phrase would be ⟨*man*, *every*⟩ instead of ⟨*every*, *man*⟩. In English, there is no phrase with such a phonology. As such a configuration of objects would, however, be compatible with the signature in (55) and respect all elements of the theory in (58), it is necessarily part of each exhaustive model of this grammar.

Similarly, if we do not explicilty exclude objects with an infinite number of components via a description as the one given in (63b), every exhaustive model of the grammar will contain such infinite objects.

In Section 3.2, we will show for an RSRL grammar that the language that it is supposed to describe really is an exhaustive model of the grammar. While this proof is hard to give for natural languages, it is relatively simple for an RSRL grammar which describes a formal language. In the case of Chapter 3 this is the semantic representation language Ty2.

2.2. AVMs and Other Conventions

RSRL as defined in the preceding section, has the functionality needed for HPSG grammars. The descriptions do not, however, look like those used in most HPSG publications, or like those that we have used in Section 1.1. This conflict is resolved in Richter 1999, where an AVM language is defined which is equivalent to RSRL as defined above. In this thesis, we do not repeat the definition of AVMs for RSRL, but we will use both the AVM and the standard notation.

In (57) we have already seen a number of explicit RSRL descriptions that correspond to AVMs as given in Section 1.1. There, we saw how the tags used in the AVMs are interpreted as variables that are bound by quantifiers. Richter 1999 also introduces many useful abbreviatory conventions. One of these conventions is that variables which are not bound explicitly in a description, are interpreted as bound by an existential quantifier which has scope over the entire description. Applying this convention, the two AVMs in (14) (page 16) and (15) are interpreted the same way, as the free variables, $\boxed{1}$, $\boxed{2}$ and $\boxed{3}$ are interpreted as being bound by a wide-scope existential quantifier, such as the quantifiers added explicitly in (15).

As a further example, consider the following AVM version of the principle in (18) (and (58b)) which constrains the phonology of a phrase to the concatenation of the phonologies of its daughters.

(64) a. $$\textit{phrase} \Rightarrow \left(\begin{array}{l} \begin{bmatrix} \textsc{phon} & \boxed{3} \\ \textsc{dtrs} & \begin{bmatrix} \textsc{h-dtr} & \boxed{1} \\ \textsc{n-dtr} & \boxed{2} \end{bmatrix} \end{bmatrix} \\ \textbf{and}\ (\mathtt{append}(\boxed{1},\boxed{2},\boxed{3})\ \textbf{or}\ \mathtt{append}(\boxed{2},\boxed{1},\boxed{3})) \end{array} \right)$$

b. $$\mathbf{E}\boxed{1}\ \mathbf{E}\boxed{2}\ \mathbf{E}\boxed{3} \left(:\sim\textit{phrase} \Rightarrow \left(\begin{array}{l} :\textsc{phon} \approx \boxed{3} \\ \textbf{and}\ :\textsc{dtrs h-dtr phon} \approx \boxed{1} \\ \textbf{and}\ :\textsc{dtrs h-dtr phon} \approx \boxed{2} \\ \textbf{and}\ (\mathtt{append}(\boxed{1},\boxed{2},\boxed{3})\ \textbf{or}\ \mathtt{append}(\boxed{2},\boxed{1},\boxed{3})) \end{array} \right) \right)$$

The AVM notation in (64a) contains the logical symbols that we also use in regular RSRL, i.e., ⇒, **and**, **or**. In addition, it contains usual AVMs, i.e., large square brackets which enclose lines that consist of a sequence of attributes followed by some (AVM) description. In (64b) we have given the regular RSRL syntax that corresponds to the AVM notation in (a). Comparing the two descriptions, it can be seen that the logical symbols are used in the same way in both syntaxes, the bracketed structure is, however, undone in regular RSRL and corresponds to a sequence of conjuncted path identities or sort assignments. We also see that there are free variables in (64a). Following the convention mentioned above, these variables are existentially bound by quantifiers which have wide scope.[4]

In our RSRL definitions, we saw that terms may either start with the special symbol ": " or with a variable. For terms that start with variables, Richter 1999 allows AVMs which are preceded by a variable. In (65), an example is given for a description which contains a variable. The description denotes all those signs which have a component with an empty phonological contribution.

(65) a. $:\sim$*sign* **and** $\mathbf{E}\boxed{1}\,(\boxed{1}$PHON$\sim$*elist*)
b. $\left[\textit{sign}\right]$ **and** $\boxed{1}\left[\text{PHON}\ \langle\rangle\right]$

The example in (65) can also be used to illustrate the effect of restricted quantification. For this purpose, we contrast the description in (65) with a corresponding description which only uses the special symbol ": ".

(66) a. $:\sim$*sign* **and** $:$ PHON$\sim$*elist*
b. $\left[\textit{sign}\right]$ **and** $\left[\text{PHON}\ \langle\rangle\right]$

If we compare the (a) descriptions in (65) and (66), it is clear that syntactically, variables (tags, in this case) are treated just as the symbol ": " in the construction of terms. In the AVM language, there is a difference: the symbol ": " can be omitted, whereas a variable must be stated explicitly. On the semantic side, the descriptions in (65) denote potentially more non-isomorphic configurations of linguistic objects than the ones in (66). Assume that we have an HPSG grammar like the one presented in Pollard and Sag 1994. In the model of that grammar, the description in (66) denotes exactly all signs which have an empty phonology, i.e., traces (pp. 161ff.), the null complementizer (pp. 126f.) and the null relativizer (pp. 213ff.). In contrast to this, the description in (65) describes all signs which are mapped by some path "$:\pi$" into a sign with empty phonology. Such a path can simply be ": ". Therefore, all signs described by (66) are in the denotation of (65) as well. Yet, (65) additionally describes all signs which (non-reflexively) dominate some null element.

In our small toy grammar, given by the signature in (55) and the theory in (58), we also have included the possibility of having words with an empty phonology (see the first disjunct in the consequent of description (58a)). Such a word would be described by (65) as well as by (66). The same is true for a phrase that dominates only words with empty phonology. Given our toy grammar, this kind of phrase is not excluded. But the description in (66) also describes phrases which do not have an empty phonology, but which dominate a sign with empty phonology.

We should, next, turn to relations. Richter 1997 (p. 40) defines an abbreviatory convention for RSRL calls. According to this convention, relational calls may contain arbitrary terms instead of just variables as in Definition 2.10. Such calls abbreviate complex expressions with a series of existential quantifiers over some new variables. The quantifiers have scope over a relational call that uses exactly these variables conjoined with identities that equate these variables with the terms in the abbreviated description. This convention can be illustrated with the constraint in (58b), i.e., a description that is met by phrases whose phonology is the concatenation of the phonologies of their daughters. We have already seen an AVM version of this constraint in (18), and an explicit RSRL encoding in (58b).

---

[4]Notice that the description in (64b) differs syntactically from the description in (58b). In the latter case the existential quantifiers only take scope in the consequent of the implication. But, as there are no occurrences of the variables in the antecedent of the implication, the two descriptions are logically equivalent.

In (67), we re-state this constraint using the above mentioned abbreviatory convention and its explicit expansion.

(67) a. $$\textit{phrase} \Rightarrow \left(\begin{array}{l}(\texttt{append}(\textsc{:dtrs h-dtr phon},\ \textsc{:dtrs n-dtr phon},\ \textsc{:phon}) \\ \textbf{or} \\ \texttt{append}(\textsc{:dtrs n-dtr phon},\ \textsc{:dtrs h-dtr phon},\ \textsc{:phon}))\end{array}\right)$$

b. $$\textit{phrase} \Rightarrow \left(\begin{array}{l}\mathbf{E}\boxed{1}\ \mathbf{E}\boxed{2}\ \mathbf{E}\boxed{3} \\ \left(\begin{array}{l}\textsc{:dtrs h-dtr phon} \approx \boxed{1} \\ \textbf{and}\ \textsc{:dtrs n-dtr phon} \approx \boxed{2} \\ \textbf{and}\ \textsc{:dtrs phon} \approx \boxed{3} \\ \textbf{and}\ \texttt{append}(\boxed{1},\boxed{2},\boxed{3})\end{array}\right) \\ \textbf{or} \\ \mathbf{E}\boxed{1}\ \mathbf{E}\boxed{2}\ \mathbf{E}\boxed{3} \\ \left(\begin{array}{l}\textsc{:dtrs h-dtr phon} \approx \boxed{1} \\ \textbf{and}\ \textsc{:dtrs n-dtr phon} \approx \boxed{2} \\ \textbf{and}\ \textsc{:dtrs phon} \approx \boxed{3} \\ \textbf{and}\ \texttt{append}(\boxed{2},\boxed{1},\boxed{3})\end{array}\right)\end{array}\right)$$

In the explicit RSRL encoding, each call is expanded separately, as indicated by the two disjuncts in (67b). Since both disjuncts look alike, we will concentrate on the first. By expanding the first relation call in (67a), existential quantifiers are introduced that bind the variables $\boxed{1}$, $\boxed{2}$ and $\boxed{3}$. These are new variables which do not occur in the abbreviated description. The relation call, $\texttt{append}(\boxed{1},\boxed{2},\boxed{3})$ uses exactly theses variables in the argument slots. In the full description, the relation call is conjoined with identities that specify that in the scope of the existential quantifiers, the variables $\boxed{1}$, $\boxed{2}$ and $\boxed{3}$ should be interpreted in such a way that they have the same denotation as the interpretation of the terms : DTRS H-DTR PHON, : DTRS N-DTR PHON and : PHON on the described object respectively.

We have already seen in (58) that a relation such as `append` is defined by a description in the theory. The way in which exhaustive models are defined, guarantees that in every exhaustive model the set $\mathsf{R}(\texttt{append})$ contains are components or chains of components of an element of U, and (ii) z is the concatenation of x and y. A description that defines this relation was given in (58c) (and (63a)).

Instead of stating an explicit principle to define a relation, it is often convenient to split up the definition into *clauses*. For the relation `append`, this is done in (68), again for both RSRL syntax and the AVM language. A special symbol "$\overset{\forall}{\Longleftarrow}$" is used to separate the head from the body of a clause. The use of this symbol shows that a clause is not a regular RSRL description but a notational convention.

(68) a. $$\texttt{append}(\boxed{1},\boxed{2},\boxed{3}) \overset{\forall}{\Longleftarrow} \left(\begin{array}{l}\boxed{1}{\sim}\textit{elist} \\ \textbf{and}\ \boxed{2} \approx \boxed{3}\end{array}\right)$$

$$\texttt{append}(\boxed{1},\boxed{2},\boxed{3}) \overset{\forall}{\Longleftarrow} \left(\begin{array}{l}\boxed{1}\textsc{first} \approx \boxed{3}\textsc{first} \\ \textbf{and}\ \texttt{append}(\boxed{1}\textsc{rest},\boxed{2},\boxed{3}\textsc{rest})\end{array}\right)$$

b. $$\texttt{append}(\boxed{1},\boxed{2},\boxed{3}) \overset{\forall}{\Longleftarrow} \left(\begin{array}{l}\boxed{1}\left[\textit{elist}\right] \\ \textbf{and}\ \boxed{2} \approx \boxed{3}\end{array}\right)$$

$$\texttt{append}(\boxed{1},\boxed{2},\boxed{3}) \overset{\forall}{\Longleftarrow} \left(\begin{array}{l}\boxed{1}\left[\begin{array}{ll}\textsc{first} & \boxed{4} \\ \textsc{rest} & \boxed{5}\end{array}\right] \textbf{and}\ \boxed{3}\left[\begin{array}{ll}\textsc{first} & \boxed{4} \\ \textsc{rest} & \boxed{6}\end{array}\right] \\ \textbf{and}\ \texttt{append}(\boxed{5},\boxed{2},\boxed{6})\end{array}\right)$$

The definition of the relation `append` is expressed in two clauses. The clause definition of an n-ary relation ρ corresponds to an RSRL description that starts with n universal quantifiers, one for each argument position of the relation. In the scope of these quantifiers, there is an equivalence. At one side of the equivalence, there is a call to the relation ρ with the variables bound by the quantifiers in their corresponding argument slots. On the other side of the equivalence arrow, there is a disjunction which contains the bodies of the clauses. In the AVM notation in (68b), there are free variables ($\boxed{4}$, $\boxed{5}$, $\boxed{6}$) in the second clause. Such free variables are interpreted as being existentially bound by a quantifier which has scope over the body of the clause.

Finally, there are some conventions for lists and chains. As we have seen above, in (63a), chains are used in a way very similar to lists. Therefore, we often do not want to give concern to the distinction between lists and chains. This formal vagueness will usually not cause any problems in understanding the descriptions used in this thesis.

In many cases, we want to equate a list and a chain. But, as lists and chains are different kinds of entities, this cannot be done directly. Instead, we define a relation `list-chain-ident` which expresses the intended identities. A pair $\langle e_1, e_2 \rangle$ such that both e_1 and e_2 are elements of $\overline{\mathsf{U}}$ stands in the relation `list-chain-ident` if and only if the following holds: if e_1 and e_2 are elements of U, then $e_1 = e_2$, otherwise, the first element of e_1 is identical to the first element of e_2, and the rest of e_1 stands in the relation `list-chain-ident` with the rest of e_2. This relation is defined in (69).

(69)
$$\texttt{list-chain-ident}(\boxed{1},\boxed{2}) \stackrel{\forall}{\Longleftarrow} \left(\begin{array}{l} (\textbf{not } \boxed{1}{\sim}\textit{chain}) \\ \textbf{and } \ (\textbf{not } \boxed{2}{\sim}\textit{chain}) \\ \textbf{and } \ \boxed{1} \approx \boxed{2} \end{array}\right)$$

$$\texttt{list-chain-ident}(\boxed{1},\boxed{2}) \stackrel{\forall}{\Longleftarrow} \left(\begin{array}{l} (\boxed{1}{\sim}\textit{chain} \ \textbf{or} \ \boxed{2}{\sim}\textit{chain}) \\ \textbf{and} \ \ \textbf{E}\boxed{3} \left(\begin{array}{l} (\boxed{1}\textsc{first} \approx \boxed{3} \ \textbf{or} \ \boxed{1}\dagger \approx \boxed{3}) \\ \textbf{and} \ \ (\boxed{2}\textsc{first} \approx \boxed{3} \ \textbf{or} \ \boxed{2}\dagger \approx \boxed{3}) \end{array}\right) \\ \textbf{and} \\ \textbf{E}\boxed{4}\textbf{E}\boxed{5} \left(\begin{array}{l} (\boxed{1}\textsc{rest} \approx \boxed{4} \ \textbf{or} \ \boxed{1}\triangleright \approx \boxed{4}) \\ \textbf{and} \ \ (\boxed{2}\textsc{rest} \approx \boxed{5} \ \textbf{or} \ \boxed{2}\triangleright \approx \boxed{5}) \\ \textbf{and} \ \ \texttt{list-chain-ident}(\boxed{4},\boxed{5}) \end{array}\right) \end{array}\right)$$

With the formal definitions given in the preceding section and the notational variant sketched in the present section, we are now able to write HPSG grammars in the way commonly used in the HPSG literature, i.e., using AVMs. We can augment these AVMs with relations, quantification and chains. In the following section, we will depart from the little toy grammar that we have used in the presentation of the formalism and give the grammar that we want to use to build a semantic fragment on.

2.3. The Syntactic Analysis

In this section, we will provide the syntactic analysis that will be assumed throughout the rest of this thesis. It will serve as a syntactic background for the integration of Lexicalized Flexible Ty2 in Chapter 4 and for the analysis of idiomatic expressions in Chapter 8.

The point of departure for our syntactic analysis is the grammar presented in Pollard and Sag 1994. We will adopt most parts of the signature assumed there and the principles used there for extraction. Nonetheless, our analysis will differ in several respects: First, as we want to use the semantic framework of LF-Ty2, we will introduce a new sort hierarchy below the sort *content*, and we dismiss with the attributes QSTORE and RETRIEVED.[5] Second, we assume an overall binary-branching syntactic structure. As a consequence, we will not use the attributes C-DTRS, ADJ-DTR, FILLER-DTR etc. Instead, we will assume a H(EAD)-DTR and a N(ON-HEAD)-DTR attribute. Binary branching will make it easier to specify the principles of grammar, but we do not claim that a binary branching structure is empirically

[5]For the use of these attributes within the grammar of Pollard and Sag 1994, see Section 1.2.

superior to the syntactic structure assumed in Pollard and Sag 1994. Furthermore, as we will mainly be concerned with transitive and intransitive verbs, there is not much of a difference between our structures and those assumed in Pollard and Sag 1994. Third, we will assume a traceless analysis of extraction, similar in spirit to the analysis sketched in Section 9.5 of Pollard and Sag 1994, using the architecture for *Lexical Rules* defined in Meurers 2000. Again, this move is primarily motivated to keep the principles needed for combinatorial semantics as simple as possible, i.e., as sketched in Section 1.3.

In the first part of this section, we will give the signature assumed for the syntactic parts of signs, being explicit only in those parts that differ from the definitions in the appendix of Pollard and Sag 1994. After this, we will give the principles needed for the construction of simple clauses. In Section 2.3.2, an analysis of passive and extraction will be presented, which employs the technique of Meurers 2000 to encode Lexical Rules in HPSG.

2.3.1. General Architecture. Just as was the case for the toy grammar of Section 1.1, the grammar that we define in this section is mainly concerned with objects of the sort *sign*. We give the sort hierarchy and the appropriateness conditions for this sort in (70).

(70) Sort hierarchy and appropriateness conditions below the sort *sign* (as used throughout the rest of Part I):

sign
PHON *list*
SYNSEM *synsem*

word
STORE *list*

phrase
DTRS *const-struc*

As can be seen, the toy sort hierarchy that we had defined in (11) on page 13 is almost identical with the real signature given in (70). The only difference lies in the fact that in the latter signature, we also declare a new attribute STORE appropriate for the sort *word*. This attribute is taken from Meurers 2000 (p. 124) and is used to encode a mechanism that has the effect of *Lexical Rules*. In the present subsection, we do not elaborate on this particular attribute, but we will come back to it in Section 2.3.2.

Again, following our toy grammar of Sections 1.1 and 2.1, the PHON value of a sign is a list of *phonstring* objects. We are not explicit about all the subsorts of *phonstring* that we assume, but they contain at least the phonology of the words of our semantic fragment of Section 1.3, i.e., *man*, *woman*, *Mary*, *walks*, *loves*, *every*, *some*, *believes*, and *that*.

For the sort *synsem*, we follow the analysis of Pollard and Sag 1994. There are two attributes appropriate to this sort: LOC(AL) and NONL(OCAL). For the data analyzed in this thesis, the non-local part is only needed for complement extraction, i.e., we will only use the SLASH attribute. These declarations are given in (71).

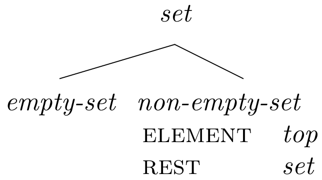
(71) a. Appropriateness conditions on the sort *synsem*:
synsem LOCAL *local*
NONLOCAL *nonlocal*

b. Appropriateness conditions on the sort *nonlocal*:
nonlocal INHERITED *nonlocal1*
TO-BIND *nonlocal1*

c. Appropriateness conditions on the sort *nonlocal1*:
nonlocal1 SLASH *set*

Following Pollard and Sag 1994, the attribute SLASH is declared to be set-valued in (71c). Richter 2000 (Section 4.4) shows how finite sets can be encoded in RSRL. For our purpose, we simply assume the following list-like sort hierarchy and appropriateness conditions for the sort *set* and assume that the grammar is written in such a way that the order of the elements in a *set* object does not matter.

(72) Sort hierarchy and appropriateness conditions for the sort *set*:



For the sort *local*, we restrict our attention to two attributes, CAT(EGORY) and CONT(ENT). The third attribute, CONTEXT, assumed to be appropriate to this sort in Pollard and Sag 1994 will not be considered. In the present section, we are only concerned with the CAT values, the following chapters of Part I of the thesis will focus on the CONTENT value. For the time being, we assume that the sort *content* does not have any subsorts nor attributes defined on it. The CAT value of a *local* object is a *category* object. In this thesis, we use all three attributes that Pollard and Sag 1994 assume for this sort: HEAD, SUBCAT, and MARKING, with the values as introduced there, i.e., the HEAD value specifies the syntactic category, the SUBCAT value is a list of *synsem* objects that indicates the valence, and the MARKING value is needed to differentiate between *that*-clauses (which are saturated verbal projections with the MARKING value *that*) and *that*-less clauses, i.e., saturated verbal projections with the MARKING value *unmarked*. The parts of the signature just mentioned are summarized in (73a).

(73) a. Appropriateness conditions on the sort *local*:

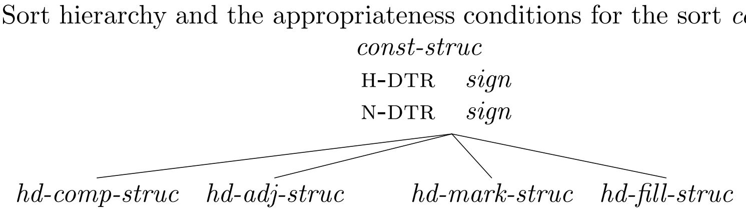
local	CATEGORY	*category*
	CONTENT	*content*

b. Appropriateness conditions on the sort *category*:

category	HEAD	*head*
	SUBCAT	*list*
	MARKING	*marking*

So far, we have mentioned the ontology assumed below the PHON and the SYNSEM values of a sign. Next, we must consider the DTRS value. Just as in the toy grammar, we assume that the attribute DTRS is defined on the sort *phrase* and the sort *constituent-structure* is appropriate for this attribute. In (74) we give the sort hierarchy and the appropriateness conditions for the sort *const-struc*.

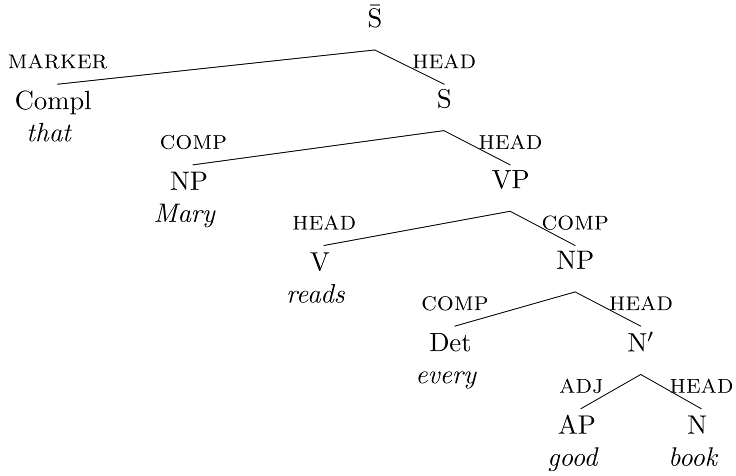
(74) Sort hierarchy and the appropriateness conditions for the sort *const-struc*:



As in our toy grammar, the attributes H(EAD)-DTR and N(ON-HEAD)-DTR are appropriate to the sort *const-struc*, and the sort *sign* is appropriate for this attribute.[6] We use the subsorts of *const-struc* to differentiate between different possibilities to combine signs in syntax. We assume that a syntactic head can combine with a complement in a *head-complement-struc* (*hd-comp-struc*), with an adjunct in a *hd-adj-struc*, with a marker in a *hd-mark-struc* and with a filler in a *hd-fill-struc*.

[6] In the grammar of Pollard and Sag 1994 most non-head daughters are assumed to be phrases. We allow words as daughters as well to avoid non-branching phrases.

FIGURE 2.1. The structure of sentence (75a):



With the declarations in (74), we have introduced all the sorts and attributes that we assume for the syntactic analysis and that differ from Pollard and Sag 1994. We can now present some principles of the grammar. In this section, we show the principles needed to analyze the sentences in (75).

(75) a. that Mary reads every good book.
b. John is loved.
c. John, Mary loved.

In (75a) we have a sentence which exhibits most of the kinds of constituent structures assumed in this thesis. In Figure 2.1 the different subsorts of *const-struc* are indicated by the labeling of the branches. The noun *book* combines in a head-adjunct structure with the adjective *good*; this N′ combines with the determiner *every* in a head-complement structure. Similarly, the resulting NP combines with the verb *reads* as a head-complement structure and so does the resulting VP with the subject *Mary*. Finally, the complementizer *that* is added to the sentence in a head-marker structure.

Tree structures like Figure 2.1 are descriptions of phrases. The description at the branch with label HEAD is a description of the DTRS H-DTR value of a phrase, the description at the other branch is a description of the DTRS N-DTR value. The label of this other branch is used to indicate the sort of the DTRS value of the described sign, i.e., in the case of the label MARKER, the DTRS value of the described phrase is of sort *hd-marker-struc*. Analoguously for the other lables.

In addition, we use syntactic category labels for the nodes and we indicate the phonology of the words at terminal nodes. Later, we will also indicate the logical form of a node as a term of Ty2, just as we did in Section 1.3.3. We will also often include AVM descriptions in nodes to emphasize specific details of the HPSG encoding.

Sentence (75b) is a passive sentence. The structure of this sentence is indicated in Figure 2.2. It only involves combinations of signs in head-complement structures. In (75c) we have a topicalization structure. There, the topicalized constituent is combined with the rest of the clause in a head-filler structure. The structure of this sentence is sketched in Figure 2.3. In this subsection, we will only be concerned with the structure of sentence (75a). In Section 2.3.2 we will come back to the other two example sentences.

FIGURE 2.2. The structure of sentence (75b):

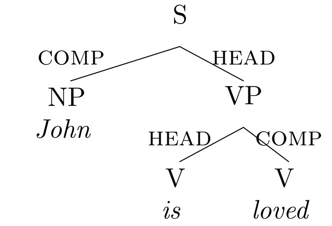
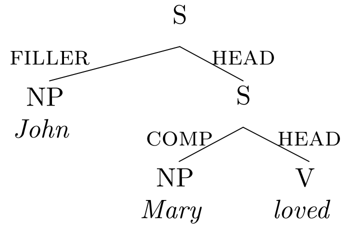


FIGURE 2.3. The structure of sentence (75c):



In order to write a grammar which licenses a fragment of English that consists of sentences such as (75a), we must furnish the theory with a number of principles. One of the central principles of an HPSG grammar is the WORD PRINCIPLE. It contains descriptions of the possible words. In fact, the way we use the WORD PRINCIPLE, it only describes those words that have an empty STORE value. We will see in Section 2.3.2 that having an empty STORE value means for a word that it is not derived through a lexical rule.

(76) The WORD PRINCIPLE:

$$\begin{bmatrix} \textit{word} \\ \text{STORE} \ \textit{elist} \end{bmatrix} \Rightarrow \left(\text{LE}_1 \ \textbf{or} \ \ldots \textbf{or} \ \ldots \text{LE}_m\right)$$

The WORD PRINCIPLE as given in (76) is an implication with a huge disjunction in its consequent. We call each of the disjuncts in the consequent a *lexical entry.* A lexical entry is a description of a word. In (77), we give some examples of lexical entries. The descriptions given in (77) are not complete lexical entries. They only mention those parts of a lexical entry that will be relevant for us. In (77) we give (such sketches of) lexical entries for the words that occur in sentence (75a).

(77) a. Parts of the lexical entry of the word *book*:

$$\begin{bmatrix} \textit{word} \\ \text{PHON} \ \langle \textit{book} \rangle \\ \text{SYNS} \begin{bmatrix} \text{LOC} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{SUBCAT} & \left\langle \begin{bmatrix} \text{LOC CAT HEAD} \ \textit{determiner} \end{bmatrix} \right\rangle \\ \text{MARKING} & \textit{unmarked} \end{bmatrix} \end{bmatrix} \\ \text{NONL} \begin{bmatrix} \text{INHER} & \begin{bmatrix} \text{SLASH} \ \textit{eset} \end{bmatrix} \\ \text{TO-BIND} & \begin{bmatrix} \text{SLASH} \ \textit{eset} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{STORE} \ \textit{elist} \end{bmatrix}$$

b. Parts of the lexical entry of the word *good*:

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{good} \rangle \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{adj} & \\ \text{MOD} & \begin{bmatrix} \textit{synsem} & \\ \text{LOC CAT HEAD} & \textit{noun} \end{bmatrix} \end{bmatrix} \\ \text{SUBCAT} & \textit{elist} \\ \text{MARKING} & \textit{unmarked} \end{bmatrix} \end{bmatrix} \\ \text{NONL} & \begin{bmatrix} \text{INHER} & \begin{bmatrix} \text{SLASH} & \textit{eset} \end{bmatrix} \\ \text{TO-BIND} & \begin{bmatrix} \text{SLASH} & \textit{eset} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{STORE} & \textit{elist} \end{bmatrix}$$

c. Parts of the lexical entry of the word *every*:

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{every} \rangle \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{determiner} & \\ \text{SPEC} & \begin{bmatrix} \textit{synsem} & \\ \text{LOC CAT} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{SBC} & \left\langle \begin{bmatrix} \textit{synsem} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{SUBCAT} & \textit{elist} \\ \text{MARKING} & \textit{unmarked} \end{bmatrix} \end{bmatrix} \\ \text{NONL} & \begin{bmatrix} \text{INHER} & \begin{bmatrix} \text{SLASH} & \textit{eset} \end{bmatrix} \\ \text{TO-BIND} & \begin{bmatrix} \text{SLASH} & \textit{eset} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{STORE} & \textit{elist} \end{bmatrix}$$

d. Parts of the lexical entry of the word *reads*:

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{reads} \rangle \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{SBC} & \left\langle \begin{bmatrix} \text{LOC CAT} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{SBC} & \langle\rangle \end{bmatrix} \end{bmatrix}, \begin{bmatrix} \text{LOC CAT} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{SBC} & \langle\rangle \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{MARKING} & \textit{unmarked} \end{bmatrix} \end{bmatrix} \\ \text{NONL} & \begin{bmatrix} \text{INHER} & \begin{bmatrix} \text{SLASH} & \textit{eset} \end{bmatrix} \\ \text{TO-BIND} & \begin{bmatrix} \text{SLASH} & \textit{eset} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{STORE} & \textit{elist} \end{bmatrix}$$

e. Parts of the lexical entry of the word *Mary*:

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{Mary} \rangle \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{SUBCAT} & \textit{elist} \\ \text{MARKING} & \textit{unmarked} \end{bmatrix} \end{bmatrix} \\ \text{NONL} & \begin{bmatrix} \text{INHER} & \begin{bmatrix} \text{SLASH} & \textit{eset} \end{bmatrix} \\ \text{TO-BIND} & \begin{bmatrix} \text{SLASH} & \textit{eset} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{STORE} & \textit{elist} \end{bmatrix}$$

f. Parts of the lexical entry of the word *that*:

$$\begin{bmatrix} \textit{word} \\ \text{PHON} & \langle \textit{that} \rangle \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{marker} \\ \text{SPEC} & \begin{bmatrix} \textit{synsem} \\ \text{LOC CAT} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{SBC} & \langle\rangle \\ \text{MARK} & \textit{unmarked}\end{bmatrix}\end{bmatrix}\end{bmatrix} \\ \text{SUBCAT} & \textit{elist} \\ \text{MARKING} & \textit{that}\end{bmatrix}\end{bmatrix} \\ \text{NONL} & \begin{bmatrix} \text{INHER} & \begin{bmatrix}\text{SLASH} & \textit{eset}\end{bmatrix} \\ \text{TO-BIND} & \begin{bmatrix}\text{SLASH} & \textit{eset}\end{bmatrix}\end{bmatrix}\end{bmatrix} \\ \text{STORE} & \textit{elist} \end{bmatrix}$$

The lexical entries given in (77) are just like those assumed in Pollard and Sag 1994, if we ignore the CONTENT specification. The lexical entry of the noun *book* in (77a) and that of the proper name *Mary* in (77e) are almost alike, but there is one element on the SUBCAT list of the noun. The proper name, however, is saturated. We follow the analysis of Pollard and Sag 1994 (p. 49), which assumes that nouns select their determiners through the attribute SUBCAT.

In the lexical entry for the verb *reads* given in (77d) we also specify a non-empty SUBCAT list. Just as in Pollard and Sag 1994 (p. 29), we assume that both the subject and the direct object of a transitive verb are on the SUBCAT list.

The lexical entry of the adjective *good* in (77b) specifies that the SUBCAT list of the adjective is empty. However, a *synsem* object is required to appear as the MOD value. This kind of selection is taken over from Pollard and Sag 1994 as well (p. 55).

In (77) we also have lexical entries of two functional words, the determiner *every* in (77c) and the marker *that* in (77f). In both cases, there is a *synsem*-valued attribute SPEC appearing inside the HEAD value. For the determiner, the SPEC value specifies the SYNSEM value of the noun with which the determiner combines. In the case of the complementizer the SPEC value indicates the verbal projection which the complementizer attaches to. Again, we follow Pollard and Sag 1994 in the use of the SPEC value (p. 46 for complementizers and p. 48 for determiners).

Now that we have seen the lexical entries needed for the analysis of sentence (75a), we can turn to some principles for phrases. Most of these principles can be taken almost directly from Pollard and Sag 1994, such as the HEAD FEATURE PRINCIPLE (HFP), the MARKING PRINCIPLE (MP), and the SPEC PRINCIPLE (SpP).[7] Due to our assumption of a strictly binary branching syntactic structure, we are forced to change the IMMEDIATE DOMINANCE PRINCIPLE (IDP). In the analysis of Pollard and Sag 1994, all complements of a head are realized as complement daughters within a single phrase. As an effect, Pollard and Sag 1994 assume non-branching structures for heads that do not have complements, and multiply branching structures for head with more than one complement. A second difference between our system and that of Pollard and Sag 1994 is that we do not use a separate SUBCATEGORIZATION PRINCIPLE, but, instead, encode the inheritance of valence information directly in the ID-Schemata.

We will first consider the IDP. Parallel to the WORD PRINCIPLE, the IDP takes the form of an implication with a disjunction in its consequent. In contrast to the WP, however, the consequent only contains a very small number of disjuncts. In our case, there is one disjunct for each subsort of *const-struc*. The IDP is given schematically in (78).

[7] In Section 2.3.2 we adopt the NONLOCAL FEATURE PRINCIPLE of Pollard and Sag 1994.

(78) The Immediate Dominance Principle (IDP):

$$phrase \Rightarrow \big(\text{HC } \textbf{or} \text{ HA } \textbf{or} \text{ HM } \textbf{or} \text{ HF}\big)$$

Following the usual HPSG terminology, we call each disjunct in the consequent of the IDP an *ID Schema*. We assume a Head-Complement Schema, indicated as HC in (78), a Head-Adjunct Schema (HA), a Head-Marker Schema (HM), and a Head-Filler Schema (HF). Let us first consider the HA Schema. It is needed to analyze the phrase *good book* as it occurs in sentence (75a).

(79) The Head-Adjunct Schema:

$$\begin{bmatrix} \textit{phrase} & \\ \text{SYNS LOC CAT SUBCAT } \boxed{2} & \\ \text{DTRS} & \begin{bmatrix} \textit{head-adj-struc} & \\ \text{H-DTR} & \begin{bmatrix} \text{SYNS } \boxed{1} \begin{bmatrix} \text{LOC CAT SUBCAT} & \boxed{2} \\ \text{NONL TB SLASH} & \textit{eset} \end{bmatrix} \end{bmatrix} \\ \text{N-DTR} & \begin{bmatrix} \text{SYNS LOC CAT HEAD MOD } \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

The HA Schema specifies that the dtrs value of a phrase is an object of sort *head-adj-struc*. Furthermore, the mod value of the nonhead daughter ($\boxed{1}$) is required to be identical to the synsem value of the head daughter. In addition, the to-bind slash value of the head daughter is empty and the subcat value is the same on the mother and the head daughter. As mentioned above, the to-bind value of a sign is part of the analysis of extraction phenomena. We will see in Section 2.3.2 that an empty to-bind value at the head daughter of a phrase has the effect that nonlocal dependencies cannot be saturated at this phrase.

Using the lexical entries for the words *book* and *good*, we can give a description of the phrase *good book*, in which we make the effect of the HA Schema explicit.

(80) Description of the phrase *good book* (1):

$$\begin{bmatrix} \textit{phrase} \\ \text{SYNS LOC CAT SBC } \boxed{2} \end{bmatrix}$$

ADJ — HEAD

$$\begin{bmatrix} \textit{word} \\ \text{PHON } \langle \textit{good} \rangle \\ \text{SYNS LOC CAT HEAD MOD } \boxed{1} \end{bmatrix} \qquad \begin{bmatrix} \textit{word} & \\ \text{PHON } \langle \textit{book} \rangle & \\ \text{SYNS } \boxed{1} & \begin{bmatrix} \text{LOC CAT SBC } \boxed{2} \big\langle \boxed{3} \big\rangle \\ \text{NONL TB SLASH } \textit{eset} \end{bmatrix} \end{bmatrix}$$

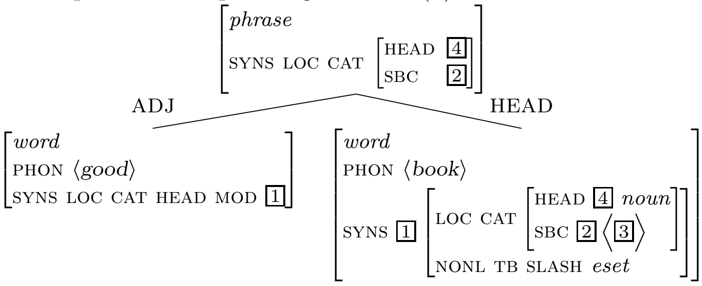
In the following, we describe the phrase in more detail, i.e., we extend the description so that it only describes objects that satisfy other principles on phrases as well. The most famous principle of HPSG in the Head Feature Principle. It requires that the head value be the same on a phrase and its head daughter. This principle is stated in (81).

(81) The Head Feature Principle (HFP):

$$phrase \Rightarrow \begin{bmatrix} \text{SYNS LOC CAT HEAD } \boxed{1} \\ \text{DTRS H-DTR SYNS LOC CAT HEAD } \boxed{1} \end{bmatrix}$$

The description of the phrase *good book* in (82) is only met by objects that satisfy the HA-Schema and the HFP.

(82) Description of the phrase *good book* (2):



The tag $\boxed{4}$ indicates that the HEAD values on the mother and on the head daughter are identical. Therefore, if a phrase is described by (82), it also satisfies the HFP.

We can introduce a further general principle on phrases: the Marking Principle. According to this principle, the MARKING value of a phrase is identical to that of its head daughter, except in a head-marker structure, in which case the MARKING value is identical to that of the non-head daughter. Consider (83):

(83) The Marking Principle (MP):

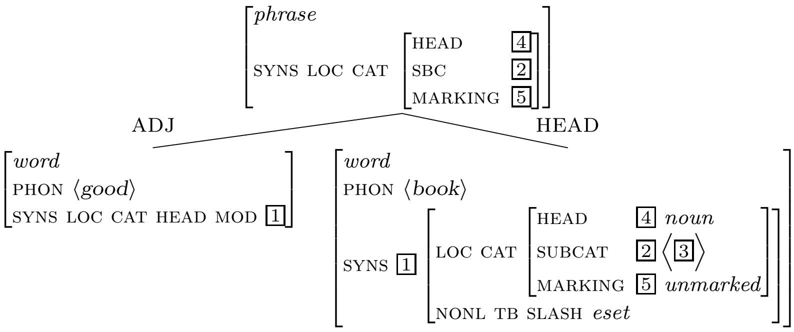
$$phrase \Rightarrow \begin{bmatrix} \text{SYNS LOC CAT MARKING } \boxed{1} \\ \text{DTRS} \begin{bmatrix} \textbf{not} \;\; hd\text{-}mark\text{-}struc \\ \text{H-DTR SYNS LOC CAT MARKING } \boxed{1} \end{bmatrix} \end{bmatrix}$$
$$\textbf{or}$$
$$\begin{bmatrix} \text{SYNS LOC CAT MARKING } \boxed{1} \\ \text{DTRS} \begin{bmatrix} hd\text{-}mark\text{-}struc \\ \text{N-DTR SYNS LOC CAT MARKING } \boxed{1} \end{bmatrix} \end{bmatrix}$$

In the first AVM in this consequent of this principle, we use the logical symbol "**not** " inside the description of the DTRS value. This use has not been introduced in the previous section. This AVM corresponds to the following explicit RSRL description:

(84) **not** :DTRS∼*hd-mark-struc*
and :SYNS LOC CAT MARKING ≈ :DTRS H-DTR SYNS LOC CAT MARKING

If we make the description in (82) even more specific to exclude phrases that do not meet the MP, a further tag, $\boxed{5}$, appears that expresses the identity of the MARKING values of the phrase and the head daughter.

(85) Description of the phrase *good book* (3):



As indicated in Figure 2.1 (page 85), the nominal projection *good book* combines with the determiner in a head-complement structure. To show how this is done, we must first give the HC Schema. This is done in (86).

(86) The HEAD-COMPLEMENT SCHEMA:

$$\begin{bmatrix} \textit{phrase} \\ \text{SYNS LOC CAT SUBCAT } \boxed{1} \\ \text{DTRS} \begin{bmatrix} \textit{head-comp-struc} \\ \text{H-DTR} \left[\text{SYNS} \left[\text{LOC CAT SUBCAT } \boxed{1} \oplus \left\langle \boxed{2} \right\rangle \right]\right] \\ \text{N-DTR} \left[\text{SYNS } \boxed{2}\right] \end{bmatrix} \end{bmatrix}$$

$$\textbf{and} \left(\left[\text{DTRS H-DTR } \textit{phrase}\right] \Rightarrow \left[\text{DTRS H-DTR SYNS NONL TB SLASH } \textit{eset}\right] \right)$$

According to the HC SCHEMA in (86), a phrase has a DTRS value of sort *head-comp-struc*. The SYNSEM value of the non-head daughter is the last element on the SUBCAT list of the head daughter. The SUBCAT value of the head daughter is the concatenation of the SUBCAT list of the phrase and the list that contains only the SYNSEM value of the nonhead daughter.

In addition, we have added an implication to the HC-Schema which relates the TO-BIND SLASH value of the head daughter to its status as a word or a phrase. If the head-daughter is a phrase, then its TO-BIND SLASH value is the empty set. If it is a word, then its TO-BIND SLASH value is not constrained by this schema. For words, the TO-BIND SLASH value is determined by the lexical entry. In the fragment that we are considering, it is always empty (see the lexical entries in (77)). In their analysis of *tough*-constructions, Pollard and Sag 1994 (Section 4.3) assume words which have a non-empty TO-BIND SLASH value.

In (86) we use the symbol "⊕" as an informal functional notation of a description that uses the relation `append`. To be formally precise, it is necessary to introduce another variable, $\boxed{3}$, for the SUBCAT value of the head daughter and impose the restriction that the relation append holds between the SUBCAT list of the phrase, a singleton list that contains just the SYNSEM value of the non-head daughter, and the SUBCAT list of the head daughter. We state the formally precise variant of the HC SCHEMA in (87).

(87) Explict formalization of the HC SCHEMA:

$$\begin{bmatrix} \textit{phrase} \\ \text{SYNS LOC CAT SUBCAT } \boxed{3} \\ \text{DTRS} \begin{bmatrix} \textit{head-comp-struc} \\ \text{H-DTR} \left[\text{SYNS} \begin{bmatrix} \text{LOC CAT SUBCAT} & \boxed{1} \\ \text{NONL TB SLASH} & \textit{eset} \end{bmatrix}\right] \\ \text{N-DTR} \left[\text{SYNS } \boxed{2}\right] \end{bmatrix} \end{bmatrix}$$

$$\textbf{and} \left(\left[\text{DTRS H-DTR } \textit{phrase}\right] \Rightarrow \left[\text{DTRS H-DTR SYNS NONL TB SLASH } \textit{eset}\right] \right)$$

$$\textbf{and E}\boxed{4} \left(\boxed{4}\begin{bmatrix} \textit{list} \\ \text{FIRST } \boxed{2} \\ \text{REST } \textit{elist} \end{bmatrix} \textbf{ and } \texttt{append}(\boxed{3},\boxed{4},\boxed{1}) \right)$$

In (88) we give a description of the phrase *every good book* as it occurs in sentence (75a). Every phrase that is described by (88) satisfies the HC SCHEMA, the HFP and the MP.

(88) Description of the phrase *every good book* (1):

$$\begin{bmatrix} \textit{phrase} \\ \text{SYNS LOC CAT} \begin{bmatrix} \text{HEAD} & \boxed{4} \\ \text{SBC} & \langle\rangle \\ \text{MARKING} & \boxed{5} \end{bmatrix} \end{bmatrix}$$

COMP:

$$\begin{bmatrix} \text{PHON } \langle \textit{every} \rangle \\ \text{SYNS } \boxed{3} \left[\text{LOC CAT HEAD SPEC } \textit{synsem}\right] \end{bmatrix}$$

HEAD:

$$\begin{bmatrix} \textit{phrase} \\ \text{SYNS} \begin{bmatrix} \text{LOC CAT} \begin{bmatrix} \text{HEAD } \boxed{4} \\ \text{SBC } \left\langle \boxed{3} \right\rangle \\ \text{MARKING } \boxed{5} \end{bmatrix} \\ \text{NONL TB SLASH } \textit{eset} \end{bmatrix} \end{bmatrix}$$

In the description in (88), the head daughter has a single element on its SUBCAT list. The HC SCHEMA has the effect that the SYNSEM value of the non-head is identical to this element. Finally, the SUBCAT list of the phrase is just like that of the head daughter minus the last element. In this case, this amounts to an empty SUBCAT list on the phrase.

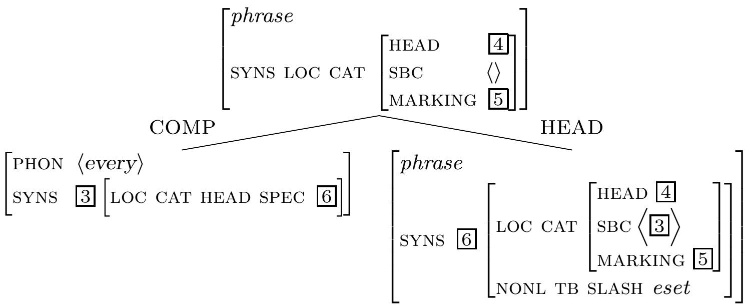
It was noted in connection with the lexical entry for *every* in (77c) that the determiner has a *synsem* object as its SPEC value. Pollard and Sag 1994 assume a SPEC PRINCIPLE which enforces that whenever the non-head daughter has a *synsem*-valued SPEC attribute defined on it, then the SYNSEM value of the head daughter is identical to the SPEC value of the non-head daughter. This principle is expressed formally in (89).[8]

(89) The SPEC PRINCIPLE (SpP):

$$\begin{bmatrix} \textit{phrase} \\ \text{DTRS} \begin{bmatrix}\text{N-DTR SYNS LOC CAT HEAD SPEC } \textit{synsem}\end{bmatrix}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{DTRS} \begin{bmatrix}\text{H-DTR SYNS } \boxed{1} \\ \text{N-DTR SYNS LOC CAT HEAD SPEC } \boxed{1}\end{bmatrix}\end{bmatrix}$$

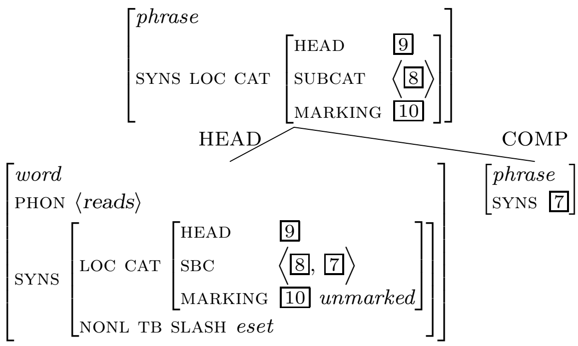
In (90) we add a tag $\boxed{6}$ to the tree in (88) to express the effect of the SpS, i.e., the identity between the SPEC value of the determiner and the SYNSEM value of the head daughter.

(90) Description of the phrase *every good book* (2):



Given the lexical entry for the verb *reads* in (77d), we can combine a word that is described by this lexical entry with a phrase that is described by (90) in a head-complement structure to form a VP. The resulting VP is described in (91).

(91) Description of the phrase *reads every good book*



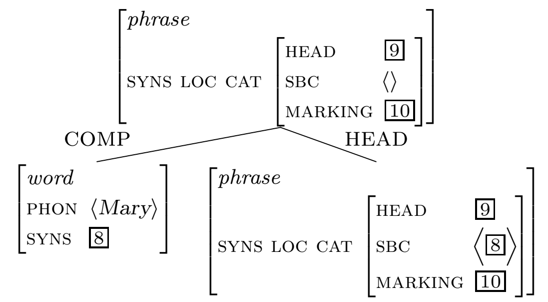
The HEAD values of the phrase and its head daughter ($\boxed{9}$) are identical by virtue of the HFP. The MP has the effect that the MARKING values ($\boxed{10}$) are identical as well. The HC SCHEMA guarantees that the SYNSEM value of the nonhead daughter ($\boxed{7}$) is identical to the

[8]The formulation of the SPEC PRINCIPLE in Pollard and Sag 1994 (p. 400) looks more complicated, because the authors assume distinct attribute names for different kinds of nonhead daughters, and a list of complement daughters.

last element on the SUBCAT list of the head daughter, and that the first element of this SUBCAT list (8) appears as the only element in the SUBCAT list of the phrase.

We can, next, add the subject *Mary*, which results in the phrase described in (92).

(92) Description of the phrase *Mary reads every good book*:



The phrase described in (92) is a saturated verb phrase. The identities expressed by the tags 9 and 10 follow from the HFP and the MP, respectively. The HC SCHEMA has the effect that the SYNSEM value of the non-head daughter (8) is identical to the single element on the 'shead daughter SUBCAT list, and that the SUBCAT list of the phrase is empty.[9]

We have almost finished the analysis of sentence (75a). What remains to be shown is the analysis of the highest local tree in Figure 2.1, i.e., how the complementizer *that* combines with the saturated verbal projection described in (92). As indicated in Figure 2.1, the complementizer is realized as a marker. Markers combine with heads according to the HEAD-MARKER SCHEMA.

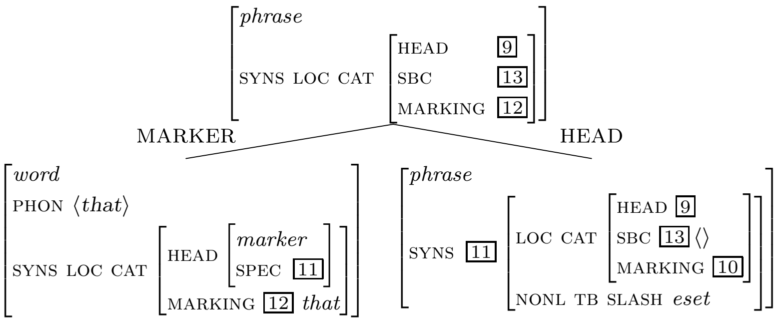
(93) The HEAD-MARKER SCHEMA:

$$\begin{bmatrix} \textit{phrase} & \\ \text{SYNS LOC CAT SUBCAT } \boxed{1} & \\ \text{DTRS} & \begin{bmatrix} \textit{head-marker-struc} & \\ \text{H-DTR} & \begin{bmatrix} \text{SYNS} & \begin{bmatrix} \text{LOC CAT SUBCAT} & \boxed{1} \\ \text{NONL TB SLASH} & \textit{eset} \end{bmatrix} \end{bmatrix} \\ \text{N-DTR} & \begin{bmatrix} \text{SYNS LOC CAT HEAD} & \textit{marker} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

The HM SCHEMA requires that the DTRS value of the phrase be an object of sort *head-marker-struc*. Furthermore, the SUBCAT values of the phrase and its head daughter are identical. The TO-BIND SLASH value of the head daughter is empty and the non-head daughter is a marker. The way the HM SCHEMA is stated, it seems as if there is no selection happening in head-marker structures. This is, however, not the case: the MP and the SpP have a non-trivial effect in head-marker structures. This is indicated in (94), which is a description of the highest local tree in the structure of sentence (75a) as sketched in Figure 2.1.

[9]The way we have stated the HC SCHEMA, the SUBCAT list is always reduced by its last element in a head-complement structure. If we assume that the phonology of a phrase is the concatenation of the phonologies of its daughters, as we did in the toy grammar of Sections 1.1 and 2.1, then this has consequences on the order of the elements on the SUBCAT list. As we are only concerned with transitive and intransitive verbs in this thesis, this issue does not arise.

(94) Description of the phrase *that Mary reads every good book*:



In the description in (94) the tag $\boxed{9}$ expresses the effect of the HFP, just as it did in (92). The identity of the SUBCAT values ($\boxed{13}$) on the phrase and its head daughter is enforced by the HM SCHEMA. As the DTRS value is of sort *head-marker-struc*, the MP in (83) requires that the MARKING value of the phrase be identical with that of the non-head daughter ($\boxed{12}$). Furthermore, as the non-head daughter has a *synsem* object in its SPEC value, the SpP (given in (89)) ensures that the SYNSEM value of the head daughter is identical with the SPEC value of the non-head daughter ($\boxed{11}$).

In this subsection, we have provided a full analysis of sentence (75a). To achieve this, it has been necessary to present adopted versions of the major principles of the grammar of Pollard and Sag 1994. It should be emphasized, again, that our syntactic analysis is not intended as a "serious" proposal; rather it provides the necessary syntax for showing how the semantic framework of LF-Ty2 can be integrated into an HPSG grammar. The use of strictly binary structures for phrases allows us to define the semantics of a phrase as simple application of the semantics of one daughter to that of the other daughter, as proposed in Section 1.3.3. Given this strictly binary branching structure, we have tried to give an analysis that follows the grammar of Pollard and Sag 1994 as closely as possible.

2.3.2. The Analysis of Passive and Complement Extraction. In the preceding subsection, we have presented the general architecture of our syntactic analysis. In this subsection, we will address two more specific phenomena which we will need in our account of idiomatic expressions in the second part of this thesis. The relevant phenomena are *passive* and *complement extraction*. We maintain the same reservations here as we did in the previous subsection; the analysis should allow for a simple formulation of the semantics, therefore syntactic motivation is only of secondary importance.

Our analysis of both phenomena relies on the availability of a mechanism to encode what has often been called *Lexical Rules* (LR) in HPSG. It is discussed in great detail in Meurers 2000 that there are two basic concepts of LRs: under the traditional use of the term, an LR is a relation between two *lexical entries*, i.e., between two descriptions. Alternatively, one could view an "LR" as a relation between two *words*, i.e., between two objects. In this thesis, we will use a mechanism that formalizes the second view. To avoid terminological confusion, we will not refer to such rules as Lexical Rules but as *Derivational Rules* (DR). In this thesis, we will exclusively use DRs.

As a simple example, in (95) we present the informal version of the DR for passivization.

(95) The Passive DR:

$$\begin{bmatrix} \textit{word} \\ \text{SYNS LOC} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD} \begin{bmatrix} \textit{verb} \\ \text{VFORM} \ \textit{psp} \end{bmatrix} \\ \text{SBC} \ \left\langle \text{NP}|\boxed{2} \right\rangle \oplus \left\langle \boxed{1} \right\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \longmapsto \begin{bmatrix} \textit{word} \\ \text{SYNS LOC} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD} \begin{bmatrix} \textit{verb} \\ \text{VFORM} \ \textit{pas} \end{bmatrix} \\ \text{SBC} \ \left\langle \boxed{1}|\boxed{2} \right\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

In its informal notation, a DR consists of two descriptions separated by the symbol "$\longmapsto$". The description on the left side of the "$\longmapsto$" symbol is called the *input specification*, the other description the *output specification*. A DR relates a word that meets the input specification to a word that meets the output specification, while it is assumed that the DR specification as given in (95) is automatically expanded so that the output word has all the properties of the input word except for those explicitly mentioned explicitly in the DR. Meurers 2000 (Section 5.3) defines such an expansion mechanism.[10]

The DR in (95) expresses the idea that there is a systematic relation between a transitive verb (as licensed by the lexical entry in (77d)) and the passivized form of this word. In (96) we describe a potential input word and a potential output word of this DR.

(96) a. Description of the active verb *loved*:

$$\begin{bmatrix} \textit{word} \\ \text{PHON} \ \boxed{1} \ \langle \textit{loved} \rangle \\ \text{SYNS} \begin{bmatrix} \text{LOC CAT} \begin{bmatrix} \text{HEAD} \begin{bmatrix} \textit{verb} \\ \text{VFORM} \ \textit{psp} \end{bmatrix} \\ \text{SUBCAT} \ \left\langle \text{NP}, \boxed{2} \ \text{NP} \right\rangle \\ \text{MARKING} \ \boxed{3} \ \textit{unmarked} \end{bmatrix} \\ \text{NONLOCAL} \ \boxed{4} \begin{bmatrix} \text{INH SLASH} \ \textit{eset} \\ \text{TB SLASH} \ \textit{eset} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

b. Description of the passive verb *loved*:

$$\begin{bmatrix} \textit{word} \\ \text{PHON} \ \boxed{1} \ \langle \textit{loved} \rangle \\ \text{SYNS} \begin{bmatrix} \text{LOC CAT} \begin{bmatrix} \text{HEAD} \begin{bmatrix} \textit{verb} \\ \text{VFORM} \ \textit{pas} \end{bmatrix} \\ \text{SUBCAT} \ \left\langle \boxed{2} \ \text{NP} \right\rangle \\ \text{MARKING} \ \boxed{3} \ \textit{unmarked} \end{bmatrix} \\ \text{NONLOCAL} \ \boxed{4} \begin{bmatrix} \text{INH SLASH} \ \textit{eset} \\ \text{TB SLASH} \ \textit{eset} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

We use tags in (96) to indicate which parts of the input word must be identical with which parts of the output word. Given this input/output pair, we can illustrate the intended effect of the Passive DR. The output word is just like the input word, with a small number of differences: First the output word is passive, i.e., it has a VFORM value of sort *pas(sive)*, whereas the input is a past participle (*psp*). Second, on the SUBCAT list of the output word, the first element of the input's SUBCAT list has been removed, and the last element of the input's SUBCAT list appears as the first element on the output's SUBCAT list.[11]

We follow the formalization of "Lexical Rules" proposed in Meurers 2000, but we use the term *derivational rule* to refer to what Meurers calls *description level lexical rules* (DL-LR). We introduce a sort *der(ivational)-rule* whose appropriateness conditions are stated in (97).

---

[10] By the existence of such a mechanism it follows that it is enough to state a DR in the form given in (95). Since the notation leaves many details of the DR implicit and since we do not have the space to present the expansion mechanism of Meurers 2000, we call the notation in of a DR as in (95) *implicit* or *informal*, in contrast to the *explicit* notation as given in (98).

[11] Note that we stated the Passive DR so that the SYNSEM object that appears as the first element on the SUBCAT list of the passivized verb is identical with the last element of the active verb. Because of this identity, the CASE value of the subject of the passivized verb is identical to the CASE value of the direct object of the input word of the DR. While this is unintuitive at first sight, it is compatible with theories of structural case as developed for example in Przepiórkowski 1999a. Alternatively, we can refine the Passive DR so that the values of all attributes except for CASE are identical on the two NPs. As we are not concerned with case in this thesis, we will ignore this issue in what follows.

(97) Appropriateness conditions on the sort *derivational-rule*:
der-rule IN *word*
OUT *word*

According to (97), a *derivational-rule* object has two *word*-valued attributes: an attribute IN which contains the input word of the DR, and an attribute OUT which contains the output word. For the DR sketched in (95), a *der-rule* object can be described the following way:

(98) Explicit description of the Passive DR:

$$\begin{bmatrix} \textit{der-rule} & \\ \text{IN} & \begin{bmatrix} \textit{word} & \\ \text{PHON} & \boxed{3} \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{psp} \\ \text{INV} & \boxed{5} \\ \text{AUX} & \boxed{6} \end{bmatrix} \\ \text{SUBCAT} & \left\langle \text{NP}|\boxed{2} \right\rangle \oplus \left\langle \boxed{1} \right\rangle \\ \text{MARKING} & \boxed{7} \end{bmatrix} \end{bmatrix} \\ \text{NONL} & \boxed{4} \end{bmatrix} \end{bmatrix} \\ \text{OUT} & \begin{bmatrix} \textit{word} & \\ \text{PHON} & \boxed{3} \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{pas} \\ \text{INV} & \boxed{5} \\ \text{AUX} & \boxed{6} \end{bmatrix} \\ \text{SUBCAT} & \left\langle \boxed{1}|\boxed{2} \right\rangle \\ \text{MARKING} & \boxed{7} \end{bmatrix} \end{bmatrix} \\ \text{NONL} & \boxed{4} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

The *der-rule* object described in (98) expresses that the word in its IN value, i.e., the input word meets the requirements expressed by the input specification in (95), and that, similarly, the word in the OUT value meets the output specification of the informal DR specification given above. Stated as in (98), the tags used in (95) can be interpreted in the regular way, i.e., as variables. To consider our example input and output words in (96), a *der-rule* object which has the active verb *loved* as its IN value and the passive verb *loved* as its OUT value meets the description in (98).

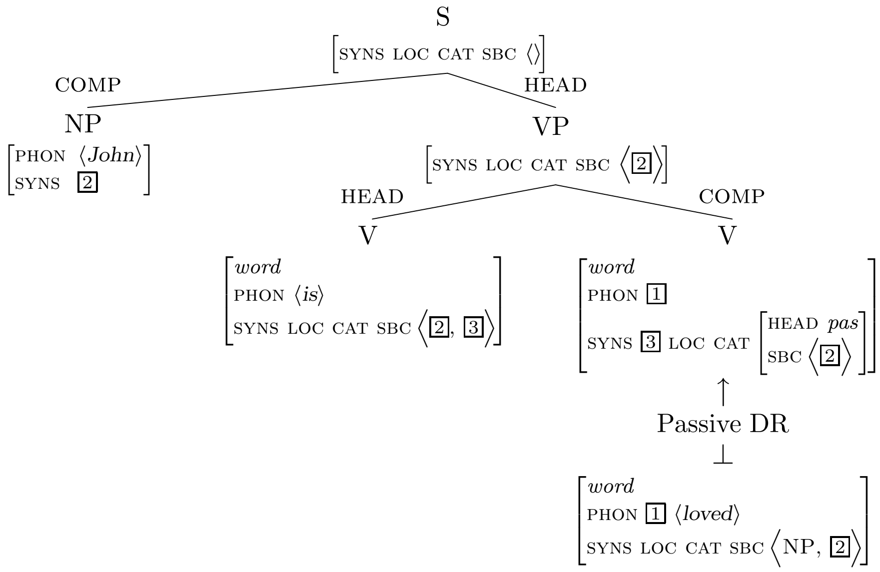
In order to constrain the possible *der-rule* objects in the denotation of a grammar, we must add a principle to the theory.

(99) The DR PRINCIPLE:
der-rule $\Rightarrow \big(\text{DR}_1$ **or** $\ldots$ **or** $\text{DR}_n\big)$

Just as the WORD PRINCIPLE and the ID PRINCIPLE, the DR PRINCIPLE is an implication with a disjunction as its consequent. We call each disjunct in the consequent of the DR PRINCIPLE a *derivational rule* (DR). In this section, we will only consider two DRs, the Passive DR, which we have already given in (98) and the Complement Extraction DR, which will be presented in (102).

Now that we have assigned DRs a formally clear status, we can show how Meurers 2000 establishes the connection between the output of a derivational rule and an actual word. For this purpose, the attribute STORE declared appropriate on the sort *word* in (70) plays a crucial role. So far, we have only considered words with an empty STORE value. In case the STORE value is not empty, it contains exactly one object of sort *derivational-rule*. We add a principle in (100) to the grammar that requires identity between a word with a non-empty STORE value and the OUT value of the *der-rule* object in the STORE list.

FIGURE 2.4. The structure of sentence (75b):



(100) The STORE PRINCIPLE:

$$\begin{bmatrix} \textit{word} \\ \text{STORE} \ \textit{nelist} \end{bmatrix} \Rightarrow \boxed{1} \begin{bmatrix} \text{STORE} \left\langle \begin{bmatrix} \textit{der-rule} \\ \text{OUT} \ \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix}$$

It is a crucial property of the DR approach to lexical generalizations that the output word contains the input word as a component, i.e., we can reach the input word via the path : STORE FIRST IN. In that respect, we can also think of a derived word as being a word which dominates its input word in a unary branching syntactic structure. Such structures differ, however, from the syntactic structures dominated by phrases, as we have met them in Section 2.3.1: there, the mother is a phrase and there are always two daughters. In the case of a derived word, there is a single daughter and the mother and the daughter are both words. Still, when we sketch the structure of a sentence with a tree diagram, we will include the application of a DR. We will do this, however, using a special symbol, "↥".

In Figure 2.4 this notational convention is illustrated with a description of the passive sentence in (75b). The active verb *loved* as described in (96a) is directly licensed by some lexical entry in the WORD PRINCIPLE. It occurs as the IN value in some *der-rule* object, which, in turn, appears as the STORE element of some derived word. The OUT value of the *der-rule* object meets the description in (96b). By the STORE PRINCIPLE in (100), the derived word is identical to its STORE FIRST OUT value. This connection is abbreviated with the "↥" symbol in Figure 2.4.

Following Pollard and Sag 1994, the passive auxiliary *is* is treated as a subject raising verb. As such, the first element on the SUBCAT list of the auxiliary is identical to the first element on the SUBCAT list of the passivized verb. This verb occurs as the syntactic complement of the passive auxiliary. In the highest local tree in Figure 2.4, the NP *John* combines in a head-complement structure with the VP to yield a saturated verbal projection.

After the consideration of simple passive sentences, we can turn to complement extraction. In (75c) we have given an example with a topicalized NP which is repeated in (101).

(101) John, Mary loved.

Within HPSG there is much debate on how sentences as (101) should be analyzed. There is, however, also some agreement on certain aspects of the construction. First, the pre-posed NP, *John* in our example, is called the *filler* and is introduced by a special ID schema. Second, the LOCAL value of the filler is assumed to be identical with the LOCAL value of the second element on the SUBCAT list of the verb *loved*. Third, all phrases that dominate the verb *loved* but not the filler are assumed to have a non-empty INHERITED SLASH value which contains the LOCAL value of the filler.[12]

There is, however, no agreement on the question whether there should be an element with empty phonology, i.e, a trace, appearing as the complement of the verb *loved* or not. This controversy is already visible in the differences between the analyses of complement extraction proposed in Chapter 4 and in Section 9.5 of Pollard and Sag 1994, where the former assumes the presence of a trace, but the latter does not.

As has already been seen in Figure 2.3 on page 86, we assume an analysis of sentence (75c) (= (101)) without a trace. Within HPSG, the argumentation of Sag and Fodor 1994 had great influence in promoting traceless analyses of extraction. As will be shown in more detail in Section 4.3, we assume a traceless analysis, because it enables us to express the SEMANTICS PRINCIPLE in a more uniform way and to maintain the assumption that there are *local* objects in the SLASH values.[13] Thus, again, this move is motivated by our attempt to provide a simple account of combinatorial semantics and to follow the analysis of Pollard and Sag 1994 as closely as possible.

Our analysis of sentence (101) builds on three ingredients: First, we assume a DR that removes an element from the SUBCAT list and "puts" it into the INHERITED SLASH set. Second, we need a principle that ensures that the SLASH specification is mediated between an extraction site and the filler. And third, a new ID schema, the HEAD-FILLER SCHEMA will be given that allows us to introduce fillers into the syntactic structure.

In (102) we state the Complement Extraction DR (CEx-DR) in the form assumed in this thesis. In (a) we give the informal specification as is common in HPSG literature, in (b) the explicit description of the DR, i.e., the DR is stated as a disjunct in the DR PRINCIPLE. Just as was the case with the informal and the formal versions of the Passive DR in (95) and (98), the informal and the formal version differ in two respects: First, the descriptions on both sides of the "$\longmapsto$" symbol of the informal specification are re-stated as descriptions of the IN and the OUT values of a *der-rule* object. Second, in the description in (102b) we explicitly state which properties of the input word are the same in the output.

(102) The Complement Extraction Derivational Rule (CEx-DR):

a. informal specification:

$$\begin{bmatrix} \textit{word} \\ \text{SYNS} \begin{bmatrix} \text{LOC CAT SBC } \boxed{1}\begin{bmatrix}\textit{nelist}\end{bmatrix} \oplus \left\langle \begin{bmatrix}\text{LOC } \boxed{2}\end{bmatrix} | \boxed{3} \right\rangle \\ \text{NONL INH SLASH } \boxed{4} \end{bmatrix} \end{bmatrix} \longmapsto \begin{bmatrix} \textit{word} \\ \text{SYNS} \begin{bmatrix} \text{LOC CAT SBC} & \boxed{1} \oplus \boxed{3} \\ \text{NONL INH SLASH} & \boxed{4} \cup \left\{\boxed{2}\right\} \end{bmatrix}\end{bmatrix}$$

[12]Of course, not all authors agree on all of these aspects. To pick out a very radical proposal, Richter 1997 does not assume a SLASH attribute altogether, nor is there a special ID schema for introducing fillers.

[13]The problem that we are facing is basically the following: If we assume a trace whose LOCAL value is identical to that of its filler, then we will automatically have two signs within the same tree that have identical CONTENT values. In the grammar of Pollard and Sag 1994 this does not cause any problems, because there the semantic contribution of complement daughters and that of filler daughters is largely ignored for combinatorial semantics. Within other linguistic frameworks such as GPSG (Gazdar et al., 1985), or the generative tradition (von Stechow, 1993; Heim and Kratzer, 1997), the logical form of a trace is a variable and, therefore, not identical to that of the filler.

b. explicit specification:

$$\begin{bmatrix} \textit{der-rule} & \\ \text{IN} & \begin{bmatrix} \textit{word} & \\ \text{PHON} & \boxed{8} \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \boxed{6} \\ \text{SUBCAT} & \boxed{1}\big[\textit{nelist}\big] \oplus \Big\langle \big[\text{LOC } \boxed{2}\big] | \boxed{3} \Big\rangle \\ \text{MARKING} & \boxed{7} \end{bmatrix} \end{bmatrix} \\ \text{NONL} & \begin{bmatrix} \text{INH SLASH} & \boxed{4} \\ \text{TB SLASH} & \boxed{5} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{OUT} & \begin{bmatrix} \textit{word} & \\ \text{PHON} & \boxed{8} \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \boxed{6} \\ \text{SUBCAT} & \boxed{1} \oplus \boxed{3} \\ \text{MARKING} & \boxed{7} \end{bmatrix} \end{bmatrix} \\ \text{NONL} & \begin{bmatrix} \text{INH SLASH} & \boxed{4} \cup \Big\{\boxed{2}\Big\} \\ \text{TB SLASH} & \boxed{5} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

It is the advantage of the informal notation that it highlights those parts of the input and the output word of a DR which differ. In the case of the CEx-DR, the difference is found in the SUBCAT and the INHERITED SLASH values. The SUBCAT list of the output word lacks one element, i.e., its SUBCAT list is the concatenation of the two lists $\boxed{1}$ and $\boxed{3}$, whereas the input word has a SUBCAT list which has one more element, described as [LOCAL $\boxed{2}$] in the input specification of the DR. While the SUBCAT list of the output is shorter than that of the input, the INHERITED SLASH set of the output has (potentially) more elements: The INHERITED SLASH value contains the elements of the INHERITED SLASH value of the input unioned with the singleton set which contains just $\boxed{2}$, i.e., the LOCAL value of the extra element on the SUBCAT list of the input.[14]

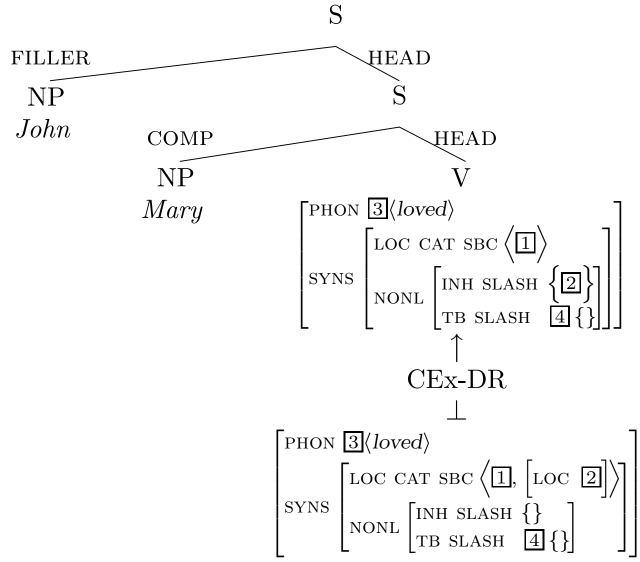
Let us turn back to the sentence in (101). In its analysis, the CEx-DR is applied to the verb *loved*. In (77d) the lexical entry of the transitive verb *reads* was given. We can assume that there is a similar lexical entry for the verb *love*. A word that is described by such a lexical entry is also described by the input specification of the CEx-DR. To derive the form of the verb needed in sentence (101), we must extract the second element of the SUBCAT list of this word. In (103a) the input word *loved* and the output of the CEx-DR are described. To illustrate the effect of the DR, we use the same tags as in (102b).

(103) a. Description of the input word *loved*:

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \boxed{8}\ \langle \textit{loved} \rangle \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \boxed{6} \begin{bmatrix} \textit{verb} \\ \text{VFORM } \textit{psp} \end{bmatrix} \\ \text{SUBCAT} & \boxed{1} \Big\langle \text{NP} \Big\rangle \oplus \Big\langle \big[\text{LOC } \boxed{2}\big] | \boxed{3}\ \langle\rangle \Big\rangle \\ \text{MARKING} & \boxed{7}\ \textit{unmarked} \end{bmatrix} \end{bmatrix} \\ \text{NONL} & \begin{bmatrix} \text{INH SLASH} & \boxed{4}\ \{\} \\ \text{TB SLASH} & \boxed{5}\ \{\} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

[14]We put "potentially" here, because the same *local* object might already be in the set $\boxed{4}$. In this case, of course, the union of $\boxed{4}$ with the set which contains just $\boxed{2}$ has exactly the same elements as the set $\boxed{4}$.

FIGURE 2.5. The structure of sentence (101) (including the CEx-DR):



b. Description of the output word *loved* with extracted direct object:

$$\begin{bmatrix} \textit{word} \\ \text{PHON } \boxed{8} \\ \text{SYNS} \begin{bmatrix} \text{LOC} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD} & \boxed{6}\begin{bmatrix}\textit{verb}\end{bmatrix} \\ \text{SUBCAT} & \boxed{1}\left\langle \text{NP}\right\rangle \oplus \boxed{3}\langle\rangle \\ \text{MARKING} & \boxed{7}\ \textit{unmarked}\end{bmatrix}\end{bmatrix} \\ \text{NONL} \begin{bmatrix} \text{INH SLASH} & \boxed{4}\{\} \cup \left\{\boxed{2}\right\} \\ \text{TB SLASH} & \boxed{5}\{\}\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

The output verb in (103b) has a single element, the subject, on its SUBCAT list. It also has a single element in its INHERITED SLASH set: the *local* object referred to by the tag $\boxed{2}$. In Figure 2.5 we have made the application of the CEx-DR and the special properties of the output word explicit. The rest of the figure is as in Figure 2.3 (page 86).

The second ingredient of the analysis of sentence (101) is the mechanism of passing "up" slash specifications. We have seen that the output verb *loved* in (103b) has a non-empty INHERITED SLASH value. In the analysis of extraction in Pollard and Sag 1994, all phrases that dominate this word but which do not dominate the filler are assumed to have the extracted element in their INHERITED SLASH value as well. For our example structure in Figure 2.5, this means that in addition to the V node, the *local* object referred to with the tag $\boxed{2}$ also appears in the slash value of the lower S node.

We follow Pollard and Sag 1994 in assuming a NONLOCAL FEATURE PRINCIPLE (NFP) which enforces the right identities of slash values. The NFP uses a new relation, `element`, which we have not defined so far. We assume that two objects o and s stand in the relation `element` iff either (i) o is the ELEMENT value of s or (ii) o stands in the relation `element` with the REST value of s. This is defined formally by the following clauses.

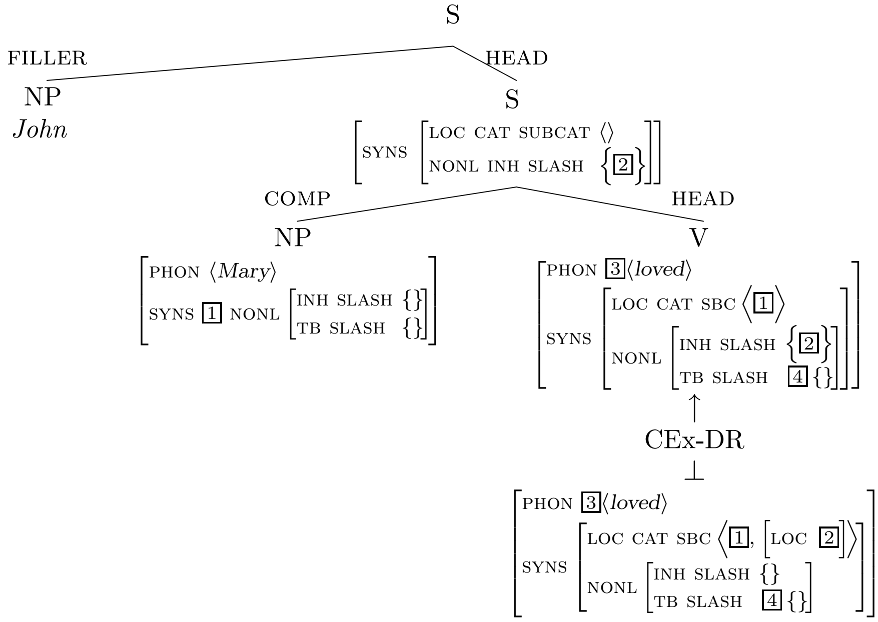
(104) The relation `element`:

$$\texttt{element}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow} \boxed{2}\begin{bmatrix}\text{ELEMENT } \boxed{1}\end{bmatrix}$$

$$\texttt{element}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow} \left(\boxed{2}\begin{bmatrix}\text{REST } \boxed{3}\end{bmatrix} \textbf{ and } \texttt{element}(\boxed{1},\boxed{3})\right)$$

Given the definition of the relation `element`, we can state the NFP in (105).

FIGURE 2.6. The structure of (101) (including the NONLOCAL specification of the lower S node):



(105) The NONLOCAL FEATURE PRINCIPLE (NFP):

phrase ⇒

([SYNS NONL INH SLASH [1], DTRS [H-DTR [SYNS NONL [INH SLASH [2], TB SHASH [3]]], N-DTR [SYNS NONL INH SLASH [4]]]]

and A[5] (((`element`([5],[2]) **or** `element`([5],[4])) **and not** `element`([5],[3])) ⇔`element`([5],[1])))

The NFP says that for each phrase the following applies: for each [5]: [5] is an element of the INHERITED SLASH value of the head daughter ([2]) or of the non-head daughter ([4]), but not of the TO-BIND SLASH set of the head daughter ([3]) if and only if [5] is also an element of the INHERITED SLASH value of the phrase ([1]). As an effect of the NFP, the INHERITED SLASH value of a phrase contains exactly those elements that appear in the INHERITED SLASH values of its daughters, except those that appear in the TO-BIND SLASH value of the head daughter.

Let us apply the NFP to our example sentence in (101). In the lexical entry for the name *Mary* in (77e) we have specified that the INHERITED SLASH value of the noun is empty. In the output of the CEx-DR, as given in (103b) and indicated in Figure 2.5, the INHERITED SLASH value contains a single element, and the TO-BIND SLASH value is empty. As in Figure 2.5, the verb *loved* and the NP *Mary* combine as a head-complement structure. The resulting phrase, the lower S node in the figure, must respect the NFP. As the TO-BIND SLASH value of the head daughter is empty, the NFP requires the element in the INHERITED SLASH of the head daughter to appear also as element in the INHERITED SLASH value of the phrase. In the structure in Figure 2.6 we have added the SLASH specification of the S node of our example sentence.

The last ingredient of our analysis of complement extraction is a new ID schema, the HEAD-FILLER SCHEMA, given in (106).

(106) The Head-Filler Schema:

$$\begin{bmatrix} \textit{phrase} \\ \text{SYNS LOC CAT SUBCAT } \textit{elist} \\ \text{DTRS} \begin{bmatrix} \textit{head-filler-struc} \\ \text{H-DTR} \begin{bmatrix} \text{SYNS} \begin{bmatrix} \text{LOC CAT} \begin{bmatrix} \text{HEAD} \begin{bmatrix} \textit{verb} \\ \text{VFORM } \textit{finite} \end{bmatrix} \\ \text{SBC } \textit{elist} \end{bmatrix} \\ \text{NONL} \begin{bmatrix} \text{INH SLASH } \boxed{1} \\ \text{TB SLASH } \left\{ \boxed{2} \right\} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{N-DTR} \begin{bmatrix} \text{SYNS LOC } \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

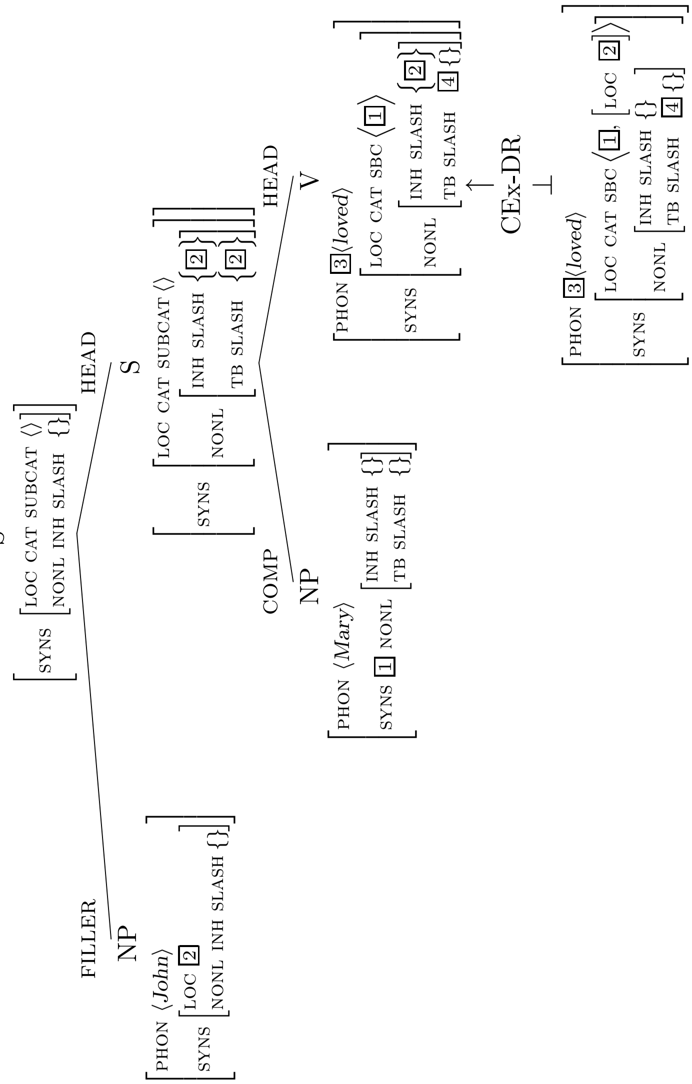
and `element`($\boxed{2}$,$\boxed{1}$)

The HF Schema is taken almost directly from Pollard and Sag 1994. It is the only schema that explicitly enforces a non-empty TO-BIND SLASH value at the head daughter. It requires a single element ($\boxed{2}$) in the TO-BIND SLASH value of the head daughter, which must also be an element in the INHERITED SLASH set of the head daughter ($\boxed{1}$) and which must be identical to the SYNSEM LOCAL value of the non-head daughter.

In Figure 2.7 (page 103) we indicate the effect of the HF Schema and of the NFP at the upper S node. The HF Schema ensures that there is an element in the TO-BIND SLASH set of the head daughter ($\boxed{2}$). This element also occurs in the INHERITED SLASH value of the head daughter and is identical with the SYNSEM LOCAL value of the filler daughter. As the *local* object referred to with the tag $\boxed{2}$ in Figure 2.7 appears in the TO-BIND SLASH value of the head daughter, the NFP prevents it from appearing in the INHERITED SLASH value of the mother.

The tree in Figure 2.7 can be seen as a summary of our analysis of complement extraction as it contains the effect of the CEx-DR, of the NFP, and of the HF Schema. With this tree, our presentation of the syntactic fragment assumed in this thesis is complete. The purpose of this presentation was twofold. First, we presented the syntactic fragment that will be used in later chapters of this thesis. In particular, we will augment this fragment with a semantic analysis in Chapter 4 and with an analysis of idiomatic expressions in Chapter 8. Second, we have chosen a syntactic analysis which is relatively close to the analysis proposed in Pollard and Sag 1994 for which we can assume some familiarity within the HPSG community. What was new, however, was the RSRL formalization of parts of this analysis. As such, we hope to have given some examples of how an RSRL grammar is written, before we turn to the RSRL grammar for the semantic representation language Ty2 of Section 1.3.2.

FIGURE 2.7. The structure of sentence (101) (including the effect of the HF SCHEMA and the NFP at the upper S node).



CHAPTER 3

Ty2 in RSRL

In the preceding chapter we have given the formal definitions of the description language RSRL, and specified a syntactic fragment that will be used throughout the rest of this thesis. Regarding AVM syntax, RSRL descriptions emulate the common notation used in the HPSG literature. In this chapter, we will ignore the syntactic fragment defined in Section 2.3, and define a new RSRL grammar, $\mathcal{TY}2$. Whereas the purpose of the grammar of Section 2.3 was to encode a syntactic fragment of English, the purpose of the grammar $\mathcal{TY}2$ will be to encode the semantic representation language Ty2 as defined in Section 1.3.2 and prove the adequacy of the given encoding. In addition, just as there is an AVM notation for RSRL (given in Richter 1999, 2000), we will show that we can give a more standard notation for descriptions that denote terms of the semantic representation language.

This chapter consists of five sections. In Section 3.1 we will define the RSRL grammar $\mathcal{TY}2$. In the process of defining this grammar, we will give some examples to illustrate how the structure of Ty2 is re-built with the technical means of RSRL. We will highlight those aspects that are essential to prove that the grammar indeed describes language Ty2.

Section 3.2 contains this proof: we will show that an interpretation is possible, which uses the language Ty2 as its universe and which is an exhaustive model of the grammar $\mathcal{TY}2$. In the main text, we only sketch this proof, which is given in Appendix 1.1. We know from this proof that the new grammar accounts for its "empirical" domain adequately, i.e., it neither underlicenses nor overlicenses.

Our ultimate goal is integrate the grammar $\mathcal{TY}2$ into a grammar of a fragment of English, such as the fragment of Section 2.3. In this case, we cannot guarantee that the terms of Ty2 are part of the intended exhaustive model. In Section 3.3 we show that even if we consider an arbitrary exhaustive model of $\mathcal{TY}2$, instead of the exhaustive model of Section 3.2, we can still regard the objects in the universe of this model as terms of Ty2. In particular, we can assign these objects a denotation with respect to a semantic frame. Thus, we show that, independent of the choice of the exhaustive model of $\mathcal{TY}2$, we can treat the objects in the universe as if they were elements of the intended model.

In Section 3.4 we show that not only can we treat objects in the denotation of the grammar $\mathcal{TY}2$ as terms of Ty2, we can also use terms of Ty2 as abbreviations of $\mathcal{TY}2$-descriptions. From here, we will be able to use terms of Ty2 as descriptions inside AVMs, consequently leading to a compact notation for linguistic objects that represent these terms.

Finally, in Section 3.5 we give a short summary of the results and show how we can integrate the extensions to the semantic representation language Ty2 that have been presented in Section 1.3.2. There, we showed that the standard logical connectives (negation, conjunction, disjunction, implication), the classical quantifiers (existential and universal) and generalized quantifiers such as *most* can be treated as syntactic sugar. For the grammar $\mathcal{TY}2$, however, we propose to integrate them as separate elements of the syntax. At the end of this chapter, we give an example of how the integration of the grammar $\mathcal{TY}2$ into some larger grammar such as the syntactic fragment of Section 2.3 can be accomplished.

3.1. The RSRL grammar $\mathcal{TY}2$

In this section, we give an RSRL grammar, called $\mathcal{TY}2$. It is the purpose of this grammar to make the terms of Ty2 available for an HPSG grammar. Just as the natural language English is the target of an HPSG grammar as given in Pollard and Sag 1994, in this section, our target is the language Ty2. In the following sections of this chapter, we will show that we have reached this target.

In this section, we will state the grammar $\mathcal{TY}2$ in the same way we have stated the grammar in Section 2.3. There, we have used some example sentences from English to illustrate how the grammar works. In this section, as our "empirical domain" is the language Ty2, we can use a term of Ty2 for illustration. Such a term is given in (107). In (a) we use the short hand that we have been employing throughout Section 1.3, in (b) we present the explicit form of the term as provided in Definition 1.5 (page 38).

(107) a. $\lambda x.\mathsf{walk}'_{@}(x_{@})$
b. $(\lambda v_{se,0}.((\mathsf{walk}'_{s(et)} v_{s,0})_{et}(v_{se,0} v_{s,0})_e)_t)_{(se)t}$

As can be seen, the term in (107) consists of (i) a natural number (0), (ii) semantic types (t, e, s, et, $s(et)$, $(se)t$), and finally, (iii) terms, built from variables and constants combined via functional application and lambda abstraction. As none of these entities is part of the RSRL grammars that we have presented so far, we will address these three kinds of entities (natural numbers, semantic types, and terms) in following subsection.

3.1.1. Natural Numbers. In the definition of Ty2, we used natural numbers to count the variables. Therefore, we must provide an equivalent to natural numbers in $\mathcal{TY}2$.

For this purpose, we define a sort *number* with two subsorts, *zero* and *non-zero*. There is an attribute NUMBER defined on the sort *non-zero* whose value is of sort *number*. The sort hierarchy and the appropriateness conditions for the sort *number* are given in (108).

(108) Sort hierarchy and appropriateness conditions for natural numbers:

number

zero *non-zero*
NUMBER *number*

number objects can be seen as strips. In our example term in (107b), there occurred just one natural number: 0. Using the sort hierarchy and appropriateness conditions as given in (108), we can describe this number as indicated in (109a). For illustration, we also describe the number 3 in (109b).

(109) a. RSRL description of the number 0:

$$\begin{bmatrix} \textit{zero} \end{bmatrix}$$

b. RSRL description of the number 3:

$$\begin{bmatrix} \textit{non-zero} & \\ \textsc{number} & \begin{bmatrix} \textit{non-zero} & \\ \textsc{number} & \begin{bmatrix} \textit{non-zero} & \\ \textsc{number} & \textit{zero} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

These descriptions show, that our RSRL encoding of natural numbers amounts to strips which are of a certain length: The natural number i is encoded as a strip of length i, i.e., as a *number* object that is mapped to an object of sort *zero* by applying the attribute interpretation of the attribute NUMBER i times.

Due to the nature of RSRL objects, these strips can be of arbitrary length, but even infinite and cyclic. However, only finite strips that correspond to natural numbers, so we must exclude infinite and cyclic configurations of objects. The properties of RSRL allow us to exclude these cases. A simple principle, the NUMBER PRINCIPLE (NP) fulfills both tasks.

(110) The NUMBER PRINCIPLE (NP):

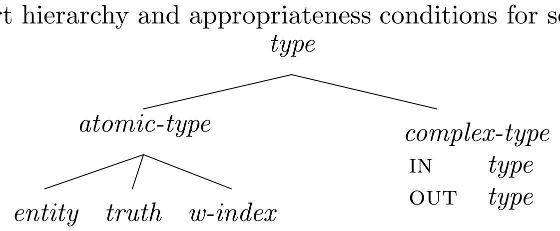
$$number \Rightarrow \mathbf{E}\boxed{1}\ \boxed{1}\begin{bmatrix} zero \end{bmatrix}$$

It should be noted that RSRL quantification always is quantification over components of the described object. By definition, a component is an object that can be reached from the described object via finite path of attributes. As a result, we can express a constraint that states that the configuration under every *number* object must contain a *zero* object. Objects of sort *zero* trivially satisfy the constraint. For objects of sort *non-zero*, it means that there can only be a finitely long path NUMBER... NUMBER, and that, at the end of this path, there is a *zero* object.

This principle excludes a cyclic configuration under a *number* object, because in a cyclic configuration, there is no object of sort *zero* at all. Similarly, in the case of an infinite configuration under a *number* object, there either is no *zero* object at all or, at least, the *zero* object cannot be reached by a finite path.

3.1.2. Types. The next kind of entity that occurs inside terms are semantic types. In Definition 1.1 on page 37 we defined the set of semantic types as the smallest set that contains the atomic types t, e and s and is closed under pairs. This is directly reflected in our RSRL encoding of semantic types as given in (111).

(111) Sort hierarchy and appropriateness conditions for semantic types:



Objects of sort *type* correspond to the elements of *Type* as defined for Ty2. In our example semantic term in (107b), we find the following semantic types: all the atomic types t, e and s, and the complex types et, se, $s(et)$ and $(se)t$. In (112) we give some AVM descriptions of *type* objects that correspond to these types.

(112) a. Description of the atomic types:

$$t\text{: } \begin{bmatrix} truth \end{bmatrix} \qquad e\text{: } \begin{bmatrix} entity \end{bmatrix} \qquad s\text{: } \begin{bmatrix} w\text{-}index \end{bmatrix}$$

b. Description of the type et:

$$\begin{bmatrix} c\text{-}type & \\ \text{IN} & entity \\ \text{OUT} & truth \end{bmatrix}$$

c. Description of the type se:

$$\begin{bmatrix} c\text{-}type & \\ \text{IN} & w\text{-}index \\ \text{OUT} & entity \end{bmatrix}$$

d. Description of the type $(se)t$:

$$\begin{bmatrix} c\text{-}type & \\ \text{IN} & \begin{bmatrix} c\text{-}type & \\ \text{IN} & w\text{-}index \\ \text{OUT} & entity \end{bmatrix} \\ \text{OUT} & truth \end{bmatrix}$$

e. Description of the type $s(et)$:

$$\begin{bmatrix} \textit{c-type} & \\ \textsc{in} & \textit{w-idex} \\ \textsc{out} & \begin{bmatrix} \textit{c-type} & \\ \textsc{in} & \textit{entity} \\ \textsc{out} & \textit{turth} \end{bmatrix} \end{bmatrix}$$

These examples illustrate how the encoding works: the atomic types are encoded as atomic sorts. Types of the form $\langle \tau, \tau' \rangle$ are encoded as objects of sort *c-type*, where the first type (τ) appears as the IN value, and the second type (τ') as the OUT value.

But, just as we saw with the encoding of natural numbers, it is not enough to provide a signature for semantic types. In order to make objects of sort *type* fully look like semantic types, we have to add three constraints to the grammar $\mathcal{TY}2$.

First, a principle is needed that ensures that in every complex type, all sub-types that look alike are actually token identical. We call this principle the TYPE IDENTITY PRINCIPLE (TyIP). We need a relation, `same-type`, to express what we mean by types that "look alike". For our purpose, two types look alike if (i), in case they are atomic, they are the same atomic type, and, (ii), in case they are complex, their components are identical types. In (113) this relation is defined.

(113) The relation `same-type`:

$$\texttt{same-type}(\boxed{1},\boxed{2}) \stackrel{\forall}{\Longleftarrow} \begin{pmatrix} \left(\boxed{1}\begin{bmatrix}\textit{entity}\end{bmatrix} \textbf{ and } \boxed{2}\begin{bmatrix}\textit{entity}\end{bmatrix}\right) \\ \textbf{or } \left(\boxed{1}\begin{bmatrix}\textit{truth}\end{bmatrix} \textbf{ and } \boxed{2}\begin{bmatrix}\textit{truth}\end{bmatrix}\right) \\ \textbf{or } \left(\boxed{1}\begin{bmatrix}\textit{w-index}\end{bmatrix} \textbf{ and } \boxed{2}\begin{bmatrix}\textit{w-index}\end{bmatrix}\right) \end{pmatrix}$$

$$\texttt{same-type}(\boxed{1},\boxed{2}) \stackrel{\forall}{\Longleftarrow} \left(\boxed{1}\begin{bmatrix} \textit{c-type} & \\ \textsc{in} & \boxed{3} \\ \textsc{out} & \boxed{4} \end{bmatrix} \textbf{ and } \boxed{2}\begin{bmatrix} \textit{c-type} & \\ \textsc{in} & \boxed{5} \\ \textsc{out} & \boxed{6} \end{bmatrix} \begin{array}{l} \textbf{and } \texttt{same-type}(\boxed{3},\boxed{5}) \\ \textbf{and } \texttt{same-type}(\boxed{4},\boxed{6}) \end{array} \right)$$

With the help of the relation `same-type`, we can state the TyIP:

(114) The TYPE IDENTITY PRINCIPLE (TyIP):

$$\textit{type} \Rightarrow \mathbf{A}\boxed{1}\mathbf{A}\boxed{2}\ (\texttt{same-type}(x,y) \Rightarrow \boxed{1} \approx \boxed{2})$$

The TyIP exploits the fact that RSRL quantifiers quantify over the set of components of the described object. Thus, the TyIP enforces identities only within a given object.

In our example types in (112) there was no type occurring twice within the same type. Consider, however, the type $(s((se)t))((s((se)t))t)$ which is the type of the basic translation of the determiner *every*, i.e., the type of the term $\lambda P \lambda Q.\forall x[P_{@}(x) \rightarrow Q_{@}(x)]$.

(115) Description of the type $(s((se)t))((s((se)t))t)$:

$$\begin{bmatrix} \textit{c-type} & & \\ \textsc{in} & \boxed{1} & \begin{bmatrix} \textit{c-type} & \\ \textsc{in} & \boxed{2}\ \textit{w-index} \\ \textsc{out} & \begin{bmatrix} \textit{c-type} & \\ \textsc{in} & \begin{bmatrix} \textit{c-type} & \\ \textsc{in} & \boxed{2} \\ \textsc{out} & \textit{entity} \end{bmatrix} \\ \textsc{out} & \boxed{3}\ \textit{truth} \end{bmatrix} \end{bmatrix} \\ \textsc{out} & \begin{bmatrix} \textit{c-type} & \\ \textsc{in} & \boxed{1} \\ \textsc{out} & \boxed{3} \end{bmatrix} & \end{bmatrix}$$

The description in (115) expresses that the type of the first semantic argument of the quantifier $s((se)t)$ is identical to that of the second semantic argument. In the AVM, we use the tag $\boxed{1}$ to express this identity. Furthermore, within this type, the type s occurs twice. The tag $\boxed{2}$ is used to indicate that in any object which is described by (115), every path that leads to an object of the sort *w-index*, leads to the same object. Finally, the tag $\boxed{3}$ refers to a *truth* object. Again, there is only one *truth* object that is a component of the object described by (115).

It is the purpose of the TyIP to avoid the situation where the objects of sort *type* express finer distinctions than the actual semantic types. For illustration, consider (116).

(116) Another description of the type $(s((se)t)((s((se)t))t)$:

$$\begin{bmatrix} \textit{c-type} & \\ \text{IN} & \boxed{1}\begin{bmatrix}\textit{c-type} & \\ \text{IN} & \boxed{2}\ \textit{w-index} \\ \text{OUT} & \begin{bmatrix}\textit{c-type} & \\ \text{IN} & \begin{bmatrix}\textit{c-type} & \\ \text{IN} & \boxed{2} \\ \text{OUT} & \textit{entity}\end{bmatrix} \\ \text{OUT} & \boxed{3}\ \textit{truth}\end{bmatrix}\end{bmatrix} \\ \text{OUT} & \begin{bmatrix}\textit{c-type} & \\ \text{IN} & \boxed{1} \\ \text{OUT} & \boxed{4}\end{bmatrix} & \end{bmatrix}$$
$$\textbf{and not } \boxed{3} \approx \boxed{4}$$

An object described by (115) differs from an object described by (116) only with respect to the paths that lead to the same object. In the first case, the paths :IN OUT OUT, :OUT IN OUT OUT, and :OUT OUT lead to the same object, referred to by the tag $\boxed{3}$. In the second case, the path :OUT OUT leads to a different object. This is indicated by the tag $\boxed{4}$ and the requirement that $\boxed{3}$ and $\boxed{4}$ refer to different objects. Within our "empirical" domain, the set *Type*, such a differentiation is not attested. Therefore, we eliminate the distinction from the denotation of the grammar $\mathcal{TY}2$ as well by imposing the maximal number of identities possible.[1]

The second restriction that we impose on *type* objects in order to make them look like "real" semantic types is a requirement that all types must be non-cyclic. To achieve this, we state a constraint, the TYPE NON-CYCLICITY PRINCIPLE (TyNP). This principle requires the relation `component`.

The relation `component` holds of two objects u and u' iff u is a component of u', i.e., either u and u' are identical or there is an attribute α such that u is a component of the α value of u'. The relation `component` can be defined for any signature with a finite set of attribute names, by listing all the attributes defined in the signature.[2] In (117) we define the relation `component` for the signature of $\mathcal{TY}2$.[3]

(117) The relation `component`

$$\texttt{component}(\boxed{1},\boxed{2}) \stackrel{\forall}{\Longleftarrow} \boxed{1} \approx \boxed{2}$$

[1]Alternatively, we could also have forbidden all identities within *type* objects. We have chosen to require identities rather than non-identities because this makes the RSRL formalization easier.

[2]A general definition of the relation `component` for each signature with a finite set of attributes is given in Section 3.1.4, number (129).

[3]In (117) we also mention those attributes that will not be introduced until Section 3.1.3. In (129), we show how for an arbitrary set of attributes, a relation `component` can be defined.

$$\texttt{component}(\boxed{1},\boxed{2}) \stackrel{\forall}{\Longleftarrow} \left(\begin{array}{l} \left(\begin{array}{l} \quad\; {}^{\boxed{2}}\left[\text{IN } \boxed{3}\right] \ \textbf{or} \ {}^{\boxed{2}}\left[\text{OUT } \boxed{3}\right] \ \textbf{or} \ {}^{\boxed{2}}\left[\text{NUMBER } \boxed{3}\right] \\ \textbf{or} \ {}^{\boxed{2}}\left[\text{ARG } \boxed{3}\right] \ \textbf{or} \ {}^{\boxed{2}}\left[\text{ARG1 } \boxed{3}\right] \ \textbf{or} \ {}^{\boxed{2}}\left[\text{ARG2 } \boxed{3}\right] \\ \textbf{or} \ {}^{\boxed{2}}\left[\text{FUNC } \boxed{3}\right] \ \textbf{or} \ {}^{\boxed{2}}\left[\text{VAR } \boxed{3}\right] \end{array}\right) \\ \textbf{and} \ \ \texttt{component}(\boxed{1},\boxed{3}) \end{array}\right)$$

Using the relation `component`, we can formalize a ban on cyclic objects of sort *type*. For atomic types, cyclicity cannot arise. To exclude cyclicity, we add a constraint to the grammar which states that for each *c-type*, there is no path other than the trivial path ": " that leads back to the object itself.

(118) The TYPE NON-CYCLICITY PRINCIPLE (TyNP):

$$\textit{complex-type} \Rightarrow \mathbf{A}\boxed{1}\ \left(\left(\left[\text{IN } \boxed{1}\right] \ \textbf{or} \ \left[\text{OUT } \boxed{1}\right]\right) \Rightarrow \textbf{not} \ \ \texttt{compontent}(:,\boxed{1})\right)$$

The way the TyNP is expressed in the TyNP, it says that for each object o of sort *c-type*, o is not a component of its own IN value, nor a component of its own OUT value.

Finally, we need a third constraint on objects of sort *type* to ensure that they are finite. To achieve this goal, we use the same technique that we have employed in Section 2.1, constraint (63b), to ensure that phrases have only a finite number of components, i.e., we make use of the finiteness requirement imposed on chains in the definition of RSRL. The TYPE FINITENESS PRINCIPLE (TyFP) states that for any *type* object, there is a chain, a, such that exactly each component, $\boxed{1}$, of the objects described is a member of a. By definition, chains are finite. As the chain required in the TyFP must contain all components of the *type* object, the object itself must be finite.

(119) The TYPE FINITENESS PRINCIPLE (TyFP):

$$\textit{type} \Rightarrow \mathbf{E}a \left(\begin{array}{l} {}^{a}\left[\textit{chain}\right] \\ \textbf{and} \ \mathbf{A}\boxed{1}\ \left(\texttt{component}(\boxed{1},:) \ \Rightarrow \texttt{member}(\boxed{1}, a)\right) \end{array}\right)$$

The TyFP uses a relation `member` which we have not defined so far. In (120), the necessary definitions are given. An object o stands in the relation `member` with some *list* object or some chain l provided (i) o is the first element of l, or (ii) o stands in the relation `member` with the rest of l.

(120) The relation `member`:

$$\texttt{member}(\boxed{1},\boxed{2}) \stackrel{\forall}{\Longleftarrow} {}^{\boxed{2}}\left[\text{FIRST } \boxed{1}\right]$$

$$\texttt{member}(\boxed{1},\boxed{2}) \stackrel{\forall}{\Longleftarrow} \left({}^{\boxed{2}}\left[\text{REST } \boxed{3}\right] \ \textbf{and} \ \texttt{member}(\boxed{1},\boxed{3})\right)$$

Note that the chain a may contain multiple occurrences of the same component of the described object. An extra relation could be added to avoid this effect, but it is not necessary, because we are only interested in finiteness, not in the actual number of components.

In this subsection we have given those parts of the signature and the theory of the grammar $\mathcal{TY}2$ that are concerned with semantic types. In the definition of semantic types in Definition 1.1 some properties of semantic types came for free, such as their finiteness and non-cyclicity. These properties must, however, be explicitly stated in our RSRL encoding. Note, furthermore, that we must make use of chains to encode the finiteness requirement. With the TyIP we have furthermore guaranteed that the grammar $\mathcal{TY}2$ does not introduce distinctions between *type* objects that are not relevant for the empirical domain, i.e., the set of semantic types.

In the next subsection, we will discuss the RSRL encoding of terms of Ty2. For that encoding, we will also provide an identity principle, a non-cyclicity principle, and a finiteness principle.

3.1.3. Terms. The availability of the signature and the necessary constraints for the objects that correspond to natural numbers and to semantic types makes it possible to address the RSRL encoding of terms of Ty2. For this purpose we introduce the sort *meaningful-expression* (*me*). Just as with the sorts *number* and *type*, we intend that an object of sort *me* corresponds to a term of Ty2.

An attribute TYPE is defined on the sort *me*. The species below *me* mimic the syntactic constructs of the language Ty2: variables, constants, application, abstraction, and equation.

(121) The sort hierarchy below *me*

me	TYPE	*type*
var(iable)	NUMBER	*number*
const(ant)		
const$_1$		
...		
const$_n$		
appl(ication)	FUNC(TOR)	*me*
	ARG(UMENT)	*me*
abstr(action)	VAR(IABLE)	*variable*
	ARG(UMENT)	*me*
equ(ation)	ARG(UMENT)1	*me*
	ARG(UMENT)2	*me*

In (126) we will give the $\mathcal{TY}2$ description of the Ty2 term given in (107b). It can, however, already be seen how the RSRL encoding of terms of Ty2 works. For illustration, let us consider variables. In Ty2, a particular variable is fully specified, and distinguished from all other terms of the language, if (i) we know that it is a variable, (ii) we know its semantic type, and (iii) we know its number. These three characteristics are reflected in the signature in (121). There, we have introduced a sort *variable*. Just as all subsorts of the sort *me*, it has an attribute TYPE defined on it which takes values of the sort *type*. Finally, as stated in (121), a further attribute NUMBER is defined on the sort *var* whose values are of sort *number*.

This indicates that the parts of the signature given in (121) express all the necessary differences that exist between distinct terms of Ty2. We will add some principles to the grammar $\mathcal{TY}2$ that will ensure that the *me* objects admitted by the grammar respect the structure of Ty2 terms as given in Definition 1.5.

In the definition of Ty2 terms in Section 1.3.2, the semantic type of complex terms, such as applications, abstraction and equation is fully predictable from the semantic types of their subterms. For example, the definition of application guarantees that the combination of the terms $v_{se,0}$ and $v_{s,0}$ in (107b) results in a term of type e. We impose a set of constraints to restrict the TYPE values of the recursive subsorts of *me* in the same way.

(122) Type restrictions:

a. $$appl \Rightarrow \begin{bmatrix} \text{TYPE} & \boxed{2} \\ \text{FUNC TYPE} & \begin{bmatrix} \text{IN} & \boxed{1} \\ \text{OUT} & \boxed{2} \end{bmatrix} \\ \text{ARG TYPE} & \boxed{1} \end{bmatrix}$$

b. $$abstr \Rightarrow \begin{bmatrix} \text{TYPE} & \begin{bmatrix} \text{IN} & \boxed{1} \\ \text{OUT} & \boxed{2} \end{bmatrix} \\ \text{VAR TYPE} & \boxed{1} \\ \text{ARG TYPE} & \boxed{2} \end{bmatrix}$$

c. $$equ \Rightarrow \begin{bmatrix} \text{TYPE} & truth \\ \text{ARG1 TYPE} & \boxed{1} \\ \text{ARG2 TYPE} & \boxed{1} \end{bmatrix}$$

The first constraint enforces that for an *application* object, the TYPE of the ARGUMENT value is identical to the TYPE IN value of the FUNCTOR value. Furthermore, the TYPE value of the APPLICATION object must be identical to the TYPE OUT value of the FUNCTOR value.

For objects of the sort *abstraction*, which correspond to lambda abstractions, the TYPE IN value is identical to the TYPE value of the VARIABLE value, and the TYPE OUT value is identical to the ARG TYPE value. These identities reflect the fact that if we form the lambda abstraction of a term ϕ_τ and a variable $v_{\tau',i}$, the resulting term, $(\lambda v_{\tau',i}.\phi_\tau)_{\tau'\tau}$ is of type $\langle \tau', \tau \rangle$.

For objects of the sort *equation*, we require that the TYPE values of the ARGUMENT1 and the ARGUMENT2 values be identical. In addition, the TYPE value of the *equation* object is restricted to be of sort *truth*.

So far, we have addressed the TYPE values of recursively built *me* objects, which are fully predictable by the constraints in (122). What remains to be addressed is the TYPE value of variables and non-logical constants, i.e., of objects of sort *constant*.

In the case of variables, we do not want to impose any restrictions on the possible TYPE values, other than requiring that the TYPE value be a *type* object. This will guarantee that in the denotation of the grammar, we can find variables of each semantic type, just as the set of variables of Ty2 contains variables of each semantic type.

For constants of Ty2, or more precisely of $\mathrm{Ty2}_\mathcal{C}$, the function $\mathcal{C}$ restricts the semantic type. To imitate this, there is a constraint for each species $const_i$ below *const* that fixes its semantic type. In (123), such a constraint is given for a non-logical constant species of type $\langle s, \langle e, t \rangle \rangle$, like walk' from example (107b), which we assume to be the species $const_{127}$.

(123) $$const_{127} \Rightarrow \begin{bmatrix} \text{TYPE} & \begin{bmatrix} \text{IN} & w\text{-}index \\ \text{OUT} & \begin{bmatrix} \text{IN} & entity \\ \text{OUT} & truth \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

We collect the type restrictions for the subsorts of *const* and those in (122) in the set of TYPE RESTRICTION PRINCIPLES (TRP). The TRP are part of the theory of $\mathcal{TY}2$.

For the sort *type*, we assumed three additional principles to guarantee (i) the maximal number of identities (via the TyIP in (114)), (ii) that types are not cyclic (via the TyNP in (118)), and (iii) that types are finite (via the TyFP in (119)). On terms, we must impose the very same kinds of restrictions.

Let us first consider the equivalent of the TyIP for terms. For *type* objects we required the maximal number of identities possible. To do the same for *me* objects, we, first, need a characterization of the circumstances in which two *me* objects are the same in the relevant sense. This is the case iff (i), they are both variables with the same TYPE and the same NUMBER value, or (ii), they are *constant* objects of the same species, or (iii) they are of the same species and have components which are the same in the relevant sense. To express this characterization, we must first state what it means for two *number* objects to be the same. This is done in the relation `same-number` in (124a). Then, we define the relation `same-term` in (124b). This relation makes use of the previously defined relation `same-type` (see (113) and of the relation `same-number` in its clause for *var* objects.

(124) a. The relation `same-number`:

$$\texttt{same-number}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow}$$
$$\boxed{1}\begin{bmatrix} \textit{zero} \end{bmatrix} \textbf{ and } \boxed{2}\begin{bmatrix} \textit{zero} \end{bmatrix}$$

$$\texttt{same-number}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow}$$
$$\boxed{1}\begin{bmatrix} \textit{non-zero} \\ \text{NUMBER} \ \boxed{3} \end{bmatrix} \textbf{ and } \boxed{2}\begin{bmatrix} \textit{non-zero} \\ \text{NUMBER} \ \boxed{4} \end{bmatrix}$$
$$\textbf{and } \texttt{same-number}(\boxed{3},\boxed{4})$$

b. The relation `same-term`:

$$\texttt{same-term}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow}$$
$$\boxed{1}\begin{bmatrix} \textit{var} & \\ \text{TYPE} & \boxed{3} \\ \text{NUMBER} & \boxed{4} \end{bmatrix} \textbf{ and } \boxed{2}\begin{bmatrix} \textit{var} & \\ \text{TYPE} & \boxed{5} \\ \text{NUMBER} & \boxed{6} \end{bmatrix}$$
$$\textbf{and } \texttt{same-type}(\boxed{3},\boxed{5})$$
$$\textbf{and } \texttt{same-number}(\boxed{4},\boxed{6})$$

for each $\textit{const}_i \sqsubseteq \textit{const}$:

$$\texttt{same-term}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow}$$
$$\boxed{1}\begin{bmatrix} \textit{const}_i \end{bmatrix} \textbf{ and } \boxed{2}\begin{bmatrix} \textit{const}_i \end{bmatrix}$$

$$\texttt{same-term}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow}$$
$$\boxed{1}\begin{bmatrix} \textit{appl} & \\ \text{TYPE} & \boxed{3} \\ \text{FUNC} & \boxed{4} \\ \text{ARG} & \boxed{5} \end{bmatrix} \textbf{ and } \boxed{2}\begin{bmatrix} \textit{appl} & \\ \text{TYPE} & \boxed{6} \\ \text{FUNC} & \boxed{7} \\ \text{ARG} & \boxed{8} \end{bmatrix}$$
$$\textbf{and } \texttt{same-type}(\boxed{3},\boxed{6})$$
$$\textbf{and } \texttt{same-term}(\boxed{4},\boxed{7})$$
$$\textbf{and } \texttt{same-term}(\boxed{5},\boxed{8})$$

$$\texttt{same-term}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow}$$
$$\boxed{1}\begin{bmatrix} \textit{abstr} & \\ \text{TYPE} & \boxed{3} \\ \text{VAR} & \boxed{4} \\ \text{ARG} & \boxed{5} \end{bmatrix} \textbf{ and } \boxed{2}\begin{bmatrix} \textit{abstr} & \\ \text{TYPE} & \boxed{6} \\ \text{VAR} & \boxed{7} \\ \text{ARG} & \boxed{8} \end{bmatrix}$$
$$\textbf{and } \texttt{same-type}(\boxed{3},\boxed{6})$$
$$\textbf{and } \texttt{same-term}(\boxed{4},\boxed{7})$$
$$\textbf{and } \texttt{same-term}(\boxed{5},\boxed{8})$$

$$\texttt{same-term}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow}$$
$$\boxed{1}\begin{bmatrix} \textit{equ} & \\ \text{TYPE} & \boxed{3} \\ \text{ARG1} & \boxed{4} \\ \text{ARG2} & \boxed{5} \end{bmatrix} \textbf{ and } \boxed{2}\begin{bmatrix} \textit{equ} & \\ \text{TYPE} & \boxed{6} \\ \text{ARG1} & \boxed{7} \\ \text{ARG2} & \boxed{8} \end{bmatrix}$$
$$\textbf{and } \texttt{same-type}(\boxed{3},\boxed{6})$$
$$\textbf{and } \texttt{same-term}(\boxed{4},\boxed{7})$$
$$\textbf{and } \texttt{same-term}(\boxed{5},\boxed{8})$$

Given the relations `same-number` and `same-term`, we can define the TERM IDENTITY PRINCIPLE (TIP) in analogy to the TyIP in (114). The TIP enforces the maximal number of identities possible within a *me* object.

(125) The TERM IDENTITY PRINCIPLE (TIP):
$me \Rightarrow \mathbf{A}\boxed{1}\mathbf{A}\boxed{2}\ (\texttt{same-term}(\boxed{1},\boxed{2}) \Rightarrow \boxed{1} \approx \boxed{2})$

As a consequence of the TIP, within a single *me* object, all *const* objects of the same species are required to be identical. The same is true for all *variable* objects that have the same NUMBER and TYPE values. For larger *me* objects, these identities are rather rare in natural examples.

Given the signature for *me* objects, the TRP and the TIP, we can describe an *me* object that corresponds to the term in (107b).

(126) Description of the term $(\lambda v_{se,0}.((\mathsf{walk}'_{s(et)} v_{s,0})(v_{se,0} v_{s,0})_e)_t)_{(se)t}$:

$$\begin{bmatrix}
\textit{abstr} & & \\
\text{TYPE} & & \begin{bmatrix} \textit{c-type} & & \\ \text{IN} & \boxed{1} & \begin{bmatrix} \textit{c-type} & & \\ \text{IN} & \boxed{2} & \textit{w-index} \\ \text{OUT} & \boxed{3} & \textit{entity} \end{bmatrix} \\ \text{OUT} & \boxed{4} & \textit{truth} \end{bmatrix} \\
\text{VAR} & \boxed{5} & \begin{bmatrix} \textit{var} & & \\ \text{TYPE} & \boxed{1} & \\ \text{NUMBER} & \boxed{6} & \textit{zero} \end{bmatrix} \\
\text{ARG} & & \begin{bmatrix} \textit{appl} & & \\ \text{TYPE} & \boxed{4} & \\ \text{FUNC} & & \begin{bmatrix} \textit{appl} & & \\ \text{TYPE} & \boxed{8} & \begin{bmatrix} \textit{c-type} & \\ \text{IN} & \boxed{3} \\ \text{OUT} & \boxed{4} \end{bmatrix} \\ \text{FUNC} & & \begin{bmatrix} \textit{const}_{127} & \\ \text{TYPE} & \begin{bmatrix} \textit{c-type} & \\ \text{IN} & \boxed{2} \\ \text{OUT} & \boxed{8} \end{bmatrix} \end{bmatrix} \\ \text{ARG} & \boxed{7} & \begin{bmatrix} \textit{var} & \\ \text{TYPE} & \boxed{2} \\ \text{NUMBER} & \boxed{6} \end{bmatrix} \end{bmatrix} \\ \text{ARG} & & \begin{bmatrix} \textit{appl} & \\ \text{TYPE} & \boxed{3} \\ \text{FUNC} & \boxed{5} \\ \text{ARG} & \boxed{7} \end{bmatrix} \end{bmatrix}
\end{bmatrix}$$

As can be seen in (126), the overall term is a lambda abstraction, and therefore corresponds to an object of sort *abstr*. The type of the overall term, $(se)t$, is described as in (112d). The variable bound by the lambda operator, $v_{se,0}$, is described as being a *var* object. It follows from the type restriction in (122b) that the TYPE value of the VAR value of the abstraction is identical to the TYPE IN value of the abstraction. This is expressed with the tag $\boxed{1}$. The ARG value of the overall abstraction is an *application* object. Again, the type restriction in (122b) enforces the type of the ARG TYPE value ($\boxed{4}$) to be identical to the TYPE OUT value of the overall *abstr* object.

Let us, next, consider the *appl* object as it is described as the ARG value in (126). Its FUNC value is, again, a *appl* object. The FUNC value of this object is a constant of sort $const_{127}$. The part of the TRP given in (123) determines that this constant has a TYPE value that corresponds to the type $s(et)$. We have said, that the species $const_{127}$ should correspond to the non-logical constant walk' of Ty2. The ARG value of this most embedded *appl* object is a variable of type s, as required by the TRP. In fact, it is the special variable $v_{s,0}$, for which we often write "@". In (126), it is described as a *var* object. Its TYPE value is the tag $\boxed{2}$ which refers to an object of sort *w-index*, the sort that corresponds to the type

s of Ty2. Furthermore, the NUMBER value is given as the tag $\boxed{6}$. This tag refers to an object of sort *zero*, i.e., to a *number* object that corresponds to the natural number 0.

The ARG value of the upper *appl* object (as reached with the path ARG ARG) is described as being syntactically complex, i.e., of sort *appl* again. Its functor is the variable bound by the lambda abstractor. In Ty2 this is expressed by using the same number and type as subscript for the variable, in our RSRL encoding we actually use the same variable, as indicated by the tag $\boxed{5}$. Finally, the argument of this variable is the special variable $v_{s,0}$, which occurred already within the most deeply embedded *appl* object (tag $\boxed{7}$).

The *me* object that we described in (126) is finite and non-cyclic. For *me* objects with this property, we can give a corresponding Ty2 term. So far the grammar does not provide that all *me* objects in the denotation of the grammar have this property. Therefore, we will add two more constraints to the grammar $\mathcal{TY}2$.

Given the relation `component` as defined in (117), we can express the TERM NON-CYCLICITY PRINCIPLE (TNP). The TNP requires all *me* objects to be acyclic. The TNP says that an *me* object must not occur as a component of the value of any of its attributes.

(127) The TERM NON-CYCLICITY PRINCIPLE (TNP):

$$me \Rightarrow \mathbf{A}\boxed{1}\left(\left(\begin{matrix}\left[\text{FUNC } \boxed{1}\right] \textbf{ or } \left[\text{ARG } \boxed{1}\right] \textbf{ or } \left[\text{ARG1 } \boxed{1}\right] \\ \textbf{or } \left[\text{ARG2 } \boxed{1}\right] \textbf{ or } \left[\text{VAR } \boxed{1}\right]\end{matrix}\right) \Rightarrow \textbf{not } \ \texttt{component}(:,\boxed{1})\right)$$

Finally, parallel to the constraint on *type*, we want all *me* objects to be finite, i.e., to contain only a finite number of components. This is expressed in the TERM FINITENESS PRINCIPLE (TFP).

(128) The TERM FINITENESS PRINCIPLE:

$$me \Rightarrow \mathbf{E}a\left({}^{a}\!\left[\mathit{chain}\right] \textbf{ and } \mathbf{A}\boxed{1}\left(\texttt{component}(\boxed{1},:) \Rightarrow \texttt{member}(\boxed{1},a)\right)\right)$$

The TFP is the last principle needed for our RSRL encoding of Ty2. We have shown in the description in (126) that there is a clear correspondence between terms of Ty2 and objects of sort *me*, just as there has been a correspondence between natural numbers and objects of sort *number* and semantic types and objects of sort *type*. In the following sections, we will make this correspondence formally precise.

Before we turn to the formal relation between the grammar $\mathcal{TY}2$ and the language Ty2, we will devote some space to streamline the grammar $\mathcal{TY}2$.

3.1.4. Simplification of $\mathcal{TY}2$. For every kind of entities in the grammar of $\mathcal{TY}2$, namely, *number*, *type* and *me* objects, a large number of principles is of the same shape. These are the principles that require the objects to have as many token identities as possible (TyIP and TIP), the non-cyclicity principles (TyNP, TNP) and the finiteness principles (TyFP and TFP). Because numbers have a very simple ontology, no identities could possibly arise and finiteness and non-cyclicity can be expressed in a single principle, the NP. In this subsection, we provide general definitions of the relations `component` and `are-copies`, which are used in the principles above. We can then reduce 7 principles to 3 basic principles.[4]

(129) The relation `component`
For each RSRL grammar with the signature Σ with $\mathcal{A}$ as the set of attributes, if $\mathcal{A}$ is finite, then there is a relation `component` in $\mathcal{R}$ such that the following description is in the theory of the grammar:

[4]Note that we also talk about components of a chain and copies of a chain in the definitions in (129) and (130). This is done by using not only the attributes and species given by the signature, but also the quasi-attributes and the quasi-species of definition 2.4.

$$\mathbf{A}\boxed{1}\mathbf{A}\boxed{2}\left(\texttt{component}(\boxed{1},\boxed{2}) \Leftrightarrow \left(\boxed{1} \approx \boxed{2} \textbf{ or } \bigvee_{\alpha\in\widehat{\mathcal{A}}} (\texttt{component}(\boxed{1},\boxed{2}\alpha))\right)\right)$$

In the description in (129), we use the symbol "$\bigvee$" as abbreviation of a disjunction, i.e., for each attribute α in the augmented signature, there is a disjunct of the form $\texttt{component}(\boxed{1},\boxed{2}\alpha)$.

The set of attributes is required to be finite, as otherwise the description given in (117) would contain an infinite disjunction. The use of the relation `component` in this thesis is unproblematic, as all the grammars proposed here assume a finite set of attributes.

(130) The relation `are-copies`:
For each RSRL grammar with $\mathcal{S}$ as the set of species and $\mathcal{A}$ as the set of attributes as provided by the signature, if $\mathcal{S}$ and $\mathcal{A}$ are finite, then there is a relation `are-copies` in $\mathcal{R}$, and the following description is in the theory of the grammar:

$$\mathbf{A}\boxed{1}\ \mathbf{A}\boxed{2}\left(\texttt{are-copies}(\boxed{1},\boxed{2}) \Leftrightarrow \left(\begin{array}{l}\bigvee_{\sigma\in\widehat{\mathcal{S}}} (\boxed{1}{\sim}\sigma \textbf{ and } \boxed{2}{\sim}\sigma) \\ \textbf{and} \bigwedge_{\alpha\in\widehat{\mathcal{A}}} (\boxed{1}\alpha \approx \boxed{1}\alpha \Rightarrow \texttt{are-copies}(\boxed{1}\alpha,\boxed{2}\alpha))\end{array}\right)\right)$$

In the definition of the relation `are-copies`, we use the symbol "$\bigvee$" to express a big disjunction, i.e., in this case, it is a disjunction such that for each species σ in the signature, there is a disjunct of the form $\boxed{1}{\sim}\sigma$ **and** $\boxed{2}{\sim}\sigma$. In the last line of the constraint in (130), there is the symbol "$\bigwedge$". Analogously to the symbol "$\bigvee$", the second symbol is used to express a big conjunction. In each of the conjuncts, there is the description $\boxed{1}\alpha \approx \boxed{1}\alpha$. This describes an object iff the attribute α is defined on the object denoted by the variable $\boxed{1}$.

The relation `are-copies` can only be defined in this way for signatures that assume a finite set of species and a finite set of attributes, because otherwise, the description in (130) would be infinite. Again, the signatures proposed in this thesis have this property, so we can safely use the relations defined in (129) and (130).

Given these general relations, we can give the three general principles instead of the seven specific principles considered so far: the GENERAL IDENTITY PRINCIPLE (GIP), the GENERAL NON-CYCLICITY PRINCIPLE (GNP) and the GENERAL FINITENESS PRINCIPLE.

(131) a. The GENERAL IDENTITY PRINCIPLE (GIP):
$$\mathbf{A}\boxed{1}\ \mathbf{A}\boxed{2}\ (\texttt{are-copies}(\boxed{1},\boxed{2}) \Rightarrow \boxed{1} \approx \boxed{2})$$
b. The GENERAL NON-CYCLICITY PRINCIPLE (GNP):
$$\mathbf{A}\boxed{1}\left(\bigvee_{\alpha\in\mathcal{A}} ((\boxed{1} \approx {:}\alpha) \Rightarrow \textbf{not}\ \texttt{component}({:},\boxed{1}))\right)$$
c. The GENERAL FINITENESS PRINCIPLE (GFP):
$$\mathbf{E}a\left({}^{a}\Big[\mathit{chain}\Big] \textbf{ and } \mathbf{A}\boxed{1}\ (\texttt{component}(\boxed{1},{:}) \Rightarrow \texttt{is-member}(\boxed{1},a))\right)$$

Thus we can assume that $\mathcal{TY}2$ contains the GIP instead of the TyIP and the TIP; the GNP instead of the NP, the TyNP and the TNP; and the GFP instead of the TyFP and the TFP. Of course, we still need the TRP.

The general relations and principles defined in this subsection are useful for linguistic theories in general, as componenthood and copyhood must be expressed frequently. The treatment of duplication in morphophonology of Höhle 1999 for example relies on copyhood rather than token identity.[5] Finiteness and acyclicity are requirements that are necessary in order to restrict the PHON list, as done in Richter and Sailer 1995 and Richter et al. 1999.

[5]In the grammar $\mathcal{TY}2$ we require that all copies be identical. Such a requirement is, of course, not used in other RSRL grammars. As mentioned above, the phenomena reported in Höhle 1999 show that there is a distinction between copyhood and identity in phonology. Similarly, the Binding Theory of Pollard and Sag

3.2. Ty2 as a model of $\mathcal{TY}2$

Given the signature, and the Number Principle (110), the Type Identity Principle (114), the Type Non-Cyclicity Principle (118), the Type Finiteness Principle (119), the Type Restriction Principle in (122) and (123), the Term Identity Principle (125), the Term Non-Cyclicity-Principle (127) and the Term Finiteness Principle (128), or their generalized counterparts, the grammar $\mathcal{TY}2$ is complete. Following King 1999 and Richter 1999, 2000, (R)SRL grammars are used to describe certain empirical domains. In our case the empirical domain is the set of Ty2 terms as defined in Section 1.3.2. An (R)SRL grammar has achieved its goal iff it can be shown that the empirical domain is an exhaustive model of the grammar. In the case of Ty2 we are in the fortunate position to know exactly how the empirical domain is structured (in contrast to what is the case in a grammar for, say, a natural language). We can show that the natural numbers, the set of semantic types, *Type*, and the set of Ty2 terms together are the universe of an exhaustive model of $\mathcal{TY}2$. All we have to add are the relations defined above with their intended interpretation.

In this section, we will show that the grammar $\mathcal{TY}2$ describes the language Ty2. To show this, we will prove the following proposition.

PROPOSITION 3.1

There is an exhaustive model $\mathsf{I}_{Ty2} = \langle \mathsf{U}_{Ty2}, \mathsf{S}_{Ty2}, \mathsf{A}_{Ty2}, \mathsf{R}_{Ty2} \rangle$ *of* $\mathcal{TY}2$ *such that* $\mathsf{U}_{Ty2} = \mathbb{N} \cup Type \cup Ty2$.

To show this proposition, we define the model $\mathsf{I}_{\mathrm{Ty2}}$, which we will call the *intended model* of the grammar $\mathcal{TY}2$. The universe of $\mathsf{I}_{\mathrm{Ty2}}$ contains exactly the natural numbers, the semantic types and the terms of Ty2.

DEFINITION 3.2 *(intended model* I_{Ty2}*)*

Let $Const = \{c_1, \ldots, c_n\}$ *be a finite set of constant symbols,*

let $Type$ *be the set of types generated over the atomic types* e, t *and* s,

let $Var = \{v_{\tau,i} | i \in \mathbb{N}, \tau \in Type\}$ *be the set of variables,*

and let $\mathcal{C}$ *be a function from* $Const$ *to* $Type$.

Let $Ty2_{\mathcal{C}}$ *be the set of terms generated according to definition in Section 1.3.2.*

We define the interpretation $\mathsf{I}_{Ty2} = \langle \mathsf{U}_{Ty2}, \mathsf{S}_{Ty2}, \mathsf{A}_{Ty2}, \mathsf{R}_{Ty2} \rangle$ *of the signature of* $\mathcal{TY}2$ *as follows:*

$\mathsf{U}_{\mathrm{Ty2}} = \mathbb{N} \cup Type \cup Ty2$,

$\mathsf{S}_{\mathrm{Ty2}}$:

- $\mathsf{S}_{Ty2}(0) = zero$,
- for each $i \in \mathbb{N}$, $\mathsf{S}_{Ty2}(i+1) = non\text{-}zero$,
- $\mathsf{S}_{Ty2}(e) = entity$,
- $\mathsf{S}_{Ty2}(t) = truth$,
- $\mathsf{S}_{Ty2}(s) = w\text{-}index$,
- for each $\alpha, \beta \in Type$, $\mathsf{S}_{Ty2}(\langle \alpha, \beta \rangle) = complex\text{-}type$,
- for each $i \in \mathbb{N}$, for each $\tau \in Type$, $\mathsf{S}_{Ty2}(v_{\tau,i}) = variable$,

1994 relies on such a difference, i.e., in a sentence such as (i), the INDEX values of the NPs *he* and *Peter* are copies of each other, but the Binding Theory requires that they be distinct *index* objects.

(i) He_i likes $\mathrm{Peter}_{j,*i}$

- *for each* $c_i \in Const$, $\mathsf{S}_{Ty2}(c_i) = const_i$,
- *for each* $\phi, \psi \in Ty2$, *for each* $v \in Var$, *for each* $\tau, \tau' \in Type$,
  - $\mathsf{S}_{Ty2}((\phi\psi)_\tau) = application$
  - $\mathsf{S}_{Ty2}((\lambda v_\tau.\phi)_{\tau'}) = abstraction$,
  - $\mathsf{S}_{Ty2}((\phi_\tau = \psi_\tau)_t) = equation$.

A_{Ty2} :

- *for each* $i \in \mathbb{N}$, $\mathsf{A}_{Ty2}(\text{NUMBER})(i+1) = i$,
- *for each* $\alpha, \beta \in Type$
  $\mathsf{A}_{Ty2}(\text{IN})(\langle\alpha,\beta\rangle) = \alpha$ *and*
  $\mathsf{A}_{Ty2}(\text{OUT})(\langle\alpha,\beta\rangle) = \beta$,
- *for each* $\tau \in Type$, *for each* $\phi_\tau \in Ty2$,
  $\mathsf{A}_{Ty2}(\text{TYPE})(\phi_\tau) = \tau$,
- *for each* $i \in \mathbb{N}$, *for each* $\tau \in Type$, *for each* $v_{\tau,i} \in Var$,
  $\mathsf{A}_{Ty2}(\text{NUMBER})(v_{\tau,i}) = i$,
- *for each* $(\phi\psi)_\tau \in Ty2$,
  $\mathsf{A}_{Ty2}(\text{FUNC})((\phi\psi)_\tau) = \phi$, *and*
  $\mathsf{A}_{Ty2}(\text{ARG})((\phi\psi)_\tau) = \psi$,
- *for each* $(\lambda v.\phi)_\tau \in Ty2$,
  $\mathsf{A}_{Ty2}(\text{VAR})((\lambda v.\phi)_\tau) = v$, *and*
  $\mathsf{A}_{Ty2}(\text{ARG})((\lambda v.\phi)_\tau) = \phi$, *and*
- *for each* $(\phi = \psi)_t \in Ty2$,
  $\mathsf{A}_{Ty2}(\text{ARG1})((\phi = \psi)_t) = \phi$, *and*
  $\mathsf{A}_{Ty2}(\text{ARG2})((\phi = \psi)_t) = \psi$

R_{Ty2} : *is chosen in such a way that it contains the right tuples for each of the relations defined above.*

The interpretation I_{Ty2} given in Definition 3.2 establishes the intuitive interpretation of the objects in the universe, i.e., of natural numbers, semantic types and terms of Ty2, under the signature of $\mathcal{TY}2$. As the grammar $\mathcal{TY}2$ is explicitly written for this particular interpretation, we call the interpretation in Definition 3.2 the *intended model*.

We can illustrate that the interpretation given above interprets the objects in the empirical domain in an intuitive way with a simple example from the set of semantic types. The type $\langle e,t\rangle$ is in the set $Types$. By the definition of S_{Ty2}, this type is assigned the sort label *complex-type*. By the attribute interpretation function A_{Ty2} as given for I_{Ty2} in Definition 3.2, the IN value of the type $\langle e,t\rangle$ is the type e, and the OUT value of this type is the type t. These two atomic types are assigned the species *entity* and *truth* respectively.

To prove the proposition, we must show the following two lemmata. The first lemma, Lemma 3.3, states that the interpretation I_{Ty2} is a model of $\mathcal{TY}2$. Being a model is a necessary condition for being an exhaustive model.

LEMMA 3.3

I_{Ty2} *is a model of the grammar* $\mathcal{TY}2$.

The second lemma specifies the further conditions that are imposed on exhaustive models, i.e., that an exhaustive model of a grammar contains at least one instance of every possible configuration of objects that is a model of the grammar.

