Negative Concord with Negative Quantifiers:
A Polyadic Quantifier Approach to Romanian Negative Concord

Gianina Nicoleta Iordăchioaia

Philosophische Dissertation
angenommen von der Neuphilologischen Fakultät
der Universität Tübingen
am 24. November 2009

Tübingen
2010

Gedruckt mit Genehmigung der Neuphilologischen Fakultät
der Universität Tübingen

Hauptberichterstatter:	PD Dr. Frank Richter
Mitberichterstatter:	Prof. Dr. Erhard W. Hinrichs
Dekan:	Prof. Dr. Johannes Kabatek

Acknowledgements

There are many people whose more or less direct support contributed to my writing this dissertation and I know that no matter how long these acknowledgments, there will always be somebody I forget or whom I cannot thank enough.

First and foremost, I thank Frank Richter, my supervisor, for the hours, days and months he spent reading, analyzing, listening and trying to understand the most undeveloped thoughts on negation I would come up with. Thanks for the unique inspiring discussions we had, thanks for initiating me into the secrets of getting a dissertation done, thanks for bringing me up, for being a real Doktorvater to me.

Next, I want to thank Erhard Hinrichs, my second advisor, who played a central role in my decision to pursue a PhD in Tübingen. Thanks for making my life so much easier at the beginning of my stay in Tübingen and for introducing me into the academic world.

I am extremely grateful to the other two committee members, Uwe Mönnich and Fritz Hamm, for always being there for me and doing their best to answer questions that I sometimes didn't understand myself. I thank Susanne Winkler and the whole committee of the defense colloquium for being so supportive that day, for telling me exactly what I needed to hear before and after the defense, for making the day so special.

I cannot imagine and I may not even want to know what would have happened, had I not met Artemis Alexiadou at the most critical point in the process of my writing this dissertation. Thank you, Artemis, for finding a way to make me useful to your department, for the emotional, financial and professional support, and for making me discover unsuspected energy resources in myself.

While working on my dissertation, I received financial support from the Open Society Institute, which awarded me a Supplementary Grant in 2007, the German Research Foundation (DFG), via various short-term project positions I held within the Collaborative Research Centers (SFB) 441 in Tübingen and 732 in Stuttgart, and the German Academic Exchange Service (DAAD), which facilitated me with a three-month PhD grant via the Academic Advisory Office of the University of Tübingen. I am grateful to Dr. Reinhard Brunner for his great help and advice while I was looking for funding options.

However, before I had to deal with the ups and downs of a dissertation, it was the presence of my university professors Alexandra Cornilescu and Emil Ionescu that had a decisive impact on my professional future. I always regretted having to leave Bucharest to pursue a PhD, but this loss was countervailed by your friendship and permanent support even from thousands of kilometers away. The few times we met since I left acted like radiant flashes of motivation and intellectual energy on me and my writing. Thank you for initiating me into linguistics in Bucharest, for your enthusiasm and for your belief in me. I am also grateful to Andrei and Larisa Avram, Ana-Maria Barbu, Dana Isac, Cristian Moroianu, and Andra Vasilescu for the engaging courses they taught in Bucharest.

And even before the university it was my high-school teacher of Romanian, Ioan Dănilă, who first stirred my interest in linguistics. He once observed that I had a special sense for language, but it was only his fascinating classes that developed this sense in me, when he first told us about Saussure and

Chomsky, and linguistics as a science. I knew right away that this was what I wanted to do.

Coming back to the time when I wrote my dissertation, I greatly benefited from stimulating discussions with and insightful comments from Henk Barendregt, Edward Keenan, Richard Larson, Manfred Sailer, and Ede Zimmermann. Thank you so much for taking your time to read, listen to, think of, and comment on various parts in this dissertation.

There is a special category of people who played a very important role during the time when I was writing this dissertation: while they started out as my teachers, Ana-Maria Barbu and Janina Radó soon became close friends, and their presence has been a joy to me every time I talked to them. Moreover, this dissertation would be much more difficult to follow, hadn't it been for Janina, who proofread it and came up with great ideas to improve the readability of the text.

Many people were just there for me when I most needed a friend, a piece of advice, or simply understanding, and I will always associate them with this time of my life. Thank you, Georgiana Dinu, Sophia Katrenko, Sandra Kübler, Mingya Liu, Susanne Lohrmann, Mihaela Marchis, Sabine Mohr, Doris Penka, Jelena Prokic, Zoulfiya Rakhmatoulina, Florian Schäfer, Jan-Philipp Söhn, Cornelia Stoll, Beata Trawiński, Julia Trushkina, Heike Winhart, Holger Wunsch, Hedde Zeijlstra, and Heike Zinsmeister. Chatting with you now and then, your friendship and support were the daily little sunrays that cheered me up and kept me going.

Special thanks are due to many other people who provided me with data judgements and/or careful comments. Besides most of the colleagues and friends just above, I mention Ana-Maria Barbu, Verginica Barbu Mititelu, Gabriela Bîlbîie, Anamaria Fălăuş, Cristina Ionică, Dana Isac, Roxana Maftei Ciolăneanu, Cătălin Mititelu, Dafina Raţiu, Oana Săvescu Ciucivara, Elena Soare, and Dragoş Trică. Thank you for your patience with the tricky data and my persistent questions.

Finally, I have to mention the least obvious, but most powerful sources of energy for my emotional survival through this process. First, I thank Cristina for her longstanding friendship despite time and distance and for her tenacious faith in me at moments when I had almost given up myself. If not this, I don't know what a true friend is. Second, I thank my family, Bibi, Zaia, and Dadu, for encouraging and accepting all my decisions, and especially for coping with the distance I had to put between me and them to follow my dreams. Finally, I thank Dragoş for his love and support over these years, for listening to endless complaints and processing more linguistics than he had ever imagined he could, and I hope he will forgive my preference for John in the data.

To my parents

Short table of contents

Contents

List of abbreviations

$\in$	Infix notation for the `member` relation: p. 163
$\oplus$	Infix notation for the `append` relation: p. 47
$\triangleleft$	Infix notation for the `subterm` relation: p. 162
$\triangleleft_{\in}$	Infix notation for the `subterm-of-member` relation: p. 170
AR	Argument Raising: p. 130
Cum	Cumulation: p. 29
It	Iteration: p. 22
L	Logical language L: p. 123
Res	Resumption: p. 32
$Ty1$	One-sorted Type Theory: p. 154
AVM	Attribute-Value Matrix
CL	Clitic
CN	Common Noun
DE	Downward Entailing
Det	Determiner
DN	Double Negation
GQT	Generalized Quantifier Theory
HPSG	Head-driven Phrase Structure Grammar
IV	Intransitive Verb
LF-Ty2	Lexicalized Flexible Ty2
LRS	Lexical Resource Semantics
M	Marker
NC	Negative Concord
NM	Negative Marker
NQ	Negative Quantifier
PE	Romanian object marking preposition *pe*
PF	Perfect
PPI	Positive Polarity Item
RF	Reflexive pronoun
RSRL	Relational Speciate Re-entrant Language
S	Sentence
SJ	Subjunctive
TV	Transitive Verb
Ty2	Two-sorted Type Theory

Chapter 1

Introduction

1.1 Negative concord

This thesis is an investigation of negative concord with application to Romanian. Negative concord (NC) languages like Romanian pose an important challenge to our common linguistic practice of composing meaning: they use several negative constituents in one sentence with an overall interpretation of single negation. The negative sentence (1a) with one negative expression (*nobody*) in a non-NC language like standard English has the Romanian counterpart in (1b), a sentence with two negative expressions: *nimeni* 'nobody' and *nu* 'not'. In English employing both negative expressions *nobody* and *not* results in an affirmative interpretation (1c), which is unavailable for the Romanian (1b):

(1) a. **Nobody** came.
$\neg\exists x\ [person'(x) \wedge come'(x)]$

b. **Nimeni nu** a venit.
nobody not has come
'Nobody came.'
i. $\neg\exists x\ [person'(x) \wedge come'(x)]$
ii. # $\neg\exists x\ [person'(x) \wedge \neg come'(x)]$

c. **Nobody** did **not** come.
$\neg\exists x\ [person'(x) \wedge \neg come'(x)]$

The fact that both *nimeni* and *nu* have negative semantics is confirmed by (2a) and (2b), where each one alone is responsible for the negative interpretation of the construction, just like in the English parallel translations:

(2) a. Cine a venit? **Nimeni**.
who has come nobody
'Who came? Nobody.'

b. Ion **nu** a venit.
John not has come
'John didn't come.'

I use the term 'negative marker' (NM) to refer to the verbal negation in NC languages, like *nu* in Romanian. The term 'n-word' introduced by Laka (1990) is employed to designate nominal and adverbial negative constituents like *nimeni* and *nobody* in both NC and non-NC languages.

This thesis aims at an analysis of NC in Romanian that accounts for the negative semantics of n-words and the NM and provides a semantic mechanism by which we can interpret two or more such negative expressions as contributing one sentential negation.

1.2 The theoretical problem and two possible solutions

The linguistic interest in NC has a rich tradition starting at least with Jespersen (1917), but the term was introduced in Labov (1972). More recently, NC has been discussed both from a crosslinguistic perspective (Ladusaw (1992), Haegeman (1995), Corblin and Tovena (2001), Zeijlstra (2004), Giannakidou (2006), Richter and Sailer (2006), Penka (2007), Tubau (2008)) and in relation to individual languages (for Spanish: Laka (1990), Suñer (1995), Herburger (2001), Catalan: Espinal (2000), Portuguese: Peres (1997), French: Déprez (1997), Mathieu (2001), de Swart and Sag (2002), Italian: Zanuttini (1991), Acquaviva (1997), Przepiórkowski (1999*a*), Tovena (2003), Romanian: Isac (1998, 2004), Ionescu (1999, 2004), Greek: Giannakidou (1998), Serbo-Croatian: Progovac (1994), Polish: Przepiórkowski and Kupść (1997, 1999), Błaszczak (1999), Richter and Sailer (1999*b*, 2004), to name just a few).

The problem that NC raises for linguistic theory, informally described above, can be formulated in more precise terms if we consider NC in relation to the principle of compositionality, which is fundamental in linguistics nowadays.

Compositionality and negation The principle of compositionality (3) states that the meaning of a complex linguistic expression must be composed from the individual meanings of its syntactic parts by means of a function that is consistent with their syntax. This function is usually referred to as the 'mode of composition'.

(3) *The principle of compositionality* (Partee (1984, p. 281))
The meaning of an expression is a function of the meanings of its parts and of the way they are syntactically combined.

To check if the principle of compositionality is respected in the interpretation of the sentences in (1), we should first identify their parts with the corresponding meanings. Let us start with (1c). This sentence has two syntactic parts: the NP *nobody* and the VP *didn't come*. If we represent the meaning of linguistic expressions in terms of a higher-order logical language (Gamut (1991)), the English n-word *nobody* corresponds to the negative quantifier in (4a) and *didn't come* to the negative property in (4b):[1]

(4) a. nobody $\rightsquigarrow \lambda P.\neg\exists x\ [person'(x) \wedge P(x)]$
b. didn't come $\rightsquigarrow \lambda v.\neg come(v)$

Combining the two parts by functional application, the typical mode of composition, gives us the derivation in FIGURE 1.1. Further β-reduction and functional application at the S level ultimately lead to the predicate logic formula that was given in (1c) as the meaning of the English sentence: $\neg\exists x\ [person'(x) \wedge \neg come'(x)]$. This shows that the interpretation of the English sentence (1c) respects the principle of compositionality with functional application as the mode of composition.

[1] I ignore here the tense and auxiliary semantics as well as the detailed syntactic information of the verb, as they are not relevant for the present purposes.

S
Nobody didn't come
$\neg\exists x\ [person'(x) \wedge \neg come'(x)]$
$\uparrow$
$\neg\exists x\ [person'(x) \wedge [\lambda v.\neg come'(v)](x)]$
$\uparrow$
$\lambda P.\neg\exists x\ [person'(x) \wedge P(x)](\lambda v.\neg come'(v))$

NP
Nobody
$\lambda P.\neg\exists x\ [person'(x) \wedge P(x)]$

VP
didn't come
$\lambda v.\neg come'(v)$

Figure 1.1: Syntactic derivation and interpretation for *Nobody didn't come*

Double negation (DN) The cooccurrence of two negations in the predicate logic formula obtained in FIGURE 1.1 makes it truth-conditionally equivalent to a positive formula, if we consider the logical law of double negation in (LEMMA 1.1), by which two logical negations cancel each other.

Lemma 1.1 The law of double negation
For every formula p, the following holds:
$\neg\neg p \Leftrightarrow p$

To apply the law of double negation to the formula in FIGURE 1.1 we have to make the two negative operators adjacent by use of logical inference rules. This is done in (5a). We first replace the existential quantifier outscoped by negation with a universal quantifier outscoping negation (LEMMA 1.2). The result contains the negation of a conjunction which can be substituted by an implication with a positive antecedent and a negative consequent (LEMMA 1.3). We thus obtain the desired adjacent negative operators that cancel each other (see the third line in (5a)). The result is the positive formula in (5a), which corresponds to our intuition concerning the English sentence (1c): see (5b).

Lemma 1.2 The law of quantifier negation
For every variable x, for every formula ψ, the following holds:
$\neg\exists x\ \psi \Leftrightarrow \forall x\ \neg\psi$

Lemma 1.3 *For all formulas ϕ and ψ, the following holds:*[2]
$\neg(\phi \wedge \psi) \Leftrightarrow (\phi \rightarrow \neg\psi)$

(5) a. $\neg\exists x\ [person'(x) \wedge \neg come'(x)]$
$\overset{L:1.2}{\Longleftrightarrow} \forall x \neg[person'(x) \wedge \neg come'(x)]$
$\overset{L:1.3}{\Longleftrightarrow} \forall x\ [person'(x) \rightarrow \neg\neg come'(x)]$
$\overset{L:1.1}{\Longleftrightarrow} \forall x\ [person'(x) \rightarrow come'(x)]$

[2]This rule is derived on the basis of the DeMorgan law $\neg(\phi \wedge \psi) \Leftrightarrow (\neg\phi \vee \neg\psi)$ and the conditional law $(\neg\phi \vee \psi) \Leftrightarrow (\phi \rightarrow \psi)$ (see Partee et al. (1990, Sec. 6.4)).

S
Nobody came
$\neg\exists x\ [person'(x) \wedge come'(x)]$
$\uparrow$
$\neg\exists x\ [person'(x) \wedge [\lambda v.come'(v)](x)]$
$\uparrow$
$\lambda P.\neg\exists x\ [person'(x) \wedge P(x)](\lambda v.come'(v))$

NP
Nobody
$\lambda P.\neg\exists x\ [person'(x) \wedge P(x)]$

VP
came
$\lambda v.come'(v)$

Figure 1.2: Syntactic derivation and interpretation for *Nobody came*

b. **Nobody** did **not** come. = Everybody came.

The fact that the cooccurrence of two negative expressions in a sentence triggers a double negation interpretation makes standard English a so-called DN language. This contrasts with NC languages like Romanian, where two negative expressions yield a NC interpretation.

The NC challenge Let us now return to the Romanian sentence in (1) to see what the principle of compositionality predicts. (1b) is made up of syntactic parts similar to those in (1c): the n-word *nimeni* 'nobody' and the negated verb *nu a venit* 'didn't come'. Assuming, as for English and as indicated by the data in (2), that the meaning of the former corresponds to the negative quantifier and the latter to the negative property in (4), the principle of compositionality allows us to derive the formula in FIGURE 1.1 as the meaning of (1b). The translation and the predicate logic formula in (1b), however, indicate that the Romanian sentence has a different interpretation, with only one negation. (1b) is synonymous with the English sentence (1a) 'Nobody came'. If we interpret (1a) we easily get the derivation in FIGURE 1.2 and the right interpretation with one negative operator.

The interpretation in FIGURE 1.2 is the one that we need for the Romanian sentence (1b) as well. The problem is that the Romanian sentence contains two negative expressions instead of one. To make it match the structure in FIGURE 1.2 we have to hypothesize that one of the two expressions is not negative, which is contrary to what the data in (2) suggest. Alternatively, we have to find a different mode of composition which yields the interpretation in FIGURE 1.2 from input expressions similar to those in FIGURE 1.1. As we will see in this thesis, this is not a trivial matter.

This conflict between the compositionally derived meaning (FIGURE 1.1) and the actual interpretation (FIGURE 1.2) of a NC sentence like (1b) illustrates the challenge that NC constructions pose to linguistic theory.

Two solutions: NPI vs. NQ approaches Comparing the Romanian and the English data in (1b) and (1c) with respect to the principle of compositionality, there are two points where the analysis for Romanian could differ from that for English: (1) the initial assignment of a negative meaning to the parts or (2) the function by which the two negative parts are composed. Let us consider each option in turn.

In the first case, a thorough empirical investigation is needed to determine if the n-word and the negative marker are indeed negative, that is, if they both contribute semantic negation in the contexts where they occur. If we can conclude that only one of them is truly negative, the compositionality problem is solved, as we can derive the interpretation by a mechanism similar to that in FIGURE 1.2.

In NC languages the NM expresses sentential negation alone:

(6) a. Mario **non** è venuto.
Mario NM is come
'Mario hasn't come.' (Italian)

b. Ion **nu** a venit.
Ion not has come
'John didn't come.' (Romanian)

This does not hold of n-words, which at least in some environments require the occurrence of the NM to make the sentence grammatical:[3]

(7) a. Mario *(**non**) ha detto **niente** (a **nessuno**).
Mario NM has said nothing (to nobody)
'Mario didn't say anything (to anybody).' (Italian)

b. Ion *(**nu**) a zis **nimic** (**nimănui**).
Ion not has said nothing nobody-Dat.
'John didn't say anything (to anybody).' (Romanian)

The data in (6) clearly indicate that the NM bears semantic negation independently of n-words. Thus it is reasonable to assume a uniform negative semantics for the NM in all the contexts, including (7). It remains to be determined whether n-words in (1b) and (7) are indeed negative.

A simple way to put NC constructions in accord with the principle of compositionality is to start with the hypothesis that n-words are non-negative. The approaches that adopt this idea usually assume that n-words are negative polarity items (NPIs) like *anything* in the English translations in (7).[4] With this assumption the NM remains the only negative component in (1b) and (7), and no compositionality problem arises. Laka (1990) is the first to take up this option in an extensive study. Ladusaw's (1992) more fine-grained approach sets the basis for a rich tradition of linguistic studies that account for NC as an instance of negative polarity.

If the empirical investigation leads to the conclusion that n-words are semantically negative just like the NM, the solution is to replace the functional application mechanism in FIGURE 1.1 by one that derives only one negation when composing two negative expressions. This direction of analysis is introduced in Zanuttini's (1991) approach to Italian, continues in Haegeman (1995), Haegeman and Zanuttini (1996), and more recently also in de Swart and Sag (2002) and Richter and Sailer (2004).

The two options described above have developed into the two main directions in the literature on NC. I will refer to the studies that take the first line of analysis as the "NPI approaches", and to the ones following the second as the "negative quantifier (NQ) approaches". For Romanian, I will argue in Chapter 3 that the NQ analysis is empirically more adequate.

[3]Parentheses express optionality, and the star outside them indicates that optionality is ungrammatical, so what is between the parentheses is obligatory.

[4]Ladusaw (1980) and Linebarger (1980) use the term NPI extensively. It is employed here for (non-negative) indefinites restricted to appear within the scope of a negative (or negative-like) operator.

1.3 The contribution of this thesis

This thesis is an NQ approach to NC and has both empirical and theoretical contributions. From an empirical point of view, it enriches the linguistic literature with an extensive investigation of Romanian n-words and NC constructions, on the one hand, and offers a more refined explanation for the dual behavior of n-words with crosslinguistic implications, on the other hand. I reject the analysis of n-words as NPIs on the basis of the fact that Romanian n-words and NC constructions lack the crucial characteristics of NPIs and their relation to the semantic licenser. Even in contexts where they occur without a NM, Romanian n-words exhibit anti-additive properties which qualify them as semantically negative. Moreover, the NM does not show anti-additivity over n-words, while it does over NPIs. This indicates that it semantically licenses NPIs, but not n-words. The availability of a DN reading with two cooccurring n-words and the similarity between their scope properties and those of true quantifiers are taken as further evidence for their negative quantifier status. A close investigation of other empirical tests provided by NPI approaches against the NQ status of n-words indicates that they are actually compatible with the claim in this thesis, if we regard negative quantifiers as a subclass of weak quantifiers (Milsark (1974)).

The theoretical contribution of this thesis is the elaboration of a systematic syntax-semantics interface for the core properties of Romanian n-words and NC. This is also an example of how we can account for NC in natural language in general if we maintain the assumption that n-words are negative quantifiers. I follow de Swart and Sag's (2002) proposal for French to analyze NC as a *resumptive negative quantifier* in an Extended Generalized Quantifier Theory (van Benthem (1989), Hamm (1989), May (1989), Keenan (1992), Keenan and Westerståhl (1997), Peters and Westerståhl (2006)). N-words and the NM are assumed to contribute a generalized negative quantifier NO of Lindström type $\langle 1, 1 \rangle$ and $\langle 0 \rangle$, respectively (Lindström (1966)). As they all contribute quantifiers with the same operator NO, a sequence of k n-words and one NM (the typical NC pattern) together can build a resumptive polyadic quantifier NO^k of type $\langle 1^k, k \rangle$, which binds k variables. The negative semantics is thus contributed only once, independently of how many n-words are involved, and we obtain the NC interpretation of sentences like (1b) and (7b). Alternatively, the monadic quantifiers NO can be combined by *iteration*, which gives us the same result as functional application.

While de Swart and Sag (2002) remains mainly programmatic with respect to the compositionality problem, I further investigate the feasibility of their suggestion to define a mode of composition called *resumption*, an alternative to functional application, that constructs resumptive polyadic quantifiers from monadic ones. I show that this operation contravenes the traditional combinatorics provided by a functional type theory with λ-calculus exemplified in FIGURE 1.1 and 1.2. Therefore, resumption cannot be formulated as a mode of composition. To offer a syntax-semantics interface for resumptive negative quantifiers, I give a logical syncategorematic definition of NO^k. Instead of defining a resumption operation, I make direct use of a k-ary resumptive (negative) quantifier. This quantifier is further integrated in Lexical Resource Semantics (LRS, Richter and Sailer (2004)), an underspecified semantics theory for the constraint-based framework of Head-driven Phrase Structure Grammar (HPSG, Pollard and Sag (1994)). LRS replaces the traditional techniques of combining syntactic expressions with a constraint-based combinatorics that observes the surface constituent structure and the well-typing of logical formulae. This allows a straightforward integration of a resumptive quantifier NO of an underspecified complexity (type) without major adjustments to the grammar. We can thus account for a core sample of Romanian NC constructions, the locality conditions on the scope of n-words, their interaction with non-negative quantifiers, as well as for the semantic and information structure conditions on DN readings.

1.4 Overview

The thesis is organized in five thematic chapters and conclusions as follows.

Chapter 2, *Theoretical background*, is a preliminary presentation of the theoretical frameworks and the empirical domain of the thesis. In Section 2.1 I describe the main assumptions of the Extended Generalized Quantifier Theory that will be used in the analysis. I introduce the so-called *polyadic lifts iteration*, *resumption*, *cumulation* and *different/ same* quantifiers as distinct semantic mechanisms to interpret a sequence of monadic quantifiers in sentences with two or more quantificational NPs. Section 2.2 contains a general characterization of Romanian to familiarize the reader with the empirical domain. I address those properties of Romanian that concern inflection, agreement and word order. Section 2.3 is a description of HPSG, the grammatical framework in which I develop the syntax-semantics interface for NC constructions. I give a small HPSG grammar for Romanian that will later be enriched with the analysis of NC.

Chapter 3, *The semantic status of Romanian n-words*, describes the empirical phenomena that motivate the choice for an NQ approach to n-words and NC in Romanian. I first show that the semantic behavior of n-words evidences their negative content which makes an NPI approach undesirable for NC. Moreover, n-words have scope properties that closely resemble those of true quantifiers and thus further support their negative quantifier status. I also investigate the scope interaction between two negative quantifiers and a non-negative one and its effects on the NC/ DN interpretation. The conclusion is that negative quantifiers in NC have idiosyncratic scope properties similar to cumulative polyadic quantifiers. This motivates a treatment of NC in terms of a polyadic quantifier as proposed in the following chapter.

Chapter 4, *Romanian NQs and NC. Towards a syntax-semantics*, has two parts: 1) a semantic analysis of Romanian NC and DN readings with polyadic quantifiers and 2) an investigation of the status of polyadic lifts in a compositional grammar. I first show that the DN reading of two cooccurring n-words can be obtained if we apply iteration to the two monadic negative quantifiers, and NC if we apply resumption instead. In further support of a polyadic approach to Romanian negation, the scope properties of the negative quantifiers in DN and NC readings are shown to match the general scope behavior of the monadic parts in polyadic quantifiers derived by iteration and resumption, respectively. Second, I investigate the possibility of defining resumption as a mode of composition. I develop a small compositional fragment for Romanian in which I show that resumption and polyadic lifts in general cannot be defined as modes of composition. This is because the traditional notion of compositionality assumes a functional type theory with λ-calculus which is used to imitate the constituent structure of natural language, and polyadic lifts, formulated in a relational type theory, cannot be captured with this combinatorics. The question that arises is how to develop a syntax-semantics interface for NC as a resumptive negative quantifier, if resumption cannot be compositional.

Chapter 5, *The HPSG analysis of Romanian NC: An LRS account*, offers a solution and proposes a syntax-semantics interface for Romanian NC in HPSG. I use an extensional higher-order representation language $Ty1$ in which I define a k-ary resumptive negative quantifier. LRS is an underspecified semantics framework and allows a direct integration of this resumptive quantifier in the grammar by formulating the right constraints consistent with its logical representation. I account for NC constructions by allowing k negative quantifiers contributed by n-words to identify their list of variables, restrictions and the nuclear scope. This means that all the negative quantifiers end up as one and the same k-ary resumptive negative quantifier. Alternatively, two negative quantifiers can stay separate, one taking scope over the other, and yield the DN reading. The next step of the analysis concerns the NM which is shown to always contribute negative semantics and to fix the scope of the negative quantifiers in NC. I thus offer an account of the locality conditions on the interpretation of n-words that occur in embedded subjunctive clauses. While this analysis is not meant to exhaustively describe n-words and NC in Romanian, it proposes a systematic syntax-semantics that accounts for basic NC constructions, the NM as sentential negation and its relation to n-words, as well as the essential properties of DN readings with n-words.

Chapter 6, *Comparison to previous approaches*, is a survey of other approaches to NC in comparison to the one in this thesis. I first consider some NPI approaches and then alternative NQ analyses. I show that my claim that n-words are negative quantifiers is consistent with several other empirical tests that the NPI approaches employ in support of their *non-negative* semantics. Furthermore, I argue that the systematicity of the present analysis makes it preferable to other non-compositional accounts for NC as for instance those making use of negation factorization. Cooper storage, a semantic mechanism usually employed to underspecify quantifier scope interaction, is shown to be unable to integrate resumption compositionally for the same reasons as the compositional grammar in Chapter 4. This makes LRS the only semantic framework of the ones considered here that can integrate resumptive quantifiers in a systematic syntax-semantics interface.

Chapter 7, *Conclusion and perspectives*, summarizes the results in this thesis and presents suggestions for future research. In particular, it calls attention to the primary reasons why we need compositionality. We generally need a compositional mechanism that allows us to account for the systematicity of meaning composition in natural language. The principle of compositionality is our mechanism at the moment and it has been successful in numerous applications. But if natural language challenges it, we should not force the empirical facts to fit our theoretical concept. We should rather reformulate the mechanism to correctly characterize the natural language, at the same time keeping the previous results. NC and other natural language polyadic quantifiers challenge our traditional principle of compositionality which is most likely in need of adjustment. LRS is a framework that allows us to account for the phenomena analyzed in compositional grammars and to also integrate NC and polyadic quantifiers. Thus it could be taken as an indicator of how we should reformulate our mechanism for compositionality.

Chapter 2

Theoretical background

The aim of this chapter is to set the theoretical background for the account of NC that will be developed in the subsequent chapters. Three main aspects are taken into consideration here: 1) the semantic framework within which NC can be accounted for, 2) the empirical domain: Romanian, and 3) the linguistic theory which can integrate the semantics in a syntactic framework. The first component is provided by the theory of *Polyadic Quantifiers* developed among others in Keenan and Westerståhl (1997). This is presented in Section 2.1. Section 2.2 is a short description of Romanian and Section 2.3 introduces Head-driven Phrase Structure Grammar (HPSG) of Pollard and Sag (1994) as the syntax-semantics framework.

2.1 Polyadic quantifiers

My account of Romanian NC continues the linguistic tradition of the "NQ approaches" mentioned in Section 1.2, more precisely, the line in de Swart and Sag (2002). The semantic apparatus is an extension of the *Generalized Quantifier Theory* usually referred to as the *Extended Generalized Quantifier Theory* or *Polyadic Quantifiers* (Keenan and Westerståhl (1997), Peters and Westerståhl (2006), a.o.).

In this section, I present the background assumptions of the Generalized Quantifier Theory and the way they are extended to polyadic quantifiers. For now, I am only concerned with the semantics of generalized quantifiers, so I do not provide a full logical language with a syntax. This will be done at a further stage, when I integrate polyadic quantifiers in a logical language that will be needed for the analysis of Romanian negative concord (Chapters 4 and 5).

Section 2.1.1 is a presentation of polyadic quantifiers as a complex extension of generalized quantifiers. In Sections 2.1.2 and 2.1.3, I introduce the operations *iteration*, *cumulation*, *resumption*, also called *polyadic lifts*, and *different/ same* quantifiers as polyadic quantifiers derived from monadic generalized quantifiers. In Section 2.1.4 I discuss the expressive power of these operations and their potential to be "reduced" to iteration.

2.1.1 Preliminaries

The main concern of the *Generalized Quantifier Theory* (GQT) – first formulated in Barwise and Cooper (1981) – is the semantic interpretation of NPs like the ones italicized in (8):

(8) a. *Everybody/ John* came/ worked hard.

b. *Every student/ No doctor/ Three students* came/ worked hard.

All these NPs combine with *one-place predicates* expressed by intransitive verbs like *came* and *worked hard* to form sentences. One-place predicates denote properties of individuals. So, given a domain E of individuals, one-place predicates denote subsets of E, i.e. sets of individuals carrying the same property (e.g. the property of coming or that of working hard). Sentences denote truth values: either 1 (*true*), or 0 (*false*).

If John is among the individuals who have the property of coming, the sentence *John came* is true; otherwise, it is false. This is the way a sentence is interpreted in first-order predicate logic, where John is represented as an individual constant j and the property is predicated of this constant: see (9a), where $[\![x]\!]$ stands for the denotation of x.[1] But in a higher-order logic, John may be represented as a second-order function (i.e. a set of properties) which takes the property as an argument. In this case, the sentence is true if coming is one of the properties that John has (see (9b)). This latter view, first introduced in Montague (1973), is the one adopted in the Generalized Quantifier Theory and the one I will follow in this thesis:

(9) John came.

a. First-order predicate logic
COME(j)
$[\![\text{COME}(j)]\!] = 1$ iff $[\![j]\!] \in [\![\text{COME}]\!]$

b. Generalized quantifier theory
JOHN(COME)
$[\![\text{JOHN(COME)}]\!] = 1$ iff $[\![\text{COME}]\!] \in [\![\text{JOHN}]\!]$

Thus in GQT terms, NPs like the ones in (8) denote second-order functions over the domain of individuals E: they map properties (subsets of E) onto truth values. This translation of an NP corresponds to the mathematical notion of a *generalized quantifier*, so Barwise and Cooper (1981) refer to NPs as denoting generalized quantifiers.

The NPs in (8a) are usually analyzed as wholes, but within the ones in (8b), the determiners *every, no, three* combine with the common nouns *student* and *doctor* to form NPs. Common nouns, like intransitive verbs, denote properties, so determiners denote functions that map properties onto generalized quantifiers of the kind denoted by NPs. Determiners are thus interpreted as functions from properties to sets of properties: a sentence like *Every student came* is represented as in (10) and is true if coming is a property of every student:

(10) Every student came.
(EVERY(STUDENT))(COME)
$[\![\text{(EVERY(STUDENT))(COME)}]\!] = 1$ iff $[\![\text{COME}]\!] \in [\![\text{(EVERY(STUDENT))}]\!]$

In Montague's tradition Barwise and Cooper (1981) use the term generalized quantifier exclusively for the denotation of NPs. However, following the mathematical tradition based on Lindström (1966), the subsequent linguistic literature refers to both NPs and determiners as denoting generalized quantifiers of different complexity. For this presentation, I adopt this latter position. To distinguish between the two types, I use the terms "NP quantifier" and "Det quantifier". Later in this section, this informal terminology will be replaced by a more precise one following Lindström's mathematical classification of generalized quantifiers.

TABLE 2.1 summarizes the correspondence between linguistic expressions and GQT notions and the notational conventions that will be assumed throughout this presentation. For NP quantifiers,

[1] I follow the common assumption that the language of generalized quantifiers is interpreted in a model M which assigns an interpretation to expressions of the language with respect to a domain E of individuals. M is viewed as the ordered pair $\langle E, [\![\ ]\!] \rangle$, such that it assigns to each expression x an interpretation $[\![x]\!]$.

which are obtained by the application of a Det quantifier to a property, I adopt CONVENTION 2.1 and simplify the notation by leaving out the brackets:

Convention 2.1 *For* Det *a Det quantifier and* N *a common noun, the following holds:*

Det(N) = Det N

Example: EVERY(STUDENT) = EVERY STUDENT

Linguistic expression	Syntactic category	Syntactic representation	Denotation
come	VP	COME	set of individuals (property)
student	N	STUDENT	set of individuals (property)
John	NP	JOHN	set of properties
every student	NP	EVERY STUDENT	set of properties
every	Det	EVERY	function from properties to sets of properties

Table 2.1: Basic assumptions in GQT

Semantics Barwise and Cooper (1981) start with the idea that a sentence of the form $[_S$ NP VP $]$ is true iff the denotation of the VP is a member of the generalized quantifier (see also (10) above). In DEFINITION 2.1, I give the truth conditions for various NP quantifiers. The notation $|\text{A}|$ stands for the cardinality of the set A. I use small caps for subsets of the domain E or other functions on E:

Definition 2.1 Semantics for NP generalized quantifiers
For a domain E, *for every* $\text{A} \subseteq \text{E}$*:*

a. $[\![\text{EVERY}]\!](\text{A}) = \{ \text{X} \subseteq \text{E} \mid \text{A} \subseteq \text{X}\}$

b. $[\![\text{SOME}]\!](\text{A}) = \{ \text{X} \subseteq \text{E} \mid \text{A} \cap \text{X} \neq \emptyset\}$

c. $[\![\text{NO}]\!](\text{A}) = \{ \text{X} \subseteq \text{E} \mid \text{A} \cap \text{X} = \emptyset\}$

d. *For every cardinal number* n *and a corresponding Det quantifier* N,
$[\![\text{N}]\!](\text{A}) = \{ \text{X} \subseteq \text{E} \mid |\text{A} \cap \text{X}| = n\}$

Given the semantics of NP quantifiers and their relation to the corresponding Det quantifiers within their structure, we can also determine the semantic contribution of the latter. Recall that Det quantifiers map properties (common nouns) onto NP quantifiers, which in turn take a property (the VP) to a truth value (the sentence). This perspective on generalized quantifiers is called *functional*, because it reflects the syntactic structure of the sentence: see FIGURE 2.1. Given the semantics of NP quantifiers containing the determiner *every* and assuming that $[\![\text{STUDENT}]\!]$ and $[\![\text{COME}]\!]$ are subsets of the domain E, the sentence in (10) is interpreted as in (11): it is true iff the property of coming contains all the individuals that have the student property:

(11) Every student came.
(EVERY(STUDENT))(COME)
$[\![(\text{EVERY(STUDENT)})(\text{COME})]\!] = 1$

S
(EVERY(STUDENT))(COME)

NP (EVERY(STUDENT)) VP COME

Det EVERY N STUDENT

Figure 2.1: Functional perspective on generalized quantifiers

$$
\begin{aligned}
&\iff \llbracket \text{COME} \rrbracket \in \llbracket (\text{EVERY}(\text{STUDENT})) \rrbracket \\
&\overset{D:2.1}{\iff} \llbracket \text{COME} \rrbracket \in \{\, \textsc{x} \subseteq \textsc{e} \mid \llbracket \text{STUDENT} \rrbracket \subseteq \textsc{x} \} \\
&\iff \llbracket \text{STUDENT} \rrbracket \subseteq \llbracket \text{COME} \rrbracket
\end{aligned}
$$

Leaving aside the hierachical structure and concentrating on a purely set-theoretic perspective, we can view the denotation of a determiner as a function taking two properties to a truth value (see van Benthem (1986*a*,*b*)). Thus a Det quantifier can be regarded as a binary second-order relation. It is binary because it takes two arguments, similarly to a binary relation denoted by a transitive verb like *love*. It is second-order because it does not apply to individuals, but to sets of individuals, i.e. properties. This is the *relational* perspective on generalized quantifiers. The two perspectives (functional and relational) are not in conflict with each other. For instance, the syntactic asymmetry between the object and the subject of the verb *love* is not in conflict with the fact that the verb denotes a binary relation between individuals. In the same way, a determiner denotes a binary relation between properties, independently of the syntactic difference between the common noun and the VP.

In my discussion on generalized quantifiers, I follow Zwarts (1983), van Benthem (1986*a*, 1989), Westerståhl (1989), Keenan (1987, 1992), Keenan and Westerståhl (1997), Peters and Westerståhl (2006) in adopting the *relational* view. In this perspective, determiners denote various binary relations between sets of individuals: *every* denotes the subset relation, *some* the non-empty intersection, *no* the empty intersection and so on, as given in DEFINITION 2.2:

Definition 2.2 Semantics for Det generalized quantifiers
For a domain E*, for every* A, B $\subseteq$ E*:*

a. $\llbracket$EVERY$\rrbracket$(A,B)= *1 iff* A $\subseteq$ B
b. $\llbracket$SOME$\rrbracket$(A,B)= *1 iff* A $\cap$ B$\neq \emptyset$
c. $\llbracket$NO$\rrbracket$(A,B)= *1 iff* A $\cap$ B$= \emptyset$
d. *For every cardinal number* n *and a corresponding Det quantifier* N*,*
$\llbracket$N$\rrbracket$(A,B)= *1 iff* $|$A $\cap$ B$|$ $= n$

Within the relational view, the sentence in (10) is represented and interpreted as in (12). The functional (11) and the relational (12) representation of the sentence have the same truth-conditions:

(12) Every student came.
EVERY(STUDENT, COME)
$\llbracket$EVERY(STUDENT, COME)$\rrbracket$= 1 iff $\llbracket$STUDENT$\rrbracket \subseteq \llbracket$COME$\rrbracket$

The truth-conditional equivalence between the functional and the relational perspective has been formalized in the work of the mathematicians Moses Schönfinkel (see Schönfinkel (1924)) and Haskell

B. Curry (see Curry (1930)).[2] There are two operations by which one can turn a relational representation of a function into a functional representation, and vice versa. These operations are commonly referred to as *currying* and *uncurrying*, respectively. They are given in DEFINITION 2.3 below, adapted from Carpenter (1997, pp. 68–69) to match the set-theoretical notation used here. Here I give the definition with application to Det quantifier functions:

Definition 2.3 $curry/\ uncurry$
For every Q_f *and* Q_r*, the functional, respectively, the relational representation of a Det quantifier, for every* A, B$\subseteq$ E*, the following hold:*

$$curry(\text{Q}_r(\text{A},\text{B})) = (\text{Q}_f(\text{A}))(\text{B})$$
$$uncurry((\text{Q}_f(\text{A}))(\text{B})) = \text{Q}_r(\text{A},\text{B})$$

For every functional expression α*, and every relational expression* β*, the following hold:*

$$curry(uncurry(\alpha)) = \alpha$$
$$uncurry(curry(\beta)) = \beta$$

The *curry/ uncurry* functions defined above allow us to freely switch between the functional and the relational representation of a Det quantifier. As indicated by the different subscripts in DEFINITION 2.3, Q_f and Q_r are not exactly the same, since they have different domains and co-domains: Q_f takes one property and returns a set of properties, while Q_r takes two properties and returns a truth value. However, there is a one-to-one correspondence between them in terms of truth conditions, since Q_r is the set of pairs (A, B), such that $(\text{Q}_f(\text{A}))(\text{B}) = 1$, and conversely, $(\text{Q}_f(\text{A}))(\text{B}) = 1$ iff $(\text{A}, \text{B}) \in \text{Q}_r$ (see also Gamut (1991, Vol. 2, pp. 85, 228)). In view of this correspondence between the relational and the functional representation, already apparent from DEFINITION 2.1 and DEFINITION 2.2, I use the same notation for both the relational and the functional quantifier. This means that in general instead of Q_r or Q_f I will simply use Q, for any quantifier Q defined on the domain E. Whether it is the relational or the functional one can be determined by examining the arguments it takes.

With respect to the quantifiers in DEFINITION 2.1 and DEFINITION 2.2, the *curry/ uncurry* functions in DEFINITION 2.3 allow us to formulate the correspondence between the relational and the functional representation as in LEMMA 2.1:

Lemma 2.1 Semantic correspondence between functional and relational Det quantifiers

For a domain E*, for every* A, B $\subseteq$ E*:*

a. $([\![\text{EVERY}]\!](\text{A}))(\text{B}) = 1$ iff $\text{B} \in \{\text{X} \subseteq \text{E} \mid \text{A} \subseteq \text{X}\} \Leftrightarrow [\![\text{EVERY}]\!](\text{A}, \text{B}) = 1$ iff $\text{A} \subseteq \text{B}$
b. $([\![\text{SOME}]\!](\text{A}))(\text{B}) = 1$ iff $\text{B} \in \{\text{X} \subseteq \text{E} \mid \text{A} \cap \text{X} \neq \emptyset\} \Leftrightarrow [\![\text{SOME}]\!](\text{A}, \text{B}) = 1$ iff $\text{A} \cap \text{B} \neq \emptyset$
c. $([\![\text{NO}]\!](\text{A}))(\text{B}) = 1$ iff $\text{B} \in \{\text{X} \subseteq \text{E} \mid \text{A} \cap \text{X} = \emptyset\} \Leftrightarrow [\![\text{NO}]\!](\text{A}, \text{B}) = 1$ iff $\text{A} \cap \text{B} = \emptyset$
d. For every cardinal number n *and a corresponding Det quantifier N,*
$([\![\text{N}]\!](\text{A}))(\text{B}) = 1$ iff $\text{B} \in \{\text{X} \subseteq \text{E} \mid |\text{A} \cap \text{X}| = n\} \Leftrightarrow [\![\text{N}]\!](\text{A}, \text{B}) = 1$ iff $|\text{A} \cap \text{B}| = n$
e. For every Det quantifier Q*,*
$((\text{Q}(\text{A}))\text{B}) = 1 \Leftrightarrow \text{Q}(\text{A}, \text{B}) = 1$

[2] According to Hindley and Seldin (2008), this idea was already present in Frege (1893, Vol 1, Sec. 4). Thanks to Janina Radó and Frank Richter for mentioning this to me.

Most of the discussion on generalized quantifiers in this thesis will be formulated within the relational perspective. However, the representation of generalized quantifiers within a model-theoretic semantics based on lambda-calculus with functional types will require that I switch to a functional representation in Section 4.3 and Chapter 5.

2.1.1.1 Monadic vs. polyadic quantifiers

Our discussion so far has concerned *monadic* (or *unary*) *quantifiers* whose arguments are *properties* that can be viewed as *unary relations* over the domain of individuals. Unary relations are denoted by linguistic expressions corresponding to common nouns (*student*) or intransitive verbs (*came*, *worked hard*). The NPs in (8) appear as subjects of intransitive verbs, so they denote monadic quantifiers taking unary relations to truth values. But NPs can also appear as direct and indirect objects of transitive and ditransitive verbs like in (13):

(13) a. *Every student* read *some book.*
b. *Three teachers* gave *every student some book.*

Unlike *came* and *worked hard* in (8), *read* and *gave* denote a binary and a ternary relation, respectively. The standard way (in the tradition of Montague (1973)) to interpret these sentences is a functional one in which the relation denoted by the verb applies to each NP quantifier in turn to derive the truth conditions of the proposition.[3] By contrast, in the GQT literature the relational view is used for these sentences as well. Thus we can think of all the NPs in each sentence in (13) as denoting one complex quantifier which maps the binary/ ternary relation onto a truth value. In GQT syntax, we represent the two sentences as in (14), where I again employ CONVENTION 2.1 with NP quantifiers:

(14) a. Every student read some book.
(EVERY STUDENT, SOME BOOK)(READ)
b. Three teachers gave every student some book.
(THREE TEACHER, EVERY STUDENT, SOME BOOK)(GIVE)

Generalized quantifiers like those in (14), which apply to arguments more complex than unary relations, are called *polyadic quantifiers*. In particular, (EVERY STUDENT, SOME BOOK) is a binary quantifier because it maps binary relations to truth values. (THREE TEACHER, EVERY STUDENT, SOME BOOK) is a ternary quantifier.

Relations Before going into the discussion on polyadic quantifiers and their relation to monadic quantifiers, we need to clarify the status of their arguments. We represented unary relations/ properties as sets of individuals from the domain E, in short, as subsets of E. Binary and ternary relations are sets of pairs (3-tuples) of individuals from the domain E. This is to say that a binary relation is a subset of the *Cartesian product* $\mathrm{E} \times \mathrm{E}$ and a ternary relation is a subset of the Cartesian product $\mathrm{E} \times \mathrm{E} \times \mathrm{E}$.

Definition 2.4 n-ary Cartesian product
For a domain E, $\mathrm{X}_1, \mathrm{X}_2, ..., \mathrm{X}_n \subseteq \mathrm{E}$, $n \in \mathbb{N}$, *the Cartesian product of* $\mathrm{X}_1, \mathrm{X}_2, ..., \mathrm{X}_n$ *is:*
$\mathrm{X}_1 \times \mathrm{X}_2 \times ... \times \mathrm{X}_n = \{(x_1, x_2, ..., x_n) | \, x_1 \in \mathrm{X}_1$ *and* $x_2 \in \mathrm{X}_2$ *and ... and* $x_n \in \mathrm{X}_n\}$

The notion of a Cartesian product allows us to define relations as sets of ordered tuples of individuals from the domain E. TABLE 2.2 shows the correspondence between linguistic expressions, their syntactic category, and their denotation as relations.

[3] This is possible provided a type-shifting mechanism is applied to the translation of the verb, so that it can take the NP quantifier as its argument. See Section 4.3.2.4, for an example.

Linguistic expression	Syntactic category	Denotation	Subset of
John/ Every student came	sentence	0-ary relation (proposition)	E^0
come, work	intransitive verb	unary relation (property)	$\mathrm{E}= \mathrm{E}^1$
student	common noun	unary relation (property)	$\mathrm{E}= \mathrm{E}^1$
read, love	transitive verb	binary relation	$\mathrm{E} \times \mathrm{E}= \mathrm{E}^2$
give	ditransitive verb	ternary relation	$\mathrm{E} \times \mathrm{E} \times \mathrm{E}= \mathrm{E}^3$
–	–	n-ary relation	$\underbrace{\mathrm{E} \times \mathrm{E} \times \ldots \times \mathrm{E}}_{n-times}= \mathrm{E}^n$

Table 2.2: Relations

Given that relations of arity n are subsets of the n-ary Cartesian product of the domain E, we can think of unary relations as subsets of the unary Cartesian product of the domain, which is E itself. This coincides with our initial representation of a property but it has the advantage that it can be integrated in the general picture of n-ary relations and their status with respect to the domain E.

Another way of viewing relations is by making appeal to the set of all subsets of E as the *power set* of E, written as P(E) and defined below:

Definition 2.5 Power set
Given a set A*, the power set of* A *is the set of all subsets of* A*:* P(A)= $\{$X$|$ X $\subseteq$ A$\}$

Lemma 2.2 *For every set* A*,* $n \in \mathbb{N}$ *such that* $|$A$|$= n, $|$P(A)$|$= 2^n.

Example:
For a set A= {a, b}, P(A)= {{a, b}, {a}, {b}, { }};
$|$E$|$= 2, so $n = 2$ and $|$P(E)$|$= $2^2 = 4$.

The power set of a non-empty set A contains at least the set A and the empty set. With the notion of the power set of a set, we can define relations as elements of the power set of a Cartesian product of E. For instance, unary relations are elements of P(E^1), binary relations are elements of P(E^2) and n-ary relations are elements of P(E^n). In this thesis, I will occasionally make use of both ways of viewing relations.

In the table above, note that the general representation of n-ary relations allows us to view propositions (i.e. the denotation of sentences) as 0-ary relations and thus subsets of the empty Cartesian product E^0. The set E^0contains only one element, the empty tuple: i.e. E^0= $\{()\}$. As a subset of E^0, a proposition can be either the set $\{()\}$, or $\{\}$, given that P(E^0)= $\{\{()\}, \{\}\}$. In the former case the proposition is true, in the latter, it is false. In the linguistic literature, a true proposition is usually represented as equal to 1 and a false one as equal to 0, so the following convention is usually adopted (see for instance Keenan (1992)):

Convention 2.2 *For any domain* E*, the power set of the set* E^0= $\{()\}$ *is the set of truth values, i.e. we have the following convention:*

$$\mathrm{P}(\mathrm{E}^0) = \{\{()\}, \{\}\} = \{1, 0\}, \textit{ and thus } \{()\} = 1 \textit{ and } \{\} = 0.$$

Here, I will use the latter notation which is more common in the literature and thus we will view a proposition as an element of the set $\{1, 0\}$.

2.1.1.2 Classification of generalized quantifiers

Extending the domain of generalized quantifiers with polyadic quantifiers requires a rigorous system within which one may characterize the properties and the complexity of each kind of quantifier. In this respect, the linguistic framework of polyadic quantifiers follows the mathematical tradition of Mostowski (1957) and Lindström (1966).

Within Lindström's classification, monadic NP quantifiers (e.g. EVERY STUDENT in (10)) are categorized as type $\langle 1 \rangle$ generalized quantifiers, binary NP quantifiers (e.g. (EVERY STUDENT, SOME BOOK) in (14a)), as type $\langle 2 \rangle$, ternary NP quantifiers (e.g. (THREE TEACHER, EVERY STUDENT, SOME BOOK) in (14b)), as type $\langle 3 \rangle$ and n-ary NP quantifiers in general, as type $\langle n \rangle$. This classification is meant to indicate that these quantifiers are functions that map one relation of arity one, two, three, and n, respectively, to a truth value. In more precise terms, their domain is P(E^1), P(E^2), P(E^3), or P(E^n), respectively, and their co-domain is P(E^0).

NP quantifiers (monadic or polyadic) take one argument relation to a truth value, so their type contains only one digit. This is in contrast with Det quantifiers which take at least two arguments, as we saw for instance in the case of EVERY in (12) which maps two arguments (STUDENT and COME) onto a truth value. This means that in Lindström's classification, the type of EVERY has two digits. Since both arguments are unary relations, it is $\langle 1, 1 \rangle$.

As shown above with respect to (14), polyadic NP quantifiers are made up of several monadic NP quantifiers viewed as building a complex quantifier together. The binary quantifier (EVERY STUDENT, SOME BOOK) is made up of the monadic EVERY STUDENT and SOME BOOK and (THREE TEACHER, EVERY STUDENT, SOME BOOK) contains the monadic NP quantifiers THREE TEACHER, EVERY STUDENT, and SOME BOOK. If we write the monadic NP quantifiers as Det quantifiers applying to a unary relation, we can represent the type $\langle 2 \rangle$ quantifier (EVERY STUDENT, SOME BOOK) as (EVERY, SOME)(STUDENT, BOOK) and the type $\langle 3 \rangle$ quantifier (THREE TEACHER, EVERY STUDENT, SOME BOOK) as (THREE, EVERY, SOME)(TEACHER, STUDENT, BOOK). The binary Det quantifier (EVERY, SOME) that we obtain is a function that takes three arguments to a truth value: the first two arguments are unary relations (i.e. STUDENT and BOOK), the third argument is the binary relation READ. Its type is $\langle 1, 1, 2 \rangle$. The ternary Det quantifier (THREE, EVERY, SOME) is a function that takes four arguments to a truth value: three unary relations (i.e. TEACHER, STUDENT, BOOK) and one ternary relation (GIVE). Its type is $\langle 1, 1, 1, 3 \rangle$.

In Lindström's general typing system, the type of a polyadic quantifier is given by a sequence of natural numbers. The number of arguments of the quantifier is the same as the length of this sequence. The last argument is expressed by a verb, the other ones are common nouns. Lindström's classification thus provides a uniform treatment of all natural language quantifiers as functions, regardless of their syntactic position. In order to distinguish the restrictions of a polyadic quantifier from its nuclear scope, sometimes angle brackets are used. Instead of $\langle 1, 1, 2 \rangle$ or $\langle 1, 1, 1, 3 \rangle$, one may write $\langle\langle 1, 1 \rangle, 2 \rangle$ and $\langle\langle 1, 1, 1 \rangle, 3 \rangle$. If all the restrictions of the monadic Det quantifiers are of the same arity like in the two cases above, we write the number of restrictions as a superscript of the arity of the relations involved: e.g. $\langle 1^2, 2 \rangle$ and $\langle 1^3, 3 \rangle$.

We call binary/ ternary/ n-ary a quantifier which takes a binary/ ternary/ n-ary relation as an argument, independently of how many other arguments of a lower arity the quantifier takes. Thus a type $\langle 1, 1, 2 \rangle$ and a type $\langle 2 \rangle$ quantifier are both binary, since the most complex relation they take as an argument is a binary one. The same holds of ternary and n-ary quantifiers in general. So, unlike in the case of relations, whose arity is given by the number of arguments they take (see TABLE 2.2), the complexity of a polyadic quantifier is not given by the number of the arguments, but by the greatest arity of their arguments.

Quantifier (Q)	Component Qs (CQ)	Type of CQ	Type of Q	Domain of Q
propositional operators (negation)	–	–	$\langle 0 \rangle$	P(E^0)
EVERY STUDENT	–	–	$\langle 1 \rangle$	P(E^1)
(EVERY STUDENT, SOME BOOK)	EVERY STUDENT SOME BOOK	$\langle 1 \rangle$ $\langle 1 \rangle$	$\langle 2 \rangle$	P(E^2)
(THREE TEACHER, EVERY STUDENT, SOME BOOK)	THREE TEACHER EVERY STUDENT SOME BOOK	$\langle 1 \rangle$ $\langle 1 \rangle$ $\langle 1 \rangle$	$\langle 3 \rangle$	P(E^3)
(EVERY, SOME)	EVERY SOME	$\langle 1, 1 \rangle$ $\langle 1, 1 \rangle$	$\langle 1^2, 2 \rangle$	P(E^1) $\times$ P(E^1) $\times$ P(E^2)
(THREE, EVERY, SOME)	THREE EVERY SOME	$\langle 1, 1 \rangle$ $\langle 1, 1 \rangle$ $\langle 1, 1 \rangle$	$\langle 1^3, 3 \rangle$	P(E^1) $\times$ P(E^1) $\times$ P(E^1) $\times$ P(E^3)
(NP_1, NP_2, ..., NP_n)	NP_1 NP_2 ... ... NP_n	$\langle 1 \rangle$ $\langle 1 \rangle$ ... ... $\langle 1 \rangle$	$\langle n \rangle$	P(E^n)
(Det_1, Det_2, ..., Det_n)	Det_1 Det_2 ... ... Det_n	$\langle 1, 1 \rangle$ $\langle 1, 1 \rangle$ ... ... $\langle 1, 1 \rangle$	$\langle 1^n, n \rangle$	P(E^1)$_1$ $\times$ P(E^1)$_2$ $\times$ ... $\times$ P(E^1)$_{n-1}$ $\times$ P(E^n)$_n$

Table 2.3: Types of generalized quantifiers

In TABLE 2.3, I summarize the classification of the quantifiers that we discussed. I give the natural language quantifiers with their monadic components, their type, and the domain of definition. The co-domain for each of them is the power set P(E^0), i.e. $\{0, 1\}$, the set of truth values. The type of a polyadic quantifier is obtained by adding up the complexity of its quantifier components. The type $\langle 2 \rangle$ quantifier (EVERY STUDENT, SOME BOOK) is made up of two quantifiers of type $\langle 1 \rangle$: EVERY STUDENT and SOME BOOK. Similarly, the complexity of the type $\langle 1^2, 2 \rangle$ quantifier (EVERY, SOME) reflects the fact that it contains two type $\langle 1, 1 \rangle$ quantifiers: EVERY and SOME. In this thesis, we will only discuss polyadic quantifiers that are derived from monadic quantifiers, so we may extend this classification to cover type $\langle n \rangle$ and type $\langle 1^n, n \rangle$ quantifiers.[4]

Lindström uses this system to also characterize propositional operators, as for instance propositional negation. He considers them generalized quantifiers without a restriction, so they take only one argument, and since the argument is a proposition, i.e. a relation of arity 0 (see table TABLE 2.2), propositional operators are quantifiers of type $\langle 0 \rangle$. This means that both their domain and their co-domain is P(E^0)= $\{0, 1\}$, the set of truth values. In Section 5.5, I will offer an analysis of the Romanian negative marker as a type $\langle 0 \rangle$ quantifier.

[4]But see Keenan and Westerståhl (1997) for more complex examples.

2.1.1.3 Syntactic representations with polyadic quantifiers

In view of the relation between polyadic NP quantifiers and the Det quantifiers within their structure, we may syntactically represent the two sentences in (13) in two different ways: with NP polyadic quantifiers and Det polyadic quantifiers. This is illustrated in (15), where I make use of one further notational convention usually adopted in the literature, that of indicating the restriction of a Det quantifier as a superscript. This means that besides CONVENTION 2.1, we have another notation for NP quantifiers to indicate their relation to the Det quantifier. This is given in CONVENTION 2.3 below. The superscript notation of the restriction also appears with polyadic Det quantifiers and is described by CONVENTION 2.4. These conventions will be used here both in syntactic and semantic representations of quantifiers.

Convention 2.3 *For a domain* E*,* Q *a type* $\langle 1, 1\rangle$ *quantifier and* A $\subseteq$ E*, we have the following convention:*

$\text{Q}(\text{A}) = \text{Q A} = \text{Q}^{\text{A}}$

Example*:* EVERY(STUDENT) = EVERY STUDENT = EVERY$^{\text{STUDENT}}$

Convention 2.4 *For a domain* E*,* Q *a type* $\langle 1^n, n\rangle$ *quantifier and* A_1, A_2, ..., $\text{A}_n \subseteq$ E*, we have the following convention:*

$\text{Q}(\text{A}_1,\text{A}_2,\ldots,\text{A}_n) = \text{Q}^{\text{A}_1,\text{A}_2,\ldots,\text{A}_n}$

Example*:* (EVERY, SOME)(STUDENT, BOOK) = (EVERY, SOME)$^{\text{STUDENT, BOOK}}$

(15) a. *Every student* read *some book.*

i. Representations with a type $\langle 2\rangle$ quantifier:
(EVERY STUDENT, SOME BOOK)(READ)
(EVERY$^{\text{STUDENT}}$, SOME$^{\text{BOOK}}$)(READ)

ii. Representations with a type $\langle 1^2, 2\rangle$ quantifier:
(EVERY, SOME)(STUDENT, BOOK, READ)
((EVERY, SOME)(STUDENT, BOOK))(READ)
(EVERY, SOME)$^{\text{STUDENT, BOOK}}$(READ)

b. *Three teachers* gave *every student some book.*

i. Representations with a type $\langle 3\rangle$ quantifier:
(THREE TEACHER, EVERY STUDENT, SOME BOOK)(GIVE)
(THREE$^{\text{TEACHER}}$, EVERY$^{\text{STUDENT}}$, SOME$^{\text{BOOK}}$)(GIVE)

ii. Representations with a type $\langle 1^3, 3\rangle$ quantifier:
(THREE, EVERY, SOME)(TEACHER, STUDENT, BOOK, GIVE)
((THREE, EVERY, SOME)(TEACHER, STUDENT, BOOK))(GIVE)
(EVERY, SOME, THREE)$^{\text{TEACHER, STUDENT, BOOK}}$(GIVE)

So far we discussed the classification of generalized quantifiers as well as their syntactic representation and the relation to their monadic components. The next issue that we are interested in is finding a way to interpret them. As in the structure of polyadic quantifiers one can easily distinguish monadic quantifiers, the main goal of the literature on polyadic quantification (see for instance van Benthem (1986*a*, 1989), Keenan (1987, 1992), Hamm (1989), Westerståhl (1994), Keenan and Westerståhl (1997), and Peters and Westerståhl (2006) a.o.) has been to describe the semantics of polyadic quantifiers on the basis of the semantics of their components. It is usually assumed that the monadic parts undergo some *polyadic operation* or *polyadic lift* which eventually gives the interpretation of the polyadic quantifier. Several such operations and quantifier combinations have been defined in the literature. In the subsequent sections I will concentrate on *iteration*, *different/ same* quantifiers, *cumulation*, and *resumption*. The last one will be used in Chapters 4 and 5 to account for Romanian negative concord.

2.1.2 Iteration

Iteration is the most common operation by which polyadic quantifiers can be derived from monadic ones. For instance, in order to derive the meaning of the polyadic quantifier (EVERY, SOME) in (15a), the two monadic quantifiers EVERY and SOME are composed by means of iteration. In this section, I show how this can be done.

To define iteration, the concept of a monadic quantifier must be slightly extended. Recall from the previous section that a quantifier Q of type $\langle 1\rangle$ maps properties to truth values. Thus Q may be viewed as reducing the arity of a relation by 1: it reduces a unary relation to a 0-ary relation and, in general, it reduces an $n+1$-ary relation to an n-ary relation as in DEFINITION 2.6. Instead of defining Q as Q: P(E^1) $\rightarrow$ P(E^0) like in TABLE 2.3, we can extend this definition to Q: P(E^{n+1}) $\rightarrow$ P(E^n):

Definition 2.6 Monadic quantifiers as 1-arity reducers
Given a universe E, *for* R $\subseteq$ E^{n+1}, $n \in \mathbb{N}$, Q *a type* $\langle 1\rangle$ *quantifier, the following holds:*

$$\text{Q(R)} = \{(a_1, ..., a_n) \in \text{E}^n | \text{Q}(\{b \in \text{E} | (a_1, ..., a_n, b) \in \text{R}\}) = 1\}$$

If a quantifier Q of type $\langle 1\rangle$ combines with a relation R of arity $n+1$, the result is a relation of arity n (a set of $(a_1, ..., a_n)$ tuples), with the property that Q yields truth when applied to each element b, the $(n+1)$-th member of the $(n+1)$-tuples $(a_1, ..., a_n, b)$ in the relation R. The relation R is thus decomposed into two relations: one of arity n (the set of n-tuples $(a_1, ..., a_n)$) and one of arity 1 (the set of b individuals). Monadic Q reduces the unary relation to a truth value (in a way similar to EVERY STUDENT in (10)). The n-ary relation contains all the tuples of n-elements which result from Q being applied to the $(n+1)$-ary relation R.

Let us illustrate DEFINITION 2.6 with a few particular cases. If n= 0, then R is a unary relation and we obtain Q(R)= $\{() \in \text{E}^0 | \text{Q}(\{b \in \text{E} |\ b \in \text{R}\}) = 1\}$. In words, the value of Q(R) is the set of empty tuples in E^0, such that Q yields truth if applied to the set of elements b in R. Note that there is a single empty tuple "()" and the set made up of this element is 1 (see CONVENTION 2.2). Moreover, the set of elements b in R is the unary relation R, itself. Thus the definition simply says that Q(R)= 1 iff Q($\{b|\ b \in \text{R}\}$)= 1, which is a tautology. If we now take $n = 1$, R is a binary relation and we obtain Q(R)= $\{a_1 \in \text{E}^1 | \text{Q}(\{b \in \text{E} |\ (a_1, b) \in \text{R}\}) = 1\}$. The value of Q(R) is a unary relation made up of all the elements a_1, such that Q$(\{b \in \text{E} |\ (a_1, b) \in \text{R}\}) = 1$. An example in which $n = 1$ will be given in (16).

As previously shown, a generalized quantifier Q of type $\langle k\rangle$ reduces a relation of arity k to a 0-ary relation: i.e. Q: P(E^k)$\rightarrow$ P(E^0). But following the model of monadic quantifiers in DEFINITION 2.6, Q can also be regarded as reducing $(n+k)$-ary relations to n-ary relations, so Q: P(E^{n+k})$\rightarrow$ P(E^k), as below:

Definition 2.7 K-ary quantifiers as k-arity reducers
Given a universe E*, for* $\mathrm{R} \subseteq \mathrm{E}^{n+k}$*,* $n \in \mathbb{N}, k \geq 1$*,* Q *a type* $\langle k \rangle$ *quantifier, the following holds:* $\mathrm{Q}(\mathrm{R}) = \{(a_1, ..., a_n) \in \mathrm{E}^n | \mathrm{Q}(\{(b_1, ..., b_k) \in \mathrm{E}^k | (a_1, ..., a_n, b_1, ..., b_k) \in \mathrm{R}\}) = 1\}$

If Q applies to an $(n+k)$-ary relation, the result is an n-ary relation (a set of $(a_1, ..., a_n)$ tuples), with the property that Q yields truth of all the k-tuples $(b_1, ..., b_k)$, such that the $k+n$-tuples $(a_1, ..., a_n, b_1, ..., b_k)$ are members of R.

Note that DEFINITION 2.6 actually represents the particular case of DEFINITION 2.7 where $k = 1$, so that the generalized type $\langle k \rangle$ quantifier is actually a monadic one. But let us concentrate on the value of n. For $n = 0$ in DEFINITION 2.7, the relation R is of arity k so $\mathrm{R} \subseteq \mathrm{E}^k$ and $\mathrm{Q}(\mathrm{R}) = \{() \in \mathrm{E}^0 | \mathrm{Q}(\{(b_1, ..., b_k) \in \mathrm{E}^k | (b_1, ..., b_k) \in \mathrm{R}\} = 1)\}$. Given that $\{() \in \mathrm{E}^0\} = \{()\} = 1$, we again obtain a tautology, namely, $\mathrm{Q}(\mathrm{R})$= 1 iff $\mathrm{Q}(\{(b_1, ..., b_k) | (b_1, ..., b_k) \in \mathrm{R}\}) = 1$. So we are dealing with the situation we already described in TABLE 2.3, where a type $\langle k \rangle$ quantifier reduces a relation of arity k to a truth value. For $n = 1$, R is a $k + 1$-ary relation, and $\mathrm{Q}(\mathrm{R}) = \{a_1 \in \mathrm{E}^1 | \mathrm{Q}(\{(b_1, ..., b_k) \in \mathrm{E}^k | (a_1, b_1, ..., b_k) \in \mathrm{R}\}) = 1\}$. So the value of $\mathrm{Q}(\mathrm{R})$ is the set of elements a_1, such that $\mathrm{Q}(\{(b_1, ..., b_k) \in \mathrm{E}^k | (a_1, b_1, ..., b_k) \in \mathrm{R}\}) = 1$.

As an illustration of the base case (i.e. $n = 0$) in DEFINITION 2.7, the type $\langle 2 \rangle$ quantifier (EVERY STUDENT, SOME$^{\text{BOOK}}$) in (15a) takes the binary relation READ to a truth value and the type $\langle 3 \rangle$ quantifier (EVERY$^{\text{STUDENT}}$, SOME$^{\text{COLLEAGUE}}$, THREE$^{\text{BOOK}}$) in (15b) takes the ternary relation GIVE to a truth value. In these two examples, Q of type k ($k = 2$ and $k = 3$, respectively) applies to a k-ary relation, so the result is always a truth value.

Interpreting polyadic quantifiers Our concern is to interpret sentences like (13) above, to which we associated the polyadic quantifiers in (14)/ (15). But at this point we have no mechanism to interpret polyadic quantifiers, we only have the semantics of monadic quantifiers summarized in LEMMA 2.1. We have seen that polyadic quantifiers are built on the basis of several monadic quantifiers. DEFINITION 2.6 helps us to interpret the sentences in (13) by only making use of the semantics of monadic quantifiers: it allows us to consider in turn each monadic quantifier within a polyadic one. DEFINITION 2.7 is helpful for generalizations with polyadic quantifiers.

For $n = 1$ in DEFINITION 2.6, in (13a) repeated below we can view the monadic quantifier SOME$^{\text{BOOK}}$ as reducing the binary relation READ to a unary relation as in (16a). This relation is then reduced to a truth value via the application of the monadic quantifier EVERY$^{\text{STUDENT}}$, as in (16b), so we can interpret the sentence on the basis of the semantics of the two monadic quantifiers:

(16) *Every student* read *some book.*

a. $[\![\text{SOME}^{\text{BOOK}}]\!]([\![\text{READ}]\!])$
$\overset{D:2.6}{=\!=} \{a_1 \in \mathrm{E}^1 | [\![\text{SOME}^{\text{BOOK}}]\!](\{b \in \mathrm{E}^1 | (a_1, b) \in [\![\text{READ}]\!]\}) = 1\}$

b. $[\![\text{EVERY}^{\text{STUDENT}}]\!]([\![\text{SOME}^{\text{BOOK}}]\!]([\![\text{READ}]\!]))$
$\overset{D:2.6,16a}{=\!=} \{() \in \mathrm{E}^0 | [\![\text{EVERY}^{\text{STUDENT}}]\!](\{a_1 \in \mathrm{E}^1 | [\![\text{SOME}^{\text{BOOK}}]\!](\{b \in \mathrm{E}^1 |$
$(a_1, b) \in [\![\text{READ}]\!]\}) = 1\}) = 1\}$
$\overset{C:2.2}{\Longleftrightarrow}$
$[\![\text{EVERY}^{\text{STUDENT}}]\!]([\![\text{SOME}^{\text{BOOK}}]\!]([\![\text{READ}]\!]))$
$= 1$ iff $[\![\text{EVERY}^{\text{STUDENT}}]\!](\{a_1 \in \mathrm{E}^1 | [\![\text{SOME}^{\text{BOOK}}]\!](\{b \in \mathrm{E}^1 |$

$(a_1, b) \in [\![\text{READ}]\!]\}) = 1\}) = 1$

In a similar way, for the sentence in (13b) we can obtain a truth value that depends on the semantics of the three monadic quantifiers it contains. By DEFINITION 2.6, SOME$^{\text{BOOK}}$ reduces the ternary relation GIVE to a binary relation as in (17a), EVERY$^{\text{STUDENT}}$ further reduces the binary relation to the unary relation in (17b), and this latter relation is mapped onto a truth value, once it becomes the argument of the monadic quantifier THREE$^{\text{TEACHER}}$ as in (17c):

(17) *Three teachers* gave *every student some book*.

a. $[\![\text{SOME}^{\text{BOOK}}]\!]([\![\text{GIVE}]\!])$

$\overset{D:2.6}{=\!=} \{(a_1, a_2) \in \mathrm{E}^2 |\ [\![\text{SOME}^{\text{BOOK}}]\!](\{b \in \mathrm{E}^1 | (a_1, a_2, b) \in [\![\text{GIVE}]\!]\}) = 1\}$

b. $[\![\text{EVERY}^{\text{STUDENT}}]\!]([\![\text{SOME}^{\text{BOOK}}]\!]([\![\text{GIVE}]\!]))$

$\overset{D:2.6,17a}{=\!=} \{a_1 \in \mathrm{E}^1 | [\![\text{EVERY}^{\text{STUDENT}}]\!](\{a_2 \in \mathrm{E}^1 | [\![\text{SOME}^{\text{BOOK}}]\!](\{b \in \mathrm{E}^1 |$
$(a_1, a_2, b) \in [\![\text{GIVE}]\!]\}) = 1\}) = 1\}$

c. $[\![\text{THREE}^{\text{TEACHER}}]\!]([\![\text{EVERY}^{\text{STUDENT}}]\!]([\![\text{SOME}^{\text{BOOK}}]\!]([\![\text{GIVE}]\!])))$

$= 1$ iff $[\![\text{THREE}^{\text{TEACHER}}]\!]\ (\{a_1 \in \mathrm{E}^1 | [\![\text{EVERY}^{\text{STUDENT}}]\!](\{a_2 \in \mathrm{E}^1 |$
$[\![\text{SOME}^{\text{BOOK}}]\!](\{b \in \mathrm{E}^1 | (a_1, a_2, b) \in [\![\text{GIVE}]\!]\}) = 1\}) = 1\}) = 1$

In conclusion, we can interpret sentences with two or three monadic quantifiers by successively applying the semantics of each quantifier to the argument relation, as suggested by the syntax in DEFINITION 2.6. As indicated in (14) and (15), in GQT these sentences are usually associated with polyadic quantifiers: a binary and a ternary one, respectively. We cannot interpret such polyadic quantifiers as wholes, but DEFINITION 2.6 provides us with a syntax that allows us to interpret them by only making use of the semantics of monadic quantifiers. This gives us a first mechanism to derive the semantics of polyadic quantifiers from that of their component monadic quantifiers. So for the polyadic quantifiers in (15a) and (15b), we have the following interpretation:[5]

(18) a. (EVERY$^{\text{STUDENT}}$, SOME$^{\text{BOOK}}$)(READ)

may be interpreted as

$[\![\text{EVERY}^{\text{STUDENT}}]\!]([\![\text{SOME}^{\text{BOOK}}]\!]([\![\text{READ}]\!]))$

b. (THREE$^{\text{TEACHER}}$, EVERY$^{\text{STUDENT}}$, SOME$^{\text{BOOK}}$)(GIVE)

may be interpreted as

$[\![\text{THREE}^{\text{TEACHER}}]\!]([\![\text{EVERY}^{\text{STUDENT}}]\!]([\![\text{SOME}^{\text{BOOK}}]\!]([\![\text{GIVE}]\!])))$

This way of combining the semantics of the monadic parts to obtain the semantics of a polyadic quantifier is known in the literature as *iteration*. In general, following Keenan and Westerståhl (1997), it is said that the monadic quantifiers have been "lifted" by iteration to a polyadic quantifier. That is, in (18a) and (18b), two/ three quantifiers of type $\langle 1 \rangle$ are lifted to a complex quantifier of type $\langle 2 \rangle$/ $\langle 3 \rangle$, such that the resulting quantifer can take the binary/ ternary relations READ/ GIVE directly to a truth value. Iteration is defined in DEFINITION 2.8 for two monadic quantifiers and a binary relation. The function composition operator "$\circ$" is used to indicate that two quantifiers are "composed by iteration", since iteration is function composition with generalized quantifiers (see Keenan and Westerståhl (1997, pp. 871–873) for further discussion):

[5]In the subsequent sections we will see that this is not the only way to interpret a polyadic quantifier.

Definition 2.8 Iteration of two type $\langle 1 \rangle$ quantifiers
For Q_1, Q_2, *quantifiers of type* $\langle 1 \rangle$, $It(\mathrm{Q}_1, \mathrm{Q}_2)$ *is the type* $\langle 2 \rangle$ *quantifier defined, for any domain* E, *any* $x, y \in \mathrm{E}$, *and any* $\mathrm{R} \subseteq \mathrm{E}^2$, *as:*

$$It(\mathrm{Q}_1, \mathrm{Q}_2)(\mathrm{R}) = (\mathrm{Q}_1 \circ \mathrm{Q}_2)(\mathrm{R}) = \mathrm{Q}_1(\{x \in \mathrm{E}^1 |\ \mathrm{Q}_2(\{y \in \mathrm{E}^1 |\ (x,y) \in \mathrm{R}\}) = 1\})$$

This definition can be extended to two quantifiers of type $\langle n \rangle$ and $\langle k \rangle$ and a relation R of $(n+k)$-arity as in DEFINITION 2.9:

Definition 2.9 Iteration
For any $n, k \in \mathbb{N}$, *for two quantifiers* Q_1 *of type* $\langle n \rangle$, Q_2 *of type* $\langle k \rangle$, $It(\mathrm{Q}_1,\mathrm{Q}_2)$ *is the type* $\langle n{+}k \rangle$ *quantifier defined, for any domain* E, *any* $(a_1, ..., a_n) \in \mathrm{E}^n$, *any* $(b_1, ..., b_k) \in \mathrm{E}^k$, *and any* $\mathrm{R} \subseteq \mathrm{E}^{n+k}$, *as:*

$$\begin{aligned} It(\mathrm{Q}_1,\mathrm{Q}_2)(\mathrm{R}) &= (\mathrm{Q}_1 \circ \mathrm{Q}_2)(\mathrm{R}) \\ &= \mathrm{Q}_1(\{(a_1, ..., a_n) \in \mathrm{E}^n |\ \mathrm{Q}_2(\{(b_1, ..., b_k) \in \mathrm{E}^k | \\ &\qquad (a_1, ..., a_n, b_1, ..., b_k) \in \mathrm{R}\}) = 1\}) \end{aligned}$$

The definitions above give us a direct interpretation for a polyadic quantifier. Take one of the quantifiers to be of type $\langle 0 \rangle$, say $n = 0$ and $k \geq 1$. In this case, $\mathrm{R} \in \mathrm{E}^k$ and we get $It(\mathrm{Q}_1,\mathrm{Q}_2)(\mathrm{R}) = (\mathrm{Q}_1 \circ \mathrm{Q}_2)(\mathrm{R}) = \mathrm{Q}_1(\{() \in \mathrm{E}^0 |\ \mathrm{Q}_2(\{(b_1, ..., b_k) \in \mathrm{E}^k | (b_1, ..., b_k) \in \mathrm{R}\}) = 1\})$. If Q_1 is the negative operator, $\mathrm{Q}_1(\{() \in \mathrm{E}^0 |\ \mathrm{Q}_2(\{(b_1, ..., b_k) \in \mathrm{E}^k | (b_1, ..., b_k) \in \mathrm{R}\}) = 1\}) = 1$ iff $\{() \in \mathrm{E}^0 |\ \mathrm{Q}_2(\{(b_1, ..., b_k) \in \mathrm{E}^k | (b_1, ..., b_k) \in \mathrm{R}\}) = 1\} = \{\} = 0$ (it is only true of a false 0-ary relation). This latter formula holds if and only if $\mathrm{Q}_2(\{(b_1, ..., b_k) \in \mathrm{E}^k | (b_1, ..., b_k) \in \mathrm{R}\}) = 0$ which is equivalent to $\mathrm{Q}_2(\mathrm{R}) = 0$. Thus if one of the two quantifiers in DEFINITION 2.9 is of type $\langle 0 \rangle$, the type of its iteration with another quantifier will have the same type as the latter quantifier. However, the type $\langle 0 \rangle$ quantifier brings its own contribution to the semantics of the iteration. Thus $It(\mathrm{Q}_1,\mathrm{Q}_2)(\mathrm{R}) \neq \mathrm{Q}_2(\mathrm{R})$. In our case, given that we took Q_1 to bear the semantics of the negative operator, $It(\mathrm{Q}_1,\mathrm{Q}_2)(\mathrm{R}) = 1$ iff $\mathrm{Q}_2(\mathrm{R}) = 0$.