LEMMA 3.4

Let $I' = \langle U', S', A', R'\rangle$ *be an interpretation of the signature of the grammar* $\mathcal{TY}2$, *then if* I' *is an exhaustive model of* $\mathcal{TY}2$, *then for each* $u' \in U'$,

$$\text{there is an object } u \in \mathsf{U}_{Ty2} \text{ such that } \langle u', I' \rangle \text{ and } \langle u, \mathsf{I}_{Ty2} \rangle \text{ are congruent.}$$

The proof of Lemma 3.3 proceeds by induction on the recursive structure of the objects in the universe of I_{Ty2}. It suffices to show that each configuration of objects in the interpretation I_{Ty2} satisfies all constraints in the theory. We can prove the lemma by induction, because the universe of I_{Ty2} is recursively defined. The proof of Lemma 3.3 is given in Appendix A.1.1.

With Lemma 3.3 we know that the natural numbers, the semantic types, and the terms of Ty2 can actually be used as the universe of an interpretation of the grammar $\mathcal{TY}2$. In the following, we must show that it is an exhaustive model, i.e., that it contains instances of all configurations that are licensed by the grammar. Note that Lemma 3.3 would hold even if we had assumed the signature of $\mathcal{TY}2$, but an empty theory. Then, however, the interpretation I_{Ty2} as given in Definition 3.2 would not be an exhaustive model. An exhaustive model of the empty theory would, for example also contain cyclic or infinite configurarions under objects of sort *number*. Such configurations are excluded in the grammar $\mathcal{TY}2$ by the NUMBER PRINCIPLE (110). Infinite or cyclic configurations do not correspond to natural numbers. In the previous section, we imposed a number of constraints on the entities in the denotation of the grammar $\mathcal{TY}2$. The validity of Lemma 3.3 shows that that we did not impose too many constraints. With the proof of Lemma 3.4, we will show, that we imposed enough constraints.

To be able to prove Lemma 3.4, we must explain what we understand under *congruence*. We have already introduced this notion together with the notion of *indiscernibility* in Section 1.1. The definition of both notions relies on the notion of a *configuration of objects*. Following Richter 2000, we write $\langle u, I \rangle$ for the configuration of objects under the object u in the interpretation I. In the figures in Section 1.1 we have shown some such configurations.

The notion of indiscernibility is defined in King 1999 (p. 343) for SRL and adapted for RSRL in Richter 2000 (p. 184). Given interpretations I and I', two configurations $\langle u, I \rangle$ and $\langle u', I' \rangle$ are *indiscernible* iff for each description δ, u is described by δ under interpretation I iff u' is described by δ under interpretation I'. While the definition of indiscernibility looks the same for SRL and for RSRL, there is, of course, a difference, as the set of descriptions is different, and the interpretation of RSRL contains more structure. In RSRL, in contrast to SRL, the set of descriptions contains quantified descriptions, relation calls and terms that start with a variable. An RSRL interpretation contains a relation interpretation, which is missing in SRL interpretations.

While indiscernibility is concerned with the description of objects, the notion *congruence* refers to the shape of a configuration. For SRL two configurations $\langle u, I \rangle$ and $\langle u', I' \rangle$ are congruent, iff there is a bijection f from the set of components of u to the set of components of u' such that $f(u) = u'$ and for each component o of u, (i) the species assigned to o under I are the same as the species assigned to $f(o)$ under I', and (ii) exactly the same attributes are defined on the components of o and on $f(o)$, and (iii) for each attribute α that is defined on o, applying f to the α value of o is the same as taking the α value of $f(o)$.

Put differently, two configurations are congruent, if every path that is defined on the matrix object of one configuration is also defined on the matrix object of the other and leads to an object of the same species. In addition, whenever two paths that are defined on the matrix object of one configuration lead to the same object, so do these paths within the other configuration.

As in SRL there are no quasi sorts, quasi attributes and no relations, this definition is enough. For RSRL, Richter 2000 (p. 183) extends King's definition to include the new concepts of RSRL as well. In his extension, Richter requires that for two configurations $\langle u, I\rangle$ and $\langle u', I'\rangle$ to be congruent, the bijection must also map each chain built from the components of u to a corresponding chain built from the components of u'. In addition, Richter adds a condition to ensure that the interpretation of relations is also the same relative to this bijection. This condition says that for each relation symbol ρ with arity i, for each i-tuple $\langle o_1, \ldots o_i\rangle$ of components or chains from u that is in the interpretation of the relation ρ in I, the i-tuple $\langle f(o_1), \ldots, f(o_i)\rangle$ is in the interpretation of ρ in I'.

When we will use the notion of congruence in our proof of Lemma 3.4, we will ignore the two additional conditions imposed on congruent configurations in Richter 2000. We will briefly show that we can safely do this. Let us first consider the case of chains. Assume that there is a bijection g from the components of an object u in an interpretation I to the components of an object u' in an interpretation I' such that g meets King's requirements for congruence. We can extend g to a bijection g' from the set of components and chains of components of u to the set of components and chains of components of u'. The extension is such that

$$
\begin{aligned}
g'(o) &= g(o), \text{ if } o \text{ is a component of } u,\\
g'(o) &= \langle\rangle, \text{ if } o \text{ is the empty chain, and}\\
g'(o) &= \left\langle g(\widehat{\mathsf{A}}(\dagger)(o)) \,\middle|\, g'(\widehat{\mathsf{A}}(\triangleright)(o)) \right\rangle, \text{ if } o \text{ is a non-empty chain.}
\end{aligned}
$$

It is clear that g' meets the three condition on the bijection that is required for congruent configurations: First, for every o which is a component or a chain of u, the (quasi-)species of o is the same as that of $g'(o)$. Second, exactly the same (quasi-)attributes are defined on o and on $g'(o)$. Third, whenever a (quasi-)attribute α is defined on o, then applying g' to the α values of o is the same as the α value of $g'(o)$.

This indicates that once we found a bijection g between components which meets the SRL requirements for congruence, we know that there is also a bijection g' which includes chains and meets the SRL requirements for congruence when applied to chains as well.

Let us next turn to the condition on the relation interpretation. As Richter defines congruence on arbitrary interpretations, it is necessary for him to include in his definition the requirement that the relation interpretations contain corresponding tuples. In Lemma 3.4, we assume that the interpretation I' is an exhaustive model. Therefore, we know that for each u' in the universe of I', the interpretation of some i-ary relation symbol ρ in I' contains every i-tuple of components or chains of components of u' that are allowed by the grammar. This means that the set of components of an object u' in the universe of some exhaustive model I' and the grammar fully determine which i-tuples of components or chains of components of u' are in the interpretation of an i-ary relation in I'.

Similarly, in our definition of the intended model $\mathsf{I}_{\mathrm{Ty2}}$ of $\mathcal{TY}2$ in Definition 3.2 we simply stated that the interpretation of the relations contain all possible tuples. Therefore, for each configuration $\langle u, \mathsf{I}_{\mathrm{Ty2}}\rangle$ in the intended model and for each configuration $\langle u', I'\rangle$ in some exhaustive model of $\mathcal{TY}2$, if there exists a bijection g between the components of u and those of u', then an i-tuple $\langle o_1, \ldots, o_i\rangle$ of components or chains of components of u' is in the interpretation of an i-ary relation ρ in $\mathsf{I}_{\mathrm{Ty2}}$, iff $\langle g'(o_1), \ldots, g'(o_i)\rangle$ is in the interpretation of ρ in I', where g' is the extension of g as given before.

This reasoning on chains and relation interpretation shows that, given an object u of the universe of $\mathsf{I}_{\mathrm{Ty2}}$ and an object u' of the universe of some exhaustive model I' of $\mathcal{TY}2$, if there is a bijection g between the objects in a configuration $\langle u, \mathsf{I}_{\mathrm{Ty2}}\rangle$ of our intended model and the objects in a configuration $\langle u', I'\rangle$ of some exhaustive model such that that g meets the requirements of King's notion of congruence, then there is also a bijection that has all the properties of Richter's notion. Thus, we need not consider those parts of a configuration that are concerned with the interpretation of the relation symbols in the proof of Lemma 3.4.

The proof of Lemma 3.4 goes, again, by induction. This time, the induction is on the structure of configurations of objects in an exhaustive model. The base cases are constituted by objects of atomic sorts, i.e., by configurations that consist of a single object and no attribute arrows. The recursive cases deal with configurations of objects whose matrix object is of a non-atomic sort. Such an induction is possible, because the theory of the grammar ensures that in every model of the grammar and in every finite and nonclyclic configuration in this model, every path will ultimately lead to an object of an atomic sort.

Given this idea, in the proof, we assume an arbitrary exhaustive model I' of $\mathcal{TY}2$. For each object u in the interpretation I', we consider the configuration of objects $\langle u, I' \rangle$. We then show that for each of these configurations, there is an object o in the interpretation $\mathsf{I}_{\mathrm{Ty2}}$ such that the configurations $\langle o, \mathsf{I}_{\mathrm{Ty2}} \rangle$ and $\langle u, I' \rangle$ are congruent. The full proof is given in Appendix A.1.1.

With Lemma 3.3 and Lemma 3.4, Proposition 3.1 follows immediately, if we fill in the definition of an exhaustive model as given in Definition 2.18. There, we have defined an exhaustive model via indiscernibility classes. We proved with Lemma 3.4 that for every configuration of objects in an arbitrary exhaustive model, the intended model has a configuration which is congruent to that configuration. We can, then, use the equivalence of congruence and indiscernibility to show the proposition.

It is a direct consequence of Proposition 3.1 that the grammar $\mathcal{TY}2$ is observationally adequate for the language Ty2, as explained at the end of Section 2.1. It is interesting to see, though, that the grammar $\mathcal{TY}2$ is a constraint-based definition of the language, in contrast to the recursive definition given in Section 1.3.2. For simplicity, consider just the set $Type$ and those parts of $\mathcal{TY}2$ which we defined in Subsection 3.1.2. In Definition 1.5, the set of semantic types is defined recursively, i.e., we are given some instructions of what the basic types are and how we can build complex types from existing types. In an RSRL grammar, the signature determines what the general shape of *type* objects is. The theory, then, is used to impose finer constraints such as the restriction to finiteness and non-cyclicity. We can now see that the set $Type$ can be adequately described by both mechanisms, while the recursive definition is admittedly more straightforward and natural as it automatically includes finiteness and non-cyclicity.

3.3. The Equivalence of $\mathcal{TY}2$ and Ty2

So far, we focussed on a particular exhaustive model. We chose terms of Ty2 to be in the denotation of the grammar $\mathcal{TY}2$, because we can interpret these terms with respect to a *semantic model* as defined in Definition 1.9. In the present section, we show that even if we consider a different exhaustive model, such a semantic interpretation can still be provided. For this purpose, we define a function "$\{\!\![\ ]\!\!\}$" which assigns indiscernibility classes of *me* objects an interpretation, just as the function "$[\![\ ]\!]$" does for terms of Ty2.

In the previous sections, the grammar $\mathcal{TY}2$ was defined and it was shown that the intended model $\mathsf{I}_{\mathrm{Ty2}}$ is an exhaustive model of $\mathcal{TY}2$. Since the universe of the interpretation $\mathsf{I}_{\mathrm{Ty2}}$ is exclusively populated by natural numbers, semantic types and terms of Ty2, it follows from this that all other exhaustive models of $\mathcal{TY}2$ differ from $\mathsf{I}_{\mathrm{Ty2}}$ only with respect to the number of indiscernible copies of the configurations under these objects.

Thus, instead of going via the interpretation of objects in the intended exhaustive model, we can also define an interpretation of equivalence classes of indiscernible objects of an arbitrary exhaustive model. We write $[u]$ for the equivalent class of objects which are indiscernible from u under a given interpretation.

In the following definitions, we assume an arbitrary exhaustive model I. For this interpretation, we assume a frame F as given by the following definition.

DEFINITION 3.5

Let $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$ *be an exhaustive model of* $\mathcal{TY}2$,

let E *be a set of individuals, and*

let W *be a set of possible worlds, then*

for each $u \in \mathsf{U}$,

$$F \text{ is a frame } F = \bigcup_{([u], \mathsf{S}(u) \sqsubseteq type)} D_{E,W,[u]}, \text{ where,}$$

for each $u \in \mathsf{U}$ *such that* $\mathsf{S}(u) = truth$,

$$D_{E,W,[u]} = \{1, 0\},$$

for each $u \in \mathsf{U}$ *such that* $\mathsf{S}(u) = entity$,

$$D_{E,W,[u]} = E,$$

for each $u \in \mathsf{U}$ *such that* $\mathsf{S}(u) = w\text{-}index$,

$$D_{E,W,[u]} = W, \text{ and}$$

for each $u \in \mathsf{U}$ *such that* $\mathsf{S}(u) = c\text{-}type$ *with* $T_{\mathsf{I}}(:\textsc{in})(u) = u'$ *and* $T_{\mathsf{I}}(:\textsc{out})(u) = u''$,

$$D_{E,W,[u]} = D_{E,W,[u'']}^{D_{E,W,[u']}}.$$

The way the frames of Ty2 and of the arbitrary exhaustive model of $\mathcal{TY}2$ are defined, it can be shown that we can use the same frame for both.

LEMMA 3.6 *Let* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$ *be an exhaustive model of* $\mathcal{TY}2$,

let E *be the domain of individuals and*

let W *be a set of possible worlds, then*

for each $\tau \in Type$,

there is an indiscernibility class $[u] \subseteq \mathsf{U}$ *such that*

$$D_{E,W,\tau} = D_{E,W,[u]},$$

and for each $u \in \mathsf{U}$, *with* $\mathsf{S}(u) \sqsubseteq type$,

there is a $\tau \in Type$ *such that*

$$D_{E,W,\tau} = D_{E,W,[u]}.$$

Lemma 3.6 shows, that if we assume identical E and W, the frame of $\mathcal{TY}2$ is identical to that of Ty2, independent of the particular choice of the exhaustive model. The proof of this lemma, given in Appendix A.1.1, goes by induction. The base cases are constituted by the basic types t, e and s, and by the atomic subsorts of *type*, i.e., the species *truth*, *entity* and *w-index*. For these cases, the identity of the frame follows directly from the definitions. For the recursive case, i.e., types of the form $\langle \tau, \tau' \rangle$ and indiscernibility classes of objects of sort *complex-type*, the identity of the frame follows from the definition using the hypothesis.

The major definition that we are aiming for in this section is that of the interpretation, or extension, of objects in the universe of an arbitrary exhaustive model of $\mathcal{TY}2$. We will, however, not address these objects directly, but we will, instead, assign an extension to *indiscernibility classes* of such objects. The indiscernibility class of an object o is the equivalence class containing all objects of the universe that are described by exactly the same descriptions as o is. As there is no way to tell indiscernible objects apart, it is reasonable to assume that we assign an extension only to equivalence classes of objects.

Before we can interpret indiscernibility classes of *me* objects, we must introduce two auxiliary notions: a *model* and a *variable assignment*. Both definitions will be parallel to the corresponding definitions for Ty2 as given in Section 1.3.2.

In the following definition, we state, what it is to be a *semantic model*.

DEFINITION 3.7

Let $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$ *be an exhaustive model of* $\mathcal{TY}2$,

let E *be a set of individuals, and let* W *be a set of possible worlds, then*

> *a semantic model is a pair* $M = \langle F, Int \rangle$ *such that*
> F *is a frame and*
> Int *is a function from the set species below* const *to* F *such that*
> *for each species* $const_i \sqsubseteq const$ *and*
> *for each* $u \in \mathsf{U}$ *with* $\mathsf{S}(u) = const_i$,
> *and for some* $u' \in \mathsf{U}$ *with and* $T_{\mathsf{I}}(:\text{TYPE})(u) = u'$,
> $Int(const_i) \in D_{E,W,[u']}$.

Notice that we call M a *semantic model*. It is not a *model of the grammar* $\mathcal{TY}2$, but a semantic model of the terms in the denotation of the grammar $\mathcal{TY}2$. The function Int is the interpretation of the *const* objects and respects their semantic types; parallel to the use of *int* for the interpretation of constants of Ty2.

Before we can interpret (indiscernibility classes) of *me* objects, we must define what a variable assignment is.

DEFINITION 3.8

Let $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$ *be an exhaustive model of* $\mathcal{TY}2$,

let $VAR = \{[u] \subseteq \mathsf{U} | u \in \mathsf{U} \text{ with } \mathsf{S}(u) = var\}$,

and let F *be a frame, then*

$$ASS = \left\{ A \in F^{VAR} \,\middle|\, \begin{array}{l} \text{for each } [u] \in VAR, \\ A([u]) \in D_{E,W,[T_{\mathsf{I}}(:\,\text{TYPE})(u)]} \end{array} \right\}$$

We call each element of ASS a *variable assignment*. Just as has been the case of the variable assignments for variables of Ty2, as defined in Definition 1.8, the variable assignments respect the type information of the variables. For the elements of ASS this means that if a *var* object u has a TYPE value t, then every element A of ASS is such $A([u]) \in D_{E,W,[t]}$.

We can now define the denotation of (indiscernibility classes of) *me* objects. In Definition 3.9, we define the denotation function "$\{\![\ ]\!\}$" with respect to a semantic model M and a variable assignment $A \in ASS$.

DEFINITION 3.9 *(Extension of me objects)*

Let $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$ *be an exhaustive model of* $\mathcal{TY}2$,

let $M = \langle F, Int \rangle$ *be a semantic model,*

let $A \in ASS$ *be a variable assignment, then*

for each $u \in \mathsf{U}$,
$\{\![u]\!\}^{M,A}$, *the extension of* $[u]$ *in a model* M *under a variable assignment* A, *is defined as follows:*
if $\mathsf{S}(u) \sqsubseteq$ *const*
$\{\![u]\!\}^{M,A} = Int(\mathsf{S}(u))$,
if $\mathsf{S}(u) \sqsubseteq$ *var*
$\{\![u]\!\}^{M,A} = A([u])$,
if $\mathsf{S}(u) \sqsubseteq$ *appl,*
$\{\![u]\!\}^{M,A} = \{\![T_{\mathsf{I}}(:\text{FUNC})(u)]\!\}^{M,A}(\{\![T_{\mathsf{I}}(:\text{ARG})(u)]\!\}^{M,A})$,
if $\mathsf{S}(u) \sqsubseteq$ *abstr such that*
there are $v, t, t' \in \mathsf{U}$ *with*
$t = T_{\mathsf{I}}(:\text{ARG TYPE})(u)$,
$t' = T_{\mathsf{I}}(:\text{VAR TYPE})(u)$, *and*
$v = T_{\mathsf{I}}(:\text{VAR})(u)$,
$\{\![u]\!\}^{M,A} = f \in D_{E,W,[t]}^{D_{E,W,[t']}}$ *such that*
for each $d \in D_{E,W,[t']}$*:*
$f(d) = \{\![T_{\mathsf{I}}(:\text{ARG})(u)]\!\}^{M,A[v/d]}$,
if $\mathsf{S}(u) \sqsubseteq$ *equ*
$\{\![u]\!\}^{M,A} = 1$ *if* $\{\![T_{\mathsf{I}}(:\text{ARG1})(u)]\!\}^{M,A} = \{\![T_{\mathsf{I}}(:\text{ARG2})(u)]\!\}^{M,A}$, *else 0.*

It should be noted that the constant interpretation function actually assigns an interpretation to species rather than to (indiscernibility classes of) objects. The extension function, $\{\![u]\!\}$, then, assigns an extension to the indiscernibility classes of objects of the various species under *const*.

The extension function on *var* objects is given by the variable assignment. Objects of sort *appl* are interpreted as functional application. Those of sort *abstr* are treated just like λ-abstraction, i.e., as forming a complex functor. Note the use of the frame in this definition. Finally, objects of sort *equ* receive the interpretation of equations.

With Definition 3.9, we are able to assign indiscernibility classes of *me* objects an extension. But before we can rely on *me* objects of an arbitrary exhaustive model, we must show that the extension assigned to the *me* object is the same extension as that of the corresponding Ty2 term.

PROPOSITION 3.10 *(Equivalence of Ty2 and exhaustive models of* $\mathcal{TY}2$*)*

Let I *be an arbitrary exhaustive model of* $\mathcal{TY}2$.

Then, for each indiscernibility class of me objects

there is a Ty2 term which is assigned the same extension.

For a comparison of Ty2 with a given grammar $\mathcal{TY}2$, it is necessary that the constants are interpreted in the same way. In Definition 3.11 we define what is needed to guarantee this identity of the constant interpretations.

DEFINITION 3.11 *(corresponding constant interpretations)*

Let F *be a frame, and let* $\mathcal{C}$ *be be a total function from* $Const$ *to* $Type$, *then*

An interpretation function, int, *from the set of constants of Ty2 (*$Const$*) to a frame* F *and an interpretation function,* Int, *from the subsorts of const to* F *are called* corresponding *iff*

there is a bijection C *from the set of species below const to* $Const$ *such that*

for each species const′ with const′$\sqsubseteq$*const,*
$int(C(const')) = Int(const')$*, and*
for each $c \in Const$ *and for some* $\tau \in Type$ *such that* $\mathcal{C}(c) = \tau$,
$c_\tau \in Ty2$, $int(c_\tau) = Int(C^{-1}(c_\tau))$.

Given this notion of correspondence for constants, we formulate a proposition that states that there is a systematic semantic correspondence between the objects in an exhaustive model of $\mathcal{TY}2$ and terms of Ty2: we can define a function that maps each indiscernibility class of objects of the sort *me* to a term such that the extension of the indiscernibility classes of objects and the extension of the term are identical.

In the following definition, we give a function SR which assigns a term ϕ of Ty2 to an equivalence class $[u]$ of *me* objects. We will show later that $[u]$ and ϕ have the same extension.

DEFINITION 3.12 *(SR)*

Let $I = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$ *be an exhaustive model of* $\mathcal{TY}2$.

Let C *be a bijection from the species below const to the set* $Const$ *as in Definition 3.11, then*

we define the function SR *from* $Pow(\mathsf{U})$ *to* $\mathbb{N} \cup Type \cup Ty2$ *such that*

for each $u \in \mathsf{U}$ *with* $\mathsf{S}(u) \sqsubseteq number$
if $\mathsf{S}(u) \sqsubseteq zero$*, then*
$SR([u]) = 0$,
if $\mathsf{S}(u) \sqsubseteq non\text{-}zero$*, then*
for some $u_1 \in \mathsf{U}$ *with* $T_\mathsf{I}(\text{:NUMBER})(u) = u_1$,
$SR([u]) = 1 + SR([u_1])$,
for $u \in \mathsf{U}$ *with* $\mathsf{S}(u) \sqsubseteq type$,
if $\mathsf{S}(u) \sqsubseteq entity$*, then*
$SR([u]) = e$,
if $\mathsf{S}(u) \sqsubseteq truth$*, then*
$SR([u]) = t$,
if $\mathsf{S}(u) \sqsubseteq w\text{-}index$*, then*
$SR([u]) = s$,
if $\mathsf{S}(u) \sqsubseteq complex\text{-}type$*, then*
for some $u_1, u_2 \in \mathsf{U}$*, with* $T_\mathsf{I}(\text{:IN})(u) = u_1$ *and*
$T_\mathsf{I}(\text{:OUT})(u) = u_2$,
$SR([u]) = \langle SR([u_1]), SR([u_2]) \rangle$,
for each $u \in \mathsf{U}$ *such that* $\mathsf{S}(u) \sqsubseteq me$,
for each const′ that is a species below const,
if $\mathsf{S}(u) \sqsubseteq const'$,
then $SR([u]) = C(const')$,
if $\mathsf{S}(u) \sqsubseteq var$*, then*
for some $u_1, u_2 \in \mathsf{U}$*, with* $T_\mathsf{I}(\text{:TYPE})(u) = u_1$ *and*
$T_\mathsf{I}(\text{:NUMBER})(u) = u_2$,
$SR([u]) = v_{SR([u_1]), SR([u_2])}$
if $\mathsf{S}(u) \sqsubseteq appl$*, then*
for some $u_1, u_2, t \in \mathsf{U}$*, with* $T_\mathsf{I}(\text{:FUNC})(u) = u_1$
$T_\mathsf{I}(\text{:ARG})(u) = u_2$*, and* $T_\mathsf{I}(\text{:TYPE})(u) = u_3$
$SR([u]) = (SR([u_1]) SR([u_2]))_{SR([u_3])}$,
if $\mathsf{S}(u) \sqsubseteq abstr$*, then*

$$\begin{array}{l} \quad\text{for some } u_1, u_2, u_3 \in \mathsf{U}, \text{ with } T_{\mathsf{I}}(:\text{VAR})(u) = u_1 \\ \quad T_{\mathsf{I}}(:\text{ARG})(u) = u_2, \text{ and } T_{\mathsf{I}}(:\text{TYPE})(u) = u_3 \\ \qquad SR([u]) = (\lambda SR([u_1]).SR([u_2]))_{SR([u_3])}, \\ \text{if } \mathsf{S}(u) \sqsubseteq equ, \text{ then} \\ \quad\text{for some } u_1, u_2, u_3 \in \mathsf{U}, \text{ with } T_{\mathsf{I}}(:\text{VAR})(u) = u_1 \\ \quad T_{\mathsf{I}}(:\text{ARG})(u) = u_2, \text{ and } T_{\mathsf{I}}(:\text{TYPE})(u) = u_3 \\ \qquad SR([u]) = (SR([u_1]) = SR([u_2]))_{SR([u_3])}. \end{array}$$

With the function SR we map an indiscernibility class of *me* objects to a corresponding term of Ty2. To prove Proposition 3.10 we must show that the *me* objects and their corresponding Ty2 terms have identical extensions. To ensure this, we must chose corresponding variable assignments. We can use the function SR to define a correspondence relation between variable assignments. This is done in Definition 3.13.

DEFINITION 3.13 *(corresponding variable assignments)*

Let $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$ *be an exhaustive model of* $\mathcal{TY}2$*, let* F *be a frame, let* Var *be the set of variables,*

then for each $a \in ass$ *and for each* $A \in ASS$*,*

$$\begin{array}{l} a \text{ and } A \text{ are corresponding iff} \\ \quad\text{for each } u \in \mathsf{U} \text{ with } \mathsf{S}(u) = var, \text{ for some } u', u'' \in \mathsf{U} \text{ with} \\ T_{\mathsf{I}}(:\text{NUMBER})(u) = u' \text{ and } T_{\mathsf{I}}(:\text{NUMBER})(u) = u'', \\ \quad A([u]) = a(v_{SR([u']),SR([u''])}) \end{array}$$

With this definition, we can show that the function SR maps an indiscernibility class $[u]$ of *me* objects to a term of Ty2 ϕ such that $[u]$ and ϕ have identical extensions under corresponding variable assignment. This is expressed in the following lemma.

LEMMA 3.14

Given a frame F*,*

an exhaustive model $I = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$ *of* $\mathcal{TY}2$*,*

and corresponding interpretation functions int *and* Int*, then*

$$\begin{array}{l} \text{the function } SR \text{ is such that} \\ \quad\text{there are variable assignments } a \text{ and } A \text{ of Ty2 and } \mathsf{I} \text{ respectively, such that} \\ \qquad\text{for each } u \in \mathsf{U}, \\ \qquad\quad \mathsf{S}(u) \sqsubseteq me, \text{ iff } \{\![[u]]\!\}^{M,A} = [\![SR([u])]\!]^{M,a} \end{array}$$

We are now ready to consider the proof of Lemma 3.14, which is given in Appendix A.1.1. The proof goes by induction on the structure of configurations of objects. Ror objects of the sorts *const* and *var*, the lemma follows from the correspondence of the interpretation of constants and sorts below *const* and the correspondence of the variable assignments For the other subsorts of *me*, we are in the recursive case of the proof. Here, the parallelism of the extension of *me* objects with that of the corresponding syntactic constructs in Ty2 is used.

Given Lemma 3.14, Proposition 3.10 follows immediately. In fact, it is the existence of the function SR that proves the proposition.

We have now achieved the second important result of this chapter: we are able to abstract away from a concrete (intended) exhaustive model, the language Ty2, to an arbitrary exhaustive model of the grammar $\mathcal{TY}2$. The proposition guarantees that when we consider an arbitrary exhaustive model of $\mathcal{TY}2$, the *me* objects in the universe of this model can be assigned a model theoretic interpretation, i.e., an extension with respect to some model M and some variable assignment A, just as if they were terms of Ty2.

Put differently, Proposition 3.10 states that (indiscernibility classes of) objects in the denotation of the grammar $\mathcal{TY}2$ are *semantically equivalent* to natural numbers, semantic types or terms of Ty2. In the next section, we will show that they are also structurally, i.e., *syntactically*, equivalent.

3.4. Terms as Descriptions

We can define a function that maps each Ty2 term ϕ to a description δ such that δ is a description under the signature of $\mathcal{TY}2$ and in each exhaustive model of $\mathcal{TY}2$, δ denotes exactly that indiscernibility class that corresponds to ϕ. For convenience, we will call a description δ *a* $\mathcal{TY}2$ *description* iff δ is a description under the signature of $\mathcal{TY}2$.

PROPOSITION 3.15 *($\mathcal{TY}2$-describability)*

Let $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$ *be an exhaustive model of* $\mathcal{TY}2$.

For each term ϕ *of Ty2, there is a* $\mathcal{TY}2$ *description* δ *such that*

$$D_I(\delta) = \{u \in \mathsf{U} | SR([u]) = \phi\}$$

To prove this proposition, we first define a function "*" from numbers, types and terms to $\mathcal{TY}2$ descriptions. This function can be seen as the mirror of the function SR: The function SR mapped indiscernibility classes *me* objects to Ty2 terms with identical extension. The function "*" maps Ty2 terms to $\mathcal{TY}2$ descriptions that denote exactly these indiscernibility classes.

First, we show that all elements denoted by the resulting description are indiscernible. Then, we show that the resulting indiscernibility classes are the semantic representations of the original number, type or term.

DEFINITION 3.16 *("*")*

"" is a function from* $\mathbb{N} \cup Type \cup Ty2$ *to the set of* $\mathcal{TY}2$ *descriptions such that,*[6]

$$
\begin{array}{l}
\textit{for each } i \in \mathbb{N}, \\
\quad \textit{if } i = 0, \textit{ then} \\
\qquad i^* = :\sim zero, \\
\quad \textit{if } i = j+1, \textit{ then} \\
\qquad i^* = \left(\begin{array}{l} :\sim non\text{-}zero \\ \textbf{and} \;\; j^*[:\textsc{number}/:] \end{array} \right), \\
\textit{for each } \tau \in Type, \\
\quad \textit{if } \tau = e, \textit{ then} \\
\qquad \tau^* = :\sim entity, \\
\quad \textit{if } \tau = t, \textit{ then} \\
\qquad \tau^* = :\sim truth, \\
\quad \textit{if } \tau = s, \textit{ then} \\
\qquad \tau^* = :\sim w\text{-}index, \\
\quad \textit{if } \tau = \langle \tau_1, \tau_2 \rangle, \textit{ then} \\
\qquad \tau^* = \left(\begin{array}{l} :\sim complex\text{-}type \\ \textbf{and} \;\; \tau_1^*[:\textsc{in}/:] \\ \textbf{and} \;\; \tau_2^*[:\textsc{out}/:] \end{array} \right),
\end{array}
$$

[6] Note that $\delta[\pi/\pi']$ is a description that differs from δ in that, wherever there is an occurrence of the path π' in δ, there is an occurrence of the path π in $\delta[\pi/\pi']$.

$$\begin{array}{l}
\textit{for each } \phi \in \textit{Ty2}, \\
\textit{if } \phi = v_{\tau,i}, \textit{ then} \\
\qquad \phi^* = \left(\begin{array}{l} \textit{:}{\sim}\textit{var} \\ \textbf{and} \;\; i^*[\text{:NUMBER}/\text{:}] \\ \textbf{and} \;\; \tau^*[\text{:TYPE}/\text{:}] \end{array} \right), \\
\textit{if } \phi = c_\tau, \textit{ then} \\
\qquad \phi^* = \left(\begin{array}{l} \textit{:}{\sim}\textit{const}_i \\ \textbf{and} \;\; \tau^*[\text{:TYPE}/\text{:}] \end{array} \right), \\
\textit{if } \phi = (\phi_1\phi_2)_\tau, \textit{ then} \\
\qquad \phi^* = \left(\begin{array}{l} \textit{:}{\sim}\textit{appl} \\ \textbf{and} \;\; \tau^*[\text{:TYPE}/\text{:}] \\ \textbf{and} \;\; \phi_1^*[\text{:FUNC}/\text{:}] \\ \textbf{and} \;\; \phi_2^*[\text{:ARG}/\text{:}] \end{array} \right), \\
\textit{if } \phi = (\lambda x.\phi)_\tau, \textit{ then} \\
\qquad \phi^* = \left(\begin{array}{l} \textit{:}{\sim}\textit{abstr} \\ \textbf{and} \;\; \tau^*[\text{:TYPE}/\text{:}] \\ \textbf{and} \;\; x^*[\text{:VAR}/\text{:}] \\ \textbf{and} \;\; \phi^*[\text{:ARG}/\text{:}] \end{array} \right), \\
\textit{if } \phi = (\phi_1 = \phi_2)_t, \textit{ then} \\
\qquad \phi^* = \left(\begin{array}{l} \textit{:}{\sim}\textit{equ} \\ \textbf{and} \;\; \text{:TYPE}{\sim}\textit{truth} \\ \textbf{and} \;\; \phi_1^*[\text{:ARG1}/\text{:}] \\ \textbf{and} \;\; \phi_2^*[\text{:ARG2}/\text{:}] \end{array} \right).
\end{array}$$

$\mathcal{TY}2$ has an interesting property. The function "*" can be used to identify indiscernibility classes in arbitrary exhaustive models of $\mathcal{TY}2$. This is possible, because (i), all objects are finite and acyclic, and (ii), there is a maximal amount of token identities in each object.

To illustrate the effect of the function "*", we can compute the description $(\mathsf{walk}'_@(x_@))^*$, i.e., the description that corresponds to a subterm of our example term in (107b). In (133) the resulting description is given in AVM form.

(132) a. $(\mathsf{walk}'_@(x_@))^*$
$= (((\mathsf{walk}'_{s(et)} v_{s,0})_{et}(v_{se,0} v_{s,0})_e)_t)^*$

b. $= \left(\begin{array}{l} \text{:}{\sim}\textit{appl} \\ \textbf{and} \;\; t^*[\text{:TYPE}/\text{:}] \\ \textbf{and} \;\; (\mathsf{walk}'_{s(et)} v_{s,0})^*_{et}[\text{:FUNC}/\text{:}] \\ \textbf{and} \;\; ((v_{se,0} v_{s,0})_e)^*[\text{:ARG}/\text{:}] \end{array} \right)$

c. $= \left(\begin{array}{l} \text{:}{\sim}\textit{appl} \\ \textbf{and} \;\; \text{:TYPE}{\sim}\textit{truth} \\ \textbf{and} \;\; \text{:FUNC}{\sim}\textit{appl} \\ \textbf{and} \;\; (et)^*[\text{:FUNC TYPE}/\text{:}] \\ \textbf{and} \;\; \mathsf{walk}'^*_{s(et)}[\text{:FUNC FUNC}/\text{:}] \\ \textbf{and} \;\; v_{s,0}[\text{:FUNC ARG}/\text{:}] \\ \textbf{and} \;\; \text{:ARG}{\sim}\textit{appl} \\ \textbf{and} \;\; e^*[\text{:ARG TYPE}/\text{:}] \\ \textbf{and} \;\; (v_{se,0})^*[\text{:ARG FUNC}/\text{:}] \\ \textbf{and} \;\; (v_{s,0})^*[\text{:ARG ARG}/\text{:}] \end{array} \right)$

$$
\text{d.} = \left(\begin{array}{l}
:\sim \mathit{appl} \\
\textbf{and } :\textsc{type}\sim \mathit{truth} \\
\textbf{and } :\textsc{func type}\sim \mathit{c\text{-}type} \\
\textbf{and } e^*[:\textsc{func type in}/:] \\
\textbf{and } t^*[:\textsc{func type out}/:] \\
\textbf{and } :\textsc{func func}\sim \mathit{const}_{127} \\
\textbf{and } s(et)^*[:\textsc{func func type}/:] \\
\textbf{and } :\textsc{func arg}\sim \mathit{var} \\
\textbf{and } s^*[:\textsc{func arg type}/:] \\
\textbf{and } 0^*[:\textsc{func arg number}/:] \\
\textbf{and } :\textsc{arg}\sim \mathit{appl} \\
\textbf{and } :\textsc{arg type}\sim \mathit{entity} \\
\textbf{and } :\textsc{arg func}\sim \mathit{var} \\
\textbf{and } (se)^*[:\textsc{arg func type}/:] \\
\textbf{and } 0^*[:\textsc{arg func number}/:] \\
\textbf{and } :\textsc{arg arg}\sim \mathit{var} \\
\textbf{and } s^*[:\textsc{arg arg type}/:] \\
\textbf{and } 0^*[:\textsc{arg arg number}/:]
\end{array} \right)
$$

$$
\text{e.} = \left(\begin{array}{l}
:\sim \mathit{appl} \\
\textbf{and } :\textsc{type}\sim \mathit{truth} \\
\textbf{and } :\textsc{func type}\sim \mathit{c\text{-}type} \\
\textbf{and } :\textsc{func type in}\sim \mathit{entity} \\
\textbf{and } :\textsc{func type out}\sim \mathit{truth} \\
\textbf{and } :\textsc{func func}\sim \mathit{const}_{127} \\
\textbf{and } :\textsc{func func type}\sim \mathit{c\text{-}type} \\
\textbf{and } s^*[:\textsc{func func type in}/:] \\
\textbf{and } (et)^*[:\textsc{func func type out}/:] \\
\textbf{and } :\textsc{func arg}\sim \mathit{var} \\
\textbf{and } :\textsc{func arg type}\sim \mathit{w\text{-}index} \\
\textbf{and } :\textsc{func arg number}\sim \mathit{zero} \\
\textbf{and } :\textsc{arg}\sim \mathit{appl} \\
\textbf{and } :\textsc{arg type}\sim \mathit{entity} \\
\textbf{and } :\textsc{arg func}\sim \mathit{var} \\
\textbf{and } :\textsc{arg func type}\sim \mathit{c\text{-}type} \\
\textbf{and } s^*[:\textsc{arg func type in}/:] \\
\textbf{and } e^*[:\textsc{arg func type out}/:] \\
\textbf{and } :\textsc{arg func number}\sim \mathit{zero} \\
\textbf{and } :\textsc{arg arg}\sim \mathit{var} \\
\textbf{and } :\textsc{arg arg type}\sim \mathit{w\text{-}index} \\
\textbf{and } :\textsc{arg arg number}\sim \mathit{zero}
\end{array} \right)
$$

$$\text{f.} = \left(\begin{array}{l}
:\sim \textit{appl} \\
\textbf{and } :\textsc{type}\sim \textit{truth} \\
\textbf{and } :\textsc{func type}\sim \textit{c-type} \\
\textbf{and } :\textsc{func type in}\sim \textit{entity} \\
\textbf{and } :\textsc{func type out}\sim \textit{truth} \\
\textbf{and } :\textsc{func func}\sim \textit{const}_{127} \\
\textbf{and } :\textsc{func func type}\sim \textit{c-type} \\
\textbf{and } :\textsc{func func type in}\sim \textit{w-index} \\
\textbf{and } :\textsc{func func type out}\sim \textit{c-type} \\
\textbf{and } e^{*}[:\textsc{func func type out in}/:] \\
\textbf{and } t^{*}[:\textsc{func func type out out}/:] \\
\textbf{and } :\textsc{func arg}\sim \textit{var} \\
\textbf{and } :\textsc{func arg type}\sim \textit{w-index} \\
\textbf{and } :\textsc{func arg number}\sim \textit{zero} \\
\textbf{and } :\textsc{arg}\sim \textit{appl} \\
\textbf{and } :\textsc{arg type}\sim \textit{entity} \\
\textbf{and } :\textsc{arg func}\sim \textit{var} \\
\textbf{and } :\textsc{arg func type}\sim \textit{c-type} \\
\textbf{and } :\textsc{arg func type in}\sim \textit{w-index} \\
\textbf{and } :\textsc{arg func type out}\sim \textit{entity} \\
\textbf{and } :\textsc{arg func number}\sim \textit{zero} \\
\textbf{and } :\textsc{arg arg}\sim \textit{var} \\
\textbf{and } :\textsc{arg arg type}\sim \textit{w-index} \\
\textbf{and } :\textsc{arg arg number}\sim \textit{zero}
\end{array}\right)$$

$$\text{g.} = \left(\begin{array}{l}
:\sim \textit{appl} \\
\textbf{and } :\textsc{type}\sim \textit{truth} \\
\textbf{and } :\textsc{func type}\sim \textit{c-type} \\
\textbf{and } :\textsc{func type in}\sim \textit{entity} \\
\textbf{and } :\textsc{func type out}\sim \textit{truth} \\
\textbf{and } :\textsc{func func}\sim \textit{const}_{127} \\
\textbf{and } :\textsc{func func type}\sim \textit{c-type} \\
\textbf{and } :\textsc{func func type in}\sim \textit{w-index} \\
\textbf{and } :\textsc{func func type out}\sim \textit{c-type} \\
\textbf{and } :\textsc{func func type out in}\sim \textit{entity} \\
\textbf{and } :\textsc{func func type out out}\sim \textit{truth} \\
\textbf{and } :\textsc{func arg}\sim \textit{var} \\
\textbf{and } :\textsc{func arg type}\sim \textit{w-index} \\
\textbf{and } :\textsc{func arg number}\sim \textit{zero} \\
\textbf{and } :\textsc{arg}\sim \textit{appl} \\
\textbf{and } :\textsc{arg type}\sim \textit{entity} \\
\textbf{and } :\textsc{arg func}\sim \textit{var} \\
\textbf{and } :\textsc{arg func type}\sim \textit{c-type} \\
\textbf{and } :\textsc{arg func type in}\sim \textit{w-index} \\
\textbf{and } :\textsc{arg func type out}\sim \textit{entity} \\
\textbf{and } :\textsc{arg func number}\sim \textit{zero} \\
\textbf{and } :\textsc{arg arg}\sim \textit{var} \\
\textbf{and } :\textsc{arg arg type}\sim \textit{w-index} \\
\textbf{and } :\textsc{arg arg number}\sim \textit{zero}
\end{array}\right)$$

In (132), the stepwise application of the function "*" is given. Applying the function to the term $\mathsf{walk}'_{@}(x_{@})$, we first get a description that specifies that the described object is of sort *application* and that we must apply the function "*" to the type of the term (t), and to the two subterms ($\mathsf{walk}'_@$, and $x_@$), where we must substitute the symbol ": " in the resulting descriptions by the paths :TYPE, :FUNC and :ARG respectively.

Applying the function "*" to the type t results in the description : $\sim$*truth*. If we replace the symbol ": " in this description by the path :TYPE, we get the description :TYPE$\sim$*truth*. This description appears as a conjunct in the description in (132c).

Similarly, applying the function "*" to the subterm $\mathsf{walk}'_@$, results in a description that specifies that the described object is of sort *appl*, i.e., we add the line :FUNC TYPE$\sim$*appl*, and we must further apply the function "*" to the semantic type et, the constant $\mathsf{walk}'_{s(et)}$, and the variable $v_{s,0}$.

In (132) all the derivation steps are shown. The resulting description is given in (132g). In (133) we repeat it in the more convenient AVM form.

(133) $$(\mathsf{walk}'_@(x_@))^* = \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{appl}} \\ \text{TYPE} & \textit{truth} \\ \text{FUNC} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{appl}} \\ \text{TYPE} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{c-type}} \\ \text{IN} & \textit{entity} \\ \text{OUT} & \textit{truth} \end{array}\right] \\ \text{FUNC} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{const}_{127}} \\ \text{TYPE} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{c-type}} \\ \text{IN} & \textit{w-index} \\ \text{OUT} & \left[\begin{array}{ll} \text{IN} & \textit{entity} \\ \text{OUT} & \textit{truth} \end{array}\right] \end{array}\right] \end{array}\right] \\ \text{ARG} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{var}} \\ \text{TYPE} & \textit{w-index} \\ \text{NUMBER} & \textit{zero} \end{array}\right] \end{array}\right] \\ \text{ARG} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{appl}} \\ \text{TYPE} & \textit{entity} \\ \text{FUNC} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{var}} \\ \text{TYPE} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{c-type}} \\ \text{IN} & \textit{w-index} \\ \text{OUT} & \textit{entity} \end{array}\right] \\ \text{NUMBER} & \textit{zero} \end{array}\right] \\ \text{ARG} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{var}} \\ \text{TYPE} & \textit{w-index} \\ \text{NUMBER} & \textit{zero} \end{array}\right] \end{array}\right] \end{array}\right]$$

The function "*" is of enormous practical use for us because it allows us to use terms of Ty2 as descriptions of *me* objects. I.e., instead of the description in (132g), we can simply write $(\mathsf{walk}'_@(x_@))^*$.

When comparing the description $(\mathsf{walk}'_@(x_@))^*$ in its AVM form in (133) with the description of this *me* object as it appears as the ARG value in the AVM in (126), it is striking that the identities are missing from (133). To consider just one case, there are three paths that lead to an object of sort *zero*: :FUNC ARG NUMBER, :ARG FUNC NUMBER and :ARG ARG NUMBER. Still, the description that results from the application of the function "*" does not enforce any identities between those objects of the same sort.

When we use the function "*", we will always describe objects in a model of the grammar $\mathcal{TY}2$. The grammar $\mathcal{TY}2$ contains the TIP (given in (125)). The TIP enforces all possible identities to be realized within a *me* object. Therefore, we know that any *me* object in a model of the grammar $\mathcal{TY}2$ that is described by the AVM in (133) will be such that the paths mentioned above lead to the same object.

In fact, we can show that the function "$*$" interacts with the the grammar $\mathcal{TY}2$ in such a way, that applying the function to some term of Ty2, results in a description which only describes objects in the same indiscernibility class. This is expressed in the next lemma.

LEMMA 3.17

For each exhaustive model $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$ *of* $\mathcal{TY}2$*, the function "$*$" is such that for each element* $i \in (\mathbb{N} \cup Type \cup Ty2)$*, and for each* $u_1, u_2 \in \mathsf{U}$*,*

if $u_1, u_2 \in D_{\mathsf{I}}(i^*)$*,*
then $\langle u_1, \mathsf{I} \rangle$ *and* $\langle u_2, \mathsf{I} \rangle$ *are congruent.*

The proof of Lemma 3.17 is given in Appendix A.1.1. It proceeds by induction on the natural numbers, the semantic types and the terms of Ty2. In each case, we will show that the description given by the function"“$*$" is precise enough to single out a class of objects which are the matrix objects of congruent configurations. For this proof to work, it is important that we use objects in a model of $\mathcal{TY}2$. Thus, we know that the objects in the universe of the considered interpretation satisfy the principles of the grammar $\mathcal{TY}2$. In particular, for complex terms, the TERM IDENTITY PRINCIPLE plays a crucial role. As can be seen in the definition of the function "$*$", the resulting descriptions do not include any path identities. Such identities are, however, enforced by the TIP and, thus, need not be part of the resulting descriptions.

With the proof of Lemma 3.17, we have shown that given some x which is a natural number, a semantic type or a term of Ty2, the description x^* denotes exactly one indiscernibility class in the denotation of the grammar $\mathcal{TY}2$. This is an important result for the proof of Proposition 3.15. In fact, all that remains to be shown is that the indiscernibility class denoted by x^* is mapped back to x by the function SR as given in Definition 3.12.

Given the validity of Lemma 3.17, we can now prove Proposition 3.15. Since we know by the lemma that the function "$*$" singles out indiscernibility classes, it is enough to show that they actually single out the right indiscernibility classes, i.e., that for each i which is a natural number, a semantic type of a term of Ty2, there is some u in the universe such that i^* identifies the class $[u]$ iff $SR([u]) = i$. To ensure that the necessary objects in the universe of the grammar exist, we consider only exhaustive models.

With the proof of this proposition, which is given in Appendix A.1.1, we have established the third major result of this chapter: for each x which is a natural number, a semantic type or a term of Ty2, the function "$*$" gives us a description that singles out the indiscernibility class, $[u]$, of objects in the denotation of the grammar $\mathcal{TY}2$ such that the function SR maps $[u]$ back to x. As both functions, "$*$" and SR are syntactic, we can say that x and the elements in $[u]$ are *syntactically equivalent*.

In Section 3.3 we showed that there is also a *semantic equivalence*: We can assign indiscernibility classes of objects in the denotation of $\mathcal{TY}2$ an extension with respect to some semantic model. Furthermore, we showed that given an indiscernibility class of *me* objects, $[u]$, the extension of $[u]$ is the same as the extension of $SR([u])$.

Both kinds of equivalences were shown to hold for arbitrary exhaustive models of $\mathcal{TY}2$. Therefore, we can use the grammar $\mathcal{TY}2$ independent from the particular choice of the exhaustive model.

If we are in the special case of considering our "intended" model $\mathsf{I}_{\mathrm{Ty2}}$ of Definition 3.2, the equivalence reduces to identity. This is expressed in the following corrolary:

COROLLARY 3.18 *In the interpretation* I_{Ty2} *as given in Definition 3.2,*

$$\begin{array}{l} \textit{for each } x \in (\mathbb{N} \cup Type \cup Ty2), \\ \quad SR([x]) = x, \textit{ and} \\ \quad D_I(x^*) = \{x\} \end{array}$$

This corollary states that if we use the intended model, the function SR maps each object in the universe to itself. Furthermore, for each object in the domain, the function "*" yields a description that describes exactly this object.

3.5. Extending and Integrating $\mathcal{TY}2$

In this final section of the present chapter we will first summarize the results of the previous sections. Then, we show how the grammar $\mathcal{TY}2$ can be extended to account for an extended semantic representation language. For this purpose, we show what additions to the grammar $\mathcal{TY}2$ are needed to incorporate (i) logical constants ($\neg$, $\wedge$, $\vee$, and $\rightarrow$), (ii) the classical quantifiers ($\exists$ and $\forall$), and (iii) generalized quantifiers such as the quantifier most introduced in (38). Finally, we will add the grammar $\mathcal{TY}2$ to the grammar of our syntactic fragment as given in Section 2.3.

In this chapter, we provided an RSRL grammar, $\mathcal{TY}2$, and showed that this grammar denotes the semantic representation language Ty2. In particular, we could prove that the semantic representation language is an exhaustive model of the grammar $\mathcal{TY}2$. We referred to this particular model as the *intended model*. The major result of Section 3.3 is that we are not forced to use the intended model. Instead, we can take any exhaustive model of $\mathcal{TY}2$ and assign an interpretation (extension) to the objects in that model.

Furthermore, not only can we use objects in an arbitrary exhaustive model for semantic interpretation, we also showed in Section 3.4 that these objects can be treated syntactically as if they were objects of the intended model. This point is of particular importance, because it allows us to use objects of the sort *me* as *logical forms* of linguistic signs. For this use, we are mainly interested in the syntax of the terms of the semantic representation language and not so much in their semantics.

As a practical result of Section 3.4, the definition of the function "*" enables us to use natural numbers, semantic types and terms of the semantic representation language as shorthand for lengthy descriptions of complex entities.

Before we turn to the extension of Ty2, and to an integration of the grammar $\mathcal{TY}2$ into a larger grammar, we want to point once more to the intimate relationship that holds between the empirical domain considered in this chapter, i.e., the language Ty2, and the grammar that is used to describe this domain, i.e., the grammar $\mathcal{TY}2$.

It is worth noting that all species and attributes introduced by the signature of $\mathcal{TY}2$ are motivated by the structure of the empirical domain, the semantic representation language. The only additional thing introduced in $\mathcal{TY}2$ are relations. In addition, differences in configurations of objects in the denotation of $\mathcal{TY}2$ always express differences that are actually present in the empirical domain. We addressed this property in our discussion of the Type Identity Principle (TyIP) in (114). There we saw that for semantic types, there is no advantage of assuming that the type $e(et)$ appears in two distinct forms; one where both occurrences of the type e are the same type, and one where the occurrences express different types. The TyIP, or alternatively the more general GIP, ensures that all configurations of objects that correspond to the type $e(et)$ are congruent.

While it was relatively easy to establish this intimate relationship for the well-defined empirical domain of the semantic representation language Ty2, we think that it should be at least a methodological principle for RSRL grammars in general to try to introduce only those attributes and sorts into the signature that are empirically motivated. We will see in our formalization of *Lexicalized Flexible Ty2* in Chapter 4 that the syntactic means of RSRL allow us to adopt this methodological principle.

After these summarizing remarks, we can turn to extensions of the grammar $\mathcal{TY}2$ which will be presupposed in the rest of this thesis. In the following, we will discuss two such extensions: the introduction of logical constants and classical quantifiers, and the introduction of generalized quantifiers.

In Section 3.1 we presented a sort hierarchy below the sort *me* in (121) which did not contain logical constants and quantifiers. We did not introduce these syntactic constructs, because the corresponding constructs were missing from the definition of the terms of Ty2 in Definition 1.5. Instead, there, we defined them as abbreviations of terms that only use application, abstraction and equation (see (33)).

If we want to use a representation language which provides an explicit encoding of quantification, negation, disjunction etc, we must extend the definition of the syntax and semantics of Ty2 terms and extend the sort hierarchy below the sort *me* to include logical constants and quantifiers as well. In (134a) we add the necessary clauses to the syntax of Ty2; in (134b) the corresponding additional clauses to the semantics of Ty2 are given.

(134) a. Additional clauses to Definition 1.5:
for each $\phi_t, \psi_t \in$ Ty2,
$(\neg\phi_t)_t \in$ Ty2,
$(\phi_t \wedge \psi_t)_t \in$ Ty2,
$(\phi_t \vee \psi_t)_t \in$ Ty2,
$(\phi_t \rightarrow \psi_t)_t \in$ Ty2,
for each $v_{\tau,i} \in Var$, and for each $\phi_t \in$ Ty2,
$(\exists v_{\tau,i}.\phi_t)_t \in$ Ty2, and
$(\forall v_{\tau,i}.\phi_t)_t \in$ Ty2.

b. Additional clauses to Definition 1.9:
for each $\phi_\tau, \psi_\tau \in$ Ty2,
$[\![(\neg\phi_t)_t]\!]^{M,a} = 1$ if $[\![\phi_t]\!]^{M,a} = 0$, else 0.
$[\![(\phi_t \wedge \psi_t)_t]\!]^{M,a} = 1$ if $[\![\phi_t]\!]^{M,a} = 1$ and $[\![\psi_t]\!]^{M,a} = 1$, else 0.
$[\![(\phi_t \vee \psi_t)_t]\!]^{M,a} = 1$ if $[\![\phi_t]\!]^{M,a} = 1$ or $[\![\psi_t]\!]^{M,a} = 1$, else 0.
$[\![(\phi_t \rightarrow \psi_t)_t]\!]^{M,a} = 1$ if $[\![\phi_t]\!]^{M,a} = 0$ or $[\![\psi_t]\!]^{M,a} = 1$, else 0.
$[\![(\exists v_{\tau,i}.\phi_t)_t]\!]^{M,a} = 1$ if there exists a $d \in D =_{E,W,\tau}$ such that $[\![\phi_t]\!]^{M,a[v_{\tau,i}/d]} = 1$, else 0.
$[\![(\forall v_{\tau,i}.\phi_t)_t]\!]^{M,a} = 1$ if for each $d \in D_{E,W,=\tau}$ such that $[\![\phi_t]\!]^{M,a[v_{\tau,i}/d]} = 1$, else 0.

Corresponding to these extensions of the description language, we can also extend the ontology of the grammar $\mathcal{TY}2$. In (135) we give those aspects of the signature of $\mathcal{TY}2$ that must be added. In particular, we introduce new sorts below the sort *me*: *negation* for negation, *logical-constant* for the binary logical constants with a separate subsort for each of these constants, and, finally the sort *quantifier*, which has a subsort for each of the quantifiers. Note that we also introduce a new attribute, SCOPE, which is only appropriate for the sort *quantifier* and its subsorts.

(135) Extensions to the sort hierarchy below *me*

:

me TYPE *type*

neg(ation) ARG *me*

l(ogical)-const(ant) AGR1 *me*
ARG2 *me*

dis(junction)

con(junction)

imp(lication)

quant(ifiers) VAR *variable*
SCOPE *me*

uni(versal)

exi(stential)

In Section 3.1.3 we added a constraint on the TYPE value of all *me* objects except for variables. Clearly, such constraints must also be imposed on the newly introduced sorts.

(136) Extensions to the type restrictions in (122):

a. $neg \Rightarrow \begin{bmatrix} \text{TYPE } truth \\ \text{ARG TYPE } truth \end{bmatrix}$

b. $l\text{-}const \Rightarrow \begin{bmatrix} \text{TYPE } truth \\ \text{ARG1 TYPE } truth \\ \text{ARG2 TYPE } truth \end{bmatrix}$

c. $quant \Rightarrow \begin{bmatrix} \text{TYPE } truth \\ \text{SCOPE TYPE } truth \end{bmatrix}$

The additional principles in (136) ensure that the argument of a *negation* object is a *me* object of the semantic type t, i.e., an object with a TYPE value of sort *truth*. Analogously, both terms that are combined by a binary logical constant must be of sort t, which is expressed in (136b) by the requirement that the ARG1 TYPE and the ARG2 TYPE values of an *l-const* object be of sort *truth*. Finally, the constraint in (136c) enforces that the SCOPE value of a *quant* object be an *me* object with a TYPE value of sort *truth* as well.

If we assume the streamlined version of the grammar $\mathcal{TY}2$ as presented in Section 3.1.4, nothing more must be added to the grammar to include the new syntactic constructs. In case we stick to the TIP, the TNP and the TFP as given in Section 3.1.3, we must add additional clauses to the relation `same-term` (defined in (124b)) for the new subsorts of *me*, and we must change the definition of the relation `component` as given in (117) to take the new attribute SCOPE into account as well.

Now that we have extended the language Ty2 and the grammar $\mathcal{TY}2$ to contain logical constants and quantifiers, we can also show that the correspondence that holds between the original definitions is equally valid for the extended definitions. In Appendix A.1.2 these extended definitions are listed. For illustration, let us consider the term in (137).

(137) $\neg\exists x[\mathsf{walk}'_{@}(x_{@})]$

In Definition 3.2, we defined the intended model of the grammar $\mathcal{TY}2$. As we have enriched the ontology, we must account for this in the specifications of the interpretation. The extended definition of the intended model as given in Appendix A.1.2 specifies that the term in (137) is assigned the species *neg* by the function S. The attribute interpretation function A is specified in such a way that the value of A at the attribute ARG is defined on this term and is the term $\exists x[\mathsf{walk}'_{@}(x_{@})]$. Similarly, this term is assigned the species *existential* by the function S, the VAR value of this term is the variable x, and the SCOPE value of this term is the term $\mathsf{walk}'_{@}(x_{@})$.

The next definition that needs to be extended is that of the function $\{\![\ ]\!\}$, i.e., Definition 3.9. For illustration, we will, again, restrict ourselves to the case of negation and existential quantification, in Appendix A.1.2, number (550), all the required definitions are given. The function $\{\![\ ]\!\}$ is extended so that for each semantic model M, and for each variable assignment A, if u is an object in the universe of an exhaustive model of $\mathcal{TY}2$, such that u is of the sort *neg*, then $\{\![u]\!\}^{M,A} = 1$ if $\{\![T_{\mathsf{I}}(:\text{ ARG})(\mathrm{u})]\!\}^{M,A} = 0$, and 0 else. Similarly, if u is of sort *exist* and v is the VAR value of u, then $\{\![u]\!\}^{M,A} = 1$, if there exists a d in the frame which is in the domain specified by the semantic type of the VAR value of u, such that $\{\![T_{\mathsf{I}}(:\text{ SCOPE})(\mathrm{u})]\!\}^{M,A[[v]/d]} = 1$, and 0 else.

It follows from the definition of the function "$\{\![\ ]\!\}$" that if we use the same frame F, corresponding constant interpretations *int* and Int, and corresponding variable assignments a and A, then the term in (137) has the same extension independent of whether we interpret it by the function $[\![\ ]\!]^{<F,int>,a}$ or by the function $\{\![\ ]\!\}^{<F,Int>,A}$.

The last two definitions that must be considered are those of the function SR (Definition 3.12) and of the function "*" (Definition 3.16). The function SR maps an indiscernibility class of *me* objects to a term of Ty2. In the extended representation language, we require that for an object u of sort *negation*, the indiscernibility class of u be mapped to a term $(\neg\phi)_t$, where ϕ is the term that the indiscernibility class of u's ARG value is mapped to. Similarly, for an object u of sort *existential*, the indiscernibility class of u is mapped by SR to a term $(\exists x.\phi)_t$, where x is the variable that the indiscernibility class of u's VAR value is mapped to, and ϕ is the term that that the indiscernibility class of u's SCOPE value is mapped to. The precise formulation of the extended definition of the function SR is given in (551) in Appendix A.1.2.

Finally, we must consider the function "*", which maps a term of Ty2 to a description. In (132), we have already shown to which description the term $\mathsf{walk}'_{@}(x_{@})$ is mapped. In (138), we illustrate the extended definition of this function with the term in (137).

(138) a. $(\neg\exists x[\mathsf{walk}'_{@}(x_{@})])^{*}$

b. $= \begin{pmatrix} :\sim negation \\ \textbf{and}\ :\text{TYPE}\sim truth \\ \textbf{and}\ (\exists x[\mathsf{walk}'_{@}(x_{@})])^{*}[:\text{ARG}/:] \end{pmatrix}$

c. $= \begin{pmatrix} :\sim negation \\ \textbf{and}\ :\text{TYPE}\sim truth \\ \textbf{and}\ :\text{ARG}\sim existential \\ \textbf{and}\ :\text{ARG TYPE}\sim truth \\ \textbf{and}\ x^{*}[:\text{ARG VAR}/:] \\ \textbf{and}\ (\mathsf{walk}'_{@}(x_{@}))^{*}[:\text{ARG SCOPE}/:] \end{pmatrix}$

In (138b) we show that the function "*", when applied to a negated term $(\neg\phi)_t$, yields a description of a *negation* object whose TYPE value is of sort *truth* and whose ARG value is described by applying the function "*" to the term ϕ. Similarly, in (138c), we show that applying the function "*" to a term of the form $(\exists x.\phi)_t$ results in a description of a *existential* object whose TYPE value is of sort *truth*, whose VAR value is described by applying "*" to the variable x and whose SCOPE value is described by applying "*" to the term ϕ.

With this brief illustration of the function "*", we close our exposition of the first extension of the semantic representation language and the grammar $\mathcal{TY}2$. In the following chapters, we will use this extended representation language.

In Section 1.3.2 we have noted that we might want to use a special syntax for generalized quantifiers such as most. Just as was the case with the existential quantifier and the universal quantifier, quantifiers such as most can be defined within Ty2 as given in Definition 1.5 and Definition 1.9. In (38) we provided such a definition for the case of most. There, we extended the set of Ty2 descriptions to include terms of the form $[\mathsf{most}\, x : \phi_t](\psi_t)_t$, where most is a generalized quantifier, x is a variable and ϕ, ψ are terms of Ty2.

It is easy to make a similar extension to the grammar $\mathcal{TY}2$. We will briefly show what needs to be done. First we extend the signature. The sort hierarchy should include a sort *generalized-quantifier* (*gen-quant*) which is a subsort of *quantifier*. Below the sort *gen-quant* we assume one species for each generalized quantifier that is needed in the semantic representation language. For our example, it is enough to assume a single such species, *gen-quant-most*. We also add a new attribute to the signature, RESTR(ICTION). This attribute is declared appropriate for the sort *gen-quant*.

In (139) we state all the attributes that are appropriate for the sort *gen-quant*.

(139) Appropriateness conditions for the sort *gen-quant*:

gen-quant	TYPE	*type*
	VAR	*var*
	RESTR	*me*
	SCOPE	*me*

The attribute TYPE is appropriate for the sort *gen-quant*, because this sort is a subsort of *me*. By virtue of being a subsort of the sort *quant*, the attributes VAR and SCOPE also appear on the sort *gen-quant*. Finally, the new attribute RESTR is declared appropriate for this sort.

In addition to the modifications in the signature, we must also add a principle to the theory. In the definition of terms that contain a generalized quantifier in (38), we require both the restriction and the scope of the quantifier to be of type t. As far as the SCOPE value is concerned, this is guaranteed by the principles collected in the TRP as given in (136c). In order to ensure the same semantic type for the RESTR value, the TRP are extended by one more principle, given in (140).

(140) *gen-quant* $\Rightarrow \left[\text{RESTR TYPE } \textit{truth}\right]$

Nothing more is necessary to include the generalized quantifier most. In order to maintain the equivalence between $\mathcal{TY}2$ and the version of Ty2 used in this thesis, we add a clause for the generalized quantifier most to the syntax and semantics of Ty2, just as done for the classical connectives and quantifiers in (134). Furthermore, we add a clause for the generalized quantifier most to the definitions of the denotation function $\{\![\;]\!\}$ (Definition 3.9), of the function SR (Definition 3.12), and of the function "*" (Definition 3.16). All these extensions are straightforward and are collected in (141).

(141) a. Additional clause to Definition 1.5:

For each $x \in Var$, for each $\phi_t, \psi_t \in$ Ty2,
$[\mathsf{most}\, x : \phi_t](\psi_t)_t \in$ Ty2

b. Additional clause to Definition 1.9:

For each $x \in Var$, for each $\phi_t, \psi_t \in$ Ty2,
$[\![[\mathsf{most}\, x : \phi_t](\psi_t)_t]\!]^{M,a} = 1$
if the cardinality of
$[\![\lambda x.\phi]\!]^{M,a} \cap [\![\lambda x.\psi]\!]^{M,a}$
is greater than the cardinality of
$[\![\lambda x.\phi]\!]^{M,a} \setminus [\![\lambda x.\psi]\!]^{M,a}$,
else 0.

c. Additional clause to Definition 3.9:

for each $\mathsf{S}(u) = \textit{gen-quant-most}$,

$\{\![[u]]\!\}^{M,A} = 1$ if the cardinality of

$\{x \in D_{E,W,T(:\text{VAR TYPE})(u)} | \{\![[T(:\text{SCOPE})(u)]]\!\}^{M,A} = 1\}$

is greater than the cardinality of

$\{x \in D_{E,W,[T(:\text{VAR TYPE})(u)]} | \{\![[T(:\text{RESTR})(u)]]\!\}^{M,A} = 1\}$
$\backslash\{x \in D_{E,W,[T(:\text{VAR TYPE})(u)]} | \{\![[T(:\text{SCOPE})(u)]]\!\}^{M,A} = 1\}$

else 0.

d. Additional clause to Definition 3.12:

if $\mathsf{S}(u) = \textit{gen-quant-most}$, then,

for some $u_1, u_2, u_3 \in \mathsf{U}$ with $T_{\mathsf{I}}(:\text{VAR})(u) = u_1$,

$T_{\mathsf{I}}(:\text{SCOPE})(u) = u_2$, and $T_{\mathsf{I}}(:\text{RESTR})(u) = u_3$,

$SR([u]) = [\mathsf{most}\, SR([u_1]) : SR([u_2])](SR([u_3]))_t$

e. Additional clause to Definition 3.16:

if $\phi = [\mathsf{most}\, x : \phi](\psi)_t$, then

$$\phi^* = \begin{pmatrix} :\sim \textit{gen-quant-most} \\ \textbf{and } :\text{TYPE}\sim \textit{truth} \\ \textbf{and } x^*[:\text{VAR}/:] \\ \textbf{and } \phi^*[:\text{RESTR}/:] \\ \textbf{and } \psi^*[:\text{SCOPE}/:] \end{pmatrix}.$$

In (141c), we have added a clause to the definition of the function $\{\![\;]\!\}$ that ensures that the extension of a *gen-quant-most* object is the same as the extension of a Ty2 description that consists of a generalized quantifier together with its restriction and its scope.

The addition in (141d) shows how we have to extend the function SR. As an effect of this definition, indiscernibility classes of *gen-quant-most* objects are mapped to Ty2 descriptions of the form $[\mathsf{most}\, x : \phi](\psi)_t$.

Finally, in (141e), we add a clause to the definition of the function "*" that has the effect of mapping a description of the form $[\mathsf{most}\, x : \phi](\psi)_t$ to a *gen-quant-most* object whose TYPE value is of sort *truth*, and whose VAR, RESTR, and SCOPE values are described by x^*, ϕ^*, and ψ^*, respectively.

With the example of the generalized quantifier most, we have indicated how an extension of the semantic representation language can be accompanied by an extension of the grammar $\mathcal{TY}2$. For the new grammar, the results of the preceding sections are equally valid.

So far, we were concerned with extending the grammar $\mathcal{TY}2$. In the following chapters, we will address a different question: the integration of the grammar $\mathcal{TY}2$ into a grammar of a natural language such as the grammar for a fragment of English given in Section 2.3. In particular, we will declare objects of sort *me* appropriate for the attribute CONTENT.

In the process of integration, we do not want to lose any of the results of this chapter. Our task is to integrate the grammar $\mathcal{TY}2$ into the grammar of Section 2.3. To achieve this, we assume (i) that the sort *top-ty2* is the top sort in the sort hierarchy of $\mathcal{TY}2$, and (ii) that every principle δ in the theory of $\mathcal{TY}2$ re-appears in the integrated grammar in the form *top-ty2* $\Rightarrow \delta$.

What remains to be done is to provide an interface between the grammar of Section 2.3 and the subgrammar $\mathcal{TY}2$. This interface consists of the appropriateness conditions of the sort *local*. In (73a) we introduced the attributes CATEGORY and CONTENT as appropriate for the sort *local*. In Section 2.3 we gave all relevant constraints on the interaction of the CATEGORY value. For the CONTENT value, however, we preliminarily assumed values of an atomic sort *content*. In (142) we change the appropriateness conditions of the sort *local* so that the sort *me* is appropriate to the attribute CONTENT.

(142) Appropriateness conditions for the sort *local*:

local	CATEGORY	*category*
	CONTENT	*me*

Given this change in the appropriateness conditions, we can include descriptions of the CONTENT value of a word in its lexical entry. For illustration, let us extend the sketch of the lexical entry for the word *every* from (77c). The revised lexical entry is given in (143).

(143) Sketch of the lexical entry of the word *every* (including the CONTENT specification):

$$\begin{bmatrix} \textit{word} \\ \textsc{phon}\ \langle \textit{every} \rangle \\ \textsc{syns} \begin{bmatrix} \textsc{loc} \begin{bmatrix} \textsc{cat} \begin{bmatrix} \textsc{head} \begin{bmatrix} \textit{determiner} \\ \textsc{spec} \begin{bmatrix} \textit{synsem} \\ \textsc{loc cat} \begin{bmatrix} \textsc{head}\ \textit{noun} \\ \textsc{sbc} \left\langle \begin{bmatrix} \textit{synsem} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \textsc{subcat}\ \textit{elist} \\ \textsc{marking}\ \textit{unmarked} \end{bmatrix} \\ \textsc{content}\ (\lambda P_{s((se)t)} \lambda Q_{s((se)t)}.\forall x_{se}[P_{@}(x) \rightarrow Q_{@}(x)])^{*} \end{bmatrix} \\ \textsc{nonl} \begin{bmatrix} \textsc{inher} \begin{bmatrix} \textsc{slash}\ \textit{eset} \end{bmatrix} \\ \textsc{to-bind} \begin{bmatrix} \textsc{slash}\ \textit{eset} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \textsc{store}\ \textit{elist} \end{bmatrix}$$

It should be noted that we use the function "*" to describe the CONTENT value of the word. The term to which we apply this function is exactly the term that was given as the basic translation of the word *every* in Table 1.1 (page 44).

In the following, we will normally not indicate the use of the function "*" explicitly when it occurs within a larger description. This means that if we write a term of Ty2, ϕ, inside an RSRL description (usually in AVM notation) this is to be understood as the RSRL description $(\phi)^*$.

Analogously to the lexical entry of the word *every* as given in (143), we can extend all the lexical entries given in (77) by specifying their CONTENT values as being described by the application of the function "*" to the basic translation of the words given in Table 1.1.

In our syntactic fragment of Section 2.3 we included some words for which there was no basic translation given in that table. The relevant translations are given in (144).

(144) Some additional basic translations:

book	$\rightsquigarrow \lambda x_{se}.\mathsf{book}'_{s(et)}(@)(x_{@})$
good	$\rightsquigarrow \lambda P_{s((se)t)}\lambda x_{se}.[P_{@}(x) \wedge \mathsf{good}'_{s(et)}(@)(x_{@})]$
reads	$\rightsquigarrow \lambda y_{se}\lambda x_{se}.\mathsf{read}'_{s(e(et))}(@)(x_{@}, y_{@})$
John	$\rightsquigarrow j_{e}$

With the indicated integration of the grammar $\mathcal{TY}2$ into the grammar of Section 2.3, we have provided an RSRL formalization of the basic translations as needed for the semantic framework LF-Ty2. In Section 1.3 we assumed that the logical form of a phrase is the result of functional application of the logical forms of the daughters. In the following chapter, we will formalize this as the SEMANTICS PRINCIPLE. The main mechanism of LF-Ty2, however, was shown to be its flexible type shifting potential. We will give a formalization of the two type-shifting rules, *argument raising* and *value raising* as defined above. Finally, we will also show how the two additional constructions treated in the syntactic fragment, passive and complement extraction, are accounted for within LF-Ty2.

CHAPTER 4

Lexicalized Flexible Ty2 in HPSG

In the preceding chapter we laid the formal ground for an integration of the semantic representation language Ty2 into an HPSG grammar. In particular, we saw that there is an RSRL grammar, $\mathcal{TY}2$, such that we can treat the objects in any exhaustive model of this grammar to represent terms of Ty2. In the last section of the preceding chapter, we extended the grammar of Section 2.3 so that it contains the grammar $\mathcal{TY}2$ as a subgrammar and that the CONTENT values of linguistic signs are terms of Ty2, i.e., objects of the sort *meaningful-expression* (*me*).

In Section 1.3 we presented the semantic framework of *Lexicalized Flexible Ty2* (LFTy2). This system is based on Hendriks 1993 and has the advantage that it accounts for scope variation without requiring a storage mechanism such as Cooper Storage (Cooper, 1975, 1983) or movement operations such as *Quantifier Raising* (May, 1977, 1985).

In this chapter, we will combine the system LF-Ty2 with the syntactic fragment of Section 2.3. In Section 4.1 we will present a first version of this integration. In the following sections, we will address the concrete RSRL formalization of this integration in more detail.

In our presentation of LF-Ty2 in Section 1.3, we saw that this semantic framework relies on three ingredients: (i) the *basic translation* of lexical elements, (ii) type shifting operations such as *argument raising* (AR) and *value raising* (VR) which apply recursively to the basic translations, and (iii) intensional functional application to compute the meaning/ the logical form of the mother node from that of the daughters in a branching tree.

In Section 3.5 we illustrated that the basic translation of lexical elements can be integrated into lexical entries as the description of the CONTENT value. In Section 4.1, we will see that this is also possible for the other two basic ingredients of LF-Ty2. The reason for this is that we can consider type shifting operations as mappings from one term ϕ to another term ψ such that ϕ is a subterm of ψ. Similarly, intensional functional application is a mapping from two terms ϕ_1 and ϕ_2 to a term ψ, where ϕ_1 and ϕ_2 are subterms of ψ. It is easy to illustrate this for intensional functional application, because there ψ is either the term $\phi_1(\lambda @.\phi_2)$ or the term $\phi_2(\lambda @.\phi_1)$. In either case, the terms ϕ_1 and ϕ_2 are subterms of ψ. For the shifting operations AR and VR this subterm relation is less obvious. But, it can be seen from the definitions of these operations as given in Section 1.3.3, that in both cases the original term is embedded in the resulting term.

In (145) we illustrate this subterm property of the shifting operation and of intensional functional application with a simple example.

(145) a. Intensional functional application of $\lambda x_{se}.\mathsf{walk}'_@(x_@)$ and m_e:
$[\lambda x_{se}.\mathsf{walk}'_@(x_@)](\lambda @.m)$
b. AR of $\lambda x_{se}.\mathsf{walk}'_@(x_@)$:
$\lambda Y.Y_@(\lambda @ \lambda y.[\lambda x_{se}.\mathsf{walk}'_@(x_@)](y))$
c. VR of $\lambda x_{se}.\mathsf{walk}'_@(x_@)$:
$\lambda y \lambda u.u_@(\lambda @.[\lambda x_{se}.\mathsf{walk}'_@(x_@)](y))$

In all the examples in (145), the input of the functions, intensional functional application, AR and VR appears as a subterm of the output. In the introduction of the framework of

LF-Ty2 in Section 1.3.3, we did not use the terms as they are given in (145), instead, we gave terms that have the same meaning, but differ syntactically, i.e., we applied λ-conversion to the terms. In (146) we give the fully λ-converted form of these terms:

(146) a. $[\lambda x_{se}.\mathsf{walk}'_{@}(x_{@})](\lambda @.m) \rightarrow_{\lambda} \mathsf{walk}'_{@}(m)$
b. $\lambda Y.Y_{@}(\lambda @ \lambda y.[\lambda x_{se}.\mathsf{walk}'_{@}(x_{@})](y)) \rightarrow_{\lambda} \lambda Y.Y_{@}(\lambda @ \lambda y.\mathsf{walk}'_{@}(y_{@}))$
c. $\lambda y \lambda u.u_{@}(\lambda @.[\lambda x_{se}.\mathsf{walk}'_{@}(x_{@})](y)) \rightarrow_{\lambda} \lambda y \lambda u.u_{@}(\lambda @.\mathsf{walk}'_{@}(y_{@}))$

There is an important difference between λ-conversion and the three operations exemplified in (145): while the latter preserve the input term(s) as subterm(s), but change the meaning, the former preserves the meaning, but the result of a one-step λ-conversion does not contain the input as a subterm in the output. In this thesis, we assume that the logical forms of signs is fully reduced with respect to λ-conversion. Therefore, we must provide the technical means to express λ-conversion within RSRL. In Section 4.2, we present two alternative encodings.

These two formalizations of λ-conversion differ in their ontological assumptions. In the first encoding, we are forced to extend the ontology of linguistic signs, i.e., we will introduce new attributes and sorts which will be used to encode the step-wise λ-conversion explicitly as components of a linguistic sign. The second encoding does not necessitate an extension of the ontology of signs. Instead, we will show that we can use chains to represent terms of Ty2 and that we can perform λ-conversion on these chains.

In Section 4.3 we will reconsider the derivational rules introduced for passive and complement extraction in Section 2.3.2. In that section, we did not specify effect of the derivational rules on the logical form of a word. We will show that all that is needed to combine our syntactic analysis of these phenomena with the semantic framework LF-Ty2 is to add a CONTENT specification to the input and the output of the derivational rules. While this extension is simple by itself, we will take it as a starting point to discuss the question at what level the shifting operations AR and VR should be applied. In our presentation of LF-Ty2 in Section 1.3 we have only considered non-derived words, to which the application of shifting rules is allowed, and regular phrases, in the calculation of whose logical form, we did not allow the application of shifting operations. With the introduction of derived words, a third kind of sign needs to be considered. We will show that shifting operations must be available for derived words as well as for non-derived words.

In Section 4.4 we will summarize the results of this chapter and provide a formalization of the simple lf-constraint presented in (53). At the end of this chapter, we have added a section in which we reflect on our use of RSRL in the encoding of LF-Ty2.

4.1. LF-Ty2 without λ-Conversion

In this section we will present all ingredients of LF-Ty2, but we will avoid λ-conversion. Instead, we will use non-reduced terms as CONTENT values of signs, as those given in (145). We will first reconsider the basic translations of words and, then, state a first version of the Semantics Principle. Finally, we will discuss two alternatives for the encoding of shifting operations.

In the presentation of LF-Ty2 in Section 1.3.3 we gave the basic translations of some words in our fragment in Table 1.1. At the end of Section 3.5 we showed how such basic translations can be integrated into lexical entries as those in (77). This was achieved by taking the basic translation of a word as the CONTENT specification in the lexical entry of the word. In (147) we give such lexical entries for the intransitive verb *walks* and for the transitive verb *loves*.

(147) a. Parts of the lexical entry of the word *walks*:

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{walks} \rangle \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{SUBCAT} & \left\langle \text{NP} \right\rangle \\ \text{MARKING} & \textit{unmarked} \end{bmatrix} \\ \text{CONT} & \lambda x_{se}.\mathsf{walk}'_{@}(x_{@}) \end{bmatrix} \\ \text{NONL} & \begin{bmatrix} \text{INHERITED SLASH} & \textit{eset} \\ \text{TO-BIND SLASH} & \textit{eset} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

b. Parts of the lexical entry of the word *loves*:

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{loves} \rangle \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{SUBCAT} & \left\langle \text{NP, NP} \right\rangle \\ \text{MARKING} & \textit{unmarked} \end{bmatrix} \\ \text{CONT} & \lambda y_{se}\lambda x_{se}.\mathsf{love}'_{@}(x_{@}, y_{@}) \end{bmatrix} \\ \text{NONL} & \begin{bmatrix} \text{INHERITED SLASH} & \textit{eset} \\ \text{TO-BIND SLASH} & \textit{eset} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

In the presentation of the framework of LF-Ty2, we saw that we can derive the logical form of some sentences without the application of shifting rules. A simple example was sentence (39a), repeated in (148).

(148) Every man walks.

The principle that is responsible for determining the logical form of phrases in HPSG is traditionally called the SEMANTICS PRINCIPLE (SP). In (29) in Section 1.2 we quoted the SEMANTICS PRINCIPLE of Pollard and Sag 1994. As we have seen in that section, the SP of Pollard and Sag 1994 incorporates the storage mechanism assumed there for the treatment of quantifier scope. For LF-Ty2, the SP only expresses that the CONTENT value of a phrase is the intensional functional application of the CONTENT values of the daughters. This principle is stated in (149).

(149) The SEMANTICS PRINCIPLE (SP):

$$\textit{phrase} \Rightarrow \left(\begin{bmatrix} \text{SYNS LOC CONT } \boxed{1} & \\ \text{DTRS} & \begin{bmatrix} \text{H-DTR SYNS LOC CONT } \boxed{2} \\ \text{N-DTR SYNS LOC CONT } \boxed{3} \end{bmatrix} \end{bmatrix} \\ \textbf{and } \texttt{intensional-functional-application}(\boxed{1},\boxed{2},\boxed{3}) \right)$$

The SP as given in (149) contains a relational call which uses the relation `intensional-functional-application` (`ifa`). In (150) we give a preliminary definition of this relation, which will be further refined, when we add the treatment of λ-conversion.

(150) The relation `intensional-functional-application` (`ifa`):

$$\texttt{ifa}(\boxed{1},\boxed{2},\boxed{3}) \overset{\forall}{\Longleftarrow} \boxed{1}\begin{bmatrix} \textit{appl} & \\ \text{FUNC} & \boxed{2} \\ \text{ARG} & \begin{bmatrix} \textit{abstr} & \\ \text{VAR} & v_{s,0} \\ \text{ARG} & \boxed{3} \end{bmatrix} \end{bmatrix} \qquad \texttt{ifa}(\boxed{1},\boxed{2},\boxed{3}) \overset{\forall}{\Longleftarrow} \boxed{1}\begin{bmatrix} \textit{appl} & \\ \text{FUNC} & \boxed{3} \\ \text{ARG} & \begin{bmatrix} \textit{abstr} & \\ \text{VAR} & v_{s,0} \\ \text{ARG} & \boxed{2} \end{bmatrix} \end{bmatrix}$$

The two clauses in the definition of the relation `ifa` encode intensional functional application. If we assume that the CONTENT value of the head daughter is the term ϕ_1 and that the CONTENT value of the nonhead daughter is the term ϕ_2, then the first clause specifies that the CONTENT value of the phrase is the term $\phi_1(\lambda @.\phi_2)$. The second clause contains the option that the nonhead daughter is taken as the semantic functor. In this case, the CONTENT value of the phrase is of the form $\phi_2(\lambda @.\phi_1)$.

Given the lexical entries of the words *every*, *man* and *walks*, and the SP, we give the structure of sentence (148), as it is predicted by our grammar in Figure 4.1 (page 145).

In the tree in Figure 4.1 we use a mixed AVM and term notation for the CONTENT values of the signs in the tree. Doing this, we can see how the relation `ifa` constrains the CONTENT values of phrases. For the three words in this sentence, the basic translation is given as the CONTENT values. For the two phrases, the NP and the S node, the CONTENT value is given as the intensional functional application of the logical forms of their daughters. In both cases, the CONTENT value of the nonhead daughter appears as the functor, i.e., as the FUNC value in the CONTENT value of the phrase ($\boxed{5}$ in the case of the NP, and $\boxed{6}$ at the S node). The logical form of the head daughter is in both cases identical with the ARG ARG value of the logical form of the phrase ($\boxed{4}$ at the NP and $\boxed{3}$ at the S node).

The tree in Figure 4.1 should be compared with the structure of the sentence given in Figure 1.10 on page 47. If the SP uses the relation `ifa` as given in (150), the semantic interpretation of the *me* object in the CONTENT value of a phrase is the same as that of the terms that we assigned the phrasal nodes in Figure 1.10. In the latter case, however, we executed all possible λ-conversions. In (151) we indicate the terms that correspond to the CONTENT values as determined by the SP in (149) in the first line, and the reduced forms of these terms in the second line.

(151) a. NP (compare the step-by-step derivation in (40)):
$(\lambda P \lambda Q.\forall x[P_@(x) \rightarrow Q_@(x)])(\lambda @ \lambda x.\mathsf{man}'_@(x_@))$
$\rightarrow_\lambda \lambda Q.\forall x[\mathsf{man}'_@(x_@) \rightarrow Q_@(x)]$
b. S (compare the step-by-step derivation in (41)):
$((\lambda P \lambda Q.\forall x[P_@(x) \rightarrow Q_@(x)])(\lambda @ \lambda x.\mathsf{man}'_@(x_@)))(\lambda @ \lambda x.\mathsf{walk}'_@(x_@))$
$\rightarrow_\lambda \forall x[\mathsf{man}'_@(x_@) \rightarrow \mathsf{walk}'_@(x_@)$

The preliminary version of the SP as found in (149) guarantees that the semantic interpretation of phrases is as defined in our initial presentation of LF-Ty2 in Section 1.3.3. As we have not yet provided the technical means to express λ-conversion, the CONTENT values of the phrases differ syntactically from the terms given as logical forms in Section 1.3.3.

The next component of the framework LF-Ty2 are the shifting rules AR and VR as given in Definition 1.10 and Definition 1.11. In our exposition of the framework LF-Ty2, we have restricted the application of these rules to words. Put differently, the logical form of a word was said to be either the basic translation of a word or the (possibly) iterated application of AR and VR to this basic translation. There are two possible formalizations of this: as the application of the shifting rules is restricted to words, we can encode these rules as derivational rules as introduced in Section 2.3.2. Alternatively, we could incorporate the option of applying shifting operations as part of the lexical entry. For both formalizations, we must encode AR and VR as relations. We will first define these two relations, and then turn to the two possible integrations of the shifting operations into our overall grammar. As VR is formally simpler than AR, we will discuss this shifting operation first. Below, we repeat the definition of the operation VR from Section 1.3.3.

DEFINITION 1.11

For each type $d \in Type$,

VR_d is a relation between two terms α and β such that

if α is of some type $a_1(\ldots(a_n b)\ldots)$.
then β is some term $\lambda x_{a_1,1}\ldots\lambda x_{a_n,n}\lambda u_{s((sb)d)}.u(@)(\lambda @.\alpha(x_1)\ldots(x_n))$.

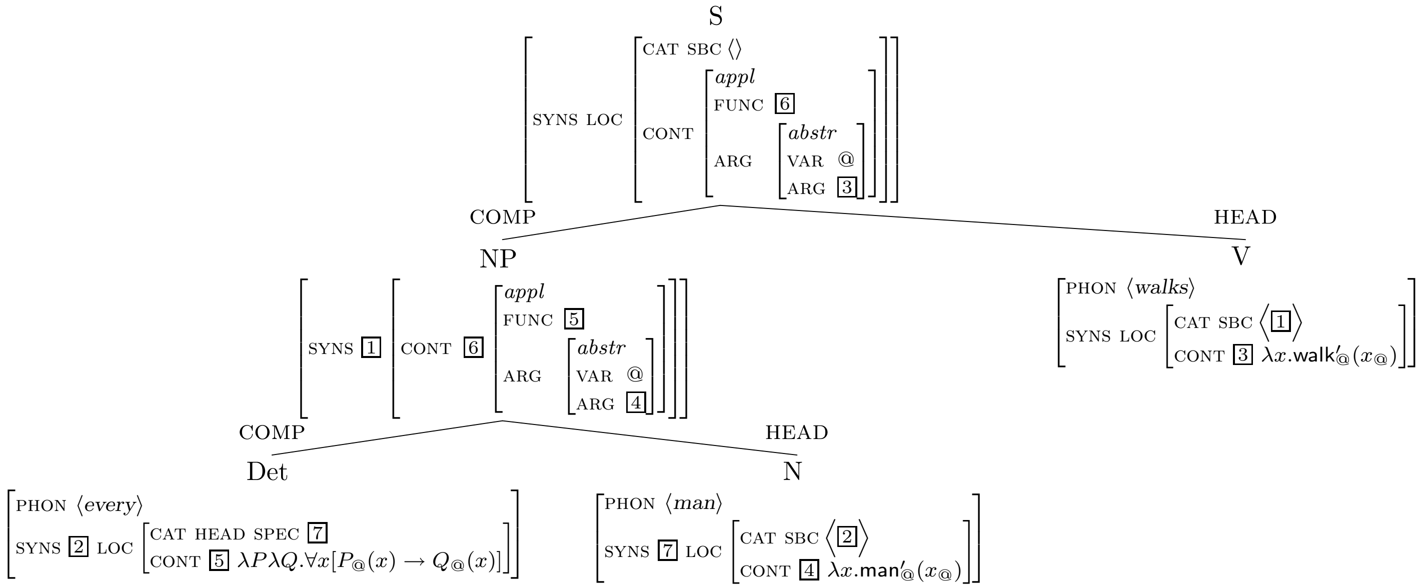


FIGURE 4.1. The structure of the sentence *Every man walks*.

As pointed out in Section 1.3.3, we use the operation VR to relate the basic translation of a proper name such as *Mary*, which is simply the term m_e, to the more complex translation assumed in Montague's work, i.e., $\lambda P.P_{@}(\lambda @.m)$. In addition, we need VR to derive the *de re* readings of sentences such as (39c), repeated in (152).

(152) Every man believes that some woman walks.

In (153) we define the relation `value-raising` (`vr`) which relates two *me* objects if the corresponding terms are related by the operation VR.

(153) The relation `value-raising` (`vr`):

$$\texttt{vr}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow} \boxed{2}\begin{bmatrix} \textit{abstr} & \\ \text{VAR} & \boxed{3} \\ \text{ARG} & \begin{bmatrix} \textit{appl} & \\ \text{FUNC} & \begin{bmatrix} \textit{appl} & \\ \text{FUNC} & \boxed{3} \\ \text{ARG} & @ \end{bmatrix} \\ \text{ARG} & \begin{bmatrix} \textit{abstr} & \\ \text{VAR} & @ \\ \text{ARG} & \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\texttt{vr}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow} \left(\boxed{3}\begin{bmatrix} \textit{appl} & \\ \text{FUNC} & \boxed{1} \\ \text{ARG} & \boxed{4} \end{bmatrix} \textbf{ and } \boxed{2}\begin{bmatrix} \textit{abstr} & \\ \text{VAR} & \boxed{4} \\ \text{ARG} & \boxed{5} \end{bmatrix} \textbf{ and } \texttt{vr}(\boxed{3},\boxed{5}) \right)$$

The definition of the relation `vr` is given in two clauses. The first clause treats the case where the term that we want to apply VR to is simply of some type b, instead of some type $a_1(\ldots(a_n b))$. Applying value raising to the constant m_e is such a case. In (154) we give an AVM description of the term m_e (in (a)) and of its value-raised correspondence (in (b)). In (154b) we use the same tags as in the first clause of the definition of the relation `vr` in (153). This makes clear that for each two objects o_1 and o_2, if o_1 is described by the AVM in (154a) and o_2 is described by the AVM in (154b), then o_1 and o_2 also meet the first clause in the definition of `vr`. As a consequence, o_1 and o_2 must be in the interpretation of the relation `vr`.

(154) a. m_e: $\begin{bmatrix} \textit{mary} & \\ \text{TYPE} & \textit{entity} \end{bmatrix}$

b. $\lambda P_{s((se)t)}.P(@)(\lambda @.m)$:

$$\boxed{2}\begin{bmatrix} \textit{abstr} & \\ \text{VAR} & \boxed{3}\ P_{s((se)t)} \\ \text{ARG} & \begin{bmatrix} \textit{appl} & \\ \text{FUNC} & \begin{bmatrix} \textit{appl} & \\ \text{FUNC} & \boxed{3} \\ \text{ARG} & @ \end{bmatrix} \\ \text{ARG} & \begin{bmatrix} \textit{abstr} & \\ \text{VAR} & @ \\ \text{ARG} & \boxed{1}\begin{bmatrix} \textit{mary} & \\ \text{TYPE} & \textit{entity} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

To illustrate the second clause in the definition of the relation `vr`, we must consider a more complex example. For this purpose, reconsider example (152). In Section 1.3.3 we showed that the derivation of the *de re* readings of this sentence requires the application of VR to the basic translation of the verb *walks*. In (155) we indicate this basic translation and its value-raised form.

(155) *walks* $\leadsto$ $\lambda x_{se}.\mathsf{walk}'_{@}(x(@))$
$\longrightarrow_{VR}$ $\lambda y_{se}\lambda u_{s((st)t)}.u(@)(\lambda @[(\lambda x.\mathsf{walk}'_{@}(x(@)))(y)])$

According to the definition of `vr`, the basic translation of the word *walks* is value-raised to the second term in (155) if the following two terms are also in the relation `vr`:

(156) a. $(\lambda x_{se}.\mathsf{walk}'_{@}(x(@)))(y)$
b. $\lambda u_{s((st)t)}.u(@)(\lambda @[(\lambda x.\mathsf{walk}'_{@}(x(@)))(y)])$

The term in (156a) is the functional application of the basic translation of the word *walks*, to the variable y. If we compare the term in (156b) to the output of the shifting operation as indicated in the last line in (155), we see that the term in (b) is exactly the argument of the outermost λ-abstractor, and that y is exactly the variable that is bound by this abstractor.

It can be seen that the terms in (156) satisfy the requirements of the first clause of the relation `vr`. For illustration, consider the following AVM description of the term in (156b), where we use the same tags as in the first clause of the definition of the relation `vr`.

(157)
$$\boxed{2}\begin{bmatrix} \textit{abstr} \\ \text{VAR} \; \boxed{3}\; u_{s((st)t)} \\ \text{ARG} \begin{bmatrix} \textit{appl} \\ \text{FUNC} \begin{bmatrix} \textit{appl} \\ \text{FUNC}\ \boxed{3} \\ \text{ARG}\ @ \end{bmatrix} \\ \text{ARG} \begin{bmatrix} \textit{abstr} \\ \text{VAR}\ @ \\ \text{ARG}\ \boxed{1}\ (\lambda x_{se}.\mathsf{walk}'_{@}(x(@)))(y) \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

This example should suffice to show how the two clauses of the definition of the relation `vr` in (153) achieve the effect of encoding the operation value raising.

The second shifting operation considered in Section 1.3.3, is *argument raising* (AR). The relation AR is needed to derive scope ambiguities among co-arguments of the same predicate. Below, we repeat the definition of the operation AR.

DEFINITION 1.10

For each $i \in \mathbb{N}$, AR_i is a relation between two terms α and β such that

if α is of some type $(a_1(\dots((sa_i)(\dots(a_n b)\dots)))))$
then β is some term
$\lambda x_{a_1,1}\dots\lambda X_{s((s((sa_i)b))b),i}\dots\lambda x_{a_n,n}.X(@)(\lambda @\lambda x_{sa_i,i}.\alpha(x_1)\dots(x_i)\dots(x_n))$

We have used this operation to derive the two readings of the sentence in (39b), which we repeat for convenience in (158).

(158) Every man loves some woman.

In (159) we give the basic translation of the verb *love* together with its argument-raised versions. As the basic translation of the verb has two semantic arguments, we can apply AR to either of them. In (159) we show the result of applying AR to the first semantic argument first ($\longrightarrow_{AR1}$), and then, the result of applying AR to the second semantic argument ($\longrightarrow_{AR2}$).

(159) *loves* $\leadsto$ $\lambda x_1\lambda x_2.\mathsf{love}'_{@}(x_2(@), x_1(@))$
$\longrightarrow_{AR1}$ $\lambda Y_1\lambda y_2.Y_1(@)(\lambda @\lambda y_1.[(\lambda x_1\lambda x_2.\mathsf{love}'_{@}(x_2(@), x_1(@)))(y_1)(y_2)])$
$\longrightarrow_{AR2}$ $\lambda y_1\lambda Y_2.Y_2(@)(\lambda @\lambda y_2.[(\lambda x_1\lambda x_2.\mathsf{love}'_{@}(x_2(@), x_1(@)))(y_1)(y_2)])$

We can define the operation AR as a relation that holds between two *me* objects. The relation `argument-raising` (`ar`) is defined in (160a). As can be seen, the definition of this relation involves an auxiliary relation, `ar-aux`, whose definition is added in (160b).

(160) a. The relation `argument-raising` (`ar`):

$$\texttt{ar}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow} \left(\boxed{3}\begin{bmatrix} \textit{appl} & \\ \textsc{func} & \boxed{1} \\ \textsc{arg} & \boxed{4}\end{bmatrix} \textbf{ and } \boxed{2}\begin{bmatrix}\textit{abstr} & \\ \textsc{var} & \boxed{5}\\ \textsc{arg} & \boxed{6}\end{bmatrix} \textbf{ and } \texttt{ar-aux}(\boxed{3},\boxed{6},\boxed{4},\boxed{5})\right)$$

$$\texttt{ar}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow} \left(\boxed{3}\begin{bmatrix} \textit{appl} & \\ \textsc{func} & \boxed{1} \\ \textsc{arg} & \boxed{4}\end{bmatrix} \textbf{ and } \boxed{2}\begin{bmatrix}\textit{abstr} & \\ \textsc{var} & \boxed{4}\\ \textsc{arg} & \boxed{5}\end{bmatrix} \textbf{ and } \texttt{ar}(\boxed{3},\boxed{5})\right)$$

b. The relation `ar-aux`

$$\texttt{ar-aux}(\boxed{1},\boxed{2},\boxed{3},\boxed{4}) \overset{\forall}{\Longleftarrow} \left(\boxed{2}\begin{bmatrix} \textit{appl} & \\ \textsc{type} & \boxed{5} \\ \textsc{func} & \begin{bmatrix}\textit{appl} & \\ \textsc{func} & \boxed{4} \\ \textsc{arg} & @\end{bmatrix} \\ \textsc{arg} & \begin{bmatrix}\textit{abstr} & \\ \textsc{var} & @ \\ \textsc{arg} & \begin{bmatrix}\textit{abstr} & \\ \textsc{var} & \boxed{3} \\ \textsc{arg} & \boxed{1}\begin{bmatrix}\textsc{type} & \boxed{5}\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix}\right)$$

$$\texttt{ar-aux}(\boxed{1},\boxed{2},\boxed{3},\boxed{4}) \overset{\forall}{\Longleftarrow} \left(\boxed{5}\begin{bmatrix} \textit{appl} & \\ \textsc{func} & \boxed{1} \\ \textsc{arg} & \boxed{6}\end{bmatrix} \textbf{ and } \boxed{2}\begin{bmatrix}\textit{abstr} & \\ \textsc{var} & \boxed{6}\\ \textsc{arg} & \boxed{7}\end{bmatrix} \textbf{ and } \texttt{ar-aux}(\boxed{5},\boxed{7},\boxed{3},\boxed{4})\right)$$

We can use the examples in (159) to illustrate how the definition of the relations `ar` and `ar-aux` work. Let us first consider the basic translation of the verb *loves* and the application of AR to the first semantic argument, as given in the second line in (159).

These two terms are in the relation `ar` if they satisfy one of the clauses in (160a). In this particular case, it is the first clause that we should consider. In (161) we insert the basic translation of the verb *loves* and the term that results from an application of AR_1 as the arguments of the relation. For the sake of clarity, we use the same tags as in the first clause of the definition of the relation.

(161)
$$\boxed{3}\begin{bmatrix} \textit{appl} & \\ \textsc{func} & \boxed{1}\ \lambda x_1 \lambda x_2.\mathsf{love}'_@(x_2(@), x_1(@)) \\ \textsc{arg} & \boxed{4}\ y_1 \end{bmatrix}$$
$$\textbf{and}\quad \boxed{2}\begin{bmatrix} \textit{abstr} & \\ \textsc{var} & \boxed{5}\ Y_1 \\ \textsc{arg} & \boxed{6}\ \lambda y_1.Y_2(@)(\lambda @ \lambda y_1.[(\lambda x_1 \lambda x_2.\mathsf{love}'_@(x_2(@), x_1(@)))(y_1)(y_2)]) \end{bmatrix}$$
$$\textbf{and}\ \texttt{ar-aux}(\boxed{3},\boxed{6},\boxed{4},\boxed{5})$$

In a next step, we must verify that the quadruple consisting of the term indicated by $\boxed{3}$ in (161), the term indicated by $\boxed{6}$, and the variables indicated by $\boxed{4}$ and $\boxed{5}$ satisfies the clauses of the relation `ar-aux`. In (162) we give these four *me* objects in term notation.

(162)
$$\begin{array}{lcl} \boxed{3} & = & (\lambda x_1 \lambda x_2.\mathsf{love}'_@(x_2(@), x_1(@)))(y_1) \\ \boxed{6} & = & \lambda y_2.Y_1(@)(\lambda @ \lambda y_1.[(\lambda x_1 \lambda x_2.\mathsf{love}'_@(x_2(@), x_1(@)))(y_1)(y_2)]) \\ \boxed{4} & = & y_1 \\ \boxed{5} & = & Y_1 \end{array}$$

To check whether this quadruple satisfies one of the clauses of the relation `ar-aux` as given in (160b), let us consider the second clause.

(163)
$$\boxed{5}\begin{bmatrix} \textit{appl} & \\ \textsc{func} & \boxed{1}\ (\lambda x_1 \lambda x_2.\mathsf{love}'_@(x_2(@), x_1(@)))(y_1) \\ \textsc{arg} & \boxed{6}\ y_2 \end{bmatrix}$$

$$\textbf{and}\quad \boxed{2}\begin{bmatrix} \textit{abstr} \\ \text{VAR}\ \boxed{6} \\ \text{ARG}\ \boxed{7}\ Y_1(@)(\lambda @ \lambda y_1.[(\lambda x_1 \lambda x_2.\mathsf{love}'_@(x_2(@), x_1(@)))(y_1)(y_2)]) \end{bmatrix}$$
$$\textbf{and}\ \texttt{ar-aux}(\boxed{5},\boxed{7},\boxed{3},\boxed{4})$$

To check whether this is the case, we must see whether the quadruple $\langle\boxed{5},\boxed{7},\boxed{3},\boxed{4}\rangle$ satisfies a clause of the relation `ar-aux`. In (164) we give the term notation of the *me* objects that these tags point to.

$$(164)\quad \begin{array}{lcl} \boxed{5} & = & (\lambda x_1 \lambda x_2.\mathsf{love}'_@(x_2(@), x_1(@)))(y_1)(y_2) \\ \boxed{7} & = & Y_1(@)(\lambda @ \lambda y_1.[(\lambda x_1 \lambda x_2.\mathsf{love}'_@(x_2(@), x_1(@)))(y_1)(y_2)]) \\ \boxed{3} & = & y_1 \\ \boxed{4} & = & Y_1 \end{array}$$

Finally, we can show that this quadruple satisfies the first clause of the relation `ar-aux`. For illustration, consider the AVM notation of the second argument of the relation, where the same tags are used as in the first clause of the relation `ar-aux`.

$$(165)\quad \boxed{2}\begin{bmatrix} \textit{appl} \\ \text{TYPE}\ \ \textit{truth} \\ \text{FUNC}\ \begin{bmatrix} \textit{appl} \\ \text{FUNC}\ \boxed{4}\ Y_1 \\ \text{ARG}\ \ @ \end{bmatrix} \\ \text{ARG}\ \begin{bmatrix} \textit{abstr} \\ \text{VAR}\ \ @ \\ \text{ARG}\ \begin{bmatrix} \textit{abstr} \\ \text{VAR}\ \boxed{3}\ y_1 \\ \text{ARG}\ \boxed{1}\ ((\lambda x_1 \lambda x_2.\mathsf{love}'_@(x_2(@), x_1(@)))(y_1)(y_2))_t \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

So far, we have illustrated the first clause of the relation `ar` and the two clauses of the relation `ar-aux`. What remains is to show an instance of the second clause of the relation `ar`, i.e., the recursive case. The application of AR to the second semantic argument of the basic translation of the verb *loves* is suited to serve as such an example. In (167) we repeat the basic translation of the verb together with the term that results from applying AR2.

$$(166)\quad \begin{array}{ll} \textit{loves} \leadsto & \lambda x_1 \lambda x_2.\mathsf{love}'_@(x_2(@), x_1(@)) \\ \longrightarrow_{AR2} & \lambda y_1 \lambda Y_2.Y_2(@)(\lambda @ \lambda y_2.[(\lambda x_1 \lambda x_2.\mathsf{love}'_@(x_2(@), x_1(@)))(y_1)(y_2)]) \end{array}$$

In the following, we are going to show that the two terms in (166) stand in the relation `ar`. This is the case, if they satisfy either of the clauses of this relation. If they satisfy the second clause, the following must hold:

$$(167)\quad \boxed{3}\begin{bmatrix} \textit{appl} \\ \text{FUNC}\ \boxed{1}\ \lambda x_1 \lambda x_2.\mathsf{love}'_@(x_2(@), x_1(@)) \\ \text{ARG}\ \ \boxed{4}\ y_1 \end{bmatrix}$$
$$\textbf{and}\quad \boxed{2}\begin{bmatrix} \textit{abstr} \\ \text{VAR}\ \boxed{4}\ y_1 \\ \text{ARG}\ \boxed{5}\ \lambda Y_2.Y_2(@)(\lambda @ \lambda y_2.[(\lambda x_1 \lambda x_2.\mathsf{love}'_@(x_2(@), x_1(@)))(y_1)(y_2)]) \end{bmatrix}$$
$$\textbf{and}\ \texttt{ar}(\boxed{3},\boxed{5})$$

To see whether this is the case, the terms referred to with the tags $\boxed{3}$ and $\boxed{5}$ must stand in the relation `ar` as well. To check this, we use the first clause of this relation.

$$(168)\quad \boxed{3}\begin{bmatrix} \textit{appl} \\ \text{FUNC}\ \boxed{1}\ (\lambda x_1 \lambda x_2.\mathsf{love}'_@(x_2(@), x_1(@)))(y_1) \\ \text{ARG}\ \ \boxed{4}\ y_2 \end{bmatrix}$$

$$\textbf{and}\quad \boxed{2}\begin{bmatrix} \textit{abstr} \\ \text{VAR}\ \boxed{5}\ Y_2 \\ \text{ARG}\ \boxed{6}\ Y_2(@)(\lambda @ \lambda y_2.[(\lambda x_1 \lambda x_2.\mathsf{love}'_@(x_2(@), x_1(@)))(y_1)(y_2)]) \end{bmatrix}$$
$$\textbf{and}\ \texttt{ar-aux}(\boxed{3},\boxed{6},\boxed{4},\boxed{5})$$

This is the case if the quadruple consisting of the terms referred to with the tags $\boxed{3}$, $\boxed{6}$, $\boxed{4}$ and $\boxed{5}$ is in the interpretation of the relation `ar-aux`. For convenience, we repeat these four terms in term notation.

(169)
$$\begin{array}{lcl} \boxed{3} & = & (\lambda x_1 \lambda x_2.\mathsf{love}'_@(x_2(@), x_1(@)))(y_1)(y_2) \\ \boxed{6} & = & Y_2(@)(\lambda @ \lambda y_2.[(\lambda x_1 \lambda x_2.\mathsf{love}'_@(x_2(@), x_1(@)))(y_1)(y_2)]) \\ \boxed{4} & = & y_2 \\ \boxed{5} & = & Y_2 \end{array}$$

In (170) we show that this is actually the case, as the terms meet the first clause in the definition of the relation `ar-aux`.

(170)
$$\boxed{2}\begin{bmatrix} \textit{appl} \\ \text{TYPE}\ \ \textit{truth} \\ \text{FUNC}\ \begin{bmatrix} \textit{appl} \\ \text{FUNC}\ \boxed{4}\ Y_2 \\ \text{ARG}\ \ @ \end{bmatrix} \\ \text{ARG}\ \begin{bmatrix} \textit{abstr} \\ \text{VAR}\ \ @ \\ \text{ARG}\ \begin{bmatrix} \textit{abstr} \\ \text{VAR}\ \boxed{3}\ y_2 \\ \text{ARG}\ \boxed{1}\ (\lambda x_1 \lambda x_2.\mathsf{love}'_@(x_2(@), x_1(@)))(y_1)(y_2) \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

With the AVM in (170) we have illustrated how the relation `ar` captures the effect of the shifting operation AR2 on the basic translation of the verb *loves* as given in (166). We have thus given examples for every clause of the definition of the relations `ar` and `ar-aux`.

Given the relations `ar` and `vr`, we can address the issue of how the basic translation of a word is related to its shifted variants. In the present section, we will present and evaluate two alternative methods for the integration of shifting rules into our fragment. As mentioned above, one of these options is to encode shifting rules as derivational rules. We will call this the DR-encoding of shifting operations. In the second encoding, we will incorporate the possible applications of shifting rules in the lexical entries. This will be referred to as the LE-encoding. In the following, we will first present the DR-encoding, then the LE-encoding, and finally discuss which of these encodings seems to be more appropriate.

To integrate shifting operations into our grammar, we can define two derivational rules, AR-DR and VR-DR, which relate an input word to an output word which differs from the input only with respect to the CONTENT value. In the case of the AR-DR, the output's CONTENT value is the application of argument raising to the CONTENT value of the input. For VR-DR, the relation between the two CONTENT values is that of value raising. In (171) we give both derivational rules in the formally precise notation, i.e., as disjuncts in the consequent of the DR PRINCIPLE in (99).

(171) a. The AR-DR:

$$\begin{bmatrix} \textit{der-rule} \\ \text{IN} & \begin{bmatrix} \textit{word} \\ \text{PHON} & \boxed{3} \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \boxed{4} \\ \text{CONT} & \boxed{1} \end{bmatrix} \\ \text{NONL} & \boxed{5} \end{bmatrix} \end{bmatrix} \\ \text{OUT} & \begin{bmatrix} \textit{word} \\ \text{PHON} & \boxed{3} \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \boxed{4} \\ \text{CONT} & \boxed{2} \end{bmatrix} \\ \text{NONL} & \boxed{5} \end{bmatrix} \end{bmatrix} \end{bmatrix} \textbf{ and } \texttt{ar}(\boxed{1},\boxed{2})$$

b. The VR-DR:

$$\begin{bmatrix} \textit{der-rule} \\ \text{IN} & \begin{bmatrix} \textit{word} \\ \text{PHON} & \boxed{3} \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \boxed{4} \\ \text{CONT} & \boxed{1} \end{bmatrix} \\ \text{NONL} & \boxed{5} \end{bmatrix} \end{bmatrix} \\ \text{OUT} & \begin{bmatrix} \textit{word} \\ \text{PHON} & \boxed{3} \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \boxed{4} \\ \text{CONT} & \boxed{2} \end{bmatrix} \\ \text{NONL} & \boxed{5} \end{bmatrix} \end{bmatrix} \end{bmatrix} \textbf{ and } \texttt{vr}(\boxed{1},\boxed{2})$$

Given this encoding, we can show the structure of a word whose CONTENT value is shifted. In (172) we describe the word *loves* with AR1 applied to the logical form as indicated in (159). Note that the tag $\boxed{1}$ appears as the CONTENT value of the input word and as a subterm of the CONTENT value of the output word.

(172) AR in the DR-encoding:

$$\boxed{0} \begin{bmatrix} \textit{word} \\ \text{PHON} & \boxed{3} \langle \textit{loves} \rangle \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \boxed{4} \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{SUBCAT} & \langle \text{NP, NP} \rangle \\ \text{MARKING} & \textit{unmarked} \end{bmatrix} \\ \text{CONT} & \boxed{2}\ \lambda Y_1 \lambda y_2 . Y_1(@)(\lambda @ \lambda y_1 . [(\boxed{1})(y_1)(y_2)]) \end{bmatrix} \\ \text{NONL} & \boxed{5} \begin{bmatrix} \text{INHER SLASH} & \textit{eset} \\ \text{TO-BIND SLASH} & \textit{eset} \end{bmatrix} \end{bmatrix} \\ \text{STORE} & \left\langle \begin{bmatrix} \textit{der-rule} \\ \text{IN} & \begin{bmatrix} \text{PHON} & \boxed{3} \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \boxed{4} \\ \text{CONT} & \boxed{1}\ \lambda x_1 \lambda x_2 . \mathsf{love}'_@(x_2(@), x_1(@)) \end{bmatrix} \\ \text{NONL} & \boxed{5} \end{bmatrix} \end{bmatrix} \\ \text{OUT} & \boxed{0} \end{bmatrix} \right\rangle \end{bmatrix}$$

$$\textbf{and } \texttt{ar}(\boxed{1},\boxed{2})$$

The derived word described in (172) has all the properties of its non-derived base, whose lexical entry was sketched in (147b). This means that its PHON value is the list ⟨*loves*⟩, it is a verb with two elements on its SUBCAT list and a MARKING value of sort *unmarked*, and the NONLOCAL values are all empty. As a derived word, it has a non-empty STORE value. The single element in this list is a *der-rule* object as described by the AR-DR. Remember that the STORE PRINCIPLE in (100) ensures that the overall word be identical to its STORE FIRST OUT value. Just as expressed in the DR, the only difference between the input of the DR and the derived word lies in the CONTENT value. The logical form of the derived word is the argument-raised form of the CONTENT value of the input word.

When we integrated DRs as part of our syntactic analysis in Section 2.3.2, we pointed out that DRs should be considered as non-branching syntactic structures where both the mother and the single daughter are of sort *word*. As a consequence, the derivational structure is an explicit part of the configuration of objects that constitutes the derived word.

There is an alternative to the encoding of AR and VR as derivational rules: we can express them directly in the lexical entries of words. This encoding is only possible at this point because of the subterm property of shifting operations, i.e., because for each of the shifting operations, the input term is a subterm of the output term. This encoding does, however, lead to a change in the lexical entries of each word. In (173) we give the modified lexical entry for the word *loves*.

(173) Parts of the lexical entry for the word *loves*, including shifting operations:

$$\mathbf{E}\boxed{1}\ \mathbf{E}\boxed{2}\ \begin{bmatrix} \textit{word} \\ \text{PHON} & \langle \textit{loves} \rangle \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{SUBCAT} & \left\langle \text{NP, NP} \right\rangle \\ \text{MARKING} & \textit{unmarked} \end{bmatrix} \\ \text{CONT} & \boxed{2} \end{bmatrix} \\ \text{NONL} & \begin{bmatrix} \text{INHER SLASH} & \textit{eset} \\ \text{TO-BIND SLASH} & \textit{eset} \end{bmatrix} \end{bmatrix} \end{bmatrix} \quad \begin{array}{l} \textbf{and}\ \boxed{1}\left[\lambda x_1 \lambda x_2.\mathsf{love}'_{@}(x_2(@), x_1(@))\right] \\ \textbf{and}\ \texttt{shifting}(\boxed{1},\boxed{2}) \end{array}$$

The lexical entry given in (173) differs from that in (147b) only with respect to the CONTENT specification. In (147b), we stated the CONTENT value directly as part of the lexical entry. In the lexical entry in (173), the CONTENT value is described by a variable, the tag $\boxed{2}$. Furthermore, there is the specification that the value of this variable must stand in the relation `shifting` with the value of the variable $\boxed{1}$. The variable $\boxed{1}$ is described to be an object of the form $\lambda x_1 \lambda x_2.\mathsf{love}'_{@}(x_2(@), x_1(@))$, i.e., the basic translation of the verb according to Table 1.1.

In the lexical entry in (173) we have used the relation `shifting`. This relation expresses the reflexive transitive closure over the relations `ar` and `vr`. In (174) the definitions for this relation are given.

(174) The relation `shifting`:

$$\begin{array}{lcl} \texttt{shifting}(\boxed{1},\boxed{2}) & \stackrel{\forall}{\Longleftarrow} & \boxed{1} \approx \boxed{2} \\ \texttt{shifting}(\boxed{1},\boxed{2}) & \stackrel{\forall}{\Longleftarrow} & \begin{pmatrix} \texttt{ar}(\boxed{1},\boxed{3}) \\ \textbf{and}\ \texttt{shifting}(\boxed{3},\boxed{2}) \end{pmatrix} \\ \texttt{shifting}(\boxed{1},\boxed{2}) & \stackrel{\forall}{\Longleftarrow} & \begin{pmatrix} \texttt{vr}(\boxed{1},\boxed{3}) \\ \textbf{and}\ \texttt{shifting}(\boxed{3},\boxed{2}) \end{pmatrix} \end{array}$$

The first clause in this definition expresses that a term o stands in the relation `shifting` with itself. Furthermore, two terms o and o' stand in this relation if there is a third term o''' such that o and o''' are in the relation `ar` or in the relation `vr`, and o''' and o'' are in the relation `shifting`.

Given this definition, it can be seen that the lexical entry of the verb *loves* in (173) licenses *word* objects that have the syntactic properties as described in the AVM. In addition, the CONTENT value can either be of the form

$$\lambda x_1 \lambda x_2.\mathsf{walk}'_{@}(x_2(@), x_1(@)),$$

i.e., the basic translation of the verb, or a shifted form thereof.

The CONTENT value of the verb is its basic translation if the variables $\boxed{1}$ and $\boxed{2}$ are interpreted as the same object. As the relation `shifting` is reflexive (by the first clause), every term stands in this relation with itself.

The lexical entry in (173) does, however, also license a verb whose the CONTENT value is of the argument-raised form

$$\lambda Y_1 \lambda y_2 . Y_1(@)(\lambda @ \lambda y_1 . [(\lambda x_1 \lambda x_2 . \mathsf{love}'_@(x_2(@), x_1(@)))(y_1)(y_2)]).$$

This is the case, because the relation `ar` holds between the basic translation of the verb, i.e., the term referred to with the tag $\boxed{1}$, and the CONTENT value, indicated by the tag $\boxed{2}$. As the relation `vr` holds between these terms, so does the relation `shifting`, which is the reflexive transitive closure over the two shifting operations assumed for LF-Ty2. In (175) we describe a word that is licensed by the lexical entry in (173), where the CONTENT is the shifted form of the basic translation.

(175) AR in the LE-encoding:

$$\begin{bmatrix} \textit{word} \\ \text{PHON } \langle \textit{loves} \rangle \\ \text{SYNS} \begin{bmatrix} \text{LOC} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{SUBCAT} & \left\langle \text{NP, NP} \right\rangle \\ \text{MARKING} & \textit{unmarked} \end{bmatrix} \\ \text{CONT } \lambda Y_1 \lambda y_2 . Y_1(@)(\lambda @ \lambda y_1 . [(\lambda x_1 \lambda x_2 . \mathsf{love}'_@(x_2(@), x_1(@)))(y_1)(y_2)]) \end{bmatrix} \\ \text{NONL} \begin{bmatrix} \text{INHERITED SLASH} & \textit{eset} \\ \text{TO-BIND SLASH} & \textit{eset} \end{bmatrix} \end{bmatrix} \\ \text{STORE } \textit{elist} \end{bmatrix}$$

Under this encoding of shifting operations, the application of shifting operations is not an explicit part of the words, i.e., there is a difference between shifting operations and derivational rules in that the second are non-branching trees, whereas the first do not feature explicitly in the structure at all.

It should be noted that this implicit encoding of the shifting operations is only possible because the basic translation of the word is always a subterm of the shifted term. Suppose that there were a shifting relation for which the subterm property did not hold. In that case, the input of this shifting relation would not be a component of the output. The output appears as the CONTENT value of the *word* described in the lexical entry, but the input does not appear as the value of any attribute in the word. Therefore, the basic translation of the word is not a component of the word. In the lexical entry, however, we require that there be a component $\boxed{1}$ of the described object that is the basic translation of this word.

Now that we have presented two different possibilities for integrating shifting operations at the word level, the question arises which of these alternative encodings is to be preferred. Prima facie, both encodings express the restriction to words, because both rely on mechanisms that are only available for words, be it DRs or lexical entries. Still, the approaches lead to different overall expectations. In the following, we will discuss three of them.

In the case of the DR-encoding, a word is seen as an object of sort *word*, whether or not it is derived. As DRs can only take words as their input, it follows without further stipulation that shifting operations cannot be applied to phrases. On the other hand, we expect that shifting operations can be applied to the output of DRs. In Section 4.3 this expectation will be confirmed, i.e., we will show that under our analysis of passive and complement extraction as developed in Section 2.3.2 we must apply shifting to the output of DRs to derive some readings.

The LE-encoding which we have exemplified in the lexical entry of *loves* in (173) leads us to different expectations. this approach locates the applicability of AR and VR in the lexical entries. Therefore, we might expect that these rules cannot be applied to the output of derivational rules, such as passivized predicates or lexical heads that miss complements. We will show in Section 4.3 that this restriction cannot be maintained. While we can still rescue this approach by changing the specification of the passive DR and the complement extraction DR slightly to allow for the application of AR and VR on their output as well, this shows that the second approach might not locate the restriction on the applicability of shifting operations correctly.

On the other hand, in the second part of this thesis, we will introduce lexical entries for irregular phrases, such as idiomatic expressions of the kind *kick the bucket* and *trip the light fantastic*. For phrases that are licensed by a lexical entry, shifting operations must be available as well. This shows that lexicalized phrases behave just like words with respect to shifting operations. On the other hand, lexicalized phrases do not undergo derivational rules such as the DRs for passivization or complement extraction. This suggests that shifting operations have a different status in the grammar than derivational rules.

There is a further property of shifting operations that points to a difference between DRs and shifting operations. Treating AR and VR as derivational rules we might expect to find words where the application of a shifting operation has a reflect in the other properties of the word, such as its phonological or syntactic properties. This is, however, not the case. Instead, the shifting operations exclusively operate on the logical form of a sign, and we must specify the shifting DRs so that all the remaining information is taken over from the input to the output. Under the LE-encoding of shifting operations, we do not expect the application of a shifting operation to have any consequences on the other properties of a sign. As this is true, the LE-encoding imposes the right restrictions in this respect.

Finally, a DR- and a LE-encoding lead to different expectations with respect to the regularity of the availability of shifting operations. The two DRs given in (171) are applicable to all words, i.e., we do not expect the existence of words to which these shifting operations are not applicable. In the case of exceptions, i.e., words that do not allow any or at least not the full range of shifting operations, we expect the exceptions to be systematic, i.e., to be restricted to natural classes. Such a restriction to natural classes is necessary for delimiting the range of the DR.

In contrast to this, under the LE-encoding, we must mention the applicability of shifting operations explicitly in every lexical entry. Therefore, we expect that there might be lexical entries that express the availability of shifting, and others that do not. In particular, it would be quite natural if the restrictions to the applicability of shifting operations were unsystematic, i.e., if they involved arbitrary, unpredictable elements in the language.

While this seems to be a promising and empirically testable difference, the data is not all that clear. For example, it seems desirable to exclude complementizers from shifting operations. We will show this for the complementizer *that* and for the phonological empty lexical head of a relative clause, as it is assumed in the analysis of Pollard and Sag 1994. As both examples are instances of functional words, it might be possible to generalize the restriction to a natural class. In this case, the data would not decide between the two alternative encodings.

Let us first address the case of the complementizer *that*. In our introduction of the framework LF-Ty2 in Section 1.3.3, we assumed a basic translation of the complementizer as given in (176). We mentioned that the intuition behind this translation was that the complementizer *that* expresses an identity function.

(176) *that* $\leadsto \lambda p_{st}.p(@)$

In our derivation of the *de re* readings of sentence (39c), repeated in (177), we saw that the type of the embedded clause is changed to enable a wide scope reading for some quantifier that occurs inside the *that*-clause.

(177) Every man believes [$_{\bar{S}}$that [$_{S}$some woman walks]].

a. *de dicto*:
$\forall x_{se}[\mathsf{man}'_{@}(x_{@}) \rightarrow \mathsf{believe}'_{@}(x_{@}, \lambda @.\exists y_{se}[\mathsf{woman}'_{@}(y_{@}) \wedge \mathsf{walk}'_{@}(y_{@}))]]$

b. $\forall\exists$-*de re*:
$\forall x_{se}[\mathsf{man}'_{@}(x_{@}) \rightarrow \exists y_{se}[\mathsf{woman}'_{@}(y_{@}) \wedge \mathsf{believe}'_{@}(x_{@}, \lambda @.\mathsf{walk}'_{@}(y_{@}))]]$

c. $\exists\forall$-*de re*:
$\exists y_{se}[\mathsf{woman}'_{@}(y_{@}) \wedge \forall x_{se}[\mathsf{man}'_{@}(x_{@}) \rightarrow \mathsf{believe}'_{@}(x_{@}, \lambda @.\mathsf{walk}'_{@}(y_{@}))]]$

The derivations of the three readings were given in Figures 1.11, 1.14 and 1.15. For all readings, we assumed that the logical form of the $\bar{\text{S}}$ node in (177) is identical to that of the embedded S node. To achieve this in the case of the two *de re* readings, we applied the same shifting rules to the complementizer that we applied to the main verb of the embedded clause.

Using the HPSG integration of LF-Ty2, there is a simpler way to treat the complementizer *that*. In the syntactic analysis of the complementizer, following Pollard and Sag 1994, we treated *that* as a marker. As such, it selects the clause that it attaches to by an attribute SPEC. The value of the SPEC attribute is a *synsem* object. In particular, it is exactly that *synsem* object that occurs as the SYNSEM value of the clause that the marker combines with. As the logical form of the clause is part of this *synsem* object, it is locally available in the lexical entry of the marker. We can use this information to give the marker a logical form which expresses the identity function under intensional functional application for an arbitrary type.

In (178a) we give this idea in informal terms. In (178b), we incorporate this new logical form into the lexical entry of the complementizer.

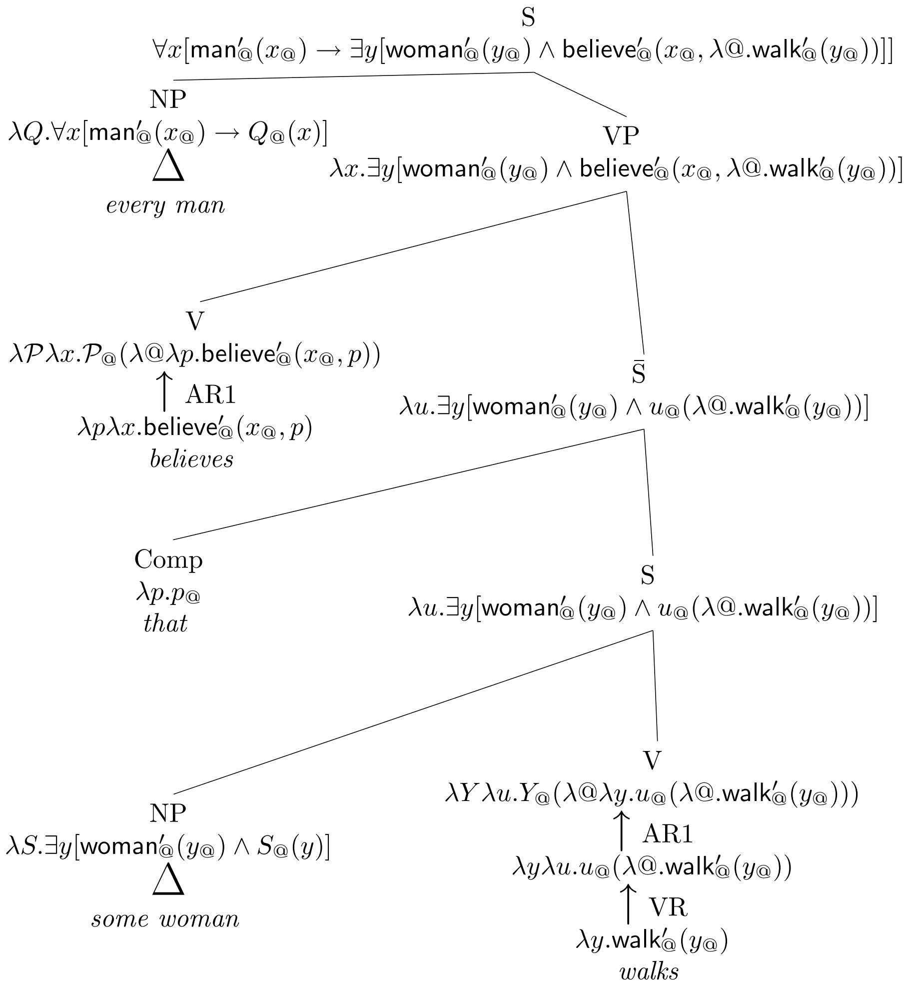
(178) a. New basic translation of the complementizer *that*:
that $\leadsto \lambda p_{s\tau}.p(@)$,
where τ is the LOCAL CONT TYPE value of the SPEC value of the complementizer.

b. Revised lexical entry for the complementizer *that*:

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{that} \rangle \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{marker} & \\ \text{SPEC} & \begin{bmatrix} \textit{synsem} & \\ \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HD} & \textit{verb} \\ \text{SBC} & \langle\rangle \\ \text{MKG} & \textit{unmarked} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{TYPE} & \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{SUBCAT} & \textit{elist} \\ \text{MARKING} & \textit{that} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \textit{abstr} & \\ \text{VAR} & \boxed{2} \\ \text{ARG} & \begin{bmatrix} \textit{appl} & \\ \text{TYPE} & \boxed{1} \\ \text{FUNC} & \boxed{2} \\ \text{ARG} & @ \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{NONL} & \begin{bmatrix} \text{INHER SLASH} & \textit{eset} \\ \text{TO-BIND SLASH} & \textit{eset} \end{bmatrix} \end{bmatrix} \\ \text{STORE} & \textit{elist} \end{bmatrix}$$

In the lexical entry, it suffices to specify that the type of the application $p(@)$ is identical to the type of the clause that the marker attaches to. Given that this is some type τ, it

FIGURE 4.2. The derivation of the ∀∃-reading of sentence (177):



follows from the restrictions on the TYPE values of *appl* objects and *abstr* objects (given in the TRP in (122)) that the variable p is of type $\langle s, \tau \rangle$.

With this lexical entry, the derivation of the *de re* readings is simplified in that we do not need to assume any shifting operations to be applied to the complementizer. In Figure 4.2 (page 156) we show the derivation of the reading in (177b). The tree in this figure should be compared with that given for the same reading in Figure 1.14.

The new lexical entry shows that we can simplify our analysis of the marker considerably, if we use the possibility of not fixing the semantic type of the marker in the lexical entry. As a consequence, there is, strictly speaking, not a single basic translation for the marker, but it is assigned a family of basic translations. This, however, makes it unnecessary to apply shifting rules to the complementizer *that*. This indicates that the HPSG encoding of LF-Ty2 can differ from the original proposal, as it may use mechanisms that are available to HPSG, such as underspecification in the lexicon, which are not strictly part of LF-Ty2.

Let us turn again to what the modified lexical entry of the complementizer *that* means for the two alternative encodings of shifting. For the LE-encoding, it is just what we would expect to find: the lexical entry is simply as given in (178b), i.e., without allowing the application of shifting operations.

For the DR-encoding, we could also assume the lexical entry just in (178b), but would be possible to apply shifting DRs to words licensed by this lexical entry. This does no harm, when it comes to the available readings, but the original intention behind the more flexible lexical entry of the marker would be lost. Alternatively, of course, we could specify the input conditions of the shifting DRs in such a way as to exclude the marker *that*. This is possible, because we can add to the input conditions of the DRs in (171) that the input word must not have a MARKING value of sort *that*.

While the case of the marker *that* does not create a decisive argument for or against one of the proposed encodings, it seems to us that the LE-encoding captures the generalization about the semantic contribution of the marker more directly and simplifies the derivations considerably. The argument is, however, relatively weak, because even under the assumption of a fixed basic translation for the complementizer of the form $\lambda p_{st}.p(@)$ as given in Table 1.1 and free application of shifting operations, all the readings can be derive. Therefore, the point that has been made here is purely aesthetic.

Matters change, if we turn to a second example of a word for which we might want to disallow the application of shifting operations. Hendriks 1993 (pp. 107f.) discusses how scope islands can be accounted for within *Flexible Montague Grammar* (FMG). In Section 1.2 we have pointed out that the *Complex NP Constraint* (CNPC) seems to hold for quantifier scope as well as for *wh*-movement. In (179) we repeat the data from (25).

(179) a. Guinevere has $[_{NP}$ a bone in every corner of the house].
b. Guinevere has $[_{NP}$ a bone $[_{S}$ which is in every corner of the house]].

It was noted that sentence (179b) cannot have a reading in which the quantifier *every corner* takes scope over *a bone*, even if considering our world knowledge, this would be the only sensible reading of this sentence. This is treated as an instance of the CNPC, because the quantifier *every corner* cannot take its scope outside the clause in which it is contained.

For further discussion, let us consider the examples in (180a) which allow us to avoid speculations on the correct representation of prepositions. But, just as has been the case in example (179b), sentence (180a) only has a reading in which the quantifier *every* takes scope inside the relative clause, i.e., the reading in (b), whereas the (c)-reading is excluded. We indicate the unavailability of the wide-scope reading with the symbol “$”.

(180) a. Some native speaker who proofread every thesis, was exhausted.
b. $\exists x[\textsf{native-speaker}'_{@}(x_{@})$
$\wedge\forall y[\textsf{thesis}'_{@}(y_{@}) \rightarrow \textsf{proofread}'_{@}(x_{@}, y_{@})] \wedge \textsf{be-exhausted}'_{@}(x_{@})]$
c. \$ $\forall y[\textsf{thesis}'_{@}(y_{@})$
$\rightarrow \exists x[\textsf{native-speaker}'_{@}(x_{@}) \wedge \textsf{proofread}'_{@}(x_{@}, y_{@}) \wedge \textsf{be-exhausted}'_{@}(x_{@})]]$

Let us assume, following Pollard and Sag 1994 (Section 5.2.2), that there is a phonologically empty relativizer which occurs as the head of the relative clause in (179b). In (181) we give the necessary lexical entries for this empty relativizer (a) and for the relative pronoun *who* (b). These lexical entries differ from those given in Pollard and Sag 1994 in several respects: First, and most importantly, we gave the content specification in accordance with the framework of LF-Ty2, and not with the semantic analysis of Pollard and Sag 1994. Second, we only mention one aspect of the nonlocal part of the analysis. In the analysis in Pollard and Sag 1994 there are two nonlocal features involved: SLASH, as the relative pronoun is assumed to be extracted, and REL a feature which is used to provide the identity of indices between the relative constituent and the noun that is modified by the relative clause. In (181a) we give some SLASH specification, while we do not consider a REL specification.

(181) a. Sketch of the lexical entry of the phonologically empty relativizer, after Pollard and Sag 1994 (p. 218):

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \textit{elist} \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{rltvzr} \\ \text{MOD N}' \end{bmatrix} \\ \text{SUBCAT} & \left\langle \boxed{1}\text{NP, VP}\left[\text{SBC}\left\langle \boxed{1}\left[\text{LOC } \boxed{2}\right]\right\rangle\right] \right\rangle \\ \text{MARKING} & \textit{unmarked} \end{bmatrix} \\ \text{CONT} & \lambda P \lambda Q \lambda R \lambda x.[R_@(x) \wedge Q_@(x) \wedge P_@(x)] \end{bmatrix} \\ \text{NONL} & \begin{bmatrix} \text{INHER SLASH} & \left\{\boxed{2}\right\} \\ \text{TO-BIND SLASH} & \left\{\boxed{2}\right\} \end{bmatrix} \end{bmatrix} \\ \text{STORE} & \textit{elist} \end{bmatrix}$$

b. Sketch of the lexical entry for the relative pronoun *who*:

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{who} \rangle \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{SUBCAT} & \langle\rangle \\ \text{MARKING} & \textit{unmarked} \end{bmatrix} \\ \text{CONT} & \lambda P \lambda x.[\mathsf{human}'_@(x_@) \wedge P_@(x)] \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

The empty relativizer as described in (181a) is used for *wh*-relative clause where the relative pronoun is the subject of the relative clause. Following Pollard and Sag 1994, the relativizer has two elements on its SUBCAT list: the relative constituent as the first element, and a finite VP as the second element. The first element of the SUBCAT list is specified as identical to the missing subject of the second element (see the use of the tag $\boxed{1}$).

As we adopt the theory of extraction of Pollard and Sag 1994, we specify in the lexical entry of the empty relativizer that it has a non-empty INHERITED SLASH value. This set contains the *local* object which also appears as the LOC value of the finite VP that is the complement of the relativizer. In (181a), we use the tag $\boxed{2}$ for this *local* object. Since the "extracted" subject is realized as a complement of the relativizer, we must block the percolation of the INHERITED SLASH value. This is done by requiring the TO-BIND SLASH value of the relativizer to contain exactly this element. Then, the NONLOCAL FEATURE PRINCIPLE given in (105) ensures that, given the nonlocal specification as in (181a), the INHERITED SLASH value of a relative clause is the empty set.

While our adaptation of the empty relativizer preserves the SLASH specifications of Pollard and Sag 1994, we do not mention the second nonlocal feature used in the analysis of Pollard and Sag 1994, REL. The REL specification seems to be unnecessary in our account, because of the way the logical form of the relativizer interacts with that of the N′ to which the relative clause attaches. This will be exemplified in the derivation of the reading in (180b) in Figure 4.3 (page 160).

In the lexical entry for the relativizer, we assert that its basic translation is the term $\lambda P \lambda Q \lambda R \lambda x.[R_@(x) \wedge Q_@(x) \wedge P_@(x)]$. The relativizer has two elements on its SUBCAT list, the rest of the relative clause, and the relative pronoun. These two "complements" correspond to the first two semantic arguments in this term. Once the relativizer has combined with its complements, the logical form of the relative clause is of the form $\lambda R_{s((se)t)} \lambda x_{se}.[R_@(x) \wedge \phi]$. As such, it is just like the logical form of an intersective adjective, as given for *good* in (144), i.e., $\lambda P_{s((se)t)} \lambda x_{se}[P_@(x) \wedge \mathsf{good}'_@(x_@)]$.

The basic translation of the relative pronoun *who* is the term $\lambda P \lambda x.[\mathsf{human}'_@(x_@) \wedge P_@(x)]$, i.e., a translation which is just like that of the quantifier *someone* but where the variable x is not bound by a quantifier, but by a lambda operator. Thus, the relative pronoun merely contributes the restriction to humans.[1]

In Figure 4.3 (page 160) we show the structure of relative clause of sentence (180a) and the derivation of the narrow scope reading of the quantifier *every*. In this derivation, we use the fully λ-converted forms of the terms, just as we did in the structures in Section 1.3.3. Note that in the derivation of the (180b) reading, we only applied the shifting rule AR to the first semantic argument of the embedded verb.

To block quantifiers which are inside the relative clause from taking scope over matrix material, it suffices to block the application of shifting operations to the relativizer. As a consequence, the logical form of the RelP will always be of the form $\lambda R_{s((se)t)} \lambda x_{se}.[R_{@}(x) \wedge \phi]$. As mentioned above, this term is of exactly the same type as the basic translation for other nominal modifiers. If we combine it with the basic translation of a noun, which usually is of type $(se)t$, we get a logical form of the form $\lambda x_{se}.\phi'_t$ If this term combines via intensional functional application with some other term ψ of type $s(((se)t)\tau)$, such as a quantifier, for example $\lambda P \lambda Q.\exists x[P_{@}(x) \wedge Q_{@}(x)]$, then it is excluded that a quantifier which occurs in ψ ends up in the scope of a quantifier that occurs in ϕ'.

If we do not exclude the application of shifting operations to the relativizer, the empirically unavailable reading in (180c) can be derived. In Figure 4.4 (page 161) we give the hypothetical structure of the Rel′ as it would occur in the derivation of this reading. In Figure 4.5 (page 162), we add the rest of the structure to form the entire NP *some native speaker who ...*. In the following, we will briefly comment on this derivation. In order to derive the reading in (180c), we would be forced to use the shifting operations to the basic translation of the empty relativizer.

To see how this derivation works, we should briefly consider the shifting operations that are applied. Let us start with the verb *proofread*. We first apply VR to the basic translation of this verb. Note, however, that in contrast to the previous examples, we do not choose the newly introduced variable u to be of type $s((st)t)$, but instead we assume it to be of type $s((s((se)t))t)$. This is the case, because a relative clause as a whole normally is of the semantic type $(s((se)t))((se)t)$. The variable u, then "indicates" the place where the semantic contribution of the rest of the NP is supposed to occur. Next, we must apply AR1 to the first semantic argument. This is done, because in the reading which we want to derive, the universal quantifier that occurs inside the relative clause is supposed to take scope outside the relative clause. After the application of AR1, the bound occurrence of the variable u is in the scope of the universal quantifier. This becomes clear at the logical form of the VP node.

We also apply shifting operations to the relativizer. In fact, just like with the verb, we first apply VR to mark the position of the matrix quantifier *some*, then we apply AR to the first semantic argument. Note that as the first argument of the relativizer is of sort $s((se)t)$, the raised form of this argument, the variable $\mathcal{P}$ is of sort $s((s((s((se)t))t))t)$. As the variable u is of type $s((s((se)t))t)$, prefixing the logical form of the VP node with the lambda abstractor $\lambda @$ results in a term which is of the same type as the variable $\mathcal{P}$. In the logical form of the Rel′ node, the universal quantifier now outscopes the semantic contribution of the relativizer, among which we also find the occurrence of the variable v.

Nothing interesting happens at the RelP node and at the N′ node (shown in Figure 4.5 on page 162). At the determiner, *some*, however, we also must apply AR. This is necessary because it must take the logical form of the N′ node as its semantic argument. The constellation of types is the same as in the local tree Rel′, where the relativizer was combined with the VP. The combination results in a logical form in which the universal quantifier

[1]In the case of more complex relative constituent, such as *whose wife*, the semantic contribution of the relative clause is more complex as well.

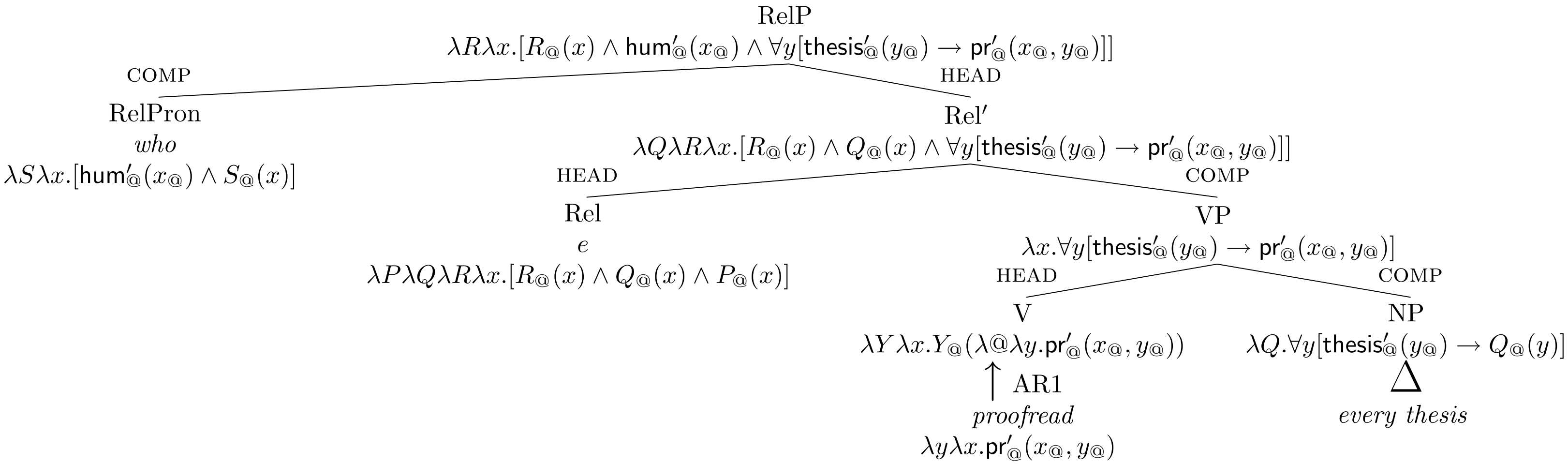


FIGURE 4.3. The structure of the narrow scope reading of sentence (180a):

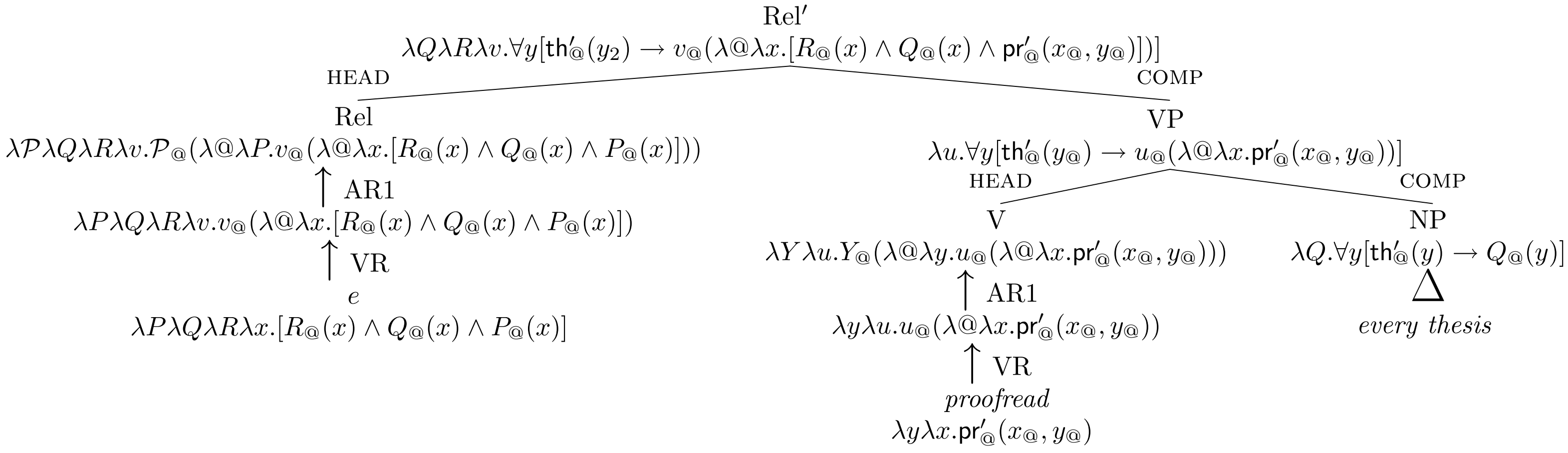


FIGURE 4.4. The structure of the Rel′ under a wide scope reading of sentence (180a):

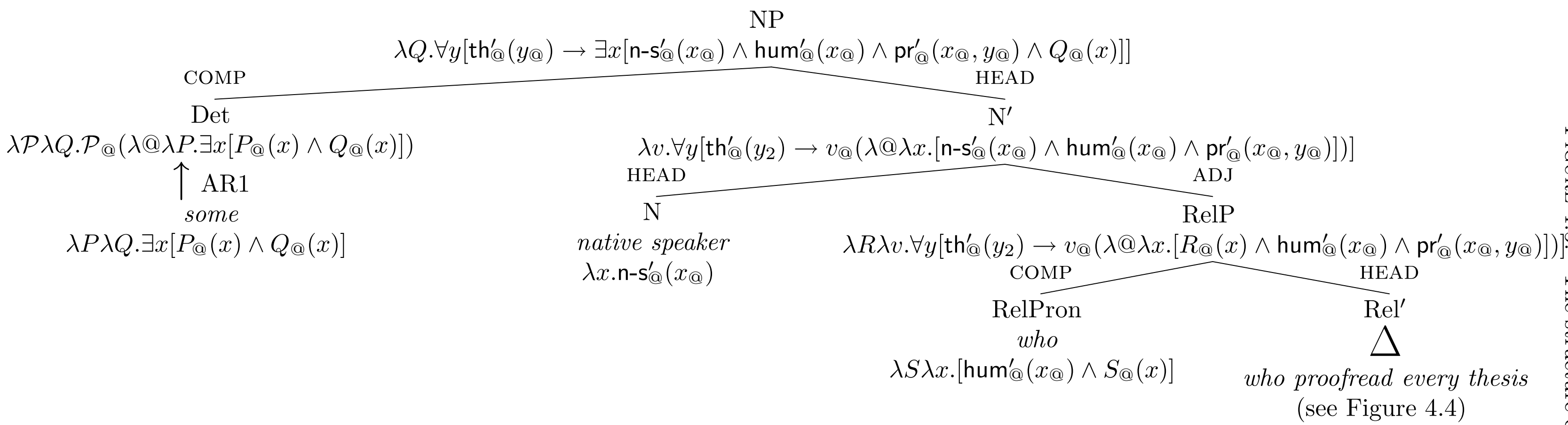


FIGURE 4.5. The structure of the wide scope reading of sentence (180a):

that was introduced inside the relative clause has scope over the existential quantifier that appeared outside the relative clause.

To sum up the discussion, we have shown that we can incorporate the CNPC into LF-Ty2 if we assume that the empty relativizer cannot undergo type shifting operations. In contrast to the case of the complementizer *that*, this restriction is not motivated on purely aesthetic grounds, it is crucial to implement the CNPC within the framework of LF-Ty2. This means that we need this restriction to express one of the major restrictions on scope which have been discussed in the literature.

Let us, finally turn back to the question whether the data in (179) can be used to decide between the DR- and the LE-encoding of shifting rules. If our grammar was enriched to contain more types of relative clauses, we might need several relativizers as well. The CNPC is, however, valid for each of them. In that respect, the non-applicability of shifting operations is valid for some natural class, i.e., all words with a HEAD value of the sort *relativizer*. Under the DR-encoding, we would, be forced to exclude all words with such a head value from meeting the input conditions of the DRs. This can easily be done by specifying the input as having a HEAD value different from *relativizer*. In the LE-encoding, the lexical entries of the relativizer(s) would simply lack the option of applying shifting operations to the basic translation.

Again, both approaches are equally fit to incorporate the non-applicability of shifting operations to relativizers. Nonetheless, our discussion of the complementizer *that* and of the relativizer has revealed two things. First, we have seen that there may be conceptual and empirical reasons to block the applicability of shifting operations from some words. Second, if we want to decide between the two encodings on the ground of empirical reasoning, we must discuss whether the words which cannot undergo shifting operations form a natural class. At this point, we cannot come to a definitive conclusion.

In this thesis, we will favor the LE-encoding, because we find it useful to make a clear distinction between shifting operations which only operate on the semantic representation and DRs which potentially change all properties of a word. Furthermore, we will include lexical entries for irregular phrases such as *kick the bucket* and *trip the light fantastic* in Chapter 8. As mentioned above, we will apply shifting operations to phrases that are licensed by such lexical entries, we will, however, not apply derivational rules to them.

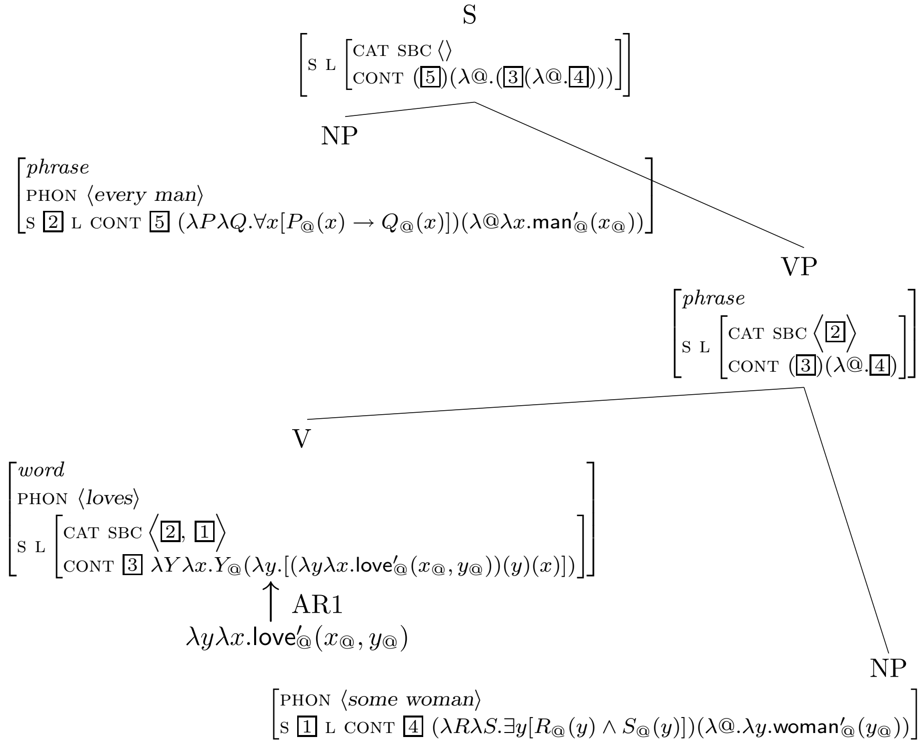
In the tree structures in this thesis, we will use the symbol "↑" to indicate the application of shifting operations, just as we did in Section 1.3.3. Note that the symbol "↑" is distinct from the symbol "$\underline{\uparrow}$" which we have introduced in Section 2.3.2 for the application of derivational rules. In Figure 4.6 contains an example structure for the ∀∃-reading of sentence (39b), repeated for convenience in (182).

(182) Every man loves some woman.
∀∃: $\forall x_{se}[\mathsf{man}'_@(x_@) \rightarrow \exists y_{se}[\mathsf{woman}'_@(x_@) \wedge \mathsf{love}'_@(x_@, y_@)]]$
∃∀: $\exists y_{se}[\mathsf{woman}'_@(x_@) \wedge \forall x_{se}[\mathsf{man}'_@(x_@) \rightarrow \mathsf{love}'_@(x_@, y_@)]]$

In the tree in Figure 4.6 we have applied AR1 to the basic translation of the verb *love*. Depending on whether this tree is interpreted under a DR-encoding or under a LE-encoding of shifting operations, the "↑AR1 " arrow means different things. In the case of a DR-encoding, everything below the label V in the structure is an abbreviation of the description in (172). The description below the arrow is understood as the CONTENT value of the input word of the DR, the AVM above the arrow is a description of the output word.

Under the perspective of a LE-encoding, the AVM above the "↑AR1 " arrow describes a word as licensed by a lexical entry. The term below the arrow is merely indicates how the CONTENT value of this word was derived starting from the basic translation. The structures

FIGURE 4.6. The structure of sentence (158) in the ∀∃-reading:



that we are going to draw in the following sections are, thus, neutral with respect to the particular encoding chosen for shifting operations.

In this section, we have provided a first encoding of LF-Ty2 in RSRL. To do this, we integrated the basic translation of a word in its lexical entry, we incorporated intensional functional application into the SEMANTICS PRINCIPLE, and, finally, we provided two alternative encodings for the application of shifting operations. The CONTENT values that resulted from this first implementation of LF-Ty2 as the semantic part of the grammar proposed in Section 2.3, have the same denotation with respect to a semantic model as the terms given in our original presentation of LF-Ty2. The terms do, however, differ syntactically, as they are not λ-converted. We think that it is motivated to assume λ-conversion to be part of the semantic system, i.e., we will assume a principle that states that the logical form of every sign is such that it is *redex-free*, i.e., that it is closed under the application of λ-conversion. The next section is devoted to the question of howλ-conversion can best be integrated into the given system. In Section 4.3 we will extend our semantic analysis to passive and complement extraction.

4.2. FORMALIZATION OF λ-CONVERSION

The semantic system developped in the previous section is a direct implementation of LF-Ty2 as presented in Section 1.3, i.e., we used the same basic translations, the same shifting operations, and intensional functional application as the only operation available at branching nodes. Still, the resulting terms differed from those given in Section 1.3. The reason for this difference is that we had not introduced λ-conversion into the RSRL encoding of LF-Ty2, while we used λ-conversion freely in Section 1.3. In this section, we will ultimately provide the means to express λ-conversion formally within RSRL. Before we turn to such a formalization, however, we will give some reasons for why we consider it useful to integrate this operation into our grammar.

Logical forms to which λ-conversion cannot be applied are of great practical advantage. First the terms are shorter and, therefore, of a better readability. For illustration, consider the CONTENT value of the NP *every man* as given in Figure 4.1 and its λ-converted form as it is given at the NP node in Figure 1.10. In (183) these two terms are repeated.

(183) every man
  a. $(\lambda P \lambda Q.\forall x[P_{@}(x) \rightarrow Q_{@}(x)])(\lambda @ \lambda x.\mathsf{man}'_{@}(x_{@}))$
  b. $\lambda Q.\forall x[\mathsf{man}'_{@}(x_{@}) \wedge Q_{@}(x)]$

Second, scope relations among quantifiers can be read off directly, whereas they are harder to detect in a non-converted representation. To see this, reconsider the CONTENT value of sentence (182) under the ∀∃-reading given in Figure 4.6. In (184) we give the non-converted and the converted term.

(184) Every man loves some woman.
  a. $((\lambda P \lambda Q.\forall x[P_{@}(x) \rightarrow Q_{@}(x)])(\lambda @ \lambda x.\mathsf{man}'_{@}(x_{@})))$
  $\lambda @.((\lambda Y \lambda x.Y_{@}(\lambda y.[(\lambda y \lambda x.\mathsf{love}'_{@}(x_{@}, y_{@}))(y)(x)]))$
  $\lambda @.((\lambda R \lambda S.\exists y[R_{@}(y) \wedge S_{@}(y)])(\lambda @.\lambda y.\mathsf{woman}'_{@}(y_{@}))))$
  b. $\forall x_{se}[\mathsf{man}'_{@}(x_{@}) \rightarrow \exists y_{se}[\mathsf{woman}'_{@}(x_{@}) \wedge \mathsf{love}'_{@}(x_{@}, y_{@})]]$

In (b) it is easy to see that the existential quantifier is in the scope of the universal quantifier. This relative scope of the two quantifiers is not transparent in the term in (a).

We will see in Section 4.4 that it is necessary to express scope relations in RSRL if we want to formalize lf-constraints as the one in (53). If we assume fully λ-converted logical forms then the formulation of such constraints is straightforward. This means that it is reasonable from the point of view of grammar writing to include λ-conversion in the semantic system.

In addition to these practical considerations, there is also a conceptual advantage. In Section 1.3.3 we saw that there are in general, many distinct ways to derive a certain reading. This was illustrated with the simple sentence in (39a), repeated in (185).

(185) Every man walks.
  $\forall x[\mathsf{man}'_{@}(x_{@}) \rightarrow \mathsf{walk}'_{@}(x_{@})]$

As shown in Figure 1.10 on page 47 it is possible to derive this reading without applying any shifting operation. In the derivation in Figure 1.16 (page 56) an alternative derivation was presented, where we applied AR to the first semantic argument of the verb and, then, combined it with the quantified NP subject. The difference between the two derivations is that in the first case, the quantified NP is taken as the semantic functor which combines with the translation of the verb. In the second case, the basic translation of the verb is shifted via AR so that it becomes a functor which takes the translation of the quantified NP as its argument. Both derivations lead to the same reading, the reading given in (185).

If we do not apply λ-conversion then the two derivations lead to different logical forms, which have, however, identical extensions. In (186) we give the translation of the NP, which is the same in both derivations. As can be seen, the resulting term in (c) is the intensional functional application of the basic translation of *every* to the basic translation of *man*.

(186) Translation of the NP *every man*:
  a. Basic translation of *man*: *man* $\leadsto \lambda x.\mathsf{man}'_{@}(x_{@})$
  b. Basic translation of *every*: *every* $\leadsto \lambda P \lambda Q.\forall x[P_{@}(x) \rightarrow Q_{@}(x)]$
  c. Translation of *every man*: $(\lambda P \lambda Q.\forall x[P_{@}(x) \rightarrow Q_{@}(x)])(\lambda @ \lambda x.\mathsf{man}'_{@}(x_{@}))$

In (187) we show the translation of sentence (185) under the first derivation. As indicated in (b), the translation of the sentence is just the intensional functional application of the the translation of the NP to the basic translation of the verb. This means, that in this derivation, the quantified NP is the semantic functor.

(187) First derivation of a translation of sentence (185)
a. Basic translation of *walks*: *walks* $\leadsto \lambda x.\mathsf{walk}'_{@}(x_{@})$
b. Translation of the sentence:
$((\lambda P \lambda Q.\forall x[P_{@}(x) \rightarrow Q_{@}(x)])(\lambda @ \lambda x.\mathsf{man}'_{@}(x_{@})))(\lambda @ \lambda x.\mathsf{walk}'_{@}(x_{@}))$

In (188) the alternative derivation is show. Again we start from the basic translation of the verb in (a). In (b), we apply AR1 to the basic translation. The resulting term is, then, combined via intensional functional application with the translation of the NP. This time, it is the translation of the verb that is the semantic functor.

(188) Second derivation of a translation of sentence (185)
a. Basic translation of *walks*: *walks* $\leadsto \lambda x.\mathsf{walk}'_{@}(x_{@})$
b. The result of applying AR1: $\lambda Y.Y_{@}(\lambda @ \lambda y.[(\lambda x.\mathsf{walk}'_{@}(x_{@}))(y)])$
c. Translation of the sentence:
$(\lambda Y.Y_{@}(\lambda @ \lambda y.[(\lambda x.\mathsf{walk}'_{@}(x_{@}))(y)]))$
$(\lambda @(\lambda P \lambda Q.\forall x[P_{@}(x) \rightarrow Q_{@}(x)])(\lambda @ \lambda x.\mathsf{man}'_{@}(x_{@})))$

What is important here is that the translations differ depending on the derivation, i.e., we find distinct terms in (187b) and (188c). In the RSRL grammar presented so far, this means that the sentence has (at least) these two logical forms. This indicates that without λ-conversion, we are (i) confronted with hardly readable logical forms, and (ii) we also give rise to a number of distinct logical forms which do not reflect distinct interpretations.[2]

In the following two subsections, we will show two ways to integrate λ-conversion into the framework presented so far. As a result, we will posit the restriction that the logical form of a sign be always redex free, i.e., fully λ-converted.

The two encodings can be seen parallel to the two alternative encodings of shifting operations discussed in the last section. To indicate where the difference lies, consider the following step-by-step λ-conversion that leads from the term in (183a) to the term in (183b).

(189)
$$\begin{aligned} &\lambda P_{s((se)t)} \lambda Q_{s((se)t)}.\forall x_{se}[P_{@}(x) \rightarrow Q_{@}(x)](\lambda @ \lambda x_{se}.\mathsf{man}'_{@}(x_{@})) \\ &= \lambda Q_{s((se)t)}.\forall x_{se}[(\lambda @ \lambda x_{se}.\mathsf{man}'_{@}(x_{@}))(@)(x) \rightarrow Q_{@}(x)] \\ &= \lambda Q_{s((se)t)}.\forall x_{se}[(\lambda x_{se}.\mathsf{man}'_{@}(x_{@}))(x) \rightarrow Q_{@}(x)] \\ &= \lambda Q_{s((se)t)}.\forall x_{se}[\mathsf{man}'_{@}(x_{@}) \rightarrow Q_{@}(x)] \end{aligned}$$

The first formalization of λ-conversion includes the conversion steps into the structure of the linguistic sign, i.e., the linguistic sign will have all the four terms of (189) as components. This is similar to the DR-encoding of shifting operations, because there, the shifted and the non-shifted term were both explicitly present in the structure of a word.

In contrast to this, under the second formalization of λ-conversion, it will only be the last term in (189) that appears as a component of the NP *every man*. The initial term as well as the intermediate terms are not explicitly present in the NP. In this sense, this is parallel to the LE-encoding of shifting operations. There, the shifting steps were not explicitly part of the CONTENT value of a word.

There is, of course, an important difference between shifting operations and λ-conversion. The two shifting operations discussed in this thesis do not preserve the meaning of the

[2] Note that this is distinct from the spurious derivations discussed in Section 1.3.3. If a sentence has two distinct derivations which lead to the same reading, i.e., to the same logical form for the root node, then, still, the logical forms of the intermediate nodes differ in the two derivations.

input, but they contain the input term as a subterm in their output. In contrast to this, λ-conversion preserves the meaning of the input but does not contain the input as a subterm. This can be seen in the λ-conversion steps given in (189). There, all four terms have the same extension with respect to a semantic model. It is, however, not the case that a lower term contains a higher term as a subterm. This formal difference between shifting and λ-conversion requires completely different mechanisms of encoding.

In addition to this formal difference between shifting and λ-conversion, there is also a difference with respect to the applicability of these operations within linguistic signs. In LF-Ty2 we do not allow shifting operations to apply at the level of phrases. λ-conversion, on the other hand, was taken to be freely available whenever possible in our presentation of LF-Ty2 in Section 1.3. Because of this difference, we will not only find distinct mechanisms for the encoding of shifting and λ-conversion, but we will also integrate the two operations in different ways into the overall architecture of our grammar.

So far, we have presupposed some familiarity with the definition of λ-conversion. For a better understanding of the RSRL encodings of this operation, it is useful to state a standard definition as a point of reference. For this purpose. we adapt the definitions from Hindley and Seldin 1986 (pp. 5–11). We cannot follow their definitions completely because they discuss an untyped λ-calculus without constants. Since we use a typed representation language with constants, we make the necessary adaptations. There is one further caveat: Throughout this section we will only consider the definition of Ty2 from Section 1.3.2 and the grammar $\mathcal{TY}2$ of Section 3.1. In Appendix A.1.3 the additional definitions are given for the extended representation language which includes logical constants, and quantifiers.

For illustration, we will use the simple term in (190a), which is related to the term in (190b) by one conversion step.

(190) a. $(\lambda x_{se}.\mathsf{walk}'(@)(x(@)))(y_{se})$
b. $\mathsf{walk}'_{@}(y(@))$

The first definition of Hindley and Seldin 1986 is that of a subterm. The term in (190a) contains the subterms in (191).

(191) $(\lambda x_{se}.\mathsf{walk}'(@)(x(@)))(y_{se})$
$(\lambda x_{se}.\mathsf{walk}'(@)(x(@)))$
x_{se}
$\mathsf{walk}'(@)(x(@))$
$\mathsf{walk}'(@)$
$x(@)$
walk'
$@$
y_{se}

DEFINITION 4.1 *(Subterm)*
The relation ϕ is a subterm of ψ *is defined by induction on* ψ*, thus:*

ϕ is a subterm of ϕ
if ϕ is a subterm of some ϕ_1 or of some ϕ_2, then
ϕ is a subterm of $(\phi_1\phi_2)$,
if ϕ is a subterm of ϕ_1 or ϕ is some variable x, then
ϕ is a subterm of $(\lambda x.\phi_1)$, and
if ϕ is a subterm of some ϕ_1 or of some ϕ_2, then
ϕ is a subterm of $(\phi_1 = \phi_2)$.

It can be seen that the terms in (191) are exactly the subterms of the term in (190a). The term $(\lambda x.\mathsf{walk}'_{@}(x(@))(y)$ is a subterm of itself by virtue of the first line in the definition. In terms $(\lambda x_{se}.\mathsf{walk}'_{@}(x(@)))$ and y_{se} are subterms of the overall term according to the second line. The term x_{se} is a subterm of $(\lambda x_{se}.\mathsf{walk}'_{@}(x(@)))$ by virtue of the third line and as such also a subterm of the overall term. Similarly, the term $\mathsf{walk}'_{@}(x(@))$ is a subterm of the term in (190a). The further subterms of $\mathsf{walk}'_{@}(x(@))$ follow from the second line in the definition of subtermhood.

The next definition is that of the notion of *free* and *bound variables*.

DEFINITION 4.2 *(free and bound variables)*

An occurrence of a variable x is bound *in a term ϕ*

iff it is a subterm of some subterm of ϕ of the form $\lambda x.\phi'$,

otherwise, it is free.

If x has at least one free occurrence in ϕ, it is called a free variable of ϕ.

The set of all free variables of ϕ is called $FV(\phi)$.

In the term in (190a), the variable x has only bound occurrences, because it occurs twice in the subterm $\lambda x.\mathsf{walk}'_{@}(x(@))$. The variables @ and y, on the other hand are free in the term. In (192) we give the set of free variables of the term (190a).

(192) $FV((\lambda x.\mathsf{walk}'_{@}(x(@)))(y)) = \{@, y\}$

With this definition, we have the necessary prerequisites to define *substitution*. We write $[\psi/x]\phi$ for a term that is the result of substituting every free occurrence of the variable x in the term ϕ by the term ψ.

DEFINITION 4.3 *(Substitution)*

For each terms ϕ, ψ, and for each variable x, we define $[\psi/x]\phi$ by induction on ϕ, as

$$
\begin{aligned}
&[\psi/x]x = \psi,\\
&\quad [\psi/x]a = a, \text{ for each variable and constant } a \neq x,\\
&\quad [\psi/x](\phi_1\phi_2) = ([\psi/x]\phi_1[\psi/x]\phi_2),\\
&\quad [\psi/x](\phi_1 = \phi_2) = ([\psi/x]\phi_1 = [\psi/x]\phi_2),\\
&\quad [\psi/x](\lambda x.\phi) = \lambda x.\phi,\\
&\quad [\psi/x](\lambda y.\phi) = \lambda y.[\psi/x]\phi, \text{ if } y \neq x \text{ and } y \notin FV(\psi) \text{ or } x \notin FV(\phi),
\end{aligned}
$$

The first four cases of Definition 4.3 are straightforward. The last two cases involve substitution in abstractions and, therefore, we must be careful not to turn a free variable of ψ or ϕ into a bound variable by substitution. In (193) we illustrate the last two cases.

(193) a. $[y/x]\lambda x.\mathsf{walk}'_{@}(x(@)) = \lambda x.\mathsf{walk}'_{@}(x(@))$
b. $[z/x]\lambda y.\mathsf{love}'_{@}(y(@), x(@)) = \lambda y.\mathsf{love}'_{@}(y(@), z(@))$
c. $[y/x]\lambda y.\mathsf{walk}'_{@}(m) = \lambda y.\mathsf{walk}'_{@}(m)$

In (193a), there is no free occurrence of the variable x in the term $\lambda x.\mathsf{walk}'_{@}(x(@))$. Therefore, there is no occurrence that could be substituted by the term y. In the next example, the variable x occurs freely in the term $\mathsf{love}'_{@}(y(@), x(@))$. Therefore, we can substitute the variable z for x. Example (193c) is another instance of the last clause of Definition 4.3. In this example, the variable y occurs freely in ψ, but the variable x is not free in ϕ. Therefore, even if we carry out the substitution, the free variable y of ψ will not be bound, because there is no occurrence of x in ϕ that would be replaced by the term ψ.

Finally, consider example (194).

(194) $[y/x]\lambda y.\mathsf{love}'_{@}(y(@), x(@))$ undefined

According to Definition 4.3 the indicated substituton is not defined. In this term, y is a free variable in ψ and the variable x is free in ϕ, i.e., in the term $\lambda y.\mathsf{love}'_{@}(y_{@}, x_{@})$. If we substitute the variable x by the variable y, then this variable would be bound in the resulting term $\lambda y.\mathsf{love}'_{@}(y(@), y(@))$. To avoid this, substitution is undefined here.

The problematic situation illustrated with example (194) can be avoided if we change the bound variable. Such a change of a bound variable does not change the meaning of a term and should, therefore be possible. This change of a bound variable can be applied inside complex terms and can be iterated. In this case, it is called α-conversion. Note that α-conversion is an equivalence relation, i.e., it is transitive, reflexive and symmetric.

DEFINITION 4.4 *(α-conversion)*

Let ϕ be a term that contains an occurrence of a term $\lambda x.\psi$, and let $y \notin FV(\psi)$.

> *The act of replacing $\lambda x.\psi$ by*
> $\lambda y.[y/x]\psi$
> *is called* change of bound variable *in ϕ.*

For two terms ϕ, ψ,

> *ϕ α-converts to ψ (or $\phi \equiv_{\alpha} \psi$) iff*
> *ψ has been obtained from ϕ by a finite (perhaps empty) series of changes of bound variables.*

To return to our example, the term in (190a) α-converts to the term in (195).

(195) $(\lambda z.\mathsf{walk}'_{@}(z(@)))(y)$

As we have seen, the term in (190a) has a subterm of the form $\lambda x.\mathsf{walk}'_{@}(x(@))$. The variable z does not occur freely in the term $\mathsf{walk}'_{@}(x(@)$. If we replace x by z, the resulting term is $\lambda z.\mathsf{walk}'_{@}(z(@))$. If we substitute this term for the occurrence of the term $\lambda x.\mathsf{walk}'_{@}(x(@)$ in (190a), we get the term given in (195).

Finally, we can define λ-conversion, which is also referred to as β-reduction (Hindley and Seldin, 1986). In our example in (190) the first term is related to the second by λ-conversion/β-reduction. Generally speaking, a term of the form $(\lambda x.\phi)\psi$ can be reduced to a term $[\psi/x]\phi$. This reduction is expressed in Definition 4.5.

DEFINITION 4.5 *(β-reduction/λ-conversion)*

A β-redex *is a term of the form*

> $(\lambda x.\phi)\psi$
> *and its* contractum *is the corresponding term*
> $[\psi/x]\phi$

A term χ β-contracts *to a term χ' (or $\chi \rightarrow_{1\beta} \chi'$) iff*

> *χ contains an occurrence of $(\lambda x.\phi)\psi$, and we replace that occurrence by $[\psi/x]\phi$, such that the result is χ'.*

A term χ β-reduces/λ-converts *to a term χ' (or $\chi \rightarrow_{\lambda} \chi'$) iff*

> *χ' is obtained from χ by a finite (perhaps empty) series of β-contractions and changes of bound variables.*

With this definition, we can show that the terms in (190) are related by β-contraction. The term in (190a) is of the form of a β-redex. The term in (190b) is the result of substituting the free occurrences of the variable x in the term $\mathsf{walk}'_@(x(@))$ by the term y:

(196) $$(\lambda x.\mathsf{walk}'_@(x(@)))(y) \quad \rightarrow_{1\beta} \quad [y/x]\mathsf{walk}'_@(x(@)) \\ = \quad \mathsf{walk}'_@(y(@))$$

We can, now, return to the example in (194). There, we saw that substitution is not defined in all cases. As β-contraction uses substitution, this means that some β-redices do not have a contractum. In (197) we give an example of this.

(197) $(\lambda x \lambda y.\mathsf{love}'_@(y(@), x(@)))(y)$

The contractum of this term would be of the form $[y/x]\lambda y.\mathsf{love}'_@(y(@), x(@))$. But this is exactly (194), i.e., a situation where substitution is not defined.

While we cannot contract the term in (197) directly, we can still β-reduce it. For this purpose, we first change the bound variable y to z. For the resulting β-redex, the contractum is defined. In (198) we give the reduction steps.

(198) $$\begin{array}{rcl} (\lambda x \lambda y.\mathsf{love}'_@(y(@), x(@)))(y) & \equiv_\alpha & (\lambda x \lambda z.\mathsf{love}'_@(z(@), x(@)))(y) \\ & \rightarrow_{1\beta} & [y/x]\lambda z.\mathsf{love}'_@(z(@), x(@)) \\ & = & \lambda z.[y/x]\mathsf{love}'_@(z(@), x(@)) \\ & = & \lambda z.\mathsf{love}'_@(z(@), y(@)) \end{array}$$

With the definition of β-reduction, we have established all the necessary ingredients for the RSRL encoding of λ-conversion. In our definitions, we will rely heavily on a relation `replace` which has the effect of substitution as given in Definition 4.3. Using this relation, we will provide an encoding of α-conversion and β-reduction.

4.2.1. First Alternative. As pointed out in the introduction to this section, the first encoding of λ-conversion assumes that the conversion steps are part of the linguistic sign proper.[3] In this subsection, we will first introduce a new sort, *reduction* which will be used in the encoding of the conversion steps. In Section 4.2.1.1 we will present this sort and the constraints that are necessary for this encoding of λ-conversion. In Section 4.2.1.2, we will, then, show how objects of the sort *reduction* can be integrated into linguistic signs to make fully reduced logical forms available in our grammar.

4.2.1.1. *Formalization.* We can now proceed with the RSRL rendering of the definitions given above. In the particular formalization that we will give in this subsection, some of the notions defined above will be expressed as relations, such as *subterm*, *free variables* and *substitution*. For others, *change of bound variable* and *β-contraction* and *β-reduction*, we will introduce new sorts in the signature.

For illustration, we will use again the terms in (190). In (199), we give an AVM description of these terms. In this description, we take the variable x to be $v_{se,0}$, the variable y to be $v_{se,1}$, and following our conventions, the variable @ to be $v_{s,0}$.

[3]The encoding proposed in this paragraph is based on the one given in Richter and Sailer 1999a (pp. 285–288). A similar encoding is also presented in Richter 2000 (pp 363–369).

(199) a. AVM description of the term $(\lambda x.\mathsf{walk}'(@)(x(@)))(y)$:

$$\begin{bmatrix} \textit{appl} & \\ \text{FUNC} & \begin{bmatrix} \textit{abstr} & \\ \text{VAR} & \boxed{1}\begin{bmatrix}\textit{var} & \\ \text{TYPE} & \boxed{2}\ \langle s,e\rangle \\ \text{NUM} & \textit{zero}\end{bmatrix} \\ \text{ARG} & \begin{bmatrix} \textit{appl} & \\ \text{TYPE} & \boxed{5}\ \textit{truth} \\ \text{FUNC} & \boxed{3}\ \mathsf{walk}'_{s(et)}(@) \\ \text{ARG} & \begin{bmatrix}\textit{appl} & \\ \text{FUNC} & \boxed{1} \\ \text{ARG} & \boxed{4}\begin{bmatrix}\textit{var} & \\ \text{TYPE} & \textit{w-index}\\ \text{NUM} & \textit{zero}\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix} \\ \text{ARG} & \boxed{6}\begin{bmatrix}\textit{var} & \\ \text{TYPE} & \boxed{2} \\ \text{NUM} & \begin{bmatrix}\text{NUM} & \textit{zero}\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

b. AVM description of the term $\mathsf{walk}'_{@}(@)(y(@))$:

$$\begin{bmatrix} \textit{appl} & \\ \text{TYPE} & \boxed{5}\ \textit{truth} \\ \text{FUNC} & \boxed{3}\ \mathsf{walk}'_{s(et)}(@) \\ \text{ARG} & \begin{bmatrix}\textit{appl} & \\ \text{FUNC} & \boxed{6}\begin{bmatrix}\textit{var} & \\ \text{TYPE} & \langle s,e\rangle \\ \text{NUM} & \begin{bmatrix}\text{NUM} & \textit{zero}\end{bmatrix}\end{bmatrix} \\ \text{ARG} & \boxed{4}\begin{bmatrix}\textit{var} & \\ \text{TYPE} & \textit{w-index} \\ \text{NUM} & \textit{zero}\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

In Definition 4.1 we defined the notion *subterm*. In RSRL, we can define a simple relation `subterm` that uses the relation `component` as defined in (117) or in (129). One *me* object is a subterm of a second iff it is a component of the second. This is expressed in the definition of the relation `subterm` in (200).

(200) The relation `subterm`:

$$\texttt{subterm}(\boxed{1},\boxed{2}) \stackrel{\forall}{\Longleftarrow} \left(\begin{array}{l} \boxed{1}\begin{bmatrix}\textit{me}\end{bmatrix} \textbf{ and } \boxed{2}\begin{bmatrix}\textit{me}\end{bmatrix} \\ \textbf{and } \texttt{component}(\boxed{1},\boxed{2}) \end{array} \right)$$

In Definition 4.2, we defined what a free occurrence of a variable in a term is. In (201) we define a relation that holds of a pair consisting of a *var* object v and a *me* object u iff v has a free occurrence in u.

(201) The relation `free-variable`:

$$\begin{array}{l} \texttt{free-variable}(\boxed{1},\boxed{2}) \stackrel{\forall}{\Longleftarrow} \\ \quad \boxed{1}\begin{bmatrix}\textit{var}\end{bmatrix} \\ \quad \textbf{and } \texttt{are-copies}(\boxed{1},\boxed{2}) \\ \texttt{free-variable}(\boxed{1},\boxed{2}) \stackrel{\forall}{\Longleftarrow} \\ \quad \boxed{2}\begin{bmatrix}\textit{appl} & \\ \text{FUNC} & \boxed{3} \\ \text{ARG} & \boxed{4}\end{bmatrix} \\ \quad \textbf{and } (\ \texttt{free-variable}(\boxed{1},\boxed{3}) \textbf{ or } \texttt{free-variable}(\boxed{1},\boxed{4})\ ) \end{array}$$

$\texttt{free-variable}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow}$
$\boxed{2}\begin{bmatrix} abstr \\ \text{VAR } \boxed{3} \\ \text{ARG } \boxed{4} \end{bmatrix}$
and (**not** $\texttt{are-copies}(\boxed{1},\boxed{3})$)
and $\texttt{free-variable}(\boxed{1},\boxed{4})$

$\texttt{free-variable}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow}$
$\boxed{2}\begin{bmatrix} equ \\ \text{ARG1 } \boxed{3} \\ \text{ARG2 } \boxed{4} \end{bmatrix}$
and ($\texttt{free-variable}(\boxed{1},\boxed{3})$ **or** $\texttt{free-variable}(\boxed{1},\boxed{4})$)

The definition in (201) follows the recursive structure of *me* objects. A pair $\langle v, o\rangle$, where v is a *var* object and o is a *me* object, is in interpretation of the relation `free-variable` iff one of the following four conditions is met: o is a copy of v (clause 1); or o is of sort *appl* and v is free in o's FUNC or in o's ARG value (clause 2); or o is of sort *abstr*, v is not a copy of o's VAR value and v is free in o's ARG value; or, finally, o is of sort *equ* and v is free in o's ARG1 value or in o's ARG2 value.

As a consequence of this definition, the pair $\langle v, o\rangle$ is in the interpretation of the relation `free-variable` iff the variable that corresponds to v has a free occurrence in the term that corresponds to o. According to this definition, the variables referred to by the tags $\boxed{4}$ (@) and $\boxed{6}$ (y) occur freely in the term described in (199a).

Having a notion of free variables defined for *me* objects, we can define a relation `replace` which encodes the effect of substitution as given in Definition 4.3. The relation `replace` has four arguments. In (202) we sketch the intuitive connection between the four arguments of the relation and the substitution operation.

(202) Intuitive characterization of the relation `replace`:
$\langle m, x, p, n\rangle \in \mathsf{R}(\texttt{replace})$ iff $SR([p]) = [SR([n])/SR([x])]SR([m])$

This intuitive characterization states that a quadruple $\langle m, x, p, n\rangle$ is in the interpretation of the relation `replace` iff their corresponding terms M, X, P, N are such that P is the result of substituting every free occurrence of X in M by N, i.e., P is $[N/X]M$.

In (203) we give the definition of the relation `replace` as it occurs in the theory of our grammar. The clauses of the definition correspond directly to the lines in Definition 4.3. For the sake of clarity, we state the corresponding line always right above the clauses.

(203) The relation `replace`:

$[\psi/x]x = \psi$
$\texttt{replace}(x, y, v, w) \overset{\forall}{\Longleftarrow}$
${}^{x}\big[var\big]$
$x \approx y$ **and** $v \approx w$

$[\psi/x]a = a$, for each variable and constant $a \neq x$
$\texttt{replace}(x, y, v, w) \overset{\forall}{\Longleftarrow}$
$\Big({}^{x}\big[var\big]$ **or** ${}^{x}\big[const\big] \Big)$
and $x \approx v$

$[\psi/x](\phi_1\phi_2) = ([\psi/x]\phi_1[\psi/x]\phi_2)$

$$\texttt{replace}(x,y,v,w) \overset{\forall}{\Longleftarrow}$$
$$x\begin{bmatrix} \textit{appl} & \\ \text{TYPE} & \boxed{1} \\ \text{FUNC} & \boxed{2} \\ \text{ARG} & \boxed{3} \end{bmatrix} \textbf{ and } v\begin{bmatrix} \textit{appl} & \\ \text{TYPE} & \boxed{1} \\ \text{FUNC} & \boxed{4} \\ \text{ARG} & \boxed{5} \end{bmatrix}$$
$$\textbf{and } \texttt{replace}(\boxed{2}, y, \boxed{4}, w) \textbf{ and } \texttt{replace}(\boxed{3}, y, \boxed{5}, w)$$

$[\psi/x](\phi_1 = \phi_2) = ([\psi/x]\phi_1 = [\psi/x]\phi_2)$

$$\texttt{replace}(x,y,v,w) \overset{\forall}{\Longleftarrow}$$
$$x\begin{bmatrix} \textit{equ} & \\ \text{TYPE} & \boxed{1} \\ \text{ARG1} & \boxed{2} \\ \text{ARG2} & \boxed{3} \end{bmatrix} \textbf{ and } v\begin{bmatrix} \textit{equ} & \\ \text{TYPE} & \boxed{1} \\ \text{ARG1} & \boxed{4} \\ \text{ARG2} & \boxed{5} \end{bmatrix}$$
$$\textbf{and } \texttt{replace}(\boxed{2}, y, \boxed{4}, w) \textbf{ and } \texttt{replace}(\boxed{3}, y, \boxed{5}, w)$$

$[\psi/x](\lambda x.\phi) = \lambda x.\phi$

$$\texttt{replace}(x,y,v,w) \overset{\forall}{\Longleftarrow}$$
$$x\begin{bmatrix} \textit{abstr} & \\ \text{VAR} & y \end{bmatrix}$$
$$\textbf{and } v \approx x$$

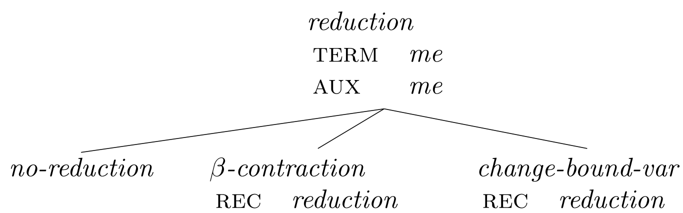
$[\psi/x](\lambda y.\phi) = \lambda y.[\psi/x]\phi$, if $y \neq x$ and $y \notin FV(\psi)$ or $x \notin FV(\phi)$

$$\texttt{replace}(x,y,v,w) \overset{\forall}{\Longleftarrow}$$
$$(\textbf{not } \boxed{2} \approx y)$$
$$\textbf{and } (\textbf{not } \texttt{free-variable}(\boxed{2}, w) \textbf{ or not } \texttt{free-variable}(y, \boxed{3})\ )$$
$$\textbf{and } x\begin{bmatrix} \textit{abstr} & \\ \text{TYPE} & \boxed{1} \\ \text{VAR} & \boxed{2} \\ \text{ARG} & \boxed{3} \end{bmatrix} \textbf{ and } v\begin{bmatrix} \textit{abstr} & \\ \text{TYPE} & \boxed{1} \\ \text{VAR} & \boxed{2} \\ \text{ARG} & \boxed{4} \end{bmatrix}$$
$$\textbf{and } \texttt{replace}(\boxed{3}, y, \boxed{4}, w)$$

The *me* object v is obtained by substituting the *me* object w for every free occurrence of the *var* object y in x. The result of this substitution is the *me* object v. There are two base cases: if x is identical to y, then the reduced version of y, namely w, is identical to v. If x is not identical to y, but an atomic term, i.e., either a variable or a constant, then x is identical to v. In the recursive case, the structure of x is copied to v, and the relation `replace` is applied to all recursive subterms of x.

The next definition is that of α-conversion, i.e., Definition 4.4. Before turning to it, we have to specify a general architecture for expressing conversion operations. We introduce a sort *reduction* which has two attributes TERM and AUX defined on it, which take values of sort *me*. We will have a constraint on this sort that guarantees that for each *reduction* object, the AUX value α-converts/β-reduces to the TERM value. In (204) we give the sort hierarchy and the appropriateness conditions for the sort *reduction*.

(204) The sort hierarchies of *reduction*:



As indicated in (204), the sort *reduction* comes in three species: either there is no reduction, i.e., we have neither changed a bound variable nor executed β-contraction. In this case, we have an instance of a *no-reduction* object. Alternatively, there can be one change of bound variable (Definition 4.4), i.e., we have an instance of a *change-bound-var* object; or, there can be one β-contraction (Definition 4.5), which is expressed in objects of the sort *β-contraction*. For these last two cases, an attribute REC is needed to account for the possibly iterative application of bound variable changes and β-contractions.

The idea of the recursive structure of *reduction* is that the AUX value always reduces to the TERM value. If this relation holds, without performing a reduction step, the AUX value equals the TERM value. If there is a reduction, the AUX value reduces to the REC AUX value by one change of a bound variable or one β-contraction. In addition, the TERM value and the REC TERM value are identical. We can illustrate this with the following AVM.

(205)
$$\begin{bmatrix} \textit{change-bound-var} & \\ \text{TERM} & \boxed{1} \\ \text{AUX} & \boxed{2}\ (\lambda x.\mathsf{walk}'_{@}(x(@)))(y) \\ \text{REC} & \begin{bmatrix} \textit{no-reduction} & \\ \text{TERM} & \boxed{1} \\ \text{AUX} & \boxed{1}\ (\lambda z.\mathsf{walk}'_{@}(z(@)))(y) \end{bmatrix} \end{bmatrix}$$

The AVM in (205) describes a *change-bound-var* object, which has the *me* object described in (199a) as its AUX value. Its REC AUX value is related to its AUX value by one change of a bound variable, changing x to z. The REC value is of sort *no-reduction*, therefore the REC AUX and the REC TERM values are identical. Furthermore, the TERM value is identical to the REC TERM value. We say that the AUX value α-converts to the TERM value.

The intuition behind this encoding can be made explicit in the two constraints given in (206) and (207), which we assume to be part of the grammar.

(206) The theory of *no-reduction*:

$$\textit{no-reduction} \Rightarrow \begin{bmatrix} \text{TERM} & \boxed{1} \\ \text{AUX} & \boxed{1} \end{bmatrix}$$

(207) The theory of *change-bound-var*:

$$\textit{change-bound-var} \Rightarrow \left(\begin{array}{l} \begin{bmatrix} \text{TERM} & \boxed{1} \\ \text{AUX} & \boxed{2} \\ \text{REC} & \begin{bmatrix} \text{TERM} & \boxed{1} \\ \text{AUX} & \boxed{3} \end{bmatrix} \end{bmatrix} \\ \textbf{and}\ \boxed{4}\begin{bmatrix} \textit{abstr} & \\ \text{VAR} & \boxed{6}\begin{bmatrix}\text{TYPE} & \boxed{7}\end{bmatrix} \\ \text{ARG} & \boxed{8} \end{bmatrix} \\ \textbf{and}\ \texttt{subterm}(\boxed{4},\boxed{2}) \\ \textbf{and}\ \boxed{5}\begin{bmatrix} \textit{abstr} & \\ \text{VAR} & \boxed{9}\begin{bmatrix}\text{TYPE} & \boxed{7}\end{bmatrix} \\ \text{ARG} & \boxed{10} \end{bmatrix} \\ \textbf{and not}\ \texttt{free-variable}(\boxed{9},\boxed{8}) \\ \textbf{and}\ \texttt{replace}(\boxed{8},\boxed{6},\boxed{10},\boxed{9}) \\ \textbf{and}\ \texttt{replace1}(\boxed{2},\boxed{4},\boxed{3},\boxed{5}) \end{array} \right)$$

The constraint on the sort *no-reduction* in (206) enforces the identities within the REC value of the *change-bound-var* object described in (205) above. Similarly, in the second constraint, the tag $\boxed{1}$ is used to indicate that the TERM value of a *change-bound-var* object is identical with its REC TERM value. The "real" conversion step relates the AUX value of a *change-bound-var* object with its REC AUX value.

To illustrate the effect of the second constraint, it is useful to re-consider the example in (205). In (208), we have indicated which subterms of the TERM and the AUX value of the object described in (205) correspond to which tags in the constraint.

(208)
$$\begin{array}{ll}
\boxed{1} & (\lambda z.\mathsf{walk}'_{@}(z(@)))(y) \\
\boxed{2} & (\lambda x.\mathsf{walk}'_{@}(x(@)))(y) \\
\boxed{3} & (\lambda z.\mathsf{walk}'_{@}(z(@)))(y) \\
\boxed{4} & \lambda x.\mathsf{walk}'_{@}(x(@)) \\
\boxed{5} & \lambda z.\mathsf{walk}'_{@}(z(@)) \\
\boxed{6} & x \\
\boxed{7} & \langle s, e \rangle \\
\boxed{8} & \mathsf{walk}'_{@}(x(@)) \\
\boxed{9} & z \\
\boxed{10} & \mathsf{walk}'_{@}(z(@))
\end{array}$$

The tag $\boxed{1}$ is the TERM value and also the REC TERM value of the object described in (205). The tag $\boxed{2}$ is its AUX value, and the tag $\boxed{3}$ is its REC AUX value (which happens to be identical to $\boxed{1}$ in this simple example). As we said before, the change of bound variable relates the AUX value to the REC AUX value, i.e., $\boxed{2}$ is changed to $\boxed{3}$.

Following Definition 4.4, there is a term $\lambda v.\phi$ which is a subterm of the AUX value. The tag $\boxed{4}$ points to such a term, as indicated in (208), i.e., to the term $\lambda x.\mathsf{walk}'_{@}(x(@))$. This term consists of a variable x (tag $\boxed{6}$) and some other term $\mathsf{walk}'_{@}(x(@))$ (tag $\boxed{8}$). We can now chose some variable z (tag $\boxed{9}$) which is not free in $\boxed{8}$, and substitute the variable z for every free occurrence of x in $\mathsf{walk}'_{@}(x(@)$. The term that results from this substitution, $\mathsf{walk}'_{@}(z(@))$, is indicated by the tag $\boxed{10}$. This substitution is enforced by the relation call `replace`($\boxed{8}$,$\boxed{6}$,$\boxed{10}$,$\boxed{9}$).

The last step is to substitute the term $\lambda z.\mathsf{walk}'_{@}(z(@))$ ($\boxed{10}$) for the term $\lambda x.\mathsf{walk}'_{@}(x(@))$ ($\boxed{4}$) in the original AUX value ($\boxed{2}$). The resulting term is given the tag $\boxed{3}$. For this substitution, we cannot use the relation `replace` as defined in (203), because in the case of the relation `replace`, the second argument must be a variable and we must replace all free occurrences of this variable. The relation that we need here is slightly different: now, we can replace an arbitrary subterm. Furthermore, we substitute only one occurrence of this subterm by a new term. The relation that has this effect is called `replace1`. The constraint in (207) requires that exactly one occurrence of the term $\boxed{4}$ (i.e., $\lambda x.\mathsf{walk}'_{@}(x(@))$) in the AUX value $\boxed{2}$ is replaced by the term $\lambda z.\mathsf{walk}'_{@}(z(@))$ (tag $\boxed{5}$) in the resulting term $\boxed{3}$. This substitution of one occurrence is expressed in the relation call `replace1`($\boxed{2}$,$\boxed{4}$,$\boxed{3}$,$\boxed{5}$).

As mentioned above, the theory of the sort *change-bound-var* uses a relation `replace1` which we have not yet defined. The relation is similar to the relation `replace` as given in (203). It is, however, also different in two respects. First, the term to be substituted need not be a variable, but can be of arbitrary complexity. Second we replace exactly one occurrence of this term. The definition of this new relation is given in (209).

(209) The relation `replace1`:

$$\begin{array}{l}
\texttt{replace1}(x, y, v, w) \stackrel{\forall}{\Longleftarrow} \\
\quad x \approx y \textbf{ and } v \approx w \\
\texttt{replace1}(x, y, v, w) \stackrel{\forall}{\Longleftarrow} \\
\quad {}^{x}\begin{bmatrix} \textit{appl} \\ \text{TYPE } \boxed{1} \\ \text{FUNC } \boxed{2} \\ \text{ARG } \boxed{3} \end{bmatrix} \textbf{ and } {}^{v}\begin{bmatrix} \textit{appl} \\ \text{TYPE } \boxed{1} \\ \text{FUNC } \boxed{4} \\ \text{ARG } \boxed{5} \end{bmatrix} \\
\quad \textbf{and} \begin{pmatrix} \texttt{replace1}(\boxed{2}, y, \boxed{4}, w) \\ \textbf{and } \boxed{3} \approx \boxed{5} \end{pmatrix} \textbf{ or } \begin{pmatrix} \texttt{replace1}(\boxed{3}, y, \boxed{5}, w) \\ \textbf{and } \boxed{2} \approx \boxed{4} \end{pmatrix}
\end{array}$$

$$\begin{array}{l}
\texttt{replace1}(x,y,v,w) \stackrel{\forall}{\Longleftarrow} \\
\quad {}^{x}\begin{bmatrix} \textit{equ} & \\ \text{TYPE} & \boxed{1} \\ \text{ARG1} & \boxed{2} \\ \text{ARG2} & \boxed{3} \end{bmatrix} \textbf{ and } {}^{v}\begin{bmatrix} \textit{equ} & \\ \text{TYPE} & \boxed{1} \\ \text{ARG1} & \boxed{4} \\ \text{ARG2} & \boxed{5} \end{bmatrix} \\
\quad \textbf{and} \left(\begin{array}{l} \texttt{replace1}(\boxed{2}, y, \boxed{4}, w) \\ \textbf{and } \boxed{3} \approx \boxed{5} \end{array} \right) \textbf{ or } \left(\begin{array}{l} \texttt{replace1}(\boxed{3}, y, \boxed{5}, w) \\ \textbf{and } \boxed{2} \approx \boxed{4} \end{array} \right) \\
\texttt{replace1}(x,y,v,w) \stackrel{\forall}{\Longleftarrow} \\
\quad (\textbf{not } \boxed{2} \approx y) \\
\quad \textbf{and } (\textbf{not } \texttt{free-variable}(\boxed{2}, w) \textbf{ or not } \texttt{free-variable}(y, \boxed{3})\ ) \\
\quad \textbf{and } {}^{x}\begin{bmatrix} \textit{abstr} & \\ \text{TYPE} & \boxed{1} \\ \text{VAR} & \boxed{2} \\ \text{ARG} & \boxed{3} \end{bmatrix} \textbf{ and } {}^{v}\begin{bmatrix} \textit{abstr} & \\ \text{TYPE} & \boxed{1} \\ \text{VAR} & \boxed{2} \\ \text{ARG} & \boxed{4} \end{bmatrix} \\
\quad \textbf{and } \texttt{replace1}(\boxed{3}, y, \boxed{4}, w)
\end{array}$$

The easiest way to understand this definition is to compare it with the corresponding clauses in the definition of the relation `replace`. In the first clause, the actual substitution takes place. It differs from the first clause in the definition of the relation `replace` in (203) because in the case of the relation `replace1` we do not require that x be a *var* object.

The second clause in (209) corresponds to the third clause of the relation `replace`, i.e., substitution within *appl* objects. In the case of the relation `replace1` we do, however, execute the substitution only in one of the subterms of the *appl* object, the other subterm is taken over without changes. This is indicated by the disjunction at the end of the clause. The clause of *equ* objects is parallel to that for *appl* objects.

For *abstr* objects, we must consider only one case, i.e., there is no clause that would correspond to the fifth clause of the relation `replace`, i.e., to the case $[\psi/x]\lambda x.\phi$. The fourth clause in (209) is exactly as the sixth clause in the definition of the relation `replace` in (203), but using the relation `replace1`.

With these definitions, we have encoded α-conversion so that the following is true: for two terms ϕ and ψ, $\phi \equiv_\alpha \psi$ holds iff there is a configuration $\langle u, I\rangle$ in some exhaustive model I of the grammar such that (i) u is of sort *reduction*, (ii) the configuration does not contain an object of sort *β-reduction*, and (iii) u's AUX value corresponds to ϕ (by the functions SR and "*") and u's TERM value corresponds to ψ (by the functions SR and "*").

After α-conversion, we can address β-contraction. We will express a β-reduction step in the form of a constraint in the theory. Parallel to the situation with the sort *change-bound-var*, the AUX value of a *β-contraction* object contains a β-redex, the REC AUX value, then contains the contractum of this redex as defined in Definition 4.5. In (210) we state the necessary constraint on the sort *β-contraction*.

(210) The theory of *β-contraction*:

$$\textit{β-contraction} \Rightarrow \left(\begin{array}{l} \begin{bmatrix} \text{TERM} & \boxed{1} \\ \text{AUX} & \boxed{2} \\ \text{REC} & \begin{bmatrix} \text{TERM} & \boxed{1} \\ \text{AUX} & \boxed{3} \end{bmatrix} \end{bmatrix} \textbf{ and } {}^{\boxed{4}}\begin{bmatrix} \textit{appl} & \\ \text{FUNC} & \begin{bmatrix} \textit{abstr} & \\ \text{VAR} & \boxed{5} \\ \text{ARG} & \boxed{6} \end{bmatrix} \\ \text{ARG} & \boxed{7} \end{bmatrix} \\ \textbf{and } \texttt{subterm}(\boxed{4},\boxed{2}) \\ \textbf{and } \texttt{replace}(\boxed{4},\boxed{5},\boxed{8}\boxed{7}) \\ \textbf{and } \texttt{replace1}(\boxed{2},\boxed{4},\boxed{3}\boxed{8}) \end{array} \right)$$

In a *β-contraction* object, the TERM value is identical to the REC TERM value, just as has been the case for *change-bound-var* objects. This is expressed with the tag $\boxed{1}$ in (210).

The AUX value ($\boxed{2}$) reduces to the REC AUX value ($\boxed{3}$) by one β-reduction ($\rightarrow_{1\beta}$). To achieve this, we require that there be a *me* object $\boxed{4}$ which is a subterm of the AUX value. This object has the form of a β-redex, i.e., it is of sort *appl* and its FUNC value is of sort *abstr*.

Given this β-redex, we can find its contractum, which is referred to with the tag $\boxed{8}$. The second relation call in (210) enforces this: the relation `replace` must hold for the quadruple consisting of the FUNC ARG value ($\boxed{6}$) of the β-redex, its FUNC VAR value ($\boxed{5}$), the contractum ($\boxed{8}$) and the ARG value ($\boxed{7}$) of the redex. Put differently, this means that the term that corresponds to the *me* object $\boxed{8}$ is the result of substituting the term that corresponds to $\boxed{7}$ for every free occurrence of the variable that corresponds to $\boxed{5}$ in the term that corresponds to $\boxed{6}$.

Once we are sure that $\boxed{8}$ is the contractum of $\boxed{4}$, we must replace one occurrence of the β-redex $\boxed{4}$ in the initial term $\boxed{2}$ by the contractum $\boxed{8}$. The *me* object that results from this replacement is the REC AUX value ($\boxed{3}$). This is expressed by the last line in (210). There, the quadruple $\langle\boxed{2},\boxed{4},\boxed{3},\boxed{8}\rangle$ is required to be in the relation `replace1`.

To illustrate the constraint on the sort *β-contraction*, consider the following description of a *β-contraction* object, which shows how *me* objects described by the AVMs in (199) can occur as components of a *β-reduction* object.

(211)
$$\begin{bmatrix} \textit{\beta-contraction} & \\ \text{TERM} & \boxed{1} \\ \text{AUX} & \boxed{2}\ (\lambda x.\mathsf{walk}'_{@}(x(@)))(y) \\ \text{REC} & \begin{bmatrix} \textit{no-reduction} & \\ \text{TERM} & \boxed{1} \\ \text{AUX} & \boxed{1}\ \mathsf{walk}'_{@}(y(@)) \end{bmatrix} \end{bmatrix}$$

To see that an object described by (211) satisfies the constraint in (210), we indicate in (212) which subterms of the TERM and the AUX value of this object correspond to which tags in the constraint.

(212)
$\boxed{1}$ $\mathsf{walk}'_{@}(y(@))$
$\boxed{2}$ $(\lambda x.\mathsf{walk}'_{@}(x(@)))(y)$
$\boxed{3}$ $\mathsf{walk}'_{@}(y(@))$
$\boxed{4}$ $(\lambda x.\mathsf{walk}'_{@}(x(@)))(y)$
$\boxed{5}$ x
$\boxed{6}$ $\mathsf{walk}'_{@}(x(@))$
$\boxed{7}$ y
$\boxed{8}$ $\mathsf{walk}'_{@}(y(@))$

In the description in (211) the tag $\boxed{1}$ expresses that the TERM and the REC TERM values are identical, just as required in (210). The REC value of an object described by (211) is of sort *no-reduction*. As this is a subsort of sort *reduction*, the appropriateness conditions on the sort *β-contraction* are met.

Turning back to the constraint in (210), we must find a β-redex which is a subterm of the AUX value $\boxed{2}$. In the example that we have chosen, the AUX value itself is a β-redex. Therefore, the tags $\boxed{2}$ and $\boxed{4}$ refer to the same object. We can, then, verify that the term $\boxed{8}$ is the contractum of the term $\boxed{4}$, i.e., it is the result of replacing every free occurrence of the variable x (tag $\boxed{5}$) in the term $\mathsf{walk}'_{@}(x(@))$ (tag $\boxed{6}$) by the term y (tag $\boxed{7}$). Finally, the contractum $\boxed{8}$ is substituted for the redex $\boxed{4}$ in $\boxed{2}$. As $\boxed{2}$ and $\boxed{4}$ are the same, the first clause of the relation `replace1` applies, and the resulting term is identical to the contractum $\boxed{8}$.

In (213) and (214) we give more complex examples for β-reduction. In (a) we show the reduction steps in term notation, in (b) we describe the corresponding *reduction* objects.

In (213) we start out with the example term of (199a). We use α-conversion to replace the bound variable x by z, which results in the term given in (195). Finally, we execute β-contraction and are left with the term in (199b).

(213) a. $(\lambda x.\mathsf{walk}'_{@}(x(@)))(y)$
$\equiv_{\alpha}(\lambda z.\mathsf{walk}'_{@}(z(@)))(y)$
$\rightarrow_{1\beta}\mathsf{walk}'_{@}(y(@))$

b. $$\begin{bmatrix} \textit{change-bound-var} & \\ \text{TERM} & \boxed{3} \\ \text{AUX} & \boxed{1}\ (\lambda x.\mathsf{walk}'_{@}(x(@)))(y) \\ \text{REC} & \begin{bmatrix} \textit{β-contraction} & \\ \text{TERM} & \boxed{3} \\ \text{AUX} & \boxed{2}\ (\lambda z.\mathsf{walk}'_{@}(z(@)))(y) \\ \text{REC} & \begin{bmatrix} \textit{no-reduction} & \\ \text{TERM} & \boxed{3} \\ \text{AUX} & \boxed{3}\ \mathsf{walk}'_{@}(y(@)) \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

In the AVM in (213b) we use the tags $\boxed{1}$, $\boxed{2}$ and $\boxed{3}$ for the *me* objects that correspond to these terms. The theory of *change-bound-var* in (207) states that the *me* object $\boxed{1}$ converts to the *me* object $\boxed{2}$ via one change of bound variable. As the REC value of the object described in (213b) is of sort *β-contraction*, the object $\boxed{2}$ reduces to $\boxed{1}$ by one β-contraction. The REC REC value of the object described by the AVM is of sort *no-reduction*. Therefore, by virtue of the constraint in (206), its AUX and its TERM values are identical, i.e., we have the result of one α-conversion and one β-contraction as the REC REC TERM value. By the theories of the sorts *change-bound-var* and *β-contraction*, the TERM values of all *reduction* objects that are components of one *reduction* object are identical.

Within each *reduction* object, we interpret the relation between the AUX value and the TERM value as β-reduction/λ-conversion, i.e., as a finite (possibly empty) sequence of applications of α-conversion and β-contraction.

The example in (214), shows two applications of β-contraction, i.e., in (b), we give the description of a *reduction* object that has two components of sort *β-contraction*.

(214) a. $(\lambda x.\mathsf{walk}'_{@}(x(@)))(\lambda @.m)$
$\rightarrow_{1\beta}\mathsf{walk}'_{@}((\lambda @.m)(@))$
$\rightarrow_{1\beta}\mathsf{walk}'_{@}(m)$

b. $$\begin{bmatrix} \textit{β-contraction} & \\ \text{TERM} & \boxed{3} \\ \text{AUX} & \boxed{1}\ (\lambda x.\mathsf{walk}'_{@}(x(@)))(\lambda @.m) \\ \text{REC} & \begin{bmatrix} \textit{β-contraction} & \\ \text{TERM} & \boxed{3} \\ \text{AUX} & \boxed{2}\ \mathsf{walk}'_{@}((\lambda @.m)(@)) \\ \text{REC} & \begin{bmatrix} \textit{no-reduction} & \\ \text{TERM} & \boxed{3} \\ \text{AUX} & \boxed{3}\ \mathsf{walk}'_{@}(m) \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

The AUX value of an object described by the AVM in (214b) is a β-redex, just as was the case in our example in (211) above. In its REC value, however, the AUX value ($\boxed{2}$) contains a proper subterm which is a β-redex, i.e., the subterm $(\lambda @.m)(@)$. Therefore, the REC value of an object described by (214b) is of sort *β-contraction*. In the REC REC AUX value, this redex is replaced by its contractum, the term m. Since the REC REC AUX value ($\boxed{3}$) does not contain any bound variables nor β-redices, the REC REC value must be of sort *no-reduction*. The theory of *no-reduction* enforces that the REC REC AUX value and the REC REC TERM value are identical, and the theory of *β-contraction* guarantees the identities of the TERM value, the REC TERM value and the REC REC TERM value.

With this illustration, we conclude our presentation of the first formalization of λ-conversion. In the introduction to Section 4.2 we characterized this first formalization as being similar in spirit to the DR-encoding of shifting operations. The parallelism lies in the fact that we explicitly encode each step in the derivation of the reduced term within a linguistic object, a *reduction* object in this case. This leads to the situation that if there are n ways to reduce a term ϕ to a term ψ, then there are n distinct configurations $\langle u, I\rangle$ in each exhaustive model of the grammar such that u is of sort *reduction*, and each configuration encodes one derivation.

In (211) and (213b) we have such a case: In both AVMs, the AUX value corresponds to the term $(\lambda x.\mathsf{walk}'_{@}(x_{@}))(y)$ and the TERM value corresponds to the term $\mathsf{walk}'_{@}(y_{@})$. The objects described by the two AVMs are, however, distinct, as they express different ways to derive the TERM values from the AUX value. The alternative formalization that we will present in Section 4.2.2 does not have this property, i.e., there, β-reduction will be fully expressed as a relation. This means that there is no additional kind of objects introduced in the linguistic universe to express the reduction steps.

After this technical remark, we can turn to the integration of the encoding of β-reduction into the RSRL formalization of LF-Ty2.

4.2.1.2. *Integration.* In our first presentation of LF-Ty2 in Section 1.3.3, we applied λ-conversion whenever possible, i.e., we applied λ-conversion to the terms that resulted from shifting operations and from intensional functional application. As shifting operations are restricted to words in our framework, and intensional functional application is only used to combine the semantic contributions of the daughters in a phrase, we must allow for λ-conversion at the level of words and at the level of phrases. This has a consequence for the formalization: since we want to apply λ-conversion to the logical form of phrases, we cannot encode it as a derivational rule, the way we encoded the shifting operations. Instead, we treat λ-conversion as an operation that is internal to the CONTENT value of a sign. As such, its application does not depend on the word/phrase status of a sign.

In Section 4.2.1.1 we provided an encoding of λ-conversion by means of objects of sort *reduction*. In the following integration of λ-conversion into our overall grammar, we must first define some place within the feature geometry of a sign where *reduction* objects may occur. We will also provide those principles that are needed to connect the newly introduced components of a sign with its logical form. This will lead to a change in the structure of the CONTENT value of signs. This change will have consequences for the definitions of the relation `ifa` (originally defined in (150) and used in the SEMANTICS PRINCIPLE), and for the encoding of shifting operations.

In order to make β-reduction available in the CONTENT value of every sign, we change the signature in the way indicated in (215). First, we change the appropriateness conditions of the sort *local* so that the attribute CONTENT is appropriate for the sort *local*, and that the sort *content* is appropriate to the attribute CONTENT. While this is exactly as specified in the appropriateness conditions of the sort *local* in (73a), we also change the appropriateness conditions of the sort *content*.

(215) Revised sort hierarchy and appropriateness conditions:

a. Revised appropriateness conditions for the sort *local*:

local	CATEGORY	*category*
	CONTENT	*content*

b. Appropriateness conditions for the sort *content*:

content	LOGICAL-FORM	*me*
	REDUCTION	*reduction*

In Section 2.3.1 we introduced the sort *content* as being atomic, i.e., as having no attributes appropriate for it. In (215b), we introduce two attributes: LOGICAL-FORM (LF) and REDUCTION (RED). The sort *me* is appropriate to the attribute LF, and the sort *reduction* is appropriate to the attribute RED. Intuitively, the value of the attribute LF will express the *logical form* of a sign, i.e., it takes over the function attributed to the CONTENT value in Section 4.1. We will enforce that the LF value of a *content* object is always *redex-free*, i.e., that it does not contain any β-redex.

The second attribute introduced in (215b), RED, is used to integrate *reduction* objects as defined in Section 4.2.1.1 into linguistic signs. We enforce a straightforward connection between the LF value and the RED value of a CONTENT object: The LF value is identical with the RED TERM value. Furthermore, the LF value must be redex-free. These two conditions are expressed in the CONTENT PRINCIPLE in (216).

(216) The CONTENT PRINCIPLE (for the first alternative):

$$content \Rightarrow \left(\begin{array}{l} \begin{bmatrix} \text{LF } \boxed{1} \\ \text{RED TERM } \boxed{1} \end{bmatrix} \\ \textbf{and not E}\boxed{2} \\ \left(\texttt{subterm}(\boxed{2},\boxed{1}) \textbf{ and } \boxed{2}\begin{bmatrix} \textit{appl} \\ \text{FUNC } \textit{abstr} \end{bmatrix} \right) \end{array} \right)$$

To illustrate this principle, consider the AVM in (217). The AVM describes only objects that satisfy the CONTENT PRINCIPLE. An object described by this AVM has a RED value as described in (214b), i.e., a RED value which expresses the reduction of the term $(\lambda x.\mathsf{walk}'_{@}(x_{@}))(\lambda @.m)$ to the term $\mathsf{walk}'_{@}(m)$. The latter term occurs as the RED TERM value of the *content* object. As an object described in (217) satisfies the CONTENT PRINCIPLE, its LF value and its RED TERM value must be identical. The term $\mathsf{walk}'_{@}(m)$ is redex-free. Thus, the second condition in the CONTENT PRINCIPLE is met as well.

$$(217)\quad \begin{bmatrix} \textit{content} \\ \text{LF} & \boxed{3}\ \mathsf{walk}'_{@}(m) \\ \text{RED} & \begin{bmatrix} \beta\textit{-contraction} \\ \text{TERM} & \boxed{3} \\ \text{AUX} & \boxed{1}\ (\lambda x.\mathsf{walk}'_{@}(x(@)))(\lambda @.m) \\ \text{REC} & \begin{bmatrix} \beta\textit{-contraction} \\ \text{TERM} & \boxed{3} \\ \text{AUX} & \boxed{2}\ \mathsf{walk}'_{@}((\lambda @.m)(@)) \\ \text{REC} & \begin{bmatrix} \textit{no-reduction} \\ \text{TERM} & \boxed{3} \\ \text{AUX} & \boxed{3} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

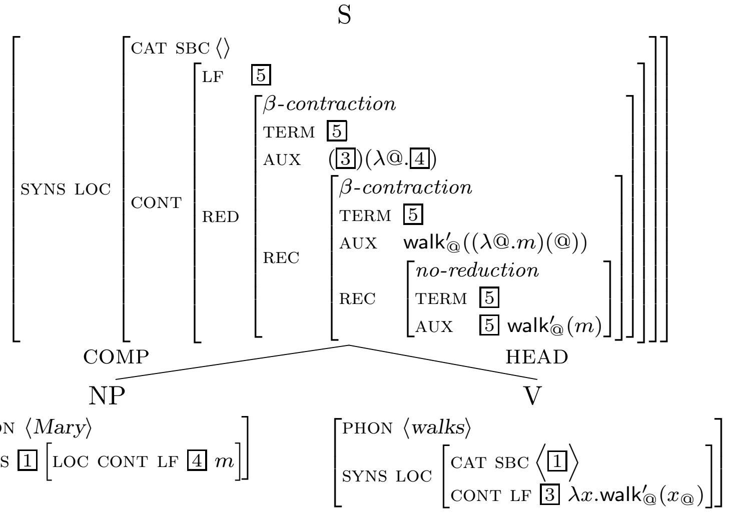
So far, we have provided an integration of *reduction* objects into the grammar presented in this thesis. As we have altered the overall feature geometry, we must adapt the SEMANTICS PRINCIPLE and the encoding of shifting operations to the new feature geometry. We will, first, re-consider the SP. In (218) we repeat the SP as given in (149) above.

(218) The SEMANTICS PRINCIPLE (SP):

$$phrase \Rightarrow \begin{bmatrix} \text{SYNS LOC CONT } \boxed{1} \\ \text{DTRS} \begin{bmatrix} \text{H-DTR SYNS LOC CONT } \boxed{2} \\ \text{N-DTR SYNS LOC CONT } \boxed{3} \end{bmatrix} \end{bmatrix}$$
$$\textbf{and } \texttt{intensional-functional-application}(\boxed{1},\boxed{2},\boxed{3})$$

The SP expresses that the logical form of a phrase is the intensional functional application of the logical forms of its daughters. The way we have defined the relation `ifa` in (150) we did not include β-reduction in functional application. Given the new architecture, it is easy to re-define the relation `ifa` so that the CONTENT LF value of a phrase is the β-reduced

FIGURE 4.7. The structure of the sentence *Mary walks*:



form of the intensional functional application. The revised definition of the relation `ifa` is given in (219).

(219) Revised definition of the relation `ifa`:

$$\texttt{ifa}(\boxed{1},\boxed{2},\boxed{3}) \stackrel{\forall}{\Longleftarrow} \left(\begin{array}{l} \boxed{2}\left[\text{LF } \boxed{4}\right] \textbf{ and } \boxed{3}\left[\text{LF } \boxed{5}\right] \\ \textbf{and } \boxed{1}\left[\text{RED AUX } \left[\begin{array}{l} appl \\ \text{FUNC } \boxed{4} \\ \text{ARG } \left[\begin{array}{l} abstr \\ \text{VAR } @ \\ \text{ARG } \boxed{5} \end{array}\right] \end{array}\right]\right] \end{array}\right)$$

$$\texttt{ifa}(\boxed{1},\boxed{2},\boxed{3}) \stackrel{\forall}{\Longleftarrow} \left(\begin{array}{l} \boxed{2}\left[\text{LF } \boxed{4}\right] \textbf{ and } \boxed{3}\left[\text{LF } \boxed{5}\right] \\ \textbf{and } \boxed{1}\left[\text{RED AUX } \left[\begin{array}{l} appl \\ \text{FUNC } \boxed{5} \\ \text{ARG } \left[\begin{array}{l} abstr \\ \text{VAR } @ \\ \text{ARG } \boxed{4} \end{array}\right] \end{array}\right]\right] \end{array}\right)$$

If ϕ is the term that corresponds to the LF value of the head daughter and ψ is the term that corresponds to the LF value of the nonhead daughter, then in this new definition of the relation `ifa`, we assume that the CONTENT RED AUX value of a phrase is of the form $\phi(\lambda @.\psi)$ or $\psi(\lambda @.\phi)$. The CONTENT PRINCIPLE, then, enforces that the LF value of the phrase corresponds to some term χ which (i) is the result of applying β-reduction to the term $\phi(\lambda @.\psi)$ or $\psi(\lambda @.\phi)$, and (ii) is redex-free.

In Figure 4.7 we illustrate this new definition with the example sentence in (220).

(220) Mary walks.

In the figure, the logical form of the head daughter is the basic translation of the verb *walks*, i.e., $\lambda x.\mathsf{walk}'_@(x_@)$. The logical form of the nonhead daughter is the basic translation of the word *Mary*, i.e., m. At the phrase, the SP guarantees that the CONTENT RED AUX value is the intensional functional application of these two terms, i.e. $(\lambda x.\mathsf{walk}'_@(x_@))(\lambda @.m)$. The CONTENT RED value of the phrase is just as described in (214b) above, i.e., it expresses two β-contractions, and the TERM value is the term $\mathsf{walk}'_@(m)$. By the CONTENT PRINCIPLE, this terms appears as the CONTENT LF value of the phrase.

The example illustrates how the new definition of the relation `ifa` allows us to assume fully reduced LF values for phrases, while still using intensional functional application to combine the logical forms of the daughters,

In the framework LF-Ty2, we do, however, also use shifting operations. Applying a shifting operation to the basic translation of a word leads to a term which contains β-redices. For illustration, consider the application of the operation AR to the basic translation of the verb *walks*, given in (221).

(221) *walks* $\leadsto \lambda x_{se}.\mathsf{walk}'_{@et}(x(@))$
$\longrightarrow_{AR} \lambda X_{s((se)t)}.X(@)(\lambda @ \lambda y.[(\lambda x.\mathsf{walk}'_{@}(x(@)))(y)])$

The last term in (221) contains a β-redex, the term $(\lambda x.\mathsf{walk}'_{@}(x(@)))(y)$. Applying one β-contraction to this term results in the term given in the last line of (222).

(222) *walks* $\leadsto \lambda x_{se}.\mathsf{walk}'_{@et}(x(@))$
$\longrightarrow_{AR} \lambda X_{s((se)t)}.X(@)(\lambda @ \lambda y.[(\lambda x.\mathsf{walk}'_{@}(x(@)))(y)])$
$\rightarrow_{1\beta} \lambda X_{s((se)t)}.X(@)(\lambda @ \lambda y.\mathsf{walk}'_{@}(y(@)))$

As we require in the CONTENT PRINCIPLE that the LF value be always redex-free, we want the last term in (222) to be the logical form of the word *walks*. To achieve this, we must adapt our formalization of shifting operations as given in Section 4.1 slightly. We will first re-consider the DR-encoding and then address the LE-encoding of shifting operations in the new architecture.

When we adjusted the definition of the relation `ifa` to the new signature, we basically used the "old" specification of the CONTENT value as a specification of the CONTENT RED AUX value. Similarly, we can change the AR-DR (or the VR-DR) in such a way that the CONTENT LF value of the input word and the CONTENT RED AUX value of the output word stand in the relation the relation `ar` (or `vr`). The constraints on the sort *reduction* and the CONTENT PRINCIPLE will, then, guarantee that the CONTENT LF value is the fully β-reduced version of the term that results from shifting the logical form of the input word. In (223) we state the adapted form of the shifting DRs from (171).

(223) a. Revised description of the AR-DR:

$$\begin{bmatrix} \textit{der-rule} & \\ \text{IN} & \begin{bmatrix} \textit{word} & \\ \text{PHON} & \boxed{3} \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \boxed{4} \\ \text{CONT} & \begin{bmatrix} \text{LF} & \boxed{1} \end{bmatrix} \end{bmatrix} \\ \text{NONL} & \boxed{5} \end{bmatrix} \end{bmatrix} \\ \text{OUT} & \begin{bmatrix} \textit{word} & \\ \text{PHON} & \boxed{3} \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \boxed{4} \\ \text{CONT} & \begin{bmatrix} \text{RED AUX} & \boxed{2} \end{bmatrix} \end{bmatrix} \\ \text{NONL} & \boxed{5} \end{bmatrix} \end{bmatrix} \end{bmatrix} \textbf{ and } \texttt{ar}(\boxed{1},\boxed{2})$$

b. Revised description of the VR-DR:

$$\begin{bmatrix} \textit{der-rule} & \\ \text{IN} & \begin{bmatrix} \textit{word} & \\ \text{PHON} & \boxed{3} \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \boxed{4} \\ \text{CONT} & \begin{bmatrix} \text{LF} & \boxed{1} \end{bmatrix} \end{bmatrix} \\ \text{NONL} & \boxed{5} \end{bmatrix} \end{bmatrix} \\ \text{OUT} & \begin{bmatrix} \textit{word} & \\ \text{PHON} & \boxed{3} \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \boxed{4} \\ \text{CONT} & \begin{bmatrix} \text{RED AUX} & \boxed{2} \end{bmatrix} \end{bmatrix} \\ \text{NONL} & \boxed{5} \end{bmatrix} \end{bmatrix} \end{bmatrix} \textbf{ and } \texttt{vr}(\boxed{1},\boxed{2})$$

In Figure 4.8 (page 184) we show an alternative derivation of sentence (220) which includes the application of shifting operations: we apply AR to the basic translation of the verb *walks* and VR to the basic translation of the proper name *Mary*. Note that in the figure, we indicate the application of the DRs by mentioning explicitly the STORE values of the derived words.

Within the CONTENT value of the word *walks*, we execute β-reduction as discussed in (222). Even though the basic translation of the word *Mary* is also shifted, there is no redex, as VR maps the term m to the redex-free term $\lambda P.P_{@}(\lambda @.m)$. Therefore, the CONTENT RED value of the derived word is of sort *no-reduction*. The SP then enforces that the CONTENT RED AUX value of the S node corresponds to the first term in (224), which reduces by six β-contractions to the second term in (224).

(224) $(\lambda X_{s((se)t)}.X(@)(\lambda @\lambda y.\mathsf{walk}'_{@}(y(@))))(\lambda @\lambda P.P_{@}(\lambda @.m))$
$\rightarrow_{\lambda}$ $\mathsf{walk}'_{@}(m)$

As can be seen in the tree in Figure 4.8, a *phrase* described by this tree contains every shifting step and every reduction step as a component. Therefore, the two alternative derivations given for sentence (220) in Figure 4.7 and in Figure 4.8 describe distinct objects.

What remains to be shown is how the encoding of λ-conversion proposed in Section 4.2.1.1 can be connected with the LE-encoding of shifting operations presented in Section 4.1. In the LE-encoding, the possibility for shifting is expressed as part of the lexical entry of a word. In (173) we gave the lexical entry of the verb *loves*, which contained a relation call to the relation `shifting`. The lexical entry described a word, if this word has a component $\boxed{1}$ which is of the form $\lambda x_1 \lambda x_2.\mathsf{love}'_{@}(x_2(@), x_1(@))$ and if the CONTENT value of this word stands in the relation `shifting` with this term.

To include β-reduction, we must require that the relation `shifting` holds between the term $\lambda x_1 \lambda x_2.\mathsf{love}'_{@}(x_2(@), x_1(@))$ and the CONTENT RED AUX value of the word. Then, the theory of the sort *reduction* and the CONTENT PRINCIPLE ensure that the CONTENT LF value of the word is the fully β-reduced form of some term that results from applying shifting operations to the basic translation of the word *loves*. In (225) we give the lexical entry of the word *loves* which respects the new signature.

FIGURE 4.8. An alternative structure of the sentence *Mary walks*:

(225) Parts of the lexical entry for the word *loves*, including shifting operations:

E[1] **E**[2]

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{loves} \rangle \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{SUBCAT} & \left\langle \text{NP, NP} \right\rangle \\ \text{MARKING} & \textit{unmarked} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{RED AUX} & \boxed{2} \end{bmatrix} \end{bmatrix} \\ \text{NONL} & \begin{bmatrix} \text{INHER SLASH} & \textit{eset} \\ \text{TO-BIND SLASH} & \textit{eset} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

and $\boxed{1}\left[\lambda x_1 \lambda x_2.\mathsf{love}'_{@}(x_2(@), x_1(@))\right]$

and `shifting`([1],[2])

When we compared the DR-encoding of shifting operations with the LE-encoding at the end of Section 4.1, we saw that the major difference between the two approaches lies in the fact that under the DR-encoding every application of a shifting operation is explicitly present in the structure of a sign. The LE-encoding does not have this property, i.e., there, it is only the resulting term that appears as the logical form of a sentence, while the single steps that led to this term cannot be recovered.

If we were to give the structure of sentence (220) under the derivation where we apply AR to the verb *walks* and VR to the proper name *Mary*, the result would be just like in Figure 4.7, but the STORE values of the head daughter and the nonhead daughter in the tree would be empty.

We have mentioned above that the encoding of λ-conversion in the form of *reduction* objects leads to a similar explicit presence of the reduction steps inside linguistic signs. So, even if we can "hide" the shifting steps in the LE-encoding, the reduction steps are explicitly present if use the present formalization of β-reduction. In Section 4.2.2 we will present an alternative formalization of λ-conversion that leaves the particular derivation steps outside the components of a linguistic object. This "implicit" encoding of λ-conversion appears to be more in the spirit of the LE-encoding of shifting operations. For this reason, we will not discuss the combination of the LE-encoding and the formalization of β-reduction as given in Section 4.2.1.1 in any more detail.

4.2.1.3. *Summary.* We provided a first formalization of λ-conversion in RSRL, and showed how this formalization can be integrated into the RSRL encoding of LF-Ty2. With this integration, we can say that we completed a rigorous RSRL-formalization of LF-Ty2 and an integration into a particular RSRL grammar of a fragment of English.

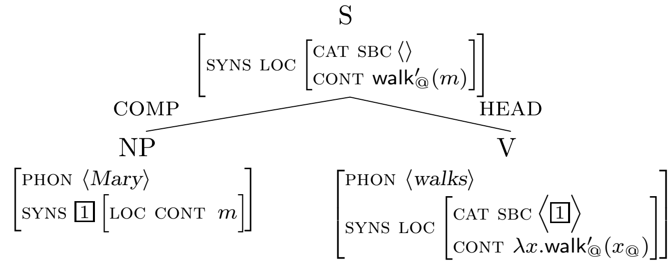
The formalization showed that we can express the definitions needed for λ-conversion in RSRL. Thus, not only can we use terms of Ty2 as semantic representations, but we can actually perform computations on these terms. The fact that we can require the logical form of a sign to be redex-free allows us for example, to express directly whether some term is in the scope of a quantifier within a larger term. If this is the case, then the first term is a subterm of the SCOPE value of the quantifier. With this potential constraints on logical forms can be expressed in a transparent way, such as the constraint in (53) which prevents a universal quantifier from taking scope in a matrix clause.

In our RSRL formalization of the definitions needed for λ-conversion, we expressed some definitions as relations, others (α-conversion and β-contraction) as sorts. The fact that a β-redex is not a subterm of its contractum forced us to introduce new sorts to our linguistic ontology. As a consequence of this new ontology, the CONTENT of a linguistic sign is not only its logical form, but also contains every step of the reduction that leads to the logical form. We consider this highly undesirable, because it introduces components to linguistic objects for whose existence there is no linguistic motivation.

In the following subsection, we will show that such an extension of the linguistic ontology is not necessary. Still, we think that the presentation of the encoding of λ-conversion given in this subsection is useful for various reasons: First, the encoding developed in the following subsection is far more abstract, but follows in the general line the present alternative. Therefore, the "simpler" case given in the present subsection is useful for an understanding of the following definitions. Second, the encoding of λ-conversion without chains provides a good illustration of the limits of RSRL quantification and relations. In Section 4.5 we will address some further issues in this direction.

4.2.2. Second Alternative. The second encoding of λ-conversion that we will present in this thesis makes use of *chains*, i.e., the list-like auxiliary structure provided by RSRL. The basic idea behind our second formalization is that we can provide an encoding of Ty2 terms

FIGURE 4.9. An example derivation of the sentence *Mary walks.*:



not only as *me* objects, but also as chains. This chain encoding will allow us to use refer to arbitrary terms of Ty2, independent of the logical form of a sign. In this way, we will be able to express λ-conversion as a relation, even though the input term to λ-conversion is not a subterm of the output term.

Let us illustrate the structure of this proposal intuitively. In LF-Ty2, we want the logical form of a phrase to be a redex-free term that results from applying λ-conversion to the intensional functional application of the logical forms of the daughters. Let us, suppose for a moment that the logical forms of the daughters are the *me* objects o and o' that correspond to some terms $\phi_{(s\tau)\tau'}$ and ψ_τ respectively. As indicated by the semantic types, we assume that ϕ is the functor, when these two terms combine. The logical form of the phrase is some *me* object u that corresponds to a term χ such that χ is redex-free and $\phi(\lambda@.\psi)$ β-reduces to *chi* ($\phi(\lambda@.\psi)\rightarrow_\lambda\chi$).

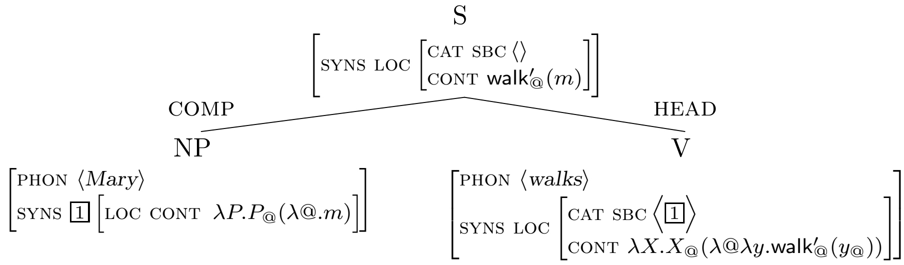
We will provide relation `chain2term` with the following property: a pair $\langle c,a\rangle$ is in the relation `chain2term` iff, c is a chain, o is a *me* object, and there is a Ty2 term ϕ such that c is the chain encoding of ϕ and o corresponds to ϕ. Thus, c and o are different data structures, but both correspond to the same term, ϕ. Furthermore, we will define a relation `reduction` that holds for a pair of chains $\langle c, c'\rangle$ iff c is the chain encoding of some term ϕ, c' is the chain encoding of some term ϕ', and ϕ β-reduces to ϕ' ($\phi\rightarrow_\lambda\phi'$). These relations, will be defined in Sections 4.2.2.1 and 4.2.2.2.

We can use these relations to carry out λ-conversion at the level of chains: Let us assume that u, o, o' are the *me* objects that appear as the logical form of a phrase and its two daughters respectively, and let $\chi_{\tau'}$, $\phi_{(s\tau)\tau'}$ and ψ_τ be the terms that correspond to u, o and o'. We, then, consider the chains, a and b, where a is the chain encoding of the term $\phi(\lambda@.\psi)$ and b is the chain encoding of the term χ. The last step is to require that the pair $\langle a, b\rangle$ is in the relation `reduction`. By the definition of the relation `reduction`, this means that $\phi(\lambda@.\psi)\rightarrow_\lambda\chi$. As b is the chain encoding of the term χ, we know that the pair $\langle b, u\rangle$ is in the relation `chain2term`. Thus, we have ensured that the logical form of the phrase is a β-reduced form of the intensional functional application of the logical forms of the daughters. In Section 4.2.2.3 we will integrate this way of expressing the relation between the logical forms of the signs in a local tree into our grammar.

While the chain encoding of Ty2 terms is straightforward, it is a so far unexploited potential of chains within RSRL. Therefore, it is of technical and conceptual interest. In Section 4.5 we will give an overview over the uses of chains that have been proposed so far within RSRL and try to locate the chain encoding of Ty2 among them.

As a result of the encoding of λ-conversion by the means of relations, the reduction steps will not be components of linguistic signs, and we can simply assume that the CONTENT value of a sign is an *me* object. In Figure 4.9 we give an example derivation of sentence (220), where we assume that the logical forms of the words *Mary* and *walks* are just their basic translations. As can be seen, the CONTENT value of the phrase is a fully β-reduced form

FIGURE 4.10. A second derivation of the sentence *Mary walks.*:



of the intensional functional application of the logical forms of the daughters, even though there is no *me* object occurring as a component of an object that is described by this tree that corresponds to the term $(\lambda x.\mathsf{walk}'_{@}(x_{@}))(\lambda @.m)$.

Similarly, in Figure 4.10, we describe an alternative derivation of sentence (220). In Figure 4.10 we assume an LE-encoding of shifting operations. Therefore, the application of AR to the basic translation of the verb and the application of VR to the basic translation of the proper name are not explicitly encoded in the structure of the sentence.

The trees given in Figures 4.9 and 4.10 should be compared to Figures 4.7 and 4.8, which show the same derivations, but use the encoding of λ-conversion presented in Section 4.2.1. Clearly, the structures that will result from the chain-based encoding of λ-conversion are more straightforward, as they avoid the conceptually problematic ontological commitment that every β-reduction step is a component of the linguistic objects.

4.2.2.1. *Terms of Ty2 as Chains.* In this subsection, we will present a chain encoding of terms of Ty2. To do this, we will define two things: First, a relation `chain2term` that holds of a pair $\langle c,o \rangle$, where c is a chain and o is a *me* object iff there is a term ϕ of Ty2 such that o corresponds to ϕ and c chain-encodes ϕ. This relation will be used to allow us to change freely between *me* objects and chains in the representation of terms. Second, we will define a function "#". This function maps every term of Ty2 ϕ to some RSRL description δ such that δ describes exactly those chains that are the chain-encoding of ϕ. This rough characterization already indicates that the function "#" is similar to the function "*" as given in Definition 3.16, but has the description of a chain as its output, not that of a *me* object. There is an intimate connection between the functions "*" and "#" and the relation `chain2term`: For each *me* object o and for each chain c: the pair $\langle c, o \rangle$ is in the relation `chain2term` iff there is some term ϕ such that o is described by ϕ^* and c is described by $\phi^{\#}$. Finally, we will define a third important relation, `chain-term`. This relation holds of a chain c iff there is a term ϕ that is chain-encoded by c. This relation is important, because it will allow us to single out exactly those chains which encode some term, without being obliged to state which term (or *me* object) is being chain-encoded.

We will encode Ty2 terms as chains that contain only objects of two distinct sorts. It is immaterial which sorts we choose, as long as none of them is a subsort of the other. This means that we could use the sorts *me* and *type*, but not for example the sorts *type* and *entity*. We know that each *me* object has at least two components, itself and its TYPE value. therefore, we consider it a reasonable choice to take the sorts *me* and *type*. Still we want to remain neutral with respect to the particular choice and simply write *s1* and *s2* for the two sorts. A chain encoding of a Ty2 term is a finite sequence of objects of the sorts *s1* and *s2*. In (226) we give the example of the chain encoding of the semantic type $\langle e, t \rangle$.

(226) The chain encoding of the semantic type $\langle e, t\rangle$:
$\langle$ *s2*, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*,
s2, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*,
s2, *s1*, *s1*, *s1*, *s1*,
s2, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*,
s2, *s1*, *s1*, *s1* $\rangle$

While for the time being, this encoding is quite opaque, the example already indicates that a chain used for encoding terms (and types and natural numbers) is internally structured: it consists of subsequences all of which start with an object of sort *s2*, followed by a finite number of *s1* objects. The chain encoding follows the encoding as linguistic objects presented in Chapter 3 quite closely. In particular, we chose the subsequences of a term-representing chain in a way that each of them represents some sort or attribute. For ease of notation, we write $s2s1^i$ for a subsequence that start with a *s2* object followed by i-many *s1* objects. In the example in (226), the subsequence $s2s1^6$, i.e., the first such subsequence, represents the sort name *complex-type*. The next subsequence, $s2s1^{12}$, represents the attribute name IN. The third subsequence, $s2s1^4$, stands for the species *entity*. The fourth subsequence, $s2s1^{13}$, is used to encode the attribute name OUT, and finally, the subsequence $s2s1^3$ encodes the sort name *truth*.

In order to be able to refer to subsequences of the form $s2s1^i$, we assume that the signature contains a family of relations `is-`i`-long` for each $1 \leq i \leq n$, with n being the sum of the number of species and attributes given by the signature under consideration.[4] For the encoding of terms of Ty2, it is enough to assume the relations `is-`i`-long` for the number of species and attributes of the grammar $\mathcal{TY}2$. The relations are defined as follows:

(227) The family of relations `is-`i`-long`:

$$\texttt{is-1-long}(a) \stackrel{\forall}{\Longleftarrow} \left({}^{a}\begin{bmatrix} \dagger & \textit{s1} \\ \triangleright & \textit{echain} \end{bmatrix} \right)$$

for each i, $2 \leq i \leq n$,

$$\texttt{is-}i\texttt{-long}(a) \stackrel{\forall}{\Longleftarrow} \left({}^{a}\begin{bmatrix} \dagger & \textit{s1} \\ \triangleright & b \end{bmatrix} \textbf{ and } \texttt{is-}(i-1)\texttt{-long}(b) \right)$$

The way we have defined terms of Ty2, semantic types and natural numbers (in the case of variables) occur as parts of term. For this reason, we will define the relation `chain2term` not only for chains and terms, but also for chains and numbers and for chains and semantic types. In fact, there will be a clause of the relation `chain2term` for each species in the signature of the grammar $\mathcal{TY}2$.

In (228) we give the two clauses needed for the chain encoding of natural numbers.

(228) The relation `chain2term` (`c2t`) as needed for natural numbers:

$$\texttt{c2t}(a, \boxed{1}) \stackrel{\forall}{\Longleftarrow}$$
$$ {}^{a}\begin{bmatrix} \dagger & \textit{s2} \\ \triangleright & b \end{bmatrix} \textbf{ and } {}^{\boxed{1}}\begin{bmatrix} \textit{zero} \end{bmatrix}$$
$$\textbf{and } \texttt{is-1-long}(b)$$

[4] This is, of course, only possible if the number of species and the number of attributes are finite.

$$
\begin{array}{l}
\texttt{c2t}(a, \boxed{1}) \stackrel{\forall}{\Longleftarrow} \\
\quad {}^{a}\begin{bmatrix} \dagger & \mathit{s2} \\ \triangleright & b \end{bmatrix} \textbf{ and } {}^{\boxed{1}}\begin{bmatrix} \mathit{non\text{-}zero} \\ \textsc{number}\ \boxed{2} \end{bmatrix} \\
\quad \textbf{and } \texttt{append}(c, d, b) \\
\quad \textbf{and } \texttt{is-2-long}(c) \\
\quad \textbf{and } {}^{d}\begin{bmatrix} \dagger & \mathit{s2} \\ \triangleright & e \end{bmatrix} \\
\quad \textbf{and } \texttt{append}(f, g, e) \\
\quad \textbf{and } \texttt{is-11-long}(f) \\
\quad \textbf{and } \texttt{c2t}(g, \boxed{2})
\end{array}
$$

To illustrate the relation `chain2term` as defined so far, consider the chain in (229a) and the description of a *number* object that corresponds to the number 1 in (229b).

(229) a. Chain encoding of the natural number 1:
⟨ *s2*, *s1*, *s1*,
s2, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*,
s2, *s1* ⟩

b. AVM description of a *number* object that corresponds to the natural number 1:

$$
\begin{bmatrix} \mathit{non\text{-}zero} \\ \textsc{number}\ \mathit{zero} \end{bmatrix}
$$

To see that the chain in (229a) and an object described by (229b) stand in the relation `chain2term`, we must first consider the second clause of (228): The first element on the chain is of sort *s2*, the *number* object is of sort *non-zero*. The rest of the chain (b) can be split into two subsequences c and d. d starts with an object of sort *s2* and its rest (e) can be split into two subsequences f and g. In (230) we give the four chains c, e, f and g.

(230) $c = \langle \mathit{s1}, \mathit{s1} \rangle$
$e =$ ⟨ *s2*, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*, *s1*,
s2, *s1* ⟩
$f = \langle \mathit{s1}, \mathit{s1}, \mathit{s1}, \mathit{s1}, \mathit{s1}, \mathit{s1}, \mathit{s1}, \mathit{s1}, \mathit{s1}, \mathit{s1}, \mathit{s1} \rangle$
$g = \langle \mathit{s2}, \mathit{s1} \rangle$

As required, the chain c is in the relation `is-2-long`, i.e., it is a sequence that contains exactly two *s1* objects. f is a sequence that contains exactly eleven *s1* objects. Therefore it is in the relation `is-11-long`. Next, we must check whether the pair consisting of g and the NUMBER value ($\boxed{2}$) of the object described by (229b) meets one of the clauses in (228).

We can show that this pair meets the first clause. The chain g as given in (230) starts with an object of sort *s2*. The NUMBER value of an object described by the AVM in (229b) is of sort *zero*. The remaining chain of g contains one *s1* object, i.e., it is in the relation `is-1-long`. Thus, the pair satisfies the first clause in (228), and, consequently, the pair consisting of the chain given in (229b) and an object that is s described by (229b) satisfy the second clause in (228).

The next kind of entities that occurs in terms of Ty2 are semantic types. Within the grammar $\mathcal{TY}2$, semantic types are expressed by objects of the sort *type*. There are four subsorts of *type*. In (231) we give the clauses of the relation `chain2type` that correspond to these four subsorts.

(231) The relation `chain2term` (`c2t`) as needed for semantic types:

$$
\begin{array}{l}
\texttt{c2t}(a, \boxed{1}) \stackrel{\forall}{\Longleftarrow} \\
\quad {}^{a}\begin{bmatrix} \dagger & \mathit{s2} \\ \triangleright & b \end{bmatrix} \textbf{ and } {}^{\boxed{1}}\begin{bmatrix} \mathit{truth} \end{bmatrix} \\
\quad \textbf{and } \texttt{is-3-long}(b)
\end{array}
$$

$$\texttt{c2t}(a,\boxed{1}) \overset{\forall}{\Longleftarrow} {}^{a}\begin{bmatrix}\dagger & \textit{s2}\\ \triangleright & b\end{bmatrix} \textbf{ and } \boxed{1}\begin{bmatrix}\textit{entity}\end{bmatrix} \textbf{ and } \texttt{is-4-long}(b)$$

$$\texttt{c2t}(a,\boxed{1}) \overset{\forall}{\Longleftarrow} {}^{a}\begin{bmatrix}\dagger & \textit{s2}\\ \triangleright & b\end{bmatrix} \textbf{ and } \boxed{1}\begin{bmatrix}\textit{w-index}\end{bmatrix} \textbf{ and } \texttt{is-5-long}(b)$$

$$\texttt{c2t}(a,\boxed{1}) \overset{\forall}{\Longleftarrow} {}^{a}\begin{bmatrix}\dagger & \textit{s2}\\ \triangleright & b\end{bmatrix} \textbf{ and } \boxed{1}\begin{bmatrix}\textit{complex-type}\\ \text{IN} & \boxed{2}\\ \text{OUT} & \boxed{3}\end{bmatrix}$$
$$\textbf{and } \texttt{append}(c,d,b) \textbf{ and } \texttt{is-6-long}(c) \textbf{ and } \texttt{append}(e,f,d)$$

$$\textbf{and } {}^{e}\begin{bmatrix}\dagger & \textit{s2}\\ \triangleright & g\end{bmatrix} \textbf{ and } \texttt{append}(h,i,g) \textbf{ and } \texttt{is-12-long}(h) \textbf{ and } \texttt{c2t}(i,\boxed{2})$$

$$\textbf{and } {}^{f}\begin{bmatrix}\dagger & \textit{s2}\\ \triangleright & j\end{bmatrix} \textbf{ and } \texttt{append}(k,l,j) \textbf{ and } \texttt{is-13-long}(k) \textbf{ and } \texttt{c2t}(l,\boxed{3})$$

The first three clauses in (231) relate chains to the atomic subsorts of *type*. They are parallel to the first clause in (228). The last clause in (231) is relatively complex. It can be seen to consist of three parts: the first part contains up to the relation call $\texttt{append}(f, g, e)$. The second and third parts correspond to the blocks below this relation call.

Intuitively the first block checks whether the initial $\textit{s2s1}^i$ subsequence of a corresponds to the sort *c-type*, the `append` relation, then, splits the rest of the chain a in such a way that the first subsequence, f, corresponds to the IN value of the *type* object $\boxed{1}$ and the second subsequence, g corresponds to the OUT value of $\boxed{1}$. In the second and the third part of the clause e and f are checked with respect to these components of $\boxed{1}$. This can be seen, because we take the sequence $\textit{s2s1}^{12}$ to correspond to the IN attribute, and the sequence $\textit{s2s1}^{13}$ to correspond the the OUT attribute. These sequences are the initial subsequences of f and g respectively. The remaining sequences i and l, then, are required to stand in the relation `c2t` with the IN and the OUT value of $\boxed{1}$.

We can now turn back to our initial example of the chain encoding of the semantic type $\langle e, t\rangle$ in (226). If our specification of the relation `chain2term` given in (231) is correct, then the chain in (226) should stand in the relation `chain2term` with a *type* object that is described by the following AVM.

(232) $$\begin{bmatrix}\textit{complex-type}\\ \text{IN} & \textit{entity}\\ \text{OUT} & \textit{truth}\end{bmatrix}$$

To show that this is the case, we give in the subsequences of the chain in (226) as required by the second clause in (231). We use the abbreviatory notation $\textit{s2s1}^i$.

(233) $a = \langle \textit{s2s1}^6, \textit{s2s1}^{12}, \textit{s2s1}^4, \textit{s2s1}^{13}, \textit{s2s1}^3 \rangle$
$b = \langle \textit{s1}^6, \textit{s2s1}^{12}, \textit{s2s1}^4, \textit{s2s1}^{13}, \textit{s2s1}^3 \rangle$
$c = \langle \textit{s1}^6 \rangle$
$d = \langle \textit{s2s1}^{12}, \textit{s2s1}^4, \textit{s2s1}^{13}, \textit{s2s1}^3 \rangle$

$$\begin{aligned}
e &= \left\langle \textit{s2s1}^{12}, \textit{s2s1}^{4} \right\rangle \\
&\quad g = \left\langle \textit{s1}^{12}\textit{s2s1}^{4} \right\rangle \\
&\quad h = \left\langle \textit{s1}^{12} \right\rangle \\
&\quad i = \left\langle \textit{s2s1}^{4} \right\rangle \\
f &= \left\langle \textit{s2s1}^{13}, \textit{s2s1}^{3} \right\rangle \\
&\quad j = \left\langle \textit{s1}^{13}, \textit{s2s1}^{3} \right\rangle \\
&\quad k = \left\langle \textit{s1}^{13} \right\rangle \\
&\quad l = \left\langle \textit{s2s1}^{3} \right\rangle
\end{aligned}$$

The second clause in (231) contains two calls for the relation `chain2term`. Therefore, we must check whether the chain i and the chain l stand in the relation `chain2term` with the IN and the OUT value of an object described by the AVM in (232). To verify this, we can use the non-recursive clauses of the relation `chain2term`: The chain i is of the form $\textit{s2s1}^4$. By the second clause in (231), this chain is in the relation with a linguistic object o iff o is of sort *entity*. This is the case for the IN value of an object described by (232). Parallel to this, the chain l is of the form $\textit{s2s1}^3$, which corresponds to objects of the sort *truth* according to the first clause in (231).

We have, thus, shown that the chain c given as our initial example in (226) and an object o described by (232) are in the relation `chain2type`. There is a semantic type, $\langle e, t \rangle$, which corresponds to o. Alternatively, we can consider c the *chain encoding* of the type $\langle e, t \rangle$.

With the clauses of the relation `chain2term` that we stated in (228) and (231), we illustrated the chain encoding of atomic sorts (the first clause in (228) and the first three clauses in (231)). The second clause in (228) is an instance of the chain encoding for a sort for which one attribute is appropriate. The fourth clause in (231) can be considered an instance of the chain encoding of a sort for which there are two attributes appropriate.

If we want to encode terms of Ty2 proper, we must write clauses for the species below *me*. As can be seen from the sort hierarchy and the appropriateness conditions given for the sort *me* in (121), there is no atomic subsort of *me*. The species below *const* have one attribute appropriate (TYPE). Thus, the clauses of the relation `chain2term` for these species are of the same form as the last clause in (228). There are two attributes that are appropriate for the sort *var*, TYPE and NUMBER. The clause for the chain encoding of variables is, thus, built after that for the encoding of complex semantic types, i.e., the last clause in (231). The other subsorts of *me* — *appl*, *abstr*, and *equ* — all require three attributes. Therefore the corresponding three clauses of the relation `chain2term` are all of the same structure, but are slightly more complicated than the other clauses.

In (234) we give the necessary clauses for the chain encoding of terms of Ty2. For a better overview over the encoding, Table 4.1 indicates which subsequence of the form $\textit{s2s1}^i$ corresponds to which species or attribute.

(234) The relation `chain2term` (`c2t`) as needed for terms of Ty2:

for each species $\textit{const}_i \sqsubseteq \textit{const}$,

$$\begin{aligned}
&\texttt{c2t}(a, \boxed{1}) \overset{\forall}{\Longleftarrow} \\
&\qquad {}^{a}\begin{bmatrix} \dagger & \textit{s2} \\ \triangleright & b \end{bmatrix} \textbf{ and } {}^{\boxed{1}}\begin{bmatrix} \textit{const}_i \\ \text{TYPE} \ \boxed{2} \end{bmatrix} \\
&\qquad \textbf{and } \texttt{append}(c, d, b) \\
&\qquad \textbf{and } \texttt{is-}(i+21)\texttt{-long}(c) \\
&\qquad \textbf{and } {}^{d}\begin{bmatrix} \dagger & \textit{s2} \\ \triangleright & e \end{bmatrix} \\
&\qquad \textbf{and } \texttt{append}(f, g, e) \\
&\qquad \textbf{and } \texttt{is-14-long}(f) \\
&\qquad \textbf{and } \texttt{c2t}(g, \boxed{2})
\end{aligned}$$

TABLE 4.1. Chart used for the chain encoding of *me* objects:

species	number	attribute	number
zero	1	NUMBER	11
non-zero	2		
truth	3	IN	12
entity	4	OUT	13
w-index	5		
complex-type	6		
variable	7	TYPE	14
application	8	FUNCTOR	15
abstraction	9	ARGUMENT	16
equation	10	VAR	17
$const_1$	22	ARG1	18
		ARG2	19
⋮	⋮		
$const_n$	$n+21$		

For the sort *var*:

$\texttt{c2t}(a,\boxed{1}) \overset{\forall}{\Longleftarrow}$

$${}^{a}\begin{bmatrix} \dagger & \mathit{s2} \\ \triangleright & b \end{bmatrix} \textbf{ and } \boxed{1}\begin{bmatrix} \mathit{var} & \\ \textsc{type} & \boxed{2} \\ \textsc{number} & \boxed{3} \end{bmatrix}$$

and $\texttt{append}(c,d,b)$
and $\texttt{is-7-long}(c)$
and $\texttt{append}(e,f,d)$

and ${}^{e}\begin{bmatrix} \dagger & \mathit{s2} \\ \triangleright & g \end{bmatrix}$
and $\texttt{append}(h,i,g)$
and $\texttt{is-14-long}(h)$
and $\texttt{c2t}(i,\boxed{2})$

and ${}^{f}\begin{bmatrix} \dagger & \mathit{s2} \\ \triangleright & j \end{bmatrix}$
and $\texttt{append}(k,l,j)$
and $\texttt{is-11-long}(k)$
and $\texttt{c2t}(l,\boxed{3})$

for the sort *appl*:

$\texttt{c2t}(a,\boxed{1}) \overset{\forall}{\Longleftarrow}$

$${}^{a}\begin{bmatrix} \dagger & \mathit{s2} \\ \triangleright & b \end{bmatrix} \textbf{ and } \boxed{1}\begin{bmatrix} \mathit{appl} & \\ \textsc{type} & \boxed{2} \\ \textsc{func} & \boxed{3} \\ \textsc{arg} & \boxed{4} \end{bmatrix}$$

and $\texttt{append}(c,d,b)$
and $\texttt{is-8-long}(c)$
and $\texttt{append}(e,l,d)$
and $\texttt{append}(f,m,l)$

and ${}^{e}\begin{bmatrix} \dagger & \mathit{s2} \\ \triangleright & g \end{bmatrix}$
and $\texttt{append}(h,i,g)$
and $\texttt{is-14-long}(h)$
and $\texttt{c2t}(i,\boxed{2})$

and ${}^{f}\begin{bmatrix} \dagger & \mathit{s2} \\ \triangleright & j \end{bmatrix}$
and $\texttt{append}(k,l,j)$
and $\texttt{is-15-long}(k)$
and $\texttt{c2t}(l,\boxed{3})$

and ${}^{m}\begin{bmatrix} \dagger & \mathit{s2} \\ \triangleright & n \end{bmatrix}$
and $\texttt{append}(o,p,n)$
and $\texttt{is-16-long}(o)$
and $\texttt{c2t}(p,\boxed{4})$

for the sort *abstr*: $\mathtt{c2t}(a,\boxed{1}) \overset{\forall}{\Longleftarrow}$

$$
{}^{a}\begin{bmatrix} \dagger & \mathit{s2} \\ \triangleright & b \end{bmatrix} \textbf{ and } \boxed{1}\begin{bmatrix} \mathit{abstr} & \\ \text{TYPE} & \boxed{2} \\ \text{VAR} & \boxed{3} \\ \text{ARG} & \boxed{4} \end{bmatrix}
$$

$\textbf{and } \mathtt{append}(c,d,b)$
$\textbf{and } \mathtt{is\text{-}9\text{-}long}(c)$
$\textbf{and } \mathtt{append}(e,l,d)$
$\textbf{and } \mathtt{append}(f,m,l)$

$$
\begin{array}{lll}
\textbf{and } {}^{e}\begin{bmatrix} \dagger & \mathit{s2} \\ \triangleright & g \end{bmatrix} & \textbf{and } {}^{f}\begin{bmatrix} \dagger & \mathit{s2} \\ \triangleright & j \end{bmatrix} & \textbf{and } {}^{m}\begin{bmatrix} \dagger & \mathit{s2} \\ \triangleright & n \end{bmatrix} \\
\textbf{and } \mathtt{append}(h,i,g) & \textbf{and } \mathtt{append}(k,l,j) & \textbf{and } \mathtt{append}(o,p,n) \\
\textbf{and } \mathtt{is\text{-}14\text{-}long}(h) & \textbf{and } \mathtt{is\text{-}17\text{-}long}(k) & \textbf{and } \mathtt{is\text{-}16\text{-}long}(o) \\
\textbf{and } \mathtt{c2t}(i,\boxed{2}) & \textbf{and } \mathtt{c2t}(l,\boxed{3}) & \textbf{and } \mathtt{c2t}(p,\boxed{4})
\end{array}
$$

for the sort *equ*:
$\mathtt{c2t}(a,\boxed{1}) \overset{\forall}{\Longleftarrow}$

$$
{}^{a}\begin{bmatrix} \dagger & \mathit{s2} \\ \triangleright & b \end{bmatrix} \textbf{ and } \boxed{1}\begin{bmatrix} \mathit{equ} & \\ \text{TYPE} & \boxed{2} \\ \text{ARG1} & \boxed{3} \\ \text{ARG2} & \boxed{4} \end{bmatrix}
$$

$\textbf{and } \mathtt{append}(c,d,b)$
$\textbf{and } \mathtt{is\text{-}10\text{-}long}(c)$
$\textbf{and } \mathtt{append}(e,l,d)$
$\textbf{and } \mathtt{append}(f,m,l)$

$$
\begin{array}{lll}
\textbf{and } {}^{e}\begin{bmatrix} \dagger & \mathit{s2} \\ \triangleright & g \end{bmatrix} & \textbf{and } {}^{f}\begin{bmatrix} \dagger & \mathit{s2} \\ \triangleright & j \end{bmatrix} & \textbf{and } {}^{m}\begin{bmatrix} \dagger & \mathit{s2} \\ \triangleright & n \end{bmatrix} \\
\textbf{and } \mathtt{append}(h,i,g) & \textbf{and } \mathtt{append}(k,l,j) & \textbf{and } \mathtt{append}(o,p,n) \\
\textbf{and } \mathtt{is\text{-}14\text{-}long}(h) & \textbf{and } \mathtt{is\text{-}18\text{-}long}(k) & \textbf{and } \mathtt{is\text{-}19\text{-}long}(o) \\
\textbf{and } \mathtt{c2t}(i,\boxed{2}) & \textbf{and } \mathtt{c2t}(l,\boxed{3}) & \textbf{and } \mathtt{c2t}(p,\boxed{4})
\end{array}
$$

It is useful to introduce an abbreviatory notation for the chain encoding of terms. In the following, we will use the number i as an abbreviation for a subsequence of the form $\mathit{s2s1}^i$. The chart in Table 4.1 lists which number corresponds to which species or attribute. In (235) we give some examples for the chain encoding of some simple terms. The first term is the variable $v_{s,0}$, for which we usually write @, the second term is the variable $v_{(se),0}$, and the third term is the the functional application of these two terms.

(235)	term	chain encoding
	$v_{s,0}$	$\langle 7,14,5,11,1\rangle$
	$v_{(se),0}$	$\langle 7,14,6,12,5,13,4,11,1\rangle$
	$(v_{(se),0}v_{s,0})_e$	$\langle 8,14,4,15,7,14,6,12,5,13,4,11,1,16,7,14,5,11,1\rangle$

To illustrate that the chain encoding is more straightforward then it might seem at the first place, we can re-write the last chain of (235) as an indentation structure. Then, it looks like the AVM description of an *me* object that corresponds to the encoded term, where the sort and attribute names are replaced by the corresponding numbers from Table 4.1.

(236) term — chain encoding — description of a *me* object

$$(v_{(se),0}v_{s,0})_e \qquad \begin{bmatrix} \mathit{8}, & & \\ 14, & \mathit{4}, & \\ 15, & \begin{bmatrix} \mathit{7}, & \\ 14, & \begin{bmatrix} \mathit{6}, & \\ 12, & \mathit{5}, \\ 13, & \mathit{4}, \end{bmatrix} \\ 11, & \mathit{1}, \end{bmatrix} & \\ 16, & \begin{bmatrix} \mathit{7}, & \\ 14, & \mathit{5}, \\ 11, & \mathit{1} \end{bmatrix} & \end{bmatrix} \qquad \begin{bmatrix} \mathit{appl} & \\ \textsc{type} & \mathit{entity} \\ \textsc{func} & \begin{bmatrix} \mathit{var} & \\ \textsc{type} & \begin{bmatrix} \mathit{c\text{-}type} & \\ \textsc{in} & \mathit{w\text{-}index} \\ \textsc{out} & \mathit{entity} \end{bmatrix} \\ \textsc{number} & \mathit{zero} \end{bmatrix} \\ \textsc{arg} & \begin{bmatrix} \mathit{var} & \\ \textsc{type} & \mathit{w\text{-}index} \\ \textsc{number} & \mathit{zero} \end{bmatrix} \end{bmatrix}$$

With the definition of the relation `chain2term` we have established a relation between *me* objects and chains. For terms of Ty2, we were able to give a function "*" that maps every term to a description of some indiscernibility class of *me* objects (see Section 3.4). Parallel to this function from terms to $\mathcal{TY}2$ descriptions of *me* objects, we can give a function "#" from terms $\mathcal{TY}2$ descriptions of chains.

The definition of the function "#" is similar to the clauses of the relation `chain2term`, in particular, there are four different kinds of cases to distinguish: (i) the atomic cases, i.e., the way the function is defined for the number 0 and the types t, e and s; (ii) the "unary" cases, i.e. the definition of the function on numbers of the form $i + 1$ and on non-logical constants; (iii) the "binary" cases, i.e., the case of types of the form $\langle\tau, \tau'\rangle$ and the case of variables; and (iv) the "ternary" cases of terms of the form $(\phi\psi)$, $\lambda x.\phi$ and $(\phi = \psi)$. In Definition 4.6 we define the function "#" for each one of these cases, and indicate in what respect the other parallel cases differ.

DEFINITION 4.6

"#" is a function from $\mathbb{N} \cup \mathit{Type} \cup \mathit{Ty2}$ *to the set of* $\mathcal{TY}2$ *descriptions such that,*

atomic cases:

if $x = 0$, *then*

$$i^{\#} = \left(\begin{array}{l} a\dagger{\sim}\mathit{s2} \\ \textbf{and } \texttt{is-1-long}(a\triangleright) \end{array} \right)$$

for the other atomic cases, replace the call `is-1-long`$(a\triangleright)$ *as indicated in the following table:*

0	t	e	s
`is-1-long`$(a\triangleright)$	`is-3-long`$(a\triangleright)$	`is-4-long`$(a\triangleright)$	`is-5-long`$(a\triangleright)$

unary cases:

if $x = y + 1$, *then*

$$x^{\#} = \left(\begin{array}{l} \mathbf{E}b\ \mathbf{E}c\ \mathbf{E}d\ \mathbf{E}f\ \mathbf{E}g \\ a\dagger{\sim}\mathit{s2} \\ \textbf{and } \texttt{append}(c,d,a\triangleright) \\ \textbf{and } \texttt{is-2-long}(c) \\ \textbf{and } d\dagger{\sim}\mathit{s2} \\ \textbf{and } \texttt{append}(f,g,d\triangleright) \\ \textbf{and } \texttt{is-11-long}(f) \\ \textbf{and } y^{\#}[g/a] \end{array} \right)$$

for each $c \in \mathit{Const}$, *where* const_i *is the species that corresponds to* c, *the definition of* $(c_\tau)^{\#}$ *is just as given for* $y+1$, *but with the following replacements:*

$y+1$	c_τ
is-2-long($a\triangleright$)	is-$(i+21)$-long($a\triangleright$)
is-11-long(f)	is-14-long(f)
$y^{\#}[g/a]$	$\tau^{\#}[g/a]$

binary cases:
if $x = \langle \tau, \tau' \rangle$, then

$$x^{\#} = \left(\begin{array}{ll} \mathbf{E}b\ \mathbf{E}c\ \mathbf{E}d\ \mathbf{E}f\ \mathbf{E}g\ \mathbf{E}h\ \mathbf{E}i\ \mathbf{E}j\ \mathbf{E}k\ \mathbf{E}l & \\ a\dagger\!\sim\! s2 & \\ \textbf{and}\ \mathtt{append}(c,d,a\triangleright) & \\ \textbf{and}\ \mathtt{is\text{-}6\text{-}long}(c) & \\ \textbf{and}\ d\dagger\!\sim\! s2 & \\ \textbf{and}\ \mathtt{append}(e,f,d) & \\ \textbf{and}\ e\dagger\!\sim\! s2 & \textbf{and}\ f\dagger\!\sim\! s2 \\ \textbf{and}\ \mathtt{append}(h,i,e\triangleright) & \textbf{and}\ \mathtt{append}(k,l,f\triangleright) \\ \textbf{and}\ \mathtt{is\text{-}12\text{-}long}(h) & \textbf{and}\ \mathtt{is\text{-}13\text{-}long}(k) \\ \textbf{and}\ \tau^{\#}[i/a] & \textbf{and}\ \tau'^{\#}[l/a] \end{array} \right)$$

For $x = v_{\tau,n}$, do the following replacements:

$\langle \tau, \tau' \rangle$	$v_{\tau,n}$
is-6-long(c)	is-7-long(c)
is-12-long(h)	is-14-long(h)
is-13-long(k)	is-11-long(k)
$\tau'^{\#}[l/a]$	$n^{\#}[l/a]$

ternary cases:
if $x = (\phi\psi)_\tau$, then

$$x^{\#} = \left(\begin{array}{ll} \mathbf{E}b\ \mathbf{E}c\ \mathbf{E}d\ \mathbf{E}f\ \mathbf{E}g\ \mathbf{E}h\ \mathbf{E}i\ \mathbf{E}j\ \mathbf{E}k\ \mathbf{E}l\ \mathbf{E}m\ \mathbf{E}o\ \mathbf{E}p & \\ a\dagger\!\sim\! s2 & \\ \textbf{and}\ \mathtt{append}(c,d,a\triangleright) & \textbf{and}\ e\dagger\!\sim\! s2 \\ \textbf{and}\ \mathtt{is\text{-}8\text{-}long}(c) & \textbf{and}\ \mathtt{append}(h,i,e\triangleright) \\ \textbf{and}\ d\dagger\!\sim\! s2 & \textbf{and}\ \mathtt{is\text{-}14\text{-}long}(h) \\ \textbf{and}\ \mathtt{append}(e,l,d) & \textbf{and}\ \tau^{\#}[i/a] \\ \textbf{and}\ \mathtt{append}(f,m,d) & \\ \textbf{and}\ f\dagger\!\sim\! s2 & \textbf{and}\ m\dagger\!\sim\! s2 \\ \textbf{and}\ \mathtt{append}(k,l,f\triangleright) & \textbf{and}\ \mathtt{append}(o,p,m\triangleright) \\ \textbf{and}\ \mathtt{is\text{-}15\text{-}long}(k) & \textbf{and}\ \mathtt{is\text{-}16\text{-}long}(o) \\ \textbf{and}\ \phi^{\#}[l/a] & \textbf{and}\ \psi^{\#}[p/a] \end{array} \right)$$

for the other two ternary cases, replace the indicated parts of the definition of $(\phi\psi)_\tau$:

$(\phi\psi)_\tau$	$(\lambda x.\phi)_\tau$	$(\phi = \psi)_\tau$
is-8-long(c)	is-9-long(c)	is-10-long(c)
is-15-long(k)	is-17-long(k)	is-18-long(k)
$\phi^{\#}[l/a]$	$x^{\#}[l/a]$	$\phi^{\#}[l/a]$
is-16-long(o)	is-16-long(o)	is-19-long(o)
$\psi^{\#}[p/a]$	$\phi^{\#}[p/a]$	$\psi^{\#}[p/a]$

If one compares the lines in the definition of the function "#" to the clauses given for the relation `chain2term` in (228), (231) and (234), there is an obvious correspondence. To highlight this correspondence, we have used the same variables. The difference between the two, then, amounts to two things: First, there is no *me* object involved in Definition 4.6. Second, whenever there is a call `chain2term`(c, $\boxed{1}$) in the clauses of the relation, there is a line $x^{\#}[c/a]$ in the definition of the function "#".

It should also be noted that every description $x^{\#}$ contains a free variable a, whereas the description x^* has no free variables. In the case of the function "*" we can say that x^* describes a linguistic object u iff $u \in D_I(x^*)$. For the function "#" we would like to say that the description $x^{\#}$ describes a chain. For this purpose, however, we cannot use the description interpretation function D directly, as the interpretation of a description is always a set of linguistic objects, i.e., it does not contain chains in its denotation. In Definition 4.7 we define what it means for a chain to be described by some description.

DEFINITION 4.7

For each interpretation $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$*, and for each description* δ *that contains exactly one free variable* $\boxed{i}$*,*

$$\begin{array}{l} A\ chain\ c \in \mathsf{U}^* \text{ is described by } \delta \\ \quad iff\ for\ some\ ass \in \mathsf{Ass_I}\ with\ ass(\boxed{i}) = c, \\ \qquad D_{\mathsf{I}}^{ass}(\delta) = \left\{ u \in \mathsf{U} \,\middle|\, c \in (\mathsf{Co}_{\mathsf{I}}^{u})^* \right\} \neq \emptyset. \end{array}$$

Definition 4.7 states that a chain c can be described by some description δ if δ contains exactly one free variable. For c to be described by such a description, there must be a variable assignment that assigns the chain c to the free variable of δ and the interpretation of δ under this variable assignment is the set of all objects u such that c contains only components of u. Intuitively this means that a description δ describes some chain c if δ is true of all objects that contain the chain c in their domain of quantification. Adding the requirement that this set be not empty, we ensure that only those chains can be described in the sense of Definition 4.7 whose elements are components of an object in the universe.

Given this definition, we can see what it means to say that the chain in (226) is described by $\langle e, t \rangle^{\#}$. In (237) we re-state the chain given in (226), using our abbreviatory convention, and the description $\langle e, t \rangle^{\#}$

(237) a. $\langle 6, 12, 4, 13, 3 \rangle$

b. $\langle e, t \rangle^{\#} =$

$$\left(\begin{array}{ll} \multicolumn{2}{l}{\mathbf{E}b\ \mathbf{E}c\ \mathbf{E}d\ \mathbf{E}f\ \mathbf{E}g\ \mathbf{E}h\ \mathbf{E}i\ \mathbf{E}j\ \mathbf{E}k\ \mathbf{E}l} \\ \multicolumn{2}{l}{a\dagger{\sim}s2} \\ \multicolumn{2}{l}{\textbf{and } \texttt{append}(c,d,a\triangleright)} \\ \multicolumn{2}{l}{\textbf{and } \texttt{is-6-long}(c)} \\ \multicolumn{2}{l}{\textbf{and } d\dagger{\sim}s2} \\ \multicolumn{2}{l}{\textbf{and } \texttt{append}(e,f,d)} \\ \textbf{and } e\dagger{\sim}s2 & \textbf{and } f\dagger{\sim}s2 \\ \textbf{and } \texttt{append}(h,i,e\triangleright) & \textbf{and } \texttt{append}(k,l,f\triangleright) \\ \textbf{and } \texttt{is-12-long}(h) & \textbf{and } \texttt{is-13-long}(k) \\ \quad \textbf{and } i\dagger{\sim}s2 & \quad \textbf{and } l\dagger{\sim}s2 \\ \quad \textbf{and } \texttt{is-4-long}(i\triangleright) & \quad \textbf{and } \texttt{is-3-long}(l\triangleright) \end{array} \right)$$

The description in (237b) contains exactly one free variable, a. We can, therefore, ask whether it describes the chain in (237a) according to Definition 4.7. A brief comparison of the numbers in the abbreviated notation of c with the calls `is-`i`-long` in (237b) shows that this is the case.