If we take both quantifiers in DEFINITION 2.9 to be of type $\langle 0 \rangle$, i.e. $k = n = 0$, then R is a proposition, i.e. $\mathrm{R} \subseteq \mathrm{E}^0$, and we obtain the following: $It(\mathrm{Q}_1,\mathrm{Q}_2)(\mathrm{R}) = (\mathrm{Q}_1 \circ \mathrm{Q}_2)(\mathrm{R}) = \mathrm{Q}_1(\{() \in \mathrm{E}^0 |\ \mathrm{Q}_2(\{() \in \mathrm{E}^0 | () \in \mathrm{R}\}) = 1\})$. To better understand how the semantics works, let Q_1 be an affirmative operator, and Q_2 the negative operator: that is, for every $\mathrm{P} \in \mathrm{E}^0$, $\mathrm{Q}_1(\mathrm{P}) = 1$ iff $\mathrm{P} = 1$, and $\mathrm{Q}_2(\mathrm{P}) = 1$ iff $\mathrm{P} = 0$. By applying the semantics of Q_1, we obtain: $\mathrm{Q}_1(\{() \in \mathrm{E}^0 |\ \mathrm{Q}_2(\{() \in \mathrm{E}^0 | () \in \mathrm{R}\}) = 1\}) = 1$ iff $\{() |\ \mathrm{Q}_2(\{() \in \mathrm{E}^0 | () \in \mathrm{R}\}) = 1\} = \{()\} = 1$. But the latter formula holds if only if $\mathrm{Q}_2(\{() \in \mathrm{E}^0 | () \in \mathrm{R}\}) = 1$ which can be simplified to $\mathrm{Q}_2(\mathrm{R}) = 1$. Thus for Q_1 an affirmative operator, we arrive at $It(\mathrm{Q}_1,\mathrm{Q}_2)(\mathrm{R}) = 1$ iff $\mathrm{Q}_2(\mathrm{R}) = 1$. If we further apply the negative semantics of Q_2, then $It(\mathrm{Q}_1,\mathrm{Q}_2)(\mathrm{R}) = 1$ iff $\mathrm{R} = 0$.

Let us now take an example with two type $\langle 1 \rangle$ quantifiers. By DEFINITION 2.8, the binary quantifier ($\mathrm{EVERY}^{\mathrm{STUDENT}}$, $\mathrm{SOME}^{\mathrm{BOOK}}$) in (15a) can be interpreted as an iteration of the two monadic quantifiers $\mathrm{EVERY}^{\mathrm{STUDENT}}$ and $\mathrm{SOME}^{\mathrm{BOOK}}$ as given in (19b):

(19) a. Every student read some book.
($\mathrm{EVERY}^{\mathrm{STUDENT}}$, $\mathrm{SOME}^{\mathrm{BOOK}}$)(READ)

b. $It([\![\mathrm{EVERY}^{\mathrm{STUDENT}}]\!], [\![\mathrm{SOME}^{\mathrm{BOOK}}]\!])([\![\mathrm{READ}]\!]) = 1$
$\overset{D:2.8}{\Longleftrightarrow} ([\![\mathrm{EVERY}^{\mathrm{STUDENT}}]\!] \circ [\![\mathrm{SOME}^{\mathrm{BOOK}}]\!])([\![\mathrm{READ}]\!]) = 1$
$\overset{D:2.8}{\Longleftrightarrow} [\![\mathrm{EVERY}^{\mathrm{STUDENT}}]\!]\ (\{x \in \mathrm{E} |\ [\![\mathrm{SOME}^{\mathrm{BOOK}}]\!](\{y \in \mathrm{E} |$

$$(x,y) \in [\![\text{READ}]\!]\}) = 1\}) = 1$$
$$\overset{L:2.1b}{\Longleftrightarrow} [\![\text{EVERY}^{\text{STUDENT}}]\!](\{x \in \text{E} |\ [\![\text{BOOK}]\!] \cap \{y \in \text{E}|$$
$$(x,y) \in [\![\text{READ}]\!]\} \neq \emptyset\}) = 1$$
$$\overset{L:2.1a}{\Longleftrightarrow} [\![\text{STUDENT}]\!] \subseteq \{x \in \text{E}|\ [\![\text{BOOK}]\!] \cap \{y \in \text{E}|\ (x,y) \in [\![\text{READ}]\!]\} \neq \emptyset\}$$

If we replace Q_1, Q_2 and R in DEFINITION 2.8 with $[\![\text{EVERY}^{\text{STUDENT}}]\!]$ (the set of properties every student has), $[\![\text{SOME}^{\text{BOOK}}]\!]$ (the set of properties some book has) and $[\![\text{READ}]\!]$ (the set of pairs of elements that are in the read relation), respectively, we obtain the first two equivalences in (19b). The interpretation of the two iterated quantifiers is obtained from the semantics of the two monadic quantifiers SOME and EVERY given in a convenient form in LEMMA 2.1.[6] The interpretation of the sentence *Every student read some book* is that the set of students is a subset of the set of book-readers.

The meaning of the ternary quantifier in (15b) can be derived by iteration in a similar way: we first apply DEFINITION 2.9 to the meaning of the monadic quantifier $\text{THREE}^{\text{TEACHER}}$ and the meaning of the binary one ($\text{EVERY}^{\text{STUDENT}}$, $\text{SOME}^{\text{BOOK}}$). We also interpret the latter via iteration as in (19b) above, so we obtain the meaning of the ternary quantifier by applying iteration twice. As will become clear in Section 2.1.3, iteration is only one of the possible interpretations that can be given for a polyadic quantifier. It is a choice that we make to interpret the quantifiers in (15) by iteration and in the case of the ternary quantifier in (15b) we make this choice twice. At each step, we could choose not to use iteration. But for illustration, we now interpret (15b) only with iteration. The interpretation we obtain for the sentence *Three teachers gave every student some book* is that the cardinality of the set intersection between the set of teachers and the set of students who were given some book is 3:

(20) a. Three teachers gave every student some book.
($\text{THREE}^{\text{TEACHER}}$, $\text{EVERY}^{\text{STUDENT}}$, $\text{SOME}^{\text{BOOK}}$)(GIVE)

b. $It([\![\text{THREE}^{\text{TEACHER}}]\!], It([\![\text{EVERY}^{\text{STUDENT}}]\!], [\![\text{SOME}^{\text{BOOK}}]\!]))([\![\text{GIVE}]\!]) = 1$
$$\overset{D:2.9}{\Longleftrightarrow} ([\![\text{THREE}^{\text{TEACHER}}]\!] \circ It([\![\text{EVERY}^{\text{STUDENT}}]\!], [\![\text{SOME}^{\text{BOOK}}]\!]))$$
$$([\![\text{GIVE}]\!]) = 1$$
$$\overset{D:2.9}{\Longleftrightarrow} [\![\text{THREE}^{\text{TEACHER}}]\!](\{z \in \text{E} \mid It([\![\text{EVERY}^{\text{STUDENT}}]\!], [\![\text{SOME}^{\text{BOOK}}]\!])$$
$$(\{(x,y) \in \text{E}^2 | (z,x,y) \in [\![\text{GIVE}]\!]\}) = 1\}) = 1$$
$$\overset{D:2.8}{\Longleftrightarrow} [\![\text{THREE}^{\text{TEACHER}}]\!](\{z \in \text{E} \mid ([\![\text{EVERY}^{\text{STUDENT}}]\!] \circ [\![\text{SOME}^{\text{BOOK}}]\!])$$
$$(\{(x,y) \in \text{E}^2 | (z,x,y) \in [\![\text{GIVE}]\!]\}) = 1\}) = 1$$
$$\overset{D:2.8}{\Longleftrightarrow} [\![\text{THREE}^{\text{TEACHER}}]\!](\{z \in \text{E} \mid [\![\text{EVERY}^{\text{STUDENT}}]\!](\{x \in \text{E} \mid$$
$$[\![\text{SOME}^{\text{BOOK}}]\!](\{y \in \text{E} | (z,x,y) \in [\![\text{GIVE}]\!]\}) = 1\}) = 1\}) = 1$$
$$\overset{L:2.1}{\Longleftrightarrow} |[\![\text{TEACHER}]\!] \cap \{z \in \text{E} \mid [\![\text{STUDENT}]\!] \subseteq \{x \in \text{E} \mid [\![\text{BOOK}]\!] \cap$$
$$\{y \in \text{E} | (z,x,y) \in [\![\text{GIVE}]\!] \} \neq \emptyset\}\}| = 3$$

A further noteworthy point is that for (20) we obtain the same truth conditions if we compose ($\text{THREE}^{\text{TEACHER}}$, $\text{EVERY}^{\text{STUDENT}}$) with $\text{SOME}^{\text{BOOK}}$. This is because iteration operates like *function composition*, and it is thus *associative* (cf. Keenan and Westerståhl (1997, p. 871), Peters and

[6]In discussing the semantics of generalized quantifiers, I will often make reference to LEMMA 2.1, rather than to DEFINITION 2.1 or DEFINITION 2.2, since the lemma shows the relational/ functional representation of the quantifier to be neutral with respect to its interpretation.

Westerståhl (2006, pp. 349–351)). This is to say that $f \circ (g \circ h) = (f \circ g) \circ h$, for all functions f, g, h. However, function composition and, implicitly, iteration are not *commutative* operations, which means that the order in which the functions are composed influences the result. Thus typically $f \circ g \neq g \circ f$. This brings us to the next topic concerning iteration, which is the scope of the quantifiers as determined by the order in which they are composed. For simplicity, the following theoretical discussion is limited to examples with binary quantifiers.

2.1.2.1 Scope of quantifiers

With iterations the order in which the monadic quantifiers are combined with the relation determines the scope interaction between them: the rightmost quantifier combines first with the relation, so what comes to its left takes wide scope. In (19b) above, only one interpretation is given for (19a), the one in which EVERY outscopes SOME. But the other order is also possible and yields another reading, that in which there is a (specific) book which was read by every student. This is obtained by first applying $\text{EVERY}^{\text{STUDENT}}$ to READ to obtain a unary relation which then becomes the argument of $\text{SOME}^{\text{BOOK}}$:[7]

(21) a. Every student read some book.
(EVERY$^{\text{STUDENT}}$, SOME$^{\text{BOOK}}$)(READ)

b. $It(⟦\text{SOME}^{\text{BOOK}}⟧, ⟦\text{EVERY}^{\text{STUDENT}}⟧)\,(⟦\text{READ}⟧^{-1}) = 1$

$\overset{D:2.8}{\Longleftrightarrow} (⟦\text{SOME}^{\text{BOOK}}⟧ \circ ⟦\text{EVERY}^{\text{STUDENT}}⟧)\,(⟦\text{READ}⟧^{-1}) = 1$

$\overset{D:2.8}{\Longleftrightarrow} ⟦\text{SOME}^{\text{BOOK}}⟧(\{x \in \text{E}|\ ⟦\text{EVERY}^{\text{STUDENT}}⟧\ (\{y \in \text{E}|$
$(x, y) \in ⟦\text{READ}⟧^{-1}\ \}) = 1\}) = 1$

$\overset{L:2.1}{\Longleftrightarrow} ⟦\text{BOOK}⟧ \cap \{x \in \text{E}|\ ⟦\text{STUDENT}⟧ \subseteq \{y \in \text{E}|\ (x, y) \in ⟦\text{READ}⟧^{-1}\ \}\} \neq \emptyset$

As can be seen from the interpretations in (19b) and (21b), $It(⟦\text{EVERY}^{\text{STUDENT}}⟧, ⟦\text{SOME}^{\text{BOOK}}⟧)\,(⟦\text{READ}⟧) \neq It(⟦\text{SOME}^{\text{BOOK}}⟧, ⟦\text{EVERY}^{\text{STUDENT}}⟧)\,(⟦\text{READ}⟧^{-1})$. While the former means that the set of students is a subset of the set of book-readers, the latter means that the intersection between the set of books and the set of things that were read by every student is non-empty. In a situation where every student read a different book the former is true but the latter is false.

Since iteration is not commutative, changing the order in which the quantifiers are composed may create different interpretations, depending on the quantifiers that are involved. There are two possibilities, given in LEMMA 2.3. In LEMMA 2.3a, we have order dependence, that is, the interpretation of the complex quantifier is dependent on the order of the simpler quantifiers. In this case we have scope interaction: on the left-hand side, Q_1 outscopes Q_2 (like in (19b)), on the right-hand side Q_2 outscopes Q_1 (like in (21b)):

Lemma 2.3 The Quantifier Scope Lemma
For a domain E, Q_1, Q_2 *type* $\langle 1 \rangle$ *quantifiers on* E, *and* $\text{R} \in \text{E}^2$, *the following possibilities are available:*

a. $\text{Q}_1(\text{Q}_2(\text{R})) \neq \text{Q}_2(\text{Q}_1(\text{R}))$ *(order dependence: scope interaction)*

b. $\text{Q}_1(\text{Q}_2(\text{R})) = \text{Q}_2(\text{Q}_1(\text{R}))$ *(order independence: scope neutrality)*

[7] $⟦\text{READ}⟧^{-1}$ is the inverse relation of ⟦READ⟧ which is now needed, since the order in which the relation applies to the two arguments is reversed.

But there are also iterations of quantifiers for which changing the order does not create a different interpretation. Peters and Westerståhl (2006, p. 349) show that for $Q_1 = Q_2 =$ SOME and $Q_1 = Q_2 =$ EVERY, the equality in LEMMA 2.3b holds. However, this should not lead us to expect that we automatically get order independence and scope neutrality with identical quantifiers, since in Section 4.1 we will see that this does not hold for $Q_1 = Q_2 =$ NO.

Conclusion I conclude at this point that iteration is one operation by which polyadic quantifiers may be interpreted merely based on the semantics of their monadic parts. As can be seen from the discussion in this section and the reasoning by which we arrived at defining iteration as a polyadic lift, this way of defining the semantics of a polyadic quantifier does not go beyond the semantics of monadic quantifiers. DEFINITION 2.6 provides us with a syntactic mechanism by which we can interpret a sentence with several monadic quantifiers that we would normally represent with a polyadic quantifier. Iteration expresses precisely this syntactic mechanism of interpreting the monadic quantifiers one by one. As a polyadic lift, iteration is a composition of monadic quantifiers in which each monadic part can be dealt with separately, bringing its own contribution to the meaning of the whole independently of the contributions of the other parts. By DEFINITION 2.8 and DEFINITION 2.9, even the most complex polyadic quantifier may eventually be reduced to several iterations of monadic ones (see for instance (20)). Iteration itself does not contribute anything additional to the semantics of the monadic quantifiers.

As we will see in the next section, other polyadic lifts, which are inherently polyadic, behave differently from iteration to the extent that either the monadic quantifiers are interpreted as dependent on each other, or the polyadic lift itself contributes some additional semantics to the interpretation of the polyadic quantifier, besides the semantics of the monadic components.

2.1.3 Other polyadic quantifiers and polyadic lifts

There are several cases of natural language polyadic quantification in the linguistic literature where iteration does not yield the correct results (see Higginbotham and May (1981), Clark and Keenan (1987), Keenan (1987, 1992), van Benthem (1989), May (1989), Keenan and Westerståhl (1997), a.o.). In such cases, other operations have to be defined in order to derive the right truth conditions. A few such examples are presented below: quantifiers with *different*/ *same*, *cumulations*, and *resumptions*.

2.1.3.1 "Different"/ "same"

Different/ *same* quantifiers are often cited in the GQT literature (especially in Keenan (1987, 1992), Keenan and Westerståhl (1997)) as a case of polyadic quantification that goes beyond the limits of iteration. Treating the second reading of each of the sentences in (22) below as a polyadic quantifier offers a straightforward account for the fact that the interpretation of *different* and *same* is dependent on the previous quantifier.

Let us take a look at the sentences in (22) which are ambiguous:

(22) a. *Two boys* in my class date *different girls*.

1. Two boys in my class date different girls from the ones we know.
2. The girls that one of the two boys dates are all different from the girls that the other boy dates.
3. Two boys in my class date various/ many girls.

b. *Two students* answered *the same questions*.

1. Two students answered the questions that we are discussing now.
2. Whatever questions one of the two students answered were also answered by the second student.

The source of the ambiguity in the two sentences resides in the interpretation that the NPs *different girls* and *the same questions* receive. The first and the third reading in (22a) and the first reading in (22b) presuppose the same interpretation for the two NPs as in (23a) and (23b), respectively. In (23a), the NP *different girls* is still ambiguous: the sentence may suggest that John dates girls who are different from the ones known in the context, e.g. different from "the ones we know" (reading 1.), or it may suggest that John dates at least two girls (reading 2.). (23b) indicates that John answered the same questions as the ones specified in the context, for instance "the questions that we are discussing now" as described in the second reading for (22b):

(23) a. John dates different girls.
1. John dates different girls from the ones we know.
2. John dates various/ many girls.

b. John answered the same questions.

The readings that concern polyadic quantification are the ones given as the second reading for each sentence in (22). For (22a), reading 2. entails that the choice of a girl who is dated is constrained to co-vary with the choice of the boy who is involved in the dating activity. In this interpretation, the sentence is false if there is a girl who has been dated by both boys. Note however that this scenario does not yield falsity for readings 1. and 3.: two boys may have dated the same girl, as long as the girl is different from "the ones we know" (for reading 1.) or as long as the two boys dated many/ at least two girls. For (22b), the corresponding interpretation (given in reading 2.) is that the choice of the answered question is constrained to be the same for both students who do the answering.

The fact that the readings 1. and 3. in (22a) and the reading 1. in (22b) are also available in the absence of the quantifier TWO (see (23a) and (23b) where *John* replaces *two boys* and *two students*) suggests that the interpretation of *different* and *same* in these readings is independent of the presence of a quantifier. But reading 2. in (22a) and (22b) is directly related to the presence of another quantifier and as we will see below, the treatment in terms of polyadic quantification takes this fact into account. Independently of what syntactic status one may assign to *different* and *same*, we are interested in providing the right semantics for reading 2. In GQT this can be done by assuming that *different*/ *same* denote quantifiers, as determiners like *every, two, some* etc. do. Thus in order to interpret the sentences in (22) under their second reading, we represent them with the two binary quantifiers below:

(24) a. *Two boys* date *different girls*.
$(\mathrm{TWO}^{\mathrm{BOY}}, \mathrm{DIFFERENT}^{\mathrm{GIRL}})(\mathrm{DATE})$
$(\mathrm{TWO}, \mathrm{DIFFERENT})^{\mathrm{BOY},\ \mathrm{GIRL}}(\mathrm{DATE})$

b. *Two students* answered *the same question*.
$(\mathrm{TWO}^{\mathrm{STUDENT}}, \mathrm{SAME}^{\mathrm{QUESTION}})(\mathrm{ANSWER})$
$(\mathrm{TWO}, \mathrm{SAME})^{\mathrm{STUDENT},\ \mathrm{QUESTION}}(\mathrm{ANSWER})$

I mentioned before that the polyadic quantifiers based on *different* and *same* (i.e. the ones in (24)) are taken as cases of polyadic quantification that iteration cannot account for. Let us see why the meaning of the binary quantifiers in (22) cannot be obtained by iteration. The answer lies in the very definition of iteration in DEFINITION 2.8. If two unary relations A, B are added as the restrictions of the two monadic quantifiers, the value of a binary quantifier It(Q$_1$,Q$_2$) with Q$_1$, Q$_2$ (each of type $\langle 1, 1\rangle$) at the triple (A, B, R) is given by DEFINITION 2.10:

Definition 2.10 Iteration of two type ⟨1,1⟩ quantifiers
For Q$_1$, Q$_2$, *quantifiers of type ⟨1,1⟩,* It(Q$_1$,Q$_2$) *is the type* $\langle 1^2, 2\rangle$ *quantifier defined, for any domain* E, *any* A, B $\subseteq$ E, *any* R $\subseteq$ E^2, *as:*

It(Q$_1$,Q$_2$)(A, B, R)= Q$_1$(A, $\{x \in$ E$|$ Q$_2$(B, $\{y \in$ E$|\ (x, y) \in$ R$\})\}$)

Given A, the value of It(Q$_1$,Q$_2$) depends on what individuals x are in the set $\{x|$ Q$_2$(B, $\{y|\ (x, y) \in$ R$\})\}$ (the domain of R). And given B, whether an individual x_1 is in this set is determined by $\{y|\ (x_1, y) \in$ R$\}$, the set of things x_1 bears R to (i.e. the co-domain of R). This means that given B, Q$_2$ decides whether to put an individual x_1 in the set $\{x|$Q$_2$(B, $\{y|\ (x, y) \in$R$\})\}$ only by checking the set of things x_1 is related to; in deciding about x_1, Q$_2$ does not have at hand the set $\{y|\ (x_2, y) \in$ R$\}$ of things some x_2 is related to, and thus cannot make its decision about x_1 contingent, for example, on whether $\{y|\ (x_1, y) \in$ R$\} \neq \{y|\ (x_2, y) \in$ R$\}$ or $\{y|\ (x_1, y) \in$ R$\} = \{y|\ (x_2, y) \in$ R$\}$.

However, the functions that are needed in order to interpret (22a) and (22b) must be sensitive to whether or not different individuals in the domain of the relation are related to the same elements in the co-domain. In (22a), A= ⟦BOY⟧, B= ⟦GIRL⟧, R= ⟦DATE⟧, and according to iteration, if a boy x_1 is in the set $\{x|$Q$_2$(⟦GIRL⟧, $\{y|\ (x, y) \in$ ⟦DATE⟧$\})\}$ is determined directly by $\{y|(x_1, y) \in$ ⟦DATE⟧$\}$. (We will follow CONVENTION 2.5 and refer to this latter set by the short notation ⟦DATE⟧$\,x_1$, i.e. the set of girls x_1 dates.) Thus the condition that ⟦DATE⟧$\,x_1 \neq$ ⟦DATE⟧$\,x_2$ for $x_1 \neq x_2$ cannot be specified. But this is exactly the way (22a) should be interpreted. Similarly, in (22b) the condition ⟦READ⟧$\,x_1$ =⟦READ⟧$\,x_2$ fails to be expressed by iteration, for every x_1, $x_2 \in$ ⟦STUDENT⟧ and $x_1 = x_2$.

Convention 2.5 *For* R$\subseteq$ E^2, $x \in$E, Rx *is the set of objects* x *bears* R *to, namely, we have the following convention:* R$x = \{y|\ (x, y) \in$R $\}$

The operations by which the interpretations of (22a) and (22b) can be obtained are given in DEFINITION 2.11 and DEFINITION 2.12, adapted from Keenan and Westerståhl (1997):

Definition 2.11 The semantics of polyadic quantifiers containing DIFFERENT
For Q, *a polyadic quantifier of type* $\langle 1^2, 2\rangle$ *containing* DIFFERENT, A, B $\subseteq$ E, R $\subseteq$ E^2, *and* H *a quantifier of type* $\langle 1, 1\rangle$, *the interpretation of* Q *is given by:*

Q$^{\mathrm{A,B}}$(R) $= 1$ *iff* *there is* A$_\subseteq \subseteq$ A [H$^{\mathrm{A}}$(A$_\subseteq$) $= 1$ *and*
for all $x, y \in$ A$_\subseteq$ $(x \neq y \Rightarrow$ B $\cap$ R$x \neq$ B $\cap$ R$y)$]

Definition 2.12 The semantics of polyadic quantifiers containing SAME
For Q, *a polyadic quantifier of type* $\langle 1^2, 2\rangle$, *containing* SAME, A, B $\subseteq$ E, R $\subseteq$ E^2, *and* H *a quantifier of type* $\langle 1, 1\rangle$, *the interpretation of* Q *is given by:*

Q$^{\mathrm{A,B}}$(R) $= 1$ *iff* *there is* A$_\subseteq \subseteq$ A [H$^{\mathrm{A}}$(A$_\subseteq$) $= 1$ *and*
for all $x, y \in$ A$_\subseteq(x \neq y \Rightarrow$ B $\cap$ R$x =$ B $\cap$ R$y)$]

The semantics of DIFFERENT/ SAME establishes a close relation to the previous quantifier, so the polyadic quantifier has to be interpreted as a whole. This is exactly what the definitions in DEFINITION 2.11 and DEFINITION 2.12 do: H is the quantifier with respect to which the semantics of DIFFERENT and SAME is expressed. For the examples in (24), H = TWO. More intuitively, in (24a), DIFFERENT brings its semantic contribution only in relation to two elements of the co-domain of the relation ⟦DATE⟧, fixed with respect to two elements in the set ⟦BOY⟧. By substituting the variables in DEFINITION 2.11, we get Q = ⟦(TWO, DIFFERENT)⟧, H = ⟦TWO ⟧, A = ⟦BOY⟧, B = ⟦GIRL⟧ and the semantics of (22a) as (25) below:

(25) ⟦(TWO, DIFFERENT)$^{\text{BOY, GIRL}}$(DATE)⟧= 1 iff
there is $[\![\text{BOY}]\!]_\subseteq \subseteq [\![\text{BOY}]\!]$
$[\ [\![\text{TWO}^{\text{BOY}}(\text{BOY}_\subseteq)]\!] = 1$ and for all $x, y \in [\![\text{BOY}]\!]_\subseteq$ $(x \neq y \Rightarrow$
$[\![\text{GIRL}]\!] \cap [\![\text{DATE}]\!]x \neq [\![\text{GIRL}]\!] \cap [\![\text{DATE}]\!]y)]$
$\overset{L:2.1}{\Longleftrightarrow}$ ⟦(TWO, DIFFERENT)$^{\text{BOY, GIRL}}$(DATE)⟧= 1 iff
there is $[\![\text{BOY}]\!]_\subseteq \subseteq [\![\text{BOY}]\!]$
$[\ |[\![\text{BOY}]\!] \cap [\![\text{BOY}]\!]_\subseteq)| = 2$ and for all $x, y \in [\![\text{BOY}]\!]_\subseteq$ $(x \neq y \Rightarrow$
$[\![\text{GIRL}]\!] \cap [\![\text{DATE}]\!]x \neq [\![\text{GIRL}]\!] \cap [\![\text{DATE}]\!]y)]$
$\Longleftrightarrow$ ⟦(TWO, DIFFERENT)$^{\text{BOY, GIRL}}$(DATE)⟧= 1 iff
there is $[\![\text{BOY}]\!]_\subseteq \subseteq [\![\text{BOY}]\!]$
$[\ |\ [\![\text{BOY}]\!]_\subseteq| = 2$ and for all $x, y \in [\![\text{BOY}]\!]_\subseteq$ $(x \neq y \Rightarrow$
$[\![\text{GIRL}]\!] \cap [\![\text{DATE}]\!]x \neq [\![\text{GIRL}]\!] \cap [\![\text{DATE}]\!]y)]$

A similar mechanism can be applied to derive the semantics of (24b), containing SAME. In this case, Q = ⟦(TWO, SAME)⟧, H = ⟦TWO⟧, A = ⟦STUDENT⟧, B = ⟦QUESTION⟧. H is the same as in (24a), i.e. ⟦TWO⟧, so we again obtain $|[\![\text{STUDENT}]\!]_\subseteq| = 2$. The semantics of (24b) is given by (26):

(26) ⟦(TWO, SAME)$^{\text{BOY, GIRL}}$(ANSWER)⟧= 1 iff
there is $[\![\text{STUDENT}]\!]_\subseteq \subseteq [\![\text{STUDENT}]\!]$
$[\ |[\![\text{STUDENT}]\!]_\subseteq| = 2$ and for all $x, y \in [\![\text{STUDENT}]\!]_\subseteq$ $(x \neq y \Rightarrow$
$[\![\text{QUESTION}]\!] \cap [\![\text{ANSWER}]\!]x = [\![\text{QUESTION}]\!] \cap [\![\text{ANSWER}]\!]y)\ ]$

In conclusion, the polyadic quantifiers in DEFINITION 2.11 and 2.12 allow us to interpret the sentences in (24) where the DIFFERENT and SAME quantifiers must be dependent on the previous quantifier. As we saw above, iteration cannot express this dependence between the monadic parts of a polyadic quantifier, because it interprets them independently of one another. Thus the polyadic lifts in DEFINITION 2.11 and 2.12 can distinguish between relations in a way that is not available for iteration. In Section 2.1.4, this intuition will be expressed in a more precise way, by proving that these polyadic quantifiers are not "reducible" to iteration.

2.1.3.2 Cumulation

Cumulative quantification is discussed in Keenan (1987), Keenan (1992), Westerståhl (1994), Keenan and Westerståhl (1997), and Peters and Westerståhl (2006), a.o. One example is the sentence below:

(27) *Forty contributors* wrote *thirty-two papers* for the Handbook.
(FORTY$^{\text{CONTRIBUTOR}}$, THIRTY-TWO$^{\text{PAPER}}$)(WRITE)
(FORTY, THIRTY-TWO)$^{\text{CONTRIBUTOR, PAPER}}$(WRITE)

If one interprets the polyadic quantifier in (27) by iteration, two readings can be obtained, depending on the order of the two monadic quantifiers (see the first proposition of LEMMA 2.3):

(28) a. $It([\![\text{FORTY}^{\text{CONTRIBUTOR}}]\!], [\![\text{THIRTY-TWO}^{\text{PAPER}}]\!])([\![\text{WRITE}]\!]) = 1$

$\overset{D:2.8}{\Longleftrightarrow} ([\![\text{FORTY}^{\text{CONTRIBUTOR}}]\!] \circ [\![\text{THIRTY-TWO}^{\text{PAPER}}]\!])([\![\text{WRITE}]\!]) = 1$

$\overset{D:2.8}{\Longleftrightarrow} ([\![\text{FORTY}^{\text{CONTRIBUTOR}}]\!])(\{x|[\![\text{THIRTY-TWO}^{\text{PAPER}}]\!](\{y| (x,y) \in [\![\text{WRITE}]\!]\}) = 1\}) = 1$

$\overset{L:2.1}{\Longleftrightarrow} |[\![\text{CONTRIBUTOR}]\!] \cap \{x|\ |[\![\text{PAPER}]\!] \cap \{y| (x,y) \in [\![\text{WRITE}]\!]\}| = 32\}| = 40$

b. $It([\![\text{THIRTY-TWO}^{\text{PAPER}}]\!], [\![\text{FORTY}^{\text{CONTRIBUTOR}}]\!])([\![\text{WRITE}]\!]^{-1}) = 1$

$\overset{D:2.8}{\Longleftrightarrow} ([\![\text{THIRTY-TWO}^{\text{PAPER}}]\!] \circ [\![\text{FORTY}^{\text{CONTRIBUTOR}}]\!])([\![\text{WRITE}]\!]^{-1}) = 1$

$\overset{D:2.8}{\Longleftrightarrow} ([\![\text{THIRTY-TWO}^{\text{PAPER}}]\!])(\{x|\ [\![\text{FORTY}^{\text{CONTRIBUTOR}}]\!](\{y| (x,y) \in [\![\text{WRITE}]\!]^{-1}\}) = 1\}) = 1$

$\overset{L:2.1}{\Longleftrightarrow} |[\![\text{PAPER}]\!] \cap \{x|\ |[\![\text{CONTRIBUTOR}]\!] \cap \{y| (x,y) \in [\![\text{WRITE}]\!]^{-1}\}| = 40\}| = 32$

These interpretations are obtained by composing the semantics of the two cardinal quantifiers. In (28a), the quantifier FORTY is the leftmost one, so it outscopes THIRTY-TWO. The interpretation is that every of the forty contributors wrote thirty-two papers, so the total number of papers that were written is 1280. The other interpretation (in (28b)), with THIRTY-TWO taking scope over FORTY, says that each of the thirty-two papers was written in a collaboration between forty contributors. Thus the number of contributors is 1280.

However, neither of the two readings in (28) is the first one conveyed by the sentence in (27). It is rather an interpretation in which there is a total of forty contributors and a total of thirty-two papers, such that each of the contributors wrote some paper (perhaps more than one, perhaps jointly with other contributors) and each of the papers was authored by some of these contributors. In this case, the two quantifiers are interpreted “cumulatively”. This reading can be obtained via a polyadic quantifier that is derived by means of *cumulation*, another polyadic lift defined in Westerståhl (1994), Keenan and Westerståhl (1997), and Peters and Westerståhl (2006):[8]

Definition 2.13 *k*-ary Cumulation of type $\langle 1, 1\rangle$ quantifiers
For any $k \geq 1$*, for* $\text{Q}_1, ..., \text{Q}_k$ *quantifiers of type* $\langle 1, 1\rangle$*, for* $\text{A}_1, \text{A}_2, ..., \text{A}_k \subseteq \text{E}$*,* $a_1 \in \text{A}_1, a_2 \in \text{A}_2, ..., a_{k-1} \in \text{A}_{k-1}, a_k \in \text{A}_k$*, and* $\text{R} \subseteq \text{E}^k$*, the polyadic cumulative quantifier* $Cum(\text{Q}_1, ..., \text{Q}_k)$ *of type* $\langle 1^k, k\rangle$ *is defined as:*

$$Cum(\text{Q}_1, ..., \text{Q}_k)^{\text{A}_1,...,\text{A}_k}(\text{R}) = \text{Q}_1^{\text{A}_1}(\{a_1|(a_1, a_2, ..., a_k) \in \text{R}\}) \wedge \text{Q}_2^{\text{A}_2}(\{a_2|(a_1, a_2, ..., a_k) \in \text{R}\}) \wedge ... \wedge \text{Q}_{k-1}^{\text{A}_{k-1}}(\{a_{k-1}|(a_1, a_2, ..., a_k) \in \text{R}\}) \wedge \text{Q}_k^{\text{A}_k}(\{a_k|(a_1, a_2, ..., a_k) \in \text{R}\}).$$

DEFINITION 2.13 describes a cumulative quantifier of type $\langle 1^k, \text{k}\rangle$ as the conjunction of the component monadic quantifiers, each applied to its restriction and the corresponding set of all the elements

[8] DEFINITION 2.13 is a modified version of Westerståhl (1994).

that occupy a certain position (1, or 2, or ..., or k) in the k-tuples that belong to the relation R. Here we will only discuss cases of binary polyadic quantifiers, so we will not make use of the complex quantifiers in DEFINITION 2.13, but only of the simpler version given in DEFINITION 2.14 which defines cumulative quantifiers of type $\langle 1^2, 2\rangle$.

Definition 2.14 Binary cumulation of type $\langle 1, 1\rangle$ quantifiers
For Q_1, Q_2 *quantifiers of type* $\langle 1, 1\rangle$, $A_1, A_2 \subseteq E$, $x \in A_1$, $y \in A_2$, $R \subseteq E^2$, $Cum(Q_1, Q_2)$, *the polyadic quantifier of type* $\langle 1^2, 2\rangle$ *is defined as:*

$$Cum(Q_1, Q_2)^{A_1,A_2}(R) = Q_1^{A_1}(\{x|(x,y) \in R\}) \wedge Q_2^{A_2}(\{y|(x,y) \in R\}).$$

With the help of DEFINITION 2.14, we can derive the cumulative interpretation for the sentence in (27). If we replace Q_1 with [[FORTY]], Q_2 with [[THIRTY-TWO]], R with [[WRITE]], A_1 with [[CONTRIBUTOR]] and A_2 with [[PAPER]] in DEFINITION 2.14, we obtain (29):

(29) $\text{(FORTY, THIRTY-TWO)}^{\text{CONTRIBUTOR, PAPER}}\text{(WRITE)}$
$Cum([\![\text{FORTY}]\!], [\![\text{THIRTY-TWO}]\!])^{[\![\text{CONTRIBUTOR}]\!], [\![\text{PAPER}]\!]}([\![\text{WRITE}]\!])$
$\overset{D:2.14}{=\!=}$ for every $x \in [\![\text{CONTRIBUTOR}]\!]$, $y \in [\![\text{PAPER}]\!]$,
$[\![\text{FORTY}]\!]([\![\text{CONTRIBUTOR}]\!], \{x|\ (x,y) \in [\![\text{WRITE}]\!]\})$
$\wedge\ [\![\text{THIRTY-TWO}]\!]([\![\text{PAPER}]\!], \{y|\ (x,y) \in [\![\text{WRITE}]\!]\})$

Given the semantics of cardinal quantifiers in DEFINITION 2.2, the truth conditions of the cumulative quantifier in (27) can be derived as in (30b):

(30) a. For a domain E, A, B $\subseteq$ E, the following hold:
$[\![\text{FORTY}]\!](A, R) = 1$ iff $|A \cap R| = 40$
$[\![\text{THIRTY-TWO}]\!](B, R) = 1$ iff $|B \cap R| = 32$

b. *Forty contributors* wrote *thirty-two papers* for the Handbook.
$\text{(FORTY, THIRTY-TWO)}^{\text{CONTRIBUTOR, PAPER}}\text{(WRITE)}$
$Cum([\![\text{FORTY}]\!], [\![\text{THIRTY-TWO}]\!])^{[\![\text{CONTRIBUTOR}]\!], [\![\text{PAPER}]\!]}([\![\text{WRITE}]\!])$
$\overset{D:2.14}{=\!=}$ for every $x \in [\![\text{CONTRIBUTOR}]\!]$, $y \in [\![\text{PAPER}]\!]$,
$[\![\text{FORTY}]\!]([\![\text{CONTRIBUTOR}]\!], \{x|\ (x,y) \in [\![\text{WRITE}]\!]\})$
$\wedge\ [\![\text{THIRTY-TWO}]\!]([\![\text{PAPER}]\!], \{y|\ (x,y) \in [\![\text{WRITE}]\!]\}) = 1$
$\overset{D:2.1,30a}{\Longleftrightarrow} |[\![\text{CONTRIBUTOR}]\!] \cap \{x|\ (x,y) \in [\![\text{WRITE}]\!]\}| = 40$
$\wedge\ |[\![\text{PAPER}]\!] \cap \{y|\ (x,y) \in [\![\text{WRITE}]\!]\}| = 32$

This interpretation captures the reading usually associated with (27): there is a total of forty contributors and a total of thirty-two papers, such that the former wrote the latter.

Scope neutrality While within iteration the order from left to right dictates the scope interaction between the component quantifiers (see also (28) above), the situation with cumulation is different. Since a cumulative quantifier is a conjunction of the monadic quantifiers, and conjunction is commutative in general, the order has no influence on interpretation: the meaning of (31) below is identical to the one in (30b):

(31) $Cum(⟦\text{THIRTY-TWO}⟧, ⟦\text{FORTY}⟧)^{⟦\text{PAPER}⟧,\ ⟦\text{CONTRIBUTOR}⟧}(⟦\text{WRITE}⟧^{-1}) = 1$

$\overset{D:2.14}{\Longleftrightarrow}$ for every $x \in ⟦\text{PAPER}⟧, y \in ⟦\text{CONTRIBUTOR}⟧$

$⟦\text{THIRTY-TWO}⟧(⟦\text{PAPER}⟧, \{x|\ (x, y) \in ⟦\text{WRITE}⟧^{-1}\})$

$\wedge\ ⟦\text{FORTY}⟧(⟦\text{CONTRIBUTOR}⟧, \{y|(x, y) \in ⟦\text{WRITE}⟧^{-1}\}) = 1$

$\overset{L:2.1,30a}{\Longleftrightarrow} |⟦\text{PAPER}⟧ \cap \{x|\ (x, y) \in ⟦\text{WRITE}⟧^{-1}\}| = 32$

$\wedge\ |⟦\text{CONTRIBUTOR}⟧ \cap \{y|\ (x, y) \in ⟦\text{WRITE}⟧^{-1}\}| = 40$

To generalize, for cumulation we have the following lemma:

Lemma 2.4 $Cum(\text{Q}_1,\text{Q}_2)(\text{R}) = Cum(\text{Q}_2,\text{Q}_1)(\text{R}^{-1})$

This means that cumulation is order-independent and thus neutral to scope, so it generally obeys the second proposition in LEMMA 2.3.

2.1.3.3 Resumption

Multiple wh-questions represent another construction that has been characterized by means of polyadic quantifiers (see Higginbotham and May (1981), May (1989), Keenan (1992, 1996), Keenan and Westerståhl (1997)). The debate on the appropriate mechanism to account for the semantics of multiple wh-questions is far from settled, but in what follows, I am only concerned with the way polyadic quantifiers have been used in this respect, and the reader is referred to Higginbotham (1995), Groenendijk and Stokhof (1997), Ginzburg and Sag (2000) for other approaches.

For questions like (32), the polyadic quantifier literature argues that the wh-quantifier quantifies over pairs that satisfy the relation CHASE (see for instance Keenan (1996)):

(32) Which dog chased which cat?
$(\text{WH}^{\text{DOG}}, \text{WH}^{\text{CAT}})(\text{CHASE})$
$(\text{WH}, \text{WH})^{\text{DOG, CAT}}(\text{CHASE})$

Keenan argues that if the question in (32) is assumed to be a binary iteration of the unary interrogative operator, it should be successfully answerable with a single NP, say *Fido*, filling the value of the first wh-quantifier in that iteration. The interpretation of this answer should be the unary interrogation *Fido chased which cat?* However, the NP *Fido* is not an appropriate answer for (32), although the pair (*Fido, Tom*) is:

(33) Which dog chased which cat?
  a. # Fido.
  b. Fido (chased) Tom.

Replacing one quantifier with an NP in an iteration usually does not affect the possibility to interpret the sentence, as can be seen for (19), given here as (34):

(34) Every student read some book.
  a. John read some book.
  b. Every student read *A Natural History of Negation*.

First, the contrast between (33a) and (34a)/ (34b) shows that composing the two wh-quantifiers in (32) by iteration is problematic. Second, (33b) indicates that (32) asks for pairs of a dog and a cat, which in GQT terms can be naturally represented by the wh-quantifier taking the properties DOG and CAT as arguments.

This operation is usually referred to as *resumption* and together with *cumulation* is used as an alternative polyadic lift to iteration. Resumption expresses the interpretation of n identical unary quantifiers as an instance of one n-ary quantifier yielding a truth value of the n-ary relation. It is commonly assumed that given a domain E, we can define a unary quantifier $\mathrm{Q_E}$ (see CONVENTION 2.6) as a relation between subsets of E, but as a general case, we may define a k-ary quantifier $\mathrm{Q}_{\mathrm{E}^k}$ as a relation between subsets of E^k. This latter quantifier corresponds to the idea of resumption as a polyadic lift. Resumptive polyadic quantifiers are defined as in DEFINITION 2.15 below, along the lines of Keenan and Westerståhl (1997).

Convention 2.6 *For a domain* E, *we have the following convention:*

$$\mathrm{Q_E}\text{: } (\mathrm{P(E^1)}\times \mathrm{P(E^1)}) \rightarrow \mathrm{P(E^0)}\ (\textit{type}\ \langle 1,1\rangle)$$
$$\mathrm{Q}_{\mathrm{E}^k}\text{: } (\underbrace{\mathrm{P(E^1)} \times \mathrm{P(E^1)} \times \ldots \times \mathrm{P(E^1)}}_{k\text{-times}} \times \mathrm{P(E}^k)) \rightarrow \mathrm{P(E^0)}\ (\textit{type}\ \langle 1^k, k\rangle)$$

Definition 2.15 *K*-ary resumption of type $\langle 1,1\rangle$ quantifiers
For a quantifier Q *of type* $\langle 1,1\rangle$, *given* E *the domain, for any* $k \geq 1$, $\mathrm{A}_1, \mathrm{A}_2, \ldots, \mathrm{A}_k \subseteq \mathrm{E}$, $\mathrm{R} \subseteq \mathrm{E}^k$, *the polyadic quantifier* $Res^k(\mathrm{Q})$ *of type* $\langle 1^k, k\rangle$ *derived from* Q *is defined as:*

$$Res^k(\mathrm{Q})_{\mathrm{E}}^{\mathrm{A}_1,\mathrm{A}_2,\ldots,\mathrm{A}_k}(\mathrm{R}) = \mathrm{Q}_{\mathrm{E}^k}^{\mathrm{A}_1\times\mathrm{A}_2\times\ldots\times\mathrm{A}_k}(\mathrm{R})$$

DEFINITION 2.15 gives us the general case with a k-number of monadic type $\langle 1,1\rangle$ quantifiers applying to a k-ary relation. Thus a unary quantifier alone can be viewed as a unary resumption of itself, for $k = 1$.

In order to account for the resumptive quantifier in (32), we define binary resumption:

Definition 2.16 Binary resumption of type $\langle 1,1\rangle$ quantifiers
For a quantifier Q *of type* $\langle 1,1\rangle$, *given* E *the domain,* A, B $\subseteq$ E, R $\subseteq$ E ^{2}, *the polyadic quantifier* $Res^2(\mathrm{Q})$ *of type* $\langle 1^2,2\rangle$ *derived from* Q *is defined as:*

$$Res^2(\mathrm{Q})_{\mathrm{E}}^{\mathrm{A,B}}(\mathrm{R}) = \mathrm{Q}_{\mathrm{E}^2}^{\mathrm{A}\times\mathrm{B}}(\mathrm{R})$$

In view of DEFINITION 2.16, we can represent the wh-question in (32) as below:

(35) Which dog chased which cat?
$(\mathrm{WH^{DOG}, WH^{CAT}})(\mathrm{CHASE})$
$(\mathrm{WH, WH})^{\mathrm{DOG,\ CAT}}(\mathrm{CHASE})$
$Res^2([\![\mathrm{WH}]\!])_{\mathrm{E}}^{[\![\mathrm{DOG}]\!],\ [\![\mathrm{CAT}]\!]}([\![\mathrm{CHASE}]\!])$
$\overset{D.2.16}{=\!=} [\![\mathrm{WH}]\!]_{\mathrm{E}^2}^{[\![\mathrm{DOG}]\!]\times[\![\mathrm{CAT}]\!]}([\![\mathrm{CHASE}]\!])$

The representation in (35) tacitly assumes that the meaning of a question is the set of its answers (as in Groenendijk and Stokhof (1997)). Thus the interpretation of (35) is given by the set of (DOG, CAT) pairs which take the CHASE relation to truth. The interpretation of the quantifier ($\mathrm{WH^{DOG}}$,

$\mathrm{WH}^{\mathrm{CAT}}$) can be derived from that of the corresponding monadic quantifier, by DEFINITION 2.16. The semantics of the monadic WH in DEFINITION 2.17a[9] may be generalized to k-ary WH-quantifiers as in DEFINITION 2.17a, and by that, we can interpret (35) as in (36):

Definition 2.17 The semantics of WH-quantifiers

a. For a domain E, $\mathrm{A}, \mathrm{R} \subseteq \mathrm{E}$,

$$[\![\mathrm{WH}]\!]_{\mathrm{E}}(\mathrm{A}, \mathrm{R}) = 1 \textit{ iff } \mathrm{A} \cap \mathrm{R} \neq \emptyset$$

b. For a domain E, $\mathrm{A}_1, \mathrm{A}_2, \ldots, \mathrm{A}_k, \subseteq \mathrm{E}$, $\mathrm{R} \subseteq \mathrm{E}^k$,

$$[\![\mathrm{WH}\ ]\!]_{\mathrm{E}^k}(\mathrm{A}_1 \times \mathrm{A}_2 \times \ldots \times \mathrm{A}_k, \mathrm{R}) = 1 \textit{ iff } (\mathrm{A}_1 \times \mathrm{A}_2 \times \ldots \times \mathrm{A}_k) \cap \mathrm{R} \neq \emptyset$$

(36) $$[\![\mathrm{WH}]\!]_{\mathrm{E}^2}^{[\![\mathrm{DOG}]\!] \times [\![\mathrm{CAT}]\!]}([\![\mathrm{CHASE}]\!]) = 1$$
$$\overset{D.2.17}{\Longleftrightarrow} ([\![\mathrm{DOG}]\!] \times [\![\mathrm{CAT}]\!]) \cap [\![\mathrm{CHASE}]\!] \neq \emptyset$$

Scope neutrality With iteration, the order of the monadic quantifiers determines the scope interaction between the monadic quantifiers (see (19) vs. (21) and (28a) vs. (28b)). In the case of cumulation, the interpretation is independent of the order of the quantifiers (31). With resumption, the question of order doesn't arise at all, since there is formally only one occurrence of the monadic quantifier, even if the linguistic construction includes two quantifiers. Changing the order of the two NP quantifiers in the linguistic example (35) has no effect on the interpretation:

(37) Which dog chased which cat?
⇔ Which cat was chased by which dog?

2.1.3.4 Conclusion

Among the four polyadic quantifiers that we have looked at, two classes can be differentiated: iterations and "non-iterations". The former are essentially monadic, the latter are derived by polyadic lifts and are inherently polyadic.

What distinguishes the two classes is first of all the way they build their semantics. As the definition of iteration and previous examples suggest, each monadic quantifier within an iteration contributes its own semantics, independently of the other quantifier(s). "Different"/"same" quantifiers, cumulations, and resumptions cannot be accounted for by iteration precisely for this reason. The meaning of these polyadic quantifiers is derived in such a way that each monadic quantifier contributes its meaning only in relation to the other one(s). This dependency relation between the semantics of the monadic quantifiers must be specified for each polyadic quantifier (see DEFINITION 2.11, vs. DEFINITION 2.12 vs. DEFINITION 2.13 vs. DEFINITION 2.15). The meaning of non-iterations can only be derived as a whole.

This characteristic is also reflected in the syntax by which the polyadic quantifiers are derived. Unlike iterations, polyadic lifts and inherently polyadic quantifiers are neutral to scope and order independent (see for instance the results in LEMMA 2.4 and (37)). For some of them, the issue doesn't

[9] One may notice that the semantics of WH in DEFINITION 2.17 is identical to that of SOME in DEFINITION 2.2. This idea has its origin in Karttunen (1977) and is well expected under the assumption that a question is true iff the set of its answers is non-empty: a wh-question is true iff there is at least one individual that can successfully replace the wh-pronoun to yield a true proposition.

arise at all, as is the case with “different” and “same”, or resumption. In Section 3.5.5, I will show that the difference between the two classes of polyadic quantifiers is also visible when they interact with other operators which are not part of their structure.

2.1.4 Reducibility

An important concern of the research on polyadic quantifiers is to answer the question whether a polyadic quantifier Q is definable from the monadic $\mathrm{Q}_1, ..., \mathrm{Q}_n$. This notion of definability is relative to the context in which one looks for an answer. From a logical point of view, Westerståhl (1989) investigates whether Q can be defined in a logic with n quantifiers. van Benthem (1989) addresses the question whether Q may be defined as a Boolean combination of iterations among $\mathrm{Q}_1, ..., \mathrm{Q}_n$. In linguistics, researchers are interested in determining whether Q may be defined in terms of $It(\mathrm{Q}_1, ..., \mathrm{Q}_n)$.

Here, we concentrate on the linguistic perspective. Polyadic quantifiers are complex higher order functions and linguists are usually reluctant to use them for the description of natural language. The concern is to keep this description simple. However, a polyadic quantifier becomes theoretically motivated once one can show that its particular interpretation, attested in the natural language, cannot be obtained by means of an iteration of monadic quantifiers. This then amounts to establishing that natural language quantification goes beyond monadicity. The analysis in Chapters 4 and 5 is built on the idea that a negative resumptive quantifier is a suitable semantic mechanism to account for the properties of negative concord in Romanian. In Section 4.2 I will address the issue of whether resumptive negative quantifiers are theoretically motivated. In this section I start with investigating the theoretical status of the polyadic quantifiers defined in Section 2.1.3.

Another important concern in the linguistic literature is interpreting complex constructions with several quantifiers in a way that corresponds to the principle of compositionality ((3) p. 2). In the tradition of Montague (1973), monadic quantifiers have been successfully accommodated within a compositional grammar. And since with iteration, the syntax-semantics of each monadic quantifier is taken into account independently of the other quantifiers, the general assumption within the theory of Polyadic Quantification is that iteration respects the principle of compositionality. Consequently, from a linguistic point of view, the definability question above is reformulated in terms of *reducibility* of the polyadic quantifier Q to the iteration of the monadic $\mathrm{Q}_1, ..., \mathrm{Q}_n$. An important technical result in this respect is Keenan (1992), which formulates a theorem that makes it possible to determine whether a polyadic quantifier can be reduced to an iteration of monadic quantifiers.

In this section, I briefly outline the advantages of iteration as a polyadic lift for linguistic theories (Section 2.1.4.1). Then I present the way the theorem in Keenan (1992) can be used to prove that the quantifiers in Section 2.1.3 cannot be reduced to iteration. The latter point will be important in Section 4.2 where I address the question whether negative resumptive quantifiers are reducible to iteration.

2.1.4.1 Monadicity

There are two reasons why iteration as a lift is preferable to the other polyadic operations: (1) its monadic character which ensures simplicity for the theory, (2) the assumed faithfulness to the principle of compositionality. In fact, the two aspects go hand in hand, but the former reflects the view from the Generalized Quantifier Theory, while the latter is relevant for linguistic theory in general.

The Generalized Quantifier Theory of Barwise and Cooper (1981) and the subsequent related literature offer a theory of monadic quantifiers that describes their formal properties and interpretation. If one distinguishes monadic quantifiers in the structure of a polyadic quantifier like the ones in Section 2.1.3 (see (24), (27) and (32)), a direct way to derive its meaning is by composing the meanings of its monadic components as they are defined in GQT. Iteration is the appropriate operation in this respect, since it does not introduce anything beyond the already defined monadic interpretations. This is transparent from DEFINITION 2.9 repeated below:

Definition 2.9 (p. 22) Iteration
For two quantifiers Q_1 *of type* $\langle n\rangle$, Q_2 *of type* $\langle k\rangle$, *for any* $n, k \in \mathbb{N}$, $It(\mathrm{Q}_1,\mathrm{Q}_2)$ *is the type* $\langle n{+}k\rangle$ *quantifier defined, for any domain* E, *any* $(a_1, ..., a_n) \in \mathrm{E}^n$, *any* $(b_1, ..., b_k) \in \mathrm{E}^k$, *and any* $\mathrm{R} \subseteq \mathrm{E}^{n+k}$, *as:*

$$\begin{aligned}\mathrm{It}(\mathrm{Q}_1,\mathrm{Q}_2)(\mathrm{R}) &= (\mathrm{Q}_1 \circ \mathrm{Q}_2)(\mathrm{R}) \\ &= \mathrm{Q}_1(\{(a_1, ..., a_n) | \ \mathrm{Q}_2(\{(b_1, ..., b_k) | (a_1, ..., a_n, b_1, ..., b_k) \in \mathrm{R}\}) = 1\})\end{aligned}$$

If the two quantifiers are not monadic but are themselves polyadic iterations (so $n, k > 1$), they may in turn be regarded as iterations of two simpler quantifiers, until in the end the $(n+k)$ iteration reduces to several binary iterations of monadic quantifiers (such an example was given in (20) for the ternary quantifier (THREE$^{\mathrm{TEACHER}}$, EVERY$^{\mathrm{STUDENT}}$, SOME$^{\mathrm{BOOK}}$)). Thus the meaning of an iteration is directly derived from the individual semantics of the monadic quantifiers and interpretation takes place within GQT.

Regarding the principle of compositionality, it is again the monadic character of iteration that makes it preferable to other polyadic lifts. Beginning with Montague's treatment of quantification in English, the linguistic literature has provided various examples of compositional accounts of (monadic) generalized quantifiers (see among others Partee (1987), Gamut (1991), Bach et al. (1995)). In these approaches the operation by which complex meanings are derived is functional application, since it is compositional.[10] As we will see in Section 4.3.3, composing two monadic quantifiers by iteration yields the same semantics as functional application. For this reason, iteration is considered the counterpart of functional application within Polyadic Quantification and thus a compositional operation for deriving polyadic quantifiers.[11]

In the next section I present the Theorem of Reducibility given in Keenan (1992), by which one can determine whether a polyadic quantifier may be reduced to an iteration.

2.1.4.2 Reducibility to iteration

The simplicity that comes with the monadic character of iteration makes it desirable for linguistic theory to reduce all natural language quantification to iteration. The question is whether we can restate the so-called inherently polyadic quantifiers in Section 2.1.3 exclusively in terms of iteration. To be precise, we need to determine whether an n-ary polyadic quantifier Q is *reducible* to $It(\mathrm{Q}_1, ..., \mathrm{Q}_n)$. For a positive answer to this question, it is sufficient to find the monadic quantifiers which by iteration yield the same result as the polyadic one. But a negative answer, as Keenan (1987), Keenan (1992), Keenan and Westerståhl (1997) indicate, needs a proof that there is no sequence of monadic quantifiers whose iteration could yield the same semantic interpretation as the corresponding polyadic lift.

In Section 2.1.3, intuitive arguments were brought to show that non-iterations are needed in the description of natural language quantification. Keenan (1987), van Benthem (1989), and Keenan

[10] A description of a compositional grammar is given in Section 4.3.2.4.

[11] In Section 4.3.3, I will show that this is not entirely correct. Although iteration yields the same semantics as functional application in a typical compositional grammar, its logical syntax does not match a surface-oriented syntax for natural language.

(1992) provide mechanisms for proving that a non-iterative polyadic quantifier is necessary and thus theoretically motivated. Here, I use the one in Keenan (1992) by which a polyadic quantifier is motivated as long as it can be proved to be unreducible to iteration. The attention is limited to polyadic quantifiers of type $\langle 2 \rangle$. Note, however, the further development in Dekker (2003) who formulates a theorem by which unreducibility can be proved for n-ary polyadic quantifiers.

Let us first define the notion of reducibility. In DEFINITION 2.18, I adapt the general definition in Dekker (2003, p. 551) to binary quantifiers. Recall from Section 2.1.2 that the function composition symbol "$\circ$" stands for iteration:

Definition 2.18 Reducibility
A type $\langle 2 \rangle$ quantifier Q *is (2)-reducible iff there are 2 type $\langle 1 \rangle$ quantifiers* Q_1 *and* Q_2*, such that* $\mathrm{Q} = \mathrm{Q}_1 \circ \mathrm{Q}_2$.

Keenan (1992) formulates two tests to check reducibility of polyadic quantifiers: *Reducibility Equivalence* and *Reducibility Characterization*. The former one is the simpler version and it is sufficient[12] for the polyadic quantifiers we have to test, so the attention will be limited to the theorem of Reducibility Equivalence as given below:[13]

Theorem 2.1 Reducibility Equivalence (RE)*:*
For every domain E *and* Q_1, Q_2*, reducible functions of type $\langle 2 \rangle$,*
$\mathrm{Q}_1 = \mathrm{Q}_2$ *iff for all* $\mathrm{A}, \mathrm{B} \subseteq \mathrm{E}$, $\mathrm{Q}_1(\mathrm{A}\times\mathrm{B}) = \mathrm{Q}_2(\mathrm{A}\times\mathrm{B})$

THEOREM 2.1 states that two reducible functions which yield the same values on all Cartesian product relations within a domain are identical.[14] Their value with respect to other binary relations need not be checked further. Let us take an example.

For the binary quantifiers defined in Section 2.1.3 we need a domain with at least 2 elements. Assume a domain E = {a, b}. The set of all its subsets is P(E) = {{ }, {a}, {b}, {a,b}} and the set of pairs of its elements is E^2 = {(a,a), (a,b), (b,a), (b,b)}. We need to determine all the binary Cartesian product relations defined on E. For this, we first determine P(E)×P(E), the set of all the possible pairs of subsets of E. By calculating the Cartesian product between the two subsets of E in each pair, we then obtain all the Cartesian product relations defined on the domain E, and thus all the relations with respect to which we have to check the truth conditions of the two binary quantifiers Q_1, Q_2.

The set P(E)×P(E)= {({}, {}), ({}, {a}), ({}, {b}), ({}, {a,b}), ({a},{}), ({a},{a}), ({a},{b}), ({a}, {a,b}), ({b},{}), ({b}, {a}), ({b}, {b}), ({b}, {a,b}), ({a,b},{}, ({a,b}, {a}), ({a,b}, {b}), ({a,b}, {a,b})}. Calculating the Cartesian product between the two sets in each pair gives us the set CP of all Cartesian product relations defined on E. The set CP is {({}×{}), ({}×{a}), ({}×{b}), ({}×{a,b}), ({a}×{}), ({a}×{a}), ({a}×{b}), ({a}×{a,b}), ({b}×{}), ({b}×{a}), ({b}×{b}), ({b}×{a,b}), ({a,b}×{}, ({a,b}×{a}), ({a,b}×{b}), ({a,b}×{a,b})} = {{}, {(a,a)}, {(a,b)}, {(a,a), (a,b)}, {(b,a)}, {(b,b)}, {(b,a), (b,b)}, {(a,a), (b,a)}, {(a,b), (b,b)}, {(a,a), (a,b), (b,a), (b,b)}}.

CP is a set of binary relations and thus a subset of P(E^2) = {{}, {(a,a)}, {(a,b)}, {(b,a)}, {(b,b)}, {(a,a), (a,b)}, {(a,a), (b,a)}, {(a,a), (b,b)}, {(a,b),(b,a)}, {(a,b), (b,b)}, {(b,a), (b,b)}, {(a,a), (a,b), (b,a)}, {(a,a), (a,b), (b,b)}, {(a,a), (b,a), (b,b)}, {(a,b), (b,a), (b,b)}, {(a,a), (a,b), (b,a), (b,b)}}. But note that P(E^2) is richer than CP, since it also contains binary relations that are not Cartesian products,

[12] See Keenan (1992) for examples of binary quantifiers whose unreducibility may only be proved by means of Reducibility Characterization.

[13] See Ben-Shalom (1994), Dekker (2003), and van Eijck (2005) for subsequent developments of Keenan's theorem.

[14] See Keenan (1992), pp. 218–219 for a detailed proof of this theorem.

as for instance {(a,a), (b,b)}, {(a,b), (b,a)}, {(a,a), (a,b), (b,a)}.[15] Solely by means of the Cartesian product on E we cannot obtain a relation like {(a,b), (b,a)}, for instance, because the Cartesian product requires that each element of the first set make up a pair with each element of the second set, while appearing first in the pair. If we have two pairs (a,b) and (b,a), both sets must contain the elements a and b. Thus the Cartesian product between these two sets must also contain the pairs (a,a) and (b,b). Iterations of monadic quantifiers can only distinguish between Cartesian product relations. Inherently polyadic quantifiers can also express truth conditions that are only met by relations that are not Cartesian products (e.g. {(a,b), (b,a)}). This is where the difference between iterations and non-interations becomes relevant for natural language and RE helps us to determine when a non-iteration cannot be restated as an iteration.

The way RE is used in proving the unreducibility of a quantifier Q_1 is to find an iteration Q_2 with a different semantics from that of Q_1, but which takes the same values on product relations. Showing that Q_2 is actually different from Q_1 is enough to contradict the initial assumption that Q_1 is also reducible. In order to show that Q_2 is different from Q_1, one has to find a binary relation which is not a Cartesian product and for which Q_2 and Q_1 yield different truth values.

We now apply this procedure to the binary quantifiers discussed in Section 2.1.3 in order to show that they are unreducible. Consider the ⟦(TWO, DIFFERENT)⟧ quantifier in (24a), repeated below:

(38) Two boys in my class date different girls.
$[\![(\text{TWO, DIFFERENT})]\!]^{[\![\text{BOY}]\!],\,[\![\text{GIRL}]\!]}([\![\text{DATE}]\!])$

There are circumstances in which $[\![(\text{TWO, DIFFERENT})]\!]^{[\![\text{BOY}]\!],\,[\![\text{GIRL}]\!]}$ yields the same truth value as the reducible iteration $0{\circ}0$ composed of the unary constant functions that are false of all unary relations. The iteration $0 \circ 0$ is thus false of all binary relations. Take the universe E to contain two boys ⟦BOY⟧ = $\{b_1, b_2\}$ and two girls ⟦GIRL⟧ = $\{g_1, g_2\}$, and $A \times B$ as a Cartesian product relation.

If the arbitrary set A contains no boys then the quantifier $[\![(\text{TWO, DIFFERENT})]\!]^{[\![\text{BOY}]\!],\,[\![\text{GIRL}]\!]}$ yields the value 0 for $A \times B$, since according to DEFINITION 2.11, the domain of the binary relation $A \times B$ (i.e. the set A) must contain at least two boys. The same value is obtained by applying $0 \circ 0$ to $A \times B$. So in this case ($[\![(\text{TWO, DIFFERENT})]\!]^{[\![\text{BOY}]\!],\,[\![\text{GIRL}]\!]}$ $(A \times B) = (0 \circ 0)(A \times B)$.

The minimal condition for a situation where it is possible to get a true value of the quantifier $[\![(\text{TWO, DIFFERENT})]\!]^{[\![\text{BOY}]\!],\,[\![\text{GIRL}]\!]}$ applied to the relation $A \times B$ is that of the set A containing two boys and the set B containing at least two girls. For any sets A and B containing less than 2 boys and 2 girls, respectively, $[\![(\text{TWO, DIFFERENT})]\!]^{[\![\text{BOY}]\!],\,[\![\text{GIRL}]\!]}$ $(A \times B)$ is always 0, so it takes the same value as $0 \circ 0$. So let us assume that A = ⟦BOY⟧ and B = ⟦GIRL⟧. Then $A \times B = \{(b_1, g_1), (b_1, g_2), (b_2, g_1), (b_2, g_2)\}$. But note that $[\![(\text{TWO, DIFFERENT})]\!]^{[\![\text{BOY}]\!],\,[\![\text{GIRL}]\!]}(A \times B)= 0$, since the girls that are dated by different boys are the same: each boy dates both girls. We can thus conclude that $[\![(\text{TWO, DIFFERENT})]\!]^{[\![\text{BOY}]\!],\,[\![\text{GIRL}]\!]}(A \times B)= (0 \circ 0)(A \times B)= 0$, for all the subsets A, B of E.

At this point, if both quantifiers were reducible, according to THEOREM 2.1, we would conclude that they are equal. But we only know for sure that $0 \circ 0$ is reducible, we do not know if $[\![(\text{TWO, DIFFERENT})]\!]^{[\![\text{BOY}]\!],\,[\![\text{GIRL}]\!]}$ is reducible as well. And it turns out that the two quantifiers are not identical, since there is a relation in E^2 for which they do not yield the same value. $[\![(\text{TWO, DIFFERENT})]\!]^{[\![\text{BOY}]\!],\,[\![\text{GIRL}]\!]}(\{(b_1, g_1),(b_2, g_2)\})= 1$, whereas $(0 \circ 0)(\{(b_1, g_1),(b_2, g_2)\})= 0$. This means that the assumed identity between the two binary quantifiers is wrong, which entails that the

[15]See Section 4.3.3 for a related discussion on the cardinality of $P(E^2)$ as being in general greater than that of $P(E) \times P(E)$ for $|E| > 2$.

assumption that both quantifiers are reducible must be false, too. Since we know that $(0 \circ 0)$ is reducible, it follows that $[\![(\text{TWO, DIFFERENT})]\!]^{[\![\text{BOY}]\!],\,[\![\text{GIRL}]\!]}$ is not reducible.

By means of RE, we can also prove that a binary cumulative quantifier is unreducible to an iteration of two unary quantifiers. Note that the cumulation $[\![(\text{FORTY, THIRTY-TWO})]\!]$ interpreted in (30) may take a relation to the same truth value as the iteration $[\![\text{EACH OF THE FORTY}]\!] \circ [\![\text{EXACTLY THIRTY-TWO}]\!]$:

(39) a. Forty contributors wrote thirty-two papers for the Handbook.
$Cum([\![\text{FORTY}]\!], [\![\text{THIRTY-TWO}]\!])^{[\![\text{CONTRIBUTOR}]\!],\,[\![\text{PAPER}]\!]}([\![\text{WRITE}]\!])$

b. Each of the forty contributors wrote exactly thirty-two articles for the Handbook.
$([\![\text{EACH OF THE FORTY}^{\text{CONTRIBUTOR}}]\!] \circ [\![\text{EXACTLY THIRTY-TWO}^{\text{PAPER}}]\!])$
$([\![\text{WRITE}]\!])$

The universe E should contain at least forty contributors and thirty-two papers for the two quantifiers above to be able to yield truth. So assume the two subsets of E $[\![\text{CONTRIBUTOR}]\!] = \{c_1, c_2, ..., c_{40}\}$ and $[\![\text{PAPER}]\!] = \{p_1, p_2, ..., p_{32}\}$.

If we take the set A to contain less than forty contributors, for instance, only thirty-nine, and B to contain thirty-two papers, both the cumulative quantifier $Cum([\![\text{FORTY}]\!], [\![\text{THIRTY-TWO}]\!])$ $^{[\![\text{CONTRIBUTOR}]\!],[\![\text{PAPER}]\!]}$ and the iteration $[\![(\text{EACH OF THE FORTY}^{\text{CONTRIBUTOR}})]\!] \circ [\![(\text{EXACTLY THIRTY-TWO}^{\text{PAPER}})]\!]$ are false of A$\times$B, because there are not forty contributors involved, which is a requirement both binary quantifiers have with respect to the monadic FORTY$^{\text{CONTRIBUTOR}}$ and EACH OF THE FORTY$^{\text{CONTRIBUTOR}}$. The same truth value is obtained if the set B contains any less than thirty-two papers. In this case, the monadic THIRTY-TWO$^{\text{PAPER}}$ and EXACTLY THIRTY-TWO$^{\text{PAPER}}$ make the two binary ones false, because there are not thirty-two papers in the co-domain of the relation A$\times$B.

The only case where the two binary quantifiers may yield truth is the one in which the sets A and B are identical to $[\![\text{CONTRIBUTOR}]\!]$ and $[\![\text{PAPER}]\!]$, respectively, so A$\times$B = $\{(c_1, p_1), (c_1, p_2), ..., (c_1, p_{32}), (c_2, p_1), (c_2, p_2), ..., (c_2, p_{32}), ..., (c_{40}, p_1), (c_{40}, p_2), ..., (c_{40}, p_{32})\}$. In this case, $Cum([\![\text{FORTY}]\!], [\![\text{THIRTY-TWO}]\!])^{[\![\text{CONTRIBUTOR}]\!],[\![\text{PAPER}]\!]}$ (A$\times$B) = 1, because there is a total of forty contributors and a total of thirty-two papers in the Cartesian product. For the iteration, $[\![(\text{EACH OF THE FORTY}^{\text{CONTRIBUTOR}})]\!] \circ [\![(\text{EXACTLY THIRTY-TWO}^{\text{PAPER}})]\!]$(A$\times$B) = 1 as well, because each of the forty contributors appears in thirty-two pairs in the Cartesian product.

Like in the previous example with DIFFERENT, if we knew that both $Cum([\![\text{FORTY}]\!], [\![\text{THIRTY-TWO}]\!])^{[\![\text{CONTRIBUTOR}]\!],[\![\text{PAPER}]\!]}$ and $[\![(\text{EACH OF THE FORTY}^{\text{CONTRIBUTOR}})]\!] \circ [\![(\text{EXACTLY THIRTY-TWO}^{\text{PAPER}})]\!]$ are reducible quantifiers, with the equality on Cartesian products, THEOREM 2.1 would lead us to conclude that the two quantifiers are equal on all binary relations. But we do not know if $Cum([\![\text{FORTY}]\!], [\![\text{THIRTY-TWO}]\!])^{[\![\text{CONTRIBUTOR}]\!],[\![\text{PAPER}]\!]}$ is reducible, so we cannot conclude this yet. And we can see that despite the identity on products, the two quantifiers are not identical, since there are binary relations on which they yield different truth values. For example, if we consider the relation $[\![\text{WRITE}]\!] = \{(c_1, p_1), (c_1, p_2), ..., (c_1, p_{32}), (c_2, p_1), (c_2, p_2), ..., (c_2, p_{32}), ..., (c_{40}, p_1), (c_{40}, p_2)\}$, then $Cum([\![\text{FORTY}]\!], [\![\text{THIRTY-TWO}]\!])^{[\![\text{CONTRIBUTOR}]\!],[\![\text{PAPER}]\!]}$ $([\![\text{WRITE}]\!])$= 1, since there are forty contributors and thirty-two papers in the $[\![\text{WRITE}]\!]$ relation. But $[\![(\text{EACH OF THE FORTY}^{\text{CONTRIBUTOR}})]\!] \circ [\![(\text{EXACTLY THIRTY-TWO}^{\text{PAPER}})]\!]([\![\text{WRITE}]\!])$= 0, because contributor c_{40} wrote only two papers, and not thirty-two as the truth conditions for the iteration $[\![(\text{EACH OF THE FORTY}^{\text{CONTRIBUTOR}})]\!] \circ [\![(\text{EXACTLY THIRTY-TWO}^{\text{PAPER}})]\!]$ require.

We can apply the same reasoning as for the example with DIFFERENT quantifiers and conclude that the cumulative quantifier $[\![$(FORTY, THIRTY-TWO)$]\!]$ is unreducible as well.

We saw that the difference between the polyadic quantifiers in Section 2.1.3 and iteration can be stated in terms of the relation between the semantics of the monadic quantifiers. In iteration, the monadic parts are independent of one another and the polyadic quantifier is a simple composition of them, so it does not bring anything besides their individual semantics. In Section 2.1.2, we arrived at iteration as a polyadic lift by a simple generalization of the notion of a monadic quantifier. By contrast, with non-iterations there is always a new relation that has to be established between the monadic parts. DIFFERENT/ SAME quantifiers, for instance, introduce a relation of non-equality/ equality between the elements of the co-domain from the perspective of a non-identity relation between the corresponding elements in the domain. In cumulations, the polyadic quantifier is a conjunction of the monadic parts.

It is precisely this difference in terms of the (in)dependence between the monadic parts that is exploited by THEOREM 2.1 in order to distinguish iterations from non-iterations, i.e. reducible quantifiers from unreducible ones. In the theorem, this distinction is formulated with respect to the (in)dependence between the domain and the co-domain of the binary relation to which a quantifier applies. In Cartesian product relations one may view the domain set and the co-domain set as independent unary relations. If the binary quantifier is an iteration, the monadic parts are interpreted with respect to each of the two sets, so two iterations yielding the same truth values on a Cartesian product must contain (semantically) equivalent monadic parts to be equal. But if the quantifier is a non-iteration, the value it takes on a Cartesian product does not fully describe its semantic behavior. A non-iteration characterizes binary relations in which the domain is independent from the co-domain (see the examples above). Given the dependence between the monadic parts of a non-iteration, the fact that a non-iteration and an iteration yield the same truth value on products does not entail that they are equal. In conclusion, it is the dependence between the monadic parts that distinguishes the semantics of non-iterations from that of iterations.

Conclusion In Section 2.1.3, I showed that quantifiers containing "different"/ "same", cumulation, and resumption are needed in order to analyze several instances of natural language quantification which cannot be accounted for by iteration. In this section, I proved that binary quantifiers containing "different" and binary cumulations are unreducible according to Keenan's theorem of Reducibility Equivalence. These cases indicate that natural language does employ unreducible polyadic quantifiers, so despite their complexity, linguistic theories should be open to the idea of using them. In Chapter 4 I will argue for an analysis of Romanian negative concord as an instance of resumption, and on that occasion I will return to the discussion of reducibility with respect to resumptions. Regarding compositionality, we will see later that polyadic quantifiers cannot be described compositionally in the traditional understanding of this notion as in Montague's Universal Grammar. This matter and its implications for the semantics of natural language quantification will be addressed in Section 4.3.3.

2.2 Romanian

In this section I offer a short theoretical background of Romanian which should facilitate the understanding of the empirical domain of this thesis. Romanian is an Eastern Romance language which, besides general characteristics shared with Western Romance languages, also displays similarities with Slavic and especially with Balkan languages. The most notable influence from Slavic is that of lexical borrowings. The Romance and the Balkan characteristics will be indicated below when I address inflection, agreement and word order.

2.2.1 Inflection

Like other Romance languages, Romanian has two grammatical numbers (i.e. singular and plural, but no dual), but unlike them, Romanian makes a three-way gender distinction between masculine, feminine and neuter.[16] Romanian distinguishes five case paradigms: nominative, accusative, dative, genitive, and vocative. The case paradigms display dative-genitive syncretism, a Balkan characteristic. In addition, Romanian also displays nominative-accusative syncretism and this brings it closer to the other Romance languages which make no case distinction. Distinct case inflections for nominative/ accusative and dative/ genitive appear only in (personal) pronominal declension.