The description $(et)^{\#}$ describes the chain in (237a) iff interpreting the variable a as this chain makes the description true for every object in the universe which has every member of the chain in (237a) as a component. The description in (237b) requires the chain referred to by a to start with a *s2* object, followed by six *s1* objects. As the chains c starts with a subsequence $s2s1^6$, it satisfies this requirement. Going through the rest of the chain c and the other lines in (237b) in an analogous way shows that the chain c is described by $(et)^{\#}$.

The way we have we have defined the grammar $\mathcal{TY}2$ in Chapter 3, only congruent configurations of objects correspond to the same entity in the semantic representation language. For the chain encoding, we will require that two chains c_1 and c_2 correspond to the same entity in the semantic representation language, if (i) they are of the same length, (ii) they only contain elements of the sort *s1* or *s2*, and (iii) the order of the *s1* and *s2* objects is the same in c_1 and c_2. This connection is formally expressed by the definition of the relation `are-chain-copies`:

(238) The relation `are-chain-copies`:

$$\texttt{are-chain-copies}(x,y) \overset{\forall}{\Longleftarrow} \left({}^{x}\Big[\textit{echain} \Big] \textbf{ and } {}^{y}\Big[\textit{echain} \Big] \right)$$

$$\texttt{are-chain-copies}(x,y) \overset{\forall}{\Longleftarrow} \left(\begin{array}{l} {}^{x}\begin{bmatrix} \dagger & \textit{s1} \\ \rhd & \boxed{1} \end{bmatrix} \textbf{ and } {}^{y}\begin{bmatrix} \dagger & \textit{s1} \\ \rhd & \boxed{2} \end{bmatrix} \\ \textbf{and } \texttt{are-chain-copies}(\boxed{1},\boxed{2}) \end{array} \right)$$

$$\texttt{are-chain-copies}(x,y) \overset{\forall}{\Longleftarrow} \left(\begin{array}{l} {}^{x}\begin{bmatrix} \dagger & \textit{s2} \\ \rhd & \boxed{1} \end{bmatrix} \textbf{ and } {}^{y}\begin{bmatrix} \dagger & \textit{s2} \\ \rhd & \boxed{2} \end{bmatrix} \\ \textbf{and } \texttt{are-chain-copies}(\boxed{1},\boxed{2}) \end{array} \right)$$

Given this relation, it can be shown that for each *me* object o and for each two chains c_1 and c_2, if both pairs $\langle c_1, o\rangle$ and $\langle c_2, o\rangle$ are in the relation `chain2term`, then the pair $\langle c_1, c_2\rangle$ is in the relation `are-chain-copies`. Furthermore, we can show that the function "#" characterizes a chain up to copyhood, i.e., for each term x, and for each chains c_1 and c_2, if $x^{\#}$ describes c_1 and c_2, then the pair $\langle c_1, c_2\rangle$ is in the relation `are-chain-copies`.

We now have defined a relation, `chain2term`, that relates some chains to objects of sort *number*, *type* and *me*. In addition, the function "#" expresses a relation between natural numbers, semantic types and terms of Ty2 on the one side and chains on the other side. The relation `chain2term` expresses the correspondence between some chain and some object in the model of the grammar. The function "#" expresses a correspondence between the entities of our semantic representation language and some chains. What we are missing, is a characterization of the circumstances in which a chain is such that it corresponds to some object in the linguistic universe or to some entity in the semantic representation language.

Such a characterization is encoded in the relations `chain-number`, `chain-type` and `chain-term`. These relations are defined so that they mimic the signature and the theory of the corresponding parts of the grammar $\mathcal{TY}2$. As chains are finite and acyclic by definition, it is neither necessary to include a finiteness requirement such as the GFP in (131c) on term encoding chains, nor are we forced to exclude cyclic chains explicitly.

In (239), we define a relation `chain-number` that holds for a chain iff this chain encodes a *number* object (or, equivalently a natural number).

(239) The relation `chain-number`:

$$\texttt{chain-number}(x) \overset{\forall}{\Longleftarrow} x = \langle 1 \rangle$$

$$\texttt{chain-number}(x) \overset{\forall}{\Longleftarrow} \left(\begin{array}{l} x = \langle 2, 11 \rangle \oplus n \\ \textbf{and } \texttt{chain-number}(n) \end{array} \right)$$

The two clauses of the relation `chain-number` correspond to the information of *number* objects given by the grammar $\mathcal{TY}2$. In the first case, the chain is just the sequence $s2s1^1$. According to Table 4.1, this corresponds to the species *zero*. The second clause describes a chain c iff the first $s2s1^i$ subsequence of c has two *s1* objects. In this case, it corresponds to the sort *non-zero* according to Table 4.1. The next $s2s1^i$ sequence of c is required to contain eleven *s1* objects, i.e., to correspond to the attribute name NUMBER. Finally, the remaining subsequence of c, indicated by n in the clause, must be in the relation `chain-number` as well. This corresponds to the appropriateness conditions of the sort *non-zero* given in (108), where the sort *number* is appropriate to the attribute NUMBER.

In Section 3.1.1, where we introduced the sort *number* we imposed a constraint that each number object has a component of sort *zero*. In the case of the chain-encoding, this follows automatically, because chains are finite. Thus, the only way that a chain would be in the relation `chain-number` is if its final $s2s1^i$ subsequence is $s2s1^1$.

The relation `chain-type` is parallel to the relation `chain-number`. A chain c is in the relation `chain-type` iff it corresponds to a *type* object in an exhaustive model of $\mathcal{TY}2$, or, equivalently, iff it corresponds to some semantic type. The relation is defined in (240). The four clauses correspond to the four subsorts of *type*.

(240) The relation `chain-type`:

$$\begin{array}{lcl}
\texttt{chain-type}(x) & \overset{\forall}{\Longleftarrow} & x = \langle 3 \rangle \\
\texttt{chain-type}(x) & \overset{\forall}{\Longleftarrow} & x = \langle 4 \rangle \\
\texttt{chain-type}(x) & \overset{\forall}{\Longleftarrow} & x = \langle 5 \rangle \\
\texttt{chain-type}(x) & \overset{\forall}{\Longleftarrow} & \begin{pmatrix} x = \langle 6, 12 \rangle \oplus i \oplus \langle 13 \rangle \oplus o \\ \textbf{and } \texttt{chain-type}(i) \\ \textbf{and } \texttt{chain-type}(o) \end{pmatrix}
\end{array}$$

The first three clauses in (240) correspond to the three atomic subsorts of *type*. The chains that meet these clauses are of the form $s2s1^3$, $s2s1^4$ or $s2s1^5$, which correspond to the sorts *truth*, *entity* and *w-index* respectively according to Table 4.1.

In the last clause in (240), a chain c is described iff c's initial $s2s1^i$ sequence contains exactly six *s1* objects, i.e., corresponds to the sort *c-type*. In addition, it must be possible to split the rest of c in two chains $c_1 = \langle 12 \rangle \oplus i$ and $c_2 = \langle 13 \rangle \oplus o$, where i and o are in the relation `chain-type` as well. It can be seen in Table 4.1 that the numbers 11 and 12 represent the attribute names IN and OUT respectively. This shows that the relation `chain-type` mimics the appropriateness conditions of the sort *type*.

It is helpful for the reading descriptions that use chain encodings of elements of the semantic representation language to see that a characterization of a chain as given in the first line in the second clause of (240) can be rendered in a form which looks almost like an AVM and, then, is parallel to a description of a *me* object. We have already used such a notation in (236). The first line of the second clause would look as in (241):

(241)
$$x = \begin{array}{ll} \langle 6 \rangle & \oplus \\ \langle 12 \rangle & \oplus i \oplus \\ \langle 13 \rangle & \oplus o \end{array}$$

Finally, we have to show that the relation also encodes the theory of the sort *type*. Just as was the case with number-encoding chains, the nature of chains as being finite sequences automatically captures the fact that semantic types are finite and acyclic. In the case of the sort *type*, we have furthermore imposed the restriction that configurations under an object of sort *type* must be such that they contain the maximally possible number of identities. This was expressed with the TYPE IDENTITY PRINCIPLE in (114). This principle was needed to ensure that non-congruent configurations always correspond to distinct semantic types. In the chain encoding, we are more permissive, i.e., chains that encoded the same semantic type need not be "congruent", it is enough if they contain the same sequence of *s1* and *s2* objects. For this reason, the conditions included in the definition of the relation `chain-type` are enough.

We define the relation `chain-term` which has the following property: A chain c is in this relation iff c corresponds to some *me* object, or, equivalently, iff c corresponds to some term of Ty2. The definition of the relation `chain-term` in (242) contains one clause for every species below *me*.

(242) The relation `chain-term`:
for each constant $c \in Const$, such that c is assigned the type τ by $\mathcal{C}$, and c corresponds to the species $const_i$,

$$\texttt{chain-term}(x) \stackrel{\forall}{\Longleftarrow}$$
$$x = \langle (i+21), 14\rangle \oplus t$$
$$\textbf{and } \tau^{\#}[t/a]$$

$$\texttt{chain-term}(x) \stackrel{\forall}{\Longleftarrow}$$
$$x = \langle 7, 14\rangle \oplus t \oplus \langle 11\rangle \oplus n$$
$$\textbf{and } \texttt{chain-type}(t)$$
$$\textbf{and } \texttt{chain-number}(n)$$

$$\texttt{chain-term}(x) \stackrel{\forall}{\Longleftarrow}$$
$$x = \langle 8, 14\rangle \oplus t \oplus \langle 15\rangle \oplus f \oplus \langle 16\rangle \oplus a$$
$$\textbf{and } \texttt{chain-type}(t)$$
$$\textbf{and } \texttt{chain-term}(f)$$
$$\textbf{and } \texttt{chain-term}(a)$$
$$\textbf{and } \texttt{find-type}(f, t_1)$$
$$\textbf{and } \texttt{find-type}(a, t_2)$$
$$\textbf{and } t_1 = \langle 6, 12\rangle \oplus t_2 \langle 13\rangle \oplus t$$

$$\texttt{chain-term}(x) \stackrel{\forall}{\Longleftarrow}$$
$$x = \langle 9, 14\rangle \oplus t \oplus \langle 17\rangle \oplus v \oplus \langle 16\rangle \oplus a$$
$$\textbf{and } \texttt{chain-type}(t)$$
$$\textbf{and } \texttt{chain-term}(v) \textbf{ and } v = \langle 7\rangle \oplus v'$$
$$\textbf{and } \texttt{chain-term}(a)$$
$$\textbf{and } \texttt{find-type}(v, t_1)$$
$$\textbf{and } \texttt{find-type}(a, t_2)$$
$$\textbf{and } t = \langle 6, 12\rangle \oplus t_1 \langle 13\rangle \oplus t_2$$

$$\texttt{chain-term}(x) \stackrel{\forall}{\Longleftarrow}$$
$$x = \langle 10, 14, 3\rangle \oplus \langle 18\rangle \oplus f \oplus \langle 19\rangle \oplus a$$
$$\textbf{and } \texttt{chain-term}(a_1)$$
$$\textbf{and } \texttt{chain-term}(a_2)$$
$$\textbf{and } \texttt{find-type}(a_1, t_1)$$
$$\textbf{and } \texttt{find-type}(a_2, t_2)$$
$$\textbf{and } \texttt{are-chain-copies}(t_1, t_2)$$

The clause for a constant c makes reference to the particular type τ assigned to the constant by the function $\mathcal{C}$ and to the species $const_i$ that corresponds to this constant. A chain that meets the clause for the constant c, then, must start with a subsequence of the form $s2s1^{i+21}$, as the number $(i + 21)$ represents the sort $const_i$ according to Table 4.1. The next $s2s1^i$ sequence of this chain is required to contain exactly fourteen objects of sort *s1*, i.e., it represents the attribute TYPE. In order to impose the right semantic type, the remaining subsequence of the chain, then, must be described by $\tau^{\#}$.

The clause for variables is constructed following the final clause of the relation `chain-type`. If it describes some chain c, then c has the following properties: c starts with a subsequence of the form $s2s1^7$ (which corresponds to the sort *var* in Table 4.1), followed by a subsequence $s2s1^{14}$, i.e., the representation of the attribute TYPE. The remaining sequence starts with some type-encoding chain t, followed by a subsequence $s2s1^{11}$. We have already seen that this subsequence is the chain-encoding of the attribute NUMBER. Finally, the rest of the chain c must be some sequence n which is in the relation `chain-number`.

The last three clauses in (242) encode the information on the other species below *me*, *appl*, *abstr* and *equ*. In the first line, we always give the overall structure of the chain. In the clause for application, the chain is of the form

$$\langle 8, 14\rangle \oplus t \oplus \langle 15\rangle \oplus f \oplus \langle 16\rangle \oplus a.$$

According to Table 4.1, the initial $s2s1^8$ subsequence represents the species *appl*. The following $s2s1^{14}$ sequence corresponds to the attribute TYPE. The sequence t is required to be a type encoding chain by virtue of the line `chain-type`(t). Next on the chain, there is a subsequence $s2s1^{15}$ (= FUNC), followed by a term-encoding sequence f. Then, there comes a sequence $s2s1^{16}$ (= ARG) and, finally, another term-encoding sequence a.

The first four lines of the clause, thus, give us the information from the appropriateness conditions of the sort *appl* as given in (121). In Section 3.1.3 we have, however, also imposed some restrictions on the semantic types that occur within a *appl* object: The TYPE value of a *appl* object must be identical with its FUNC TYPE OUT value, and the ARG TYPE value must be identical with its FUNC TYPE IN value (see (122a)).

In order to encoded these restrictions in the clause of the relation `chain-term`, we use an auxiliary relation `find-type` which has a pair $\langle c,t\rangle$ in its denotation iff the first $s2s1^i$ sequence of c corresponds to a species below *me*, the second $s2s1^i$ sequence of c corresponds to the attribute TYPE (i.e., $s2s1^{14}$), and the rest of c starts with the sequence t, such that t is chain encoding of a semantic type.[5]

(243) The relation `find-type`:

$$\texttt{find-type}(x,y) \overset{\forall}{\Longleftarrow} \begin{pmatrix} x = \langle s2\rangle \oplus s \oplus \langle 14\rangle \oplus y \oplus r \\ \textbf{and} \begin{pmatrix} \texttt{is-1-long}(s) \\ \textbf{or} \ldots \\ \textbf{or}\ \texttt{is-10-long}(s) \\ \textbf{or}\ \texttt{is-22-long}(s) \\ \textbf{or} \ldots \\ \textbf{or}\ \texttt{is-}(n+21)\texttt{-long}(s) \end{pmatrix} \\ \textbf{and}\ \texttt{chain-type}(y) \end{pmatrix}$$

With the relation `find-term`, we can access the subsequence of a term-encoding that encodes its semantic type. In the clause that corresponds to the sort *appl*, we, now, include the requirement that those subsequences of x, f and a that encode the semantic types, i.e., t, t_1 and t_2 stand in the right relation, i.e., that t_1 corresponds to a complex type whose input type is t_2 and whose output type is t. Having integrated this type restriction into the clause that corresponds to the sort *appl*, this clause encodes all the restrictions that we impose on objects of sort *appl*.

The next clause in (242) expresses the restrictions on the sort *abstr*. As such, it is almost like the previous clause, but uses different numbers, since the species and attribute names differ. Furthermore, it requires that the chain that corresponds to the variable bound by the lambda abstractor, the sequence v, be the encoding of a variable, i.e., v must be in the relation `chain-term` and start with a subsequence $s2s1^7$. The constraint on the semantic type of a lambda abstraction as given in (122b) for the sort *abstr* is given in the last three lines of the clause.

The final clause in (242) corresponds to the sort *equ*. It is very similar to the preceding two clauses. The main difference lies in the encoding of the constraint on the semantic types. As we know that an equation is always of type t, we can directly require the initial part of a chain that meets the last clause to be of the form $\langle s2s1^{10}, s2s1^{14}, s2s1^{3}\rangle$, which corresponds to the species *equ*, the attribute TYPE and the species *truth* according to Table 4.1. In addition, we require that the subsequences a_1 and a_2 which correspond to the two subterms of an equation, be of the same type. For term-encoding chains, this means that the subsequences that correspond to their semantic type be copies of each other.

[5]As the relation `find-type` does not have a recursive clause, we could, in principle, replace the relation calls `find-type`(x, y) in (242) by the description in the body of the clause in (243). As this would turn the clauses in (242) quite lengthy, we prefer to use the auxiliary relation for the convenience of the reader.

The relation `chain-term`, as given in (242) provides the means to identify exactly those chains that encode a term of Ty2. This will be necessary in our encoding of λ-conversion. There, we must manipulate subsequences of term-encoding chains. We must be sure, however, that what we manipulate are actually term-encoding subsequences and not just arbitrary sequences of *s1* and *s2* objects.

We will use the relation `chain2term` in Section 4.2.2.2 to encode λ-conversion as a relation, not as a new kind of linguistic object, as we did in Section 4.2.1. The function "#" will be used in Section 4.2.2.3 to allow for a LE-encoding of shifting operations that interacts with the chain encoding of λ-conversion.

4.2.2.2. *λ-Conversion defined on Chains.* In our formalization of λ-conversion in Section 4.2.1, we introduced a sort, *reduction* with two attributes AUX and TERM such that the term that corresponds to the AUX value reduces to the term that corresponds to the TERM value. Instead of enriching the ontology with such a sort, we will, now, define λ-conversion as a relation between two term-encoding chains. A pair $\langle c_1, c_2 \rangle$ is in the relation `reduction` iff there are terms ϕ_1 and ϕ_2 such that c_1 is described by ${\phi_1}^{\#}$ and c_2 is described by ${\phi_2}^{\#}$, and ϕ_1 β-reduces to ϕ_2. But before we can turn to λ-conversion, we must give the analogs to the definitions of *subterm*, *free variables*, and *substitution*.

In the constraint on *β-reduction* objects, we could use the relation `component` to find a redex within the input term. In the case of chains, however, a subterm of a term-representing chain is not a component of that chain, but a string within it. In order to be able to identify such term-representing sub-strings of chains, we define the relation `chain-subterm` in (244).

To keep the clauses in the definitions of the following relations readable, we will use the symbol "$\oplus$" in infix notation for the relation `append`. Furthermore, we will use the natural number i as abbreviations for the description of a subsequence of the form $s2s1^i$.

(244) The relation `chain-subterm`:

$$\begin{array}{l}
\texttt{chain-subterm}(x,y) \stackrel{\forall}{\Longleftarrow} \\
\quad \texttt{chain-term}(y) \\
\quad \textbf{and } \texttt{are-chain-copies}(x,y) \\
\texttt{chain-subterm}(x,y) \stackrel{\forall}{\Longleftarrow} \\
\quad \texttt{chain-term}(y) \\
\quad \textbf{and } y = \langle 7{,}14 \rangle \oplus t \oplus \langle 17 \rangle \oplus v \oplus \langle 16 \rangle \oplus a \\
\quad \textbf{and } (\texttt{chain-subterm}(x,v) \textbf{ or } \texttt{chain-subterm}(x,a)) \\
\texttt{chain-subterm}(x,y) \stackrel{\forall}{\Longleftarrow} \\
\quad \texttt{chain-term}(y) \\
\quad \textbf{and } y = \langle 8{,}14 \rangle \oplus t \oplus \langle 15 \rangle \oplus f \oplus \langle 16 \rangle \oplus a \\
\quad \textbf{and } (\texttt{chain-subterm}(x,f) \textbf{ or } \texttt{chain-subterm}(x,a)) \\
\texttt{chain-subterm}(x,y) \stackrel{\forall}{\Longleftarrow} \\
\quad \texttt{chain-term}(y) \\
\quad \textbf{and } y = \langle 9{,}14 \rangle \oplus t \oplus \langle 18 \rangle \oplus a_1 \oplus \langle 19 \rangle \oplus a_2 \\
\quad \textbf{and } (\texttt{chain-subterm}(x,a_1) \textbf{ or } \texttt{chain-subterm}(x,a_2))
\end{array}$$

The four clauses in the definition of the relation `chain-subterm` correspond to the four cases of Definition 4.1. Notice that we must require that a chains c can only be a subterm of a chain c' if both are term-encoding chains. It is, however, sufficient to require this property explicitly of the second argument of the relation, because if it is a term representing chain, then the first argument must be term representing as well.

We must be able to characterize a free variable, even if represented as a chain. This is achieved in the relation `chain-free-variable`. A pair $\langle c, c' \rangle$ is in the relation `chain-free-variable` iff c is the chain-encoding of some variable v and c' is the chain encoding of some term ϕ and v occurs freely in ϕ. We will follow the definition in (201) as closely as possible.

(245) The relation `chain-free-variable`:

$\texttt{chain-free-variable}(x,y) \overset{\forall}{\Longleftarrow}$
$\quad \texttt{chain-term}(x)$
$\quad \textbf{and } x = \langle 7 \rangle \oplus a$
$\quad \textbf{and } \texttt{are-chain-copies}(x,y)$

(clause for application:)

$\texttt{chain-free-variable}(x,y) \overset{\forall}{\Longleftarrow}$
$\quad \texttt{chain-term}(y)$
$\quad \textbf{and } y = \langle 8{,}14 \rangle \oplus t \oplus \langle 15 \rangle \oplus f \oplus \langle 16 \rangle \oplus a$
$\quad \textbf{and } \left(\begin{array}{l} \texttt{chain-free-variable}(x,f) \\ \textbf{or } \texttt{chain-free-variable}(x,a) \end{array} \right)$

(clause for abstraction:)

$\texttt{chain-free-variable}(x,y) \overset{\forall}{\Longleftarrow}$
$\quad y = \langle 9{,}14 \rangle \oplus t \oplus \langle 17 \rangle \oplus v \oplus \langle 16 \rangle \oplus a$
$\quad \textbf{and not } \texttt{are-copies}(x,v)$
$\quad \textbf{and } \texttt{chain-free-variable}(x,a)$

(clause for equation:)

$\texttt{chain-free-variable}(x,y) \overset{\forall}{\Longleftarrow}$
$\quad \texttt{chain-term}(y)$
$\quad \textbf{and } y = \langle 10{,}14 \rangle \oplus t \oplus \langle 18 \rangle \oplus a_1 \oplus \langle 19 \rangle \oplus a_2$
$\quad \textbf{and } \left(\begin{array}{l} \texttt{chain-free-variable}(x,a_1) \\ \textbf{or } \texttt{chain-free-variable}(x,a_2) \end{array} \right)$

The relation `chain-free-variable` is needed for the formalization of the notion *substitution* as given in Definition 4.3. In (246) we define a relation `chain-replace` parallel to the definition of the relation `replace` in (203). Like in (203) we will indicate to which line of Definition 4.3 the clauses correspond.

(246) The relation `chain-replace`:

$[\psi/x]x = \psi$

$\texttt{chain-replace}(x,y,v,w) \overset{\forall}{\Longleftarrow}$
$\quad \texttt{chain-term}(x)$
$\quad \textbf{and } \texttt{chain-term}(v)$
$\quad \textbf{and } x = \langle 7 \rangle \oplus a$
$\quad \textbf{and } \texttt{are-chain-copies}(x,y)$
$\quad \textbf{and } \texttt{are-chain-copies}(v,w)$

$[\psi/x]a = a$, for each variable and constant $a \neq x$

$\texttt{chain-replace}(x,y,v,w) \overset{\forall}{\Longleftarrow}$
$\quad \texttt{chain-term}(x)$
$\quad \textbf{and } \left(\begin{array}{l} x = \langle 7 \rangle \oplus a \\ \textbf{or } x = \langle 22 \rangle \oplus a \\ \textbf{or } \ldots \\ \textbf{or } x = \langle (n+21) \rangle \oplus a \end{array} \right)$
$\quad \textbf{and not } \texttt{are-chain-copies}(x,y)$
$\quad \textbf{and } \texttt{are-chain-copies}(x,v)$

$[\psi/x](\phi_1\phi_2) = ([\psi/x]\phi_1[\psi/x]\phi_2)$

$\texttt{chain-replace}(x, y, v, w) \stackrel{\forall}{\Longleftarrow}$
$\quad\texttt{chain-term}(x)$
$\quad\textbf{and}\ \texttt{chain-term}(v)$
$\quad\textbf{and}\ x = \langle 8, 14\rangle \oplus t \oplus \langle 15\rangle \oplus f \oplus \langle 16\rangle \oplus a$
$\quad\textbf{and}\ x = \langle 8, 14\rangle \oplus t' \oplus \langle 15\rangle \oplus f' \oplus \langle 16\rangle \oplus a'$
$\quad\textbf{and}\ \texttt{are-chain-copies}(t, t')$
$\quad\textbf{and}\ \texttt{chain-replace}(f, y, f', w)$
$\quad\textbf{and}\ \texttt{chain-replace}(a, y, a', w)$

$[\psi/x](\phi_1 = \phi_2) = ([\psi/x]\phi_1 = [\psi/x]\phi_2)$

$\texttt{chain-replace}(x, y, v, w) \stackrel{\forall}{\Longleftarrow}$
$\quad\texttt{chain-term}(x)$
$\quad\textbf{and}\ \texttt{chain-term}(v)$
$\quad\textbf{and}\ x = \langle 10, 14\rangle \oplus t \oplus \langle 18\rangle \oplus a_1 \oplus \langle 19\rangle \oplus a_2$
$\quad\textbf{and}\ x = \langle 10, 14\rangle \oplus t' \oplus \langle 18\rangle \oplus a_1' \oplus \langle 19\rangle \oplus a_2'$
$\quad\textbf{and}\ \texttt{are-chain-copies}(t, t')$
$\quad\textbf{and}\ \texttt{chain-replace}(a_1, y, a_1', w)$
$\quad\textbf{and}\ \texttt{chain-replace}(a_2, y, a_2', w)$

$[\psi/x](\lambda x.\phi) = \lambda x.\phi$

$\texttt{chain-replace}(x, y, v, w) \stackrel{\forall}{\Longleftarrow}$
$\quad\texttt{chain-term}(x)$
$\quad\textbf{and}\ x = \langle 9, 14\rangle \oplus t \oplus \langle 17\rangle \oplus u \oplus \langle 16\rangle \oplus a$
$\quad\textbf{and}\ \texttt{are-chain-copies}(u, y)$
$\quad\textbf{and}\ \texttt{are-chain-copies}(x, v)$

$[\psi/x](\lambda y.\phi) = \lambda y.[\psi/x]\phi$, if $y \neq x$ and $y \notin FV(\psi)$ or $x \notin FV(\phi)$

$\texttt{chain-replace}(x, y, v, w) \stackrel{\forall}{\Longleftarrow}$
$\quad\texttt{chain-term}(x)$
$\quad\textbf{and}\ \texttt{chain-term}(v)$
$\quad\textbf{and}\ x = \langle 9, 14\rangle \oplus t \oplus \langle 17\rangle \oplus u \oplus \langle 16\rangle \oplus a$
$\quad\textbf{and}\ v = \langle 9, 14\rangle \oplus t' \oplus \langle 17\rangle \oplus u' \oplus \langle 16\rangle \oplus a'$
$\quad\textbf{and}\ \begin{pmatrix}\textbf{not}\ \texttt{chain-free-variable}(u, w) \\ \textbf{or not}\ \texttt{chain-free-variable}(y, a)\end{pmatrix}$
$\quad\textbf{and}\ \texttt{are-chain-copies}(t, t')$
$\quad\textbf{and}\ \texttt{are-chain-copies}(u, u')$
$\quad\textbf{and}\ \texttt{chain-replace}(a, y, a', w)$

In Section 4.2.1.1 we saw that we need a relation `replace1` that would encode the substitution of a single occurrence of some subterm in addition to a relation `replace` that encodes the substitution of all free occurrences of some variable. Similarly, for the chain encoding, such a second relation is needed as well. In (247) we define this relation `chain-replace1`.

(247) The relation `chain-replace1`

$\texttt{chain-replace1}(x, y, v, w) \stackrel{\forall}{\Longleftarrow}$
$\quad\texttt{chain-term}(x)$
$\quad\textbf{and}\ \texttt{chain-term}(v)$
$\quad\textbf{and}\ x = \langle 7\rangle \oplus a$
$\quad\textbf{and}\ \texttt{are-chain-copies}(x, y)$
$\quad\textbf{and}\ \texttt{are-chain-copies}(v, w)$

$$
\begin{array}{l}
\texttt{chain-replace1}(x, y, v, w) \stackrel{\forall}{\Longleftarrow} \\
\quad \texttt{chain-term}(x) \\
\quad \textbf{and } \texttt{chain-term}(v) \\
\quad \textbf{and } x = \langle 8, 14\rangle \oplus t \oplus \langle 15\rangle \oplus f \oplus \langle 16\rangle \oplus a \\
\quad \textbf{and } x = \langle 8, 14\rangle \oplus t' \oplus \langle 15\rangle \oplus f' \oplus \langle 16\rangle \oplus a' \\
\quad \textbf{and } \texttt{are-chain-copies}(t, t') \\
\quad \textbf{and } \left(\begin{array}{l} \left(\begin{array}{l} \texttt{chain-replace1}(f, y, f', w) \\ \textbf{and } \texttt{are-chain-copies}(a, a') \end{array} \right) \\ \textbf{or} \\ \left(\begin{array}{l} \texttt{chain-replace1}(a, y, a', w) \\ \textbf{and } \texttt{are-chain-copies}(f, f') \end{array} \right) \end{array} \right)
\end{array}
$$

$$
\begin{array}{l}
\texttt{chain-replace1}(x, y, v, w) \stackrel{\forall}{\Longleftarrow} \\
\quad \texttt{chain-term}(x) \\
\quad \textbf{and } \texttt{chain-term}(v) \\
\quad \textbf{and } x = \langle 10, 14\rangle \oplus t \oplus \langle 18\rangle \oplus a_1 \oplus \langle 19\rangle \oplus a_2 \\
\quad \textbf{and } x = \langle 10, 14\rangle \oplus t' \oplus \langle 18\rangle \oplus a'_1 \oplus \langle 19\rangle \oplus a'_2 \\
\quad \textbf{and } \texttt{are-chain-copies}(t, t') \\
\quad \textbf{and } \left(\begin{array}{l} \left(\begin{array}{l} \texttt{chain-replace1}(a_1, y, a'_1, w) \\ \textbf{and } \texttt{are-chain-copies}(a_2, a'_2) \end{array} \right) \\ \textbf{or} \\ \left(\begin{array}{l} \texttt{chain-replace1}(a_2, y, a'_2, w) \\ \textbf{and } \texttt{are-chain-copies}(a_1, a'_1) \end{array} \right) \end{array} \right)
\end{array}
$$

$$
\begin{array}{l}
\texttt{chain-replace1}(x, y, v, w) \stackrel{\forall}{\Longleftarrow} \\
\quad \texttt{chain-term}(x) \\
\quad \textbf{and } \texttt{chain-term}(v) \\
\quad \textbf{and } x = \langle 9, 14\rangle \oplus t \oplus \langle 17\rangle \oplus u \oplus \langle 16\rangle \oplus a \\
\quad \textbf{and } v = \langle 9, 14\rangle \oplus t' \oplus \langle 17\rangle \oplus u' \oplus \langle 16\rangle \oplus a' \\
\quad \textbf{and } \left(\begin{array}{l} \textbf{not } \texttt{chain-free-variable}(u, w) \\ \textbf{or not } \texttt{chain-free-variable}(y, a) \end{array} \right) \\
\quad \textbf{and } \texttt{are-chain-copies}(t, t') \\
\quad \textbf{and } \texttt{are-chain-copies}(u, u') \\
\quad \textbf{and } \texttt{chain-replace1}(a, y, a', w)
\end{array}
$$

The definition of the relation `chain-replace1` follows that of the relation `replace1` in (209), but uses the chain notions instead of descriptions of *me* objects.

After the definitions of the auxiliary relations, we can turn to the relation `reduction`. A pair $\langle x, y\rangle$ of term-encoding chains is in the relation `reduction` iff either x and y are copies, or there is a chain z such that the pair $\langle x, z\rangle$ stands in any of the relations `change-bound-var` or β-`contraction`, and the pair $\langle z, y\rangle$ is in the relation `reduce`. Clearly, the three clauses of the relation `reduction` mimic the sort-hierarchy below *reduction* in (204).

(248) The relation `reduction`:

$$
\begin{array}{l}
\texttt{reduction}(x, y) \stackrel{\forall}{\Longleftarrow} \\
\quad \texttt{chain-term}(x) \\
\quad \textbf{and } \texttt{are-chain-copies}(x, y)
\end{array}
$$

$$
\begin{array}{l}
\texttt{reduction}(x, y) \stackrel{\forall}{\Longleftarrow} \\
\quad \texttt{chain-term}(x) \\
\quad \textbf{and } \texttt{change-bound-var}(x, z) \\
\quad \textbf{and } \texttt{reduction}(z, y)
\end{array}
$$

$\texttt{reduction}(x, y) \overset{\forall}{\Longleftarrow}$
  $\texttt{chain-term}(x)$
  and β-$\texttt{contraction}(x, z)$
  and $\texttt{reduction}(z, y)$

Of the three clauses in (248), the first, the case of an empty sequence of applications of reduction operations uses the relation `are-chain-copies` as defined in (238). As chains that are in this relation encode the same term, the first clause in (248) specifies the case that for each term ϕ, it holds trivially that $\phi \rightarrow_\lambda \phi$. The non-trivial cases of β-reduction are encoded by the additional relations `change-bound-var` and β-`contraction`, used in the second and the third clause of the definition of the relation `reduction`.

In (249) we give the definition of the relation `change-bound-var`. A pair $\langle c_1, c_2 \rangle$ is in this relation iff c_1 encodes some term ϕ_1 and c_2 encodes some term ϕ_2 and ϕ_2 results from changing a bound variable in ϕ_1 according to Definition 4.4 above.

(249) The relation `change-bound-var`:

$\texttt{change-bound-var}(x, y) \overset{\forall}{\Longleftarrow}$
  $\texttt{chain-subterm}(u, x)$
  and $u = \langle 9,14 \rangle \oplus t \oplus \langle 17 \rangle \oplus v \oplus \langle 16 \rangle \oplus a$
  and $w = \langle 9,14 \rangle \oplus t' \oplus \langle 17 \rangle \oplus v' \oplus \langle 16 \rangle \oplus a'$
  and **not** $\texttt{chain-free-variable}(v', a)$
  and $\texttt{find-type}(v, t_v)$
  and $\texttt{find-type}(v', t'_v)$
  and $\texttt{are-chain-copies}(t_v, t'_v)$
  and $\texttt{chain-replace}(a, v, a', v')$
  and $\texttt{chain-replace}(x, u, y, w)$

This definition is parallel to the constraint on the sort *change-bound-var* in (207). For illustration consider the following schematic example. Assume that the term ϕ contains an occurrence of a term $\lambda x_1.\psi$, and let x_2 be a variable that is not free in ψ. Let, then, χ be the term that results from substituting $\lambda x_2.[x_2/x_1]\psi$ for $\lambda x_1.\psi$ in ϕ. In (250) we indicate which variables in the definition of the relation `change-bound-var` correspond to which of these terms:

(250)

x	$=$	ϕ	v	$=$	x_1
y	$=$	χ	v'	$=$	x_2
u	$=$	$\lambda x_1.\psi$	a	$=$	ψ
w	$=$	$\lambda x_2.[x_2/x_1]\psi$	a'	$=$	$[x_2/x_1]\psi$

With these correspondences in mind, it is relatively easy to read the definition of the relation `change-bound-var`. The chain x has a subsequence that corresponds to the chain u. The chain u chain-encodes some term $\lambda x_1.\psi$, where v is the chain that encodes the Ty2-variable x_1, and a is the chain that encodes the term ψ. This is indicated in the characterization of u as given in the second line of the clause. In the third line, we get the characterization of the chain w. This chain encodes a term $\lambda x_2.\psi'$, where ψ' is just $[x_2/x_1]\psi$. The variable x_2 is encoded as the chain v', which is a subsequence of w, the term ψ' is encoded as the chain a'. We use the relation `chain-free-variable` to state that the chain v' must not be free in the chain a, which corresponds to the requirement that the variable x_2 be not free in ψ.

In addition, we require that the chains v and v' encode terms of the same semantic type. With the relation `find-type` we isolate the subsequences of v and v' that encode the semantic type, i.e., the subsequences t_v and t'_v. Consequently, we use the relation `are-chain-copies` to state that these two subsequences encode the same semantic type, Finally, we state the substitution. The call `chain-replace`(a, v, a', v') specifies the same as saying that ψ' is the term $[x_2/x_1]\psi$. In the last line, we use the relation `chain-replace1` to substitute the sequence w for one occurrence of the sequence u in x.

The third clause in the definition of the relation `reduction` in (248) uses the relation β-`contraction`. In (251), we give the clause that is needed to define this relation. Again, the parallelism to the constraint in (210) is obvious.

(251) The relation β-`contraction`:

β-`reduction`$(x, y) \overset{\forall}{\Longleftarrow}$
 `chain-term`(x)
 and `chain-subterm`(r, x)
 and $r = \langle 8{,}14\rangle \oplus t \oplus \langle 15\rangle \oplus f \oplus \langle 16\rangle \oplus s$
 and $f = \langle 9, 14\rangle \oplus t' \oplus \langle 17\rangle \oplus v \oplus \langle 16\rangle \oplus a$
 and `chain-replace`(a, v, c, s)
 and `chain-replace1`(x, r, y, c)

In this clause, we require the chain x to be a term-encoding chain. The chain x must contain a sequence which is a subterm of the term represented by x. We use the variable r to refer to this sequence. The sequence r is characterized as being the chain-encoding of a β-redex, i.e., its initial *s2s1*^{i} sequence contains exactly eight *s1* objects which makes it correspond to a term of the form $(\phi_1\phi_2)$. The chain r is split up between its semantic type t, its functor f and its argument s. The chain f is characterized as corresponding to a λ-abstraction (as it starts with *s2s1*^{9}). This characterization makes r the encoding of a β-redex. In the last two lines of the clause, we state that the usual substitutions are found between x and y.

We have given the necessary details for a chain encoding for formula shifting operations such as λ-conversion. We will, next, consider how this technique enables us to integrate λ-conversion into a linguistic theory.

4.2.2.3. *Integration.* In the following, we will integrate the definitions of Section 4.2.2.2 into our grammar. Notice that while we introduced a number of relations in Section 4.2.2.2, we did not introduce any new sorts or attributes. This means that we can still assume the CONTENT value of a sign to be an object of sort *me*. We will first show how the SEMANTICS PRINCIPLE, or more precisely the relation `intensional-functional-application` (`ifa`), can be changed from the original formulation in (150) to a new formulation which includes λ-conversion. Furthermore, we will impose a constraint on the CONTENT value of signs requiring that the logical form of a sign be redex-free. Finally, we will show what consequences the new approach to λ-conversion has for the encoding of shifting operations.

Given the encoding of λ-conversion via chains as presented in Section 4.2.2.2, we can define the SEMANTICS PRINCIPLE in such a way that we require the CONTENT value of a phrase to be a λ-converted form of the intensional functional application of the CONTENT values of the daughters. In (252) we repeat the SEMANTICS PRINCIPLE from (149).

(252) The SEMANTICS PRINCIPLE (SP):

$$phrase \Rightarrow \begin{bmatrix} \text{SYNS LOC CONT } \boxed{1} \\ \text{DTRS} \begin{bmatrix} \text{H-DTR SYNS LOC CONT } \boxed{2} \\ \text{N-DTR SYNS LOC CONT } \boxed{3} \end{bmatrix} \end{bmatrix}$$

and `ifa`($\boxed{1}$,$\boxed{2}$,$\boxed{3}$)

What needs to be done, next, is to re-define the relation `ifa` in such a way that it expresses that the intensional functional application of its second and its third argument β-reduces to its first argument. This is done in the following definition.

(253) Revised definition of the relation `ifa`:

$\texttt{ifa}(\boxed{1},\boxed{2},\boxed{3}) \stackrel{\forall}{\Longleftarrow}$
$\quad \texttt{chain2term}(p, \boxed{1})$
$\quad \textbf{and } \texttt{chain2term}(h, \boxed{2})$
$\quad \textbf{and } \texttt{chain2term}(n, \boxed{3})$
$\quad \textbf{and } \texttt{chain-term}(c)$
$\quad \textbf{and } \texttt{chain-term}(a)$
$\quad \textbf{and } c = \langle 8, 14\rangle \oplus t \oplus \langle 15\rangle \oplus h \oplus \langle 16\rangle \oplus a$
$\quad \textbf{and } a = \langle 9, 14\rangle \oplus t' \oplus \langle 17, 7, 14, 5, 11, 1\rangle \oplus \langle 16\rangle \oplus n$
$\quad \textbf{and } \texttt{reduction}(c, p)$

$\texttt{ifa}(\boxed{1},\boxed{2},\boxed{3}) \stackrel{\forall}{\Longleftarrow}$
$\quad \texttt{chain2term}(p, \boxed{1})$
$\quad \textbf{and } \texttt{chain2term}(h, \boxed{2})$
$\quad \textbf{and } \texttt{chain2term}(n, \boxed{3})$
$\quad \textbf{and } \texttt{chain-term}(c)$
$\quad \textbf{and } \texttt{chain-term}(a)$
$\quad \textbf{and } c = \langle 8, 14\rangle \oplus t \oplus \langle 15\rangle \oplus n \oplus \langle 16\rangle \oplus a$
$\quad \textbf{and } a = \langle 9, 14\rangle \oplus t' \oplus \langle 17, 7, 14, 5, 11, 1\rangle \oplus \langle 16\rangle \oplus h$
$\quad \textbf{and } \texttt{reduction}(c, p)$

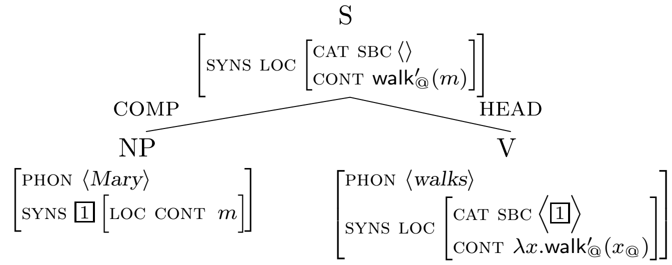
In the clauses of the relation `ifa`, the variables p, h and n refer to the chains that encode the *me* objects which occur as the CONTENT values of a phrase, its head daughter and its nonhead daughter respectively. We need two more chains, c and a both of which are term-encoding chains, as stated in the calls `chain-term`(c) and `chain-term`(a). In the first clause c is the chain that encodes the intensional functional application of the logical form of the head daughter to the logical form of the nonhead daughter. The characterization of the chain c says that c encodes an application (as it starts with *s2s1*^{8}), the functor of which is the chain that encodes the logical form of the head daughter (h), the argument is some term-encoding chain a. The chain a is characterized as a chain that encodes a λ-abstraction (*s2s1*^{9}). The abstractor binds the variable encoded as the sequence $\langle 7, 14, 5, 11, 1\rangle$. Remember from (235) that this sequence represents the variable $v_{s,0}$ (i.e., @). The scope of the abstraction is the chain n that corresponds to the logical form of the nonhead daughter.

The second clause differs from the first clause of the relation `ifa` only in that it uses the chain n as the functor in the chain c, instead of h.

In the final line in the first clause, we specify that the pair $\langle c, m\rangle$ be in the relation `reduction`. This means that the chain c encodes a term which β-reduces to the term which is encoded by the chain p. According to the first line of the clause, p is the chain encoding of the logical form of the phrase. Thus, we know that the logical form of a phrase is a β-reduced form of the intensional functional application of the logical forms of the daughters.

So far we have not enforced that the logical form of a phrase is redex-free. In Section 4.2.1.2 we expressed this condition with the CONTENT PRINCIPLE in (216). In (254) we re-state this constraint within the present architecture.

FIGURE 4.11. An example derivation of the sentence *Mary walks.*:



(254) The CONTENT PRINCIPLE (for the second alternative):

$$local \Rightarrow \left(\begin{bmatrix}\textsc{content}\ \boxed{1}\end{bmatrix} \textbf{ and not } \mathbf{E}\boxed{2} \left(\begin{array}{l} \texttt{subterm}(\boxed{2},\boxed{1}) \\ \textbf{and } \boxed{2}\begin{bmatrix} appl \\ \textsc{func}\ \ abstr \end{bmatrix} \end{array} \right) \right)$$

Given the SEMANTICS PRINCIPLE which uses the relation ifa as defined in (253), and the CONTENT PRINCIPLE, the CONTENT value of a phrase is a fully β-reduced *me* object. In contrast to the encoding of λ-conversion presented in Section 4.2.1, the reduction steps do not appear as components of the linguistic sign.

In the introduction to Section 4.2.2 we saw structures as they result from an architecture which uses chain encodings. In Figure 4.9 (page 186) and in Figure 4.10 (page 187), we gave the structure for two alternative derivations of the sentence *Mary walks.* For clarity, let us illustrate the first of these derivations again. In Figure 4.11 we repeat Figure 4.9.

The logical form of the daughters is just their basic translation. As these do not contain a β-redex, they satisfy the CONTENT PRINCIPLE in (254). The SEMANTICS PRINCIPLE enforces that the logical form of the mother stands in the relation ifa with the logical forms of the daughters. To check whether this is the case, we must see whether the triple of terms $\langle \mathsf{walk}'_@(m), \lambda x.\mathsf{walk}'_@(x_@), m\rangle$ satisfies one of the clauses of the relation ifa as given in (253). Let us check whether the triple is described by the first clause, i.e., the clause that takes the head daughter as the semantic functor. In (255) we give the terms that correspond to the CONTENT values of the signs in Figure 4.11 and the chain-encoding of these terms. We assume that the constant m corresponds to the species $const_{24}$, and that the constant walk' corresponds to the species $const_{127}$. In order to make the chains more readable, we set apart those subsequences that correspond to frequently used subterms, and indicate the corresponding term below the subsequences.

(255) a. Chain encoding of the logical form of the phrase $(\mathsf{walk}'(v_{s,0})(m))$:

$$p = \left\langle 8, 14, 3, 15, \frac{8,14,6,12,4,13,3,15,148,14,6,12,5,13,6,12,4,13,3,16,7,14,5,11,1,}{\mathsf{walk}'_@} 16, \frac{45,14,4}{m} \right\rangle$$

b. Chain encoding of the logical form of the head daughter $([\lambda v_{(se),0}.\mathsf{walk}'(v_{s,0})](v_{se,0}(v_{s,0}))$:

$h=$

$$\Big\langle 9, 14, 6, 12, 6, 12, 5, 13, 4, 13, 3, 17, \frac{7,14,6,12,5,13,4,11,1,}{v_{se,0}} 16,$$
$$8, 14, 3, 15, \frac{8,14,6,12,4,13,3,15,148,14,6,12,5,13,6,12,4,13,3,16,7,14,5,11,1,}{\mathsf{walk}'_@}$$
$$16, 8, 14, 4, 15, \frac{7,14,6,12,5,13,4,11,1,}{v_{se,0}} 16, \frac{7,14,5,11,1}{v_{s,0}} \Big\rangle$$

c. Chain encoding of the logical form of the nonhead daughter (m_e):

$n = \left\langle \frac{45,14,4}{m} \right\rangle$

Given these three chains, the first clause in (253) requires us to assume two more chains, c and a where a contains the chain n as a subsequence and c contains h and a as subsequences. In (256) we give these two chains, according to the characterization given in (253). Below each chain, we indicate the term that this chain corresponds to.

(256) a. The term-encoding chain a:

$$a = \left\langle 9, 14, 6, 12, 5, 13, 4, 17, \frac{7,14,5,11,1,}{v_{s,0}} 16 \right\rangle \oplus n,$$

i.e., $\lambda v_{s,0}.m_e$

b. The term-encoding chain c:

$$c = \langle 8, 14, 4, 15 \rangle \oplus h \oplus \langle 16 \rangle \oplus a,$$

i.e., $(\lambda v_{se,0}.\mathsf{walk}'_{@}(v_{se,0} v_{s,0}))(\lambda v_{s,0}.m_e)$

As can be seen from the term below the chain c, the chain is the encoding of term that expresses the intensional functional application of the logical form of the head daughter to that of the nonhead daughter. We can, then, take this chain as the starting point for carrying out β-reduction. The term encoded by c is a redex. Its contractum is just like the sequence in the second line of the chain h in (255), but with the sequence a substituted for the subsequence sequence that corresponds to the variable $v_{se,0}$. The resulting chain is given in (257).

(257)
$$c' = \Big\langle 8, 14, 3, 15, \frac{8,14,6,12,4,13,3,15,148,14,6,12,5,13,6,12,4,13,3,16,7,14,5,11,1,}{\mathsf{walk}'_{@}} 16,$$
$$8, 14, 4, 15, \frac{9,14,6,12,5,13,4,17, \frac{7,14,5,11,1,}{v_{s,0}} 16, \frac{45,14,3}{m}}{\lambda @.m} 16, \frac{7,14,5,11,1}{v_{s,0}} \Big\rangle$$

The chain c' as given in (257) contains a β-redex, which is given by the second line in (257). This subsequence corresponds to the term $(\lambda v_{s,0}.m)(v_{s,0})$. As the variable $v_{s,0}$ does not have a free occurrence in the term m, the contractum of the redex is simply the term m. Similarly, the chain c' reduces to the chain c'' by replacing the chain that encodes the redex by the chain that encodes the constant m. The resulting chain is given in (258).

(258) $c'' =$
$$\Big\langle 8, 14, 3, 15, \frac{8,14,6,12,4,13,3,15,148,14,6,12,5,13,6,12,4,13,3,16,7,14,5,11,1,}{\mathsf{walk}'_{@}} 16, \frac{45,14,3}{m} \Big\rangle$$

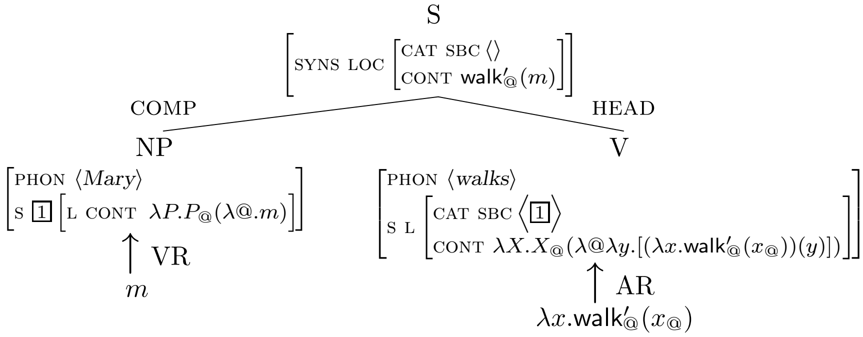
The chain in (258), however, is nothing but the chain encoding of the logical form of the phrase, given as p in (255a). Thus, we have shown that the triple consisting of the CONTENT value of the phrase, the head daughter and the nonhead daughter of Figure 4.11 is in the relation `ifa`.

With this example, we have illustrated how the logical form of a phrase is determined given the logical form of its daughters. We must, next, consider whether the approach to λ-conversion presented so far is compatible with our treatment of shifting operations. When we presented shifting operations in Section 4.1, the result of shifting the basic translation of some word was shown to be a complex term which contains the basic translation of the word as a subterm. While some of the resulting terms are redex free, such as the application of VR to the basic translation of the proper name *Mary* ($m \longrightarrow_{VR} \lambda P.P(@)(\lambda @.m)$), this is not the case in general. In (221), we saw that the application of AR to the basic translation of the verb *walks* results in a term that contains a β-redex. In (259) the basic translation of the verb is repeated, together with the term that results from applying AR to it.

(259) *walks* $\rightsquigarrow$ $\lambda x_{se}.\mathsf{walk}'_{@et}(x(@))$

$\longrightarrow_{AR}$ $\lambda X_{s((se)t)}.X(@)(\lambda @ \lambda y.[(\lambda x.\mathsf{walk}'_{@}(x(@)))(y)])$

FIGURE 4.12. A second derivation of the sentence *Mary walks.* (no λ-conversion at the word level):



As the term in the second line of (259) is not redex-free, the CONTENT PRINCIPLE in (216) excludes it from occurring as the logical form of a word. As a consequence, the CONTENT PRINCIPLE as it stands, excludes the application of most shifting operations. There are two ways out of this situation. First, we could require redex-freeness only for the logical form of phrases, but allow redices in the logical form of words. Second, we could include λ-conversion into the shifting operations, the same way we did this for intensional functional application in (253).

Let us consider the first alternative. In this case, the requirement for redex-freeness could be incorporated into the SEMANTICS PRINCIPLE, and there is no need for an additional CONTENT PRINCIPLE. A revised form of the SEMANTICS PRINCIPLE is given in (260).

(260) The revised SEMANTICS PRINCIPLE (SP):

$$phrase \Rightarrow \begin{bmatrix}\textsc{synS loc cont } \boxed{1} \\ \textsc{dtrs} \begin{bmatrix}\textsc{h-dtr synS loc cont } \boxed{2} \\ \textsc{n-dtr synS loc cont } \boxed{3}\end{bmatrix}\end{bmatrix}$$
$$\textbf{and } \texttt{intensional-functional-application}(\boxed{1},\boxed{2},\boxed{3})$$
$$\textbf{and not E}\boxed{4} \begin{pmatrix}\texttt{subterm}(\boxed{4},\boxed{1}) \\ \textbf{and } \boxed{4}\begin{bmatrix}appl \\ \textsc{func } abstr\end{bmatrix}\end{pmatrix}$$

The last line in (260) is just like the last line in the formulation of the CONTENT PRINCIPLE in (254). It expresses that the logical form of a phrase does not contain a subterm $\boxed{2}$ such that $\boxed{2}$ is a β-redex.

Assuming this new formulation of the SP and removing the CONTENT PRINCIPLE from the grammar, does not lead to any change with respect to the structure of the sentence *Mary walks* as given in Figure 4.11.

We saw in Figure 4.8 and Figure 4.10 that there is an alternative derivation of the logical form of the sentence. In this derivation, we apply VR to the basic translation of the proper name *Mary*, and AR to the basic translation of the verb *walks*. The resulting structure is given in Figure 4.12 where we use the "$\uparrow$AR/VR " notation to indicate the application of shifting rules without committing ourselves to either the DR- or the LE-encoding. In the tree in Figure 4.12, the CONTENT value of the head daughter contains a β-redex, the term $(\lambda x.\mathsf{walk}'_@(x_@))(y)$. As we do no longer require the CONTENT value of every sign to be redex-free, but only that of phrases, this derivation is possible.

It seems unnatural to exclude λ-conversion from the logical forms of words, but to enforce it for phrases. Therefore, we will show in the remainder of this subsection, how such an encoding can be achieved.

In Section 4.1, we have introduced the relations `vr` in (153) and `ar` in (160) that hold between of a pair of *me* objects $\langle o, u\rangle$ iff o corresponds to some term ϕ and u corresponds to some term ψ such that u results from o by applying a shifting operation as given in Definition 1.10 or in Definition 1.11. Similar to these relations, we can give two relations `chain-ar` and `chain-vr` that hold of a pair of term-encoding chains in the same circumstances. In (261) we give the functional specification of these relations. Their formalization is given in Appendix 1.3.2, because it is just a transformation of the clauses for the relations `ar` and `vr` into chain concepts.

(261) a. The relation `chain-vr` (see (556) in Appendix 1.3.2):
For each term-encoding chains c_1, c_2,
$\langle c_1, c_2\rangle \in \mathsf{R}(\texttt{chain-vr})$ iff
there are terms ϕ_1 and ϕ_2 such that
c_1 is described by ${\phi_1}^{\#}$,
c_2 is described by ${\phi_2}^{\#}$, and
ϕ_1 and ϕ_2 are related via VR as given in Definition 1.11.

b. The relation `chain-ar` (see (557) in Appendix 1.3.2):
For each term-encoding chains c_1, c_2,
$\langle c_1, c_2\rangle \in \mathsf{R}(\texttt{chain-ar})$ iff
there are terms ϕ_1 and ϕ_2 such that
c_1 is described by ${\phi_1}^{\#}$,
c_2 is described by ${\phi_2}^{\#}$, and
ϕ_1 and ϕ_2 are related via AR as given in Definition 1.10.

Given the relations `chain-ar` and `chain-vr` as specified in (261) and as defined in the Appendix, we can incorporate λ-conversion into the encoding of shifting operations. We will first consider the DR-approach to shifting operations. In (262) we give the revised descriptions of the shifting DRs.

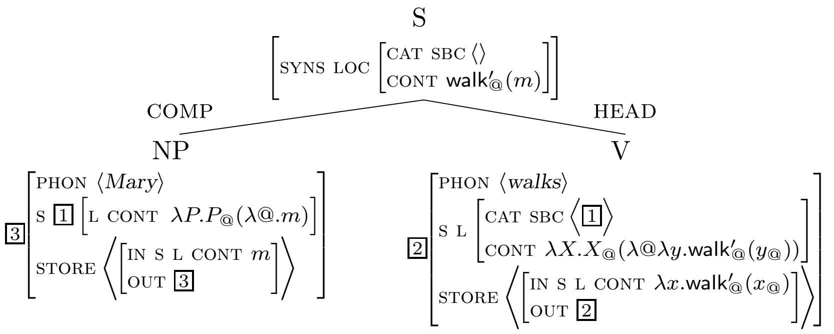
(262) a. Revised description of the AR-DR:

$$\begin{bmatrix} \textit{der-rule} \\ \text{IN} & \begin{bmatrix} \textit{word} \\ \text{PHON} & \boxed{3} \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \boxed{4} \\ \text{CONT} & \boxed{1} \end{bmatrix} \\ \text{NONL} & \boxed{5} \end{bmatrix} \end{bmatrix} \\ \text{OUT} & \begin{bmatrix} \textit{word} \\ \text{PHON} & \boxed{3} \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \boxed{4} \\ \text{CONT} & \boxed{2} \end{bmatrix} \\ \text{NONL} & \boxed{5} \end{bmatrix} \end{bmatrix} \end{bmatrix} \quad \begin{array}{l} \textbf{and } \texttt{chain2term}(a, \boxed{1}) \\ \textbf{and } \texttt{chain-ar}(a, b) \\ \textbf{and } \texttt{reduction}(b, c) \\ \textbf{and } \texttt{chain2term}(c, \boxed{2}) \end{array}$$

b. Revised description of the VR-DR:

$$\begin{bmatrix} \textit{der-rule} \\ \text{IN} & \begin{bmatrix} \textit{word} \\ \text{PHON} & \boxed{3} \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \boxed{4} \\ \text{CONT} & \boxed{1} \end{bmatrix} \\ \text{NONL} & \boxed{5} \end{bmatrix} \end{bmatrix} \\ \text{OUT} & \begin{bmatrix} \textit{word} \\ \text{PHON} & \boxed{3} \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \boxed{4} \\ \text{CONT} & \boxed{2} \end{bmatrix} \\ \text{NONL} & \boxed{5} \end{bmatrix} \end{bmatrix} \end{bmatrix} \quad \begin{array}{l} \textbf{and } \texttt{chain2term}(a, \boxed{1}) \\ \textbf{and } \texttt{chain-vr}(a, b) \\ \textbf{and } \texttt{reduction}(b, c) \\ \textbf{and } \texttt{chain2term}(c, \boxed{2}) \end{array}$$

FIGURE 4.13. A second derivation of the sentence *Mary walks.* (assuming a DR encoding of shifting and fully reduced logical forms for words):



The description in (262a) states that the logical form of the input word is encoded as a chain a. There is a second chain, b, such that the pair $\langle a, b\rangle$ is in the relation `chain-ar`. The chain b, furthermore, β-reduces to a term-encoding chain c. Chain c is the encoding of the logical form of the output word of the DR. The description of the DR for value raising is analogous to this, but uses the relation `chain-vr` instead of the relation `chain-ar`.

With the shifting DRs as specified in (262), we can, again, include in the grammar the requirement that the logical form of a sign must be redex-free. Assuming the DRs as just given, the SEMANTICS PRINCIPLE in (252), and the CONTENT PRINCIPLE in (216), we can re-consider the second derivation of the sentence *Mary walks.* In Figure 4.13, we give the semantic analysis for this sentence under the assumption that both daughters have a shifted logical form. The tree in Figure 4.13 differs from that in Figure 4.12 in two respects. First, we indicate explicitly that we use a DR-encoding, and second, we have a redex-free terms as the logical form of the head daughter.

The tree in Figure 4.13 is such that the logical form of each sign is fully β-reduced. Still, there is no auxiliary structure inside the CONTENT value such as the REDUCTION value of Section 4.2.1.2. But, since we use a DR-encoding of shifting, the shifting steps are explicitly part of the overall linguistic sign.

Finally, we can show that it is possible to include λ-conversion into the LE-encoding of shifting operations as well. As the LE-encoding locates the applicability of shifting operations in the lexical entry, we must include the application of β-reduction in the lexical specification of words. Let us, again, consider the lexical entry of the word *walks*.

(263) Parts of the lexical entry for the word *loves*, including shifting operations:

$$\mathbf{E}\boxed{1}\ \mathbf{E}a\ \mathbf{E}b\ \mathbf{E}c \left[\begin{array}{l} \textit{word} \\ \text{PHON}\ \langle \textit{walks} \rangle \\ \text{SYNS} \left[\begin{array}{l} \text{LOC} \left[\begin{array}{l} \text{CAT} \left[\begin{array}{ll} \text{HEAD} & \textit{verb} \\ \text{SUBCAT} & \langle \text{NP} \rangle \\ \text{MARKING} & \textit{unmarked} \end{array}\right] \\ \text{CONT}\ \boxed{1} \end{array}\right] \\ \text{NONL} \left[\begin{array}{ll} \text{INHER SLASH} & \textit{eset} \\ \text{TO-BIND SLASH} & \textit{eset} \end{array}\right] \end{array}\right] \end{array}\right] \begin{array}{l} \textbf{and}\ (\lambda x.\mathsf{walk}'_{@}(x(@)))^{\#} \\ \textbf{and}\ \texttt{chain-shifting}(a, b) \\ \textbf{and}\ \texttt{chain2term}(b, \boxed{1}) \end{array}$$

The AVM in the lexical entry in (263) specifies the phonology as being $\langle$*walks*$\rangle$. Furthermore, a word that is described by this lexical entry must be a verb with a single NP on its subcat list. What is new in (263) is the description of the logical form of this word. In the lexical entry, we state that there is a chain a which is the encoding of the basic translation of the verb *walks*. Note that the function "#" maps a term to a $\mathcal{TY}2$ description that contains exactly one free variable, a, i.e., to description of the chain a. This chain, furthermore, stands in the relation `chain-shifting` with some chain b. The relation `chain-shifting` is the closure over the relations `chain-ar`, `chain-vr` and `reduction`. Finally, this shifted and β-reduced chain b is considered a chain that encodes the logical form of the word.

In (264) we define the relation `chain-shifting`. A pair of term-encoding chains $\langle c, c' \rangle$ is in this relation iff the term that is encoded by c can be shifted to some term which is β-reduced to the term encoded by c'.

(264)
$$\begin{array}{l}
\texttt{chain-shifting}(x,y) \overset{\forall}{\Longleftarrow} \\
\quad\quad \texttt{reduction}(x,y) \\
\texttt{chain-shifting}(x,y) \overset{\forall}{\Longleftarrow} \\
\quad\quad \texttt{chain-ar}(x,z) \\
\quad\quad \textbf{and } \texttt{chain-shifting}(z,y) \\
\texttt{chain-shifting}(x,y) \overset{\forall}{\Longleftarrow} \\
\quad\quad \texttt{chain-vr}(x,z) \\
\quad\quad \textbf{and } \texttt{chain-shifting}(z,y)
\end{array}$$

Given this new lexical entry of the verb *walks*, we can show that the following AVM describes a word that is licensed by the lexical entry in (263).

(265)
$$\begin{bmatrix}
\textit{word} \\
\text{PHON} \ \langle \textit{walks} \rangle \\
\text{SYNS} \begin{bmatrix}
\text{LOC} \begin{bmatrix}
\text{CAT} \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{SUBCAT} & \langle \text{NP} \rangle \\ \text{MARKING} & \textit{unmarked} \end{bmatrix} \\
\text{CONT} \ \lambda X.X_{@}(\lambda @ \lambda y.\mathsf{walk}'_{@}(y_{@}))
\end{bmatrix} \\
\text{NONL} \begin{bmatrix} \text{INHER SLASH} & \textit{eset} \\ \text{TO-BIND SLASH} & \textit{eset} \end{bmatrix}
\end{bmatrix}
\end{bmatrix}$$

A word described by (265) has the phonological and syntactic properties as specified in the lexical entry in (263). We must, then, show that it also satisfies the conditions on its CONTENT value. This is the case if there is some chain a that is the chain encoding of the basic translation of the verb *walks* as given in Table 1.1, i.e., a chain that is described by $(\lambda x.\mathsf{walk}'_{@}(x_{@}))^{\#}$. The existence of such a chain is guaranteed if we can ensure that the word described in (265) contains at least one component of the sort *s1* and one component of the sort *s2*. If we chose these sorts to be *me* and *type*, this is the case. Next, we must find a chain, b, such that the pair $\langle a, b \rangle$ is in the interpretation of the relation `chain-shifting`.

We can take a chain that is described by $(\lambda X.X_{@}(\lambda @ \lambda y.\mathsf{walk}'_{@}(y_{@})))^{\#}$. This chain is in the relation `chain-shifting` with the chain a, because there is a chain b' which is described by $(\lambda X.X_{@}(\lambda @ \lambda y.\mathsf{walk}'_{@}(y_{@})))^{\#}$ and which, therefore is in the relation `chain-ar` with a. As the pair $\langle b', b \rangle$ is in the relation `reduction`, it is also in the relation `chain-shifting` by virtue of the first clause in (264). Thus, the pair $\langle a, b \rangle$ is in this relation as well.

In addition, the chain b is the term encoding of the CONTENT value of a word described by the AVM in (265). Since this term is redex-free, the CONTENT PRINCIPLE is equally met by every word that is described by (265).

With the LE-encoding of shifting operations, the structure of the second derivation of the sentence *Mary walks* need not contain the application of DR. Instead, we can have a structure as given above in Figure 4.10 (page 187). This structure accounts for the intuition that both shifting and β-reduction are automatic processes whose application need not be explicitly visible in the linguistic objects.

The chain encoding of terms and *me* object enables us to work with the logical forms of signs in a natural way, i.e., we can freely apply shifting operations and λ-conversion without worrying about the question whether some input term is a subterm of the output term of some operation. Such concerns are unavoidable in the case of the encoding given in Section 4.2.1, they are, however, unnatural under a semanticists perspective. The chain encoding of λ-conversion (and shifting operations) allows us to save the intuition that the logical form should be a redex-free term, while the particular reduction steps are of no linguistic significance. It is mainly because of this last point that we favor the second encoding of λ-conversion, in combination with the LE-encoding of shifting operations.

With the chain-encoding of terms of Ty2 and of *me* objects, we have presented a so-far unexplored use of chains in RSRL. Therefore, it useful to reconsider what properties of RSRL made such an encoding possible in the first place and what further potential might be hidden behind the concept of chains for the architecture of grammar. As such principled considerations are not of direct relevance to the present enterprise of providing an integration of LF-Ty2 into HPSG, we have added them in a short section, Section 4.5, at the end of this chapter.

4.3. Interaction with DRs

In the preceding sections, we presented a fully formalized integration of LF-Ty2 into HPSG. So far, we restricted our attention to the syntactic fragment presented in Section 2.3.1. This fragment does not include passive and complement extraction, i.e., phenomena which we analyzed in Section 2.3.2 by using *Derivational Rules*.

In the present section, we will address this remaining issue and show that it is straightforward to include a logical form specification in the formalization of the Passive DR and the Complement Extraction DR. This will also give us the opportunity to reconsider at what kind of signs we must allow the application of shifting rules. We will show that it is necessary to allow the application of shifting rules to the output of the passive DR. On the other hand, it less clear whether it is necessary to apply them to the output of the Complement Extraction DR, though such an application is probably unproblematic.

In Section 2.3.2 we presented a DR to relate an active verb to its passivized form. The passive verb was characterized as differing from the active verb with respect to (i) its VFORM specification, and (ii) the length of the SUBCAT list. For illustration, let us reconsider the transitive verb *love*. In (266) we give a description of the output verb of the DR as presented in Section 2.3.2.

$$
(266)\quad \begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{loved} \rangle \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{pas} \end{bmatrix} \\ \text{SUBCAT} & \left\langle \text{NP} \right\rangle \\ \text{MARKING} & \textit{unmarked} \end{bmatrix} \end{bmatrix} \\ \text{NONL} & \begin{bmatrix} \text{INHER SLASH} & \textit{eset} \\ \text{TO-BIND SLASH} & \textit{eset} \end{bmatrix} \end{bmatrix} \\ \text{STORE} & \textit{nelist} \end{bmatrix}
$$

A word described by the AVM in (266) has a single element on its SUBCAT list. As such, it is incompatible with a semantic specification such as the basic translation of the verb *love*, which we repeat in (267).

(267) *love* $\rightsquigarrow \lambda y \lambda x.\mathsf{love}'_{@}(x(@), y(@))$

The basic translation of the verb indicates that it requires two semantic arguments. As the syntax of the passivized form of this verb provides only one syntactic complement, we must also reduce the semantic valence of the passivized form. In (268) we show the translation that we want to assume for the passivized form.

(268) $\lambda y.\exists x[\mathsf{love}'_{@}(x(@), y(@))]$

This translation of the passivized verb has just one semantic argument, indicated by the abstractor λy. The second semantic argument is bound by the existential quantifier $\exists x$. Thus, the semantic valence of the term in (268) corresponds to the syntactic valence of the output word of an application of the passive DR to an active word licensed by the lexical entry of the word *love*. In (269) we enrich the Passive DR by a specification of the logical form of its output.

(269) The Passive DR (informal, including the CONTENT specification):

$$\begin{bmatrix} \textit{word} \\ \text{SYNS LOC} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD} \begin{bmatrix} \textit{verb} \\ \text{VFORM} \ \textit{psp} \end{bmatrix} \\ \text{SBC} \ \left\langle \text{NP}|\boxed{2} \right\rangle \oplus \left\langle \boxed{1} \right\rangle \end{bmatrix} \\ \text{CONT} \ \lambda y \lambda z_n \ldots \lambda z_1 \lambda x \lambda u_m \ldots \lambda u_1.\phi \end{bmatrix} \end{bmatrix} \longmapsto$$

$$\begin{bmatrix} \textit{word} \\ \text{SYNS LOC} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD} \begin{bmatrix} \textit{verb} \\ \text{VFORM} \ \textit{pas} \end{bmatrix} \\ \text{SBC} \ \left\langle \boxed{1}|\boxed{2} \right\rangle \end{bmatrix} \\ \text{CONT} \ \lambda z_n \ldots \lambda z_1 \lambda y \lambda u_m \ldots \lambda u_1.\exists x \phi \end{bmatrix} \end{bmatrix}$$

(where the list $\boxed{2}$ has exactly n elements)

The CONTENT value of the input has the following schematic form:

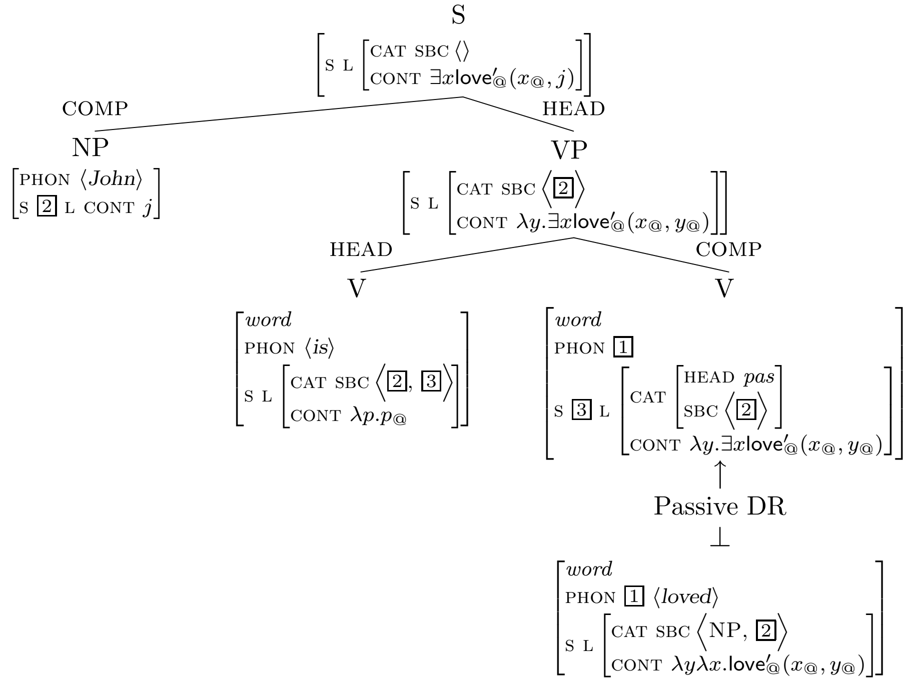
$$\lambda y \lambda z_n \ldots \lambda z_1 \lambda x \lambda u_m \ldots \lambda u_1.\phi$$

There, y is the variable that corresponds to the last element of the SUBCAT list of the input word, and x is the variable that corresponds to the first element of this SUBCAT list. The variables z_i correspond to further complements, possibly none. The informally stated condition that there be exactly as many further syntactic complements as there are variables z_i expresses this. Finally, we allow that the semantic valence of the input is greater than its syntactic valence. As we have seen, such cases arise by the application of value raising, a shifting operation that adds a further semantic valence, but has no effect on the syntactic valence.

In the DR, the CONTENT of the output is specified in such a way that it has one semantic argument less than the input ($\exists x$ in the output instead of λx). Furthermore, the order of the remaining semantic arguments is changed in the same way the order of the complements is altered on the SUBCAT lists. This means that the first semantic argument of the input, λy, re-appears after the other semantic arguments that correspond to syntactic complements ($\lambda z_n \ldots \lambda z_1$). Finally, the further semantic arguments ($\lambda u_m \ldots \lambda u_1$) are added without changing the order.

It can be seen that the term in (268) is the logical form of the output word of the passive DR, if we assume the basic translation of the input word as given in (267). Note that the sequence $\lambda z_n \ldots \lambda z_1$ is empty in the case of a transitive verb such as *love*.

FIGURE 4.14. The structure of sentence (270):



In Figure 4.14 we give the syntactic structure together with the logical forms of the simple passive sentence (75b), repeated as (270).

(270) John is loved.

Figure 4.14 is just like Figure 2.4 (page 97), i.e., in both cases, the passive form of the verb *loved* is derived via some Passive DR. This verb, then, combines with the passive auxiliary *is*, which behaves like a raising verb. Finally the subject NP *John* is added.

At the semantic side, the basic translation of the verb *love* appears in the input to the Passive DR. At the output, the term in (268) features. The semantics of the passive auxiliary is that of the identity function, just as assumed for the complementizer *that*. At the S node, the basic translation of the proper name *John*, the constant j, is added. This results in the following logical form for sentence (270):

(271) $\exists x[\mathsf{love}'_@(x_@, j)]$

We will next give an example that shows that we must allow the application of shifting rules to the output of the Passive DR. This is the case if we want the subject of a passive clause to take scope over material in some superordinate clause. A relevant example is given in (272). In (272) we indicate the three readings below the example sentence. Notice that these readings are parallel to the three readings attested for sentence (39c), i.e. for a complex sentence with a finite complement clauses headed by an intransitive verb. For a better comparison, we repeat example (39c) together with its readings in (273).

(272) Every man believes that some woman is loved.

a. *de dicto* reading:
$\forall x[\mathsf{man}'_@(x_@) \rightarrow \mathsf{believe}'_@(x_@, (\lambda @.\exists y[\mathsf{woman}'_@(y) \wedge \exists z[\mathsf{love}'_@(z_@, y_@)]]))]$

b. $\forall\exists$-*de re* reading:
$\forall x[\mathsf{man}'_@(x_@) \rightarrow \exists y[\mathsf{woman}'_@(y_@) \wedge \mathsf{believe}'_@(x_@, \lambda @.(\exists z[\mathsf{love}'_@(z_@, y_@)]))]]$

c. $\exists\forall$-*de re* reading:
$\exists y[\mathsf{woman}'_@(y_@) \wedge \forall x[\mathsf{man}'_@(x_@) \rightarrow \mathsf{believe}'_@(x_@, \lambda @.(\exists z[\mathsf{love}'_@(z_@, y_@)]))]]$

(273) Every man believes that some woman walks.
a. *de dicto* reading:
$\forall x_{se}[\mathsf{man}'_@(x_@) \rightarrow \mathsf{believe}'_@(x_@, \lambda @.\exists y_{se}[\mathsf{woman}'_@(y_@) \wedge \mathsf{walk}'_@(y_@)])]$
b. $\forall\exists$-*de re* reading:
$\forall x_{se}[\mathsf{man}'_@(x_@) \rightarrow \exists y_{se}[\mathsf{woman}'_@(y_@) \wedge \mathsf{believe}'_@(x_@, \lambda @.\mathsf{walk}'_@(y_@))]]$
c. $\exists\forall$-*de re* reading:
$\exists y_{se}[\mathsf{woman}'_@(y_@) \wedge \forall x_{se}[\mathsf{man}'_@(x_@) \rightarrow \mathsf{believe}'_@(x_@, \lambda @.\mathsf{walk}'_@(y_@))]]$

We can derive the first reading in (272), i.e., the *de dicto* reading without applying any shifting rule, just as we could derive the logical form of the *de dicto* reading of sentence (273) without applying any shifting operations (see Figure 1.11 (page 48)). In the case of the two *de re* readings in (273b) and (273c), however, we showed that it is necessary to apply first VR and then AR to the embedded verb. In our discussion of the *de re* readings of sentence (39c), we gave the following derivation for the logical form of the embedded intransitive verb *walks*.

(274) *walks* $\leadsto$ $\lambda y.\mathsf{walk}'_@(y_@)$
$\longrightarrow_{VR}$ $\lambda y \lambda u.u_@(\lambda @.\mathsf{walk}'_@(y_@))$
$\longrightarrow_{AR1}$ $\lambda Y \lambda u.Y_@(\lambda @ \lambda y.u_@(\lambda @.\mathsf{walk}'_@(y_@)))$

Applying the Passive DR to a transitive verb such as *love* results in a word whose logical form is very similar to that of the intransitive verb *walks*. Therefore, under the assumption that we can apply shifting operations to the output of DRs, we can derive the logical form of the embedded verb as required for the *de re* readings parallel to the derivation given in (274). This derivation is indicated in (275). In the derivation, we also include the effect of the Passive DR on the logical form.

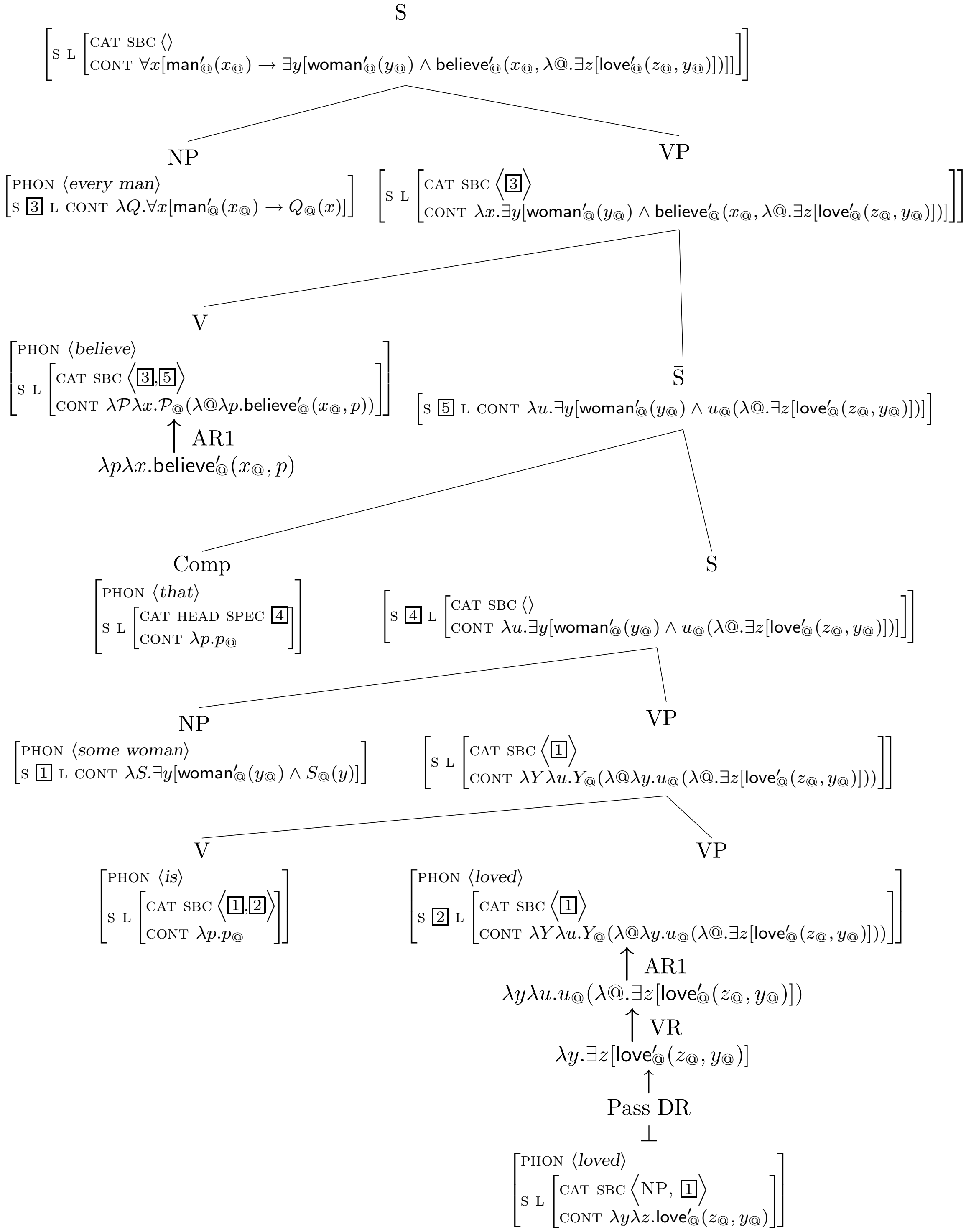
(275) *loved* $\leadsto$ $\lambda y \lambda z.\mathsf{love}'_@(z_@, y_@)$
$\longmapsto_{\text{Pass DR}}$ $\lambda y.\exists z[\mathsf{love}'_@(z_@, y_@)]$
$\longrightarrow_{VR}$ $\lambda y \lambda u.u_@(\lambda @.\exists z[\mathsf{love}'_@(z_@, y_@)])$
$\longrightarrow_{AR1}$ $\lambda Y \lambda u.Y_@(\lambda @ \lambda y.u_@(\lambda @.\exists z[\mathsf{love}'_@(z_@, y_@)]))$

In the derivation in (275) we start from the basic translation of the transitive verb *loved*. Using the active word as an input to the Passive DR in (269), results in an output word whose logical form is as given in the second line in (275). If we apply first VR to this term, followed by AR1, then the we get the term that is required for the *de re* readings of sentence (272). In Figure 4.15 (page 218) we show the derivation of the $\forall\exists$-reading.

In the embedded clause, the logical form of the passivized verb *loved* is derived as indicated in (275). The passivized verb combines with the passive auxiliary. As the passive auxiliary is assumed to be just the identity function, the logical form of the resulting VP is the same as that of the verb *loved*. This VP combines with the embedded subject. As we applied AR after VR in the derivation of the logical form of the verb *loved*, the quantifier contributed by the embedded subject has scope over an occurrence of the variable u in the logical form of the embedded S node. To this S, we add the complementizer *that*. In order to keep the structure simple, we use the lexical entry of the complementizer as given in (178b), i.e., we assume that logical form of the complementizer is the identity function.

The logical form of the matrix verb *believes* is the result of applying AR to its first semantic argument. When this derived logical form combines with the logical form of the embedded clause, the quantifier contributed by the embedded subject has the matrix predicate $\mathsf{believe}'$ in its scope, i.e., we have an instance of a *de re* reading. Finally, the

FIGURE 4.15. The structure of the ∀∃-*de re* reading of sentence (272):



matrix subject is added. The quantifier that is part of its logical form takes widest scope in the logical form of the sentence.

As shown in Figure 4.15, the assumption that we can apply shifting rules to the output of a DR allows us to derive the *de re* readings of sentence (272). We can, now, show that the opposite assumption will not lead to this result, i.e., that disallowing the application of shifting rules to the output of the Passive DR will block the derivation of the *de re* readings. In (276) we give the result of applying the shifting operations VR and AR1 in the same order as in (275), but before the application of the Passive DR.

(276) *loved*
$$\begin{array}{ll} \rightsquigarrow & \lambda y \lambda z.\mathsf{love}'_@(z_@, y_@) \\ \longrightarrow_{VR} & \lambda y \lambda z \lambda u.u_@(\lambda @.\mathsf{love}'_@(z_@, y_@)) \\ \longrightarrow_{AR1} & \lambda Y \lambda z \lambda u.Y_@(\lambda @ \lambda y.u_@(\lambda @.\mathsf{love}'_@(z_@, y_@))) \\ \longmapsto_{\text{Pass DR}} & \lambda Y \lambda u.\exists z[Y_@(\lambda @ \lambda y.u_@(\lambda @.\mathsf{love}'_@(z_@, y_@)))] \end{array}$$

This derivation leads to a reading where the existential quantifier introduced in the DR has wide scope over the variable u introduced by VR. Thus, if we use the resulting logical form of the verb, the existential quantifier $\exists z$ has wide scope over the matrix predicate $\mathsf{believe}'$, and, in fact, also over *some woman*, giving rise to the following reading.

(277) $\forall x[\mathsf{man}'_@(x_@) \rightarrow \exists z \exists y[\mathsf{woman}'_@(y_@) \wedge \mathsf{believe}'_@(x_@, \lambda @.\mathsf{love}'_@(z_@, y_@))]]$

Even though such a reading might exist, the problematic aspect is that forbidding the application of shifting to the output of the Passive DR makes the prediction that whenever the subject of the passive clause has scope over the matrix predicate, so does the existentially bound underlying active subject.

We conclude that the application of shifting operations to the output of the Passive DR must be possible. In (278) we give a formally more precise version of the Passive DR. We give the variant that uses the LE-encoding of shifting operations and the chain encoding of λ-conversion presented in Section 4.2.2.

(278) Formal specification of the Passive DR:

$$\mathbf{E}\boxed{1}\ \mathbf{E}\boxed{2}\ \mathbf{E}\boxed{3}\ \mathbf{E}\boxed{4}\ \mathbf{E}\boxed{5}\ \mathbf{E}\boxed{6}\ \mathbf{E}\boxed{7}\ \mathbf{E}\boxed{8}\ \mathbf{E}\boxed{9}\ \mathbf{E}a\ \mathbf{E}b\ \mathbf{E}c$$

$$\begin{bmatrix} \textit{derivation-rule} & \\ \text{IN} & \begin{bmatrix} \textit{word} & \\ \text{PHON} & \boxed{3} \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{psp} \\ \text{INV} & \boxed{5} \\ \text{AUX} & \boxed{6} \end{bmatrix} \\ \text{SUBCAT} & \left\langle \text{NP}—\boxed{2} \right\rangle \oplus \left\langle \boxed{1} \right\rangle \\ \text{MARKING} & \boxed{7} \end{bmatrix} \\ \text{CONT} & \boxed{8}\ \lambda y \lambda z_n \ldots \lambda z_1 \lambda x \lambda u_m \ldots \lambda u_1 . \phi \end{bmatrix} \\ \text{NONL} & \boxed{4} \end{bmatrix} \end{bmatrix} \\ \text{OUT} & \begin{bmatrix} \textit{word} & \\ \text{PHON} & \boxed{3} \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{pas} \\ \text{INV} & \boxed{5} \\ \text{AUX} & \boxed{6} \end{bmatrix} \\ \text{SUBCAT} & \left\langle \boxed{1}—\boxed{2} \right\rangle \\ \text{MARKING} & \boxed{7} \end{bmatrix} \\ \text{CONT} & \boxed{9} \end{bmatrix} \\ \text{NONL} & \boxed{4} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\textbf{and } (\lambda z_n \ldots \lambda z_1 \lambda y \lambda u_m \ldots \lambda u_1 . \exists x \phi)^{\#}$$
$$\textbf{and } \texttt{chain-shifting}(a, b) \textbf{ and } \texttt{chain2term}(b, \boxed{9})$$

The specification of the DR as given in (278) includes all the syntactic specifications that were given in Section 2.3.2 already. In addition, it also contains the specification of the logical form of the input word and the output word. The characterization of the input is just as given in the informal version of this DR in (269) above. The output term that appear in the informal version of the DR is, however, not taken directly to be the CONTENT value of the output word. Instead, we say that there is a chain a that corresponds to this output term by the function "#". This chain can be shifted and β-reduced to another chain b. The chain b is the chain encoding of the CONTENT value of the output word. This shows that we have included the applicability of shifting operations and β-reduction in the formulation of the DR.

It should be noticed that the formulation of the Passive DR in (278) is just *more* formal than that in (269), it is, however, not a *full* formalization. This is indicated by the use of "…" in the logical form of the input and the characterization of the chain a. In a fully formal version, we would need some new relation `passive-lf` that relates the logical form of the input word to the chain a. The definition of the relation is given in (559) in Appendix 1.3.3.

In the present section we remain neutral with respect to the particular encoding of shifting operations and λ-conversion. Yet, in the version of the Passive DR in (278), we encoded the possibility of applying shifting and λ-conversion to the output of the DR directly as part of the DR. This indicates that we have given a version of the DR that assumes an LE-encoding of shifting and a chain-encoding of λ-conversion. In (279) we give the formalization of the Passive DR under the three alternative combinations of encodings.

(279) Alternative Formalizations of the Passive DR:

a. DR-encoding of shifting, chain-encoding of λ-conversion:

$$\mathbf{E}\boxed{1}\ \mathbf{E}\boxed{2}\ \mathbf{E}\boxed{3}\ \mathbf{E}\boxed{4}\ \mathbf{E}\boxed{5}\ \mathbf{E}\boxed{6}\ \mathbf{E}\boxed{7}\ \mathbf{E}\boxed{8}\ \mathbf{E}\boxed{9}\ \mathbf{E}a\ \mathbf{E}b$$

$$\begin{bmatrix} \textit{derivation-rule} & \\ \text{IN} & \begin{bmatrix} \textit{word} & \\ \text{PHON} & \boxed{3} \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{psp} \\ \text{INV} & \boxed{5} \\ \text{AUX} & \boxed{6} \end{bmatrix} \\ \text{SUBCAT} & \left\langle \text{NP}\text{—}\boxed{2} \right\rangle \oplus \left\langle \boxed{1} \right\rangle \\ \text{MARKING} & \boxed{7} \end{bmatrix} \\ \text{CONT} & \boxed{8}\ \lambda y \lambda z_n \ldots \lambda z_1 \lambda x \lambda u_m \ldots \lambda u_1 . \phi \end{bmatrix} \\ \text{NONL} & \boxed{4} \end{bmatrix} \end{bmatrix} \\ \text{OUT} & \begin{bmatrix} \textit{word} & \\ \text{PHON} & \boxed{3} \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{pas} \\ \text{INV} & \boxed{5} \\ \text{AUX} & \boxed{6} \end{bmatrix} \\ \text{SUBCAT} & \left\langle \boxed{1}\text{—}\boxed{2} \right\rangle \\ \text{MARKING} & \boxed{7} \end{bmatrix} \\ \text{CONT} & \boxed{9} \end{bmatrix} \\ \text{NONL} & \boxed{4} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

and $(\lambda z_n \ldots \lambda z_1 \lambda y \lambda u_m \ldots \lambda u_1 . \exists x \phi)^{\#}$
and `reduction`(a, b) **and** `chain2term`$(b, \boxed{9})$

b. LE-encoding of shifting, encoding of λ-conversion by the sort *reduction*:

$$\mathbf{E}\boxed{1}\ \mathbf{E}\boxed{2}\ \mathbf{E}\boxed{3}\ \mathbf{E}\boxed{4}\ \mathbf{E}\boxed{5}\ \mathbf{E}\boxed{6}\ \mathbf{E}\boxed{7}\ \mathbf{E}\boxed{8}$$

$$\begin{bmatrix} \textit{derivation-rule} & \\ \text{IN} & \begin{bmatrix} \textit{word} & \\ \text{PHON} & \boxed{3} \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{psp} \\ \text{INV} & \boxed{5} \\ \text{AUX} & \boxed{6} \end{bmatrix} \\ \text{SUBCAT} & \left\langle \text{NP}\text{—}\boxed{2} \right\rangle \oplus \left\langle \boxed{1} \right\rangle \\ \text{MARKING} & \boxed{7} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{LF} & \lambda y \lambda z_n \ldots \lambda z_1 \lambda x \lambda u_m \ldots \lambda u_1 . \phi \end{bmatrix} \end{bmatrix} \\ \text{NONL} & \boxed{4} \end{bmatrix} \end{bmatrix} \\ \text{OUT} & \begin{bmatrix} \textit{word} & \\ \text{PHON} & \boxed{3} \\ \text{SYNS} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{pas} \\ \text{INV} & \boxed{5} \\ \text{AUX} & \boxed{6} \end{bmatrix} \\ \text{SUBCAT} & \left\langle \boxed{1}\text{—}\boxed{2} \right\rangle \\ \text{MARKING} & \boxed{7} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{RED AUX} & \boxed{9} \end{bmatrix} \end{bmatrix} \\ \text{NONL} & \boxed{4} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

and $^{\boxed{8}}\begin{bmatrix}(\lambda z_n \ldots \lambda z_1 \lambda y \lambda u_m \ldots \lambda u_1 . \exists x \phi)^{*}\end{bmatrix}$ **and** `shifting`$(\boxed{8}, \boxed{9})$

c. DR-encoding of shifting, encoding of λ-conversion by the sort *reduction*:

$$\mathbf{E}\boxed{1}\ \mathbf{E}\boxed{2}\ \mathbf{E}\boxed{3}\ \mathbf{E}\boxed{4}\ \mathbf{E}\boxed{5}\ \mathbf{E}\boxed{6}\ \mathbf{E}\boxed{7}$$

$$\begin{bmatrix} \textit{derivation-rule} \\ \text{IN}\ \begin{bmatrix} \textit{word} \\ \text{PHON}\ \boxed{3} \\ \text{SYNS}\ \begin{bmatrix} \text{LOC}\ \begin{bmatrix} \text{CAT}\ \begin{bmatrix} \text{HEAD}\ \begin{bmatrix} \textit{verb} \\ \text{VFORM}\ \ \textit{psp} \\ \text{INV}\ \ \boxed{5} \\ \text{AUX}\ \ \boxed{6} \end{bmatrix} \\ \text{SUBCAT}\ \ \left\langle \text{NP}\!-\!\boxed{2} \right\rangle \oplus \left\langle \boxed{1} \right\rangle \\ \text{MARKING}\ \boxed{7} \end{bmatrix} \\ \text{CONT}\ \begin{bmatrix} \text{LF}\ \ \lambda y \lambda z_n \ldots \lambda z_1 \lambda x \lambda u_m \ldots \lambda u_1 . \phi \end{bmatrix} \end{bmatrix} \\ \text{NONL}\ \boxed{4} \end{bmatrix} \end{bmatrix} \\ \text{OUT}\ \begin{bmatrix} \textit{word} \\ \text{PHON}\ \boxed{3} \\ \text{SYNS}\ \begin{bmatrix} \text{LOC}\ \begin{bmatrix} \text{CAT}\ \begin{bmatrix} \text{HEAD}\ \begin{bmatrix} \textit{verb} \\ \text{VFORM}\ \ \textit{pas} \\ \text{INV}\ \ \boxed{5} \\ \text{AUX}\ \ \boxed{6} \end{bmatrix} \\ \text{SUBCAT}\ \left\langle \boxed{1}\!-\!\boxed{2} \right\rangle \\ \text{MARKING}\ \boxed{7} \end{bmatrix} \\ \text{CONT}\ \begin{bmatrix} \text{RED AUX}\ \ \lambda x_n \ldots \lambda x_1 \lambda y \lambda u_m \ldots \lambda u_1 . \exists x \phi \end{bmatrix} \end{bmatrix} \\ \text{NONL}\ \boxed{4} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

The formalization given in (279a) assumes that shifting is done by some DRs as well, as indicated in (171). Therefore, we do not need to include a call to the relation `chain-shifting` in this version of the Passive DR. What is required, however, is a chain-description of the output of the Passive DR, given as the result of the function "#". A chain a that is described by this chain-description is β-reduced to another chain b, which is the chain-encoding of the final logical form of the output word.

In the next two encodings, we do not use chains, i.e., we assume an encoding of λ-conversion as presented in Section 4.2.1. In (279b), we combine this with an LE-encoding of shifting. This means that we include the application of shifting operations to the output of the Passive DR directly in the DR. For this reason, we assume that there is some *me* object $\boxed{8}$ that meets the description that results from applying the function "*" to the term which was given as the output of the DR in the informal specification in (269). This *me* object $\boxed{8}$ stands in the relation `shifting` with the CONTENT AUX value of the output word. Since the CONTENT value of the output word is of sort *content* as introduced in (215b) in Section 4.2.1, the term that corresponds to $\boxed{8}$ reduces to some redex-free term ϕ, which corresponds to the CONTENT LF value of the output word.

Finally, the encoding of the Passive DR as given in (279c) needs no relation calls. There, the CONTENT AUX value of the output word is the term that corresponds directly to the characterization of the output semantics in the informal specification of the Passive DR. This term reduces to the redex-free term that corresponds to the CONTENT LF value of the output word. This means that in this formalization, the application of shifting operations is not part of the formalization of the Passive DR, but is taken care of by other DRs, i.e., those given in (171).

These four alternative formalizations are equally capable of capturing the data. They differ, however, in the ontological commitments they make. In fact, the choice between them will largely depend on the encoding that has been chosen for shifting operations and for λ-conversion independent of the Passive DR. Comparing the four encodings given in (278) and (279), it should, however, be clear how one formalization can be transformed into the other, and how the information provided in the informal specification of this DR in (269) re-appears in the formalizations.

The second DR that we introduced in Section 2.3.2 was the Complement Extraction DR (CEx-DR), given in (102) on page 98. Remember from Section 2.3.2 that our analysis of unbounded dependencies does not assume traces. Instead, the CEx-DR "removes" an element from the SUBCAT list of a word and "puts" its LOCAL value in the INHERITED SLASH set of the output word. This SLASH specification also appears at signs that dominate the output of the CEx-DR by the NONLOCAL FEATURE PRINCIPLE (NFP) in (105). For topicalization structures, we assume a special ID SCHEMA, the HEAD-FILLER-SCHEMA in (106). This schema licenses phrases which have a head daughter with a single *local* object in its TO-BIND SLASH value which is identical with some member of the INHERITED SLASH value of the head daughter and with the LOCAL value of the nonhead daughter. In (280) we repeat the informal specification of this DR.

(280) Informal specification of the Complement Extraction DR (ignoring CONTENT):

$$\begin{bmatrix} \textit{word} \\ \text{SYNS} \begin{bmatrix} \text{LOC CAT SUBCAT } \boxed{1}\big[\textit{nelist}\big] \oplus \Big\langle \big[\text{LOC } \boxed{2}\big] | \boxed{3} \Big\rangle \\ \text{NONL INH SLASH } \boxed{4} \end{bmatrix} \end{bmatrix} \longmapsto \begin{bmatrix} \textit{word} \\ \text{SYNS} \begin{bmatrix} \text{LOC CAT SUBCAT } \boxed{1} \oplus \boxed{3} \\ \text{NONL INH SLASH } \boxed{4} \cup \Big\{\boxed{2}\Big\} \end{bmatrix} \end{bmatrix}$$

The SUBCAT list of the output word of this DR is one element shorter than that of the input word. The LOCAL value of the missing element appears as a member in the INHERITED SLASH set of the output, but not necessarily of the input.

In the present section, we want to consider the semantic side of our analysis of unbounded dependencies. The basic intuition behind our semantic analysis is that a sign that has a non-empty INHERITED SLASH value is not saturated semantically. For illustration, consider the sentence in (281) which is given with its logical form.

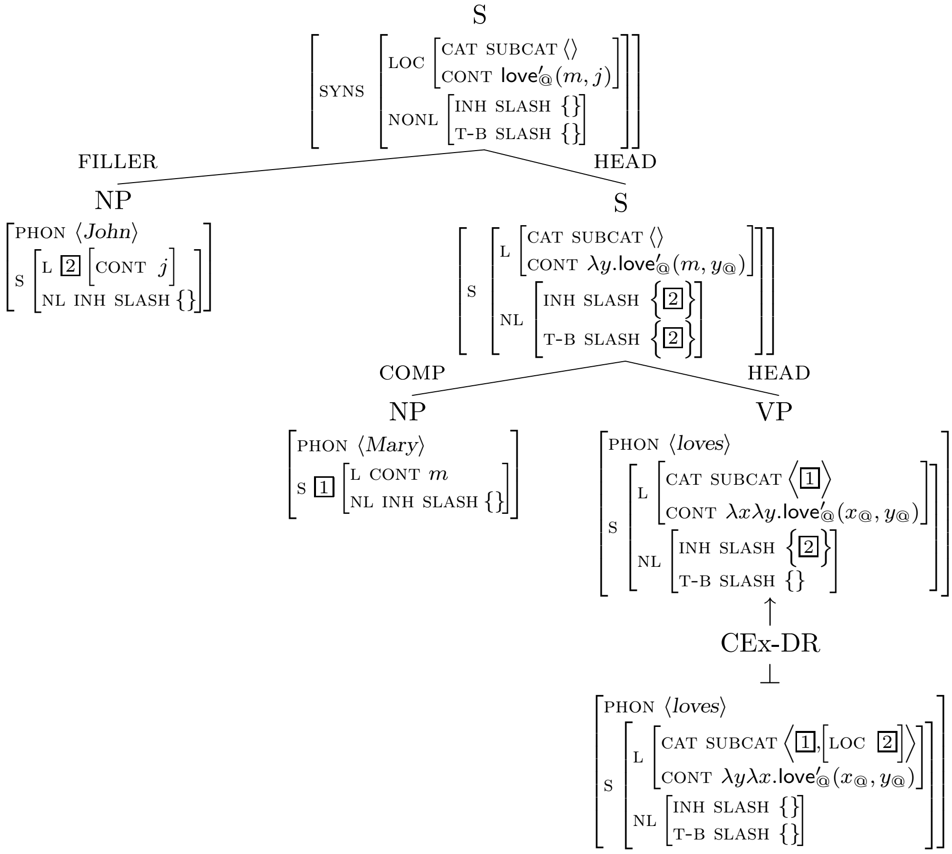
(281) John, Mary loves.
$\mathsf{love}'_{@}(m, j)$

By the CEx-DR, the syntactic valence of the verb *loves* is changed in such a way that it first combines with the subject NP, *Mary*, and then, in a head-filler structure, with the filler *John*. Semantically, the filler has exactly the same function as the direct object in the non-extracted case. Given these considerations, we assume that the basic translation of the verb *loves* is changed by the application of the CEx-DR as indicated in (282).

(282) *loves* $\rightsquigarrow$ $\lambda y \lambda x.\mathsf{love}'_{@}(x_{@}, y_{@})$
$\longmapsto_{\text{CEx-DR}}$ $\lambda x \lambda y.\mathsf{love}'_{@}(x_{@}, y_{@})$

In Figure 4.16, we show the structure of sentence (281). Applying the CEx-DR to the verb *loves* leads to a change on the SUBCAT list, i.e., instead of having two elements on the SUBCAT list, the output word has a single element, $\boxed{1}$ left. Since the second syntactic complement is "removed" from the SUBCAT list, we assume that it is also removed from its position in the logical form. Therefore the first λ-abstractor in the logical form of the output word binds the variable that corresponds to the subject (λx). The output word has a nonempty INHERITED SLASH value. Since we treat elements in the INHERITED SLASH value as semantic arguments, it is necessary to add another semantic argument to the logical form of the output. This leads to the logical form of the output as given in the last line of (282), i.e., to the term $\lambda x \lambda y.\mathsf{love}'_{@}(x_{@}, y_{@})$.

FIGURE 4.16. The structure of sentence (281):



At the S node, the subject NP *Mary* combines with the output of the CEx-DR. The S node has an empty SUBCAT list, but, as its INHERITED SLASH value is not empty. Therefore, it must still have a semantic argument missing. Our logical form at this node indicates this correctly, as it is the term $\lambda y.\mathsf{love}'_{@}(x_{@}, y_{@})$. Finally, at the higher S node, the filler is introduced into the structure. Now, the semantic argument indicated by the abstractor λy is satisfied. The resulting logical form is just as that of a sentence without topicalization.

After this simple example, we can enhance the informal specification of the CEx-DR by adding a CONTENT specification. This is done in (283). Semantically, the CEx-DR has the effect of putting one of the semantic arguments to the end of the λ-abstractors.

(283) Informal specification of the Complement Extraction DR (including CONTENT):

$$\begin{bmatrix} \textit{word} \\ \text{SYNS} \begin{bmatrix} \text{LOC} \begin{bmatrix} \text{CAT SUBCAT } \boxed{1}\begin{bmatrix}\textit{nelist}\end{bmatrix} \oplus \left\langle \begin{bmatrix}\text{LOC } \boxed{2}\end{bmatrix} \middle| \boxed{3} \right\rangle \\ \text{CONT } \lambda x_n \ldots \lambda x_i \ldots \lambda x_1.\phi \end{bmatrix} \\ \text{NONL INH SLASH } \boxed{4} \end{bmatrix} \end{bmatrix} \longmapsto$$

$$\begin{bmatrix} \textit{word} \\ \text{SYNS} \begin{bmatrix} \text{LOC} \begin{bmatrix} \text{CAT SUBCAT } \boxed{1} \oplus \boxed{3} \\ \text{CONT } \lambda x_n \ldots \quad \ldots \lambda x_1 \lambda x_i.\phi \end{bmatrix} \\ \text{NONL INH SLASH } \boxed{4} \cup \left\{\boxed{2}\right\} \end{bmatrix} \end{bmatrix}$$

(where $\begin{bmatrix}\text{LOC } \boxed{2}\end{bmatrix}$ is the i-th element on the SUBCAT list of the input word.)

So far, we only considered an instance of extraction within the same clause. As the following sentences show, topicalization can cross clause boundaries.

(284) a. John, Mary loves.
$\mathsf{love}'_@(m, j)$
b. John, Bill believes Mary loves.
$\mathsf{believe}'_@(b, \lambda @.\mathsf{love}'_@(m, j))$
c. John, Peter claims Bill believes Mary loves.
$\mathsf{claim}'_@(p, \lambda @.\mathsf{believe}'_@(b, \lambda @.\mathsf{love}'_@(m, j)))$

The (a) example is just the sentence whose structure we indicated in Figure 4.16. In (284b) the direct object of the verb *loves*, the NP *John*, appears as the filler one clause higher than the verb *loves*. In (284c), the filler is realized even two clauses up.

For our semantic analysis to work in these cases as well, we must apply value raising to the basic translation of the verb prior to applying the CEx-DR. In (285) we indicate how the logical form of the verb *loves* as needed for example (284b) is derived. In (285) we always indicate the semantic type of the variables introduced by VR.

(285) *loves* $\rightsquigarrow$ $\lambda y_{se} \lambda x_{se}.\mathsf{love}'_@(x_@, y_@)$
$\longrightarrow_{VR}$ $\lambda y \lambda x \lambda u_{s((st)((se)t))}.u_@(\lambda @.\mathsf{love}'_@(x_@, y_@))$
$\longrightarrow_{VR}$ $\lambda y \lambda x \lambda u \lambda v_{s((s((se)t))t)}.v_@(\lambda @.u_@(\lambda @.\mathsf{love}'_@(x_@, y_@)))$
$\longmapsto_{\text{CEx-DR}}$ $\lambda x \lambda u \lambda v \lambda y.v_@(\lambda @.u_@(\lambda @.\mathsf{love}'_@(x_@, y_@)))$

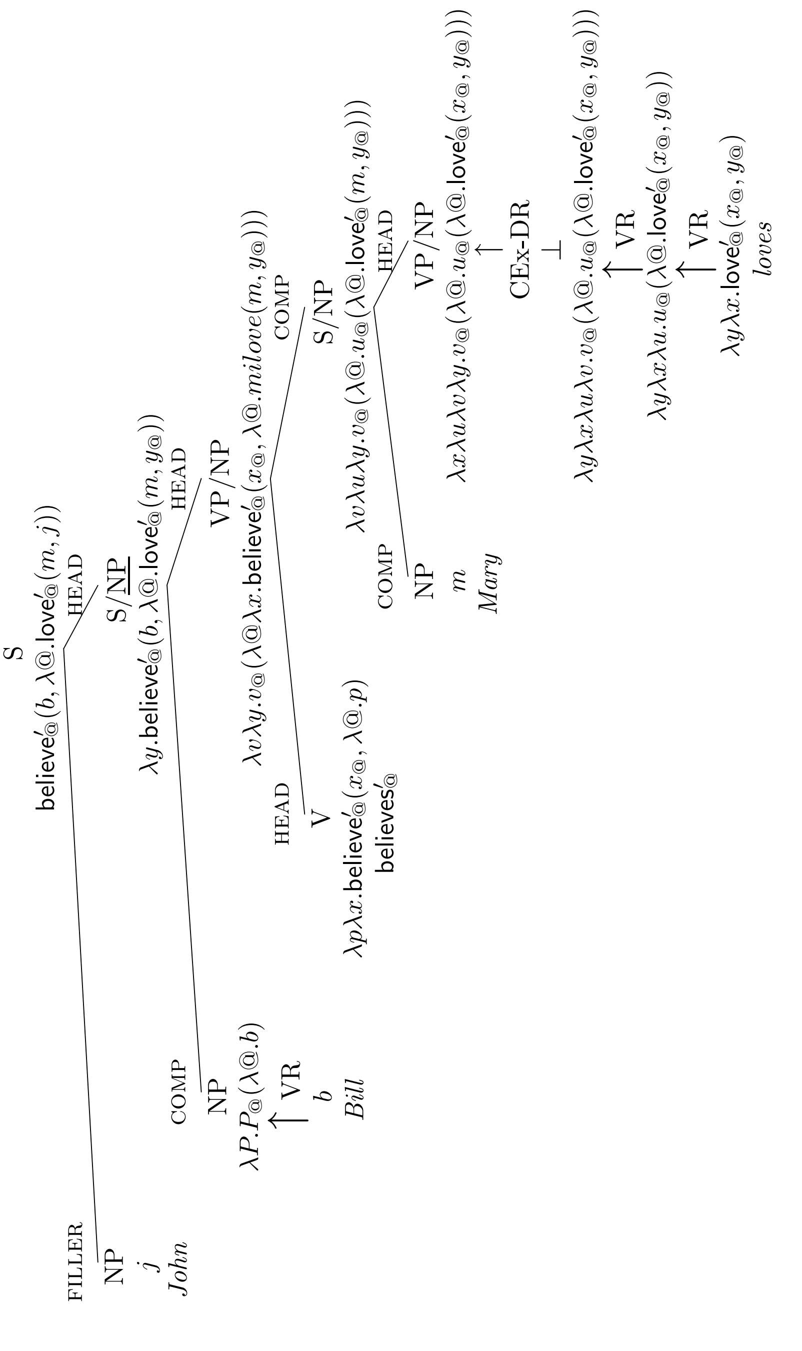
Given the derivation of the logical form of the verb *loves* as needed for example (284b), we indicate the structure of this example in Figure 4.17 (page 226). To keep the figure smaller, we do not use the AVM notation for the nodes. Instead, we indicate the syntactic category with the usual symbols, V, VP, S, and NP. In addition we write VP/NP for a VP sign that has a *local* object that corresponds to an NP in its INHERITED SLASH set. We write S/<u>NP</u> for an S node which has a single NP *local* in its INHERITED SLASH and in its TO-BIND SLASH value.

In Figure 4.17, the embedded verb *loves* has a non-empty INHERITED SLASH value, as indicated by the specification VP/NP. Its logical form is the result of the derivation given in (285). This verb combines with the embedded subject, *Mary*. The resulting phrase is syntactically saturated, i.e., it is an S node, but it inherits the slash specification from its head daughter by the NONLOCAL FEATURE PRINCIPLE. Therefore we write S/NP at this node. The logical form of this node is the result of combining the logical forms of the daughters with intensional functional application, using the logical form of the head daughter as the functor.

For the verb *believe*, we assume a logical form as given by its basic translation. This is a term of sort $((st)((se)t))$. As the variable u introduced by the first application of VR is of sort $s((st)((se)t))$, we can use the logical form of the embedded clause as the semantic functor in the calculation of the logical form of the matrix VP. The NFP, again, determines that the INHERITED SLASH value of the matrix VP contains the same element as the INHERITED SLASH set of the complement clause.

The matrix VP combines with the matrix subject. The subject NP, *Bill*, has as its logical form the value-raised version of the basic translation. As such, it is of the semantic type $(s((se)t))t$, i.e., the type of a quantified NP. But this is also the right semantic type to combine with the logical form of the matrix VP by intensional functional application, as we have introduced the variable v by the second application VR such that v is of type $s((s((se)t))t)$ (see (285)). As a result of this combination, we get a term of sort $(se)t$, i.e., a term whose remaining semantic argument corresponds to the extracted NP.

FIGURE 4.17. The structure of sentence (284b):



At the syntactic side, the INHERITED SLASH value of the matrix VP also appears at the lower matrix S node, as indicated by the notation S/NP. At the higher matrix S node, however, we have an instance of a head-filler construction. Therefore, the lower matrix S node is required to have a non-empty TO-BIND SLASH set. As said above, we indicate this by underlining in the category label, i.e., we write S/<u>NP</u>. The NFP then stops the NP from occurring in the INHERITED SLASH value of the highest S node in Figure 4.17. Semantically, in the highest local tree in the structure, the logical form of the filler daughter, i.e., the semantic constant j, combines with the logical form of the head daughter. The result is the term given in (284b).

This example has illustrated that it is necessary to allow the application of shifting rules before we apply the CEx-DR. With VR we have transformed the basic translation of the verb in such a way that it becomes the semantic functor along the entire extraction path.

Because of the traceless syntactic analysis, we were able to maintain a simple and uniform mode of semantic combination at the level of phrases, i.e., intensional functional application. If we had proposed a trace, we would have been forced to make other assumptions which would have been in need for motivation. Let us briefly sketch two options. First, we could follow the analysis of Pollard and Sag 1994 and assume a trace and the identity of LOCAL values between the trace and the filler. As the logical form is part of the LOCAL value of a sign, this means that the logical form of the filler/trace would appear twice in the structure, once as the CONTENT value of the trace, and once as that of the filler. We would, then, be forced to introduce some mechanism to prevent the same semantic contribution from occurring twice in the overall logical form. One such mechanism would be to change the SEMANTICS PRINCIPLE in such a way that the logical form of the nonhead daughter is ignored in the case of a head-filler structure.

Alternatively, it would have been possible to restrict the identity between filler and trace to some structure smaller than *local*, which does not include the logical form. As such, identities of CATEGORY values would be a promising candidate. The logical form of the trace could, then, be some complex semantic functor which has the same effect on the logical form of the head that it combines with as the effect encoded in the CEx-DR above.[6]

Both alternatives, thus, require to make more decisions, none of which can be motivated in this thesis and both would lead to more complicated semantic system. It is for this reason that we have adopted a traceless analysis of extraction.

In the discussion of the Passive DR, we have seen that we need to apply this DR *before* applying VR. The CEx-DR on the other hand was applied after VR. The question arises whether it is necessary to apply shifting operations to the result of the CEx-DR. The answer to this question depends on the judgments for sentences such as those in (286).

(286) a. Every man believes that John loves some woman.
b. Every man believes that some woman, John loves.

For a sentence such as (286a) we have followed the analysis of Montague in assuming that the scope of the existential quantifier *some woman* can be wider than the clause in which it appears. In particular, we have assumed that there are *de re* readings of this sentence where the quantified NP *some woman* has scope over the matrix predicate $\mathsf{believe}'$ and, possibly, even over the quantifier contributed the matrix subject *every man*, i.e., we expect to find one *de dicto* reading, and two *de re* readings, which could again be called ∀∃-*de re* and ∃∀-*de re*.

Example (286b) is similar to the previous example, but the embedded direct object *some woman* is topicalized inside the embedded clause. Sentence (286b) sounds admittedly odd in isolation, but similar sentences are fine in some appropriate context. If one accepts this sentence, it is at least possible to think of a reading in which there is a particular woman such that every man believes that John loves that woman. This reading corresponds to the ∃∀-*de re* reading in previous examples.

[6] In the semantic analysis of Generalized Phrase Structure Grammar (Gazdar et al., 1985) which we will present briefly in Section 7.1, the semantic contribution of the trace is different from that of the filler, and, furthermore, the semantic combination in a head-filler structure is different from that in other structures.

If such a reading is available, we must allow the application of shifting operations to the output of the CEx-DR. In (287) we show the derivation of the logical form of the verb *loves* as required for the ∃∀-*de re* reading of sentence (286b).

(287) *loves*
$$
\begin{array}{ll}
\leadsto & \lambda y \lambda x.\mathsf{love}'_{@}(x_{@}, y_{@})\\
\longmapsto_{\text{CEx-DR}} & \lambda x \lambda y.\mathsf{love}'_{@}(x_{@}, y_{@})\\
\longrightarrow_{VR} & \lambda x \lambda y \lambda u.u_{@}(\lambda @.\mathsf{love}'_{@}(x_{@}, y_{@}))\\
\longrightarrow_{AR2} & \lambda x \lambda Y \lambda u.Y_{@}(\lambda @ \lambda y.u_{@}(\lambda @.\mathsf{love}'_{@}(x_{@}, y_{@})))
\end{array}
$$

In (287), we first apply the CEx-DR to the verb *love* in its basic translation. This has the effect of re-ordering the semantic arguments of the verb. Then, we apply VR to add a further semantic argument, that corresponds to the material from the matrix clause. Finally, we apply AR to achieve that the embedded direct object can have scope over material in the matrix clause. In Figure 4.18 (page 229), we show how the wide scope reading of the embedded direct object quantifier can be derived, using the logical form of the verb *loves* as given in the last line in (287).

In Figure 4.18 we use the abbreviatory notation for the nonlocal values introduced above: We write VP/NP for the output of the CEx-DR, i.e., for a verb which has one element on its SUBCAT list and an NP-*local* in its INHERITED SLASH value. We write S/<u>NP</u> for a verb with an empty SUBCAT value and the same NP-*local* in its INHERITED SLASH set and in its TO-BIND SLASH set.

In the figure, we have left out the derivation of the logical form of the embedded verb *loves*, as it is given in (287). We must, however, also apply some shifting operations to the basic translation of the matrix verb. These are indicated in the structure. The shifting operations applied are exactly those that we used for the derivation of the ∀∃-*de re* reading in Figure 1.15 (page 55).

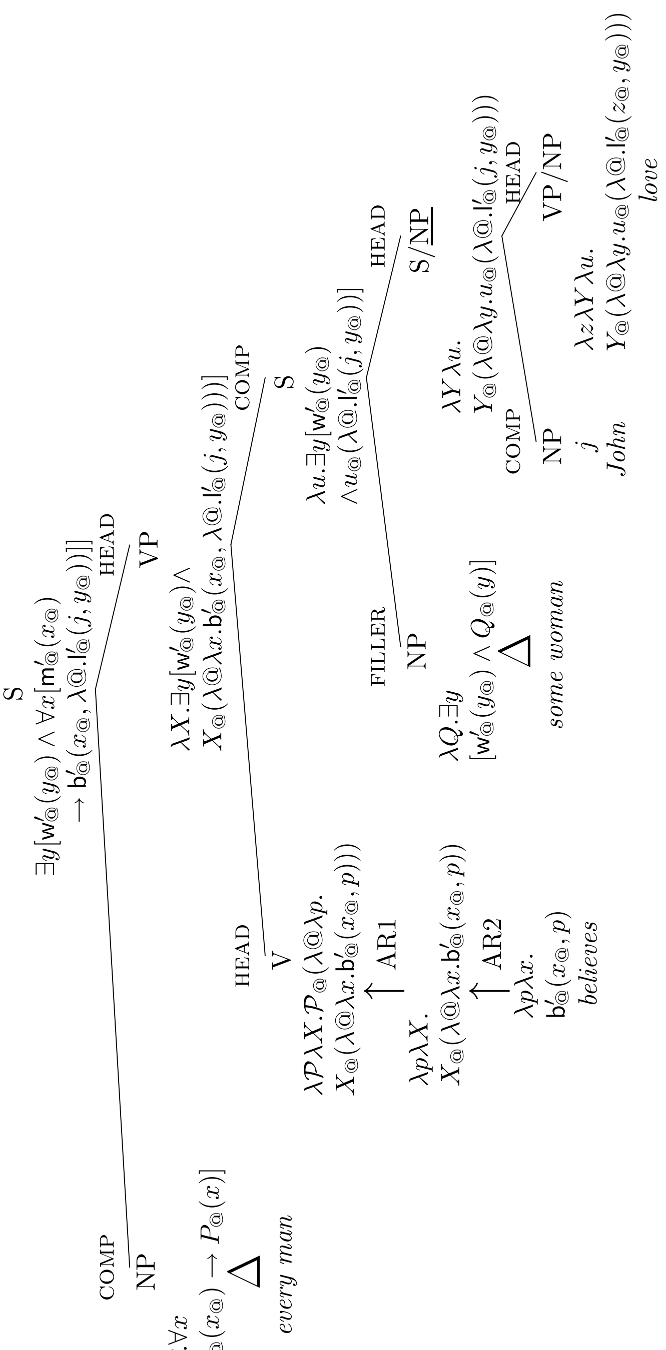
Given the considerations around the examples in (286), we assume that shifting operations may apply to the output of the CEx-DR. We can, then, state the DR in a more formal way than we stated in (283). In (288) we give the CEx-DR as it is needed for a LE-encoding of shifting operations with the approach to λ-conversion given in Section 4.2.2.

(288) Formal specification of the Complement Extraction DR (CEx-DR):

$$
\mathbf{E}\boxed{1}\ \mathbf{E}\boxed{2}\ \mathbf{E}\boxed{3}\ \mathbf{E}\boxed{4}\ \mathbf{E}\boxed{5}\ \mathbf{E}\boxed{6}\ \mathbf{E}\boxed{7}\ \mathbf{E}\boxed{8}\ \mathbf{E}\boxed{9}\ \mathbf{E}a\ \mathbf{E}b\ \mathbf{E}c
$$
$$
\left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{der-rule}}\\
\text{IN} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{word}}\\
\text{PHON} & \boxed{5}\\
\text{SYNS} & \left[\begin{array}{ll}
\text{LOC} & \left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}
\text{HEAD} & \boxed{6}\\
\text{SUBCAT} & \boxed{1}\left[\textit{nelist}\right] \oplus \left\langle \left[\text{LOC}\ \boxed{2}\right] \middle| \boxed{3}\right\rangle\\
\text{MARKING} & \boxed{7}
\end{array}\right]\\
\text{CONT} & \lambda x_n \dots \lambda x_i \dots \lambda x_1.\phi
\end{array}\right]\\
\text{NONL} & \left[\begin{array}{ll}
\text{INH SLASH} & \boxed{4}\\
\text{T-B SLASH} & \boxed{8}
\end{array}\right]
\end{array}\right]
\end{array}\right]\\
\text{OUT} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{word}}\\
\text{PHON} & \boxed{5}\\
\text{SYNS} & \left[\begin{array}{ll}
\text{LOC} & \left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}
\text{HEAD} & \boxed{6}\\
\text{SUBCAT} & \boxed{1} \oplus \boxed{3}\\
\text{MARKING} & \boxed{7}
\end{array}\right]\\
\text{CONT} & \boxed{9}
\end{array}\right]\\
\text{NONL} & \left[\begin{array}{ll}
\text{INH SLASH} & \boxed{4} \cup \left\{\boxed{2}\right\}\\
\text{T-B SLASH} & \boxed{8}
\end{array}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right]
$$

and $(\lambda x_n \dots \ \dots \lambda x_1 \lambda x_i.\phi)^{\#}$

FIGURE 4.18. The structure of the $\exists\forall$-*de re* reading of sentence (286b):



and `chain-shifting`(a, b)
and `chain2term`$(b, \boxed{9})$

The structure of this DR is just like that of the Passive DR given in (278), i.e., below the AVM, a chain a is specified as the chain-encoding of the term that was given as the logical form of the output word in the informal version of the DR. This chain is related via shifting operations and λ-conversion to a chain b which is the chain-encoding of the logical form of the output word.

Again, just as in (278) the version of the DR given in (288) is not a complete formalization, as we use the "…" notation in the characterization of the input and the output term. It is, however, straightforward to define a relation `cex-lf` in analogy to the relation `passive-lf` that relates the input word, its logical form and the chain a.

In this section, we extended the framework LF-Ty2 to include the analysis of passive and complement extraction as presented in Section 2.3.2. In the syntactic analysis, we made use of Derivational Rules (DRs). A DR establishes a link between two *word* objects. We interpreted this kind of link as a unary tree with a *word* object, the output of the DR, as the mother and another *word*, the input of the DR, as the single daughter. In the present section, we have shown that both DRs of Section 2.3.2 are such that the logical form of the output cannot, in general, be identical to that of the input. Therefore, we added some CONTENT specification to the formulation of the DRs.

In our first presentation of the semantic framework LF-Ty2 in Section 1.3 we only considered two kinds of signs: words and phrases, or lexical and non-lexical elements. At that stage, it could be assumed that words and lexical elements are the same, and, similarly, that phrases and non-lexical elements are the same. We defined the logical form of a word as its basic translation or the result of a finite number of applications of shifting operations to its basic translation. The logical form of a phrase, on the other hand, is simply the result of intensional functional application of the logical forms of the daughters.

With the introduction of derived words, the notion of a word is no longer synonymous to that of an element that is directly licensed by some lexical entry: the output of a DR is a word, but this word is **not** licensed by some lexical entry. In this sense, it is a word but not a lexical element. By looking at the possible readings for passive sentences and also for topicalization structures, we came to the conclusion that shifting operations must be applicable to all words, be they derived or non-derived.[7]

4.4. Summary

In this chapter, we combined the syntactic analysis presented in Section 2.3 with the RSRL formalization of the semantic representation language Ty2 as provided in Chapter 3 to yield an RSRL grammar that fulfills the requirements expounded in Chapter 1. The resulting grammar, thus, furnishes each linguistic sign with a CONTENT value that should be considered its logical form. By the results of Chapter 3 we know that these logical forms stand in direct correspondence to terms of Ty2 and, therefore, can be interpreted directly with respect to a semantic model. Because of the choice of Ty2, we are in a position to integrate the work carried out in the tradition of Montague Grammar in HPSG. In this chapter, we showed that the analysis of quantifiers and intensional verbs as presented in Montague 1974b can be used within an HPSG grammar.

While the semantic analysis of the sentences is as proposed in the work of Montague, the syntactic fragment is a modified version of the grammar of Pollard and Sag 1994. We also followed Pollard and Sag 1994 in the assumption that the CONTENT value of a sign is its logical form, i.e., some representation of its meaning. We differ, however, with respect to the semantic representation language, and, most importantly, with respect to the mechanisms that we use to constrain the CONTENT value of a phrase with respect to those

[7] In Part II of this thesis, we will consider yet another new kind of entities, phrases that are licensed directly by some lexical entry. We will show that the application of shifting operations must be possible for these elements as well.

of its daughters. In Section 1.2 we presented the semantic treatment of Pollard and Sag 1994, showing that it uses some storage mechanism to account for scope ambiguities. This storage mechanism is implemented by stipulating additional components of linguistic signs that lack independent motivation, such as the QSTORE and the RETRIEVED values of a sign.

In contrast to this, it was shown that the semantic analysis provided in the preceding sections does not introduce any additional components to a sign apart from a CONTENT value which is a term of the semantic representation language. This was possible because we used the flexible system LF-Ty2 that is based on Hendriks 1993. This system makes it unnecessary to use explicit storage mechanisms. Instead, it allows us to shift from basic translations of words to complex functors which encode the scoping possibilities. We, thus, consider the approach to semantics presented in this chapter a progress in comparison to the analysis given in Pollard and Sag 1994 in two respects: first, because it is clear that and how the logical forms presented in this chapter are interpreted, and second, because we do not assume structure for which there is no independent motivation.

After these general remarks on the architecture for semantics presented in this chapter, one part of the analysis presented in Section 1.3.3 remains to be discussed: the formulation of constraints on logical forms. This issue is central, because it is the existence or absence of such constraints that will give the ultimate argument in the debate whether or not there is a logical form as a level of representation. In Section 1.3.3 we presented a simple lf constraint. In (53) we gave the following formulation:

(289) (= (53))
For each node of category S,
the logical form of S does not contain a subterm of the form $\lambda x.\psi$ such that ψ has a subterm $\forall v\ \phi$ which has a free occurrence of the variable x.

This constraint was used to exclude a universal quantifier to take scope over material which is outside the clause in which the quantifier is introduced. For illustration, we considered example (51) which is repeated as (290). This sentence does not have a reading in which for each woman there is some man who believes that that woman walks.

(290) Some man believes that every woman walks.

The constraint in (289) is depicted in its formalization in (291).

(291) Formalization of the lf constraint in (53):

$$\mathbf{A}\boxed{1}\left(\left[\text{SYNS LOC}\left[\begin{array}{ll}\text{CAT} & \left[\begin{array}{ll}\text{HEAD} & \textit{verb}\\ \text{SUBCAT} & \textit{elist}\end{array}\right]\\ \text{CONT} & \boxed{1}\end{array}\right]\right] \Rightarrow \begin{array}{l}\textbf{not}\ \mathbf{E}\boxed{2}\ \mathbf{E}\boxed{3}\ \mathbf{E}\boxed{4}\\ \left(\begin{array}{l}\boxed{2}\left[\begin{array}{l}\textit{abstr}\\ \text{VAR}\ \boxed{3}\end{array}\right]\ \textbf{and}\ \boxed{4}\left[\textit{universal}\right]\\ \textbf{and}\ \texttt{subterm}(\boxed{2},\boxed{1})\\ \textbf{and}\ \texttt{subterm}(\boxed{4},\boxed{2})\\ \textbf{and}\ \texttt{free-variable}(\boxed{3},\boxed{4})\end{array}\right)\end{array}\right)$$

The principle in (291) expresses the constraint in the following way: for each $\boxed{1}$, if $\boxed{1}$ is the logical form of a saturated verb (i.e., of an S node), then there are no $\boxed{2}$, $\boxed{3}$, $\boxed{4}$ such that $\boxed{2}$ is a subterm of $\boxed{1}$ of the form $\lambda\boxed{3}.\psi$ and $\boxed{4}$ is a subterm of $\boxed{2}$ and the variable $\boxed{3}$ is a free variable of the term $\boxed{4}$.

The formalization in (291) shows how the CONTENT value of a sign can be constrained depending on some other properties of the sign. These mutual constraints are possible because the logical form is a component of the sign just as all other properties of the sign.

When we first introduced the constraint in (289) in Section 1.3.3, we pointed out that it is not obvious how the constraint could be expressed in frameworks that assume a syntactic

LF (May, 1985; von Stechow, 1993) or frameworks that assume a direct interpretation of syntactic structures (Lappin, 1991). As shown in (291), in our HPSG-rendering of LF-Ty2, such a constraint is very natural. This indicates that the present proposal for a logical form might make different predictions on the kinds of constraints on interpretation that are available in a grammar.

With this short indication of how lf constraints can be expressed in LF-Ty2, we conclude the presentation of this framework. Since, in the present chapter, we integrated the semantic fragment of Section 1.3.3 and the syntactic fragment of Section 2.3, we achieved the goal of Part I of this thesis: we presented a framework for combinatorial semantics which reflects the high degree of regularity attested in the form-meaning correlation. In particular, we have directly implemented the principle of compositionality, which we repeat in (292).

(292) The *Principle of Compositionality*

The meaning of a non-lexical element is a function of the meaning of its parts.

In the first part of this thesis, we encountered two classes of *non-lexical elements*: phrases and derived words. In both cases, the meaning of these elements was characterized as a function of the meaning of the parts. The meaning of a phrase is the intensional functional application of the meaning of its daughters, and the meaning of a derived word is a function of the meaning of the input word of a Derivation Rule. Thus, the principle of compositionality is directly incorporated into our architecture of combinatorial semantics.

The only kind of *lexical* elements that we looked at so far are non-derived words. For them, we simply assumed that the logical form is specified in the lexical entry. In Part II of this study, we will show that there are also some phrases which must be considered as lexical elements. Furthermore, we will investigate some requirements of words on the linguistic context in which they may occur. Both these phenomena cannot be captured in the framework developed so far. What both phenomena have in common is that they are instances of idiosyncratic and irregular behavior of signs. Thus, we will present an extension of the present architecture of grammar which incorporates a module of irregularity.

4.5. Excursion: The Potential of Chains and Quantification in RSRL

The language RSRL differs from its predecessor *Speciate Re-entrant Logic* (SRL) as presented in King 1989, 1994, 1999 in several ways. The major innovation of RSRL consists in the introduction of relations, quantification and chains. In Section 1.1 and in Chapter 2 we commented on some examples for the use of these innovations. In the following, we will concentrate on the question of what chains contribute to the grammar. This question is intimately linked to the interpretation of descriptions of the form $\mathbf{E}v\ \delta$ or $\mathbf{A}v\ \delta$. This is the case, because the descriptions in the theory of a grammar do not contain free variables. Therefore, the only possibility to use chains is if some quantified description introduces a variable. Such a variable can, then, be assigned a chain by the variable assignment function. As a consequence, if we want to explore the effect of chains in the grammar, it is useful to see what happens if we give a different interpretation to descriptions of the form $\mathbf{E}v\ \delta$ or $\mathbf{A}v\ \delta$, i.e., an interpretation which does not involve chains.

In (293) we give three possible interpretations for descriptions of the form $\mathbf{E}v\ \delta$. First, we could restrict the range of quantification to components of the described object, i.e., to elements of Co^u_I (293a). Second, we let quantifiers range over components and chains of components, just as is done in Definition 2.13. This interpretation is repeated in (293b). Third, we let quantifiers range over arbitrary objects of the universe, as indicated in (293c).

(293) Conceivable interpretations of descriptions of the form $\mathbf{E}x\ \delta$:

a. Quantification over components:

for each $v \in \mathcal{VAR}$, for each $\delta \in \mathcal{D}^{\Sigma}$,

$$D_{\mathsf{I}}^{ass}(\mathbf{E}v\ \delta) = \left\{ u \in \mathsf{U} \;\middle|\; \begin{array}{l} \text{for some } u' \in \mathsf{Co}_{\mathsf{I}}^{u}, \\ u \in D_{\mathsf{I}}^{ass\frac{u'}{v}}(\delta) \end{array} \right\},$$

b. Quantification over components or chains of components:

for each $v \in \mathcal{VAR}$, for each $\delta \in \mathcal{D}^{\Sigma}$,

$$D_{\mathsf{I}}^{ass}(\mathbf{E}v\ \delta) = \left\{ u \in \mathsf{U} \;\middle|\; \begin{array}{l} \text{for some } u' \in \overline{\mathsf{Co}_{\mathsf{I}}^{u}}, \\ u \in D_{\mathsf{I}}^{ass\frac{u'}{v}}(\delta) \end{array} \right\},$$

c. Quantification over the universe:

for each $v \in \mathcal{VAR}$, for each $\delta \in \mathcal{D}^{\Sigma}$,

$$D_{\mathsf{I}}^{ass}(\mathbf{E}v\ \delta) = \left\{ u \in \mathsf{U} \;\middle|\; \begin{array}{l} \text{for some } u' \in \mathsf{U}, \\ u \in D_{\mathsf{I}}^{ass\frac{u'}{v}}(\delta) \end{array} \right\},$$

It has been demonstrated in Richter et al. 1999 and Richter 2000 that the first alternative is not satisfying from the point of view of the grammar writer, because linguists want to express complex relations between parts of a linguistic object. In many cases, the objects stand in these relations only indirectly. As an example, re-consider part (a) of the SEMANTICS PRINCIPLE in Pollard and Sag 1994 (p. 401), which was illustrated in Section 1.2 and which we repeat in (294).