Let us look at a few examples of nominal inflection: feminine *carte* ('book'), masculine *băiat* ('boy'), neuter *tablou* ('painting'). In (40) I give both the bare form of the noun (on the left) and the one containing the definite article (on the right). Romanian resembles Balkan languages in placing the definite article in post-nominal position:

(40) a. Inflection of *fată* ('girl') - bare

CASE	SING	PLURAL
NOM-ACC	fat-ă	fet-e
DAT-GEN	fet-e	fet-e
VOCATIVE	fat-o	-

- with definite article

CASE	SING	PLURAL
NOM-ACC	fat-a	fete-le
DAT-GEN	fete-i	fete-lor
VOCATIVE	-	fetelor

b. Inflection of *băiat* ('boy') - bare

CASE	SING	PLURAL
NOM-ACC	băiat	băieţ-i
DAT-GEN	băiat	băieţ-i
VOCATIVE	băiet-e	băieţi

- with definite article[17]

CASE	SING	PLURAL
NOM-ACC	băiat-(u)l	băieţi-i
DAT-GEN	băiat-(u)lui	băieţi-lor
VOCATIVE	băiat-(u)l-e	băieţilor

c. Inflection of *tablou* ('painting') - bare

CASE	SING	PLURAL
NOM-ACC	tablou	tablo-uri
DAT-GEN	tablou	tablo-uri
VOCATIVE	-	-

- with definite article

CASE	SING	PLURAL
NOM-ACC	tablou-l	tablouri-le
DAT-GEN	tablou-lui	tablouri-lor
VOCATIVE	-	-

The vocative case inflection is solely used with animate nouns. Moreover, it has only two specific endings: *-o* for feminine, and *-e* for masculine. The remaining vocative forms are borrowed from other cases: nominative-accusative or dative-genitive.

Unlike nominal inflection, verbal inflection in Romanian is very rich, just like in other Romance languages and Latin. Verbs are classified according to four conjugations and inflect for mood and tense. Aspectual differences are not grammaticalized in Romanian. There are five personal (finite)

[16] Based on the lack of semantic individuality of neuter nouns, and the fact that they display morphological syncretism with masculine singular and feminine plural forms, neuter has been argued not to be a gender class. A recent approach is that of Bateman and Polinsky (2006), but here I follow the traditional view in GA (1966) which treats neuter as a gender.

[17] The vowel *u* is a phonological connector for the two consonants *t* and *l*.

moods (indicative, subjunctive, conditional-optative, imperative and presumptive), four non-finite moods (infinitive, past participle, present participle[18] and supine), and three diatheses (active, passive, reflexive).

Simple verb forms in Romanian include only the lexical root, suffixes and endings corresponding to persons. I give an example for the verb *(a) chema* ('to call') with its simple inflection forms in (42a) below. Complex verb forms are made up of auxiliary verbs which are added to some simple form of the base verb (see *a chema* in (42b)). The subjunctive mood in Romanian contains the conjunction *să* which in subjunctive clauses also functions like a clause connector (i.e. marker), as for instance in (41) below:

(41) Ion vrea să citească.
John wants SĂ read
'John wants to read.'

Romanian abandons the typical Romance use of the infinitive form, and follows the Balkan tendency of employing the subjunctive instead.

(42) Inflection of *a chema* ('to call'), 2nd person, singular, active diathesis

a. Simple verb forms

MOOD	TENSE	VERBAL FORM
indicative	present	*chem-i*
	imperfect	*chem-a-i*
	simple perfect	*chem-a-şi*
	past perfect	*chem-a-se-şi*
subjunctive	present	*să chem-i*
imperative		*cheamă*
infinitive	present	*a chem-a*
present participle		*chem-înd*
past participle		*chem-a-t*
supine		*(de) chem-a-t*

b. Complex verb forms

MOOD	TENSE	VERBAL FORM
indicative	present perfect	*ai chemat*
	future	*vei chema*
	future perfect	*vei fi chemat*
subjunctive	perfect	*să fi chemat*
conditional	present	*ai chema*
	perfect	*ai fi chemat*
presumptive	present	*vei fi chemînd*
		să fi chemînd
		ai fi chemînd
	perfect	*vei fi chemat*
infinitive	perfect	*a fi chemat*

[18]In the Romanian linguistic literature, this form is also referred to as "gerund" (from Romanian *gerunziu*), although the functionality of this non-finite verb form is more similar to the English present participle than to the gerund.

Romanian has three auxiliary verbs that are used in building complex verb forms: *a avea*, *a fi*, and *a vrea*. The auxiliary *a avea* ('to have', 2nd person singular *ai*) takes part in forming indicative present perfect and conditional mood. The verb *a fi* ('to be') contributes two auxiliary forms: the short infinitive *fi* takes part in the formation of all perfect forms except for present perfect and in the formation of presumptive mood; the present indicative form of *a fi* is used in building the passive diathesis (e.g. *eşti chemat* - '(you) are called (for)'). The auxiliary *a vrea* ('to want', 2nd person singular *vei*) is part of both future forms, of perfect presumptive, and it also appears in one of the present presumptive forms.

Simple verbal forms which are used in deriving the complex ones are: past participle (*chemat* in (42b)) which appears in all the (complex) perfect forms; short infinitive which is part of the present future and the present conditional; and present participle (see *chemînd* above) which appears in present presumptive verb forms (see (42)).

2.2.2 Agreement

There are three types of agreement in Romanian: noun - specifier, noun - adjective, and subject - verb. Noun - specifier agreement means that determiners agree in case, number and gender with the noun they specify. This can already be seen in the nominal paradigms with the definite article (a post nominal specifier) in (40). The definite article *-a* combines with nominal forms which carry nominative case, singular number and feminine gender. It forms a minimal pair with (feminine) *-i* with respect to case, with *-le* with respect to number, and with *-(u)l* with respect to gender. Similarly, the article *-l* which carries nominative case, singular number and masculine gender forms a minimal pair with *-i* in terms of number, and with *-lui* in terms of case. The noun such determiners combine with tells us the gender: e.g. the article *-l* has neuter gender with the neuter noun *tablou* and masculine with *băiat*.

Noun - adjective agreement also concerns all three nominal inflection paradigms: case, number and gender. (43) is an example of noun modification by the adjective *frumos* ('beautiful'), applied to all three categories of nouns in (40)[19]:

(43) Noun - adjective agreement

a. Feminine

CASE	SINGULAR	PLURAL
NOM-ACC	fat-ă frumoas-ă	fet-e frumoas-e
DAT-GEN	fet-e frumoas-e	fet-e frumoas-e

b. Masculine

CASE	SINGULAR	PLURAL
NOM-ACC	băiat frumos	băieţ-i frumoş-i
DAT-GEN	băiat frumos	băieţ-i frumoş-i

[19]Adjectives do not exhibit a special ending for vocative case, and nouns in vocative form cannot be modified by adjectives. Thus a sentence like *#Fato frumoasă, vino aici!* ('Beautiful girl, come here!') is not attested.

c. Neuter

CASE	SINGULAR	PLURAL
NOM-ACC	tablou frumos	tablo-uri frumoas-e
DAT-GEN	tablou frumos	tablo-uri frumoas-e

Note that adjectives in Romanian usually follow the noun they modify (like in Romance languages in general). Some adjectives can be preposed, but the construction is highly marked.

(43) shows that adjectives have two inflectional endings: *-ă* vs. *-e* for feminine, and $\emptyset$ vs. *-i* for masculine. Like with specifiers, the agreement features can be determined on the basis of the noun: *frumoase* in *fete frumoase* has feminine gender, but it is neuter in *tablouri frumoase*.

Subject and verb agree in person and number. See for instance the present indicative forms of the verb *a chema* for all three person specifications and singular/ plural number:

(44) Complete verbal inflection for present indicative of *a chema* ('to call')

NUMBER	1ST PERSON	2ND PERSON	3RD PERSON
SINGULAR	eu chem (I call)	tu chem-i (you call)	el/ ea cheam-ă (he/ she calls)
PLURAL	noi chem-ă-m (we call)	voi chem-a-ţi (you call)	ei/ ele cheam-ă (they$_{m/f}$ call)

The person and number of a verb form are indicated by the agreement with the subject: thus, *cheamă* in *ea cheamă* is singular, and in *ele cheamă* is plural.

2.2.3 Word order

Romanian is a free word order language, although like in other Romance languages, the order is much less flexible than in Latin. In principle, syntactic constituents exhibit free order in the sentence, but they cannot be split (with the exception of the VP). Thus a sentence with a subject, a transitive verb, a direct object and an adverb allows 24 permutations of the four constituents and they are all grammatical. Most of them have slightly different interpretations triggered by a change in the information structure. In (45) I give a few such permutations:

(45) a. Ion a spart un geam ieri.
John has broken a window yesterday
'John broke a window yesterday.'

b. A spart Ion un geam ieri.

c. Ieri Ion un geam a spart.

d. Un geam ieri a spart Ion.

e. ...

(45a) is the most common word order in a sentence, which means that Romanian tends to be an SVO[20] language. The sentence in (45b) is also neutral with respect to information structure, but in (45c), the

[20]In the GB/ Minimalist tradition, Cornilescu (1997) argued that Romanian is a VSO language, and the subject in preverbal position is a case of topicalization in the sense of Rizzi (1997). This claim is also confirmed by the fact that a neutral answer to a question like 'What happened?' is the one in (45b) which displays a VSO order. We will not attempt to determine whether Romanian is an SVO or a VSO language, since this has no influence on the analysis in this thesis.

adverb is understood as a topic, while in (45d) the direct object is understood as topicalized and the adverb as focused.

The flexible word order and the case syncretism between nominative and accusative would normally lead to ambiguity between the subject and the direct object if both of them can be interpreted as agents (see also Niculescu (1965), Cornilescu (2000*b*), Ionescu (2001)). In Romanian, this is avoided by a special marking of the direct object with the preposition *pe* 'on' which loses its original predicative status. Thus the sentence in (46a), which is ambiguous with respect to whether *fata* ('the girl') or *băiatul* ('the boy') is the subject, will be disambiguated by means of *pe* as (46b) or (46c):

(46) a. A certat fata băiatul.
has scolded girl-the boy-the

b. L-a certat fata *pe* băiat.
CL-has scolded girl-the PE boy
'The girl scolded the boy.'

c. A certat-o *pe* fată băiatul.
has scolded-CL PE girl boy-the
'The boy scolded the girl.'

Besides *pe* marking, Romanian makes heavy use of clitic doubling which can be observed in (46b) - (46c) and which goes beyond verb - direct object constructions.[21]

In the following section I sketch a grammar for the basic sentence structure of Romanian in the framework of HPSG which will be later used in my account of negative concord.

2.3 HPSG

Having looked at the semantic framework and a few general observations about Romanian, let us now concentrate on HPSG, the linguistic theory that will be employed to provide a syntax-semantics interface for Romanian negative concord in Chapter 5.

HPSG is a generative linguistic theory that evolved in the tradition of Generalized Phrase Structure Grammar (GPSG, Gazdar et al. (1985)), and was mostly influenced by Lexical Functional Grammar (LFG, Bresnan (1982)), Government and Binding (GB, Chomsky (1981)) and Categorial Grammar (CG, Ajdukiewicz (1935)). Unlike GB, HPSG is a non-derivational framework, that is, linguistic principles do not apply in a successive order. Furthermore, HPSG is a monostratal theory in which various linguistic aspects interact simultaneously.

In this section I briefly present the basic ideas and mechanisms of HPSG as described in Pollard and Sag (1994) for which I employ RSRL (Relational Speciate Re-entrant Language) of Richter (2004*b*) as the logical formalism. I start with a short informal description of the logical foundations of HPSG as a model-theoretic grammar framework in Section 2.3.1 and then I develop an HPSG grammar of a fragment of Romanian in Section 2.3.2. This grammar will be extended to include an account of negative concord in Chapter 5.

2.3.1 HPSG as a model-theoretic grammar

Grammars describe fragments of natural language. In the model-theoretic view, we write a grammar as a logical theory and define models of it. A certain model, in our case, the exhaustive model, will

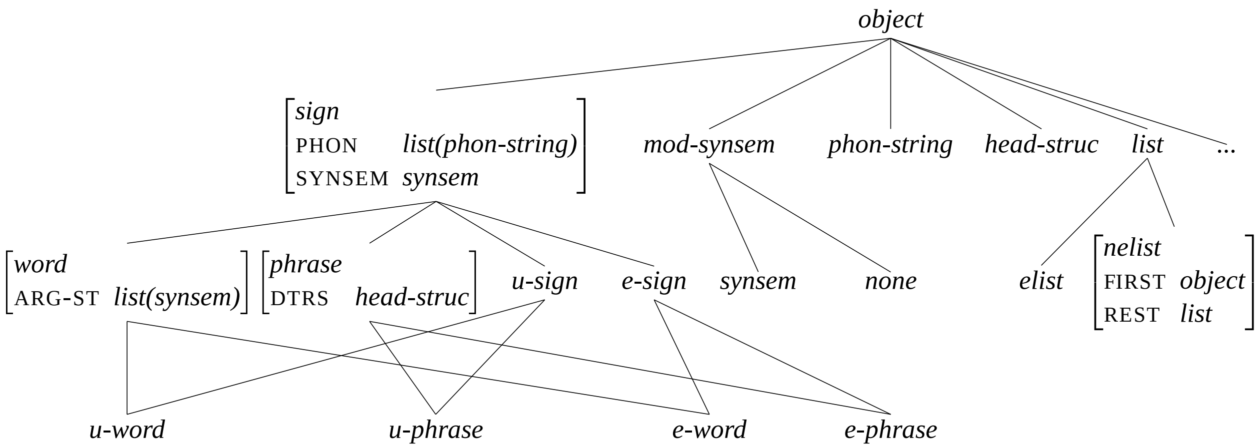
[21] See Dobrovie-Sorin (1994) for an extended discussion of Romanian clitic doubling.

give us the natural language fragment that we want to describe with the grammar: the objects in the exhaustive model are the objects of the natural language.

HPSG is a model-theoretic grammar framework in which a grammar Γ is constructed as a pair of a signature Σ and a theory Θ: $\Gamma = \langle \Sigma, \Theta \rangle$ (Richter (2004*b*)). The signature is the alphabet: it specifies the potential linguistic objects. The theory determines which of these objects are actual linguistic objects in the denotation of the grammar.

The signature Σ The signature declares a set of *sorts* (the non-logical symbols) organized in a *sort hierarchy*, the set of *attributes* and the *appropriateness conditions* between sorts and corresponding attributes, as well as a set of *relations* with their arity specification. Let us take the sort hierarchy in (47) as an example.

(47) *Sort hierarchy: Graph notation*



All the sorts are *subsumed* by one most general sort, in our case *object*. The more general sorts are called *supersorts*, the ones that they subsume are *subsorts*. If a sort A subsumes a sort B and there is no other sort C that subsumes B while being subsumed by A, we say that A *immediately subsumes* B. Thus *object* subsumes all the sorts in the signature, but immediately subsumes only the sorts *sign*, *mod-synsem*, *phon-string*, *head-struc*, and *list*. When a subsort is immediately subsumed by two sorts, we have *multiple inheritance*. This is the case for instance with *u(nembedded)-word* which is subsumed by both *word* and *u(nembedded)-sign*.[22] The sorts that do not subsume any further sorts are called maximally specific sorts, or *species*. The set of species in (47) contains the following elements: *u-word, u-phrase, e-word, e-phrase, synsem, none, phon-string, head-struc, elist,* and *nelist*.

Besides the subsumption relation between sorts, the sort hierarchy also specifies the appropriate *attributes* for each sort. Attributes are usually written in capital letters and receive a value of a certain sort. For instance, the sort *sign* has two attributes PHON and SYNSEM which specify the sign's "phonological" structure and its "syntax-semantics", respectively. The value for the former must be a *list* of *phon-string*s (phonological strings). The value of the latter is of sort *synsem*, a subsort of the more general sort *mod-synsem* which will be explained in relation to (56) below.

[22]In Chapter 5 we will use the sort *u-sign* to formulate constraints on utterances (see Richter (2007) for the importance of unembedded signs). For more background and a detailed discussion of how to distinguish *u-signs* from *e(mbedded)-signs*, see also Richter (1997, pp. 135–136).

The descriptions of linguistic objects which contain the information about their attributes with appropriate values are called *attribute-value matrices* (AVMs). In the sort hierarchy above, the sorts *sign*, *word*, *phrase*, and *nelist* introduce new attributes which are given within AVMs. Attributes are *inherited* by the more specific sorts from the sort that subsumes them. Thus, besides their attribute DTRS ("daughters"), *phrases* inherit the attributes PHON-STR and SYNSEM from *sign*. The same holds for *words* which also introduce an attribute ARG-ST ("argument structure").

The sort *list* is partitioned into *elist* (denoting empty lists) and *nelist* (denoting non-empty lists). While the former has no attributes, the latter has internal structure organized through the attributes FIRST with value of sort *object* and REST with value of sort *list*. So non-empty lists can contain elements of any sort subsumed by *object* in the hierarchy. The parametric sorts *list(phon-string)*/ *list(synsem)* of the attributes PHON/ ARG-ST are used as a short notation for a list that contains only elements of sort *phon-string*/ *synsem*. For a technical discussion, see Penn (1998, 2000).

In the practice of HPSG grammar writing, sort hierarchies become very complex and less transparent, so linguists usually present only those parts of the hierarchy which are directly relevant for the discussion. At the same time, extensive use is made of abbreviations, especially within AVMs, and this practice will be adopted here as well.

Besides the sort hierarchy, the attributes, and the appropriateness conditions, the signature also declares the *relations* that are employed in the grammar. Relations are used to formulate the principles in the theory of the grammar. The meaning of relation symbols is fixed together with the other principles in the theory, so the definitions of the relations are principles themselves. One frequently used relation in HPSG grammars is `append`. The notation `append/3` gives us the name of the relation and its arity.

Relations are not sorts in the signature, so they cannot be placed in the sort hierarchy in (47). There is, however, another notational variant of an HPSG signature, usually employed in grammar implementation, where we also declare the relations with their arity. This is given in (48). The hierarchical structure of the sorts in this notation of a signature is represented as indentation. This notation will be used in Chapter 5.

(48) *Sort hierarchy: Notational variant*

```
object
  sign PHON list(phon-string)
       SYNSEM synsem
    word ARG-ST list(synsem)
      u-word
      e-word
    phrase DTRS head-struc
      u-phrase
      e-phrase
    u-sign
      u-word
      u-phrase
    e-sign
      e-word
      e-phrase
  mod-synsem
    synsem
    none
  phon-string
  head-struc
```

```
  list
    elist
    nelist FIRST object
           REST  list
Relations
append/3
```

The theory Θ The theory is a set of descriptions that employ non-logical symbols from the signature and logical operators like conjunction '$\wedge$', disjunction '$\vee$', implication '$\rightarrow$', double implication '$\leftrightarrow$', universal '$\forall$' and existential '$\exists$' quantification. It should be noted that '$\forall$' and '$\exists$' are not the first order logic quantifiers, although they have a similar behavior (see Richter (2004*b*, Sec. 4.1)).

There are two kinds of principles in the grammar: those that define the meaning of relation symbols (formulated as double implications) and those that constrain the objects in the grammar (usually formulated as implications). The constraints introduced by principles apply to all the objects in the denotation of the grammar.

Let us take a look at THE append PRINCIPLE:

(49) THE append PRINCIPLE

$$\forall\boxed{1}\ \forall\boxed{2}\ \forall\boxed{3}\left(\begin{array}{l}\texttt{append}(\boxed{1},\boxed{2},\boxed{3}) \leftrightarrow \\ \left(\left(\boxed{1}[\textit{elist}] \wedge \boxed{2}[\textit{list}] \wedge \boxed{2} = \boxed{3}\right) \vee \\ \exists\boxed{4}\ \exists\boxed{5}\ \exists\boxed{6}\left(\begin{array}{l}\boxed{1}\begin{bmatrix}\textit{list}\\ \text{FIRST} & \boxed{4}[\textit{object}]\\ \text{REST} & \boxed{5}[\textit{list}]\end{bmatrix} \wedge \boxed{3}\begin{bmatrix}\textit{list}\\ \text{FIRST} & \boxed{4}[\textit{object}]\\ \text{REST} & \boxed{6}[\textit{list}]\end{bmatrix}\\ \wedge\ \texttt{append}(\boxed{5},\boxed{2},\boxed{6})\end{array}\right)\right)\end{array}\right)$$

(49) defines the meaning of the relation append. To the left of the double implication we have the relation with its three arguments and to the right we specify the conditions that have to be fullfilled for the relation to hold. In our case, any three lists $\boxed{1}$, $\boxed{2}$, and $\boxed{3}$ are in the append relation if and only if one of the two conditions holds: 1) $\boxed{1}$ is an empty list and $\boxed{2} = \boxed{3}$, or 2) the first element on the list $\boxed{1}$ appears as the first element on the list $\boxed{3}$ and the append relation holds of the rest $\boxed{5}$ of list $\boxed{1}$, the list $\boxed{2}$ and the rest $\boxed{6}$ of the list $\boxed{3}$. In HPSG, the append relation is often written as an infix operation by means of the symbol '$\oplus$'. In the functional notation, we represent the value of $\boxed{3}$ as $\boxed{1} \oplus \boxed{2}$, which means that append($\boxed{1}$,$\boxed{2}$,$\boxed{3}$) holds.

Principles that do not define relations are usually formulated as implications. For instance THE IMMEDIATE DOMINANCE (ID) PRINCIPLE says that each object of sort *phrase* must obey one of a number n of ID schemata formulated in the grammar. The ID schemata are descriptions that constrain the kinds of phrases that can be part of the grammar. The ID schemata for the present grammar will be formulated in (67).

(50) THE ID PRINCIPLE
phrase $\rightarrow$ (SCHEMA-1 $\vee$ SCHEMA-2 $\vee$ ... $\vee$ SCHEMA-n)

The denotation of an HPSG grammar In HPSG there is a correspondence between the grammar and the natural language such that the latter can be viewed as a particular model of the former.

To determine the models of an HPSG grammar we first need to assume a universe U that contains all the objects denoted by the grammar. We then define a function S that assigns a denotation (objects from U) to each sort in the signature, via the species that it subsumes. Each object in the denotation of the grammar instantiates a particular species, so supersorts are collections of objects of various species.

The attribute interpretation function A provides a denotation for the attribute symbols. This function respects all the appropriateness conditions in Σ: if an attribute is appropriate for a species it will also be interpreted for all the objects of that species (i.e. objects in the denotation of the grammar must be complete) and its value will be a collection of objects in the denotation of the grammar.

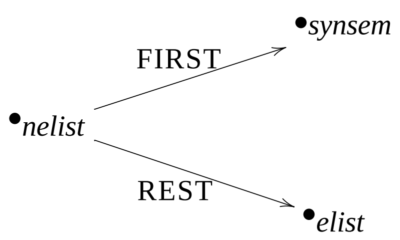
The function R interprets n-ary relations by assigning them the corresponding sets of n-tuples of objects in the universe U. On the basis of the domain U and the functions S, A and R, we can now define the notion of an interpretation of a signature Σ:[23]

Definition 2.19 *For each signature Σ,* I *is a Σ interpretation iff:*
I *is a quadruple $\langle U, S, A, R\rangle$,*
U is a set that contains all the objects of the domain,
S is a total function from U to the set of species in Σ,
A is a total function from the set of attributes in Σ to the set of partial functions from U to U,[24]
R is a total function from the set of n-ary relations in Σ to the set of n-tuples in U^n.

In grammar writing the signature generates descriptions like (51). We interpret (51) as a collection of non-empty list objects in U whose single element is an object of sort *synsem*. We represent these objects (i.e. the interpretation of (51)) by means of a graph as in (52). The nodes symbolize objects in U and are labeled by their sorts. The arrows stand for the interpretation of attributes and are labeled by their attribute names. The origin of the arrow is an object in the domain of the partial function that the attribute denotes and its endpoint is an object in the range of that function. In (52) we have a non-emtpy list object whose first element is a synsem object and whose rest is an empty list.[25] The denotation of the relation append in (52) is the empty set.

(51) $\begin{bmatrix} \textit{nelist} & \\ \text{FIRST} & \textit{synsem} \\ \text{REST} & \textit{elist} \end{bmatrix}$

(52) Interpretation of (51)



append = {}

[23]DEFINITION 2.19 is only an informal version of the precise definition in Richter (2004*b*, pp. 77–78) and Richter (2004*a*, pp. 21–22). In particular, the definition of the function R is more complex in a way that is not relevant for the present discussion.

[24]Each attribute denotes a partial function from entities to entities (Richter (2004*b*)).

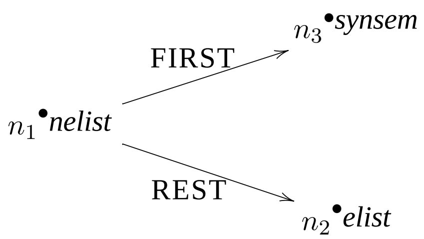
[25]The sort *synsem* has no attributes in our signature (47), so it doesn't receive any attributes in the interpretation (52) either. Later in this section we will also consider attributes of synsem objects, but for now we keep this example simple.

We usually say that the non-empty lists with a unique element of sort *synsem* in (52) *satisfy* the description in (51). The objects of sort *synsem* and *elist* in (52), however, do not satisfy (51), because they are not non-empty lists. A configuration is *licensed* by a description if every node (i.e. every object in it) satisfies the description. Thus (51) does not license (52), because the objects synsem and elist in the latter do not satisfy the description in (51). Principles in the theory of the grammar are also descriptions. Every object in the intended interpretation of our grammar must be licensed by all the principles in the grammar.

We define a *model* of an HPSG grammar as an interpretation of the signature in which every object is licensed by each description in the theory Θ. Let us check whether the interpretation in (52), call it I_{52}, is a model of the grammar developed so far.

Our grammar consists of the signature Σ and the theory Θ. As shown above, I_{52} is an interpretation of Σ. The theory Θ contains two principles: the append PRINCIPLE and the ID PRINCIPLE. The ID PRINCIPLE constrains objects of sort *phrase*.[26] In I_{52} there are no objects of sort *phrase*, so the ID PRINCIPLE principle is vacuously satisfied by I_{52}. Consider now the append PRINCIPLE. The append relation has an empty denotation in I_{52}. However, I_{52} contains objects of sort *list* and the append PRINCIPLE enforces that they are in an appropriate relationship with respect to the relation append. If we label the non-empty list node n_1 and the empty list node n_2, the denotation of the append relation will contain three tuples: $(n_2, n_2, n_2), (n_2, n_1, n_1)$, and (n_1, n_2, n_1). In conclusion, I_{52} is not a model of our grammar, because the append PRINCIPLE is not satisfied. We can give another interpretation similar to I_{52}, call it I_{53}, which in addition contains the full denotation of append (see (53)). I_{53} is licensed by both our principles, so it is a model of our grammar:

(53) A model of our grammar



append = $\{(n_2, n_2, n_2), (n_2, n_1, n_1), (n_1, n_2, n_1)\}$

We can find an infinite number of models of a consistent grammar. Obviously not all the models of an HPSG grammar can be identified with the natural language fragment that we want to be denoted by our grammar. For instance, the above exemplified model is too poor. A model for an HPSG grammar should also contain unembedded/ embedded phrases (of sort *u-phrase*/ *e-phrase*), unembedded/ embedded words (of sort *u-word*/ *e-word*) etc. For the denotation of an HPSG grammar we need a so-called *exhaustive model*. Informally, an exhaustive model contains instances of all the potential configurations of objects that are well-formed with respect to the signature and are licensed by the principles of the grammar. Thus we can identify the intended exhaustive model of an HPSG grammar with the natural language fragment that the grammar is written to denote.

[26]The ID PRINCIPLE formulated in (50) uses a disjunction of ID Schemata which we have not defined for our grammar yet (they are given in (67)). To keep the grammar simple, consider for the moment that these schemata are just a finite number of different phrases: phrase-1, phrase-2, .., phrase-n.

2.3.2 An HPSG grammar of a fragment of Romanian

In this section I will go into the details of HPSG grammar writing with a direct application to a small fragment of Romanian. This grammar will later be taken as the starting point for the HPSG analysis of Romanian negative concord.

2.3.2.1 Words and phrases

The lexicon In HPSG, the lexicon is defined as a finite set of lexical entries (L-1, L-2, ..., L-n) which denote the words admitted by the grammar and a set LR of words that are licensed as the output of lexical rules (Höhle (1999), Meurers (1999)). Technically, the lexicon is part of the theory of an HPSG grammar and is specified as a constraint on words:

(54) THE *word* PRINCIPLE
$$\textit{word} \rightarrow \left(\text{L-1} \vee \text{L-2} \vee \ldots \vee \text{L-}n \vee \text{LR}\right)$$

Given the disjunction in the consequent of THE *word* PRINCIPLE, every object of sort *word* in the grammar has to satisfy one of the given lexical entries or be the output of a lexical rule.

Lexical entries are partial descriptions of words and specify all the particular information about a word that is not provided by the signature or the principles in the grammar. The inflectional variants of a word are usually obtained by means of a lexical rule.[27] Thus various verb forms like *read, reads, reading* receive a single lexical entry which contains the least marked form and the other forms are derived by lexical rules. The word that undergoes a lexical rule is called *input* and the result is the *output* of the lexical rule.

Before we exemplify lexical entries, let us take a look at objects of sort *synsem*, as they are the most important in our grammar:

(55)
$$\begin{bmatrix} \textit{synsem} & \\ \text{LOC} & \begin{bmatrix} \textit{local} & \\ \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{head} \\ \text{VAL} & \textit{valence} \\ \text{MARKING} & \textit{marking} \end{bmatrix} \\ \text{CONT} & \textit{content} \end{bmatrix} \\ \text{NLOC} & \textit{nonlocal} \end{bmatrix}$$

Objects of sort *synsem* come with two attributes: LOC ("local") with value of sort *local* and NLOC ("nonlocal") with value of sort *nonlocal*. The latter is useful in the analysis of unbounded dependency constructions and will not be addressed here.[28] Objects of sort *local* carry the local information about the syntax-semantics of an object and have at least the following two attributes: CAT ("category") with value of sort *category* and CONT ("content") with value of sort *content*. The CAT attribute specifies the (morpho-)syntactic information of a sign, except for its constituent structure which is given under DTRS for phrases. CONT hosts the semantic information of a sign. The information under CAT is distributed over the HEAD, VAL ("valence"), and MARKING attributes with values of sort *head*, *valence*, and *marking*, respectively.

---

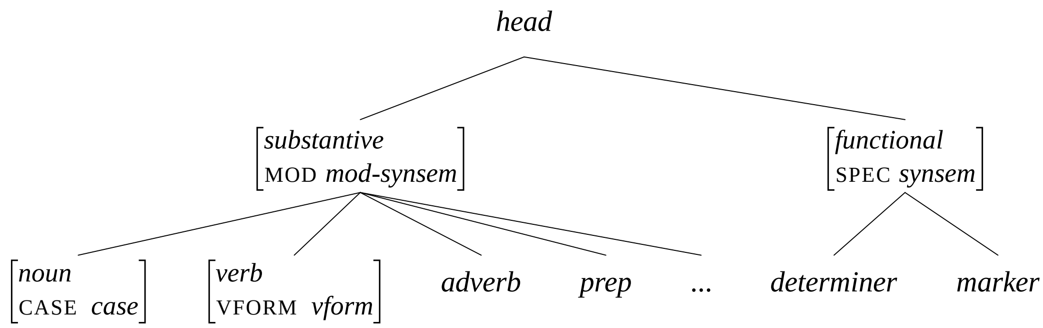
[27] I assume the view on lexical rules in Meurers (1999, Ch. 5) in a formalization that can be integrated in RSRL (Richter (2004*b*, pp. 318–319)).

[28] The reader is referred to Pollard and Sag (1994, Ch. 4), Ginzburg and Sag (2000, Ch. 5), and Bouma et al. (2001) for the value of the NLOC attribute.

Note that with the description of synsems, we introduce new sorts and thus enrich our signature. The newly introduced sorts that do not have a supersort will be immediately subsumed by *object* in (47). This is the case of *local*, *nonlocal*, *head*, *valence*, *marking*, and *content* in (55).

Words of different syntactic categories are distinguished on the basis of the HEAD value which we specify in terms of the following sort hierarchy for *head* objects.

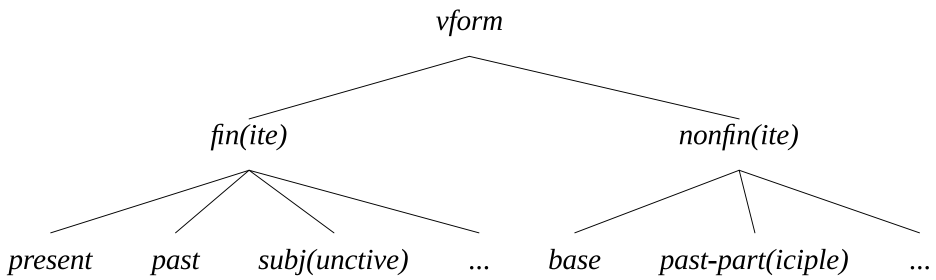
(56)



Head has two immediate subsorts corresponding to *substantive* and *functional* categories. The sort *functional* usually includes *determiner*s and *marker*s (i.e. complementizers). Functional categories have an attribute SPEC, whose value is the synsem object within the sign that they “specify”.

Substantive categories (including nouns, verbs, adverbs, and prepositions) may modify (MOD) other synsems. The value for MOD can be *none* or *synsem* (see (47)). If a sign modifies another sign, the value for MOD is *synsem* and is identified with the SYNSEM value of the sign that is modified. If a sign does not modify other signs, its MOD value is of sort *none*. Among substantive categories, nouns specify their CASE value as *case* (e.g. *nominative*, *accusative*, *genitive*, *dative*, for Romanian). The case information of a noun is not important for our study, so I will not pay particular attention to it and I will assume that we have a theory of case that gives us the right results. Similarly, no particular position is taken with respect to agreement. Any kind of analysis integrating agreement should in principle be compatible with our grammar. Verbs specify their tense/ mood form under VFORM with value of sort *vform*. For the sort *vform* we assume the subsorts in (57).

(57)

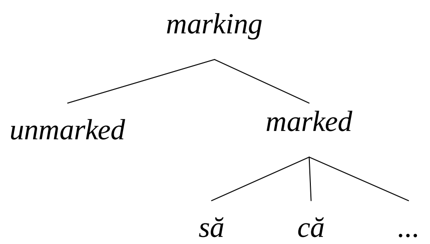


The attribute VAL in (55) describes the subcategorization properties of a sign. The sort *valence* has three attributes: SUBJ (“subject”), SPR (“specifier”) and COMPS (“complements”). The values of these attributes give us the subject, specifier or complements that the sign subcategorizes for. The value for the three valence attributes in (58) is a list of synsems. This means that heads subcategorize only for syntax-semantic information and not for full linguistic signs.

(58) $$\begin{bmatrix} \textit{valence} & \\ \textsc{subj} & \textit{list(synsem)} \\ \textsc{spr} & \textit{list(synsem)} \\ \textsc{comps} & \textit{list(synsem)} \end{bmatrix}$$

The attribute MARKING indicates whether a linguistic object is marked by a marker or not: see the *marking* subsorts in (60), where the species *să* and *că* stand for the subjunctive marker and the ‘that’-complementizer in Romanian.[29]

(60)



Now we have enough information about synsems to give an example of a lexical entry. Take the Romanian verb *citi* ‘read’ below:

(61) *citi* (‘read’)

$$\begin{bmatrix} \textit{word} \\ \text{PHON} \ \langle \texttt{citi} \rangle \\ \text{SYNSEM} \begin{bmatrix} \textit{synsem} \\ \text{LOC} \begin{bmatrix} \textit{local} \\ \text{CAT} \begin{bmatrix} \textit{category} \\ \text{HEAD} \begin{bmatrix} \textit{verb} \\ \text{VFORM} \ \textit{base} \end{bmatrix} \\ \text{VAL} \begin{bmatrix} \textit{valence} \\ \text{SUBJ} \ \langle \text{NP} \rangle \\ \text{COMPS} \ \langle \text{NP} \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

This lexical entry says that the word *citi* has a phonology list made up of one phon-string `citi` and a HEAD value of sort *verb* with a base verbal form, and that it subcategorizes for a list of one subject (an NP) and a list of one (NP) complement. We represent lists by means of angle brackets.

The information that is not given in the lexical entry comes from the signature and the principles of the grammar. The signature provides us with the information that the CONT value is of sort *content* and that the MOD value is of sort *mod-synsem*. Other pieces of information come from principles. For instance, all the words in the grammar that are not markers (i.e. their HEAD value is not *marker*) receive a MARKING value of sort *unmarked*, so we can formulate the principle in (62):

(62) $$\begin{bmatrix} \textit{word} \\ \text{SYNSEM}|\text{LOC}|\text{CAT}|\text{HEAD} \ \neg \ \textit{marker} \end{bmatrix} \rightarrow \begin{bmatrix} \text{SYNSEM}|\text{LOC}|\text{CAT}|\text{MARKING} \ \textit{unmarked} \end{bmatrix}$$

Similarly, we know that verbs always receive an empty list for the SPR attribute: only subjects are subcategorized for by verbs, while specifiers appear in the nominal domain. The principle in (63) allows us to specify this generalization for all the verbs:

[29] An example with the subjunctive marker is given in (59):

(59) Ion i-a cerut Mariei *să* vină.
John CL-has asked Mary SJ come
‘John asked Mary to come.’

Clauses with a subjunctive marker will be analyzed in Section 5.7, where I discuss the locality conditions on negative concord.

(63) $\begin{bmatrix} \textit{word} \\ \text{SYNSEM}|\text{LOC}|\text{CAT}|\text{HEAD}\ \ \textit{verb} \end{bmatrix} \rightarrow \begin{bmatrix}\text{SYNSEM}|\text{LOC}|\text{CAT}|\text{VAL}|\text{SPR}\ \langle\rangle\end{bmatrix}$

Thus, in addition to the information in the lexical entry of *citi*, we also have the specifications $[\text{MARKING}\ \ \textit{unmarked}]$ and $[\text{SPR}\ \langle\rangle]$ from the two principles above.

From the lexical entry of *citi* we can derive another inflectional form of the verb by means of a lexical rule. The lexical rule in (64) derives the past participle from a base verbal form:

(64) THE PAST PARTICIPLE LEXICAL RULE

$$\begin{bmatrix} \textit{word} \\ \text{PHON}\ \ \boxed{1} \\ \text{SS}|\text{LOC}\ |\text{CAT}\ |\text{HEAD} \begin{bmatrix}\textit{verb} \\ \text{VFORM}\ \ \textit{base}\end{bmatrix} \end{bmatrix} \longmapsto \begin{bmatrix} \text{PHON}\ \ PastPart(\boxed{1}) \\ \text{SS}\ |\ \text{LOC}\ |\text{CAT}\ |\text{HEAD} \begin{bmatrix}\textit{verb} \\ \text{VFORM}\ \ \textit{past-part}\end{bmatrix} \end{bmatrix}$$

The input description in the lexical rule (to the left) refers to a verb in its base form and the output description (to the right) to the same verb in the past participle form. The function $PastPart$ specifies how the phonological string of the input is modified in the output. The past participle of the word *citi* is *citit*, so in this case $PastPart$ would stipulate that if the input phon-string $\boxed{1}$ ends in `-i`, the output phon-string will be added `-t`.

In lexical rules, we only specify that piece of information about a word which undergoes a change via the lexical rule. All other information is transmitted unchanged to the output. If the verb *citi* undergoes the PAST PARTICIPLE LEXICAL RULE, the output is the word *citit* which has the complex specification in (65):

(65) $$\begin{bmatrix} \textit{word} \\ \text{PHON}\ \ \langle\texttt{citit}\rangle \\ \text{SYNSEM} \begin{bmatrix} \textit{synsem} \\ \text{LOC} \begin{bmatrix} \textit{local} \\ \text{CAT} \begin{bmatrix} \textit{category} \\ \text{HEAD} \begin{bmatrix}\textit{verb} \\ \text{VFORM}\ \ \textit{past-part}\end{bmatrix} \\ \text{VAL} \begin{bmatrix} \textit{valence} \\ \text{SUBJ}\ \ \langle\text{NP}\rangle \\ \text{COMPS}\ \ \langle\text{NP}\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

For the account of negative concord in this thesis we are not directly interested in the derivational history of verb forms or other expressions. Thus I will only describe the necessary inflectional form of a linguistic expression and refer to it as a lexical entry, even though in a carefully written grammar that inflectional form would be licensed as the output of a lexical rule and not by a lexical entry. I will call a particular description the output of a lexical rule only in those cases when I make use of a lexical rule written in this grammar.

With lexical entries and lexical rules we describe words. We now concentrate on phrases. Unlike words, phrases are objects with constituent structure which is carried by the *head-struc* value of the attribute DTRS. Headed structures are constituent combinations that are licensed as (headed) phrases in the grammar. The sort *head-struc* has the following subsorts:

(66)

$$\begin{bmatrix} \textit{head-struc} & \\ \text{HEAD-DTR} & \textit{sign} \end{bmatrix}$$

$$\begin{bmatrix} \textit{head-subj-str} & \\ \text{SUBJ-DTR} & \textit{sign} \end{bmatrix} \begin{bmatrix} \textit{head-spr-str} & \\ \text{SPR-DTR} & \textit{sign} \end{bmatrix} \begin{bmatrix} \textit{head-comp-str} & \\ \text{COMP-DTR} & \textit{sign} \end{bmatrix} \begin{bmatrix} \textit{head-adj-str} & \\ \text{ADJ-DTR} & \textit{sign} \end{bmatrix} \begin{bmatrix} \textit{head-mrk-str} & \\ \text{MRK-DTR} & \textit{sign} \end{bmatrix}$$

The sort hierarchy above presents five constituent structures that can be values for the attribute DTRS of a phrase. All headed structures have an attribute HEAD-DTR which specifies the head of the phrase. Besides this attribute, individual headed structures introduce their specific attribute that specifies the non-head daughter in the phrase. Thus *head-subj-str*s ("head-subject structure") have an attribute SUBJ-DTR, *head-spr-str*s ("head-specifier structure") a SPR-DTR attribute, *head-comp-str*s ("head-complement structure") a COMP-DTR attribute. These structures are all related to the valence requirements of a head. Besides them, we also license *head-adj-str*s ("head-adjunct structure") with an attribute ADJ-DTR and *head-mrk-str*s ("head-marker-structure") with an attribute MRK-DTR.

Importantly, the *sign* value of the attributes in a headed structure is to be contrasted with the value of the valence attributes SUBJ, SPR, COMP, and the head attributes MOD, SPEC, where we have lists of synsem objects. This is because phrases are made up of full signs, including phonology etc, while a sign subcategorizes for/ modifies/ specifies a syntax-semantics specification, independently of the phonology that it is associated with.

To license only the kinds of phrases that describe linguistic complex objects, our grammar must constrain the way signs are put together in phrases. At this point we turn to the grammar principles that make up the theory of the HPSG grammar.

2.3.2.2 Important grammar principles

ID Schemata The IMMEDIATE DOMINANCE (ID) SCHEMATA in (67) give us the kinds of phrases that our grammar allows. The ID PRINCIPLE in (50) excludes from the grammar any phrase that does not match one of the five schemata in (67).

(67) ID SCHEMATA

a. $\text{SCHEMA-1} \equiv \begin{bmatrix} \text{SS}|\text{LOC}|\text{CAT}|\text{VAL} & \begin{bmatrix} \textit{valence} & \\ \text{SUBJ} & \textit{elist} \\ \text{SPR} & \textit{elist} \\ \text{COMPS} & \textit{elist} \end{bmatrix} \\ \text{DTRS } \textit{head-subj-str} & \end{bmatrix}$

b. $\text{SCHEMA-2} \equiv \begin{bmatrix} \text{SS}|\text{LOC}|\text{CAT}|\text{VAL} & \begin{bmatrix} \textit{valence} & \\ \text{SUBJ} & \textit{elist} \\ \text{SPR} & \textit{elist} \\ \text{COMPS} & \textit{elist} \end{bmatrix} \\ \text{DTRS} & \begin{bmatrix} \textit{head-spr-str} & \\ \text{HEAD-DTR} & \begin{bmatrix} \textit{sign} \\ \text{SS } \boxed{0} \end{bmatrix} \\ \text{SPR-DTR} & \begin{bmatrix} \textit{sign} \\ \text{SS}|\text{LOC}|\text{CAT}|\text{HEAD}|\text{SPEC } \boxed{0} \end{bmatrix} \end{bmatrix} \end{bmatrix}$

c. $$\text{S}\textsc{chema}\text{-3} \equiv \begin{bmatrix} \textsc{ss}|\textsc{loc}|\textsc{cat}|\textsc{val} \begin{bmatrix} \textit{valence} \\ \textsc{comps} \;\; \textit{list} \end{bmatrix} \\ \textsc{dtrs} \; \textit{head-comp-str} \end{bmatrix}$$

d. $$\text{S}\textsc{chema}\text{-4} \equiv \begin{bmatrix} \textsc{dtrs} \begin{bmatrix} \textit{head-adj-str} \\ \textsc{head-dtr} \begin{bmatrix} \textit{sign} \\ \textsc{ss} \; \boxed{1} \end{bmatrix} \\ \textsc{adj-dtr} \begin{bmatrix} \textit{sign} \\ \textsc{ss}|\textsc{loc}|\textsc{cat}|\textsc{head}|\textsc{mod} \; \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

e. $$\text{S}\textsc{chema}\text{-5} \equiv \begin{bmatrix} \textsc{ss}|\textsc{loc}|\textsc{cat} \begin{bmatrix} \textsc{val} \begin{bmatrix} \textsc{subj} & \textit{list} \\ \textsc{spr} & \textit{elist} \\ \textsc{comps} & \textit{elist} \end{bmatrix} \\ \textsc{marking} \; \boxed{2} \end{bmatrix} \\ \textsc{dtrs} \begin{bmatrix} \textit{head-mrk-str} \\ \textsc{head-dtr} \begin{bmatrix} \textit{sign} \\ \textsc{ss} \; \boxed{1} \end{bmatrix} \\ \textsc{mrk-dtr} \begin{bmatrix} \textit{sign} \\ \textsc{ss}|\textsc{loc}|\textsc{cat} \begin{bmatrix} \textsc{head} \begin{bmatrix} \textit{marker} \\ \textsc{spec} \;\; \boxed{1} \end{bmatrix} \\ \textsc{marking} \; \boxed{2} \; \textit{marked} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

The ID Schema in (67a) enforces head-subject phrases to have all valence requirements satisfied: SUBJ, SPR and COMP lists must be empty. Given the signature, these phrases will also have a subject daughter. Head-specifier phrases (67b) are also required to have a satisfied subcategorization frame. Moreover, their specifier daughter must identify its SPEC value with the SYNSEM value of the head daughter. The use of the tag $\boxed{0}$ for both the value of the attribute SPEC in the specifier daughter and the value of SYNSEM (SS) in the head daughter indicates that the two values are the same.

Head-complement phrases (67c) have a possibly non-empty list as the COMP value. According to the signature, they also have a complement daughter. Given that head-complement phrases have only one COMP-DTR (see (66)), only binary branching structures are licensed in the grammar: in case a head requires more complements, they combine with the head one by one.[30]

SCHEMA-4 licenses head-adjunct phrases and is intended to account for modifiers of verbal projections.[31] It enforces the MOD value of the adjunct daughter to be identified with the synsem of the head daughter via the tag $\boxed{10}$. By not stating any particular requirements on the valence lists of the phrase, we allow adjuncts to modify any projection level: the lexical head, a phrase containing some or all the complements required by the head, or even full phrases with subjects.

SCHEMA-5 constrains head-marker phrases to inherit the MARKING specification $\boxed{2}$ from the marker daughter and their marker daughter to identify its SPEC value $\boxed{1}$ with the synsem of the head-daughter. In this grammar I only consider markers for verbal projections (see (60)), so the SPR list of the head-marker phrase will be empty (cf. (63)). SCHEMA-5 also constrains head-marker phrases to have an empty COMPS list. This means that a phrase cannot further combine with a complement if it has been marked. The SUBJ list can be empty or not. It will always be empty for the *că* complementizer which marks full clauses, but it may be empty or non-empty for *să*, which can

[30] Pollard and Sag (1994) make use of multiple branching structures, thus the subcategorization requirements for complements are saturated all at once (see SCHEMA 2 in Pollard and Sag (1994, p. 38)). In this thesis I assume a binary branching structure which is easier to extend to the semantic representations with quantifiers in Chapter 4.

[31] Our grammar in Section 5.4.3 will only contain one such modifier.

mark both VPs and full clauses.[32] Related to markers, I also assume a principle that enforces a phrase to inherit the MARKING specification of the head-daughter if it is not a head-marker phrase. This marking specification would usually be *unmarked*.[33]

The constraints on the MOD value of adjunct daughters and the SPEC value of specifier and marker daughters can be formulated independently of the ID Schemata, as two principles. Pollard and Sag (1994) for instance give a Spec Principle. To keep our theory simple, we enforce these conditions within the ID Schemata.

Valence Principle Another constraint necessary for a theory of constituent structure is the VALENCE PRINCIPLE. Its role is to relate the SUBJ-/ SPR-/ COMP-DTR to the subcategorization requirements of the head daughter. Together with the ID Schemata above, it licenses the phrases in the grammar.

(68) THE VALENCE PRINCIPLE

a. The value of the SUBJ attribute of the head daughter in a head-subject phrase is a list whose first element is the SYNSEM value of its subject daughter and whose rest is the phrase's SUBJ value. The SPR and COMPS values of the phrase are identical to those of the head daughter.

$$\begin{bmatrix} \textit{phrase} \\ \text{DTRS } \textit{head-subj-str}\end{bmatrix} \rightarrow \begin{bmatrix} \text{SS}\mid\text{LOC}\mid\text{CAT}\mid\text{VAL} & \begin{bmatrix}\text{SUBJ} & \boxed{1}\\ \text{SPR} & \boxed{3} \\ \text{COMPS} & \boxed{4}\end{bmatrix} \\ \text{DTRS} & \begin{bmatrix} \text{HEAD-DTR}\mid\text{SS}\mid\text{LOC}\mid\text{CAT}\mid\text{VAL} & \begin{bmatrix}\text{SUBJ} & \langle \boxed{2} \mid \boxed{1}\rangle \\ \text{SPR} & \boxed{3}\\ \text{COMPS} & \boxed{4}\end{bmatrix} \\ \text{SUBJ-DTR}\mid\text{SS} & \boxed{2}\end{bmatrix}\end{bmatrix}$$

b. The value of the SPR attribute of the head daughter in a head-specifier phrase is a list whose first element is the SYNSEM value of its specifier daughter and whose rest is the phrase's SPR value. The SUBJ and COMPS values of the phrase are identical to those of the head daughter.

$$\begin{bmatrix} \textit{phrase} \\ \text{DTRS } \textit{head-spr-str}\end{bmatrix} \rightarrow \begin{bmatrix} \text{SS}\mid\text{LOC}\mid\text{CAT}\mid\text{VAL} & \begin{bmatrix}\text{SUBJ} & \boxed{3}\\ \text{SPR} & \boxed{1} \\ \text{COMPS} & \boxed{4}\end{bmatrix} \\ \text{DTRS} & \begin{bmatrix} \text{HEAD-DTR}\mid\text{SS}\mid\text{LOC}\mid\text{CAT}\mid\text{VAL} & \begin{bmatrix}\text{SUBJ} & \boxed{3} \\ \text{SPR} & \langle \boxed{2} \mid \boxed{1}\rangle\\ \text{COMPS} & \boxed{4}\end{bmatrix} \\ \text{SPR-DTR}\mid\text{SS} & \boxed{2}\end{bmatrix}\end{bmatrix}$$

c. The value of the COMPS attribute of the head daughter in a head-complement phrase is a list whose first element is the SYNSEM value of its complement daughter and whose rest is the phrase's COMPS value. The SPR and SUBJ values of the phrase

[32] In Section 5.7 we will have an example with a matrix control verb, so *să* will mark an embedded VP with a non-empty SUBJ value.

[33] See (Pollard and Sag, 1994, fn. 51, p. 45) for a formulation of this principle.

are identical to those of the head daughter.

$$\begin{bmatrix} \textit{phrase} \\ \text{DTRS} & \textit{head-comp-str} \end{bmatrix} \rightarrow \begin{bmatrix} \text{SS|LOC|CAT|VAL} & \begin{bmatrix} \text{SUBJ} & \boxed{3} \\ \text{SPR} & \boxed{4} \\ \text{COMPS} & \boxed{1} \end{bmatrix} \\ \text{DTRS} & \begin{bmatrix} \text{HEAD-DTR|SS|LOC|CAT|VAL} & \begin{bmatrix} \text{SUBJ} & \boxed{3} \\ \text{SPR} & \boxed{4} \\ \text{COMPS} & \langle \boxed{2} | \boxed{1} \rangle \end{bmatrix} \\ \text{COMP-DTR|SS} & \boxed{2} \end{bmatrix} \end{bmatrix}$$

d. The SPR, SUBJ and COMPS values of a head-adjunct phrase are identical to those of the head daughter.

$$\begin{bmatrix} \textit{phrase} \\ \text{DTRS} & \textit{head-adj-str} \end{bmatrix} \rightarrow \begin{bmatrix} \text{SS|LOC|CAT|VAL} & \begin{bmatrix} \text{SUBJ} & \boxed{1} \\ \text{SPR} & \boxed{2} \\ \text{COMPS} & \boxed{3} \end{bmatrix} \\ \text{DTRS} & \begin{bmatrix} \text{HEAD-DTR|SS|LOC|CAT|VAL} & \begin{bmatrix} \text{SUBJ} & \boxed{1} \\ \text{SPR} & \boxed{2} \\ \text{COMPS} & \boxed{3} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

e. The SPR, SUBJ and COMPS values of a head-marker phrase are identical to those of the head daughter.

$$\begin{bmatrix} \textit{phrase} \\ \text{DTRS} & \textit{head-mrk-str} \end{bmatrix} \rightarrow \begin{bmatrix} \text{SS|LOC|CAT|VAL} & \begin{bmatrix} \text{SUBJ} & \boxed{1} \\ \text{SPR} & \boxed{2} \\ \text{COMPS} & \boxed{3} \end{bmatrix} \\ \text{DTRS} & \begin{bmatrix} \text{HEAD-DTR|SS|LOC|CAT|VAL} & \begin{bmatrix} \text{SUBJ} & \boxed{1} \\ \text{SPR} & \boxed{2} \\ \text{COMPS} & \boxed{3} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

The VALENCE PRINCIPLE must be understood as a conjunction of the five constraints in (68a) to (68e). Phrases usually inherit the valence values of the head daughter, unless the non-head daughter saturates (part of) one such value of the head daughter. Head-adjunct phrases and head-marker phrases inherit all the valence specifications of the head daughter, since their non-head daughters are not subcategorized for by the head daughter.

For the phrases in which the non-head daughter reduces some valence list of the head daughter (in (68a), (68b), and (68c)), the valence principle constrains the relationship between the valence values of the head daughter and the SYNSEM value of the non-head daughter. Note that the valence requirements are saturated in the order in which they appear on the valence lists of the head daughter. For instance, in (68c) the *phrase* inherits the REST value ($\boxed{1}$) of the COMPS list of the head daughter. The first element $\boxed{2}$ on the COMPS list of the head daughter is identified with the SYNSEM value of the COMP-DTR. The notation $\langle \boxed{2} | \boxed{1} \rangle$ stands for a list with the FIRST value $\boxed{2}$ and the REST value $\boxed{1}$.

We also need to make sure that the SUBJ and SPR attributes are lists of length at most one.[34] This can be done by the two principles below which enforce the SUBJ/ SPR value in valence objects to be either the empty list or a list made up of only one synsem element:

[34]Note that in the case of specifiers, this constraint may be too restrictive, since it would rule out structures like *all the students* in English if both *all* and *the* are considered specifiers.

(69) THE UNIQUE SUBJECT/ SPECIFIER PRINCIPLES

$$\textit{valence} \rightarrow \begin{bmatrix}\textsc{subj} & \textit{elist} \vee \langle \textit{synsem} \rangle\end{bmatrix}$$

$$\textit{valence} \rightarrow \begin{bmatrix}\textsc{spr} & \textit{elist} \vee \langle \textit{synsem} \rangle\end{bmatrix}$$

Argument structure In Pollard and Sag (1994, Ch. 1–8) all the valence requirements are placed on a SUBCAT (“subcategorization”) list. Given the subsequent convention of keeping valence properties separated from one another (Pollard and Sag (1994, Ch. 9), Sag (1997) and others, all following Borsley (1987)), the attribute ARG-ST (“argument structure”) is introduced on words to collect all the valence specifications on a single list of *synsem* objects. In Sag et al. (2003), the realization of ARG-ST is formulated as a principle which I import into our grammar:

(70) THE ARGUMENT REALIZATION PRINCIPLE

$$\textit{word} \rightarrow \begin{bmatrix}\textsc{synsem}|\textsc{loc}|\textsc{cat}|\textsc{val} & \begin{bmatrix}\textsc{subj} & \boxed{1} \\ \textsc{spr} & \boxed{2} \\ \textsc{comps} & \boxed{3}\end{bmatrix} \\ \textsc{arg-st} & \boxed{1} \oplus \boxed{2} \oplus \boxed{3}\end{bmatrix}$$

The Head Feature Principle One more constraint to be mentioned here is the HEAD FEATURE PRINCIPLE (HFP). It is given below:

(71) THE HEAD FEATURE PRINCIPLE (HFP)

$$\textit{phrase} \rightarrow \begin{bmatrix}\textsc{synsem}|\textsc{local}|\textsc{cat}|\textsc{head} & \boxed{1} \\ \textsc{dtrs}|\textsc{head-dtr}|\textsc{synsem}|\textsc{local}|\textsc{cat}|\textsc{head} & \boxed{1}\end{bmatrix}$$

The HFP ensures that phrases inherit the morphosyntactic (HEAD) specification of their head daughters. It thus rules out a phrase like (72) which would be allowed by the signature in combination with the other principles that we mentioned:

(72) An example of a phrase type ruled our by the HFP

$$\begin{bmatrix}\textit{phrase} \\ \textsc{synsem}|\textsc{loc}|\textsc{cat}|\textsc{head}\ \textit{verb} \\ \textsc{dtrs}|\textsc{head-dtr}|\textsc{synsem}|\textsc{loc}|\textsc{cat}|\textsc{head}\ \textit{adverb}\end{bmatrix}$$

2.3.2.3 Abbreviations

As mentioned before, HPSG grammar writing often employs abbreviations of AVMs for readability. In this thesis I make use of the following abbreviations:

(73) a. $$\text{NP} \equiv \begin{bmatrix}\textit{synsem} & \\ \textsc{loc}|\textsc{cat} & \begin{bmatrix}\textsc{head} & \textit{noun} \\ \textsc{val} & \begin{bmatrix}\textit{valence} & \\ \textsc{subj} & \langle\rangle \\ \textsc{spr} & \langle\rangle \\ \textsc{comps} & \langle\rangle\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

b. $$N \equiv \begin{bmatrix} \textit{synsem} & \\ \text{LOC}|\text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \textit{valence} & \\ \text{SUBJ} & \langle\rangle \\ \text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

c. $$S \equiv \begin{bmatrix} \textit{synsem} & \\ \text{LOC}|\text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \textit{valence} & \\ \text{SUBJ} & \langle\rangle \\ \text{SPR} & \langle\rangle \\ \text{COMPS} & \langle\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

d. $$VP \equiv \begin{bmatrix} \textit{synsem} & \\ \text{LOC}|\text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \textit{valence} & \\ \text{SUBJ} & \langle \boxed{1} \rangle \\ \text{SPR} & \langle\rangle \\ \text{COMPS} & \langle\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

e. $$V \equiv \begin{bmatrix} \textit{word} & \\ \text{SYNSEM} & \\ \text{LOC}|\text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \textit{valence} & \\ \text{SUBJ} & \langle \boxed{1} \rangle \\ \text{SPR} & \langle\rangle \\ \text{COMPS} & \textit{list} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

f. $$M \equiv \begin{bmatrix} \textit{synsem} & \\ \text{LOC}|\text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{marker} \\ \text{VAL} & \begin{bmatrix} \textit{valence} & \\ \text{SUBJ} & \langle\rangle \\ \text{SPR} & \langle\rangle \\ \text{COMPS} & \langle\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

g. $$Adv \equiv \begin{bmatrix} \textit{synsem} & \\ \text{LOC}|\text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{adverb} \\ \text{VAL} & \begin{bmatrix} \textit{valence} & \\ \text{SUBJ} & \langle\rangle \\ \text{SPR} & \langle\rangle \\ \text{COMPS} & \langle\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

h. $$Det \equiv \begin{bmatrix} \textit{synsem} & \\ \text{LOC}|\text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{determiner} \\ \text{VAL} & \begin{bmatrix} \textit{valence} & \\ \text{SUBJ} & \langle\rangle \\ \text{SPR} & \langle\rangle \\ \text{COMPS} & \langle\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

2.3.2.4 The syntactic structure of a Romanian sentence

We now have the main ingredients of an HPSG grammar to analyze a sentence of Romanian. Let us derive (74) to see how the signature and the constraints in the theory interact in licensing grammatical sentences.

(74) Ion citeşte o carte.
John reads a book
'John is reading a book.'

We start by specifying the contribution of the lexical items in the sentence: *Ion*, *citeşte*, *o*, and *carte*. Given the lexical entry for *citi* in (61), the specification in (75b) below describes objects that are licensed as the output of a lexical rule giving us the present tense (third person singular) form of the verb.

(75) a. *Ion* ('John')

$$\left[\begin{array}{ll} \multicolumn{2}{l}{\textit{word}} \\ \text{PHON} & \langle \texttt{ion} \rangle \\ \text{SYNSEM } \boxed{2} & \left[\text{LOC}|\text{CAT} \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{category}} \\ \text{HEAD} & \boxed{1}\ \textit{noun} \\ \text{VAL} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{valence}} \\ \text{SPR} & \langle\rangle \\ \text{COMPS} & \langle\rangle \end{array}\right] \end{array}\right]\right] \\ \text{ARG-ST } \langle\rangle & \end{array}\right]$$

b. *citeşte* ('reads')

$$\left[\begin{array}{ll} \multicolumn{2}{l}{\textit{word}} \\ \text{PHON} & \langle \texttt{citeşte} \rangle \\ \text{SYNSEM}|\text{LOC}|\text{CAT} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{category}} \\ \text{HEAD} & \boxed{0} \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{verb}} \\ \text{VFORM} & \textit{present} \end{array}\right] \\ \text{VAL} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{valence}} \\ \text{SUBJ} & \langle \boxed{2}\ \text{NP} \rangle \\ \text{COMPS} & \langle \boxed{3}\ \text{NP} \rangle \end{array}\right] \end{array}\right] \\ \text{ARG-ST } \langle \boxed{2}, \boxed{3} \rangle & \end{array}\right]$$

c. *o* ('a')

$$\left[\begin{array}{ll} \multicolumn{2}{l}{\textit{word}} \\ \text{PHON} & \langle \texttt{o} \rangle \\ \text{SYNSEM } \boxed{6} & \left[\text{LOC}|\text{CAT} \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{category}} \\ \text{HEAD} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{det}} \\ \text{SPEC} & \boxed{4}\ \text{N} \end{array}\right] \\ \text{VAL} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{valence}} \\ \text{SPR} & \langle\rangle \\ \text{COMPS} & \langle\rangle \end{array}\right] \end{array}\right]\right] \\ \text{ARG-ST } \langle\rangle & \end{array}\right]$$

d. *carte* ('book')

$$\left[\begin{array}{ll} \textit{word} & \\ \text{PHON} & \langle \texttt{carte} \rangle \\ \text{SYNSEM} & \boxed{4} \left[\begin{array}{ll} \text{LOC}|\text{CAT} & \left[\begin{array}{ll} \textit{category} & \\ \text{HEAD} & \boxed{5}\ \textit{noun} \\ \text{VAL} & \left[\begin{array}{ll} \textit{valence} & \\ \text{SPR} & \langle \boxed{6}\ \text{Det} \rangle \\ \text{COMPS} & \langle\rangle \end{array}\right] \end{array}\right] \end{array}\right] \\ \text{ARG-ST} & \langle \boxed{6} \rangle \end{array}\right]$$

To keep the lexical entries simple, we introduce another principle (similar to (62) and (63) above) that constrains all the words that are not verbs to have an empty SUBJ list:

(76) $$\left[\begin{array}{ll} \textit{word} & \\ \text{SYNSEM}|\text{LOC}|\text{CAT}|\text{HEAD} & \neg\ \textit{verb} \end{array}\right] \rightarrow \left[\text{SYNSEM}|\text{LOC}|\text{CAT}|\text{VAL}|\text{SUBJ}\ \langle\rangle\right]$$

Thus all the lexical items in (75), except for *citeşte*, receive a SUBJ empty list value.

Although the tags with only one occurrence in (75) (e.g. $\boxed{0}$, $\boxed{1}$, $\boxed{4}$, $\boxed{5}$) do not play any particular role they indicate token-identity in the complex structure in FIGURE 2.2, where these lexical descriptions and their pieces are present at different places within the description of the whole sentence.

Now we can license phrases by means of the lexical descriptions, the ID SCHEMATA in (67) and the VALENCE PRINCIPLE in (68). SCHEMA-2 and the principle in (68b) license the phrase *o carte* ("a book") in (77). SCHEMA-3 and principle (68c) license the phrase *citeşte o carte* ("is reading a book"), and by means of SCHEMA-1 and principle (68a), we can license the whole sentence *Ion citeşte o carte* ("John is reading a book"). For readability, I leave unmarked the token-identity between the valence lists of a phrase and those of its head daughter. For instance in (77), we should label the COMPS and the SUBJ values of the phrase with the same tags as the COMPS and SUBJ values of the word *carte* in (75d). See a full specification of these token-identities in the tree structure in FIGURE 2.3.

Regarding the PHON value of phrases, I tacitly assume a principle in the grammar that restricts this value in a sensible way so that it contains all and only the PHON values of the daughters in the intended linear order.[35]

(77) *o carte* ('a book')

$$\left[\begin{array}{ll} \textit{phrase} & \\ \text{PHON} & \langle \texttt{o carte} \rangle \\ \text{SS} & \boxed{3} \left[\begin{array}{ll} \text{LOC}|\text{CAT} & \left[\begin{array}{ll} \textit{category} & \\ \text{HEAD} & \boxed{5}\ \textit{noun} \\ \text{VAL} & \left[\begin{array}{ll} \textit{valence} & \\ \text{SUBJ} & \langle\rangle \\ \text{SPR} & \langle\rangle \\ \text{COMPS} & \langle\rangle \end{array}\right] \end{array}\right] \end{array}\right] \\ \text{DTRS} & \left[\begin{array}{ll} \textit{head-spr-str} & \\ \text{HEAD-DTR} & \boxed{75d} \\ \text{SPR-DTR} & \boxed{75c} \end{array}\right] \end{array}\right]$$

[35]Given that the value of PHON is a list, we should have a comma between the phonological string elements and write $\left[\text{PHON}\ \langle \texttt{o, carte} \rangle\right]$ in (77). For simplicity I leave out the commas and only use a blank to delimit the individual phonological strings.

(78) *citeşte o carte* (‘reads a book’)

$$\begin{bmatrix} \textit{phrase} & \\ \text{PHON} & \langle \texttt{citeşte o carte} \rangle \\ \text{SS}|\text{LOC}|\text{CAT} & \begin{bmatrix} \textit{category} & \\ \text{HEAD} & \boxed{0} \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{present} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \textit{valence} & \\ \text{SUBJ} & \langle \boxed{2}\ \text{NP} \rangle \\ \text{SPR} & \langle \rangle \\ \text{COMPS} & \langle \rangle \end{bmatrix} \end{bmatrix} \\ \text{DTRS} & \begin{bmatrix} \textit{head-comp-str} & \\ \text{HEAD-DTR} & \boxed{75b} \\ \text{COMP-DTR} & \boxed{77} \end{bmatrix} \end{bmatrix}$$

(79) *Ion citeşte o carte* (‘John reads a book’)

$$\begin{bmatrix} \textit{phrase} & \\ \text{PHON} & \langle \texttt{ion citeşte o carte} \rangle \\ \text{SS}|\text{LOC}|\text{CAT} & \begin{bmatrix} \textit{category} & \\ \text{HEAD} & \boxed{0} \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{present} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \textit{valence} & \\ \text{SUBJ} & \langle \rangle \\ \text{SPR} & \langle \rangle \\ \text{COMPS} & \langle \rangle \end{bmatrix} \end{bmatrix} \\ \text{DTRS} & \begin{bmatrix} \textit{head-subj-str} & \\ \text{HEAD-DTR} & \boxed{78} \\ \text{SUBJ-DTR} & \boxed{75a} \end{bmatrix} \end{bmatrix}$$

A detailed AVM description of the phrase in (79) is given in FIGURE 2.2, p. 63. A tree structure notation is given in FIGURE 2.3, p. 64, where the branches under each phrase correspond to the attributes of the head-structure object that is the value of DTRS in the phrase. For the upmost phrase in the tree the left branch stands for the SUBJ-DTR attribute and the right branch for the HEAD-DTR attribute.

The reader may verify in FIGURE 2.3 the correct application of the constraints given in Section 2.3.2.2. The HFP is respected since the phrase *o carte* inherits the HEAD value $\boxed{5}$ of its head daughter *carte*, and the phrases *citeşte o carte* and *Ion citeste o carte* inherit the HEAD value $\boxed{0}$ of their head daughters. The ID PRINCIPLE is respected since we only have phrases with a DTRS value of sorts *head-subj-str*, *head-spr-str*, and *head-comp-str* licensed by the schemata in (67). The correct application of the VALENCE PRINCIPLE can be verified by observing that non-head daughters saturate the corresponding valence requirements of the head daughters, while the other valence values are inherited by a phrase from its head daughter. The word *carte* gets its SPR value saturated by the specifier $\boxed{8}$, so the phrase *o carte* has the empty list value for SPR. The same procedure applies for the word *citeşte* and the phrase *citeşte o carte* whose COMPS and SUBJ values are saturated by $\boxed{10}$ and $\boxed{12}$, respectively, so the phrases *citeşte o carte* and *Ion citeşte o carte* have empty COMPS and SUBJ values. Apart from the saturated valence values, *o carte* inherits the SUBJ and COMPS values $\boxed{13}$/ $\boxed{14}$ from the head daughter *carte*, *citeşte o carte* inherits the SUBJ and SPR values $\boxed{15}$/ $\boxed{16}$ from the head daughter *citeşte*, and *Ion citeşte o carte* inherits the SPR and COMPS values $\boxed{16}$/ $\boxed{17}$ from the head daughter *citeşte o carte*.

$$
\left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{phrase}}\\
\textsc{phon} & \left\langle \texttt{ion citește o carte} \right\rangle\\
\textsc{ss}|\textsc{loc}|\textsc{cat} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{category}}\\
\textsc{head} & \boxed{0}\\
\textsc{val} & \left[\begin{array}{l}\textit{valence}\\ \textsc{subj}\ \langle\rangle\\ \textsc{spr}\ \langle\rangle\\ \textsc{comps}\ \langle\rangle\end{array}\right]
\end{array}\right]\\
\textsc{dtrs} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{head-subj-str}}\\
\textsc{h-dtr} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{phrase}}\\
\textsc{phon} & \left\langle \texttt{citește o carte} \right\rangle\\
\textsc{ss}|\textsc{loc}|\textsc{cat} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{category}}\\
\textsc{head} & \boxed{0}\\
\textsc{val} & \left[\begin{array}{l}\textit{valence}\\ \textsc{subj}\ \left\langle \boxed{2}\,\mathrm{NP} \right\rangle\\ \textsc{spr}\ \langle\rangle\\ \textsc{comps}\ \langle\rangle\end{array}\right]
\end{array}\right]\\
\textsc{dtrs} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{head-comp-str}}\\
\textsc{h-dtr} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{word}}\\
\textsc{phon} & \left\langle \texttt{citește} \right\rangle\\
\textsc{ss}|\textsc{loc}|\textsc{cat} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{category}}\\
\textsc{head} & \boxed{0}\,\textit{verb}\\
\textsc{val} & \left[\begin{array}{l}\textit{valence}\\ \textsc{subj}\ \left\langle \boxed{2}\,\mathrm{NP} \right\rangle\\ \textsc{spr}\ \langle\rangle\\ \textsc{comps}\ \left\langle \boxed{3}\,\mathrm{NP} \right\rangle\end{array}\right]
\end{array}\right]\\
\textsc{arg-st} & \left\langle \boxed{2}, \boxed{3} \right\rangle
\end{array}\right]\\
\textsc{c-dtr} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{phrase}}\\
\textsc{phon} & \left\langle \texttt{o carte} \right\rangle\\
\textsc{ss} & \boxed{3}\left[\begin{array}{ll}
\textsc{loc}|\textsc{cat} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{category}}\\
\textsc{head} & \boxed{5}\\
\textsc{val} & \left[\begin{array}{l}\textit{valence}\\ \textsc{subj}\ \langle\rangle\\ \textsc{spr}\ \langle\rangle\\ \textsc{comps}\ \langle\rangle\end{array}\right]
\end{array}\right]
\end{array}\right]\\
\textsc{dtrs} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{head-spr-str}}\\
\textsc{h-dtr} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{word}}\\
\textsc{phon} & \left\langle \texttt{carte} \right\rangle\\
\textsc{ss} & \boxed{4}\left[\begin{array}{ll}
\textsc{loc}|\textsc{cat} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{category}}\\
\textsc{head} & \boxed{5}\,\textit{noun}\\
\textsc{val} & \left[\begin{array}{l}\textit{valence}\\ \textsc{subj}\ \langle\rangle\\ \textsc{spr}\ \left\langle \boxed{6} \right\rangle\\ \textsc{comps}\ \langle\rangle\end{array}\right]
\end{array}\right]
\end{array}\right]\\
\textsc{arg-st} & \left\langle \boxed{6} \right\rangle
\end{array}\right]\\
\textsc{spr-dtr} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{word}}\\
\textsc{phon} & \left\langle \texttt{o} \right\rangle\\
\textsc{ss} & \boxed{6}\left[\begin{array}{ll}
\textsc{loc}|\textsc{cat} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{category}}\\
\textsc{head} & \left[\begin{array}{l}\textit{det}\\ \textsc{spec}\ \boxed{4}\,\mathrm{N}\end{array}\right]\\
\textsc{val} & \left[\begin{array}{l}\textit{valence}\\ \textsc{subj}\ \langle\rangle\\ \textsc{spr}\ \langle\rangle\\ \textsc{comps}\ \langle\rangle\end{array}\right]
\end{array}\right]
\end{array}\right]\\
\textsc{arg-st} & \langle\rangle
\end{array}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right]\\
\textsc{sj-dtr} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{word}}\\
\textsc{phon} & \left\langle \texttt{ion} \right\rangle\\
\textsc{ss} & \boxed{2}\left[\begin{array}{ll}
\textsc{loc}|\textsc{cat} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{category}}\\
\textsc{head} & \boxed{1}\,\textit{noun}\\
\textsc{val} & \left[\begin{array}{l}\textit{valence}\\ \textsc{subj}\ \langle\rangle\\ \textsc{spr}\ \langle\rangle\\ \textsc{comps}\ \langle\rangle\end{array}\right]
\end{array}\right]
\end{array}\right]\\
\textsc{arg-st} & \langle\rangle
\end{array}\right]
\end{array}\right]
\end{array}\right]
$$

Figure 2.2: *Ion citeşte o carte*

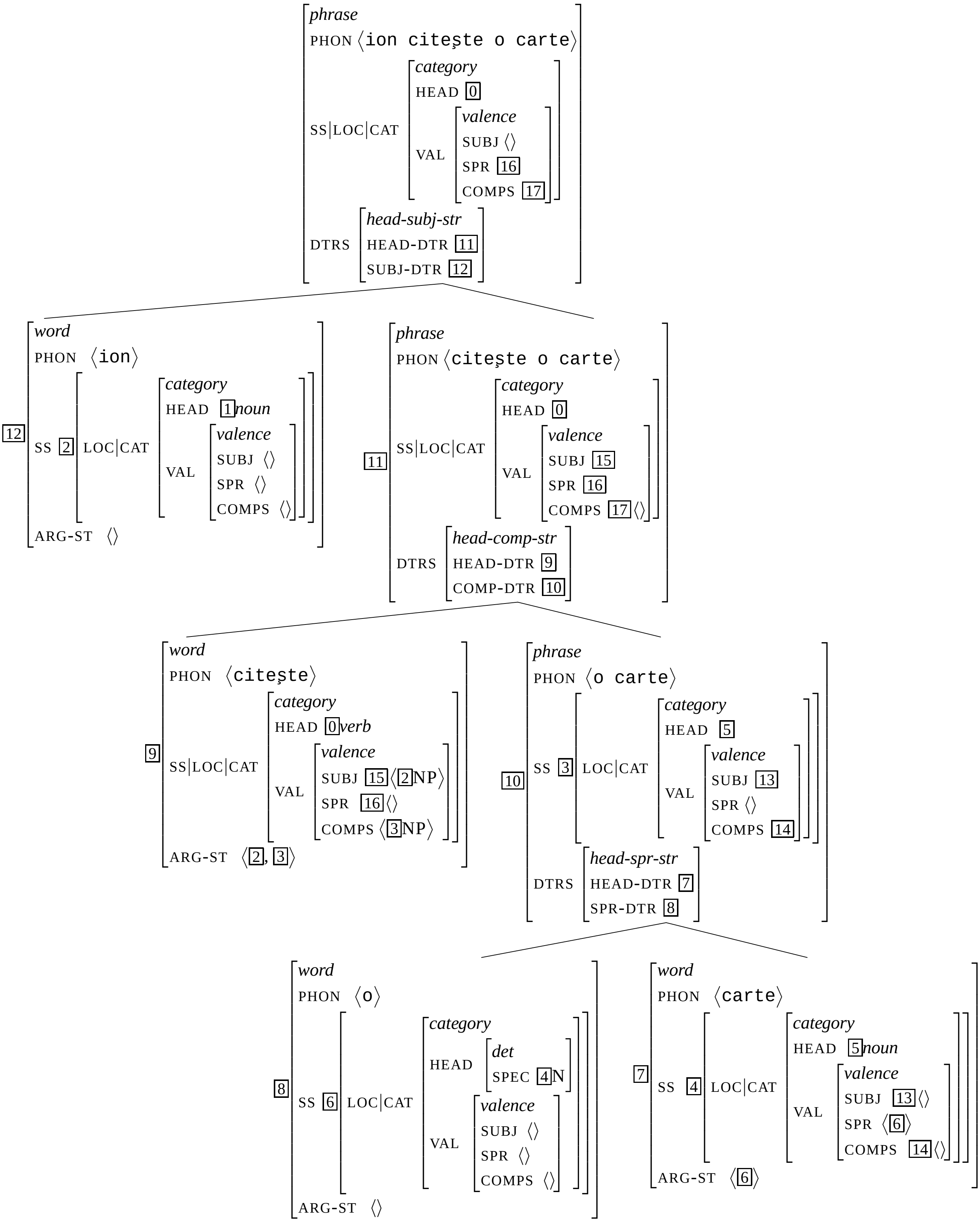


Figure 2.3: Tree representation for *Ion citeşte o carte*

2.3.2.5 Further issues: Semantics

So far, we have discussed the organization of an HPSG grammar from a syntactic point of view. That is, we paid attention to objects of sort *category*, the value of the attribute CAT. In this section I briefly address the semantics in HPSG, i.e. objects of sort *content*, as the value of the attribute CONT.

Semantics in the HPSG tradition of Sag and Pollard (1987) and Pollard and Sag (1994) is based on the *Situation Semantics* framework of Barwise and Perry (1983), but it also makes use of other semantic mechanisms imported for instance from Cooper (1983) (‘Cooper storage’). The values of the attributes for *content* objects characterize a semantic ontology specific to HPSG and not shared by other frameworks. In Chapter 5, I will replace this kind of ontology with another one, based on model-theoretic semantic representations generally assumed in semantic theories. For this reason, at this point I only give an informal description of the values for the CONT attribute. This short presentation is necessary to understand a proposal in de Swart and Sag (2002) which will be addressed in Section 4.3.1.

Let us concentrate on the semantic representation of a verb:

(80)
$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \textit{list(phon-string)} \\ \text{SYNSEM} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \textit{local} & \\ \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \textit{valence} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \textit{content} & \\ \text{QUANTS} & \text{“list of scopal elements”} \\ \text{NUCL} & \text{“main predication”} \\ \text{STORE} & \text{“set of quantifiers”} \end{bmatrix} \end{bmatrix} \\ \text{NLOC} & \textit{nonlocal} \end{bmatrix} \\ \text{ARG-ST} & \textit{list(synsem)} \end{bmatrix}$$

The structure in (80) is the result of several empirically motivated changes proposed in Pollard and Yoo (1998) and Przepiórkowski (1998), which modified the original semantic representation of signs as viewed in Pollard and Sag (1994).

The CONT value of a verb contains the attributes NUCL (“nucleus”), QUANTS (“quantifiers”) and STORE. NUCL hosts the semantic relation expressed by the verb. The value of QUANTS is a list of quantifiers (scopal operators) which take scope in the order dictated by the list: the leftmost quantifier has widest scope. They all take scope over the nucleus of the verb. The interpretation of quantifiers on the QUANTS list is mediated by a Cooper storage mechanism encoded under the attribute STORE. The value of STORE is a set of quantificational operators. It is non-empty for quantificational determiners, NPs that contain a quantifier, and verbs that have quantificational arguments. The STORE value is inherited by NPs from their determiners, so a quantifier NP like *fiecare student* ‘every student’ will have the specification $\left[\text{SS}|\text{LOC}|\text{CONT}|\text{STORE}\ \left\{\text{EVERY}^{\text{STUDENT}}\right\}\right]$. Verbs inherit the STORE specification of their arguments. The verb *citeşte* in (81) has two quantificational arguments *fiecare student* ‘every student’ and *o carte* ‘some book’ with a non-empty STORE set, so its STORE $\boxed{3}$ is a set of two quantifiers.[36]

(81) *Fiecare student* citeşte *o carte*.
every student reads a book

[36] Note that in (82) I employ the notation in de Swart and Sag (2002) for the NUCL value of the verb: the relation READ is different from the notation in Pollard and Sag (1994).

i. ‘Every student is such that s/he reads a book.’
ii. ‘A book is such that every student reads it.’

(82)
$$
\begin{bmatrix}
\textit{word} & \\
\text{PHON} & \left\langle \texttt{citește} \right\rangle \\
\text{SS}|\text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \textit{category} & \\ \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \textit{valence} & \\ \text{SUBJ} & \left\langle \boxed{1}\ \text{NP} \right\rangle \\ \text{SPR} & \langle\rangle \\ \text{COMPS} & \left\langle \boxed{2}\ \text{NP} \right\rangle \end{bmatrix} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \textit{content} & \\ \text{QUANTS} & \boxed{4} \\ \text{NUCL} & \text{READ} \\ \text{STORE} & \boxed{3}\left\{\text{EVERY}^{\text{STUDENT}}, \text{SOME}^{\text{BOOK}}\right\} \end{bmatrix} \end{bmatrix} \\
\text{ARG-ST} & \left\langle \boxed{1}\left[\text{SS}|\text{L}|\text{C}|\text{STORE}\ \left\{\text{EVERY}^{\text{STUDENT}}\right\}\right], \boxed{2}\left[\text{SS}|\text{L}|\text{C}|\text{STORE}\ \left\{\text{SOME}^{\text{BOOK}}\right\}\right] \right\rangle
\end{bmatrix}
$$

$\wedge\ \texttt{retrieve}(\boxed{3}, \boxed{4})$

We represent sets with curly brackets to distinguish them from the angle brackets for lists. Elements in a set are not ordered, so the set $\boxed{3}$ does not say anything about the scope interaction between the two quantifiers. This is fixed on the QUANTS list of the verb by means of a relation `retrieve`. The relation `retrieve` takes a set $\boxed{3}$ and returns a (ordered) list $\boxed{4}$ of the elements of that set. Thus in (82) the variable $\boxed{4}$ may take two different list values: $\langle \text{EVERY}^{\text{STUDENT}}, \text{SOME}^{\text{BOOK}} \rangle$ and $\langle \text{SOME}^{\text{BOOK}}, \text{EVERY}^{\text{STUDENT}} \rangle$. The former gives us the first interpretation in (81) and the latter the second one.

Chapter 3

The semantic status of Romanian n-words

This chapter addresses the main empirical facts concerning Romanian NC. The aim is to determine the semantic status of n-words and their role in NC constructions.[1] I argue that Romanian n-words are negative quantifiers, and that their behavior within NC resembles that of inherently polyadic quantifiers discussed in Section 2.1.3.

The chapter begins with a general presentation of NC languages (Section 3.1), and of the basic NC data in Romanian (Section 3.2). The NPI and the NQ approaches to NC mentioned in Section 1.2 are considered here in relation to Romanian. Section 3.3 provides several arguments against an NPI approach, and for an NQ analysis. In Section 3.4 more empirical support is brought for the negative semantics of Romanian n-words. In the last part, Section 3.5, the scope interaction between NQs and other operators is investigated. The similarity between NC and cumulative polyadic quantifiers leads to a proposal to treat NC as an inherently polyadic quantifier.

3.1 N-words and NC languages

The term *n-word*, originary from Laka (1990), has become very popular in the literature on negation and is used for nominal and adverbial negative constituents (like the Spanish *nadie*, 'nobody' *nada*, 'nothing', *ningun*, 'no' , *nunca*, 'never'), as opposed to the negative sentential operator, usually an adverb or an adverbial particle attached to the verb and referred to as the Negative Marker (NM).

3.1.1 DN vs. NC Languages

A central distinction that crosslinguistic studies on negation make is that between Double Negation (DN) and Negative Concord (NC) languages.[2] In DN languages the cooccurrence of a negative constituent with the NM or another negative constituent results in a DN effect, i.e. the sentence is understood as affirmative. In NC languages such a cooccurrence receives a NC reading, i.e. the sentence

[1]The discussion here is not intended to be exhaustive in characterizing negation or even negative concord in Romanian. For related issues not addressed here the reader is referred to the overview on Romanian negation in Barbu Mititelu and Maftei Ciolăneanu (2004).

[2]The term *negative concord* comes from Labov (1972) and is equivalent to Jespersen's (1917) *double attraction* and Klima's (1964) *neg-incorporation*.

is understood as negative. The class of DN languages includes most Germanic languages, while Romance and Slavic are standardly taken to belong to the class of NC languages. Let us take a few examples from both classes:

(83) DN languages

a. Standard English

i. John did**n't** say that.

ii. John did**n't** say **nothing**. (= 'John said something.') (DN)

b. German

i. Hans hat das **nicht** gekauft.
Hans has NM bought
'Hans didn't buy that.'

ii. Hans hat **nicht nichts** gekauft.
Hans has NM nothing bought
'Hans didn't buy nothing.' (= 'Hans bought something.') (DN)

c. Dutch

i. Jan loopt **niet**.
Jan walks NM
'Jan doesn't walk.' (Zeijlstra (2002, p. 186))

ii. Frank heeft **niet niemand** gezien.
Frank has NM nobody seen
'It is not the case that Frank didn't see anybody.' (DN)
(Giannakidou (2006, p. 329))

(84) NC languages

a. Italian

i. Gianni **non** è venuto.
Gianni NM is come
'Gianni didn't come.'

ii. Gianni *(**non**) ha visto **nessuno**.
Gianni NM has seen nobody
'Gianni didn't see anybody.' (NC)

b. Spanish

i. Pedro **no** ha visto a Juan.
Pedro NM has seen Juan
'Pedro didn't see Juan.'

ii. Pedro *(**no**) ha visto a **nadie**.
Pedro NM has seen nobody
'Pedro didn't see anybody.' (NC)

c. Polish

i. Jan **nie** lubi Marysi.
John NM likes Mary
'John doesn't like Mary.'

ii. Marysia *(**nie**) dała **niczego** Piotrowi.
Mary NM gave nothing Piotr
'Mary didn't give anything to Peter.' (NC)
(Przepiórkowski and Kupść (1999, pp. 212–213))

The interpretation of sentences (83a-ii), (83b-ii), (83c-ii) qualifies standard English, German, and Dutch as DN languages, while (84a-ii), (84b-ii), and (84c-ii) indicate that Italian, Spanish, and Polish are NC languages. As will be shown in Section 3.2.2, Romanian belongs to the second class.

Besides the DN effect, in DN languages negative constituents like *nothing*, *nichts*, *niemand* also yield sentential negation alone, while *nessuno*, *nadie*, *niczego* do not. Compare the negative meaning of (85) below to the obligatoriness of the NM in (84a-ii), (84b-ii), and (84c-ii):[3]

(85) Negative quantifiers in DN languages

a. John said **nothing**. (English)

b. Hans hat **nichts** gekauft.
Hans has nothing bought
'Hans didn't buy anything.' (German)

c. Frank heeft **niemand** gezien.
Frank has nobody seen
'Frank didn't see anybody.' (Dutch)

Negative constituents in DN languages are usually called *negative quantifiers*, while the notion *n-word* is used for negative constituents in NC languages. In this thesis I will use the term *n-word* for negative constituents in both NC and DN languages. A defining property of n-words is their ability to appear in contexts where they independently contribute negative meaning, so we can formulate this as a condition for qualifying a constituent as an n-word (86):[4]

(86) An expression is an *n-word* if there are contexts where it independently contributes negative meaning.

Fragmentary answers are one context that satisfies the condition (86) in all NC languages, just like in DN languages. Thus a question like the English (87) can be answered with an n-word in each of the three NC languages below, and the answer is invariably interpreted as negative:

(87) Who came?

a. **Nessuno.** (Italian)

b. **Nadie.** (Spanish)

c. **Nikt.** (Polish)

'Nobody.'

3.1.2 Strict vs. non-strict NC

Within the class of NC languages, Giannakidou (2006) distinguishes strict NC from non-strict NC. The former refers to languages where the presence of an n-word in a sentence always requires the cooccurrence of the NM on the verb, regardless of the syntactic position that the n-word occupies.

[3] Italian *nessuno* and Spanish *nadie* can yield sentential negation in some special cases described in Section 3.1.2, but the contrast above still holds, since in (84a-ii) and (84b-ii) they wouldn't be able to.

[4] See also Giannakidou (2006) for a more restricted definition by which n-words only appear in NC languages.

In non-strict NC languages, an n-word preceding the verb is incompatible with the NM and is able to license NC with other n-words. The NC constructions of the kind in (88a-ii) are usually called 'negative spread' (den Besten (1986)).

Slavic languages typically belong to the former class, most Romance languages to the latter. The examples in (88a) and (88b) illustrate the contrast between the two NC classes:

(88) a. Non-strict NC (Italian)

i. **Nessuno** (***non**) è venuto.
nobody (*NM) is come
'Nobody came.'

ii. **Nessuno** (***non**) ha visto **nessuno**.
nobody (*NM) has seen nobody
'Nobody saw anybody.'

b. Strict NC (Polish)

i. **Nikt** *(**nie**) dał Marysi książki.
nobody NM gave Mary book
'Nobody gave Mary a/ the book.'

ii. **Nikt** *(**nie**) uderzył **nikogo**.
nobody NM hit noone
'Nobody hit anybody.' (Przepiórkowski and Kupść (1999, p. 213))

The asymmetric conditions imposed on the presence of the NM by preverbal and postverbal n-words indicate that Italian displays the non-strict variety of NC ((88a) vs. (84a-ii)).[5] The Polish examples in (88b) repeat the situation already observed with postverbal n-words in (84c-ii), and thus establish that Polish is a strict NC language. In Section 3.2.2 it will be shown that, unlike other Romance languages, Romanian is typically a strict NC language.

3.2 Negation and NC in Romanian

3.2.1 General facts about negation in Romanian

GA (1966) and Avram (1986) describe the use of the NM *nu* (with the phonological variant *n-*)[6] before the main verb as the common way to negate a sentence in Romanian. The sentence in (89b) is the negative counterpart to (89a):

(89) a. Studenţii au citit romanul.
students-the have read novel-the
'The students read the novel.'

b. Studenţii **nu** au citit romanul.
students-the NM have read novel-the
'The students didn't read the novel.'

[5]The presence of the NM with a preverbal n-word is not completely ungrammatical, as (88a) may suggest. Under special intonational conditions, the two sentences may receive a DN interpretation (see Zanuttini (1991)). But for a NC reading, the presence of the NM is excluded.

[6]The reduced form *n-* is optionally used under certain phonological conditions described in Section 5.5.6.

Besides negating the main verb of a sentence, *nu* is also able to negate a constituent, like in (90). In this case, the sentence is affirmative, because the verb is not negated. This role of *nu* is usually referred to as "constituent negation" and is syntactically distinguished from the one in (89), the "negative marker". I will gloss constituent negation *nu* with *not*, and the negative marker with *NM*:

(90) a. **Nu** studenţii au citit romanul.
not students-the have read novel-the
'It was not the students who read the novel.'

b. Studenţii au citit **nu** romanul, ci prefaţa.
students-the have read not novel-the, but preface-the
'The students read not the novel, but the preface.'

The NM *nu* appears with all the finite verb forms (including the subjunctive), and with the infinitive. The other non-finite forms, i.e. present/ past participle and supine, become negative by means of the prefix *ne*-, attached to the verb:

(91) **nu** va scrie/ **nu** ar scrie/ sa **nu** scrie/ a **nu** scrie/ **ne**scriind/ **ne**scris/ de **ne**scris
NM will write/ NM would write/ SJ NM write/ to NM write/ un-writing/ un-written/ of un-written

Although negated non-finite verb forms cannot contribute negation to the main clause, they do express negation of the predication within absolute clauses or reduced relative clauses:

(92) a. Maria a mers mai departe, **ne**acordînd atenţie oamenilor din jur.
Maria has gone further, un-paying attention people around
'Maria walked further, without paying attention to the people around.'

b. Acest articol **ne**citat de către critici este de fapt foarte interesant.
this article un-cited by critics is in fact very interesting
'This article, which wasn't cited by critics, is actually very interesting.'

Another means of negating a constituent within a sentence is the preposition *fără* 'without', which can negate an NP, but also an infinitival or a subjunctive clause:

(93) Maria a rezolvat problema **fără** ajutor/ a cere ajutor/ să ceară ajutor.
Maria has solved problem-the without help/ to ask help/ SJ ask help
'Maria solved the problem without help/ asking for help.'

3.2.2 N-words and NC

Besides the NM contributing negation to the verb, GA (1966) and Avram (1986) mention n-words as negative constituents which give rise to negative concord:[7]

(94) a. Studenţii *(**nu**) au citit **niciun** roman.
students-the NM have read no novel
'The students read no novel.'

[7] GA (1966) and Avram (1986) use the terms "double negation" (for NC with one n-word) and "multiple negation" (for NC with two or more n-words).

b. Studenţii *(**nu**) au citit **niciodată niciun** roman.
students-the NM have read never no novel
'The students never read any novel.'

c. **Niciun** student *(**nu**) a citit romanul.
no student NM has read novel-the
'No student read the novel.'

d. **Niciun** student *(**nu**) a citit **niciun** roman.
no student NM has read no novel
'No student read any novel.'

Romanian, like other languages, has both bare n-words and what could be called 'n-determiners':

(95) Romanian n-words

- Nouns (pronouns):
  - **nimeni** ('nobody'), with dative-genitive **nimănui**
  - **nimic** ('nothing')
  - **niciunul/ niciuna** (masculine/ feminine of 'no one', 'none') with dative-genitive **niciunuia/ niciuneia**
- Adverbs:
  - **niciodată** ('never')
  - **nicăieri/ niciunde** ('nowhere')
  - **nici(de)cum** ('nohow', 'nowise'), **deloc** ('at all')
- Determiners:
  - **niciun/ nicio** (masculine/ feminine singular of 'no') with dative-genitive **niciunui/ niciunei**

(94) shows that the presence of an n-word always requires the NM on the verb. With respect to Giannakidou's distinction in Section 3.1.2, Romanian qualifies as a strict NC language: the preverbal n-word in (94c) and (94d) doesn't make the presence of the NM on the verb any more optional than the postverbal n-words in (94a) and (94b).[8]

NC also appears in non-finite and 'without'-constructions, as long as the prefix *ne-* or the negative preposition *fără* is present:

(97) a. Maria a mers mai departe, *(**ne**)acordînd atenţie **nimănui**.
Maria has gone further, un-paying attention nobody
'Maria walked on, without paying attention to anybody.'

b. Acest articol *(**ne**)citat de **niciun** critic este de fapt foarte interesant.
this article un-cited by no critic is in fact very interesting
'This article, which wasn't cited by any critic, is actually very interesting.'

[8] Ionescu (1999) and Iordăchioaia (2004) show that strict NC does not hold for all instances of negation involving n-words in Romanian. In past participial constructions, an n-word preceding the affirmative verb form negates it:

(96) articol de **nimeni** citat
article by nobody cited
'article which hasn't been cited by anybody'

These constructions are quite rare and usually stylistically marked. Our discussion at this point only takes typical strict NC constructions into account.

c. Maria a rezolvat problema **fără** **niciun** ajutor/ a cere ajutor **nimănui**/ să
Maria has solved problem-the without no help/ to ask help nobody/ SJ
ceară ajutor **nimănui**.
ask help nobody
'Maria solved the problem without any help/ asking anybody for help.'

In the sentences above, *ne*- and *fără* exhibit strict NC the same way as *nu* does in (94). This pattern is not particular to Romanian, since NC languages usually allow NC within non-finite clauses, and under *without*:

(98) a. Spanish
Pedro compró el terreno **sin** contarselo a **nadie**.
Pedro bought the land without telling-CL.CL. to nobody
'Pedro bought the land without telling anybody.' (Herburger (2001), p. 297)

b. Polish
Zaczął **bez** **żadnych** wstępów.
started without none introductions
'He started straight away.' (Przepiórkowski and Kupść (1999), p. 218)

c. French
Anne est partie **sans** **rien** dire.
Anne has left without nothing say
'Anne has left without saying anything.' (de Swart and Sag (2002), p. 411)

In conclusion, Romanian n-words can be licensed within strict NC constructions by the NM *nu* (94), the prefix *ne*-, and the negative preposition *fără* 'without' (97). In what follows, the discussion will concern contexts like (94), but the conclusions will be formulated in a way that will allow an extension to cover the cases in (97).

3.2.3 NPIs

In addition to n-words, Romanian has a class of indefinites sensitive to negation, which best resemble English negative polarity items like *any*. They are ungrammatical in positive contexts, and are licensed only under various forms of negative(-like) licensers:

(99) a. * Mary bought *any* book.
b. **Few** students bought *any* book.

(100) a. * Maria a cumpărat *vreo* carte.
Maria has bought any book
b. **Puţini** studenţi au cumpărat *vreo* carte.
few students have bought any book
'Few students bought any book.'

(101) Romanian NPIs
- Nouns (pronouns):
  - **cineva** ('some-/ anybody') with dative-genitive **cuiva**

  - **ceva** (‘some-/ anything’)
  - **vreunul/ vreuna** (masculine/ feminine of ‘anyone’) with dative-genitive **vreunuia/ vreuneia**
- Adverbs:
  - **vreodată** (‘ever’)
- Determiners:
  - **vreun/ vreo** (masculine/ feminine singular of ‘any’) with dative-genitive **vreunui/ vreunei**

As the English translation already indicates, the bare nouns *cineva* and *ceva* are ambiguous between a specific and a non-specific interpretation. Just like English *some*-indefinites in (102b), they outscope negation (102ai), but they can also be interpreted in the scope of negation (102aii), like typical English *any*-NPIs in (102c):

(102) a. Maria **nu** a zis ca a văzut pe *cineva*.
Maria NM has said that has seen some-/ anybody
i. $\exists > \neg$: ‘There is somebody who Maria didn’t mention to have seen.’
ii. $\neg > \exists$: ‘Maria didn’t say that she had seen anybody.’

b. Maria did**n’t** say that she saw *somebody*.
i. $\exists > \neg$ ii. # $\neg > \exists$

c. Maria did**n’t** say that she saw *anybody*.
i. # $\exists > \neg$ ii. $\neg > \exists$

This ambiguity disappears in the case of indefinites containing *vre*-, which are unambiguously interpreted within the scope of negation:

(103) Maria **nu** a zis ca a văzut *vreun* hoţ.
Maria NM has said that has seen any thief
i. # $\exists > \neg$
ii. $\neg > \exists$: ‘Maria didn’t say that she had seen any thief.’

3.3 N-words between NPIs and NQs

In order to determine which of the two analyses in Section 1.2 is appropriate for Romanian, we first have to establish the semantic status of n-words, that is, whether they are NPIs or negative quantifiers. This section brings arguments against the NPI hypothesis for Romanian n-words.

After a general presentation of the NPI licensing conditions (Section 3.3.1), in Section 3.3.2 I present the reasons why an NPI analysis is not desirable for Romanian. The subsequent sections bring additional arguments for the negative character of n-words.

3.3.1 NPIs

Ladusaw (1980) addresses two main problems concerning negative polarity items like *any*: 1) NPIs are licensed by some operators but not by others, 2) the operator has to precede the NPI in the syntax:

(104) a. **Few**/***Many** people saw *anybody*.

b. He did*(**n’t**) see *anybody*.

c. * *Anybody* did**n't** see him.

(104a) and (104b) indicate that English *anybody* can be interpreted in the scope of *few* and *not*, but not in the scope of *many* or that of an affirmative verb. The ungrammaticality of (104c) adds to this condition a syntactic observation: it is not enough to interpret an NPI within the semantic scope of a licenser, it also has to be preceded by that licenser in the syntax. The former constraint is referred to as the 'semantic licensing' of NPIs, and the latter as the 'syntactic licensing'.

With respect to 'semantic licensing', the idea put forward by Ladusaw is that NPI *any* is an existential quantifier which must be licensed in the scope of a negative operator that is at least *downward entailing*.

3.3.1.1 A hierarchy of negative licensers

van der Wouden (1997) and Zwarts (1998) give a semantic characterization of negative contexts which sheds more light on the semantic licensing differences among NPI classes. They distinguish between downward entailing (DE), anti-additive, and antimorphic operators:

Definition 3.1 *Given X and Y, a function F is*
a. downward entailing *iff:*
$X \subseteq Y \rightarrow F(Y) \subseteq F(X)$
b. anti-additive *iff:*
$F(X \text{ or } Y) = F(X) \text{ and } F(Y)$
c. antimorphic *iff:*
$F(X \text{ or } Y) = F(X) \text{ and } F(Y)$
$F(X \text{ and } Y) = F(X) \text{ or } F(Y)$

DE operators are the largest class of the three, and are characterized by the least negative semantics satisfying the condition in DEFINITION 3.1a, which is the weakest. Anti-additive operators are a subclass of DE operators, characterized by a stronger negativity constraint. The most negative operators are the antimorphic ones, constituting a further restricted subclass of the anti-additive operators. Thus there is an inclusion relation between the three classes of negative operators, in the order in which they are presented in DEFINITION 3.1. This relation is directly reflected in the examples below.

In (105) there are three DE operators: the quantifier *few*, the preposition *without*, and the NM *not*. *Many*, a positive operator, does not obey the DE condition in DEFINITION 3.1, which explains the ungrammatical version of the sentence in (104a).

(105) For X= ⟦spinach⟧and Y= ⟦vegetable⟧, ⟦spinach⟧$\subseteq$ ⟦vegetable⟧:

a. **Many** people eat vegetables. $\not\rightarrow$ **Many** people eat spinach.

b. **Few** people eat vegetables. $\rightarrow$ **Few** people eat spinach.

c. John ate his sandwich **without** vegetables.$\rightarrow$ John ate his sandwich **without** spinach.

d. John does**n't** eat vegetables. $\rightarrow$ John does**n't** eat spinach.

If we take X to stand for "flower" and Y for "book" in (106), we can check the three expressions above for anti-additivity:

(106) a. **Few** people brought flowers *or* books. $\neq$ **Few** people brought flowers *and* **few** people brought books.

b. John came **without** flowers *or* books. = John came **without** flowers *and* John came **without** books.

c. John did**n't** bring flowers *or* books. = John did**n't** bring flowers *and* John did**n't** bring books.

Without and *not* in (106b) and (106c) meet the condition in DEFINITION 3.1b and thus qualify as anti-additive. The lack of equivalence in (106a) indicates that *few people*, although DE (105b), is not anti-additive. Negative indefinites (n-words) containing *no*, like *nobody*, *nothing*, *no student* are also anti-additive (see van der Wouden (1994)).

In (107), it can be seen that the class of antimorphic expressions is even more restricted than that of anti-additive expressions: *without* does not pass the second antimorphicity test in (107b):

(107) a. John came **without** flowers *and* books. $\neq$ John came **without** flowers *or* John came **without** books.

b. John did**n't** bring flowers *and* books. = John did**n't** bring flowers *or* John did**n't** bring books.

In conclusion, *not* is the strongest negative expression of the three considered here, since it is the only one that fulfills the antimorphicity conditions.

3.3.1.2 Licensing of NPIs

Given the hierarchy of negative functions in DEFINITION 3.1 and the proposal in Ladusaw (1980) that NPIs are licensed by DE operators, it follows that English *any*-NPIs should be grammatical in the scope of *few*, *without*, and *not*, but not in the scope of *many*. This is confirmed by the data in (104a) and (104b), and (108) below:

(108) He managed **without** *any* help.

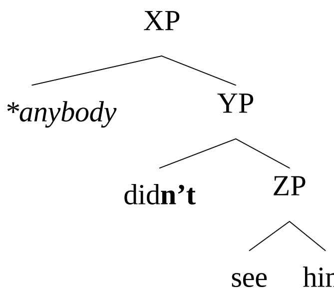
Furthermore, (104c) shows that the NPI has to be preceded by the licensing operator in the syntax. To account for this, Ladusaw proposes that NPIs must be *c-commanded* by a DE operator. A common definition of c-command is the one below:

(109) In a tree, a node A *c-commands* node B iff

- neither dominates the other, and
- every (branching) node dominating A also dominates B.

The tree in (110) represents the structure of sentence (104c) repeated as (111a). Note that *anybody* c-commands *didn't*, but not vice versa, because the first branching node (YP) dominating *didn't* does not dominate *anybody*. In Ladusaw's account, this explains the ungrammaticality of (111a).

(110)



(111) a. * *Anybody* did**n't** see him.

b. He did**n't** see *anybody*.

In (110), *didn't* c-commands the direct object position, which explains why (104b)/ (111b), with the NPI *anybody* in the object position, is grammatical.

3.3.1.3 Classes of NPIs

van der Wouden (1994, 1997) and Zwarts (1998) show that there are three classes of NPIs, which are semantically licensed by the three classes of negative operators. Within van der Wouden's (1994) terminology, DE operators license *weak* NPIs, anti-additive operators license NPIs of *medium* strength, while antimorphic operators license *strong* NPIs.

Considering the hierarchy of negative operators presented above, weak NPIs should be successfully licensed by each of the three kinds of operators, a fact that is confirmed in (104a) and (104b), and (108) by *any* which is grammatical with DE *few*, anti-additive *without*, and antimorphic *not*. In (112), *any* is also licensed by the DE *at most*, and the anti-additive *nobody*:

(112) a. **At most** three people brought *any* flowers.
b. **Nobody** brought *any* flowers.

NPIs of medium strength like *yet* are licensed by anti-additive operators, but not by DE ones:

(113) a. * **At most** three people brought flowers *yet*.
b. **Nobody** has brought the flowers *yet*.
c. John has**n't** brought the flowers *yet*.

Finally, the strong NPI *a bit* can only be licensed by the antimorphic operator *not*:

(114) a. * **At most** three linguists were *a bit* happy about these facts.
b. * **No one** was *a bit* happy about these facts.
c. Chomsky was**n't** *a bit* happy about these facts. (van der Wouden (1994), p. 19)

Some medium and strong NPIs have been noticed to display collocational properties: they can also appear in positive contexts, but they are interpreted as NPIs only under appropriate negative licensers. For instance, *yet* is synonymous with *still* in positive contexts (115a), but not in negative ones, where it gets an NPI reading (115b):

(115) a. *Yet*, John is a nice guy. = *Still*, John is a nice guy.
b. **Nobody** was there *yet*. $\neq$ **Nobody** was *still* there.

The same contrast appears with *a bit*, which as a non-NPI is synonymous with *a little* (116a), a fact that does not hold for NPI *a bit* (116b):

(116) a. John is *a bit* upset. = John is *a little* upset.
b. John is **not** *a bit* happy. $\neq$ John is **not** *a little* happy.

Ladusaw's analysis of NPIs concentrates on the properties of *any*, i.e. 'weak' NPIs. The other two classes of NPIs are semantically more restricted than *any*, so they satisfy the licensing conditions imposed on *any*, plus their specific restrictions. Thus *yet* and *a bit* – in their NPI form – cannot be licensed by the non-DE quantifier *many*, as (117a) and (117b) indicate: the former is totally ungrammatical, while the latter can only receive the non-NPI reading.

(117) a. * **Many** students were there *yet*.
b. **Many** students were *a bit*/ *a little* upset.

Moreover, they must be c-commanded by their negative licenser. Otherwise, they again lose the NPI reading (see (118a) and (118b)).

(118) a. There was *yet/ still* **nobody** to answer.

b. ?? They were *a bit/ a little* **not** happy.

In conclusion, the observations in Ladusaw (1980) concerning both the semantic and the syntactic licensing of *any* carry over to stronger NPIs, which are semantically more constrained.

3.3.1.4 Roofing

In Ladusaw (1980), NPI *any* is assumed to contribute an existential quantifier. But this quantifier does not behave like a typical existential quantifier contributed by an indefinite, since no other operator is allowed to intervene between it and its licenser. (119a) has a reading in which the universal quantifier intervenes between the negative operator *not* and the existential quantifier carried by the indefinite *a student*. In (119b) this reading is not available anymore, because the existential quantifier is contributed by the NPI *any*. The only reading is the one in which no operator intervenes between the negation and the existential quantifier (119bii):

(119) a. Meg did**n't** read *every* book to **a** student.
$\neg\forall x[book(x) \rightarrow \exists y(student(y) \wedge read(Meg, x, y))]$

b. Meg did**n't** read *every* book to **any** student.
i. # $\neg\forall x[book(x) \rightarrow \exists y(student(y) \wedge read(Meg, x, y))]$
ii. $\neg\exists y[student(y) \wedge \forall x(book(x) \rightarrow read(Meg, x, y))]$

In view of this observation, an extra stipulation has to be made about the semantic licensing of *any*.[9] Ladusaw (1992) gives up the assumption that NPI *any* contributes an existential quantifier, in favor of a general definition of NPIs in terms of Heimian 'indefinites' (cf. Heim (1982)). Thus NPIs are considered to be variables with descriptive content and no inherent quantificational force, which become existentially bound at some point in the interpretation. The existential binding is only available when the indefinite falls in the restriction or the nuclear scope of an operator. This binding operator is called 'the *roof* of the indefinite'. With the notion of a 'roof' the immediateness between the licenser and the NPI comes for free and no additional stipulation is necessary.

In conclusion, the semantic licensing of NPIs is formulated as a general *roofing* condition: the roof must be an appropriately negative operator. *Any*-NPIs, as a subclass, must be semantically roofed by a DE operator, and c-commanded by it in the syntax. In Section 3.3.2.1 below I compare the licensing of NPIs with that of n-words.

3.3.2 N-words as NPIs

In what follows, it will be shown that the assumption that Romanian n-words are NPIs encounters three major problems concerning: 1) the status of the licenser, 2) locality conditions, and 3) modification by *almost*. They are addressed in this order in Sections 3.3.2.2, 3.3.3, and 3.3.4, after a presentation of the main claims of the NPI approaches with respect to n-words (Section 3.3.2.1).

3.3.2.1 Ladusaw (1992)

The first influential NPI analysis is given in Ladusaw (1992) which mainly addresses non-strict NC Romance languages and English NC varieties. This proposal has been implemented in various semantic and/ or syntactic-semantic frameworks (see for instance Richter and Sailer (1999*b*)'s HPSG

[9] See for instance the Immediate Scope Constraint in Linebarger (1980).

analysis with Ty2 expressions, Przepiórkowski and Kupść's (1999) HPSG analysis within Situation Semantics, and Zeijlstra's (2004) Minimalist account). Given the great impact that Ladusaw (1992) had on NPI approaches, I take this proposal as most representative for the NPI analysis of n-words.

As briefly described in the previous section, Ladusaw (1992) redefines NPI licensing in terms of semantic roofing by a negative operator. This is the most general condition on NPIs, and Ladusaw argues that a language can display various classes of NPIs, which are licensed via a particularization of the general semantic roofing condition. These classes also include n-words.

We now consider the operator that roofs n-words as a kind of NPIs. The class of NPIs of the *any* type is broader than that of *nobody*, since they accept roofing under any DE operator, so they are more permissive. Ladusaw (1992) argues that n-words impose a stronger restriction on their roof, that of anti-additivity (DEFINITION 3.1b). This is confirmed by (120), where the Italian n-word *niente* is grammatical in the scope of *senza* ('without'), but not in the scope of *pochi* ('few'), an appropriate context for the NPI *alcunché* ('anything'):

(120) a. **Pochi** capiscono *alcunché*/ ***niente** di logica.
'Few people understand anything about logic.' (Italian, Zanuttini (1991))

b. ... **senza** capire **niente** di logica
'without understanding anything about logic.'

Ladusaw's theory also predicts the grammaticality of (121a) and (121b) below. The sentential negation *n't* and the n-word *nobody* count each as anti-additive operators. But the ungrammaticality of (121c) comes unexpected if we consider that *nothing* is anti-additive, just like *nobody* in (121b). Thus (121c) must violate a syntactic condition. This is not c-command, since *nobody* is c-commanded by *nothing*. Moreoveor, *anyone* is grammatical in (121d):

(121) a. She did**n't** give **nothing** to **nobody**.

b. **Nobody** said **nothing**.

c. * She gave **nothing** to **nobody**.

d. She gave **nothing** to **anyone**.

(Ladusaw (1992), pp. 249–250)

The kind of contrast between (121a) – (121b) and (121c) is attested in English NC varieties and in non-strict NC languages like Italian and Spanish. The grammaticality of (121a) suggests that the expressor of negation must be associated with the head of the sentence (i.e. the verb). At the same time, the sentence in (121b) is fine, which indicates that an n-word preceding the head of the clause can license another n-word. For Italian, Zanuttini (1991) formulates the constraint that negation must have sentential scope, which only happens if the negative operator c-commands the verb. In this way, one can explain how the NM in (121a) and the n-word in (121b) license the postverbal n-words.

Ladusaw offers a more elegant solution: he starts from the idea that n-words in NC are NPIs that have to be roofed by an anti-additive operator. But this operator doesn't need to be part of a lexical meaning, it can also be constructional, in the sense that it is related to a structural feature that is not visible in the clause. Thus the operator is simply added in at some point in the interpretation of a sentence, and n-words are taken to act as licensers for its insertion.

Ladusaw (1992) gives the outlines of a syntax-semantics both in GPSG (Generalized Phrase Structure Grammar) and in GB (Government and Binding). In GPSG he proposes that there is a [neg] feature inherently specified for all negative phrases. This feature must be part of the lexical specification of the head of a clause in order to trigger sentential negation, and this only happens when the feature is already on the verb (122a) or it gets there by percolation from an n-word specifier (122b) or an adjoined sister node (122c):

(122) a. John did**n't** speak.
b. **Nobody** spoke.
c. John **never** spoke.

The transmission of the [neg] feature from the n-word to the verbal head in (122b) and (122c) is made possible by the principle in (123):

(123) (Ladusaw (1992), p. 254)
A category inherits the feature [neg] from a specifier sister or an adjoined sister.

The sentences in (122) are all correctly interpreted as negative within such an analysis, and the NC instances in (121) can be explained if one understands n-words as roughly *self-licensing NPIs*.

In conclusion, the core idea of the NPI analysis in Ladusaw (1992) is that n-words as NPIs have to be semantically licensed in the scope of an anti-additive operator which must be syntactically licensed either by a head already marked as negative or by an n-word appearing in a special configuration (specifier-head or adjunct-head phrase) with the head.

3.3.2.2 A semantic licenser for Romanian n-words?

The central claim of the NPI analysis that n-words need to be semantically licensed by an operator is refuted here on the basis of the semantic independence of Romanian n-words.[10]

Romanian negative licensers The classification of negative operators given in DEFINITION 3.1 can also be applied to Romanian. The three negative contexts discussed above correspond to the Romanian *puţini* ('few'), *fără* ('without'), and *nu* ('not'). *Mulţi* ('many') is not even DE:

(124) a. **Mulţi** oameni mănîncă legume. $\not\rightarrow$ **Mulţi** oameni mănîncă spanac.
many people eat vegetables many people eat spinach

b. **Puţini** oameni mănîncă legume. $\rightarrow$ **Puţini** oameni mănîncă spanac.
few people eat vegetables few people eat spinach

c. Ion mănîncă sandwich-ul **fără** legume. $\rightarrow$ Ion mănîncă sandwich-ul **fără** spanac.
John eats sandwich without vegetables John eats sandwich without spinach

d. Ion **nu** mănîncă legume. $\rightarrow$ Ion **nu** mănîncă spanac.
John not eats vegetables John not eats spinach

The sentences in (124) are parallel to the English ones in (105), and they show that *puţini*, *fără*, and *nu* are DE, while *mulţi* is not.

The examples in (125), the Romanian counterpart to (106), indicate that *fără* and *nu* are also anti-additive, while *puţini* is not:

(125) a. **Puţini** oameni au adus flori *sau* cărţi. $\neq$ **Puţini** oameni au adus flori *şi* **puţini** oameni au adus cărţi.
few people have brought flowers or books. few people have brought flowers and few people brought books.

[10]Variants of an NPI analysis for NC in Romanian, which assume a semantic licensing mechanism for n-words, have been proposed in Ionescu (1999, 2004) and Barbu Mititelu and Maftei Ciolăneanu (2004). A close consideration of the motivation behind these approaches in comparison to the present analysis is postponed for Chapter 6. In this section, I concentrate on the empirical evidence that supports the present NQ analysis.

b. Ion a venit **fără** flori *sau* cărţi. = Ion a venit **fără** flori *şi*
John has come without flowers or books John has come without flowers and
Ion a venit **fără** cărţi.
John has come without books.

c. Ion **nu** a adus flori *sau* cărţi. = Ion **nu** a adus flori *şi* Ion
John not has brought flowers or books John not has brought flowers and John
nu a adus cărţi.
not has brought books

Finally, Romanian *nu*, like English *not* in (107b), is also antimorphic, but *fără* 'without' is not:

(126) a. Ion a venit **fără** flori *şi* cărţi. ≠ Ion a venit **fără** flori *sau*
John has come without flowers and books John has come without flowers or
Ion a venit **fără** cărţi.
John has come without books.

b. Ion **nu** a adus flori *şi* cărţi. = Ion **nu** a adus flori *sau* Ion
John not has brought flowers and books John not has brought flowers or John
nu a adus cărţi.
not has brought books

The syntactic condition on Romanian n-words The syntactic licensing contrast illustrated in (121), which shows up in non-strict NC languages, does not arise in Romanian, a strict NC language. The ungrammaticality of the Romanian counterpart of (121c), given in (127c), must be due to the general constraint on NC that the NM be present on the verb, which we saw in (94c) and (94d), repeated below as (127a) and (127b):

(127) a. **Niciun** student *(**nu**) a citit romanul.
no student NM has read novel-the
'No student read the novel.'

b. **Niciun** student *(**nu**) a citit **niciun** roman.
no student NM has read no novel
'No student read any novel.'

c. *(**Nu**) a dat **nimănui nimic**.
NM has given nobody nothing
'S/he gave nobody anything.'

Thus the principle in (123) is not necessary in Romanian.

The semantic status of the licenser If Romanian n-words are treated as NPIs, the NM[11] is a reasonable choice for a licenser: its obligatory presence with n-words is similar to that of a DE operator with NPIs.[12] But with the NM as a licenser, there is a puzzling asymmetry in the licensing conditions of n-words and NPIs: n-words are syntactically more independent, and semantically more restricted than NPIs. This casts serious doubt on the claim that n-words are a class of NPIs.

[11] I will assume here that *fără* 'without' and *ne-* 'un-' in NC structures count as NMs, too.
[12] Przepiórkowski and Kupść's (1999) analysis of Polish NC has the same starting point: the NM is the NPI-licenser.

First, Romanian n-words do not need to be c-commanded by the NM, as the contrast in (128) indicates. This suggests that n-words are syntactically less constrained than *any* NPIs:[13]

(128) a. **Niciun** student **nu** a venit.
no student NM has come
'No student has come.'

b. * *Vreun* student **nu** a venit.
any student NM has come

Let us consider Ladusaw's approach in relation to (128) step by step. Ladusaw (1992, fn. 12, p. 251) states that n-words do not need to be licensed at surface structure (i.e. the c-command condition does not apply) when they license the negative operator, i.e. in (121d), (122b), (122c), where principle (123) applies. This would explain the grammaticality of (128a), if the n-word licensed the negative operator in that position. But we saw above that this principle is unmotivated for Romanian: n-words never need to license a [neg] feature on the verb, because this feature is always there in NC, it is carried by the obligatory NM. If Ladusaw's principle does not apply to Romanian n-words, the grammaticality of (128a) remains unexplained in comparison to the NPI in (128b). The only answer is that the c-command condition does not apply to Romanian n-words in general.

Thus n-words are syntactically less restricted than NPIs. However, they are more restricted in what concerns the semantic value of the licenser: they are excluded in a DE context like the nuclear scope of *puţini* 'few' (129).

(129) **Puţini** oameni ştiu *__niciun__/ *vreun* detaliu despre el.
few people know no/ any detail about him
'Few people know any details about him.'

Ladusaw (1992) claims that anti-additive operators are appropriate licensers for n-words. This explains the ungrammaticality of the n-word in (129), since *puţini* is DE, but not anti-additive (cf. (125a)). The Romanian NMs *nu* 'not' and *fără* 'without' are anti-additive (125), so they are correctly predicted to license n-words in sentences like (94) and (97c), pp. 71–73.

According to our discussion in Section 3.3.1.3, all the NPIs that need a stronger licenser than *any* also need to be c-commanded by their licenser (just like *any*). The semantic licensing cannot take place if the syntactic restrictions are not met. From this point of view, n-words exhibit a contradictory behavior for NPIs: they require a semantically stronger licenser, but they are more independent syntactically. Their syntactic independence, unavailable for typical NPIs, indicates that the semantic licensing does not take place with n-words.

In addition to this, there are two more reasons why the idea that n-words are NPIs licensed by an anti-additive operator cannot be right: 1) the semantics of n-words is negative independently of the NM and 2) in NC the NM does not semantically license the n-words.

First of all, in contexts where the presence of the NM is not required Romanian n-words display anti-additivity, which qualifies them as semantically negative (130):

(130) a. articol [de **nimeni** citat *sau* lăudat] = articol [de **nimeni** citat *şi* de **nimeni** lăudat]
article by nobody cited or praised article by nobody cited and by nobody praised

[13]It should be noted that with Ladusaw's (1992) assumption that n-words are licensed by an abstract operator, one could argue for the existence of such an operator in a syntactic position from where it c-commands the n-word *niciun* in (128). In this thesis I use a surface-oriented syntax which disallows covert operators, so I will not pursue this kind of approach. But see Zeijlstra (2004) for an alternate account.

‘article which hasn’t been cited or praised by anybody’ = ‘article which hasn’t been cited and which hasn’t been praised by anybody’

b. A: Who was at the door?

B: **Nimeni** cunoscut *sau* important. = **Nimeni** cunoscut *şi* **nimeni** important.
nobody known or important nobody known and nobody important

This property obviously differentiates n-words from NPIs, since the latter cannot be interpreted at all in the absence of a licenser. Moreover, if n-words needed to be semantically licensed by an anti-additive operator (Ladusaw (1992)) it would remain unexplained why they cannot license one another in (131), although they can license an NPI:

(131) articol de **nimeni** citat la ***nicio**/ *vreo* conferinţă
article by nobody cited at no/ any conference
‘article that hasn’t been cited at any conference’

Second, if we test the anti-additivity of the NM when n-words are involved (132), the interpretation of the sentence indicates that this property is not preserved over n-words:

(132) Ion **nu** a citit **nicio** carte *sau* **niciun** articol.
John NM has read no book or no article

a. Anti-additivity

$\neq$ Ion **nu** a citit **nicio** carte *şi* Ion **nu** a citit **niciun** articol.
John NM has read no book and John NM has read no article

‘John didn’t read any book and John didn’t read any article.’

b. Ellipsis

= Ion **nu** a citit **nicio** carte *sau* Ion **nu** a citit **niciun** articol.
John NM has read no book or John NM has read no article

‘John read no book or John read no article.’

The most natural interpretation of (132) is the one in (132b), where the NM does not take scope over the disjunction between the two n-words. The sentence is understood as elliptical, i.e. as a disjunction between two negative clauses. The situation is different in the case of *vreun* NPIs (133), where the first available reading is the one in which the NM takes scope over the disjunction of the two NPIs (133a), so the final interpretation is a conjunction of two negative sentences. This indicates that, unlike with n-words, the NM displays anti-additivity with respect to NPIs:

(133) Ion **nu** a citit **vreo** carte *sau* **vreun** articol.
John NM has read any book or any article

a. Anti-additivity

= Ion **nu** a citit **vreo** carte *şi* Ion **nu** a citit **vreun** articol.
John NM has read any book and John NM has read any article

‘John hasn’t read any book and John hasn’t read any article’

b. Ellipsis

= Ion **nu** a citit **vreo** carte *sau* [Ion **nu** a citit] **vreun** articol.
John NM has read any book or John NM has read any article

‘John hasn’t read any book or John hasn’t read any article’

The interpretation (133b) with ellipsis is also possible for the NPI, as it is with any other item. What is important is that there is a contrast between (132) and (133) which casts serious doubt on the assumption that n-words are NPIs.

The data in (132) and (133) raise an additional question: how is it possible for a negative operator to be anti-additive with respect to some items (NPIs) and not anti-additive with respect to others (n-words)? The unavailability of the anti-additive reading in (132) is most likely the effect of the syntactic conditions that govern NC in Romanian (see Section 3.3.3). Importantly, anti-additivity is possible for a marginal sentence like (134a). If one forces[14] *nu* to take scope over the disjunction of the two n-words, the effect is an interpretation containing a conjunction of two sentences, as predicted by anti-additivity. But in this case both sentences are interpreted as affirmative, which means that a DN effect occurs between *nu* and each of the two n-words. Note here that I speak of *nu* and not of the NM. In Section 5.5.2, I will show that this *nu* is syntactically different from the NM *nu* (cf. Barbu (2004)). This difference will also explain the marginality of the sentence in (134a): there is no NM to (syntactically) license the two n-words, although the intended semantic effect can be obtained if we disregard the syntax.

(134) a. ?? Ion **NU** a citit **nicio** carte *sau* **niciun** articol.
John NM has read no book or no article

b. = Nu e adevărat că Ion nu a citit nicio carte *şi* nu e adevărat că Ion
NM is true that John NM has read no book and NM is true that John
nu a citit niciun articol. = Ion a citit cărţi *şi* Ion a citit articole.
NM has read no article John has read books and John has read articles

'It is not true that John read no book and it is not true that John read no article.' = 'John did read books and John did read articles.'

The equivalence in (134) suggests that it is not only *nu* that is negative in (134a), but also each of the two n-words. Given these observations, it is impossible to maintain the assumption that n-words are semantically licensed by the negative marker.

NC constructions with *fără* 'without' or with *ne-* 'un-' display a behavior similar to that of the NM *nu* when it comes to anti-additivity. They exhibit anti-additivity over NPIs (135a)/ (135b), but not over n-words (136)/ (137):

(135) a. Ion a venit **fără** **vreo** floare *sau* **vreo** carte. = Ion a venit **fără** **vreo**
John has come without any flower or any book John has come without any
floare *şi* **fără** **vreo** carte
flower and without any book

'John came without any flower or book.' = 'John came without any flower and without any book.'

b. Maria a mers mai departe, **ne**acordînd atenţie **vreunui** coleg *sau* **vreunui**
Mary has gone further un-paying attention any colleague or any
student.
student

'Mary walked on not paying attention to any colleague or student.'

[14] This usually involves a special emphasis on *nu* (marked in (134) by capital letters) immediately followed by an intonational break.

= Maria a mers mai departe, **ne**acordînd atenţie **vreunui** coleg *şi*
Mary has gone further un-paying attention any colleague and
neacordînd atenţie **vreunui** student.
un-paying attention any student

'Mary walked on not paying attention to any colleague and not paying attention to any student.'

(136) Ion a venit **fără** **nicio** floare *sau* **nicio** carte.
John has come without no flower or no book

a. ≠ Ion a venit **fără** **nicio** floare *şi* **fără** **nicio** carte
John has come without no flower and without no book

'John came without any flower and without any book.'

b. = Ion a venit **fără** **nicio** floare *sau* Ion a venit **fără** **nicio** carte
John has come without no flower or John has come without no book

'John came without any flower or without any book.'

(137) Maria a mers mai departe, **ne**acordînd atenţie **niciunui** coleg *sau* **niciunui** student.
Mary has gone further un-paying attention any colleague or any student

a. ≠ Maria a mers mai departe, **ne**acordînd atenţie **niciunui** coleg *şi*
Mary has gone further un-paying attention no colleague and
neacordînd atenţie **niciunui** student.
un-paying attention no student

'Mary walked on not paying attention to any colleague and not paying attention to any student.'

b. = Maria a mers mai departe, **ne**acordînd atenţie **niciunui** coleg *sau*
Mary has gone further un-paying attention no colleague or
neacordînd atenţie **niciunui** student.
un-paying attention no student

'Mary walked on not paying attention to any colleague or not paying attention to any student.'

The NM *nu*, *fără* and *ne-* are all obligatory in the respective NC constructions, so they *are* licensers for the presence of n-words. But given that n-words are anti-additive themselves and their licensers do not exhibit anti-additivity over them, the licensing cannot be semantic like in the case of NPIs. In the next section I will propose that this licensing is *syntactic*.[15]

In conclusion, this section has shown that assuming that Romanian n-words are NPIs leads to both syntactic and semantic problems. First, they are syntactically less restricted than other classes of NPIs which is contradictory for the notion of semantic licensing that we know from NPIs. Second, their interpretation is not dependent on the presence of the licenser, since they are anti-additive. The semantic independence and the syntactic flexibility make the NPI hypothesis undesired for the analysis of n-words. In the next two sections I address two more issues that support this conclusion: the locality conditions on NC and modification by *almost*.

[15] We will see that this *syntactic* licensing of n-words is of a different nature from the 'syntactic licensing' of NPIs, which is c-command.

3.3.3 Locality

If the negative marker is not the semantic licenser of n-words, there are two more questions that need an answer: what is the role of the NM with respect to n-words and what does this tell us about the status of n-words?

The role of the NM A function that the NM obviously plays is that of fixing the sentential scope of the negative quantifier (NQ). The NM is required on the verb with respect to which the negation of the NQ is interpreted. For instance, in a complex sentence containing a subjunctive clause that hosts an n-word the NM can be placed either on the main verb (138a)[16] or on the embedded one (138b). As the English translation shows[17], the negation of the NQ is interpreted in the clause of the negatively marked verb:[18]

(138) a. **Nu** i-aş cere să se mărite cu **nimeni**.
NM CL-would ask SJ RF marry with nobody
'There is nobody I would ask her to marry.'

b. I-aş cere să **nu** se mărite cu **nimeni**.
CL-would ask SJ NM RF marry with nobody
'I would ask her not to marry anybody./ I would ask her to stay unmarried.'

By comparing the sentence in (138) with a similar one in English, it can be observed that the n-word *no one* exhibits the ambiguity that would arise in Romanian, too, if the NM weren't a condition for the presence of the n-word:

(139) (Klima (1964), p. 285)
I will force you to marry **no one**.

a. 'I wo**n't** force you to marry anyone.'

b. 'I would force you **not** to marry anyone.'

Thus the English interpretations in (139a) and (139b) can be regarded as the counterparts of the Romanian sentences in (138a) and (138b) with the NM resolving the scope ambiguity of the n-word.

Locality conditions on NC The idea that in NC the NM marks the scope of the NQ leads to another test (first proposed by Giannakidou) for determining if n-words are empirically closer to NQs or to NPIs. In what follows, it will be shown that the licensing of n-words is subject to the same locality conditions as the scope of bona fide quantifiers. This counts as evidence for the quantificational status of n-words. The licensing of NPIs is less constrained with respect to locality, which differentiates them from n-words.

Subjunctive in Romanian is not a barrier for NC (138a), and apparently neither is it for the scope of a quantifier like *fiecare* ('every'). In (140) below, although *fiecare* appears in the embedded subjunctive clause, there is a reading (140b), where it outscopes the main clause existential:

[16] Note that scrambling the n-word in the embedded clause of (138a) may have effects on the grammaticality of the construction, but this will not concern us here.

[17] CL stands for "clitic", and RF for "reflexive pronoun".

[18] Neg-Raising verbs (Horn (1989), Sailer (2006)) will not be considered here, because they have an exceptional behavior. But given the assumed lexical nature of Neg-Raising, leaving it aside does not compromise the present conclusions.

(140) Un student a încercat să citească fiecare carte.
a student has tried SJ read every book
'A student tried to read every book.'
a. $\exists > \forall$: A (certain) student tried to read every book.
b. $\forall > \exists$: For every book there is a student who tried to read it.

But *fiecare* cannot take scope out of an indicative complement clause over the complementizer *că* 'that', and neither can an n-word be licensed by the NM if an indicative clause boundary intervenes:

(141) a. Un student a zis că a citit fiecare carte.
a student has said that has read every book
'A student said that he read every book.'
i. $\exists > \forall$: A (certain) student said that he read every book.
ii. # $\forall > \exists$: For every book there is a student who said that he read it.
b. * **Nu** a zis că a citit **nicio** carte.
NM has said that has read no book

The data in (140) and (141) suggest a close similarity between Romanian n-words and quantifiers, since the restrictions on their scope are parallel. In addition, English n-words, commonly assumed to be negative quantifiers, display the same scope limitation that we observed for Romanian n-words in (141b). In (142) below, *no book* can take sentential scope within the embedded clause, but not within the matrix clause:

(142) John said that he read **no** book.
a. 'John said that he didn't read any book.'
b. # 'John didn't say that he read any book.'

These facts do not only support the NQ analysis, they also make it unlikely for n-words to be NPIs. As (143) shows, indicative clauses are not barriers for NPI licensing in English or Romanian:

(143) a. Ion **nu** a zis ca a citit *vreo* carte.
John NM has said that has read any book
b. John did**n't** say that he read *any* book.

Syntactic islands provide further supportive evidence: adjunct and relative clauses constitute barriers for NC, but not for NPIs:

(144) a. **Nu** am dezvăluit secrete [care să-l fi expus pe ***niciun**/ *vreun* coleg].
NM have revealed secrets that SJ-CL be exposed PE no/ any colleague
'I didn't reveal secrets that exposed any colleague.'
b. **Nu** am spus asta [pentru că mi-o ceruse ***niciun**/ *vreun* prieten].
NM have said this because CL-CL asked no/ any friend
'I didn't say that because any friend had asked me to (but because I wanted to.)'

In (144a), the n-word *niciun* 'no' embedded in a relative clause cannot be licensed by the NM placed on the matrix verb. In the same context, the NPI *vreun* 'any' is unproblematic. A similar situation holds of (144b), where the n-word and the NPI appear within an adjunct clause. English NPIs in the corresponding translations are also unproblematic when embedded in relative and adjunct clauses.

Finally, it should be noted that the quantificational status of n-words has also been observed in NPI analyses of Romanian NC like Ionescu (1999, 2004). In particular, the 1999 analysis recognizes the scope marking role of the NM with respect to n-words as genuine quantifiers. However, these accounts differ from the present one, to the extent that they deny the negative contribution of n-words in favor of the NM as the sole carrier of negation[19], which fails to explain the facts in Section 3.3.2.2 and in Section 3.4 below.

3.3.4 The *almost*-test

An empirical test widely used in order to establish the semantic status of n-words is modification by *almost* (see Zanuttini (1991), Déprez (1997), Richter and Sailer (1999*b*), among others). The basic generalization is that *almost* can modify universal, but not existential quantifiers:

(145) a. *Almost* everybody came.
b. * *Almost* somebody came.

Zanuttini (1991), a proponent of the NQ analysis for n-words, uses modification by *almost* to support the idea that n-words are universal and not existential negative quantifiers. Representing a negative quantifier in standard predicate logic, presupposes a choice between an existential and a universal quantifier as interacting with negation, as suggested by the truth-conditional equivalence in (146):

(146) $\neg\exists x[P(x) \wedge Q(x)] = \forall x[P(x) \rightarrow \neg Q(x)]$

Existential quantifiers cannot be modified by *almost* (145b), but n-words can (147), so Zanuttini concludes that n-words must be universal (negative) quantifiers:

(147) **Non** ha detto *quasi* **niente**.
NM has said almost nothing
'He said almost nothing.' (Zanuttini (1991), p. 117)

NPIs, which are commonly assumed to be existential quantifiers, cannot be modified by *almost* either and this distinguishes them from n-words. For this reason, Zanuttini (1991) uses *almost* also as a test against an NPI analysis for n-words:

(148) a. *Almost* **nobody** came.
b. * I could**n't** see *almost* anything.

Zanuttini's conclusion is that NPIs and n-words are two distinct paradigms: the former are existential quantifiers, and the latter universal (negative) quantifiers, a claim that is consistent with their (in)compatibility with *almost*.

The asymmetry between n-words and NPIs with respect to *almost* carries over to Romanian, which again points at the empirical differences between the two classes:

(149) a. **Nu** am putut vedea *aproape* **nicio** casă în întuneric.
NM have could see almost no house in darkness
b. * **Nimeni nu** a putut vedea *aproape* vreo casă în întuneric.
nobody NM has could see almost any house in darkness

[19] For a comparison between these accounts and the one developed in this thesis, see Chapter 6.

Penka (2006) There has been much controversy on how reliable *almost*-modification is as a test for the status of n-words (see Richter and Sailer (1999*b*) and Giannakidou (2006)). Penka (2006) has recently argued against its validity in this respect. She proposes a unitary semantic analysis for *almost* as evaluating alternatives on an ordered Horn scale which has existential quantifiers at the bottom, and universal quantifiers at the top. This account predicts the incompatibility between *almost* and existential quantifiers in positive contexts like (145b) to the extent that existentials being at the bottom of the ordered scale, there is no lower value below them that could be evaluated as an alternative (150).

(150) Quantifier scale in positive contexts

$$\xrightarrow{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}$$

$\exists$ $\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\forall$

Penka (2006) argues that the scale is reversed in the scope of negation, such that existentials are at the top, so lower alternatives can be considered in this case (151). In her terms, this means that representing an n-word as an existential quantifier outscoped by negation does not interfere with its possibility of being modified by *almost*. The incompatibility of NPIs with *almost* in (148b) is explained in Penka (2006) by means of apparent intervention effects between two operators evaluating alternatives. *Almost* is such an operator and so is *even*. *Even* is taken to be obligatorily associated with the presence of an NPI. Thus the impossibility of *almost* to modify NPIs is determined by the intervention effects triggered by the cooccurrence of *even* and *almost*.

(151) Reversed quantifier scale in negative contexts (Penka (2006))

$$\xleftarrow{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}$$

$\exists$ $\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\forall$

There are several issues about this analysis of *almost* which taken together show that it neither contradicts the assumption that n-words are negative quantifiers, nor does it support the idea that they are existential and not universal quantifiers.

First of all, while this analysis brings *almost*-modification in accord with representing n-words as existential quantifiers, it does not exclude the other option, that n-words are universal quantifiers. In (146) the universal quantifier outscopes the negative operator. If the n-word is represented as a universal negative quantifier, it is still at the top of the scale since the scale is not reversed and Penka's analysis predicts *almost*-modification to be available.

Second, Penka's account permits an existential analysis of n-words only under the assumption that the scale is reversed under negation. This means that *almost* actually modifies the whole negative quantifier: the existential quantifier outscoped by the negative operator (see (146)). So one cannot say that it is only the existential that is modified by *almost*.

This is an issue that Penka (2007) takes into account. She argues that *almost* must always take scope over the negation, because it is a positive polarity item (PPI). She gives the following examples to illustrate the supposed incompatibility between *almost* and negative contexts:

(152) (Penka (2007, p. 213))

a. Antimorphic context:
?? I have**n't** read *almost* every book by Chomsky.

b. Anti-additive context:
?? **None** of the guests stayed *almost* until midnight.

c. Downward entailing context:
?? John **rarely** sleeps *almost* eight hours.

The claim that *almost* is a PPI is meant to explain the fact that *almost* must modify the whole negative proposition and cannot modify only the existential quantifier in the scope of negation. However, there are two problems with this claim as well. First, it seems to me that the examples in (152) are not as bad in Romanian, while slightly modified versions are perfectly fine:[20]

(154) a. Antimorphic context:

Ion **nu** a citit *aproape* **toată** cartea, mai are jumătate din ea.
John NM has read almost all book-the, still has half of it

'John didn't read almost the entire book, he still has half of it to read.'

b. Anti-additive context:

Niciun student nu a citit *aproape* **toată** cartea.
no student NM has read almost all book-the

'No student read almost the entire book. (= No student is close to finishing the book.)'

c. Downward entailing context:

Puţini studenţi au citit *aproape* **toată** cartea.
few students have read almost all book-the

'Few students read almost the entire book. (= Few students are close to finishing the book.)'

Second, the idea that *almost* is a PPI cannot explain its incompatibility with the negated universal quantifier *nu toţi* 'not all' in (155):

(155) (**Aproape*) **Nu toţi** studenţii au adus cărţi.
almost not all students have brought books

'(*Almost) Not all the students brought books.'

I think that this can be explained in Penka's (2006) analysis of *almost*, if we reformulate the scale in (151) as the one in (156) with negation and quantifiers. This scale confirms the predictions of the analysis: $\neg\forall$ is at the bottom, so there are no alternatives available and *almost* is ungrammatical in (155). $\neg\exists$ is at the top, so alternatives are available and *almost* is grammatical.

(156) Quantifier scale in negative contexts

$\longleftarrow$

$\neg\exists$ $\qquad\qquad\qquad\qquad$ $\neg\forall$

In conclusion, assuming that *almost* evaluates alternatives, it can modify either a negative quantifier (the top in (156)) or a universal quantifier (with or without negation in its scope: the top in (150)), but not the existential quantifier alone (the bottom in (150)) or the universal outscoped by negation (the bottom in (156)). So the fact that n-words can be modified by *almost* is only compatible with this analysis if n-words are negative quantifiers.

Further research is needed to determine whether *almost* is a PPI at all as Penka suggests and whether this could be the case in some languages and not in others. (154) suggests that it is not a PPI

[20]Note that the continuation in (154a) indicates that this is not an instance of metalinguistic negation like the one below given in Penka (2007, p. 213):

(153) I haven't read ALMOST every book by Chomsky – I have read every single one.

in Romanian. If *almost* can be conclusively argued to be a PPI, this test is at best irrelevant for the status of n-words. As we will see below, the arguments for the negative quantifier status of Romanian n-words are independent of *almost*-modification.

3.3.5 Conclusion for the choice of the analysis

In the last three sections, various empirical and theoretical arguments have been brought mainly against the NPI hypothesis, and partly in favor of an NQ analysis for Romanian n-words. It has been shown that, with respect to n-words, the NM fails to play the role that is expected of a typical NPI licenser: it does not need to c-command them in the syntax and it does not act like a semantic licenser for them. However, the NM is a syntactic licenser for n-words to the extent that it regulates their scope possibilities, which resemble the ones of uncontroversial quantifiers, and those of negative quantifiers in English. This indicates that n-words also have quantificational force, besides the negative semantics shown by their anti-additive properties.

The contrast between the behavior of n-words and that of NPIs with respect to the NM, locality conditions, and *almost*-modification make the NPI analysis untenable for Romanian n-words. In the next two sections, further arguments will be brought in support of the negative contribution of n-words and their behavior as negative quantifiers.

3.4 The negative status of n-words and double negation

Having shown that n-words do behave like quantifiers, at least with respect to locality conditions on scope, in this section I present arguments in favor of their negative content. In Section 3.4.1 I discuss empirical contexts where n-words express negation on their own and in Section 3.4.2, I argue for their negative semantics on the basis of the observation that two co-occurring n-words can yield double negation readings.

3.4.1 Negative contribution in non-NC contexts

Fragmentary answers Although the typical context where n-words show up is that of NC with a NM, in some constructions they can appear alone and express negation. Fragmentary answers are such a case: in (157a), the n-word *nimic* 'nothing' has a negative interpretation:

(157) What did he buy?

a. **Nimic**.
'Nothing.'

b. * *Anything*.

Fragmentary answers have also been used as an argument against the NPI analysis, since an NPI like the English *anything* is excluded in such a context (157b).

NPI analyses reject the idea that n-words contribute negation in fragmentary answers. Giannakidou (1998, 2000, 2006) argues that these contexts are elliptical, and negation is actually contributed by the NM in the elided material indicated by the strikethrough in (158b):

(158) a. Ce a cumpărat?
what has bought

b. [~~**Nu** a cumpărat~~] **nimic**.
NM has bought nothing

First of all, such an explanation does not provide an answer as to why the NPI *anything* is not grammatical in the same context. If the negation in (158b) is contributed by the negative marker, and the n-word is an NPI, (159) should also be grammatical as an answer to (158a):

(159) * [~~He did**n't** buy~~] *anything*.

Second, on the basis of Merchant's (2001) analysis of ellipsis, Watanabe (2004) argues that the negative interpretation of constructions like (157a) and (158b) can only come from the n-word. Merchant shows that ellipsis resolution presupposes *semantic identity* between the elided material and its antecedent.

Consider the following question-answer pair:

(160) a. Ce a cumpărat?
what has bought

b. O carte.
a book

i. # [~~**Nu** a cumpărat~~] o carte.
NM has bought a book

ii. [~~A cumpărat~~] o carte.
has bought a book

There are two possible constructions for which (160b) can stand, containing negative (160b-i) or positive (160b-ii) elided material. However, only the positive one is available in response to the question in (160a), because only this one is semantically equivalent to the positive antecedent *a cumpărat* provided by the question. If the question provides a negative antecedent (*nu a cumpărat*), the negative material is interpreted as having been elided (161):

(161) a. Ce **nu** a cumpărat?
what NM has bought

b. O carte.
a book

i. [~~**Nu** a cumpărat~~] o carte.
NM has bought a book

ii. # [~~A cumpărat~~] o carte.
has bought a book

In view of the semantic identity between the elided material and the antecedent, it is obvious now that in (158), the n-word *nimic* is the one contributing negation. The question in (158a) provides the positive antecedent *a cumpărat*, which is semantically identical to the elided part *nu a cumpărat*. Thus the negative marker in (158b) does not contribute semantic negation, unlike in (160b-i), where the negation it carries makes the elided material incompatible with the positive antecedent. Notice that the difference is made by the n-word: it is only its presence that prevents the NM from contributing negation in (158b).[21]

As a confirmation that the n-word is indeed the negative component in (158b), consider also the negative question with an n-word elliptical answer below:

[21]In Section 5.5, I will argue that the NM is a syntactic licenser in NC and does not contribute its negation independently of the n-word(s). This is in accord with the observation above about ellipsis.

(162) a. Ce **nu** a cumpărat?
what NM has bought

b. **Nimic**. (A cumpărat totul.)
nothing (has bought everything)

i. There is nothing he didn't buy. (He bought everything.) (DN)

ii. # He didn't buy anything. (#NC)

The question above provides a negative antecedent for the elided material in the answer, so what is elided is negative. Since the n-word is also negative, the answer in (162b) can only be interpreted with two negations leading to a DN reading.

Thus we may conclude with Watanabe (2004) that n-words in fragmentary answers are negative. This holds at least for the Romanian data discussed above.

There are other contexts where n-words appear without a NM and receive a negative interpretation. In what follows I exemplify gapping, comparative, and past participial constructions.

Gapping constructions Bîlbîie (2008) points out that n-words contribute negation in gapping constructions where they establish a contrast with the affirmative verb and a PPI like *cam* 'pretty', *tot* 'still', *mai* 'still' or/ and with another constituent (163c) in the complete clause.[22]

(163) a. Maria *tot mai* citeşte, dar Ion (**niciodată**) **nimic**.
Maria still still reads, but John never nothing
'Maria still reads, but John never does.'

b. Maria *cam* exagerează, dar Ion **niciodată**.
Maria pretty exaggerates, but John never
'Maria pretty much exaggerates, but John never does.'

c. Maria *mai* citeşte *cîte o carte*, dar Ion **nimic**/ **niciuna**.
Maria still reads each a book, but John nothing/ none
'Maria still reads a book from time to time, but John doesn't read anything/ any.'

As there is no negation in these constructions apart from that contributed by the n-words, there is no way to argue that the negative meaning of the second conjunct in (163a) – (163c) comes from some source other than the n-word.

Comparative constructions Another context where n-words contribute negation independently of the NM is that of comparative constructions and disjunctive *ori ... ori* 'either ... or' structures like in (164a) and (164b), respectively:

(164) a. Ion e înalt ca **nimeni** altul de la el din clasă.
John is tall like nobody else from him from class
'John is taller than everybody else in his class. (Nobody in John's class is as tall as he is.)'

b. Mă duc ori la mare, ori **nicăieri** (altundeva).
me go or to seaside or nowhere (else)
'I'll either go to the seaside or nowhere.

[22]Farkas (2002) shows that *cîte* in contexts like (163c) is a dependent indefinite with a co-varying interpretation. It can be translated as 'each' with a co-varying interpretation (e.g. The boys received one book each).

Some n-words are often used in relatively idiomatic expressions, where they also contribute negation: the adverb *niciodată* ('never') with the comparative, and *nimeni* ('nobody') within a possessive construction:

(165) a. Ca **niciodată**, Ion a vorbit foarte mult cu ceilalti invitati.
like never, John has chatted very much with the other guests
'Uncharacteristically, John chatted a lot with the other guests.'

b. Noi sîntem ai **nimănui**.
we are of nobody
'We belong to nobody.'

Past participial constructions An even more straightforward context that indicates the negative contribution of an n-word is that of past participial constructions. An n-word preceding the affirmative verb form makes the whole construction negative:

(166) Acest articol, [**niciodată**/ **de nimeni**] citat, a rămas uitat.
this article never/ by nobody cited has remained forgotten
'This article, which has never been cited/ which hasn't been cited by anybody, has been forgotten.'

A NC construction with the preposed n-word and the negative marker on the participle is excluded. If the NM appears on the participle, the only possibility to interpret the construction is double negation. This indicates that both the n-word and the NM on the verb contribute negation in this context:

(167) Acest articol, [**niciodată**/ ?**de nimeni**] **ne**citat, a devenit foarte cunoscut.
this article never/ by nobody un-cited has become very well-known
'This article, which is always cited/ which is cited by everybody, has become very well-known.' (DN/ #NC)

3.4.2 Double negation and denial

The previous section provided arguments for the negative semantics of n-words on the basis of their ability to yield negation in the absence of a NM. Here, we focus on DN readings with n-words.

Although Romanian is a NC language, there are particular contexts were a DN reading can be obtained. So far we have seen that this is possible in those contexts where an n-word contributes negation on its own, as in question-answer pairs (162) and past participial constructions (167). In Fălăuş (2007), DN readings are shown to occur in Romanian finite sentences as well. Thus sentence (168) allows both a NC and a DN reading, while (169) favors a DN reading, since pragmatic reasons exclude a NC interpretation in which humans are immortal:

(168) **Nimeni** nu vine de **nicăieri**.
nobody NM comes from nowhere

a. Nobody comes from anywhere. (NC)
b. Nobody comes from nowhere. (Everybody comes from somewhere.) (DN)

(169) **Nimeni** nu moare **niciodată**.
nobody NM dies never

a. # Nobody ever dies. (#NC)
b. Nobody never dies. (Everybody dies one day.) (DN)

Denial Abstracting away from pragmatic considerations, DN usually occurs in a finite sentence if this is interpreted as the *denial* of a negative statement already provided by the context. The term *denial* comes from Van der Sandt (1991) and Geurts (1998) who use it similarly to the terms *radical negation* (Seuren (1988)), and *metalinguistic negation* (Horn (1985, 1989)). Here it will stand for *the role played by negation in a well-defined discourse that presupposes two distinct consecutive sequences of which the second one is negative and objects to a statement made in the first one*.[23] The data in (170) are a case of denial: the affirmative statement in (170a) is denied by the negative one in (170b):

(170) a. Speaker A: The cook killed her.
b. Speaker B: The cook did **NOT** kill her. (He has an alibi.)

All the references above note that denial is intonationally marked, which I will indicate by means of capital letters.

If the statement made by Speaker A is negative, Speaker B can employ an n-word to deny it, and thus DN occurs:

(171) a. Speaker A: Aceşti oameni **nu** iubesc pe **nimeni**, nici măcar pe ei înşişi.
these people NM love PE nobody not even PE them themselves
'These people don't love anybody, not even themselves.'
b. Speaker B: **NIMENI nu** iubeşte pe **nimeni**. (Toată lumea iubeşte pe cineva.)
nobody NM loves PE nobody all people love PE somebody
i. 'Nobody loves nobody. (Everybody loves somebody.)' (DN)
ii. # 'Nobody loves anybody.' (#NC)

Unlike (162) and (167), (171b) is crucially a full finite sentence with a NM: it is neither a short answer without a verb, nor a past participial construction. So the n-word brings its negative contribution although it would be expected to build NC together with the NM and the other n-word. The DN reading in (171b) is a clear confirmation of the negative meaning of the n-word.

N-words, DN and the NM A denial context like (171b) only yields a DN reading if two n-words are involved. That is, the sentence that is denied must already contain an n-word (171a). If it doesn't, the DN effect does not obtain between an n-word and the NM:

(172) a. Speaker A: Aceşti oameni **nu**-l plac pe Ion.
these people NM-CL like PE John
'These people don't like John.'
b. Speaker B: **NIMENI nu**-l place pe Ion. (# Toată lumea îl place pe Ion.)
nobody NM likes PE John all people CL like PE John
i. # 'Nobody doesn't like John. (Everybody likes John.)' (#DN)
ii. 'Nobody likes John.' (NC)

In (172) the first utterance provides a negative statement, but the n-word *nimeni* in the second one does not yield a DN reading: see the unnaturalness of the continuation with 'Everybody likes John' in (172b). By comparison to (171), this means that an n-word and a NM that are clausemates cannot contribute their negations independently of one another, but only in a concord reading.

[23] In Van der Sandt (1991), affirmative sentences used to contradict a previous negative statement are also instances of denial. Although we concentrate our attention on negative sentences, such an example is given in (178).

The difference between (171) and (172) has implications both for n-words and for the NM. For n-words, it supports the claim that they are negative quantifiers to the extent that two of them can yield a DN interpretation. The fact that the same does not hold of a single n-word with the NM indicates that the negation of the NM must always concord with the negation introduced by the n-word. This means that in (171b), where DN arises, the NM only (syntactically) licenses the presence of the n-words[24], but its semantic negation does not play any role with respect to interpretation. For this reason, in the rest of this chapter I will focus on n-words as NQs. The NM will be addressed in Section 5.5.