(294) In a headed phrase, the RETRIEVED value is a list whose set of elements is disjoint from the QSTORE value set, and the union of those two sets is the union of the QSTORE values of the daughters.

Even if we ignore the additional complications introduced by the set/list distinction, the principle expresses a complex relation between (i) the RETRIEVED value of a phrase, (ii) its QSTORE value and (iii) the QSTORE values of its daughters. The relation is mediated by some construct, namely the union of (i) and (ii) which is required to be the same as the union of all values of (iii). This union is, however, not necessarily a component of the described object.

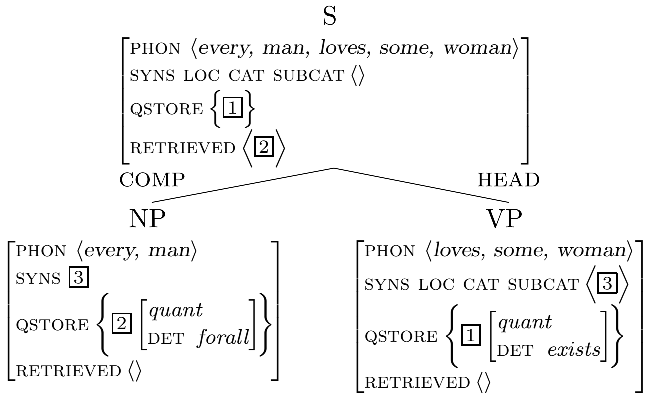
For illustration, consider the example in (295a). We are interested in a reading as given in (295b), i.e., a reading in which the quantifier that is contributed by the embedded direct object *some woman* has wide scope over the matrix predicate *believe*, whereas the quantifier contributed by the subject of the embedded clause takes scope inside the embedded clause.

(295) a. John believes [$_S$ every man [$_{VP}$ loves some woman]].

b. $\exists y[\mathsf{woman}'_{@}(y_{@}) \wedge \mathsf{believe}'_{@}(j, (\lambda @.\forall x[\mathsf{man}'_{@}(x_{@}) \rightarrow \mathsf{love}'_{@}(x_{@}, y_{@})]))]$

If we want to derive this reading in the analysis of Pollard and Sag 1994, the two quantifiers *every* and *some* have non-empty QSTORE values. Since the quantifier contributed by the determiner *some* has wide scope over the matrix predicate, this quantifier must be an element of the QSTORE value of the embedded S node. In contrast, the quantifier contributed by the determiner *every* must be retrieved either at the embedded S node. In (296) we describe the embedded S node, focusing on the QSTORE and RETRIEVED values of the phrase and its daughters.

(296) Description of the embedded S node of example (295a) in the intended reading:



The S node as described in (296) satisfies the SEMANTICS PRINCIPLE of Pollard and Sag 1994 as given above in (294): the disjoint union of the QSTORE value of the phrase and the set containing the elements of the RETRIEVED list is the set $\{\boxed{1}, \boxed{2}\}$. This set is the same as the union of the QSTORE values of the daughters.

The problem is, however, that the set $\{\boxed{1}, \boxed{2}\}$ is not a component of the phrase described in (296). Therefore, we cannot use an interpretation of quantification that uses only components of the described object, i.e., the interpretation of quantification given in (293a) is not sufficient for the grammar as envisaged in Pollard and Sag 1994.

The definitions in (293b) and (293c) can, however, capture the SEMANTICS PRINCIPLE of Pollard and Sag 1994. In fact, in the formalization of all principles of the appendix of Pollard and Sag 1994, Richter 2000 shows that whenever some additional structure is required, it is sufficient to assume some list-like structure whose elements are components of the described object. This is also the case in our example, i.e., the quantifiers $\boxed{1}$ and $\boxed{2}$ are components of the phrase, but the set $\{\boxed{1}, \boxed{2}\}$ is not.

The difference between the interpretation of quantification given in (293b) and (293c) is that in the first case, there is no other way for the auxiliary structure than to be list-like and to contain components of the described object. In the second interpretation, it would be a mere coincidence.

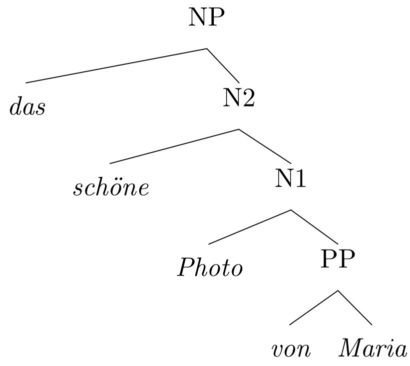
To sum up, Richter 2000 (Appendix C) shows that all of Pollard and Sag 1994 can be formalized in RSRL by quantification over objects directly or, in the case of more complex relations, by quantifying over "virtual" lists that can be used to express a relation between components of an object.

In Sailer 1997 this use of quantification is extended to encode chunk structures in chains. Building on Continuing from Abney's (1991; 1992; 1996) line of thought, Sailer 1997 argues that chunks should be considered the unit that is relevant for word order, instead of constituents, as is done in many of the publications on word order in HPSG (Sag, 1987; Reape, 1994; Kathol, 1995; Richter and Sailer, 1995; Richter, 1997).

For illustration, consider the following German example, taken from Sailer 1997 (p. 302). According to Abney's theory of chunks, the NP in (297a) consists of two chunks, each of which starts with a functional word, the determiner *das* (*the*) or the preposition *von* (*of*), and ends with a substantive word, the noun *Photo* (*picture*) and the proper name *Maria* respectively. The phonology of these chunks is indicated by the bracketing in (297a). The syntactic structure of the NP is as given in (297b). What is important is that the chunk *das schöne Photo* in (a) does not correspond to a constituent in the tree in (b).

(297) a. [das schöne Photo] [von Maria]
the beautiful picture of Maria

b.



In Sailer 1997, a chunk is characterized as a list that contains some nodes of a tree. There are certain requirements for chunks, in particular, the nodes in a chunk must be a partial order under the dominance relation. Chunk can be formalized as a chains which contain signs.

The two chunks in (297a) can, thus, be represented as chains which contain subtrees of the tree in (297b). In (298), the chunks are given by enumerating the root labels of the subtrees of the overall structure which are part of the chunks. The first chunk contains all subtrees of the NP except for those which are dominated by the PP node. These remaining subtrees, on the other hand, belong to the second chunk. Clearly, the subtrees in each of the chunks in (298) are a partial order under the dominance relation.

(298) The chunks of the NP:
⟨NP, *das*, N2, *schöne*, N1, *Photo*⟩
⟨PP, *Maria*, *von*⟩

A complication arises once the chunk structure of a clause is considered. As chunks constitute the minimal units for linearization, the chunk structure of a clause consists of several chunks. But, chunks are encoded as chains, so chunk structures should be a "chain of chains". This is not possible because an element of a chain must be an object from the universe U and cannot be a chain again. The solution proposed in Sailer 1997 relied on standard techniques of mapping a list of lists to a list. In the particular application, a simple chain was used to represent a chain of chains: To combine the two chunks in (298) to form the chunk structure of the NP in (297), the two chunks are concatenated and a separation symbol is inserted between them. This is illustrated in (299).

(299) The chunk structure of the NP:
⟨NP, *das*, N2, *schöne*, N1, *Photo*, *elist*, PP, *Maria*, *von*⟩

In this example, we use an object of the sort *elist* as separation symbol. The separation symbol must fulfill certain requirements: First, it must be a component of the described object. Due to the way linguistic signs are defined — by the finiteness of the SUBCAT list or the PHON list for example — we can safely assume that every linguistic sign has a component of sort *elist*. Second, the separation element must be different from those elements that can appear inside chunks. As chunks consist of *sign* objects, an object of sort *elist* cannot be a proper part of a chunk.

Once such structures are defined, we need relations to operate on them. For chunk structures, these are concatenation, permutation of chunks on a chunk structure, a relation to determine whether a given chunk is on a chunk structure etc. Some of these relations are given in Sailer 1997.

This use of chains is essentially different from the "standard" use explained earlier: in the case of the SEMANTICS PRINCIPLE, the chain contained only elements that were of direct linguistic relevance and the only relevant aspect of the structure of the chain was its capacity to store these elements. In chunk structures, we use special elements such as the *elist* object in addition to those objects which are of real linguistic relevance, i.e., the signs. Also, the structure of the chain becomes extremely important and we must access it by specific relations.

If we allow such uses of chains, we have, indirectly, extended the range of quantification. Now, we do not only quantify over components and virtual lists of components of an object, but we also quantify over other data types which contain components of the object, such as chunk structures.

After this brief overview of the uses of chains that have been proposed so far within RSRL, we can show that the use of chains in Section 4.2.2 is even more abstract. We have defined chains that encode terms of our semantic representation language. The elements on these chains were not subterms of these terms, instead they were arbitrary objects of two — almost arbitrary — sorts, *s1* and *s2*. Thus, in the chain-encoding of terms, it is the *structure* of the chain that relates to the term or to some *me* object, not the *elements* on the chain. We could, for example assume two term-encoding chains that have the same *s1* and *s2* objects, but in a different order. In that case, the chains necessarily encode different terms. On the other hand, two chains might use distinct objects of sort *s1* and *s2*, but have the same $s2s1^i$ sequences. In this case, the two chains express the same term.

We also defined some specific relations on term-encoding chains, such as the relations `chain-subterm`, `chain-replace` etc. These relations allowed us to work with term-encoding chains as if they were objects of sort *me*, or terms of Ty2. This encoding was possible because objects of sort *me* and terms of Ty2 are of a relatively simple structure. In particular, they are finite, acyclic, the constraints on semantic types are very local, and the distinction whether some subterms are identical or merely "look alike" is not used in the grammar.

If we can give a chain encoding for all objects of a certain kind, independent of the object that we are actually describing, a quantifier which binds a chain variable, then, can be seen to range over all these virtual objects. To come back to our grammar of LF-Ty2, let us assume that the sorts *s1* and *s2* that are used for the chain-encoding of terms/*me*-objects are the sorts *me* and *type*. In this case, every object of sort *me* is such that for each term ϕ, there is a chain c which consists only of components of this *me* object and which encodes the term ϕ. This means that if the description of some *me* object contains a sub-description of the form $\mathbf{E}v\,\delta$, then the variable v implicitly ranges over all terms of the semantic representation language.

We exploited this property in our formulation of the SEMANTICS PRINCIPLE in (252). There the term that represents the unreduced intensional functional application of the logical forms of the daughters does not appear as a component of the sign. Yet it is available to us, because we can refer to a chain that encodes this term.

As a result, using the chain-encoding of *me* objects, we can use quantification in such a way that we implicitly quantify over all *me* objects, whenever we describe an object that has components of the sort *s1* and *s2*. But, as this is the case, the question arises why we do not quantify over the entire universe in the first place, i.e., why do we not use (293c) as the definition of the interpretation of quantified descriptions? In Richter et al. 1999 (pp. 288f.) and Richter 2000 (pp. 148f.) it is argued that the way the principles of grammar are stated in Pollard and Sag 1994 and other HPSG sources implies that non-components are never taken into consideration.

In general, adopting an interpretation as given in (293c) would lead to a formal system which is crucially different from RSRL. In particular the model theory of RSRL relies on the particular type of quantification. For illustration, let us just mention one such difference: In our presentation of RSRL in Section 1.1 and in Chapter 2 we pointed out that we can determine what kinds of configurations of object we will find in the model of a grammar, but we cannot determine how many copies of congruent configurations there are. Under an interpretation of quantification as indicated in (293c), however, we can specify the cardinality of the model. Consider the description in (300).

(300) $\mathbf{A}\boxed{1}\ \mathbf{A}\boxed{2} \left(\begin{pmatrix} \boxed{1}\left[\textit{elist} \right] \\ \textbf{and}\ \boxed{2}\left[\textit{elist} \right] \end{pmatrix} \Rightarrow \left(\boxed{1} \approx \boxed{2} \right) \right)$

Under the regular RSRL interpretation, the description in (300) describes an object *o* iff there is at most one component of this object which is of sort *elist*. Under an interpretation as given in (293c) the description in (300) describes an object iff there is at most one object of sort *elist* in the universe. This indicates that changing the interpretation of quantified descriptions leads to major changes in the model theory and should, therefore, be avoided unless there is massive evidence against the present model-theoretic assumptions.

Beside these general reasons, there is also a more concrete reason for us for not using an interpretation of quantification as given in (293c). If we use (293c) then we do not have chains as part of the denotation of descriptions. So far, there is no known way to express the finiteness requirement on objects of sort *me* in RSRL without using chains (see the use of chains in the General Finiteness Principle in (131c)). This means that giving up chains would also lead to giving up the intimate correspondence between objects of sort *me* and terms of Ty2.

In addition, there are some limitations as to which objects can be represented as chains. As mentioned above, the chain-encoding of *me* objects is possible, because these objects are finite. Had we chosen a different kind of linguistic objects for which we do not require finiteness, the existence of a chain encoding would not be guaranteed. As a consequence, the linguistic universe may contain configurations for which we cannot give a chain encoding. This indicates that there is a difference in the range of quantification between (293c) and the definition used in RSRL: in the first case, quantification ranges over all objects, whereas in the second case it ranges over the components of some object and, implicitly, over configurations for which we can give a chain encoding.

Let us, finally, consider yet another possible use of chains. This is a use which has, so far, not been explored in practice within any RSRL grammar. Still, it is interesting from a conceptual point of view, because it shows that using chains, we can also implicitly quantify over chain-encoded structures which are not part of the linguistic universe. As such, this shows that if we use quantification over chains, we may quantify over structures which do not correspond to any configuration in the universe, i.e., over structures which we could not access under the interpretation of quantified descriptions given in (293c).

So far we have taken chains to represent entities that are built in accordance with the signature. It is, however, conceivable, to define chain encodings of structure that do not exist at the object level. A simple example would be an RSRL grammar which uses a signature that does not contain lists. In this case, chains would introduce a virtual list structure, but there would not be an actual list structure in the model of the grammar. If we put this in the context of quantification, this means that with (293b), we are able to quantify over virtual structures, whereas under (293c) we can only quantify over concrete, existing structures.

To give an example for this use which is not completely implausible, we can re-consider the chain encoding of terms of Ty2. Given the encoding of terms as presented above, we can, in principle eliminate terms of Ty2 from our linguistic universe. To be more precise, assume a signature as the one given in Section 2.3, but without the attribute CONTENT and the sort *content*. In such a grammar, there are no objects of sort *me* that occur as components of signs. This means that the signs in the denotation of this grammar do not have an explicit logical form as one of their components. Still, we can use the chain encoding of terms to relate these signs with a "virtual logical form".

Under such an approach, we define a relation `basic-translation` that holds of a pair $\langle w, c\rangle$ iff w is a word and c is a term-encoding chain such that c encodes the basic translation of the word w. To give an example, this relation holds between a word described by the lexical entry for the word *Mary* and some chains described by $m^{\#}$, or between a word described by the lexical entry for *walks* and some chains described by $(\lambda x.\mathsf{walk}'_{@}(x_{@}))^{\#}$.

Given this relation, we can define a relation `logical-form` that holds of a pair $\langle s,c\rangle$ iff s is a sign and c is a term-encoding chain, and one of the following conditions is met: First, if s is a word, then there is a term-encoding chain c' such that the pair $\langle s, c'\rangle$ is in the relation `basic-translation` and the pair $\langle c', c\rangle$ is in the relation `chain-shifting`. Or second, if s is a phrase than there are term-encoding chains c_1 and c_2 such that each daughter of s stands in the relation `logical-form` with one of these chains, and c expresses the intensional functional application of the terms encoded by c_1 and c_2. Without much effort, we could require in addition that the chain c be fully β-reduced.

This can be illustrated with our simple example sentence *Mary walks*. A word described by the lexical entry for *Mary* stands in the relation `logical-form` with the chains described by $m^{\#}$, or with the value-raised form of m, i.e., $(\lambda P.P_{@}(\lambda @.m))^{\#}$, etc. Similarly, the relation holds between a word described by the lexical entry for *walks* and some chains described by $(\lambda x.\mathsf{walk}'_{@}(x_{@}))^{\#}$, its argument-raised form $(\lambda X.X_{@}(\lambda @\lambda x.\mathsf{walk}'_{@}(x_{@})))^{\#}$, etc. According to the second case, a phrase with the phonology *Mary walks* stands in the relation `logical-form` with a chain described by $(\mathsf{walk}'_{@}(m))^{\#}$. We can say that if a pair $\langle s, c\rangle$ is in the relation `logical-form`, then the chain c is a "virtual logical form" of the sign s.

Given these relations, for each sign in the denotation of the grammar, there will be a set of its virtual logical forms. We can, then, define the model-theoretic interpretation of a sign as the union of the extensions of the terms that correspond to the virtual logical forms of the sign. To come back to our little example, the phrase *Mary walks* has as its model-theoretic interpretation the extension of the term $\mathsf{walk}'_{@}(m)$.

The difference between such a virtual logical form and the concrete logical form as we will assume them throughout this thesis is difficult to grasp. One difference certainly lies in the fact that it is easier to impose constraints on possible logical forms of a sign if the logical form is an actual component of the sign, than if it is just a chain. On the other hand, assuming virtual logical forms instead of CONTENT values is closer to the intuitions behind S-structure Interpretivism as put forth in Lappin 1991.

The discussion in this section has revealed that the introduction of chains offers the grammar writer possibilities which would not be available otherwise. Even if we allowed quantifiers to range over the entire linguistic universe, the potential of chains is at the same time more restricted and less restricted than such an extended quantification. It is more restricted because we cannot give a chain encoding for infinite objects in general, it is less restricted because we can encode structures that do not correspond to any configuration in the denotation of the grammar.

As we have seen in this chapter, the integration of a semantic representation language into an HPSG grammar offers many opportunities to investigate the formalism of RSRL, and especially, the use of chains within grammars. So far, in actual grammars, chains have been used only to a limited extent. Mainly, this is due to the fact that we lack enough experience with this formal tool, though the need for chains could be established already based on the grammar of Pollard and Sag 1994. The novel uses of chains proposed in this thesis will hopefully give rise to more discussion on the ontological status of chains and will help to clarify the role that chains can and should play in the grammar of natural languages.

II

Idiomatic Expressions

CHAPTER 5

Introduction

In the first part of this thesis, we have established the means for building larger signs from smaller signs. We assumed that such combinations underly a small set of very simple rules. This became especially clear in the way we constructed the logical form of a phrase: by simple functional application of the logical forms of the daughters. The resulting system can be called *compositional*, in the sense given in the introduction. In Part II of this thesis, we will address a class of phenomena that is problematic for the system assumed so far, and, indeed, problematic for any system that assumes simple principles of regular combination. The problematic empirical domain is that of *irregularity*.

It has often been observed that irregularity abounds in natural language. On the one hand, irregularities are important in the discovery of rules, i.e., we can find cases of an overgeneralized application of a rule, where an exception is appropriate. On the other hand, irregularities are also a challenge for formal grammars, as it is far from clear how the great diversity of irregularities should be incorporated into a precise description of the language. In this part of the thesis, it is exactly this second question that we will be concerned with. In particular, we will show how we can in principle integrate a module of irregularity into a grammar that assumes strict principles of regular combination such as the grammar presented in Part I.

In this part of the thesis, we will confine ourselves to a hand full of VPs which exhibit a varying degree of irregularities. For lack of a beter term, we will refer to these VPs as instances of *idiomatic expressions* (IE). It would be beyond the scope of this thesis, and beyond its purpose, to try and achieve a full classification of idiomatic expressions. Instead, we will focus on some concrete examples of idiomatic expressions, which we will distinguish by certain semantic and syntactic properties. All the expressions that we will consider in this chapter are VPs, at least in their most natural form. By looking at the differences that exist among idiomatic VPs, we will see what fine-grained distinctions a formal theory of idiomatic expressions must be able to cover.

It is not *a priori* clear what should count as an idiomatic expression. We use the term *idiomatic expression* (IE) as a cover term for what might be considered an *idiom*, a *construction*, an *idiomatically combining expression*, a *collocation* or a *fixed combination*.[1] In all these cases, it is assumed that there is something “irregular” to these expressions. Instead of providing a definition of the term, we will give examples that will be considered here and explain in what respect these VPs exhibit irregular behavior. The idiomatic expressions that will be treated in this chapter are listed in (301).

(301) a. trip the light fantastic (‘dance (nimbly)’)
b. kick the bucket (‘die’)
c. spill the beans (‘reveal information’)
d. make headway (‘make progress’)
e. pull the strings (‘use connections’)
f. make a decision (‘decide’)

[1]See van der Wouden 1997 for an overview over the terminology and its problems.

As a *tertium comparationes*, we will consider regular freely combined VPs such as the one in (302).

(302) Peter [$_{VP}$ read a book].

In our interpretation of the data, we will distinguish two kinds of irregularity: an *internal* and an *external* irregularity. A given linguistic entity is internally irregular, if it is not built or interpreted according to the ordinary rules of the language. The expression *by and large* is a prototypical example of such an internally irregular linguistic entity, as its syntactic shape does not occur elsewhere in English. As we will see, only the first two IEs in the list in (301) (*trip the light fantastic*, *kick the bucket*) will be internally irregular in this sense.

The other type of irregularity has to do with the question of what constitutes a possible linguistic context for a given expression. Outside the domain of IEs, such context sensitivity is widely acknowledged. To consider a prominent example, an anaphor can only felicitously occur in a sentence if it is bound, i.e., if there is another NP that stands in the right syntactic constellation to the anaphor (be this c-command or (local) o-command) and that is co-indexed with the anaphor. This externally irregular behavior of anaphora can be illustrated with the following examples.

(303) a. John$_i$ shaves himself$_i$.
b. * John$_i$ shaves himself$_j$.

(304) a. Mary wants John$_i$ to shave himself$_i$.
b. * Mary wants himself$_i$ to shave John$_i$.

In (303b) the anaphor *himself* is not co-indexed with any other NP in the sentence, therefore, the context requirements of the anaphor are not satisfied, which yields ungrammaticality. In both examples in (304), the NP *John* is co-indexed with the anaphor. It is, however, only in (304a) that *John* is in the required syntactic constellation. We can call this the external irregularity of the anaphor, because, leaving aside the binding requirement, the anaphor can be found in any linguistic context that is appropriate for an NP.

We will show that a similar point can be made for the parts that constitute IEs such as *spill the beans*, *make headway*, *pull strings* and *make a decision*. We will show that these IEs fulfill all the criteria of internal regularity. Because of this, we can give an analysis of these IEs in terms of a regular combination of their parts. What makes them idiomatic expressions is that these parts, while combining in a standard way, can only occur in a constellation which also contains their IE-mates.

To give a concrete example which will be elaborated on in more detail below, consider the IE *spill the beans*. We will assume that in this combination, the word *beans* occurs in a meaning similar to that of *information* or *secret*. The word *spill*, as used in the IE, has the meaning of *uncover/reveal*. Given this meaning assignment to the two major parts of the IE, we can interpret the VP *spill the beans* in a regular, i.e. compositional, manner. We must, however, prevent the noun *beans* from occurring in this meaning outside the IE, i.e., we must impose context restrictions on the occurrence of the word *beans* in the meaning of *information*. Similar context restrictions must be made for the verb *spill* under the given meaning.

The interpretation of the data and the resulting analysis stands in the tradition of the work on regularity within idioms such as Fraser 1970 on the syntactic flexibility of certain idioms and Ernst 1981, Wasow et al. 1983 and Nunberg et al. 1994, which emphasizes the semantic flexibility of some idioms. Nonetheless the approach is novel in that it is the first to draw an explicit borderline between internal and external aspects of irregularity and that it integrates these notions into the architecture of a fully formalized grammatical framework such as HPSG.

In Chapter 6, we will provide data for some IEs and evaluate them according to a set of criteria of regularity. As a result, we will substantiate the claim that of the above mentioned IEs, only the first two are internally irregular, whereas the others are constructed in a regular way. They are still IEs in the sense that they exhibit some irregularity, but this irregularity is not to be found in the way the parts of the IE are combined, but in linguistic contexts in which the parts of the IE can occur. Chapter 7 presents the analyses made in other frameworks of formal grammar. We will discuss the approach of GPSG (Gazdar et al., 1985) and TAG (Abeillé, 1995), as well as the HPSG proposal made in Riehemann 1997.These proposals can be considered the main sources that were used to develop the present HPSG analysis. In Chapter 8, we will provide our own analysis that is based on the distinction between internal and external irregularity. In Chapter 9, the results of this part will be summarized.

CHAPTER 6

The Data

6.1. Criteria of Regularity

In this section, we use some tests to distinguish between several sorts of IEs. Before we apply them to what we consider to be IEs, we first show how regular VPs as *read a book* in (302) behave with respect to them. We distinguish semantic and syntactic aspects of an expression. Let us, first, consider the semantic aspects. There, we can give four characteristics of regularity.

1. Every element in the VP can be attributed some meaning with which it occurs also outside the particular combination under consideration.
2. The meaning of the entire VP is arrived at by combining the meanings of its parts in a regular way.
3. Parts of the VP can be semantically modified.
4. In a V-NP combination, it is in principle possible to have a pronoun referring to the NP.

We can illustrate these criteria with the VP *read a book*. We can assign the elements in the VP some regular meaning, such as the term $\lambda y \lambda x.\mathsf{read}'_{@}(x_{@}, y_{@})$ for the word *read*, an existential quantifier to the indefinite article and the term $\lambda x.\mathsf{book}'_{@}(x_{@})$ to the word *book*. These words occur with the same meaning in other combinations, such as:

(305) a. Peter *read* the newspaper.
b. Peter wrote *a* paper.
c. Peter bought this *book*.

We apply this criterion, however, in a stronger way. For a VP to satisfy this criterion means that all its parts can freely occur in other combinations, preserving their meaning. Thus, the verb *read* appears in its meaning read' not only with the NP *a book* as in (302) or the NP *the newspaper* as in (305a), but it combines with anything that can be read.

Let us turn to the second criterion, which is not fully independent of the first: assigning parts of a VP some meaning only makes sense if this assignment tells us something about the meaning of the entire VP, i.e., the meaning assignment to parts happens under the assumption that the overall meaning of the VP is arrived at compositionally.

In the case of the VP in (302), there is a compositional derivation of the meaning of the entire VP. Without going into details of combinatorial semantics, which has been the topic of Chapter 4, we can say that all common semantic frameworks such as the PTQ framework of Montague 1974b or the framework sketched for HPSG in Pollard and Sag 1994 will treat the derivation of the meaning of this VP as an instance of regular combinatorial (or compositional) semantics. In this thesis, we assume a semantic representation, i.e., a logical form for the sentence in (302) of the kind in (306).

(306) $\exists x[\mathsf{book}'_{@}(x_{@}) \wedge \mathsf{read}'_{@}(p, x_{@})]$

Clearly, the logical form in (306) contains all the meaning components contributed by all the elements in the sentence and only these, and combines them in a predictable way. It is in this sense, that the second semantic criterion is met.

The third criterion is also related to compositionality. It concerns the question how much the semantics of a VP is open to internal modification. For illustration, consider the sentences in (307).

(307) a. Peter read a recent book.
b. Peter recently read a book.

Clearly, the two sentences are not equivalent: in (a), the modifier establishes the novelty of the book, in (b) it refers to the time when Peter performed the act of reading a book.

In (307a) there is syntactic material added to the direct object, an adjective (a) which leads to a further determination of the direct object. When a syntactically modified constituent is also semantically modified, we speak of *internal modification*. Internal modification is the rule when it comes to regular combinations, as would be expected under the assumption of a compositional semantics. The notion internal modification contrasts with *external modification* which is sometimes attested even with regular combinations.[1]

(308) a. An occasional sailor came into the bar.
= Occasionally, a sailor came into the bar. (Nunberg et al., 1994, fn. 15)

Apart from a small class of examples of external modification in freely combined VPs as the one in (308), a modifier syntactically adjoins to the element whose semantics it modifies.

Our fourth semantic criterion can be illustrated with an example such as (309).

(309) Peter read [a book]$_i$. It$_i$ was very interesting.

The pronoun *it* in the second clause can refer to the NP *a book* in the preceding sentence. The way we stated the criterion, we carefully added "in principle", because independent factors might block the possibility of anaphoric reference. A salient case of these is if we negate the first sentence in (309). In that case, illustrated in (310), no binding across the sentence is possible, as is well documented in the literature on dynamic binding (Kamp, 1981; Heim, 1982).

(310) Peter didn't read [a book]$_i$. * It$_i$ was very interesting.

Another case which is relevant in the context of idiomatic expressions arises when the NP is used non-referentially. Sentence (302) has a reading in which the NP is not used to refer to a particular book but in which the entire VP expresses that Peter is doing some "book-reading". Under this interpretation, pronominal reference is excluded as well. As we will see, for the direct object in IEs, a non-referential reading is often the only available reading. Therefore, we consider it a sign of regularity that the NP can in principle also have a referential reading, i.e., the criterion is met if pronominal reference is possible at least in some reading of the VP.

Taken together, our semantic criteria give us a good characterization of the semantic properties of a freely combined VP, and, as we will soon see, provides us with criteria to differentiate between several classes of IEs.

In addition to the four semantic criteria given above, we will employ a series of syntactic criteria as well. These are listed below:[2]

[1]The terminology *internal* and *external* modification was introduced by Ernst 1981. In his paper, however, Ernst applies these terms only to idiomatic expressions.

[2]In this thesis, we are only concerned with IEs of a certain shape, i.e., with VPs that are of the form V NP. For other IEs, different criteria of syntactic regularity are needed.

1. Every element in the VP occurs in the same form in some other combination.
2. Syntactically, the VP is of a regularly built shape.
3. If it is a V-NP combination, the direct object can be modified syntactically.
4. If it is a V-NP combination, it can be passivized, (and further raised).
5. If it is a V-NP combination, the direct object can be topicalized.
6. If it is a V-NP combination, the direct object can have the shape of a relative pronoun.

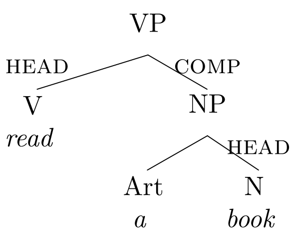
This is quite a long list of criteria, which can be ordered in two groups. The first three criteria relate to the internal structure of the VP as such. These are the questions (i) whether the words used in the VP have an independent life as items of the language or not, (ii) whether the internal structure of the VP is regular and, (iii) whether it can be slightly extended internally by adding a modifier. The next three of the criteria do some major changes to the VP under consideration, i.e., they characterize the VP's ability to undergo processes that have been characterized as transformations or movement in Transformational Grammar or GB respectively. Within these criteria, there are some that relate to A-movement, such as passivization and raising, others relate to non-A-movement such as topicalization and relativization.

Just as we did with the semantic criteria, we will illustrate the syntactic criteria with the freely combined VP *read a book* to see that free combinations have all the properties listed above. Consider, again sentence (302). All the words in this sentence appear independently of each other in precisely the form they have in sentence (302).

(311) a. (= (302)) Peter read a book.
b. Peter *read* some journal.
c. Peter met *a* friend on the bus.
d. Peter took the green *book* from the shelf.

The sentences in (311) illustrate that all elements of the VP under consideration exist independently in English. Albeit similar, our first syntactic criterion differs from our first semantic criterion in that it only looks at the form of the words, not at their meaning. So, homonyms will still count as one thing with respect to the criterion. For illustration, consider the word *band* which has two basic meanings: a strip, or a group of some kind. As we are only looking at the syntactic properties of the word in the first syntactic criterion, the homonyms count as one: they are both singular nouns. The usefulness of this criterion will become obvious, when we want to distinguish IEs which contain words which are not present in the language outside the IE (example (301d)) form those which contain "normal words" but with either no identifiable proper meaning (example (301b)) or with a meaning that cannot be attributed to this word outside the IE (example (301c)).

The second syntactic criterion concerns the internal syntactic structure of the VP. For the sake of concreteness, let us assume the following structure of the VP *read a book*:

(312)


The same syntactic structure can be found with other VPs, such as those listed in (313).

(313) a. call the police
b. drive a car
c. rejected my paper

The VPs in (314), even though regular according to our criteria as well, have a different syntactic structure.

(314) a. gave Mary a kiss b. run the entire day
c. read an interesting book d. read it.

In (a), there is a ditransitive VP, i.e., there would be one more complement NP in the structure. In (b), the tree would actually be similar to the one given in (312). Still the NP *the entire day* is an adjunct rather than a complement. In (c), the only difference concerns the internal structure of the NP. Still, there is an adjunction which is not present in the original structure. Finally, in (d), the NP consists of a single word, a personal pronoun in the example. Thus, it is syntactically less complex than the NP in (312). Therefore, the syntactic structure of the VPs in (314) is distinct from that in (312).

Let us now turn to the third and last of the "non-transformational" syntactic criteria. This criterion is related to the third semantic criterion. On the syntactic side, we are only interested in the question whether the structure of the NP in V-NP combinations can be slightly extended. This is clearly the case for a VP such as *read a book*, as is illustrated in (315).

(315) a. Peter read an interesting book.
b. Peter read a book from the library.

In these examples, the adjuncts to the direct object can be of different syntactic shape, an adjective in (a), or a PP in (b).

The second group of syntactic criteria concerns the ability of a direct object to undergo movement. For these criteria, we must require that the original meaning of the VP be preserved. This is not a problem in the case of freely combined VPs, but, as we will see in (343a), an idiomatic interpretation may be lost under passivization or topicalization. There is an additional problem with the way we have formulated the remaining syntactic criteria. We do not want to suggest a transformational analysis of the constructions, but we do find it useful to see whether what appears with a certain meaning as a VP in an active clause can also appear with the same meaning in a passive clause, or under topicalization.

Let us consider the different syntactic constellations in turn. The VP *read a book* can undergo passivization.

(316) A book was read by Peter.

Once the NP *a book* appears as a subject, it can be further raised, be it by subject-to-subject raising (a) or by subject-to-object raising (b).

(317) a. A book seems to be read by Peter (whenever he sits in the bus).
b. Mary expects a book to be read in every British school.

The last two syntactic criteria concern the question whether the direct object in a V-NP combination can participate in an unbounded dependency or not. In the list of criteria, we have elected topicalization as an instance of unbounded dependency. We did not chose the possibility to form a wh-question as this would have required some change in the shape of the NP, such as changing from *a book* to *which book*. As illustrated in (318), the NP *a book* can be topicalized.

(318) A book, Peter read on Tuesday.

Another instance of unbounded dependency is the formation of a relative clause. There are, in principle two kinds of relative clauses possible: the VP under consideration is in the matrix clause and the relative clause modifies the NP, or the VP appears in the relative clause, with the NP taking the form of a fronted relative pronoun. The two possibilities are shown in (319).

(319) a. Peter read a book [which he liked a lot].
b. Peter forgot the title of a book [which he read recently].

Of these two options for relative clause formation, we are mostly interested in the second, because there the NP does not occur explicitly as a filler to something extracted from the VP, as it does in topicalization. Instead, the relative pronoun functions as a filler and is related to the explicit NP by some other means (index identity in Pollard and Sag 1994).

This concludes our traversal of the criteria of regularity used in this thesis. We have shown that the VP *read a book* is regular with respect to all of these criteria. In Table 6.1 on page 267, we have compiled the results of the application of these criteria to some idiomatic expressions and to the regular combination *read a book*. As can be seen in the first row, the regular combination passes all the tests.

6.2. Applying the Criteria to some IEs

After this overview of the semantic and syntactic criteria of regularity, we will now present a number of less regular VPs. As mentioned above, we will use the cover term idiomatic expression (IE) for every syntactically complex combination which exhibits some irregularity. In the present work we will call a VP, and in particular a V-NP combination an IE, if it fails to satisfy at least one of the criteria above.

In the introduction to this chapter, we listed the following VPs. We will show how they behave with respect to our criteria. Table 6.1 summarizes the results of these tests.

(320) a. trip the light fantastic ('dance (nimbly)')
b. kick the bucket ('die')
c. spill the beans ('reveal information')
d. make headway ('make progress')
e. pull strings ('use connections')
f. make a decision ('decide')

trip the light fantastic. The idiomatic expression *trip the light fantastic* (*dance*) can be illustrated by the following example.

(321) Let's go out tonight and trip the light fantastic. (Ammer, 1997)

According to Ammer 1997, the IE goes back to a poem by John Milton, *L'Allegro* (1632):

Come and trip it as ye go,
On the light fantastick toe.

In this quote, the verb *trip* and the combination *the light fantastic* are separate from each other. In particular, *the light fantastic* is part of the NP *the light fantastic toe*, and this NP is not the direct object of *trip* (which is *it*), but rather part of an adjunct PP. The expression has appeared in its present form in a song by James W. Blake, *The Sidewalks of New York* (1894):

East Side, West Side,
All around the town,
The tots sang 'Ring-a-Rosie',
'London Bridge is Falling Down!

Boys and girls together,
Me and Mamie O'Rorke,
Tripped the light fantastic
On The Sidewalks Of New York.

If we apply the first semantic criterion, we can assign the verb *trip* the meaning *move with light steps*, which occurs independent of the particular IE.[3]

(322) The young lovers were tripping hand in hand through a meadow filled with wild flowers.

Matters are different when it comes to *the light fantastic*. Here, the most plausible assignment of meaning would be to say that the entire combination means *a/some dance*. This interpretation is, however problematic, as *a/some dance* cannot be used with *trip* (outside poetry).

(323) a. They tripped the light fantastic.
b. ?* They tripped some/a dance.

Finally, as the combination *the light fantastic* does not occur in other contexts at all, it cannot replace the NP *some/a dance* in other contexts either (see (324)).

(324) a. They danced some dance/ *the light fantastic.
b. We really should practice some dance/ *the light fantastic for tonight's party.
c. In some dance/ *the light fantastic, the partners are not supposed to look at each other.

Thus, even though we can attribute to parts of the VP some meaning which participates in the meaning of the whole, at least for *the light fantastic* this meaning does not occur outside the IE. A further complication arises from the fact that, while we might assign the combination *the light fantastic* a meaning, we cannot distribute this meaning in any sensible way further over the words that constitute this part of the IE. Thus, the first semantic criterion is not met by the IE.

With the meaning assignment proposed above, we can construct the meaning of the VP from that of the verb and that of the rest of the combination. However, because of the impossibility to give a meaning assignment to all words in the IE, the compositionality criterion is not fully met.

The IE also fails on the third criterion. The only adjectival modifier that our informants accepted was *proverbial*, but even with this modifier, the grammaticality of the sentence is judged quite low. With the modifier *proverbial*, the external and internal modification yield synonymous interpretations.

(325) a. ?* Let's go out tonight and trip the proverbial light fantastic.
b. ?? Let's go out tonight and proverbially trip the light fantastic.

Other adjectival modifiers were judged impossible.

(326) a. * Let's go out tonight and trip the nimble light fantastic.
Let's go out tonight and nimbly trip the light fantastic.
b. * They tripped the gay light fantastic.
They tripped the light fantastic with gaiety.

We conclude that there is no instance of real internal modification with this IE, i.e., if an adjectival modifier is possible at all, it is always interpreted semantically as modifying the entire VP. Therefore, the IE is not regular according to our third semantic criterion either.

To apply the fourth semantic criterion, we must construct a sequence of sentences which in principle allows a pronominal reference to the direct object. Example (327a) gives such a context using the NP *a waltz* instead of *the light fantastic*. If we use the IE in its full form, as in (327b), such a pronominal reference is impossible.

[3] I am grateful to Jesse Tseng for this nice example and for references on the origin of this expression.

(327) a. Mary danced [a waltz]$_i$. It$_i$ was the “Kaiserwalzer”.
b. Mary tripped [the light fantastic]$_i$. * It$_i$ was the “Kaiserwalzer”.

It should, however, be noted that the impossibility of pronominal reference might be reduced to independent factors. In its most natural reading, the NP *the waltz* in (328) is interpreted non-referentially. In this use, pronominal reference is impossible. In contrast to example (327b), sentence (328) does have a reading with pronominal reference.

(328) Mary danced [the waltz]$_i$. ?? It$_i$ was the “Kaiserwalzer”.

To conclude, pronominal reference is not possible to *the light fantastic*, thus, the IE fails on the fourth semantic criterion just as it did on the other three.

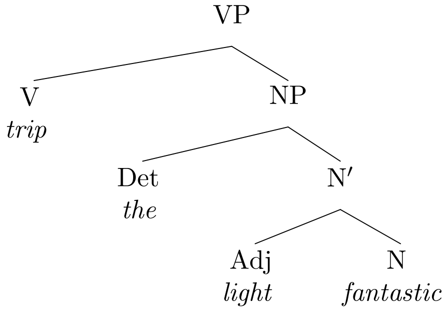
Let us, next, consider the six syntactic criteria given above. The IE does, in fact satisfy the first syntactic criterion, as all of its words do occur outside the IE, even though not necessarily with the same meaning. In (329) one such independent occurrence is given for each word in the IE. In the case of *light* and *fantastic*, the syntactic category in which they occur in the IE is not obvious. For this reason, we give an example for each of these words as adjectives and nouns. In (329c) and (329e), the words *light* and *fantastic* occur as adjectives, in (d) and (f), they are used as nouns.

(329) a. Mary *tripped* to New York last weekend.
b. John watched *the* movie.
c. Peter was looking for some *light* refreshment.
d. James turned off the *light*.
e. Everyone considers her a *fantastic* singer.
f. Peter prefers the *fantastic* to the real.

In contrast to all the semantic criteria, the IE meets the first syntactic criterion.

Matters are less clear when it comes to the second syntactic criterion, i.e., to the question whether the IE has a syntactically regular shape. In Wasow et al. 1983 (p. 104), *trip the light fantastic* is listed among expressions as *by and large* and *Believe you me!*, which are not built according to the ordinary rules of English syntax.[4] The fact that we were not sure about the syntactic category of *fantastic* in the IE suggests that Wasow et al. 1983 are indeed right in their claim that the IE is syntactically irregularly built.

It could, however, be argued that the syntactic shape of the IE is as indicated in (330). In this structure, the IE is treated as a normal V-NP combination.

(330)


Even though this structure would be regular, it was not in line with the intuitions of our informants. According to them, *light* is used as a noun in the IE, modified by the adjective *fantastic*. There is a small number of constructions in English that would allow such a post-nominal modification. The first, and intuitively most plausible is exemplified

[4]The expression *Believe you me!* shows that semantic regularity does not necessarily require syntactic regularity, i.e., this expression satisfies the first and, possibly also the second semantic criterion, but, still, is not built in a syntactically regular way.

in (331a). This construction is, however limited to a certain class of nouns in English, such as *something*, *nothing*, *everyone*, etc.

(331) a. She heard something interesting last night.
b. * She read the book interesting.

(332) a. She hammered the metal flat.
b. She ate the meat raw.

The second construction is a resultative construction which contains a small clause. Also the third example, the so-called depictive construction, might be interpreted as combining the noun and the adjective in a small clause. Even though my informants assigned the words in the IE syntactic categories that would match with those in the examples in (331a) and (332), none of these constructions seemed them applicable to the IE.[5]

If we assume either the structure in (330) or a structure parallel to the examples in (331), then the IE meets the second syntactic criterion, if we follow Wasow et al. 1983 and the intuitions of our informants, it does not. It is for this reason that we have put a question mark in the relevant slot in the Table in figure 6.1 (page 267).

In example (325), we have seen that the adjective *proverbial* seems to be the only adjectival modifier that is allowed inside the IE. The example with an adjectival modifier is repeated in (333)

(333) ?* Let's go out tonight and trip the proverbial light fantastic.

As the modification is relatively bad, even with this particular adjective, we are inclined to say that the IE does not allow a VP internal modifier syntactically. Again, we must put a question mark in Table 6.1.

The first three syntactic criteria gave some unclear results, which pointed to a non-regular syntactic behavior of the IE. This will be confirmed by the second group of syntactic criteria. In order to apply these criteria, we must assume that the combination *the light fantastic* is an NP, even though we are not forced to make a commitment to a particular internal structure for this NP. We continue the discussion on the assumption that the IE is a V-NP combination.

It is impossible to form a passive with the present IE:[6]

(334) * The light fantastic was tripped in this hall from time to time.

The meaning of sentence (334) could be something like *There was some dancing in this hall from time to time*. Still, the sentence is ungrammatical. Embedding the passive inside a raising predicate does not yield a grammatical sentence either.

(335) a. * The light fantastic seems to be tripped in this hall from time to time.
b. * Mary expects the light fantastic to be tripped in this hall from time to time.

[5]Gazdar et al. 1985 (p. 244, fn. 33) give sentence (i), not (ii) as an example for a putative passive.
(i) * The light was tripped fantastic.
(ii) * The light fantastic was tripped.
This choice of example suggests that, to these authors as well, (330) is not the correct syntactic structure for the IE.

[6]Whichever of the syntactic structures in (331a) or (332) we assume, passive and raising is possible in the case of free combinations, but excluded for the IE:
(i) The metal was hammered entirely flat.
(ii) The meat is often eaten raw in this area.
(iii) * The light was tripped fantastic. (Gazdar et al., 1985, p. 244, fn. 33)

In addition to the ungrammaticality of passive and raising, it is also impossible to topicalize *the light fantastic*. Example (336) shows that the IE fails on the fifth of our syntactic criteria.[7]

(336) * The light fantastic, she has always loved to trip.

With regards to the last criterion, the relativization criterion, it is already problematic to construct examples, because, as we have seen in the discussion of the first semantic criterion, the combination *the light fantastic* in the meaning *some dance* does not occur independently of the verb *trip*. But, even if the IE occurs in the matrix clause and in the relative clause, the resulting sentence is ungrammatical.

(337) a. * She tripped the light fantastic which she has learned in school.
b. She danced a dance which she has learned in school.

(338) a. * She tripped the light fantastic which her father had tripped when he was younger.
b. She danced a dance which her father had danced when he was younger.

The ungrammaticality of the (a) examples clearly indicates that it is impossible to form a relative cause that depends on *the light fantastic*.

If we contrast our findings about *trip the light fantastic* with those of *read a book*, we see that the former is extremely irregular. In particular, it is irregular with respect to its semantics and to its ability to appear in syntactic constructions other than its active VP form. In that form, however, its internal structure is not entirely clear.

kick the bucket. The second example in the list of VP-IEs given in (320) is the VP *kick the bucket*. In (339a), there is an example sentence with this IE.

(339) a. He said if he *kicked the bucket* his deputy could run the business as well as he had. (McCaig and Manser, 1986)
b. All of my goldfish kicked the bucket while we were on vacation. (Ammer, 1997)

The IE *kick the bucket* has the meaning *die*. Even though all the words that appear in the IE can appear independently, none of them does so in any meaning that could be considered part of the meaning of the IE.[8] Thus, it is not possible to assign the parts of the IE a meaning which will both contribute to the overall meaning of the VP and will occur independently. The IE *kick the bucket* does not meet the first semantic criterion.

As we cannot distribute the meaning of the IE over its components, it follows automatically that the overall meaning cannot be computed by regular means from that of the components. Therefore, the second semantic criterion fails as well.

As far as the third criterion is concerned, we find the following examples for syntactically internal modification. In all instances, however, the parallel sentence with syntactically external modification is equivalent.

(340) a. Pat kicked the proverbial bucket.
= Pat proverbially kicked the bucket. (Wasow et al., 1983, p. 110f.)

---

[7]Again, a different syntactic structure would not predict this ungrammaticality of topicalization, as it is possible out of a small clause.

(i) The metal she hammered entirely flat, but the plastic she didn't even touch.
(ii) The meat people in this area often eat raw, but they do cook the vegetables.
(iii) * The light she has always loved to trip fantastic.

[8] According to Ammer 1997, the word *bucket* refers to some kind of beam form which pigs were suspended by their heels after being slaughtered.

b. With that dumb remark at the party last night, I really kicked the social bucket.
= Socially, I kicked the bucket. (Ernst, 1981, p. 51)

We can conclude from these examples that it is not possible to modify parts of the IE semantically, even though a syntactic modification is possible. Thus, the third semantic criterion is not met by the IE *kick the bucket*.

Checking the fourth semantic criterion, we observe that pronominal reference is not possible to the NP-part of the IE either, as shown in example (341). This means that this criterion fails for the present IE.

(341) * Pat kicked [the bucket]$_i$ and Harry kicked it$_i$, too.

As noted in the table in Table 6.1, the IE *kick the bucket* fails to satisfy any of our semantic criteria of regularity. It does, in fact, share this behavior with the previously discussed IE *trip the light fantastic*.

We will, next, consider the syntactic criteria. As noted above, all words that occur in the IE occur independently in other contexts, where they have a clear meaning. In (342), we list an occurrence of each of the words that occur in the IE.

(342) a. Peter *kicked* a ball.
b. Mary read *the* book.
c. John carried the heavy *bucket*

Not only does the IE *kick the bucket* consist of ordinary words, it also combines them in an ordinary way. The syntactic structure of the IE is exactly as that of the freely combined VP *read a book* shown in (312). This is the reason why there is a literal meaning to the VP, in addition to the idiomatic reading *die*. Thus, the second syntactic criterion is met as well.

In the discussion of the semantic properties, we have already seen that it is syntactically possible to have an adjective as part of the NP in the IE. The relevant examples were given in (340). It follows from these examples that the third syntactic criterion also applies.

So far, the IE *kick the bucket* patterns with *trip the light fantastic* with respect to the semantic criteria, but it patterns with the free combination *read a book* with respect to the first three syntactic criteria. Our further criteria will confirm the similarities of the two IEs rather than that of the *kick the bucket* with a free combination: None of the “transformational” criteria is met by *kick the bucket*.

Let us consider the relevant sentences, where a ‘*’ indicates ungrammaticality under the idiomatic reading.

(343) a. * The bucket was kicked by Pat.
b. * The (social) bucket, Pat really kicked (with his dumb remark at the party last night).
c. * The old lady kicked the bucket that the murderer had planned for her.
d. * The old lady wasn’t aware of the bucket that she would soon kick.

As indicated in (343a), passivization is not possible with the idiomatic reading, nor is topicalization (343b). Thus, our fourth and fifth syntactic criteria fail to hold for the IE. In addition, the NP *the bucket* is not a candidate for an attachment of a relative clause. The sentences are ungrammatical independently of whether the verb *kick* is a clause-mate of *the bucket* (343c), or whether it appears in the relative clause as in (343d). This shows that the last syntactic criterion is not met either.

spill the beans. As a third IE, we will consider the VP *spill the beans* (*reveal information/a secret*). In (344) we give some example sentences for this IE.

(344) a. Pat spilled the beans. (Wasow et al., 1983, p 102)
b. That's just like Marcia to spill the beans and tell Peter we're having a party for him! (McCaig and Manser, 1986)

The usual paraphrases of this IE are all of a syntactic shape which is very close to that of the VP. Thus, we can give the following paraphrases for the sentence in (344a):

(345) a. Pat revealed a secret.
b. Pat divulged information.

Therefore, we can assign the verb *spill* the meaning *reveal* or *divulge*. In slang usage the verb *spill* occurs in that meaning independent of the NP *the beans*.

(346) a. She spilled the story to the media.
b. He spilled all of the facts to the jury.

Under this interpretation, *the beans* must be assigned the meaning *a secret*. The NP *the beans* does, however, not occur with this meaning outside the IE under consideration. Again, the asterisk in the examples in (347) means ungrammaticality under the interpretation *a secret* for the NP *spill the beans*.

(347) a. She divulged the story/ *the beans to the media.
b. He divulged all the facts/ *the beans to the jury.

As the NP *the beans* does not occur with the meaning *a secret* in environments other than the IE under consideration, the IE does not meet the first semantic criterion.

When it comes to compositionality, however, we find ourselves in the position to attribute the elements of the IE a meaning all of which will be combined to form the meaning of the overall IE: The verb *spill* is used in its slang meaning *divulge*, the definite article is used in a generic meaning and, finally *beans* is used in the meaning of *secret*. From these meaning components, we can regularly derive the meaning of the IE. For this reason, the second semantic criterion is met.

When we turn to the next semantic criteria, there is a great amount of divergence in judgments among speakers and in the literature. On the one side, there is Schenk 1995 according to whom, the IE *spill the beans* does not allow modification. On the other hand, the IE is claimed to pattern with other expressions that allow for modification in Wasow et al. 1983. The behavior of this particular IE seemed to be not fully clear to the speakers that we have consulted either. In this section, we present the data as given to us by two relatively congruent informants. Being aware of the existing differences in judgments, we will point to other judgment patterns found in the literature in footnotes and come back to the picture that emerges from these alternative judgments in Section 6.3.

The IE *spill the beans* allows for semantic modification of the noun. Thus, syntactically internal and syntactically external modification does not lead to the same interpretation.[9]

(348) a. Pat spilled the inadvertent beans.
$\neq$ Pat inadvertently spilled the beans.
b. Pat spilled the sordid beans.
$\neq$ Pat sordidly spilled the beans.

[9] The data given in the main text agree with the empirical expectations of Wasow et al. 1983, Gazdar et al. 1985, Nunberg et al. 1994 and Abeillé 1995. According to Schenk 1995 (p. 262), however, the IE does not allow modification. In (i), we give Schenk's grammaticality indication:
(i) * Mary spilled the well-kept beans.
Thus, speakers that share the judgments in Schenk 1995, the IE does not meet the third semantic criterion.

In addition, there are cases in which no external reading of the modifier would be possible.

(349) a. Pat spilled the well-guarded beans.
b. Pat spilled the juicy beans.

In all of the examples in (348) and (349), the adjective semantically modifies only the kind of information, not the telling thereof as a whole. Thus, the IE satisfies the third criterion of semantic regularity.

The following example illustrates that pronominal reference is possible to *the beans* in this IE. Thus, the IE is semantically regular with respect to the fourth criterion as well.

(350) I was worried that the beans might be spilled, but they weren't.
(Wasow et al., 1983, p. 112)

Such a reference is, however only possible if the pronoun appears as part of the same IE, at least semantically: in the second clause of (350), the elided verb corresponds to *spill*. If this extra condition is not met, pronominal reference is impossible, as shown in (351).[10]

(351) * When Pat spilled the beans, she thought that they would shock her parents.

In summary, the IE *spill the beans* semantically deviates from complete regularity only with respect to the first semantic criterion, i.e., the fact that the NP *the beans* does not occur in its idiomatic reading in other contexts.

When we address the syntactic criteria, we find that all the words in the IE have an independent existence in the language, satisfying the first syntactic criterion.

(352) a. Pat *spilled* some oil.
b. Mary took *the* picture from the wall.
c. The kids don't like *beans*.

The second criterion is equally met, as the VP has the same syntactic structure as the free combination *read a book* in (312). The third syntactic criterion is satisfied as well, since the examples with adjectives given in (348) show that it is in principle possible to have a modifier inside the NP without losing the idiomatic meaning of the expression.[11]

In contrast to the IEs that we have looked at so far, *spill the beans* can be passivized and raised. As the reader might have noticed, example (350), repeated as (353b) is an instance of a passive use of the IE, embedded under a modal auxiliary, i.e., a subject-to-subject raising verb.[12]

(353) a. The beans were spilled in this article.
b. I was worried that the beans might be spilled, but they weren't. (Wasow et al., 1983, p. 112)
c. The beans tend to be spilled. (Schenk, 1995, p. 260)
d. John believes the beans to be spilled. (Schenk, 1995, p. 260)

[10]With respect to pronominalization, Schenk 1995 only considers examples of the type in (351). On page 262, Schenk gives the following example, which is also judged ungrammatical by all our informants.
(i) * Alexander spilled the beans, since he did not know they were secret.
It is not clear what Schenk's judgments would be in cases such as (350).
[11]As pointed out in footnote 9, some speakers might reject modification of the noun *beans*. We do, however, lack data whether this is a syntactic or a semantic restriction. As our informants did in any case accept at least the modification by the adjective *proverbial*, the restriction seems to be more likely semantic in nature, just as it had been attested for the IE *kick the bucket*.
(i) Pat spilled the proverbial beans.
[12]All references we found agreed on this property of the IE.

When it comes to topicalization, however, the IE stops to show regular behavior. According to our informant and to Schenk 1995 (p. 259), topicalization is excluded with the NP in this IE. Thus, as indicated by the asterisk in (354), topicalization is ungrammatical under the idiomatic reading.[13]

(354) * The beans John spilled.

The last syntactic criterion involves relative clauses. In the case of *spill the beans*, no instances of relative clauses can be found. This sharply contrasts with the clear grammaticality of the parallel examples using the paraphrase *reveal the information* in (356).

As relative clauses involve both modification and extraction, again, a mismatch in judgments arises between the expectations of Schenk 1995, Wasow et al. 1983 and the judgments of our informants. According to our informants, relative clauses with the IE are considerably bad, as indicated in (355).

(355) a. * The alleged arms dealer spilled the beans [that made the party leader resign].
b. * The beans [that the alleged arms dealer spilled] made the party leader resign.
c. * The party leader resigned because of the beans that the alleged arms dealer had spilled.

(356) a. The alleged arms dealer revealed the information [that made the party leader resign].
b. The information [that the alleged arms dealer revealed] made the party leader resign.
c. The party leader resigned because of the information that the alleged arms dealer had revealed.

In the (a) sentence of (355), the entire IE is part of the matrix clause. In (b) and (c), the noun *beans* is part of the matrix, whereas the verb *spill* appears in the embedded clause. As can be seen, none of the sentences is grammatical.[14]

We conclude that the behavior of this particular IE seemed to be not fully clear either: The IE *spill the beans* behaves almost like a free combination with respect to our criteria. Following the judgments given in the main text, it has only one semantic irregularity, i.e., the fact that the noun *beans* does not occur in its idiomatic meaning outside this IE, and two syntactic irregularity, the ban on topicalization and on relative clauses.

According to the judgments given in Schenk 1995, the IE would be less regular: it would only be regular with respect to the second semantic criterion (regular combination of the semantic contributions of its parts) and to the first four syntactic criteria.

If we followed the empirical classification given in Wasow et al. 1983, we would expect the IE *spill the beans* to behave more regularly than indicated by the judgments in the main text. In fact, the IE *spill the beans* should be parallel to the judgments we give for the expression *pull strings* below. In Table 6.1 the results results are given for the three grammaticality patterns that we found for this IE.

[13]While Wasow et al. 1983 or Gazdar et al. 1985 do not explicitly discuss topicalization data with this IE, their presentation suggests that *spill the beans* belongs to those IEs that allow for topicalization in principle. While all our informants judged sentence (354) ungrammatical, there is an increased acceptance if the direct object contains more material:
(i) ?* The well-guarded beans John spilled.
Still, there seems to be a lot of context required to accept this sentence, which leads us to indicate it rather as ungrammatical.

[14]While no relative clause data for this IE are given in Wasow et al. 1983, such data should be grammatical in principle for these authors. In fact, the following sentences, while structurally parallel to (355b) are considered less ungrammatical by our informants.
(i) ?* The beans Pat spilled weren't so damaging as what Fred divulged.
(ii) ?* The beans (that) Pat spilled caused a scandal.

make headway. The next IE that we want to address is the VP *make headway* (*progress*), as exemplified in sentence (357).

(357) The government's not making headway in the battle against inflation. (McCaig and Manser, 1986)

Just as in the case of *spill the beans*, we can give a paraphrase which is syntactically very close to the IE.

(358) The government is not making progress in the battle against inflation.

This suggests that we can assign the words in the IE some meaning: *make* appears in the IE in its function to take a noun and express the action of achieving what is expressed by the noun. This use of *make* occurs with several nouns. In these uses, the verb *make* is called a *light verb* or a *support verb.*

(359) a. make progress ('progress')
b. make a difference ('differentiate')
c. make a decision ('decide')
d. make a move ('move')
e. make a speech

The way we interpret the first semantic criterion requires it to be possible to use a given part of a VP freely in its meaning in other combinations as well. We show in the following that the support verb *make* does not meet this criterion. For this purpose, we argue that at least for some support verbs, we must list the nouns that they combine with. From this, we conclude that the first semantic criterion is not satisfied.

For the verb *make* to meet the first semantic criterion, it should be able to co-occur with nearly any noun which can occur in a light verb construction. This is, however, not the case. In (360), we list some potential light verb constructions of English.

(360) a. do the laundry
b. wage war
c. commit a crime
d. take a shower

In (361), we show which of the nouns from this list can combine with the light verb *make.* An asterisk indicates that the combination with *make* is not a light verb construction.

(361) a. * make the laundry
b. make war
c. * make a crime
d. * make a shower

For some of the light verbs, it seems possible to predict the kind of nouns that they combine with on semantic grounds, i.e., we can restrict the nouns that occur as direct objects of these light verbs by the regular means of semantic restrictions. For example, one can only *wage* a war, a campaign or some other kind of fight. Similarly, in its light verb use, *commit* can only combine with something illegal.

It is an open question whether such a semantic characterization can be given for all support verbs. If we consider the nouns that combine with *make* in (359), there is no obvious semantic class that would comprise these, but not the nouns of (360) that cannot combine with the support verb use of *make.* The following considerations of support verbs in German will show that some cases, there seems to be no hope that an appropriate semantic class can be found. There are two papers within HPSG that address support verbs in German, Krenn and Erbach 1994 and Kuhn 1995. In both papers, support verbs are assumed to select for certain lexemes rather than for semantic classes. Krenn and Erbach 1994 give the following examples to suggest that in some cases nouns which might be considered synonyms are not compatible with the same support verb.[15]

(362) a. eine Entscheidung/ * einen Beschluss treffen
a decision/ a resolution meet
b. eine Entscheidung/ * einen Beschluss fällen
a decision/ a resolution fell
c. * eine Entscheidung/ einen Beschluss fassen
a decision/ a resolution grasp

The examples in (362) show that the support verbs *treffen* (*meet*), *fällen* (*fell*) and *fassen* (*grasp*) require different nouns as their complements and that, even though the nouns *Entscheidung* (*decision*) and *Beschluss* (*resolution*) are semantically similar, the combinatorial requirements of the support verbs must be able to differentiate between them.

A similar, but maybe even stronger, example is the following. The German verb *einjagen* combines only with the noun *Angst* (*fear*) or the NP *einen Schrecken* (*a scare*).[16] It forms a support verb construction in the meaning of *give somebody a fright/scare*. This is indicated in (363). We could, then, assume that *Angst* and *Schrecken* belong to one, very small, semantic class, and that the verb *einjagen* selects for this class.

(363) Peter hat mir Angst/ einen Schrecken eingejagt.
Peter has me fear/ a scare ??
'Peter gave me a scare.'

In (364) we show that only the noun *Angst* can combine with the support verb *machen* (*make*) to form a support verb construction which is synonymous to the example in (363).

(364) Peter hat mir Angst/ * einen Schrecken gemacht.
Peter has me fear/ a scare made
'Peter gave me a scare.'

If the support verb *machen* selects a semantic class which contains all nouns compatible with *machen* in a support verb construction, then this class contains *Angst* and for example *Vorschlag* (*proposal*), but not *Schrecken*, while the latter is semantically very similar to *Angst*. This is particularly strange, because we would be forced to say that *Angst* und *Schrecken* form a semantic class on independent grounds, since they are the only nouns that combine with *einjagen* (see (363))

Consequently, the data in (363) and (364) are better characterized in terms of selecting particular lexical items. The verb *einjagen* selects two lexical items, *Angst* and *Schrecken*, whereas the support verb *machen* selects a great variety of lexical items, but not *Schrecken*.

Under the assumption that at least some support verbs select single lexical items, the question arises of whether the support verbs themselves are ambiguous, i.e., whether we must assume multiple lexical entries for support verbs such as *make*. We think that there is evidence that the support verb *make* must be distinct from a main verb *make*, but we do not see a clear difference between several support verb uses of *make*. The following data illustrate the contrast between the support verb uses and the main verb uses of *make*.

(365) a. * The making of progress/headway is a precondition for success.
b. The making of cars is an interesting topic for a documentary.

---

[15]As with all German examples in this thesis, we have adapted the examples of Krenn and Erbach 1994 (pp. 379f.) to the new spelling rules.

[16]In addition to these, the *Deutsches Wörterbuch* of Jacob and Wilhelm Grimm gives older quotes with *Schau(d)er* (*shudder*) and *Zorn* (*wrath*). The combination *Schauer einjagen* might still be marginally acceptable, but to us, the combination *Zorn einjagen* seems to be impossible in today's German.

Schenk 1995 (p. 267) points out that support verbs cannot appear as nominal gerunds, i.e., as *-ing*-forms that take an *of*-PP as their syntactic complement. This is shown in (365a). As indicated in (365b), nominal gerunds are possible with the main verb use of *make*.[17] According to Zwicky and Sadock 1975, if we can find such a difference in the linguistic constructions in which an element can occur, we have some evidence to postulate ambiguity. Thus, the contrast in (365) indicates that there are at least two verbs *make* that must be distinguished in English: a main verb in the sense of *produce*, and a support verb.

It is unclear, however, whether all support verb uses can be captured with a single lexical entry. For the few support verb constructions that we consider in this study, this general problem reduces to the question of whether the verb *make* in *make headway* and in *make a decision* is licensed by the same lexical entry or by distinct lexical entries. As we will not provide a detailed semantic analysis for support verbs, we cannot not give an answer to this question. In the following, we assume that a single lexical entry is enough, but an alternative assumption that there are two lexical entries which share many properties would be equally compatible with our analysis.[18]

To conclude, the examples given above make it at least questionable whether we can express the selectional requirements of support verbs fully in terms of semantic restrictions. While there certainly exists some predictability within the distribution of support verbs, there are also completely idiosyncratic cases. In this thesis, we will simply consider the cases of *make headway* and, later in the text, *make a decision* as idiosyncratic in this sense.

The paraphrase in (357) suggests that we can assign the noun *headway* the meaning *progress*. Nonetheless, the word *headway* does not occur in English outside the present IE. Therefore, the first semantic criterion is not met for this part of the IE either.

(366) a. Everyone is in favor of progress/ *headway.
b. Her recent progress/ *headway impressed her boss.

As far as the second semantic criterion is concerned, however, we can easily compute the meaning of the VP by assuming the indicated meaning assignment. This tells us, that the IE satisfies the second semantic criterion.

Just as its paraphrase, the IE allows for semantic modification of the direct object in some cases (367a). The semantic contribution of the verb *make* is quite weak, therefore it is in most cases not clear whether there is an instance of internal or external modification, as the two interpretations would mean the same (368).

(367) a. He is making interesting headway.
$\neq$ He is making progress in an interesting way.
b. He is making interesting progress.
$\neq$ He is making progress in an interesting way.

[17] See for example Abeillé 1988 for observations on differences in the potential of extraction between the light verb constructions and other combinations. We have not included the ability to form nominal gerunds in our syntactic criteria, because we will not give an account of gerunds in this thesis.

[18] A similar case is investigated in Lüdeling 1995 for the main verb uses of German *haben* (*have*). Lüdeling formulates her analysis within the framework of two-level semantics of Bierwisch 1983, 1989. Bierwisch assumes that there are two levels of interpretation: a level of *Semantic Form* (SF) and a level of *Conceptual Structure* (CS). This distinction allows Lüdeling to assume a single SF — and therefore also a single lexical entry — for the main verb *haben* as it appears for example in (i) and (ii). At the level of CS, this general relation is made more specific depending on the context. Thus, in (i) the verb is contextually specified as expressing a possession, in (ii) it relates an individual to its physical state.

(i) Chris hat eine Villa in Südfrankreich.
Chris has a villa in the South of France

(ii) Chris hat Fieber.
Chris has a fever

If Lüdeling's approach can be applied support verbs then it is likely that we can assume a single semantic specification for the support verb uses of *make*.

(368) a. He is making considerable headway.
= He is making headway considerably.
b. He is making considerable progress.
= He is making progress considerably.

From these data we conclude that semantically internal modification is possible with the IE *make headway*, i.e., the third semantic criterion is met.

When we consider pronominal reference, we observe that it is impossible to refer to the noun *headway* by a pronoun, as evidenced by the following sentences. This contrasts with the semantically equivalent *progress* in (370).

(369) a. * Mary made considerable headway$_i$ and John made it$_i$, too.
b. * I thought that considerable headway$_i$ was made in the last 15 years, but it$_i$ wasn't made.

(370) a. Mary made considerable progress$_i$ and John made it$_i$, too.
b. I thought that considerable progress$_i$ was made in the last 15 years, but it$_i$ wasn't made.

If we look at the results for the semantic criteria, we see that the IE is semantically less regular than *spill the beans*, but more regular than *kick the bucket*.

Let us turn to the syntactic criteria. As mentioned above, the word *headway* does not occur in the English language outside the present IE. Therefore, the first syntactic criterion of regularity is not met.

Even though the word *headway* is special to the IE, we can treated it as a noun. This makes the entire IE structurally regular, i.e., it has the same structure as any of the following VPs with a bare noun direct object. Therefore it satisfies the second syntactic criterion.

(371) drink milk, read books, love children

As evidenced by the modification examples in (367), the IE allows for an adjectival modifier to be part of the direct object, which shows that the third syntactic criterion is met as well.

The IE *make headway* can also undergo passivization and raising, thus meeting the fourth syntactic criterion.

(372) a. Considerable headway was made over the last 15 years.
b. Considerable headway seems to be made in this area of research.
c. The researchers expect considerable headway to be made in the near future.

The data are less clear with topicalization. In its bear form, the noun *headway* seems not to be topicalizable. If *headway* is modified, however, the acceptability of topicalization rapidly increases.

(373) a. ?* Headway our research makes constantly.
b. That much headway only a huge research institute like ours could make in a single year.

(374) a. ?* Progress our research makes constantly.
b. That much progress only a huge research institute like ours could make in a single year.

As the same pattern arises with the VP *make progress* in (374), we interpret this as an indication that the oddness of sentence (373a) is due to independent constraints on what kinds of elements can be topicalized in English.

Relativization seems to be possible for the IE as well. This is unquestionably the case if the entire IE is within the same clause and the relative clause is merely attached to the noun *headway*, see example (375a). If the verb *make* is contained in the relative clause, as in (375b), there is a slight decline in acceptability. This decline does not arise with the noun *progress*.

(375) a. You have made considerable headway/progress that will be of tremendous use for the entire project.
b. I admire the ?headway/progress that you have made since we met last time.

If we have the IE appear in both the matrix and the relative clause, the sentence is fine:

(376) She has made headway that nobody has ever made before in this area of research.

The discussion revealed that the IE *make headway* exhibits considerably regular behavior. Still, it is not a regular VP because the support verb *make* exhibits idiosyncratic selectional properties and the word *headway* does not appear in other contexts.

pull the strings. The next IE that we want to consider exhibits even more regularity than the others. It is illustrated by the following sentence.

(377) a. Pat pulled strings to get Chris the job.
b. You can often achieve more by pulling strings rather than writing letters or trying to persuade people of your point of view. (McCaig and Manser, 1986)

The following paraphrase can be used as a rough correspondence.

(378) Pat used connections to get Chris the job.

Under the working assumption that the two sentences convey exactly the same meaning, we can attribute the meaning *use* to the verb *pull* and interpret the noun *strings* as *connections*. Using this assignment, each element in the IE contributes to the meaning of the IE in a compositional way. Nonetheless, neither the verb *pull* nor the noun *strings* freely occur in this sense elsewhere. For the verb *pull* there is just one other combination in which it takes the same meaning as in *pull strings*: the synonymous IE *pull wires*.

(379) Pat pulled wires to get Chris the job.

For all other combinations, the words *pull* and *strings* are not used in the meaning *use* and *connections* respectively. This can be illustrated with the following examples. In each case, replacing the words *use* or *connections* by *pull* or *strings* does not preserve the meaning.

(380) a. Pat used strings to get Chris the job.
b. Pat pulled connections to get Chris the job.

(381) a. Pat used the bike to get to the department.
b. $\neq$ Pat pulled the bike to get to the department.

(382) a. Pat has connections to a member of the parliament.
b. $\neq$ Pat has strings to a member of the parliament.

These examples show that our first criterion is not met. Still, under the meaning assignment characterized above, the IE can be interpreted compositionally. We conclude that the second semantic criterion is satisfied.

The IE under consideration also allows for internal modification which is non-synonymous to external modification. To see this, consider the following data.[19]

(383) a. Pat pulled official strings to help Chris.
$\neq$ Pat officially pulled strings to help Chris.
b. Pat pulled obvious strings to help Chris.
$\neq$ Pat obviously pulled strings to help Chris.

There are, of course instances of external modification with that IE as well.

(384) Pat pulled political strings to help Chris.
= In the political domain, Pat pulled strings to help Chris.

Because of the IE's potential to allow for semantically internal modification, it satisfies the third semantic criterion.

To test the fourth criterion, we can simply modify the sentence used for the IE *spill the beans* in (350).

(385) I hoped that Pat would pull the strings to help Chris, but he didn't even think of pulling them.

As the example shows, the IE allows for pronominal reference, thus also satisfying the last semantic criterion.

When we turn to the syntactic criteria, the regular properties of the IE become even more obvious. First, all the words that appear in the IE also appear independently, although with a different interpretation. Thus, the first syntactic criterion is met as well.

(386) a. Kim *pulled* a knife on the attacker.
b. Pat tied up the parcel with two *strings*.

Similarly, the IE syntactically has the same structure as the freely combined VP *read a book* in (312). Therefore, the second syntactic criterion is equally satisfied.

Turning next to modification. As the IE allows semantically internal modification and as the examples in (383) show, an adjectival modifier can appear inside the IE. Thus, the third syntactic criterion is met.

The IE can also appear in all syntactic constructions used in our criteria. As such, it can occur in passivized form, and with raising predicates.

(387) a. Those strings were pulled by Pat.
b. These strings seem to have been pulled by Pat.
c. Chris actually expected these strings to be pulled by Pat.

Topicalization is possible as well:

(388) Those strings, he wouldn't pull for you. (Nunberg et al., 1994, p. 13)

When it comes to relative clauses, both types of relative clauses are possible.

(389) a. Pat pulled the strings [that got Chris the job].
b. The strings [that Pat pulled] got Chris the job.

If we compare the behavior of the IE *pull the strings* and that of the freely combined VP *read a book*, it is obvious that *pull the strings* is entirely regular, except for the fact that the words are used in a nonstandard interpretation.

[19]In fact, the IE *pull strings* is subject to very much the same variation attested for *spill the beans* above. As the data that we got on this IE were more in line with those given in Wasow et al. 1983 than with the pattern assumed in Schenk 1995, we do not repeat Schenk's position here.

make a decision. The final IE that we want to consider in this thesis is the VP *make a decision*, as in (390).

(390) After all this discussion, you must finally make a decision.

If we run our criteria on this VP, it appears to be extremely regular: The word *decision* appears in its regular meaning, and the verb *make* has its light verb function, i.e., a relatively vacuous meaning, serving the function of creating a verbal context for the noun (*decision*, in this case). As such, both words appear in the given meaning independent of each other within the language. As we have seen, however, in the case of the IE *make headway*, the light verb use of *make* is not predictable and, thus, we must conclude that it violates the first semantic criterion.

The word *decision* clearly appears in other VPs as well, some being even synonymous to *make a decision.*

(391) a. They finally reached/took a decision.
b. I can only admire such a brave decision.

This indicates that it is only the verb *make* in this combination that exhibits irregularity.

As to combinatorial semantics, the meaning of the VP can be compositionally derived from the meanings of its parts. Thus, the second semantic criteria is satisfied.

As the meaning contribution of the verb *make* is minor, modification of the nominal part of the IE is often synonymous with modification of the entire VP. This can easily be tested by replacing *make a decision* by the simple verb *decide.*

(392) a. She made the right decision.
= She decided the right way.
b. She made a difficult decision.
= She decided with great difficulties.

Still, we think that in all these cases, the adjectival modifier can be taken to semantically modify the noun rather than the entire VP, because the Adj-N combinations can occur with the same meaning in other contexts (393).[20]

(393) a. At first sight, this seemed to be the right decision.
b. It was a difficult decision to sell the house.

Finally, pronominalization is also possible, which means that the fourth criterion is met.

(394) a. Whenever the members of the party make [a decision]$_i$, the party leader must defend it$_i$ in the parliament.
b. While Pat makes [the decisions]$_i$, Sandy just accepts them$_i$ as they are.

Turning to the syntactic criteria, we see that all the words contained in the VP occur independently as well, and that the way they combine syntactically is just as that of the freely combining VP *read a book* in (312). As seen in the modification example in (392), an adjectival modifier can occur inside the NP.

Passivization and raising are both easily possible with the IE, satisfying the fourth syntactic criterion.

(395) a. Obviously, this decision was made without asking those that will be affected by it.
b. Sometimes unpleasant decisions must be made.
c. I don't expect an important decision to be made before next Monday.

[20]Note that in the case of *headway*, we were not able to give such examples because the noun cannot appear outside the IE *make headway.*

Table 6.1. Regularity criteria and idiomatic expressions

	Semantic criteria				Syntactic criteria					
	1	2	3	4	1	2	3	4	5	6
read a book	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok
trip the light fantastic	*	*	*	*	ok	?	?	*	*	*
kick the bucket	*	*	*	*	ok	ok	ok	*	*	*
spill the beans (Schenk, 1995)	*	ok	*	*	ok	ok	ok	ok	*	*
spill the beans (main text)	*	ok	ok	ok	ok	ok	ok	ok	*	*
spill the beans (Wasow et al., 1983)	*	ok	ok	ok	ok	ok	ok	ok	ok	ok
make headway	*	ok	ok	*	*	ok	ok	ok	ok	ok
pull the strings	*	ok	ok	ok	ok	ok	ok	ok	ok	ok
make a decision	*	ok	ok	ok	ok	ok	ok	ok	ok	ok

Also, topicalization is possible, if the noun is made prominent enough to meet the constraints on topicalization.

(396) Such a decision, we will never make without asking for your approval.

Finally, relative clauses, both with the V- and the NP-part of the IE in the same clause and separated are fine.

(397) a. Mary made a decision that affected the rest of her life.
b. The decision that Mary made affected the rest of her life.

We conclude that the VP *make a decision* satisfies all the syntactic criteria and only fails on the first semantic criterion. As can be seen from Table 6.1, the result is the same that we got for the IE *pull strings*.

There is a syntactic difference between *pull strings* and *make a decision*. As illustrated in (365a), it is impossible to have a nominal gerund with the light verb use of *make*. Nominal gerunds are, however, possible with other kinds of IEs. Fraser 1970 (pp. 38ff.) explicitly mentions these gerunds to be possible with *spill the beans* and *pull strings*.

(398) a. I did not appreciate Pat's spilling of the beans to the media.
b. Pat's pulling of strings got Chris the job.
c. * I do not like his making of all decisions for our family.

6.3. Interpreting the Data

The results of our survey of the syntactic and semantic properties of some English VPs are collected in Table 6.1. On the horizontal axes, we have put the semantic and syntactic criteria as we have presented them above. The vertical axes lists the VPs whose regularity we have investigated. For comparison, the behavior of the regularly combined VP *read a book* is included as well. An "ok" in a slot of the table marks that the VP under consideration satisfies the indicated criterion. As can be seen, the freely combined VP satisfies all the regularity criteria. The other rows contain some "*" symbols which indicate that a certain criterion is not met. In the case of *trip the light fantastic*, we also find a "?" symbol twice. The occurrence of this symbol shows that it might be possible that the criterion is met, but it is not entirely clear. In the discussion of this IE, the problematic status of its internal structure was shown. In Table 6.1, we have not only indicated the results for the IE *spill the beans* as used in the main text, but also the results for speakers that follow the judgments in Schenk 1995, and the reconstructed judgments of Wasow et al. 1983.

Now that we have gathered some differences between the VPs listed in (320), we can try to interpret these data from a theory neutral perspective. This interpretation of the data will prepare the reader for the particular analysis chosen in Chapter 8.

In this section, we will try to interpret the data collected above from a perspective which is theory neutral as far as possible. Still, there are certain basic assumptions that we are making. They should, however, be rather uncontroversial. Our interpretation of the data will be particularly concerned with the distinction between what we have called *internal* and *external* irregularity in the introduction to this part of the thesis.

In order to be able to work with these notions of irregularity, we must give at least a tentative definition. In the introduction, we took the adverbial *by and large* as an example of an internally irregular phrase. Our criterion for internal regularity of a complex expression is twofold: Syntactically, the expression must have a structure which is regularly attested in the language. Semantically, there must be a meaning assignment to the parts of the expression such that the meaning of the entire expression can be computed from this meaning assignment according to general principles of combinatorial semantics, such as those illustrated in Part I. These two criteria correspond to the second syntactic and the second semantic criterion of Section 6.1 respectively. The way the semantic part of internal regularity is formulated, it is clear that an expression cannot be syntactically irregular but semantically regular. This is a consequence of the standard notion of compositionality.

In Table 6.1, there are only two expressions that fail on the second semantic criterion, *trip the light fantastic* and *kick the bucket*. For this reason, they are the only IEs from our list in (320) that we consider to be internally irregular.

It is much harder to give a clear characterization of external irregularity, and, to some extend, this characterization will depend on the particular theoretic perspective and the amount of theoretical insight that is available in a particular empirical domain. Still, for any theory and at any level of understanding of the data, we will find instances of external irregularity. For the purpose of this thesis, we will offer a general characterization of external irregularity and, then, show which of the regularity criteria introduced in Section 6.1 are indicative of external (ir)regularity.

In its most general form, external irregularity can be characterized in the following way: a linguistic entity is externally irregular, if its distribution cannot be predicted by its internal properties and the principles of the grammar. In principle, there are two kinds of deviations from the regular distributional pattern: An expression is banned from contexts where one would expect it, given its meaning and its syntactic category. Alternatively, the expression could be allowed in contexts where similar expressions do not occur. In this thesis, we will only be concerned with externally irregular expressions that have a more restricted distribution than what would be expected given the regular rules of grammar.

The syntactic and semantic criteria of regularity contain some that give us clear indication of external irregularity. Consider the first syntactic criterion: If a word such as *headway* cannot be used outside a certain VP, this is a clear indication of its external irregularity. The same is true for the first semantic criterion. If a word such as *beans* associated with a certain meaning, such as the of *secret/information* is bound to occur as a direct object of a certain verb, this indicates a very restricted distribution, which proves the external irregularity of parts of this expression. As can be seen in Table 6.1, all the IEs of (320) fail to satisfy the first semantic criterion. Restricting ourselves to those IEs that are internally regular, this means that these IEs contain at least one word which does not occur freely in the given meaning. For some words such as *headway* this means that they do not occur at all outside the IE, for others, such as *beans* in the meaning of *information*, it is the special form-meaning pairing that does not occur outside the IE. For others, like *pull*, occur in a certain group of contexts, but not freely. Finally, in the case of *make* this group of contexts is considerably large. Still, it could be shown that, in its light verb meaning, *make* cannot occur with every noun which is a candidate for a light verb construction.

Based on the notions of internal and external (ir)regularity, we can classify the data summarized in Table 6.1. For each of the examined IEs, we will give a brief characterization in what respect the entire VP or parts thereof are internally or externally (ir)regular.

In the case of a freely combined VP such as *read a book*, we have three words *read*, *a*, and *book* which are all combined syntactically in a standard way and the meaning of the entire VP results from a regular combination of the meanings of these words, given the syntactic structure used to combine these words. In addition all these words freely occur with the same meaning in other combinations. Thus, the entire VP is constructed in an internally regular way from the words that it contains. All these words are externally regular.

In the discussion of the IE *trip the light fantastic*, we could not assign the VP a clear syntactic structure, nor could we provide any sensible meaning assignment to its parts. The only analytical option we can see is that the entire VP must be treated as internally irregular. Still, there must be some internal structure to the VP, as the verb *trip* receives its regular inflection. Therefore, we conclude that the VP as a whole is assigned the meaning *dance*. The only thing that is known about the internal structure of the VP is that its head is the verb *trip*. As the meaning assignment is done to the VP as a whole, it follows that we cannot alter the VP syntactically under the same meaning. The IE, as a whole is, thus, internally irregular.

The case of *kick the bucket* is very similar to this. Even though we can assign this VP a regular syntactic structure, this structure cannot be used for semantic interpretation. Instead, we must assign the meaning *die* to the VP as a whole. As the VP is of a regular structure, we can allow for some syntactic modifiers to enter the structure. Parts of the IE do not have a meaning that will participate in the overall meaning of the VP. Therefore, the only way for an adjunct to make a semantic contribution is by being interpreted as an external modifier. Such an interpretation has been shown to be possible, in principle, for freely combined VPs as well. We can again, block greater syntactic flexibility by requiring that the meaning assignment be made to the VP only if it is of the right shape.

These two IEs were the only ones in our list that were absolutely incapable of participating in syntactic operations, and, in fact, the only two whose meaning could not be spread over the components of the IE. We took the noncompositionality (or non-decomposability) of these IEs as a criterion to classify them as internally irregular. Eventually, this will lead us to treat them as irregular complex entities.

Let us turn to the other extreme, the VP *make a decision*. We indicated in Table 6.1 that this VP shows almost fully regular behavior, the only problem being that not all potential candidates of English nouns can combine with the light verb *make*. As we want to assume regularity wherever possible, we claim that *make a decision* really is built up in a regular way from its components, the light verb *make*, the determiner *a* and the noun *decision*. It is, however, a lexical property of the light verb *make* that it accepts only some NPs as its complement. Such NPs are for example those headed by *decision*, but not those headed by *laundry*. It is clear that this requirement need not be fulfilled locally, i.e., within the same VP. In the data we have collected, the light verb *make* can also occur if its direct object is extracted or takes the form of a relative pronoun. In both cases, however, there must ultimately be a relation to a noun that falls into the class of elements that can be combined with the verb *make* in its light verb reading.

To summarize, the VP *make a decision* is regularly built, but the light verb *make* comes with the distributional requirement that its direct object be of a certain kind, i.e., be related to an occurrence of a noun like *decision*. What makes us call the VP *make a decision* an IE is precisely the fact that, even though it is internally freely combined, it contains a word, *make*, which is very restricted in its occurrence possibilities. This leads us to classify the IE as internally regular, but one part of the IE, the verb *make*, exhibits external irregularities.

We have thus encountered two different sources of irregularity in the IEs considered: a VP like *trip the light fantastic* is internally a fully irregular, but syntactically complex entity. A VP like *make a decision* is internally fully regular, but part of it, the verb *make* in its light verb use is restricted to co-occur with a noun like *decision*. We will see that the other VPs considered in this chapter, while being internally regular, all consist of two elements with particular distributional properties, i.e., with external irregularities.

Let us first reconsider the IE *make headway* . In this IE, as we have seen in the discussion of the data, the verb *make* is used as a light verb, just as in the case of *make a decision*. *Headway* can be considered a noun, whose meaning is roughly that of *progress*. As a light verb, *make* has the distributional restrictions mentioned above. The noun *headway* combines with this verb in a syntactically and semantically regular way. It is, however, irregular in so far as it does not occur outside the given IE (the first syntactic criterion). This property can be seen as a external irregularity of the noun *headway*, just as the restriction on the light verb *make*.

The IE *pull strings* differs from *make headway* in two respects: first, it is more regular, as all its parts occur independently in the language. On the other hand, it is less regular, as none of its parts occurs with the meaning used in the IE outside this particular combination, i.e., *pull* in the meaning of *use* and *strings* in the meaning of *connections*. Under this meaning assignment, the syntactic structure and the meaning of the IE can be regularly derived. We must, however, prevent the words from occurring in this meaning outside the IE. Treating this case parallel to that of *make headway*, we can impose strong distributional requirements on both elements of the IE, in this case, the verb *pull* is restricted to occur in its idiomatic meaning only in the context of the word *strings* (and the word *wires*). The distributional restrictions on *strings* in the idiomatic reading are parallel to those of *headway*, i.e., it must occur as the direct object of a particular verb.

The last IE that we treated in the preceding section is the VP *spill the beans*. As our criteria evidenced, we can treat the VP as regular, once we accept an exceptional meaning assignment to its parts. Thus. the combination of words is regular, but the words exhibit special distributional requirements, which ensures that the they occur in this particular reading only when they are in combination with each other.

Treated under this perspective of internal and external irregularity, we would expect there to be a clear distinction between the syntactic and semantic behavior of *trip the light fantastic* and *kick the bucket* on the one hand and the rest of the IEs in (320) on the other hand. Indeed, this distinction exists: as mentioned above, these IEs are the only VPs in our collection that do not passivize, or participate in any of the other syntactic operation.

What is more surprising, is the difference in behavior between the more restricted IEs *spill the beans* and *make headway* on one hand and the very flexible *pull the strings* and *make a decision* on the other. Why is it that the former two have less syntactic freedom than the latter? In the rest of this section we will try to derive this contrast from the semantic properties of the NPs under consideration.

The IE *make headway* differs from the VP *make a decision* only in not allowing for pronominal reference to the noun. The examples used to show this are repeated below.

(399) a. * Mary made considerable headway$_i$ and John made it$_i$, too.
b. * I thought that considerable headway$_i$ was made in the last 15 years, but it$_i$ wasn't made.

(400) a. Mary made considerable progress$_i$ and John made it$_i$, too.
b. I thought that considerable progress$_i$ was made in the last 15 years, but it$_i$ wasn't made.

Our judgment was motivated by the contrast between *make headway* and the synonymous expression *make progress*. The pattern can, however, be explained if we consider what kind of entity is denoted by the noun *headway*. *Headway* clearly denotes a mass. The data in (399) are, indeed, parallel to what we find with an ordinary mass noun such as *milk*.

(401) a. * Mary drank (a lot of) milk$_i$ and John drank it$_i$, too.
b. Mary drank (a lot of) milk$_i$ and John drank some$_i$, too.

In (401a), the personal pronoun *it* cannot refer to the mass noun *milk*. The indefinite term *some* in (401b), on the other hand can. Changing *it* to *some* in the case of *headway* will also result in a grammatical sentence.

(402) Mary made considerable headway$_i$ and John made some$_i$, too.

This shows that we can account for the pronominalization data, by assuming that the noun *headway* has only a mass noun reading. In this respect, it is special, because most nouns, even if they have a mass noun reading, can also occur as count nouns.

(403) There is still one milk in the fridge, please put it on the table for breakfast.

This brief discussion shows that, while the fact that the noun *headway* cannot be pronominalized is irregular, it can be reduced to a general property of mass expressions. Thus, all that must be assumed is that the noun denotes mass expressions.

Matters are different in the case of the IE *spill the beans*. There, we gave three slightly different patterns with respect to our criteria of regularity: In the main text, we observed that this IE satisfies all the syntactic criteria except for the last two, i.e., the NP *the beans* cannot be topicalized (criterion 5) nor can there be a relative clause attached to *the beans* (criterion 6). The relevant examples are repeated in (404) and (405).

(404) * The beans John spilled.

(405) a. * The alleged arms dealer spilled the beans [that made the party leader resign].
b. * The beans [that the alleged arms dealer spilled] made the party leader resign.

Following the way other authors such as Schenk 1995 treat topicalization, we might assume that the distinction follows from more general restrictions on what kinds of entities can be topicalized.[21] These restrictions have to do with the degree of meaningfulness of a certain term. Schenk assumes that topicalization is only possible if the topicalized constituent contains some "non-idiomatic" part. It may, as well, contain idiomatic material which is seen to be "pied-piped" together with the non-idiomatic material. To see his point, consider the following data.[22]

(406) a. Pete pulled Mary's leg.
b. Mary's leg Pete pulled.

---

[21] Note, however, that Schenk 1995 assumes that *beans* in *spill the beans* is basically meaningless, and, thus, has trouble to account for the difference in flexibility between the passivizable *spill the beans* and the fully fixed IE *kick the bucket*.

[22] Schenk 1995 (p. 260) states:

> Because *Mary* is a free argument in [(406)], it is possible to topicalize constituents containing an idiomatic subpart, whereas it is impossible to topicalize constituents that do not contain a meaningful subpart, as in the earlier examples.

The IE *pull someone's leg* allows the direct object to be topicalized. In the examples in (406), the direct object contains the proper name *Mary*, which is taken to be a non-idiomatic part. In Schenk's perspective, as *Mary* is a non-idiomatically interpreted word, it should be possible to topicalize it. Due to syntactic constraints of English, captured for example in the *Left Branch Condition*, it is impossible to extract a possessive NP out of the NP which contains it. This is illustrated in (407).[23]

(407) a. John bought Pete's book.
b. Pete's book John bought.
c. * Pete's John bought book.
d. * Pete John bought 's book.

In the case of the IE *pull strings*, topicalization of the direct object is possible if it contains some element other than *strings*, i.e., some non-idiomatic bit. This includes a determiner, an adjective etc. As can be seen in (408), there is the demonstrative *those* in the topicalized NP.

(408) Those strings, he wouldn't pull for you. (Nunberg et al., 1994, p. 13)

There is, however, a problem with Schenk's explanation. In contrast to Schenk's judgments, our informants allowed for internal modification in the case of the IE *spill the beans* (see the examples in (348)). Thus, there may be semantically non-vacuous material occurring inside the NP *the beans*. Still, topicalization is far from grammatical for our informants.

(409) a. Pat spilled the well-guarded beans.
b. ?* The well-guarded beans Pat spilled.

The ungrammaticality of (409b) is unexpected under Schenk's generalization. In principle, the explanation offered in Schenk 1995 points to the possibility to link the extraction data to general constraints on topicalization. The existence of such constraints cannot be denied, even though they are still poorly understood. Some of these restrictions certainly have to do with the content of a topicalized constituent. Still, we think that under Schenk's explanation the contrast in (409) would not be expected.[24]

It must be noted that according to the judgments in Schenk 1995, both sentences in (409) are ungrammatical. Such a grammaticality pattern is in line with Schenk's generalization, as the NP *the well-guarded beans* is considered ungrammatical in the first place. Similarly, if we accept the data pattern assumed in Wasow et al. 1983, Schenk's generalization seems to make the right predictions. According to Wasow et al. 1983, both sentences in (409) are grammatical: the (b) sentence, i.e., the topicalization should be possible, as it contains non-idiomatic material.

---

[23]Examples (406) and (407) are taken from Schenk 1995 (p. 260).

[24]It should be noted that there are more exceptions to Schenk's generalization. As pointed out in Nunberg et al. 1994, in German, there are some IEs such as *ins Gras beißen* (*bite the dust*) (literally: *bite into the grass*) or *den Vogel abschießen* (*to steal the show*) (literally: *shoot of the bird*) whose parts cannot be assigned a compositional meaning. Still, many speakers allow the PP or the NP part of these IEs to be fronted in verb second clauses. This fronting is usually analyzed by a slash mechanism just as the one used for topicalization in English (see Uszkoreit 1987 for such an analysis within GPSG, and Pollard 1996 or Kiss 1995 for HPSG).

(i) Er hat [ins Gras] gebissen.
He has into the grass bitten 'He died.'

(ii) [Ins Gras] hat er gebissen.
(iii) * [gebissen] hat er ins Gras.

Surprisingly, in the light of the generalization of Schenk 1995, the "meaningless" PP *ins Gras* can be fronted. Schenk's generalization is, however, met with respect to the verb *gebissen* which may not be fronted if the idiomatic meaning is to be conserved. But note that sentence (iii) is grammatical under the literal meaning, i.e., when the verb *gebissen* is "contentful".

In this part of the thesis, we will assume the data to be as given in the main text. Speakers for whom the IE *spill the beans* is more regular, would simply have to assume an analysis as the one given for the IE *pull strings*. For speakers that are more restrictive than we have indicated, i.e., speakers that share the judgments of Schenk 1995, we will point out in footnotes those places in the analysis that need to be changed to be adapted to their judgments.

This discussion of the IEs *make headway* and *spill the beans* has shown that some of the syntactic irregularities of IEs might be reducible to other properties of certain parts of the IE. For example, in the case of *make headway*, we could predict its non-pronominalizability by a general property of mass terms. In the case of *spill the beans*, however, we could not reduce the absence of topicalization to any general restriction. This shows that while both IEs are classified as internally regular, there is still a lot of idiosyncratic variation within this class of IEs to account for.

CHAPTER 7

Existing Approaches

With the discussion of some VPs with irregular behavior, we have presented a data space which must be accounted for in a linguistic theory. An account of irregularities is particularly challenging for frameworks that try to achieve a fully formalized, or at least fully formalizable, description of natural language, because formal systems usually emphasize their potential of expressing generalizations and regularities. In this chapter, we will discuss three different approaches to IEs, formulated in three different frameworks: The analysis of Gazdar et al. 1985 carried out in the framework of *Generalized Phrase Structure Grammar* (GPSG), the *Tree Adjoining Grammar* (TAG) account of Abeillé 1995, and finally the approach of Riehemann 1997, which is stated within a version of HPSG that we will call *constructional HPSG*.

As all three frameworks differ significantly from the version of HPSG adopted in this thesis, we will in all cases first present how the free combination *read a book* is accounted for. Then, we will discuss the different IEs which we have considered in the previous chapter. We will show whether and how the approaches can handle the diversity that these IEs exhibit with respect to our criteria of regularity. It will turn out that the GPSG account is best suited for internally regular IEs such as *spill the beans*, although the mechanism assumed there is relatively complicated and unintuitive. The TAG approach, on the other hand, seems to be ideal for internally irregular IEs such as *kick the bucket*. The approach given in constructional HPSG appears to be equally well-suited for both internally regular and internally irregular IEs and it avoids some of the problems of the GPSG-account. It will be shown, however, to be problematic from the point of view of its formal architecture, as HPSG is less construction-oriented than TAG.

As a result of the survey in this chapter and from the tension between regularity and irregularity, a clearer picture emerges of what an HPSG theory of the IE data should look like. Such a theory will then be presented in the following chapter.

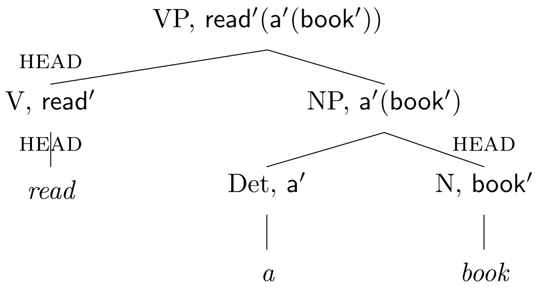
7.1. IEs in GPSG

Gazdar et al. 1985 (pp. 236–242) provide a formal account of the empirical insights gained in Wasow et al. 1983 within the famework of *Generalized Phrase Structure Grammar* (GPSG). The authors explicitly address the two IEs *spill the beans* and *pull strings*. Their analysis of these IEs is based on two assumption: (i), that the combinations are basically regularly built and interpreted; and (ii), that the interpretation of a term is a partial function, i.e., it need not be defined on all arguments.

Gazdar et al. 1985 treat IEs as a basically semantic phenomenon. For this reason, we do not need to present the details of the syntactic aspects of GPSG. The syntactic structure that the authors assume for a VP such as *read a book* is similar to the one given in (312).[1] What is important to notice, however, is that the authors assume two basic kinds of trees: First, terminated local trees, i.e., local trees whose (single) leaf is a terminal symbol and second, local trees that are admitted by some rule of the grammar and by the

[1]Technically, the syntactic labels used in (312) are considered to be sets of featue-value pairs in GPSG.

FIGURE 7.1. The GPSG analysis of the VP *read a book*:



(universal and language-specific) principles. Larger trees are composed from local trees and the grammaticality of a larger tree is fully derivable from that of all the local trees that it is composed of.

Gazdar et al. 1985 provide an explicit model-theoretic semantic system along with the syntax of GPSG.[2] Following Montague, the authors assume that the meaning of an expression can be captured as the interpretation of some term of *Intensional Logic* (IL). Thus, every syntactic category is associated with some term of IL. In the case of terminated local trees, it is just the mother that is a syntactic category and, therefore, has an IL term associated with it. For local trees that are admitted by a rule, the IL term associated with the mother is the result of applying functional application to the IL terms associated with the immediate daughters.[3]

To give a concrete example, the word *read* is associated with the following terminated local tree:

(410) V, $\mathsf{read}'_{\langle NP, \langle NP, S\rangle\rangle}$
|
read

This indicates that the word *read* enters a syntactic tree with the category V and the IL term read′ which is of the semantic type $\langle NP, \langle NP, S\rangle\rangle$, which is the type assumed in Gazdar et al. 1985 for transitive verbs. The type S abbreviates the type $\langle s, t\rangle$, and NP abbreviates the type $\langle s, \langle\langle s, \langle e, t\rangle\rangle, t\rangle\rangle$ (Klein and Sag, 1985, p. 170).

The authors explicitly allow the same phonological string to be associated with several terminated local trees, and, in particular, with several IL terms. For raising verbs and passive verb forms, the authors assume operators f_{R} and f_{P} respectively that alter the order of the arguments, i.e., if read′ is the semantic constant associated with the active verb *read*, then $f_{\mathrm{P}}(\mathsf{read}')$ is the term associated with the passive form. The effect of the operator becomes clear, once the verb is combined with its arguments. Let ϕ be the IL term associated with the active subject and ψ be the term associated with the direct object, then, a meaning postulate is evoked to ensure that $\mathsf{read}'(\psi)(\phi)$ is equivalent to $f_{\mathrm{P}}(\mathsf{read}')(\phi)(\psi)$.[4]

In local trees whose leaves are non-terminals, the IL term associated with the mother is the functional application of the IL terms on the daughters. To illustrate this, consider Figure 7.1 which shows the tree for the VP *read a book*, enriched with IL terms.

---

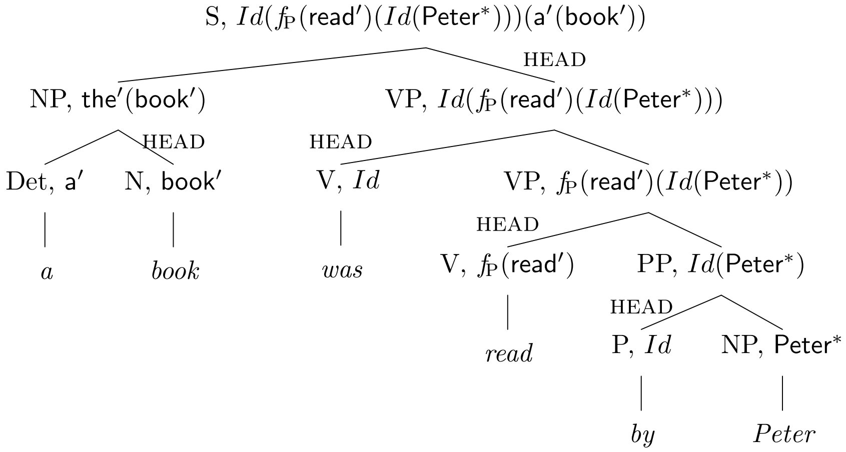
[2]This semantic system is presented in more detail, but with fewer phenomena in Klein and Sag 1985.

[3]The final formulation of *Semantic Interpretation Schema* is explicitly fine-tuned to the syntactic analysis. See below for the treatment of extracted constituents.

[4]It is not strictly necessary to do this by a meaning postulate. Instead, the functor f_{P} could be specified as indicated in (i), where β is either the type NP or the type S, and V is a variable of type $\langle \alpha_1, \langle \ldots \langle \alpha_n, \langle \beta, \langle NP, S\rangle\rangle\rangle \ldots\rangle\rangle$.

(i) $\lambda V \lambda x_{NP} \lambda a_{1,\alpha_1} \ldots \lambda a_{n,\alpha_n} \lambda y_{\beta}.V(a_1)\ldots(a_n)(y)(x)$

FIGURE 7.2. The GPSG analysis of sentence (411b):



We can now understand the derivation of the VP in Figure 7.1: The noun *book* is associated with a term book' of type $s(et)$, abbreviated as Nom in Gazdar et al. 1985 and Klein and Sag 1985. The determiner *a* is associated with a term a' of type $\langle Nom, NP \rangle$, i.e., it combines with a term of type Nom (a noun) to yield a term of type NP (an NP). Thus, in the local tree whose mother is the NP, the term associated with the syntactic category is $\mathsf{a}'(\mathsf{book}')$, which is of type NP. In the highest local tree, this term combines via functional application with the constant read' to yield the given semantic term for the entire tree.

In Gazdar et al. 1985, it is assumed that passive and topicalization do not change the meaning of a sentence. Thus, all sentences in (411) have the interpretation given in (412).[5]

(411) a. Peter read a book.
b. A book was read by Peter.
c. A book, Peter read.

(412) $\mathsf{read}'(\mathsf{a}'(\mathsf{book}'))(\mathsf{Peter}^*)$

In Figure 7.1, we have shown the structure of the VP in the active sentence (411a). For the passive sentence in (411b), the structure can be seen in Figure 7.2.

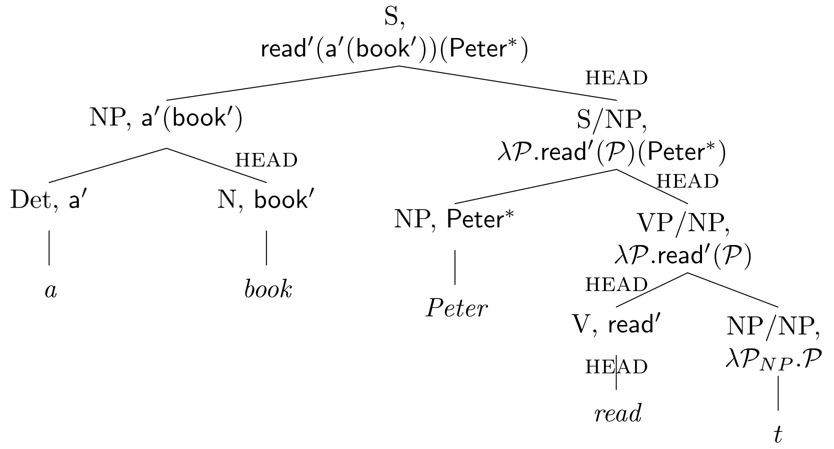
The semantics of passive sentences is accounted for through the special operator f_P. In addition, there are two lexical elements in this tree that we have not considered so far: the passive auxilliary *was* and the preposition *by*. Klein and Sag 1985 (p. 199 fn. 33) assume that the passive auxiliary *be* is interpreted as the identity function. For this reason, we have given the node that dominates the verb *was* the identity function, Id, of the appropriate type as semantic term. Similarly, selected PPs such as the PP headed by passive *by* or *for* in the case of the verb *wait for NP* are considered to be of the same semantic type as NP complements, i.e. $\langle s, \langle \langle s, \langle e, t \rangle \rangle, t \rangle \rangle$. The underlying prepositions are, thus, of type $\langle NP, NP \rangle$ and are interpreted as the identity function (Klein and Sag, 1985, p. 167).

Given the semantic types and the terms at the terminal local trees, the term that is associated with the top S node can be computed. The following equations prove that this semantic term is equivalent to that in (412).

(413) $Id(f_P(\mathsf{read}')(Id(\mathsf{Peter}^*)))(\mathsf{a}'(\mathsf{book}'))$
$= f_P(\mathsf{read}')(\mathsf{Peter}^*)(\mathsf{a}'(\mathsf{book}'))$ (leaving out Id)

[5] The constant Peter^* is of type NP and is interpreted as $\lambda P. {}^{\wedge}P({}^{\vee}\mathsf{p})$, where P is a variable of type $\langle s, \langle e, t \rangle \rangle$ and p is a constant of type $\langle s, e \rangle$ which denotes for each index w the individual referred to as *Peter* in w.

FIGURE 7.3. The GPSG analysis of the sentence *A book Peter reads*:



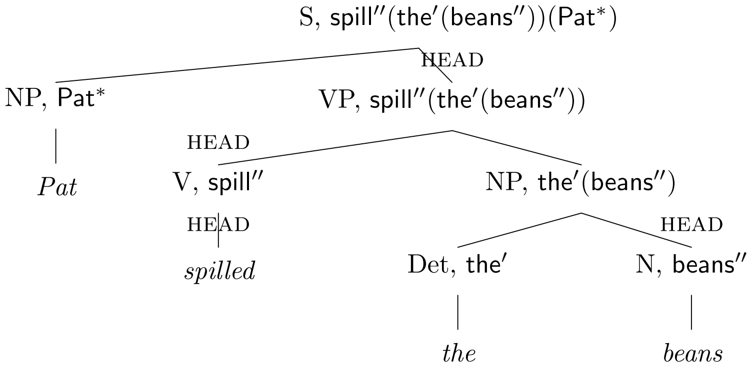
$$= [\lambda V \lambda x \lambda y.V(y)(x)](\mathsf{read}')(\mathsf{Peter}^*)(\mathsf{a}'(\mathsf{book}')) \quad \text{(replacing } f_{\mathrm{P}}\text{)}$$
$$= [\lambda x \lambda y.\mathsf{read}'(y)(x)](\mathsf{Peter}^*)(\mathsf{a}'(\mathsf{book}')) \quad (\lambda\text{-conversion})$$
$$= [\lambda y.\mathsf{read}'(y)(\mathsf{Peter}^*)](\mathsf{a}'(\mathsf{book}')) \quad (\lambda\text{-conversion})$$
$$= \mathsf{read}'(\mathsf{a}'(\mathsf{book}'))(\mathsf{Peter}^*) \quad (\lambda\text{-conversion})$$

In the case of unbounded dependencies, a variable of type NP is introduced. Syntactically, unbounded dependencies are analyzed using a trace, which introduces a SLASH, i.e., the syntactic category of the trace is $\alpha \cup \{\langle \text{SLASH}, \alpha \rangle\}$, where α is some syntactic category, i.e., a set of feature-value specifications. In local trees, the slash value percolates from the daughter(s) to the mother, up to the point where a special Immediate Dominance Rule introduces a "filler" which, then, blocks further percolation of the slash value.

Semantically, a trace is analyzed as a variable of type NP. In order to be able to eliminate this variable at some stage, the semantic interpretation schema of Gazdar et al. 1985 is a little bit more complex than we presented it above. The semantics of the mother node in a local tree is arrived at by functional application alone just in the case that none of the nodes in the local tree has a SLASH specification. If, however, there is such a specification, it is first ignored in the computation of the resulting term, but then added as a lambda abstraction. This can best be seen in the analysis of a simple sentence with topicalization as indicated in Figure 7.3, where we give the syntactic category together with the semantic term in the notation used in Gazdar et al. 1985.

To illustrate this extra effect of the semantics principle, consider the terminated local tree that contains the trace. Neglecting the SLASH specification, the term associated with the syntactic category of the trace is simply the variable $\mathcal{P}$. As the node contains a SLASH specification, its semantics must be added as a lambda abstractor to the IL term at the node in the tree, resulting in the term $\lambda\mathcal{P}.\mathcal{P}$. In the next local tree, whose mother node is the VP, we see that the semantic term of the head daughter (read') combined with the basic semantics of the nonhead to yield $\mathsf{read}'(\mathcal{P})$. To this term, we must add, again, the SLASH specification, which gives us the term $\lambda\mathcal{P}.\mathsf{read}'(\mathcal{P})$. We proceed in this way until we finally reach the top local tree. There, the slash value is stopped from percolating higher up in the tree by a special Immediate Dominance Rule. This rule, or more precisely the feature specifications that it induces, also triggers the effect that the lambda abstractor is not ignored in the calculation of the resulting semantic term this time. Therefore, at the upper S node, we calculate the semantic term of the mother by applying the functor $\lambda\mathcal{P}.\mathsf{read}'(\mathcal{P})(\mathsf{Peter}^*)$ to the term $\mathsf{a}'(\mathsf{book}')$, which results in the term $\mathsf{read}'(\mathsf{a}'(\mathsf{book}'))(\mathsf{Peter}^*)$ as shown in the tree.

Figure 7.4. The analysis of sentence (415):



After these general remarks on the way semantic interpretation is handled in Gazdar et al. 1985, we can turn explicitly to their treatment of IEs. As mentioned above, the authors are mainly concerned with the IEs *spill the beans* and *pull strings*. Based on the fact that both IEs can take internal modifiers (our third semantic criterion), that both can passivize (our fourth syntactic criterion), and that, at least with the IE *pull strings*, the direct object can be topicalized, the authors argue that the meaning of the IEs should be distributed over its parts and, furthermore, that one should assume a regular syntactic and semantic combination for these IEs.

To achieve such an analysis, Gazdar et al. 1985 introduce separate non-logical constants for the idiomatic reading of *spill* and *beans*, spill″ and beans″:

> "To illustrate, the verb *spill* might be assigned two senses ..., which we can represent as two distinct expressions of intensional logic: spill′ (representing the literal sense) and spill″ (representing the idiomatic sense — roughly (but not exactly) the sense of *divulge*). Similarly, *beans* is assigned two senses: beans′ and beans″, the latter of which has roughly (but not exactly) the sense of *information*." (Gazdar et al., 1985, p. 238)

They propose the following two terminated local trees for the words *spill* and *beans*.

(414) a. V, spill′$\langle NP, \langle NP, S\rangle\rangle$ — *spill*   V, spill″$\langle NP, \langle NP, S\rangle\rangle$ — *spill*

b. V, beans′$_{Nom}$ — *beans*   V, beans″$_{Nom}$ — *beans*

Once new semantic constants are introduced for the idiomatic reading of parts of an IE, it is easy to derive the regular effects of the IE. A sentence such as (415) receives the literal interpretation, if the first terminated local trees in (414a) and (414b) are used, but it will get the idiomatic reading, in case the second terminated local trees are used.

(415) Pat spilled the beans.

In Figure 7.4 we show the structure of sentence (415), where we chose to take the idiomatic interpretation of the words *spill* and *beans*.

Under this analysis, the question of how we can block the use of the beans″ reading arisis, when the noun *beans* combines with verbs other than spill, but synonymous with its spill″ sense. Gazdar et al. 1985 achieve this by two assumptions: First, there are no absolute synonyms in the language, i.e., there are no two constants whose intensions are identical. Second, the interpretation of a constant is a partial function, i.e., it may not be defined on some functor-argument pairs, even if the combination is semantically well-typed. Combining these two assumptions, Gazdar et al. 1985 claim that the interpretation of the constant spill″ is such that it is only defined on the denotation of beans″, but not on that of information′.[6] This restriction prevents the idiomatic use of *spill* from being completely interchangable with the verb *divulge*.

Analogously, as also noted in Pulman 1993, we must impose a restriction on all other constants: the application of any other constant to the denotation of beans″ must not be defined. This extra assumption, which is not made explicit in Gazdar et al. 1985, is needed to prevent the idiomatic use of *beans* from freely occurring in places where *information* can occur (compare the examples in (347) above).

(416) a. She divulged the story/ *the beans to the media.
b. We were hoping to have information/ *the beans on this dramatic incident soon.

The object NP *the beans* consists of the idiomatically interpreted noun *beans* and the definite article. In their own brief discussion, Gazdar et al. 1985 (p. 238) state that

> "the normal principles of compositional semantics will thus assign an idiomatic interpretation to *the beans* (we will represent this interpretation informally as the-beans″), which serves as the argument of spill″."

In the tree given above, we have not used the term the-beans″, because, under the assumption of a regular combination, as expressed in the quote, the resulting term should be the′(beans″) (whatever the sense of the′ is). In either case, the semantic term associated with the object NP combines with that of the finite verb.

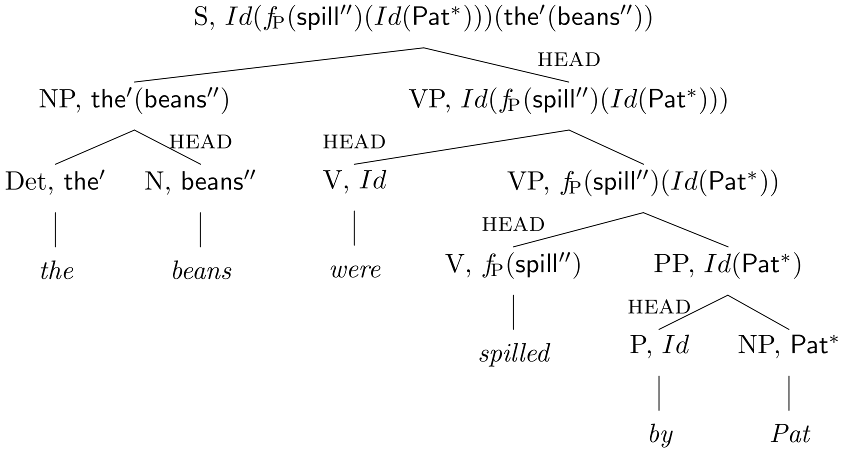
The elegance of the analysis of IEs in Gazdar et al. 1985 comes from the simple assumption of only partially defined interpretation. This assumption enables them to treat the IE as a regular combination, i.e., the syntactic and semantic flexibility of the IE follows. As we saw, in order to achieve the idiomatic reading for sentence (415), we took the idiomatic terminated local trees for *spill* and *beans* and applied the regular means of syntactic and semantic combination. To illustrate that this account is equally capable of capturing the syntactic freedom of the IE, we will show the analysis of this IE, when used in the passive, as in sentence (417). The analysis of this sentence is given in Figure 7.5. This figure shows that the structure of the passive sentence is just like that in Figure 7.2 above.

(417) The beans were spilled by Pat.

Under the IE reading, the second terminated local trees of (414) are used for the words *spill* and *beans*. In the case of *spill*, when used in the passive, the passive operator f_P appears as part of the semantic term associated with the mother node in the local tree. In the lower VP, this term combines with that of the *by*-PP whose semantics is that of its NP complement, Pat* in this example. In the upper VP, the VP *spilled by Pat* appears as the nonhead daughter. The head daughter in this VP is the finite passive auxiliary verb *were*. As noted above, the semantics of this verb is the identity function.

[6] This is, of course, only needed, if we ignore the slang use of *spill*, otherwise, no such restriction is needed for spill″. We do, however, need such a restriction in the case of other IEs such as *pull strings* or *keep tabs on someone*.

FIGURE 7.5. The analysis of sentence (417):



The next step is to show that the first argument of the constant spill'' is in the denotation of the constant beans''. Otherwise, the entire sentence could not be assigned a denotation. In the term associated with the S node, it is not immediately clear that the required constellation is found: The constant spill'' does not appear as a functor at all, but as an argument of the passive operator f_{P}. The term $\mathsf{the}'(\mathsf{beans}'')$ occurs as the second argument to the identity function of type $\langle VP, VP\rangle$.[7]

But, as shown in (413), the term that is associated with a passive sentence is equivalent to that associated with the corresponding active sentence. The reason for this is that the passive auxilliary and the preposition *by* denote the identity function and that the passive operator f_{P} has the effect of reordering the arguments of the verb. In the case in Figure 7.5, the operator has the same form as for the verb *read*, i.e., $\lambda V \lambda x \lambda y.V(y)(x)$. We can, therefore, show that in the interpretation of the sentence, the constant spill'' is in fact interpreted with respect to the right domain, i.e., the denotation of $\mathsf{the}'(\mathsf{beans}'')$.

(418)
$$
\begin{aligned}
& Id(f_{\mathrm{P}}(\mathsf{spill}'')(Id(\mathsf{Pat}^*)))(\mathsf{the}'(\mathsf{beans}''))\\
&= f_{\mathrm{P}}(\mathsf{spill}'')(\mathsf{Pat}^*)(\mathsf{the}'(\mathsf{beans}''))\\
&= [\lambda V \lambda x \lambda y.V(y)(x)](\mathsf{spill}'')(\mathsf{Pat}^*)(\mathsf{the}'(\mathsf{beans}''))\\
&= [\lambda x \lambda y.\mathsf{spill}''(y)(x)](\mathsf{Pat}^*)(\mathsf{the}'(\mathsf{beans}''))\\
&= [\lambda y.\mathsf{spill}''(y)(\mathsf{Pat}^*)](\mathsf{the}'(\mathsf{beans}''))\\
&= \mathsf{spill}''(\mathsf{the}'(\mathsf{beans}''))(\mathsf{Pat}^*)
\end{aligned}
$$

The term that we receive after having performed lambda conversion shows that in the case of a passive, the constant spill'' combines with the intension of beans'', which is the only kind of argument on which the interpretation of spill'' is defined.

This analysis accounts in a natural way for the syntactically and semantically regular behavior of the IE. It attributes all the idiosyncrasies of the combination to some unusual semantic properties of the constants that are involved, i.e., the restricted domain on which the interpretation of the relation denoted by the verb is defined.

Given this approach, we would expect the IE reading to be possible in all syntactic environments whose semantics is that of the active sentence *Pat spilled the beans*. In (411) we listed some of these contexts: passive and topicalization. In (417), we have seen how this result is achieved for the passivized form of the IE. For topicalization, we can have a tree like that in Figure 7.3, but using the constants spill'', the' and beans'' instead of read',

[7] VP abbreviates the type $\langle NP, S\rangle$.

a′ and book′. Such a tree would be accepted by the grammar and would also be assigned a denotation. It is semantically equivalent to a sentence which contains the topicalized constituent in the place of the trace.

Our empirical overview in Section 6.2, however, revealed that — at least for some speakers — the IE *spill the beans* does not allow topicalization of the NP *the beans*, whereas such topicalization is readily available with the IEs *pull strings*, *make headway* and *make a decision*. This observation brings us to a discussion of the problematic aspects of the analysis of IEs in Gazdar et al. 1985. Our criticism will concern three points: (i) the treatment is too coarse to allow for a differentiation in the behavior of several apparently compositional IEs such as *spill the beans* and *pull strings*, (ii) the use of partial meaning functions to express the occurrence restrictions of parts of IEs is problematic, and (iii) there is no straightforward way to include IEs of the type *trip the light fantastic* and *kick the bucket* into the overall architecture of GPSG.

Let us address the first point. As mentioned above, the IEs *spill the beans* and *pull strings* differ in their ability to conserve the idiomatic reading if the direct object is topicalized. The relevant examples from Section 6.2 are repeated below.[8]

(419) a. = (354) * The beans John spilled.
b. = (388) Those strings, he wouldn't pull for you.

Since both IE can be passivized, they must be analyzed as basically regular, only exhibiting some irregularity in the interpretation of the constants. Gazdar et al. 1985 treat all such IEs alike and predicts that they behave the same with respect to all syntactic operations that are semantically neutral, which include topicalization. Therefore the difference in (419) cannot be captured in their framework as part of the grammar.

As this particular problem is not addressed in Gazdar et al. 1985, we can only speculate in what direction a solution should be sought. In Section 6.3 we have argued that there is no principled reason why topicalization is allowed for *pull strings*, but not possible with the IE *spill the beans*. Under the assumption that this contrast really is an idiosyncrasy of the IEs, the distinction cannot be expressed with the means given in Gazdar et al. 1985, because in their system any combination of a V and an NP that can be combined to form an active VP should also appear in a passive clause and in a topicalization construction.[9]

Let us address the second problematic aspect of the approach to IEs in Gazdar et al. 1985. It has to do with the notion of partial function which is taken as the basis for the entire treatment of IEs. Pulman 1993 (pp. 257–261) tries to show the consequences of the partial function approach to the possible semantic models. To see his argument, let us take the two IEs *spill the beans* and *pull strings*.[10] Pulman first shows that in cases such as that of the IE *spill the beans*, it is not enough to restrict the domain of the interpretation of the constant spill″ to the denotation of the constant beans″. One must also restrict the domain of the interpretation of all other semantic constants in such a way that they cannot combine in an interpretable way with the denotation of the constant beans″. If this is not done, we will find the idiomatic reading of *beans* freely occurring.

[8]The judgments in (419) are those given above in the main text. In footnotes, we have pointed out that for Wasow et al. 1983 both sentences should be grammatical. Under this assumption, of course, there is no problem for the GPSG account. The problem remains, however, if we want to give an account of the judgments as referred in Schenk 1995. There passive is assumed to be possible with most IEs, whereas topicalization should be excluded.

[9]Technically, we can solve this problem, if we introduce a new feature, EXTRACTABLE, with values + or −. All ID-rules that mention a SLASH specification must be changed to include the specification +EXTRACTABLE. The idiomatic word *beans* would inherit the feature specification −EXTRACTABLE from the lexicon. Even if such an ad hoc modification works, it is still true that the difference between the two IEs cannot be directly expressed in the grammar, i.e., with the inventory of features, rules and principles in Gazdar et al. 1985.

[10]Pulman actually uses the IEs *axe to grind* (*point to make*) and *break the ice* (*ease the tension*).

The system of restrictions does, however, get still more complicated, as Pulman 1993 (p. 259f.) shows. Compare the following two sentences:

(420) a. * The strings got Chris the job.
b. The strings [that Pat pulled] got Chris the job.
c. Pat pulled the strings [that got Chris the job].

Sentence (420a) is ungrammatical under an idiomatic reading of the noun *strings*. This means that the interpretation of the constant associated with the verb *get* must be such that it does not have the intension of the constant strings″ in its domain. Under this interpretation, however, we would expect the sentences in (420b) and (420c) to be equally ungrammatical under an idiomatic interpretation. Such an interpretation is, however, clearly available. As Pulman 1993 points out, this problem can be solved if we are willing to assume further lexical ambiguity. In the case of (420c), we would have to consider the verb in the relative clause as being idiomatic in such a way that its interpretation is only defined if it combines with a relative pronoun whose denotation is that of the idiomatic interpretation of *strings*. Similarly, in (420b), there must be an idiomatic interpretation of the verb *get* which takes as its domain the intension of idiomatically pulled idiomatic strings.

As such an additional interpretation must be available for every verb and for every IE which allows for these types of relative clause, there is an explosion of semantic constants and a complex system of domain restrictions. We conclude with Pulman that

> "This is technically possible, of course (though the resulting adjustments to the model theory of Intensional Logic are not trivial), but is beginning to seem a somewhat less than elegant solution." (Pulman, 1993, p. 261)

Finally, we address a third shortcoming of the proposal. Beside the IEs classified as internally regular above, there are also some which are internally irregular. Gazdar et al. 1985 (p. 244, fn. 33) mention the cases of *trip the light fantastic* and *kick the bucket*:

> "Certain idioms are of course semantically unanalyzable, e.g. *kick the bucket* and *trip the light fantastic*. These are to be analyzed as syntactically complex lexical items associated with a single, undecomposable semantic interpretation. From this it follows that their parts are never distributed in complex syntactic constructions ..."

As we can see from this quote, Gazdar et al. 1985 basically assume the same distinction between internal and external (ir)regularity that we introduced in the preceding section. For them, the IEs *kick the bucket* and *trip the light fantastic* are semantically unanalyzable and should therefore be considered syntactically complex lexical items.

It is, however, far from clear how the system of Gazdar et al. 1985 could be extended to handle syntactically complex lexical entries. The problem stems from the fact that in GPSG the grammaticality of a tree should be decidable on the basis of the local trees that it is composed of alone. As there is arguably some internal structure at least to the IE *kick the bucket*, this IE cannot be encoded as a single local tree. Under the reasonable assumption that the internal structure of the VP *kick the bucket* is just like that of other transitive VPs such as that of *read a book* in Figure 7.1, there are three terminated local trees and two non-terminated local trees. For that reason, probably, Gazdar et al. 1985 speak about a "syntactically complex lexical item". It is unclear how the grammar should differentiate formally between those five local trees which need not obey the regular criteria of well-formedness of complex expressions and regular combinations of local trees. Furthermore, under the assumption of complex lexical items, it remains surprising that the IE *kick the bucket* allows adjectival modifiers inside the NP, as attested in the examples in (340), repeated below.

(421) a. Pat kicked the proverbial bucket.
= Pat proverbially kicked the bucket. (Wasow et al., 1983, p. 110f)
b. With that dumb remark at the party last night, I really kicked the social bucket.
= Socially, I kicked the bucket. (Ernst, 1981, p. 51)

We conclude that the approach to IEs in Gazdar et al. 1985 builds on the observation that some IEs are internally regular. These are analyzed as regular combinations of words. The system of Gazdar et al. 1985 is not fine-grained enough to account for the differences in the behavior of internally regular IEs such as *spill the beans* and *pull strings*. In addition, accounting for the limited distribution of the idiomatic reading of parts of a regularly combining IE by a partial meaning function leads to a serious complication in the way semantic denotation is defined. Finally, it is unclear how internally irregular IEs (*kick the bucket* and *trip the light fantastic*) could be handled.

In Chapter 8 we will see that our own proposal will have many similarities to that of Gazdar et al. 1985. In particular, we will make the same distinction between internally regular IEs, which are handled by regular means of syntactic and semantic combination and internally irregular IEs which will be treated as complex lexical entities. We will provide a precise formalization of this notion which is fully integrated into the overall formalism of HPSG. For the internally regular IEs, we will introduce a mechanism to restrict the occurrence of the elements that the IE consists of, such as *beans* and *spill*. While we will follow Gazdar et al. 1985 in assuming new semantic constants for the idiomatic uses of these words, we will not impose restrictions on the interpretation of these constants but on their occurrence within the CONTENT value of a clause. In addition, we will be able to account for the difference between the IEs *spill the beans* and *pull strings* by incorporating syntactic notions into our theory of distributional restrictions.

We will now turn to a proposal made within the framework of *Tree Adjoining Grammar*. There, the opposite perspective on IEs is taken: in general there is no problem with the notion of a syntactically complex lexical entity, as the theory takes all lexical entities to be tree-like. Thus, all IEs are modelled similar to the internally irregular IEs.

7.2. IEs in TAG

Whereas GPSG has made an attempt to capture as much as possible of the behavior of IEs in terms of regular combination, the analysis within the framework of *Tree Adjoining Grammar* (TAG) (Joshi, 1987; Schabes et al., 1988) proposed in Abeillé 1995 takes exactly the opposite perspective.[11] As we will see, TAG takes potentially complex trees as the basic building blocks of linguistic analysis. From this point of view, internally irregular IEs such as *kick the bucket* are just extreme cases of such complex trees. Abeillé 1995 then tries to show how the semantically more transparent and syntactically more flexible IEs can be handled in TAG as well. Her proposal, thus, reduces external irregularity to a special instance of internal irregularity.

Before we can present her proposal in more detail, a short outline is necessary of the framework of TAG and the variant thereof used in Abeillé 1995, *synchronous Tree Adjoining Grammars* (Shieber and Schabes, 1990). In our brief introduction, we will first give an example of a simple TAG for the syntactic analysis of a sentence and then demonstrate how this can be complemented with a second TAG that is used to derive a semantic representation. As the syntactic analysis and the semantic representation of a sentence are derived simultaneously, this framework is called synchronous TAG. The possibility to have

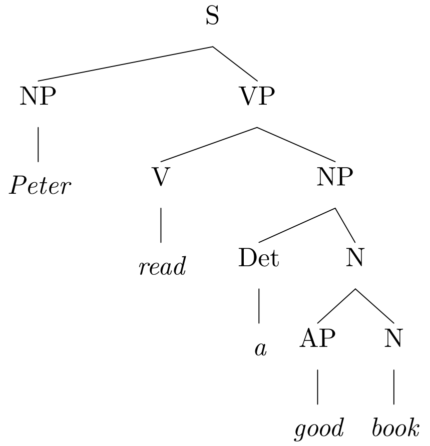
[11]We are grateful to Laura Kallmeyer for knowledgeable advice and comments on this section.

non-parallel syntactic and semantic derivations will play a crucial role in t he analysis of IEs presented in Section 7.2.2.

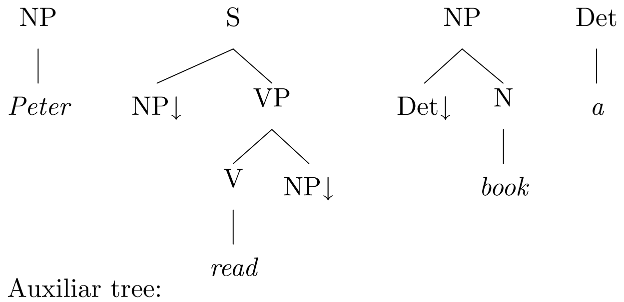
7.2.1. A Brief Introduction to Synchronous TAG. A Tree Adjoining Grammar (TAG) is a set of trees. The trees in the grammar are called the *elementary trees*. They can be combined by two operations (*substitution* and *adjunction*) to form larger trees. The elementary trees come in two varieties: a set of *intial trees* and a set of *auxiliary trees*.[12]

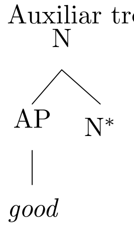
We will take the sentence in (422a) as a concrete example. It will eventually be assigned the syntactic structure in (422b).

(422) a. Peter read a good book

b.


In (423), we give the elementary trees that are needed to analyze the sentence in (422a).

(423) a. Initial trees:


b. Auxiliar tree:


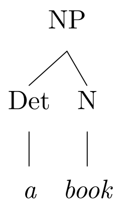
Starting with these elementary trees, we can perform the operations *substitution* and *adjunction* to derive the syntactic structure of sentence (422a). Both operations combine two trees γ_1 and γ_2 in such a way that one node in γ_1 is replaced with the tree γ_2. The difference between the two operation lies in the choice of the node which is going to be replaced: in the case of substitution this node is a leaf of γ_1, in the case of adjunction it is an intermediate node of γ_1.

[12] For an introduction to the formal properties of TAG, see Joshi 1987.

Some of the initial trees in (423) contain no-terminal leaves, such as the node labelled Det in the initial tree for the noun *book*. Nodes like this are called *substitution nodes*. To make it easier to identify the substitution nodes, they are conventionally marked with a downarrow (↓). Given a tree γ with some substitution node n with label X, we can take some initial tree α from the grammar whose root node is also labelled X and replace the node n in the tree γ with the initial tree α.

This can be illustrated with our toy grammar in (423). The initial tree for the noun *book* contains one substitution node, i.e., a leaf which has a nonterminal label: the node labelled Det. In the grammar, there is one initial tree whose root has the label Det: the initial tree for the determiner *a*. We can now combine these two trees by substitution. To do this, we substitute the initial tree of the determiner for the node labelled Det in the initial tree of the noun. This substitution gives us the *derived tree* in (424).

(424)

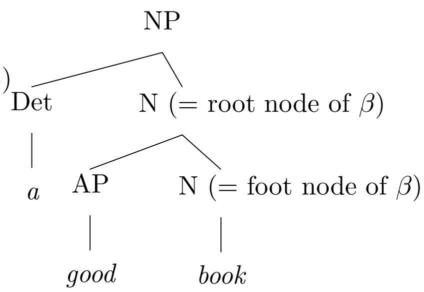


The second way to combine trees is by *adjunction*. Here, auxiliary trees come into play. Auxiliary trees have the special property that they contain one designated leaf. The label of this leaf must be the same as the label of the root node of the auxiliary tree. The designated node is called the *foot node* of the auxiliary tree and it is conventionally indicated by a star (*) following the label of the node. In the grammar in (423), we have just one auxiliary tree, the elementary tree for the adjective *good*. The leaf with the non-terminal label N is the designated node of this tree, as indicated by the star. The label of this foot node is the same as the label of the root node of the auxiliary tree.

We can combine some tree γ with an auxiliary tree β by adjunction. Adjunction proceeds the following way: We need some tree γ which contains a node n which has the label X, and some auxiliary tree β whose root node is also labelled X. We can now replace the node n in γ by the auxiliary tree β and put the structure that was dominated by the node n in γ below the foot node of α.

For illustration, consider the tree for the NP *a book* in (424), which serves as our γ. It contains a node with label N. This node corresponds to the n in our abstract characterization. The node n dominates the leaf with the terminal label *book*. The node n has the label N, which is also the top label of the root node of the auxiliary tree for *good* (i.e., the tree β). We can, now, combine these two trees by adjunction. The result is depicted in (425).

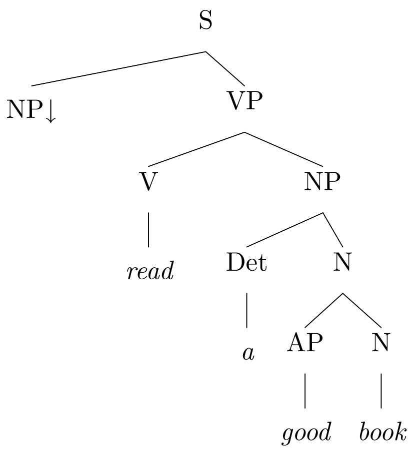
(425)



The tree shows that the node n has been replaced with the auxiliary tree of *good*. The foot node of this auxiliary tree, now, dominates whatever was dominated by the node n in the tree for *the book*, i.e., the terminal symbol *book*.

Given the derivation of the NP *a good book*, it is straightforward to derive the structure of the entire sentence in (422a): We take the initial tree for the verb *read*. It has two substitution nodes, both labelled NP. We can substitute the elementary tree of *book* for the lower NP node in the initial tree of *read*. We will, then add the initial tree for the determiner *a* and the auxiliary tree for *good* to this, in the way illustrated in (424) and (425). The result of these three operations is shown in (426).