3.5 Scope properties of n-words as negative quantifiers

We provided arguments for the empirical and theoretical inadequacy of an NPI-analysis, and for the quantificational behavior and the negative content of Romanian n-words. Since n-words are able to express negation on their own (Section 3.4.1), and to yield DN (Section 3.4.2), the theoretical premise here is that they are negative quantifiers.

If Romanian n-words are NQs, the NC reading of two n-words remains a dilemma. The aim of this section is to further investigate the way n-words behave as NQs, in order to identify those specific properties that may lead us to an appropriate analysis of NC. I will examine the scope properties of n-words in NC constructions.

After some general considerations on the scope interaction between non-negative and negative quantifiers (Sections 3.5.1 and 3.5.2), I will focus on the scope conditions under which NC readings occur when NQs interact with non-negative quantifiers (Section 3.5.3). In Section 3.5.4 I investigate the scope conditions on the DN reading and in Section 3.5.5 I show that some complex quantificational constructions discussed in Section 2.1 display similar properties to those of NC when they interact with external quantifiers. Since such quantificational complexes have been successfully accounted for as inherently polyadic quantifiers, this similarity will be taken as supportive evidence for a treatment of NC as a polyadic quantifier.

3.5.1 General considerations

An objective investigation of the scope properties of n-words as NQs with respect to other quantifiers must rely on data that do not involve existential or universal quantifiers. These quantifiers display special scope interaction with negative quantifiers: universal quantifiers usually take narrow scope (173a), while existentials take wide scope (173b):

(173) a. **Niciun** student nu a citit **fiecare** carte.
no student NM has read every book
 i. # $\forall >$ NO: 'For each book it is the case that no student read it.'
 ii. NO $> \forall$: 'No student read every book.'

b. **Niciun** student nu a citit **o** carte.
no student NM has read a book
 i. $\exists >$ NO: 'There is a book such that no student read it.'
 ii. # NO $> \exists$: 'No student read any book.'

The linear order of quantifiers in (173b-ii) may be available, but with two readings that are different from the typical existential quantifier reading. In one the indefinite determiner is interpreted as a

[24]In view of the general observations in section Section 3.2.

minimizer. In this case, special intonation is required for *o carte* and the translation of (173b) would be 'None of the students read one single book'. The second possible reading is one where each of the students read any n number of books, except for $n = 1$. In this case, the indefinite determiner is understood as the cardinal quantifier *one* which in Romanian is homophonous with the indefinite determiner. These two readings are special and would not be represented like in (173b-ii), where a plain existential quantifier like *some* is intended.

Reversing the linear order of the negative and the universal/ existential quantifiers slightly modifies the availability of the disfavored readings, but the general picture remains the same. The wide scope reading of the universal over the negative quantifier is not completely excluded, but it is highly marked (174a-i). The two auxiliary interpretations available when the existential takes narrow scope with respect to negation are slightly harder to obtain in (174a-ii), but still possible. The typical existential reading is again excluded.

(174) a. **Fiecare** student nu a citit **nicio** carte.
every student NM has read no book

i. ?? $\forall >$ NO: 'For each student it is the case that he read no book.'

ii. NO $> \forall$: 'No book was read by every student.'

b. **Un** student nu a citit **nicio** carte.
a student NM has read no book

i. $\exists >$ NO: 'There is a student such that he read no book.'

ii. # NO $> \exists$: 'No book was read by any student.'

The unavailability of wide scope for universal quantifiers and narrow scope for existentials with respect to negative quantifiers can be explained by the competition between the constructions in (173a-i), (174a-i), (173b-ii), and (174b-ii) and one in which another n-word replaces the universal/ existential quantifier. The sentence in (175) expresses the reading that the four constructions above fail to convey:

(175) **Niciun** student nu a citit **nicio** carte.
no student NM has read no book
'No student read any book.'/ 'No book was read by any student.'

Note that the interpretation of (175) is truth-conditionally equivalent to the unavailable interpretations in (173) and (174), if we take into account the three-way logical equivalence[25] between a negative quantifier, an existential quantifier outscoped by negation and a universal quantifier outscoping negation. This equivalence is formulated below:

(176) Logical representations of a negative statement:

a.	NO$x[P(x) \wedge Q(x)]$	Generalized negative quantifier
b.	$\neg\exists x[P(x) \wedge Q(x)]$	Existential quantifier
c.	$\forall x[P(x) \rightarrow \neg Q(x)]$	Universal quantifier

The choice between the logical representations in (176) corresponds to the claim that n-words are existential ((176b) in Giannakidou (2006), Zeijlstra (2004), Penka (2007)) or universal ((176c) in Giannakidou (1998)) negative polarity items.[26] Since in this thesis I treat n-words as negative

[25]Giannakidou (1998, 2006) makes extensive use of this logical equivalence to explain the crosslinguistic ambiguity of n-words (see Section 6.1.1).

[26]A variant of the term 'existential NPI' is that of 'Heimian indefinite', after Ladusaw (1992), which suggests that the n-word is a free variable bound by existential closure. This is the terminology that Zeijlstra (2004) and Penka (2007) use.

quantifiers (and thus not NPIs), I will only make use of the representation in (176a), where NO stands for the *generalized negative quantifier* given in DEFINITION 2.2c, p. 12.

3.5.2 Two quantifiers

To investigate the scope properties of NQs in interaction with non-negative quantifiers, I restrict the discussion to MANY and FREQUENTLY, for which no special behavior has been noted in negative contexts. To distinguish the characteristic properties of NQs in NC, I will compare them with cardinal quantifiers which I likewise consider in their interaction with MANY and FREQUENTLY.

To my knowledge, quantifier scope in Romanian has not been studied in detail yet. In this section I will use strictly parallel constructions to compare NQs with cardinal quantifiers. This way any differences between the two classes of quantifiers must be only due to their scope properties. I thus keep away from any debate on general quantifier scope behavior in Romanian.

Although Romanian quantifiers exhibit relatively free scope interactions, preference is usually given to linear order.[27] Thus for (177) speakers first obtain the reading in (177a), which is the linear order of the quantifiers. The reading in (177b), although available, requires a context:[28,29]

(177) **Doi** studenţi au citit **multe** cărţi.
two students have read many books
'Two students read many books.'

a. 2 $>$ MANY: 'Two students are such that they each read many books.'

b. MANY $>$ 2: 'There are many books such that for each of them it is the case that there are (at least) two students who read it.'

An appropriate context for the interpretation in (177b) is the following:

(178) a. Speaker A: Probabil că sînt **puţine** cărţi pe care să le fi citit (măcar) **doi** studenţi.
probably that are few books PE which SJ them be-PF read at least two students
'There are probably few books which have been read by at least two students.'

b. Speaker A: **Doi** studenţi au citit **MULte** cărţi.
two students have read many books
MANY $>$ 2: 'There are (actually) many books which have been read by (at least) two students.'

In Van der Sandt's (1991) broad understanding of denial, (178b) is an (affirmative) denial of the statement made by (178a). Thus the quantifier MANY 'denies' FEW, and the intonational emphasis is used to indicate this. In this presentation it will usually be the case that the inverse scope reading requires an emphasis on the lower quantifier and possibly also a contrastive context similar to (178).[30]

[27] The linear order preference may be due to the already indicated free word order character of Romanian (cf. Section 2.2). The speaker's choice of a particular linear order usually also indicates his/ her choice with respect to quantifier scope.

[28] We will leave aside possible cumulative readings for the moment, since they will be addressed later.

[29] The symbols "??", "?" mark the degree of (un)grammaticality of a sentence, or the (un)availability of an interpretation for a given sentence: "??" stands for "rather unacceptable, but not excluded", "?" for "pretty acceptable in an appropriate context". For an ungrammatical sentence we use "*", and for a totally unavailable reading "#".

[30] In Section 5.4.2 I will associate this emphasis in denial/ contrastive contexts with 'contrastive focus'.

The linear scope of quantifiers becomes more important in scope interactions between NQs and non-negative quantifiers. In (179) and (180) below, the inverse scope readings in (b.) are less available than the ones in (a.), even if an appropriate context is provided:

(179) **Niciun** student n-a citit **multe** cărţi.
no student NM-has read many books
'No student read many books.'
a. NO > MANY: 'No student is such that s/he read many books.'
b. ? MANY > NO: 'There are many books such that no student read them.'

(180) **Mulţi** studenţi n-au citit **nicio** carte.
many students NM-have read no book
'Many students read no book.'
a. MANY > NO:' Many students are such that they didn't read any book.'
b. ?? NO > MANY: 'For no book is it the case that many students read it.'

3.5.3 Two NQs and a non-negative quantifier

I now consider the scope interaction between two negative quantifiers and an intervening non-negative quantifier (MANY and FREQUENTLY), since they make visible the properties of n-words as NQs, and the particularities of the NC interpretation. This brings us closer to an explanation for the nature of NC as a semantic effect in the interpretation of two NQs. As we will see in Section 3.5.5, negative quantifiers are not unique in creating such readings: previously discussed polyadic quantifiers (Section 2.1.3) exhibit similar properties.

3.5.3.1 Scope interaction with MANY

Consider the scope interaction between two NQs in their NC reading and the quantifier MANY in (181) below:

(181) **Niciun** scriitor n-a recomandat **multor** studenţi **nicio** carte.
no writer NM-has recommended many-Dat students no book
a. ? NO (writer) > MANY > NO (book): 'No writer recommended books to many students.'
b. NO (writer) – NO (book) > MANY: 'There is no writer and no book such that the writer recommended the book to many students.'
c. MANY > NO (writer) – NO (book): 'Many students are such that they weren't recommended any book by any writer.'

Since NC is the most natural reading for a sentence with two n-words in Romanian, negation is logically expressed only once, and the scope interaction between the two n-words is irrelevant. Thus the sentence in (181) accepts three different readings, given in (a), (b), and (c). Contrary to the expectations based on the linear order, the scope order in (181a) is not the most natural one. This is due to the fact that the intervention of a non-negative quantifier between two negative quantifiers forces both negative quantifiers to contribute their negation, and the resulting interpretation is DN, as we will see in Section 3.5.4. Here I only take into account the NC reading, since the denial context is not provided for DN to be possible.[31]

The preferred scope is (181b), where both n-words take scope together over MANY. It says that there is no (writer, student) pair, such that there have been many book recommendations from the former towards the latter. Apparently, given the n-word *niciun* as the linearly first quantifier in the sentence, the other n-word takes scope over the preceding MANY more easily than in a construction where there is no other n-word: see the asymmetry between (180b) and (181b). Note, though, that the contrast is not so sharp as it may appear by directly comparing the two sentences. The fact that the linearly first quantifier is a subject in (180b) makes it more difficult for the direct object negative quantifier to take wide scope. In (182) the direct object negative quantifier can more easily outscope a preceding indirect object. This is the reading in (182b):

(182) [**Multor**/ La **mulţi**] studenţi n-am recomandat **nicio** carte.
many-Dat/ to many students NM-have recommended no book
'I recommended no book to many students.'

a. MANY > NO: 'There are many students to whom I didn't recommend any book.'

b. ? NO > MANY: 'There is no book such that I recommended it to many students.'

(181b) and (182b) are similar with respect to the syntactic role of the two quantifiers (carried by *many students* and *no book*), but we can still notice that it is easier for the negative quantifier to take wide scope over preceding MANY if another negative quantifier precedes MANY.

Similarly, in (181c) the first negative quantifier in linear order takes narrow scope with respect to MANY due to the presence of another negative quantifier that follows MANY. Compare the availability of (181c) with that of (179b). The syntactic position (*many* in (179) as a direct object vs. an indirect object in (181)) does not make a difference, since the same scope behavior can be found in (183), where *many* is an indirect object:

(183) **Niciun** scriitor n-a recomandat **multor** studenţi "Nostalgia".
no writer NM-has recommended many students "Nostalgia"
'No writer recommended the book "Nostalgia" to many students.'

a. NO > MANY: 'No writer is such that he recommended "Nostalgia" to many students.'

b. ? MANY > NO: 'Many students are such that they weren't recommened "Nostalgia" by any writer.'

The readings in (181b) and (181c) are more natural than those in (182b) and (183b), which indicates that the preference for the two n-words to be interpreted as scope-adjacent is stronger than the linear order of the quantifiers.

This conclusion is further supported by the observation that two non-negative quantifiers instead of the negative ones in (181) would make the linear order scope interpretation most natural, as (184) indicates:

(184) **Doi** scriitori au recomandat **multor** studenţi **trei** cărţi.
two writers have recommended many-Dat students three books

[31] This scope order is not excluded with a NC reading, because speakers tend to interpret it cumulatively (see the examples with cumulative readings in Section 2.1.3.2). MANY is a nominal quantifier that expresses cardinality, NQs can also be interpreted as expressing the cardinality 0 of a set intersection, and thus the most salient interpretation of (181a) is that there are zero writers who recommended books to many students, and there are zero books that were recommended to many students by writers.

a. $2 > \text{MANY} > 3$: 'Two writers have each recommened to each of many students (at least) three books.'

b. ?? $2 > 3 > \text{MANY}$: 'Two writers have each recommended each of (at least) three books to many students.'

c. ? $\text{MANY} > 2 > 3$: 'For each of many students there are (at least) two writers such that each of the writers recommended (at least) three books to the student.'

In (184), we leave aside the inverse scope readings between the two cardinal quantifiers 2 and 3, as we did for NQs in (181). The scope order in (184b) is only available with a cumulative reading (see Section 3.5.5 below).

I conclude that a NC interpretation requires the scope-adjacency of the negative quantifiers.

3.5.3.2 Scope interaction with FREQUENTLY

This conclusion is further supported by the even stronger effects that can be observed when two NQs interact with an adverbial quantifier like FREQUENTLY. In this case, the linear scope interpretation in (185a) is less acceptable than in (181):[32]

(185) **Niciun** student nu a citit **frecvent** **nicio** carte.
no student NM has read frequently no book

a. ?? NO (student) $>$ FREQUENTLY $>$ NO (book): 'None of the students were frequent book-readers.'

b. NO (student) $-$ NO (book) $>$ FREQUENTLY: 'There is no student and no book such that the student read the book frequently.'

c. ? FREQUENTLY $>$ NO (student) $-$ NO (book): 'It was frequently the case that no student read any book.'

Like in the case of MANY (184), if we replace the two NQs with non-negative quantifiers, we obtain opposite scope tendencies. The linear scope reading is the most natural (186a); the other two readings are less available:

(186) **Doi** studenţi au recitat **frecvent** **trei** poezii.
two students have recited frequently three poems
'Two students frequently recited three poems.'

a. $2 >$ FREQUENTLY > 3: 'For two students it was frequently the case that they each recited (at least) three poems.'

b. ?? $2 > 3 >$ FREQUENTLY: 'Two students each recited each of (at least) three poems frequently.'

c. ? FREQUENTLY $> 2 > 3$: 'It was frequently the case that there were (at least) two students such that each of them recited (at least) three poems.'

Similarly to (184b), the lower cardinal quantifier 3 can outscope FREQUENTLY only if it forms a cumulative quantifier with 2 (see also Section 3.5.5).

[32]This is most likely due to the fact that a cumulative reading is harder to obtain between an adverbial and a nominal quantifier: cf. footnote 31.

3.5.4 DN readings with an intervening quantifier

The scope interaction between NQs and MANY/ FREQUENTLY in (181) and (185) indicates that whatever semantic mechanism we choose to derive the NC reading in Romanian, it should take into account the scope-adjacency condition on NQs.

Moreover, a DN reading is only available for the quantifier scope in which the non-negative quantifier intervenes between the two negative ones. DN can be obtained for (181) only in the scope order in (181a), and for (185), only in (185a). In (187) and (188) below, the two sentences are integrated in a context that favors denial, and thus yield a DN reading:[33]

(187) a. Speaker A: Am înţeles că Mircea Cărtărescu n-a recomandat
have understood that Mircea Cărtărescu NM-has recommended
MULtor studenţi **nicio** carte.
many students no book

i. *MANY > NO: 'I've heard that there are many students to whom M. C. didn't recommend any book.'*

ii. # NO > MANY: 'I've heard that there is no book such that M.C. recommended it to many students.'

b. Speaker B: **NICIun** scriiTOR n-a recomandat MUL**tor** studenţi **nicio** carte.
no writer NM-has recommended many students no book

NO (writer) > MANY > NO (book): 'No writer is such that there are many students to whom s/he didn't recommend any book.'
('If there is a writer such that there are students to whom s/he didn't recommend any book, then there were only a few (not many) such students.')

In a context where somebody utters (187a) with the interpretation in (187a-i), another person can deny this statement by (187b). That is, if Speaker A complains that there are many students to whom Mircea Cărtărescu didn't recommend any book, Speaker B, having more knowledge about the book recommendations, objects to that and says that for none of the writers were there many students to whom s/he didn't recommend any book: if there were students such that a writer didn't recommend any book to them, then there must have been only a few (not many) such students. The interpretation in (187b) is an instance of both n-words contributing their negative quantifier. The situation is similar in (188b), where the only difference is that MANY is replaced by FREQUENTLY:

(188) a. Speaker A: Am înţeles că Ion n-a recitat **frecVENT nicio** poezie.
have understood that John NM-has recited frequently no poem

i. FREQUENTLY > NO*: 'I've heard that it was frequently the case that John didn't recite any poem.'*

ii. # NO > FREQUENTLY: 'I've heard that there is no poem such that John recited it frequently.'

b. Speaker B: **NICIun** stuDENT nu a recitat **frec**VENT **nicio** poezie.
no student NM has recited frequently no poem

NO (student) > FREQUENTLY > NO (poem): 'No student is such that s/he frequently didn't recite any poem.'
('If there was a student who happened to not recite any poem, then this happened seldom (not frequently).'

[33]Small capitals indicate an intonational emphasis on the quantifier which is repeated from the previous statement. The new emphasis contributed by the sentence is marked with large capital letters.

In conclusion, the NC interpretation is idiosyncratic regarding the linear order between the (negative) quantifier components: they have to be immediately adjacent to one another. If this adjacency condition is not met, the availability of NC is remarkably reduced, since this reading then competes with a scope interaction between the negative quantifiers, which yields DN.

3.5.5 Scope properties of cumulative quantifiers

In this section I show that, like NQs in NC, Romanian cardinal quantifiers also display scope idiosyncracy when they interact with external quantifiers in their cumulative reading. Although I focus on cumulative readings here, the same properties can easily be shown to also hold of *different*/ *same* and resumptive quantifiers, which were discussed in Section 2.1.3 as instances of inherently polyadic quantifiers.

Depending on the scope interaction between the two cardinal quantifiers, the sentence in (189) can receive any of the following three interpretations:

(189) **Patruzeci** de colaboratori au scris **treizeci şi două** de articole pentru volum.
forty of contributors have written thirty and two of articles for volume
'Forty contributors wrote thirty-two articles for the volume.'

a. $40 > 32$: 'Forty contributors wrote each thirty-two articles.'

b. $32 > 40$: 'Thirty-two articles were each written by forty contributors.'

c. $40 - 32$: 'There is a total of forty contributors who wrote and a number of thirty-two articles that were written for the volume.'

The first reading is the one in which 40 has wide scope over 32, so there is a total of twelve hundred eighty articles. In the second one, 32 takes scope over 40, and there are twelve hundred eighty contributors. But the most natural interpretation is the one in (189c), in which neither of the two cardinal quantifiers takes scope over the other, and 40 and 32 specify the total number of contributors and articles, respectively, such that the former wrote the latter for the volume. As already discussed in Section 2.1.3.2, this interpretation is known as the 'cumulative' reading, and it only occurs in constructions with at least two cardinal quantifiers. For our discussion, the cumulative reading is special as for (189c) to be available, the two quantifiers must have a different scope behavior from that in (189a) and (189b) (see also Section 2.1.3.2): they are scope neutral with respect to each other. I use the notation ($40 - 32$), to indicate that there is no scope interaction between the two cardinal quantifiers.

In constructions where two quantifiers are cumulatively interpreted, if another quantifier intervenes, the cumulative reading is lost. This is shown by the example below:

(190) **Patruzeci** de colaboratori au scris **frecvent** **treizeci şi două** de articole pentru
forty of contributors have written frequently thirty and two of articles for
volum.
volume
'Forty contributors frequently wrote thirty-two articles for the volume.'

a. $40 >$FREQUENTLY> 32: 'For forty contributors it was frequently the case that they wrote thirty-two articles for the volume.'

b. FREQUENTLY$> 40 - 32$: 'It was frequently the case that a total of forty contributors wrote thirty-two articles for the volume.'

In (190), the quantifier FREQUENTLY intervenes in linear order between 40 and 32, which in (189c) were part of the polyadic quantifier $(40, 32)$. If we interpret the sentence with the scope order in which adjacency between 40 and 32 is not maintained, the only possible reading is one in which 40 takes scope over FREQUENTLY, and the latter takes scope over 32 (190a). In this case, for every one of the forty contributors it was frequently the case that s/he wrote thirty-two articles for the volume. The number of articles that were written for the volume is a multiple of 1280.

The cumulative reading in (189c) can only be obtained in (190) if 40 and 32 are scope-adjacent. The most natural order is the one in (190b), where FREQUENTLY outscopes everything else. This allows 40 and 32 to build the polyadic quantifier $(40 - 32)$, over which FREQUENTLY takes scope. The reading is: it happened frequently that there was a total of forty contributors who wrote a total of thirty-two articles. Theoretically, the scope order $40 - 32 >$FREQUENTLY is also possible, but the interpretation is pragmatically strange, since it means that there are forty contributors and thirty-two articles, such that the former wrote the latter frequently. It is somewhat unnatural to think of somebody writing the same thing frequently, unless one thinks of "writing" as "rewriting".

The cumulative reading of cardinal quantifiers resembles NC to the extent that they both build a quantificational complex with idiosyncratic scope properties: the monadic quantifiers in cumulative readings do not scopally interact with each other, just like negative quantifiers in NC. In particular, the data on the scope intervention of FREQUENTLY indicate the similarity between the cumulative reading and NC with respect to the examples in (190b)/ (190a), and the ones in (185)/ (188). On the one hand, the cumulative interpretation is possible once two cardinal quantifiers are scope-adjacent (190b); the NC reading of two NQs is most natural under the same circumstances, as (185b) and (185c) show. On the other hand, the intervention of FREQUENTLY between the two cardinal quantifiers imposes a scopal interpretation on them (cf. (190a)). Such an intervention between two n-words derives a DN reading in (188), i.e. the scopal interpretation of the two negative quantifiers.

The difference between NC and cumulative readings concerns their occurrence frequency in comparison to that of the corresponding scopal reading. On the one hand, NC is the default interpretation of two NQs in a NC language like Romanian, so it usually wins the competition with the DN reading. For DN special contextual conditions are necessary. On the other hand, the cumulative and the scopal reading of two numeral quantifiers freely occur in parallel. There is only a slight preference to associate a cumulative interpretation with quantifiers that express a large cardinality ((189) and (190)), and a scopal interpretation with quantifiers of a small cardinality (186).

Thus NC functions more like a general principle for the scope interpretation of two negative quantifiers, which is not the case for the cumulative reading of cardinal quantifiers. This contrast, however, is a matter of language use, and does not contradict the scope similarity attested here between the two quantificational complexes.

3.6 Conclusion

To summarize this chapter, we have reached three important results concerning the semantic status of Romanian n-words: 1) the inadequacy of the NPI analysis to account for their semantic properties, 2) the negative content and the quantificational properties which indicate their negative quantifier status, and 3) their particular scope properties in NC, which resemble those of inherently polyadic quantifiers.

First, I showed that the NPI assumption is not motivated for Romanian n-words for several reasons concerning the empirical differences between NPIs and n-words. Most importantly, unlike NPIs, n-words do not need a semantic licenser: their negative content, indicated by their anti-additive property, is apparent in the absence of the NM as well. The locality conditions between n-words and the NM suggested that the latter marks the scope of the negative quantifier carried by the n-word. This will be made explicit in Section 5.5. Moreover, locality tests indicated that n-words have scope properties similar to those of typical quantifiers in Romanian and NQs in DN languages like English.

Besides the evidence for their quantificational properties, I argued that n-words have a negative semantic contribution, attested by their negative interpretation when they precede the past participle, in fragmentary answers, gapping and comparative constructions, where the NM is absent. Contrary to the claims made by the NPI approaches, I showed that analyzing fragmentary answers as elliptical supports the idea that n-words contribute negation alone. Denial contexts provide a further argument for the negative semantics of n-words, since they create the pragmatic conditions for a DN interpretation of two n-words.

Finally, I discussed the scope properties of n-words in NC. The scope interaction with non-negative quantifiers showed that n-words in NC interpretations must be scope-adjacent, so they do not permit the intervention of another quantifier. If a quantifier does intervene, the NC reading is seriously degraded due to the competition with a DN reading. This means that an intervening quantifier creates the right conditions for the scopal/ DN interpretation of the two n-words. The same scopal behavior was shown to characterize cumulative readings of cardinal quantifiers which in Section 2.1.3 were argued to belong to the class of inherently polyadic quantifiers, together with *different/ same* and resumptive quantifiers. In particular, I showed that the (scope-neutral) cumulative reading of two cardinal quantifiers can be obtained if the two quantifiers are scope-adjacent. If another quantifier intervenes, a scope interaction appears between the two cardinal quantifiers and the cumulative reading is excluded.

The claim that n-words are negative quantifiers provides no explanation as to why two cooccurring negative quantifiers should give rise to NC readings, rather than to DN. N-words were shown to behave like typical quantifiers and to be negative independently of the NM. So the explanation for the NC reading must be found within their semantics as negative quantifiers. The idiosyncratic scope properties of NQs in NC provide us with an indication of how this happens: the NC reading of two negative quantifiers in Romanian is most likely the effect of their scope-adjacency. This is supported by the contrast between the scope interactions available in a sentence with two n-words and a MANY/ FREQUENTLY quantifier, on the one hand, and the scope possibilities that arise between two non-negative quantifiers and a MANY/ FREQUENTLY quantifier, on the other hand (see (181) vs. (184), and (185) vs. (186)). Moreover, the fact that cumulative quantifiers present this kind of scope idiosyncracy as well suggests that it is not NC that has an exceptional nature alone. It seems to be often the case that some quantifier complexes may receive special interpretations that cannot be accounted for by a direct scope interaction between the monadic quantifiers.

Given the similarity to cumulative polyadic quantifiers, we can relate NC to the semantic framework of Polyadic Quantifiers where we can provide an answer for the NC effect. Natural language presents various cases of quantification that go beyond our theoretical expectations, restricted by the idea that a complex of two (or more) monadic quantifiers must be interpreted by means of iteration/ scope interaction (Section 2.1.3). Several other operations must be used instead to properly derive the semantic contribution of these complex quantifiers. *Different/ same* quantifiers, cumulative and resumptive polyadic quantifiers were shown to need such operations. Within this picture, the NC reading of negative quantifiers represents another such polyadic quantifier. In the following chapter, I will use Polyadic Quantifiers to account for the DN and the NC reading of two negative qauntifiers in Romanian. I will show that DN can be easily obtained by means of iteration, and I will analyze NC as a resumptive interpretation of negative quantifiers.

Chapter 4

Romanian NQs and NC. Towards a syntax-semantics

In Chapter 3 I concluded that Romanian n-words are negative quantifiers. In this chapter I develop the semantic basis for a syntax-semantics analysis of negative concord. In Section 3.4.2, the negative marker was shown to have no semantic contribution to the double negation interpretation of a sentence that contains two n-words. For this reason, the discussion in this chapter exclusively concerns n-words (as negative quantifiers) and the negative marker will be addressed in Chapter 5.

As already indicated, a sentence like (191) may receive two interpretations: NC and DN.[1]

(191)	**Niciun**	student	nu	a	citit	**nicio**	carte.
	no	student	NM	has	read	no	book

i. 'No student read any book.' (NC)
ii. 'No student read no book. (Every student read some book.)' (DN)

An analysis of the syntax-semantics of n-words in Romanian should account for both interpretations. In this chapter I will show that this can be done in the framework of polyadic quantifiers which allows the two negative quantifiers to be interpreted either by resumption or by iteration.

In Section 4.1, I describe DN readings in Romanian as derived by interpreting two monadic negative quantifiers as a binary iteration. NC is shown in Section 4.2 to be properly analyzed by means of resumption. Since resumptive quantifiers are non-iterations, in view of our discussion in Section 2.1.4, I also investigate whether resumption of negative quantifiers is reducible to iteration. As we will see, a resumptive negative quantifier is reducible to an iteration of a negative and an existential quantifier. But despite reducibility, I will argue that resumption of negative quantifiers best accounts for the special properties of Romanian NC and the negative semantics of n-words.

The second part of this chapter is an investigation of the status of resumption with respect to compositionality. Compositionality is an essential requirement for linguistic analyses, but it is often understood to be restricted to functional application as a mode of composing meaning. de Swart and Sag (2002) argue that resumption is important enough for natural language quantification to be taken as an alternative mode of composition to functional application. The attempt to define resumption as a mode of composition in the algebraic system of Montague (1970) turns out to be impossible (Section 4.3). This result leads to several methodological questions concerning, on the one hand, the significance of resumption and polyadic lifts in general for natural language semantics and, on the

[1]Note that the DN reading appears provided that the contextual conditions presented in Section 3.4.2 are met.

other hand, the adequacy of the current notion of compositionality for natural language and linguistic theory. This discussion is presented in Section 4.4, where I also motivate my decision to integrate resumption in Lexical Resource Semantics, a task that will be pursued in Chapter 5.

4.1 Iteration and negation

In this section, I present a GQT account of the DN reading of sentences like (191), as obtained by iteration of two negative quantifiers. I adopt the GQT representation of a negative quantifier, so an n-word will be represented as the generalized quantifier NO, with the semantics in DEFINITION 2.2c and DEFINITION 2.1c and repeated below in the more convenient form of LEMMA 2.1c.

> **Lemma 2.1c** (p. 13)
> For a domain E, for every A, B $\subseteq$ E:
> $([\![\text{NO}]\!](\text{A}))(\text{B}) = 1$ iff $\text{B} \in \{\text{X} \subseteq \text{E} \mid \text{A} \cap \text{X} \neq \emptyset\} \Leftrightarrow [\![\text{NO}]\!](\text{A}, \text{B}) = 1$ iff $\text{A} \cap \text{B} = \emptyset$

In GQT, we represent the sentence in (191) by means of a binary quantifier taking the relation READ to a truth value:

(192) **Niciun** student nu a citit **nicio** carte.
no student NM has read no book

$(\text{NO}^{\text{STUDENT}}, \text{NO}^{\text{BOOK}})(\text{READ})$
$(\text{NO}, \text{NO})^{\text{STUDENT, BOOK}}(\text{READ})$

To interpret the type $\langle 2\rangle$ quantifier $(\text{NO}^{\text{STUDENT}}, \text{NO}^{\text{BOOK}})$, we may apply one of the polyadic lifts presented in Section 2.1. If we apply iteration as in DEFINITION 2.8, we obtain the truth conditions in (193):

> **Definition 2.8** (p. 22) Iteration of two type $\langle 1\rangle$ quantifiers
> *For* Q_1, Q_2, *quantifiers of type* $\langle 1\rangle$, $It(\text{Q}_1, \text{Q}_2)$ *is the type* $\langle 2\rangle$ *quantifier defined, for any domain* E, *any* $x, y \in \text{E}$, *and any* $\text{R} \subseteq \text{E}^2$, *as:*
> $$It(\text{Q}_1, \text{Q}_2)(\text{R}) = (\text{Q}_1 \circ \text{Q}_2)(\text{R}) = \text{Q}_1(\{x \in \text{E}^1 \mid \text{Q}_2(\{y \in \text{E}^1 \mid (x,y) \in \text{R}\}) = 1\})$$

(193)
$$It([\![\text{NO}^{\text{STUDENT}}]\!], [\![\text{NO}^{\text{BOOK}}]\!])([\![\text{READ}]\!]) = 1$$
$$\overset{D:2.8}{\Longleftrightarrow} ([\![\text{NO}^{\text{STUDENT}}]\!] \circ [\![\text{NO}^{\text{BOOK}}]\!])([\![\text{READ}]\!]) = 1$$
$$\overset{D:2.8}{\Longleftrightarrow} [\![\text{NO}^{\text{STUDENT}}]\!](\{x \mid [\![\text{NO}^{\text{BOOK}}]\!](\{y \mid (x,y) \in [\![\text{READ}]\!]\})\}) = 1$$
$$\overset{L:2.1}{\Longleftrightarrow} [\![\text{STUDENT}]\!] \cap \{x \mid [\![\text{BOOK}]\!] \cap \{y \mid (x,y) \in [\![\text{READ}]\!]\} = \emptyset\} = \emptyset$$

The truth conditions in (193) suggest a DN interpretation: the intersection between the set of students and the set of people who didn't read any book is empty. DN is the consequence of both NO's contributing their negative semantics to the meaning of the binary quantifier. Thus iteration of two negative quantifiers accounts for the DN interpretation in (191).

Scope of NQs within DN A negative quantifier is logically equivalent to a universal quantifier outscoping negation or an existential outscoped by negation.[2] Taking into account the semantics of NO, EVERY, and SOME in DEFINITION 2.2, p. 12, this equivalence can be established in GQT terms as in (194). The symbol "¬" is used as in DEFINITION 4.1 taken from Peters and Westerståhl (2006, p. 92):

(194) a. $[\![\text{NO}]\!](\text{A,B}) = 1$ iff $\text{A} \cap \text{B} = \emptyset$

b. $[\![\text{EVERY}]\!](\text{A, B}) = 1$ iff $\text{A} \subseteq \text{B}$

$\Leftrightarrow [\![\text{EVERY}]\!](\text{A, B}) = 1$ iff $\text{A} \cap \text{B} = \text{A}$

$\Leftrightarrow [\![\text{EVERY}]\!](\text{A, B}) = 1$ iff $\text{A}\cap\neg\text{B} = \emptyset$

$\overset{(194a)}{\Longrightarrow} [\![\text{NO}]\!](\text{A, B}) = [\![\text{EVERY}]\!](\text{A}, \neg\text{B})$

c. $[\![\text{SOME}]\!](\text{A, B}) = 1$ iff $\text{A} \cap \text{B} \neq \emptyset$

$\overset{(194a)}{\Longrightarrow} [\![\text{NO}]\!](\text{A, B}) = \neg[\![\text{SOME}]\!](\text{A, B})$

Definition 4.1 *For* Q *a quantifier of type* $\langle 1 \rangle$*, a domain* E*, and* A$\subseteq$ E*, we define the following negative operations on quantifiers:*

$\text{Q}(\neg\text{A}) = \text{Q}(\text{E} - \text{A})$ *(inner negation/ postcomplement)*

$(\neg\text{Q})(\text{A}) = \neg(\text{Q}(\text{A}))$ *(outer negation/ complement)*

Considering the interaction between iteration and inner/ outer negation in LEMMA 4.1 below[3], the iteration of two negative quantifiers is equivalent to the iteration of a universal and an existential quantifier (195), which explains the resulting positive interpretation of a DN reading.

Lemma 4.1 Iteration and inner/ outer negation*:*

$(\text{Q}_1\ \neg)(\neg\ \text{Q}_2) = \text{Q}_1 \circ \text{Q}_2$ (Peters and Westerståhl (2006, p. 348))

(195) $[\![\text{NO STUDENT}]\!] \circ [\![\text{NO BOOK}]\!]$

$\overset{(194)}{=\!=} ([\![\text{EVERY STUDENT}]\!]\neg) \circ (\neg[\![\text{SOME BOOK}]\!])$

$\overset{L:4.1}{=\!=} [\![\text{EVERY STUDENT}]\!] \circ [\![\text{SOME BOOK}]\!]$

Note that to be able to apply LEMMA 4.1 in (195), we must represent the first negative quantifier with a universal outscoping negation and the second negative quantifier as an existental outscoped by negation. In Section 2.1.2 we saw that iteration of a universal and an existential quantifier displays order dependence and implicitly, scope interaction. Thus the order of the negative quantifiers plays an important role, since it determines whether the universal quantifier is restricted by STUDENT or by BOOK. The order in (195) yields the reading in (191), "every student read some book" (see also (196a)). The other order gives the interpretation "every book was read by a student", expressed by (196b).

(196) a. $([\![\text{NO STUDENT}]\!] \circ [\![\text{NO BOOK}]\!])([\![\text{READ}]\!])$

$\overset{(195)}{=\!=} ([\![\text{EVERY STUDENT}]\!] \circ [\![\text{SOME BOOK}]\!])([\![\text{READ}]\!])$

[2] See also (176), p. 97.

[3] Note that LEMMA 4.1 is the GQT version of the logical law of DN given in Section 1.2.

b. $([\![\text{NO BOOK}]\!] \circ [\![\text{NO STUDENT}]\!])([\![\text{READ}]\!]^{-1})$

$\overset{(194)}{=} (([\![\text{EVERY BOOK}]\!]\neg) \circ (\neg[\![\text{SOME STUDENT}]\!]))([\![\text{READ}]\!]^{-1})$

$\overset{L:4.1}{=} ([\![\text{EVERY BOOK}]\!] \circ [\![\text{SOME STUDENT}]\!])([\![\text{READ}]\!]^{-1})$

The two readings in (196) are not equivalent, so if two monadic negative quantifiers are composed by iteration, their linear order determines their scope, and thus has effects on interpretation.[4]

Despite the equivalence between the NO $\circ$ NO binary quantifier and the positive EVERY $\circ$ SOME, not all instances of DN can be directly reduced to iterations of positive quantifiers. It is the case for (191), because there is no other operator intervening between the two monadic quantifiers. The two negations can be represented as adjacent to one another, and thus make possible the application of LEMMA 4.1. However, the intervention of another operator between the two negative quantifiers prevents this, and then no equivalence arises between the two negations and a positive iteration. This is the case with the scopal readings of the two NO's in Section 3.5.3 (see (187b), (188b)), where the intervening quantifiers MANY and FREQUENTLY make a direct equivalence between the two NO's and the positive binary quantifier EVERY $\circ$ SOME unavailable. I generally use the term "double negation" for both cases.

In conclusion, we associate a DN interpretation with the binary quantifier obtained via iteration of two negative quantifiers which contribute their semantics independently of each other. DN is also the "scopal reading" of a sentence with two negative quantifiers, since there is a scope interaction between the two quantifiers: the leftmost quantifier takes scope over the rightmost one. As will become obvious in Section 4.2.2, these two properties (the independent semantic contribution of the monadic quantifiers and the possible scope interaction between them) characterize iteration, but not resumption of negative quantifiers.

4.2 Romanian NC as resumption

Resumption as a polyadic lift, defined in Section 2.1.3 has been suggested to account for instances of NC in dialects of English, where the interpretation of (197) is that there is no (MAN, WOMAN) pair in the LOVE relation (see van Benthem (1989), May (1989), Keenan (1992), Keenan and Westerståhl (1997)):

(197) **No** man loves **no** woman.

The same idea will be used here for Romanian NC, and I will show that lifting several negative quantifiers to a resumptive polyadic quantifier correctly accounts for the characteristics of the NC interpretation in Romanian.

4.2.1 NC as NO^k

In Section 4.1 it was shown that iteration unambiguously yields a DN reading in a sentence with two n-words (192), so NC remains unaccounted for. In Section 3.5.5, I showed that there are important similarities between NC and unreducible polyadic quantifiers with respect to interpretation

[4]There are two explanations for why linear order fixes the scope of two negative quantifiers. One has to do with the general characteristics of quantifier scope in Romanian, as illustrated in Section 3.5.2. The other concerns the condition on DN that it occur in denial contexts. Since the left periphery of a sentence is most active in relation to the discourse, the n-word that brings about denial has to appear in this area and will thus usually precede the other n-word in linear order.

and scope.[5] This observation suggests that NC may be accounted for by one of the (polyadic) lifts alternative to iteration. Since in NC the negative semantics carried by two n-words is interpreted only once, the representation of NC resembles that of multiple wh-questions (see (32) repeated below as (198)) which were analyzed in Section 2.1.3 by means of the resumptive quantifier $\mathrm{WH}_{\mathrm{E}^2}$:

(198) Which dog chased which cat?
$(\mathrm{WH}^{\mathrm{DOG}}, \mathrm{WH}^{\mathrm{CAT}})(\mathrm{CHASE})$
$(\mathrm{WH}, \mathrm{WH})^{\mathrm{DOG, CAT}}(\mathrm{CHASE})$
$[\![\mathrm{WH}]\!]_{\mathrm{E}^2}^{[\![\mathrm{DOG}]\!] \times [\![\mathrm{CAT}]\!]}([\![\mathrm{CHASE}]\!])$

For the NC reading in (191), I suggest an account in terms of a resumptive quantifier $\mathrm{NO}_{\mathrm{E}^2}$, which we also write as NO^2, according to the convention below:

Convention 4.1 *For a domain* E *and a quantifier* $\mathrm{Q}_{\mathrm{E}^k}$, *we have the following convention:*

$$\mathrm{Q}_{\mathrm{E}^k} = \mathrm{Q}^k$$

If we apply binary resumption (DEFINITION 2.16) in order to interpret the polyadic quantifier in (192), we obtain the type $\langle 1, 1, 2\rangle$ quantifier in (199b).

Definition 2.16 (p. 32) Binary resumption of type $\langle 1, 1\rangle$ quantifiers
For a quantifier Q *of type* $\langle 1, 1\rangle$, *given* E *the domain,* $\mathrm{A}, \mathrm{B} \subseteq \mathrm{E}$, $\mathrm{R} \subseteq \mathrm{E}^2$, *the polyadic quantifier* $Res^2(\mathrm{Q})$ *of type* $\langle 1^2,2\rangle$ *derived from* Q *is defined as:*
$$Res^2(\mathrm{Q})_{\mathrm{E}}^{\mathrm{A,B}}(\mathrm{R}) = \mathrm{Q}_{\mathrm{E}^2}^{\mathrm{A}\times\mathrm{B}}(\mathrm{R})$$

(199) a. **Niciun** student nu a citit **nicio** carte.
no student NM has read no book
$(\mathrm{NO}^{\mathrm{STUDENT}}, \mathrm{NO}^{\mathrm{BOOK}})(\mathrm{READ})$
$(\mathrm{NO}, \mathrm{NO})^{\mathrm{STUDENT, BOOK}}(\mathrm{READ})$

b. $Res^2([\![\mathrm{NO}]\!])_{\mathrm{E}}^{[\![\mathrm{STUDENT}]\!],\, [\![\mathrm{BOOK}]\!]}([\![\mathrm{READ}]\!])$
$\overset{D:2.16}{=\!=} [\![\mathrm{NO}]\!]_{\mathrm{E}^2}^{[\![\mathrm{STUDENT}]\!] \times [\![\mathrm{BOOK}]\!]}([\![\mathrm{READ}]\!])$

Given the semantics of NO (i.e. NO^1) in DEFINITION 2.2c, we define the meaning of NO^k as in DEFINITION 4.2 and we interpret the binary quantifier in (199b) as in (200). In its NC interpretation, the sentence in (199) means that there are no (STUDENT, BOOK) pairs in the READ relation: the intersection between the set of (STUDENT, BOOK) pairs and the set of pairs of objects in the READ relation is empty.

Definition 2.2c (p. 12) The semantics of NO
For a domain E, *for every* $\mathrm{A}, \mathrm{B} \subseteq \mathrm{E}$:

c. $[\![\mathrm{NO}]\!](\mathrm{A}, \mathrm{B}) = 1$ iff $\mathrm{A} \cap \mathrm{B} = \emptyset$

[5] This matter will be addressed in more detail in Section 4.2.4.2, where I argue for the general incompatibility between NC and the mechanism of iteration.

Definition 4.2 The semantics of NO^k
For a domain E*, for every* $k \in \mathbb{N}^0$*, for every* $\text{A}_1, \text{A}_2, ..., \text{A}_k \subseteq \text{E}$, $\text{R} \subseteq \text{E}^k$*:*

$$[\![\text{NO}^k]\!](\text{A}_1 \times \text{A}_2 \times ... \times \text{A}_k, \text{R}) = 1 \text{ iff } (\text{A}_1 \times \text{A}_2 \times ... \times \text{A}_k) \cap \text{R} = \emptyset$$

(200) $[\![\text{NO}]\!]_{\text{E}^2}^{[\![\text{STUDENT}]\!] \times [\![\text{BOOK}]\!]}([\![\text{READ}]\!]) = 1$
$\overset{D:4.2}{\Longleftrightarrow} ([\![\text{STUDENT}]\!] \times [\![\text{BOOK}]\!]) \cap [\![\text{READ}]\!] = \emptyset$

Crucially, the polyadic quantifier NO^2 in (200) expresses only one negation, just like a monadic one, and this yields the NC interpretation of (199a).

The same result may be obtained for more complex NC constructions with three or more n-words by following the general definition of k-ary resumption in DEFINITION 2.15. An example is given in (201), where ternary resumption applies to NO:

Definition 2.15 (p. 32) *K*-ary resumption of type $\langle 1, 1 \rangle$ quantifiers
For a quantifier Q *of type* $\langle 1, 1 \rangle$*, given* E *the domain, for any* $k \geq 1$, $\text{A}_1, \text{A}_2, ..., \text{A}_k \subseteq \text{E}$, $\text{R} \subseteq \text{E}^k$*, the polyadic quantifier* $Res^k(\text{Q})$ *of type* $\langle 1^k, k \rangle$ *derived from* Q *is defined as:*
$Res^k(\text{Q})_{\text{E}}^{\text{A}_1, \text{A}_2, ..., \text{A}_k}(\text{R}) = \text{Q}_{\text{E}^k}^{\text{A}_1 \times \text{A}_2 \times ... \times \text{A}_k}(\text{R})$

(201) a. **Nimeni** nu a dat **nimănui nimic**.
nobody NM has given nobody nothing
'Nobody gave anybody anything.'
$(\text{NO}^{\text{PERSON}}, \text{NO}^{\text{PERSON}}, \text{NO}^{\text{THING}})(\text{GIVE})$

b. $Res^3([\![\text{NO}]\!])_{\text{E}}^{[\![\text{PERSON}]\!], [\![\text{PERSON}]\!], [\![\text{THING}]\!]}([\![\text{GIVE}]\!]) = 1$
$\overset{D:2.15}{\Longleftrightarrow} [\![\text{NO}]\!]_{\text{E}^3}^{[\![\text{PERSON}]\!] \times [\![\text{PERSON}]\!] \times [\![\text{THING}]\!]}([\![\text{GIVE}]\!]) = 1$
$\overset{D:4.2}{\Longleftrightarrow} ([\![\text{PERSON}]\!] \times [\![\text{PERSON}]\!] \times [\![\text{THING}]\!]) \cap [\![\text{GIVE}]\!] = \emptyset$

In conclusion, by means of k-ary resumption, we may account for NC readings of sentences with any number of n-words. In what follows we will be concerned with binary resumptions in particular, but at times, we may consider ternary examples as well.

4.2.2 DN vs. NC

I have just shown how iteration and resumption of two negative quantifiers can account for the DN and the NC reading, respectively, of a sentence with two n-words. In this section I briefly address the question of how the different properties of iterations vs. inherently polyadic lifts (or "non-iterations") are reflected in the properties of the constructions that they account for (i.e. DN and NC). I discuss two issues: 1) the impact that the order of the monadic quantifiers has on the interpretation of the whole (i.e. scope interaction vs. scope neutrality), and 2) the way in which the semantics of the monadic quantifiers is contributed to the semantics of the whole.

I showed that in non-iterations the order in which the monadic quantifiers are composed has no effect on the interpretation of the polyadic one (Section 2.1.3.4). This is either because the order is pre-established in the semantics of the polyadic quantifier (for instance DIFFERENT/ SAME quantifiers) or because the semantics of the polyadic quantifier makes the order irrelevant (cumulation and resumption). For iterations, the order in which the monadic quantifiers are composed was shown to influence the final interpretation. The few apparent exceptions have to do with the semantics of particular monadic quantifiers, which yields equivalent truth conditions even if the order is changed.

This difference between iterations and non-iterations carries over to the DN readings as iterations and NC readings as resumptions. For DN, the order in which the two monadic quantifiers are composed indicates the scope of the quantifiers (see Section 4.1). If we consider resumption in (200) and (201b), the order question does not arise, because there is only one operator (see also Section 2.1.3.3).

With respect to the way the individual semantics of the two quantifiers is contributed to the whole, in iterations each of the monadic quantifiers contributes its own semantics independently of the others (Section 2.1.2). The same mechanism is at work with DN: the semantics in (193) indicates that both negative quantifiers contribute their semantic negation which can be truth-conditionally checked, independently of the other quantifier(s).

The situation is different with resumption and NC. Although in (199a) and (201) there are two/three monadic negative quantifiers, the interpretation of the two sentences contains only one (binary/ternary) negative quantifier (see (200) and (201b), respectively). This is the effect of the resumption operation which applies only to quantifiers that bear the same operator, and ensures that the semantics of the operator is contributed to the polyadic quantifier only once. This is how NC arises. With resumption, the meaning of each monadic quantifier is contributed to that of the polyadic quantifier only by making sure that the same meaning is contributed by the other quantifier(s), too. So, truth conditions are verified for the whole polyadic quantifier at once.

The contrast between the ways in which the semantics for DN and NC is built confirms the generalization in Section 2.1.3.4 concerning iterations and non-iterations, if we view the two readings as particular instantiations of the two kinds of polyadic lifts.

4.2.3 Reducibility of NO^2

In Section 2.1.3 inherently polyadic lifts were proposed in situations where iteration could not derive the right interpretation of particular polyadic quantifiers which appear in natural language. With respect to negation, it was shown above that iteration only derives the DN reading, but not the NC one. This gives us a first motivation for employing resumption.

In Section 2.1.4 I showed how we can test if a non-iteration is theoretically necessary for a semantic description. This is the case if there is no iteration that yields the same interpretation. The Reducibility Equivalence theorem of Keenan (1992) (THEOREM 2.1) helps us to determine if a polyadic quantifier is reducible to an iteration of monadic quantifiers. We saw that non-iterations containing DIFFERENT and cumulations are indeed unreducible. In what follows I investigate the status of the resumptive quantifier NO^2 with respect to reducibility.

> **Theorem 2.1** (p. 36) Reducibility Equivalence (RE):
> *For every domain* E *and* Q_1, Q_2, *reducible functions of type* $\langle 2 \rangle$,
> $\mathrm{Q}_1 = \mathrm{Q}_2$ *iff for all* $\mathrm{A}, \mathrm{B} \subseteq \mathrm{E}$, $\mathrm{Q}_1(\mathrm{A}\times\mathrm{B}) = \mathrm{Q}_2(\mathrm{A}\times\mathrm{B})$

Let us consider the constructions below, with the resumptive NO^2 and the iteration NO $\circ$ SOME:[6]

[6]Note that checking whether NO^2 is reducible to NO $\circ$ SOME is reminiscent of the NPI analyses of NC which interpret a sentence like (202a) by an iteration of one negative quantifier followed by an existential quantifier, or truth-conditionally equivalent variations thereof.

(202) a. **Niciun** student nu a citit **nicio** carte.
no student NM has read no book
'No student read any book.'
$[\![\text{NO}]\!]^{[\![\text{STUDENT}]\!] \times [\![\text{BOOK}]\!]}_{\text{E}^2}([\![\text{READ}]\!])$

b. **Niciun** student nu a citit **o** carte.
no student NM has read a book
'No student read a book.'
$([\![\text{NO}]\!]^{[\![\text{STUDENT}]\!]} \circ \text{SOME}^{[\![\text{BOOK}]\!]})([\![\text{READ}]\!])$

We test if the two quantifiers are identical. Assume a domain E containing the subsets $[\![\text{STUDENT}]\!] = \{s_1, s_2\}$, $[\![\text{BOOK}]\!] = \{b_1, b_2\}$. If A and B simultaneously contain at least one student and one book, respectively, both $[\![\text{NO}^2]\!]$ and the iteration $[\![\text{NO}]\!] \circ [\![\text{SOME}]\!]$ yield falsity, since the Cartesian product A×B does contain one (or more) (STUDENT, BOOK) pair(s).

If A = $\emptyset$ or B= $\emptyset$, then A×B= $\emptyset$. Applying $[\![\text{NO}^2]\!]$ to A×B, an empty set, we obtain truth, since $([\![\text{STUDENT}]\!] \times [\![\text{BOOK}]\!]) \cap \emptyset = \emptyset$, as required by the truth conditions of $[\![\text{NO}^2]\!]$. If we apply $[\![\text{NO}]\!] \circ$ $[\![\text{SOME}]\!]$ to A×B in these conditions, we again obtain truth, because $[\![\text{STUDENT}]\!] \cap ([\![\text{BOOK}]\!] \cap \emptyset)$= $[\![\text{STUDENT}]\!] \cap \emptyset$= $\emptyset$.

In conclusion, the resumption $[\![\text{NO}^2]\!]$ and the iteration $[\![\text{NO}]\!] \circ [\![\text{SOME}]\!]$ have the same truth conditions on cross-product relations. If we knew that $[\![\text{NO}^2]\!]$ is a reducible function, by RE we would now conclude that the two are identical. However, this is something we do not know. In previous examples (with DIFFERENT quantifiers and with cumulation) in Section 2.1.4, each time we found a binary relation for which the non-iteration and the iteration did not yield the same value. This was enough to conclude that the non-iteration is unreducible. For $[\![\text{NO}^2]\!]$ and $[\![\text{NO}]\!] \circ [\![\text{SOME}]\!]$ it is hard to find such a relation, since as we will see, the two functions are identical, so the former is reducible to the latter.

In order to show that the two functions are identical, we assume that they yield different values on the same relations and this will lead to a contradiction which will indicate the falsity of the initial assumption. Consider our domain E, A, B $\subseteq$ E, $x, y \in$ E, and R $\subseteq$ E^2. Take now $[\![\text{NO}^2]\!]$(A, B, R)= 0 and $[\![\text{NO}]\!] \circ [\![\text{SOME}]\!]$(A, B, R)= 1. Let us follow the implications of these two statements:[7]

(203) a. $[\![\text{NO}^2]\!](\text{A}, \text{B}, \text{R}) = 0$
$\overset{D:4.2}{\Longleftrightarrow} (\text{A} \times \text{B}) \cap \text{R} \neq \emptyset$
$\Longrightarrow \text{A} \cap \text{R}y \neq \emptyset$ and $\text{B} \cap \text{R}x \neq \emptyset$

b. $[\![\text{NO}]\!] \circ [\![\text{SOME}]\!](\text{A}, \text{B}, \text{R}) = 1$
$\overset{D:2.10}{\Longleftrightarrow} [\![\text{NO}]\!](\text{A}, \{x|\ [\![\text{SOME}]\!](\text{B}, \{y|(x,y) \in \text{R}\}) = 1\}) = 1$
$\overset{D:2.2}{\Longleftrightarrow} \text{A} \cap \{x|\ \text{B} \cap \{y|(x,y) \in \text{R}\} \neq \emptyset\} = \emptyset$
$\Longrightarrow \text{A} \cap \text{R}y = \emptyset$ and $\text{B} \cap \text{R}x \neq \emptyset$

The two conjunctions in the last lines of (203a) and (203b) cannot be true at the same time, and this entails that the initial assumption that $[\![\text{NO}^2]\!]$(A, B, R)= 0 and $[\![\text{NO}]\!] \circ [\![\text{SOME}]\!]$(A, B, R)= 1 is false. This proves that $[\![\text{NO}^2]\!]$(A, B, R) = $[\![\text{NO}]\!] \circ [\![\text{SOME}]\!]$(A, B, R). In conclusion, the resumption NO^2 is reducible to the iteration NO ∘ SOME.

[7] Recall that Rx = $\{y \mid (x,y) \in \text{R}\}$ and Ry = $\{x \mid (x,y) \in \text{R}\}$ (CONVENTION 2.5, p. 2.5).

4.2.4 Consequences of the reducibility of NO^2

If NO^2 is reducible to NO ∘ SOME, the immediate question to ask is whether analyzing NC readings with resumption is necessary. I will show below that considering the properties of n-words and NC in Romanian, the resumptive quantifier is more adequate than the logically equivalent iteration.

4.2.4.1 The monadic quantifiers

The resumptive quantifier is built of two negative quantifiers. As indicated in Section 4.1, the iteration of the two negative components does not derive NC, the reading obtained by resumption. The iteration NO ∘ SOME contains only one negative quantifier which (necessarily) outscopes an existential quantifier. An analysis of NC as the iteration NO ∘ SOME makes several predictions with respect to the properties of NC.

First of all, it predicts that in (202a) above, the n-word *niciun student* corresponds to a negative quantifier NO STUDENT, while *nicio carte* corresponds to an existential quantifier SOME BOOK. This suggests that the determiner *niciun* (feminine *nicio* in (202a)), and determiner n-words in general, are lexically ambiguous between negative and existential quantifiers. However, Romanian n-words have a systematic behavior from a syntax-semantics point of view and nothing indicates that some n-words may be lexically negative, others existential (i.e. non-negative) and yet others ambiguous between the two.[8] Any n-word can fill the first argument slot of a relation and thus express a negative quantifier: see also the bare *nimeni* and *nimic* in both linear order possibilities in (204a) and (204b) below:

(204) a. **Nimeni** nu a citit **nimic**.
nobody NM has read nothing
'Nobody read anything.'
$([\![\mathrm{NO}]\!]^{[\![\mathrm{PERSON}]\!]} \circ [\![\mathrm{SOME}]\!]^{[\![\mathrm{THING}]\!]})([\![\mathrm{READ}]\!])$

b. **Nimic** nu a fost citit de către **nimeni**.
nothing NM has been read by nobody
'Nothing has been read by anybody.'
$([\![\mathrm{NO}]\!]^{[\![\mathrm{THING}]\!]} \circ [\![\mathrm{SOME}]\!]^{[\![\mathrm{PERSON}]\!]})([\![\mathrm{READ}]\!]^{-1})$

In Section 3.3.2.2, I showed that Romanian n-words have anti-additive properties. The examples in (130) are repeated below:

(205) a. articol [de **nimeni** citat *sau* lăudat] = articol [de **nimeni** citat *şi* de **nimeni** lăudat]
article by nobody cited or praised article by nobody cited and by nobody praised
'article which hasn't been cited or praised by anybody' = 'article which hasn't been cited and which hasn't been praised by anybody'

b. A: Who was at the door?
B: **Nimeni** cunoscut *sau* important. = **Nimeni** cunoscut *şi* **nimeni** important.
nobody known or important nobody known and nobody important

[8]See Richter and Sailer (1999*a*) for an account of various French negative (polarity) elements assuming lexical ambiguity. French n-words seem to exhibit a high degree of flexibility with respect to their negative contribution, which is not the case in Romanian.

If in a sentence with two n-words, the first one were negative and the other one non-negative – as the iteration NO ∘ SOME suggests – we would expect the second n-word to not exhibit the anti-additive property anymore. As the example below indicates, this is not the case: both the first and the second n-word display anti-additivity:

(206) **Niciun student** înalt *sau* blond nu a lăsat **nicio carte** galbenă *sau* roşie.
no student tall or blond NM has left no book yellow or red
'No tall or blond student left any yellow or red book.'

= **Niciun student** înalt *şi* **niciun student** blond nu a lăsat **nicio carte** galbenă *şi*
no student tall and no student blond NM has left no book yellow and
nicio carte roşie.
no book red

Thus the test indicates that the second n-word in linear order carries negative semantics just like the first one.

For some Romance languages, it has been proposed that in some contexts n-words are negative, but in some others they are not (see Zanuttini (1991) and Giannakidou (2006)). For Italian, Zanuttini (1991) argues that n-words are negative quantifiers in declarative sentences (see (207a) and (207b)), but they are non-negative NPIs in questions (see (207c) below). She uses the *almost*-test (Section 3.3.4) as indicative of this contrast. The incompatibility between *nessuno* and *quasi* in the question (207c) is taken to indicate the NPI status of the former:

(207) a. *Quasi* **nessuno** ha telefonato.
almost nobody has called

b. Non a telefonato *quasi* **nessuno**.
NM has called almost nobody
'Almost nobody called.'

c. * Ha telefonato *quasi* **nessuno**?
has called almost nobody
*'Has almost anybody called?'

(Zanuttini (1991, pp. 116–117))

If in a sentence with two n-words in Romanian, the first n-word were negative and the second an NPI, we would expect the same contrast as in Italian with respect to *almost*-modification, but this is not the case, as (208) shows: both n-words may be modified by *almost* at the same time or separately.

(208) a. *Aproape* **niciun** student nu a citit **nicio** carte.
almost no student NM has read no book

b. **Niciun** student nu a citit *aproape* **nicio** carte.
no student NM has read almost no book

c. *Aproape* **niciun** student nu a citit *aproape* **nicio** carte.
almost no student NM has read almost no book
'Almost no student read any book.'

I conclude here that the lexical ambiguity assumption suggested by an analysis of NC as the iteration NO ∘ SOME contravenes the empirical evidence for Romanian NC, and it should be avoided.

One way to account for NC in terms of iteration but without assuming lexical ambiguity for n-words is to represent NC in (202a) and (204) as the iteration $\neg([\![\text{SOME}]\!] \circ [\![\text{SOME}]\!])$, i.e. the negation of an iteration of two existential quantifiers. This suggests that negation comes from somewhere outside the semantics of n-words, and n-words are all existential quantifiers. Negation can be argued to be contributed by the NM, since the latter is always present in NC and it must be semantically negative, given that it yields sentential negation (see (89) repeated below as (209)):

(209) a. Studenţii au citit romanul.
students-the have read novel-the
'The students read the novel.'

b. Studenţii **nu** au citit romanul.
students-the NM have read novel-the
'The students didn't read the novel.'

But such an assumption is also problematic. This account is similar to an NPI analysis of NC in which the NM is the only carrier of negation and n-words are semantically licensed by it. This was argued in Section 3.3 to be inappropriate for Romanian NC. There are two basic reasons why such an approach fails: one concerns the relationship between the NM and n-words, the other the semantics of n-words. First, we saw that the NM does not qualify as a semantic licenser for n-words (Section 3.3.2.2), since it fails anti-additivity with respect to n-words, which means that the negation of the NM must concord with that of the n-word (see (132) repeated below as (210)):

(210) Ion **nu** a citit **nicio** carte *sau* **niciun** articol.
John NM has read no book or no article

a. Anti-additivity
≠ Ion **nu** a citit **nicio** carte *şi* Ion **nu** a citit **niciun** articol.
John NM has read no book and John NM has read no article
'John didn't read any book and John didn't read any article.'

b. Ellipsis
= Ion **nu** a citit **nicio** carte *sau* Ion **nu** a citit **niciun** articol.
John NM has read no book or John NM has read no article
'John read no book or John read no article.'

Secondly, assuming, contrary to the conclusion in Chapter 3, that n-words are not negative, one cannot account for the anti-additivity in (205) and (206), or the DN contexts in Sections 3.4 and 3.5.4.

4.2.4.2 Scope dissimilarities between NC and iteration

The special properties of NC in (scope) interaction with other operators also indicate that iteration is not the right mechanism to account for NC.

We saw that NC readings require that the negative quantifiers (expressed by n-words) be scope-adjacent (Section 3.5). Once another operator intervenes between them, the NC reading is replaced by a DN reading (see (185), slightly modified in (211) below):

(211) **Niciun** student nu a citit **frecvent** **nicio** carte.
no student NM has read frequently no book

a. NO (STUDENT) > FREQUENTLY > NO (BOOK) ??NC/ DN

b. NO (STUDENT) > NO (BOOK) > FREQUENTLY NC/ *DN

c. FREQUENTLY $>$ NO (STUDENT) $>$ NO (BOOK) ?NC/ *DN

As can be noticed in (211a), the intervention of a non-negative quantifier between the two negative ones makes DN the preferred interpretation; the NC reading is barely available in this case.

As its semantics is built step by step (Section 2.1.2), iteration can freely apply to non-similar monadic quantifiers regardless of the order. This is what happens with NO and FREQUENTLY in the DN reading above. The same is possible with the iteration NO $\circ$ SOME in (212) whose interpretation is equivalent to the NC reading in (211a):

(212) **Niciun** student nu a citit **frecvent** **o** carte.
no student NM has read frequently a book
NO $>$ FREQUENTLY $>$ SOME: For no student is it the case that s/he was frequently involved in reading books.

Since in (212), *o* ("a") is a typical existential quantifier (unlike the n-word *nicio* in (211)), the scope reading with the quantifier FREQUENTLY intervening between NO and SOME is fully available. For NC, the corresponding intervention in (211a) is not allowed. Analyzing NC as an instance of the iteration NO $\circ$ SOME would predict that the two constructions have similar scope properties, and thus that (212) receive the interpretation (211a), which is not the case.

This observation leads to the conclusion that the iteration NO $\circ$ SOME fails to explain the (idiosyncratic) properties displayed by NC with respect to the scope interaction between the monadic parts and external non-negative quantifiers. Resumption, on the other hand, establishes a close connection between the monadic parts with no scope interaction between them, and thus resembles NC. It also accounts for the opacity of NC constructions to scope interaction with external quantifiers. Resumption only applies to monadic quantifiers with the same operator, so it cannot incorporate any other operator. As observed in (211), the same property characterizes NC, which suggests an account in terms of resumption.

4.2.4.3 Reducibility of WH^2

If we test reducibility of WH^2 ((35), repeated below as (213)) in terms of the iteration WH $\circ$ SOME in (214), we reach the conclusion that resumptive WH^2, like NO^2, is reducible.

(213) Which dog chased which cat?
(WH$^{\text{DOG}}$, WH$^{\text{CAT}}$)(CHASE)
$[\![\text{WH}]\!]_{\text{E}^2}^{[\![\text{DOG}]\!]\times[\![\text{CAT}]\!]}([\![\text{CHASE}]\!])$

(214) Which dog chased a cat?
(WH$^{\text{DOG}}$, SOME$^{\text{CAT}}$)(CHASE)
= $([\![\text{WH}]\!]^{[\![\text{DOG}]\!]} \circ [\![\text{SOME}]\!]^{[\![\text{CAT}]\!]})([\![\text{CHASE}]\!])$

Consider a domain containing two sets $[\![\text{DOG}]\!] = \{d_1, d_2\}$, $[\![\text{CAT}]\!] = \{c_1, c_2, c_3\}$, and A, B arbitrary subsets of the domain. If A contains at least one dog and B at least one cat, $[\![\text{WH}]\!]^2(\text{A} \times \text{B}) = 1$, since there is at least one (DOG, CAT) pair that gives a true answer to the question in (213).[9] $[\![\text{WH}]\!] \circ [\![\text{SOME}]\!](\text{A} \times \text{B}) = 1$ as well, since there is at least one cat that was chased by a dog, and at least one dog that chased a cat.

[9] See the semantics of $[\![\text{WH}]\!]$ in DEFINITION 2.17, p. 33.

If $\text{A} = \emptyset$ or $\text{B} = \emptyset$, then $\text{A} \times \text{B} = \emptyset$, and both $[\![\text{WH}]\!]^2$ and $[\![\text{WH}]\!] \circ [\![\text{SOME}]\!]$ are false of $\text{A} \times \text{B}$, because there is no (DOG, CAT) pair in $\text{A} \times \text{B}$ (for $[\![\text{WH}^2]\!]$), and there is no dog such that it chased a cat (for $[\![\text{WH}]\!] \circ [\![\text{SOME}]\!]$).

At this point, if we knew that $[\![\text{WH}^2]\!]$ is reducible, we would conclude by RE that $[\![\text{WH}^2]\!]$ is identical to the iteration $[\![\text{WH}]\!] \circ [\![\text{SOME}]\!]$. But we do not know if $[\![\text{WH}^2]\!]$ is reducible and we cannot find a binary relation on which it yields a different truth value from that of $[\![\text{WH}]\!] \circ [\![\text{SOME}]\!]$, so the intuition is that the two binary quantifiers are identical.

In order to show that this is the case, we assume like for $[\![\text{NO}^2]\!]$ above that the two quantifiers yield different values on the same relations. Since by this assumption we arrive at a contradiction (i.e. the two conjunctions in (215a) and (215b) cannot hold at the same time), we may conclude that the assumption is false, so the two binary quantifiers are identical. See the reasoning below for $[\![\text{WH}^2]\!]$ and $[\![\text{WH}]\!] \circ [\![\text{SOME}]\!]$ which resembles the one for $[\![\text{NO}^2]\!]$ and $[\![\text{NO}]\!] \circ [\![\text{SOME}]\!]$ in (203):

(215) a. $[\![\text{WH}^2]\!](\text{A}, \text{B}, \text{R}) = 0$
$\overset{D:2.17}{\iff} (\text{A} \times \text{B}) \cap \text{R} = \emptyset$
$\Longrightarrow \text{A} \cap \text{R}y = \emptyset$ or $\text{B} \cap \text{R}x = \emptyset$

b. $[\![\text{WH}]\!] \circ [\![\text{SOME}]\!](\text{A}, \text{B}, \text{R}) = 1$
$\overset{D:2.10}{\iff} [\![\text{WH}]\!](\text{A}, \{x | \, [\![\text{SOME}]\!](\text{B}, \{y | (x, y) \in \text{R}\}) = 1\}) = 1$
$\overset{D:2.17}{\iff} \text{A} \cap \{x | \, \text{B} \cap \{y | (x, y) \in \text{R}\} \neq \emptyset\} \neq \emptyset$
$\Longrightarrow \text{A} \cap \text{R}y \neq \emptyset$ and $\text{B} \cap \text{R}x \neq \emptyset$

The fact that both NO^2 and WH^2 are reducible to iterations may indicate that resumption as a polyadic lift in general is theoretically superfluous for describing natural language quantification. But Peters and Westerståhl (2002, pp. 192–194) show that resumptive MOST^2 is unreducible.[10] So not all resumptive quantifiers are reducible to iterations.

Moreover, Moltmann (1995, 1996) argues that resumption is determinant in accounting for exceptive sentences like (218):[11]

(218) **Niciun** student nu a citit **nicio** carte, *în afară de* Maria, "Syntactic Structures".
no student NM has read no book, except Maria, S.S.
'No student read any book, except Maria "Syntactic Structures".'

So resumption in general is theoretically motivated and necessary for natural language description.

[10] See also the additional remarks in Section 7.2

[11] Zeijlstra (2004) rejects Moltmann's analysis on the claim that resumption is unable to account for the de dicto reading of (216) and thus wrongly predicts that the devil exists:

(216) **Personne** ne veut parler á **personne**, *sauf* Jean au diable.
nobody NM wants speak to nobody except Jean to devil
'Nobody wants to talk to anybody, except John to the devil.' (Zeijlstra (2004, p. 205))

Although the interaction between intensional verbs and polyadic quantification is a complex issue, note that according to native speakers' intuitions the Romanian sentence corresponding to (216) only has a de re interpretation:

(217) **Nimeni** nu vrea să vorbească cu **nimeni**, *în afară de* Ion cu dracul.
nobody NM wants SJ speak with nobody except John with devil
'Nobody wants to talk to anybody, except John to the devil.'

4.2.5 Conclusion

In conclusion, I argued that resumption provides an appropriate account for NC in Romanian. Moreover, resumption as a polyadic lift can account for several particularities of natural language quantification which cannot be successfully analyzed by means of iteration. For Romanian NC, I gave two important reasons why an analysis in terms of resumption is superior to one based on iteration. First, it is consistent with the negative semantics of n-words, and second, it explains the special scope behavior of NC, by building the semantics of the polyadic quantifier on the basis of the same conditions as those that are required by NC readings. Thus, despite the reducibility of NO^2 to the iteration NO $\circ$ SOME, the former mechanism is more appropriate for the analysis of NC.

In a framework with polyadic quantifiers, the interplay between the NC and the DN interpretations of two Romanian n-words may be uniformly explained via lifting monadic quantifiers with resumption or with iteration.

The rest of this chapter is a close investigation of resumption in its relation to compositionality, traditionally considered an essential property of linguistic theories and of linguistic mechanisms used to describe natural language.

4.3 Resumption and compositionality

The main problem that negative concord raises for linguistic theories is that the presence of several negative constituents in a sentence with a unique sentential negation meaning disobeys the principle of compositionality in its standard understanding (see also Section 1.2). Resumption, as presented in Section 4.2, offers a semantic mechanism to derive NC readings, but we still need to see whether it respects compositionality.

de Swart and Sag (2002) is the first to bring into discussion the compositional status of resumption in a syntax-semantics interface (but see also May (1989)). In their HPSG-based analysis of NC and DN de Swart and Sag argue that resumption is just as motivated as a mode of composition as functional application, viewed as iteration, is. In Section 4.3.1 I give a summary of their account which will show that it does not clarify the status of resumption, because the analysis does not directly address compositionality.

This issue will be discussed in the rest of this chapter. I start by presenting the main assumptions of a compositional interpretation following Hendriks (1993) (Section 4.3.2). In Section 4.3.3, I describe the problems that one encounters in the attempt to define a mode of composition that derives polyadic quantifiers. We will see that both resumption and iteration fail to be compositional because their syntax as polyadic lifts violates the phrase structure syntax of natural language. Moreover, the semantics of resumption cannot be expressed compositionally, because it disregards the semantics of the monadic parts.

4.3.1 de Swart and Sag (2002)

Resumption has been employed to account for NC in de Swart (1999), de Swart and Sag (2002), Corblin et al. (2004), de Swart (2010). de Swart and Sag (2002) is the first attempt to develop a syntax-semantics interface for NC as resumption. The account is fleshed out within the HPSG framework.

de Swart and Sag (2002) focus on French negative sentences with two n-words like (219). This sentence is ambiguous between a DN (219a) and a NC (219b) reading:

(219) **Personne** n'aime **personne**.
nobody NM-loves nobody

a. Nobody loves nobody. (Everybody loves somebody.) (DN)
b. Nobody loves anybody. (NC)

Within polyadic quantification, de Swart and Sag (2002) derive DN by composing the two negative quantifiers with iteration, and NC by applying resumption. In short, the two interpretations of (219) are obtained similarly to the corresponding Romanian ones in Sections 4.1 and 4.2:[12]

(220) **Personne** n'aime **personne**.
nobody NM-loves nobody

a. $It([\![\text{NO}]\!]^{[\![\text{PERSON}]\!]}, [\![\text{NO}]\!]^{[\![\text{PERSON}]\!]})([\![\text{LOVE}]\!]) = 1$
$\overset{D:2.8}{\Longleftrightarrow} [\![\text{NO}]\!]^{[\![\text{PERSON}]\!]} \circ [\![\text{NO}]\!]^{[\![\text{PERSON}]\!]} = 1$
$\overset{L:2.1}{\Longleftrightarrow} [\![\text{PERSON}]\!] \cap \{x \in \text{E} | [\![\text{PERSON}]\!] \cap \{y \in \text{E} | (x,y) \in [\![\text{LOVE}]\!]\} = \emptyset\} = \emptyset$ (DN)

b. $Res^2([\![\text{NO}]\!])_{\text{E}}^{[\![\text{PERSON}]\!], [\![\text{PERSON}]\!]}([\![\text{LOVE}]\!]) = 1$
$\overset{D:2.16}{\Longleftrightarrow} \text{NO}_{\text{E}^2}^{[\![\text{PERSON}]\!] \times [\![\text{PERSON}]\!]}([\![\text{LOVE}]\!]) = 1$
$\overset{D:4.2}{\Longleftrightarrow} ([\![\text{PERSON}]\!] \times [\![\text{PERSON}]\!]) \cap [\![\text{LOVE}]\!] = \emptyset$ (NC)

The polyadic quantifiers sketched in (220) are integrated in a syntax-semantics interface based on Situation Semantics (see Barwise and Perry (1983)), in the HPSG tradition of Pollard and Sag (1994). de Swart and Sag (2002) make use of the Lexical Quantifier Retrieval mechanism proposed in Przepiórkowski (1998) and Manning et al. (1999), roughly defined below:[13,14]

(221) *Lexical Quantifier Retrieval*

$$verb \rightarrow \begin{bmatrix} \text{SS|LOC|CONT} & \begin{bmatrix}\text{QUANTS } retrieve\Big((\Sigma_1 \cup ... \cup \Sigma_n) \dot{-} \Sigma\Big)\end{bmatrix} \\ \text{ARG-ST} & \Big\langle \begin{bmatrix}\text{STORE } \Sigma_1\end{bmatrix}, ... \begin{bmatrix}\text{STORE } \Sigma_n\end{bmatrix} \Big\rangle \\ \text{STORE} & \Sigma \end{bmatrix}$$

By lexical retrieval, the quantifiers that the verb collects from its arguments on its own STORE value get to be interpreted (i.e. retrieved) on the verb's QUANTS list (see also Section 2.3.2.5). The interpretation of quantifiers in a sentence appears directly on the verb, at the *word*-level, hence "lexical" retrieval. For (219), the two interpretations of the sequence of two negative quantifiers are given below: (222a) and (222b) are the syntax-semantics representations of the iteration in (220a) and the resumption in (220b), respectively, as interpreted on the verb under QUANTS:

(222) a. Double negation:

$$\begin{bmatrix} \text{PHON} & \Big\langle \textit{n'aime} \Big\rangle \\ \text{SS|LOC|CONT} & \begin{bmatrix} \text{QUANTS} & \Big\langle \text{NO}_{\text{E}}^{\text{PERSON}}, \text{NO}_{\text{E}}^{\text{PERSON}} \Big\rangle \\ \text{NUCL} & \text{LOVE} \end{bmatrix} \\ \text{ARG-ST} & \Big\langle \begin{bmatrix}\text{STORE } \Big\{\text{NO}_{\text{E}}^{\text{PERSON}}\Big\}\end{bmatrix}, \begin{bmatrix}\text{STORE } \Big\{\text{NO}_{\text{E}}^{\text{PERSON}}\Big\}\end{bmatrix} \Big\rangle \\ \text{STORE} & \{\} \end{bmatrix}$$

[12] For clarity, I continue using the notational conventions in this thesis when presenting the analysis in de Swart and Sag (2002).

[13] Note that the principle in (221) is simplified. de Swart and Sag (2002) generalize lexical retrieval so that not only verbs can retrieve quantifiers but also a preposition like *sans* 'without' which can retrieve negative quantifiers.

[14] The symbol $\dot{-}$ designates a relation of contained set difference which is identical to the familiar set difference with the condition that $\Sigma_1 \dot{-} \Sigma_2$ is defined only if Σ_2 is a subset of Σ_1.

b. Negative concord:

$$\begin{bmatrix} \text{PHON} & \langle \textit{n'aime} \rangle \\ \text{SS}\mid\text{LOC}\mid\text{CONT} & \begin{bmatrix} \text{QUANTS} & \langle \text{NO}_{\text{E}^2}^{\text{PERSON,PERSON}} \rangle \\ \text{NUCL} & \text{LOVE} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \left[\text{STORE } \{\text{NO}_{\text{E}}^{\text{PERSON}}\}\right], \left[\text{STORE } \{\text{NO}_{\text{E}}^{\text{PERSON}}\}\right] \right\rangle \\ \text{STORE} & \{\} \end{bmatrix}$$

Applying resumption and iteration under the value of QUANTS is possible by defining the two operations within the relation *retrieve* in (221). de Swart and Sag (2002) define retrieval as below:

(223) *Retrieve*

Given a set Σ of generalized quantifiers defined on a domain E and a partition of Σ into two sets Σ_1 and Σ_2, where Σ_2 is either empty or else for any $\mathrm{R}_1, \ldots, \mathrm{R}_n \subseteq \mathrm{E}$, $\Sigma_2 = \{\mathrm{NO}^{R_1}, \ldots, \mathrm{NO}^{R_n}\}$, then

$retrieve(\Sigma) = iteration(\Sigma_1 \cup resumption(\Sigma_2))$ (de Swart and Sag (2002, p. 394))

Retrieval of a set Σ of quantifiers usually means composing them by iteration. If there are negative quantifiers, they can either be iterated together with the non-negative ones, in which case Σ_2 is empty, or they can be composed by resumption, and the polyadic quantifier they form undergoes iteration together with the non-negative quantifiers (the set Σ_1). Given $\Sigma = \{\mathrm{NO}_{\mathrm{E}}^{\mathrm{PERSON}}, \mathrm{NO}_{\mathrm{E}}^{\mathrm{PERSON}}\}$ in (220), in order to obtain DN, the set Σ_2 was considered empty. For the NC reading, $\Sigma_2 = \Sigma = \{\mathrm{NO}_{\mathrm{E}}^{\mathrm{PERSON}}, \mathrm{NO}_{\mathrm{E}}^{\mathrm{PERSON}}\}$, and Σ_1 is empty. By applying resumption to Σ_2, we obtain a singleton set containing the binary resumptive quantifier $\mathrm{NO}_{\mathrm{E}^2}^{\mathrm{PERSON,PERSON}}$.

de Swart and Sag (2002) argue that natural language quantification is more complex than iteration predicts, and that for instance resumption should be an alternative to iteration. Their motivation relies heavily on the observations made by the proponents of the Polyadic Quantification framework, already discussed in Section 2.1. They argue that their HPSG-account has the advantage of offering a flexible syntax-semantics interface which can accommodate both iteration and resumption.

However, this account is merely programmatic, and de Swart and Sag do not discuss how their mechanism of quantifier retrieval relates to the traditional matter of compositionality, and where it belongs within the algebraic system developed in Montague (1970) ("Universal Grammar"). Besides this, it is unclear how the two operations *iteration* and *resumption* are to be formulated in the syntax of a logical language for which compositionality can be shown to hold. This will be investigated in the subsequent sections.

4.3.2 Compositionality

To get a better understanding of resumption (and polyadic quantifiers in general) within a compositional grammar, we first need to understand what compositionality is and what levels of the grammar it involves. After an informal presentation of the principle of compositionality, in Section 4.3.2.1 I define a logical language L (similar to that in Hendriks (1993, Ch. 2)) on the basis of which we can compositionally interpret a natural language fragment for Romanian in Section 4.3.2.2. In Section 4.3.2.3, I give a precise description of how the Romanian fragment can be compositionally interpreted in L. I close this section with an example of a Romanian sentence derived and interpreted in accord with the principle of compositionality.

The most general formulation of the principle of compositionality illustrating Montague's understanding of compositionality in his paper "Universal Grammar" (Montague (1970)) is the one in (224) below, previously given in (3), Section 1.2:

(224) *The principle of compositionality* (Partee (1984, p. 281))
The meaning of an expression is a function of the meanings of its parts and of the way they are syntactically combined.

There are three aspects of the principle concerning how the meaning of a complex expression is compositionally built: 1) the *syntax* by which its parts are combined, 2) the *meanings* that the parts carry and 3) the *function* between the two, i.e. the semantic interpretation.

The syntax consists of a collection of basic (lexical) expressions and a set of syntactic operations that recursively derive new syntactic expressions on the basis of other (basic or derived) expressions. The syntax is viewed as an algebra $\langle A, F_\gamma \rangle_{\gamma \in \Gamma}$ where A is the set of all expressions (basic and derived), F_γ is a set of syntactic operations, Γ is a set of indices that identify the syntactic operations, and A is closed under $(F_\gamma)_{\gamma \in \Gamma}$. In principle, no restriction is imposed on the nature or form of the expressions, but in linguistics it is usually assumed that they are strings over some alphabet. Apart from that, they may be empty, overlap, include one another etc. Similarly, no restrictions are imposed on the way the syntactic operations combine the expressions: for instance, they may concatenate them, insert, permute or delete material in them and so on.

Parallel to the syntactic algebra, there must be a structurally similar algebra of meanings $\langle B, G_\gamma \rangle_{\gamma \in \Gamma}$, where B is the set of basic and derived meanings and $(G_\gamma)_{\gamma \in \Gamma}$ is a set of operations that build complex meanings from simpler ones. As in the case of the syntax, no restrictions are imposed on the ways in which the meanings can be affected by the operations.

The principle of compositionality requires that for every n-place syntactic operation F_i there be an n-place semantic operation G_j. That is, G_j interprets semantically what F_i forms syntactically. A semantic interpretation for a language is defined as a *homomorphism* from $\langle A, F_\gamma \rangle_{\gamma \in \Gamma}$ to $\langle B, G_\gamma \rangle_{\gamma \in \Gamma}$. This means that the semantic interpretation of a language is viewed as a function h such that for each n-place syntactic operation F_i and its corresponding n-place semantic operation G_j, for each sequence of expressions $\alpha_1, ..., \alpha_n$, the following holds:

(225) $h(F_i(\alpha_1, ..., \alpha_n)) = G_j(h(\alpha_1), ..., h(\alpha_n))$

(226) is a simple example in which a binary syntactic operation `syntactic-combination-of`, a binary semantic operation `semantic-function-of`, and the semantic interpretation function *meaning-of* combine, so that the interpretation of *John came* is compositionally derived:

(226) *meaning-of*(`syntactic-combination-of`(John, came))
= `semantic-function-of`(*meaning-of*(John), *meaning-of*(came))

4.3.2.1 The logical language

It is common practice in the linguistic literature, especially the literature addressing compositionality (see Montague (1970), Halvorsen and Ladusaw (1979), Dowty et al. (1981), Janssen (1986, 1997), Gamut (1991), Hendriks (1993)) to assign meaning to natural language expressions via an intermediate logical language.

In a simplified formulation, this procedure involves three basic components where the relationship between them is observed by homomorphic functions. The syntax of a logical language is defined as a syntactic algebra to which meaning is assigned via a homomorphism with a semantic algebra, the semantics of the logical language. The syntactic algebra of the natural language is related to the logical syntax by a translation function which must also be homomorphic. The composition of the homomorphism between the logical and the semantic algebra with the translation homomorphism between the natural language syntax and the logical algebra is also a homomorphism and thus ensures that the natural language expressions in the syntactic algebra are compositionally assigned meaning in the (logical) semantic algebra.

Let us define the syntax and semantics of the language L, a language of extensional typed logic. The presentation here closely follows in structure and conventions that in Hendriks (1993, Ch. 2).

The syntax of L We first define the set of semantic types with the two basic types e for individuals and t for truth values:

Definition 4.3 $Type$
Let $Type$ *be the smallest set such that*
$e, t \in Type$,
for each $\tau, \tau' \in Type, \tau\tau' \in Type$.

For every type $\tau \in Type$, there are two sets of basic expressions: a set Var_τ consisting of the *variables* of type τ and a set $Const_\tau$ consisting of the *constants* of type τ:

Definition 4.4 Var
Let Var_τ *be the smallest family of sets such that for each* $\tau \in Type$ *and for each* $i \in \mathbb{N}$,
$v_{i,\tau} \in Var_\tau$.

Definition 4.5 $Const$
$Const_\tau$ *is defined as follows:*
$Const_e = \{j\}$,
$Const_{et} = \{student', book', talk'\}$
$Const_{e(et)} = \{read'\}$
$Const_\tau = \emptyset$ *for* $\tau \notin \{e, (et), (e(et))\}$.

The logical language L is the indexed family of sets $(L_\tau)_{\tau \in Type}$ of well-formed expressions which are defined below:

Definition 4.6 Terms in L_τ
For every type $\tau \in Type$*, the set* L_τ *of* well-formed expressions *of type* τ *is the smallest set such that:*

1. $Var_\tau \subset L_\tau$,
2. $Const_\tau \subset L_\tau$,
3. *for each* $\alpha \in L_t$, $F_1(\alpha) \in L_t$,
4. *for each* $\alpha, \beta \in L_t$, $F_2(\alpha, \beta) \in L_t$,
5. *for each* $\alpha, \beta \in L_{et}$, $F_3(\alpha, \beta) \in L_t$,

6. *for each* $\tau' \in Type$*, for each* $\alpha \in L_{\tau'\tau}$, $\beta \in L_{\tau'}$, $F_{4:\tau':\tau}(\alpha, \beta) \in L_\tau$,
7. *for each* $\tau' \in Type$*, for each* $\alpha \in L_\tau$, $F_{5:\tau':\tau:i}(\alpha) \in L_{\tau'\tau}$,
8. *for each* $\alpha, \beta \in L_\tau$, $F_{6:\tau}(\alpha, \beta) \in L_t$.

In our compositional grammar I introduce the negative quantifier by means of the function F_3 taking two expressions of type et to a truth value. In this section, we limit our attention to the monadic negative quantifier. The more complex quantifiers will be introduced in Section 4.3.3.

The functions involved in deriving complex expressions in L_τ are given in DEFINITION 4.7, where $\tau, \tau' \in Type$ and $i \in \mathbb{N}$. Note that in a language that makes exclusive use of functional types (see DEFINITION 4.3) NO in DEFINITION 4.7.3 corresponds to the GQT functional representation of the negative quantifier NO (Section 2.1). Its syncategorematic representation should be read as application from left to right, i.e. as $(NO(\alpha))(\beta)$.

Definition 4.7 Syntactic operations

1. $F_1 : L_t \to L_t$, *where* $F_1(\alpha) = \neg\alpha$,
2. $F_2 : L_t \times L_t \to L_t$, *where* $F_2(\alpha, \beta) = [\alpha \wedge \beta]$,
3. $F_3 : L_{et} \times L_{et} \to L_t$, *where* $F_3(\alpha, \beta) = NO(\alpha)(\beta)$,
4. $F_{4:\tau':\tau} : L_{\tau'\tau} \times L_{\tau'} \to L_\tau$, *where* $F_{4:\tau':\tau}(\alpha, \beta) = [[\alpha](\beta)]$,
5. $F_{5:\tau':\tau:i} : L_\tau \to L_{\tau'\tau}$, *where* $F_{5:\tau':\tau:i}(\alpha) = \lambda v_{i,\tau'}.\alpha$,
6. $F_{6:\tau} : L_\tau \times L_\tau \to L_t$, *where* $F_{6:\tau}(\alpha, \beta) = [\alpha = \beta]$.

The logical language L includes the set of expressions in the syntactic algebra $\langle (L_\tau)_{\tau \in Type}, (F_\gamma)_{\gamma \in \Gamma} \rangle$ with F_γ as in DEFINITION 4.7, where for every $\tau, \tau' \in Type$ and $i \in \mathbb{N}$, Γ = {1, 2, 3, 4:τ':τ, 5:τ':τ:i, 6:τ}. We now turn to the semantic algebra in which we interpret the syntactic expressions of L.

The interpretation of L An *interpretation* of the language L is based on some non-empty set E, the *domain of individuals*. We define the following domains of objects:

Definition 4.8 Domains of objects

1. $D_{E,e} = E$,
2. $D_{E,t} = \{0, 1\}$, *and*
3. $D_{E,\tau'\tau} = D_{E,\tau}^{D_{E,\tau'}}$.

$D_{E,\tau'\tau}$ is thus the set of functions from $D_{E,\tau'}$ to $D_{E,\tau}$.

Definition 4.9 Frame
For a domain of individuals E*, for every* $\tau \in Type$*, we define a frame* $\mathcal{F}$ *as the family of domains* $D_{E,\tau}$ *indexed by the types:*

$$\mathcal{F} = (D_{E,\tau})_{\tau \in Type}.$$

Definition 4.10 Model
Given a set of constants $Const$ and a set of individuals E,

a model is a pair $M = \langle \mathcal{F}, Int \rangle$, such that
$\mathcal{F}$ is a frame, and
Int is a function from $Const$ to $\mathcal{F}$ such that
for each $c_\tau \in Const$, $Int(c) \in D_{E,\tau}$.

Constants will be interpreted by means of the function Int in DEFINITION 4.10 and variables will be assigned values in the domain by means of the assignment function a defined below:

Definition 4.11 Variable assignment
A variable assignment is a function $a : Var_\tau \rightarrow D_{E,\tau}$, such that

for each $v_{i,\tau}$, $a(v_{i,\tau}) \in D_{E,\tau}$.

Ass is the set of all variable assignments:

$Ass = \{a|$ *for each* $i \in \mathbb{N}$, *for each* $\tau \in Type$, $a(v_{i,\tau}) \in D_{E,\tau}\}$.

Now we can define the way we interpret the terms in L:

Definition 4.12 Interpretation of L in M
The interpretation of an expression α_τ in a model M is given by the function $in_M(\alpha)$[15] *from variable assignments into the set $D_{E,\tau}$, as follows:*

1. *for each $\alpha \in Var_\tau$, $in_M(\alpha) = \{\langle a, d\rangle | a \in Ass$ and $d = a(\alpha)\}$,*
2. *for each $\alpha \in Const_\tau$, $in_M(\alpha) = \{\langle a, d\rangle | a \in Ass$ and $d = Int(\alpha)\}$,*
3. $in_M(F_1(\alpha)) = G_{M,1}(in_M(\alpha))$,
4. $in_M(F_2(\alpha, \beta)) = G_{M,2}(in_M(\alpha), in_M(\beta))$
5. $in_M(F_3(\alpha, \beta)) = G_{M,3}(in_M(\alpha), in_M(\beta))$,
6. $in_M(F_{4:\tau':\tau}(\alpha, \beta)) = G_{M,4:\tau':\tau}(in_M(\alpha), in_M(\beta))$,
7. $in_M(F_{5:\tau':\tau:i}(\alpha)) = G_{M,5:\tau':\tau:i}(in_M(\alpha))$,
8. $in_M(F_{6:\tau}(\alpha, \beta)) = G_{M,6:\tau}(in_M(\alpha), in_M(\beta))$.

The functions G introduced in DEFINITION 4.12 are given in DEFINITION 4.13:[16]

Definition 4.13 Semantic operations

1. $G_{M,1} : D_{E,t}^{Ass} \rightarrow D_{E,t}^{Ass}$, *where* $G_{M,1}(\phi) =$
$\{\langle a, 1\rangle | \langle a, 0\rangle \in \phi\} \cup$
$\{\langle a, 0\rangle | \langle a, 1\rangle \in \phi\}$,

[15] Note that the function in_M corresponds to the interpretation function $[\![ ]\!]$ which we used in the GQT presentation in Section 2.1.
[16] For $\tau \in Type$, $D_{E,\tau}^{Ass}$ is the set of functions from Ass (the set of variable assignments) to the domain $D_{E,\tau}$.

2. $G_{M,2} : D^{Ass}_{E,t} \times D^{Ass}_{E,t} \to D^{Ass}_{E,t}$, *where* $G_{M,2}(\phi, \psi) =$
$\{\langle a, 1\rangle | \langle a, 1\rangle \in \phi$ *and* $\langle a, 1\rangle \in \psi\} \cup$
$\{\langle a, 0\rangle | \langle a, 0\rangle \in \phi$ *or* $\langle a, 0\rangle \in \psi\}$

3. $G_{M,3} : D^{Ass}_{E,et} \times D^{Ass}_{E,et} \to D^{Ass}_{E,t}$, *where* $G_{M,3}(\phi, \psi) =$
$\{\langle a, 1\rangle |$ *for every* $d \in D_{E,e} : \langle a, 0\rangle \in \phi(d)$ *or* $\langle a, 0\rangle \in \psi(d)\} \cup$
$\{\langle a, 0\rangle |$ *for some* $d \in D_{E,e} : \langle a, 1\rangle \in \phi(d)$ *and* $\langle a, 1\rangle \in \psi(d)\}$

4. $G_{M,4:\tau':\tau} : D^{Ass}_{E,\tau'\tau} \times D^{Ass}_{E,\tau'} \to D^{Ass}_{E,\tau}$, *where* $G_{M,4:\tau':\tau}(\phi, \psi) =$
$\{\langle a, f(d)\rangle | \langle a, f\rangle \in \phi$ *and* $\langle a, d\rangle \in \psi\}$

5. $G_{M,5:\tau':\tau:i} : D^{Ass}_{E,\tau} \to D^{Ass}_{E,\tau'\tau}$, *where* $G_{M,5:\tau':\tau:i}(\phi) =$
$\{\langle a, f\rangle |$ *for every* $d \in D_{E,\tau'} : \langle a[v_{i,\tau'}/d], f(d)\rangle \in \phi\}$[17]

6. $G_{M,6:\tau} : D^{Ass}_{E,\tau} \times D^{Ass}_{E,\tau} \to D^{Ass}_{E,t}$, *where* $G_{M,6:\tau}(\phi, \psi) =$
$\{\langle a, 1\rangle | \langle a, d\rangle \in \phi$ *and* $\langle a, d'\rangle \in \psi$ *and* $d = d'\} \cup$
$\{\langle a, 0\rangle | \langle a, d\rangle \in \phi$ *and* $\langle a, d'\rangle \in \psi$ *and* $d \neq d'\}$

Compositionality In order to interpret the language L, DEFINITION 4.8 to DEFINITION 4.13 build the semantic algebra in which the syntactic algebra $\langle (L_\tau)_{\tau \in Type}, (F_\gamma)_{\gamma \in \Gamma}\rangle$ can be interpreted. Let us call the semantic algebra $\langle (S_\tau)_{\tau \in Type}, (G_{M,\gamma})_{\gamma \in \Gamma}\rangle$. The set of semantic objects S_τ is identical to our domain of objects $D_{E,\tau}$ given in DEFINITION 4.8. They are derived as in DEFINITION 4.12, where the functions Int and a assigning meaning to constants and variables derive the basic semantic objects, and the semantic operations $G_{M,i}$ derive the complex semantic objects. The function in_M which assigns meaning to the expressions in L is given by DEFINITION 4.12 as a homomorphism between the syntactic algebra $\langle (L_\tau)_{\tau \in Type}, (F_\gamma)_{\gamma \in \Gamma}\rangle$ and the semantic algebra $\langle (S_\tau)_{\tau \in Type}, (G_\gamma)_{M,\gamma \in \Gamma}\rangle$, since as required by the principle of compositionality in (224) for every n-place syntactic operation F_i there is an n-place semantic operation $G_{M,i}$, such that for every n-sequence of syntactic expressions $\alpha_1, ..., \alpha_n$, $in_M(F_i(\alpha_1, ..., \alpha_n)) = G_{M,i}(in_M(\alpha_1), ..., in_M(\alpha_n))$. In our case, $n \in \{1, 2\}$, so we only have unary and binary syntactic and semantic operations. Thus the interpretation of the logical language L is done compositionally.

4.3.2.2 The natural language

In order to interpret a natural language fragment we must *translate* it into the logical language. That is, the natural language fragment must be defined as a syntactic algebra $\langle (R_c)_{c \in C}, (H_\delta)_{\delta \in \Delta}\rangle$ which can then be rewritten in the logical algebra $\langle (L_\tau)_{\tau \in Type}, (F_\gamma)_{\gamma \in \Gamma}\rangle$ by means of a homomorphic function $tr(anslation)$. On the basis of the homomorphism in_M between the syntactic and the semantic logical algebras, the natural language algebra can be compositionally interpreted in the logical semantic algebra $\langle (S_\tau)_{\tau \in Type}, (G_\gamma)_{M,\gamma \in \Gamma}\rangle$. This is because the composition of the two homomorphisms $in_M \circ tr$ is also a homomorphism from the natural language algebra to the logical semantic algebra.

Let us first describe the fragment of Romanian that we want to interpret. The Romanian expressions in the set R belong to *syntactic categories* $c \in C$ and each c is associated with a type of L via the function $\sigma(c)$:

[17] The expression $a[v_{i,\tau}/d]$ stands for the variable assignment a' such that: 1) $a'(v_{i,\tau}) = d$ and 2) $a'(v_{i',\tau'}) = a(v_{i',\tau'})$ if $i' \neq i$ or $\tau' \neq \tau$.

(227) *Syntactic categories*

c	*description*	$\sigma(c)$
S	sentence	t
NP	noun phrase	(et)t
CN	common noun	et
IV	intransitive verb	et
TV	transitive verb	e(et)
Det	determiner	(et)((et)t)

As we will see below, to combine two linguistic expressions using logical operations, we sometimes need to enable a syntactic category c to correspond to more than one logical type. For instance, I will show that the type $e(et)$ of a transitive verb must be "shifted" or "raised" to the type $((et)t)(et)$ so that it can combine with an NP of type $(et)t$ by functional application (see the discussion after DEFINITION 4.17).

Every lexical expression α of category c is assigned one *lexical translation* $lextr(\alpha)$, an expression of type $\sigma(c)$ in L. See the lexical expressions of our fragment in (228):

(228) *Lexical translations*

α	*category of* α	$lextr(\alpha)$
student	CN	$\lambda x_1.student'(x_1)$
carte	CN	$\lambda x.book'(x)$
vorbi	IV	$\lambda x.talk'(x)$
citi	TV	$\lambda x_2\lambda x_1.read'(x_1, x_2)$
niciun	Det	$\lambda P\lambda Q.NO(\lambda x_1.P(x_1))(\lambda x_2.Q(x_2))$
nicio	Det	$\lambda P\lambda Q.NO(\lambda x_1.P(x_1))(\lambda x_2.Q(x_2))$

The syntax of the fragment is made up of two sets: the set of syntactic terms of category c $(S_c)_{c\in C}$ and the set of Romanian expressions of category c $(R_c)_{c\in C}$, as defined below for $C = \{$S, NP, CN, IV, TV, Det$\}$. The difference between the two sets will be addressed below.

Definition 4.14 The syntax of the fragment

For each $c \in C$, S_c and R_c are the smallest sets such that:

1. a. *if α of category c appears in (228), then $\lfloor\alpha_c\rfloor \in S_c$*
   b. *if $\alpha \in S_{NP}$ and $\beta \in S_{IV}$, then $\lfloor\underline{H}_1\alpha\beta\rfloor \in S_S$,*
   c. *if $\alpha \in S_{Det}$ and $\beta \in S_{CN}$, then $\lfloor\underline{H}_2\alpha\beta\rfloor \in S_{NP}$,*
   d. *if $\alpha \in S_{TV}$ and $\beta \in S_{NP}$, then $\lfloor\underline{H}_3\alpha\beta\rfloor \in S_{IV}$.*
2. a. *if α of category c appears in (228), then $\alpha_c \in R_c$*
   b. *if $\alpha \in R_{NP}$ and $\beta \in R_{IV}$, then $H_1(\alpha, \beta) \in R_S$,*

c. *if* $\alpha \in R_{Det}$ *and* $\beta \in R_{CN}$*, then* $H_2(\alpha, \beta) \in R_{NP}$*,*

d. *if* $\alpha \in R_{TV}$ *and* $\beta \in R_{NP}$*, then* $H_3(\alpha, \beta) \in R_{IV}$*.*

In DEFINITION 4.14.1, $\underline{H}_i$ only stands for symbols. It refers to complex syntactic terms viewed at an abstract level as syntactic units available in natural languages in general. By contrast, the functions H_i are defined as operations on strings within a particular natural language, so they build the complex Romanian expressions in our algebra:

Definition 4.15 Operations on strings

1. $H_1 : R_{NP} \times R_{IV} \rightarrow R_S$*, where* $H_1(\alpha, \beta) = \alpha\beta$*,*
2. $H_2 : R_{Det} \times R_{CN} \rightarrow R_{NP}$*, where* $H_2(\alpha, \beta) = \alpha\beta$*,*
3. $H_3 : R_{TV} \times R_{NP} \rightarrow R_{IV}$*, where* $H_3(\alpha, \beta) = \alpha\beta$*.*

To better understand the difference between $\underline{H}_i$ and H_i, take for instance the function $\underline{H}_2$ which combines a Det with a CN into an NP. In Romanian, it would derive the syntactic term $\lfloor \underline{H}_2 \lfloor niciun \rfloor \lfloor student \rfloor \rfloor$ and in English, the same function would derive $\lfloor \underline{H}_2 \lfloor no \rfloor \lfloor student \rfloor \rfloor$. However, only the former will be correlated with a Romanian expression (*niciun student*) derived by means of the function H_2 which is defined on $(R_c)_{c \in C}$, the set of Romanian expressions. The syntactic term $\lfloor \underline{H}_2 \lfloor no \rfloor \lfloor student \rfloor \rfloor$ will have to be correlated to a function H_i different from H_2, since it will be defined on the set of English expressions. This correlation between syntactic terms and natural language expressions is done by a function called *evaluation* which is given in DEFINITION 4.16 for our Romanian fragment. It evaluates a given syntactic term in $(S_c)_{c \in C}$ as a Romanian expression in $(R_c)_{c \in C}$:

Definition 4.16 The evaluation function *ev*
The function ev*:* $(S_c)_{c \in C} \rightarrow (R_c)_{c \in C}$ *evaluates each syntactic term in* S_c *as a Romanian expression in* R_c*:*

1. $ev(\lfloor \alpha_c \rfloor) = \alpha$*,*
2. $ev(\lfloor \underline{H}_1 \alpha\beta \rfloor) = H_1(ev(\alpha), ev(\beta))$*,*
3. $ev(\lfloor \underline{H}_2 \alpha\beta \rfloor) = H_2(ev(\alpha), ev(\beta))$*,*
4. $ev(\lfloor \underline{H}_3 \alpha\beta \rfloor) = H_3(ev(\alpha), ev(\beta))$*.*

We now define the translation function between the Romanian algebra and the logical syntactic algebra:

Definition 4.17 Translation
Each syntactic term $\alpha \in S_c$ *is assigned a translation* $tr(\alpha) \in L_{\sigma(c)}$*:*

1. $tr(\lfloor \alpha_c \rfloor) = lextr(\alpha)$ *for* α *of category* c *in (228)*
2. $tr(\lfloor \underline{H}_1 \alpha\beta \rfloor) = F_{4:et:t}(tr(\alpha), tr(\beta))$*,*
3. $tr(\lfloor \underline{H}_2 \alpha\beta \rfloor) = F_{4:et:(et)t}(tr(\alpha), tr(\beta))$*,*

4. $tr(\lfloor \underline{H}_3 \alpha\beta \rfloor) = F_{4:((et)t):et}(tr(\alpha), tr(\beta))$.

For the translation of the Romanian expressions in this fragment we only need the function F_4 in DEFINITION 4.7.[18] I repeat below the three instantiations of the function F_4 as used in DEFINITION 4.17:

(229) a. $F_{4:et:t} : L_{(et)t} \times L_{et} \rightarrow L_t$, where $F_{4:et:t}(\alpha, \beta) = [[\alpha](\beta)]$,
b. $F_{4:et:(et)t} : L_{(et)((et)t)} \times L_{et} \rightarrow L_{(et)t}$, where $F_{4:et:(et)t}(\alpha, \beta) = [[\alpha](\beta)]$,
c. $F_{4:((et)t):et} : L_{((et)t)(et)} \times L_{(et)t} \rightarrow L_{et}$, where $F_4(\alpha, \beta) = [[\alpha](\beta)]$.

A remark is in order here with respect to DEFINITION 4.17.4 and (229c). According to DEFINITION 4.16.4, $\underline{H}_3$ is evaluated in terms of the syntactic operation H_3. H_3 combines two Romanian expressions of categories TV and NP into an expression of category IV. The translation function associated with $\underline{H}_3$ and thus indirectly with H_3 is $F_{4:((et)t):et}$. This translation function should combine an expression of type $e(et)$ (the TV) with one of type $(et)t$ (the NP) into one of type et (the IV). But the former two logical types do not match, so none of the syntactic functions in DEFINITION 4.7 can apply to the two expressions. To cope with this problem, the type $e(et)$ of the TV must be "raised" to the type $((et)t)(et)$, so that its argument matches the type $(et)t$ of the NP. Given the new type $((et)t)(et)$ of the TV, the syntactic function $F_{4:((et)t):et}$ may be applied and the two logical expressions representing the TV and the NP are combined into an expression of type et as in (229c). The raising mechanism from $e(et)$ to $((et)t)(et)$ is provided by *Argument Raising* (AR), a type shifting rule. In DEFINITION 4.18, I give AR and highlight in bold the type that undergoes raising and the corresponding variable.

Definition 4.18 Argument Raising (AR)
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 $(a_1(...(\mathbf{a_i}(...(a_n b)))))$
then β is some term
$\lambda x_{1,a_1}...\lambda \mathbf{X}_{\mathbf{i},(\mathbf{a_i}\mathbf{b})\mathbf{b}}...\lambda x_{n,a_n}.\mathbf{X}(\lambda \mathbf{x}_{\mathbf{i},\mathbf{a_i}}.\alpha(x_1)...(\mathbf{x_i})...(x_n))$

Hendriks (1993) makes type shifting rules available for the translation mechanism by allowing a syntactic category c to be assigned not only one logical type but a set of types.[19] This procedure is called *flexible type assignment*. In our grammar fragment, it is important for the category TV to be assigned a set of two logical types: $e(et)$ (i.e. σ(TV)) and $((et)t)(et)$, which is obtained by applying AR to the first argument of σ(TV).[20] Thus words of certain categories are assigned multiple logical types. In Hendriks's system, AR is represented as a new syntactic operation on strings, say H_{AR}, whose role is to only change the type of the natural language expression. Thus in our fragment an operation $H_{AR:1}$ would apply to *citi* $\in R_{e(et)}$ and produce *citi* $\in R_{(e(et))(et)}$.

As a consequence of the flexible type assignment to syntactic categories, every Romanian expression α is associated with a set of translations $Tr(\alpha)$. This set consists of the syntactic terms whose evaluation coincides with α:

Definition 4.19 Translations set
If $\alpha \in R_c$, then $Tr(\alpha) = \{tr(\gamma) | \gamma \in S_c$ and $ev(\gamma) = \alpha\}$

[18]But see Hendriks (1993) for a fragment of English which is syntactically more complex and makes use of more logical operations than the ones given in DEFINITION 4.7.

[19]For more details, the reader is referred to Hendriks (1993, Ch. 2, Sec. 5).

[20]However, in principle a type shifting rule can be applied more than once.

Given that AR does not change the expression at all, the argument raised translation of an expression will also be included in the set of its translations. To illustrate this with an example, the transitive verb *citi* given in (228) will receive the following two translations in our fragment:[21]

(230) The translations set for *citi*

$$Tr(\textit{citi}) = \{tr(\lfloor citi_{TV} \rfloor), tr(\lfloor \underline{H}_{AR:1} \lfloor citi_{TV} \rfloor \rfloor)\}$$
$$= \{\lambda x_{2,e} \lambda x_{1,e}.read'(x_1, x_2), \lambda X_{2,(et)t} \lambda x_{1,e}.X_2(\lambda x_{2,e}.read'(x_1, x_2))\}$$

In order to interpret the Romanian expressions, we compose the two functions tr and in_M, the former relating the Romanian expression to a logical formula and the latter assigning an interpretation to the logical formula. Syntactic terms in S_c and Romanian expressions in R_c are assigned meaning according to the definition below:

Definition 4.20 Interpretation of natural language expressions
For each $\alpha \in S_c$, the interpretation of α is given by

$$in_M(tr(\alpha))$$

An expression $\alpha \in R_c$ can be associated with a set of interpretations in the model M:

$$\{in_M(\beta) | \beta \in Tr(\alpha)\}$$

4.3.2.3 Compositional interpretation

Let us summarize the algebras involved in a compositional interpretation of our natural language fragment in order to understand the entire procedure. We defined a logical language L which we interpreted compositionally, as a homomorphism in_M between the syntactic algebra $\langle (L_\tau)_{\tau \in Type}, (F_\gamma)_{\gamma \in \Gamma} \rangle$ and the semantic one $\langle (S_\tau)_{\tau \in Type}, (G_{M,\gamma})_{\gamma \in \Gamma} \rangle$ (Section 4.3.2.1). We described a fragment of Romanian as another syntactic algebra $\langle (R_c)_{c \in C}, (H_\delta)_{\delta \in \Delta} \rangle$ which we want to interpret via the logical language L. To do that it is enough to reformulate the Romanian algebra in terms of the syntactic algebra, that is, to define a translation function tr between them as a homomorphism, and the Romanian algebra will be indirectly assigned an interpretation in the semantic algebra interpreting L.

Let us concentrate on the homomorphism tr between the Romanian algebra and the logical one. In DEFINITION 4.14 we defined two sets that form the syntax of the natural language fragment: S_c, the set of syntactic terms of category c, and R_c, the set of Romanian expressions of category c. The relation between the two sets and their corresponding sets of operations is regulated by the evaluation function ev given in DEFINITION 4.16. When we speak of interpreting the Romanian algebra, we speak of the algebra based on the set R_c of Romanian expressions (i.e. $\langle (R_c)_{c \in C}, (H_\delta)_{\delta \in \Delta} \rangle$). However, the translation function tr in DEFINITION 4.17 which establishes a homomorphism between the Romanian and the logical algebras is defined on the set S_c of syntactic terms. This allows us to define the translation function at the abstract level of syntactic terms and not for each particular Romanian expression. Thus a translation function $tr(\lfloor \underline{H}_i(\alpha\beta) \rfloor) = F_j(tr(\lfloor \alpha \rfloor), tr(\lfloor \beta \rfloor))$ for two given operations $\underline{H}_i$ and F_j may be employed in the translation of several natural languages, but there will be a different H_i for each language. This is because H_i is defined on the set of the expressions of each language (see also the discussion under DEFINITION 4.14 and DEFINITION 4.15 above).

There is an important difference between logical and natural language algebras. While the terms in the former can be analyzed in a unique way, this does not hold of the expressions in the latter. This

[21] This example will be used in Section 4.3.2.4.

is true of natural language in general, although our small Romanian fragment does not contain ambiguity. Ambiguity appears very often in English, where an expression like *talk* is ambiguous between the categories IV and CN (it can be both an intransitive verb and a noun). This ambiguity should appear in the lexicon already. The expression *Mary walks and talks fast* is syntactically ambiguous between [*Mary* [[*walks and talks*] *fast*]] and [*Mary* [*walks and* [*talks fast*]]]. The difference between the two would be expressed in the order in which the corresponding functions H_i are applied (see Hendriks (1993, p. 140) for a full explanation within an English fragment). So natural language algebras are *syntactically ambiguous*, while logical algebras are *unambiguous* because they are defined that way.[22] As a consequence of the syntactic ambiguity, when translating a natural language algebra into a logical algebra, one cannot speak of the interpretation of an expression, but only of the interpretation of that expression with respect to its *category* and its *derivational history*.

To refer to the category and the derivational history of an expression, Hendriks introduces the notion of a syntactically unambiguous *term algebra*. To simplify the discussion, I will not go into details on term algebras but the reader is referred to Hendriks (1993, p. 141) for definitions and a detailed explanation. If we name the Romanian algebra $A = \langle (R_c)_{c\in C}, (H_\delta)_{\delta\in\Delta}\rangle$, its corresponding term algebra is $T_{A,K} = \langle (T_{A,K,c})_{c\in C}, (H^T_\delta)_{\delta\in\Delta}\rangle$. K is the set of the lexical expressions in (228), as given in (231). The operations $(H^T_\delta)_{\delta\in\Delta}$ on $(T_{A,K,c})_{c\in C}$ are defined in terms of $(\underline{H}_\delta)_{\delta\in\Delta}$ in DEFINITION 4.14.1 (see (232)):

(231) The set K of lexical expressions

$K = \{K_S, K_{NP}, K_{CN}, K_{IV}, K_{TV}, K_{Det}\}$,
$K_S = K_{NP} = \emptyset$, $K_{CN} = \{$*student, book*$\}$,
$K_{IV} = \{$*vorbi*$\}$, $K_{TV} = \{$*citi*$\}$,
$K_{Det} = \{$*niciun, nicio*$\}$.

(232)
a. $H^T_1 : T_{A,K,NP} \times T_{A,K,IV} \to T_{A,K,S}$, where $H^T_1(\alpha,\beta) = \lfloor \underline{H}_1\alpha\beta \rfloor$,
b. $H^T_2 : T_{A,K,Det} \times T_{A,K,CN} \to T_{A,K,NP}$, where $H^T_2(\alpha,\beta) = \lfloor \underline{H}_2\alpha\beta \rfloor$,
c. $H^T_3 : T_{A,K,TV} \times T_{A,K,NP} \to T_{A,K,IV}$, where $H^T_3(\alpha,\beta) = \lfloor \underline{H}_3\alpha\beta \rfloor$.

To assign meaning to the expressions in the Romanian algebra $A = \langle (R_c)_{c\in C}, (H_\delta)_{\delta\in\Delta}\rangle$, we have to assign meaning to the expressions in the Romanian term algebra $T_{A,K} = \langle (T_{A,K,c})_{c\in C}, (H^T_\delta)_{\delta\in\Delta}\rangle$ which is syntactically unambiguous and keeps track of the category and the derivational history of the Romanian expressions in R.

Now the translation procedure is straightforward. The homomorphism tr defined in DEFINITION 4.17 associates a logical expression in $(L_\tau)_{\tau\in Type}$ to every Romanian syntactic term (lexical or derived). Each operation $\underline{H}_i$ is translated into a logical operation F_j and once we have the translation of $\underline{H}_i$, we can get the translation of its corresponding operation H^T_i in the term algebra. This way, the elements of the term algebra $T_{A,K}$ receive a logical translation in the algebra $\langle (L_\tau)_{\tau\in Type}, (F_\gamma)_{\gamma\in\Gamma}\rangle$, where they can be assigned meaning. The elements of the term algebra are evaluated as elements of the Romanian algebra $A = \langle (R_c)_{c\in C}, (H_\delta)_{\delta\in\Delta}\rangle$ via the evaluation function ev defined in DEFINITION 4.16. So the logical translations and their corresponding meanings can be related to the Romanian expressions.

A schema of the three algebras that we described here and their interaction towards a compositional interpretation of a natural language fragment is given in FIGURE 4.1. Since the composition of two homomorphisms is also a homomorphism (see Hendriks (1993, p. 145) for a proof), $tr \circ in_M$

[22] See Hendriks (1993, p. 140) for the definition of a *free algebra*, in our terms an *unambiguous* algebra.

$$T_{A,K} = \langle (T_{A,K,c})_{c\in C}, (H^T_\delta)_{\delta\in\Delta}\rangle$$
$$\downarrow tr$$
$$B = \langle (L_\tau)_{\tau\in Type}, (F_\gamma)_{\gamma\in\Gamma}\rangle$$
$$\downarrow in_M$$
$$S = \langle (S_\tau)_{\tau\in Type}, (G_\gamma)_{\gamma\in\Gamma}\rangle$$

Figure 4.1: The algebras involved in a compositional interpretation (*simplified*)

$$\begin{array}{ll}
 & T_{A,K} = \langle (T_{A,K,c})_{c\in C}, (H^T_\delta)_{\delta\in\Delta}\rangle \\
 & \downarrow tr \\
B = \langle (L_\tau)_{\tau\in Type}, (F_\gamma)_{\gamma\in\Gamma}\rangle & \Pi(B) \\
\downarrow \mathcal{I} & \downarrow \mathcal{I}^{MP} \\
\mathcal{S} = \langle (\mathcal{I}_\tau)_{\tau\in Type}, (\mathcal{G}_\gamma)_{\gamma\in\Gamma}\rangle & \Pi(\mathcal{S}^{MP})
\end{array}$$

Figure 4.2: The algebras involved in a compositional interpretation (Hendriks (1993, p.176))

is a homomorphism, so the interpretation of the Romanian term algebra in the semantic algebra $\langle (S_\tau)_{\tau\in Type}, (G_\gamma)_{\gamma\in\Gamma}\rangle$ is compositional.

Before closing this section, let me raise one further point which is fundamental for the correct understanding of the principle of compositionality, although it is of minor importance for our present purposes. It relates to the schema in FIGURE 4.1 which is only an informal simplified version of the exhaustive schema in FIGURE 4.2. There are only two algebras that the two schemas have in common: the syntactic algebra ($T_{A,K}$) and the logical algebra ($B = \langle (L_\tau)_{\tau\in Type}, (F_\gamma)_{\gamma\in\Gamma}\rangle$).

One difference between FIGURE 4.1 and FIGURE 4.2 concerns the logical algebra B which is interpreted in the semantic algebra $\mathcal{S}$, a restriction of the algebra S in FIGURE 4.1. Both Janssen (1986) and Hendriks (1993) agree that the semantic algebra must contain only the meanings and the operations necessary in assigning meaning to the logical expressions in the algebra B, so other meanings that may be generated in S are eliminated and thus, the result is an algebra $\mathcal{S}$. The fact that $\mathcal{S}$ assigns meaning to all and only the logical expressions in B is ensured by the condition of *epimorphicity*[23] on the *interpretation* function $\mathcal{I}$ between B and $\mathcal{S}$. The epimorphism $\mathcal{I}$ is a function from models to interpretations in models such that $\mathcal{I}(\alpha)(M) = in_M(\alpha)$ for all models M.

The second difference concerns the translation procedure for the syntactic algebra $T_{A,K}$ which, according to Janssen (1986) and Hendriks (1993), is not done directly as in FIGURE 4.1, but via another algebra derived from the logical algebra B. In Hendriks' formalization this new algebra is $\Pi(B)$, the *polynomial closure* of the algebra B (see Hendriks (1993, Ch. 2, Sec. 3.1)). Moreover, the compositional interpretation of a syntactic algebra must accommodate a set of *meaning postulates* MP which are often necessary to formulate semantic relations between linguistic lexical expressions (see Montague (1970)). Thus $\mathcal{I}^{MP}$, a restriction of $\mathcal{I}$, is an epimorphism from the logical algebra B to the semantic algebra $\mathcal{S}^{MP}$ in which the meaning postulates are true.

[23] An epimorphism is a surjective homomorphism.

The translation of a term algebra often requires additional syntactic operations in the defined logical algebra. The algebra $\Pi(B)$ in FIGURE 4.2 is an extension of the algebra B which includes all the operations unavailable in B, but necessary for the translation of $T_{A,K}$. This extension is formulated in such a way that the epimorphism $\mathcal{I}^{MP}$ still holds between $\Pi(B)$ and $\Pi(\mathcal{S}^{MP})$. Thus $\Pi(\mathcal{S}^{MP})$ is the extension of the algebra $\mathcal{S}^{MP}$ which additionally provides all and only those meanings and semantic operations that correspond to the logical expressions and operations introduced in $\Pi(B)$ besides the ones in B. With these revisions, the composition $tr \circ \mathcal{I}^{MP}$ of the translation homomorphism tr and the interpretation epimorphism $\mathcal{I}^{MP}$ is a homomorphism from the syntactic algebra $T_{A,K}$ to the semantic algebra $\Pi(\mathcal{S}^{MP})$ (Hendriks (1993, pp. 169–171)).

As one may have already noticed, the translation of the Romanian fragment described in Section 4.3.2.2 does not require the formulation of meaning postulates and does not necessitate additional operations to those in B. The syntactic operations $(\underline{H}_\delta)_{\delta\in\Delta}$ (and the corresponding $({H^T}_\delta)_{\delta\in\Delta}$ in $T_{A,K}$) are translated by three variations of the function F_4 defined in DEFINITION 4.7 (cf. DEFINITION 4.17). For this reason, for the present Romanian fragment, it is enough to refer to the schema in FIGURE 4.1 as reflecting the compositional mechanism of assigning interpretation.

4.3.2.4 An example

Having shown how the mechanism of compositional interpretation functions theoretically, let us take the natural language example below for illustration. I will show how the meaning of the complex Romanian expression in (233) is derived compositionally. As expected, we will see that the only interpretation that the sentence receives in a compositional grammar is the one with double negation:

(233) Niciun student nu a citit nicio carte.
no student NM has read no book
'No student read no book.' ('Every student read some book.')

Let us start with the lexical expressions that appear in this sentence: *niciun*, *student*, *nu a citit*, *nicio*, *carte*, all elements of the set $(R_c)_{c\in C}$. I further ignore the syntax and the semantics of the NM *nu* for this fragment, so I will take *nu a citit* to be a derivational version of *citi*, just as *a citit* is a derivational version of *citi*. All the lexical expressions are associated with a category and a lexical translation in (228) which I repeat below in a more convenient notation of the variables:

(234) The lexical expressions

niciun$_{Det}$	$\rightsquigarrow$	$\lambda A_{et}\lambda B_{et}.NO(\lambda x_{1,e}.A(x_1))(\lambda x_{2,e}.B(x_2))$
student$_{CN}$	$\rightsquigarrow$	$\lambda x_{3,e}.student'(x_3)$
nicio$_{Det}$	$\rightsquigarrow$	$\lambda C_{et}\lambda D_{et}.NO(\lambda x_{4,e}.C(x_4))(\lambda x_{5,e}.D(x_5))$
carte$_{CN}$	$\rightsquigarrow$	$\lambda x_{6,e}.book'(x_6)$
nu a citit$_{TV}$	$\rightsquigarrow$	$\lambda x_{8,e}\lambda x_{7,e}.read'(x_7, x_8)$

To derive the sentence *Niciun student nu a citit nicio carte* and its interpretation, I use a syntactic tree which is closer to common linguistic representations and thus makes it easier to follow the mechanism of compositional interpretation. In FIGURE 4.3 the Romanian expression appears on top, the corresponding syntactic term underneath, and the logical translation at the bottom. The former two expressions are connected via the evaluation function ev, the latter two by the translation function tr. Each function is labeled with the corresponding definition. The last line represents the logical expression in $(L_\tau)_{\tau\in Type}$ which is the reduced translation of the linguistic expression.

niciun student nu a citit nicio carte

$$\uparrow ev \text{ (D:4.16.2)}$$

$$\lfloor \underline{H}_1 \lfloor \underline{H}_2 \lfloor \textit{niciun}_{Det} \rfloor \lfloor \textit{student}_{CN} \rfloor \rfloor \lfloor \underline{H}_3 \lfloor \textit{nu a citit}_{TV} \rfloor \lfloor \underline{H}_2 \lfloor \textit{nicio}_{Det} \rfloor \lfloor \textit{carte}_{CN} \rfloor \rfloor \rfloor \rfloor$$

$$\downarrow tr \text{ (D:4.17.2)}$$

$$F_{4:et:t}(F_{4:et:(et)t}(tr(\lfloor \textit{niciun}_{Det} \rfloor), tr(\lfloor \textit{student}_{CN} \rfloor)), F_{4:((et)t):et}(tr(\lfloor \textit{nu a citit}_{TV} \rfloor), F_{4:et:(et)t}(tr(\lfloor \textit{nicio}_{Det} \rfloor), tr(\lfloor \textit{carte}_{CN} \rfloor))))$$

$$=$$

$$NO(\lambda x_{1,e}.student'(x_1))(\lambda x_{2,e}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.read'(x_2, x_5)))$$

niciun student

$$\uparrow ev \text{ (D:4.16.3)}$$

$$\lfloor \underline{H}_2 \lfloor \textit{niciun}_{Det} \rfloor \lfloor \textit{student}_{CN} \rfloor \rfloor$$

$$\downarrow tr \text{ (D:4.17.3)}$$

$$F_{4:et:(et)t}(tr(\lfloor \textit{niciun}_{Det} \rfloor), tr(\lfloor \textit{student}_{CN} \rfloor))$$

$$=$$

$$\lambda B_{et}.NO(\lambda x_{1,e}.student'(x_1))(\lambda x_{2,e}.B(x_2))$$

nu a citit nicio carte

$$\uparrow ev \text{ (D:4.16.4)}$$

$$\lfloor \underline{H}_3 \lfloor \textit{nu a citit}_{TV} \rfloor \lfloor \underline{H}_2 \lfloor \textit{nicio}_{Det} \rfloor \lfloor \textit{carte}_{CN} \rfloor \rfloor \rfloor$$

$$\downarrow tr \text{ (D:4.17.4)}$$

$$F_{4:((et)t):et}(tr(\lfloor \textit{nu a citit}_{TV} \rfloor), F_{4:et:(et)t}(tr(\lfloor \textit{nicio}_{Det} \rfloor), tr(\lfloor \textit{carte}_{CN} \rfloor)))$$

$$=$$

$$\lambda x_{7,e}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.read'(x_7, x_5))$$

nu a citit

$$\uparrow ev \text{ (D:4.16.1)}$$

$$\lfloor \textit{nu a citit}_{TV} \rfloor$$

$$\downarrow tr \text{ (228; 234)}$$

$$\lambda x_{8,e} \lambda x_{7,e}.read'(x_7, x_8)$$

$$\downarrow AR_1$$

$$\lambda X_{8,(et)t} \lambda x_{7,e}.X_8(\lambda x_{8,e}.read'(x_7, x_8))$$

nicio carte

$$\uparrow ev \text{ (D:4.16.3)}$$

$$\lfloor \underline{H}_2 \lfloor \textit{nicio}_{Det} \rfloor \lfloor \textit{carte}_{CN} \rfloor \rfloor$$

$$\downarrow tr \text{ (D:4.17.3)}$$

$$F_{4:et:(et)t}(tr(\lfloor \textit{nicio}_{Det} \rfloor), tr(\lfloor \textit{carte}_{CN} \rfloor))$$

$$=$$

$$\lambda D_{et}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.D(x_5))$$

nicio

$$\uparrow ev \text{ (D:4.16.1)}$$

$$\lfloor \textit{nicio}_{Det} \rfloor$$

$$\downarrow tr \text{ (228; 234)}$$

$$\lambda C_{et} \lambda D_{et}.NO(\lambda x_{4,e}.C(x_4))(\lambda x_{5,e}.D(x_5))$$

carte

$$\uparrow ev \text{ (D:4.16.1)}$$

$$\lfloor \textit{carte}_{CN} \rfloor$$

$$\downarrow tr \text{ (228; 234)}$$

$$\lambda x_{6,e}.book'(x_6)$$

Figure 4.3: The translation tree for *Niciun student nu a citit nicio carte*

Let us describe the tree in FIGURE 4.3. As can be noticed from the tree and from our previous discussion in Section 4.3.2.3, the syntactic terms built as in DEFINITION 4.14.1 on the basis of the functions $\underline{H}_i$ mediate the translation between Romanian expressions and logical terms, so they will be considered in parallel with the other two kinds of objects. I start with the complex expression *nicio carte* derived from the two lexical ones *nicio* and *carte*. The corresponding lexical syntactic terms obtained via DEFINITION 4.14.1a are $\lfloor nicio_{Det} \rfloor$ and $\lfloor carte_{CN} \rfloor$. In (228), *nicio* and *carte* appear as lexical expressions, so according to the definition of translation (DEFINITION 4.17.1) they get their translation (via their corresponding syntactic terms) from (228), reformulated in (234). Given the two syntactic categories Det and CN, the function $\underline{H}_2$ in DEFINITION 4.14.1c tells us that they build a syntactic term of category NP. The operation applied to the two expressions is the operation H_2 related to $\underline{H}_2$ via the function ev (DEFINITION 4.16.3). Since H_2 concatenates two expressions of category Det and CN, we obtain the expression *nicio carte* of category NP. This gets its logical translation via the operation $\underline{H}_2$ which is assigned a logical translation by the function tr as specified in DEFINITION 4.17.3. The procedure can be represented as in (235). The NP *nicio carte* is assigned a set containing one translation:

(235) Translation for *nicio carte*

Tr(*nicio carte*) = $\{tr(\lfloor \underline{H}_2 \lfloor nicio_{Det} \rfloor \lfloor carte_{CN} \rfloor \rfloor)\}$ (D:4.19)

$tr(\lfloor \underline{H}_2 \lfloor nicio_{Det} \rfloor \lfloor carte_{CN} \rfloor \rfloor)$

$= F_{4:et:(et)t}(tr(\lfloor nicio_{Det} \rfloor), tr(\lfloor carte_{CN} \rfloor))$ (D:4.17.3)

$= F_{4:et:(et)t}(\lambda C_{et} \lambda D_{et}.NO(\lambda x_{4,e}.C(x_4))(\lambda x_{5,e}.D(x_5)), \lambda x_{6,e}.book'(x_6))$ (234)

$= [\lambda \mathbf{C_{et}} \lambda D_{et}.NO(\lambda x_{4,e}.\mathbf{C}(x_4))(\lambda x_{5,e}.D(x_5))](\lambda \mathbf{x_{6,e}}.\mathbf{book'}(\mathbf{x_6}))$ (D:4.7)

$= \lambda D_{et}.NO(\lambda x_{4,e}.[\lambda \mathbf{x_{6,e}}.book'(\mathbf{x_6})](\mathbf{x_4}))(\lambda x_{5,e}.D(x_5))$ (β-reduction)

$= \lambda D_{et}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.D(x_5))$ (β-reduction)

Tr(*nicio carte*) = $\{\lambda D_{et}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.D(x_5))\}$

In a similar way, we derive the Romanian expression *nu a citit nicio carte* and its translation. This is given in (236). Note that we do not use the basic translation of the transitive verb *nu a citit*, we need to employ one derived by *Argument Raising* as explained after (236):

(236) Translation for *nu a citit nicio carte*

Tr(*nu a citit nicio carte*) = $\{tr(\lfloor \underline{H}_3 \lfloor nu\ a\ citit_{TV} \rfloor \lfloor \underline{H}_2 \lfloor nicio_{Det} \rfloor \lfloor carte_{CN} \rfloor \rfloor \rfloor)\}$ (D:4.19)

$tr(\lfloor \underline{H}_3 \lfloor nu\ a\ citit_{TV} \rfloor \lfloor \underline{H}_2 \lfloor nicio_{Det} \rfloor \lfloor carte_{CN} \rfloor \rfloor \rfloor)$

$= F_{4:((et)t):et}(tr(\lfloor nu\ a\ citit_{TV} \rfloor), tr(\lfloor \underline{H}_2 \lfloor nicio_{Det} \rfloor \lfloor carte_{CN} \rfloor \rfloor))$ (D:4.17.4)

$= F_{4:((et)t):et}(tr(\lfloor nu\ a\ citit_{TV} \rfloor), F_{4:et:(et)t}(tr(\lfloor nicio_{Det} \rfloor), tr(\lfloor carte_{CN} \rfloor)))$ (D:4.17.3)

$= F_{4:((et)t):et}(\lambda X_{8,(et)t} \lambda x_{7,e}.X_8(\lambda x_{8,e}.read'(x_7, x_8)),$
$\lambda D_{et}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.D(x_5)))$ (237; 235)

$= [\lambda \mathbf{X_{8,(et)t}} \lambda x_{7,e}.\mathbf{X_8}(\lambda x_{8,e}.read'(x_7, x_8))]$
$(\lambda \mathbf{D_{et}}.\mathbf{NO}(\lambda \mathbf{x_{4,e}}.\mathbf{book'}(\mathbf{x_4}))(\lambda \mathbf{x_{5,e}}.\mathbf{D}(\mathbf{x_5}))))$ (D:4.7)

$= \lambda x_{7,e}.[\lambda \mathbf{D_{et}}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.\mathbf{D}(x_5))]$
$(\lambda \mathbf{x_{8,e}}.\mathbf{read'}(\mathbf{x_7}, \mathbf{x_8}))$ (β-reduction)

$= \lambda x_{7,e}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.[\lambda \mathbf{x_{8,e}}.read'(x_7, x_8)](\mathbf{x_5}))$ (β-reduction)

$= \lambda x_{7,e}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.read'(x_7, x_5))$ (β-reduction)

Tr(*nu a citit nicio carte*) = $\{\lambda x_{7,e}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.read'(x_7, x_5))\}$

The basic translation of the verb *nu a citit* in (228) and (234) is $\lambda x_{8,e}\lambda x_{7,e}.read'(x_7, x_8)$. But in the tree (governed by the Romanian syntactic terms formation in DEFINITION 4.14.1), it has to combine with the expression *nicio carte* which is of type $(et)t$, so we have a type mismatch since no logical syntactic operation in DEFINITION 4.7 can apply to combine the two expressions. As a solution, we may use the type shifting operation *Argument Raising* in DEFINITION 4.18. AR is lexically available for every translation of the expressions in our fragment and allows deriving other possible translations. For *nu a citit*, AR_1 yields the result in (237) which was used in FIGURE 4.3 and in (236) in order to derive the translation of the complex expression *nu a citit nicio carte*. Given the translation $\lambda X_{8,(et)t}\lambda x_{7,e}.X_8(\lambda x_{8,e}.read'(x_7, x_8))$ of *nu a citit*, the syntactic operation F_4 can now apply it to the translation $\lambda D_{et}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.D(x_5))$ of the NP.

(237) AR_1 for *nu a citit*

$\lambda x_{8,e}\lambda x_{7,e}.read'(x_7, x_8)$

$\downarrow AR_1$

$\lambda X_{8,(et)t}\lambda x_{7,e}.X_8(\lambda x_{8,e}.read'(x_7, x_8))$

In FIGURE 4.3, the NP *niciun student* is derived similarly to the NP *nicio carte* in (235), so I skip to the final expression, the sentence *niciun student nu a citit nicio carte*. Since there is nothing new about its derivation and translation compared to (235) and (236), I give the procedure directly in (238):

(238) Translation for *niciun student nu a citit nicio carte*

Tr(*niciun student nu a citit nicio carte*)

$= \{tr(\lfloor \underline{H}_1 \lfloor \underline{H}_2 \lfloor \textit{niciun}_{Det} \rfloor \lfloor \textit{student}_{CN} \rfloor \rfloor \lfloor \underline{H}_3 \lfloor \textit{nu a citit}_{TV} \rfloor \lfloor \underline{H}_2 \lfloor \textit{nicio}_{Det} \rfloor \lfloor \textit{carte}_{CN} \rfloor \rfloor \rfloor \rfloor)\}$

$tr(\lfloor \underline{H}_1 \lfloor \underline{H}_2 \lfloor \textit{niciun}_{Det} \rfloor \lfloor \textit{student}_{CN} \rfloor \rfloor \lfloor \underline{H}_3 \lfloor \textit{nu a citit}_{TV} \rfloor \lfloor \underline{H}_2 \lfloor \textit{nicio}_{Det} \rfloor \lfloor \textit{carte}_{CN} \rfloor \rfloor \rfloor \rfloor)$

$= F_{4:et:t}(F_{4:et:(et)t}(tr(\lfloor \textit{niciun}_{Det} \rfloor), tr(\lfloor \textit{student}_{CN} \rfloor)), F_{4:((et)t):et}(tr(\lfloor \textit{nu a citit}_{TV} \rfloor),$
$F_{4:et:(et)t}(tr(\lfloor \textit{nicio}_{Det} \rfloor), tr(\lfloor \textit{carte}_{CN} \rfloor))))$ (D:4.17.2; 4.17.3; 4.17.4)

$= F_{4:et:t}(\lambda B_{et}.NO(\lambda x_{1,e}.student'(x_1))(\lambda x_{2,e}.B(x_2)),$
$\lambda x_{7,e}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.read'(x_7, x_5)))$ ($\sim$235; 236)

$= [\lambda \mathbf{B_{et}}.NO(\lambda x_{1,e}.student'(x_1))(\lambda x_{2,e}.\mathbf{B}(x_2))]$
$(\lambda \mathbf{x_{7,e}}.\mathbf{NO}(\lambda \mathbf{x_{4,e}}.\mathbf{book'}(\mathbf{x_4}))(\lambda \mathbf{x_{5,e}}.\mathbf{read'}(\mathbf{x_7}, \mathbf{x_5})))$ (D:4.7)

$= NO(\lambda x_{1,e}.student'(x_1))(\lambda x_{2,e}.$
$[\lambda \mathbf{x_{7,e}}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.read'(\mathbf{x_7}, x_5))](\mathbf{x_2}))$ (β-reduction)

$= NO(\lambda x_{1,e}.student'(x_1))$
$(\lambda x_{2,e}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.read'(x_2, x_5)))$ (β-reduction)

Tr(*niciun student nu a citit nicio carte*)

$= \{NO(\lambda x_{1,e}.student'(x_1))(\lambda x_{2,e}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.read'(x_2, x_5)))\}$

We thus obtain the translation of the sentence (233) as given in FIGURE 4.3.

In order to assign meaning to this sentence, one assigns an interpretation to its logical translation, and thus everything is a matter of interpretation of the logical language L. The translation set of the natural language expression *niciun student nu a citit nicio carte*$_S$ is made up of one element, given in (238). According to DEFINITION 4.20, p. 131, this means that it receives a single interpretation given by the function $in_M(tr($*niciun student nu a citit nicio carte*$_S))$. The value of this function can be calculated in at least two ways. One may apply in_M to the complex translation including all the syntactic functions $(F_\gamma)_{\gamma\in\Gamma}$ as given by the translation from functions $(\underline{H}_\delta)_{\delta\in\Delta}$. This procedure is shown in (239), where the lexical translations will have to be written in terms of functions $(F_\gamma)_{\gamma\in\Gamma}$ which will then be turned into interpretation functions $(G_\gamma)_{\gamma\in\Gamma}$. Alternatively, one may apply in_M directly to the reduced translation which can be restated in terms of the functions $(F_\gamma)_{\gamma\in\Gamma}$ (see (240)). Both procedures yield the same interpretation, but the latter is simpler, since the translation contains fewer operations.

(239) The interpretation of *niciun student nu a citit nicio carte*

$$\begin{aligned}
& in_M(tr(\lfloor \textit{niciun student nu a citit nicio carte}_S \rfloor)) \\
&= in_M(F_{4:et:t}(F_{4:et:(et)t}(tr(\lfloor \textit{niciun}_{Det} \rfloor), tr(\lfloor \textit{student}_{CN} \rfloor)), \\
&\quad F_{4:((et)t):et}(tr(\lfloor \textit{nu a citit}_{TV} \rfloor), F_{4:et:(et)t}(tr(\lfloor \textit{nicio}_{Det} \rfloor), tr(\lfloor \textit{carte}_{CN} \rfloor))))) \quad (238) \\
&= G_{M,4:et:t}(G_{M,4:et:(et)t}(in_M(tr(\lfloor \textit{niciun}_{Det} \rfloor)), in_M(tr(\lfloor \textit{student}_{CN} \rfloor))), \\
&\quad G_{M,4:((et)t):et}(in_M(tr(\lfloor \textit{nu a citit}_{TV} \rfloor)), \\
&\quad G_{M,4:et:(et)t}(in_M(tr(\lfloor \textit{nicio}_{Det} \rfloor)), in_M(tr(\lfloor \textit{carte}_{CN} \rfloor))))) \quad \text{(D:4.12.6)}
\end{aligned}$$

where

$in_M(tr(\lfloor \textit{niciun}_{Det} \rfloor)) = in_M(\lambda A_{et} \lambda B_{et}.NO(\lambda x_{1,e}.A(x_1))(\lambda x_{2,e}.B(x_2))) =$ ... etc.

$in_M(tr(\lfloor \textit{student}_{CN} \rfloor)) = in_M(\lambda x_{3,e}.student'(x_3)) =$ ... etc.

$in_M(tr(\lfloor \textit{nu a citit}_{TV} \rfloor)) = in_M(\lambda X_{8,(et)t} \lambda x_{7,e}.X_8(\lambda x_{8,e}.read'(x_7, x_8))) =$ ... etc.

$in_M(tr(\lfloor \textit{nicio}_{Det} \rfloor)) = in_M(\lambda C_{et} \lambda D_{et}.NO(\lambda x_{4,e}.C(x_4))(\lambda x_{5,e}.D(x_5))) =$ ... etc.

$in_M(tr(\lfloor \textit{carte}_{CN} \rfloor)) = in_M(\lambda x_{6,e}.book'(x_6)) =$ ... etc.

(240) The interpretation of *niciun student nu a citit nicio carte*

$$\begin{aligned}
& in_M(tr(\lfloor \textit{niciun student nu a citit nicio carte}_S \rfloor)) \\
&= in_M(NO(\lambda x_{1,e}.student'(x_1))(\lambda x_{2,e}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.read'(x_2, x_5)))) \\
&= in_M(F_3(F_{5:e:t:1}(F_{4:e:t}(student', x_1)), F_{5:e:t:2}(F_3(F_{5:e:t:4}(F_{4:e:t}(book', x_4)), \\
&\quad F_{5:e:t:5}(F_{4:e:t}(F_{4:e:et}(read', x_5), x_2)))))) \quad \text{(D:4.7)} \\
&= G_{M,3}(G_{M,5:e:t:1}(G_{M,4:e:t}(in_M(student'), in_M(x_1))), \\
&\quad G_{M,5:e:t:2}(G_{M,3}(G_{M,5:e:t:4}(G_{M,4:e:t}(in_M(book'), in_M(x_4))), \\
&\quad G_{M,5:e:t:5}(G_{M,4:e:t}(G_{M,4:e:et}(in_M(read'), in_M(x_5)), in_M(x_2)))))) \quad \text{(D:4.12.5)}
\end{aligned}$$

In (240), the value for the $(G_\gamma)_{\gamma\in\Gamma}$ operations will be inserted from DEFINITION 4.13, p. 126. According to DEFINITION 4.12, the function a (DEFINITION 4.11) will assign an interpretation to the variables x_1, x_2, x_4, x_5 and the function Int (in DEFINITION 4.10) will assign an interpretation to the constants $student', read', book'$. The last expression in (240) contains two semantic functions G_3 which are negative, so the interpretation of the sentence in (233) in this compositional fragment is double negation. In the next section, I will address the possibility of integrating polyadic quantifiers in this fragment, so that we may derive both the double negation and the negative concord reading of this sentence by making use of the flexibility of interpretation that iteration and resumption as polyadic quantifiers can offer.

4.3.3 Iteration and resumption as modes of composition?

Having shown how the principle of compositionality applies in the interpretation of a natural language fragment, in this section I investigate the status of polyadic quantifiers with respect to compositionality. The ultimate goal is to test the feasibility of the suggestion in de Swart and Sag (2002) to give resumption a compositional status similar to that of functional application.

As we will see, this attempt turns out to be impossible for two related reasons concerning: 1) the syntax of polyadic lifts and 2) their high expressive power. First, the syntax of polyadic lifts in general cannot be made compositional with a surface-oriented natural language syntax. From this point of view, iteration is just as non-compositional as resumption. Second, I will show that binary non-iterations, including resumption, have a higher expressive power than any combinations of two monadic quantifiers. This means that their semantics cannot be restated in terms of the semantics of their (monadic) parts, as a compositional interpretation with λ-calculus and functional types requires.

In Section 4.3.3.1, I introduce some modifications of the logical language L to integrate polyadic quantifiers and I show that only iteration can be defined as a compositional function in L. In Section 4.3.3.2, I present the problems that one encounters when trying to make iteration a mode of composition (between the logical and the natural language). In Section 4.3.3.3, I present the general problem that the expressive power of polyadic quantifiers raises with respect to a compositional interpretation entirely based on a functional type theory with λ-calculus.

4.3.3.1 Polyadic quantifiers in L

We saw that iteration can account for double negation readings in Romanian (Section 4.1) and resumption for negative concord (Section 4.2). In this section I propose a precise formulation of iteration and resumption in the logical language L. I focus on the simplest cases with binary quantifiers, so I repeat below DEFINITION 2.10 and DEFINITION 2.16 for binary iteration and binary resumption, respectively:

> **Definition 2.10** (p. 27) Iteration of two type $\langle 1,1\rangle$ quantifiers
> *For* Q_1, Q_2, *quantifiers of type* $\langle 1,1\rangle$, $It(\mathrm{Q}_1,\mathrm{Q}_2)$ *is the type* $\langle 1^2,2\rangle$ *quantifier defined, for any domain* E, *any* $\mathrm{A},\mathrm{B} \subseteq \mathrm{E}$, *any* $\mathrm{R} \subseteq \mathrm{E}^2$, *as:*
> $$It(\mathrm{Q}_1,\mathrm{Q}_2)^{\mathrm{A},\mathrm{B}}(\mathrm{R}) = \mathrm{Q}_1(\mathrm{A}, \{x \in \mathrm{E} |\ \mathrm{Q}_2(\mathrm{B}, \{y \in \mathrm{E} |\ (x,y) \in \mathrm{R}\})\})$$
>
> **Definition 2.16** (p. 32) Binary resumption of type $\langle 1,1\rangle$ quantifiers
> *For a quantifier* Q *of type* $\langle 1,1\rangle$, *given* E *the domain*, $\mathrm{A},\mathrm{B} \subseteq \mathrm{E}$, $\mathrm{R} \subseteq \mathrm{E}^2$, *the polyadic quantifier* $Res^2(\mathrm{Q})$ *of type* $\langle 1^2,2\rangle$ *derived from* Q *is defined as:*
> $$Res^2(\mathrm{Q})^{\mathrm{A},\mathrm{B}}_{\mathrm{E}}(\mathrm{R}) = \mathrm{Q}^{\mathrm{A}\times\mathrm{B}}_{\mathrm{E}^2}(\mathrm{R})$$

As quantifiers of Lindström type $\langle 1^2,2\rangle$ (241), iteration and resumption receive the logical types in (242). The same type is assigned to the binary quantifier $Q^{A\times B}_{E^2}$ which is the value of $Res^2(\mathrm{Q})^{\mathrm{A},\mathrm{B}}_{\mathrm{E}}$ defined above.[24] Since in our grammar fragment we only make use of the quantificational operator NO, I will limit my attention to NO^2. To make Res a mode of composition, we must write the function Res as applying to two distinct quantifiers, just like It (see also the discussion under DEFINITION 4.22). But note that the function Res in (242) is different from Res^2 above, although they both receive a resumptive interpretation of the monadic quantifier(s) to which they apply.

(241) *Correspondence between Lindström types and $Type$*

[24] Note the slight modifications of the GQT notation from our previous discussion (small caps now turned into italic big caps), meant to better suit the notation in this fragment and to distinguish the generalized quantifiers as part of the logical fragment from the pure GQT notions.

$\langle 1 \rangle$ corresponds to $(et)t$;

$\langle 1,1 \rangle$ corresponds to $(et)((et)t)$

$\langle 2 \rangle$ corresponds to $(e(et))t$;

$\langle 1^2, 2 \rangle$ corresponds to $(et)((et)((e(et))t))$

$\langle n \rangle$ corresponds to $\underbrace{(e(...(e}_{n-times}\, t)))t$;

$\langle 1^n, n \rangle$ corresponds to $\underbrace{(et)(...((et)}_{n-times}((\underbrace{e(...(e}_{n-times}\ t)))t)))$

(242)

Polyadic quantifier	Lindström type	$Type$
$It(Q_1)(Q_2)$	$\langle 1^2, 2\rangle$	$(et)((et)((e(et))t))$
$Res(Q_1)(Q_2)$	$\langle 1^2, 2\rangle$	$(et)((et)((e(et))t))$
$Q^{A\times B}_{E^2}$; NO^2	$\langle 1^2, 2\rangle$	$(et)((et)((e(et))t))$

Modifications of the language L Keenan and Westerståhl (1997) speak of iteration and resumption as *polyadic lifts*, that is, as higher-order functions taking, in our case, two monadic quantifiers as arguments and yielding the binary quantifiers $It(Q_1)(Q_2)$ and $Res(Q_1)(Q_2)$. In the logical language L, I defined the negative quantifier as the syntactic operation F_3 (see DEFINITION 4.7). Since this definition is syncategorematic, the quantifier NO alone cannot be selected by It or Res. We need to redefine NO as a logical constant of type $(et)((et)t)$, i.e. to give it a categorematic status (see also Gamut (1991, vol. II, pp. 114–115)). Recall from Section 4.3.2.3 that we have to keep our logical algebra unambiguous. So we eliminate the syntactic operation F_3 and we define the constant NO as in DEFINITION 4.21, where I also redefine the set of constants $Const$ previously given in DEFINITION 4.5. Similarly, NO^2 is defined as a constant of type $(et)((et)((e(et))t))$:

Definition 4.21 $Const$
Let $Const_\tau$ be be defined as follows
$Const_e = \{j\}$,
$Const_{et} = \{student', book', talk'\}$
$Const_{e(et)} = \{read'\}$
$Const_{(et)((et)t)} = \{NO\}$
$Const_{(et)((et)((e(et))t))} = \{NO^2\}$
$Const_\tau = \emptyset$ *for* $\tau \notin \{e, (et), (e(et)), ((et)((et)t)), ((et)((et)((e(et))t)))\}$.

Unlike other constants which may get a different interpretation with respect to each model, the logical constants NO and NO^2 receive the same semantics $Int(NO)$ and $Int(NO^2)$, respectively, in all models, as given in DEFINITION 4.22:

Definition 4.22 *The semantics of NO and NO^2*

1. $in_M(NO) = \{\langle a, d\rangle | a \in Ass$ *and* $d = Int(NO)\}$
$Int(NO) = f \in D_{E,t}^{D_{E,et}^{D_{E,et}}}$ *, such that for* $f_1, f_2 \in D_{et}$,
$(f(f_1))(f_2) = 1$, *iff for every* $d_1 \in D_{E,e}$,

$f_1(d_1) = 0$, *or* $f_2(d_1) = 0$.

2. $in_M(NO^2) = \{\langle a, d\rangle | a \in Ass$ *and* $d = Int(NO^2)\}$

$Int(NO^2) = f \in D_{E,t}^{D_{E,e(et)}^{D_{E,et}^{D_{E,et}}}}$, *such that for* $f_1, f_2 \in D_{et}$, $g \in D_{e(et)}$, $((f(f_2))(f_1))(g) = 1$, *iff for every* $d_1, d_2 \in D_{E,e}$,

$f_1(d_1) = 0$, *or* $f_2(d_2) = 0$, *or* $g(d_1, d_2) = 0$.

Iteration and resumption Similarly to the way we defined NO and NO^2, we could also define It and Res as logical constants of type $((et)((et)t))(((et)((et)t))((et)((et)((e(et))t))))$. But since we are only interested in obtaining $It(Q_1)(Q_2)_{(et)((et)((e(et))t))}$ and $Res(Q_1)(Q_2)_{(et)((et)((e(et))t))}$, and we do not need to make It and Res available for selection by even higher-order functions, we may also define them syncategorematically. Moreover, a syncategorematic definition is in the spirit of de Swart and Sag (2002), who regard *iteration* and *resumption* as two different "modes of composition". In our terms, this means that they should be represented as syntactic operations $(F_\gamma)_{\gamma\in\Gamma}$ (with corresponding semantic operations $(G_\gamma)_{\gamma\in\Gamma}$) in the language L, which would translate corresponding syntactic operations $(\underline{H}_\delta)_{\delta\in\Delta}$ given in the Romanian fragment above.

Let us consider $F_{7:It}$ and $F_{8:Res}$, the two syntactic operations in L that derive $It(Q_1)(Q_2)$ and $Res(Q_1)(Q_2)$. At this point, the set of indices for logical operations in L is $\Gamma = \{1, 2, 4{:}\tau'{:}\tau, 5{:}\tau'{:}\tau{:}i$, $6{:}\tau$, $7{:}It$, $8{:}Res\}$, since we eliminated F_3:

Definition 4.23 Terms in L with iteration and resumption

1. *for each* $\alpha, \beta \in L_{(et)((et)t)}$, $F_{7:It}(\alpha, \beta) \in L_{(et)((et)((e(et))t))}$
2. *for each* $\alpha, \beta \in L_{(et)((et)t)}$, $F_{8:Res}(\alpha, \beta) \in L_{(et)((et)((e(et))t))}$

Definition 4.24 The syntactic operations for iteration and resumption

1. $F_{7:It} : L_{(et)((et)t)} \times L_{(et)((et)t)} \rightarrow L_{(et)((et)((e(et))t))}$, *where* $F_{7:It}(\alpha, \beta) = It(\alpha)(\beta)$
2. $F_{8:Res} : L_{(et)((et)t)} \times L_{(et)((et)t)} \rightarrow L_{(et)((et)((e(et))t))}$, *where* $F_{8:Res}(\alpha, \beta) = Res(\alpha)(\beta)$

As binary quantifiers built on the basis of polyadic lifts, $It(\alpha)(\beta)$ and $Res(\alpha)(\beta)$ are derived by similar syntactic operations in L (see DEFINITION 4.24), but the corresponding semantic operations must yield the interpretations given by DEFINITION 2.10 and DEFINITION 2.16. Considering that in our fragment there is one constant of type $(et)((et)t)$ which is NO, we can only build $It(NO)(NO)$ and $Res(NO)(NO)$.