(426)



Finally, all that remains to be done is to substitute the initial tree of the name *Peter* for the remaining NP↓ node. This substitution results in the tree given in (422b) above.

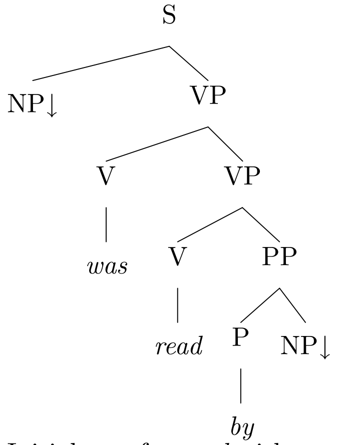
In our little example, we have presented all elements needed for a TAG: two kinds of elementary trees and two operations for combining trees. Before we can present the treatment of IEs within this framework, there are two more issues that must be addressed: First, we will briefly show how syntactic constructions such as passive and topicalization are accounted for within TAG. This is needed for a treatment of the syntactic flexibility of some IEs (and the lack thereof of others). Second, we present how the syntax-semantics interface is handled within TAGs. This latter point needs more discussion as it will involve the extension from simple TAG to synchronous TAG.

In order to account for a larger variety of phenomena without changing the simplicity of the underlying formalism, we introduce more elementary trees. Our study is mainly interested in passive and topicalization. We will also be concerned with relative clauses to a certain degree, as these constructions appeared in the syntactic criteria for regularity of Chapter 6. Let us first consider the derivation of the following three sentences.

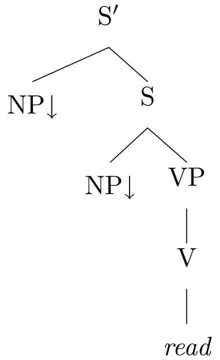
(427) a. Peter read a good book.
b. A good book was read by Peter.
c. A good book, Peter read.

Our little grammar in (423) can easily be extended to account for these sentences. Since in TAG a grammar is nothing but a set of elementary trees, adding two initial trees to the grammar is enough. In (428a) we give an elementary tree for the passive use of the verb *read*. The elementary tree in (428b) encodes a structure in which the direct object of *read* has been extracted.

(428) a. Initial tree for passivized *read*:



b. Initial tree for *read* with extracted direct object:

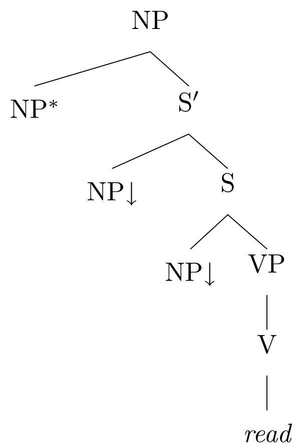


In order to derive the passivized sentence in (427b), we must take the initial tree in (428a). This tree expresses the structure of a passive sentence. As can be seen, initial trees can contain more than one terminal node. In the case of a passive initial tree, we usually find terminal symbols for (i) the passivized verb (*read* in our example), (ii) the passive auxiliary *be* and (iii) the preposition *by*. Given this initial tree, we substitute its upper NP node, i.e., the subject node, by the elementary tree for *book*, to which we can, then, add the trees for *a* and *good* just as illustrated above. Substituting the lower NP node (the NP embedded in the *by*-PP) by the initial tree for *Peter* results in a derivation of sentence (427b).

Similarly, to derive sentence (427c), we start from the initial tree in (428b). This tree need not contain any terminal nodes except *read*. It differs from the initial tree for *read* given in (423) by lacking an NP node in direct object position. Instead, there is an S′ node which dominates the S node. The S′ node immediately dominates the position of the extracted direct object, the S node dominates the VP and the subject position, as usual. Again, if we generate the structure of *a good book* below the upper NP node and substitute the initial tree of *Peter* for the lower, we achieve a derivation of sentence (427c).

As we can see, passive and topicalization are all handled by distinct elementary trees. In general, there must be a distinct elementary tree for every construction in which a given word may occur. Abeillé 1995 calls the set of elementary trees that are associated with a word its *Tree family*. In addition to the trees for the verb *read* given in (423), (428a) and (428b), there will be further trees, expressing the use of *read* in topicalization structures with a fronted subject, in infinitival constructions, in a relative clause, etc. As relative clauses are among the environments included in our syntactic criteria for IEs, we give a relative clause elementary tree for *read*.

(429) Auxiliary tree for *read* as a relative clause:



It should be noted that, in contrast to the other elementary trees for *read*, the tree in (429) is an auxiliary tree, as it will be adjoined to an NP. The internal structure of the relative clause itself is parallel to that of the topicalization structure given in (428b), i.e., the S′ node dominates an S node and the extracted direct object. Further details of the grammar will insure that only NPs which can serve as relative constituents are possible in the upper NP position, and that there is some agreement between the relative constituent and the NP that the relative clause attaches to.

We have discussed four out of numerous elementary trees that are part of the Tree family of the word *read*. As the Tree families of other transitive verbs certainly contain the same kinds of elementary trees that we find for *read*, it seems that there are some regularities in the language that need to be expressed. Work such as Vijay-Shanker and Schabes 1992 has provided some means to avoid the need to state every member of a Tree family for every word separately in a TAG. We do not want to discuss these kinds of generalizations over elementary trees at this point.[13] Instead, we simply assume that there are some techniques to generate the right Tree family to a given expression. In our discussion of IEs in the following section, we will give one elementary tree and note what other kinds of trees are available for a certain expression.

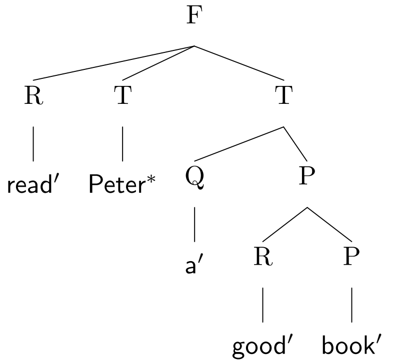
So far we have shown how syntactic structures can be derived given a set of (construction-specific) elementary trees. We did not mention how the appropriate semantics can be associated with the syntactic derivation. The analysis in Abeillé 1995 makes use of the formalism of *synchronous* TAG as presented in Shieber and Schabes 1990. This formalism can be used to achieve a semantic representation built parallel to the syntactic derivation. The underlying idea of synchronous TAG is that two structures are constructed simultaneously. In the present context, these will be a syntactic structure and a semantic representation, i.e., a logical form.

To return to our example in (422a), we now want the grammar to derive not only the syntactic structure in (422b), but also the semantic representation given in (430).[14]

[13]We will address a related issue in our discussion of the HPSG approach to IEs in Riehemann 1997.

[14]Note that the star in Peter* is part of the terminal symbol of our semantic representation language (see footnote 5), and does not indicate that the node is a foot node.

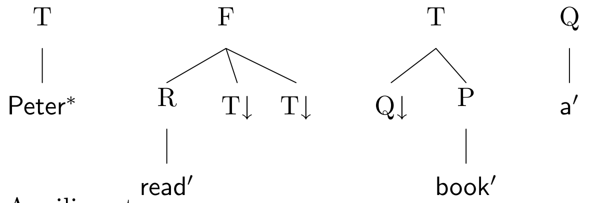
(430)



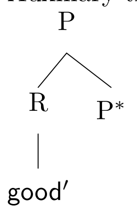
In order to avoid complications with quantifier scope, we assume semantic representations as those given for GPSG in Section 7.1, i.e., we assume that verbs take quantified NPs as arguments. Thus, we depart in various respects from the examples given in Shieber and Schabes 1990, but not from the formalism they introduce. We assume that, with the exception of the symbol R, each nonterminal in a semantic tree is a semantic type. We take F(ormula) to be the type $\langle s, t \rangle$ (or S in Gazdar et al. 1985). The label T(erm) is used for quantified NPs, i.e., expressions of the type abbreviated as NP in GPSG. For quantifiers (type *Det* in GPSG), we use the symbol Q. Finally, we use the label P(roperty) for the semantic type of nouns, i.e. $\langle s, \langle e, t \rangle \rangle$ (or *Nom* in Gazdar et al. 1985). The label R(elation) is a functor which is assumed to have just the right semantic type to combine with the semantic types of its sisters to yield an expression of the type required by the mother node. With these assumptions, the semantic representation in (430) is just a notational variant of that used in GPSG; it is straightforward to rewrite it as read′(a′(good′(book′)))(Peter*).

The semantic representation can be derived via a TAG which contains the following elementary trees.

(431) a. Initial trees:



b. Auxiliary trees:

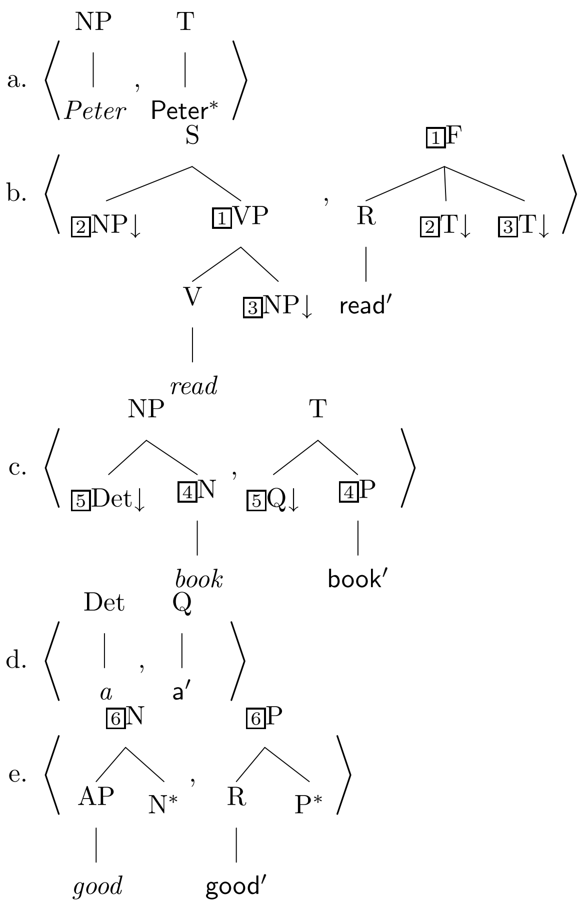


Taken alone, this set of elementary trees can derive the semantic representation given in (430): starting from the initial tree for the constant read′, we can substitute the intial tree of Peter* for the first T node from left. Consequently, we replace the second T node by the tree for book′. In the resulting tree, there is a substitution node Q↓ which we replace by the initial tree of the quantifier a′. Finally, we adjoin the auxiliary tree of the modifier good′ to the P node that dominates book′.

This derivation of the semantic term is parallel to the derivation of the syntactic structure as shown above. Building on this parallelism, it is the basic idea of synchronous TAG to synchronize these two derivations, i.e., to derive the syntactic and the semantic structure simultaneously. In order to achieve this, a synchronous TAG consist of pairs of elementary trees, instead of single elementary trees. Each of these pairs contains a syntactic and a semantic elementary tree. In our case, clearly, the elementary tree for the terminal symbol *read* in (423) and that for the terminal symbol read′ in (431) would be paired together. The other pairings are also made in the intuitive way.

In a synchronous TAG nodes in two paired trees may also be *linked* to each other. In (432), we state the synchronous TAG that arises under the intuitive combination of the syntactic TAG in (423) and the semantic TAG in (431). Following the convention of Shieber and Schabes 1990, we indicate links between nodes by boxed integers.[15]

(432)



The new set of pairs of elementary trees is a synchronous TAG. The boxed integers show which nodes on the syntactic side correspond to which nodes on the semantic side. For example, the highest NP in the elementary tree for *read* corresponds to the first semantic argument of the relation read′ in the semantic tree. This is expressed by the boxed integer 2 which precedes the linked nodes in both trees in this pair.[16]

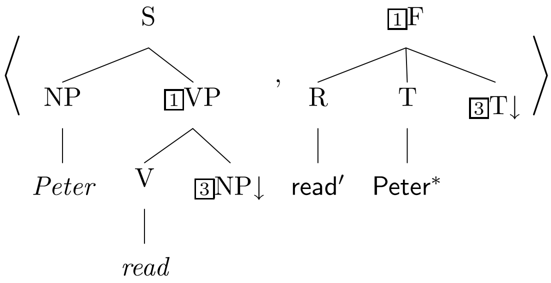
Given this grammar, we can execute the synchronous derivation of the syntactic and semantic representation of example (422a). Links are crucial in the derivation, as tree-combining operations must be executed on both trees in a pair. Assume that there is a pair of trees $\langle \alpha_1, \alpha_2 \rangle$ which we want to combine with a pair of elementary trees $\langle \beta_1, \beta_2 \rangle$, then we can only substitute/adjoin β_1 for/to a node n_1 in α_1, if there is a node n_2 in α_2 which is

[15]Note that in synchronous TAG, boxed integers are used to express links, not identity (or description language variables) as in HPSG.

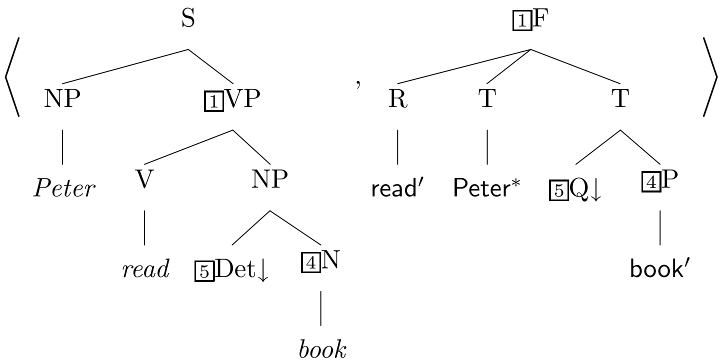
[16]For a larger fragment, it might be necessary to add more links to the elementary trees in (432). For example, to allow for the attachment of relative clauses to the NP node, we would need another link from the NP node in the syntactic tree in (432c) to the P node in the semantic tree.

linked to n_1 and for/to which we can substitute/adjoin the tree β_2. The derivation of our example sentence will make clear how this works.

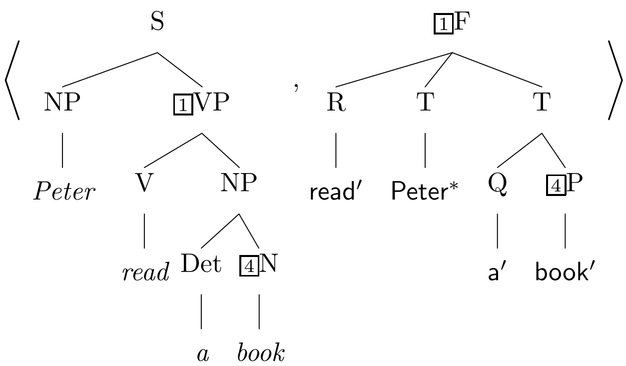
Let us first take the pair of initial trees given in (432b). In this pair, there are substitution nodes that are linked to each other by the boxed integer $\boxed{2}$. The root nodes of the trees in the pair (432a) have the right labels to be substituted for the substitution nodes in (432b). This substitution results in the tree given in (433)

(433)


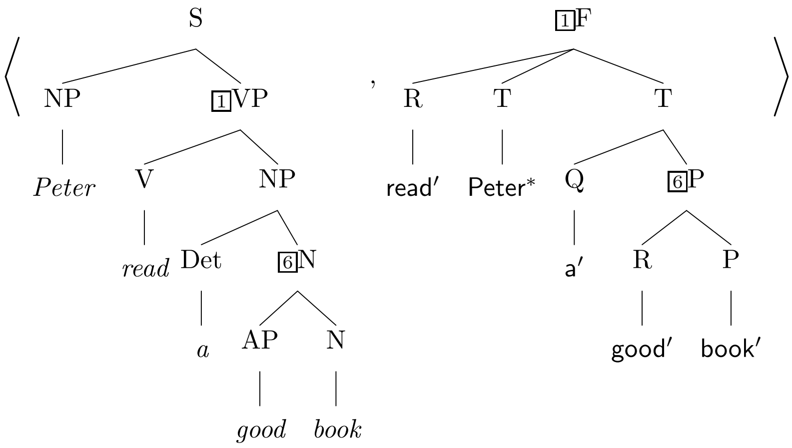
Note that in the resulting pair of trees, all links are preserved except for the one that has been used for the combination of the trees. In the next step, we choose the nodes linked by the boxed integer $\boxed{3}$ in the trees in (433) and substitute for them the trees contained in the pair (432c). The result of this substitution is given in (434).

(434)


After this substitution, there is one more pair of substitution nodes in the resulting pair of trees: the nodes linked by the boxed integer $\boxed{5}$. We can use the pair of elementary trees given in (432d) to execute the substitution. The result is shown in (435).

(435)


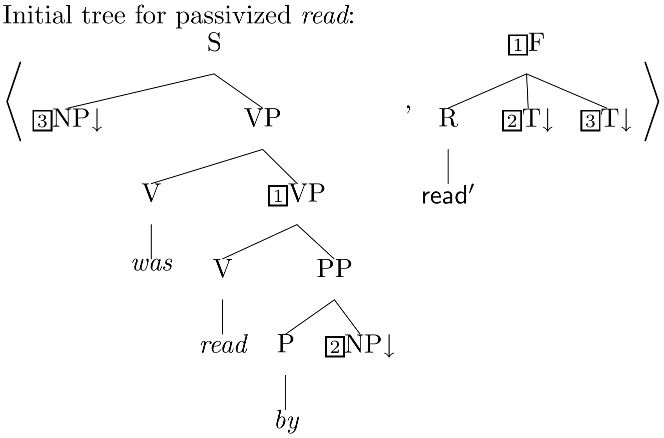
Finally, we can add the auxiliary trees in (432e) to this pair of trees by adjunction to the nodes linked by the boxed integer $\boxed{4}$.

(436) 

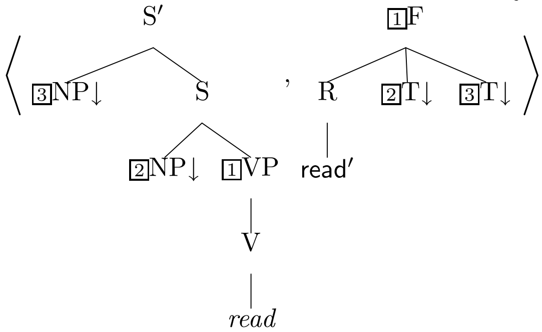
When we combine the trees in (435) with those in (432e), we use the link 4. But, as the trees in (432e) contain the link 6, this link appears in the overall structure in (436). Thus, in the resulting structure, there are still two links, 1 and 6, which indicate that this pair of trees may be further used for adjoining other modifiers to the N node that dominates *good book* or to the VP.

The treatment of passivization and topicalization in synchronous TAG is straightforward; we need only introduce pairs of elementary trees for passive, topicalization and relative clause formation. On the syntactic side, we take the trees specified in (428) and (429). Under the assumption that both passive and topicalization do not contribute any meaning, the elementary semantic tree for both syntactic trees in (428) will be the tree given for read′ in (431). For the relative clause it is necessary to introduce a new semantic tree: Since the elementary tree of the relative clause construction is an NP, the semantic tree is labelled T and merely embeds the formula that contains the predicate read′. We will not specify the details of the semantic treatment of relative clauses, but briefly consider the pairs of elementary trees needed for passive and topicalization.

(437) a. Initial tree for passivized *read*:



b. Initial tree for *read* with extracted direct object:



In both trees, there are three links: the VP node is linked to the semantic F node by $\boxed{1}$. In the passive tree, the NP inside the *by*-PP is linked to the first semantic argument of the predicate read′ by $\boxed{2}$. Finally, there is a link between the passive subject and the second semantic argument of read′, indicated with the boxed integer $\boxed{3}$.

In the pair of trees in (437b), the highest NP, i.e., the topicalized constituent is linked to the second argument of the semantic predicate, and the subject NP is linked to the first argument of read′ at the semantic side.

With these brief remarks on passive and topicalization within synchronous TAG, we want to close the introduction to the framework used in Abeillé (1995). We did not present all aspects of her framework, but concentrated rather on those that will be relevant for the following discussion. In particular, we did not address the fact that Abeillé assumes that the nodes in the syntactic trees are enriched with feature structures which serve to determine agreement and case government as well as other phenomena.[17]

7.2.2. A TAG Analysis of IEs. We can now discuss the specific analysis of IEs presented in Abeillé 1995. There are, however, two ways in which we will depart from Abeillé's analysis. The analysis of IEs in Abeillé 1995 is presented on the basis of French examples. In the present section, we will stick to the English IEs introduced in Chapter 6 and try to apply Abeillé's approach to these data as truthfully as possible. In addition, Abeillé 1995 is neutral with respect to a particular semantic representation language. We will simply assume the language that we have used in the presentation of synchronous TAG, i.e., a tree structure representation of IL terms.

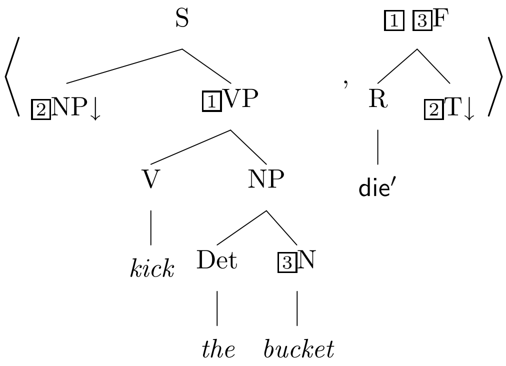
Our presentation of the framework of synchronous TAG has revealed the following properties which are important for the analysis of IEs in Abeillé 1995: (i) Within TAG, trees are the basic (and only) available data structure. This has two consequences: first, a given lexical entry will not specify a terminal node in a tree, but an entire tree; and second, there are no phrase structure rules. (ii) There is a Tree family of elementary trees associated with every word which specifies all the constructions in which the word may appear.

In contrast to the approach of Gazdar et al. 1985, Abeillé 1995 focusses on the irregular aspects of IEs. For her, independent of any alledged regularity that can be found within IEs, all IEs are treated as units.[18] For such an approach, an IE such as *kick the bucket* is an ideal example. In (438), we give the pair of elementary trees for this IE.[19]

[17]In fact, each node has two such feature structures: an "upper" and a "lower" feature structure. See example (439c) and footnote 20 for an application of this idea.

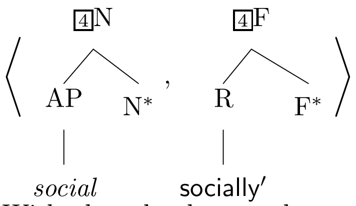
[18]With the exception of light verb constructions. See below for details.

[19]The trees in (438) are taken from Shieber and Schabes 1990 (p. 256). We have only added the links $\boxed{1}$ and $\boxed{3}$ to account for the modification data, incorporating the analysis of French idioms in Abeillé 1995.

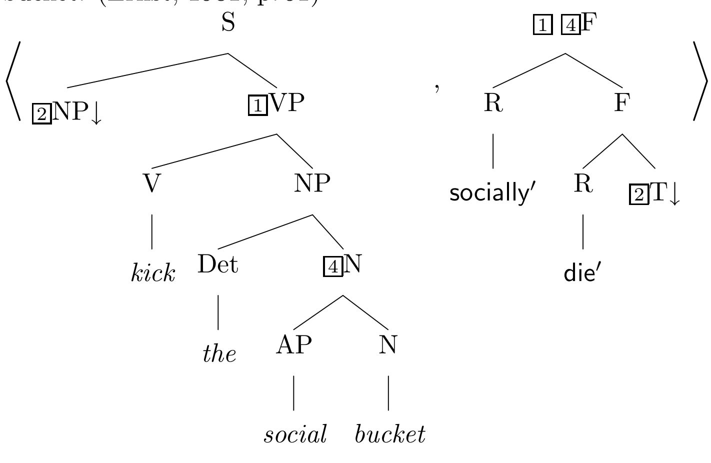
(438) 

The pair of trees in (438) expresses all the properties of the IE *kick the bucket*. To illustrate this, we check the list of criteria applied in the previous chapter. First, the parts of the IE cannot be attributed a meaning. This is achieved by the fact that the IE is stated as a whole instead of putting it together from smaller elementary trees. Second, the meaning is not derived in a combinatorial way, because it is given in an elementary tree, i.e., a tree which is not the result of a derivation. Crucially, in synchronous TAG, nothing forces the syntactic tree and the semantic tree in a pair to be of a similar structure. Thus, the syntactic tree in (438) is that of a sentence headed by a transitive verb, whereas the semantic structure has the form normally associated with intransitive sentences.

Third, there cannot be any internal modification of the nominal part of the IE. This is achieved by the fact that the N node, which would host syntactic adjunction, is linked to the F node in the semantic structure. Adopting the example of Abeillé 1995 (p. 36), we assume the elementary tree for *social* in (439a) to adjoin to the noun *bucket* in syntax, as attested in sentence (439b). In the pair of initial trees for *kick the bucket*, there is a link, 3, between the N node in the syntactic structure and the F node in the semantic representation. The semantic effect of external modification is achieved by using this link for the adjunction of the auxiliary tree for *social*. The resulting pair of trees is shown in (439c).

(439) a. 

b. With that dumb remark at the party last night, I really kicked the social bucket. (Ernst, 1981, p. 51)

c. 

In the pair of trees in (439c), the link $\boxed{3}$ does not re-occur as it was used for the adjunction. Instead, the link $\boxed{4}$ now links the upper N node in the syntactic structure to the upper F node in the semantic representation. The link $\boxed{1}$ re-appears on the upper F node in the semantic structure.[20] The adjective *social*, while syntactically attached inside the NP, semantically modifies the entire expression.

While Abeillé 1995 contains explicit suggestions for how to account for the behavior of the IE with respect to the first three semantic criteria, she does not elaborate on the fourth semantic criterion, pronominalizability, (Abeillé, 1995, p. 24). For this reason, we will ignore this criterion in the present section.

Let us turn to the syntactic criteria. Since all the terminal nodes are explicitly given in the syntactic structure in (438), it is not clear whether the word *kick* as it appears in the IE has anything in commun with the normal word *kick*.[21] Even though the first syntactic property of the IE is not fully accounted for, the syntactic structure in (438) is exactly that found in other sentences with a transitive verb, i.e., the second syntactic observation is captured. The fact that there is a link ($\boxed{3}$) at the N node makes it open for syntactic modification, thus accounting for the IE's behavior with respect to the third criterion.

The fourth, fifth and sixth syntactic criteria involve the possibility of undergoing passivization, extracting the NP part of the IE, and having the nominal part of the IE occur as a term that is modified by a relative clause which contains the rest of the IE. As we have seen in the discussion of these three constructions in TAG, the availability of a passive, a topicalization or a relative clause depends on whether a certain elementary tree is part of the grammar or not. All that is needed to block the IE from occurring in any of these constructions, is to assume that there are no such additional elementary trees in the Tree family of this IE. Thus, as indicated by the asterisk, the sentence in (440) cannot have an idiomatic interpretation, simply because there is no initial tree in the grammar that it could possibly be built from.

(440) * The bucket he kicked.

Of course, the literal meaning can be derived by starting with an elementary tree for the verb *kick* that, in all relevant respects, looks like the topicalization structure given in (437b).

Note also that the analysis does not run the risk of an idiomatic *kick* or *bucket* to occur outside the IE, because these words do not exist independently in the grammar, i.e., they do not have a Tree family of their own, but occur exclusively in the Tree family of the idiomatic expression *kick the bucket*.

After having considered the IE *kick the bucket* in some detail, we can now turn to the syntactically more flexible and semantically more transparent IE *spill the beans*. The pair of elementary trees for the canonical use of this IE, i.e., in active voice and a simple tense form, is given in (441a). In (441b), we indicate which further elementary trees will be needed to account for its syntactic behavior.[22]

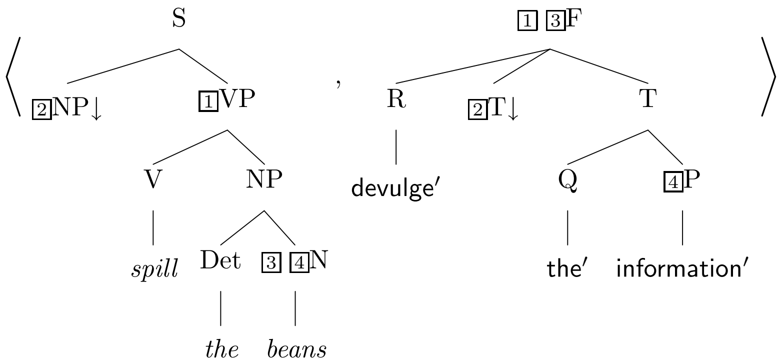
---

[20] We have not explained in detail how this is achieved. But, in general, each node has "upper" properties and "lower" properties. If a node is doubled by an adjunction, just as the F node in (438), the "lower" properties stay with the lower node, whereas the "upper" properties appear at the higher copy of the node. Assuming that the link $\boxed{1}$ is part of the "upper" properties of the F node ensures that it will appear at the top node in (439c).

[21] In the case of the word *bucket*, this lack of a connection to the normal word *bucket* might be etymologically correct (see footnote 8).

[22] The trees in (441) reflect the judgments given in Section 6.2. For more permissive speakers, as those assumed in Wasow et al. 1983, the Tree family contains auxiliary trees for topicalization and relative clauses as well. For less permissive speakers (Schenk, 1995), the link $\boxed{4}$ would be missing, and, possibly, even the link $\boxed{3}$.

(441) a.



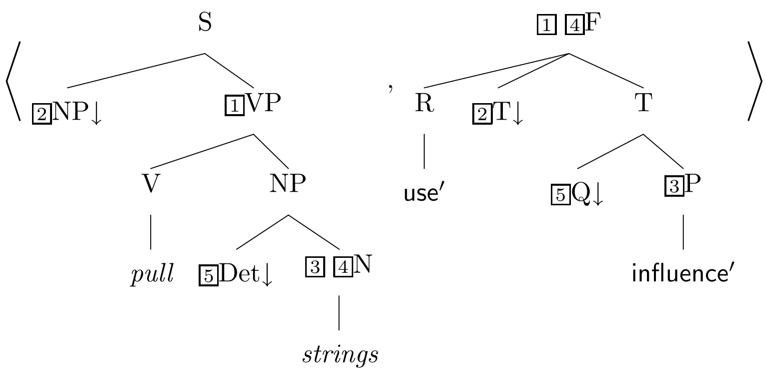
b. Further elementary trees: passive yes
topicalization no
relative clause no

The elementary trees for *spill the beans* differ from those of *kick the bucket* in various respects: First, the semantic tree expresses a two-place relation, which makes it look like the semantic tree normally found in the case of a transitive verb. In addition, there is a link ([4]) from the N node that dominates the noun *beans* to the P node dominating information′. This link is used in cases of semantically internal modification. For semantically external modification, the link [3] can be used in the way we illustrated in (439). In (441b) it is indicated that a corresponding passive elementary tree exists, which accounts for the greater syntactic flexibility of the IE. There are, however, no trees for topicalized or relative clause structures.

Given the elementary tree for *spill the beans*, we can compare this Abeillé-style analysis to that of Gazdar et al. 1985 presented in Section 7.1. Treating the entire IE as one entity makes it possible (but not necessary) to assume the regular semantic constants divulge′ and information′ instead of the highly restricted constants spill′′ and beans′′. More generally, Abeillé's proposal is free of the problematic assumption of restricting the domain of the interpretation of some semantic constants in such a way that only the right constants are successfully combined. To achieve this, however, the entire IE must be stated as a unit and the compositional aspects of the combination are lost.

Next, we want to address the expression *pull strings*. Again, we give one pair of elementary trees (442a) and indicate which additional elementary trees are available (442b).

(442) a.

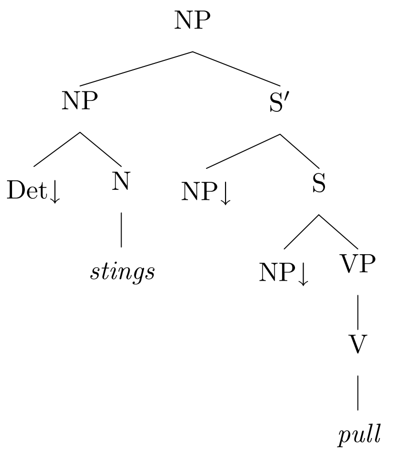


b. Further elementary trees: passive yes
topicalization yes
relative clause yes

The elementary trees in (442a) have more non-terminal leaves than those of the IE *spill the beans*. In particular, the determiner is not fixed, but given as a substitution node which is linked to the quantifier node in the semantic tree by [5]. This additional substitution node accounts for the greater freedom of this IE with respect to the determiners that may occur.

As can be seen in (442b), the Tree family of the IE contains an elementary tree for a relative clause in which the direct object takes the form of a relative pronoun. The resulting elementary tree does, however, differ from that given in (429). In the case of the verb *read*, the resulting tree is an auxiliary which attaches to an NP. For the IE, we must give the full form of the NP to which the relative clause attaches, since the terminal element *strings* would otherwise be lost. In (443) we give the relevant syntactic tree.

(443)



This elementary tree is not an auxiliary tree, as it does not contain any foot node. Thus, it lacks one of the most important properties of the elementary tree for regular relative clauses. For this reason, a relative clause made from an IE is different from a relative clause that is part of the Tree family of a regular verb. We conclude that while this approach accounts for the distribution of the IE, it fails to treat the relative clause formation potential as part of the regular behavior of the IE.

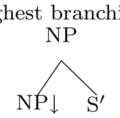
However, this problem is specific to relative clauses and does not arise with other syntactic constructions such as passive, topicalization or clefts (Abeillé, 1995): in all these constructions, there still is one node that directly corresponds to the direct object node in the tree in (442), i.e., the relevant information about the direct object node simply occurs at a different place in the structure. Only in the case of a relative clause, a foot node appears instead of a substitution node. Thus, the kind of node is changed. This does no harm in the case of free combinations, but it leads to a difference between the analysis of free combinations and IEs.[23]

Next we can consider the two IEs, *make headway* and *make a decision*. Both IEs are instances of light verb constructions. Abeillé 1988 (section 3) presents an analysis of French light verb constructions in terms of elementary trees. As Abeillé 1988 is only concerned with the syntax, no semantic trees are given.

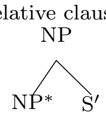
Abeillé 1988 assumes that for every noun that can appear in a light verb construction, there is a special elementary tree for the light verb construction, in addition to the regular elementary tree for the noun. In (444) we give these trees for the English noun *decision*.

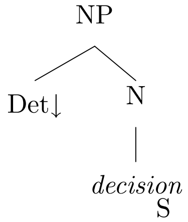
[23]This difference disappears if one assumes a substitution node for ordinary relative clauses as well, i.e., the highest branching in such a relative clause would be like in (i) instead of being like in (ii).

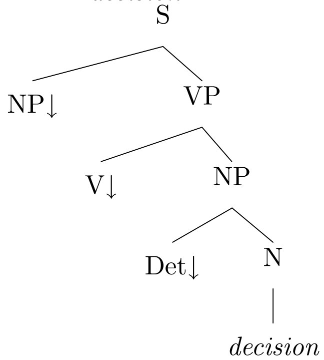
(i)



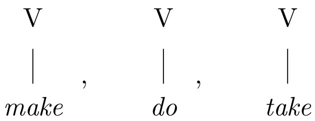
(ii)



(444) a.


b.


The interesting feature of this analysis is that the verb in a light verb construction is introduced via a substitution node, i.e., the elementary tree in (444b) is not that of a verb but that of a noun. Therefore, the class of light verbs can be identified by special elementary trees: verbal trees that do not project up to an S node. In (445), some such trees are given.

(445)


This ingenious treatment of light verbs runs the risk of overgeneration: Even if we consider just the three light verbs in (445), the proposal is unable to account for the fact that the noun *decision* allows only the verbs *make* and *take* in a light verb construction, but not the verb *do*.[24]

Beside this overgeneration problem, we think that the elementary trees for the support verbs are not in line with the basic concept of elementary trees. Abeillé 1995 (p. 26) lists some criteria that must be satisfied by elementary trees. Among these there are the requirement for *predicate-argument co-occurrence* and the requirement for *semantic consistency.* She formulates these requirements as follows:

> "**Predicate-argument co-occurrence:** Elementary trees must correspond to complete argument structures. The elementary tree(s) associated with a given predicate must comprise one node (a foot node or a sustitution node) for each of its arguments. Subjects and complements both belong to the elementary trees of their predicate, as do both fillers and gaps.
> **Semantic consistency:** Each elementary tree corresponds to one non-vacuous semantic unit. This principle excludes functional elements such as case-marking prepositions as autonomous lexical anchors, if they are semantically vacuous: They are thus defined as coanchors and belong to the same elementary tree as the corresponding predicate(s) ...."

According to the second requirement, if there are elementary trees for the light verbs, these verbs must be non-vacuous semantic units. It is very likely that the light verbs are interpreted as semantic functors that transform a noun-like semantics into a verb-like

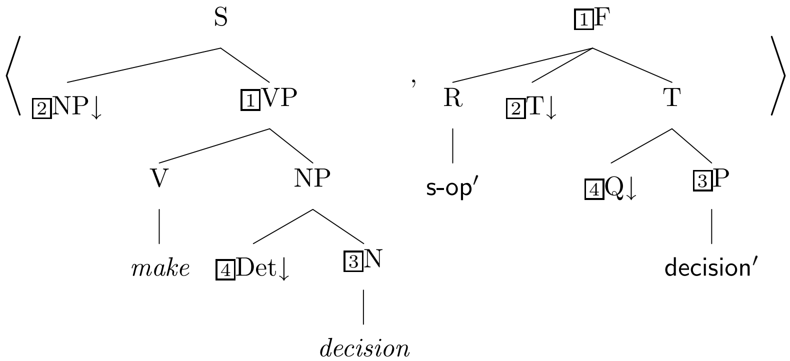
[24]The grammar presented in Abeillé 1988 is intended for parsing. Thus, the problem of overgeneration might not be relevant for the purpose of that paper.

semantics. If they are functors, the first requirement forces their arguments to be present as a substitution or a foot node in the elementary tree. However, this is not the case in the elementary trees in (445).

On the other hand, we could assume that the semantic functor is part of the light-verb construction tree of the noun. In this case, the predicate-argument co-occurrence condition would be satisfied, but there would be no semantic unit expressed by the light verb. Thus, the elementary trees for the light verbs would violate semantic consistency. We conclude that the treatment of light verb constructions in Abeillé 1988 is conceptually problematic, given the standard assumptions about elementary trees made in TAG.

This problem can be avoided, if the light verb construction as a whole is treated in the way proposed for idioms in Abeillé 1995. In this case, we will find the pair (446a) of elementary trees in the grammar. For simplicity, we assume a semantic constant s(upport)-op(erator)′ which is an operator that turns a noun-like semantics into a verb-like semantics.

(446) a.



b. Further elementary trees:

passive	yes
topicalization	yes
relative clause	yes

This idiomatic analysis captures the behavior of the light verb construction. In particular, the idiosyncratic selection of the light verb is no longer a problem.

There is a fundamental conceptual difference between this analysis of light verb constructions and the one that was originally conceived in Abeillé 1988: for Abeillé 1988, the light verb construction tree would be automatically generated for every noun in the lexicon.[25] Under an IE analysis such as the one proposed in (446), we are forced to assume that for every light verb-noun combination a separate elementary tree must be stated explicitly in the grammar. We cannot generate the elementary trees for the light verb construction automatically, because it is not predictable which noun will demand which light verb.[26]

[25] In Abeillé 1988 the basic motivation for this analysis comes from the fact that extraction out of an NP is only possible if the NP appears in a light verb construction. As she needs the structure merely for the purposes of extraction, there is no need to assume it in cases where the noun takes no complements.

> "We consider all nouns taking complements as having corresponding support verbs that they subcategorize." (Abeillé, 1988, p. 11).

In that case, however, the overgeneralization problem is even stronger because certain verb-noun combinations would not be recognized as a light verb construction. To consider a concrete example, take the noun *shower*. It certainly does not take arguments. Still it requires the light verb *take* to appear in a light verb construction (*take a shower*). If this combination is not analyzed as a light verb construction, we must assume a regular verb *take* with a light verb meaning. In that case, we would have a similar ambiguity for the light verb *do* because of the combination *do the dishes*. Then, again, we are not able to exclude a light verb construction reading for combinations such as **do a shower* and **take the dishes*.

[26] Laura Kallmeyer (p.c.) points out that there are ways to avoid the overgeneralization problem. For example, it is possible to give every noun an attribute whose value indicates which particular light verb it requires. Given such an attribute, it is possible to generate elementary trees as those in (446) automatically, where the particular choice of the terminal symbol dominated by the V node is determined by the value

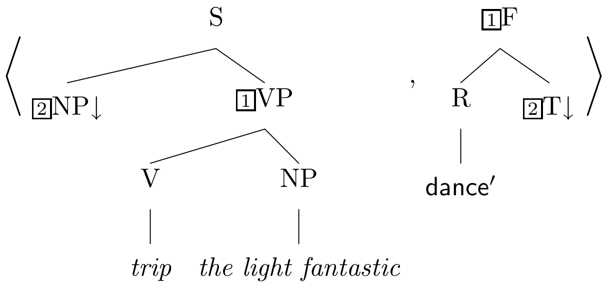
We can now address another of the IEs discussed in Chapter 6: *make headway*. This IE clearly shares the properties of a light verb construction. In addition, it is irregular in so far as the noun *headway* does not occur in the language outside this construction. We can propose an analysis which is just like that of *make a decision* in (446a), the only difference being that, whereas there exists an elementary tree for an NP headed by the word *decision*, there is no such elementary tree for the noun *headway*.

The discussion of light verb constructions is particularly enlightening as the proposal in Abeillé 1988 assumes more regularity for these constructions than for idioms. Still, as we have shown, this regularity must be given up to avoid overgeneration and to avoid the violation of basic conceptual principles that underly the idea of what an elementary tree should be.

So far, we have not discussed the IE *trip the light fantastic*. All the expressions considered in Abeillé 1995 are built according to a regular syntactic pattern of the language. This is certainly not an oversight, as Abeillé 1995 (p. 31) explicitly assumes that "idiomatic elementary trees follow the regular syntactic rules of the language". As we have argued above, this need not be the case for *trip the light fantastic*.

It is fairly unclear how IEs of this kind can be handled in TAG. Certainly, the entire IE would be expressed with a single elementary tree. But what exactly should be the structure of this tree? It can of course be very idiosyncratic, as nothing excludes a particular structure from being found in only one elementary tree in the entire langugage. Still, we are forced to decide which structure should be assigned to the IE in question. For practical reasons, any structure would do. Such an assignment would, however, fail to acknowledge the fact that English native speakers have no intuition on the internal structure of this IE. Alternatively, we could assume a syntactic tree which contains a leaf which has a sequence of terminal symbols instead of a single terminal symbol.[27] Under such an analysis, the pair of elementary trees for the IE would be as in (447).

(447) a.



b. Further elementary trees:

passive	no
topicalization	no
relative clause	no

In the elementary trees in (447a), there are only links for adding an adjunct to the VP (the link [1]) and the subject (the link [2]). Thus, the IE is even less regular than the IE *kick the bucket*, as we do not allow for external modification. Furthermore, as indicated in (447b), there are no other elementary trees in the Tree family of this IE. So far, all is in line with the analysis of Abeillé 1995 for IEs which are not flexible but built in a regular syntactic way. We do, however, consider it a problematic aspect of any TAG analysis that we are forced to assign the VP a particular structure, even if the structure is as indicated

of this attribute. A solution along these lines, certainly captures the regularity of the construction better than the fully idiomatic analysis given in the main text. Its adequacy depends, however, on the question how well-motivated such an attribute is.

[27]We are grateful to Laura Kallmeyer (p.c.) for suggesting this solution to us.

in (447a). This does not match with the perception of native speakers, who seem to have no clear intuitions about the internal syntactic structure of this IE.

After our hypothetical application of the TAG analysis of French IEs in Abeillé 1995 to the English data of the preceding chapter, it is time for some summarizing remarks.

As trees are the smallest syntactic units in a TAG, it is a natural assumption that some elementary trees are less flexible than others, i.e., more specific than others. In addition, the framework of synchronous TAG, while providing a simultaneous construction, is not forced to assume a parallel structure in both the syntactic and the semantic derivation. This leads to an extremely elegant account of IEs such as *kick the bucket* where the syntactic material is fixed and the semantic structure is considerably different from the syntactic tree. Furthermore, the analysis shows that there is a gradual difference in the flexibility of IEs which can be easily seen by the number of substitution nodes and links available in an elementary tree and the size of the Tree family.

This approach encounters conceptual problems once an IE cannot be captured within a regular syntactic structure. In the preceding discussion, we have seen three instances of this problem: First, certain IE such as *trip the light fantastic* are not constructed according to the general rules of English syntax. Second, relative clauses in which part of an IE takes the form of a relative pronoun require a different kind of elementary tree than normal relative clauses. Third, in cases where expressions seem to be more flexible such as light verb constructions, the TAG architecture forces them into the same structure that is found in more irregular expressions.

We want to elaborate this third point a little further. The problematic aspect of the analysis was evidenced by the more compositional treatment of light verb constructions in Abeillé 1988. As we have shown, such a freer treatment cannot be maintained. Yet, Abeillé 1995 (p. 39) states in her concluding remarks:

> "Although the TAG formalism itself does not require all idioms to be noncompositional, it offers a natural representation for the flexibility of noncompositional idioms."

While Abeillé is certainly right in the second part of this quotation, we have doubts about the first part. We certainly must understand the term "compositional" in this quote to refer to the way the parts of an IE combine. In this sense, IEs which have substitution nodes and links in their elementary trees are treated more compositionally than IEs which contain all the terminal symbols. As we have shown for light verb constructions, the TAG approach is forced to give a noncompositional analysis, i.e., including a substitution node for the light verb leads to overgeneration problems, because the particular light verb that is required cannot be explicitly selected.

In our own HPSG analysis of IEs in Chapter 8 we will combine the techniques and insights of both the GPSG and the TAG approach to IEs. The most important insight that we are going to adopt from the TAG approach is that, at least in some cases, the basic unit of description cannot be the word, but must be the entire phrase. We will adopt from the GPSG approach to stick to words as minimal entitities in the lexicon as far as possible. In order to do this, we will provide some mechanism to express the special kind of occurrence restrictions found within the syntactically and semantically freer IEs. In sum, this means that our treatment of IEs such as *kick the bucket* will follow the TAG analysis, whereas we will give an account similar to that of Gazdar et al. 1985 for those IEs that show a fair amount of syntactic and semantic flexibility. Before we can turn to our own approach, however, we will first address the HPSG proposal of Riehemann 1997.

7.3. IEs in Constructional HPSG

In the present section, we discuss the approach of Riehemann 1997 and Riehemann and Bender 2000.[28] Riehemann's theory of idiomatic expressions is expressed within HPSG, but builds on basic architectural assumptions which differ significantly from those that underly our own proposal. In general, Riehemann takes all IEs as units and, thus, patterns with the TAG analysis presented in the previous section. As IEs are not words, but phrases, Riehemann treats them as special instances of *constructions*, i.e., as special subsorts of the sort *phrase*. In addition, Riehemann relies on a particular approach to semantic composition, called *Minimal Recursion Semantics* (Copestake et al., 1995, 1997). All parts of the basic architecture of grammar which underlies Riehemann's analysis were proposed on independent grounds to account for phenomena not related to IEs, or should be useful for the account of other phenomena. As we will see, Riehemann uses this machinery to achieve a local treatment of all IEs, i.e., an IE can be fully described in its relevant properties without considering its internal syntactic structure.

In this section, we will first introduce the necessary technical notions that Riehemann presupposes and then describe her treatment of IEs in more detail. As the kind of HPSG theory that Riehemann proposes differs significantly from the one used in this thesis, we will refer to it with the term *constructional HPSG*. In our presentation of constructional HPSG, we will point out where these differences are. When we discuss the approach to IEs of Riehemann 1997, we will see that her focus on the constructional aspects leads to problems in the account of the distributional constraints of parts of an idiom.

7.3.1. A Brief Introduction to Constructional HPSG. In this section, we will point out three major differences between the approach to HPSG taken in Riehemann 1997 and in the present thesis. The differences lie in the overall architecture of the grammar. First Riehemann assumes a constructional perspective on HPSG. Second, Riehemann proposes a new attribute on each phrase that collects all the words dominated by the phrase. Finally, the semantic formalism assumed in Riehemann 1997 is very different from the one presented in Chapter 4 of this thesis and requires some explanation. In this section, we will address these differences in turn. Although we think that in most cases it is not an empirical question whether the architecture proposed in this thesis or the one followed in Riehemann 1997 is more adequate, we want to explain where the differences lie.

The main difference between Riehemann's perspective on HPSG grammars and the one taken in this thesis is that Riehemann assumes most generalizations to be captured in elaborate hierarchies of signs. In Pollard and Sag 1994, the sort *sign* has two subsorts, *word* for non-recursive signs and *phrase* for phrasal signs, i.e., signs which embed other signs via a syntactic dominance relation as expressed by some *sign*-valued attribute(s). In the HPSG stream of research which Riehemann 1997 is part of, both these subsorts of *sign* have received a huge number of subsorts. The sort hierarchy below *word* can be called the *lexical hierarchy*, the subsorts below *phrase* are usually referred to as *constructions*.[29] In order to be able to differentiate terminologically between the approach to HPSG taken in this thesis and that in Riehemann 1997, we have introduced the term *constructional HPSG* for the

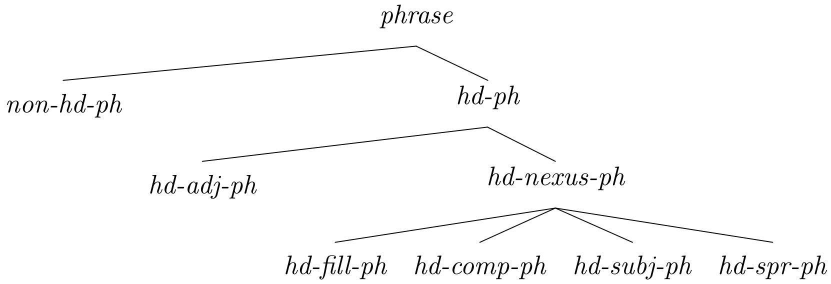
[28]At the time of writing, Riehemann (2001) was still in preperation. We confine our discussion to the papers mentioned in the main text, but see footnote 42 for a short discussion.

[29]In this exposition, we simplify considerably, as it is not clear whether it is adequate to treat the lexical hierarchy as a sort hierarchy. Such an interpretation is possible for the hierarchy proposed in Davis 1997. Alternatively, and more in the spirit of Pollard and Sag 1987, the lexical hierarchy is sometimes considered a hierarchy of descriptions. Unfortunately, there is no extension of any HPSG formalism that includes the concept of a hierarchy of descriptions. Therefore, such an interpretation of the lexical hierarchy would fall outside the HPSG grammar proper. For the purpose of this thesis, we can assume that the lexical hierarchy is just the sort hierarchy below *word*. Under this interpretation, it has a formally clear status.

latter. The constructional approach to HPSG goes back to Pollard and Sag 1987 for the lexicon, and to Sag 1997 for phrasal constructions.

Sag 1997 provides an elaborate sort hierarchy below *phrase*. Part of this sort hierarchy is given in (448). In this hierarchy, the sort *phrase* has subsorts which correspond to the ID-Schemata of Pollard and Sag 1994 and the subsorts of *constituent-structure* found there.

(448) Sort hiearchy below *phrase*, cited after Sag 1997, p. 439, nr. (9):



The sorts *non-headed-phrase* (*non-hd-phrase*) and *headed-phrase* (*hd-ph*) are the immediate subsorts of *phrase*. The latter is a supersort of all phrases with a syntactic head, i.e., with a head daughter, whereas the former covers all phrases without a head. Sag 1997 does not give an example of a non-headed phrase. The sort *hd-ph* has a number of subsorts: one for the combination of a head with an adjunct (*hd-adj-ph*), one for the introduction of a filler in an unbounded dependency (*hd-fill-ph*), and one for the realization of a subject, a complement and a specifier (the sorts *hd-subj-ph*, *hd-comp-ph* and *hd-spr-ph* respectively).

An effect of this sort hierarchy is that many principles of grammar are stated as implicational constraints with a single sort in the antecedent. This can be illustrated with the Head Feature Principle, which is a constraint on objects of the sort *hd-ph* in Sag 1997.

(449) The Head Feature Principle in Sag 1997:[30]

$$\textit{hd-ph} \Rightarrow \begin{bmatrix} \text{SYNSEM LOC CAT HEAD } \boxed{1} \\ \text{HEAD-DTR } \begin{bmatrix} \text{SYNSEM LOC CAT HEAD } \boxed{1} \end{bmatrix} \end{bmatrix}$$

The principle in (449) ensures the identity of head values between a phrase and its head daughter. As this principle is a constraint on the sort *hd-ph*, it is obeyed by all headed phrases. This version of the Head Feature Principle differs from that of Pollard and Sag 1994 — and from that introduced for our syntactic fragment in Section 2.3.1 — by having a sort as its antecedent, whereas the latter has a complex description. In (450), we formalize the Head Feature Principle of Pollard and Sag 1994 (p. 399) in RSRL.

(450) The Head Feature Principle in Pollard and Sag 1994:

$$\begin{bmatrix} \textit{phrase} \\ \text{DTRS } \textit{headed-structure} \end{bmatrix} \Rightarrow \begin{bmatrix} \text{SYNSEM LOC CAT HEAD } \boxed{1} \\ \text{DTRS HEAD-DTR } \begin{bmatrix} \text{SYNSEM LOC CAT HEAD } \boxed{1} \end{bmatrix} \end{bmatrix}$$

Similarly, equivalents to the ID-Schemata of Pollard and Sag 1994 are stated as constraints on the respective subsorts of *phrase*. For example, while the Head-Adjunct Schema of Pollard and Sag 1994 (p. 403) is a disjunct in the ID-Principle, it appears as a simple constraint on the sort *hd-adj-ph* in the grammar of Sag 1997.

The sort hierarchy given in (448) is not the entire sort hierarchy below *phrase*. In particular, none of the sorts that occur in the hierarchy in (448) is maximally specific. This comes as a surprise, considering that in RSRL, the sort hierarchy could be eliminated entirely — in fact, there is no sort hierarchy in the works of Paul King (King, 1989, 1994,

[30]Note that the attribute HEAD-DAUGHTER is defined on objects of the sort *hd-ph*. The signature of Sag 1997 does not have an attribute DAUGHTERS at all.

1999), which underlie the basic ontological assumptions of RSRL. In the sort hierarchies in the previous chapters of this thesis, the leaves stood for maximally specific sorts. This is different in the hierarchies normally given in papers written in the framework of constructional HPSG: there, the authors usually indicate just some small bits of the assumed hierarchy and normally leave out the most specific sorts.

The hierarchy in (448) captures only the partition of phrases with respect to their headedness, i.e., whether or not there is a head and what the relation between the head and the nonhead is in case there is a head. Sag assumes that there are other aspects with respect to which one can partition the sort *phrase*. In Sag 1997 the second aspect considered is that of *clausality*. Sag assumes subsorts of *phrase* for different kinds of clauses: first, a phrase need not be a clause at all, this is the case for NPs and VPs, for example. In that case, they are phrases of sort *non-clause*. If a phrase is a clause, then it can be either an imperative clause (*imp-cl*), a declarative clause (*decl-cl*), an interrogative clause (*inter-cl*) or a relative clause (*rel-cl*). We can add the clausality sorts to the sort hierarchy in (448). The resulting hierarchy is given in Figure 7.6. We follow Sag 1997 in indicating the different kinds of partitioning below phrase by so called "boxed types" in the sort hierarchy. The purpose of such a boxed type is to indicate that every phrase must be a subsort of each of the boxed types in the sort hierarchy.

As said above, every phrase must be of a subsort of some sort below each of the boxed types. In Figure 7.6 (page 306), however, there are no sorts that would satisfy this condition, i.e., in this hierarchy, there is no common subsort of clausality and headedness. This shows, that the hierarchy in Figure 7.6 is just the tip of an iceberg: for an actual phrasal sign to be of a subsort of both headedness and clausality, Sag 1997 must introduce many more subsorts of *phrase*. For illustration, consider the relative clause in (451).

(451) (a person) [who left]

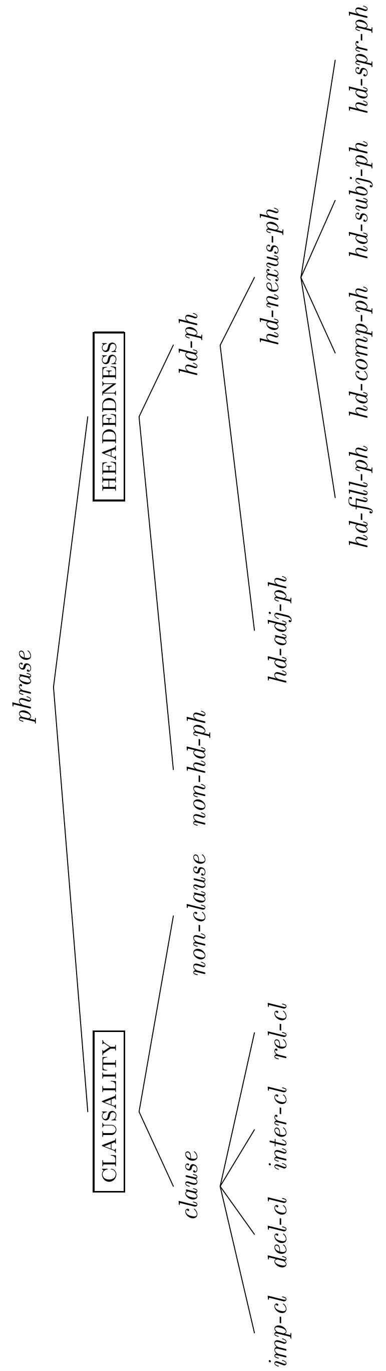
It is assumed to be of the sort *wh-subj-rel-cl*. This sort is a subsort of *rel-cl*, as it is a relative clause, and of *hd-subj-ph* as the nonhead daughter in the relative clause takes the function of a subject with respect to the verb *left*.[31]

This example already indicates that in a constructional approach to HPSG such as Sag 1997, the sort hierarchy below *phrase* contains a subsort for every single type of construction. In his discussion of the sort hierarchy below *word*, Meurers 2000 notes that such hierarchies lead to a duplication of information rather than to the expression of generalizations. His critisism can be applied validly to the sort hierarchy below *phrase*: the sort *wh-subj-rel-cl* expresses that the phrase is a relative clause which is finite and has a wh-constituent which is the subject at the same time. All this information is already expressed within the phrase itself: there is a VFORM feature whose value is *finite*, the nonhead daughter's SYNSEM value is identical to the element in the SUBJECT list of the head daughter, and the nonhead daughter is clearly identifyable as a wh-relative constituent by its REL value. The fact that such a combination of specifications is possible in English is expressed in constructional HPSG by introducing a sort such as *wh-subj-rel-cl*. In case a certain combination of specifications is not possible in a given language, such as infinite relative clauses in German for example, the sort hierarchy lacks the sorts that would characterize and license this construction.

In this sense, constructional HPSG can be considered more "constructive" than a non-constructional approach. A non-constructional approach, on the other hand, is in some intuitive sense more constraint-based, as it is necessary to exclude non-well-formed objects by principles of the grammar which exclude certain combinations of feature specifications.

[31]In addition, it is a subsort of the sorts *wh-rel-cl* and *fin-subj-ph*, as the relative clause contains a wh-constituent and is finite.

FIGURE 7.6. Sort hiearchy below *phrase*, cited after Sag 1997, p. 443, nr. (16):



The main practical difference between the constructional approach to HPSG and the one taken in this thesis and many other works within HPSG is that we assume a fixed signature, including the sort hierarchy, which should be as small as possible. All variation is, therefore, expressed by the principles of the grammar. This means that we want to impose as little ontological restrictions as possible. As a result, we assume that we often can improve a grammar simply by adding or changing some principles in the theory, while leaving the signature intact.

In a constructional approach, on the other hand, the sort hierarchy below the sort *phrase* is a vital part of the linguistic analysis. As a consequence, we can never be given the full sort hierarchy of a language, as every new word will potentially lead to the creation of new sorts. Furthermore, as all principles of grammar "live" on sorts in this approach, nearly every adjustment of the theory will lead to a modification in the sort hierarchy as well.

In short, we hope to have emphasized how different a constructional approach to HPSG is from the approach taken in this thesis. We cannot see whether there are empirical criteria to favor one approach over the other. We will show that it is enough to add a single attribute to the appropriateness conditions of the sort *sign* to express constructional effects under our view of HPSG as well.

A discussion of the differences between constructional and non-constructional HPSG was necessary to understand the context of the approach in Riehemann (1997). As we will see, Riehemann assumes a particular subsort of *phrase* for each IE. Thus, each IE has its own sort (with possibly additional subsorts).

In addition to the constructional aspect of her analysis, Riehemann 1997 assumes that each phrase has a set-valued attribute WORDS defined on it.[32] The WORDS set contains all words that are dominated by the phrase. The WORDS set assumed in Riehemann 1997 has the effect of providing local access to all words that occur in a phrase, without being forced to look into the syntactic structure. In the case of our simple example sentence (302), repeated as (452a), this means that the overall phrase can be described by the AVM in (452b).

(452) a. Peter read a book.

b.
$$\begin{bmatrix} \textit{decl-fin-subj-cl} \\ \text{PHON} \quad \langle \textit{Peter read a book} \rangle \\ \text{WORDS} \left\{ \begin{bmatrix} \textit{word} \\ \text{PHON} \; \langle \textit{Peter} \rangle \end{bmatrix}, \begin{bmatrix} \textit{word} \\ \text{PHON} \; \langle \textit{read} \rangle \end{bmatrix}, \begin{bmatrix} \textit{word} \\ \text{PHON} \; \langle \textit{a} \rangle \end{bmatrix}, \begin{bmatrix} \textit{word} \\ \text{PHON} \; \langle \textit{book} \rangle \end{bmatrix} \right\} \end{bmatrix}$$

Riehemann 1997 acknowledges that the WORDS set is not proposed elsewhere in the literature, but suggests that it can be used in an account of word order facts. Concerning the attribute WORDS, she says:

> "This attribute might also be useful for Linearization approaches to syntax, although it is distinct from the DOM attribute used in the approach developed at OSU and elsewhere, which does not contain all words individually." (Riehemann, 1997, footnote 5)

The following remarks demonstrate that this particular attribute is quite distinct from those used in linearization approaches to the extend that they would conceptually reject such an attribute.

Linearization approaches such as Reape 1990, 1994, Kathol 1995, Richter and Sailer 1995, Richter 1997, Penn 1999b,a, Donohue and Sag 1999 assume that word order cannot be determined within local trees. Instead, there are larger domains that must be considered. This can be illustrated with the relatively free word order in the German *Mittelfeld* (Uszkoreit, 1986; Jacobs, 1988). In the following examples, the relative order of the direct object *den Schläger* (*the racket*) and the indirect object *einem Fan* (*(to) a supporter*) is free. Note that furthermore, there is an adjunct *nach dem Spiel* (*after the game*) which intervenes between the two complements.

[32] Note that sets are not part of RSRL as defined in Chapter 2. Richter 1999 and Richter 2000 show how finite sets can be handled within RSRL. Finite sets are sufficient for the purpose of Riehemann 1997.

(453) a. Boris hat den Schläger nach dem Spiel einem Fan gegeben.
Boris has the racket after the game a supporter given
'After the game Boris gave his racket to a supporter.'
b. Boris hat einem Fan nach dem Spiel den Schläger gegeben.

Under the standard HPSG assumptions of Pollard and Sag 1994, none of the sentences in (453) can be analyzed. The reason for this lies in two assumptions which impose contradictory demands on these example sentences. First, adjuncts appear in the syntactic structure higher than the complements. Second, the phonology of a phrase is the concatenation of the phonologies of its daughters. With these two assumptions there is no way to have the phonology of the adjunct appear between that of the complements.

To overcome this problem, at least one of these assumptions must be dropped (or weakened). Some authors drop the first assumption, while sticking to the second: in the analyses of Kasper 1994 and Kiss 1995, 1997 data as those in (453) are accounted for in a theory which does not require the extension of the linearization domain from local trees. To achieve this, they must allow adjuncts to be introduced into the syntactic structure together with complements. Linearization approaches, on the other hand, drop the second assumption. Thus, they allow for word order to be determined within a domain as big as a clause.

The technical implementation of this simple idea varies a lot in the papers mentioned above. Some approaches assume an attribute such as DOMAIN that contains those bits of the daughters of the phrase that are necessary to determine the right ordering. In almost all analyses, these "bits" do not have all the information of a sign (in general, at least daughters information is absent), but might contain information of units larger than words (such as some representation of an entire NP). For the example in (453a), Kathol 1995 would assume the following value for the attribute DOM.[33]

(454)
$$\left\langle \begin{bmatrix}\text{PHON} & \langle \textit{Boris}\rangle \\ \text{SYNSEM} & \textit{NP}\end{bmatrix}, \begin{bmatrix}\text{PHON} & \langle \textit{hat}\rangle \\ \text{SYNSEM} & \textit{V}\end{bmatrix}, \begin{bmatrix}\text{PHON} & \langle \textit{den Schläger}\rangle \\ \text{SYNSEM} & \textit{NP}\end{bmatrix}, \begin{bmatrix}\text{PHON} & \langle \textit{nach dem Spiel}\rangle \\ \text{SYNSEM} & \textit{PP}\end{bmatrix}, \begin{bmatrix}\text{PHON} & \langle \textit{einem Fan}\rangle \\ \text{SYNSEM} & \textit{NP}\end{bmatrix}, \begin{bmatrix}\text{PHON} & \langle \textit{geschenkt}\rangle \\ \text{SYNSEM} & \textit{V}\end{bmatrix} \right\rangle$$

This list is different from a set that contains all the words of the sentence: First of all, it is a list whose order reflects that of the surface string. Second, the list contains only six elements, whereas there are ten words in the sentence. Third, the elements in the list are not of sort *sign* but of sort *dom-obj*. This sort has the attributes PHON and SYNSEM, but does not have a DTRS attribute. The motivation for having six instead of ten elements on this list stems from the fact that the word order of the sentences in (453) can be determined by considering just the permutations of these six elements, i.e., in German, the string *den Schläger* may not be seperated by phonological material that belongs to a different constituent in the clause. All linearization approaches, including Donohue and Sag 1999 for Warlpiri, are based on the observation that it is not necessary to consider the permutation of all words in a phrase, but that some words cluster together in an inseparable way.

Penn 1999a,b is perhaps the most radical linearization approach within HPSG. In Penn's approach, the DOM list of sentence (453a) would actually contain ten domain objects. In that respect, it is closer to Riehemann's proposal, at least from the point of view of the number of basic entities. Yet, these domain objects are not words. They are units which contain phonological and word order information, but no semantic information and at most very little syntactic information.

[33]As far as we know, there is just a single workshop handout, Richter and Sailer 1996b, which proposes a word order component in which the linearization rules operate on a list of words. This proposal has, however, never been pursued further.

As we will see below, the treatment of IEs in Riehemann 1997 crucially relies on a WORDS value which contains an element for every word in the phrase and for each of these elements bears at least phonological, semantic and subcategorization information. Under these conditions, Riehemann's WORDS list cannot be reduced to an attribute present in any of the currently available approaches to linearization in HPSG. Furthermore, such a WORDS set is incompatible with the aim of linearization approaches to restrict the information that is available in the linearization component to the minimum.[34]

Another difference between the approach of Riehemann 1997 and the one taken in this thesis is concerned with the treatment of semantics. In Part I of this thesis, we showed how an HPSG grammar such as that of Pollard and Sag 1994 can be furnished with a traditional architecture for combinatorial semantics. In contrast to this, Riehemann assumes the framework of *Minimal Recursion Semantics* (MRS) as presented in Copestake et al. 1995 and Copestake et al. 1997.

It would lead us too far astray to present the framework of MRS in detail. The main intuition behind MRS is that the semantic representation should be as flat as possible, i.e., the semantic representation of a sentence is not a single term which potentially consists of many subterms, but a list of some of these subterms. Let us consider a concrete example. According to the semantic system introduced in Chapter 4, the CONTENT value of sentence (302) (=(452a)) would be the following term:

(455) Peter read a book.
$\exists x[\mathsf{book}'_{@}(x_{@}) \wedge \mathsf{read}'_{@}(p, x_{@})]$

This term has a complex internal structure as it contains many subterms such as $[\mathsf{book}'_{@}(x_{@}) \wedge \mathsf{read}'_{@}(p, x_{@})]$, $\mathsf{book}'_{@}(x_{@})$, $\mathsf{read}'_{@}(p, x_{@})$ etc. The semantic representation chosen for in MRS avoids recursively structured terms, hence the name *Minimal Recursion Semantics*. Instead of having terms embedding other terms, the semantic representation is a list of subterms of a certain kind, called *relations*. The mutual embeddings/dependencies between these relations are indicated by special attributes. To see how this is done, we can augment the AVM in (452b) with a description of the semantic representation assumed in Copestake et al. 1997. This more detailed description of the sentence is given in Figure 7.7.[35]

Figure 7.7 needs some comments: The sorts *book-rel*, *read-rel* etc. stand for constants of the semantic representation language.[36] These constants are in part parallel to the analysis given in chapter 9 of Pollard and Sag 1994. For example, the constant *book-rel* has one argument which is given in the INST(ANCE) attribute. For verbs, the arguments are called ACT(OR) and UND(ERGOER), following a proposal of Davis 1997. In addition, Copestake et al. 1997 also assume an event variable for verbs, which is expressed as the value of the EVENT attribute. For a quantifier such as *a-rel*, the attribute B(OUND-)V(ARIABLE) indicates which variable is being bound. Further attributes, RESTR(ICTION) and SCOPE specify the restriction and the scope of the quantifier respectively.

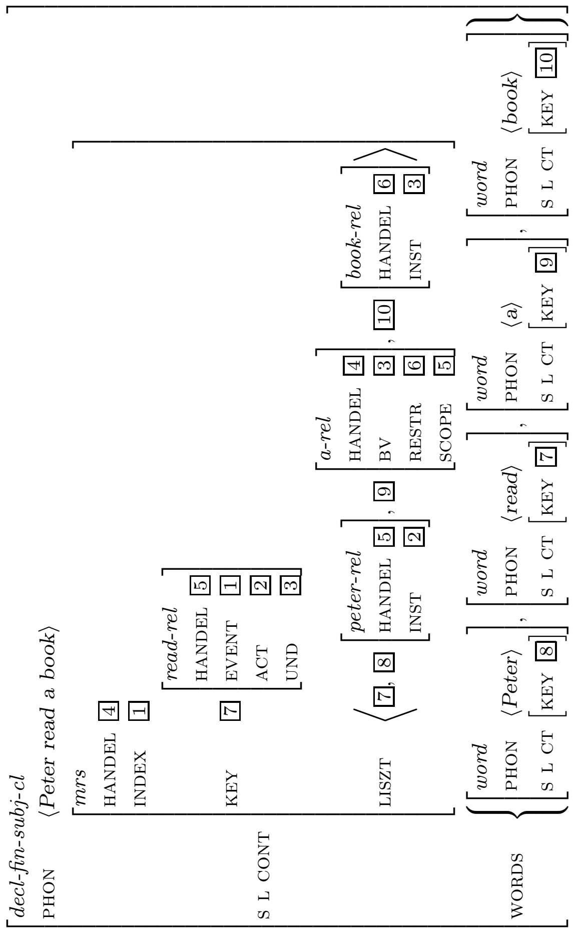
The values of the attributes used for the semantic argument slots come in two sorts: some of them are indices, i.e., of sort *index*, others are what Copestake et al. 1997 call *handles*. The sort *index* is known from Pollard and Sag 1994 and used in MRS in roughly the same

[34] Within an RSRL reformulation of Riehemann's theory, it would be possible to use a chain instead of an attribute. A chain would also make the words dominated by the idiomatic phrase locally available. In order to construct this chain it would, however, be necessary to traverse the syntactic structure dominated by this phrase recursively. This introduces a kind of non-locality which is avoided in Riehemann 1997.

[35] In Figure 7.7, S, L and CT abbreviate the attribute names SYNSEM, LOCAL and CONTENT, respectively.

[36] Copestake et al. 1997 (p. 12) state that they want to treat names as predicates. They do not give an explicit HPSG encoding of this proposal, but suggest the following two representations for the name *Sandy*: $\mathsf{Sandy}'(x)$, or $\mathsf{name}'(\mathrm{x}, \mathit{Sandy})$. In Figure 7.7 we have incorporated the first proposal, because it can be encoded more straightforwardly.

FIGURE 7.7. A description of the sentence *Peter read a book.*



way, with the difference that verbs are assumed to have an index as well. Handles are objects of sort *handel*. A handle corresponds to a formula. In the example, the restriction and the scope of the quantifier are given as handles, and furthermore, every constant has an attribute HANDEL. The restriction of the quantifier *a-rel* is given as the handle 6. This very same handle appears as the HANDEL value of the *book-rel* object. This is interpreted in such a way that the *book-rel* object is the restriction of the quantifier. The SCOPE value of the quantifier is the handle 5. In the LISZT value of the sentence, there are two relations which have this handle as their HANDEL values: the relation *read-rel* and the relation *peter-rel*. In this case, the interpretation of the handle 5 is the conjunction of the two relations.

So far, we have mainly considered the LISZT value of the sentence, but its CONTENT value has more attributes: as mentioned above, verbs and verbal projections are assumed to have an index in Copestake et al. 1997. Therefore, there is an INDEX attribute defined on the CONTENT value of the sentence. This index is identical to the EVENT value of the relation *read-rel*. Another attribute in the CONTENT value is KEY. The KEY value expresses the major semantic contribution of the lexical head of a sign. In the case of the sentence in Figure 7.7, this clearly is the *read-rel* object (7). Finally, the CONTENT value has a HANDEL attribute defined on it. This handle indicates which of the relations in the LISZT value is the semantics of the sign. In our case, this is the HANDEL value of the quantifier, $\boxed{4}$. Thus, the meaning of the sentence is identical to the meaning of this relation.[37]

In MRS, the CONTENT value of a phrase is the result of appending the LISZT values of the daughters and possibly adding construction specific relations. The KEY and the INDEX values are identical on the mother and on the head daughter in a headed phrase. There are additional principles that take care of assigning scope bearing elements the right scope.

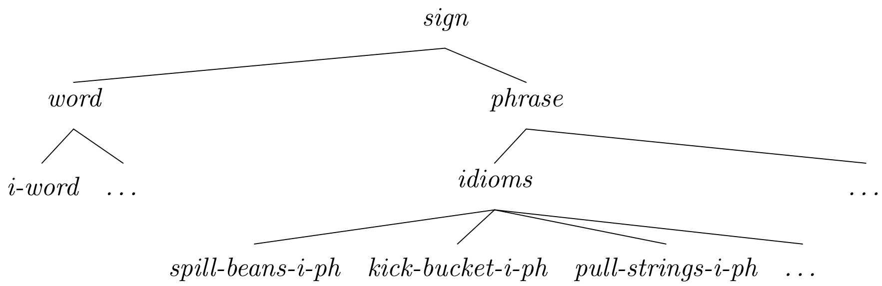
For our discussion of IEs it is important to note all the words that occur in a phrase and the semantic contribution they make, are locally available at that phrase. This is indicated in Figure 7.7 by the fact that every element on the LISZT list of the sentence is identical to the KEY value of some word on the WORDS list. We will see in the next subsection that Riehemann 1997 uses this to require for an idiomatic phrase the presence of a particular set of words with a particular meaning.

Now that we have presented the constructional HPSG, the WORDS set and the framework of MRS, we are ready to consider treatment of IEs proposed in Riehemann 1997.

7.3.2. A Constructional HPSG Analysis of IEs. In Riehemann 1997 the theoretical assumptions presented in the preceding subsection are put together to achieve an analysis of IEs which shares many aspects with the TAG analysis of Abeillé 1995. Riehemann's analysis has the following attractive properties: (i) in cases of semantically regular IEs, the overall semantics is constructed in a regular combinatorial way; (ii) yet it is not necessary to have lexical entries for words such as *beans* in their idiomatic reading of beans″; (iii) (limited) syntactic flexibility is achieved by specifying as many properties as necessary for the syntactic relations that must hold between the parts of the IE.

As Riehemann's account is constructional, she enriches the sort hierarchy below the sort *sign*: below *word*, Riehemann introduces a subsort *i(diomatic)-word*, below *phrase*, there is a subsort *idioms* which has an immediate subsort for each idiomatic phrase in the language. In our case, the subsorts of *idioms* contain the sorts *spill-beans-i(diom)-ph(rase)*, *kick-bucket-i-ph*, and *pull-strings-i-ph*. The new part of the sort hierarchy is given in (456).

(456)



Riehemann's analysis can best be illustrated with the IE *spill the beans*. In (457), we give the description of this IE as found in Riehemann 1997 (p. 8).

(457)
$$\begin{bmatrix} \textit{spill-beans-idiom-phrase} \\ \text{WORDS} \left\{ \begin{array}{l} \begin{bmatrix} \textit{i-word} \\ \text{S L CT KEY} \begin{bmatrix} \textit{i-spill-rel} \\ \text{UND } \boxed{1} \end{bmatrix} \end{bmatrix} \stackrel{<}{\sqcap} \begin{bmatrix} \textit{spill} \end{bmatrix}, \\ \begin{bmatrix} \textit{i-word} \\ \text{S L CT KEY} \begin{bmatrix} \textit{i-bean-rel} \\ \text{INST } \boxed{1} \end{bmatrix} \end{bmatrix} \stackrel{<}{\sqcap} \begin{bmatrix} \textit{bean} \end{bmatrix}, \dots \end{array} \right\} \end{bmatrix}$$

[37]See Egg 1998 for an explicit translation from an MRS-like system to predicate logic, or Richter and Sailer 2001 for an MRS-style treatment of the semantic representation language Ty2.

The AVM in (457) describes a phrase of sort *spill-beans-i-ph* which has two idiomatic words in its WORDS set. One of these words has a KEY value of sort *i-spill-rel* in its content, the other has a KEY value of the sort *i-bean-rel*. These special relations correspond to the semantic constants spill'' and beans'' of Gazdar et al. 1985, though they need not be interpreted as partial functions. In (457), a new symbol, $\overset{<}{\sqcap}$, is used. It is explained as follows:

> "…the idiomatic word …on the left side of the $\overset{<}{\sqcap}$ symbol is just like the literal word on the right, except for the properties that are explicitly changed." (Riehemann and Bender, 2000, p. 9)

The precise formal interpretation of the $\overset{<}{\sqcap}$ symbol is rather hard to grasp, but if we see this symbol as an abbreviatory convention to make the basic idea behind (457) clearer, there are no problems.[38] What is expressed by a description such as that of the first-mentioned element in the WORDS set in (457) is the following: the described word has a KEY value as given and is described by the lexical entry of *spill* with respect to all aspects except of the KEY value. A formal explication of this idea would raise many questions and depend on details of the way inflection, passive and other lexical processes are handled. It seems to us, however, that whatever the approach to lexical generalizations, there is always a way to eliminate the $\overset{<}{\sqcap}$ symbol by an explicit description. As these issues are independent of Riehemann's analysis, the $\overset{<}{\sqcap}$ symbol gives her a useful abbreviation.

In the following, we describe how Riehemann 1997 analyzes several IEs in this framework. We will show why in the form it is presented in Riehemann 1997, the approach heavily overgenerates, allowing idiomatic words to appear everywhere in the language and not ensuring that an idiomatic phrase is present whenever the right idiomatic words occur.

But, before we detail our critique, we should first consider the way the analysis of Riehemann 1997 is intended to work. Consider the following two sentences.

(458) a. Pat spilled the beans.
b. I was worried that the beans might be spilled.

These sentences illustrate the syntactic flexibility of the IE: in (458a), the noun *beans* appears as the direct object, in (458b), the noun is the subject and appears in a raising construction. In Figure 7.8, we give the syntactic structure of the *that*-clause of sentence (458b).

In the description of the sort *spill-beans-i-ph* in (457), only the semantic relation between the two idiomatic words is mentioned. This leaves open how these two words combine syntactically. In the case of (458a), the noun *beans* is the head of the direct object; in the case of (458b), the syntactic relation is mediated through a series of raising verbs.[39]

The analysis of the IE *pull strings* is parallel to that of *spill the beans*. We can see that also the topicalization and relative clause data of this IE are easily accounted for under Riehemann's approach.

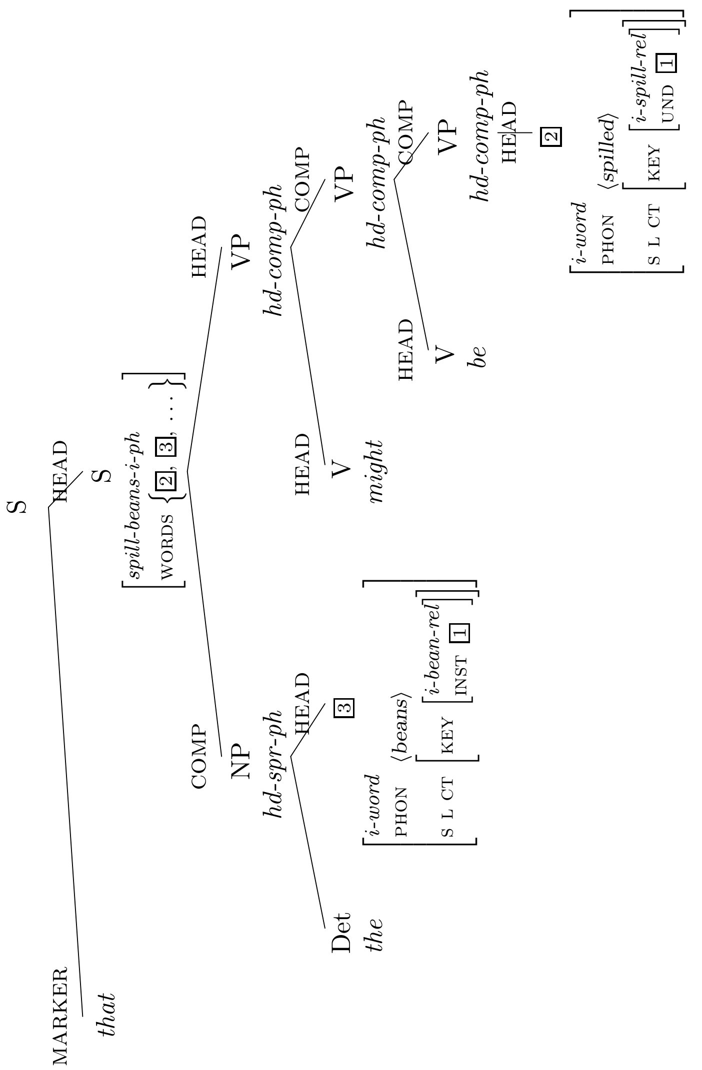
(459) a. Those strings, he wouldn't pull for you.
b. The strings that Pat pulled got Chris the job.

In (459a), the topicalized NP has the index of its head noun, *strings*. The LOCAL value of this NP appears as the direct argument on some list of the idiomatic verb *pull*, probably on the ARG(UMENT)-ST(RUCTURE) list. There the NP is linked to the UNDERGOER value of the verb. Thus, again, the required semantic relation holds.

[38]Riehemann 1997 (footnote 6) explicitly states that the $\overset{<}{\sqcap}$ symbol is interpreted as an instruction to some compiler of a parsing system and is not part of the HPSG grammar proper.

[39]Similarly, internal modification is possible because all that is required is that the index of the noun *beans* appear as the UNDERGOER value of the verb *spill*. Modification does not change the index.

FIGURE 7.8. The structure of the *that*-clause in (458b)



The same is true in the case of the relative clause. Here, the idiomatic phrase must dominate at least the entire relative clause and the head noun of the NP, *beans*. Under both the relative clause analysis of Pollard and Sag 1994 and that of Sag 1997, the relative pronoun has the same index as the head noun to which the relative clause attaches. In the case of the relative clause in (459b), the UDC mechanism together with the assumptions about argument linking will further provide the identity of this index with the UNDERGOER value of the idiomatic verb *pull*, parallel to what happens in topicalization.

To exclude topicalization and relative clause formation in the case of the IE *spill the beans*, Riehemann would have to assume that there is no common subsort of *spill-beans-i-ph* and the sort *hd-fill-ph*, nor is there a common subsort of the sort *spill-beans-i-ph* and the sort *rel-cl*. This indicates the close similarity between Riehemann's constructional HPSG analysis and the TAG analysis of Abeillé 1995: in those cases where Abeillé 1995 assumes the absense of some elementary trees in the Tree family, Riehemann 1997 would assume the absense of a common subsort of a particular IE and a certain constructional sort. Such cases are discussed in detail in Riehemann and Bender 2000.

Next, let us briefly consider yet another type of IEs whose existence is taken as a clear indication for the necessity of a constructional approach: the IE *kick the bucket*. In (460), we state the description of a *kick-bucket-idiomatic-phrase* as given in Riehemann 1997 (p. 10).[40]

$$
(460)\quad \begin{bmatrix} \textit{kick-bucket-idiom-phrase} & \\ \text{WORDS} & \left\{ \begin{array}{l} \begin{bmatrix} \textit{i-word} & \\ \text{S L} & \begin{bmatrix} \text{CAT VAL} & \begin{bmatrix} \text{SUBJ} & \left\langle \begin{bmatrix}\text{L CT INDEX } \boxed{1}\end{bmatrix} \right\rangle \\ \text{COMPS} & \left\langle \begin{bmatrix}\text{L CT KEY } \boxed{2}\end{bmatrix}\right\rangle \end{bmatrix} \\ \text{CT KEY} & \textit{empty-rel} \end{bmatrix} \end{bmatrix} \overset{<}{\sqcap} \begin{bmatrix}\textit{kick}\end{bmatrix}, \\ \begin{bmatrix} \textit{i-word} \\ \text{S L CT KEY } \boxed{3}\ \textit{empty-rel} \end{bmatrix} \overset{<}{\sqcap} \begin{bmatrix}\textit{the}\end{bmatrix}, \\ \begin{bmatrix} \textit{i-word} & \\ \text{S L} & \begin{bmatrix} \text{CAT VAL} & \begin{bmatrix}\text{SPR} \left\langle \begin{bmatrix}\text{L CT KEY } \boxed{3}\end{bmatrix}\right\rangle\end{bmatrix} \\ \text{CT KEY} & \boxed{2}\ \textit{empty-rel} \end{bmatrix} \end{bmatrix} \overset{<}{\sqcap} \begin{bmatrix}\textit{bucket}\end{bmatrix}, \ldots \end{array}\right\} \\ \text{CXCONT} & \begin{bmatrix}\text{LISZT} \left\langle \begin{bmatrix}\textit{i-kick-bucket-rel} \\ \text{ACT } \boxed{1}\end{bmatrix}\right\rangle\end{bmatrix} \end{bmatrix}
$$

The description of this IE is much more complex than that of the semantically and syntactically more regular IEs such as *spill the beans* and *pull strings*. In the WORDS set, we find the verb *kick*, the determiner *the* and the noun *bucket*. For all three of them, the KEY value is given as *empty-rel*. This is a special relation which Riehemann must introduce for those parts of an IE to which one cannot attribute a meaning. The description of the verb further specifies that the element in the COMPS list be some NP headed by the idiomatic noun *bucket*. This noun must take as its specifier the idiomatic determiner *the*, which follows from the SPR value of the noun in (460). By these details on the syntactic relations that hold among the three idiomatic words of the IE, Riehemann accounts for the lack of syntactic flexibility of the expression *kick the bucket*.

In the AVM in (460), there is one attribute that we have not discussed before, the attribute CXCONT (CONSTRUCTION-CONTENT). This attribute is part of the MRS semantic theory. Under a constructional HPSG, it is assumed that constructions may make semantic contributions of their own, i.e., the meaning of a construction can contain elements that are not present in the words that the construction is built from. In combinatorial semantics, Copestake et al. 1997 assume that the LISZT value of a phrase is the union of the LISZT values of all its daughters and its own CXCONT LISZT value.

For the example of *kick the bucket* this has the effect that while all the words in the IE have an empty semantic contribution, i.e., they have a KEY value of sort *empty-rel*, the overall idiomatic phrase has the intended meaning as the relation *i-kick-bucket-rel* is contributed by the phrase itself.

After this presentation of the analysis in Riehemann 1997, we will consider a major problem of the approach. Certainly, there are ways to solve this problem within Riehemann's account[41], but we think that it points to a basic misconception in the constructional account. We will take the IE *spill the beans* for illustration.

It is actually the first semantic criterion that goes to the heart of the problem: as we have noted in Chapter 6, the words *spill* and *beans* do not occur in their idiomatic meaning outside the IE *spill the beans*. Riehemann wants to achieve this by not assuming special

[40] We have eliminated a typo from Riehemann 1997, as in the AVM given there, the SUBJ value of the idiomatic word *kick* is described as $\left\langle [\ldots \text{KEY } \boxed{1}] \right\rangle$. This cannot be intended as the KEY value is of the sort *relation*, whereas the ACTOR attribute of the *i-kick-bucket-rel* takes an index as its value.

[41] See Riehemann (2001) and the short discussion in footnote 42.

lexical entries for these words. As we have seen in (456), however, there are objects of the sort *idiomatic-word*. Every idiomatic part of an IE will be of this sort, thus, there will at most be a very vague description of the objects of sort *idiomatic-word* in the grammar. As the idiomatic word *spill* is certainly part of the English language, there is an idiomatic word that can be described by the following AVM.

(461)
$$\begin{bmatrix} \textit{idiomatic-word} \\ \text{PHON } \langle \textit{spilled} \rangle \\ \text{SYNS LOC} \begin{bmatrix} \text{CAT HEAD } \textit{verb} \\ \text{CONTENT} \begin{bmatrix} \textit{mrs} \\ \text{INDEX } \boxed{1} \\ \text{KEY} \begin{bmatrix} \textit{i-spill-rel} \\ \text{EVENT } \boxed{1} \\ \text{ACT } \textit{index} \\ \text{UND } \textit{index} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

The AVM in (461) describes the word *spill* as it occurs in sentences like those in (458) above, repeated in (462).

(462) I was worried that the beans might be spilled.

Clearly, the idiomatic word *spill* need not be directly dominated by a *spill-beans-i-ph*. In (462), there is a number of VPs that dominate the idiomatic verb *spill* before the idiomatic word *beans* is introduced. This can be seen in the constituent structure of the *that* clause of sentence (462) which was sketched in Figure 7.8 (page 313).

In the tree in Figure 7.8, there are at least three phrases which have the idiomatic word *spill* on their WORDS set but which are not of the sort *spill-beans-i-ph*. These are the three VPs which are indicated as being of the sort *hd-comp-ph*. Still, the sentence is grammatical. This indicates that Riehemann's system must allow for phrases which are not idiomatic but contain idiomatic elements. If this is the case, then there is nothing in the grammar that prevents sentences which do not contain an occurrence of a *spill-beans-i-ph* at all, but which do contain an idiomatic word such as the idiomatic word *spill* in (461). The grammar, thus, cannot exclude the following sentence in which the word *spill* is used as an idiomatic word.

(463) * Pat spilled the secret.

Neither does the grammar enforce that whenever the idiomatic verb *spill* is combined with the idiomatic noun *beans*, there is also an instance of a *spill-beans-idiomatic-phrase*. This means that sentence (462) has an analysis which contains the idiomatic words *spill* and *beans* and, thus has the idiomatic reading, but the analysis differs from that given in Figure 7.8 in that the S node would not be of sort *spill-beans-i-ph* but of the subtype of *hd-subj-ph* used for regular, i.e., non-idiomatic, combinations.

To prevent this, a constraint could be added, saying that whenever there is a phrase which has the idiomatic words *spill* and *beans* in its WORDS set, this phrase must be a *spill-beans-i-ph*. Clearly, such an ad hoc constraint would not solve the problem, because it would require that in the case of sentence (462), all phrases which dominate the NP *the beans* be of sort *spill-beans-i-ph*. This would include the S′ node in the tree in Figure 7.8 as well as the phrase which dominates the entire sentence. Even if this were acceptable, which we think it is not, it does not offer a solution to the overgeneration problem.

The problem indicated here, is, however, more general still: in fact, we could have any idiomatic word occurring almost anywhere in the language. As there is no particular restriction on the shape of an idiomatic word in the grammar, an idiomatic word could be of any shape compatible with the rest of the principles of grammar. In particular, we could have idiomatic words, which do not have anything in common with the ordinary words of English. Thus, instead of the idiomatic word *spill*, we could also have an idiomatic word with an arbitrary phonology (*bace*) and an arbitrary meaning (*snore-rel*). The following sentence would then be a regular sentence which happens to contain an idiomatic word. The meaning would be that of the sentence "*Pat snores.*".

(464) * Pat baces.

The problem that we have just illustrated arises from the fact that in Riehemann 1997 only the restrictions on the idiomatic phrases are provided, i.e., whenever there is a certain idiomatic phrase, it must be composed of the right idiomatic words. What is missing in this proposal is a constraint that a certain idiomatic word must occur as part of a certain idiomatic phrase. In our example, a restriction must be added that states that the idiomatic word *spill* may only occur if it is dominated by some *spill-beans-i-ph* within an appropriate syntactic domain. The same is true for the idiomatic word *beans*. Therefore, what we need are constraints on the distribution of idiomatic words in larger contexts.

Another way to explain this basic problem of Riehemann's approach is to compare it with the TAG analysis of Abeillé 1995. In both approaches, the IE *spill the beans* is treated as a unit, but the same basic idea has different consequences in the two frameworks. In TAG, the idiomatic use of the verb *spill* cannot occur outside the IE, because it only exists as part of the elementary tree that belongs to the IE. In HPSG, complex signs have components which must be licensed by the grammar independently. Therefore, Riehemann is forced to introduce the sort *i-word*. If these signs exist independently, their distribution must be accounted for. This is not done in Riehemann 1997. Thus, Abeillé is right in pointing out that the notion of locality in TAG is wider than in HPSG. In the case of IEs, this makes a difference with respect to the analytic alternatives that are available.

The basic problem of the account in Riehemann 1997 is that while an attempt is made to treat idioms locally at the level of a construction, the analysis is not able to constrain appropriately neither the distribution of the idiomatic words nor the distribution of the idiomatic phrases. The approach that we will present in the next chapter, will expand on means of expressing distributional constraints for lexical items. Thus, it will be possible to attach to the idiomatic word *spill* the restriction that it only occurs in clauses which also contain the relevant parts of the idiomatic word *beans*, and the other way around. In addition, we will show that once we can express restrictions on the distribution of lexical items, there is no need for particular constructions in the analysis of internally regular IEs.[42]

7.4. Summary

In this chapter we have discussed three formal theories whose purpose is to account for the IE data presented in the previous chapter. The approaches were formulated in different linguistic frameworks, GPSG, TAG and constructional HPSG respectively. The choice of the framework had an enormous influence on the particular analysis:

[42] The probem of overgeneralization is explicitly addressed in Riehemann (2001, Section 5.2.1). To solve it, additional sorts, attributes and constraints are introduced to collect the idiomatic words that occur in a sentence, and to keep track of which idiomatic words are needed for the idioms in the sentence. At the level of a root clause, it is then ascertained, whether or not the clause contains all the idiomatic words, and respectively only those idiomatic words required by the idioms in the given clause. While this solves the problem at a technical level, the solution requires a massive apparatus that is not motivated or needed for the description of other linguistic phenomena. This shows that the constructional approach is genuinly problematic for an account of IEs in HPSG.
Furthermore, there are at least two remaining issues. The first issue is that this analysis is inconclusive about which phrasal node should be considered a *spill-beans-i-ph* in Figure 7.8. Second, as noted by Riehemann (2001, p. 207) herself, certain pronominalization data remain unaccounted for. Riehemann gives the following example:

(i) Eventually she spilled all the beans. But it took her a few days to spill them all.

The problem is that the *but*-clause contains an idiomatic *spill*, but no idiomatic *beans*.

The GPSG approach of Gazdar et al. 1985 emphasized the regular aspects of IEs, providing a technique to avoid free occurrences of idiomatic readings of particular words. This approach, however, had no obvious way to account for more irregular IEs such as *kick the bucket* and *trip the light fantastic*.

The TAG account given in Abeillé 1995 was forced to rely on an analysis that emphasized the irregular aspects of IEs, because in TAG trees are the only available data structure. Thus, the paradigm examples of the approach was the IE *kick the bucket*. In case of more regular combinations such as light verb constructions, the TAG architecture was forced to adopt an idiomatic approach, even though Abeillé 1988 claimed that a more combinatorial account could be possible.

Finally, we discussed the approach of Riehemann 1997, which is formulated within constructional HPSG. In spirit, this approach was very close to the TAG analysis and tried to account for the data similarly. The fundamental formal differences between TAG and HPSG led to problems which remained unsolved within constructional HPSG.

Let us reconsider these results in the light of the classification of the data that we have given in Section 6.3. There, we made a distinction between internally irregular IEs and IEs which are internally regular but may contain parts which show external irregularities. Of the list of IEs which we are considering in this thesis, only two are internally irregular: *trip the light fantastic* and *kick the bucket*. In both cases, the meaning of the IE cannot be arrived at by regular combination of the semantic contribution of their parts.

All other IEs have been shown to be internally regular. Still, some parts of the IEs were shown to have distributional, i.e., what we call external, irregularities: in the case of the IE *spill the beans* the irregularity is that none of the words *spill* or *beans* occurs in their idiomatic meaning outside the IE. In most light verb constructions, the light verbs could be classified as showing external irregularity. The IE *make headway* is a special case as there, in addition to the usual irregularity of the light verb *make*, we have the noun *headway* which does not exist at all outside the IE.

In the light of this classification of the data, the approach of Gazdar et al. 1985 can be considered successful for internally regular IEs, but lacks any formal means to account for internally irregular IEs. An analysis of these IEs requires integration of syntactically complex lexical items into the theory.

The TAG analysis provided an elegant account of the internally irregular IE *kick the bucket* as TAG is exactly the kind of theory that has syntactically complex lexical items at its disposal. As these are the only lexical items there are, the approach seems to miss generalizations in the case of internally regular IEs. Still, it must be acknowledged that the TAG approach is the only one presented in this chapter in which we were able to give an account for all IEs under consideration and to describe all the data given in Chapter 6.

As the strengths of the GPSG approach and the TAG appproach complement each other, it is desirable to combine insights of both approaches into an HPSG theory. Riehemann 1997 addresses the basic problem of Gazdar et al. 1985, and the framework of constructional HPSG is used to provide a theory of syntactically complex entities with idiosyncratic properties. HPSG being, however, too close in spirit to GPSG withstands a simple constructional approach. For this reason, we must seek an analysis of IEs which incorporates more of the ideas of the GPSG account. In the following chapter, such an analysis will be presented. While adopting the analytic approach of GPSG, we will combine this with the logical form representations introduced in Part I. As a result, we will be able to substitute the complicated mechanism of partially defined denotations by simple restrictions on semantic representations.

CHAPTER 8

The Analysis

After the presentation of the data in Chapter 6 and the discussion of previous analyses in Chapter 7, we can now present our own approach. This approach is based on the distinction between internally regular and internally irregular idiomatic expressions. For the treatment of internally irregular IEs, we will need a notion of a syntactically complex lexical element. Thus, in our approach, IEs such as *kick the bucket* and *trip the light fantastic* are treated as phrases which are listed in the lexicon. For the treatment of internally regular IEs, we will present a mechanism that will allow us to state exactly in which linguistic structures a given word may occur. From this characterization, it can be seen that our approach falls in the middle between the TAG analysis for internally irregular IEs and the GPSG analysis for internally regular IEs.

Technically, we will use a non-constructional approach to HPSG, i.e., we will not assume a different subsort of *phrase* for idiomatic expressions. Instead, we will introduce a new attribute that will be useful for the account of both internally regular and internally irregular IEs. On the one hand, we will use this attribute to differentiate between regularly built phrases and phrases with internal irregularities. For example, this attribute can be used to differentiate between the literal and the non-literal use of the VP *kick the bucket*. On the other hand, the value of this attribute will indicate the precise context in which an element is required to occur. Thus, in the case of the idiomatic use of *beans* as it occurs in the IE *spill the beans*, the value of this attribute will indicate that the idiomatic use of *beans* is only possible if there is an occurrence of the non-logical constant spill′′ in the same clause such that the idiomatic beans are an argument of the idiomatic spilling.

As this short preview of our account already demonstrates, the new attribute will play a crucial role in our analysis. We will call it COLL, which is an abbreviation for *Context Of Lexical Licensing*. The assumption behind this name is that all and only those elements that are licensed by some lexical entry (of a word or of an irregular phrase) can in principle impose distributional restrictions on their linguistic contexts. Put differently, a lexical element (word or phrase) can only occur in a linguistic context that is compatible with the distributional requirements of this lexical element. The COLL attribute will be used to specify these requirements.[1]

The name COLL might also be associated with the term *collocation*. Unfortunately, this term is used in many different ways, to refer to combinations such as *commit a crime* or to refer to the fact that most texts that have an occurrence of the word *cow* also have an occurrence of the word *milk*. What most of the uses of the term have in common, is that they assume that the presence of a certain lexical element suggests/requires/enables/triggers the presence of a particular second lexical element. There are at least some authors (Krenn and Erbach, 1994; van der Wouden, 1997) that call the phenomena treated in the present

[1]This attribute was first introduced in the analysis of negative concord in Polish in Richter and Sailer 1999c, and used subsequently for an analysis of negative concord in French (Richter and Sailer, 1999a). In both papers no further motivation was given for this attribute. We implicitly assumed the analysis of idiomatic expressions to be the basic motivation for this attribute. This motivation is, finally being presented in this thesis.

part of this work *collocational phenomena*. We have, however, preferred to use the term *idiomatic expression* which seems to be less theoretically charged.

In this chapter, we will extend the use of the COLL feature step by step, until we finally see its full functionality in Section 8.3. In our analysis of internally irregular IEs in Section 8.1 we will only use it to establish the notion of a syntactically complex lexical element. In the analysis of internally regular IEs in Section 8.2, we will use the attribute only to specify the licensing context for words. In Section 8.3, finally, we will show that it is useful to assume a non-trivial context of licensing for all lexical elements. Section 8.4 should be understood as an appendix to this chapter. There, we will take up the issue of how semantically external modification can be handled in our framework.

8.1. Internally Irregular IEs

In this section, we will examine how the two internally irregular IEs of our list (*trip the light fantastic* and *kick the bucket*) can be handled within (non-constructional) HPSG. Our discussion of the previous analyses revealed that it is desirable to treat these IEs as internally complex lexical items. In contrast to the approach of Riehemann 1997, however, we will not use an elaborate sort hierarchy below *phrase* for this purpose. Instead, we will introduce a new attribute, COLL (*Context Of Lexical Licensing*). In the first instance, this attribute will merely serve to tell regular and irregular phrases apart, but, as the name suggests, it will also be of great importance in our analysis of the more flexible IEs.

For the purpose of internally irregular IEs, it is enough to declare this attribute appropriate for the sort *phrase*. We assume, for the time being, that it takes boolean values. This leads to a change in the appropriateness conditions of the sort *phrase* as given in (70) and used throughout Part I. The modified sort hierarchy and appropriateness conditions below the sort *sign* are given in (465).

(465) Sort hierarchy and appropriateness conditions below the sort *sign* (preliminary):

sign
PHON *list*
SYNSEM *synsem*

word
STORE *list*

phrase
DTRS *const-struc*
COLL *boolean*

As can be seen in (465), the attribute COLL is defined on the sort *phrase*. We assume that regular phrases have a COLL value of sort *minus*, whereas irregular phrases have a COLL value of sort *plus*. As an example, consider the literal and the idiomatic interpretation of the phrase *kick the bucket*. In its literal interpretation, the VP has a COLL value *minus*, as an IE, its COLL value is *plus*.

In a second step, we change all principles of grammar that are concerned with the regular combination of signs in such a way that they only constrain phrases with a [COLL −] specification. This effects all the principles discussed in Section 2.3, i.e., the Immediate Dominance Principle in (78), the Head Feature Principle in (81), the Marking Principle in (83), the Spec Principle in (89), and the Nonlocal Feature Principle in (105). In addition, the Semantics Principle, in any of its versions given in Chapter 4, is equally effected.[2] In (466) we state the revised version of the Head Feature Principle.

[2] The list of principles must also include a principle for the regular combination of the PHON values such as the Constituent Ordering Principle of Pollard and Sag 1987.

(466) The HEAD FEATURE PRINCIPLE (HFP), relativized to regular phrases:

$$\begin{bmatrix} \textit{phrase} \\ \textsc{dtrs}\ \ \textit{headed-struc} \\ \textsc{coll}\ \ \textit{minus} \end{bmatrix} \Rightarrow \begin{bmatrix} \textsc{syns loc cat}\ \ \textsc{head}\ \boxed{1} \\ \textsc{dtrs h-dtr}\ \ \begin{bmatrix} \textsc{syns loc cat head}\ \boxed{1} \end{bmatrix} \end{bmatrix}$$

The new version of the HFP only differs from that given in Pollard and Sag 1994 (formalized as (81) on page 89) by an additional specification in the antecedent of the principle. This new HFP says that the identity of HEAD values is only required for regular phrases. Applying similar trivial changes to the other principles of the grammar results in a grammar in which phrases with a [COLL +] specification are exempt from all requirements of regular combination. But this is exactly what we wanted to achieve.

In Figure 8.1, we give the structure of a simple sentence with the IE *kick the bucket*. At the phrasal nodes, we indicate the syntactic category, the logical form and the COLL value.

The tree in Figure 8.1 shows that we assume only one phrase in the structure of the sentence to be internally irregular: the VP. For all the other nodes in the tree, we assume a regular behavior. Thus, the words *bucket* and *kick* appear with their normal meaning bucket' and kick'. At the VP node, however, these meanings are not combined in the usual way, i.e., we do not have the CONTENT $\lambda x.[\mathsf{the}\ y : \mathsf{bucket}'_{@}(y_{@})](\mathsf{kick}'_{@}(x_{@}, y_{@}))$. Instead, the irregular phrase introduces a semantic constant which is not present in the daughters and even completely ignores the semantic contribution of the daughters. As the VP has a [COLL +] specification, it is exempt from the SEMANTICS PRINCIPLE. Thus, the principles of grammar are not violated in a tree as the one in Figure 8.1.

So far, we have not said how we can avoid arbitrary phrases from bearing the specification [COLL +]. This is done by a principle which we call the INTERNAL IRREGULARITY PRINCIPLE (IIP). In (467), we give the general form of the IIP. As can be seen, the IIP is an implication with a big disjunction in the consequent. Each disjunct in the consequent describes an internally irregular sign. We can assume that one of the disjuncts is a description of the IE *kick the bucket*.

(467) The INTERNAL IRREGULARITY PRINCIPLE (IIP):

$$\begin{bmatrix} \textsc{coll}\ \ \textit{plus} \end{bmatrix} \Rightarrow \big(\mathrm{PLE}_1\ \textbf{or}\ \ldots \textbf{or}\ \ \mathrm{PLE}_n\big)$$

Following Pollard and Sag 1994 in Section 2.3, we introduced two principles which take the form of an implication with a disjunction in the consequent: the WORD PRINCIPLE and the ID PRINCIPLE. The first contains a description of all words, the latter a description of all phrases. Due to our modification of the appropriateness for the sort *phrase*, i.e., the introduction of the attribute COLL, and the subsequent relativization of the ID PRINCIPLE, the IIP fills a systematic gap in the principles that apply to signs. For the sake of completeness, we give the WORD PRINCIPLE and the ID PRINCIPLE schematically, similar to the IIP above.[3]

(468) a. The WORD PRINCIPLE: (from (76))

$$\begin{bmatrix} \textit{word} \\ \textsc{store}\ \ \textit{elist} \end{bmatrix} \Rightarrow \big(\mathrm{LE}_1\ \textbf{or}\ \ \ldots \textbf{or}\ \ \ldots \mathrm{LE}_m\big)$$

b. The ID PRINCIPLE: (from (78))

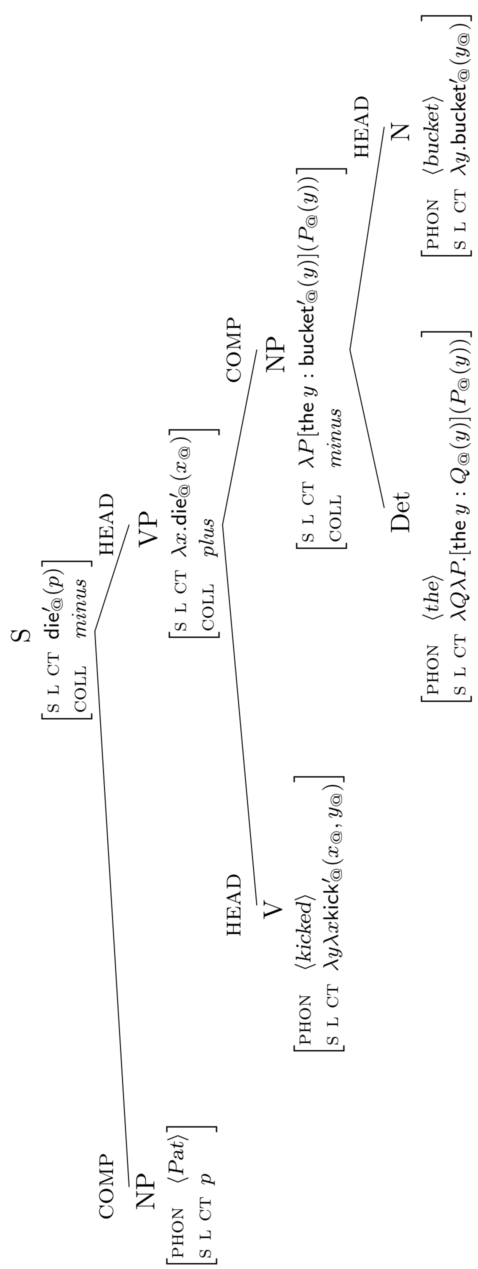
$$\begin{bmatrix} \textit{phrase} \\ \textsc{coll}\ \ \textit{minus} \end{bmatrix} \Rightarrow \big(\mathrm{HC}\ \textbf{or}\ \mathrm{HA}\ \textbf{or}\ \mathrm{HM}\ \textbf{or}\ \mathrm{HF}\big)$$

[3]The WORD PRINCIPLE differs from that in Meurers 2000 (p. 124). We assume that all the lexical entries LE_i contain the specification [STORE *elist*], i.e., that lexical entries are given only for words which are not the output of a derivational rule. For those words that are the output of such a rule, we assume the STORE PRINCIPLE given in (100), and repeated below:

(i) The STORE PRINCIPLE:

$$\begin{bmatrix} \textit{word} \\ \textsc{store}\ \ \textit{nelist} \end{bmatrix} \Rightarrow \boxed{1} \begin{bmatrix} \textsc{store}\ \left\langle \begin{bmatrix} \textit{lexical-rule} \\ \textsc{out}\ \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix}$$

FIGURE 8.1. The structure of the sentence *Pat kicked the bucket.*



Traditionally, the disjuncts in the WORD PRINCIPLE are called *lexical entries* (LE), the disjuncts in the ID PRINCIPLE are referred to as *Immediate Dominance schemata.* We will refer to the disjunct in the IIP as *phrasal lexical entries* (PLE), following the idea of Gazdar et al. 1985 that certain IEs are treated as complex lexical entities. The comparison of these two principles with the IIP shows that we assume a similar principle for all kinds of signs.[4]

[4]In the Section 8.3, we will simplify this approach even further by integrating the WORD PRINCIPLE and the IIP into a single LEXICON PRINCIPLE.

In the present section, we will only be concerned with two disjuncts of the IIP, the one needed for the IE *kick the bucket* and the one for the IE *trip the light fantastic*. In the tree in Figure 8.1 we have already seen the structure of a sentence which contains the IE *kick the bucket*. In (469), we give a first version of the the necessary disjunct of the IIP.[5]

(469) The phrasal lexical entry for the IE *kick the bucket* (first version):

$$
\mathbf{E}\boxed{1}\ \mathbf{E}\boxed{2}\ \mathbf{E}\boxed{3}\ \mathbf{E}\boxed{4}\ \mathbf{E}\ \boxed{5}
\left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{phrase}} \\
\text{PHON} & \boxed{3} \oplus \boxed{4} \\
\text{S L} & \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll}\text{HEAD} & \boxed{1}\\ \text{SUBCAT} & \langle\boxed{2}\rangle\end{array}\right] \\ \text{CONT} & \lambda x.\mathsf{die}'_{@}(x_{@})\end{array}\right]\\
\text{DTRS} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{head-compl-struc}} \\
\text{H-DTR} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{word}} \\
\text{PHON} & \boxed{3}\ \langle \textit{kick} \rangle \\
\text{S L} & \left[\text{CAT} \left[\begin{array}{ll}\text{HEAD} & \boxed{1}\ \textit{verb}\\ \text{SUBCAT} & \left\langle \boxed{2}, \boxed{5} \right\rangle\end{array}\right]\right]
\end{array}\right] \\
\text{N-DTR} & \left[\begin{array}{ll}
\text{PHON} & \boxed{4} \\
\text{SYNS} & \boxed{5} \left[\text{L} \left[\begin{array}{ll} \text{C} & \left[\begin{array}{ll}\text{HEAD} & \textit{noun}\\ \text{SUBCAT} & \textit{elist}\end{array}\right] \\ \text{CT} & \lambda P[\mathsf{the}\ y : \mathsf{bucket}'_{@}(y_{@})](P_{@}(y))\end{array}\right]\right] \\
\text{DTRS} & \left[\begin{array}{ll}
\text{H-DTR}\ldots & \left[\text{H-DTR} \left[\begin{array}{ll}\text{PHON} & \langle \textit{bucket} \rangle \\ \text{SYNS} & \text{N}\end{array}\right]\right] \\
\text{N-DTR} & \left[\begin{array}{ll}\text{PHON} & \langle \textit{the} \rangle \\ \text{SYNS} & \text{Det}\end{array}\right]
\end{array}\right] \\
\text{COLL} & \textit{minus}
\end{array}\right]
\end{array}\right]\\
\text{COLL} & \textit{plus}
\end{array}\right]
$$

The AVM in (469) describes a phrase which has a couple of regular properties and a number of irregular properties. Let us first consider the regular properties: The phrase is a VP and its HEAD value is identical to that of its head daughter (tag $\boxed{1}$). As the phrase has a [COLL +] specification, this identity is not a consequence of the HEAD FEATURE PRINCIPLE, but must be stated explicitly in the description of the IE. The PHON value of this phrase is just the concatenation of the PHON values of its daughters (the tags $\boxed{3}$ and $\boxed{4}$). The phrase also shows regular behavior with respect to the realization of complements: the SYNSEM value of the nonhead daughter is identical to the second element in the SUBCAT list of the head daughter (the tag $\boxed{5}$). The first element of this SUBCAT list appears as the only element in the SUBCAT list of the phrase (tag $\boxed{2}$). In regular phrases, these identities would follow from the HEAD COMPLEMENT SCHEMA in (86).[6]

Beside these regular properties of the phrase as a whole, all of its daughters are assumed to be regular as well: the head daughter is an instance of a regular verbal word. Even though the word is regular, the overall phrase does not allow arbitrary normal words to occur as its head daughter but imposes certain requirements: (i) the word must have the phonology *kick*, (ii) it must be a transitive verb, (iii) it must be in an active voice form.

Similarly, the complement daughter is required to be regular (thus the [COLL −] specification on the complement daughter). In addition, it must be an NP with a determiner with phonology *the* and a lexical head with the phonology *bucket*. The notation "H-DTR. . . " is

[5]Throughout this work, we ignore inflectional morphology, i.e., in the lexical entries we indicate some base form of the phonology.

[6]Note that we adopt a syntactic structure as introduced in Section 2.3, i.e., all subsorts of *const-struc* have the same two attributes, H-DTR and N-DTR.

meant to indicate that the lexical head need not be directly dominated by this NP, but can be further embedded in a path which contains only the attribute H-DTR. Such embedding is attested in the case of modification (*kick the proverbial bucket*).

In addition to these regular aspects, the phrase also contains an irregular property: so far, we have seen how the effects of several principles of regular combination are incorporated into the description in (469). We did, however, not mention the SEMANTICS PRINCIPLE. Instead of obeying this principle, the phrase introduces the semantic constant die' and does not specify any relation between its own CONTENT value and those of its daughters.

Before we proceed in the discussion of the properties of the proposal made in (469), we should notice that the "…"-notation used in the AVM above is quite informal, but it is easy to make it explicit, using a two-place relation `lexical-head` that holds between a sign and its lexical head. This relation is defined in (470).

(470) The relation `lexical-head`:

$$\texttt{lexical-head}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow} \left(\boxed{1}\left[word \right] \ \textbf{and}\ \ \boxed{1} \approx \boxed{2} \right)$$

$$\texttt{lexical-head}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow} \left(\begin{array}{l} \mathbf{E}\ \boxed{3} \\ \boxed{1}\begin{bmatrix} phrase & \\ \text{DTRS H-DTR} & \boxed{3} \end{bmatrix} \ \textbf{and}\ \ \texttt{lexical-head}(\boxed{3},\boxed{2}) \end{array} \right)$$

The relation `lexical-head` holds between a sign $\boxed{1}$ and a word $\boxed{2}$ either if $\boxed{1}$ is a word and $\boxed{1}$ and $\boxed{2}$ are identical, or if $\boxed{1}$ is a phrase and the word $\boxed{2}$ stands in the relation `lexical-head` with the head daughter of the phrase $\boxed{1}$. If we use this relation, we can replace the informal description of the complement daughter in (469) by a formally precise notation. In (471) we repeat the description of the IE *kick the bucket*, using this relation.

(471) The phrasal lexical entry for the IE *kick the bucket* (second version):

$$\mathbf{E}\boxed{1}\ \mathbf{E}\boxed{2}\ \mathbf{E}\boxed{3}\ \mathbf{E}\boxed{4}\ \mathbf{E}\boxed{5}\ \mathbf{E}\boxed{6}\ \mathbf{E}\boxed{7}$$

$$\begin{bmatrix} phrase & \\ \text{PHON} & \boxed{3} \oplus \boxed{4} \\ \text{S L} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \boxed{1} \\ \text{SUBCAT} & \langle \boxed{2} \rangle \end{bmatrix} \\ \text{CONT} & \lambda x.\mathsf{die}'_{@}(x_{@}) \end{bmatrix} \\ \text{DTRS} & \begin{bmatrix} head\text{-}compl\text{-}struc & \\ \text{H-DTR} & \begin{bmatrix} word & \\ \text{PHON} & \boxed{3}\ \langle kick \rangle \\ \text{S L} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \boxed{1}\ verb \\ \text{SUBCAT} & \left\langle \boxed{2}, \boxed{5} \right\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{N-DTR} & \begin{bmatrix} \text{PHON} & \boxed{4} \\ \text{S}\ \boxed{5}\ \text{L} & \begin{bmatrix} \text{C} & \begin{bmatrix} \text{HEAD} & noun \\ \text{SUBCAT} & elist \end{bmatrix} \\ \text{CT} & \lambda P[\mathsf{the}\ y : \mathsf{bucket}'_{@}(y_{@})](P_{@}(y)) \end{bmatrix} \\ \text{DTRS} & \begin{bmatrix} \text{H-DTR} & \boxed{6} \\ \text{N-DTR} & \begin{bmatrix} \text{PHON} & \langle the \rangle \\ \text{SYNS} & \text{Det} \end{bmatrix} \end{bmatrix} \\ \text{COLL} & minus \end{bmatrix} \end{bmatrix} \\ \text{COLL} & plus \end{bmatrix}$$

$$\textbf{and}\ \ \boxed{7}\begin{bmatrix} \text{PHON} & \langle bucket \rangle \\ \text{SYNS} & \text{N} \end{bmatrix} \ \textbf{and}\ \ \texttt{lexical-head}(\boxed{6},\boxed{7})$$

In the formally explicit notation in (471), we replaced the description of the head daughter of the NP by a tag $\boxed{6}$. Below the AVM, we added a relation call. This call specifies that the head daughter of the NP, i.e., the sign referred to by the tag $\boxed{6}$ stands in the relation `lexical-head` with a word that satisfies the description of the object referred to by the tag [7], i.e., this word must be a noun with the phonology *bucket*.

With these explications, it can be seen that the VP node in the tree structure given in Figure 8.1 is described by the AVM in (471). Below the VP node, there is a regular transitive verb with the phonology *kicked*. In English, there is such a verb, and its CONTENT value contains the constant kick′. Furthermore, the VP has a complement daughter which is an NP. This NP has the determiner *the* and a lexical head with phonology *bucket*. In English, the regular definite article and the normal word *bucket* fulfill these requirements. Thus, there is an NP with the regular meaning of *the bucket* as the complement daughter of the VP. As indicated in Figure 8.1, the VP itself contributes the semantic constant die′. This property is also encoded in the description in (471). Thus, the VP in Figure 8.1 satisfies the description of the IE *kick the bucket* as given in (471). Above the VP node, everything combines in a regular way: The subject *Pat* is realized via the HEAD-SUBJECT SCHEMA and its semantics combines with that of the VP via functional application, i.e., according to the SEMANTICS PRINCIPLE. Note that in the highest local tree in Figure 8.1, it is only the head daughter that is specified as [COLL +]. The mother is [COLL −] and, therefore, all the regular principles of grammar apply.

This analysis results in the following picture: the phrases that are dominated by the irregular VP are regular combinations, and the phrases that dominate the VP are regular, too. It is just the VP that is capable of blocking material from its daughters from appearing higher in the structure or of introducing new material.

We can show that the description in (471) accounts for the semantic and syntactic properties of the IE *kick the bucket* as collected in Chapter 6. According to the first and the second semantic criteria, the overall meaning of the IE cannot be arrived at by a regular combination of the meanings of its parts. This property follows from the fact that the CONTENT value of the phrase introduces material which is not present in the daughters.

We consider it an advantage of the present proposal that, as indicated by the CONTENT specifications in the tree in Figure 8.1, the words that constitute the IE appear in their regular meaning. Still this meaning is blocked from appearing as parts of the overall logical form of the sentence, as the VP is an irregular phrase and specifies its CONTENT value independently of that of the daughters. In that respect, our approach differs from Riehemann 1997 and Abeillé 1995. As we saw in (460) on page 314, Riehemann assumes special idiomatic versions of the words *kick*, *the* and *bucket* to constitute the IE. These idiomatic words differ from their regular counterparts, as they make an empty semantic contribution. In the TAG approach of Abeillé, given in (438) on page 294, there is also nothing that would indicate that the words used in the IE *kick the bucket* are regular words of English.

According to the observations concerning semantic modification, we have seen that the IE *kick the bucket* does not allow for semantically internal modification of the noun *bucket*. In the description of the nonhead daughter of the VP this is accounted for by the full specification of the logical form of the restriction of the generalized quantifier *the*. Thus, as the nonhead is required to have a logical form as given in (472), there cannot be any semantic material intervening between the semantic contribution of the lexical head of the NP (i.e., the noun *bucket* and that of the determiner *the*).

(472) $\lambda P.[\mathsf{the}\ y : \mathsf{bucket}'_{@}(y_{@})](P_{@}(y))$

Consider the sentence with an adjective inside the NP. In (473b) we give the logical form of the NP that would arise if the adjective is interpreted as an internal modifier. Clearly, this logical form does not have the shape required by the idiomatic VP, i.e., it is different from the term given in (472). This shows that we successfully exclude internal modification by a precise specification of the VP, without introducing internal irregularity into the NP.

(473) a. ... I kicked the social bucket.
b. $\lambda P.[\mathsf{the}\ y : \mathsf{social}'_{@}(y_{@}) \wedge \mathsf{bucket}'_{@}(y_{@})](P_{@}(y))$

We have shown how internal modification can be excluded in the case of the IE *kick the bucket*. It remains, however, to be shown how we can handle the fact that an adjectival modifier can be interpreted as modifying the entire VP, i.e., the fact that sentence (473a) is not ungrammatical but is synonymous to sentence (474). The description of the CONTENT value of the IE given in (469) does not allow such an interpretation.

(474) Socially, I kicked the bucket.

We will demonstrate that the phrasal lexical entry for the IE can be changed to make so called external modification possible. To do this, however, we will first have to provide a sketch of an account of external modification in general. As external modification does not directly concern our treatment of internally irregular IEs, we will devote a special short section at the end of this chapter (Section 8.4) to this topic.

The analysis that we are going to present in Section 8.4 is in two significant respects similar to the account in Abeillé 1995. There, an IE which allowed for semantically external modification contained a link from an N node in its syntactic tree to an F node in its semantic tree. In addition, a special kind of adjective was postulated that has the potential of attaching syntactically to a noun (an N node), but semantically to a formula (an F node).[7] Similarly, we will weaken the description of the CONTENT value of the VP *kick the bucket* and the NP *the bucket* in the phrasal lexical entry for the IE in such a way that the semantic contribution of an adjectival modifier can be applied to the semantic contribution of the IE. This is our analogy to the link in the TAG analysis. In addition, we will assign some adjectives a semantic type that allows them to take the VP instead of an N as their semantic arguments. This, of course, is parallel to the assumption of special auxiliary trees for the adjectives in Abeillé 1995

The phrasal lexical entry of the IE given in (471) provides a direct account of the ban on semantically internal modification. We have to be more vague, however, in the way we address the behavior of the IE with respect to the fourth semantic criterion, the question on pronominalizability. This is due to the fact that, within HPSG, we lack a general theory of pronominalizability which we could build our analysis on. In Chapter 6 we have seen that it is not possible to refer to the NP *the bucket* by a pronoun and at the same time preserve the idiomatic meaning. In (475), we repeat the relevant example.

(475) * Pat kicked [the bucket]$_i$ and Harry kicked it$_i$, too.

In Pollard and Sag 1994, nothing is said on the conditions under which discourse entities are introduced that may be referred to by a personal pronoun. For our purposes it is enough to mention that, if a discourse entity is introduced by the NP *the bucket*, it can always be eliminated at the level of the idiomatic VP *kick the bucket*.

We have now seen how the present approach accounts for the behavior of the IE *kick the bucket* with respect to the semantic criteria of regularity. Next, we will consider the syntactic criteria. We can show that the description of the IE given in (471) captures all the syntactic observations of Chapter 6.

[7]See the pair of elementary trees for the IE *kick the bucket* in (438) on page 294 and the elementary trees for the adjective *social* in (439a).

In Chapter 6 we have classified the IE *kick the bucket* as regular with respect to the first syntactic criterion, because it only consists of words which occur independently in the language. The description of the IE in (471) is only a description of the VP, therefore all of its daughters must be licensed by some other principle. In the case of the words, the only principle available is the Word Principle. As we have encoded a regular combination of the phon values of the daughters in the description of the IE, the words which compose the IE make their regular phonological contribution.

Our analysis, thus, appears to account for this syntactic regularity in a more adequate way than the approaches in Abeillé 1995 and Riehemann 1997. In Abeillé 1995, there was no relation at all between the normal words *kick*, *the*, and *bucket* and their occurrence in the IE. In Riehemann 1997 there is such a relation, but it is a relation between the lexical entries of the normal words and some idiomatic words which occur as parts of the IE. The idiomatic words do, however, differ significantly from their non-idiomatic counterparts as they have a different content value.

The IE *kick the bucket* is also regular with respect to the second syntactic criterion. In (471) the VP *kick the bucket* is specified in such a way that it dominates a syntactic structure which is just like that of a regular transitive VP, i.e., its head daughter is a verbal word which has two elements on its subcat list, and its nonhead is an internally regular NP ([coll −]) whose synsem value is identical to the second element on the verb's subcat list. The similarity to regular transitive VPs is enforced for the highest local tree by an explicit incorporation of the effect of the Head-Complement Schema. In that respect our proposal is similar to that of Abeillé 1995, where the entire syntactic structure of the IE is stipulated in the elementary tree. The difference between our approach and that of Abeillé is that the internally regular shape of the NP follows from the principles of the grammar. Thus, while we do impose quite strong requirements on the phonological and semantic contribution of the NP, it still counts as regular for the grammar.

The third syntactic criterion has to do with the syntactic aspects of modification, i.e., with the question of whether a modifier is allowed to appear inside the NP at all, independent of its interpretation. As shown in Chapter 6 and as illustrated again in example (473a), the IE *kick the bucket* allows modifiers to occur inside the NP. In our account, this is possible because all that we require, is that the NP start with the determiner *the* and have the noun *bucket* as its lexical head. The further syntactic details are left unexpressed. Therefore, an adjective such as *social* can occur inside the NP. It should, be noted though that the strong requirements on the content of the NP will exclude most adjectives from this position.

The fourth syntactic criterion concerns the possibility of an IE to occur in the passive. As we have seen in Chapter 6, the IE *kick the bucket* cannot be passivized. In the description of the IE in (471), we require the presence of an NP complement daughter in the VP. Thus, there is no way the IE can occur in passive. This specification indicates that a passivized form of the VP does not satisfy the description of the IE and, thus, sentence (476) cannot have the idiomatic meaning, which we indicate by the asterisk.

(476) * The bucket was kicked by Pat.

Similarly, we exclude topicalization (the fifth syntactic criterion) by the structural requirements imposed on the nonhead daughter. We require this daughter to be an NP headed by some word with the phonology *bucket*. Clearly, if the direct object is extracted, there is no such phonology. In the traceless analysis of extraction that we assume in this thesis, there is no NP complement at all, and, therefore, the entire syntactic structure would not have a VP that is compatible with the requirements of the idiomatic VP.[8]

[8] In an analysis with traces, such as the one in Pollard and Sag 1994, the complement daughter would be an NP with an empty phonology, thus equally violating the requirements on the phonology.

(477) * The bucket Pat kicked.

From this point on, it follows directly that relative clauses cannot be formed. The direct object NP cannot be extracted and, thus, it cannot take the form of a fronted relative pronoun (see example (478a)). Additionally, due to the ban on semantically internal modification, there may not be a relative clause attached to the noun *bucket*, i.e., sentence (478b) is equally excluded under a non-literal meaning.

(478) a. * The old lady wasn't aware of the bucket that she would soon kick.
b. * The old lady kicked the bucket that the murderer had planned for her.

This overview of the criteria of regularity has shown that our analysis accounts for the behavior of the IE.

In this section, we also want to provide an analysis of the second internally irregular IE from our list, the IE *trip the light fantastic*. In the discussion of the data, this IE was shown to be even more irregular than *kick the bucket*, because its internal syntactic structure appears to be unclear and it does not allow for modifiers to occur inside the expression. In our account, the IE will be treated as just another disjunct in the IIP, i.e., we will not introduce any new sorts or attributes to account for this IE. In (479) we indicate the disjunct in the IIP needed for this IE.

(479) The phrasal lexical entry for the IE *trip the light fantastic*:

$$\begin{bmatrix} \textit{phrase} & \\ \text{PHON} & \boxed{2} \oplus \boxed{3} \\ \text{S L} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \boxed{1} \\ \text{SUBCAT} & \left\langle \left[\textit{synsem}\right] \right\rangle \end{bmatrix} \\ \text{CONT} & \lambda x.\mathsf{dance}'_{@}(x_{@}) \end{bmatrix} \\ \text{DTRS} & \begin{bmatrix} \text{H-DTR} & \begin{bmatrix} \textit{word} & \\ \text{PHON} & \boxed{2}\ \langle \textit{trip} \rangle \\ \text{S L C HEAD} & \boxed{1}\ \textit{verb} \end{bmatrix} \\ \text{N-DTR} & \begin{bmatrix} \textit{phrase} & \\ \text{PHON} & \boxed{3}\ \langle \textit{the light fantastic} \rangle \end{bmatrix} \end{bmatrix} \\ \text{COLL} & \textit{plus} \end{bmatrix}$$

Interestingly, even though this IE is the most irregular in our collection, it can be described be a relatively simple AVM. In (479), we specify that the overall semantic contribution is that of dance'. As the entire phrase is given the specification [COLL +], it is identified as internally irregular. Therefore, the overall content need not stand in a combinatorial relation to the contents of the daughters. The specification of the daughters in (479) is much more vague than what we had to assume for the IE *kick the bucket* in (471). In the case of *trip the light fantastic*, it suffices to specify that the head daughter is the verb *trip* and that the nonhead daughter has the phonology *the light fantastic*. The phonologies are combined by simple concatenation ($\boxed{2} \oplus \boxed{3}$).

The effect of the phrasal lexical entry in (479) can best be illustrated if we go through the criteria of regularity. In Chapter 6 we have seen that the IE *trip the light fantastic* fails all semantic criteria of regularity, as does the IE *kick the bucket*. This is accounted for in the same way in our treatment of these IEs: in both cases, the CONTENT value of the idiomatic VP does not combine the CONTENT values of its daughters, but introduces completely new material into the logical form of the sentence, while preventing the semantic contributions of its daughters from being part of the logical form of the sentence.

The two IEs differ, however, with respect to their syntactic regularity. The only uncontroversially regular syntactic aspect of the IE *trip the light fantastic* is that all the words that it is composed of also occur in free combinations. But, as we have pointed out in Chapter 6, the syntactic category of the words *light* and *fantastic* cannot be determined. The description in (479) captures this by specifying that its daughters are a verb *trip* and some phrase with the phonology *the light fantastic*. Any phrase that provides exactly this phonology can serve as a nonhead daughter for the idiomatic VP, i.e., we are not forced to make a commitment to a particular syntactic structure of this complex. We know from the discussion in Chapter 6, page 254, that English provides some constructions which would generate exactly this phonology.

Whereas the IE *kick the bucket* has the internal structure of an ordinary transitive VP, the syntactic structure of *trip the light fantastic* is unclear. This is reflected by the "underspecified" characterization of the nonhead daughter in (479). This means, however, that the IE *trip the light fantastic* has some fully specified syntactic structure in every concrete utterance which contains the IE. The syntactic structures that the IE takes will, however, differ largely from one utterance to the other. This accounts quite nicely for the fact that, as indicated in Table 6.1, we are not certain on whether the IE is of a syntactically regular shape or not.

The third syntactic criterion of Chapter 6 concerns the question of whether a modifier is permitted to occur inside the IE. In our discussion of the data, we have seen that only the modifier *proverbial* can be used in this way, but, even in this case the sentence is judged relatively low on the grammaticality scale. In (480), we repeat the relevant example.

(480) ?* Let's go out tonight and trip the proverbial light fantastic.

As it stands, the phrasal lexical entry for *trip the light fantastic* does not allow for any modification inside the complex *the light fantastic* because we have fixed the phonology of the nonhead daughter. To make the grammar admit sentence (480), we can simply make the adjective *proverbial* an optional part of the IE, i.e., the PHON specification in for the IE would have to be changed to be as in (481).

(481)
$$\left[\begin{array}{l} \ldots \\ \text{DTRS} \left[\begin{array}{l} \ldots \\ \text{N-DTR} \left[\text{PHON } \boxed{3} \langle \textit{the (proverbial) light fantastic}\rangle\right]\end{array}\right]\end{array}\right]$$

This is of course ad hoc. For a more adequate solution, more must be known about the special status of the adjective *proverbial*. In particular, we would have to know whether the adjective has any effect on the truth-conditional meaning of an utterance or whether it only contributes meta-linguistic information. In the latter case, there is no hope of a more adequate solution until a theory of meta-linguistic use has been proposed for HPSG.

With respect to the other syntactic criteria (passivizability, extractability, and the ability of having a relative pronoun as the nonhead daughter), the IE behaves like the IE *kick the bucket*, i.e., it is fully fixed. This is accounted for, as in the case of *kick the bucket*, by the fact that we have specified the phonology of the nonhead daughter.

This discussion of our account of the behavior of *trip the light fantastic* with respect to the criteria of regularity has shown that the phrasal lexical entry in (479) captures the fact that the IE is irregular in nearly all respects. In particular, the syntactic irregularity which is not present in the case of the IE *kick the bucket* has received a straightforward account by simply leaving the internal syntactic structure of the IE unspecified, but by fully specifying the resulting phonologies.

In this section, we have provided an HPSG analysis of two internally irregular IEs. Building on the empirical insights of Wasow et al. 1983 and the analytic proposal of Abeillé 1995, we have provided an architecture for HPSG that allows for syntactically complex lexical items, i.e., internally irregular phrases. These irregular phrases are licensed by what we have called phrasal lexical entries (PLE).

PLEs license signs which share properties of both words and phrases: just like words, they do not have a regular internal structure and the principle that licenses these signs, the IIP, contains many disjuncts like the Word Principle, not just a handful as in the case of the ID Principle. On the other hand, signs licensed by a PLE are syntactically complex, just like phrases, i.e., in contrast to words, they have a DTRS attribute. Moreover, as they are phrases, derivational rules cannot be applied to them.

When we introduced the semantic framework LF-Ty2 in Section 1.3.3, we provided shifting operations that are needed to account for scope phenomena in natural language semantics. In our integration of these shifting rules into our RSRL grammar in Chapter 4, we have shown that we must allow the application of these rules on derived and non-derived words (see in particular Section 4.3). So far, we did not apply these rules to phrases.

In (482) and (483) we indicate that there are *de dicto* and *de re* readings for the subject NP of the IEs *kick the bucket* and *trip the light fantastic*.

(482) John believes that a goldfish kicked the bucket (while he was on vacation).
a. *de dicto* reading: $\mathsf{believe}'_@(j, \lambda @.\exists y[\mathsf{goldfish}'_@(y_@) \wedge \mathsf{die}'_@(y_@)])$
b. *de re* reading: $\exists y[\mathsf{goldfish}'_@(y_@) \wedge \mathsf{believe}'_@(j, \lambda @.\mathsf{die}'_@(y_@))]$

(483) John believes that some woman trips the light fantastic.
a. *de dicto* reading: $\mathsf{believe}'_@(j, \lambda @.\exists y[\mathsf{woman}'_@(y_@) \wedge \mathsf{dance}'_@(y_@)])$
b. *de re* reading: $\exists y[\mathsf{woman}'_@(y_@) \wedge \mathsf{believe}'_@(j, \lambda @.\mathsf{dance}'_@(y_@))]$

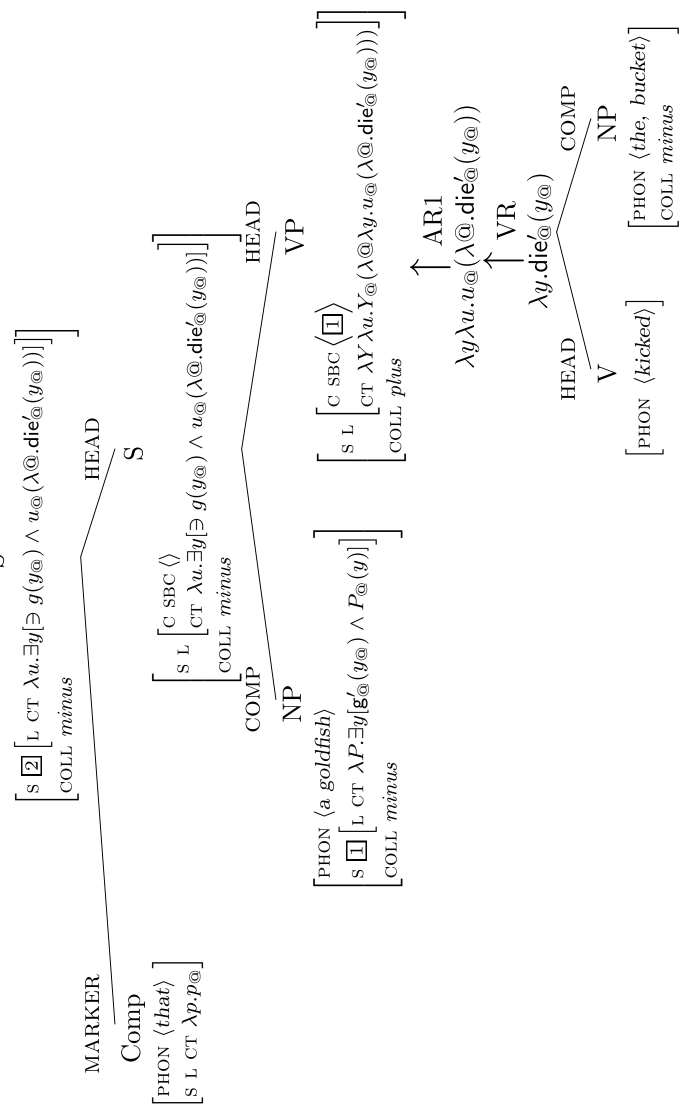
In the discussion of the framework LF-Ty2 we have seen that to derive a *de re* reading as those indicated in (482b) and (483b), we must apply *value raising* in combination with *argument raising* to the logical form of the main semantic functor of the embedded clause. In the cases that we studied so far, the main semantic functor of the embedded clause was the embedded verb. In the idiomatic cases, the main semantic functor of the embedded clause cannot be the embedded verb, *kick* or *trip*, because the idiomatic VPs block the semantic contribution of these verbs from occurring in the logical form of the embedded clause. Instead, the logical form of the idiomatic VPs must be considered the main semantic functors of the embedded clause.

To derive the *de re* readings it must be possible to apply shifting operations to the logical form of idiomatic VPs. In Figure 8.2 (page 331) we give the derivation of the embedded clause of sentence (482) under the *de re* reading. In Figure 8.3 (page 332) we indicate how this combines with the matrix clause to yield the overall logical form given in (482b).

As shown in Figure 8.2, the embedded VP of sentence (482) is irregular, i.e., it has the specification [COLL *plus*]. It dominates the verb *kick* and the regular NP *the bucket*. So far, everything is just as discussed for Figure 8.1 (page 322). The logical form of the VP is, however, not $\lambda y.\mathsf{die}'_@(y_@)$, but a shifted form thereof. In the derivation of the *de re* reading of this sentence, we apply exactly the same shifting operations that we used in the derivation of the *de re* readings of sentence (39c) in Section 1.3.3 above. In (484a), we repeat this sentence, together with the logical form of the embedded verb *walks* (b), and that of the embedded clause (c), as needed for the *de re* readings. The derivation of the *de re* readings was given in Figures 1.14 and 1.15.

(484) a. Every man believes that some woman walks.
b. Logical form of the embedded verb under the *de re* readings:
$\lambda Y \lambda u.Y_@(\lambda @ \lambda y.u_@(\lambda @.\mathsf{walk}'_@(y_@)))$
c. Logical form of the embedded clause under the *de re* readings:
$\lambda u.\exists y[\mathsf{woman}'_@(y_@) \wedge u_@(\lambda @.\mathsf{walk}'_@(y_@))]$

FIGURE 8.2. The embedded clause in the derivation of the *de re* reading of sentence (482):

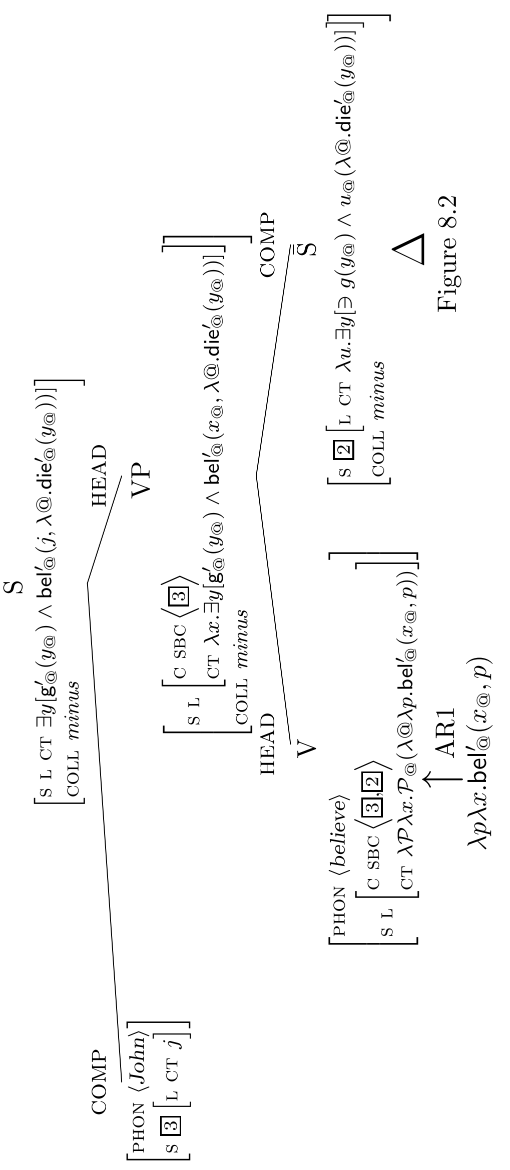


As can be seen in Figure 8.2, the logical form of the embedded VP for the *de re* reading of sentence (482) is of the same shape as the logical form given in (484b).

The irregular VP combines with its subject in a regular way to yield a logical form that is of the same shape as the logical form in (484c). For the complementizer *that* we assume a logical form as given in (178b), i.e., an identity function. Therefore, the logical forms of the embedded S node and of the embedded S̄ node are the same.

In Figure 8.3 it is shown how the embedded S̄ node is integrated into the matrix clause. For this purpose, we must execute some shifting operation on the matrix verb, just as we did in the derivation of the ∀∃-*de re* reading of sentence (484a) in Figure 1.14. The shifted logical form of the matrix verb combines with the logical form of the embedded clause which leads to a logical form in which the constant believe′ is in the scope of the quantifier contributed by the embedded subject. This combines with the logical form of the matrix

FIGURE 8.3. The matrix clause in the derivation of the *de re* reading of (482):



subject, *John*, by intensional functional application to yield the logical form of the *de re* reading of the sentence.

The derivation of the *de re* reading of sentence (482) has shown that we must allow shifting operations to apply to irregular phrases just as they are applicable to simple and derived words. For our architecture of grammar, this has the consequence that we cannot encode shifting operations as derivational rules. Thus, we can no longer maintain the DR-encoding to shifting operations as presented in Section 4.1, but we must adopt the LE-encoding developed in the same section. Thus, even though we could not find empirical arguments in favor of either encoding of shifting operations in Section 4.1, the combination of LF-Ty2 with our analysis of internally irregular IEs forces us to chose for a LE-encoding.

Under the LE-encoding, we include the availability of shifting operations in the lexical entry. For *phrasal lexical entries* of irregular VPs such as *kick the bucket* or *trip the light fantastic*, this means that we should change the PLEs given so far to allow for the application of shifting operations. In (485) this is done for the PLE of *trip the light fantastic*.

(485) The phrasal lexical entry for the IE *trip the light fantastic*, including the availability of shifting operations:

$$\mathbf{E}\boxed{1}\ \mathbf{E}\boxed{2}\ \mathbf{E}\boxed{3}\ \mathbf{E}\boxed{4}\ \mathbf{E}\boxed{5}\ \left[\begin{array}{ll} \textit{phrase} & \\ \text{PHON} & \boxed{2} \oplus \boxed{3} \\ \text{S L} & \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \text{HEAD} & \boxed{1} \\ \text{SUBCAT} & \left\langle \left[\textit{synsem}\right] \right\rangle \end{array}\right] \\ \text{CONT} & \boxed{4} \end{array}\right] \\ \text{DTRS} & \left[\begin{array}{ll} \text{H-DTR} & \left[\begin{array}{l} \textit{word} \\ \text{PHON}\ \boxed{2}\ \langle \textit{trip} \rangle \\ \text{S L C HEAD}\ \boxed{1}\ \textit{verb} \end{array}\right] \\ \text{N-DTR} & \left[\begin{array}{l} \textit{phrase} \\ \text{PHON}\ \boxed{3}\ \langle \textit{the light fantastic} \rangle \end{array}\right] \end{array}\right] \\ \text{COLL} & \textit{plus} \end{array}\right] \quad \begin{array}{l} \textbf{and}\ \boxed{5}\left[\lambda x.\mathsf{dance}'_{@}(x_{@})\right] \\ \textbf{and}\ \texttt{shifting}(\boxed{5},\boxed{4}) \end{array}$$

The PLE in (485) differs from that in (479) in the way the logical form is specified. In (479), the term $\lambda x.\mathsf{dance}'_{@}(x_{@})$ appeared as the description of the CONTENT value of the VP directly, whereas in (485) it is the description of some *me* object $\boxed{5}$. This *me* object stands in the relation `shifting` with some other *me* object, $\boxed{4}$, which is the CONTENT value of the VP. We consider the term $\lambda x.\mathsf{dance}'_{@}(x_{@})$ the *basic translation* of the VP.

In (485) we have chosen the format of the LE-encoding as exemplified in (173), i.e., the PLE in (483) includes shifting operations, but not λ-conversion. It is straightforward to change this PLE to take one of the formats discussed in Section 4.2. To avoid complications in further sections of this chapter, we will usually indicate the basic translations of non-derived words and internally irregular phrases as the CONTENT specification in the lexical entries. This is to be understood as a shorthand for some LE-encoding of shifting operations.

This discussion shows that our analysis of internally irregular IEs interacts in a natural way with the semantic framework LF-Ty2 as developed in Part I of this study.

In this section, we have considered the PLEs for two internally irregular idiomatic expressions. Surprisingly enough, while we introduce internally irregular phrases into HPSG, this does not mean that whatever is dominated by such an internally irregular phrase is irregular as well. Instead, we claim that an internally irregular phrase normally dominates fully regular signs. The irregularity lies in the fact that the internally irregular phrase need not combine the properties of its daughters in the usual, i.e., regular way. In the examples that we have considered in this section, in particular, the semantic contribution of the irregular phrase was not a combination of the semantic contributions of its daughters.

The proposed analysis of internally irregular IEs is inspired by the proposals made in Abeillé 1995 and Riehemann 1997, but it also differs from them significantly.

Like Abeillé 1995, we assume that the logical form of an IE need not be composed of the regular semantic contributions of its parts. In addition, internally irregular IEs are treated as a unit in both approaches. Yet, we differ from her approach in various respects. In Abeillé's account, the IEs that we considered in this section are fully specified elementary trees. In our approach, the PLEs that license the IEs do specify large parts of the properties of the components of the IE, these component, however, are realized as regular signs. Thus, in our analysis of a sentence such as *Pat kicked the bucket* in Figure 8.1 on page 322, there is just one node in a tree that is idiomatic: the VP. In Abeillé's account, everything dominated by this VP would be part of the elementary tree of the idiomatic expression.

The approach in Riehemann 1997 shares the last mentioned property with our approach. In contrast to our analysis, however, Riehemann assumes that internally irregular IEs are composed of idiomatic words. Riehemann is forced to this assumption, because she applies the regular SEMANTICS PRINCIPLE to all idioms. In our approach, all internally irregular phrases are liberated from the SEMANTIC PRINCIPLE and can, therefore, ignore the semantic contribution of the daughters. As a final consequence of this, we can allow the daughters of irregular phrases to be fully regular.

In the following section, we will extend our analysis to internally regular IEs. This extension will reveal further similarities between our account and that of Gazdar et al. 1985. Still, while the overall architecture of the analysis is similar, we will see that the concrete technical solution adopted here is new.

8.2. Internally Regular IEs

In the summarizing remarks on the data in Section 6.3 we made a clear distinction between IEs that we classified as internally irregular and those, classified as internally regular. In the presentation of the account of IEs in Gazdar et al. 1985 the very same distinction was made. While the authors did not provide an explicit account of internally irregular IEs, they showed that it is in principle possible to treat internally regular IEs very much like free combinations, provided one has some mechanism at one's disposal which permits the expression of contextual restrictions. In Gazdar et al. 1985 such a mechanism was the use of partial functions for the interpretation of semantic constants.

In the preceding section, we presented an analysis of internally irregular IEs as phrases, licensed by a special principle, the IIP. For internally regular IEs we will follow the assumption of Gazdar et al. 1985 and treat the VPs as regular combinations which do, however, contain elements that are subject to strict occurrence restrictions. Our analysis differs from the GPSG approach in the kind of mechanism that we assume for expressing the occurrence restrictions. The mechanism that we are going to propose is based on an extension of the use of the COLL feature: we will allow the COLL feature to take as value a specification of the linguistic context within which a certain word is allowed to occur.

The idiomatic expressions that we are going to analyze here are *spill the beans*, *pull strings*, *make headway* and *make a decision*. For them, our syntactic criteria of regularity in Chapter 6 have shown that they are of a syntactically regular internal structure and that there is some range of syntactic environments in which the parts of the IEs may occur while conserving the meaning of the IE. Similarly, our semantic criteria have revealed that for all these IEs, we can assign some meaning to their parts and that we can calculate the meaning of the overall expression by combining the meaning of its parts in a regular way.

In the present section, we will first concentrate on one particular IE, *spill the beans*. In Section 8.2.1, we give an informal sketch of our analysis. Following this, we present a formalization of this analysis based on the COLL feature in Section 8.2.2. The remaining IEs are then addressed in Section 8.2.3.

8.2.1. An Informal Outline of the Analysis.

The basic difference between internally irregular and internally regular IEs lies in the fact that the latter seem to obey more principles of regular combination than the former. In addition, internally regular IEs are more flexible with respect to semantic modification and the syntactic constructions that they can occur in. In the first two parts of this section, we will restrict our attention to a single, internally regular IE *spill the beans*.

In (486) we sketch lexical entries for the words *spill* and *beans* as they occur in the IE. Note that these lexical entries basically differ from those of the normal words by introducing a different semantics. For ease of exposition, we choose the constants spill'' for the idiomatic meaning of *spill* and beans'' for the idiomatic meaning of *beans*, just as assumed in Gazdar et al. 1985.[9] Just as in Gazdar et al. 1985, we assume that the interpretation of the constant spill'' is more or less like that of *disclose*, and the interpretation of beans'' is more or less that of *information*. We are, however, not forced to assume a partial interpretation function nor to assume that there are any two constants with different interpretations.

(486) a. Lexical entry for the idiomatic use of *spill* (preliminary):

$$\begin{bmatrix} \textit{word} & \\ \textsc{phon} & \langle \textit{spill} \rangle \\ \textsc{s l} & \begin{bmatrix} \textsc{cat} & \begin{bmatrix} \textsc{head} & \textit{verb} \\ \textsc{subcat} & \left\langle \text{NP, NP, (PP}[\textit{to}]) \right\rangle \end{bmatrix} \\ \textsc{cont} & \lambda y \lambda x.\mathsf{spill}''_{@}(x_{@}, y_{@}) \end{bmatrix} \end{bmatrix}$$

b. Lexical entry for the idiomatic use of *beans* (preliminary):

$$\begin{bmatrix} \textit{word} & \\ \textsc{phon} & \langle \textit{beans} \rangle \\ \textsc{s l} & \begin{bmatrix} \textsc{cat} & \begin{bmatrix} \textsc{head} & \textit{noun} \\ \textsc{subcat} & \left\langle \text{Det} \right\rangle \end{bmatrix} \\ \textsc{cont} & \lambda x.\mathsf{beans}''_{@}(x_{@}) \end{bmatrix} \end{bmatrix}$$

Given these two lexical entries, we can see how the idiomatic reading of the IE is derived as a regular syntactic and semantic combination of the words licensed by these lexical entries. In Figure 8.4 we give the structure for a simple example sentence.

The tree given in Figure 8.4 has the same structure as the sentence containing the IE *kick the bucket* (Figure 8.1 on page 322). The crucial difference between the two structures is, however, that in Figure 8.4 all phrases have the specification [COLL −], whereas in the case of the IE *kick the bucket*, the VP node was assumed to be [COLL +]. As a consequence of the [COLL −] specification in the case of Figure 8.4, all principles of regular combination apply to the VP. In the tree in Figure 8.4 we have indicated the CONTENT values of the signs. As can be seen, the normal SEMANTIC PRINCIPLE is satisfied, i.e., the CONTENT value of the VP is the result of applying the CONTENT value of the head daughter to that of the nonhead daughter.[10]

Before we consider what is still missing in this proposal, let us first see the effect that it already has as it stands. So far, all we did was stipulate two new lexical entries for the words *spill* and *beans*. As these words can combine freely, and as each of them makes its own semantic contribution, we arrive at the idiomatic VP by regular syntactic combination and at the idiomatic meaning by regular semantic combination. As there is no internally irregular phrase involved, we expect a certain degree of syntactic and semantic flexibility, which is actually attested: adjectival modifiers are interpreted as modifying the noun *beans* (487a), and the noun *beans* need not be realized as the syntactic direct object, but it can appear as a grammatical subject in passive and raising (487b).

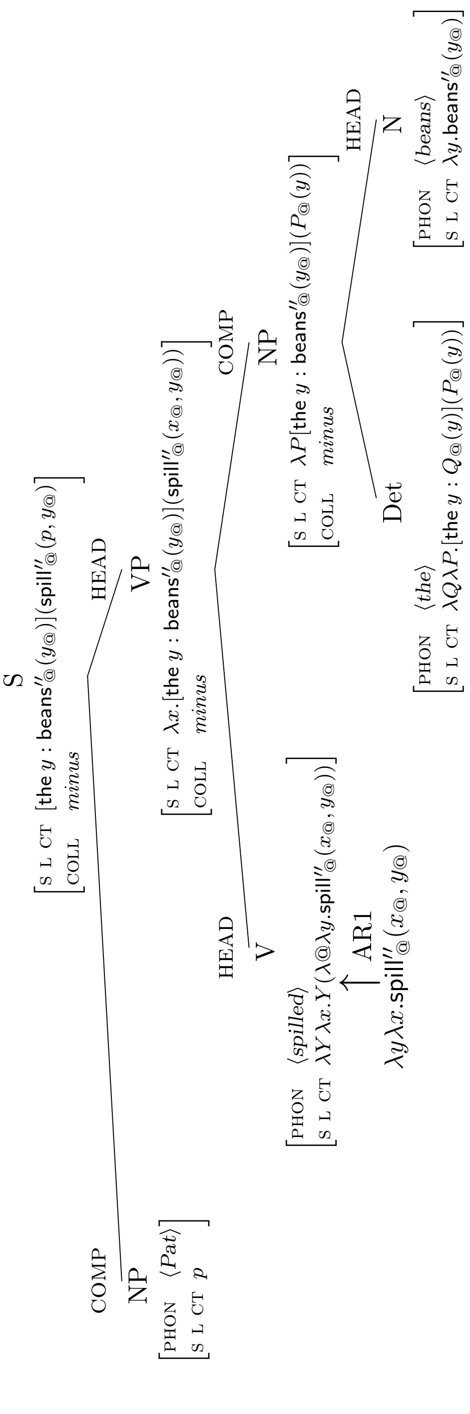
(487) a. He spilled the sordid beans to his parents.
b. I was worried that the beans might be spilled.

So far, the approach is able to cover those observations that confirmed the regular character of the IE. If we assume just the two lexical entries in (486), we fail to account for

[9]See the terminated local trees assumed in Gazdar et al. 1985, given in (414) on page 279.

[10]Note that we had to raise the semantic type of the verb in order to make the types of the verb and the direct object compatible with each other. In the structure in Figure 8.4, we have explicitly stated the logical form of the verb before and after the application of the shifting rule Argument Raising of Section 1.3.3.

FIGURE 8.4. The structure of the sentence *Pat spilled the beans.*:



the irregular properties of the IE. Among these irregular properties, we listed the following in Chapter 6: (i) the idiomatic reading of the words *spill* and *beans* is not possible outside the particular IE, (ii) the NP *the beans* cannot be topicalized, and (iii) the NP cannot take the form of a fronted relative pronoun.

The first irregular property has been demonstrated by the absence of the idiomatic interpretation of the noun *beans* in sentences where we replace the idiomatic *spill* by a near synonym. Similarly, replacing the noun *beans* renders the sentence ungrammatical in non-slang uses.

(488) a. She divulged the story to the media.
b. * She divulged the beans to the media.
c. * She spilled the story to the media.

The examples in (488) seem to indicate that the idiomatic words *spill* and *beans* require the presence of each other within the same sentence. What kind of requirement is this? In the approach of Gazdar et al. 1985 this was expressed as a restriction on the interpretability of semantic constants. While we agree with Gazdar et al. 1985 that the restriction should be formulated in terms of the semantic constants, we state the restriction as a condition on the possible logical forms in which the two constants spill'' and beans'' can occur. Before we present the RSRL formalization of this idea, we will work with an informal formulation of the distributional requirements of the idiomatic uses of *spill* and *beans*. Such an informal statement is given in (489):

(489) a. The word *spill* with the meaning spill'' can only occur in a sentence if, in the logical form of this sentence, the constant spill'' occurs in the following constellation:
$$[\mathsf{the}\ y : \ldots \mathsf{beans}''_{@}(y_{@}) \ldots](\ldots \mathsf{spill}''_{@}(x_e, y_{@}) \ldots)$$
b. The word *beans* with the meaning beans'' can only occur in a sentence if, in the logical form of this sentence, the constant beans'' occurs in the following constellation:
$$[\mathsf{the}\ y : \ldots \mathsf{beans}''_{@}(y_{@}) \ldots](\ldots \mathsf{spill}''_{@}(x_e, y_{@}) \ldots)$$

Clearly the logical forms of the sentences in Figure 8.4 and (487) satisfy these two occurrence conditions. In (490) we indicate the CONTENT values of the relevant sentences.

(490) a. Pat spilled the beans.
$[\mathsf{the}\ y : \mathsf{beans}''_{@}(y_{@})](\mathsf{spill}''_{@}(p, y_{@}))$
b. He spilled the sordid beans (to his parents).
$[\mathsf{the}\ y : \mathsf{sordid}'_{@}(y_{@}) \wedge \mathsf{beans}''_{@}(y_{@})](\mathsf{spill}''_{@}(x_{@}, y_{@}))$
c. (I was worried that) the beans might be spilled.
$\exists x([\mathsf{the}\ y : \mathsf{beans}''_{@}(y_{@})](\Diamond \mathsf{spill}''_{@}(x_{@}, y_{@})))$

The logical form in (490a) is that of the clause in Figure 8.4. The two idiomatic words satisfy the occurrence restrictions in (489): both words occur in a sentence whose logical form matches the required pattern.

In the logical form of sentence (490b) there is additional semantical material in the restriction of the quantifier the. Still, the logical form of the sentence is just as required in (489). As both idiomatic words occur in this sentence, the distributional requirements stated above are satisfied.

Finally, in sentence (490c) there is additional material in the nuclear scope of the quantifier, the modality operator contributed by the modal verb *might*. Following the tradition of GPSG and Pollard and Sag 1994, we assume that a passive sentence has basically the same logical form as its active counterpart. The only difference between the active and the passive sentence that can be seen in the example is that we assume an existential quantifier to bind the unexpressed first semantic argument. All these changes, however, do not effect the fact that the logical form of the clause which contains the words *spill* and *beans* is as required by the distributional requirements in (489).

Let us, next, consider the logical forms of the ungrammatical sentences in (488). There, only one of the constants spill'' and beans'' is present. Thus, the requirements given above are not satisfied.

(491) a. * She divulged the beans (to the media).
$[\mathsf{the}\ y : \mathsf{beans}''_{@}(y_{@})](\mathsf{divulge}'_{@}(x_{@}, y_{@}))$
b. * She spilled the news (to the media).
$[\mathsf{the}\ y : \mathsf{news}'_{@}(y_{@})](\mathsf{spill}''_{@}(x_{@}, y_{@}))$

In sentence (491a), the word *spill* does not occur at all. Therefore, the constant spill'' is absent from the logical form of this sentence. Thus, clearly the distributional requirements of the idiomatic use of *beans* are not satisfied.

Similarly, in the case of sentence (491b), the idiomatic word *spill* occurs. But, as there is no occurrence of the idiomatic word *beans*, the constant beans'' is not present in the logical form of the sentence. Thus, the requirements for the idiomatic word *spill* are not met.

This discussion shows that adopting the distributional restrictions in (489) we achieve an effect similar to that of the partial function approach in Gazdar et al. 1985: the idiomatic uses of the words *spill* and *beans* are restricted to sentences in which both occur. Still, we think, that expressing these conditions in terms of the CONTENT values of a particular form has an important advantage over the partial function approach.

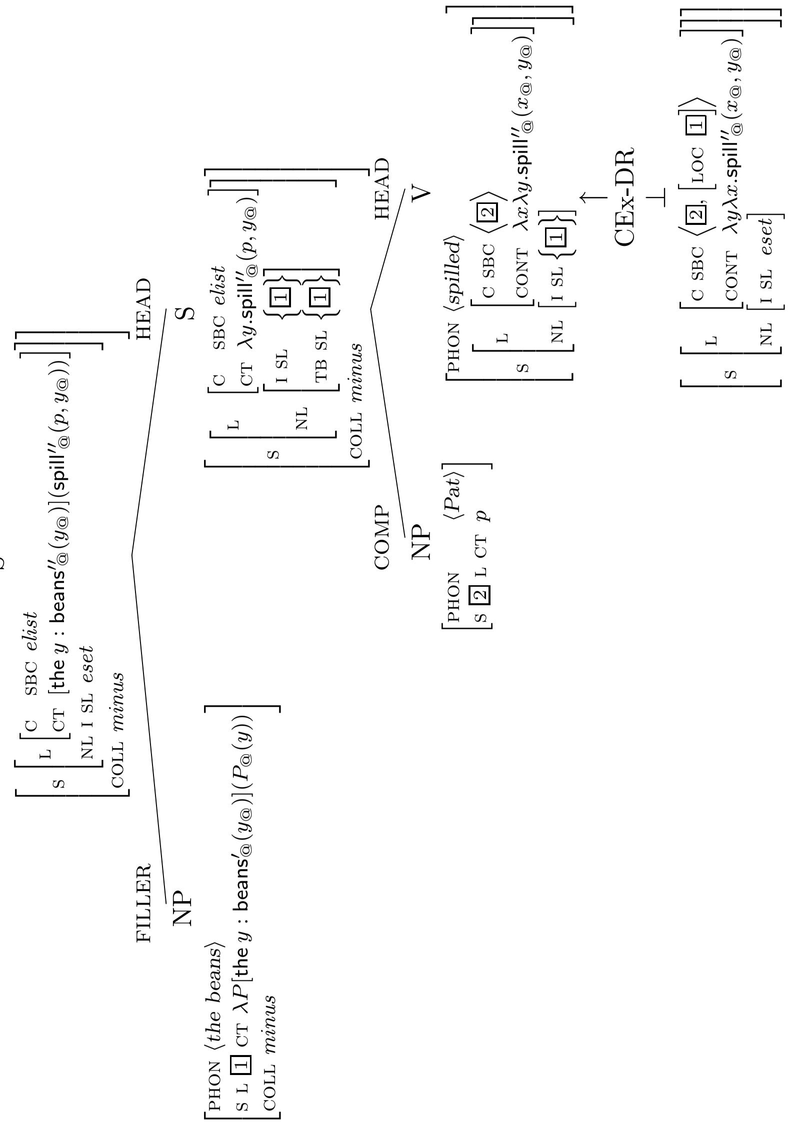
In our presentation of the partial function approach in Section 7.1 we discussed the objections of Pulman 1993. Pulman has pointed out that the GPSG account necessitates the assumption of a huge number of idiomatic semantic constants in order to allow parts of an IE to occur separated by other material. In the simplest example, Pulman argues, at least a special quantifier the'' would be needed to mediate the combination of the idiomatic spill'' and the idiomatic beans''. In our logical form approach, no such additional idiomatic constants are needed, i.e., the only special constants that we must introduce are those needed for the words that constitute the given idiomatic expression. For all other words, we can assume the usual semantic representation.

This illustration of the effect of the informally stated distributional requirements of the words *spill* and *beans* in (489) has, so far, focussed on the shape of the logical form of some constituent that dominates a given word. This allowed us to account for the first of the three mentioned irregular properties of the IE *spill the beans*, the fact that the idiomatic use of one word must co-occur with the idiomatic use of the other word. Without argument, we considered the CONTENT value of the smallest clause that dominates the idiomatic use of the word *spill* or *beans* to be the relevant logical form. Next, we will next discuss whether this was the right assumption. We will use the other two irregular aspects of the IE for this argument, i.e., the observation that the NP *the beans* can neither be topicalized nor given the shape of a fronted relative pronoun. We will first consider the case of topicalization.

In Chapter 4 we presented how topicalization structures are interpreted in the framework of LF-Ty2. We assumed a traceless analysis. We defined a derivational rule, the Complement Extraction DR (CEx-DR) that removes an element from the SUBCAT and has the effect of altering the order of the semantic arguments of the verb in such a way that the lambda operator that binds the variable associated with the argument slot of the extracted argument appears as the last lambda operator. As a consequence, the semantics of the filler constituent will not be present until the filler is overtly realized. For illustration, consider the tree in Figure 8.5 which gives the hypothetical syntactic structure of the ungrammatical sentence in (492).

(492) * The beans Pat spilled.

Figure 8.5. The structure of the sentence **The beans Pat spilled.*:



The figure not only gives us the syntactic structure, it also shows the effect of the Complement Extraction DR (CEx-DR). For ease of reference, in (493), we repeat the CEx-DR in the informal notation given in (283).

(493) Informal specification of the Complement Extraction DR as given in (283):

$$\begin{bmatrix} \textit{word} \\ \text{SYNS} \begin{bmatrix} \text{LOC} \begin{bmatrix} \text{CAT SUBCAT } \boxed{1}\begin{bmatrix}\textit{nelist}\end{bmatrix} \oplus \left\langle \begin{bmatrix}\text{LOC } \boxed{2}\end{bmatrix} | \boxed{3} \right\rangle \\ \text{CONT } \lambda x_n \ldots \lambda x_i \ldots \lambda x_1.\phi \end{bmatrix} \\ \text{NONL INH SLASH } \boxed{4} \end{bmatrix} \end{bmatrix} \longmapsto$$

$$\begin{bmatrix} \textit{word} \\ \text{SYNS} \begin{bmatrix} \text{LOC} \begin{bmatrix} \text{CAT SUBCAT } \boxed{1} \oplus \boxed{3} \\ \text{CONT } \lambda x_n \ldots \quad \ldots \lambda x_1 \lambda x_i.\phi \end{bmatrix} \\ \text{NONL INH SLASH } \boxed{4} \cup \left\{\boxed{2}\right\} \end{bmatrix} \end{bmatrix}$$

(where $\begin{bmatrix}\text{LOC } \boxed{2}\end{bmatrix}$ is the i-th element on the SUBCAT list of the input word.)

The input to the DR is a word as licensed by the lexical entry of the idiomatic use of *spill* in (486a), i.e., it is a transitive verb with a CONTENT value of the form $\lambda y \lambda x.\mathsf{spill}''_{@}(x_{@}, y_{@})$. The lexical rule removes the second element from the SUBCAT list and introduces an element into the INHERITED SLASH set whose LOCAL value is identical with that of the removed element. In addition, the order of the semantic arguments is changed, i.e., the lambda operator λy which corresponds to the second element on the subcat list is put to the end of the lambda operators in the output content value. Thus, the verb enters the syntactic structure with a subcat list with a single element on it, with a CONTENT value of the form $\lambda x \lambda y.\mathsf{spill}''_{@}(x_{@}, y_{@})$ and with a non-empty INHERITED SLASH set.

This verbal word combines directly with its subject, the NP *Pat*. The mother node inherits the slash specification of the head daughter according to the NONLOCAL FEATURE PRINCIPLE. The resulting phrase has an empty SUBCAT list, but is not yet saturated semantically, i.e., it has a CONTENT value of the form $\lambda y.\mathsf{spill}''_{@}(p, y_{@})$.

In the highest local tree in Figure 8.5 the slash value of the head daughter gets bound. Following the analysis in Pollard and Sag 1994, this is done by introducing a TO-BIND SLASH specification at the S node. The NONLOCAL FEATURE PRINCIPLE, then, blocks the inherited slash value of the head daughter from appearing at the mother, and the FILLER-HEAD SCHEMA guarantees that an appropriate filler is realized as the nonhead daughter.

Having considered the structure of sentence (492), the question arises why it is that this sentence is ungrammatical: it contains the idiomatic uses of the words *spill* and *beans*, and the S′ node has a logical form which satisfies the distributional requirements of these two words as stated in (489): as we assume that topicalization does not change the logical form of a sentence, the CONTENT value of the overall sentence is identical to that of the normal sentence given in Figure 8.4.

There is, however, a difference between the two structures. In the topicalization structure in Figure 8.5, the logical form of the S node does not satisfy the distributional requirements of the word *spill*: the CONTENT value of this node is simply $\lambda y.\mathsf{spill}''_{@}(p, y)$. Thus, there is no occurrence of the non-logical constant $\mathsf{beans}''_{@}$ in the content of the S node.

This gives us a clear indication of which syntactic node must be considered to determine whether the logical form of this node is as required by the word *spill*: the lowest S node that contains this word. For the purpose of this thesis, we can define the lowest S node as the first sign that dominates the word under consideration and whose head is of sort *verb* and whose SUBCAT list is empty. We will use the term *minimal clause* of a word w for the phrase that satisfies this description. This definition is stated in a more precisely in (494).

(494) For two signs x and y, such that $x \neq y$, x is the *minimal clause* for y iff,
  (i) x dominates y, and
  (ii) x has a SUBCAT value of sort *empty-list* and a HEAD value of sort *verb*, and
  (iii) there is no sign z which is dominated by x and satisfies (i) and (ii).

In our example in Figure 8.5, the S node is the minimal clause of the word *spill*, the S′ node is the minimal clause of the noun *beans*. The S node does not count as the minimal clause for the word *beans*, because it does not dominate this word. The S′ node, on the other hand, is not the minimal clause for the word *spill*, because it dominates the S node and the S node already satisfies the first two criteria of the definition in (494).

We can now use the notion of *minimal clause* to account for the ungrammaticality of sentence (492). All that needs to be done is to assume that the distributional requirements of the word *spill* must be satisfied by the CONTENT value of the minimal clause of this word. Using the notion of minimal clause, we can re-state the distributional requirements for the idiomatic use of *spill* and *beans*. In (495) we bold face the change made with respect to the formulation in (489).[11]

(495) a. The word *spill* with the meaning spill'' can only occur in a sentence if, in the logical form of **the minimal clause of the word**, the constant spill'' occurs in the following constellation:

$$[\mathsf{the}\, y : \ldots \mathsf{beans}''_{@}(y_{@}) \ldots](\ldots \mathsf{spill}''_{@}(x_e, y_{@}) \ldots)$$

b. The word *beans* with the meaning $\mathsf{beans}''_{@}$ can only occur in a sentence if, in the logical form of **the minimal clause of the word**, the constant $\mathsf{beans}''_{@}$ occurs in the following constellation:

$$[\mathsf{the}\, y : \ldots \mathsf{beans}''_{@}(y_{@}) \ldots](\ldots \mathsf{spill}''_{@}(x_e, y_{@}) \ldots)$$

With this more precise formulation of the distributional requirements of the words *spill* and *beans*, we can account for the fact that the topicalization of the NP *the beans* is excluded. As we have seen above, the S node is the minimal clause for the verb *spill* in Figure 8.5. As the word *spill* occurs with the meaning spill'', its occurrence must satisfy the requirement in (495a). This requirement demands a logical form of the S node of the form $[\mathsf{the}\, y : \ldots \mathsf{beans}''_{@}(y_{@}) \ldots](\ldots \mathsf{spill}''_{@}(x_e, y_{@}) \ldots)$. The S node, however, has a CONTENT value of the form $\lambda y.\mathsf{spill}''_{@}(p, y_{@})$. Thus, the condition in (495a) is not met.

It is, thus the word *spill* whose distributional conditions are not satisfied. As far as the word *beans* is concerned in sentence (492), everything is fine: its minimal clause is the S′ node. The logical form of this node is $[\mathsf{the}\, y : \mathsf{beans}''_{@}(y_{@})](\mathsf{spill}''_{@}(x_e, y_{@}))$, just as required by the distributional restriction in (495b).

The distributional requirements in (495) also allow us to account for the other aspect of irregular behavior found with the IE *spill the beans*: it is impossible for the NP to take the form of a fronted relative pronoun. In (496), we repeat the relevant sentences.

(496) a. * The beans [that the alleged arms dealer spilled] made the party leader resign.
b. * The party leader resigned because of the beans [that the alleged arms dealer had spilled].

For our purpose, it is enough to consider the verb *spill* in the relative clauses in (496a) and (496b). In both cases, the direct object has been extracted, i.e., we have a situation parallel to that in Figure 8.5 where the semantic contribution of the direct object is still absent from the minimal clause of the verb *spill*.

[11] The restrictions given in (495) account for the data as presented in the main text in Section 6.2. For speakers that allow topicalization of the NP *the beans* in principle (Wasow et al., 1983), the syntactic domain within which the right logical form must be found can be larger. For speakers that share the judgments of Schenk 1995, i.e., speakers that do not allow internal modification in the case of this IE, the logical form requirement must be formulated in a more restrictive way. As indicated in (i), for these speakers, the logical form of the minimal clause may not contain any (internal) modification of the noun *beans*.

(i) $[\mathsf{the}\, y : \mathsf{beans}''_{@}(y_{@})](\ldots \mathsf{spill}''_{@}(x_e, y_{@}) \ldots)$

The final property of the IE that we have not yet addressed is the question of pronominalizability. In Chapter 6 we found the following contrast:

(497) a. I was worried that the beans might be spilled, but they weren't. (Wasow et al., 1983, p. 112)
b. * When Pat spilled the beans, she thought that they would shock her parents.

In the discussion of the data, we concluded that pronominal reference was only possible if the pronoun appears as part of the same idiomatic expression. Thus, pronominalization is possible in the case of (497a), because in the elliptic *but*-clause the idiomatic expression is indirectly present. This is not the case in the clause that contains *they* in sentence (497b). There, the pronoun must refer to some particular secret(s) that Pat had disclosed. To account for this data, again, we would need a theory of discourse entities which is not available in HPSG. But, roughly speaking, we can assume that the NP *the beans* does not introduce a discourse entity that can be referred to by a pronoun in another sentence. Thus, we can exclude the (b) sentence in (497).

Matters are different in the (a) sentence, however. The sentence is elliptic, i.e., the main verb *spill* is not overtly present in the *but*-clause. Unfortunately, we lack a theory of elliptic constructions in HPSG. Still, it is clear that there is a difference between the two sentences which is independent of the IE. For the time being we can do nothing but hope that the different ability of pronominalization of *the beans* can be reduced to this independently present distinction.[12]

To sum up the discussion: We propose to handle the IE *spill the beans* as a free combination which contains words with very strict occurrence restrictions. The kind of restrictions that we assume are different, though, from those used in Gazdar et al. 1985. We have replaced their restrictions on the interpretation function for semantic constants by a restriction on logical forms. In doing this, we reduce the number of semantic constants that must be stipulated. In addition, we have introduced a syntactic boundary node, the minimal clause, at which the logical form restriction must hold. This gave us the means to constrain the syntactic flexibility of the IE in the desired way, i.e., while we do allow for passive and raising, we successfully exclude topicalization and relative clause formation with this IE. Remember from our discussion of the GPSG approach, that topicalization, like passive, is assumed to have no effect on the interpretation of the sentence. This led to a serious problem because within the interpretive account of the distribution of the semantic constants there is no way to account for the impossibility of topicalization.

So far, the distributional requirements of the words *spill* and *beans* were merely stated in an informally. In the next subsection, we will provide a rigid RSRL formalization of these restrictions. To do this, we extend the functionality of the COLL that we have used in our account of internally irregular IEs.

8.2.2. Formalization of the Analysis. In the previous subsection, we have argued that we can account for the behavior of the internally regular IE *spill the beans* if we assume the existence of words *spill* and *beans* which differ from the normal words in two respects: First, they contribute a different non-logical constant (spill'' and beans''). Second these words may only occur in very restricted contexts.

[12]Compare (497b) to (i), the example that was problematic for Riehemann (2001) (see footnote 42):
(i) Eventually she spilled all the beans. But it took her a few days to spill them all.
In (i), the idiomatic *spill* needs to be licensed. Whatever accounts for the restricted pronominalizability of *the beans* in (497a) will make the special constant beans'' available in (i), too. Thus, the required logical form is present to allow for an occurrence of spill''. Such an analysis is not possible in Riehemann's account, because she relies on the presence of particular idiomatic words.

In the lexical entries in (486a), we encoded the special semantic contribution made by these words. In (495) we expressed the distributional requirements of the words informally. The HPSG theory developed in Pollard and Sag 1994 does not provide the necessary means to express conditions as those in (495).

The distributional restrictions given for the words *spill* and *beans* in their idiomatic use are idiosyncratic properties of the particular words, because there are no other words in English that have exactly these distributional restrictions. As idiosyncratic properties, they should be part of the information expressed in the lexical entries for these words. Thus, the lexical entry for the idiomatic word *spill* for example should contain both the information in (486a) and in (495a). In (498) we have compiled these two kinds of information.

(498) Lexical entry for the idiomatic use of *spill* (preliminary):

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{spill} \rangle \\ \text{S L} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{SUBCAT} & \left\langle \text{NP, NP, (PP}[\textit{to}]) \right\rangle \end{bmatrix} \\ \text{CONT} & \lambda y \lambda x.\mathsf{spill}''_{@}(x_{@}, y_{@}) \end{bmatrix} \end{bmatrix}$$

and

this word can only occur in a sentence if, in the logical form of the minimal clause of the word, the constant spill'' occurs in the following constellation:

$$[\mathsf{the}\ y : \ldots \mathsf{beans}''_{@}(y_{@}) \ldots](\ldots \mathsf{spill}''_{@}(x_e, y_{@}) \ldots)$$

In (498) we have collected all the idiosyncratic information needed for the idiomatic word *spill*. This information consists of two parts: First, internal properties of the word *spill*, i.e., its phonology, its semantics, its syntactic category and its subcategorization requirements. These bits of information are those given already in (486a). The second kind of information is information about the linguistic context in which the word may occur, i.e., it is information which is external to the word itself. This information was given in (495a) and expresses the distributional requirements of the word.

If we try to formalize the distributional properties of the word *spill*, we need to make the linguistic context in which the word occurs an internal property of the word itself. While this sounds paradoxical, it is a consequence of the way HPSG works: all that can be constrained is the interaction of the components of a given object. We have already discussed this property in Chapter 2. It follows from the fact that RSRL formulae are interpreted with respect to a given object. It is enough to consider this object and its components to determine whether a formula describes this object or not. Thus, an RSRL formula can only refer to properties of the object itself.

As a consequence, we must make the linguistic context of the word *spill* locally available at the word level in order to be able to impose occurrence constraints on this word. So far, there is no attribute defined on the sort *word* that we could use for this purpose. In the preceding section, we have introduced the attribute COLL on the sort phrase. Now that we need a special attribute for words, we can simply define the attribute for all subsorts of the sort *sign*. For phrases, it seemed to be enough to assume boolean values for this attribute, as we only needed the attribute to differentiate between phrases which are built in a regular way and phrases which are exempt from the principles of regular combination of signs. For words, it is not sufficient to make a difference between idiomatic and non-idiomatic words. Here, in the case of an idiomatic word, we need to have the possibility of having a sign occur as the COLL value. In the case of the idiomatic use of the word *spill* this sign is the minimal clause which contains the word.

In the table in (499) we have given the COLL values as assumed so far. The first column consists of example expressions. The second column indicates the subsort of *sign* of the examples. In the third column, we specify the sort of the COLL value. As examples we use three VPs and their three verbal heads. The VPs are: the regular VP *read a book*, the internally irregular VP *kick the bucket* and the internally regular IE *spill the beans*.

(499)

example	sort	COLL value
read a book	*phrase*	*minus*
kick the bucket	*phrase*	*plus*
spill the beans	*phrase*	*minus*
read	*word*	??
kick	*word*	??
spill	*word*	*sign*

Following the specification in the last section, we assume that the internally irregular IE *kick the bucket* has a COLL value of sort *plus*, and the freely combined VP *read a book* has the specification [COLL −]. As we have indicated in the tree for the IE *spill the beans* in Figure 8.4, the VP node is also assumed to be a regular combination, i.e., to have a [COLL −] specification.

For the verbal heads of these VPs we make the following assumptions: In the case of the idiomatic use of the word *spill*, the COLL value must specify the minimal clause which dominates the word. For the words *read* and *kick*, we put question marks in the table. We will address the question of their COLL values in Section 8.3 below. For the time being, we will assume the following sort hierarchy and appropriateness conditions for the sort *sign*.

(500) Sort hierarchy and appropriateness conditions below the sort *sign* (preliminary)

sign
PHON *list*
SYNSEM *synsem*

word
STORE *list*
COLL *sign-or-minus*

phrase
DTRS *const-struc*
COLL *boolean*

In this new sort hierarchy, we have declared the attribute COLL appropriate for the sort *word*. On the sort *word*, the COLL feature does, however, not take boolean values but it takes either a sign as its value or *minus*.[13] We assume that words that are licensed by some lexical entry, i.e., by some disjunct in the WORD PRINCIPLE have a sign as their COLL value. Words that are licensed by some derivational rule, on the other hand, have the COLL value *minus*. This distribution is ensured by the following principle.

(501) The STORE-COLL PRINCIPLE:

$$\begin{bmatrix} \textit{word} \\ \text{STORE} \;\; \textit{elist} \end{bmatrix} \Leftrightarrow \begin{bmatrix} \textit{word} \\ \text{COLL} \;\; \textit{sign} \end{bmatrix}$$

The principle in (501) has the effect that exactly those words have sign-valued COLL specification that have an empty STORE value, i.e., those words that are not the output of a derivational rule. For words with a non-empty STORE value, we require a COLL value of sort *minus*. The fact that we assume the value [COLL −] for derived words indicates that we treat derivational rules parallel to internally regular phrases. Put differently, we conceptualize

[13]In RSRL the appropriateness function is defined in such a way that we must assume a sort *sign-or-minus* which is a supersort of the sorts *sign* and *minus*. It would not be possible to allow values of these two sorts without introducing a common supersort for them.

derivational rules as kinds of unary branching trees. This point of view is justified, because a derivational rule allows us to predict all the properties of the output word from the properties of the input word. In that respect, the output word of a derivational rule is parallel to the regular combination of signs in syntactic structures. For the time being, we will concentrate on the COLL specification for non-derived words. But, as we treat passive and topicalization via derivational rules, we will come back to the issue of derived words later in this subsection.

For non-derived words, the STORE-COLL PRINCIPLE enforces a sign-valued COLL specification. In the case of the words that constitute internally regular IEs this sign will be some larger bit of structure which satisfies the distributional requirements of this word. For words which do not have any particular distributional restrictions, we simply allow an arbitrary sign to occur in the COLL value.[14] It is obvious that while we use the same attribute on phrases and words, the COLL feature seems to fulfill completely different tasks in both cases. In Section 8.3, we will show that there is an intimate relation between the two uses.

With the modified sort hierarchy below *sign*, we can give an account of the distributional restrictions of the idiomatic use of the word *spill*. For this purpose, we consider the COLL value of this word. Under the assumption of a sign-valued COLL feature, we can now incorporate the distributional requirements into the lexical entry for the word *spill* formally.

(502) Lexical entry for the idiomatic use of *spill*:

$$\boxed{1}\begin{bmatrix} \textit{word} \\ \text{PHON} \ \langle \textit{spill} \rangle \\ \text{S L} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{SUBCAT} & \left\langle \text{NP, NP, (PP}[\textit{to}]) \right\rangle \end{bmatrix} \\ \text{CONT} \ \lambda y \lambda x.\mathsf{spill}''_{@}(x_{@}, y_{@}) \end{bmatrix} \\ \text{COLL} \ \boxed{2} \begin{bmatrix} \textit{sign} \\ \text{S L CONT} \ \ldots [\mathsf{the}\ v : \ldots \mathsf{beans}''_{@}(v_{@}) \ldots](\ldots \mathsf{spill}''_{@}(w_e, v_{@}) \ldots) \ldots \end{bmatrix} \end{bmatrix}$$
$$\textbf{and} \ \texttt{minimal-clause}(\boxed{2},\boxed{1})$$

This new lexical entry specifies that there is a COLL value $\boxed{2}$. This value is a sign which has the CONTENT value which contains the constants spill'' and beans'' in the constellation specified above in the distributional requirements for the word *spill*. In addition, the relational call at the bottom of the lexical entry specifies that the sign in the COLL value is the minimal clause of the word *spill*.

To express this condition, we make use of a binary relation `minimal-clause` which holds between two signs iff the first sign is the minimal clause of the second sign as defined above in (494). For the sake of clarity, we state which line of the formal definition in (503) corresponds to which line in the informal characterization given above.

(503) The relation `minimal-clause`:

$$\texttt{minimal-clause}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow} \begin{array}{l} \text{(i) } \texttt{dominate}(\boxed{1},\boxed{2}) \\ \text{(ii) } \textbf{and} \ \ \boxed{1}\begin{bmatrix} \text{SYNS LOC CAT} \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{SUBCAT} & \textit{elist} \end{bmatrix} \end{bmatrix} \\ \text{(iii) } \textbf{and not E}\boxed{3} \left(\begin{array}{l} \texttt{dominate}(\boxed{1},\boxed{3}) \\ \textbf{and not} \ \boxed{1} \approx \boxed{3} \\ \textbf{and} \ \texttt{dominate}(\boxed{3},\boxed{2}) \\ \textbf{and} \ \boxed{3}\begin{bmatrix} \text{S L C} \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{SUBCAT} & \textit{elist} \end{bmatrix} \end{bmatrix} \end{array} \right) \end{array}$$

[14]In (505) we present a principle which imposes some constraints on the possible COLL values. As an effect of this principle, it is guaranteed that even if the COLL value is not restricted in the lexical entry of a word, it can only contain a sign which is part of the same syntactic structure that this word is contained in.

As specified above, two signs x and y stand in the relation `minimal-clause` iff x dominates y and is a fully saturated verbal projection. In addition, x is required to be the smallest sign that satisfies these two conditions, i.e., x does not dominate a sign z which satisfies the first two conditions.

Parallel to the lexical entry for the idiomatic use of the word *spill* in (502), we can now state the lexical entry for the noun *beans* which contains all the information given in (486b), together with the COLL specification that occurs in the lexical entry of the verb *spill*. The entire lexical entry for the noun is given in (504).

(504) Lexical entry for the idiomatic use of *beans*:

$$\boxed{1}\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{beans} \rangle \\ \text{S L} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{SUBCAT} & \left\langle \text{Det} \right\rangle \end{bmatrix} \\ \text{CONT} & \lambda x.\mathsf{beans}''_{@}(x_{@}) \end{bmatrix} \\ \text{COLL} & \boxed{2}\begin{bmatrix} \textit{sign} \\ \text{S L CONT} \ \ldots[\mathsf{the}\ v : \ldots \mathsf{beans}''_{@}(v_{@})\ldots](\ldots \mathsf{spill}''_{@}(w_e, v_{@})\ldots)\ldots \end{bmatrix} \end{bmatrix}$$

and `minimal-clause(`$\boxed{2}$,$\boxed{1}$`)`

Given the lexical entries in (502) and (504), we can consider the structure of a sentence which contains the IE. In Figure 8.6 (page 347), we have added the COLL value of the words *spill* and *beans* to the tree from Figure 8.4. As indicated, the overall sentence is the minimal clause for both words, i.e., their COLL value. The CONTENT value of this sign satisfies the condition imposed on the COLL value in the lexical entries of the two words.

The COLL specification of the word *spill* guarantees that there is some minimal clause for the word which has a logical form of the right shape. What remains to be done, however, is to ensure that the sign in the COLL value is the actual structure in which the word occurs. This means that so far, nothing in the grammar ensures that the COLL value of the words *spill* and *beans* is actually identical to some node in the tree in Figure 8.6.

To solve this problem, we assume the COLL PRINCIPLE given in (505). This principle enforces that for each sign, and for each word that is dominated by this sign, the COLL value of this word either dominates the sign or is dominated by it.

(505) The COLL PRINCIPLE:

$$\textit{sign} \Rightarrow \mathbf{A}\boxed{1}\ \mathbf{A}\boxed{2}\left(\left(\begin{array}{l}\texttt{dominate}(:,\boxed{1})\\ \textbf{and}\ \ \boxed{1}\begin{bmatrix}\textit{word}\\ \text{COLL}\ \boxed{2}\ \textit{sign}\end{bmatrix}\end{array}\right) \Rightarrow \left(\begin{array}{l}\texttt{dominate}(:,\boxed{2})\\ \textbf{or}\ \texttt{dominate}(\boxed{2},:)\end{array}\right)\right)$$

In order to check that the COLL PRINCIPLE is satisfied, let us first consider the words *spill* and *beans*. As we use a reflexive definition of the dominance relation, the word *spill* dominates itself. Its COLL value is the overall sentence. This sentence does contain the word *spill*. Thus the COLL PRINCIPLE is satisfied by this word. Similarly, the principle is true of the word *beans*. In a slightly less trivial way, the principle also holds for the phrases that occur in the sentence in Figure 8.6. The NP *the beans* dominates the word *beans*. This word has the sentence $\boxed{1}$ as its COLL value. Clearly, this sentence dominates the NP. Similarly, in the case of the VP, there are the words *spill* and *beans* that are dominated by this phrase. Both have the sentence $\boxed{1}$ as their COLL value. Again, this sentence dominates the VP, thus, the COLL PRINCIPLE is satisfied. Finally, the S node reflexively dominates itself.

If we embed the sentence in Figure 8.6 further, as indicated in sentence (506), every node of the matrix clause that dominates the words *spill* and *beans* also dominates their minimal clause. Again, the requirements expressed in the COLL PRINCIPLE are met.

(506) Mary thought [Pat spilled the beans.]

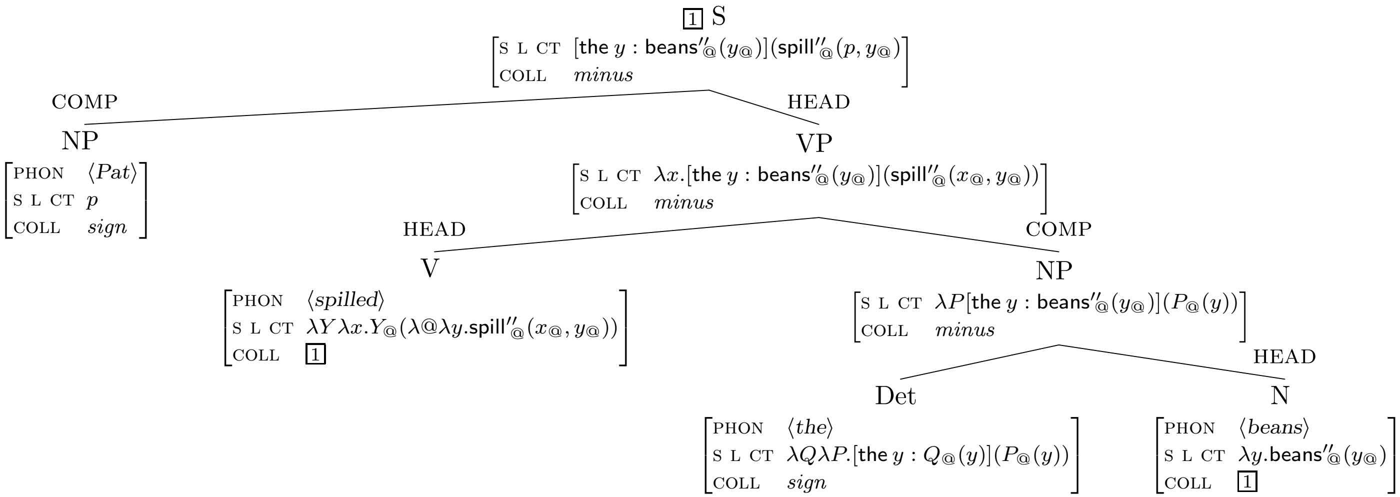


FIGURE 8.6. The structure of the sentence *Pat spilled the beans*.:

So far, we have ignored the COLL values of the other words in the sentence, i.e., the COLL specification of the words *Pat* and *the*. As indicated above, as these words do not seem to have distributional restrictions, we do not assume that these words impose any restrictions on their COLL values. Thanks to the COLL PRINCIPLE, however, the COLL values of these words are not arbitrary. Instead, the COLL values must be signs which dominate these words as they occur in the sentence in Figure 8.6, i.e., the words themselves, the direct object NP or the VP (in the case of *the*), or the overall S node, or any sign which dominates the S node of the example sentence (such as the matrix clause in example (506)).

This simple example has shown that we can account for the distributional restrictions of the idiomatic uses of the words *spill* and *beans* if we introduce the COLL feature on words and postulate the COLL PRINCIPLE.

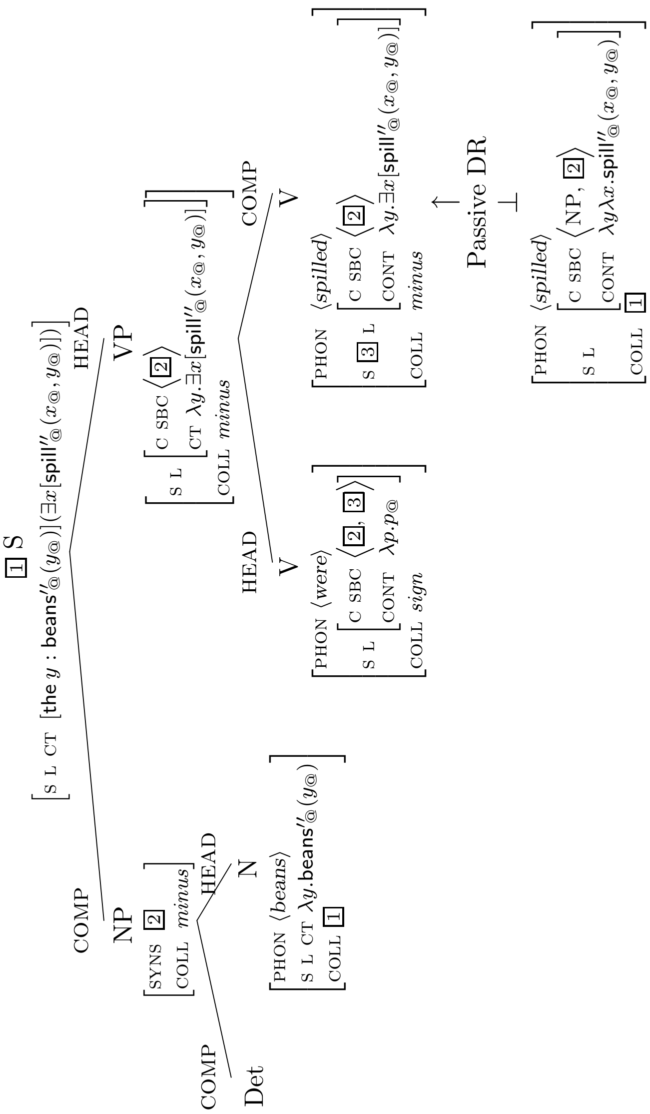
It should be noted at this point that the COLL feature is notably different from valence features such as SUBCAT or non-local features such as SLASH.

One difference is that the COLL value of a word is a sign, whereas we find *synsem* objects on the SUBCAT list and *local* objects in the SLASH values (see Section 2.3). We must require an entire sign as the COLL value, because we want to express (i) that the element in the COLL value of a word *dominates* this word and (ii) that it stands in a certain tree-configurational relation to this word: In the case of the idiomatic use of *spill*, the COLL value is the smallest clause that dominates the word. In the following subsection, we will show that the syntactic domain within which the occurrence restrictions of the word must be satisfied differs from IE to IE. Both requirements can only be expressed inside the lexical entry of the idiomatic word *spill* if we assume a sign-valued COLL attribute.

Another difference lies in the fact that the COLL value does not "percolate". We have seen in the presentation of the syntactic fragment in Section 2.3 that in a phrase an element from the SUBCAT list of the head daughter re-occurs on the SUBCAT list of the phrase if it is not realized as the nonhead daughter. Similarly, an element in the INHERITED SLASH set of a daughter re-occurs in the INHERITED SLASH set of the phrase unless it is retrieved. The COLL specification of a word, on the other hand does not appear in the COLL value of any phrase that dominates it. The motivation for this is twofold. First, as we saw in the analysis of internally irregular IEs in Section 8.1, we also use the COLL specification to distinguish phrases that obey the regular principles of syntax and semantics from those that may violate them. If the COLL value of a word could percolate, we would need another attribute to indicate the regularity status of a phrase. Second, even if we percolated the COLL value of a word we would still be forced to express how far up in the tree this percolation may go. Thus, we would still have to assume a sign-valued COLL attribute, and we would still need the relation `minimal-clause` in the lexical entries of the idiomatic word *spill*. This indicates that percolating the COLL value would not lead to any simplification in the analysis and is, therefore, better avoided.

We think that these two differences between the COLL value on one side and valence or slash values on the other side is fully justified because the COLL value is designed for a very special class of phenomena: for irregularities that cannot be captured within the regular apparatus of the grammar. Since we assume a sign-valued attribute, we have chances to capture even the most exotic context restrictions; and since the COLL specification of a word does not percolate, we keep our analysis of irregularities separate from the rest of the grammar. As we will see in Section 8.3, a non-trivial COLL value will, ultimately, be confined to lexical elements, and, thus, it will not interact directly with those principles of the grammar that express the regular combination of complex signs.

FIGURE 8.7. The structure of the sentence *The beans were spilled.*:



The example sentence that we have used to illustrate our analysis was admittedly simple. Turning next to passive and topicalization, we will show that our theory captures the distribution of the IE in these constructions as well.

In Figure 8.6, there was no instance of an application of a derivational rule. In Figure 8.7, we show the structure of the passive sentence (507) that contains the IE *spill the beans*.

(507) The beans were spilled.

As shown in Chapter 4, we follow the analysis of passive in Pollard and Sag 1994 and assume a derivational rule that relates a non-passive verb form to a passive participle with altered valence requirements. For ease of reference, we state the Passive Derivational Rule in (508). Notice that we have added the COLL specification to the output of the DR.

(508) Informal specification of the Passive DR, as given in (269):

$$\begin{bmatrix} \textit{word} \\ \text{SYNS LOC} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD} \begin{bmatrix} \textit{verb} \\ \text{VFORM} \ \textit{psp} \end{bmatrix} \\ \text{SBC} \ \left\langle \text{NP}|\boxed{2} \right\rangle \oplus \left\langle \boxed{1} \right\rangle \end{bmatrix} \\ \text{CONT} \ \lambda y \lambda z_n \ldots \lambda z_1 \lambda x \lambda u_m \ldots \lambda u_1 . \phi \end{bmatrix} \end{bmatrix} \mapsto$$

$$\begin{bmatrix} \textit{word} \\ \text{SYNS LOC} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD} \begin{bmatrix} \textit{verb} \\ \text{VFORM} \ \textit{pas} \end{bmatrix} \\ \text{SBC} \ \left\langle \boxed{1}|\boxed{2} \right\rangle \end{bmatrix} \\ \text{CONT} \ \lambda z_n \ldots \lambda z_1 \lambda y \lambda u_m \ldots \lambda u_1 . \exists x \phi \end{bmatrix} \\ \text{COLL} \ \ \textit{minus} \end{bmatrix}$$

(where the list $\boxed{2}$ has exactly n elements)

We can observe the effect of this derivational rule in Figure 8.7: The transitive verb *spill* is related to its passive form. This passive verb has just one element left on its subcat list, the synsem object referred to by the tag $\boxed{2}$. Also, semantically the passive verb is of a lower valence than its active counterpart, because the lambda operator λx in the input is replaced by an existential quantifier $\exists x$ in the output of the DR. Finally, the COLL value of the output word is *minus*, as enforced by the STORE-COLL PRINCIPLE in (501).

We treat the passive auxiliary as a raising verb, following Pollard and Sag 1994, i.e., the first element on its subcat list is identical to the remaining element on the subcat list of the passivized verb. The content of the passive auxiliary is just the identity function. In addition, the content of the finite VP is identical to that of the passive verb. The NP *the beans* is combined just as shown in Figure 8.6. This NP is realized as the syntactic subject in the passive. As indicated by the tag $\boxed{2}$, the SYNSEM value of this NP appears as the first element on the subcat lists of the passive auxiliary and of the passivized verb *spilled*, but also as the second argument on the subcat list of the non-passive form of the verb.

If we are to check now whether the distributional requirements of the words in Figure 8.7 are met, this can be done for the noun *beans* just as we had seen in the case of the active sentence: it is realized as the syntactic subject and its minimal clause is the S node. The logical form of this node is as required in the lexical entry of the noun.

Since the IE *spill the beans* contains two elements with strict distributional requirements, it is not enough to check that the requirements of the word *beans* are met. We must also verify those of the word *spill*. In the structure in Figure 8.7 there are two words that have the phonology *spilled*: the input and the output of the Passive DR. The output word has a COLL value of sort *minus*, thus, it does not express distributional restriction. The input, however, is the word *spill* as licensed by the lexical entry in (502). It has a sign-valued COLL specification. Under the usual understanding of the dominance relation in syntactic trees, the input of a derivational rule is not really part of the syntactic structure, and, therefore, not dominated by any node in this structure. As we have pointed out before, we take a slightly different position on derivational rules. We see them as unary branching trees at word level. From this perspective, it is very natural to extend the dominance relation so that the output word of a derivational rule is said to dominate the input word.

With this small extension of the definition of the dominance relation, we can account for the use of parts of internally regular IEs in syntactic structures that involve derivational rules. To illustrate this, reconsider the tree in Figure 8.7. We assume that in addition to the dominance relation among the phrases and their daughters in the structure, the output of the Passive DR (and every node that dominates the output word) also dominates the input word. As indicated in the figure, the COLL value of the input verb is the S node (tag

$\boxed{1}$). This node is the minimal clause of the input word: it is a saturated verbal projection and dominates the input word under the extended notion of domination. The CONTENT value of the S node is exactly as required in the lexical entry of the verb *spill* in (502).

We can also show that the COLL PRINCIPLE is met under the extended dominance relation: since we assume a reflexive dominance relation, the input word of the DR dominates itself. Its COLL value is the S node. As shown above, the S node dominates the input word. Thus, the input of the DR satisfies the COLL PRINCIPLE. Similarly, all nodes that dominate the input word also meet the requirements of this principle with respect to this word.

We have seen that the simple extension of the dominance relation allows us to account for the distribution of IEs in constructions involving derivational rules. In (509) we state the full definition of the relation `dominate` as we assume it to occur for example in the COLL PRINCIPLE and in the definition of the relation `minimal-clause` in (503).

(509) Revised definition of the relation `dominate`:

$$\texttt{dominate}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow} \boxed{1} \approx \boxed{2}$$

$$\texttt{dominate}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow} \mathbf{E}\boxed{3} \begin{pmatrix} \boxed{1}\begin{bmatrix}\text{DTRS H-DTR } \boxed{3}\end{bmatrix} \\ \textbf{and } \texttt{dominate}(\boxed{3},\boxed{2}) \end{pmatrix}$$

$$\texttt{dominate}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow} \mathbf{E}\boxed{3} \begin{pmatrix} \boxed{1}\begin{bmatrix}\text{DTRS N-DTR } \boxed{3}\end{bmatrix} \\ \textbf{and } \texttt{dominate}(\boxed{3},\boxed{2}) \end{pmatrix}$$

$$\texttt{dominate}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow} \mathbf{E}\boxed{3} \begin{pmatrix} \boxed{1}\begin{bmatrix}\text{STORE } \left\langle \begin{bmatrix} \textit{der-rule} \\ \text{IN } \boxed{3} \end{bmatrix} \right\rangle \end{bmatrix} \\ \textbf{and } \texttt{dominate}(\boxed{3},\boxed{2}) \end{pmatrix}$$

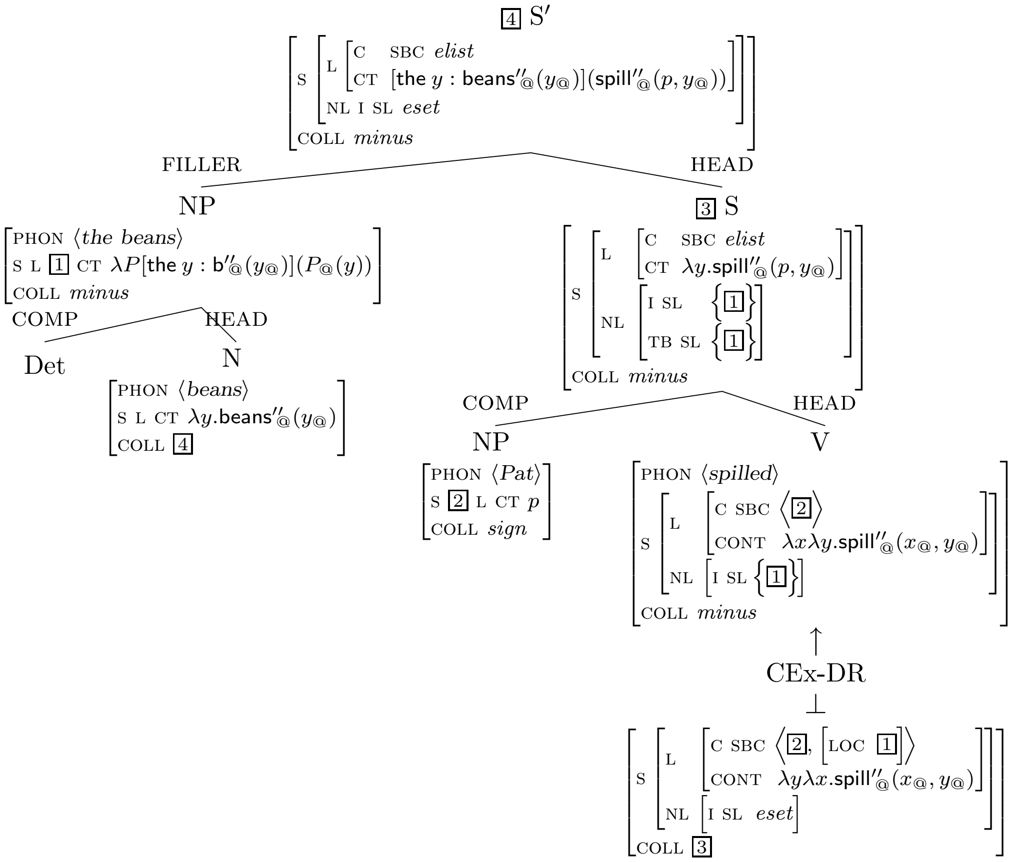
Two signs x and y stand in the relation `dominate` iff either the two signs are identical (first clause), or there is a sign z which is one of the daughters of the sign x and stands in the relation `dominate` with the sign y (second and third clause). So far, the definition of the relation `dominate` expresses just domination in syntactic structures.

The difference between the extended definition in (509) and a more traditional one lies in the fourth clause. According to this clause, x and y also stand in the relation `dominate` if there is a word $\boxed{1}$ which is the input word of a derivational rule which has the sign x as its output and this sign $\boxed{1}$ stands in the relation `dominate` with y.

Using this definition of the relation `dominate`, we can show the S node "dominates" the input word of the derivational rule in Figure 8.7: The input word of the DR dominates itself by the first clause of the definition. The input is the IN value of the STORE element of the passive verb. Thus, this verb dominates the input by the fourth clause. This passive verb is the nonhead daughter of the VP. As a consequence of the third clause, the VP dominates the input word. Finally, the VP is the head daughter of the S node. By the second clause of the definition in (509), the S node and the input verb stand in the relation `dominate`.

We have shown that with the dominance relation as defined in (509), the lexical entries and the COLL PRINCIPLE given above allow for the syntactic freedom attested for the IE *spill the beans*. For the sake of completeness, we will show that the behavior of this IE with respect to topicalization is also accounted for. In Figure 8.5 on page 339 we gave the syntactic structure of a sentence which contains the IE *spill the beans* but where the direct object has been moved outside the minimal clause of the verb by topicalization. In Figure 8.8 we repeat this structure, enriched by the COLL values. As indicated above, the sentence is ungrammatical because the distributional requirements of the verb are not met within its minimal clause, which is the S node with tag $\boxed{3}$ in Figure 8.8. Notice that, like with passive, we have to consider the input word of the derivational rule, not the output.

FIGURE 8.8. The structure of the sentence **The beans Pat spilled.*:



With this discussion of the most prominent syntactic configurations we have shown that via a simple extension of the feature geometry, i.e., allowing a sign-valued attribute COLL for non-derived words, and a simple principle, the COLL PRINCIPLE, we provided a system that enables us to formulate distributional restrictions for words as parts of their lexical entries. In the present subsection, we have concentrated on a single IE, *spill the beans*. In the following subsection, we will show how this mechanism can be used to account for the other internally regular IEs.

8.2.3. The Other IEs. In Section 8.2.2 we have used the IE *spill the beans* to illustrate and motivate our extended use of the attribute COLL. In this subsection, we will present an analysis of the remaining IEs within the framework developed above. The discussion of these other IEs will reveal the flexibility of the present approach. We will turn to the IEs *pull strings*, *make headway* and *make a decision*.

pull strings. In the preceding chapters, we have pointed out that this IE is very similar to the IE *spill the beans* but shows even more signs of regularity. In (510), we give the lexical entries for the words *pull* and *strings* as they occur in the IE *pull strings*.

(510) a. The lexical entry for the idiomatic use of *pull*:

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{pull} \rangle \\ \text{S L} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{SBC} & \left\langle \text{NP, NP} \right\rangle \end{bmatrix} \\ \text{CONT} & \lambda y \lambda x.\mathsf{pull}''_{@}(x_{@}, y_{@}) \end{bmatrix} \\ \text{COLL} & \begin{bmatrix} \text{S L CONT} & \ldots \mathsf{strings}''_{@}(w_{@}) \ldots \mathsf{pull}''_{@}(v_e, w_{@}) \ldots \end{bmatrix} \end{bmatrix}$$

b. The lexical entry for the idiomatic use of *strings*:

$$\boxed{1}\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{strings} \rangle \\ \text{S L} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{SBC} & \left\langle \text{Det} \right\rangle \end{bmatrix} \\ \text{CONT} & \lambda y.\mathsf{strings}''_{@}(y_{@}) \end{bmatrix} \\ \text{COLL} & \boxed{2}\begin{bmatrix} \text{S L CONT} & \ldots \mathsf{strings}''_{@}(w_{@}) \ldots \mathsf{pull}''_{@}(v_e, w_{@}) \ldots \end{bmatrix} \end{bmatrix}$$

and `minimal-clause`($\boxed{2}$,$\boxed{1}$)

The overall shape of the lexical entries for the words *pull* and *strings* strongly resembles those of the words *spill* and *beans* above: The words have a content which introduces a new constant, pull″ in the case of the verb and strings″ for the noun. In addition, the COLL value is a sign whose logical form must be of a certain kind. In the lexical entry for the noun *strings* we also require that the COLL value stand in some specific syntactic relation to the word. The sign in the COLL value is referred to with the tag $\boxed{2}$ in the lexical entry of the noun. In the case of *strings*, the sign $\boxed{2}$ must be the minimal clause of the word licensed by the lexical entry, just as it was above for *beans*.

For the verb *pull*, we specify the logical form of the sign in the COLL value, but we do not restrict the syntactic relation between this sign and the word *pull*. The COLL PRINCIPLE ensures that the sign in the COLL value dominates the word. Thus, all that is required in the case of *pull* is that the COLL value is a sign that dominates the word and has a certain content specification. This can be any sign, in particular the overall utterance.

Parallel to the case of the components of the IE *spill the beans*, the logical form of the sign in the COLL value is further specified. For both words, we require the logical form of the element in the COLL value to contain an occurrence of the semantic constants pull″ and strings″ such that the second argument of pull″ is identical to the argument of strings″. In contrast to the case of *spill the beans*, however, the logical form required for the words *pull* and *strings* is less constrained. In particular, no special quantifier is given, nor is it required that the semantic constants be in the restriction and the scope of some quantifier.[15]

With the lexical entries in (510) we can go through the syntactic and semantic properties of the IE *pull strings* as collected in Chapter 6.

According to the first semantic criterion, the components of the IE do not occur with their idiomatic meaning outside the IE. This property is directly accounted for by the fact that the COLL values of the words *pull* and *strings* impose the restriction that whenever one of these words occurs in a sentence, the other must be present as well.

As noted in the discussion of the second semantic criterion, we can, however, assign each element in the IE some meaning such that the meaning of the IE as a whole can be calculated by combining the semantic contributions of its parts in a regular way. Again, this property is accounted for if we assume that the IE is internally regular, i.e., that the

[15]In fact, the requirements on the logical form of the COLL value is very similar to those specified in the WORDS set in the analysis of Riehemann 1997. This is the case because we refer to subterms, where, in MRS, reference is made to an object of sort *rel* in the LISZT list.

IE is not listed directly in the WORD PRINCIPLE or the IIP, but only comes into existence by the free combination of the words that compose it.

According to the third semantic criterion, internal modification of the noun *strings* is possible. In (511) we repeat the relevant example from Chapter 6 with its logical form.

(511) Pat pulled official strings (to help Chris).
$\exists y[\mathsf{strings}''_{@}(y_{@}) \wedge \mathsf{official}'_{@}(y_{@}) \wedge \mathsf{pull}''_{@}(p, y_{@})]$

Clearly, the logical form of the sentence (511) meets the distributional requirements expressed in the lexical entries in (510).

According to the fourth semantic criterion, the IE allows for pronominal reference. This indicates that the noun *strings* must be able to introduce a discourse referent.

Let us next consider the behavior of this IE with respect to the syntactic criteria. According to the first criterion, all words that occur in the IE also occur independently within the language. The way we have formulated the lexical entries of the idiomatic uses of the words *pull* and *strings*, there is no relation between the idiomatic use of the words and their non-idiomatic uses, i.e., we must assume distinct lexical entries for the idiomatic use of *pull* and for the non-idiomatic use of this verb. This clearly misses a generalization as in most cases the components of an internally regular IE occur outside the IE, albeit with a different meaning. Among the formal accounts of IEs that we considered in Chapter 7, only Riehemann 1997 made an attempt to capture this fact. As our analysis of internally regular IEs is very much like that of Gazdar et al. 1985, we also inherit the problem that there is no way we could relate the idiomatic use of a word such to its non-idiomatic uses.

The second and third syntactic properties are accounted for in a straightforwardly. Since we only specify the words that are contained in the IE, these words must combine in a regular way to form the IE as a whole. This also accounts for the syntactic flexibility to allow other material such as modifiers to occur inside the NP.

The same is true for passivization and raising. As we have seen in the discussion of the IE *spill the beans*, these two phenomena do not introduce a minimal clause boundary between the verb and its direct object, i.e., for both words the minimal clause is the same node. As such, the minimal clause of the noun *strings* also dominates the word *pull*, and, therefore, the logical form of the minimal clause is already of the required shape.

In the case of topicalization such as in sentence (512), the minimal clause of the noun is different from that of the verb. It should be remembered from the discussion of the topicalization data of the IE *spill the beans* that the minimal clause of the topicalized constituent is the lowest node that dominates the filler. In the case of example (512) this is the entire sentence. The logical form of this sentence contains both constants pull'' and $\mathsf{strings}''$ in the right constellation. Thus, the requirements of the word *strings* are met.

(512) Those strings, he wouldn't pull for you. (Nunberg et al., 1994, p. 13)

Let us, next, turn to the requirements of the verb *pull*. In the case of the IE *spill the beans*, topicalization was excluded. We derived the ungrammaticality of topicalization structures by making the distributional restrictions of the verb *spill* a requirement of the minimal clause of the verb. For the idiomatic use of *pull* it is sufficient if the distributional requirements are satisfied by any sign in the structure, not necessarily the minimal clause of the verb. As the overall utterance in (512) contains the necessary semantic material, we can assume that the COLL value of the verb is the overall utterance.[16]

[16]Note that in sentence (512) the COLL values of the words *pull* and *strings* are identical, as the minimal clause of the noun co-indices with the utterance that contains the verb. This need not be the case in general:
(i) Mary thinks that [those strings he wouldn't pull for you].
In the sentence in (i), the minimal clause for the noun is the bracketed constituent, whereas the overall

The last syntactic criterion involves the possibility of the IE to occur in relative clauses. The relevant examples are repeated in (513).

(513) a. Pat pulled the strings [that got Chris the job].
b. The strings [that Pat pulled] got Chris the job.

In both cases the minimal clause of the noun *strings* is the entire sentence. The verb occurs in the matrix clause in (513a) and in the relative clause in (513b). Since the relevant syntactic domain for the occurrence restriction of the verb is not delimited, this difference does not play a role. Thus, for both words and in both sentences in (513), the overall utterances are the relevant syntactic domains. In the logical forms of these sentences, the two constants pull'' and $\mathsf{strings}''$ occur in the right constellation. Therefore, the distributional requirements of the two words are met.

The presentation above has shown that the lexical entries in (510) account adequately for the behavior of the IE *pull strings*. We also captured the differences between the IEs *spill the beans* and *pull strings*. In that respect our approach is more flexible than the GPSG account: the GPSG account did not have any syntactic aspects, but assumed that topicalization is semantically without influence. Therefore, it was unable to account for the fact that not all IEs which allow passive also allow topicalization. Furthermore, our approach is also more elegant than the TAG analysis, because we do not have to state all the syntactic constructions in which the IE can occur, but we can influence the range of syntactic constructions by the requirements on the linguistic context. Finally, in contrast to Riehemann 1997, the overgeneralization problem does not arise in our account.

make headway. The next IE we will address is the expression *make headway*. As pointed out above, this IE is an instance of a support verb construction. The IE shows regular behavior with respect to most of the criteria in Chapter 6, with two exceptions. First, the word *headway* does not occur in English outside this expression. Second, the verb *make* in its support verb meaning is restricted to an (arbitrary) group of nouns, which contain, *inter alia*, the nouns *headway* and *decision*. We do not intend to make any commitments with respect to the right semantic analysis of support verb constructions. For the sake of concreteness, though, we make the simplified assumption that the support verb *make* contributes a semantic constant make'' which takes two individuals as its semantic arguments.

In (514) we depict the lexical entry of the noun *headway*. It is very similar to the entries of the nouns *beans* and *strings*, shown in (504) and (510b). It introduces a semantic constant $\mathsf{headway}'$ which will be interpreted as roughly synonymous to the constant contributed by the noun *progress*. Furthermore, the COLL value is a sign which is the minimal clause of the noun and requires the presence of a support verb construction in its logical form.

(514) The lexical entry for the word *headway*:

$$\mathbf{E}\boxed{1}\ \mathbf{E}\boxed{2}$$
$$\boxed{1}\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{headway} \rangle \\ \text{S L} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{SBC} & \textit{elist} \end{bmatrix} \\ \text{CONT} & \lambda x.\mathsf{headway}'_{@}(x_{@}) \end{bmatrix} \\ \text{COLL} & \boxed{2}\begin{bmatrix} \text{S L CONT} & \ldots \mathsf{headway}'_{@}(w_{@}) \ldots \mathsf{make}''_{@}(v_e, w_{@}) \ldots \end{bmatrix} \end{bmatrix}$$
$$\textbf{and}\ \ \texttt{minimal-clause}(\boxed{2},\boxed{1})$$

utterance is the entire sentence. The distributional requirements are, nonetheless, met. The COLL value of the verb could be any node that dominates the clause in brackets.

While the lexical entry for *headway* is similar to the one for *strings*, the lexical entry for the support verb *make* is slightly different from those that we have seen so far. In the previous cases, we could always characterize the linguistic context in terms of the logical form of some node in the structure. For support verbs, it might not be enough to require the presence of some specific non-logical constant in the CONTENT value, but we must require a concrete word to occur as (the lexical head of) the direct object of the support verb.[17] For this reason, we assume that the support verb imposes a restriction on the phonology of the lexical head of the NP that functions as its second semantic argument. This is expressed in the following lexical entry for the support verb *make*.

(515) The lexical entry for the support verb *make*:

$$\mathbf{E}\boxed{1}\ \mathbf{E}\boxed{2}\ \mathbf{E}\boxed{3}\ \mathbf{E}\boxed{4}\ \mathbf{E}\boxed{5}$$

$$\boxed{1}\begin{bmatrix} \textit{word} & \\ \textsc{phon} & \langle \textit{make} \rangle \\ \textsc{s l} & \begin{bmatrix} \textsc{cat} & \begin{bmatrix} \textsc{head} & \textit{verb} \\ \textsc{subcat} & \left\langle \text{NP}, \text{NP}\begin{bmatrix}\textsc{loc}\ \boxed{3}\end{bmatrix} \right\rangle \end{bmatrix} \\ \textsc{cont} & \lambda y \lambda x.\mathsf{make}''_{@}(x,y) \end{bmatrix} \\ \textsc{coll} & \boxed{2} \end{bmatrix}$$

$$\textbf{and}\ \ \texttt{dominate}(\boxed{2},\boxed{4})$$

$$\textbf{and}\ \ \boxed{4}\begin{bmatrix}\textsc{s loc}\ \boxed{3}\end{bmatrix}$$

$$\textbf{and}\ \ \texttt{lexical-head}(\boxed{4},\boxed{5})$$

$$\textbf{and}\ \ \boxed{5}\begin{bmatrix} \textsc{phon} & \langle \textit{headway} \rangle\ \textbf{or}\ \ \langle \textit{decision} \rangle\ \textbf{or}\ \ \ldots \\ \textsc{s l c} & \begin{bmatrix}\textsc{head}\ \textit{noun}\end{bmatrix} \end{bmatrix}$$

Let us examine this lexical entry in detail. The word *make* is described as a verb with two NPs on its SUBCAT list. The first NP is the subject, the second is the direct object. As we will need the direct object for later reference, we have put the tag [3] on its LOCAL value. The CONTENT value of the verb is that of a regular transitive verb, i.e., it is of the type $(se)((se)t)$. As noted above, this is a simplification of the facts.

The COLL value of the verb is marked with the tag [2]. We do not specify the syntactic relation between this sign and the verb. This is parallel to the specification on the verb *pull* in (510a) and motivated by the fact that in the IE *pull strings* and in support verb constructions, the direct object may be absent from the minimal clause of the verb.

The first relational call in the lexical entry states that the sign [2] dominates a sign [4]. The sign [4] is further specified as having the same LOCAL value as the second element on the verb's SUBCAT list, indicated by the tag [3].

In the second relational call, we establish that the tag [5] is used to refer to the lexical head of the sign [4]. This lexical head must be a noun and it must have one of the phonologies compatible with the support verb. In the case of the English verb *make*, the disjunction in the PHON specification contains the strings *headway* and *decision*, but not the string *shower*, as the VP *make a shower* is not a grammatical support verb construction in English, in contrast to the VP *take a shower*.

The relation between the verb *make* and the sign which has the required phonology is quite indirect, i.e., we need the sign [4] to mediate between the COLL value and the sign [5]. This is due to two complicating factors: First, the support verb construction allows for the extraction of its direct object. In our theory of extraction, all we know is that the filler has the same LOCAL value as the element which (originally) occurred on the SUBCAT list of the verb. Thus, we know that the sign [4] has the LOCAL value [3]. Second, the direct object may be syntactically complex. Therefore, the lexical head of the sign [4] need not have the LOCAL

[17] See the discussion in Section 6.2 and especially examples (359) to (364).

value [3]. The support verb, however, imposes a restriction on the lexical head of the direct object. When we go through the criteria of regularity, we will see how the specifications in the lexical entry interact to achieve the required distribution of the IE.

The IE *make headway* is irregular with respect to the first semantic criterion simply due to the fact that the noun *headway* does not occur at all in English outside the particular IE. This observation is captured adequately if we assume that there is no other lexical entry in the grammar with the phonology specification ⟨*headway*⟩. The IE is, however, regular with respect to the second semantic criterion, i.e., under the meaning assignment made in the lexical entries above, the meaning of the IE is simply the result of combining the semantic contributions of the words in a regular way.

Being an internally regular IE, the expression *make headway* allows for internal modification. We have pointed out in Chapter 6 that, because the semantic contribution of the support verb is very limited in support verb constructions, it is sometimes hard to make a difference between semantic modification of the noun and semantic modification of the entire VP. In the analysis of support verb constructions as internally regular IEs, allowing for semantically internal modification is the default case. Consider the example in (367a) which is a repeated in (516a). In (516b) we have sketched a logical form for this sentence.

(516) a. He is making interesting headway.
b. $\exists y[\mathsf{headway}'_{@}(y_{@}) \wedge \mathsf{interesting}'_{@}(y_{@}) \wedge \mathsf{make}''_{@}(x_{@}, y_{@})]$

Let us first consider the noun *headway*. The minimal clause of the noun is the entire sentence. Within the logical form of this sentence as given in (516b), the constant $\mathsf{headway}'$ stands in the right relation to an occurrence of the constant make''. Thus, the minimal clause of the noun is its COLL value and satisfies the description attached to it.

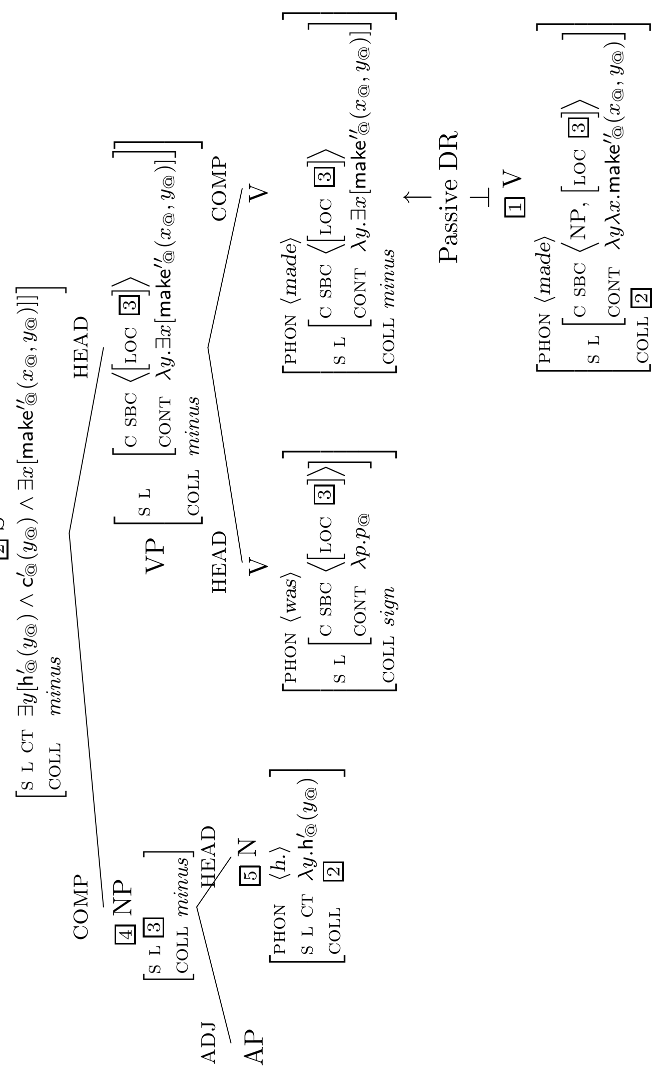
Matters are slightly more complicated for the verb *make*. The COLL value of the verb is not syntactically restricted. We can therefore assume that it is the overall sentence in (516a). The verb is transitive, and, thus, has a second element on its SUBCAT list. As the sentence is in the active voice, the direct object is realized syntactically as the sister of the verb. The direct object has a SYNSEM value which is identical to the second element on the verb's SUBCAT list and its phonology is *interesting headway*. The overall sentence, i.e., the COLL value of the verb, dominates this NP. Thus, it dominates an NP whose LOCAL value is identical to that of the second element on the verb's SUBCAT list. This NP has a lexical head, the noun *headway*. The phonology of this lexical head is among those specified as compatible with the support verb *make*. These considerations show that the distributional requirements of the verb *make* are met in the sentence (516a).

As far as pronominalization is concerned, we have observed in Chapter 6 that the noun *headway* does not allow for pronominal reference by a personal pronoun, but does allow the indefinite *some*. We have argued that this behavior is parallel to that of mass nouns and that, therefore, the noun *headway* should simply be analyzed parallel to other mass nouns. The logical forms used in this thesis do not express the count/mass distinction. We assume that once our semantic analysis is enriched by this distinction, the pronominalization data for the noun *headway* will follow automatically.

We can, next, address the syntactic criteria. Not much needs to be said about the first three syntactic criteria. The fact that the noun *headway* does not occur outside the support verb construction follows from its COLL specification in the lexical entry. The words that occur in the IE further combine according to general ID schemata. As there are no restrictions on the syntactic complexity of the direct object, an adjectival modifier can occur inside the NP. See the discussion of sentence (516a) above.

Similarly to modification, passive is also possible. Consider the following sentences, repeated from Chapter 6.

FIGURE 8.9. The structure of sentence *Considerable headway was made*:

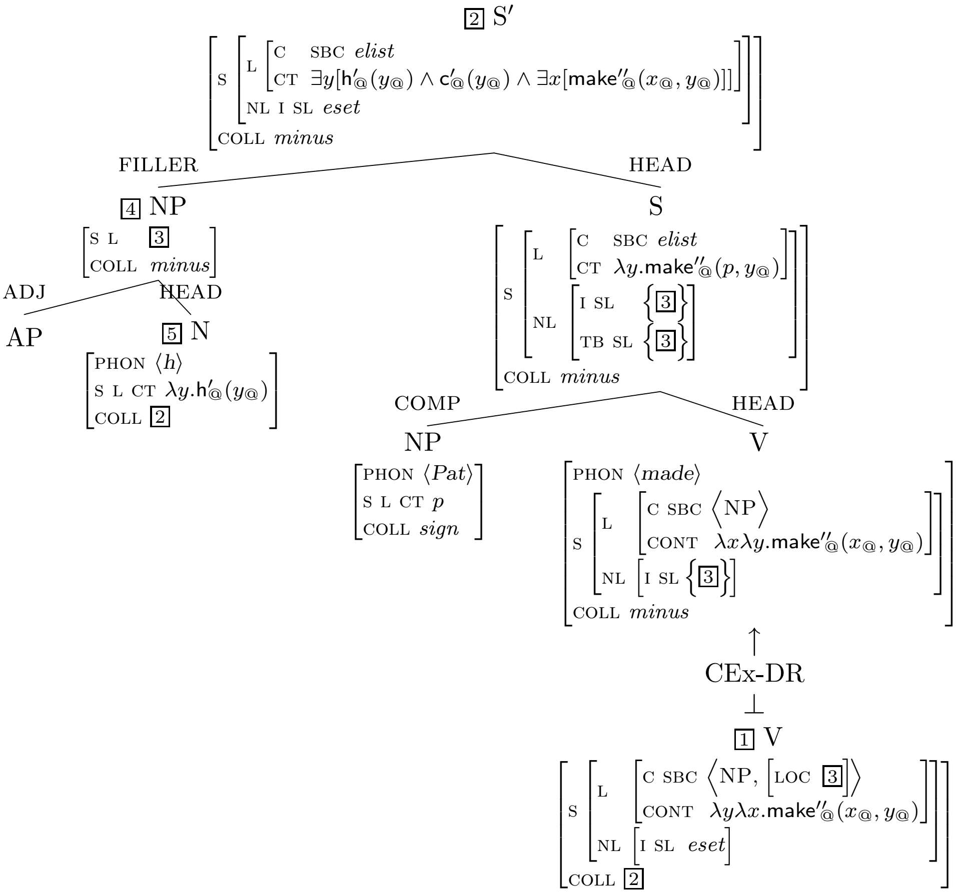


(517) a. Considerable headway was made over the last 15 years.
b. Considerable headway seems to be made in this area of research.
c. The researchers expect considerable headway to be made in the near future.

In Figure 8.9 we have sketched the structure of a simplified version of sentence (517a). In this tree, we only use those tags which occur in the lexical entry of the verb *make* in (515). We have abbreviated the semantic constant headway′ by h′ and the constant considerable′ as c′. To simplify the structure, we have not specified the AP in detail, nor have we given any indication where the existential quantifier that binds the variable y comes from. This quantifier will be provided by whatever treatment of mass nouns is assumed.

We can, now, go through the requirements in the lexical entry to see that all these requirements are met by the verb *made* that is the input to the derivational rule in the given structure. The tag [1] refers to this verb itself. As required, this verb has two elements on its subcat list, the last one's LOCAL value being referred to with the tag [3]. The COLL value of the verb is the sign with the tag [2]. This sign dominates the verb [1], since we

FIGURE 8.10. The structure of sentence *Considerable headway Pat made.*:



interpret dominance as reaching into the input of a derivational rule. Therefore, the sign [2] fulfills the requirements of the COLL PRINCIPLE.

The sign [2] further dominates a sign [4]. This sign is the subject of the passive sentence. Under the standard analysis of passive and raising in HPSG, the SYNSEM value of the subject in the passive is identical to the second element on the subcat list of the active counterpart of the passivized verb. Due to this synsem identity, the LOCAL value of the subject, i.e. of the sign [4] in Figure 8.9, is identical to the LOCAL value of the second element on the subcat list of the verb that is the input to the passive DR. This LOCAL value is referred to by the tag [3]. Consequently, the sign [4] is as described in the lexical entry of the verb *make*. The subject NP [4] has the noun [5] as its lexical head. This noun has the phonology *headway* which is among the disjunction of possibilities in the description of the sign [5] in the lexical entry of the support verb. Thus, the word *make* as it occurs as the input to the derivational rule is described by the lexical entry in (515).

As we have seen in the empirical chapter, the direct object may also be topicalized in the IE *make headway*. Again, the account of topicalization is straightforward. In Figure 8.10 we give the structure of a simple sentence with topicalization. We add the tags that are mentioned in the lexical entry of the verb *make*. As before, the requirements of the word that serves as input to the extraction DR need to be considered.

What is crucial in the syntactic structure in Figure 8.10 is the way the topicalized NP *considerable headway* ([4]) is related to the verb [1] which is the input to the derivational rule.

The second element on this verb's subcat list has the LOCAL value [3]. This value re-appears in the output of the complement extraction DR in the INHERITED SLASH value. By the NONLOCAL FEATURE PRINCIPLE this slash value also appears on the S node. Finally, the HEAD FILLER SCHEMA requires the *local* object [3] to be the LOC value of the filler daughter ([4]). As a consequence, the filler daughter is the only sign in the utterance whose LOCAL value is identical to that of the second element on the subcat list of the verb [1]. Finally, as the lexical head of the filler is a noun whose phonology is compatible with the requirements of the support verb *make*, the verb's distributional restrictions are all met in this sentence.

After the rather detailed discussion of passive and topicalization structures with support verbs, we can turn to the next criterion of syntactic regularity: relative clauses. In (518) we repeat the sentences from (375) in Chapter 6.

(518) a. You have made considerable headway/progress that will be of tremendous use for the entire project.
b. I admire the ?headway/progress that you have made since we met last time.

We can account for sentence (518a) without any changes: the noun *headway* is the lexical head of the direct object of the support verb *make*. Matters are more difficult in the case of sentence (518b). There, the support verb is inside the relative clause, but the noun *headway* is not. In this thesis, we have not presented an analysis of relative clauses. In the analysis of relative clauses in Pollard and Sag 1994 (chapter 5) and Sag 1997 the LOCAL value of the noun to which a relative clause is attached is not required to be identical to the LOCAL value of any sign inside the relative clause. As a consequence, the LOCAL value of the noun *headway* in sentence (518b) is not identical to the element in the slash value of the verb *make* in the relative clause. The only identity that is assumed in these two analyses is index identity. As the attribute INDEX is part of the content in Pollard and Sag 1994 and Sag 1997, but not defined on the kind of content values assumed in this thesis, we cannot take over any of these approaches directly in the present framework.

In an account of the relative clause data, we basically have two options: either we try to build our account of relative clauses in such a way that identity of LOCAL values is achieved, or we change the lexical entry of the verb *make* in such a way that it does not require the identity of LOCAL values, but of some smaller structure within a *local* object. We leave the decision to further research.

In short, in our analysis of the IE *make headway*, we assumed that the verb *make* is just the normal support verb *make* which also occurs with other nouns such as *decision*. The distributional requirements of the support verb express a restriction of the words that may appear as the lexical head of their direct object. The noun *headway*, on the other hand, is very similar to the idiomatic uses of the nouns *beans* and *strings* as they occur in the other IEs discussed in this section. The distributional requirements of this noun constrain the logical form of the clause which contains the word.

make a decision. In the last paragraph of this section, we address the IE *make a decision*. As we have seen in the discussion of the data, this IE shows hardly any signs of irregularity. In this paragraph, we will not go through all the criteria of regularity, instead, we will concentrate on the differences between the expressions *make headway* and *make a decision*.

The only irregular aspect is that the verb *make* cannot combine with all nouns to form a support verb construction. In the lexical entry for the support verb *make* in (515), the noun *decision* was included in the list of possible direct objects for this support verb. It will, however, not be included in the similar lists that are part of the lexical entries for the support verbs *do*, *wage* and *commit*. Thus, in the combination *make a decision*, the noun *decision* is licensed by the same lexical entry that licenses it in free contexts as well.

It follows that this combination is also regular with respect to the first syntactic criterion. A further difference between *make headway* and *make a decision* lies in the fact that the noun *headway* cannot be pronominalized. In the preceding paragraph, we have taken this fact to follow from the mass term properties of the noun *headway*. As the noun *decision* is not a mass noun, the attested contrast is expected.

In this section, we have completed the presentation of the analysis of internally regular IEs. It has been shown that the data collected in Chapter 6 can be captured in the presented architecture. For many properties we have pointed to advantages of our approach over the accounts presented in Chapter 7.

In the next section, we will attempt to streamline the use of the COLL feature. So far, we have assumed boolean values on phrases and either a sign or a *minus* as possible values on words. We will present a more uniform feature geometry where the attribute COLL is list-valued on all subsorts of *sign*. We will motivate this extension with an example that indicates that not only words, but also internally irregular phrases can have idiosyncratic distributional restrictions.

8.3. A Unified Treatment of the COLL Attribute

In this section we reconsider the appropriateness conditions for the sort *sign*. We will optimize our treatment of the COLL feature across the subtypes of *sign* by making the COLL feature list-valued. We will demonstrate that with this change, we can preserve the distinctions that were assumed in the previous sections. We will provide and discuss discuss additional data in support of this new architecture.

In the previous sections of this chapter, we worked with the sort hierarchy and appropriateness for the sort *sign* presented in (500) which we repeat in (519) for convenience.

(519) Sort hierarchy and appropriateness conditions below the sort *sign* (from (500)):

sign
PHON *list*
SYNSEM *synsem*

word
STORE *list*
COLL *sign-or-minus*

phrase
DTRS *const-struc*
COLL *boolean*

As pointed out above, while we assume a single attribute, COLL, for our treatment of IEs, the COLL specifications assumed for words and phrases do not indicate a unified concept of the function of this attribute. In Section 8.1, we used the $+/-$ distinction to differentiate between regular and irregular phrases. IEs such as *kick the bucket* were treated as irregular phrases. In Section 8.2, we differentiated between words that are licensed by a lexical entry and those that are licensed by some derivational rule. For the former we postulated a sign-valued COLL specification, for the latter the COLL value *minus* was assumed. The fact that we use the attribute COLL for both internally irregular phrases such as *kick the bucket* and the words that constitute internally regular IEs such as the words *spill* and *beans* makes it necessary to have a COLL specification for all phrases and for all words.

In the following, we will not discuss all conceivable alternatives, but present one concrete proposal and its advantages. In (520), the final version of the sort hierarchy and the appropriateness conditions of the sort *sign* are given as assumed in this thesis.

(520) Sort hierarchy and appropriateness conditions below the sort *sign* (final version):

sign
PHON *list*
SYNSEM *synsem*
COLL *list*

word — STORE *list*
phrase — DTRS *const-struc*

In this new sort hierarchy, the sort *sign* is still assumed to have just two subsorts, *word* and *phrase*. The treatment of the COLL value is, however, quite different from what we have defined in (500) in the preceding section. Instead of assuming a boolean value for phrases and a sign value for some words, we make the attribute list-valued for all signs.

The sort *list* has two subsorts, *empty-list* and *nonempty-list*. We can use these two subsorts to take over the function of the boolean values *minus* and *plus* that we used in Section 8.1, i.e., we replace the specification [COLL −] used so far by the specification [COLL *elist*]. For internally irregular phrases such as the VP *kick the bucket*, we assume the specification [COLL *nelist*], instead of [COLL +].

Analogously, we replace the [COLL −] specification on derived words (which is enforced by the STORE-COLL PRINCIPLE in (501)) by the specification [COLL *elist*]. For non-derived words, on the other hand we use the value *nelist* instead of having a sign in the COLL feature.

This has the desirable effect that internally irregular phrases and non-derived words are considered as a natural class with respect to their COLL specification: both have the value *nelist*. Empirically, they also form a natural class that contains all elements whose internal properties are not fully predictable from their parts.

This can best be illustrated with combinatorial semantics. For regular phrases, the SEMANTICS PRINCIPLE defines the way we combine the semantic contribution of the daughters to form that of the phrase. Similarly in the case of derived words: Given the content of the input word of a derivational rule and the specification of the derivational rule, the content of the output is fully predictable. There are two cases where the CONTENT value of a sign is not predictable in this way: First, obviously, if the sign is a non-derived word, then it does not contain any smaller signs whose semantic contribution could be used as the basis for the logical form of the word. Second, if the sign is a phrase, but internally irregular. Then, its logical form need not stand in a transparent relation to that of its daughters. Thus, in the case of non-derived words and internally irregular phrases, the CONTENT value is not predictable. Similarly for the other properties of a sign: while they are predictable for regular phrases and derived words, they are arbitrary in the case of non-derived words and internally irregular phrases.

All in all, this similarity between non-derived words and internally irregular phrases is captured in the same COLL specification. This specification also enables us to combine the WORD PRINCIPLE and the INTERNAL IRREGULARITY PRINCIPLE into a single principle which lists descriptions of all signs that have a COLL value of sort *nelist*. We call this combined principle the LEXICON PRINCIPLE (LexP).

(521) The LEXICON PRINCIPLE (LexP):

$$\begin{bmatrix} \textit{sign} \\ \textsc{coll} \ \textit{nelist} \end{bmatrix} \Rightarrow (\mathrm{LE}_1 \textbf{ or } \ldots \mathrm{LE}_m \textbf{ or } \mathrm{PLE}_1 \textbf{ or } \ldots \mathrm{PLE}_n)$$

The consequent of the LexP in (521) is a big disjunction of all the lexical entries (LE_i) from the old WORD PRINCIPLE and all the phrasal lexical entries from the old IIP (PLE_i).

After this re-ordering of the principles on phrases, we finally address the question of what is on the COLL list of words and internally irregular phrases. We assume that the COLL value contains at most one sign. To consider a concrete example, the lexical entry of the word *read* as it occurs in the free combination *read a book* contains the following COLL specification.

(522) $$\begin{bmatrix} \textit{word} \\ \vdots \\ \textsc{coll}\ \left\langle \begin{bmatrix} \textit{sign} \end{bmatrix} \right\rangle \end{bmatrix}$$

As a consequence of this extremely vague specification, any sign may occur in the COLL value. There is, however, a restriction, imposed by the COLL PRINCIPLE in (505). This principle requires that in every sign s, the signs in the COLL value of the words dominated by s must either dominate s or be dominated by s. Thus, whatever is in the COLL value of the normal word *read* is a sign which contains a grammatical occurrence of this word. For utterances, this means that whatever is in the COLL values of the words that the utterance is composed of is, again, a subpart of this utterance. In this way, the COLL PRINCIPLE guarantees that the COLL value does not introduce undesired material.

The reader will have noticed that our formulation of the COLL PRINCIPLE in (505) is not adapted to the new feature geometry below the sort *sign*: it assumes a sign-, instead of a list-valued COLL attribute on words and does not consider the COLL values of phrases. In (523) we give the final version of the COLL PRINCIPLE.

(523) The COLL PRINCIPLE (final version):

$$\textit{sign} \Rightarrow \left(\begin{array}{l} \mathbf{A}\boxed{1}\ \mathbf{E}\boxed{3}\ \mathbf{A}\boxed{2} \\ \left(\begin{array}{l} \texttt{dominate}(:,\boxed{1}) \\ \textbf{and}\ \ \boxed{1}\begin{bmatrix}\textsc{coll}\ \boxed{3}\end{bmatrix} \\ \textbf{and}\ \ \texttt{member}(\boxed{2},\boxed{3}) \end{array} \right) \Rightarrow \left(\begin{array}{l} \texttt{dominate}(:,\boxed{2}) \\ \textbf{or}\ \ \texttt{dominate}(\boxed{2},:) \end{array} \right) \end{array} \right)$$

This new version of the COLL PRINCIPLE fits the revised sort hierarchy below *sign*. Just as in the earlier formulation, the consequent of this principle contains an implication. In the antecedent of this implication, the tag $\boxed{1}$ is used for the signs (words or phrases) that are dominated by the described sign. The tag $\boxed{3}$ denotes the COLL list of the sign referred to by $\boxed{1}$. The tag $\boxed{2}$ then is used to pick out the elements on the COLL lists of all the signs dominated by the considered sign. The consequent of the embedded implication, then, requires for every COLL element $\boxed{2}$ that the described sign either dominate $\boxed{2}$ or be dominated by $\boxed{2}$.

Having adjusted the COLL PRINCIPLE to the new feature geometry, we can consider the COLL values for the four classes of signs, i.e., internally regular phrases, derived words, non-derived words and internally irregular phrases. In regularly combined signs, i.e., regular phrases and derived words, the COLL list is empty. In the case of non-derived words we assumed a single sign as the COLL value in Section 8.2. Thus, it is most natural to assume that the COLL list is at most a singleton list which contains exactly the sign that appeared as the COLL value in earlier sections. This assumption is fully unproblematic in the case of words, but the following question arises: is there motivation for having a sign inside the COLL value of irregular phrases? We think that there is evidence supporting that there are some irregular phrases for which we must assume idiosyncratic distributional requirements. But, before we turn to these examples, we should briefly indicate how the lexical entries given in the previous section must be changed to accommodate the new feature geometry.

In (502) we gave the lexical entry of the idiomatic use of the word *spill*. There, we indicated a sign in the COLL value. Now that we have altered the appropriateness conditions for the COLL feature, we must modify this lexical entry as well.

(524) The lexical entry for the idiomatic use of *spill*:

$$
\mathbf{E}\boxed{1}\ \mathbf{E}\boxed{2}\ \mathbf{E}\boxed{3}\ \boxed{1}\begin{bmatrix} \textit{word} \\ \text{PHON}\ \langle \textit{spill} \rangle \\ \text{S L} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{SUBCAT} & \left\langle \text{NP, NP, (PP}[\textit{to}]) \right\rangle \end{bmatrix} \\ \text{CONT}\ \lambda y \lambda x.\mathsf{spill}''_{@}(x,y) \end{bmatrix} \\ \text{COLL}\ \left\langle \boxed{2} \right\rangle \end{bmatrix}
$$

$$
\textbf{and}\ \boxed{2}\begin{bmatrix} \textit{sign} \\ \text{S L CONT}\ \ldots [\mathsf{the}\ v : \ldots \mathsf{beans}''_{@}(v_{@}) \ldots](\ldots \mathsf{spill}''_{@}(w_e, v_{@}) \ldots) \ldots \end{bmatrix}
$$

$$
\textbf{and}\ \texttt{minimal-clause}(\boxed{2},\boxed{1})
$$

In (524) we have simply changed the COLL specification from a sign to a singleton list that contains a sign.

In the new architecture, we have a non-empty COLL value for internally irregular phrases. This leads us to the expectation that we find internally irregular phrases with idiosyncratic distributional properties. This expectation, we think, is actually born out, even though none of the IEs in our investigation is of this kind. Consider, however, the following idiomatic expression in German.

(525) Hans weiß, wo (der) Barthel den Most holt.
Hans knows, where the ?? the ?? gets
'Hans knows every trick in the book.'

As indicated by the "??" in the glosses the meaning of the words *Barthel* and *Most* in these expressions is not transparent to most speakers of German. And, in addition, the overall meaning of the expression does not help to assign these words a meaning.

This lack of transparency can be illustrated with the information given for this expression in the *Deutsches Wörterbuch von Jacob und Wilhelm Grimm*. There, it is speculated that the noun *Bart(h)el* might be related to the name *Bartholomäus*. The word *Most* is uncommented, which might be an indication that it is taken to be *cider*, as there is a word *Most* with this meaning in German. This dictionary entry ends with the remark "der ursprung dieser sprichwörter liegt im dunkel" (the origin of these proverbs is unknown). According to Wolf 1956, the noun *Barthel* is related to the Hebrew word for *iron*, which is used in the meaning of *crowbar*. The word *Most*, then, is related to the Hebrew word for *money* which also appears in the slang term *Moos* (*cash*) in today's German. Thus, the expression originally meant *to know where to get cash with a crowbar*.[18]

The complement clause *wo (der) Barthel den Most holt* seems to be internally irregular: First, as pointed out before, the words *Barthel* and *Most* do not occur in the meaning that they have in this expression elsewhere in German. Second, the meaning of the clause cannot be computed by combining the meanings of its parts, even under the etymologically correct interpretation of the words *Barthel* and *Most*. Third, the syntactic form of the embedded clause is frozen. As the following examples show, the sentence cannot be used in passive (a), nor with a past tense form of the verb (b), nor may it contain any other lexical material (c,d). In the examples, we use the nouns *crowbar* and *cash* to gloss *Barthel* and *Most*.

(526) a. * Er weiß, wo der Most vom/ mit (dem) Barthel geholt wird.
he knows, where the cash by the/ with the crowbar got is
b. Er wusste, wo (der) Barthel den Most holt/ ?? holte
he knew where the crowbar the cash gets/ got

[18] I am grateful to Martin Engelhard and Michael Kaschek for pointing me to references for the origin of this expression.

c. * Er weiß, wo (der) Barthel morgen/ gerade den Most holt.
he knows where the crowbar tomorrow/ right now the cash gets

d. * Er weiß, wo (der) Barthel viel Most holt.
he knows where the crowbar a lot of cash gets

These data indicate, that we must assume an internally irregular phrase with the phonology *wo (der) Barthel den Most holt* and the meaning *every trick in the book*. While this phrase is internally irregular, it is also bound to occur as the complement clause to the verb *wissen* (*know*). In (527) we give several matrix verbs that take embedded interrogative clauses as complements. As indicated, only the combination with the verb *wissen* (*know*) is possible. As we are no longer concerned with the internal structure of the phrase *wo (der) Barthel den Most holt*, we gloss it as *every trick in the book* in the following examples.

(527) a. Er weiß, wo (der) Barthel den Most holt.
he knows every trick in the book

b. * Er ahnt, wo (der) Barthel den Most holt.
he suspects every trick in the book

c. * Er fragt sich, wo (der) Barthel den Most holt.
he asks himself every trick in the book

d. * Er wundert sich darüber, wo (der) Barthel den Most holt.
he wonders himself about every trick in the book

The combination of the verb *wissen* with the phrase *wo (der) Barthel den Most holt* is, however, regular. In particular, the complement clause can be topicalized (a). In addition, the complement clause must establish some discourse referent which can be referred to by a personal pronoun (b,c).

(528) a. [Wo (der) Barthel den Most holt], weiß Peter schon lang.
every trick in the book knows Peter already for a long time

b. Peter weiß, wo (der) Barthel den Most holt, aber Hans weiß es nicht.
Peter knows every trick in the book but Hans knows it not
'Peter knows every trick in the book, but Hans doesn't.'

c. Peter weiß, wo (der) Barthel den Most holt, und Hans ahnt es.
Peter knows every trick in the book and Hans suspects it
'Peter knows what's what, and Hans suspects it.'

But note that the full form of the complement clause cannot occur with the matrix predicate *ahnen* (*suspect/guess*)

(529) * Hans ahnt, wo (der) Barthel den Most holt, und Peter weiß es.
Hans suspects every trick in the book and Peter knows it

From these data we conclude that the irregular phrase *wo (der) Barthel den Most holt* combines with the verb *wissen* (*know*) in a regular way. But the irregular phrase has an idiosyncratic distributional property that it cannot combine with any other verb. In the new feature geometry, we can account for this fact: as the phrase *wo (der) Barthel den Most holt* is internally irregular, it has a non-empty COLL list. By the COLL PRINCIPLE, we know that whatever is in the COLL list, must be a sign that dominates the irregular phrase. We can, now, impose a further restriction on the COLL element: we require it to dominate an occurrence of the verb *wissen* (*know*) on whose SUBCAT list we find the SYNSEM value of the irregular phrase.

In (530), we sketch the phrasal lexical entry for the irregular phrase *wo (der) Barthel den Most holt*. We only state the information that is necessary to express the distributional requirements of the phrase. We leave out its semantics or other syntactic properties.

(530) The PLE for the internally irregular phrase *wo (der) Barthel den Most holt*:

$$
\begin{array}{l}
\mathbf{E}\boxed{1}\ \mathbf{E}\boxed{2}\ \mathbf{E}\boxed{3} \\
\left[\begin{array}{ll}
\textit{phrase} & \\
\text{PHON} & \langle \textit{wo, (der), Barthel, den, Most, holt} \rangle \\
\text{SYNSEM} & \boxed{1} \\
\text{COLL} & \left\langle \boxed{2} \right\rangle
\end{array}\right] \\
\textbf{and}\ \ \texttt{dominate}(\boxed{2},\boxed{3}) \\
\textbf{and}\ \ \boxed{3}\left[\begin{array}{l}
\textit{word} \\
\text{PHON}\ \langle \textit{wissen} \rangle \\
\text{SYNS LOC CAT SBC} \left\langle \textit{synsem},\ \boxed{1} \right\rangle
\end{array}\right]
\end{array}
$$

It should be stressed that within the the architecture of the COLL specification presented in Section 8.1 we would not have been able to express the distributional requirements of the irregular phrase. In Section 8.1 phrases only had boolean values in their COLL feature. Thus, we could only indicate whether a phrase is regular or not. With the new architecture we can not only specify the (ir)regularity of a phrase, but, additionally, we can restrict the distribution of irregular phrases if necessary.

The expression *wissen, wo (der) Barthel den Most holt* is an example which clearly shows that irregular phrases can in principle also have distributional restrictions. Therefore it gives empirical support to the changes in the sort hierarchy that we propose in this section. Our point would certainly be stronger, if we could give a whole list of expressions which show similar behavior. We leave it to further research to gather more examples.

To show that the architecture presented in this section is not merely a stipulation, we would also be forced to prove that there are no regular phrases or derived words that have idiosyncratic distributional properties which are not predictable from properties of their components. I.e., we must justify the assumption that signs which have predictable internal properties also have a free (or at least predictable) distribution as expressed by the empty COLL list assumed in this section. Although we were not confronted with any counterexamples, we cannot prove that this assumption is correct. We present our assumption in the form of an hypothesis which still remains to be proven in (531).

(531) The *Predictability Hypothesis*:
For every sign whose internal properties are fully predictable, the distributional behavior of this sign is fully predictable as well.

Further research might show that the *Predictability Hypothesis* is a methodological principle rather than an empirically testable hypothesis, just as was shown for compositionality in semantics (Janssen, 1997).

In this section we have presented a unified account of the COLL feature. We use the value of this attribute to differentiate between signs whose properties are fully predictable from the properties of the signs that compose them, i.e., derived words and regular phrases, and signs whose properties are not, i.e., non-derived words and irregular phrases. For the first group of signs, we have proposed an empty COLL value, for the second group, the COLL value is non-empty. Every sign that has a non-empty COLL value is listed in the lexicon, i.e., as a disjunct in the LEXICON PRINCIPLE. We have further assumed that lexical signs can impose distributional requirements on the linguistic contexts in which they may occur. To formalize these requirements, we have made the linguistic context locally available for lexical signs in their COLL element. We have shown that there are instances of such requirements attested for both non-derived words and irregular phrases.

8.4. Appendix: External Modification

After the presentation of our analysis of internally irregular and internally regular idiomatic expressions, we will briefly return to an issue that is peripheral to our approach, but normally arises in the context of idiomatic expressions: the question of how external modification can be accounted for. In (532), we repeat the two examples of semantically external modification that we have encountered in previous sections and chapters.

(532) a. An occasional sailor came into the bar.
= Occasionally, a sailor came into the bar. (Nunberg et al., 1994, fn. 15)
b. With that dumb remark at the party last night, I really kicked the social bucket.
= Socially, I kicked the bucket. (Ernst, 1981, p. 51)

The special property of these sentences is that the adjectival modifier, while occurring inside an NP, seems to modify some verbal projection. It has been observed in Ernst 1981 that this phenomenon is quite common for idiomatic expressions such as *kick the bucket*. Wasow et al. 1983 point out that it is, however, not restricted to idiomatic expressions but also attested in free combinations, as can be seen in the example in (532a).

In the presentation of our account of internally irregular IEs in Section 8.1, we mentioned that the phrasal lexical entry given for *kick the bucket* in (471) excludes such modification, because the idiomatic phrase disallows semantic material in the NP complement which does not stem from either the determiner *the* or the head noun *bucket*. In this section, we will moderately change the PLE for this IE in order to allow for modification as illustrated in the example in (532b).

The aim of this section is quite modest. We will not provide a discussion of the empirical phenomenon of external modification, all that we want to achieve is to propose some analysis within the framework of LF-Ty2 which works for externally interpreted modification in the examples in (532). We assume that the logical form of these sentences is as given in (533). It is, thus, our goal to provide a system that assigns these logical forms to the sentences.

(533) a. The logical form of (532a):
$\mathsf{occasionally}'_{@}(\lambda @.\exists x[\mathsf{sailor}'_{@}(x_{@}) \wedge \mathsf{come\text{-}into\text{-}the\text{-}bar}'_{@}(x_{@})])$
b. The logical form of (532b);
$\mathsf{socially}'_{@}(\lambda @.\mathsf{die}'_{@}(i))$

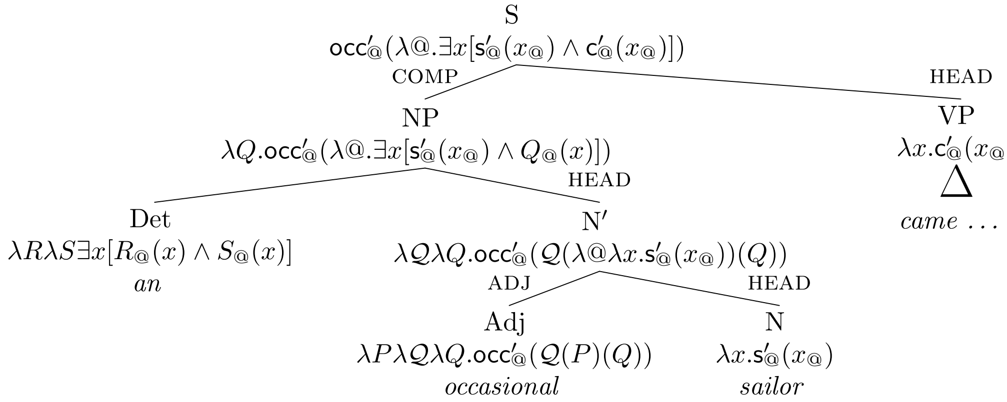
We will first concentrate on the example in (532a), i.e., we will show how we can allow an adjectival modifier to out-scope the NP in which it is contained. To do this, as we will see, it is enough to assign the adjective *occasional* a slightly more complex logical form than that used for internal modification.

In the semantic framework introduced in Chapter 4, an adjectival modifier normally has the semantic type $s((s(et))(s(et)))$, i.e., it takes the intension of a property as input type and returns a property as its output type. To account for the reading in (533a), we assume that the adjective *occasional* has a logical form which is of a more complex semantic type. In (534), we give the lambda term which we assume in the derivation of this reading: here, the first semantic argument is the restrictor of a quantifier, its second semantic argument is the quantifier and, finally, its third semantic argument is the scope of the quantifier.

(534) The logical form of the adjective *occasional*:
$\lambda P \lambda \mathcal{Q} \lambda Q.\mathsf{occ}'_{@}(\mathcal{Q}(P)(Q))$

This specification is, in fact, all that is needed to account for the external modification interpretation in sentence (532a). To see how we achieve this interpretation, consider the structure of sentence (532a) as given in Figure 8.11. The adjective combines with the noun

FIGURE 8.11. The structure of sentence (532a):



sailor and forms a functor which takes the quantifier *a* as its argument. Internally, the semantic contribution of the noun is taken to be the first argument of the variable $\mathcal{Q}$, which is of the semantic type of a quantifier.

The tree in Figure 8.11 shows the adjective, while combining with the noun *sailor*, introduces a non-logical constant which out-scopes the rest of the semantic material in the sentence. Technically this is achieved mainly by making the adjective a functor which takes the determiner as its semantic argument, instead of being the argument of the determiner as in the case of semantically internal modification.

We can apply this analysis of external modification to our treatment of the IE *kick the bucket*. For this purpose, we must assume first that the adjective *social* has a logical form similar to that of the adjective *occasional*.[19] This logical form is given in (533b).

(535) The logical form of the adjective *social* as needed for external modification:
$\lambda P \lambda \mathcal{Q} \lambda Q.\mathsf{socially}'_{@}(\mathcal{Q}(P)(Q))$

Using this logical form for the adjective *social* in the analysis of sentence (532b), we arrive at the following logical form for the NP *the social bucket*.

(536) The logical form of the NP *the social bucket*:
$\lambda Q.\mathsf{socially}'_{@}(\lambda @.[\mathsf{the}\ y : \mathsf{bucket}'_{@}(y_{@})](Q_{@}(y)))$

Note that the logical form in (536) is that of a regular NP, i.e., it is derived in exactly the same way as the logical form of the NP *an occasional sailor* in the tree in Figure 8.11.

In order to allow for semantically external modification of the IE *kick the bucket*, we must create the possibility for the predicate $\mathsf{socially}'$ to scope over the semantic contribution of the idiomatic VP, i.e., the term $\mathsf{die}'_{@}(x_{@})$ instead of over the semantic contribution of the determiner *the*. In (537), we compare the logical form of the NP *the social bucket* in (536) with that of the VP, which is just like the logical form of the entire sentence in (533b), but still needs to combine with the subject NP.

(537) *the social bucket* $\lambda Q.\mathsf{socially}'_{@}(\lambda @.[\mathsf{the}\ y : \mathsf{bucket}'_{@}(y_{@})](Q_{@}(y)))$
kicked the social bucket $\lambda x.\mathsf{socially}'_{@}(\lambda @.\mathsf{die}'_{@}(x_{@}))$

We achieve the desired logical form if we ignore the lambda abstractor in the two terms, and replace the subterm ϕ of $\mathsf{socially}'_{@}(\lambda @.\phi)$ with the term $\mathsf{die}'_{@}(x_{@})$. Then, $\mathsf{socially}'$ has

[19] In the discussion of the TAG analysis in Abeillé 1995 we have seen that the same assumption was needed. There, we considered a pair of elementary trees for the adjective *social* such that the syntactic tree adjoined to an N node, whereas the semantic tree adjoined to an F node (cf. number (439a) on page 295).

the rest of the logical form of the sentence as its argument. In the formalization of the semantic framework of LF-Ty2, we have introduced a relation `replace` which can be used for the term replacement necessary in the case of the idiomatic expression as well.

In order to allow for semantically external modification in the case of the IE *kick the bucket*, we modify the CONTENT specification in the description of the IE as indicated in (538) and add some more relation calls. The crucial change made in comparison to the PLE in (471) is that we add the replacement operation. The description given in (538) is the final version of the phrasal lexical entry of the IE *kick the bucket*, as we assume it to occur as a disjunct in the LexP.

(538) The phrasal lexical entry for the IE *kick the bucket* (final version):

$$
\mathbf{E}\boxed{7}\ \mathbf{E}\boxed{10}\ \mathbf{E}\boxed{11}
\begin{bmatrix}
\textit{phrase}\\
\text{PHON}\ \boxed{3}\oplus\boxed{4}\\
\text{S L}\ \begin{bmatrix}\text{CAT}\ \begin{bmatrix}\text{HEAD} & \boxed{1}\\ \text{SUBCAT} & \langle\boxed{2}\rangle\end{bmatrix}\\ \text{CONT}\ \lambda x.\boxed{8}\end{bmatrix}\\
\text{DTRS}\ \begin{bmatrix} \textit{head-compl-struc}\\ \text{H-DTR}\ \begin{bmatrix}\textit{word}\\ \text{PHON}\ \boxed{3}\ \langle\textit{kick}\rangle\\ \text{S L}\ \begin{bmatrix}\text{CAT}\ \begin{bmatrix}\text{HEAD}\ \boxed{1}\begin{bmatrix}\textit{verb}\\ \text{VFORM}\ \textbf{not}\ \textit{pas}\end{bmatrix}\\ \text{SUBCAT}\ \left\langle\boxed{2},\boxed{5}\right\rangle\end{bmatrix}\end{bmatrix}\end{bmatrix}\\
\text{N-DTR}\ \begin{bmatrix}\text{PHON}\ \boxed{4}\\ \text{SYNS}\ \boxed{5}\begin{bmatrix}\text{L}\begin{bmatrix}\text{C}\begin{bmatrix}\text{HEAD} & \textit{noun}\\ \text{SUBCAT} & \textit{elist}\end{bmatrix}\\ \text{CT}\ \lambda Q.\boxed{9}\end{bmatrix}\end{bmatrix}\\ \text{DTRS}\begin{bmatrix}\text{H-DTR}\ \boxed{6}\\ \text{N-DTR}\begin{bmatrix}\text{PHON} & \langle\textit{the}\rangle\\ \text{SYNS} & \text{Det}\end{bmatrix}\end{bmatrix}\\ \text{COLL}\ \langle\rangle\end{bmatrix}\end{bmatrix}\\
\text{COLL}\ \left\langle\begin{bmatrix}\textit{sign}\end{bmatrix}\right\rangle
\end{bmatrix}
$$

$$
\textbf{and}\ \boxed{7}\begin{bmatrix}\text{PHON} & \langle\textit{bucket}\rangle\\ \text{SYNS} & \text{N}\end{bmatrix}\ \textbf{and}\ \texttt{lexical-head}(\boxed{6},\boxed{7})
$$

$$
\textbf{and}\ \boxed{10}\left[[\mathsf{the}\,y:\mathsf{bucket}'_{@}(y_{@})](Q_{@}(y))\right]\ \textbf{and}\ \texttt{component}(\boxed{9},\boxed{10})
$$

$$
\textbf{and}\ \boxed{11}\left[\mathsf{die}'_{@}(x_{@})\right]\ \textbf{and}\ \texttt{replace}(\boxed{8},\boxed{11},\boxed{9},\boxed{10})
$$

Even though the description in (538) looks admittedly complicated, it is a direct encoding of the effect of semantically external modification. Let us consider the changes made to the description in (471) above. First, the CONTENT value of the VP is not given as the term $\lambda x.\mathsf{die}'_{@}(x_{@})$. Instead, we allow for further semantic material to intervene between the lambda operator and the term $\mathsf{die}'_{@}(x_{@})$. This is done in the following way: the CONTENT value of the VP as given in (538) is given as $\lambda x.\boxed{8}$, i.e., it is an *abstraction* object with the variable x as its VAR value and the tag $\boxed{8}$ as its ARG value. The term $\boxed{8}$ contains as a subterm the term referred to with the tag $\boxed{11}$. In the last line in (538), this tag is described as referring to the term $\mathsf{die}'_{@}(x_{@})$.

Similarly, the CONTENT value of the NP is not given as $\lambda Q[\mathsf{the}\,y:\mathsf{bucket}'_{@}(y_{@})](Q_{@}(y))$ as in (471). Instead, an external modifier is allowed to occur between the lambda operator λQ and the term $[\mathsf{the}\,y:\mathsf{bucket}'_{@}(y_{@})](Q_{@}(y))$. We can state the content of the NP as having the form $\lambda Q.\boxed{9}$, where the tag $\boxed{9}$ stands for a term which contains a subterm of the form [the y : $\mathsf{bucket}'_{@}(y_{@})](Q_{@}(y))$, i.e., the term referred to by the tag $\boxed{10}$ in (538). The second relational call in (538) guarantees that the term $\boxed{10}$ is a subterm of the term $\boxed{9}$. Remember in LF-Ty2, one term is a subterm of the second, in case it is a component of the second. We can, therefore, use the relation `component` to express subtermhood.

Because of the fact that the term $\boxed{10}$ is enforced to be a subterm of the term $\boxed{9}$ and because of the way we have defined the relation `replace` in Chapter 4, the term $\boxed{11}$ must be a subterm of $\boxed{8}$. The relation `replace` holds of a quadruple of *me* objects $\langle x, v, y, w\rangle$ if and only if x is just like y, except that it contains the term v as a subterm where y has the subterm w. We can illustrate this with the sentence in (532b). In (536) we saw the CONTENT value of the NP *the social bucket*. We repeat this logical form in (539), showing that it is exactly of the form required in (538).