DEFINITION 2.16 predicts that the semantics of $Res(NO)(NO)$ is the same as the semantics of NO^2, which was given in DEFINITION 4.22.2. This semantic correlation creates an unsolvable problem in defining resumption as a mode of composition: it doesn't allow us to define the semantic operation $G_{M,8:Res}$ such that the interpretation function in_M is a homomorphism between $F_{8:Res}$ and $G_{M,8:Res}$ as required by the principle of compositionality and given in DEFINITION 4.25:

Definition 4.25 in_M for iteration and resumption

1. $in_M(F_{7:It}(\alpha, \beta)) = G_{M,7:It}(in_M(\alpha), in_M(\beta))$

2. $in_M(F_{8:Res}(\alpha, \beta)) = G_{M,8:Res}(in_M(\alpha), in_M(\beta))$

$G_{M,8:Res}$ must be defined in such way that it combines the interpretations of the two constants α and β. Since with resumption, $\alpha = \beta = NO$, consider $in_M(NO) = \langle a, p\rangle$ and $G_{M,8:Res}(in_M(NO), in_M(NO)) = \langle a, q\rangle$. Then $q \in D_{E,t}^{D_{E,e(et)}^{D_{E,et}^{D_{E,et}}}}$ must be defined on the basis of p. According to DEFINITION 2.16, $G_{M,8:Res}(in_M(NO), in_M(NO)) = \langle a, q\rangle$ has the same value as $in_M(NO^2) = \langle a, f\rangle$, with f as in DEFINITION 4.22.2. The value of f depends on the interpretation functions f_1, f_2, g for the restrictions and the nuclear scope of the quantifier NO^2, but it does not use an interpretation function $n \in D_{E,t}^{D_{E,et}^{D_{E,et}}}$ of the monadic quantifier NO as equivalent to the function f in DEFINITION 4.22.1. Thus the value of $G_{M,8:Res}$ depends on the interpretation functions f_1, f_2, g of the two restrictions and the nuclear scope of the quantifier $Res(\alpha)(\beta)$, but not on the functions $in_M(\alpha)$ and $in_M(\beta)$ as DEFINITION 4.25 requires.

Why should meaning assignment be problematic for $Res(NO)(NO)$ and not for NO^2? It is precisely because $Res(NO)(NO)$ is derived by a syntactic operation ($F_{8:Res}$) which combines two parts NO and NO, while NO^2 is a constant (i.e. a syntactic term in itself within which we need not distinguish subparts) and the interpretation function assigns meaning to the whole.[25] For the former, compositionality requires that in_M be a homormophism between $F_{8:Res}$ and $G_{M,8:Res}$, while for the latter, in_M is given directly by the function Int, according to DEFINITION 4.12.2, p. 126.

I conclude at this point that resumption cannot get a compositional status in the logical language L, since its semantics as formulated in DEFINITION 2.16 fails to meet compositionality. In Section 4.3.3.2 we will see that the syntax of iteration is problematic for the translation of the natural language into the logical language, so iteration is not compositional, either. Moreover, in Section 4.3.3.3, I will show that the expressive power of polyadic lifts in general raises an important problem for a λ-calculus with functional types, the basic combinatoric system of compositional grammars in linguistics.

4.3.3.2 Iteration as a mode of composition?

Let us define the semantic function $G_{M,7:It}$ in DEFINITION 4.25 which assigns meaning to iterations. Since the semantics of iteration is defined on the basis of the monadic quantifiers (DEFINITION 2.10), $G_{M,7:It}$ in DEFINITION 4.26 can be specified in terms of the two parts, such that in_M is a homomorphism between $F_{7:It}$ and $G_{M,7:It}$:

Definition 4.26 The semantic operation for iteration

$$G_{M,7:It} : D^{Ass}_{E,(et)((et)t)} \times D^{Ass}_{E,(et)((et)t)} \rightarrow D^{Ass}_{E,(et)((et)((e(et))t))}$$

where $G_{M,7:It}(\phi, \psi) =$

$\{\langle a, f(f_1)(f_2)\rangle | \langle a, f_1\rangle \in \phi$ *and* $\langle a, f_2\rangle \in \psi$,

and for every $h_1, h_2 \in D^{Ass}_{E,(et)}, g \in D^{Ass}_{E,e(et)}, g_{AR:1} \in D^{Ass}_{E,(e(et))(et)}$,

where $g_{AR:1}$ *is the result of applying* AR_1 *to* g,

$$((((f(f_1))(f_2))(h_1))(h_2))(g) = (f_1(h_1))(g_{AR:1}(f_2(h_2)))\}$$

[25]Of course, one may define $Res(NO)(NO)$ as a constant of type $(et)((et)((e(et))t))$ like NO^2, but then resumption cannot be used as a mode of composition.

$$It(NO)(NO)(student')(book')(read')_t$$
$$\uparrow=$$
$$F_{4:e(et):t}(It(NO)(NO)(student')(book'), read')$$

$$It(NO)(NO)(student')(book')_{(e(et))t} \qquad read'_{e(et)}$$
$$\uparrow=$$
$$F_{4:et:(e(et))t}(It(NO)(NO)(student'), book')$$

$$It(NO)(NO)(student')_{(et)((e(et))t)} \qquad book'_{et}$$
$$\uparrow=$$
$$F_{4:et:(et)((e(et))t)}(It(NO)(NO), student')$$

$$It(NO)(NO)_{(et)((et)((e(et))t))} \qquad student'_{et}$$
$$\uparrow=$$
$$F_{7:It}(NO, NO)$$

$$NO_{(et)((et)t)} \qquad NO_{(et)((et)t)}$$

Figure 4.4: Compositional derivation of $It(NO)(NO)(student')(book')(read')_t$

Given the two functions $F_{7:It}$ and $G_{M,7:It}$ and the homomorphism in_M, iteration is now defined as a "mode of composition" in L. Thus we can compositionally derive the logical expression $((((It(NO)(NO))(student'))(book'))(read')) \in L_t$, as in FIGURE 4.4. Note that for simplicity I derive this logical expression on the basis of constants in the set $Const_\tau$ (instead of the equivalent λ-abstracted expressions) and the functions $(F_\gamma)_{\gamma\in\Gamma}$, where $\Gamma = \{1, 2, 4{:}\tau'{:}\tau,\ 5{:}\tau'{:}\tau{:}i,\ 6{:}\tau,\ 7{:}It\}$. It will be interpreted on the basis of the functions Int and $(G_\gamma)_{\gamma\in\Gamma}$. In FIGURE 4.4, I indicate the syntactic operation which is applied at each step in the tree.

Iteration and the natural language syntax In terms of polyadic quantifiers, the logical expression $(It(NO)(NO)(student')(book')(read'))_t$ in FIGURE 4.4 should translate the Romanian sentence in (233) (*Niciun student nu a citit nicio carte.* 'No student read no book.') in its double negation reading. But we will see below that this idea turns out to be problematic for the surface-oriented syntax that is assumed here.

Although iteration may be viewed as a mode of composition in the logical language L, the syntax of Romanian given in Section 4.3.2.2 and exemplified in FIGURE 4.3, p. 135 is different from the one in FIGURE 4.4. Most importantly, there is no syntactic rule $\underline{H}_{It}$ which combines two determiners into a complex syntactic term. Assuming that we translate both Romanian determiners *niciun* and *nicio* with the constant $NO_{(et)((et)t)}$, $F_{7:It}(NO, NO)$ in FIGURE 4.4 should translate a complex syntactic term $\lfloor \underline{H}_{It} \lfloor niciun_{Det} \rfloor \lfloor nicio_{Det} \rfloor \rfloor$ which does not exist in Romanian or in any other natural language. The typical syntax for natural language is the one given in FIGURE 4.5, where I write only the logical translations of the Romanian expressions and maintain the lexical translations given in (234). Note that unlike in FIGURE 4.4 here I use the λ-abstracted expressions in (234), since we want to derive the translation of a complex natural language expression.

$$S$$
$$It(NO)(NO)(\lambda x_{1,e}.student')(\lambda x_{4,e}.book')(\lambda x_{2,e}\lambda x_{5,e}.read')$$
$$\uparrow ??$$
$$?\mathbf{F_x}(\lambda B_{et}.NO(\lambda x_{1,e}.student'(x_1))(\lambda x_{2,e}.B(x_2)),$$
$$\lambda x_{9,e}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.read'(x_9,x_5)))$$

$$NP \qquad\qquad IV$$
$$\lambda B_{et}.NO(\lambda x_{1,e}.student'(x_1))(\lambda x_{2,e}.B(x_2)) \qquad \lambda x_{9,e}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.read'(x_9,x_5))$$
$$\uparrow =$$
$$F_a(\lambda x_{8,e}\lambda x_{7,e}.read'(x_7,x_8),$$
$$\lambda R_{e(et)}\lambda x_{9,e}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.R(x_9,x_5)))$$

$$TV \qquad\qquad NP$$
$$\lambda x_{8,e}\lambda x_{7,e}.read'(x_7,x_8) \qquad \lambda R_{e(et)}\lambda x_{9,e}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.R(x_9,x_5))$$
$$\uparrow \mathbf{L}^{2,1}$$
$$\lambda D_{et}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.D(x_5))$$
$$\uparrow =$$
$$F_{4:et:(et)t}(\lambda C_{et}\lambda D_{et}.NO(\lambda x_{4,e}.C(x_4))(\lambda x_{5,e}.D(x_5)),$$
$$\lambda x_{6,e}.book'(x_6))$$

$$Det \qquad\qquad CN$$
$$\lambda C_{et}\lambda D_{et}.NO(\lambda x_{4,e}.C(x_4))(\lambda x_{5,e}.D(x_5)) \qquad \lambda x_{6,e}.book'(x_6)$$

Figure 4.5: Syntactic tree for the logical translation of *Niciun student nu a citit nicio carte.*

$$F_x(\alpha,\beta)$$

$$\alpha_{(et)t} \qquad\qquad \beta_{et}$$
$$\lambda B_{et}.X_t \qquad\qquad \lambda x_{9,e}.Y_t$$
$$\lambda B_{et}.NO(\lambda x_{1,e}.student'(x_1))(\lambda x_{2,e}.B(x_2)) \qquad \lambda x_{9,e}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.read'(x_9,x_5))$$

Figure 4.6: The syntax of a compositional function

The tree in FIGURE 4.5 differs from the one in FIGURE 4.3 in one important respect: the derivation of the logical expression representing the IV node. It concerns the way the logical expression $\lambda x_{8,e}\lambda x_{7,e}.read'(x_7, x_8)$ (standing for the transitive verb) combines with the quantificational one $\lambda D_{et}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.D(x_5))$ (standing for the NP). In FIGURE 4.3, Argument Raising applied to the first argument of the transitive verb (by AR_1), to make it match the type of the NP. In FIGURE 4.5, we raise the argument λD_{et} of the NP so that it matches the type $e(et)$ of the transitive verb.[26] This is done by means of a lifting operation defined in van Eijck (2005) for polyadic quantifiers and given below in DEFINITION 4.27. The expression $\lambda D_{et}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.D(x_5))$[27] is lifted to $\lambda R_{e(et)}\lambda x_{9,e}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.R(x_9, x_5))$, as described in (243)[28], where it replaces the variable $\mathbf{Q}_{(et)t}$. This mechanism of lifting the type of the NP, instead of that of the TV like in FIGURE 4.3, brings us closer to the GQT idea that quantifiers take the relation of the verb as their argument and not the other way around. Still, the result at the IV-level is the same.

Definition 4.27 Lifting of type $\langle 1\rangle$ quantifiers
A type $\langle 1\rangle$ function $\mathbf{Q}$ *on the universe E can be lifted to a function (*$\mathbf{L}^{(n+1),n}\mathbf{Q}$*) from (*$n+1$*)-ary relations to* n*-ary relations as follows:*
$(\mathbf{L}^{(n+1),n}\mathbf{Q}) = \lambda \mathbf{Q}_{\mathbf{(et)t}}\lambda R_{e^{n+1}t}\lambda(x_{1,e}, ..., x_{n,e}).\mathbf{Q}(\lambda z_e.R(x_1, ..., x_n, z))$
(van Eijck (2005, p. 88))

(243)
$$\begin{aligned}&(\mathbf{L}^{2,1}\mathbf{Q}) = \lambda \mathbf{Q}_{(et)t}\lambda R_{e(et)}\lambda x_{9,e}.\mathbf{Q}(\lambda z_e.R(x_9, z))\\&[\lambda \mathbf{Q}_{(et)t}\lambda R_{e(et)}\lambda x_{9,e}.\mathbf{Q}(\lambda z_e.R(x_9, z))](\lambda D_{et}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.D(x_5))) =\\&\lambda R_{e(et)}\lambda x_{9,e}.[\lambda D_{et}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.D(x_5))](\lambda z_e.R(x_9, z)) =\\&\lambda R_{e(et)}\lambda x_{9,e}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.[\lambda z_e.R(x_9, z)](x_5)) =\\&\lambda R_{e(et)}\lambda x_{9,e}.NO(\lambda x_{4,e}.book'(x_4))(\lambda x_{5,e}.R(x_9, x_5))\end{aligned}$$

At the S-level, we have to combine two expressions of the same types as the ones in FIGURE 4.3. But this time, we would like to use a syntactic operation that would give us an expression that contains the polyadic quantifier $It(NO)(NO)$. If we just use functional application (i.e. the operation $F_{4:et:t}$) like in FIGURE 4.3, we do not integrate the polyadic quantifier. If we make use of the polyadic quantifier, the expression we should obtain is the one under S, i.e. $It(NO)(NO)(\lambda x_{1,e}.student')(\lambda x_{4,e}.book')(\lambda x_{2,e}\lambda x_{5,e}.read')$.

The first thing to notice is that the function $F_{7:It}$ compositionally defined in L is not useful here, since under NP and IV there are two expressions of type $(et)t$ and et, respectively, so $F_{7:It}$ does not apply. We could instead define a new function F_x which applies to such expressions, but this would not solve the problem. This is because this function F_x would have to look inside the two expressions and rearrange their parts. It should collect the quantificational operators (NO and NO) and the restriction of the quantifier within each expression ($\lambda x_{1,e}.student'$ and $\lambda x_{4,e}.book'$, respectively) and rearrange them within the structure of the polyadic quantifier $It(NO)(NO)$.

[26] In order to combine the two expressions, we need a new syntactic operation similar to F_4, but which reverses the order of the functor and the argument: in our language L the former precedes the latter, and here, we need the latter to precede the former. I will not go into details, since they would take us too far from the focus of the argumentation, but I assume for this the operation F_a (which is similar to the operation F_y compositionally defined in Hendriks (1993, p. 135)) such that:

1. $F_a : L_{e(et)} \times L_{(e(et))(et)} \to L_{et}$, where $F_a(\alpha, \beta) = [[\beta](\alpha)]$

[27] According to Hendriks's flexible type assignment, this lifting operation should be performed at the Det-level (in the lexicon), where we lift D_{et} to $R_{e(et)}$ as in (243) and C_{et} stays the same.

[28] In (243), I use the variable x_9 instead of x_1 to avoid confusion with the variable x_1 already used in our grammar.

A compositional function cannot be defined to operate this way. It only has access to the whole expression and its type. To illustrate this, take a look at the tree in FIGURE 4.6, where the function F_x applies to the two expressions $\alpha_{(et)t}$ and β_{et}. As an alternative to functional application, F_x can combine the two expressions such that $F_x(\alpha, \beta) = \lambda W_{et}\lambda u_e.\alpha(W) * \beta(u)$, for instance, where "$*$" stands for any binary operator defined in the logical language (conjunction, disjunction etc). This is possible because the type of the two expressions allows F_x to see what arguments α and β require. F_x can also combine the two type t expressions X and Y by some binary operator, but it has no access to their components, i.e. the quantificational operator NO and the two relevant restrictions.[29]

Iteration vs. functional application In conclusion, the syntax in FIGURE 4.5 does not allow us to formulate a compositional function that would provide the polyadic quantifier $It(NO)(NO)$. Recall that iteration was claimed in de Swart and Sag (2002) to have a compositional status similar (if not identical) to that of functional application. Moreover, the GQT literature (Keenan and Westerståhl (1997), Peters and Westerståhl (2006), a.o.) often points out the similarity between iteration and functional application. As a consequence of our discussion, it should be clear that this 'similarity' is limited to the level of the logical interpretation, but it does not hold for the natural language syntax. As we have seen above, iteration as a polyadic lift cannot be formulated as a compositional function that obeys the syntax of the natural language.

The similarity between the semantics of iteration and that of functional application raises a further question: how is it possible that iteration can be formulated compositionally within the logical language L (via the function $F_{7:It}$) but not in the syntax of the natural language? Functional application and iteration are both compositional in L, they combine the same syntactic pieces and yield the same semantics, but still only the syntax of the former is compositional in relation to the natural language. The final expression in the tree in FIGURE 4.4 is interpreted by the function $G_{M,7:it}$ in DEFINITION 4.26. Given f the semantics of $It(NO)(NO)$, f_1, f_2 for the semantics of the first and second NO, respectively, h_1, h_2 for the semantics of $\lambda x_{1,e}.student'(x_1)$ and $\lambda x_{4,e}.book'(x_4)$, and g for the semantics of $\lambda x_{2,e}\lambda x_{5,e}.read'(x_2, x_5)$, the definition says that $(((((f(f_1))(f_2))(h_1))(h_2))(g) = (f_1(h_1))(g_{AR:1}(f_2(h_2)))$. In FIGURE 4.5, if we apply functional application at the S-level, we obtain the expression $NO(\lambda x_{1,e}.student'(x_1))(\lambda x_{2,e}.NO(\lambda x_{4,e}.book'(x_4))\ (\lambda x_{5,e}.read'(x_2, x_5)))$ which is interpreted by the same semantic object $(f_1(h_1))(g_{AR:1}(f_2(h_2)))$. Both semantic interpretations are homomorphic to the logical syntax: for iteration, it is the function $F_{7:It}$, for functional application it is the function F_4. But while in the case of functional application, the interpretation $(f_1(h_1))(g_{AR:1}(f_2(h_2)))$ is the one established by the homomorphism with the logical syntax (which also corresponds to the natural language syntax), in the case of iteration, it is the expression $(((((f(f_1))(f_2))(h_1))(h_2))(g)$ that is established by the homomorphism with the logical syntax and this differs from the natural language (surface-oriented) syntax. So the equivalence between the two syntactic expressions is the effect of the way the semantic function $G_{M,7:It}$ is formulated: despite the homomorphism with $F_{7:It}$ (see DEFINITION 4.25), the interpretation assigned by $G_{M,7:It}$ introduces a syntax which is different from the syntax of $F_{7:It}$ and the homomorphism with the natural language syntax cannot be established.

In conclusion, iteration and functional application as modes of composition get the same truth conditions, but the way they put the parts together differs. In particular, the syntax of iteration as a polyadic quantifier is not taken into account by the semantic function interpreting iteration, and for this reason it is impossible to formulate the polyadic lift *iteration* as a mode of composition like functional application.

[29] For more discussion, see also Zimmermann (1990, Sec. 4.6), in particular, pp. 108–109.

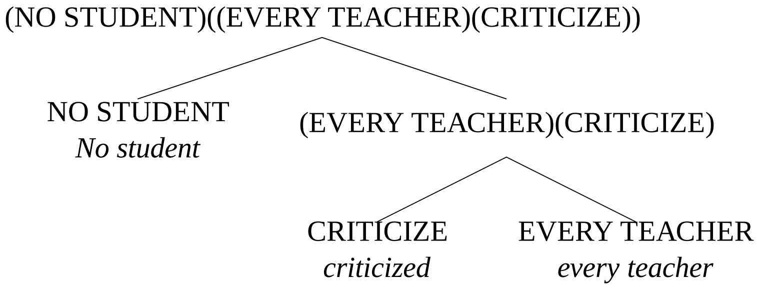


Figure 4.7: Compositional syntactic tree with generalized quantifiers (Keenan (1992, p. 201))

4.3.3.3 Polyadic quantifiers and λ-calculus with functional types

In this section I propose an explanation for why we cannot define resumption (and possibly other binary quantifiers) compositionally in a logical language with lambda-calculus and functional types. In Section 4.3.3.1 we only saw the intuitive problem: the semantics of resumption does not make direct use of the semantics of the syntactic parts, i.e. the monadic quantifiers. Here I will show that there are binary relations which cannot be expressed as a combination of two unary relations and accordingly, there are binary quantifiers that can distinguish between these relations in a way that combinations of two monadic quantifiers cannot. This discussion comes as a continuation of Section 2.1.4.

I start with a brief summary of the general claims concerning the syntax of polyadic quantifiers and the conclusions we reached here with respect to the status of iteration and resumption in a compositional grammar. Then I focus on why some binary quantifiers like resumption cannot be defined compositionally in the logical language.

Keenan (1992) talks about the assumptions that are made with respect to the syntax of polyadic quantifiers. He starts with the compositional syntactic structure of a sentence with two quantifiers like our example in FIGURE 4.5, which he describes by means of generalized quantifiers. In FIGURE 4.7, I give the tree presented by Keenan, as the similarity to FIGURE 4.5 is straightforward.

With respect to FIGURE 4.7, Keenan (1992, p. 201) writes:

> "Observe now that it makes sense to compose type $\langle 1 \rangle$ functions. Thus the last line in (2) [i.e. FIGURE 4.7] equals
>
> $$[(\text{NO STUDENT}) \circ (\text{EVERY TEACHER})](\text{CRITICIZE})$$
>
> where $[(\text{NO STUDENT}) \circ (\text{EVERY TEACHER})]$ maps binary relations to truth values and is thus a function of type $\langle 2 \rangle$."

Keenan (1992), and the literature on polyadic quantifiers in general, is interested in accounting for those binary quantifiers which are not 'reducible' to the composition (i.e. iteration) of two monadic quantifiers. But nothing more is said about the 'new syntax' introduced with the function $[(\text{NO STUDENT}) \circ (\text{EVERY TEACHER})]$ above. For this reason, the reader is left with the impression that this function should be compositional (together with its syntax), since its origin is the compositional structure in FIGURE 4.7. As we just saw, this is an erroneous assumption, since functional application and iteration do not have the same syntax. Composing the two unary quantifiers in FIGURE 4.7 into a binary quantifier as suggested by Keenan forces us to adopt the syntax in FIGURE 4.8 if we want such a function to obey compositionality in a logical language. This syntax does not match the syntax of the natural language, which is why we cannot have a mode of composition iteration (Section 4.3.3.2).

[(NO STUDENT) ∘ (EVERY TEACHER)](CRITICIZE)

[(NO STUDENT) ∘ (EVERY TEACHER)] CRITICIZE

NO STUDENT EVERY TEACHER

Figure 4.8: Syntactic tree with a binary iteration

We saw that, unlike iteration, resumption cannot be made compositional even in the logical language (Section 4.3.3.1). For the syntax of binary quantifiers in FIGURE 4.8, this means that we cannot find two monadic quantifiers that could give us the binary resumptive quantifier Q^2 in a compositional way. The question is why this is the case.

Let us call the two monadic quantifiers that we need to determine Q_1 and Q_2.[30] I assume that in the logical language L they are represented as the constants Q_1/ Q_2 of type $(et)((et)t)$. CRITICIZE is the constant $criticize'$ of type $e(et)$, and TEACHER the constant $teacher'$ of type et. Thus the GQT expressions in FIGURE 4.8 can be replaced by the logical ones in FIGURE 4.9.

$(Q^2)(student')(teacher')(criticize')$

$\lambda V_{3,e(et)}.(Q^2)(student')(teacher')(V_3)$ $\quad$ $criticize'_{e(et)}$

$\delta_{(e(et))t} = F_y(\alpha, \beta)$

$\lambda V_{1,et}.(Q_1)(student')(V_1)$ $\quad$ $\lambda V_{2,et}.(Q_2)(teacher')(V_2)$

$\alpha_{(et)t}$ $\quad$ $\beta_{(et)t}$

Figure 4.9: Syntactic tree with binary resumption/ non-iteration

With iteration we know the two monadic quantifiers α and β and compose them to obtain δ, the binary one. The same procedure applies both in the syntax and the semantics and thus iteration is compositional in the logical language. With resumption we have the two syntactic parts α and β which undergo the syntactic operation F_y to build the binary quantifier δ. But in Section 4.3.3.1 we defined the semantics of the binary quantifier in a way that did not make use of the semantics of the two syntactic parts. The question now is whether there is a way to express the semantics of δ as the semantics of $F_y(\alpha, \beta)$.

The binary quantifier $\delta_{(e(et))t}$ is a function with the domain $P(E^2)$ and the co-domain $P(E^0)$. The binary quantifier $F_y(\alpha, \beta)$, which is a combination of the two monadic quantifiers $\alpha_{(et)t}$ and $\beta_{(et)t}$, has the domain $P(E) \times P(E)$ and the co-domain $P(E^0)$. We need to determine $\alpha_{(et)t}$ and $\beta_{(et)t}$, such that $F_y(\alpha, \beta)$ and $\delta_{(e(et))t}$ are identical, i.e. they return the same truth value with respect to all binary relations in the domain.

[30] We ignore for now the fact that the two quantifiers should have the same operator in resumption. I will return to this issue at the end of the section.

In the general case, to be able to reformulate every binary quantifier δ as a combination $F_y(\alpha, \beta)$ of two monadic ones, all the binary relations that the former distinguishes between should be similarly told apart by the latter. We should first be able to restate all the binary relations in $P(E^2)$ as also elements of $P(E) \times P(E)$.[31] The domain $P(E) \times P(E)$ may contain more binary relations than $P(E^2)$, but not the other way around. We further need an operation that gives us for each relation in $P(E^2)$ a direct correspondent in $P(E) \times P(E)$. Once we have that, every binary relation V_3 in $P(E^2)$ can be restated as some logical operation "$\otimes$" between two unary relations, i.e. $V_3 = V_1 \otimes V_2$. In this case, the distinctions that the binary quantifier δ can make for the elements of a relation V_3 can also be made by an appropriate operation "$\bullet$" on two monadic quantifiers, each applying to one of the two unary relations V_1 and V_2, such that $\delta = \alpha \bullet \beta$.

However, Henk Barendregt (p. c.) points out to me that this correspondence cannot be established in general because the cardinality of $P(E^2)$ is usually higher than that of $P(E) \times P(E)$. If the domain E contains n elements, such that $n > 2$, the cardinality of $P(E^2)$ is always higher than the cardinality of $P(E) \times P(E)$. The cardinality of $P(E^2)$ is $2^{(n^2)}$ and that of $P(E) \times P(E)$ is 2^{2n} (as indicated in (244)), and for instance for $n = 3$, the former equals 2^9, while the latter is 2^6:

Lemma 2.2 (p. 15) *For every set* A, $n \in \mathbb{N}$ *such that* $|\text{A}| = n$, $|\text{P(A)}| = 2^n$.

(244) $$|E| = n \Rightarrow |E^2| = n^2 \overset{L:2.2}{\Longleftrightarrow} |P(E^2)| = 2^{(n^2)}$$
$$|P(E) \times P(E)| = 2^n \times 2^n = 2^{2n}$$

For $n = 1$, we have the only case in which $|P(E) \times P(E)| > |P(E^2)|$, since $2^2 > 2^{(1^2)}$. For $n = 2$ and $n = 0$ we have the identity $|P(E) \times P(E)| = |P(E^2)|$, since $2^4 = 2^{(2^2)}$ and $2^0 = 2^0$. Apart from these three cases, that is, for $n > 2$, $|P(E^2)| > |P(E) \times P(E)|$.

A way to put the two unary relations together and get a binary relation is by means of the Cartesian product. We can define the binary relation V_3 as equal to $V_1 \times V_2 = \lambda v_{1,e} \lambda v_{2,e}.V_1(v_1) \wedge V_2(v_2)$. In this case, our logical operator $\otimes$ is the Cartesian product, i.e. $\otimes = \times$, and the corresponding operation "$\bullet$" between the monadic quantifiers is functional composition/ iteration, i.e. $\bullet = \circ$. But, as pointed out in Keenan (1992), we can only obtain binary iterations in this way (see Section 2.1.4.2). So all those binary quantifiers $\delta = \alpha \circ \beta$ are iterations. As shown above, they are also compositional in the language L.

Some non-iterations can be restated as *Boolean combinations* of iterations (van Benthem (1989)). Peters and Westerståhl (2006, p. 351) views a binary cumulative quantifier as a conjunction of two iterations. It remains to be shown how and if Boolean combinations of iterations can also be made compositional in a logical language.

Given the cardinality difference between the domain $P(E^2)$ of binary relations and $P(E) \times P(E)$, the domain of binary combinations of unary relations, there are binary quantifiers that express the truth conditions of some binary relations in the set difference $P(E^2) - P(E) \times P(E)$ which cannot be expressed by combinations of monadic quantifiers and are thus non-compositional. For the Generalized Quantifier Theory, this cardinality difference predicts that the expressive power of a binary quantifier is higher than that of the composition of two monadic quantifiers. This is exactly the idea that the literature on polyadic quantifiers exploits: there are binary quantifiers which can be reduced to a composition (i.e. iteration) of two monadic ones, but natural language also employs other binary quantifiers which cannot. Keenan (1992), Keenan and Westerståhl (1997), Peters and Westerståhl (2006) and others concentrate on these 'unreducible' binary quantifiers, for which they abandon the idea of compositionality.

[31] I leave aside the matter of how we could make the pieces in the syntax of α and β fit the syntax of δ, namely, how we could put the (unary) operators and restrictions together in δ (see also the discussion after FIGURE 4.6).

Let us consider what this result tells us about resumptive quantifiers, argued here to account for NC. We saw that the resumptive NO^2 is reducible to the iteration $NO \circ SOME$ (Section 4.2.3). This means that NO^2 does not express the truth conditions of a binary relation in the set difference $P(E^2) - P(E) \times P(E)$. However, the two monadic quantifiers that are composed to give us this semantics are not the same two quantifiers that undergo the syntactic operation resumption. If we have a syntactic operation resumption between two monadic quantifiers NO and we interpret it by composing the semantics of two quantifiers NO and $SOME$ our operation is again non-compositional. We saw before that the semantic status of n-words in NC requires that we treat all their occurrences as negative quantifiers, which makes a treatment of NC in terms of the iteration $NO \circ SOME$ inadequate. Moreover, not all resumptive quantifiers are reducible to iteration. As mentioned before, Peters and Westerståhl (2002) argues that $MOST^2$ is unreducible. For our discussion, this means that $MOST^2$ characterizes binary relations in the set difference $P(E^2) - P(E) \times P(E)$.

Thus resumptive quantifiers allow us to express special truth conditions that cannot be obtained in any other way (e.g. $MOST^2$) and to provide a systematic account for our empirical observations (e.g. NO^2). For Romanian NC, I showed that resumptive negative quantifiers best capture the semantic status of n-words and their scope behavior (Section 4.2). The non-compositional status of resumptive quantifiers indicates that a logic with lambda calculus and functional types is not powerful enough to accommodate them.

4.4 Conclusion and discussion

In this chapter I first showed that iteration and resumption of two negative quantifiers are well-suited to account for the DN and NC readings of sentences with two n-words in Romanian. Specifically, I argued that a resumptive quantifier NO^2 can account for the idiosyncratic scope properties of NC and the negative quantifier status of n-words, while the semantically equivalent iteration NO $\circ$ SOME cannot.

I then investigated the possibility to define the polyadic lifts resumption and iteration in a compositional syntax-semantics of a Romanian fragment. I showed that the way the semantics of resumption is defined does not allow a direct access to the semantic contribution of the monadic parts. This means that resumption cannot be defined as a mode of composition. I further showed that the syntax of polyadic quantifiers prevents us from formulating even iteration as a mode of composition. While iteration can be defined compositionally in the logical language L, its syntax does not match that of natural language, so iteration fails to be compositional at the interface with the natural language algebra. Finally, I gave an explanation for why we cannot directly integrate polyadic quantifiers in a compositional fragment. This has to do with the expressive power of binary quantifiers, which is higher than that of a combination of two monadic ones. The domain of the former ($P(E^2)$) is usually richer than the domain of the latter ($P(E) \times P(E)$). So no structural correspondence can be established between the two domains to allow us to express every binary quantifier in terms of a combination of two monadic ones, as required by compositionality.

The source of the incompatibility between polyadic quantifiers and the principle of compositionality in linguistics is the way compositionality is traditionally defined in linguistics: 1) in a functional type theory and 2) by using functional application (or other lambda-calculus techniques with functional types) to imitate natural language syntax. To be more precise about the latter procedure, note that type shifting mechanisms like argument raising are employed to allow a full match between the constituent structure of natural language and a combinatorics with λ-calculus and functional types (see Section 4.3.2.4).

It is difficult to envisage a reformulation of the principle of compositionality to allow the integration of polyadic quantifiers. We can start by eliminating the limitative properties of compositional grammars that prevent us from defining polyadic quantifiers, at the same time trying to keep the previous results that the principle provides for linguistic theory. Given the two issues mentioned above, we have two options: 1) to replace the functional type theory (employed in Montague (1970) following Church (1940) and assumed in compositional grammars) with a more powerful type theory or 2) to replace the compositional combinatorics based on λ-calculus and functional types with a natural language surface-oriented syntax.

The first option was brought to my attention by Fritz Hamm (p. c.) who mentions that one may be able to define polyadic quantifiers compositionally if one starts with an intuitionistic type theory (Martin-Löf (1984)) instead of a simple type theory usually assumed with the principle of compositionality in linguistics. The intuitionistic type theory is largely used in computer science, but it has occasionally been employed for linguistics as well (e.g. Sundholm (1989), Ranta (1991, 1994)), and it crucially has more expressive power than the simple type theory which it yields as a special case. It thus presents itself as an option in defining polyadic quantifiers so they match a more flexible notion of compositionality that is to be formulated in this general setting. However, compositionality with a simple type theory has a long history in linguistics and covers a wide spectrum of phenomena which must be accounted for with the new notion of compositionality not yet available, before we may pursue an extension to polyadic quantifiers. Such an attempt is too complex to be made here.

The other option is to compose complex expressions by strictly following the constraints of the natural language constituent structure instead of the lambda calculus techniques employed by compositional grammars. This is the path I follow in Chapter 5, where I present a systematic syntax-semantics for resumptive quantifiers by making use of underspecified representations in the semantic framework Lexical Resource Semantics (LRS) (Richter and Sailer (2004)). LRS keeps the traditional practice of a functional type theory as the representation language, but gives up the traditional combinatorics based on lambda-calculus, when deriving complex expressions. It uses the constituent structure provided by a surface-oriented syntax instead. This innovation allows a direct and precise implementation of resumptive k-ary quantifiers and thus a systematic account of Romanian NC as a resumptive quantifier.

Chapter 5

The HPSG analysis of Romanian NC: An LRS account

The aim in this chapter is to propose a systematic syntax-semantics for Romanian NC as resumptive quantification. The limitative effects that the principle of compositionality has on the description of natural language quantification has led the Polyadic Quantifiers literature to disregard it. This is possible within the Generalized Quantifier Theory where the focus is on the semantics of quantifiers and the natural language syntax is left aside. But to offer an adequate theoretical description of the linguistic phenomenon of negative concord we need to account for both its syntax and its semantics.

In this chapter I show that recent developments regarding semantic description undertaken within the tradition of constraint-based formalisms, in particular HPSG, enable us to articulate the syntax-semantics of negative concord that we need: one that takes into account both the resumptive semantics of NC and a natural constituent structure for the Romanian sentence. The syntax employed here follows the general lines of the HPSG fragment developed in Section 2.3. The semantic representation language is a simplified type theory without possible worlds $Ty1$ (cf. $Ty2$ of Gallin (1975)).

Two semantic frameworks have been proposed for HPSG which make use of $Ty2$ semantic representations: *Lexicalized Flexible Ty2* (LF-Ty2) in Sailer (2003) and *Lexical Resource Semantics* (LRS) in Richter and Sailer (2004) and Richter (2004*a*). LF-Ty2 is a direct encoding of $Ty2$ in the grammar formalism of HPSG that uses the classical combinatorial system with lambda-calculus and functional application. LRS is a meta-theory of semantic representation which combines $Ty2$ semantic representations with constraint-based techniques of linguistic description, in particular underspecified representations. As shown in the previous chapter, polyadic quantifiers cannot be given a syntax-semantics in a combinatorial system with lambda calculus and functional types, because they are not compositional. For this reason, in this chapter I take up the semantic framework of LRS rather than LF-Ty2. We will see that the constraint-based mechanisms dealing with underspecification in LRS can successfully account for Romanian negative concord as a resumptive quantifier.

The chapter begins with the description of the logical language $Ty1$ (Section 5.1) in which I represent resumptive quantifiers in such a way that they can be used in LRS. In Section 5.2, I present the RSRL grammar of $Ty1$ (Γ_{Ty1}) which allows us to use $Ty1$ expressions as semantic representations in HPSG. I continue in Section 5.3 with a general presentation of the LRS framework, the theoretical background for the subsequent analysis of NC (Section 5.4.1) and DN (Section 5.4.2). In Section 5.5 I address the semantic and syntactic properties of the Romanian negative marker and I integrate them in the overall analysis of NC. After a few technical considerations in Section 5.6, in Section 5.7 I illustrate how the present analysis can account for the locality conditions on NC.

5.1 The representation language: Polyadic quantifiers in $Ty1$

In this section I describe the representation language that will be used in the rest of the chapter. There are no major differences between this language and the language L in Chapter 4. But the different goals of the two chapters require different ways of presenting the logical language. To investigate the compositional status of polyadic quantifiers, the presentation in the previous chapter had to follow particular conventions from Hendriks (1993) which in the context of this chapter would impede understanding. Moreover, in this chapter I will often rely on previous work whose aim was to integrate logical representations in HPSG (especially Sailer (2003)). To allow an immediate understanding of this material within that context, I adopt the conventions of the presentation in Sailer (2003).

Sailer (2003) uses Two-sorted Type Theory ($Ty2$ of Gallin (1975)) as the representation language for semantic descriptions in HPSG. But as we have seen in the previous chapter, the discussion on polyadic quantifiers does not involve the *world type* s, which is the second basic type besides e in $Ty2$. So I will exclusively use a *One-sorted Type Theory* $Ty1$. This does not mean that the $Ty1$ definitions below cannot be extended to the world type s, and thus to $Ty2$.

5.1.1 The syntax of $Ty1$

The syntax of the language $Ty1$ is defined below:

Definition 5.1 Type

Let Type be the smallest set such that

$e, t \in Type$,

for each $\tau, \tau' \in Type, \tau \rightarrow \tau' \in Type$.

Each element of the set *Type* is called a *(semantic) type*. The basic types e, t stand for individuals and truth values, respectively.

Convention 5.1 Type Notation
1. We write $\tau \rightarrow \tau'$ *as* $\tau\tau'$.
2. We write $\underbrace{(\tau \rightarrow (... \rightarrow (\tau}_{n\text{-times}} \rightarrow \tau')...))$ *as* $\tau^n\tau'$.
3. We make use of parentheses $(\ )$ *only when disambiguation is necessary.*

Definition 5.2 Var

Let Var *be the smallest set such that*
for each $\tau \in Type$ *and for each* $i \in \mathbb{N}^+$, $v_{i,\tau} \in Var$.

Each element of the set *Var* is called a *variable*. Note that I do not use the variable $v_{0,\tau}$, so i must be a positive number.

Definition 5.3 Const

Let $Const$ *be the smallest set such that*
for each $\tau \in Type$ *and for each* $i \in \mathbb{N}^+$, $c_{i,\tau} \in Const$.

Each element of the set *Const* is called a *constant*.

Definition 5.4 $Ty1$ Terms
Ty1 is the smallest set such that:

$Var \subset Ty1$,
$Const \subset Ty1$,

for each $\tau, \tau' \in Type$, *for each* $\alpha_{\tau\tau'}, \beta_\tau \in Ty1$,

$$(\alpha_{\tau\tau'}\beta_\tau)_{\tau'} \in Ty1,$$

for each $\tau, \tau' \in Type$, *for each* $v_{i,\tau} \in Var$, *and for each* $\alpha_{\tau'} \in Ty1$,

$$(\lambda v_{i,\tau}.\alpha_{\tau'})_{(\tau\tau')} \in Ty1,$$

for each $\tau \in Type$, *and for each* $\alpha_\tau, \beta_\tau \in Ty1$,

$$(\alpha_\tau = \beta_\tau)_t \in Ty1$$

for each $\alpha_t \in Ty1$,

$$(\neg\alpha_t)_t \in Ty1,$$

for each $\alpha_t, \beta_t \in Ty1$,

$$(\alpha_t \wedge \beta_t)_t \in Ty1, \qquad \textit{(analogously for } \vee, \rightarrow, \leftrightarrow\textit{)}$$

for each $\tau \in Type$, *for each* $n \in \mathbb{N}^0$, *for each* $i_1, i_2, ..., i_n \in \mathbb{N}^+$, *for each* $v_{i_1,\tau}, v_{i_2,\tau}, ..., v_{i_n,\tau} \in Var$, *for each* $\alpha_{t1}, \alpha_{t2}, ..., \alpha_{tn}, \beta_t \in Ty1$,

$$(NO(v_{i_1,\tau}, ..., v_{i_n,\tau})(\alpha_{t1}, ...\alpha_{tn})(\beta_t))_t \in Ty1,$$

for each $\tau \in Type$, *for each* $n \in \mathbb{N}^+$, *for each* $i_1, i_2, ..., i_n \in \mathbb{N}^+$, *for each* $v_{i_1,\tau}, v_{i_2,\tau}, ..., v_{i_n,\tau} \in Var$, *for each* $\alpha_{t1}, \alpha_{t2}, ..., \alpha_{tn}, \beta_t \in Ty1$,

$$(SOME(v_{i_1,\tau}, ..., v_{i_n,\tau})(\alpha_{t1}, ...\alpha_{tn})(\beta_t))_t \in Ty1,$$

for each $\tau \in Type$, *for each* $n \in \mathbb{N}^+$, *for each* $i_1, i_2, ..., i_n \in \mathbb{N}^+$, *for each* $v_{i_1,\tau}, v_{i_2,\tau}, ..., v_{i_n,\tau} \in Var$, *for each* $\alpha_{t1}, \alpha_{t2}, ..., \alpha_{tn}, \beta_t \in Ty1$,

$$(EVERY(v_{i_1,\tau}, ..., v_{i_n,\tau})(\alpha_{t1}, ...\alpha_{tn})(\beta_t))_t \in Ty1.$$

$Ty2$ standard results about higher-order languages have shown that the first three functions in DEFINITION 5.4 (application, abstraction, and equality) are sufficient to add quantifiers and the other logical operators (Gallin (1975)). In addition to the terms above, we can thus use the universal and the existential quantifier as syntactic sugar in our language $Ty1$:

(245) a. true : $[\lambda x_t.x_t = \lambda x_t.x_t]$
b. $\forall x_\tau \sigma_t : [\lambda x_\tau.\sigma_t = \lambda x_\tau \text{true}]$
c. $\exists x_\tau \sigma_t : \neg\forall x_\tau \neg\sigma$

(Sailer (2003, p. 40))

Generalized quantifiers in $Ty1$ In the logical language L in the previous chapter, we initially represented the monadic quantifier NO syncategorematically (see the function F_3 in DEFINITION 4.7, p. 125). To investigate the compositional status of polyadic quantifiers, in Section 4.3.3 we had to redefine it categorematically, so that it could be the argument of a polyadic lift like Res. We then defined Res syncategorematically (as applying to two monadic quantifiers) and the quantifier NO^2 categorematically to represent binary resumptive quantifiers. We saw that Res could not be defined compositionally because a corresponding semantic operation could not be constructed. For NO^2, treated as a constant, we defined the semantics under the interpretation function for constants Int.

In the language $Ty1$ I adopt another way of representing resumptive quantifiers. I give a syncategorematic representation of the monadic quantifier NO and I generalize it to stand for a quantifier NO of any complexity: monadic or polyadic. This matches Lindström's view of a generalized quantifier as a class of quantifiers of n complexity. All Romanian n-words can be represented as contributing negative quantifiers of Lindström type $\langle 1, 1\rangle$, so any resumptive quantifier representing negative concord will be of type $\langle 1^n, n\rangle$. Thus I define the generalized quantifier NO in $Ty1$ as corresponding to the Lindström type $\langle 1^n, n\rangle$. Similarly for other generalized quantifiers like $SOME$ and $EVERY$.

The generalized quantifiers in $Ty1$ take the following arguments: n variables of type τ (possibly the same variable more than once if for instance $i_k = i_{k+j}$, for every $k, j \in \mathbb{N}^+$, such that $k + j \leq n$), a corresponding n number of type t expressions which act as the restriction of the quantifier and one type t expression which is the nuclear scope, and return a truth value. So generalized quantifiers are expressions of type $\tau^n(t^n(tt))$.

For the quantifier NO we allow $n = 0$, since in Section 5.5 we will need this to represent the Romanian negative marker *nu* as a type $\langle 0\rangle$[1] quantifier which in $Ty1$ corresponds to an expression of type tt. For the other quantifiers ($SOME$ and $EVERY$), $n \geq 1$. In the next section I present the semantics of $Ty1$.

5.1.2 The semantics of $Ty1$

Definition 5.5 Frame

Let E be a set of individuals, then $F = \bigcup_{\tau\in Type} D_{E,\tau}$ is a frame where,

$D_{E,t} = \{1, 0\}$,

$D_{E,e} = E$,

for each $\tau, \tau' \in Type$,

$$D_{E,\tau\tau'} = D_{E,\tau'}^{D_{E,\tau}}.$$

Definition 5.6 Model

Given a set of constants $Const$, a set of individuals E,

a $Ty1$ model is a pair $M = \langle F, Int\rangle$, such that

F is a frame, and

Int is a function from $Const$ to F such that

for each $c_\tau \in Const$,

$Int(c) \in D_{E,\tau}$.

[1]See de Swart and Sag (2002) for a similar approach.

Definition 5.7 Variable Assignment
Ass is the set of functions F^{Var} (from Var to F) such that,

Ass= $\{a \in F^{Var} |$ *for each* $i \in \mathbb{N}^+$, *for each* $\tau \in Type$, $a(v_{i,\tau}) \in D_{E,\tau}\}$.

Definition 5.8 The Semantics of $Ty1$ Terms
For each term $\alpha_\tau \in Ty1$, for each model M and for each variable assignment $a \in Ass$,

$[\![\alpha_\tau]\!]^{M,a}$*, the extension of α_τ in a model $M = \langle F, Int\rangle$ under a variable assignment $a \in Ass$, is defined as follows:*

[constants]
for each $\tau \in Type$, for each $i \in \mathbb{N}^+$, for each $c_{i,\tau} \in Const$,
$[\![c_{i,\tau}]\!]^{M,a} = Int(c)$,
[variables]
for each $\tau \in Type$, for each $i \in \mathbb{N}^+$, for each $v_{i,\tau} \in Var$,
$[\![v_{i,\tau}]\!]^{M,a} = a(v_{i,\tau})$,

[application]
for each $\tau, \tau' \in Type$, for each $\alpha_{\tau\tau'} \in Ty1$, for each $\beta_\tau \in Ty1$,
$[\![(\alpha_{\tau\tau'}\beta_\tau)_{\tau'}]\!]^{M,a} = [\![\alpha_{\tau\tau'}]\!]^{M,a}([\![\beta_\tau]\!]^{M,a})$,
[abstraction]
for each $\tau, \tau' \in Type$, for each $v_{i,\tau} \in Var$, for each $\alpha_{\tau'} \in Ty1$,
$[\![(\lambda v_{i,\tau}.\alpha_{\tau'})_{\tau\tau'}]\!]^{M,a} = f \in D_{E,\tau'}^{D_{E,\tau}}$ *such that*
for each $d \in D_{E,\tau}$: $f(d) =$ $[\![\alpha_{\tau'}]\!]^{M,a[v_{i,\tau}/d]}$,
[equation]
for each $\tau \in Type$, for each $\alpha_\tau, \beta_\tau \in Ty1$,
$[\![(\alpha_\tau = \beta_\tau)_t]\!]^{M,a} = 1$ *if* $[\![\alpha_\tau]\!]^{M,a} =[\![\beta_\tau]\!]^{M,a}$, *else* 0,

[logical operators]
for each $\alpha_t \in Ty1$,
$[\![(\neg\alpha_t)_t]\!]^{M,a}= 1$ *if* $[\![\alpha_t]\!]^{M,a}= 0$, *else* 0,
for each $\alpha_t, \beta_t \in Ty1$,
$[\![(\alpha_t \wedge \beta_t)_t]\!]^{M,a} = 1$ *if* $[\![\alpha_t]\!]^{M,a} = 1$ *and* $[\![\beta_t]\!]^{M,a} = 1$, *else* 0,
for each $\alpha_t, \beta_t \in Ty1$,
$[\![(\alpha_t \vee \beta_t)_t]\!]^{M,a}= 1$ *if* $[\![\alpha_t]\!]^{M,a}= 1$ *or* $[\![\beta_t]\!]^{M,a}= 1$, *else* 0,
for each $\alpha_t, \beta_t \in Ty1$,
$[\![(\alpha_t \rightarrow \beta_t)_t]\!]^{M,a}= 1$ *if* $[\![\alpha_t]\!]^{M,a}= 0$ *or* $[\![\beta_t]\!]^{M,a}= 1$, *else* 0,
for each $\alpha_t, \beta_t \in Ty1$,

$[\![(\alpha_t \leftrightarrow \beta_t)_t]\!]^{M,a} = 1$ *if*
$[\![\alpha_t]\!]^{M,a} = 1$ *and* $[\![\beta_t]\!]^{M,a} = 1$ *or*
$[\![\alpha_t]\!]^{M,a} = 0$ *and* $[\![\beta_t]\!]^{M,a} = 0$, *else* 0,

[quantifiers]

for each $\tau \in Type$, *for each* $n \in \mathbb{N}^0$, *for each* $i_1, i_2, ..., i_n \in \mathbb{N}^+$, *for each* $v_{i_1,\tau}, v_{i_2,\tau}, ..., v_{i_n,\tau} \in Var$, *for each* $\alpha_{t1}, \alpha_{t2}, ..., \alpha_{tn}, \beta_t \in Ty1$,
$[\![NO(v_{i_1,\tau}, ..., v_{i_n,\tau})(\alpha_{t1}, ..., \alpha_{tn})(\beta_t)]\!]^{M,a} = 1$
iff for every $d_{i_1}, d_{i_2}, ..., d_{i_n} \in D_{E,\tau}$,
$[\![\alpha_{t1}]\!]^{M,a[v_{i_1,\tau}/d_{i_1}]} = 0$ *or* $[\![\alpha_{t2}]\!]^{M,a[v_{i_2,\tau}/d_{i_2}]} = 0$ *or* ...
or $[\![\alpha_{tn}]\!]^{M,a[v_{i_n,\tau}/d_{i_n}]} = 0$ *or* $[\![\beta_t]\!]^{M,a[(v_{i_1},...,v_{i_n})/(d_{i_1},...,d_{i_n})]} = 0$,

for each $\tau \in Type$, *for each* $n \in \mathbb{N}^+$, *for each* $i_1, i_2, ..., i_n \in \mathbb{N}^+$, *for each* $v_{i_1,\tau}, v_{i_2,\tau}, ..., v_{i_n,\tau} \in Var$, *for each* $\alpha_{t1}, \alpha_{t2}, ..., \alpha_{tn}, \beta_t \in Ty1$,
$[\![SOME(v_{i_1}, ..., v_{i_n})(\alpha_{t1}, ..., \alpha_{tn})(\beta_t)]\!]^{M,a} = 1$
iff there exist $d_{i_1}, d_{i_2}, ..., d_{i_n} \in D_{E,\tau}$ *such that*
$[\![\alpha_{t1}]\!]^{M,a[v_{i_1,\tau}/d_{i_1}]} = 1$ *and* $[\![\alpha_{t2}]\!]^{M,a[v_{i_2,\tau}/d_{i_2}]} = 1$ *and* ...
and $[\![\alpha_{tn}]\!]^{M,a[v_{i_n,\tau}/d_{i_n}]} = 1$ *and* $[\![\beta_t]\!]^{M,a[(v_{i_1},...,v_{i_n})/(d_{i_1},...,d_{i_n})]} = 1$,

for each $\tau \in Type$, *for each* $n \in \mathbb{N}^+$, *for each* $i_1, i_2, ..., i_n \in \mathbb{N}^+$, *for each* $v_{i_1,\tau}, v_{i_2,\tau}, ..., v_{i_n,\tau} \in Var$, *for each* $\alpha_{t1}, \alpha_{t2}, ..., \alpha_{tn}, \beta_t \in Ty1$,
$[\![EVERY(v_{i_1}, ..., v_{i_n})(\alpha_{t1}, ..., \alpha_{tn})(\beta_t)]\!]^{M,a} = 1$
iff for every $d_{i_1}, d_{i_2}, ..., d_{i_n} \in D_{E,\tau}$,
if $[\![\alpha_{t1}]\!]^{M,a[v_{i_1}/d_{i_1}]} = 1$ *and* $[\![\alpha_{t2}]\!]^{M,a[v_{i_2}/d_{i_2}]} = 1$ *and* ...
and $[\![\alpha_{tn}]\!]^{M,a[v_{i_n}/d_{i_n}]} = 1$, *then* $[\![\beta_t]\!]^{M,a[(v_{i_1},...,v_{i_n})/(d_{i_1},...,d_{i_n})]} = 1$.

Let us take some examples of generalized quantifiers to illustrate how they are interpreted. For $n = 0$, we can only have the quantifier NO which applies to an expression of type t, say $come'(j)$, where $j, come' \in Const$, see (246a). This quantifier will be used in Section 5.5 to represent the negative marker *nu* in Romanian. For $n = 3$, we can build ternary quantifiers with NO, $SOME$ and $EVERY$. Considering that we have three distinct variables of type e as is usually the case in natural language (i.e. $i_1 \neq i_2 \neq i_3$), we simplify the notation and use the variables x, y, z to stand for $v_{i_1}, v_{i_2}, v_{i_3}$, respectively. Let us take $\alpha_{t1} = teacher'(x)$, $\alpha_{t2} = book'(y)$, $\alpha_{t3} = student'(z)$ and $\beta_t = give'(x, y, z)$. With these specifications, we can build the following quantifiers in (246):

(246) Examples of generalized quantifiers in natural language:

a. For $n = 0$, $[\![NO()()(come'(j))]\!]^{M,a} = 1$ iff $[\![come'(j)]\!]^{M,a} = 0$

b. For $n = 3$, $v_{i_1} = x, v_{i_2} = y, v_{i_3} = z$, $\alpha_{t1} = teacher'(x)$, $\alpha_{t2} = book'(y)$, $\alpha_{t3} = student'(z)$ and $\beta_t = give'(x, y, z)$,
$[\![NO(x, y, z)(teacher'(x), book'(y), student'(z))(give'(x, y, z))]\!]^{M,a} = 1$ iff
for every $d_1, d_2, d_3 \in D_{E,e}$,

$[\![teacher'(x)]\!]^{M,a[x/d_1]} = 0$ or $[\![book'(y)]\!]^{M,a[y/d_2]} = 0$ or
$[\![student'(z)]\!]^{M,a[z/d_3]} = 0$ or $[\![give'(x,y,z)]\!]^{M,a[(x,y,z)/(d_1,d_2,d_3)]} = 0$

c. For $n = 3$, $v_{i_1} = x, v_{i_2} = y, v_{i_3} = z$, $\alpha_{t1} = teacher'(x)$, $\alpha_{t2} = book'(y)$, $\alpha_{t3} = student'(z)$ and $\beta_t = give'(x,y,z)$,
$[\![SOME(x,y,z)(teacher'(x), book'(y), student'(z))(give'(x,y,z))]\!]^{M,a} = 1$ iff there exist $d_1, d_2, d_3 \in D_{E,e}$,
$[\![teacher'(x)]\!]^{M,a[x/d_1]} = 1$ and $[\![book'(y)]\!]^{M,a[y/d_2]} = 1$ and
$[\![student'(z)]\!]^{M,a[z/d_3]} = 1$ and $[\![give'(x,y,z)]\!]^{M,a[(x,y,z)/(d_1,d_2,d_3)]} = 1$

d. For $n = 3$, $v_{i_1} = x, v_{i_2} = y, v_{i_3} = z$, $\alpha_{t1} = teacher'(x)$, $\alpha_{t2} = book'(y)$, $\alpha_{t3} = student'(z)$ and $\beta_t = give'(x,y,z)$,
$[\![EVERY(x,y,z)(teacher'(x), book'(y), student'(z))(give'(x,y,z))]\!]^{M,a} = 1$ iff for every $d_1, d_2, d_3 \in D_{E,e}$,
if $[\![teacher'(x)]\!]^{M,a[x/d_1]} = 1$ and $[\![book'(y)]\!]^{M,a[y/d_2]} = 1$ and
$[\![student'(z)]\!]^{M,a[z/d_3]} = 1$, then $[\![give'(x,y,z)]\!]^{M,a[(x,y,z)/(d_1,d_2,d_3)]} = 1$

The semantics of the generalized quantifiers given in DEFINITION 5.8 can also be expressed in terms of the minimum of the $Ty1$ syntax (application, abstraction, equation) with the syntactic sugar in (245). (247) illustrates how this can be done. Thus defining generalized quantifiers does not involve any extensions of the language $Ty1$:

(247) *Generalized Quantifiers*

a. for each $\tau \in Type$, for each $n \in \mathbb{N}^0$, for each $i_1, i_2, ..., i_n \in \mathbb{N}^+$,
for each $v_{i_1,\tau}, v_{i_2,\tau}, ..., v_{i_n,\tau} \in Var$, for each $\alpha_{t1}, \alpha_{t2}, ..., \alpha_{tn}, \beta_t \in Ty1$,

$$NO(v_{i_1}, ..., v_{i_n})(\alpha_{t1}, ..., \alpha_{tn})(\beta_t) :=$$
$$\exists A_{i_1,\tau t} ... \exists A_{i_n,\tau t} \exists B_{\tau^n t}$$
$$((A_{i_1} = \lambda v_{i_1}.\alpha_{t1} \wedge ... \wedge A_{i_n} = \lambda v_{i_n}.\alpha_{tn} \wedge B = \lambda v_{i_1} ... \lambda v_{i_n}.\beta)$$
$$\wedge \forall v_{i_1} ... \forall v_{i_n} [(A_{i_1}(v_{i_1}) \wedge ... \wedge A_{i_n}(v_{i_n})) \rightarrow \neg B(v_{i_1}, ..., v_{i_n})]).$$

b. for each $\tau \in Type$, for each $n \in \mathbb{N}^+$, for each $i_1, i_2, ..., i_n \in \mathbb{N}^+$,
for each $v_{i_1,\tau}, v_{i_2,\tau}, ..., v_{i_n,\tau} \in Var$, for each $\alpha_{t1}, \alpha_{t2}, ..., \alpha_{tn}, \beta_t \in Ty1$,

$$SOME(v_{i_1}, ..., v_{i_n})(\alpha_{t1}, ..., \alpha_{tn})(\beta_t) :=$$
$$\exists A_{i_1,\tau t} ... \exists A_{i_n,\tau t} \exists B_{\tau^n t}$$
$$((A_{i_1} = \lambda v_{i_1}.\alpha_{t1} \wedge ... \wedge A_{i_n} = \lambda v_{i_n}.\alpha_{tn} \wedge B = \lambda v_{i_1} ... \lambda v_{i_n}.\beta)$$
$$\wedge \exists v_{i_1} ... \exists v_{i_n} [A_{i_1}(v_{i_1}) \wedge ... \wedge A_{i_n}(v_{i_n}) \wedge B(v_{i_1}, ..., v_{i_n})]).$$

c. for each $\tau \in Type$, for each $n \in \mathbb{N}^+$, for each $i_1, i_2, ..., i_n \in \mathbb{N}^+$,
for each $v_{i_1,\tau}, v_{i_2,\tau}, ..., v_{i_n,\tau} \in Var$, for each $\alpha_{t1}, \alpha_{t2}, ..., \alpha_{tn}, \beta_t \in Ty1$,

$$EVERY(v_{i_1}, ..., v_{i_n})(\alpha_{t1}, ..., \alpha_{tn})(\beta_t) :=$$
$$\exists A_{i_1,\tau t} ... \exists A_{i_n,\tau t} \exists B_{\tau^n t}$$
$$((A_{i_1} = \lambda v_{i_1}.\alpha_{t1} \wedge ... \wedge A_{i_n} = \lambda v_{i_n}.\alpha_{tn} \wedge B = \lambda v_{i_1} ... \lambda v_{i_n}.\beta)$$
$$\wedge \forall v_{i_1} ... \forall v_{i_n} [(A_{i_1}(v_{i_1}) \wedge ... \wedge A_{i_n}(v_{i_n})) \rightarrow B(v_{i_1}, ..., v_{i_n})]).$$

5.2 $Ty1$ in RSRL

In order to make use of $Ty1$ terms as semantic representations in the constraint-based framework of HPSG, $Ty1$ has to be encoded in RSRL, the description language of HPSG. We have to define the grammar $\Gamma_{Ty1}=\langle\Sigma_{Ty1},\Theta_{Ty1}\rangle$, and prove that it describes exactly the language $Ty1$. The signature Σ_{Ty1} must specify the sorts and the attributes for describing $Ty1$ expressions, and the theory Θ_{Ty1} must ensure that all and only the well-formed expressions of $Ty1$ are in the denotation of the new sorts. Then it must be proved that $Ty1$ is an exhaustive model of Γ_{Ty1} (see also Section 2.3.1).

This kind of encoding and the corresponding proofs have been done for the language $Ty2$ in Sailer (2003) and both LF-Ty2 and LRS use it. Since the language $Ty1$ is a restricted version (lacking the world type) of the language $Ty2$ plus the (Lindström) generalized quantifiers, I take the work done by Sailer for the grammar of $Ty2$ to also cover the grammar Γ_{Ty1}, with the exception of the generalized quantifiers in $Ty1$ for which I add the necessary extensions.

In what follows, I give the description of the grammar of $Ty1$ (Γ_{Ty1}). For a more detailed discussion, the reader is referred to (Sailer, 2003, Ch. 3).

The Signature Σ_{Ty1} FIGURE 5.1 below presents the signature for a grammar of $Ty1$. It follows the general assumptions in Sailer (2003), Penn and Richter (2004), Richter (2004*a*) and Richter and Kallmeyer (2007), but introduces a few modifications meant to deal with the extensions of $Ty1$ introduced in Section 5.1.1.

All the objects in Γ_{Ty1} are subsumed by the sort *ty1* which, together with the sort *list*, will be an immediate subsort of the sort *object* in the HPSG sort hierarchy given in (47), Section 2.3.1. The *meaningful expressions* of $Ty1$ are subsumed by the sort *me*. They have an attribute TYPE whose value specifies their semantic type. Simple expressions (*variable*s and *constant*s) also get a positive natural number index (*non-zero*), the value of the attribute NUM-INDEX. This sort - attribute specification is generally assumed in the LF-Ty2 and LRS tradition.

The signature contains an extended structure of quantifiers, where the RESTR(iction) is separated from SCOPE, so all quantifiers are treated as generalized quantifiers (*gen-quantifier*), as in Richter and Kallmeyer (2007). To accommodate resumptive quantifiers, the value of the attributes VAR and RESTR is of sort *list*. These additions are meant to match the syntax of generalized quantifiers in $Ty1$, as presented in DEFINITION 5.4 above. The signature also contains some additional relations which are needed for the formulation of the constraints in the theory of $Ty1$ and which will be described as part of the theory of $Ty1$ in the next section.

The Theory Θ_{Ty1} The theory of the grammar of $Ty1$ consists of a set of constraints on the *ty1* (sub)sorts which guarantee that these sorts correspond to the natural numbers (for *integer*s), the semantic types (for *type*s), and the well-formed expressions of $Ty1$ (for *me*s). All the constraints are given below:

(248) THE THEORY Θ_{Ty1}

1. THE NATURAL NUMBERS PRINCIPLE:

   integer$\rightarrow \exists x\ ^{x}[zero]$

2. THE COMPLEX TERM PRINCIPLES:

```
ty1
  me  TYPE  type
    variable  NUM-INDEX  non-zero
    constant  NUM-INDEX  non-zero
    application  FUNCTOR  me
                 ARG  me
    abstraction  VAR  me
                 BODY  me
    equation  ARG1  me
              ARG2  me
    negation  ARG  me
    l-const  ARG1  me
             ARG2  me
      disjunction
      conjunction
      implication
      bi-implication
    gen-quantifier  VAR  list
                    RESTR  list
                    SCOPE  me
      every
      some
      no
  type
    atomic-type
      entity
      truth
    complex-type  IN  type
                  OUT  type
  integer
    zero
    non-zero PRE integer
list
    elist
    nelist FIRST me
           REST list
Relations
copy/2
member/2
same-length/2
same-type-list/2
subterm/2
truth-list/1
ty1-component/2
variable-list/1
```

Figure 5.1: The signature Σ_{Ty1}

$$application \rightarrow \begin{bmatrix} \text{TYPE } \boxed{2} \\ \text{FUNCTOR}|\text{ TYPE } \begin{bmatrix} \text{IN} & \boxed{1} \\ \text{OUT} & \boxed{2} \end{bmatrix} \\ \text{ARG}|\text{ TYPE } \boxed{1} \end{bmatrix} \qquad abstraction \rightarrow \begin{bmatrix} \text{TYPE } \begin{bmatrix} \text{IN} & \boxed{1} \\ \text{OUT} & \boxed{2} \end{bmatrix} \\ \text{VAR}|\text{ TYPE } \boxed{1} \\ \text{BODY}|\text{ TYPE } \boxed{2} \end{bmatrix}$$

$$equation \rightarrow \begin{bmatrix} \text{TYPE } truth \\ \text{ARG1}|\text{ TYPE } \boxed{1} \\ \text{ARG2}|\text{ TYPE } \boxed{1} \end{bmatrix} \qquad negation \rightarrow \begin{bmatrix} \text{TYPE } truth \\ \text{ARG}|\text{ TYPE } truth \end{bmatrix}$$

$$l\text{-}const \rightarrow \begin{bmatrix} \text{TYPE } truth \\ \text{ARG1}|\text{ TYPE } truth \\ \text{ARG2}|\text{ TYPE } truth \end{bmatrix}$$

$$gen\text{-}quantifier \rightarrow \begin{bmatrix} \text{TYPE } truth \\ \text{VAR } \boxed{1} \\ \text{RESTR } \boxed{2} \\ \text{SCOPE}|\text{ TYPE } truth \end{bmatrix}$$
$$\wedge \texttt{variable-list}(\boxed{1}) \wedge \texttt{same-type-list}(\boxed{3},\boxed{1})$$
$$\wedge \texttt{truth-list}(\boxed{2}) \wedge \texttt{same-length}(\boxed{1}, \boxed{2})$$

3. THE $Ty1$ NON-CYCLICITY PRINCIPLE:
$$ty1 \rightarrow \forall\boxed{1}\left(\left(\bigvee\left\{\left[\alpha\ \boxed{1}\right] \mid \alpha \in \mathcal{A}_{Ty1}\right\}\right) \rightarrow \neg\ \texttt{ty1-component}(:, \boxed{1})\right)$$

4. THE $Ty1$ FINITENESS PRINCIPLE:
$$ty1 \rightarrow \exists\boxed{1}\ \forall\boxed{2}\Big(\texttt{ty1-component}(\boxed{2}, :) \rightarrow \texttt{member}(\boxed{2}, \boxed{1}[chain])\Big)$$

5. THE $Ty1$ IDENTITY PRINCIPLE:
$$ty1 \rightarrow \forall\boxed{1}\ \forall\boxed{2}\ \Big(\texttt{copy}\big(\boxed{1}, \boxed{2}\big) \rightarrow \boxed{1} = \boxed{2}\Big)$$

6. THE `ty1-component` PRINCIPLE:
$$\forall\boxed{1}\ \forall\boxed{2}\left(\begin{array}{l} \texttt{ty1-component}\big(\boxed{1}, \boxed{2}\big) \leftrightarrow \\ \left(\begin{array}{l} \boxed{1} = \boxed{2} \vee \\ \bigvee\left\{\exists\boxed{3}\left(\begin{array}{l} \boxed{2}\left[\alpha\ \boxed{3}\right] \wedge \\ \texttt{ty1-component}\big(\boxed{1}, \boxed{3}\big) \end{array}\right) \middle|\ \alpha \in \mathcal{A}_{Ty1}\right\} \end{array}\right) \end{array}\right)$$

7. THE `copy` PRINCIPLE:
$$\forall\boxed{1}\ \forall\boxed{2}\left(\begin{array}{l} \texttt{copy}\big(\boxed{1}, \boxed{2}\big) \leftrightarrow \\ \left(\begin{array}{l} \bigvee\left\{\boxed{1}\left[\sigma\right] \wedge \boxed{2}\left[\sigma\right] \mid \sigma \in \mathcal{S}_{Ty1}\right\} \wedge \\ \bigwedge\left\{\forall\boxed{3}\left(\begin{array}{l} \boxed{1}\left[\alpha\ \boxed{3}\right] \rightarrow \\ \exists\boxed{4}\left(\boxed{2}\left[\alpha\ \boxed{4}\right] \wedge \texttt{copy}\big(\boxed{3}, \boxed{4}\big)\right) \end{array}\right) \middle|\ \alpha \in \mathcal{A}_{Ty1}\right\} \end{array}\right) \end{array}\right)$$

8. THE `subterm` PRINCIPLE:
$$\forall\boxed{1}\ \forall\boxed{2}\ \left(\texttt{subterm}\big(\boxed{1}, \boxed{2}\big) \leftrightarrow \left(\begin{array}{l} \boxed{1}\left[me\right] \wedge \boxed{2}\left[me\right] \wedge \\ \texttt{ty1-component}\big(\boxed{1}, \boxed{2}\big) \end{array}\right)\right)$$

9. THE `variable-list` PRINCIPLE:

$$\forall\boxed{1}\left(\begin{array}{l}\texttt{variable-list}(\boxed{1})\leftrightarrow\\ \left(\begin{array}{l}\boxed{1}\big[\textit{elist}\big]\vee\\ \exists\boxed{2}\exists\boxed{3}\left(\boxed{1}\begin{bmatrix}\text{FIRST} & \boxed{2}\big[\textit{variable}\big]\\ \text{REST} & \boxed{3}\end{bmatrix}\wedge\texttt{variable-list}(\boxed{3})\right)\end{array}\right)\end{array}\right)$$

10. THE `member` PRINCIPLE:

$$\forall\boxed{1}\ \forall\boxed{2}\left(\texttt{member}(\boxed{1},\boxed{2})\leftrightarrow\left(\begin{array}{l}\boxed{2}\begin{bmatrix}\textit{list}\\ \text{FIRST} & \boxed{1}\big[\textit{me}\big]\end{bmatrix}\vee\\ \exists\boxed{3}\left(\boxed{2}\begin{bmatrix}\textit{list}\\ \text{REST} & \boxed{3}\big[\textit{list}\big]\end{bmatrix}\wedge\texttt{member}(\boxed{1},\boxed{3})\right)\end{array}\right)\right)$$

11. THE `truth-list` PRINCIPLE:

$$\forall\boxed{1}\left(\begin{array}{l}\texttt{truth-list}(\boxed{1})\leftrightarrow\\ \left(\begin{array}{l}\boxed{1}\big[\textit{elist}\big]\vee\\ \exists\boxed{2}\exists\boxed{3}\left(\boxed{1}\begin{bmatrix}\text{FIRST} & \boxed{2}\begin{bmatrix}\textit{me}\\ \text{TYPE} & \textit{truth}\end{bmatrix}\\ \text{REST} & \boxed{3}\end{bmatrix}\wedge\texttt{truth-list}(\boxed{3})\right)\end{array}\right)\end{array}\right)$$

12. THE `same-length` PRINCIPLE:

$$\forall\boxed{1}\forall\boxed{2}\left(\begin{array}{l}\texttt{same-length}(\boxed{1},\boxed{2})\leftrightarrow\\ \left(\begin{array}{l}\left(\boxed{1}\big[\textit{elist}\big]\wedge\boxed{2}\big[\textit{elist}\big]\right)\vee\\ \exists\boxed{3}\exists\boxed{4}\left(\boxed{1}\begin{bmatrix}\text{FIRST} & \textit{me}\\ \text{REST} & \boxed{3}\end{bmatrix}\wedge\boxed{2}\begin{bmatrix}\text{FIRST} & \textit{me}\\ \text{REST} & \boxed{4}\end{bmatrix}\wedge\texttt{same-length}(\boxed{3},\boxed{4})\right)\end{array}\right)\end{array}\right)$$

13. THE `same-type-var` PRINCIPLE:

$$\forall\boxed{1}\ \forall\boxed{2}\left(\begin{array}{l}\texttt{same-type-list}(\boxed{1},\boxed{2})\leftrightarrow\\ \left(\boxed{1}\big[\textit{type}\big]\wedge\left(\begin{array}{l}\boxed{2}\big[\textit{elist}\big]\vee\\ \exists\boxed{3}\left(\boxed{2}\begin{bmatrix}\text{FIRST} & \begin{bmatrix}\textit{me}\\ \text{TYPE} & \boxed{1}\end{bmatrix}\\ \text{REST} & \boxed{3}\end{bmatrix}\wedge\texttt{same-type-list}(\boxed{1},\boxed{3})\right)\end{array}\right)\right)\end{array}\right)$$

Regarding the principles in (248), note that quantification in RSRL always applies to components of the described object (Richter (2004*b*), p. 152). A component is by definition an object that can be reached via a path of attributes.

The NATURAL NUMBERS PRINCIPLE ensures the correspondence between the objects in the grammar of $Ty1$ denoted by *integer* and natural numbers. For a *non-zero* integer, the number of PRE attributes that it has corresponds to the natural number that it represents. The principle in (248.1) specifies that every *integer* object should contain a *zero* value of the attribute PRE. Thus infinite and cyclic numbers are excluded.

The COMPLEX TERM PRINCIPLES in (248.2) guarantee the proper typing of $Ty1$ complex terms according to the conditions specified in the $Ty1$ syntax (terms): *application* of a functor to an argument $((\alpha_{\tau\tau'}\beta_{\tau})_{\tau'})$, lambda *abstraction* $((\lambda v_{i,\tau}.\alpha_{\tau'})_{\tau\tau'})$, *equation* $((\alpha_{\tau} = \beta_{\tau})_{t})$, *negation* $((\neg\alpha_{t})_{t})$, complex expressions made up of two expressions of type *truth* which are connected by a logical constant ($\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$) and denoted here by *l-const*, and generalized quantifiers (*gen-quantifier*) (e.g. $(NO(v_{i_1,\tau}, ..., vi_n, \tau)(\alpha_{t1}, ..., \alpha_{tn})(\beta_t))_t$). The constraint on generalized quantifiers ensures that the members of the value list for VAR are variables and have the same type, that the ones in the value list for RESTR have the type *truth*, that the value of SCOPE is also of type *truth*, and that the two lists that stand for the values of VAR and RESTR have the same length, i.e. the number of expressions in the restriction of the quantifier is the same as the number of the variables bound by the quantifier.

The next three principles (248.3-5) guarantee that the objects denoted by *ty1* correspond to the expressions of the language $Ty1$. The $Ty1$ NON-CYCLICITY PRINCIPLE in (248.3) excludes cyclic objects from the grammar. The symbol ":" is a reserved variable of RSRL expressing the identity function on objects. Here it is used to say that a path can never lead back to the same object. In the RSRL specification of the grammar of $Ty1$, $\mathcal{A}_{Ty1}$ is the set of attributes in the signature of the grammar of $Ty1$. The $Ty1$ FINITENESS PRINCIPLE enforces that every component of a *ty1* object be part of a *chain*. Given that a *chain* (cf. Richter (2004*b*, p. 158)) is finite, *ty1* objects must have a finite structure. The $Ty1$ IDENTITY PRINCIPLE enforces token-identity as often as possible on components of *ty1* objects.

The rest of the principles determine the meaning of the relation symbols which have been or will be used in the other principles: `ty1-component` (248.6), `copy` (248.7), `subterm` (248.8), `variable-list` (248.9), `member` (248.10), `truth-list` (248.11), `same-length` (248.12), and `same-type-list` (248.13). The first argument $\boxed{1}$ of the `ty1-component` relation is a component of the second argument $\boxed{2}$, if and only if the two arguments are identical, or $\boxed{1}$ is a component of the value $\boxed{3}$ of any of the attributes in the finite set of attributes $\mathcal{A}$ specified for $\boxed{2}$. The relation `copy` holds of two *ty1* objects iff they have the same attributes with values of the same sort. In the RSRL formalization of the grammar of $Ty1$, $\mathcal{S}_{Ty1}$ is the set of most specific sorts in the signature of the grammar of $Ty1$. Two meaningful expressions $\boxed{1}$ and $\boxed{2}$ are in the `subterm` relation iff $\boxed{1}$ is a ty1-component of $\boxed{2}$. This relation will be further used in its infix notation, i.e. $\boxed{1} \triangleleft \boxed{2}$ as equivalent to `subterm`($\boxed{1}$,$\boxed{2}$).

The `variable-list` relation guarantees that an object of sort *list* only contains elements of sort *variable*. Thus the relation holds of $\boxed{1}$ iff $\boxed{1}$ is of sort *elist* or the value of its attribute FIRST is of sort *variable* and `variable-list` holds of the value of the attribute REST. An object $\boxed{1}$ is a `member` of a list $\boxed{2}$ iff $\boxed{1}$ is the first element on the list $\boxed{2}$, or it is a `member` of the rest of $\boxed{2}$. Like `append` (see (49) p. 47), the `member` relation is quite often used in HPSG grammars in general. Here it is defined for lists made up of meaningful expressions, but later on it will be used as referring to lists made up of *object* elements (i.e. the most general sort in the sort hierarchy in Section 2.3.1).