(539) a. The logical form of the NP *the social bucket*:
$\lambda Q.\mathsf{socially}'_{@}(\lambda @.[\mathsf{the}\ y : \mathsf{bucket}'_{@}(y_{@})](Q_{@}(y)))$

b. ...in AVM form:

$$\begin{bmatrix} \textit{abstr} & \\ \text{VAR} & Q \\ \text{ARG} & \boxed{9}\begin{bmatrix} \textit{appl} & \\ \text{FUNC} & \textit{socially} \\ \text{ARG} & \begin{bmatrix} \textit{abstr} & \\ \text{VAR} & @ \\ \text{ARG} & \boxed{10}\ [\mathsf{the}\ y : \mathsf{bucket}'_{@}(y_{@})](Q_{@}(y)) \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Similar to this, we can give the CONTENT value of the VP in the case of the example sentence in (532b). In (540), we first give a term notation in (a) of the logical form of the VP, followed by an AVM notation in (b) which includes the tags used in (538).

(540) a. The logical form of the VP *kick the social bucket*:
$\lambda x.\mathsf{socially}'_{@}(\mathsf{die}'_{@}(x_{@}))$

b. ...in AVM notation:

$$\begin{bmatrix} \textit{abstr} & \\ \text{VAR} & x \\ \text{ARG} & \boxed{8}\begin{bmatrix} \textit{appl} & \\ \text{FUNC} & \textit{socially} \\ \text{ARG} & \begin{bmatrix} \textit{abstr} & \\ \text{VAR} & @ \\ \text{ARG} & \boxed{11}\ \mathsf{die}'_{@}(x_{@}) \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

It is easy to see that the terms referred to by the tags $\boxed{8}$, $\boxed{11}$, $\boxed{9}$ and $\boxed{10}$ stand in the relation `replace`: The terms $\boxed{8}$ and $\boxed{9}$ are of the same species, *application*. Their FUNCTOR values are of the same species as well, *socially*. The ARG value of $\boxed{8}$ is the intension of the term $\boxed{11}$. As this is the subterm that is replaced, the ARG value of $\boxed{9}$ is correctly given as the intension of the term $\boxed{10}$.[20]

It should be noted that the new version of the description of the IE *kick the bucket* does not influence the ban on semantically internal modification: the term referred to with the tag $\boxed{10}$ still specifies that there may not be any semantic contribution occurring in the restriction of the quantifier except of the contribution of the noun *bucket*.

[20]If there is no modifier occurring inside the NP, then, trivially the terms referred to by the tags $\boxed{9}$ and $\boxed{10}$ are identical, and, by virtue of the definition of the relation `replace`, the terms referred to by the tags $\boxed{8}$ and $\boxed{11}$ are also identical.

Let us summarize the treatment of modification. We have excluded semantically internal modification in the case of the IE *kick the bucket* by imposing a very strict logical form on the NP. In the case of semantically external modification, we have first shown how this phenomenon could be handled within the framework of LF-Ty2 in general. We have, then, formulated the restrictions on the logical form of the NP in the IE *kick the bucket* so that external modifiers are allowed. Furthermore, the specification of the CONTENT value of the idiomatic VP is such that any external modifier to the NP will apply as modifier to the VP.

This approach to semantically external modification is very similar to the solution proposed in Abeillé 1995: We must assume that there are certain adjectives which are functors of a higher type as they take a quantifier as their argument. Second, an internally irregular IE which allows for an external interpretation of an adjectival modifier must specify this possibility explicitly. In TAG this was done by a link between the N node in the syntactic structure and the F node in the semantic structure. In our case this is achieved by allowing external modifiers to appear in the logical form of the VP, whereas internal modifiers are excluded. However, our proposal differs from Abeillé's, in the case of free combinations, the external modification reading comes for free, whereas such reading would require far more encoding in Abeillé's analysis.[21]

[21]Note that in Abeillé's analysis, the necessary link goes from the N node to the F node. In the case of an elementary tree for transitive verbs, however, there would just be an NP substitution node. Thus, in TAG, one is forced to postulate new elementary trees for every possible case of external modification.

CHAPTER 9

Conclusion

In Part II, we provided an account of idiomatic expressions in HPSG. We consider it important to have such an account in order to show that the semantic framework developed in the first part of the thesis, and especially in Chapter 4, is empirically robust. The framework of LF-Ty2 is robust in the sense that notorious counterexamples against a regular combinatorial combination of logical forms are captured by the newly introduced mechanism, whose core element is the attribute COLL. In the present concluding chapter to the second part of this thesis, we will address several issues that enable us to put our analysis in Chapter 8 into a broader perspective. First, we will compare the analysis with the proposals presented in Chapter 7. This discussion will take into account only the analyses of IEs in the different frameworks. Second, we will demonstrate that the COLL feature can equally well be used for other linguistic phenomena. We will sketch how, in principle, constructions other than IEs can be captured in a way parallel to the treatment of internally irregular IEs in Section 8.1. Furthermore, we address some other words that show idiosyncratic distributional properties which might be accounted for with COLL specifications in the lexical entry instead of with general principles. As an example of such words, we will come back to the case of anaphora which was already mentioned in the introduction.

9.1. Comparison to the Other Approaches

In the preceding chapters, we have already tried to indicate what similarities and differences we see between the approach developed in Chapter 8 and the approaches referred in Chapter 7. In this section we will compile these remarks to a coherent overall picture.

The basis of our approach is the distinction between IEs that are internally regular and those that are internally irregular. The empirical foundation of this distinction goes back to Wasow et al. 1983 and served as the basis for the approach in Gazdar et al. 1985 as well. While the analysis of Gazdar et al. 1985 is silent with respect to the treatment of internally irregular IEs, internally regular IEs are analyzed as regular combinations of words that underlie idiosyncratic distributional restrictions. In our discussion of the GPSG analysis, we pointed to three problems: (i) the account cannot differentiate between the IEs *spill the beans* and *pull strings*. (ii) the partial function approach to semantic interpretation causes very complicated restrictions on the interpretation of the semantic constants, which, so far, have not been fully spelled out in detail. And (iii), in GPSG, there is no notion of a syntactically complex lexical item, which is needed to account for internally irregular IEs.

As far as the first problem is concerned, we must keep in mind that the semantic analysis of extraction is quite similar in GPSG and in our proposal. In both cases, the semantics of the extracted constituent is introduced into the logical form of the sentence only at the node that dominates the filler. Still, once the semantic contribution of the filler has been introduced, the overall logical form of a sentence with extraction does not differ from a sentence without a topicalization structure. In the case of GPSG, this fact caused the problem that a purely interpretive account of the distributional restrictions of lexical elements cannot make reference to a particular syntactic configuration. In our approach, on the other hand, we included a syntactic component into the mechanism for distributional restrictions. As shown in the lexical entry of the idiomatic word *spill* in (502), we can impose the restriction that the particular distributional requirements must be met at some syntactic node, i.e., the minimal clause in the case of the verb *spill*. This node does not include the filler, and, thus, does not contain the filler's semantic contribution in its logical form.

Let us next turn to the second point of criticism. It would, in principle, have been possible for us to simply adopt the model-theoretic assumptions of Gazdar et al. 1985 as far as the interpretation of semantic constants is concerned. Under such an analysis, however, we would not only have encountered the same problem as Gazdar et al. 1985 regarding the fine-grained differences among IEs, we would further have been led to the assumption that the mutual restrictions of parts of an IE should not be captured within the HPSG grammar. This claim needs some explanation.

The semantic framework of LF-Ty2, as presented in Part I, provides terms of Ty2 as CONTENT values of signs. Principles such as the SEMANTICS PRINCIPLE enable us to combine the semantic contributions of the daughters in a phrase to form the semantic contribution of the phrase. The CONTENT values are, however, terms of the semantic representation language, not the interpretation of these terms. Put differently, the CONTENT values of signs are conceived as logical forms of the signs, not as their semantic interpretation.

In Chapter I we defined the function $\{\![\ ]\!\}$ which assigns *me* objects a model theoretic interpretation. It is doubtlessly the case that there are some semantic relations that hold between the model theoretic interpretation of semantic constants, such as antonymy, hyponymy etc. These relations are part of the world knowledge and, as such, part of the restrictions on the way we can possibly interpret semantic constants. Within the HPSG grammar, we have no way to impose a restriction on the actual semantic interpretation of *constant* objects. The logical forms are all we can restrict, i.e., the terms which can occur as CONTENT values of signs under which conditions.

In our perspective, the fact that the idiomatic use of *beans* always has to co-occur with the idiomatic use of *spill* is not a restriction on the possible semantic models. Instead, it should be captured as a restriction on the distribution of the two idiomatic words. The way we have expressed this distributional requirement is as a restriction on the possible logical forms of sentences that contain an idiomatic word.

It should be noted that the logical form treatment of the distribution of idiomatic words makes the restrictions on the semantic interpretation of the constants spill'' and beans'' superfluous, i.e., these constants might even be real synonyms to the constants $\mathsf{divulge}'$ and secret'. This leads to an architecture with fewer limitations on the possible semantic models. While we consider this already a strong advantage of our system, the real difference between the two approaches can be seen once we take into account the further consequences of the partial function analysis as sketched in the criticism of Pulman 1993, which we have outlined in Section 7.1. Pulman shows that the approach of Gazdar et al. 1985 has as its consequence a duplication of semantic constants. As an example, the definite article in the IE *spill the beans* must be a special semantic constant, different from the normal definite article, but also different from the definite article which occurs in the case of other IEs such as *bite the bullet*, *break the ice*, etc. Such a duplication is avoided under our logical form account. It follows that the logical form treatment of the distributional restrictions leads to a smaller inventory of semantic constants. Specifically, special semantic constants are only necessary for the words whose distribution we must constrain, not for all the constants that can co-occur with these constants in the same clause.

Finally, we can address the third criticism to the GPSG account. In our analysis of internally irregular IEs we introduced phrases which are exempt from all principles of regular combination. These phrases are identifiable by their COLL specification and they must be listed in the lexicon. Normally, the components of such irregular phrases are themselves regular, and the irregular phrases are not fully fixed with respect to their internal structure. This was illustrated for the internally irregular VP *kick the bucket* in Section 8.1 and Section 8.4. There, we saw that the NP may contain free syntactic material and that the semantics of this material re-appears in the semantics of the VP as a whole. Due to the fact that the syntactic rules of licensing in GPSG work strictly on local trees, it is very difficult to envisage an extension of GPSG that would actually allow for syntactically complex lexical elements. In Section 8.3 we unified our treatment of the COLL feature to arrive at a lexicon which comprises both the lexical entries for words and the phrasal lexical entries for internally irregular IEs.

We conclude that our approach avoids the problematic properties of the GPSG analysis, while it maintains its major empirical and analytical insights. In particular, we are strongly influenced by the distinction between internally regular and internally irregular IEs and the insight that some mechanism is needed to account for idiosyncratic distributional properties.

The analysis of Abeillé 1995 expressed within the framework of TAG provided a very natural account of internally irregular IEs such as *kick the bucket*. The reason why it is easy to incorporate this type of IEs into a TAG grammar comes from the notion of locality that TAG assumes: There, the smallest structural entities in grammar are not nodes or local trees, but trees that are complete with respect to all obligatory dependencies, i.e., trees that contain nodes for all arguments and, if necessary, for semantically vacuous material or fillers. In HPSG, however, every single linguistic object must satisfy all principles of the grammar. Therefore, we cannot consider complex syntactic structures as basic entities of the language, as it is done in TAG. In our account we managed to include complex lexical entries into our grammar, while still preserving the stronger locality notion of HPSG. This was achieved by introducing a special kind of phrases which trivially satisfy all the principles of regular combination. As a result, we allow for individual non-terminal nodes in the structure of a sentence whose internal properties are not predictable from those of its parts, but which are licensed by a phrasal lexical entry. Still, these nodes being phrasal, they dominate some structure on whose form the phrasal lexical entry may impose constraints.

Another difference between the TAG approach and our HPSG analysis lies in the way we encode the syntactic flexibility of an IE. In TAG, the syntactic flexibility is a consequence of the number of trees present in the Tree family of the particular IE. In the HPSG account, we encoded the syntactic flexibility of an IE indirectly. First, we used internal properties of the elements that constitute an IE. For internally irregular IEs such as *kick the bucket*, we specified in the phrasal lexical entry that this IE always be a VP with the NP *the bucket* being realized as the direct object. This automatically excluded the option of extracting the NP or realizing it as the syntactic subject, as in both cases there would not be a direct object NP node with the required phonology.

In the case of the syntactically more flexible, internally regular IEs, we have used certain locality restrictions within which all elements that constitute the IE must be found. An example of this would be that for restrictive speakers, we have assumed that the contextual requirements of the verb *spill* in the IE *spill the beans* must be satisfied within the minimal clause of the verb. For more flexible IEs, or more permissive speakers, these requirements must only be satisfied in some larger syntactic domain. This domain restriction has the effect of excluding certain syntactic configurations, such as extraction, because in extraction, the filler is realized outside the minimal clause of the verb.

To sum up, the TAG approach of Abeillé 1995 and our approach both seem to be flexible enough to account for the great diversity found in the empirical area of IEs. The differences between the analyses are due to fundamental differences between the grammatical frameworks.

Finally, we shall compare our analysis to the approach of Riehemann 1997. Again, many differences simply lie in the fact that Riehemann assumes a constructional approach to HPSG, whereas we do not. In addition, while Riehemann collects all the properties of a syntactically complex IE locally in a phrase (in the WORDS set and the LISZT list), we allow the use relations which refer to elements deeper in the syntactic and the semantic structure.

The main architectural change proposed in Chapter 8 is that signs may have an attribute whose value contains some larger sign which dominates the sign under consideration. We used this attribute, COLL, to indicate that a sign is licensed by some lexical entry (phrasal or not) and to specify the idiosyncratic distributional restrictions of the particular sign. In Riehemann 1997, no mechanism is assumed that would allow encoding these distributional properties. Instead, the approach assumes that the necessary restrictions can be incorporated into the construction that contains elements with idiosyncratic distributional properties. As we showed, however, having constructions that allow for the occurrence of distributionally constrained elements is not sufficient to prevent these elements from occurring in other constructions.

What is particularly remarkable in the contrast between our approach and that of Riehemann 1997 is that for each IE, Riehemann assumes a special subsort below *phrase* but still needs idiomatic words for the crucial elements that constitute the IE. In our approach, we either have a phrasal lexical entry for the IE as a whole, as is the case with *kick the bucket*, or we have lexical entries for the parts of the IE. In the latter case, however, the IE as a whole comes to existence by the combination of its parts. Thus, the number of idiomatic entities assumed in the grammar is smaller in our approach than it is in Riehemann 1997.

In the following subsection, we will briefly address the general role that the COLL feature can play in the architecture of grammar. In particular we show that phrasal lexical entries can be used to encode constructions in general, and that there are more lexical elements that show distributional irregularities such as those we found in the case of the elements that constitute internally regular IEs.

9.2. The Role of the COLL Feature

In this section, we will emphasize the fact that IEs are not the only elements in the language that show internal and external irregularity. As such, as such we can assume that *constructions* in general are instances of internally irregular phrases. Furthermore, words such as anaphora or personal pronouns can equally well be considered elements that exhibit idiosyncratic distributional properties. We will not go into these issues in great detail, but we would like to outline what picture of grammar emerges once the COLL feature is taken into consideration.

Other Internally Irregular Phrases. In Section 8.1 we presented our use of the COLL feature to allow for phrases which are exempt from the principles of regular syntactic and semantic combination. Doing this, we opened the door for the treatment of irregularities at the phrasal level. It is natural that the treatment of internally irregular IEs is just the first step towards a broader investigation of constructions in general. In fact, one of the contributions of this work is to incorporate phrases with internal idiosyncrasies into the architecture of HPSG, by extending the lexicon to contain phrasal lexical entries as well as lexical entries for non-derived words. Something similar has already been done, in principle, in constructional HPSG (Sag, 1997; Riehemann, 1997). However, it has never been proposed under a non-constructional perspective on HPSG.

Given this new architecture, we should ask what would be the analogue of *constructions*. In (541) we give such a characterization.

(541) A *construction* is a phrase whose internal properties cannot be derived by combining its daughters in a regular way (i.e., a phrase with [COLL *nelist*]).

As this characterization indicates, we need a notion of "regularity" before we can identify exceptions. This means that it is, ultimately, a theory-internal distinction that should be made between internally regular and internally irregular phrases, i.e., constructions. There are clear cases of constructions, such as the internally irregular IEs explored in this part of the thesis. For other phenomena, the classification is not as straightforward. Consider, for example, the treatment of relative clauses in Pollard and Sag 1994 and Sag 1997. In Pollard and Sag 1994, an analysis is assumed that relies on the existence of a phonologically empty relativizer. Using this special word, all properties of a relative clause can be derived by means of regular combination. In contrast to this Sag 1997 does not use a phonologically empty functional head for relative clauses, but assumes special phrases, i.e., what he calls constructions, which may have properties that differ from those of other phrases. To name just one of these properties, the relative clause is a verbal projection with a noun in its MOD value, whereas regular verbal projections do not have a *synsem* object as their MOD value.

It is beyond our objectives to take a position in the debate whether there is empirical motivation for an empty relativizer, or for verbs with non-trivial MOD values in English. Our goal is merely to emphasize that we have provided a formal integration of internally irregular phrases into non-constructional HPSG. Thus, the linguist can express his or her insights or intuitions about the distinction between regular and irregular aspects of language in a formally explicit way. This has the important advantage that we are not forced to adopt a particular analysis, such as the assumption of an empty relativizer which was unavoidable within the theory of Pollard and Sag 1994. Instead, we can simply declare certain kinds of phrases exceptional and define their properties in terms of phrasal lexical entries.

The reader should also be warned that our analogue to constructions as characterized in (541) will not identify the same entities that are called *construction* in *Construction Grammar* (Kay, 1997). The case of IEs illustrates this. In Construction Grammar, like in TAG and constructional HPSG, internally regular IEs as *spill the beans* are treated as units, i.e., as a construction. In our approach, following GPSG, we located the irregularity of this IE in the distributional restrictions on the words *spill* and *beans*. This difference is due to the fact that in Construction Grammar, just as in TAG and constructional HPSG, there is no theory of external irregularities, i.e., of distributional idiosyncrasies.

We should finally expand on another distinction between the present approach to constructions and the one taken in constructional HPSG. As we have seen in the treatment of the IE *kick the bucket* in Riehemann 1997, Copestake et al. 1997 introduce an attribute CXCONT (CONSTRUCTIONAL-CONTENT). The semantics principle is given so that the LISZT value of a phrase contains exactly the elements in the LISZT values of the daughters and those in the CXCONT LISZT value of the mother. This means that constructions can only add semantic content but not remove relations that are present in the daughters. The consequences of this can be seen in the treatment of the IE *kick the bucket*. In the description of the *kick-bucket-idiom-phrase* in (460) on page 314, Riehemann 1997 assumes idiomatic words *kick*, *the* and *bucket* which make an empty semantic contribution. In our own analysis, these words occur with their regular semantics as parts of the irregular phrase *kick the bucket*, and it is just one of the signs of internal irregularity of this phrase that it does not integrate the semantic contributions of its daughters into its own logical form.

To conclude, the introduction of the COLL feature creates the necessary architecture for an analysis of special constructions within non-constructional HPSG. Such a non-constructional HPSG perspective on phenomena treated as constructions will certainly reveal interesting insights because we will have to address questions which are not in the focus within the other frameworks, namely: what are the minimal parts of a particular "construction" for which we must assume internal irregularity? What are the distributional requirements of these parts of the "construction"? These questions were asked for idiomatic expressions in the framework of GPSG for example, but, to our knowledge, they have not been addressed in other cases of constructions.

Other Words with Idiosyncratic Distributional Properties. In the section on internally regular IEs such as *spill the beans*, we introduced a formalization of distributional restrictions on non-derived words. In Section 8.3, we argued that there are also some internally irregular phrases that show idiosyncratic distributional properties. Finally, we have presented an architecture where, in principle, every sign that is licensed by some lexical entry (phrasal or not) can, in principle, impose idiosyncratic distributional requirements. We think that there are at least some prominent cases of signs which have such a idiosyncratic distribution. If we consider the grammar of Pollard and Sag 1994, we can take the following two cases: traces and signs that are subject to binding conditions.

Since traces are the simpler case, we consider them first. Remember that, contrary to our analysis, the grammar in the appendix of Pollard and Sag 1994 assumes an analysis of complement extraction which contains traces. A trace is a sign that has an empty PHON value and a single element in its INHERITED SLASH set which is identical to the LOC value of the trace. Assuming such a sign, there must be a mechanism to prevent it from freely occurring in the language. In Pollard and Sag 1994 this is done in the TRACE PRINCIPLE, which is repeated in (542).

(542) The TRACE PRINCIPLE (Pollard and Sag, 1994, p. 400):

> The SYNSEM value of any trace must be a (noninitial) member of the SUBCAT list of a substantive word.

The intention of this principle is to enforce that traces only occur as complement daughters. As pointed out in Richter et al. 1999, such a principle must be interpreted as a restriction on the structures that may contain a trace. As it is an idiosyncratic distributional property of some lexical element, it is most naturally re-expressed as a restriction on the COLL value of a trace inside the trace's lexical entry. In (543), we give the revised lexical entry for the trace, which incorporates the effect of the TRACE PRINCIPLE.

(543) Lexical entry for the trace (including the TRACE PRINCIPLE):

$$\mathbf{E}\boxed{1}\ \mathbf{E}\boxed{2}\ \mathbf{E}\boxed{3}\ \mathbf{E}\boxed{4}$$
$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle\rangle \\ \text{SYNS} & \boxed{1}\begin{bmatrix}\text{LOC} & \boxed{2} \\ \text{NONLOC} & \begin{bmatrix} \text{INHERITED} & \begin{bmatrix}\text{SLASH} & \{\boxed{2}\} \\ \text{QUE} & \textit{eset} \\ \text{REL} & \textit{eset}\end{bmatrix} \\ \text{TO-BIND} & \begin{bmatrix}\text{SLASH} & \textit{eset}\\ \text{QUE} & \textit{eset} \\ \text{REL} & \textit{eset}\end{bmatrix}\end{bmatrix}\end{bmatrix} \\ \text{COLL} & \langle\boxed{3}\rangle \end{bmatrix}$$
$$\textbf{and}\ \ \texttt{dominate}(\boxed{3},\boxed{4})$$
$$\textbf{and}\ \ \boxed{4}\begin{bmatrix}\textit{word} & \\ \text{SYNS LOC CAT} & \begin{bmatrix}\text{HEAD} & \textit{substantive} \\ \text{SBC} & \left\langle \begin{bmatrix}\textit{synsem}\end{bmatrix}, \ldots, \boxed{1}, \ldots \right\rangle\end{bmatrix}\end{bmatrix}$$

The first AVM in the lexical entry of the trace contains exactly the information given in the lexical entry for the trace in Pollard and Sag 1994: The PHON value is an empty list and the LOCAL value of the trace ($\boxed{2}$) is identical to the single element in the INHERITED SLASH set, while all other nonlocal values are empty. In (543) we have augmented this lexical entry with the COLL specification. The conditions on the COLL value ($\boxed{3}$) encode the TRACE PRINCIPLE: every trace must have a COLL value of a certain form. Due to the COLL PRINCIPLE, this means that every trace must occur in a structure that has certain properties. In the case of traces, the structure must be such that it dominates some sign $\boxed{4}$ such that $\boxed{4}$ is (i) a word, (ii) has a HEAD value of sort *substantive*, and (iii) contains the SYNSEM value of the trace ($\boxed{1}$) as a non-initial member on its SUBCAT list.

Other words which have idiosyncratic distributional properties are pronouns or anaphora. These elements are, however, identifiable via a certain CONTENT value in the feature geometry of Pollard and Sag 1994, i.e., anaphora have a CONTENT of sort *anaphor*, for pronouns it is of sort *personal-pronoun*, and for referential expressions it is of sort *nonpronoun*. Binding Theory expresses a generalization of the contexts in which these three classes of elements may occur. Again, as pointed out in Richter et al. 1999, the principles of Binding Theory are constraints on the structures that contain signs with certain CONTENT values.

In our new architecture, we can express the conditions of Binding Theory as distributional properties. For illustration, consider PRINCIPLE B of Binding Theory in Pollard and Sag 1994 (p. 401), repeated in (544).

(544) PRINCIPLE B:

A personal pronoun must be locally o-free.

The details of the definitions of the HPSG Binding Theory are not important in this context. We simply assume that an element is locally o-free iff there is no element in the structure that locally o-binds it.[1] We can, therefore, restate PRINCIPLE B as a restriction on the COLL element of all words that are personal pronouns.

(545) A formalization of PRINCIPLE B:

$$\begin{bmatrix} \textit{word} \\ \text{SYNS} \begin{bmatrix}\text{LOC CONT } \textit{personal-pronoun}\end{bmatrix} \\ \text{COLL } \textit{nelist}\end{bmatrix} \Rightarrow \left(\mathbf{E}\boxed{1}\ \mathbf{E}\boxed{2} \left(\begin{bmatrix} \textit{word}\\ \text{SYNS } \boxed{1} \\ \text{COLL } \langle \boxed{2} \rangle \end{bmatrix} \textbf{ and not } \mathbf{E}\boxed{3} \left(\begin{array}{l} \texttt{component}(\boxed{3},\boxed{2}) \\ \textbf{and } \texttt{locally-o-bind}(\boxed{3},\boxed{1}) \end{array}\right)\right)\right)$$

In this formalization of PRINCIPLE B, it is obvious that the principle is a constraint on the distribution of personal pronouns: the antecedent of the principle is satisfied by all (non-derived) words whose CONTENT value is of sort *personal-pronoun*.[2] For these words ($\boxed{1}$), it is then required in the consequent of the principle that they occur in a context in which it is true for every sign $\boxed{3}$ that occurs in this context ($\boxed{2}$) as well, that $\boxed{3}$ does not locally bind the personal pronoun.

This reformulation of PRINCIPLE B is slightly different from the way we have integrated the TRACE PRINCIPLE into the grammar. In the case of the TRACE PRINCIPLE, we have

[1]For an RSRL formalization of the Binding Theory of Pollard and Sag 1994, see Richter 2000 (pp. 252–258).

[2]Note that this captures the fact that in the grammar of Pollard and Sag 1994 case marking prepositions that occur with a pronoun as their complement, such as the PP *to her* in (i), are considered pronouns with respect to Binding Theory.

(i) Mary$_i$ didn't talk to her$_{*i,j}$ anymore.

simply included the distributional restriction of the trace in its lexical entry. In the case of Principle B, we have conserved the fact that it is a generalization. It is, however, a generalization about the distributional requirements of some lexical elements. Alternatively, of course, we could have stated the principle explicitly as part of every lexical entry of a personal pronoun. What we want to illustrate with the encoding chosen in (545) is, however, that we can easily generalize distributional restrictions for larger classes of words.

We want to conclude this discussion with the observation that even though the grammar of Pollard and Sag 1994 does not address idiomatic expressions, it still needs to give distributional restrictions for certain words. As the grammar is not given in a fully formalized way, we can only guess what mechanisms the authors have in mind to express these constraints. In Richter et al. 1999, it was shown that the constraints are restrictions on the structures in which particular words may occur. Still, the distributional restrictions are presented in the book as if they were properties of these words. In our new architecture, assuming the COLL feature, we can formalize these distributional restrictions in a way that shows that they are both, idiosyncratic properties of the particular words, and restrictions on the structures which may contain these words.

9.3. Compositionality?

In Part II we complemented the analysis of regular combinations from Part I with an account of irregularity phenomena as attested in the domain of idiomatic expressions. We presented the data on idiomatic expressions so that there are two classes of problems for the grammar of Part I. First, we considered the case of IEs such as *kick the bucket* that seem to show that the meaning of a complex expression is not related to the meaning of its parts. Second, we considered IEs such as *spill the beans* for which we can assume a regular combination of the meanings of their parts, but under such an assumption, the system presented in Part I cannot capture the distributional restrictions of parts of the IE.

In Part I we were only concerned with the regular combination of smaller signs to form larger signs. We presented a semantic architecture which has the effect of also combining the semantic representations of the smaller signs in a regular way to arrive at the logical form of the larger sign. We chose a semantic framework which was based on flexible type shifting operations as those introduced in Hendriks 1993 combined with Ty2 (Gallin, 1975) as the underlying semantic representation language.

Idiomatic expressions constitute an extreme case of potential counterexamples to such a regular treatment of semantic combination. In the chapters of this part of the thesis, we encountered two basic questions. The first question arises from the observations made about internally irregular IEs: How can it be that the VP *kick the bucket* means something like *die*? The second critical question for our architecture of regular combination is concerned with internally regular IEs: Why is it that the noun *beans* means something like *secret* only in combination with the verb *spill* (in the meaning of *divulge*)?

Both of these questions received an answer in Chapter 8. The answer given there located the idiosyncrasies in both cases in the lexicon. This had the effect of leaving our architecture of combinatorial semantics untouched. We argued that in the case of *kick the bucket*, we come across internally irregular properties. To account for them, we assumed that the VP is not the result of regular combination of smaller signs, but that it is directly licensed by some (phrasal) lexical entry. As such, the VP is naturally freed from being forced to obey the regular principles of semantic combination. We used the COLL feature to identify lexical entities such as this IE.

Thus, we solved the first problem by introducing phrasal lexical entities: Semantic combination is only regular for non-lexical signs. This assumption is in fact very natural.

It is, however, not in all cases obvious to decide what should count as a lexical sign and what should not. As an analogy, consider the theory of formal languages. In any formal language, the basic entities of combination, usually called the members of the *alphabet*, are not assumed to be internally structured, i.e., to obey the construction rules of the formal language. Similarly for the non-recursive cases of signs, i.e., non-derived words, it would be unmotivated to postulate a predictable form-meaning relation. What is new in our architecture is that we assume the existence of elements that are recursively built, i.e., which are composed out of smaller bits of the same data structure, but which must still be considered lexical in the sense of being idiosyncratic. To come back to our analogy with formal languages, we are in a position where we assume an alphabet which contains not only elements such as a and b, but also elements of the form ab.[3] For the purposes of semantic and syntactic combination the internal structure of the basic entities is of no importance. Thus, in the light of the analysis in Chapter 8, we claim that simple principles of regular combination hold for all elements in the language with the exception of elements that are directly licensed by some (possibly phrasal) lexical entry.

This re-adjustment of what should be considered the basic entities of regular combination, provides an answer to our first question. Still, we must address the second problem mentioned above. In our discussion of internally regular IEs we observed that the distribution of some lexical entities cannot be predicted on the basis of the distribution of other entities that are of the same syntactic category or that have the same meaning. As an example, we saw that the distribution of the noun *beans* in the meaning of *secret* is neither parallel to the distribution of the noun *beans* in its regular meaning nor to the noun *secret*. In fact, the noun occurs with this particular semantics only in combination with the verb *spill* in the meaning of *divulge*. This distributional property of the noun *beans* is not shared by any other noun in the language. It was this observation that led us to the claim that distributional idiosyncrasies are lexical properties, i.e., they should not (and cannot) be captured anywhere else but in the lexicon.

In our analysis in Chapter 8 we proposed a relatively simple extension of the HPSG feature geometry: we introduced a single new attribute, COLL, whose value is a list with at most one sign on it. The COLL value of a sign specifies the possible contexts in which it can occur. If the COLL value is empty then this sign does not have any distributional restrictions. If it is non-empty, however, the sign can in principle have such restrictions. Technically, the COLL PRINCIPLE ensures that the occurrence restrictions imposed on the COLL value are actually met in every structure that contains a certain sign.

In the discussion of IEs, we showed that the distributional restrictions of signs can greatly vary: in the case of the idiomatic use of *beans* and *spill*, the restrictions concerned the logical form of some clause, in the case of the light verb *make*, the restriction required the presence of some particular word within the sentence. We gave examples of (non-derived) words that have idiosyncratic distributional properties, but also, in Section 8.3, of an internally irregular phrase which is distributed idiosyncratically.

All these entities, non-derived words and irregular phrases are treated as lexical in our architecture. Thus, we formulated the hypothesis that only lexical entities can, in principle, have idiosyncratic distributional properties. Again, we achieved a reduction of apparent problems for the assumption of strong regularity to idiosyncratic properties of lexical entities. As a consequence, signs that show distributional idiosyncrasies are combined in a regular way, but the constraints on their occurrence possibilities will exclude some combinations from occurring in the language. It is in this sense that we restored the assumption that complex signs are built in a regular way: To form phrases, there is a

[3]For analogy, consider the Spanish spelling system. While it uses symbols from the latin alphabet, until recently (1994), the multi-letter combinations *ch* and *ll* were considered basic entities for alphabetic ordering.

small number of syntactic possibilities of combination, the ID-SCHEMATA, and just two possibilities to combine the semantics (taking either the head daughter or the non-head daughter as the functor and the other as the argument). For derived words, Derivational Rules specify exactly what the syntactic and semantic properties of the output word are in relation to those of the input. For signs that are licensed by a lexical entry, on the other hand, we allow a high degree of internal and external idiosyncrasy.

The major difference between our approach and previous analysis is that we located all idiosyncrasy in the lexical entry, even the distributional idiosyncrasy. Among the approaches considered in Chapter 7, Abeillé 1995 and Riehemann 1997 did not acknowledge the existence of distributional idiosyncrasies. In Gazdar et al. 1985 the existence of distributional idiosyncrasies was accepted, but these idiosyncrasies were built into the model theoretic interpretation of semantic constants, and they were not part of the lexical entries (or the terminal local trees) of the words. In our approach, we did not impose any conditions on the way the non-logical constants are interpreted, but formulated the distributional constraints in terms of properties of the larger structures in which the words can occur.

To conclude this chapter, we briefly recall the motivation for choosing the particular name COLL for the attribute that is in the center of our analysis. As mentioned earlier, this name suggests two things: First, it can be seen as an abbreviation of *collocation*, because the phenomenon of idiomatic expressions is sometimes referred to as a collocational phenomenon. Second, and more correctly in the light of the role that we attribute to this feature, it is the acronym for *Context Of Lexical Licensing*. We showed that only lexical entities, i.e., non-derived words and irregular phrases, have a non-empty COLL list. This means that the COLL value indicates whether a sign is licensed by a lexical entry or not. In addition, in the case of a non-empty COLL list, the element on the list specifies the linguistic context in which the sign occurs. We tried to show in the last chapter and in Section 9.2 that the COLL feature is of central importance in the grammar, once irregularities are taken into account. As such, the COLL feature and the COLL PRINCIPLE is an irregularity module that can, in principle, be added to any HPSG grammar to make it capable of coping with irregularity as well.[4]

[4]This is true, at least, for any HPSG grammar that assumes the Derivational Rule approach of Meurers 2000. For other approaches to lexical generalizations, further adaptations of the COLL PRINCIPLE might be necessary.

Summary

In this thesis, we expanded the architecture of grammar of Pollard and Sag 1994 with a model-theoretic semantics and with an irregularity module. The first was located in the CONTENT value of a linguistic sign, the second in its COLL value. In this summary, we focus on two general points. First, we argue that the architecture for combinatorial semantics presented in Part I has several conceptual and practical advantages over the way combinatorial semantics is done in Pollard and Sag 1994. Second, we show how our use of the COLL feature introduced in Part II relates to the principles of compositionality and contextuality to which we referred in the introduction.

In Part I, we presented an interpretation of HPSG which is sign-based, in the sense that the signs in the denotation of a grammar contain both phonological and semantic information. We did not assume that a linguistic sign contained its *interpretation* directly as its CONTENT value. Instead, we assumed that the CONTENT value of a sign is a *representation* of its interpretation, i.e. a term of a semantic representation language. The introduction of such a *logical form* has a number of advantages over the semantic analysis found in Pollard and Sag 1994.

First, the terms are taken from a semantic representation that is familiar to most linguists working within formal linguistics, in contrast to the rather idiosyncratic data structure employed for semantic representations in Pollard and Sag (1994). While this is just a practical advantage, it has the consequence that the work done within the framework of HPSG semantics as presented in this thesis should be directly accessible to a large community of linguists.

Second, all parts that are relevant to the semantic representation are located in the CONTENT value. This increases the modularity of the grammar. Remember from Section 1.2 that in Pollard and Sag 1994 there are also the attributes QSTORE and RETRIEVED which are appropriate for the sort *sign* and which are needed to encode the particular storage mechanism. In later proposals (Pollard and Yoo, 1998; Przepiórkowski, 1998), it was shown that this distribution of semantic information over different parts of a sign leads to incorrect predictions and must be given up. Still, none of these proposals is as radical in including all semantic information in a single term as the present one.

Third, we took a clear position about the ontological status of the CONTENT value. This makes it possible to evaluate whether a logical form of this kind is really necessary, or whether it at least facilitates the account of natural language phenomena. The simple lf constraint introduced in (53) and integrated into our grammar in (291) is a typical example. The constraint accommodates the fact that the scope of a universal quantifier is clause bound, whereas an existential quantifier appears to be able to take scope outside the clause in which it is introduced. As we have argued at the end of Section 1.3, such a constraint can be formulated easily within the present framework, while it is not clear how to express it under the assumption of an LF along the lines of May 1985 or without any level of semantic representation as in Lappin 1991. The explicit formulation of such a constraint is, however, only possible if one has a clear concept of the way semantics is treated within the grammar.

Fourth, not only have we taken a clear position in the question of the ontological status of the CONTENT value, we have also chosen a concrete semantic representation language, Ty2. This choice was motivated by the facts that (i) every term of Intensional Logic can be translated into a term of Ty2 (Gallin, 1975), (ii) Ty2 is a technical improvement over Montague's *Intensional Logic* (Zimmermann, 1989), and (iii) Ty2 has been used in semantic work such as in Groenendijk and Stokhof 1982 and in work concerned with the syntax/semantics interface (von Stechow, 1993; Beck, 1996). While we fully acknowledge that Ty2 has its deficiencies, we have not yet encountered an alternative that would allow us to substitute it for Ty2 in the way Ty2 can replace Intensional Logic. The semantic representation language used in Pollard and Sag 1994 is left relatively vague. Therefore it is not possible to evaluate its strengths and weaknesses as precisely as for Ty2.

Fifth, which is actually a point related to the previous one, by choosing Ty2 as the semantic representation language, we automatically get the link between the logical form of a sign and its model theoretic meaning through the definition of the extension function for Ty2. This means that our HPSG grammars do not only have an interpretation as a set of linguistic objects, but they make empirically falsifiable predictions about meaning. We can, thus, test whether a certain semantic relation holds between two signs, based on the interpretation of their logical forms. As long as such a connection to "real" interpretation is not made, all claims about such relations are mere plausibility arguments. The descriptions of the CONTENT values of signs as they are given in Pollard and Sag 1994 are very suggestive, but for the lack of a clear relation to semantic interpretation, we are ultimately not able to verify whether the right meanings are accounted for.

Finally, our adaptation of a flexible system allows us to account for scope ambiguity without assuming syntactic movement (such as QR) or storage mechanisms. This flexible system is extremely appealing, because it is a simple and direct way to account for scope phenomena and, in combination with the fact that we use explicit semantic representations inside the grammar, the resulting terms can be easily constrained to exclude unavailable readings. The system of Pollard and Sag 1994 relies on a storage mechanism which is technically at least as complex as the flexible system. Furthermore, shifting operations have been proposed for other phenomena in natural language.[5] Thus, a flexible account only extends the application of techniques that have been proposed independently.

These six arguments in favor of our approach over that of Pollard and Sag 1994 might be considered unfair, because it was not the purpose of Pollard and Sag 1994 to make any definite decisions on the status and the structure of the CONTENT value. While the position of Pollard and Sag 1994 is justified, we hope to have shown that it is equally useful to explore a concrete proposal about the way semantics should be approached within HPSG. It was necessary to make some concrete assumptions before we developed a formal theory of idiomatic expressions because the licensing conditions of (the element of) idiomatic expressions rely on the existence of a Ty2-based logical form as part of every linguistic sign.

This brings us to the summary of the crucial points of our account of irregularity as it was illustrated with the analysis of idiomatic expressions in Chapter 8. Our discussion of the data in Chapter 6 confirmed the claim put forth in the introduction of this thesis that we must assume the existence of syntactically complex lexical elements, i.e., the existence of irregular phrases. To account for these, we have, ultimately, changed the definition of the lexicon. In the first part of the thesis, we assumed that the lexicon licenses exactly the non-derived words, i.e., the words that have an empty STORE value. In the new architecture, developed in Part II, the lexical elements are exactly those signs that have a non-empty COLL value. Since signs may be either words or phrases, it follows that we allow for both

[5]See Hendriks 1993 (pp. 43–56) for discussion.

kinds of signs to appear as lexical elements. The COLL value of a sign indicates its status as a lexical ([COLL *nelist*]) or as a non-lexical element ([COLL *elist*]). Therefore the COLL value is also directly relevant for the principle of compositionality.

(546) *Principle of Compositionality*

The meaning of a non-lexical element is a function of the meaning of its parts.

In the grammar of this thesis, we used the COLL value to implement this "principle": We changed all principles on phrases introduced in Part I in such a way that they only express non-trivial restriction on phrases with an empty COLL value. In particular, as far as the meaning of a sign is concerned, we assume that the meaning of a non-lexical element is either the intensional functional application of the meaning of the daughters, in the case of a regular phrase, or as specified in a Derivational Rule, in the case of a derived word. Thus, for all signs that have an empty COLL value, the logical form is a function of the logical forms of their parts.

In Part I we defined the lexical elements as those elements that have an empty STORE value. In Section 8.3 we proposed a new characterization of lexical elements in terms of the COLL value. In the COLL value, the contextual requirements of a sign are expressed. Following the principle of contextuality, we assume that, in principle, each lexical element has contextual requirements. These requirements should be observable, i.e., we can test whether a certain element occurs only in certain contexts. In this sense, the new characterization of lexical elements is more empirical than the old one: it is fully theory-internal whether a word is considered derived or non-derived, but, at least for some lexical elements, an investigation of their occurrence contexts delivers clear restrictions. In Chapter 6, we carried out exactly this kind of context investigation for the parts that constitute syntactically flexible idiomatic expressions. In addition, in Chapter 9 we showed that Binding Theory can be characterized as the study of such contextual requirements for particular words, pronouns and anaphora.

With the assumption of a non-empty COLL list for every lexical element and the COLL PRINCIPLE, we provided a direct implementation of the principle of contextuality.

(547) The *Principle of Contextuality*:

A lexical element can impose idiosyncratic restrictions on the linguistic contexts in which it may occur.

This formulation of the principle of contextuality raises the question of how many lexical elements there are in a language that do not have restrictions on their context of occurrence. The answer to this question certainly differs depending on the framework and the interest of investigation. If one includes, for example, socio-linguistic considerations, it may be possible to find a large number of contextual restrictions. Under such a perspective, it is likely that one will find contextual restrictions for almost every lexical element. If one stays within a single register of the language, the distributional restrictions on lexical items are different and, maybe, fewer.

We consider it inappropriate to evaluate the use of the principle of contextuality on the basis of the sheer quantity of lexical elements with contextual restrictions versus those for which there are no obvious restrictions attested. Therefore, we provide a theory in which every lexical element can, in principle, be contextually restricted, but need not be so. If a lexical element seems to underlie no co-occurrence restrictions, then the lexical entry that licenses such a non-restricted lexical element contains a maximally underspecified COLL specification ([COLL *nelist*]). In this sense, we can also see the difference between the "idiomatic" use of the word *spill* as it occurs in the IE *spill the beans* and the "normal" use of the word: We assume two lexical entries. The COLL specification in the lexical entry for the normal use is maximally underspecified, whereas the COLL specification of the idiomatic word is highly specific. Thus, by looking at the lexical entry, we can see how "regular" a lexical element is, and we expect to find a continuum of restrictiveness for lexical elements. Thus, while we assume that all lexical elements can express context requirements, in practice, they do so only to a certain degree. This degree can be seen from the COLL specification in the lexical entries.

For non-lexical elements, on the other hand, our architecture makes different predictions. Since we postulate that the COLL value is the empty list for all non-lexical elements, we have implemented the *Predictability Hypothesis* in the grammar (see (531)):

(548) The *Predictability Hypothesis*:

> For every sign whose internal properties are fully predictable, the distributional behavior of this sign is fully predictable as well.

This hypothesis is the mirror image of the principle of contextuality: whereas the principle of contextuality expresses the observation that lexical elements have idiosyncratic distributional properties, the predictability hypothesis says that for non-lexical elements there are no idiosyncratic contextual effects.

The predictability hypothesis is central for the present approach. It enables us to treat both aspects of irregularity studied here as lexical phenomena. In fact, we can use the availability of contextual restrictions as the defining criterion of a lexical element, i.e., with the introduction of a single new attribute, COLL, we capture both phenomena of irregularity. We are, thus, preserving the lexicalism of HPSG.

At least in principle, the predictability hypothesis is stated in such a way that we know what would constitute a potential counterexample: a sign that cannot be considered lexical on the basis of its internal properties, but which shows idiosyncratic occurrence restrictions. If such signs can be found, then we must seriously re-consider our architecture of grammar.

In this work, we addressed two fundamental issues of formal grammar: the relation between form and meaning and the relation between rules and exceptions. While these issues touch the basics of linguistic theory, we have not dealt with them in an abstract way. Instead, we chose a particular linguistic framework and implemented very specific assumptions. With HPSG, we chose a framework which is based on clear and precise mathematical foundations. Yet so far, none of these two issues received a treatment or an extensive discussion within this framework. With the present work, we hope to fill this gap to a certain extent. This means two things. For research within HPSG that is not concerned with any of these issues, the existence of the present work guarantees some empirical robustness of the framework. For research that explicitly addresses theses issues, our proposal is precise enough to raise interesting questions, to develop alternatives and to discuss their consequences in a constructive way.

APPENDIX A

Proofs and Additional Definitions

1.1. Proofs for Chapter 3

In this section of the appendix, we give the proofs of the lemmata and propositions of Sections 3.2–3.4. In the main text, we already highlighted the important parts of the proofs. For this reason, we will only repeat the lemmata and propositions and give their proofs without comment.

PROPOSITION 3.1
There is an exhaustive model $\mathsf{I}_{Ty2} = \langle \mathsf{U}_{Ty2}, \mathsf{S}_{Ty2}, \mathsf{A}_{Ty2}, \mathsf{R}_{Ty2} \rangle$ *of* $\mathcal{TY}2$ *such that* $\mathsf{U}_{Ty2} = \mathbb{N} \cup Type \cup Ty2$.

LEMMA 3.3
I_{Ty2} *is a model of the grammar* $\mathcal{TY}2$.

PROOF OF LEMMA 3.3

For each $i \in \mathbb{N}$,

> n is acyclic and finite by definition, thus the NP is satisfied.

For each $\tau \in Type$,

> τ is acyclic and finite by definition, thus is satisfies the TyNP and the TyFP.
> As $Type$ is a set, there is just one occurrence of each type. Thus, several occurrences of the same type within a larger type are necessarily identical. So, TyIP is satisfied.

For each $\phi \in$ Ty2,

> By definition, the elements of Ty2 are acyclic and finite (satisfying TNP and TFP).
> As Ty2 is a set, each term is contained only once. Thus, atomic terms trivially satisfy the TIP.
> Complex terms are always constructed form simpler terms which are also in Ty2. Therefore, if a term contains to subterms of identical shape, these are necessarily identical. Thus, the TIP is satisfied.
> For each $\tau \in Type$, for each $\phi_\tau \in$ Ty2,
> $\phi_\tau \in \mathsf{U}_{\text{Ty2}}$ with $\mathsf{S}_{\text{Ty2}}(\phi_\tau) \sqsubseteq me$,
> TYPE is defined on ϕ_τ and
> $\mathsf{S}_{\text{Ty2}}(\mathsf{A}_{\text{Ty2}}(\text{TYPE})(\phi_\tau)) \sqsubseteq type$.
> For each $i \in \mathbb{N}$ and for each $\tau \in Type$,
> $v_{\tau,i} \in \mathsf{U}_{\text{Ty2}}$.
> The appropriateness condition on the sort *var* is satisfied, as there are two attributes, TYPE and NUMBER with values of the appropriate sort.

For each $c_i \in Const$,

$c_{i,\mathcal{C}(c_i)} \in$ Ty2, therefore, $c_{i,\mathcal{C}(c_i)} \in \mathsf{U}_{\text{Ty2}}$.

$\mathcal{TY}2$ contains a type restriction on the constant $const_i$ which is only satisfiable by $const_i$ objects whose TYPE value is the *type* object $\mathcal{C}(c_i)$.

Let $\tau, \tau' \in Type$ and $\phi_{\langle\tau',\tau\rangle}, \phi'_{\tau'} \in \text{Ty2}_{\mathcal{C}}$ satisfy all constraints of $\mathcal{TY}2$, then $u = (\phi_{\langle\tau',\tau\rangle}\phi'_{\tau'}) \in \mathsf{U}_{\text{Ty2}}$. and

$\mathsf{S}_{\text{Ty2}}(u) = appl$,

$\mathsf{A}_{\text{Ty2}}(\text{FUNC})(u) = \phi_{\langle\tau',\tau\rangle}$ and $\mathsf{A}_{\text{Ty2}}(\text{ARG})(u) = \phi'_{\tau'}$.

u satisfies the appropriateness conditions and the TRP on *appl* objects.

Thus, by induction, u satisfies all constraints of $\mathcal{TY}2$.

Let $i \in \mathbb{N}$, and let $\tau, \tau' \in Type$ and $\phi_\tau \in$ Ty2, and $v_{\tau',i} \in Var$ satisfy all constraints of $\mathcal{TY}2$, then $u = (\lambda v_{\tau',i}.\phi_\tau)_{\tau'\tau} \in \mathsf{U}_{\text{Ty2}}$. and

$\mathsf{S}_{\text{Ty2}}(u) = abstr$,

$\mathsf{A}_{\text{Ty2}}(\text{VAR})(u) = v_{\tau',i}$ and $\mathsf{A}_{\text{Ty2}}(\text{ARG})(u) = \phi_\tau$.

u satisfies the appropriateness conditions and the TRP on *abstr* objects.

Thus, by induction, u satisfies all constraints of $\mathcal{TY}2$.

Let $\tau \in Type$ and $\phi_\tau, \phi'_\tau \in \text{Ty2}_{\mathcal{C}}$ satisfy all constraints of $\mathcal{TY}2$, then $u = (\phi_\tau = \psi_\tau)_t \in \mathsf{U}_{\text{Ty2}}$, and

$\mathsf{S}_{\text{Ty2}}(u) = equ$,

$\mathsf{A}_{\text{Ty2}}(\text{TYPE})(u) = t$,

$\mathsf{A}_{\text{Ty2}}(\text{ARG1})(u) = \phi_\tau$ and

$\mathsf{A}_{\text{Ty2}}(\text{ARG2})(u) = \phi'_\tau$.

u satisfies the appropriateness conditions.

As $\mathsf{A}_{\text{Ty2}}(\text{TYPE})(u) = t$ and

$\mathsf{A}_{\text{Ty2}}(\text{TYPE})(\mathsf{A}_{\text{Ty2}}(\text{ARG1})(u)) = \mathsf{A}_{\text{Ty2}}(\text{TYPE})(\mathsf{A}_{\text{Ty2}}(\text{ARG1})(u)) = \tau$,

the TRP is satisfied.

Thus, by induction, u satisfies all constraints of $\mathcal{TY}2$.

LEMMA 3.4

Let $I' = \langle U', S', A', R'\rangle$ be an interpretation of the signature of the grammar $\mathcal{TY}2$, then if I' is an exhaustive model of $\mathcal{TY}2$,

then for each $u' \in U'$,

there is an object $u \in \mathsf{U}_{Ty2}$ such that $\langle u', I'\rangle$ and $\langle u, \mathsf{I}_{Ty2}\rangle$ are congruent.

PROOF OF LEMMA 3.4

Let $I' = \langle U', S', A', R'\rangle$ be an exhaustive model of $\mathcal{TY}2$.

For each $u \in U'$, consider the configuration $\langle u, I'\rangle$,

if $S'(u) \sqsubseteq number$, then

$S'(u) = zero$.

In U_{Ty2}, there is just one *zero* object, o. as there are no attributes defined on the sort *entity*, $\langle u, I'\rangle$ and $\langle o, \mathsf{I}_{\text{Ty2}}\rangle$ are congruent.

$S'(u) = number$.

By hypothesis, there is an object $o \in \mathsf{U}_{\text{Ty2}}$ such that the configurations $\langle A'(\text{NUMBER})(u), I'\rangle$ and $\langle o, \mathsf{I}_{\text{Ty2}}\rangle$ are congruent.

By the definition of $\mathbb{N}$, then, there is also an object $o+1 \in \mathsf{U}_{\text{Ty2}}$. $o+1$ is of sort *non-zero* and its NUMBER value, $\langle A'(\text{NUMBER})(u), I'\rangle$ and

$\langle o, \mathsf{I}_{\mathrm{Ty2}} \rangle$ are congruent. Thus, $\langle u, I' \rangle$ and $\langle o+1, \mathsf{I}_{\mathrm{Ty2}} \rangle$ are congruent as well.

If $S'(u) \sqsubseteq type$, then

$S'(u) = entity$.

In $\mathsf{U}_{\mathrm{Ty2}}$, there is just an *entity* object, o. as there are no attributes defined on the sort *entity*, $\langle u, I' \rangle$ and $\langle o, \mathsf{I}_{\mathrm{Ty2}} \rangle$ are congruent.

Analogically for the other subsorts of *atomic-type*.

$S'(u) = complex\text{-}type$.

By hypothesis, there are objects $o', o'' \in \mathsf{U}_{\mathrm{Ty2}}$ such that the configurations $\langle A'(\textsc{in})(u), I' \rangle$ and $\langle o', \mathsf{I}_{\mathrm{Ty2}} \rangle$ are congruent and $\langle A'(\textsc{out})(u), I' \rangle$ and $\langle o'', \mathsf{I}_{\mathrm{Ty2}} \rangle$ are congruent.

The TyIP requires token identities within *type* objects wherever possible.

Thus, we know that there is an object $o \in \mathsf{U}_{\mathrm{Ty2}}$ which is of sort *complex-type* and has IN and OUT values which are congruent with $\langle A'(\textsc{in})(u), I' \rangle$ and $\langle A'(\textsc{out})(u), I' \rangle$ respectively.

Thus, this object and u are of the same species, have congruent components and, by the TyIP, all possible token identities among components must be there. Thus, $\langle u, I' \rangle$ and $\langle o, \mathsf{I}_{\mathrm{Ty2}} \rangle$ are congruent.

If $S'(u) \sqsubseteq me$, then

$S'(u) = var$.

By hypothesis, there are objects $i, t \in \mathsf{U}_{\mathrm{Ty2}}$ such that the configurations $\langle i, \mathsf{I}_{\mathrm{Ty2}} \rangle$ and $\langle A'(\textsc{number})(u), I' \rangle$ are congruent and $\langle t, \mathsf{I}_{\mathrm{Ty2}} \rangle$ and $\langle A'(\textsc{type})(u), I' \rangle$ are congruent.

For every type t and for every number i, there is a *var* object $v \in \mathsf{U}_{\mathrm{Ty2}}$ such that $\mathsf{A}_{\mathrm{Ty2}}(\textsc{type})(v) = t$ and $\mathsf{A}_{\mathrm{Ty2}}(\textsc{number})(v) = a$.

Thus, $\langle u, I' \rangle$ and $\langle v, \mathsf{I}_{\mathrm{Ty2}} \rangle$ are congruent as well.

$S'(u) = const_i$.

Due to the type restrictions on the species of *const*, there is only one possible type for each species. Thus, the congruence is derived analogically to the previous case.

$S'(u) = appl$.

By hypothesis, there are objects $o', o'' \in \mathsf{U}_{\mathrm{Ty2}}$ such that the configurations $\langle A'(\textsc{func})(u), I' \rangle$ and $\langle o', \mathsf{I}_{\mathrm{Ty2}} \rangle$ are congruent and $\langle A'(\textsc{arg})(u), I' \rangle$ and $\langle o'', \mathsf{I}_{\mathrm{Ty2}} \rangle$ are congruent.

The TIP on the sort *me* requires token identities within *me* objects wherever possible.

The definition of $\mathrm{Ty2}_{\mathcal{C}}$ guarantees the existence of an *appl* object with the indicated components, and the TIP ensures that the identities in u and in this object are the same.

Thus, $\langle u, I' \rangle$ and $\langle o, \mathsf{I}_{\mathrm{Ty2}} \rangle$ are congruent.

$S'(u) = abstr$.

By hypothesis, there are objects $o', o'' \in \mathsf{U}_{\mathrm{Ty2}}$ such that the configurations $\langle A'(\textsc{var})(u), I' \rangle$ and $\langle o', \mathsf{I}_{\mathrm{Ty2}} \rangle$ are congruent and $\langle A'(\textsc{arg})(u), I' \rangle$ and $\langle o'', \mathsf{I}_{\mathrm{Ty2}} \rangle$ are congruent.

The TIP on the sort *me* requires identities within *me* objects wherever possible.

The definition of Ty2 guarantees the existence of an *abstr* object o with the indicated components, and the TIP ensures that the identities in u and in this object are the same.

Thus, $\langle u, I' \rangle$ and $\langle o, \mathsf{I}_{\mathrm{Ty2}} \rangle$ are congruent.

$S'(u) = equ$.

By hypothesis, there are objects $o', o'' \in \mathsf{U}_{\mathrm{Ty2}}$ such that the configurations $\langle A'(\text{ARG1})(u), I' \rangle$ and $\langle o', \mathsf{I}_{\mathrm{Ty2}} \rangle$ are congruent and $\langle A'(\text{ARG2})(u) \rangle$ and $\langle o'', \mathsf{I}_{\mathrm{Ty2}} \rangle$ are congruent.

The TIP on the sort *me* requires identities within *me* objects wherever possible.

The definition of Ty2 guarantees the existence of an *equ* object with the indicated components, and the TIP ensures that the identities in u and in this object are the same.

Thus, $\langle u, I' \rangle$ and $\langle o, \mathsf{I}_{\mathrm{Ty2}} \rangle$ are congruent.

PROOF OF PROPOSITION 3.1

Let $\mathsf{I}_{\mathrm{Ty2}}$ be the interpretation of the grammar $\mathcal{TY}2$ as given in Definition 3.2, then

$\mathsf{I}_{\mathrm{Ty2}}$ is a model of $\mathcal{TY}2$ by Lemma 3.3, and

for each interpretation $I = \langle U, S, A, R \rangle$ and for each $\theta \subseteq \mathcal{D}_0$ if I is a model of $\mathcal{TY}2$, then

if $\Theta_I(\theta') \neq \emptyset$,

then there exists some $u \in U$ such that $u \in \Theta_I(\theta')$.

By Lemma 3.4, we know that in this case, there is also some $o \in \mathsf{U}_{\mathrm{Ty2}}$ such that $\langle u, I \rangle$ and $\langle o, \mathsf{I}_{\mathrm{Ty2}} \rangle$ are congruent.

As congruent configurations are always indiscernible, it follows that $o \in \Theta_{\mathsf{I}_{\mathrm{Ty2}}}(\theta')$, and, therefore, $\Theta_{\mathsf{I}_{\mathrm{Ty2}}}(\theta') \neq \emptyset$.

This means, that $\mathsf{I}_{\mathrm{Ty2}}$ is an exhaustive model of $\mathcal{TY}2$.

LEMMA 3.6

Let $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$ *be an exhaustive model of* $\mathcal{TY}2$,

let E *be the domain of individuals and*

let W *be a set of possible worlds, then*

for each $\tau \in Type$,

there is an indiscernibility class $[u] \subseteq \mathsf{U}$ *such that*

$D_{E,W,\tau} = D_{E,W,[u]}$,

and for each $u \in \mathsf{U}$, *with* $\mathsf{S}(u) \sqsubseteq type$,

there is a $\tau \in Type$ *such that*

$D_{E,W,\tau} = D_{E,W,[u]}$.

PROOF OF LEMMA 3.6

(by induction)

First part of the lemma:

Base

$D_{E,W,t} = \{0, 1\}$.

For $u \in \mathsf{U}$ with $\mathsf{S}(u) = truth$,

$D_{E,W,[u]} = \{0, 1\}$

analogously for $D_{E,W,e}$ and $D_{E,W,s}$.

Hypothesis

Assume that we have shown for each $\tau, \tau' \in Type$ that

there are $u, u' \in \mathsf{U}$ with $\mathsf{S}(u) \sqsubseteq type$ and $\mathsf{S}(u) \sqsubseteq type$, such that

$$D_{E,W,\tau} = D_{E,W,[u]} \text{ and } D_{E,W,\tau'} = D_{E,W,[u']}.$$

Step

Then, $D_{E,W,(\tau\tau')} = D_{E,W,\tau'}^{D_{E,W,\tau}}$.
By virtue of U being the universe of an exhaustive model of $\mathcal{TY}2$,
there exist some $u'' \in \mathsf{U}$ with $\mathsf{S}(u'') = \textit{c-type}$ such that $T_{\mathsf{I}}(:\text{IN})(u'') = u$ and $T_{\mathsf{I}}(:\text{OUT})(u'') = u'$.
By Definition 3.5, $D_{E,W,[u'']} = D_{E,W,[u']}^{D_{E,W,[u]}}$.
But, by application of the hypothesis, this is identical to
$D_{E,W,[u']}^{D_{E,W,\tau}} = D_{E,W,(\tau,\tau')}$.

Second part of the lemma:

analogously.

PROPOSITION 3.10
(Equivalence of Ty2 and exhaustive models of $\mathcal{TY}2$)

Let I be an arbitrary exhaustive model of $\mathcal{TY}2$.

Then, for each indiscernibility class of me objects

there is a Ty2 term which is assigned the same extension.

LEMMA 3.14
Given a frame F,

an exhaustive model $I = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$ of $\mathcal{TY}2$,

and corresponding interpretation functions int and Int, then

the function SR is such that
there are variable assignments a and A of Ty2 and I respectively, such that
for each $u \in \mathsf{U}$,
$\mathsf{S}(u) \sqsubseteq me$, *iff* $\{\![u]\!\}^{M,A} = [\![SR([u])]\!]^{M,a}$

PROOF OF LEMMA 3.14

Let *int* and *Int* be corresponding constant interpretations, and
let a and A be corresponding variable assignments.

The proof then goes via induction on the recursive structure of *me* objects.

Base

Take $u \in \mathsf{U}$ with $\mathsf{S}(u) = \textit{const}'$ for some species below *const*, then,
$\{\![u]\!\}^{M,A} = Int(\textit{const}')$ (by Definition 3.9)
$= int(C(\textit{const}'))$ (because *int* and *Int* are corresponding)
$= [\![C(\textit{const}')]\!]^{M,a}$ (by Definition 1.9)
$= [\![SR([u])]\!]^{M,a}$ (by Definition 3.12)
Take $u \in \mathsf{U}$ with $\mathsf{S}(u) = \textit{var}$ and $u' = T_{\mathsf{I}}(:\text{NUMBER})(u)$ and $u'' = T_{\mathsf{I}}(:\text{TYPE})(u)$, then

$\{\![u]\!\}^{M,A} = A([u])$ (by Definition 3.9)
$= a(v_{SR([u']),SR([u''])})$ (because a and A are corresponding)
$= [\![v_{SR([u']),SR([u''])}]\!]^{M,a}$ (by Definition 1.9)
$= [\![SR([u])]\!]^{M,a}$ (by Definition 3.12)

Hypothesis

Assume that we have shown the lemma for each u.

Step

Take $u \in \mathsf{U}$ with $\mathsf{S}(u) = appl$ and $u' = T_{\mathsf{I}}(:\text{FUNC})(u)$ and $u'' = T_{\mathsf{I}}(:\text{ARG})(u)$, and t, t', t'' as their respective TYPE values, then
$\{\![u]\!\}^{M,A} = \{\![u']\!\}^{M,A}(\{\![u'']\!\}^{M,A})$ (by Definition 3.9)
$= [\![SR([u'])]\!]^{M,a}([\![SR([u''])]\!]^{M,a})$ (by hypothesis)
$= [\![SR([u'])(SR([u'']))_{SR([t'])SR([t''])}]\!]^{M,a}$ (by Definition 3.12)
$= [\![(SR([u'])(SR([u''])))_{SR(t)}]\!]^{M,a}$ (by hypothesis)
$= \{\![SR([u])]\!\}^{M,A}$
Take $u \in \mathsf{U}$ with $\mathsf{S}(u) = abstr$ and $u' = T_{\mathsf{I}}(:\text{VAR})(u)$ and $u'' = T_{\mathsf{I}}(:\text{ARG})(u)$, and t, t', t'' as their respective TYPE values, then
$\{\![u]\!\}^{M,A} = f$ such that for each d,
$f(d) = \{\![T_{\mathsf{I}}(:\text{ARG})(\mathrm{u})]\!\}^{M,A[[u']/d]}$, (by Definition 3.9)
then, $f(d) = [\![SR([u''])]\!]^{M,a[SR([u'])/d]}$ (by hypothesis)
therefore,
$\{\![u]\!\}^{M,A} = [\![(\lambda SR([u']).SR([u'']))_{(SR([t'])SR([t'']))}]\!]^{M,a}$ (by Definition 1.9)
$= [\![(\lambda SR([u']).SR([u'']))_{SR([t])}]\!]^{M,a}$ (by hypothesis)
$= [\![SR([u])]\!]^{M,a}$ (by Definition 3.12)
Take $u \in \mathsf{U}$ with $\mathsf{S}(u) = equ$ and $u' = T_{\mathsf{I}}(:\text{ARG1})(u)$ and $u'' = T_{\mathsf{I}}(:\text{ARG2})(u)$, and t, t', t'' as their respective TYPE values, then
$\{\![u]\!\}^{M,A} = 1$ if $\{\![u']\!\}^{M,A} = \{\![u'']\!\}^{M,A}$, else 0. (by Definition 3.9)
Thus, $\{\![u]\!\}^{M,A} = 1$ if $[\![SR([u'])]\!]^{M,a} = [\![SR([u''])]\!]^{M,a}$, else 0. (by hypothesis)
Thus, $\{\![u]\!\}^{M,A} = 1$ if $[\![SR([u']) = (SR([u'']))]\!]^{M,a} = 1$, else 0. (by Definition 3.12)
Thus, $\{\![u]\!\}^{M,A} = [\![SR([u']) = SR([u''])_{SR([t])}]\!]^{M,a}$ (by hypothesis)
Thus, $\{\![u]\!\}^{M,A} = \{\![SR([u])]\!\}^{M,A}$

PROOF OF PROPOSITION 3.10
Let $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R}\rangle$ be an exhaustive model of $\mathcal{TY}2$,
let int and Int be corresponding constant interpretation functions,
let a and A be corresponding variable assignments,
then for each $u \in \mathsf{U}$ with $\mathsf{S}(u) \sqsubseteq me$,

$SR([u]) \in$ Ty2, and
$\{\![u]\!\}^{M,A} = [\![SR([u])]\!]^{M,a}$ (by Lemma 3.14).

PROPOSITION 3.15
Let $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R}\rangle$ *be an exhaustive model of* $\mathcal{TY}2$.

For each term ϕ *of Ty2, there is a* $\mathcal{TY}2$ *description* δ *such that*

$$D_I(\delta) = \{u \in \mathsf{U} | SR([u]) = \phi\}$$

LEMMA 3.17
For each exhaustive model $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$ *of* $\mathcal{TY}2$*, the function "*" is such that for each element* $i \in (\mathbb{N} \cup Type \cup Ty2)$*, and for each* $u_1, u_2 \in \mathsf{U}$*,*

if $u_1, u_2 \in D_{\mathsf{I}}(i^*)$*,*
then $\langle u_1, \mathsf{I} \rangle$ *and* $\langle u_2, \mathsf{I} \rangle$ *are congruent.*

In the following proof, we only consider a single exhaustive model I of the grammar $\mathcal{TY}2$. For convenience, we will say that two objects u_1 and u_2 in the universe of I are *congruent* iff the configurations under these objects, i.e. $\langle u_1, \mathsf{I} \rangle$ and $\langle u_2, \mathsf{I} \rangle$ are congruent.

PROOF OF LEMMA 3.17

For $\mathsf{S}(u_1) \sqsubseteq number$:

$i = 0$, then $D_{\mathsf{I}}(i^*) = D_{\mathsf{I}}(:\sim zero)$.
As the sort *zero* is atomic, all configurations of objects under an object of sort *zero* are congruent.
$i = j + 1$,
then $D_{\mathsf{I}}(i^*) = D_{\mathsf{I}}(:\sim non\text{-}zero \wedge j^*[:\text{NUMBER}/:])$.
Each $u_1, u_2 \in D_{\mathsf{I}}(i^*)$ is of species *non-zero*.
By hypothesis, we know that the configurations $\langle \mathsf{A}(\text{NUMBER})(u_1), \mathsf{I} \rangle$ and $\langle \mathsf{A}(\text{NUMBER})(u_2), \mathsf{I} \rangle$ are congruent.
Thus, so are $\langle u_1, \mathsf{I} \rangle$ and $\langle u_2, \mathsf{I} \rangle$.

For $\mathsf{S}(u_1) \sqsubseteq type$:

if $\tau = e$, then
$D_{\mathsf{I}}(\tau^*) = D_{\mathsf{I}}(:\sim entity)$.
As the sort *entity* is atomic, all configurations of objects under an objects of sort *zero* are congruent.
analogously for the atomic types t and s.
if $\tau = \langle \tau_1, \tau_2 \rangle$, then
$D_{\mathsf{I}}(\tau^*) = D_{\mathsf{I}}(:\sim complex\text{-}type \wedge \tau_1^*[:\text{IN}/:] \wedge \tau_2^*[:\text{OUT}/:])$.
Each $u_1, u_2 \in D_{\mathsf{I}}(\tau^*)$ is of species *complex-type*.
By hypothesis, we know that the IN and OUT values of u_1 and u_2 are congruent.
The TyIP enforces on both, u_1 and u_2, the maximal number token identities possible, thus, the two objects are congruent as well.

For $\mathsf{S}(u_1) \sqsubseteq me$:

if $\phi = v_{\tau,i}$, then
$D_{\mathsf{I}}(\phi^*) = D_{\mathsf{I}}(:\sim var \wedge i^*[:\text{NUMBER}/:] \wedge \tau^*[:\text{TYPE}/:])$.
Each $u_1, u_2 \in D_{\mathsf{I}}(\phi^*)$ is of species *var*.
We have already shown that the TYPE and NUMBER values of u_1 and u_2 are congruent.
The TIP enforces on both, u_1 and u_2, the maximal number token identities possible, thus, the two objects are congruent as well.
analogously for the species below *const*.
if $\phi = (\phi_1 \phi_2)_\tau$, then
$D_{\mathsf{I}}(\phi^*) = D_{\mathsf{I}}(:\sim appl \wedge \tau^*[:\text{TYPE}/:] \wedge \phi_1^*[:\text{FUNC}/:] \wedge \phi_2^*[:\text{ARG}/:])$.
Each $u_1, u_2 \in D_I(\phi^*)$ are of sort *appl*.
We know by hypothesis that the FUNC and ARG values are congruent.
It follows from the TRP that u_1 and u_2 also have congruent TYPE values.

The TIP guarantees that u_1 and u_2 show the same structure sharings.
Thus, u_1 and u_2 are congruent as well.
analogously for the other subsorts of *me*.

PROOF OF PROPOSITION 3.15
Given Lemma 3.17, it suffices to show that

for each for each $x \in \mathbb{N} \cup Type \cup \text{Ty2}$, and for each $u \in U$, $u \in D_I(x^*)$ iff $SR([u]) = x$.

The proof proceeds by induction on the recursive structure of u.

Let us first consider the case $\mathsf{S}(u) \sqsubseteq number$:

Base
Take $\mathsf{S}(u) = zero$ and $x = 0$.
Then, $u \in D_{\mathsf{I}}\colon {\sim}zero = D_{\mathsf{I}}(0^*) = D_{\mathsf{I}}(x^*)$.
In addition, $SR([u]) = 0 = x$.
Assume that $x \neq 0$.
Then $D_{\mathsf{I}}(x^*) \neq D_{\mathsf{I}}\colon {\sim}zero$, and
because $D_{\mathsf{I}}(x^*)$ describes an indiscernibility class, $u \notin D_{\mathsf{I}}(x^*)$.
In addition, $SR([u]) = 0 \neq x$.
Thus, $\mathsf{S}(u) = 0$ iff $x = 0$.
Hypothesis
Assume that we have shown the proposition for all u' and for all y.
Step
Take $\mathsf{S}(u) = non\text{-}zero$, and $x = y + 1$.
let u' be the NUMBER value of u.
Then, $u \in D_{\mathsf{I}}(x^*)$
iff $u \in D_{\mathsf{I}}(\colon {\sim}non\text{-}zero$ **and** $y^*[\colon \text{NUMBER}/\colon])$,
iff $T_{\mathsf{I}}(\colon \text{NUMBER})(u) \in D_i(y^*)$,
iff $SR([T_{\mathsf{I}}(\colon \text{NUMBER})(u)]) = y$,
iff $SR([u']) = y$
But this holds by hypothesis.

Let us next consider the case $\mathsf{S}(u) \sqsubseteq type$:

Base
For the sort *entity*:
Take $\mathsf{S}(u) = entity$ and $x = e$.
Then, $u \in D_{\mathsf{I}}\colon {\sim}entity = D_{\mathsf{I}}(e^*) = D_{\mathsf{I}}(x^*)$.
In addition, $SR([u]) = e = x$.
Assume that $x \neq 0$.
Then $D_{\mathsf{I}}(x^*) \neq D_{\mathsf{I}}\colon {\sim}entity$, and
because $D_{\mathsf{I}}(x^*)$ describes an indiscernibility class, $u \notin D_{\mathsf{I}}(x^*)$.
In addition, $SR([u]) = e \neq x$.
Thus, $\mathsf{S}(u) = entity$ iff $x = e$.
analogously for the sorts *truth* and *w-index*.
Hypothesis
Assume that we have shown the proposition for each u_1, u_2 of sort *type*, and for each $y_1, y_2 \in \mathbb{N} \cup Type \cup \text{Ty2}$.
Step
Let $\mathsf{S}(u) = c\text{-}type$, and $T_{\mathsf{I}}(\colon\text{IN})(u) = u_1$, and $T_{\mathsf{I}}(\colon\text{OUT})(u) = u_2$,

$u \in D_{\mathsf{I}}(x^*)$
iff $x \in Type$ and there are $y_1, y_2 \in Type$ such that
$x = \langle y_1, y_2 \rangle$, and
$u_1 \in D_{\mathsf{I}}(y_1^*)$ and $u_2 \in D_{\mathsf{I}}(y_2^*)$
But then, $u \in D_{\mathsf{I}}(x^*)$
iff $u \in D_{\mathsf{I}}(: \sim c\text{-}type\ \textbf{and}\ y_1^*[:\textsc{in}/:]\textbf{and}\ y_2^*[:\textsc{out}/:])$
and $u_1 \in D_{\mathsf{I}}(y_1^*)$ and $u_2 \in D_{\mathsf{I}}(y_2^*)$
As $\mathsf{S}(u) = c\text{-}type$,
$u \in D_{\mathsf{I}}(: \sim c\text{-}type\ \textbf{and}\ y_1^*[:\textsc{in}/:]\textbf{and}\ y_2^*[:\textsc{out}/:])$
iff $u_1 \in D_{\mathsf{I}}(y_1^*)$ and $u_2 \in D_{\mathsf{I}}(y_2^*)$.
By hypothesis, this holds iff
$SR([u_1]) = y_1$ and $SR([u_2]) = y_2$
which is the same as
$SR([T_{\mathsf{I}}(:\textsc{in})(u)]) = y_1$ and $SR([T_{\mathsf{I}}(:\textsc{out})(u)]) = y_2$
But as $SR([u]) = \langle SR([T_{\mathsf{I}}(:\textsc{in})(u)]), SR([T_{\mathsf{I}}(:\textsc{out})(u)]) \rangle$
this holds iff
$SR([u]) = \langle y_1, y_2 \rangle = x$

For the case of $\mathsf{S}(u) \sqsubseteq me$ the proof is analogous.

1.2. Extended Definitions

In Section 3.5 we extended the description language Ty2 and the grammar $\mathcal{TY}2$ to contain logical constants and quantifiers explicitly.

(549) Extensions to Definition 3.2:

$\mathsf{S}_{\mathrm{Ty2}}$:

- for each $\phi, \psi \in$ Ty2, for each $v \in Var$,
  - $\mathsf{S}_{\mathrm{Ty2}}((\neg\phi)_t) = negation$,
  - $\mathsf{S}_{\mathrm{Ty2}}((\phi \wedge \psi)_t) = conjunction$,
  - $\mathsf{S}_{\mathrm{Ty2}}((\phi \vee \psi)_t) = disjunction$,
  - $\mathsf{S}_{\mathrm{Ty2}}((\phi \rightarrow \psi)_t) = implication$,
  - $\mathsf{S}_{\mathrm{Ty2}}((\forall v.\phi)_t) = universal$,
  - $\mathsf{S}_{\mathrm{Ty2}}((\exists v.\phi)_t) = existential$,

$\mathsf{A}_{\mathrm{Ty2}}$:

- for each $\phi, \psi \in$ Ty2, for each $v \in Var$,
  - for each $(\neg\phi)_t \in$ Ty2,
    $\mathsf{A}_{\mathrm{Ty2}}(\textsc{arg})((\neg\phi)_t) = \phi$
  - for each $(\phi \wedge \psi)_t \in ty$,
    $\mathsf{A}_{\mathrm{Ty2}}(\textsc{arg1})((\phi \wedge \psi)_t) = \phi$, and
    $\mathsf{A}_{\mathrm{Ty2}}(\textsc{arg2})((\phi \wedge \psi)_t) = \psi$,
  - for each $(\phi \vee \psi)_t \in ty$,
    $\mathsf{A}_{\mathrm{Ty2}}(\textsc{arg1})((\phi \vee \psi)_t) = \phi$, and
    $\mathsf{A}_{\mathrm{Ty2}}(\textsc{arg2})((\phi \vee \psi)_t) = \psi$,
  - for each $(\phi \rightarrow \psi)_t \in ty$,
    $\mathsf{A}_{\mathrm{Ty2}}(\textsc{arg1})((\phi \rightarrow \psi)_t) = \phi$, and
    $\mathsf{A}_{\mathrm{Ty2}}(\textsc{arg2})((\phi \rightarrow \psi)_t) = \psi$,
  - for each $(\forall v.\phi_t)_t \in$ Ty2,
    $\mathsf{A}_{\mathrm{Ty2}}(\textsc{var})((\forall v.\phi)_t) = v$, and
    $\mathsf{A}_{\mathrm{Ty2}}(\textsc{scope})((\forall v.\phi)_t) = \phi$,
  - for each $(\exists v.\phi_t)_t \in$ Ty2,

$\mathsf{A}_{\mathrm{Ty2}}(\textsc{var})((\exists v.\phi)_t) = v$, and
$\mathsf{A}_{\mathrm{Ty2}}(\textsc{scope})((\exists v.\phi)_t) = \phi$.

(550) Extension to Definition 3.9:
if $\mathsf{S}(u) \sqsubseteq$ *neg*,
$\{\![[u]]\!\}^{M,A} = 1$ if $\{\![[T_{\mathsf{I}}(:\textsc{arg})(\mathrm{u})]]\!\}^{M,A} = 0$, else 0.
if $\mathsf{S}(u) \sqsubseteq$ *con*,
$\{\![[u]]\!\}^{M,A} = 1$ if
$\{\![[T_{\mathsf{I}}(:\textsc{arg}1)(\mathrm{u})]]\!\}^{M,A} = 1$ and $\{\![[T_{\mathsf{I}}(:\textsc{arg}2)(\mathrm{u})]]\!\}^{M,A} = 1$,
else 0.
if $\mathsf{S}(u) \sqsubseteq$ *dis*,
$\{\![[u]]\!\}^{M,A} = 1$ if
$\{\![[T_{\mathsf{I}}(:\textsc{arg}1)(\mathrm{u})]]\!\}^{M,A} = 1$ or $\{\![[T_{\mathsf{I}}(:=)(\mathrm{u}arg2)]]\!\}^{M,A}1$,
else 0.
if $\mathsf{S}(u) \sqsubseteq$ *imp*,
$\{\![[u]]\!\}^{M,A} = 1$ if
$\{\![[T_{\mathsf{I}}(:\textsc{arg}1)(\mathrm{u})]]\!\}^{M,A} = 0$ or $\{\![[T_{\mathsf{I}}(:\textsc{arg}2)(\mathrm{u})]]\!\}^{M,A} = 1$,
else 0.
if $\mathsf{S}(u) \sqsubseteq$ *existential*,
for some $v \in \mathsf{U}$ with $v = T_{\mathsf{I}}(:\textsc{var})(u)$,
$\{\![[u]]\!\}^{M,A} = 1$ if
there exists a $d \in D_{E,W,[T_{\mathsf{I}}(:\,\mathrm{TYPE})(u)]}$
such that $\{\![[T_{\mathsf{I}}(:\textsc{scope})(\mathrm{u})]]\!\}^{M,A[[v]/d]} = 1$,
else 0.
if $\mathsf{S}(u) \sqsubseteq$ *universal*,
for some $v \in \mathsf{U}$ with $v = T_{\mathsf{I}}(:\textsc{var})(u)$,
$\{\![[u]]\!\}^{M,A} = 1$ if
for each $d \in D_{E,W,[T_{\mathsf{I}}(:\,\mathrm{TYPE})(u)]}$
$\{\![[T_{\mathsf{I}}(:\textsc{scope})(\mathrm{u})]]\!\}^{M,A[[v]/d]} = 1$,
else 0.

(551) Extension to Definition 3.12:
for each $u \in \mathsf{U}$ such that $\mathsf{S}(u) \sqsubseteq$ *me*,
if $\mathsf{S}(u) =$ *neg*, then,
$SR([u]) = \neg SR([T_{\mathsf{I}}(:\textsc{arg})(u)])$
if $\mathsf{S}(u) =$ *con*, then,
$SR([u]) = (SR([T_{\mathsf{I}}(:\textsc{arg}1)(u)]) \wedge SR([T_{\mathsf{I}}(:\textsc{arg}2)(u)]))_t$,
if $\mathsf{S}(u) =$ *dis*, then,
$SR([u]) = (SR([T_{\mathsf{I}}(:\textsc{arg}1)(u)]) \vee SR([T_{\mathsf{I}}(:\textsc{arg}2)(u)]))_t$,
if $\mathsf{S}(u) =$ *imp*, then,
$SR([u]) = (SR([T_{\mathsf{I}}(:\textsc{arg}1)(u)]) \rightarrow SR([T_{\mathsf{I}}(:\textsc{arg}2)(u)]))_t$,
if $\mathsf{S}(u) =$ *univ*, then,
$SR([u]) = (\forall SR([T_{\mathsf{I}}(:\textsc{var})(u)]).SR([T_{\mathsf{I}}(:\textsc{arg})(u)]))_t$
if $\mathsf{S}(u) =$ *exist*, then,
$SR([u]) = (\exists SR([T_{\mathsf{I}}(:\textsc{var})(u)]).SR([T_{\mathsf{I}}(:\textsc{arg})(u)]))_t$

(552) Extensions to Definition 3.16:
for each $\phi \in \mathrm{Ty2}$,
if $\phi = (\neg\psi)_t$, then

$$\phi^* = \begin{pmatrix} :\sim neg \\ \textbf{and}\ :\textsc{type}\sim truth \\ \textbf{and}\ \psi^*[:\textsc{arg}/:] \end{pmatrix},$$

if $\phi = (\psi_1 \wedge \psi_2)_t$, then

$$\phi^* = \left(\begin{array}{l} :\sim con \\ \textbf{and}\ :\text{TYPE}\sim truth \\ \textbf{and}\ \psi_1^*[:\text{ARG1}/:] \\ \textbf{and}\ \psi_2^*[:\text{ARG2}/:] \end{array}\right),$$

if $\phi = (\psi_1 \vee \psi_2)_t$, then

$$\phi^* = \left(\begin{array}{l} :\sim dis \\ \textbf{and}\ :\text{TYPE}\sim truth \\ \textbf{and}\ \psi_1^*[:\text{ARG1}/:] \\ \textbf{and}\ \psi_2^*[:\text{ARG2}/:] \end{array}\right),$$

if $\phi = (\psi_1 \rightarrow \psi_2)_t$, then

$$\phi^* = \left(\begin{array}{l} :\sim imp \\ \textbf{and}\ :\text{TYPE}\sim truth \\ \textbf{and}\ \psi_1^*[:\text{ARG1}/:] \\ \textbf{and}\ \psi_2^*[:\text{ARG2}/:] \end{array}\right),$$

if $\phi = \forall x.\psi$, then

$$\phi^* = \left(\begin{array}{l} :\sim univ \\ \textbf{and}\ :\text{TYPE}\sim truth \\ \textbf{and}\ x^*[:\text{VAR}/:] \\ \textbf{and}\ \psi^*[:\text{SCOPE}/:] \end{array}\right), \text{ and}$$

if $\phi = \exists x.\psi$, then

$$\phi^* = \left(\begin{array}{l} :\sim exist \\ \textbf{and}\ :\text{TYPE}\sim truth \\ \textbf{and}\ x^*[:\text{VAR}/:] \\ \textbf{and}\ \psi^*[:\text{SCOPE}/:] \end{array}\right).$$

1.3. Additional Definitions for Chapter 4

1.3.1. Additional Definitions for Section 4.2.1.

(553) Additional clauses for the relation `free-variable` in (201):

$$\texttt{free-variable}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow} \left(\boxed{2}\begin{bmatrix} neg \\ \text{ARG}\ \boxed{3} \end{bmatrix} \textbf{ and } \texttt{free-variable}(\boxed{1},\boxed{3}) \right)$$

$$\texttt{free-variable}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow} \left(\begin{array}{l} \boxed{2}\begin{bmatrix} l\text{-}const \\ \text{ARG1}\ \boxed{3} \\ \text{ARG2}\ \boxed{4} \end{bmatrix} \\ \textbf{and } \left(\begin{array}{l} \texttt{free-variable}(\boxed{1},\boxed{3}) \\ \textbf{or } \texttt{free-variable}(\boxed{1},\boxed{4}) \end{array} \right) \end{array} \right)$$

$$\texttt{free-variable}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow} \left(\begin{array}{l} \boxed{2}\begin{bmatrix} exi \textbf{ or } uni \\ \text{VAR}\ \boxed{3} \\ \text{SCOPE}\ \boxed{4} \end{bmatrix} \\ \textbf{and } (\textbf{not } \texttt{are-copies}(\boxed{1},\boxed{3})) \\ \textbf{and } \texttt{free-variable}(\boxed{1},\boxed{4}) \end{array} \right)$$

$$\texttt{free-variable}(\boxed{1},\boxed{2}) \overset{\forall}{\Longleftarrow} \left(\begin{array}{l} \boxed{2}\begin{bmatrix} gen\text{-}quant \\ \text{VAR}\ \boxed{3} \\ \text{RESTR}\ \boxed{4} \\ \text{SCOPE}\ \boxed{5} \end{bmatrix} \\ \textbf{and } (\textbf{not } \texttt{are-copies}(\boxed{1},\boxed{3})) \\ \textbf{and } \left(\begin{array}{l} \texttt{free-variable}(\boxed{1},\boxed{4}) \\ \textbf{or } \texttt{free-variable}(\boxed{1},\boxed{5}) \end{array} \right) \end{array} \right)$$

(554) Additional clauses for the relation `replace` of (203):

$[\psi/x](\neg\phi) = \neg[\psi/x]\phi$

$$\texttt{replace}(x,y,v,w) \stackrel{\forall}{\Longleftarrow} \left(\begin{array}{l} {}^{x}\begin{bmatrix} neg \\ \text{ARG } \boxed{1} \end{bmatrix} \textbf{ and } {}^{v}\begin{bmatrix} neg \\ \text{ARG } \boxed{2} \end{bmatrix} \\ \textbf{and } \texttt{replace}(\boxed{1}, y, \boxed{2}, w) \end{array} \right)$$

For each species σ below *l-const*:

$[\psi/x](\phi_1 \wedge \phi_2) = ([\psi/x]\phi_1 \wedge [\psi/x]\phi_2)$

$$\texttt{replace}(x,y,v,w) \stackrel{\forall}{\Longleftarrow} \left(\begin{array}{l} {}^{x}\begin{bmatrix} \sigma \\ \text{TYPE } \boxed{1} \\ \text{ARG1 } \boxed{2} \\ \text{ARG2 } \boxed{3} \end{bmatrix} \textbf{ and } {}^{v}\begin{bmatrix} \sigma \\ \text{TYPE } \boxed{1} \\ \text{ARG1 } \boxed{4} \\ \text{ARG2 } \boxed{5} \end{bmatrix} \\ \textbf{and } \texttt{replace}(\boxed{2}, y, \boxed{4}, w) \\ \textbf{and } \texttt{replace}(\boxed{3}, y, \boxed{5}, w) \end{array} \right)$$

For each species σ below *quant*:

$[\psi/x](\exists x.\phi) = \exists x.\phi$

$$\texttt{replace}(x,y,v,w) \stackrel{\forall}{\Longleftarrow} \left(\begin{array}{l} {}^{x}\begin{bmatrix} \sigma \\ \text{VAR } y \end{bmatrix} \\ \textbf{and } v \approx x \end{array} \right)$$

For each $\sigma \in \{exist, univ\}$:

$[\psi/x](\exists y.\phi) = \exists y.[\psi/x]\phi$, if $y \neq x$ and $y \notin FV(\psi)$ or $x \notin FV(\phi)$

$$\texttt{replace}(x,y,v,w) \stackrel{\forall}{\Longleftarrow} \left(\begin{array}{l} (\textbf{not } \boxed{2} \approx y) \\ \textbf{and } \left(\begin{array}{l} \textbf{not } \texttt{free-variable}(\boxed{2}, w) \\ \textbf{or not } \texttt{free-variable}(y, \boxed{3}) \end{array} \right) \\ \textbf{and } {}^{x}\begin{bmatrix} \sigma \\ \text{TYPE } \boxed{1} \\ \text{VAR } \boxed{2} \\ \text{SCOPE } \boxed{3} \end{bmatrix} \textbf{ and } {}^{v}\begin{bmatrix} \sigma \\ \text{TYPE } \boxed{1} \\ \text{VAR } \boxed{2} \\ \text{SCOPE } \boxed{4} \end{bmatrix} \\ \textbf{and } \texttt{replace}(\boxed{3}, y, \boxed{4}, w) \end{array} \right)$$

For each species σ below *gen-quant*:

$[\psi/x]([\mathsf{most}\, y : \phi](\phi')) = [\mathsf{most}\, y : [\psi/x]\phi]([\psi/x]\phi')$

if $y \neq x$ and $y \notin FV(\psi)$ or $x \notin FV(\phi) \cup FV(\phi')$

$$\texttt{replace}(x,y,v,w) \stackrel{\forall}{\Longleftarrow} \left(\begin{array}{l} (\textbf{not } \boxed{2} \approx y) \\ \textbf{and } \left(\begin{array}{l} \textbf{not } \texttt{free-variable}(\boxed{2}, w) \\ \textbf{or not } \texttt{free-variable}(y, \boxed{3}) \\ \textbf{or not } \texttt{free-variable}(y, \boxed{4}) \end{array} \right) \\ \textbf{and } {}^{x}\begin{bmatrix} \sigma \\ \text{TYPE } \boxed{1} \\ \text{VAR } \boxed{2} \\ \text{RESTR } \boxed{3} \\ \text{SCOPE } \boxed{5} \end{bmatrix} \textbf{ and } {}^{v}\begin{bmatrix} \sigma \\ \text{TYPE } \boxed{1} \\ \text{VAR } \boxed{2} \\ \text{RESTR } \boxed{4} \\ \text{SCOPE } \boxed{6} \end{bmatrix} \\ \textbf{and } \texttt{replace}(\boxed{3}, y, \boxed{4}, w) \\ \textbf{and } \texttt{replace}(\boxed{5}, y, \boxed{6}, w) \end{array} \right)$$

(555) Additional clauses for the relation `replace1` of (209):

For each species σ below *l-const*:

$$\texttt{replace1}(x,y,v,w) \stackrel{\forall}{\Longleftarrow}$$
$$\begin{array}{l} {}^{x}\begin{bmatrix} \sigma \\ \text{TYPE } \boxed{1} \\ \text{ARG1 } \boxed{2} \\ \text{ARG2 } \boxed{3} \end{bmatrix} \textbf{ and } {}^{v}\begin{bmatrix} \sigma \\ \text{TYPE } \boxed{1} \\ \text{ARG1 } \boxed{4} \\ \text{ARG2 } \boxed{5} \end{bmatrix} \\ \textbf{and } \left(\begin{array}{l} \texttt{replace1}(\boxed{2}, y, \boxed{4}, w) \\ \textbf{and } \boxed{3} \approx \boxed{5} \end{array} \right) \textbf{ or } \left(\begin{array}{l} \texttt{replace1}(\boxed{3}, y, \boxed{5}, w) \\ \textbf{and } \boxed{2} \approx \boxed{4} \end{array} \right) \end{array}$$

For each $\sigma \in \{exist, univ\}$:

$$
\begin{array}{l}
\texttt{replace1}(x, y, v, w) \overset{\forall}{\Longleftarrow} \\
\qquad (\textbf{not } \boxed{2} \approx y) \\
\qquad \textbf{and } (\textbf{not } \texttt{free-variable}(\boxed{2}, w) \textbf{ or not } \texttt{free-variable}(y, \boxed{3})\ ) \\
\qquad \textbf{and } {}^{x}\begin{bmatrix} \sigma & \\ \text{TYPE} & \boxed{1} \\ \text{VAR} & \boxed{2} \\ \text{SCOPE} & \boxed{3} \end{bmatrix} \textbf{ and } {}^{v}\begin{bmatrix} \sigma & \\ \text{TYPE} & \boxed{1} \\ \text{VAR} & \boxed{2} \\ \text{SCOPE} & \boxed{4} \end{bmatrix} \\
\qquad \textbf{and } \texttt{replace1}(\boxed{3}, y, \boxed{4}, w)
\end{array}
$$

For each species σ below *gen-quant*:

$$
\begin{array}{l}
\texttt{replace1}(x, y, v, w) \overset{\forall}{\Longleftarrow} \\
\qquad (\textbf{not } \boxed{2} \approx y) \\
\qquad \textbf{and } (\textbf{not } \texttt{free-variable}(\boxed{2}, w) \textbf{ or not } \texttt{free-variable}(y, \boxed{3})\ ) \\
\qquad \textbf{and } {}^{x}\begin{bmatrix} \sigma & \\ \text{TYPE} & \boxed{1} \\ \text{VAR} & \boxed{2} \\ \text{SCOPE} & \boxed{3} \\ \text{RESTR} & \boxed{5} \end{bmatrix} \textbf{ and } {}^{v}\begin{bmatrix} \sigma & \\ \text{TYPE} & \boxed{1} \\ \text{VAR} & \boxed{2} \\ \text{SCOPE} & \boxed{4} \\ \text{RESTR} & \boxed{5} \end{bmatrix} \\
\qquad \textbf{and } \texttt{replace1}(\boxed{3}, y, \boxed{4}, w)
\end{array}
$$

For each species σ below *gen-quant*:

$$
\begin{array}{l}
\texttt{replace1}(x, y, v, w) \overset{\forall}{\Longleftarrow} \\
\qquad (\textbf{not } \boxed{2} \approx y) \\
\qquad \textbf{and } (\textbf{not } \texttt{free-variable}(\boxed{2}, w) \textbf{ or not } \texttt{free-variable}(y, \boxed{3})\ ) \\
\qquad \textbf{and } {}^{x}\begin{bmatrix} \sigma & \\ \text{TYPE} & \boxed{1} \\ \text{VAR} & \boxed{2} \\ \text{SCOPE} & \boxed{5} \\ \text{RESTR} & \boxed{3} \end{bmatrix} \textbf{ and } {}^{v}\begin{bmatrix} \sigma & \\ \text{TYPE} & \boxed{1} \\ \text{VAR} & \boxed{2} \\ \text{SCOPE} & \boxed{5} \\ \text{RESTR} & \boxed{4} \end{bmatrix} \\
\qquad \textbf{and } \texttt{replace1}(\boxed{3}, y, \boxed{4}, w)
\end{array}
$$

1.3.2. Additional Definitions for Section 4.2.2. The definitions of the chain-encoding of λ-conversion follow the model given in Section 4.2.2. For the species and attributes that are not part of the signature used there, the relevant numbers that are needed for the encoding are given in Table 1.1 on page 400. Notice that the species *neg*, having two attributes defined on it (TYPE, ARG), follows the pattern of the species *c-type*, but with the *s2s1*^{i}-subsequences determined by the numbers in Table 1.1. Similarly, the species below *l-const* follow the pattern of *appl*. The pattern for the generalized quantifiers is a bit more complicated, as they have four attributes. But, apart from this, the definitions have the same structure.

In (556) we define the relation `chain-vr` which is used to perform value raising on term-encoding chains. The functional specification of this relation is given in (261a) in Section 4.2.2.3. The clauses of the relation follow those of the relation `vr` given in (153). The difference between the relations `vr` and `chain-vr` is that the former holds of pairs of *me* objects, whereas the latter holds of pairs of term-encoding chains.

TABLE 1.1. Chart used for the chain encoding of *me* objects, including logical connectives, classical quantifiers and generalized quantifiers:

species	number	attribute	number
zero	1	NUMBER	11
non-zero	2		
truth	3	IN	12
entity	4	OUT	13
w-index	5		
complex-type	6		
variable	7	TYPE	14
application	8	FUNCTOR	15
abstraction	9	ARGUMENT	16
equation	10	VAR	17
$const_1$	22	ARG1	18
		ARG2	19
⋮	⋮		
$const_n$	$n+21$		
negation	$n+22$	SCOPE	20
disjunction	$n+23$		
conjunction	$n+24$		
implication	$n+25$		
existential	$n+26$		
universal	$n+27$		
most	$n+28$	RESTR	21
⋮	⋮		

(556) The relation `chain-vr`

$$\begin{array}{lllllll}
\texttt{chain-vr}(x,y) \stackrel{\forall}{\Longleftarrow} & y = & \langle 9 \rangle & \oplus & & & \\
& & \langle 14 \rangle & \oplus t \oplus & & & \\
& & \langle 17 \rangle & \oplus v \oplus & & & \\
& & \langle 16, & 8 \rangle & \oplus & & \\
& & & \langle 14 \rangle & \oplus t_1 \oplus & & \\
& & & \langle 15, & 8 \rangle \oplus & & \\
& & & & \langle 14 \rangle & \oplus t_2 \oplus & \\
& & & & \langle 15 \rangle & \oplus v' \oplus & \\
& & & & \langle 16 \rangle & \oplus @^{\#} \oplus & \\
& & & \langle 16, & 9 \rangle & \oplus & \\
& & & & \langle 14 \rangle & \oplus t_3 \oplus & \\
& & & & \langle 17 \rangle & \oplus @^{\#} \oplus & \\
& & & & \langle 16 \rangle & \oplus x &
\end{array}$$

$$\texttt{chain-vr}(x,y) \overset{\forall}{\Longleftarrow} \left(\begin{array}{l} a = \begin{array}{ll} \langle 8\rangle & \oplus \\ \langle 14\rangle & \oplus t \oplus \\ \langle 15\rangle & \oplus x \oplus \\ \langle 16\rangle & \oplus v \end{array} \\ \\ \textbf{and } y = \begin{array}{ll} \langle 9\rangle & \oplus \\ \langle 17\rangle & \oplus v' \oplus \\ \langle 16\rangle & \oplus b \oplus \end{array} \\ \\ \textbf{and } \texttt{are-chain-copies}(v,v') \\ \textbf{and } \texttt{chain-vr}(a,b) \end{array}\right)$$

In (557a) we define the relation `chain-ar` which is used to perform argument raising on term-encoding chains. The functional specification of this relation is given in (261b) in Section 4.2.2.3. The clauses of the relation follow those of the relation `ar` given in (160a). The difference between the relations `ar` and `chain-ar` is that the former holds of pairs of *me* objects, whereas the latter holds of pairs of term-encoding chains. Just as with the relation `ar`, we need an auxiliar relation to encode argument raising. This relation, `chain-ar-aux` is specified in (557b).

(557) a. The relation `chain-argument-raising` (`chain-ar`):

$$\texttt{chain-ar}(x,y) \overset{\forall}{\Longleftarrow} \begin{array}{l} a = \begin{array}{ll} \langle 8\rangle & \oplus \\ \langle 14\rangle & \oplus t \\ \langle 15\rangle & \oplus x \\ \langle 16\rangle & \oplus v \end{array} \\ \textbf{and } y = \begin{array}{ll} \langle 9\rangle & \oplus \\ \langle 14\rangle & \oplus t' \oplus \\ \langle 17\rangle & \oplus v' \oplus \\ \langle 16\rangle & \oplus b \end{array} \\ \textbf{and } \texttt{chain-ar-aux}(a,b,v,v') \end{array}$$

$$\texttt{chain-ar}(x,y) \overset{\forall}{\Longleftarrow} \begin{array}{l} a = \begin{array}{ll} \langle 8\rangle & \oplus \\ \langle 14\rangle & \oplus t \oplus \\ \langle 15\rangle & \oplus x \oplus \\ \langle 16\rangle & \oplus v \oplus \end{array} \\ \textbf{and } y = \begin{array}{ll} \langle 9\rangle & \oplus \\ \langle 14\rangle & \oplus t' \oplus \\ \langle 17\rangle & \oplus v' \oplus \\ \langle 16\rangle & \oplus b \oplus \end{array} \\ \textbf{and } \texttt{are-copies}(v,v') \textbf{ and } \texttt{chain-ar}(a,b) \end{array}$$

b. The relation `chain-ar-aux`:

$$
\texttt{ar-aux}(x,y,v,u) \overset{\forall}{\Longleftarrow}\; y = \begin{array}{llllll}
\langle 8\rangle & \oplus & & & & \\
\langle 14\rangle & \oplus t \oplus & & & & \\
\langle 15, & 8\rangle & \oplus & & & \\
 & \langle 15\rangle & \oplus u \oplus & & & \\
 & \langle 16\rangle & \oplus @^{\#} \oplus & & & \\
\langle 16, & 9\rangle & \oplus & & & \\
 & \langle 14\rangle & \oplus t_1 \oplus & & & \\
 & \langle 17\rangle & \oplus @^{\#} \oplus & & & \\
 & \langle 16, & 9\rangle & \oplus & & \\
 & & \langle 14\rangle & \oplus t_2 \oplus & & \\
 & & \langle 17\rangle & \oplus v \oplus & & \\
 & & \langle 16\rangle & \oplus x & &
\end{array}
$$

$$
\textbf{and } \texttt{find-type}(x,t') \textbf{ and } \texttt{are-chain-copies}(t,t')
$$

$$
\texttt{ar-aux}(x,y,v,u) \overset{\forall}{\Longleftarrow}\; a = \begin{array}{ll}
\langle 8\rangle & \oplus \\
\langle 14\rangle & \oplus t \oplus \\
\langle 15\rangle & \oplus x \oplus \\
\langle 16\rangle & \oplus z \oplus
\end{array}
$$

$$
\textbf{and } y = \begin{array}{ll}
\langle 9\rangle & \oplus \\
\langle 14\rangle & \oplus t' \oplus \\
\langle 17\rangle & \oplus z' \oplus \\
\langle 16\rangle & \oplus b
\end{array}
$$

$$
\textbf{and } \texttt{are-chain-copies}(z,z')
$$

$$
\textbf{and } \texttt{chain-ar-aux}(\text{A,B,V,U})
$$

1.3.3. Additional Definitions for Section 4.3.

(558) Formal specification of the Passive DR:

$$
\mathbf{E}\boxed{1}\ \mathbf{E}\boxed{2}\ \mathbf{E}\boxed{3}\ \mathbf{E}\boxed{4}\ \mathbf{E}\boxed{5}\ \mathbf{E}\boxed{6}\ \mathbf{E}\boxed{7}\ \mathbf{E}\boxed{8}\ \mathbf{E}\boxed{9}\ \mathbf{E}\boxed{10}\ \mathbf{E}a\ \mathbf{E}b\ \mathbf{E}c
$$

$$
\begin{bmatrix}
\textit{derivation-rule} & \\
\text{IN} & \boxed{10}\begin{bmatrix}
\textit{word} & \\
\text{PHON} & \boxed{3} \\
\text{SYNS} & \begin{bmatrix}
\text{LOC} & \begin{bmatrix}
\text{CAT} & \begin{bmatrix}
\text{HEAD} & \begin{bmatrix}
\textit{verb} & \\
\text{VFORM} & \textbf{not}\ \ \textit{pas} \\
\text{INV} & \boxed{5} \\
\text{AUX} & \boxed{6}
\end{bmatrix} \\
\text{SUBCAT} & \left\langle \text{NP}\text{—}\boxed{2} \right\rangle \oplus \left\langle \boxed{1} \right\rangle \\
\text{MARKING} & \boxed{7}
\end{bmatrix} \\
\text{CONT} & \boxed{8}\ \lambda y \lambda z_n \ldots \lambda z_1 \lambda x \lambda u_m \ldots \lambda u_1 . \phi
\end{bmatrix} \\
\text{NONL} & \boxed{4}
\end{bmatrix}
\end{bmatrix} \\
\text{OUT} & \begin{bmatrix}
\textit{word} & \\
\text{PHON} & \boxed{3} \\
\text{SYNS} & \begin{bmatrix}
\text{LOC} & \begin{bmatrix}
\text{CAT} & \begin{bmatrix}
\text{HEAD} & \begin{bmatrix}
\textit{verb} & \\
\text{VFORM} & \textit{pas} \\
\text{INV} & \boxed{5} \\
\text{AUX} & \boxed{6}
\end{bmatrix} \\
\text{SUBCAT} & \left\langle \boxed{1}\text{—}\boxed{2} \right\rangle \\
\text{MARKING} & \boxed{7}
\end{bmatrix} \\
\text{CONT} & \boxed{9}
\end{bmatrix} \\
\text{NONL} & \boxed{4}
\end{bmatrix}
\end{bmatrix}
\end{bmatrix}
$$

$$
\textbf{and } \texttt{passive-lf}(\boxed{10},\boxed{8},a) \textbf{ and } \texttt{chain-shifting}(a,b)
$$

$$
\textbf{and } \texttt{reduction}(b,c) \textbf{ and } \texttt{chain2term}(c,\boxed{9})
$$

The relation `passive-lf` relates the input word of the DR $\boxed{10}$, its logical form $\boxed{8}$ and the term-encoding chain a iff the following conditions are met: (i) the SUBCAT list of the input word is of length $n+2$, (ii) the logical form of the input word corresponds to a term of the form $\lambda y \lambda z_n \ldots \lambda z_1 \lambda x \lambda u_m \ldots \lambda u_1.\phi$, and (iii) the chain a encodes the term $\lambda z_n \ldots \lambda z_1 \lambda y \lambda u_m \ldots \lambda u_1.\exists x \phi$. The relation `passive-lf` takes care of the first abstractor of the input term (λy). The relation `passive-lf-aux1` is responsible for the part of the input and the output term, i.e., for $\lambda z_n \ldots \lambda z_1 \lambda x$ and $\lambda z_n \ldots \lambda z_1 \lambda y$ respectively. Finally, the relation `passive-lf-aux2` treats the final part of the input term, i.e., the subterm $\lambda u_m \ldots \lambda u_1.\phi$, and that of the output term, i.e., $\lambda u_m \ldots \lambda u_1.\exists x \phi$.

(559) a. The relation `passive-lf`:

$$\texttt{passive-lf}(\boxed{1},\boxed{2},x) \overset{\forall}{\Longleftarrow} \left(\begin{array}{l} \boxed{1}\left[\text{SYNS LOC CAT SUBCAT REST } \boxed{3}\right] \textbf{ and } \boxed{2}\left[\begin{array}{l} \textit{abstr} \\ \text{VAR } \boxed{4} \\ \text{ARG } \boxed{5} \end{array}\right] \\ \textbf{and } \texttt{passive-lf-aux}(\boxed{3},\boxed{5},x,\boxed{4}) \end{array}\right)$$

b. The relation `passive-lf-aux1`:

$$\texttt{passive-lf-aux1}(\boxed{1},\boxed{2},x,\boxed{3}) \overset{\forall}{\Longleftarrow} \left(\begin{array}{l} \boxed{1}\left[\begin{array}{l} \textit{nelist} \\ \text{REST } \boxed{4}\ \textit{nelist} \end{array}\right] \textbf{ and } \boxed{2}\left[\begin{array}{l} \textit{abstr} \\ \text{VAR } \boxed{5} \\ \text{ARG } \boxed{6} \end{array}\right] \\ \textbf{and } x = \begin{array}{ll} \langle 9 \rangle & \oplus \\ \langle 14 \rangle & \oplus t \oplus \\ \langle 17 \rangle & \oplus v \oplus \\ \langle 16 \rangle & \oplus a \end{array} \\ \textbf{and } \texttt{chain2term}(v,\boxed{5}) \\ \textbf{and } \texttt{passive-lf-aux1}(\boxed{4},\boxed{6},a,\boxed{3}) \end{array}\right)$$

$$\texttt{passive-lf-aux1}(\boxed{1},\boxed{2},x,\boxed{3}) \overset{\forall}{\Longleftarrow} \left(\begin{array}{l} \boxed{1}\left[\begin{array}{l} \textit{nelist} \\ \text{REST } \textit{elist} \end{array}\right] \textbf{ and } \boxed{2}\left[\begin{array}{l} \textit{abstr} \\ \text{VAR } \boxed{4} \\ \text{ARG } \boxed{5} \end{array}\right] \\ \textbf{and } x = \begin{array}{ll} \langle 9 \rangle & \oplus \\ \langle 14 \rangle & \oplus t \oplus \\ \langle 17 \rangle & \oplus v \oplus \\ \langle 16 \rangle & \oplus a \end{array} \\ \textbf{and } \texttt{chain2term}(v,\boxed{3}) \\ \textbf{and } \texttt{passive-lf-aux2}(\boxed{5},a,\boxed{4}) \end{array}\right)$$

c. The relation `passive-lf-aux2`:

$$\texttt{passive-lf-aux2}(\boxed{1},x,\boxed{2}) \overset{\forall}{\Longleftarrow} \left(\begin{array}{l} \boxed{1}\left[\begin{array}{l} \textit{abstr} \\ \text{VAR } \boxed{3} \\ \text{ARG } \boxed{4} \end{array}\right] \\ \textbf{and } x = \begin{array}{ll} \langle 9 \rangle & \oplus \\ \langle 14 \rangle & \oplus t \oplus \\ \langle 17 \rangle & \oplus v \oplus \\ \langle 16 \rangle & \oplus a \end{array} \\ \textbf{and } \texttt{chain2term}(v,\boxed{3}) \\ \textbf{and } \texttt{passive-lf-aux2}(\boxed{4},a,\boxed{2}) \end{array}\right)$$

$$\texttt{passive-lf-aux2}(\boxed{1}, x, \boxed{2}) \overset{\forall}{\Longleftarrow} \begin{pmatrix} \boxed{1}\begin{bmatrix} me \\ \text{TYPE} \;\; truth \end{bmatrix} \\ \textbf{and } x = \begin{array}{ll} \langle 26 \rangle & \oplus \\ \langle 14, & 3 \rangle \oplus \\ \langle 17 \rangle & \oplus v \oplus \\ \langle 20 \rangle & \oplus s \end{array} \\ \textbf{and } \texttt{chain2term}(v, \boxed{2}) \\ \textbf{and } \texttt{chain2term}(s, \boxed{1}) \end{pmatrix}$$

Bibliography

Abeillé, Anne (1988). Parsing French with Tree Adjoining Grammar: Some Linguistic Accounts. In *Proceedings of COLING '88*.

Abeillé, Anne (1995). The Flexibility of French Idioms: A Representation with Lexical Tree Adjoining Grammar. In M. Everaert, E.-J. v. d. Linden, A. Schenk, and R. Schreuder (Eds.), *Idioms. Structural and Psychological Perspectives*, pp. 15–42. Lawrence Erlbaum Associates.

Abney, Steven (1991). Parsing by Chunks. In R. Berwick, S. Abney, and C. Tenney (Eds.), *Principle-Based Parsing*. Kluwer Academic Publishers.

Abney, Steven (1992). Prosodic Structure, Performance Structure and Phrase Structure. In *Proccedings, Speech and Natural Language Workshop*, pp. 425–428. Morgan Kaufmann Publisher, San Mateo, CA.

Abney, Steven (1996). Part-of-Speech Tagging and Partial Parsing. In K. Church, S. Young, and G. Bloothooft (Eds.), *Corpus-Based Methods in Language and Speech*. Kluwer Academic Publishers.

Ammer, Christine (1997). *The American Heritage Dictionary of Idioms*. Houghton Mifflin.

Barwise, Jon (1989). *The Situation in Logic*. CSLI Press.

Barwise, Jon and Cooper, Robin (1981). Generalized Quantifiers and Natural Language. *Linguistics and Philosophy 4*, 159–219.

Beck, Sigrid (1996). *Wh-Constructions and Transparent Logical Form*. Ph. D. thesis, Universität Tübingen.

Bierwisch, Manfred (1983). Semantische und konzeptuelle Repräsentation lexikalischer Einheiten. In R. Růžička and W. Motsch (Eds.), *Untersuchungen zur Semantik*, pp. 61–99. Akademie Verlag, Berlin.

Bierwisch, Manfred (1989). Event Nominalization: Proposals and Problems. In W. Motsch (Ed.), *Wortstruktur und Satzstruktur*, Number 194 in Linguistische Studien, Reihe A, Arbeitsberichte, pp. 1–73. Akademie der Wissenschaften der DDR, Zentralinstitut für Sprachwissenschaft.

Bouma, Gosse (1994). Calculated Flexibility. In H. Bunt, R. Muskens, and G. Rentier (Eds.), *Proceedings of the International Workshop on Computational Semantics*, pp. 32–40. Katholieke Universiteit Brabant.

Calcagno, Mike (1999). Some Thoughts on Tough Movement. In V. Kordoni (Ed.), *Tübingen Studies in HPSG*, Arbeitspapiere des SFB 340, Nr. 132, Volume 1, pp. 198–230. Universität Stuttgart and Universität Tübingen.

Cooper, Robin (1975). *Montague's Semantic Theory and Transformational Grammar*. Ph. D. thesis, University of Massachusetts, Amherst.

Cooper, Robin (1983). *Quantification and Syntactic Theory*. Reidel, Dordrecht.

Copestake, Flickinger, Dan, Malouf, Robert, Riehemann, Susanne, and Sag, Ivan (1995). Translation Using Minimal Recursion Semantics. In *Proceedings of The Sixth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-95)*, Leuven.

Copestake, Flickinger, Dan, and Sag, Ivan (1997). Minimal Recursion Semantics. An Introduction. Manuscript, Stanford University.

Davis, Anthony (1997). *Lexical Semantics and Linking and the Hierarchical Lexicon*. Ph. D. thesis, Stanford University.

Dekker, Paul (1993). *Ups and Downs in Dynamic Semantics*. Ph. D. thesis, Universiteit van Amsterdam.

Donohue, Cathryn and Sag, Ivan A. (1999). Domains in Warlpiri. Paper presented at HPSG-99, University of Edinburgh, available at `ftp://www-csli.stanford.edu/~sag/papers/warlpiri.ps`.

Eberle, Kurt (1997). Flat Underspecified Representation and its Meaning for a Fragment of German. Arbeitspapiere des SFB 340, Nr. 120, Universität Stuttgart and Universität Tübingen.

Egg, Markus (1998). *Wh*-questions in Underspecified Minimal Recursion Semantics. *Journal of Semantics 15*, 37–82.

Ernst, Thomas (1981). Grist for the Linguistic Mill: Idioms and 'Extra' Adjectives. *Journal of Linguistic Research 1*, 51–68.

Fodor, Jerrold A. and Sag, Ivan (1982). Referential and Quantificational Indefinites. *Linguistics and Philosophy 8*, 355–398.

Frank, Anette and Reyle, Uwe (1995). Principle Based Semantics for HPSG. In *Proceedings of the Seventh Conference of the European Chapter of the Association for Computational Linguistics*, pp. 9–16. Association for Computational Linguistics.

Fraser, Bruce (1970). Idioms within a Transformational Grammar. *Foundations of Language 6*, 22–42.

Gallin, Daniel (1975). *Intensional and Higher-Order Modal Logic*. North-Holland, Amsterdam.

Gamut, L. T. F. (1991a). *Logic, Language and Meaning. Volume I: Introduction to Logic*. University of Chicago Press.

Gamut, L. T. F. (1991b). *Logic, Language and Meaning. Volume II: Intensional Logic and Logical Grammar*. University of Chicago Press.

Gazdar, Gerald, Klein, Ewan, Pullum, Geoffrey, and Sag, Ivan (1985). *Generalized Phrase Structure Grammar*. Harvard University Press, Cambridge, Mass.

Ginzburg, Jonathan and Sag, Ivan A. (2000). *Interrogative Investigations. The Form, Meaning, and Use of English Interrogatives*. CSLI Publications.

Grimm, Jacob and Grimm, Wilhelm. *Deutsches Wörterbuch*. Deutscher Taschenbuch Verlag, München. Nachdruck.

Groenendijk, Jeroen and Stokhof, Martin (1982). Semantic Analysis of wh-Complements. *Linguistics and Philosophy 5*, 175 –233.

Groenendijk, Jeroen and Stokhof, Martin (1991). Dynamic Predicate Logic. *Linguistics and Philosophy 14*, 39–100.

Heim, Irene (1982). *The Semantics of Definite and Indefinite Noun Phrases*. Ph. D. thesis, University of Massachusetts.

Heim, Irene and Kratzer, Angelika (1997). *Semantics in Generative Grammar*. Blackwell.

Hendriks, Herman (1993). *Studied Flexibility*. ILLC Dissertation Series 1995-5. Institute for Logic, Language and Computation, Amsterdam.

Hindley, J. R. and Seldin, J. P. (1986). *Introduction to Combinators and the λ-Calculus*. Cambridge University Press.

Höhle, Tilmann N. (1999). An Architecture for Phonology. In R. D. Borsley and A. Przepiórkowski (Eds.), *Slavic in HSPG*, pp. 61–90. CSLI Publications.

Jacobs, Joachim (1988). Probleme der freien Wortstellung im Deutschen. *Sprache und Pragmatik 5*, 8–37.

Janssen, Theo M. V. (1997). Compositionality. In J. van Benthem and A. ter Meulen (Eds.), *Handbook of Logic and Language*, pp. 417–473. Elsevier.

Joshi, Aravind K. (1987). An Introduction to Tree Adjoining Grammars. In A. Manaster-Ramer (Ed.), *Mathematics of Language*, pp. 87–114. John Benjamins Publishing Company, Amsterdam.

Kamp, Hans (1981). A Theory of Truth and Semantic Representation. In J. A. G. Groendijk, T. M. V. Janssen, and M. B. J. Stokhof (Eds.), *Truth, Representation and Information*, pp. 277–322. Dordrecht.

Kamp, Hans and Reyle, Uwe (1993). *From Discourse to Logic*. Kluwer Academic Publishers.

Kasper, Robert (1994). Adjuncts in the Mittelfeld. In J. Nerbonne, K. Netter, and C. Pollard (Eds.), *German in Head-Driven Phrase Structure Grammar*, pp. 39–69. CSLI Publications. Lecture Notes 46.

Kathol, Andreas (1995). *Linearization-Based German Syntax*. Ph. D. thesis, Ohio State University.

Kathol, Andreas and Pollard, Carl J. (1995). Extraposition via Complex Domain Formation. In *Proceedings of the Thirty-Third Annual Meeting of the ACL*, Cambridge, MA, pp. 174–180. ACL.

Katz, Jerrold J. and Fodor, Jerry A. (1964). The Structure of a Semantic Theory. In J. A. Fodor and J. Katz (Eds.), *The Structure of Language*, pp. 479–518. Prentice Hall, Englewood Cliffs, New Jersey. Reprint from *Language 39*, 1963, pp. 170–210.

Kay, Paul (1997). *Words and the Grammar of Context*. CSLI Publications.

Kayne, Richard (1983). Connectedness. *Linguistic Inquiry 14*, 223–249.

Keenan, Edward L. and Westerståhl, Dag (1997). Generalized Quantifiers in Linguistics and Logic. In J. van Benthem and A. ter Meulen (Eds.), *Handbook of Logic and Language*, pp. 837–993. Elsevier.

King, Paul J. (1989). *A logical Formalism for Head-Driven Phrase Structure Grammar*. Ph. D. thesis, University of Manchester.

King, Paul J. (1994). An Expanded Logical Formalism for Head-Driven Phrase Structure Grammar. Arbeitspapiere des SFB 340, Nr. 59, Universität Stuttgart and Universität Tübingen.

King, Paul J. (1999). Towards Truth in HPSG. In V. Kordoni (Ed.), *Tübingen Studies in HPSG*, Arbeitspapiere des SFB 340, Nr. 132, Volume 2, pp. 301–352. Universität Stuttgart and Universität Tübingen.

Kiss, Tibor (1995). *Merkmale und Representationen. Eine Einführung in die deklarative Grammatikanalyse*. Westdeutscher Verlag, Opladen.

Kiss, Tibor (1997, July). Notes from the Border: Scope and Variable Binding in German. unpublished manuscript.

Klein, Ewan and Sag, Ivan A. (1985). Type-Driven Translation. *Linguistics and Philosophy 8*, 163–201.

Krenn, Brigitte and Erbach, Gregor (1994). Idioms and Support Verb Constructions. In J. Nerbonne, K. Netter, and C. Pollard (Eds.), *German in Head-Driven Phrase Structure Grammar*, pp. 365–396. CSLI Publications. Lecture Notes 46.

Kuhn, Jonas (1995, April). Die Behandlung von Funktionsverbgefügen in einem HPSG-basierten Übersetzungsansatz. Verbmobil-Report 66, Universität Stuttgart.

Lappin, Shalom (1991). Concepts of Logical Form in Linguistics and Philosophy. In A. Kasher (Ed.), *The Chomskyan Turn*, pp. 300–333. Basil Blackwell.

Lappin, Shalom and Pollard, Carl (1999). A Hyperintensional Theory of Natural Language Interpretation without Indices or Situations. Draft of July 14.

Lüdeling, Anke (1995, June). Untersuchungen zu einer konzeptuellen Fundierung der lexikalischen Semantik am Beispiel des Vollverbs *haben*. Technical Report 11, Institute for Logic and Linguistic.

May, Robert (1977). *The Grammar of Quantification*. Ph. D. thesis, MIT.

May, Robert (1985). *Logical Form: Its Structure and Derivation*. MIT Press.

May, Robert (1989). Interpreting Logical Form. *Linguistics and Philosophy 12*, 387–435.

May, Robert (1991). Syntax, Semantics and Logical Form. In A. Kasher (Ed.), *The Chomskyan Turn*, pp. 334–359. Basil Blackwell.

McCaig, Isabel and Manser, Martin H. (1986). *A Learner's Dictionary of English Idioms.* Oxford University Press. cited after the fourth impression, 1991.

Meurers, Walt Detmar (2000, February). Lexical Generalizations in the Syntax of German Non-finite Constructions. Arbeitsberichte des SFB 340, Nr. 145, Universität Stuttgart and Universität Tübingen. Phd thesis, Universität Tübingen.

Montague, Richard (1974a). English as a Formal Language. In R. H. Thomason (Ed.), *Formal Philosophy. Selected Papers of Richard Montague*, pp. 188–221. Yale University Press.

Montague, Richard (1974b). The Proper Treatment of Quantification in Ordinary English. In R. H. Thomason (Ed.), *Formal Philosophy. Selected Papers of Richard Montague*, pp. 247–270. Yale University Press.

Nerbonne, John (1992). Constraint-Based Semantics. In P. Dekker and M. Stokhof (Eds.), *Proceedings of the 8th Amsterdam Colloquium*, pp. 425–444. Institute for Logic, Language and Information.

Nunberg, Geoffrey, Sag, Ivan A., and Wasow, Thomas (1994). Idioms. *Language 70*, 491–538.

Partee, Barbara H., Meulen, Alice ter, and Wall, Robert E. (1993). *Mathematical Methods in Linguistics.* Kluwer Academic Press, Dordrecht. corrected second printing of the first edition.

Penn, Gerald (1999a). A Generalized-Domain-Based Approach to Serbo-Croatian Second Position Clitic Placement. In G. Bouma, , E. Hinrichs, G. Kruijff, and R. Oehrle (Eds.), *Constraints and Resources in Natural Language Syntax and Semantics*, pp. 119–136. CSLI Publications.

Penn, Gerald (1999b). Linearization and WH-Extraction in HPSG: Evidence from Serbo-Croatian. In R. D. Borsley and A. Przepiórkowski (Eds.), *Slavic in HPSG*, pp. 247–282. CSLI Publications.

Penn, Gerald (1999c). An RSRL Formalization of Serbo-Croatian Second Position Clitic Placement. In V. Kordoni (Ed.), *Tübingen Studies in HPSG*, Arbeitspapiere des SFB 340, Nr. 132, Volume 1, pp. 177–197. Universität Stuttgart and Universität Tübingen.

Pollard, Carl and Sag, Ivan (1987). *Information Based Syntax and Semantics. Vol.1: Fundamentals.* CSLI Lecture Notes 13.

Pollard, Carl and Sag, Ivan A. (1994). *Head-Driven Phrase Structure Grammar.* University of Chicago Press.

Pollard, Carl J. (1996). On Head Non-Movement. In H. Bunt and A. van Horck (Eds.), *Discontinuous Constituency*, Number 6 in Natural language processing, pp. 279–305. Mouton de Gruyter, Berlin, New York.

Pollard, Carl J. (1999). Strong Generative Capacity in HPSG. In G. Webelhuth, J.-P. Koenig, and A. Kathol (Eds.), *Lexical and Constructional Aspects of Linguistic Explanation*, pp. 281–297. CSLI Publications.

Pollard, Carl J. and Yoo, Eun Jung (1998). A Unified Theory of Scope for Quantifiers and *wh*-Phrases. *Journal of Linguistics 34*, 415–445.

Progovac, Ljiljana (1988). *A Binding Approach to Polarity Sensitivity.* Ph. D. thesis, University of Southern California.

Przepiórkowski, Adam (1997). Quantifiers, Adjuncts as Complements, and Scope Ambiguities. Manuscript, Universität Tübingen. Draft of June 20, 1997.

Przepiórkowski, Adam (1998). 'A Unified Theory of Scope' Revisited. In *Proceedings of the FHCG-98, 14–16 August 1998, Saarbrücken*, pp. 185–195.

Przepiórkowski, Adam (1999a). *Case Assignment and the Complement-Adjunct Dichotomy: A Non-Configurational Constraint-Based Approach.* Ph. D. thesis, Universität Tübingen.

Przepiórkowski, Adam (1999b). Negative Polarity Questions and Italian Negative Concord.

In V. Kordoni (Ed.), *Tübingen Studies in HPSG*, Arbeitspapiere des SFB 340, Nr. 132, Volume 2, pp. 353–400. Universität Stuttgart and Universität Tübingen.

Przepiórkowski, Adam and Kupść, Anna (1999). Eventuality Negation and Negative Concord in Polish and Italian. In R. D. Borsley and A. Przepiórkowski (Eds.), *Slavic in HPSG*, pp. 211–246. CSLI Publications.

Pulman, Stephen G. (1993). The Recognition and Interpretation of Idioms. In C. Cacciari and P. Tabossi (Eds.), *Idioms: Processing, Structure, and Interpretation*, Chapter 11, pp. 249–270. Lawrence Erlbaum Associates.

Reape, Mike (1990). Getting Things in Order. In *Proceedings of the Symposium on Discontinuous Constituency*. Institute for Language Technology and Artificial Intelligence.

Reape, Mike (1992). *A Formal Theory of Word Order: A Case Study of West Germanic.* Ph. D. thesis, University of Edinburgh.

Reape, Mike (1994). Domain Union and Word Order Variation in German. In J. Nerbonne, K. Netter, and C. Pollard (Eds.), *German in Head-Driven Phrase Structure Grammar*, pp. 151–197. CSLI Publications. Lecture Notes 46.

Reinhard, Sabine (2001). *Deverbale Komposita an der Morphologie-Syntax-Semantik-Schnittstelle: ein HPSG-Ansatz.* Ph. D. thesis, Universität Tübingen.

Richter, Frank (1997). Die Satzstruktur des Deutschen und die Behandlung langer Abhängigkeiten in einer Linearisierungsgrammatik. Formale Grundlagen und Implementierung in einem HPSG-Fragment. In E. Hinrichs, W. D. Meurers, F. Richter, M. Sailer, and H. Winhart (Eds.), *Ein HPSG-Fragment des Deutschen. Teil 1: Theorie*, Arbeitspapiere des SFB 340, Nr. 95, pp. 13–187. Universität Stuttgart and Universität Tübingen.

Richter, Frank (1999). RSRL for HPSG. In V. Kordoni (Ed.), *Tübingen Studies in HPSG*, Arbeitspapiere des SFB 340, Nr. 132, Volume 1, pp. 74–115. Universität Stuttgart and Universität Tübingen.

Richter, Frank (2000). A Mathematical Formalism for Linguistic Theories with an Application in Head-Driven Phrase Structure Grammar. Dissertation, submitted to the Neuphilologische Fakultät of the Universität Tübingen, Version of April 28th, 2000.

Richter, Frank and King, Paul J. (1997). On the Existence of Exhaustive Models in a Relational Feature Logic for Head-Driven Phrase Structure Grammar. unpublished manuscript university of Tübingen.

Richter, Frank and Sailer, Manfred (1995). Remarks on Linearization. Reflections on the Treatment of LP-Rules in HPSG in a Typed Feature Logic. Magisterarbeit, Seminar für Sprachwissenschaft, Universität Tübingen.

Richter, Frank and Sailer, Manfred (1996a). PTQ, LF, TFL. Zum theoretischen Status der "Übersetzung in eine formale Semantiksprache". Ausarbeitung zu einem Vortrag bei *Sinn und Bedeutung. 1. Jahrestagung der Gesellschaft für Semantik*, Tübingen, 19.–21.12.1996.

Richter, Frank and Sailer, Manfred (1996b). Regions and Word Order. Ausarbeitung zu einem Vortrag beim "International Center Workshop on Computational Linguistics", 20.9.1996 Tübingen.

Richter, Frank and Sailer, Manfred (1999a). A Lexicalist Collocation Analysis of Sentential Negation and Negative Concord in French. In V. Kordoni (Ed.), *Tübingen Studies in Head-Driven Phrase Structure Grammar*, Arbeitspapiere des SFB 340, Nr. 132, Volume 1, pp. 231–300. Universität Stuttgart and Universität Tübingen.

Richter, Frank and Sailer, Manfred (1999b). Lexicalizing the Left Periphery of German Finite Sentences. In V. Kordoni (Ed.), *Tübingen Studies in HPSG*, Arbeitspapiere des SFB 340, Nr. 132, Volume 1. Universität Stuttgart and Universität Tübingen.

Richter, Frank and Sailer, Manfred (1999c). LF Conditions on Expressions of Ty2: An HPSG Analysis of Negative Concord in Polish. In R. D. Borsley and A. Przepiórkowski (Eds.), *Slavic in HPSG*, pp. 247–282. CSLI Publications.

Richter, Frank and Sailer, Manfred (1999d). Underspecified Semantics in HPSG. In H. Bunt and R. Muskens (Eds.), *Computing Meaning*, Studies in Linguistics and Philosophy, pp.

95–112. Kluwer academic publishers.

Richter, Frank and Sailer, Manfred (2001). On the Left Periphery of German finite Sentences. In W. D. Meurers and T. Kiss (Eds.), *Constraint-Based Approaches to Germanic Syntax*, pp. 257–300. Stanford: CSLI Publications.

Richter, Frank, Sailer, Manfred, and Penn, Gerald (1999). A Formal Interpretation of Relations and Quantification in HPSG. In G. Bouma, E. Hinrichs, G.-J. M. Kruijff, and R. T. Oehrle (Eds.), *Constraints and Resources in Natural Language Syntax and Semantics*, pp. 281–298. CSLI Publications.

Riehemann, Susanne (1997). Idiomatic Constructions in HPSG. Paper presented at the *Fourth International Conference on Head-Driven Phrase Structure Grammar*, Ithaca. `http://doors.stanford.edu/~sr/idioms.ps`.

Riehemann, Susanne Z. (2001). *A Constructional Approach to Idioms and Word Formation.* Ph. D. thesis, Stanford University.

Riehemann, Susanne Z. and Bender, Emily (2000). Absolute Constructions: On the Distribution of Predicative Idioms. In S. Bird, A. Carnie, J. Haugen, and P. Norquest (Eds.), *Proceedings of the 18th WCCFL*, pp. 476–489. Cascadilla Press, Somerville.

Rodman, Robert (1976). Scope Phenomena, "Movement Transformations", and Relative Clauses. In B. Partee (Ed.), *Montague Grammar*, pp. 165–176. Academic Press.

Sag, Ivan A. (1987). Grammatical Hierarchy and Linear Precedence. In G. Huck and A. Ojeda (Eds.), *Discontinuous Constituency*, Syntax and Semantics, Nr. 20, pp. 303–340. Academic Press, New York. Also published as Report CSLI-88-129. Stanford: Center for the Study of Language and Information.

Sag, Ivan A. (1997). English Relative Clause Constructions. *Journal of Linguistics 33*, 431–483.

Sag, Ivan A. and Fodor, Janet D. (1994). Extraction Without Traces. In R. Aranovich, W. Byrne, S. Preuss, and M. Senturia (Eds.), *Proceedings of the Thirteenth West Coast Conference on Formal Linguistics*, Volume 13. CSLI Publications/SLA, Stanford.

Sailer, Manfred (1997). Adjunkte in einer Linearisierungsgrammatik: Syntax, Semantik und Wortstellung. In E. Hinrichs, W. D. Meurers, F. Richter, M. Sailer, and H. Winhart (Eds.), *Ein HPSG-Fragment des Deutschen. Teil 1: Theorie*, Arbeitspapiere des SFB 340, Nr. 95, pp. 249–318. Universität Stuttgart and Universität Tübingen.

Schabes, Y., Abeille, A., and Joshi, A. K. (1988). Parsing Strategies with 'Lexicalized' Grammars: Application to Tree Adjoining Grammars. In *Proceedings of the 12th COLING*, Budapest, Hungary, pp. 578–583.

Schenk, André (1995). The Syntactic Behavior of Idioms. In M. Everaert, E.-J. v. d. Linden, A. Schenk, and R. Schreuder (Eds.), *Idioms. Structural and Psychological Perspectives*, pp. 253–271. Lawrence Erlbaum Associates.

Shieber, Stuart M. and Schabes, Yves (1990). Synchronous Tree-Adjoining Grammars. In *Proceedings of COLING-90*, pp. 253–258.

Uszkoreit, Hans (1986). Constraints on Order. CSLI-Report 86-46, CSLI.

Uszkoreit, Hans (1987). *Word Order and Constituent Structure in German.* CSLI Lecture Notes, Nr. 8. CSLI, Stanford.

Wouden, Ton van der (1997). *Negative Contexts. Collocation, Polarity and Multiple Negation.* Routledge.

Vijay-Shanker, K. and Schabes, Yves (1992). Structure Sharing in Lexicalized Tree-Adjoining Grammars. In *Proceedings of COLING-92*, pp. 205–211.

Stechow, Arnim von (1993). Die Aufgaben der Syntax. In J. Jacobs, A. von Stechow, W. Sternefeld, and T. Vennemann (Eds.), *Syntax. Ein internationales Handbuch zeitgenössischer Forschung*, Volume 1, pp. 1–88. Walter de Gruyter.

Wasow, Thomas, Sag, Ivan A., and Nunberg, Geoffrey (1983). Idioms: An Interim Report. In S. Hattori and K. Inoue (Eds.), *Proceedings of the XIIIth International Congress of Linguists*, pp. 102–115.

Westerståhl, Dag (1989). Quantifiers in Formal and Natural Languages. In D. Gabbay and F. Guenthner (Eds.), *Handbook of Philosophical Logic*, pp. 1–131. D. Reidel Publishing Company.

Wolf, Siegmund A. (1956). *Wörterbuch des Rotwelschen. Deutsche Gaunersprache*. Bibliographisches Institut, Mannheim. Second edition, 1985, Buske.

Zimmermann, Thomas Ede (1989). Intensional Logic and Two-Sorted Type Theory. *The Journal of Symbolic Logic 54*(1), 65–77.

Zimmermann, Thomas Ede (1993). On the Proper Treatment of Opacity in Certain Verbs. *Natural Language Semantics 1*(2), 149–179.

Zwicky, Arnold M. and Sadock, Jerrold M. (1975). Ambiguity Tests and How to Fail Them. In J. P. Kimball (Ed.), *Syntax and Semantics*, Volume 4, pp. 1–36. Academic Press, New York.