The `truth-list` relation functions similarly to the `variable-list` relation and constrains the elements of a list to have the type *truth*. The relation `same-length` enforces the same length on two lists: it is true of two empty lists, or of two lists which have the first element of sort *me* and whose REST values $\boxed{3}$ and $\boxed{4}$ are in the `same-length` relation. Finally, the `same-type-list` relation enforces the elements of a list to have the same type. It holds of any type $\boxed{1}$ and a list $\boxed{2}$ which is either empty or contains only meaningful expressions of type $\boxed{1}$. This relation ensures that the variables in a VAR list have the same type (see THE COMPLEX TERM PRINCIPLE for generalized quantifiers).

As an example of how $Ty1$ expressions can be described in an AVM syntax within the grammar Γ_{Ty1}, see the description of the $Ty1$ expression $\lambda v_{e,0}.constant_{et,1}(v_{e,0})$ below, slightly modified from Richter (2004*a*, p. 172):

(249) AVM description of $\lambda v_{e,1}.constant_{et,2}(v_{e,1})$:

$$
\left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{abstraction}} \\
\text{TYPE } \boxed{4} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{c-type}} \\ \text{IN} & \boxed{1}\left[\textit{entity}\right] \\ \text{OUT} & \boxed{2}\left[\textit{truth}\right] \end{array}\right] \\
\text{VAR } \boxed{3} & \left[\begin{array}{l} \textit{var} \\ \text{TYPE } \boxed{1} \\ \text{NUM-INDEX} \mid \text{PRE } \boxed{5}\textit{zero} \end{array}\right] \\
\text{BODY} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{application}} \\
\multicolumn{2}{l}{\text{TYPE } \boxed{2}} \\
\text{FUNC} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{constant}} \\
\multicolumn{2}{l}{\text{NUM-INDEX} \mid \text{PRE} \mid \text{PRE } \boxed{5}\textit{zero}} \\
\text{TYPE } \boxed{4} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{c-type}} \\ \text{IN} & \boxed{1} \\ \text{OUT} & \boxed{2} \end{array}\right]
\end{array}\right] \\
\multicolumn{2}{l}{\text{ARG } \boxed{3}}
\end{array}\right]
\end{array}\right]
$$

The token-identity between the various attribute values in (249) is enforced by the principles in Θ_{Ty1}. According to the COMPLEX TERM PRINCIPLE for *abstraction* in (248.2), the value of the path TYPE|IN is identical to that of the path VAR|TYPE (i.e. $\boxed{1}$), and the value of the path TYPE|OUT to that of BODY|TYPE (i.e. $\boxed{2}$). The token-identities labeled $\boxed{3}$, $\boxed{4}$ and $\boxed{5}$ are a consequence of the $Ty1$ IDENTITY PRINCIPLE in (248.5). The constant $constant_{et}$ stands for predicate constants of type et: e.g. $walk'$, $student'$, $book'$. When added to the signature as subsumed by *constant*, these predicate constants have different values for the attribute NUM-INDEX. For instance, $walk'$ could be $constant_{et,201}$, $student'$ $constant_{et,130}$, and $book'$ $constant_{et,40}$.

$Ty1$ as a model of Γ_{Ty1} An RSRL grammar is used to describe a certain empirical domain and it can be said to have attained its goal if the empirical domain is proved to be an exhaustive model of the grammar. Γ_{Ty1} has been developed to describe the $Ty1$ expressions defined in Section 5.1.1, so now it has to be shown that $Ty1$ is an exhaustive model of Γ_{Ty1}. Sailer (2003) proves the same with respect to the language $Ty2$ of Gallin (1975). Since $Ty1$ is a simplified version of $Ty2$ I take the results in Sailer (2003) to hold for $Ty1$ as well. The grammar that Sailer develops has been extended to also include generalized quantifiers and lists made up of meaningful expressions. In order to prove that $Ty1$ is a model of the grammar Γ_{Ty1}, we have to prove the proposition below:

> **Proposition 5.1** *There is an exhaustive model* $\mathrm{I}_{Ty1} = \langle U_{Ty1}, S_{Ty1}, A_{Ty1}, R_{Ty1}\rangle$ *such that* $U_{Ty1} = \mathbb{N} \cup Type \cup Ty1 \cup \mathcal{L}$.
>
> (modified from Sailer (2003, p. 117))

In PROPOSITION 5.1, U_{Ty1} is the universe of $Ty1$ objects, i.e. the union between the set of natural numbers, the set of types, the set of $Ty1$ expressions, and the set $\mathcal{L}$ of lists of meaningful expressions, as given in the signature Σ_{Ty1}. S_{Ty1} and A_{Ty1} have already been introduced as the set of maximally specific sorts and of attributes in the signature, respectively. R_{Ty1} is the set of relations in the signature (recall our discussion from Section 2.3.1).

PROPOSITION 5.1 can be proved by constructing a model of Γ_{Ty1}, the *intended model* I_{Ty1}, which must then be proved to be an exhaustive model of Γ_{Ty1}. Sailer (2003) constructs such a model for most of the terms in $Ty1$, except for the quantifiers. In Appendix A under (440), I give the necessary extensions to Sailer's definitions to include *gen-quantifier*s. In Γ_{Ty1} I make use of lists of meaningful expressions as auxiliary symbols to define polyadic quantifiers so I also include *list*s in the extensions in (440).

We can further show that there is a systematic semantic correspondence between the objects in any exhaustive model of Γ_{Ty1} and the terms of $Ty1$. To prove this, a function SR must be defined, which assigns a term α of $Ty1$ an equivalence class $[u]$ of *me* objects in Γ_{Ty1}. Then it must be proved that $[u]$ and α have the same extension. This ensures that for any arbitrary exhaustive model of Γ_{Ty1}, the *me* objects in its universe can be assigned a model-theoretic interpretation just as if they were terms of $Ty1$. Thus every Γ_{Ty1} exhaustive model functions as a model of $Ty1$. Sailer (2003, Sec. 3.3) has done the same for $Ty2$ and in order to extend this result to polyadic quantifiers, I give the interpretation of the *gen-quantifier* and I extend the definition of SR in Appendix A under (441) and (442).

To be able to use $Ty1$ representations instead of AVMs in Γ_{Ty1}, we have to show that the objects in the denotation of the grammar Γ_{Ty1} behave like the natural numbers, the semantic types, the terms, and the sequences (i.e. *list*s) of terms in $Ty1$ (cf. Sailer (2003, Sec. 3.4)). Sailer (2003) defines a function "*" which produces an AVM description for every number, type, expression, and sequence of expressions of the representation language, such that the description denotes that natural number, type, expression or sequence of expressions of the language (in our case, $Ty1$) in the exhaustive model of its corresponding grammar (Γ_{Ty1}). As a result, when working with Γ_{Ty1} the standard notation for a $Ty1$ expression, natural number, type or sequence/ list can be used freely in place of the more complicated AVM formula describing it. In grammar writing this has a considerable practical advantage if we compare the two notations, exemplified in (249): the $Ty1$ symbols are much simpler and more straightforward than the AVM descriptions. The additional specification of the function "*" in Appendix A (443) ensures that generalized quantifiers and sequences/ lists in the $Ty1$ notation receive an appropriate AVM description when used in the grammar Γ_{Ty1}.

In this section I presented a way to encode the language of $Ty1$ (defined in Section 5.1.1) in RSRL as the grammar Γ_{Ty1}. In a way similar to the system in Sailer (2003) $Ty1$ is an exhaustive model of Γ_{Ty1} and $Ty1$ symbols can be used instead of AVM descriptions in grammar writing. This provides us with the possibility of using the language of $Ty1$ as the semantic representation language within HPSG. We can now go on with our HPSG semantic account within LRS.

5.3 LRS

Unlike LF-Ty2 of Sailer (2003), which was developed to import standard model-theoretic semantics in HPSG, *Lexical Resource Semantics* (Richter and Sailer (2004), Richter (2004*a*)) was designed to allow underspecification in HPSG semantics. It maintains the language of $Ty2$ for semantic representations, but unlike LF-Ty2, LRS gives up the restrictive tradition of using lambda-calculus with functional application to imitate the natural language syntax. It uses constraints that are linked to a surface-oriented syntax instead. In addition to this, the type theory ensures the type matching between objects that combine with each other and the well-typing of the derived objects. The combinatorics is regulated via LRS-specific constraints formulated in the logic of HPSG.

I will show that with its constituent structure-based combinatorics, LRS can easily incorporate polyadic quantification, in particular resumptive negative quantifiers, proposed here to account for Romanian NC. After a short presentation of the basic principles of LRS in Section 5.3.1, I will briefly present an LRS account of NC without resumptive quantifiers, as done by Richter and Sailer (2004) for Polish (Section 5.3.2). In Section 5.4, I will develop an LRS analysis of NC with resumptive quantifiers for Romanian.

5.3.1 The basic principles of LRS

LRS makes a distinction between lexical/ local and compositional semantics (see Sailer (2004)).[2] Local semantics is specified as the value *content* of the CONT attribute and is relevant for argument linking, semantic selection of heads, and binding phenomena. The value of CONT hosts an INDEX and a MAIN attribute, the latter specifying the *meaningful expression* that the sign contributes. The INDEX value is split between VAR, the variable associated with the *sign*, and PHI giving the corresponding phi-features[3] The noun *girl* in (250) is third person, singular number and feminine gender, and its MAIN semantic contribution is the constant $girl'$:

(250) The value of CONT for the noun *girl*

$$\begin{bmatrix} \textit{content} & \\ \text{INDEX} & \begin{bmatrix} \textit{extended-index} & \\ \text{VAR} & \textit{variable} \\ \text{PHI} & \begin{bmatrix} \textit{index} & \\ \text{PERS} & \textit{third} \\ \text{NUM} & \textit{sg} \\ \text{GEN} & \textit{fem} \end{bmatrix} \end{bmatrix} \\ \text{MAIN} & girl' \end{bmatrix}$$

Compositional semantics is described under the value of a new *sign*-level attribute LF (Logical Form) and is thus independent of the semantic and syntactic selection by heads. Since NC is a matter of compositional semantics, we will be concerned with the LF value of signs. The value of LF is a new sort *lrs* which we add to the HPSG sort hierarchy in (47), Section 2.3.1, directly under *object*:

(251) THE SORT *lrs*

```
lrs  EX(TERNAL-)CONT(ENT)  me
     IN(TERNAL-)CONT(ENT)  me
     PARTS  list(me)
```

Objects of sort *lrs* have three attributes: INCONT, EXCONT and PARTS. The internal content of a sign is the scopally lowest meaningful expression that the semantic head of the sign contributes within its syntactic projection. The external content of a sign is usually the meaning contribution of its maximal syntactic projection to the meaning of the overall expression. The attribute PARTS contains all the meaningful pieces that a sign contributes to the meaning of a linguistic expression. The values of the three attributes are specified in terms of meaningful expressions (*me*s) defined in the $Ty1$ signature in FIGURE 5.1.

The theory of the LRS grammar contains the INCONT PRINCIPLE, the EXCONT PRINCIPLE, the LRS PROJECTION PRINCIPLE, and the SEMANTICS PRINCIPLE. Each of them is addressed below:

[2]"Compositional" semantics refers to how the semantic representation of a phrase results from the semantics representations of the daughters independently of the principle of compositionality.

[3]The reader familiar with the grammar in Pollard and Sag (1994) should note that the value *index* of the PHI attribute in (250) is the same as the value of INDEX in Pollard and Sag (1994). This allows the lexical semantics phenomena accounted for in that formalism to be easily imported in an LRS grammar. For instance, the binding theory and the agreement mechanisms in Pollard and Sag can be maintained.

(252) LRS PRINCIPLES

a. THE INCONT PRINCIPLE

In each *lrs*, the INCONT value is an element of the PARTS list and a component of the EXCONT value.

$$lrs \rightarrow \left(\begin{bmatrix} \text{EXCONT} & \boxed{1} \\ \text{INCONT} & \boxed{2} \\ \text{PARTS} & \boxed{3} \end{bmatrix} \wedge \boxed{2} \in \boxed{3} \wedge \boxed{2} \triangleleft \boxed{1} \right)$$

b. THE EXCONT PRINCIPLE

1. In every *phrase*, the EXCONT value of the non-head daughter is an element of the non-head daughter's PARTS list.

$$phrase \rightarrow \left(\begin{bmatrix} \text{DTRS} & \boxed{3} \end{bmatrix} \wedge \texttt{non-hd-dtr}(\boxed{3}, \boxed{4}\begin{bmatrix} \text{LF} & \begin{bmatrix} \text{EXCONT} & \boxed{1} \\ \text{PARTS} & \boxed{2} \end{bmatrix} \end{bmatrix}) \wedge \boxed{1} \in \boxed{2} \right)$$

2. In every utterance, every subexpression of the EXCONT value of the utterance is an element of its PARTS list, and every element of the utterance's PARTS list is a subexpression of the EXCONT value.

$$u\text{-}sign \rightarrow \forall\boxed{1}\,\forall\boxed{2}\,\forall\boxed{3}\,\forall\boxed{4} \left(\begin{array}{l} \left(\begin{bmatrix} \text{LF} & \begin{bmatrix} \text{EXCONT} & \boxed{1} \\ \text{PARTS} & \boxed{2} \end{bmatrix} \end{bmatrix} \wedge \boxed{3} \triangleleft \boxed{1} \wedge \boxed{4} \in \boxed{2} \right) \rightarrow \\ \left(\boxed{3} \in \boxed{2} \wedge \boxed{4} \triangleleft \boxed{1} \right) \end{array} \right)$$

c. THE LRS PROJECTION PRINCIPLE

In each *phrase*,

1. the EXCONT values of the head and the mother are identical,

$$phrase \rightarrow \begin{bmatrix} \text{LF}\,|\,\text{EXCONT} & \boxed{1} \\ \text{DTRS}\,|\,\text{HEAD-DTR}\,|\,\text{LF}\,|\,\text{EXCONT} & \boxed{1} \end{bmatrix}$$

2. the INCONT values of the head and the mother are identical,

$$phrase \rightarrow \begin{bmatrix} \text{LF}\,|\,\text{INCONT} & \boxed{1} \\ \text{DTRS}\,|\,\text{HEAD-DTR}\,|\,\text{LF}\,|\,\text{INCONT} & \boxed{1} \end{bmatrix}$$

3. the PARTS value contains all and only the elements of the PARTS values of the daughters.

$$phrase \rightarrow \left(\begin{array}{l} \begin{bmatrix} \text{LF}\,|\,\text{PARTS} & \boxed{1} \\ \text{DTRS} & \boxed{4}\begin{bmatrix} \text{HEAD-DTR}\,|\,\text{LF}\,|\,\text{PARTS} & \boxed{2} \end{bmatrix} \end{bmatrix} \\ \wedge\ \texttt{non-hd-dtr}(\boxed{4}, \boxed{5}\begin{bmatrix} \text{LF}\,|\,\text{PARTS} & \boxed{3} \end{bmatrix}) \wedge \texttt{append}(\boxed{2}, \boxed{3}, \boxed{1}) \end{array} \right)$$

(253) THE `non-hd-dtr` PRINCIPLE:

$$\forall \boxed{1}\, \forall \boxed{2}$$
$$\left(\texttt{non-hd-dtr}(\boxed{1},\boxed{2}) \leftrightarrow \left(\boxed{1}\begin{bmatrix} \textit{head-struc} & \\ \text{SUBJ-DTR} & \boxed{2}[\textit{sign}] \end{bmatrix} \lor \boxed{1}\begin{bmatrix} \textit{head-struc} & \\ \text{SPR-DTR} & \boxed{2}[\textit{sign}] \end{bmatrix} \lor \boxed{1}\begin{bmatrix} \textit{head-struc} & \\ \text{COMP-DTR} & \boxed{2}[\textit{sign}] \end{bmatrix} \lor \boxed{1}\begin{bmatrix} \textit{head-struc} & \\ \text{ADJ-DTR} & \boxed{2}[\textit{sign}] \end{bmatrix} \lor \boxed{1}\begin{bmatrix} \textit{head-struc} & \\ \text{MRK-DTR} & \boxed{2}[\textit{sign}] \end{bmatrix} \right)\right)$$

The theory of LRS makes use of the relations `append` (already discussed in Section 2.3.1, p. 49), `subterm`, `member`, and `non-hd-dtr`. The relations `subterm` and `member` were introduced in the signature of the $Ty1$ grammar and described in (248.8) and (248.10). They are used here in their infix notation symbolized by "$\triangleleft$" and "$\in$", respectively. The `non-hd-dtr` relation is introduced in (253). It delivers the non-head daughter of a phrase, be it a subject, specifier, complement, adjunct, or marker as the value of the attributes SUBJ-DTR, SPR-DTR, COMP-DTR, ADJ-DTR and MRK-DTR of *head-struc* objects (see Section 2.3).

The INCONT PRINCIPLE enforces the presence of the INCONT value of a sign among the elements of its PARTS value, and as a component of the EXCONT value. By the first clause of the EXCONT PRINCIPLE, the EXCONT value of a non-head daughter appears on its PARTS list. The second clause establishes a close relation between the EXCONT and the PARTS value of an utterance, such that every subexpression of its external content is an element of its PARTS list, and every element on the PARTS list is a subexpression of its external content. The LRS PROJECTION PRINCIPLE specifies the LF value of a phrase. Thus the mother node inherits the EXCONT and the INCONT value of the head daughter (clauses 1. and 2.) and its PARTS value is the list obtained by appending the PARTS value of the head daughter and that of the non-head daughter (clause 3.).

The SEMANTICS PRINCIPLE in LRS specifies restrictions on combining the meaning of different kinds of syntactic and semantic daughters. In (254) below I give the relevant clauses for quantificational expressions and for head-marker phrases, as they will be used later in this chapter:

(254) THE SEMANTICS PRINCIPLE

1. if the specifier daughter is a quantifier, then its INCONT value is of the form $Q(v, \phi, \psi)$, the INCONT value of the head is a component of a member[4] of the list ϕ, and the INCONT value of the non-head daughter is identical to the EXCONT value of the head daughter:

$$\forall \boxed{1} \forall \boxed{2} \forall \boxed{3}$$
$$\left(\begin{bmatrix} \text{DTRS}|\text{SPR-DTR}|\text{SS}|\text{LOC} \begin{bmatrix} \text{CAT}|\text{HEAD} & \textit{det} \\ \text{CONT}|\text{MAIN} & \textit{gen-quantifier} \end{bmatrix} \end{bmatrix} \rightarrow \left(\begin{bmatrix} \text{DTRS} \begin{bmatrix} \text{H-DTR}|\text{LF} \begin{bmatrix} \text{EXCONT} & \boxed{1} \\ \text{INCONT} & \boxed{2} \end{bmatrix} \\ \text{SPR-DTR}|\text{LF} \begin{bmatrix} \text{INCONT} & \boxed{1} \begin{bmatrix} \textit{gen-quantifier} \\ \text{RESTR}\ \boxed{3} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \land \boxed{2} \triangleleft_{\in} \boxed{3} \right) \right)$$

2. if the non-head is a quantified NP with an EXCONT value of the form $Q(v, \phi, \psi)$, then the INCONT value of the head is a component of ψ:

$$\forall \boxed{1} \forall \boxed{3} \forall \boxed{4}$$

[4]The symbol "$\triangleleft_{\in}$" is the infix notation of the relation `subterm-of-member` defined in (255).

$$\left(\begin{array}{l}\left(\left[\text{DTRS } \boxed{3}\right] \wedge \boxed{4}\begin{bmatrix}\text{SS}|\text{LOC}|\text{CAT NP} \\ \text{LF}|\text{EXCONT} \begin{bmatrix}\textit{gen-quantifier} \\ \text{SCOPE } \boxed{1}\end{bmatrix}\end{bmatrix} \wedge \texttt{non-hd-dtr}(\boxed{3}, \boxed{4})\right) \\ \rightarrow \exists\boxed{2}\left(\begin{array}{l}\left[\text{DTRS}|\text{ H-DTR }|\text{LF }|\text{INCONT } \boxed{2}\right] \\ \wedge \boxed{2} \triangleleft \boxed{1}\end{array}\right)\end{array}\right)$$

3. if the non-head is a marker, then its INCONT value is identical to the INCONT value of the head:

$$\forall\boxed{1}$$
$$\left(\left[\text{DTRS} \mid \text{MRK-DTR} \begin{bmatrix}\text{SS}|\text{LOC}|\text{CAT}|\text{HEAD } \textit{marker} \\ \text{LF}|\text{INC } \boxed{1}\end{bmatrix}\right] \rightarrow \left[\text{DTRS}|\text{ H-DTR}\left[\text{LF}|\text{INCONT } \boxed{1}\right]\right]\right)$$

4. [other clauses]

(255) THE `subterm-of-member` PRINCIPLE

$$\forall\boxed{1}\,\forall\boxed{2}$$
$$\left(\begin{array}{l}\texttt{subterm-of-member}(\boxed{1}, \boxed{2}) \leftrightarrow \\ \exists\boxed{3}\left(\texttt{subterm}(\boxed{1}, \boxed{3}) \wedge \texttt{member}(\boxed{3}, \boxed{2})\right)\end{array}\right)$$

In (254), $Q(v, \phi, \psi)$ is the shorthand notation for the description of a generalized quantifier with the VAR value a list v, the RESTR value a list ϕ, and the SCOPE value ψ:

(256)
$$\begin{bmatrix}\textit{gen-quantifier} \\ \text{VAR} \quad v \\ \text{RESTR} \quad \phi \\ \text{SCOPE} \quad \psi\end{bmatrix}$$

The first clause of the SEMANTICS PRINCIPLE concerns phrases in which there is a quantificational determiner. It guarantees that the INCONT value of the noun head is a component of one of the elements on the restriction list of the generalized quantifier, and that the EXCONT value of the head is the generalized quantifier itself. By the first clause of the PROJECTION PRINCIPLE in (252c), the generalized quantifier will then become the EXCONT value of the mother NP. The second clause of the SEMANTICS PRINCIPLE refers to phrases in which the non-head daughter is a quantified NP, and ensures that the INCONT value of the head daughter is a component of the scope of the generalized quantifier carried by the NP. This clause generally applies to phrases with a verbal head daughter.

The third clause of the principle concerns head-marker phrases. For the grammar fragment here I assume that markers have no semantic contribution. Thus the SEMANTICS PRINCIPLE enforces markers to identify their INCONT value with the INCONT value of the head. This clause will be made use of in Section 5.7.

5.3.1.1 An LRS example

Let us use the example below to illustrate how the LRS principles interact in deriving the interpretation of a sentence:

(257) a. A student came.

b. $some(x, student'(x), come'(x))$

We concentrate here on the attribute specifications relevant for the semantics. For more details on syntactic descriptions, the reader is referred back to the examples in Section 2.3.2.4. The sentence in (257a) is associated with the logical interpretation in (257b).

In this example and the one in Section 5.3.2, we do not need polyadic quantifiers yet, so all quantifiers are monadic. This means that the value of VAR is a singleton list of *variable*s, and the value of RESTR is a singleton list of *me*s for these quantifiers. In order to simplify the notation in these examples, we use CONVENTION 5.2 and represent the values for VAR and RESTR directly as objects of sort *variable* and *me*. That is, we dispense with the list notation. This way our representations will be similar to the ones in the LRS literature where only monadic quantifiers are considered (see for instance Richter and Sailer (2004) and Richter and Kallmeyer (2007)). Polyadic quantifiers will be used in the account of Romanian NC starting with Section 5.4.

Convention 5.2 *For a monadic quantifier* $Q((x), (\alpha), \beta)$, *we write directly* $Q(x, \alpha, \beta)$.

In AVM notation: for $\begin{bmatrix} \textit{gen-quantifier} & \\ \text{VAR} & \begin{bmatrix} \textit{nelist} & \\ \text{FIRST} & x \\ \text{REST} & \textit{elist} \end{bmatrix} \\ \text{RESTR} & \begin{bmatrix} \textit{nelist} & \\ \text{FIRST} & \alpha \\ \text{REST} & \textit{elist} \end{bmatrix} \end{bmatrix}$, *we write* $\begin{bmatrix} \textit{gen-quantifier} & \\ \text{VAR} & x \\ \text{RESTR} & \alpha \end{bmatrix}$

In (258), I introduce the relevant parts of the lexical entries for *a*, *student* and *came*.[5,6]

(258) a. $$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \text{a} \rangle \\ \text{SS} \mid \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{det} & \\ \text{SPEC} & \boxed{0}\text{N} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SUBJ} & \langle\rangle \\ \text{SPR} & \langle\rangle \\ \text{COMPS} & \langle\rangle \end{bmatrix} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{INDEX}\mid\text{VAR} & \boxed{1a}\, x \\ \text{MAIN} & \boxed{1}\, some(x, \alpha, \beta) \end{bmatrix} \end{bmatrix} \\ \text{LF} & \begin{bmatrix} \textit{lrs} & \\ \text{EXCONT} & \textit{me} \\ \text{INCONT} & \boxed{1}\, some(x, \alpha, \beta) \\ \text{PARTS} & \langle \boxed{1}, \boxed{1a} \rangle \end{bmatrix} \end{bmatrix} \wedge x \triangleleft \alpha \wedge x \triangleleft \beta$$

[5]The subscript tags in (258) indicate the LOC| CONT| INDEX| VAR value of the *synsem*s.
[6]For simplicity, we ignore the tense property of the verb *came*.

b.

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \texttt{student} \rangle \\ \text{SS }\boxed{0}\text{| LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{SUBJ} & \langle\rangle \\ \text{SPR} & \left\langle \text{Det}_{\boxed{1a}} \right\rangle \\ \text{COMPS} & \langle\rangle \end{bmatrix} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{INDEX | VAR} & \boxed{1a} \\ \text{MAIN} & student' \end{bmatrix} \end{bmatrix} \\ \text{LF} & \begin{bmatrix} \textit{lrs} & \\ \text{EXCONT} & \begin{bmatrix} \textit{gen-quantifier} & \\ \text{VAR} & \boxed{1a} \end{bmatrix} \\ \text{INCONT} & \boxed{2}\, student'(\boxed{1a}) \\ \text{PARTS} & \langle \boxed{2}, \boxed{2a}\, student' \rangle \end{bmatrix} \end{bmatrix}$$

c.

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \texttt{came} \rangle \\ \text{SS| LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{SUBJ} & \left\langle \text{NP}_{\boxed{1a}} \right\rangle \\ \text{SPR} & \langle\rangle \\ \text{COMPS} & \langle\rangle \end{bmatrix} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{MAIN} & \boxed{3a}\, come' \end{bmatrix} \end{bmatrix} \\ \text{LF} & \begin{bmatrix} \textit{lrs} & \\ \text{EXCONT} & \textit{me} \\ \text{INCONT} & \boxed{3}\, come'(\boxed{1a}) \\ \text{PARTS} & \langle \boxed{3}, \boxed{3a} \rangle \end{bmatrix} \end{bmatrix}$$

The semantic contribution of a determiner usually consists of a generalized quantifier and the variable that the quantifier binds. Thus the internal content of *a* in (258a) is the existential quantifier $some(x, \alpha, \beta)$. The EXCONT value is not lexically determined, so it can be any *meaningful expression*. On the PARTS list of the determiner, we include the INCONT value $\boxed{1}$ and the variable x. Two subterm constraints ensure that the variable x is a component of both the restriction (α) and the scope of the quantifier (β).

The lexical entry of a bare noun like *student* specifies that the noun inherits the variable $\boxed{1a}$ of the determiner it subcategorizes for, and that the EXCONT value is a generalized quantifier that binds this variable. The semantic contribution of the noun *student* is the predicate $student'$ as the value of MAIN, and the internal content is the predication $student'(\boxed{1a})$.

The verb *came* in (258c) semantically contributes the predicate $come'$, but its internal content is the predication $come'(\boxed{1a})$, where $\boxed{1a}$ is the variable of the subject the verb subcategorizes for. The EXCONT value is lexically undetermined. On the PARTS list we include the MAIN value ($\boxed{3a}$) and the INCONT value ($\boxed{3}$).

On the basis of the lexical items above, we derive the tree structure in FIGURE 5.2. The application of the LRS principles allows us to specify the lexically undetermined values in (258), and thus to interpret the sentence in (257a).

The structure of the NP *a student* in FIGURE 5.2 is obtained by applying the first clause of the SEMANTICS PRINCIPLE. Thus the EXCONT value of N is identical to the INCONT value $\boxed{1}$ of Det. The subterm constraint $\boxed{2} \triangleleft \alpha$ specifies the INCONT value of N as a subterm of the RESTR value α of the generalized quantifier carried by the Det. By the LRS PROJECTION PRINCIPLE, the NP mother inherits the EXCONT and INCONT values from its head-daughter (N), and its PARTS list collects all

S

$$\begin{bmatrix} \text{EXCONT} & \boxed{4}\, some(x, student'(x), come'(x)) \\ \text{INCONT} & \boxed{3} \\ \text{PARTS} & \left\langle \boxed{1}, \boxed{1a}, \boxed{2}, \boxed{2a}, \boxed{3}, \boxed{3a} \right\rangle \end{bmatrix} \wedge \boxed{3} \lhd \beta \wedge \boxed{1} \lhd \boxed{4}$$

NP

$$\begin{bmatrix} \text{EXCONT} & \boxed{1}\, some(x, \alpha, \beta) \\ \text{INCONT} & \boxed{2} \\ \text{PARTS} & \left\langle \boxed{1}, \boxed{1a}, \boxed{2}, \boxed{2a} \right\rangle \end{bmatrix} \wedge \boxed{2} \lhd \alpha$$

V

$$\begin{bmatrix} \text{EXCONT} & \boxed{4} \\ \text{INCONT} & \boxed{3}\, come'(x) \\ \text{PARTS} & \left\langle \boxed{3}, \boxed{3a}\, come' \right\rangle \end{bmatrix}$$

Det

$$\begin{bmatrix} \text{EXCONT} & \boxed{1} \\ \text{INCONT} & \boxed{1}\, some(x, \alpha, \beta) \\ \text{PARTS} & \left\langle \boxed{1}, \boxed{1a}\, x \right\rangle \end{bmatrix}$$

N

$$\begin{bmatrix} \text{EXCONT} & \boxed{1} \\ \text{INCONT} & \boxed{2}\, student'(x) \\ \text{PARTS} & \left\langle \boxed{2}, \boxed{2a}\, student' \right\rangle \end{bmatrix}$$

Figure 5.2: LRS analysis of (257a) *A student came*

the PARTS elements of the daughters.

The semantic specification of the S node is determined by the second clause of the SEMANTICS PRINCIPLE, which enforces the INCONT value of V to be a subterm of the SCOPE value of the quantifier carried by the NP (i.e. $\boxed{3} \lhd \beta$). The values for the EXCONT, INCONT and PARTS attributes of the S node are given by the LRS PROJECTION PRINCIPLE. The second subterm constraint on the node S ($\boxed{1} \lhd \boxed{4}$) comes from the second clause of the EXCONT PRINCIPLE which requires that all the elements on the PARTS list of an utterance also be subexpressions of the EXCONT value. In our case, $\boxed{1} = \boxed{4}$, because there is only one operator (the quantifier $\boxed{1}$), so there is no scope ambiguity and the sentence receives only one interpretation.

Note that in this section we again used *lists* made up of *synsem* objects, although in writing the $Ty1$ grammar we considered only lists made up of *me*s. In Section 2.3, lists were specified as containing *object*s, so we are free to use any sorts of elements subsumed by *object* on a list.

5.3.2 Polish NC in LRS: Richter and Sailer (2004)

Having illustrated how LRS principles interact to derive the interpretation of an utterance with a monadic generalized quantifier, we can now take a look at how the NC phenomenon can be analyzed in LRS with monadic quantifiers. In particular, I will discuss the approach taken in Richter and Sailer (2004) and Richter and Kallmeyer (2007) to NC in Polish. The data discussion here follows Richter and Sailer (2004), but I adopt the technical adjustments in Richter and Kallmeyer (2007) where quantifiers are represented as generalized quantifiers, so they can easily be used in our grammar fragment.

Polish is usually described as a strict NC language (259b) (see Błaszczak (1999), Przepiórkowski and Kupść (1997), Przepiórkowski and Kupść (1999), Richter and Sailer (1999*b*), Przepiórkowski (1999*b*)). Both the NM and the n-word express negation alone (259a, 259c), like in Romanian. But unlike in Romanian, the presence of two n-words never triggers a DN reading in Polish. The only reading for (259d) is NC:

(259) a. Janek **nie** pomaga ojcu.
Janek NM helps father
'Janek doesn't help his father.'

b. **Nikt** *(**nie**) przyszedł.
nobody NM came
'Nobody came.'

c. Kogo widziałeś? **Nikogo**.
who you-saw? nobody
'Who have you seen? Nobody.'

d. **Nik** **nikomu nie** powiedziałe.
Nothing nobody NM I-told
'I didn't tell anybody anything.' (Richter and Sailer (2004), pp. 107-112)

The LRS structure of sentence (259b) is given in FIGURE 5.3. Following Kupść (2000), Richter and Sailer (2004) assume that the NM *nie* is a prefix, that is, it forms a morphological unit with the verb. The lexical entry for the n-word *nikt* contains a generalized quantifier, thus its LF value resembles the LF value of the NP ***a student*** in FIGURE 5.2:

(260) a. *nikt* ('nobody')

$$\left[\begin{array}{ll} \multicolumn{2}{l}{\textit{word}} \\ \textsc{phon} & \langle \texttt{nikt} \rangle \\ \textsc{ss} \mid \textsc{loc} \mid \textsc{cont} & \left[\begin{array}{ll} \textsc{index} \mid \textsc{var} & \boxed{1b}\ x \\ \textsc{main} & \boxed{3}\ some(x, \alpha, \beta) \end{array}\right] \\ \textsc{lf} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{lrs}} \\ \textsc{excont} & \boxed{3}\ some(x, \alpha, \beta) \\ \textsc{incont} & \boxed{1}\ person'(\boxed{1b}x) \\ \textsc{parts} & \langle \boxed{1}, \boxed{1a}\ person', \boxed{1b}, \boxed{2}\ \neg\gamma, \boxed{3} \rangle \end{array}\right] \end{array}\right] \wedge \boxed{3} \triangleleft \gamma \wedge \boxed{1} \triangleleft \alpha \wedge x \triangleleft \alpha \wedge x \triangleleft \beta$$

b. *nie przyszedł* ('NM came')

$$\left[\begin{array}{ll} \multicolumn{2}{l}{\textit{word}} \\ \textsc{phon} & \langle \texttt{nie przyszedl} \rangle \\ \textsc{ss} \mid \textsc{loc} & \left[\begin{array}{ll} \textsc{cat} \mid \textsc{val} \mid \textsc{subj} & \left\langle \mathrm{NP}_{\boxed{1b}} \right\rangle \\ \textsc{cont} \mid \textsc{main} & \boxed{4a}\ come' \end{array}\right] \\ \textsc{lf} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{lrs}} \\ \textsc{excont} & \boxed{0} \\ \textsc{incont} & \boxed{4}\ come'(\boxed{1b}) \\ \textsc{parts} & \langle \boxed{4}, \boxed{4a}\ come', \boxed{5}\ \neg\eta \rangle \end{array}\right] \end{array}\right] \wedge \boxed{4} \triangleleft \eta \wedge \boxed{5} \triangleleft \boxed{0}$$

Richter and Sailer (2004) and Richter and Kallmeyer (2007) do not make direct use of negative generalized quantifiers: they represent a negative quantifier as an existential generalized quantifier preceded by logical negation, as in (260a). The external content of the n-word only contains the existential quantifier, although logical negation is also an element on the PARTS list of the n-word and it must outscope the EXCONT value ($\boxed{3} \triangleleft \gamma$). This ensures that the existential quantifier is always outscoped by negation.

In the lexical specification of the verb *nie przyszedł* the logical negation on the PARTS list represents the semantic contribution of the prefix *nie*. The first constraint ($\boxed{4} \triangleleft \eta$) states that the semantics of the verb is in the scope of the negative operator. Unlike in the case of *nikt*, the negative operator has to be a subexpression of the EXCONT value of the negated verb ($\boxed{5} \triangleleft \boxed{0}$). This way, the scope of negation is restricted to the clause headed by the verb. Apart from negation and the subterm constraints associated with it, the semantic specification of the verb (i.e. $come'(x)$) is similar to that of the affirmative verb *przyszedł* given in (258c) for the English counterpart *came*.

FIGURE 5.3 gives the semantic structure of the sentence *Nikt nie przyszedł*. Just like in the case of *A student came* (FIGURE 5.2), the second clause of the SEMANTICS PRINCIPLE adds the constraint by which the INCONT value of the verb must be a subpart of the scope of the quantifier contributed by the NP ($\boxed{4} \triangleleft \beta$). The second clause of the EXCONT PRINCIPLE requires all the PARTS elements to be subterms of the EXCONT value of an utterance, thus $\boxed{2}$ must be a subterm of $\boxed{0}$:

S

$$\begin{bmatrix} \text{EXC} & \boxed{0} \\ \text{INC} & \boxed{4} \\ \text{PARTS} & \left\langle \boxed{1}, \boxed{1a}, \boxed{1b}, \boxed{2}, \boxed{3}, \boxed{4}, \boxed{4a}, \boxed{5} \right\rangle \end{bmatrix} \wedge \boxed{4} \triangleleft \beta \wedge \boxed{2} \triangleleft \boxed{0}$$

NP

$$\begin{bmatrix} \text{EXC} & \boxed{3}\, some(x,\alpha,\beta) \\ \text{INC} & \boxed{1}\, person'(\boxed{1b}x) \\ \text{PARTS} & \left\langle \boxed{1}, \boxed{1a}\, person', \boxed{1b}, \boxed{2}\, \neg\gamma, \boxed{3} \right\rangle \end{bmatrix} \wedge \boxed{3} \triangleleft \gamma \wedge \boxed{1} \triangleleft \alpha$$

V

$$\begin{bmatrix} \text{EXC} & \boxed{0} \\ \text{INC} & \boxed{4}\, come'(\boxed{1b}) \\ \text{PARTS} & \left\langle \boxed{4}, \boxed{4a}\, come', \boxed{5}\, \neg\eta \right\rangle \end{bmatrix} \wedge \boxed{4} \triangleleft \eta \wedge \boxed{5} \triangleleft \boxed{0}$$

Figure 5.3: LRS analysis of (259b) *Nikt nie przyszedł*

For the structure above, our LRS theory allows three possibilities to disambiguate the EXCONT value $\boxed{0}$, listed below:

(261) a. $\neg\neg\, some(x, person'(x), come'(x)) = some(x, person'(x), come'(x))$ (DN)

i. $\boxed{5} = \boxed{0} \wedge \boxed{2} = \eta \wedge \boxed{3} = \gamma$ or

ii. $\boxed{2} = \boxed{0} \wedge \boxed{3} = \eta \wedge \boxed{5} = \gamma$

b. $\neg\, some(x, person'(x), \neg\, come'(x))$ (DN)

$\boxed{2} = \boxed{0} \wedge \boxed{3} = \gamma \wedge \boxed{5} = \beta$

c. $\neg some(x, person'(x), come'(x))$ (NC)

$\boxed{5} = \boxed{2} = \boxed{0} \wedge \boxed{3} = \gamma = \eta$

The EXCONT value $\boxed{0}$ in FIGURE 5.3 depends on the scope interaction between the two negative expressions $\boxed{2}$ and $\boxed{5}$. The one that contains/ outscopes the other gets identified with $\boxed{0}$. The interpretation in (261a) is obtained by interpreting the negative quantifier $\boxed{2}$ in the scope of the negative expression $\boxed{5}$ ($\boxed{2} = \eta$). This way the verbal negation has widest scope. The same interpretation can be obtained if $\boxed{5}$ is in the scope of $\boxed{2}$ ($\boxed{5} = \gamma$), but outscopes the existential quantifier ($\boxed{3} = \eta$). In this case, the negation contributed by the quantifier has widest scope, but the existential quantifier is outscoped by the verbal negation. In the second interpretation (261b) the verbal negation gets narrowest scope since it appears in the scope of the existential quantifier ($\boxed{5} = \beta$). The interpretation in (261c) comes from imposing token-identity between $\boxed{2}$ and $\boxed{5}$, and thus making the two negations identical. This last reading is actually the only one available for our sentence.

In order to exclude the two unavailable readings in (261a) and (261b), Richter and Sailer (2004) posit the constraint below:

(262) THE NEGATION COMPLEXITY CONSTRAINT

For each *sign*, there may be at most one negation that is a component of the EXCONT value and has the MAIN value as its component.

The NEGATION COMPLEXITY CONSTRAINT is language-specific. Since Polish does not allow double negation readings, there may be at most one sentential negation. Richter and Sailer formulate this constraint in the spirit of various linguistic generalizations, according to which languages of the world present a general strategy to minimize the number of semantic negations in a clause and this strategy gets grammaticalized at a certain threshold (see for instance Corblin (1995) for French and Corblin and Tovena (2001) for other Romance languages). While for French this threshold is set to be two negations, for Polish it is only one negation.

An important characteristic of NC in Polish is the obligatory presence of the NM. This was indicated in (259b) where the absence of the NM would yield ungrammaticality. Richter and Sailer (2004) account for this fact by positing a principle that resembles the NEG CRITERION introduced in Zanuttini (1991) and Haegeman and Zanuttini (1991):

(263) THE NEG CRITERION

For every finite verb, if there is a negation in the external content of the verb that has scope over the verb's MAIN value, then the negation must be an element of the verb's PARTS list.

While the NEG CRITERION of Haegeman and Zanuttini is syntactic in nature, Richter and Sailer formulate it as a constraint on semantic representations. To understand how it works, let us go back to our example. In FIGURE 5.3, a wide scope negation under the EXCONT value of the node S could also come from the quantifier alone and by the PROJECTION PRINCIPLE, it would appear on the EXCONT value of the verbal head. This negation would have scope over the verb's MAIN value $come'$. However, it would not appear on the PARTS list of the verb if the verb were not negative. Sentences in which a negation outscopes a lexically affirmative verb are ungrammatical in Polish. The NEG CRITERION regulates this by only allowing negation to outscope the MAIN value of a verb if the verb itself is negative (i.e. it has negation on its PARTS list).

Conclusion The analysis of Polish NC in Richter and Sailer (2004) and Richter and Kallmeyer (2007) heavily relies on the underspecification strategies within LRS and the HPSG-specific mechanism of token-identity. The interaction between token-identity and the NEGATION COMPLEXITY CONSTRAINT for Polish ensure that only a NC reading is available for a Polish sentence with at least two negative expressions.

5.4 NC as resumption in LRS

We have seen how the LRS principles interact to account for a phenomenon like NC with underspecification means. In this section I present a way to use resumptive quantifiers in the analysis of NC. More precisely, I integrate the semantic analysis of NC in Section 4.2 within LRS.

In the first part of Chapter 4, I showed how the polyadic lifts *resumption* and *iteration* can account for NC and DN readings in Romanian within the Generalized Quantifiers Theory. We concluded that these polyadic lifts as defined in GQT cannot be integrated in a compositional grammar. But the interpretations that we derive with iteration and resumption can be obtained even if we do not make explicit use of the corresponding polyadic lifts. The interpretation derived by means of iteration can easily be obtained in LRS by allowing one of the two monadic quantifiers take scope over the other (e.g. (261a), (261b)). The interpretation of n-ary resumption of a quantifier Q is the interpretation of Q^n, which is provided by the language $Ty1$ and the corresponding grammar Γ_{Ty1}. Now we are going to use these two alternatives in analyzing the iterative and the resumptive interpretations of Romanian negative quantifiers within LRS.

In this section I concentrate on the way we can account for the NC and the DN readings of the two sentences below:

(264) a. **Niciun** student nu a citit **nicio** carte.
no student NM has read no book

i. 'No student read any book.' (NC)
ii. 'No student read no book. (Every student read some book.)' (DN)

b. **Niciun** student nu a citit *frecvent* **nicio** carte.
no student NM has read frequently no book

i. 'There is no student and no book, such that the former read the latter frequently.' (NC)
ii. 'It was frequently the case that no student read any book.' (NC)
iii. 'For no student was it frequently the case that s/he read no book.' (DN)

With the analysis of the sentence in (264b), I propose a way to account for the scope properties of Romanian negative quantifiers interacting with non-negative quantifiers described in Section 3.5.3. In Section 5.4.1 I analyze the NC readings of the sentences in (264) and in Section 5.4.2 I address the DN readings. For now, I take only n-words into consideration. The NM will be addressed in Section 5.5.

5.4.1 The NC reading

We start with the lexical information on the words in (264a): *niciun*, *student*, *nu a citit*, *nicio*, *carte*. In (265) below, I concentrate on the lexical information that is relevant for our semantic analysis, i.e. the one under the attributes SS|LOC|CONT and LF. The syntactic information (under SS|LOC|CAT and DTRS) is similar to that in Section 2.3.2.4, p. 60. The determiners *niciun* and *nicio* only differ with respect to gender, which Sailer (2004, p. 208) places under SS|LOC|CONT|INDEX|PHI|GENDER in the local semantics and which has no influence on the compositional semantics that we are interested in, so I give only the lexical entry for *niciun*. Similarly, the lexical entry of the noun *carte* carries similar semantic information to *student*, so I provide it directly in the tree in FIGURE 5.5.

(265) a. *niciun* (‘no’)

$$\left[\begin{array}{l} \textit{word} \\ \text{PHON}\ \langle \texttt{niciun} \rangle \\ \text{SS} \mid \text{LOC} \left[\begin{array}{l} \text{CAT} \mid \text{HEAD} \left[\begin{array}{l} \textit{det} \\ \text{SPEC}\ \ \text{N}_{\boxed{1a}} \end{array}\right] \\ \text{CONT} \left[\begin{array}{l} \text{INDEX} \mid \text{VAR}\ \boxed{1a}\, x \\ \text{MAIN}\ \boxed{1}\, no(v,\alpha,\beta) \end{array}\right] \end{array}\right] \\ \text{LF} \left[\begin{array}{ll} \textit{lrs} & \\ \text{EXC} & \textit{me} \\ \text{INC} & \boxed{1}\, no(v,\alpha,\beta) \\ \text{PARTS} & \langle \boxed{1}, \boxed{1a}\, x \rangle \end{array}\right] \end{array}\right] \wedge x \in v \wedge x \lhd_{\in} \alpha \wedge x \lhd \beta$$

b. *student* (‘student’)

$$\left[\begin{array}{l} \textit{word} \\ \text{PHON}\ \langle \texttt{student} \rangle \\ \text{SS}\ \boxed{7} \mid \text{LOC} \left[\begin{array}{l} \text{CAT} \left[\begin{array}{l} \text{HEAD}\ \textit{noun} \\ \text{VAL} \mid \text{SPR}\ \ \left\langle \text{DetP}_{\boxed{1a}} \right\rangle \end{array}\right] \\ \text{CONT} \left[\begin{array}{l} \text{INDEX} \mid \text{VAR}\ \boxed{1a} \\ \text{MAIN}\ \boxed{2a}\, student' \end{array}\right] \end{array}\right] \\ \text{LF} \left[\begin{array}{ll} \textit{lrs} & \\ \text{EXC} & \textit{gen-quantifier} \\ \text{INC} & \boxed{2}\, student'(\boxed{1a}) \\ \text{PARTS} & \langle \boxed{2}, \boxed{2a} \rangle \end{array}\right] \end{array}\right]$$

c. *nu a citit* (‘NM has read’, without the contribution of the NM)

$$\left[\begin{array}{l} \textit{word} \\ \text{PHON}\ \langle \texttt{nu a citit} \rangle \\ \text{SS} \mid \text{LOC} \left[\begin{array}{l} \text{CAT} \mid \text{VAL} \left[\begin{array}{l} \text{SUBJ}\ \ \left\langle \text{NP}_{\boxed{1a}} \right\rangle \\ \text{COMP}\ \left\langle \text{NP}_{\boxed{6a}} \right\rangle \end{array}\right] \\ \text{CONT} \mid \text{MAIN}\ \ \boxed{3a}\, read' \end{array}\right] \\ \text{LF} \left[\begin{array}{ll} \textit{lrs} & \\ \text{EXC} & \boxed{0} \\ \text{INC} & \boxed{3}\, read'(\boxed{1a}, \boxed{6a}) \\ \text{PARTS} & \langle \boxed{3}, \boxed{3a} \rangle \end{array}\right] \end{array}\right] \wedge \boxed{3} \lhd \boxed{0}$$

Note that for now we treat *nu a citit* ‘not has read’ as an affirmative verb, so we ignore the semantic contribution of the NM, which is addressed in Section 5.5. Both the auxiliary verb *a* ‘has’ and the NM *nu* have affixal status, so the verb form *nu a citit* is a *word*, the output of a lexical rule, and not a *phrase*. The affixal status of auxiliary verbs in Romanian is argued for in Barbu (1999). For the affixal status of the NM, motivation will be provided in Section 5.5.2.

The negative determiner *niciun* has the semantics of a negative generalized quantifier which appears as the internal content value. Its lexical entry is similar to that of the determiner *a* in (258a). But recall that (258a) was simplified, because we only dealt with monadic quantifiers and we used the meaningful expression value instead of the singleton list (*variable* for a list of variables under VAR and *me* for a list of meaningful expressions under RESTR). If we use resumptive quantifiers, we allow a generalized quantifier to bind more than just one variable so we have to represent the value for the attributes VAR and RESTR as lists of *variable*s and *me*s, as specified in the $Ty1$ grammar.

In (265a), we have to distinguish between the one variable which the determiner contributes itself (i.e. the variable $\boxed{\text{1a}}\ x$) and the list v of variables – possibly including variables contributed by other determiners – that the quantifier operator may bind. This distinction correlates with the one between the local and the compositional semantics (cf. Sailer (2004)). The determiner alone contributes the variable x under its local semantics, i.e. SS| LOC| CONT| INDEX| VAR, and this is the value that gets identified with the variable of which the common noun *student* predicates the student property (see also the value of SS| LOC| CAT| HEAD| SPEC in (265a), the place where the two variables get identified). It is the variable x that the agreement information concerning number, person and gender under SS| LOC| CONT| INDEX| PHI is posited of. But v, the list of variables that the quantificational operator binds, has to do with the compositional semantics, the way the quantifier interacts with the other quantifiers within an utterance possibly building a polyadic quantifier together. This list of variables appears under LF| INC| VAR. To ensure that the local variable x introduced by the determiner gets bound by the quantifier contributed by the same determiner, we add the constraint that x is a member of the list v: $x \in v$. The local variable x is also the one that appears on the PARTS list of the determiner. The other two constraints in the lexical entry of *niciun* ($x \triangleleft_{\in} \alpha$, $x \triangleleft \beta$) ensure that the restriction and the nuclear scope of the polyadic quantifier also contain the variable x.

Given the lexical entries for the determiner and the noun, we can derive the LF value of the NP *niciun student* in FIGURE 5.4. In view of the first clause of the EXCONT PRINCIPLE, the EXC(ONT) value of the determiner is identified with its INC(ONT) value: the EXC value must be an element of the PARTS list and since by the INCONT PRINCIPLE the INC value is a component of the EXC value, the two become equal in an NP. The first clause of the SEMANTICS PRINCIPLE enforces the identity between the EXC value of the noun and the INC value of the determiner and the fact that the INC value of the noun must be a subterm of a member of the RESTR list of the polyadic quantifier ($\boxed{2} \triangleleft_{\in} \alpha$). The LRS PROJECTION PRINCIPLE determines the EXC and INC values of the NP as identical to the EXC and INC values of the noun (the head-daughter) and the PARTS list of the NP as collecting all the parts of the daughters. The NP *nicio carte* is derived in a similar way to *niciun student* in FIGURE 5.4, so I introduce it directly in the tree in FIGURE 5.5.

NP

$$\begin{bmatrix} \text{EXC} & \boxed{1} \\ \text{INC} & \boxed{2} \\ \text{PARTS} & \left\langle \boxed{1}, \boxed{\text{1a}}, \boxed{2}, \boxed{\text{2a}} \right\rangle \end{bmatrix} \wedge \boxed{2} \triangleleft_{\in} \alpha$$

Det

$$\begin{bmatrix} \text{EXC} & \boxed{1} \\ \text{INC} & \boxed{1}\ no(v, \alpha, \beta) \\ \text{PARTS} & \left\langle \boxed{1}, \boxed{\text{1a}}\ x \right\rangle \end{bmatrix}$$

N

$$\begin{bmatrix} \text{EXC} & \boxed{1}\ no(v, \alpha, \beta) \\ \text{INC} & \boxed{2}\ student'(\boxed{\text{1a}}\ x) \\ \text{PARTS} & \left\langle \boxed{2}, \boxed{\text{2a}}\ student' \right\rangle \end{bmatrix}$$

Figure 5.4: LRS analysis of *niciun student*

On the basis of the lexical items above and the model of deriving NPs, we can now represent the combinatorics of the sentence (264a) in FIGURE 5.5.

In this tree, at the VP level the second clause of the SEMANTICS PRINCIPLE imposes the constraint that the INC value of the verb be a subterm of the nuclear scope of the NP, i.e. $\boxed{3} \triangleleft \psi$. Due to the LRS PROJECTION PRINCIPLE, the EXC and the INC values of the VP are identical to those of

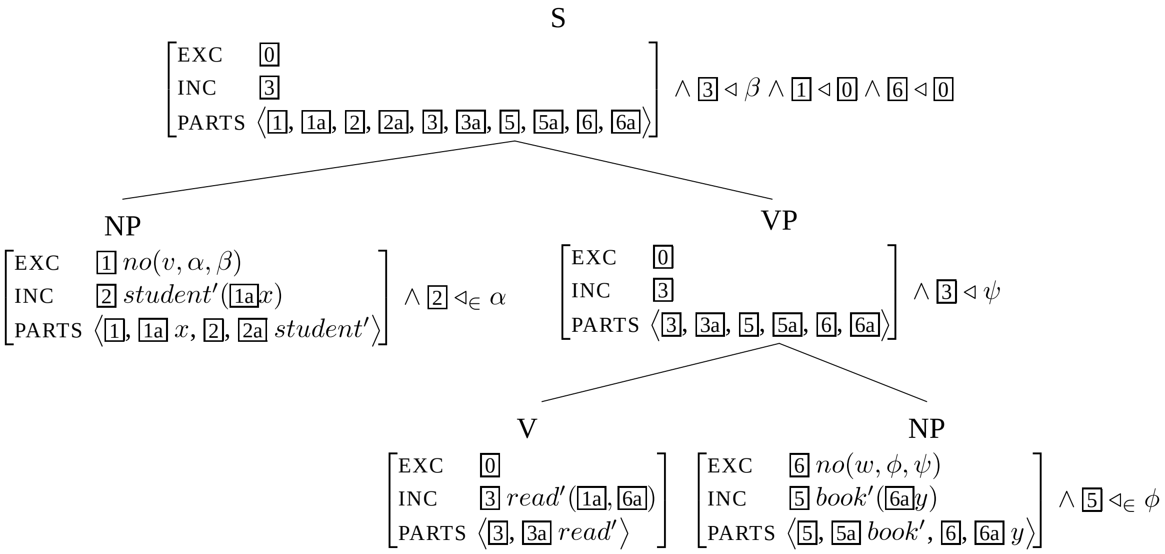


Figure 5.5: LRS analysis of *Niciun student nu a citit nicio carte* (without the NM)

the V and the PARTS list collects all the PARTS elements of the two daughters. At the S level, the same principles apply with parallel effects. Moreover, the second clause of the EXCONT PRINCIPLE requires that the EXC values of the two quantifiers, as members of the PARTS list, be subterms of the EXC value of the sentence (see $\boxed{1} \triangleleft \boxed{0}$, $\boxed{6} \triangleleft \boxed{0}$).

Interpretations The possible interpretations for the sentence in FIGURE 5.5 depend on the value $\boxed{0}$ of the EXC attribute on the S node. In order to determine $\boxed{0}$, we have to take into account the scope interaction between the two negative quantifiers $\boxed{1}$ and $\boxed{6}$ contributed by the NPs. The subterm constraints in FIGURE 5.5 in combination with the grammar of $Ty1$ expressions in Section 5.2 lead to the following possible values of $\boxed{0}$:

(266) a. $no(x, student(x), no(y, book(y), read(x, y)))$ (DN)

$\boxed{0} = \boxed{1} \wedge \boxed{6} \triangleleft \beta$

b. $no((x, y), (student(x), book(y)), read(x, y))$ (NC)

$\boxed{0} = \boxed{1} = \boxed{6}$

Given the two subterm conditions on $\boxed{1}$ and $\boxed{6}$ as subterms of $\boxed{0}$, the decisive factor for the interpretation of the sentence is the relation between the EXC values of the two quantifiers, i.e. the scope interaction between them. There are two possibilities: either one of the quantifiers is a subterm of the nuclear scope of the other, or their EXC values are identified (i.e. we have token-identity between $\boxed{1}$ and $\boxed{6}$), so they are equal. In the first case two negations are contributed to the interpretation, so we get a DN reading like in (266a). In the second case the underspecified values of the two quantifiers become identical, so they contribute one resumptive negative quantifier and a NC reading obtains.

The reader may note that in (266) I only considered the case when the quantifier contributed by the subject NP has wide scope (in DN) or its variable appears first (in the resumptive quantifier). For NC, the order of the variables does not trigger a difference in interpretation.[7] The other possibility of

[7]For more discussion on the order of the variables in a resumptive quantifier and on the status of variables in LRS see Section 5.6.

ordering the variables x, y in (266b) yields a NC interpretation which is truth-conditionally equivalent to this one (see our earlier discussion on the scope neutrality of the negative quantifiers in a NC reading, Section 4.2). For the DN readin the variation in the scope order of the negative quantifiers leads to different interpretations (Section 4.1), so $\boxed{0}$ may take one more value, different from (266a), which also yields a DN reading. But as we know from Chapter 3, the DN interpretation only appears under special contextual conditions. These conditions will be addressed in Section 5.4.2 and that analysis will also cast light on the question whether the utterance may be ambiguous between two different DN readings.

For now we retain the fact that the NC reading of a sentence with two n-words may be obtained by enforcing token-identity between the negative quantifiers, which thus contribute one resumptive negative quantifier together. As illustrated in Section 5.3.2, the same mechanism is used in Richter and Sailer (2004) and Richter and Kallmeyer (2007) to account for Polish NC. The difference is that for Polish they use a higher-order logic with monadic quantifiers, while the analysis here employs polyadic quantifiers.

5.4.2 The DN reading

Let us now concentrate on the DN readings available for sentence (264a) represented in FIGURE 5.5. Depending on which quantifier has wide scope, we obtain the following two values for $\boxed{0}$:

(267) a. $no(x, student(x), no(y, book(y), read(x, y)))$ (DN)

$\boxed{0} = \boxed{1} \wedge \boxed{6} \triangleleft \boxed{1}$

b. $no(y, book(y), no(x, student(x), read(x, y)))$ (DN)

$\boxed{0} = \boxed{6} \wedge \boxed{1} \triangleleft \boxed{6}$

The first value is the one in (266a), where the quantifier contributed by the subject has wide scope. The second one appears if the quantifier contributed by the direct object has wide scope.

As discussed in Section 3.4.2, a DN reading is available for this sentence only in a denial context where one n-word is used to deny a previous utterance that contains the other n-word. Such a context is provided in (268) for the interpretation in (267a) and in (269) for (267b):

(268) a. Speaker A: Un student nu a citit **nicio carte**.
one/a student NM has read no book
'One/a student read no book.'

b. Speaker B: **NIciun stuDENT** nu a citit **nicio carte**.
no student NM has read no book
'No student read no book. (= Every student read some book.)'

(269) a. A: **Niciun student** nu a citit "Nostalgia".
no student NM has read "Nostalgia"
'No student read "Nostalgia".'

b. B: ? **Niciun student** nu a citit **NIcio CArte**.
no student NM has read no book
'No book was read by no student. (= Every book was read by some student.)'

Theoretically both readings in (267) should be equally available, but in practice the second one is more difficult to obtain. This has to do with the general conditions on quantifier scope in Romanian, already discussed in Section 3.5: the leftmost quantifier takes wide scope. The inverse scope is available under special information structural conditions which can be provided for DN (see (269)). But given the limited availability of DN in a NC language, the sentence in (269b) is less natural than the one in (268b). For the rest of this discussion I concentrate on (268b) and the corresponding interpretation in (267a).

In information structure terms, the n-word *niciun* in (268b) carries contrastive focus (cf. also Göbbel (1995)) and the second n-word is part of the background that is negated. *Niciun* has a high pitch accent followed by a low accent and the rest of the sentence is deaccented (Göbbel (2003), p.c.). Note however that an n-word with informational focus can also trigger DN readings, if the background information contains another n-word. This is the case in (sentential) answers to negative wh-questions like (270a):

(270) a. Cine nu a citit **nicio carte**?
who NM has read no book
'Who read no book?'

b. **NImeni** nu a citit **nicio carte**.
nobody NM has read no book
'Nobody read no book.' (DN)

c. **Nimeni**. (DN)
Nobody.

Although the answer in (270c) also receives a DN interpretation, I will not address it in this analysis. This DN interpretation is obtained only in relation to the preceding question: if the question were *who read the book?*, the answer would be interpreted as simply negative. Thus the interpretation of an n-word in a fragmentary answer is determined in the discourse and not within one utterance alone. In (270b) the preceding question motivates the information structural status of the two n-words, but the DN interpretation is only dependent on this status and not on the previous question: if the two n-words receive the appropriate accent, the DN reading is available. This is also the case in denial contexts like (268).

While an exhaustive characterization of the information structural and phonological particularities of DN readings is not the principal aim here, it is important for the present analysis to correctly describe the situations in which a sentence like (264a) receives a NC reading and those where it receives a DN one. The two readings exclude each other on information structural grounds. In what follows I propose a sketch of these conditions.

Information structure in HPSG To incorporate the above information structural conditions into our HPSG analysis of NC and DN readings in Romanian, I concentrate on the HPSG architecture of information structure developed in Engdahl and Vallduví (1996), De Kuthy (2002), and De Kuthy and Meurers (2006).

Engdahl and Vallduví (1996) propose an attribute INF(ORMATION)-STR(UCTURE) to integrate the information structure specification of a sign. The location of this attribute has been subject to dispute: while Engdahl and Vallduví (1996) assume that it is appropriate for *context* objects as values of a local attribute SS| LOC| CTXT, De Kuthy (2002) places it at the *sign*-level. Otherwise, the information structure specification as a local attribute would be shared between fillers and gaps in a trace-based analysis of unbounded dependencies (see for instance, Pollard and Sag (1994), Sag et al. (2003)). This would predict that gaps carry information structure specification, which is theoretically dubious and impossible to test. In the present analysis we are not concerned with unbounded dependency phenomena, so no appeal to gaps needs to be made. But we should not exclude the possibility of introducing gaps if needed, so I follow De Kuthy (2002) and place the attribute INF-STR at the *sign*-level with *inf-str* objects as a value:

(271)
$$\begin{bmatrix} \textit{sign} & \\ \text{INF-STR} & \begin{bmatrix} \textit{inf-str} & \\ \text{FOC} & \textit{list} \\ \text{TOP} & \textit{list} \\ \text{BKGR} & \textit{list} \end{bmatrix} \end{bmatrix}$$

I assume three attributes to characterize *inf-str* objects: FOC(US), TOP(IC) and B(AC)KGR(OUND). Under FOC I include both contrastive focus (as in denial contexts) and informational focus (as in answers to wh-questions), since they have parallel effects with respect to the DN reading.[8] The separation between TOP and BKGR allows us to distinguish between topicalized constituents like *cartea asta* 'this book' in (272) and the non-topicalized old information carried by the NP *nicio carte* 'no book' in (268b), for instance. This difference minimally accounts for the phonological contrast between the two: while a topic carries a rising accent (i.e. a low accent immediately followed by a high one), the NP *nicio carte* in (268b) is deaccented. In HPSG terms *cartea asta* in (272) has a non-empty TOP specification and *nicio carte* in (268b) carries non-empty BKGR information.

(272)

Cartea asta	nu	a	citit-o	nimeni.
book this	NM	has	read-it	nobody

'Nobody read this book.'

The value of FOC, TOP, BKGR As indicated in (271), I take the values of the information structure features to be lists of objects (cf. De Kuthy (2002) and De Kuthy and Meurers (2006)). This way one can account for multiple foci as in answers to multiple wh-questions, for instance:

(273) A: Who bought what?
B: JOHN bought a BOOK.

The kind of objects that appear on these lists represent another subject of debate. There have been two proposals: Engdahl and Vallduví (1996) take them to be *sign*s, while De Kuthy (2002) and De Kuthy and Meurers (2006) consider them to be logical objects. Against the former proposal, doubts have been raised concerning the relevance of the syntactic specification of a *sign* for information structure. De Kuthy (2002) follows Kuhn (1995) and assumes that the objects forming the value of the information structure attributes are of a semantic nature. For De Kuthy, they are *meaningful expressions* compositionally derived in the LF-Ty2 representations of Sailer (2003).

In this chapter we use LRS as a semantic formalism, so I assume that the lists in (271) are made up of *lrs* objects. Note that having only EXC or INC values as members of these lists would not allow us to distinguish between a focused V, a focused VP and a focused S as exemplified in (274), since they would have the same EXC/ INC value (as required by the SEMANTICS PRINCIPLE, cf. the V, VP and S nodes in FIGURE 5.5). Since *lrs* objects also include the PARTS list which disambiguates between words and phrases, the LF values of the V, VP and S in (274) will be different and will correctly identify the focused material.[9]

[8]In a closer investigation (even with respect to n-words), a distinction between the two may however turn out to be important (see for instance Göbbel (1995) and references therein).

[9]Another option would be to assume that the objects on the information structure lists are of sort *content*, so they coincide with the local semantics of a sign. But it is not clear how this semantic specification is built for phrases. The

(274) a. A: What is new?
B: [John read the BOOK]$_F$.

b. A: What did John do?
B: John [read the BOOK]$_F$.

c. A: What did John do with the book?
B: John [READ]$_F$ the book.

To ensure that the objects on the list values of the information structure attributes in (271) are of sort *lrs*, we posit the constraint below on objects of sort *inf-str*:

(275)
$$inf\text{-}str \rightarrow \left(\begin{bmatrix} \text{FOC} & \boxed{1} \\ \text{TOP} & \boxed{2} \\ \text{BKGR} & \boxed{3} \end{bmatrix} \wedge \texttt{lrs-list}(\boxed{1}) \wedge \texttt{lrs-list}(\boxed{2}) \wedge \texttt{lrs-list}(\boxed{3}) \right)$$

(276) THE `lrs-list` PRINCIPLE

$$\forall\boxed{1} \left(\begin{array}{l} \texttt{lrs-list}(\boxed{1}) \leftrightarrow \\ \left(\begin{array}{l} \boxed{1}\,[elist] \vee \\ \exists\boxed{2}\,\exists\boxed{3}\left(\boxed{1}\begin{bmatrix}\text{FIRST} & \boxed{2}\,[lrs] \\ \text{REST} & \boxed{3}\end{bmatrix} \wedge \texttt{lrs-list}(\boxed{3})\right)\end{array}\right)\end{array}\right)$$

The projection of information structure In determining the INF-STR of phrases, I assume a principle according to which a mother node collects all the FOC/TOP/BKGR values of its daughters. Recall that our HPSG phrase structure rules (ID-Schemata) in Section 2.3 are formulated with binary branching, so phrases have only one non-head daughter (SUBJ-DTR, SPR-DTR, COMP-DTR, ADJ-DTR or MRK-DTR).[10]

(277) THE INF-STR PROJECTION PRINCIPLE

$$phrase \rightarrow \left(\begin{array}{l} \begin{bmatrix} \text{INF-STR} & \begin{bmatrix} \text{FOC} & \boxed{1} \oplus \boxed{2} \\ \text{TOP} & \boxed{3} \oplus \boxed{4} \\ \text{BKGR} & \boxed{5} \oplus \boxed{6} \end{bmatrix} \\ \text{DTRS}\ \boxed{7} & \begin{bmatrix} \text{H-DTR} & \begin{bmatrix} \text{INF-STR} & \begin{bmatrix} \text{FOC} & \boxed{1} \\ \text{TOP} & \boxed{3} \\ \text{BKGR} & \boxed{5} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \wedge\ \texttt{non-hd-dtr}\left(\boxed{7}, \boxed{8}\begin{bmatrix} \text{INF-STR} & \begin{bmatrix} \text{FOC} & \boxed{2} \\ \text{TOP} & \boxed{4} \\ \text{BKGR} & \boxed{6} \end{bmatrix} \end{bmatrix}\right) \end{array} \right)$$

The principle in (277) suffices to describe the information structural conditions on the two n-words in a sentence that receives a DN interpretation, but the reader is referred to Engdahl and Vallduví

usual assumption is that it is inherited from the head daughter. This specification would then raise similar problems as the ones indicated for the EXC/ INC values with respect to (274). In any case, a different assumption here would not affect the present analysis of DN.

[10] See De Kuthy (2002) for a similar principle for flat structures with several non-head daughters.

(1996), De Kuthy (2002), De Kuthy and Meurers (2006) and references therein, for a discussion of various complications that arise with focus/ topic projections, including the differentiation between the various constructions in (274), for instance.

The DN principle Having provided this apparatus to describe information structure in HPSG, we can now formulate the information structure constraint on DN readings in Romanian. The DN reading only occurs when one n-word carries focus and has a falling accent, while the other n-word belongs to the background and is deaccented. I will not address the accents here, since there is a clear association between the falling accent and focus, as well as between deaccentuation and background.[11] So the kind of information structure that the n-word bears is enough to indicate its accent.[12]

In FIGURE 5.6, I represent the INF-STR information of the sentence in (264a) under its most natural DN reading (i.e. with the first n-word in linear order taking wide scope: (267a)). The NP quantifier in the object position has a BKGR contribution ($\boxed{7}$), while the one in the subject position has a FOC contribution ($\boxed{6}$). Note that in (268) only the determiner *niciun* is in focus, but the whole NP *nicio carte* belongs to the background. Both values are transmitted to the S node by means of the INF-STR PROJECTION PRINCIPLE.

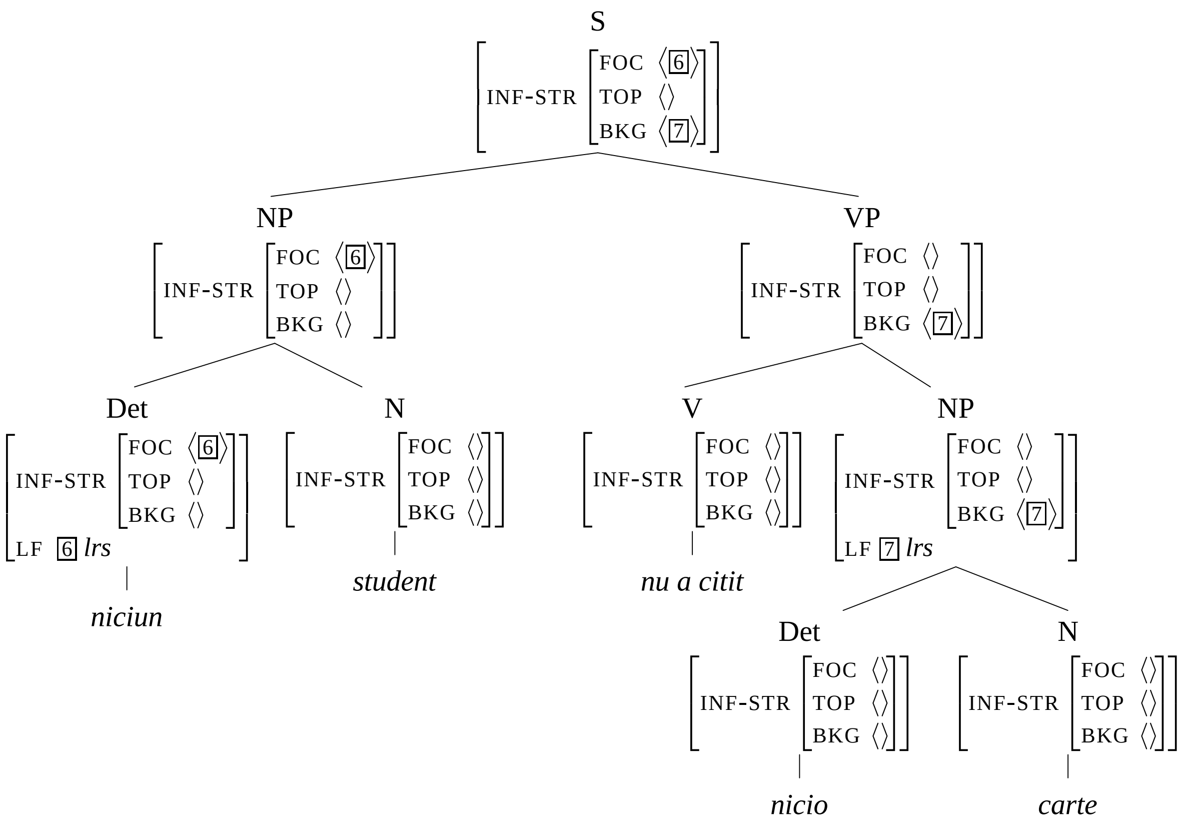


Figure 5.6: INF-STR analysis of *Niciun student nu a citit nicio carte* (DN reading)

Given the INF-STR specification of the sentence in its DN reading, we can now formulate the

[11] But see Göbbel (2003) for a discussion on accents in Romanian, De Kuthy (2002) and De Kuthy and Meurers (2006) for how accents can be integrated in HPSG grammars.

[12] See Maekawa (2004) for a more complex account on the interaction between focus and word order by means of linearization principles.

DN PRINCIPLE in (278). The antecedent of the DN PRINCIPLE introduces the two signs (Dets, NPs or PPs) contributing the quantifiers as having empty SPR and COMPS values and each carrying a negative quantifier. It also specifies the DN interpretation of an utterance, i.e. there are two negative quantifiers $\boxed{3}$ and $\boxed{4}$, such that the latter is a subterm of the nuclear scope of the other ($\boxed{4} \lhd \beta$). Since our observations only concern DN in full utterances, we limit this constraint to (*u(nembedded)-signs*).[13,14]

(278) THE DN PRINCIPLE

$$
\forall\boxed{1}\ \forall\boxed{2}\ \forall\boxed{3}\ \forall\boxed{4}
\left(\begin{array}{l}
\left(\begin{array}{l}
\textit{u-sign} \wedge \boxed{1}\begin{bmatrix} \text{SS}|\ \text{LOC}|\ \text{CAT}\begin{bmatrix}\text{HEAD}\ \textit{det} \vee \textit{noun} \vee \textit{prep}\\ \text{VAL}\begin{bmatrix}\text{SPR}\ \langle\rangle\\ \text{COMPS}\ \langle\rangle\end{bmatrix}\end{bmatrix}\\ \text{LF}\ \begin{bmatrix}\text{EXC}\ \boxed{3}\ no(v,\alpha,\beta)\end{bmatrix}\end{bmatrix}\\
\wedge \boxed{2}\begin{bmatrix} \text{SS}|\ \text{LOC}|\ \text{CAT}\begin{bmatrix}\text{HEAD}\ \textit{det} \vee \textit{noun} \vee \textit{prep}\\ \text{VAL}\begin{bmatrix}\text{SPR}\ \langle\rangle\\ \text{COMPS}\ \langle\rangle\end{bmatrix}\end{bmatrix}\\ \text{LF}\ \begin{bmatrix}\text{EXC}\ \boxed{4}\ no(w,\phi,\psi)\end{bmatrix}\end{bmatrix} \wedge \boxed{4} \lhd \beta
\end{array}\right)\\
\rightarrow \exists\boxed{5}\ \exists\boxed{6}\ \exists\boxed{7}\ \exists\boxed{8}\left(\begin{bmatrix}\text{INF-STR}\begin{bmatrix}\text{FOC} & \boxed{6}\\ \text{BKGR} & \boxed{7}\end{bmatrix}\end{bmatrix} \wedge \boxed{8}\begin{bmatrix}\text{EXC}\ \boxed{3}\end{bmatrix} \in \boxed{6}\ \wedge \boxed{5}\begin{bmatrix}\text{EXC}\ \boxed{4}\end{bmatrix} \in \boxed{7}\right)
\end{array}\right)
$$

The principle says that if an utterance contains two negative quantifiers[15] $\boxed{3}$ and $\boxed{4}$ such that one of them outscopes the other ($\boxed{4} \lhd \beta$), then the former is the EXC value of some *lrs* $\boxed{8}$ which is a member of the utterance's FOC list $\boxed{6}$, while the latter is the EXC value of some *lrs* $\boxed{5}$ which is an element on the utterance's BKGR list $\boxed{7}$.

The empty SPR list of $\boxed{1}$ and $\boxed{2}$ guarantees that the two signs are maximal projections if their head is a noun. According to the first clause of the SEMANTICS PRINCIPLE, a noun which selects a quantifier as a specifier identifies its EXC value with the INC value of the quantifier. So the nouns *student* and *carte* in FIGURE 5.6 have negative quantifiers as their EXC values (see FIGURE 5.4). But the information structure conditions on the DN reading have to do with the determiners and not with the nouns, so we have to ensure that the objects on which we impose the INF-STR conditions are either determiners (e.g. *nicio*, *niciun*), or NP-quantifiers (e.g. *nimeni* 'nobody'). The restriction on the SPR list to be empty gives us the correct result, since neither determiners nor NPs select for specifiers.

The empty COMPS list of $\boxed{1}$ and $\boxed{2}$ ensures that neither of the two signs be an argument-marking preposition alone. Argument-marking prepositions identify their EXC value with that of their NP complement (see Sag et al. (2003, Ch. 7)). Thus a P could have a negative quantifier as EXC value, if it takes an NP complement with this EXC value. But like a simple N, the information structure contribution of the P alone has no effect on the interpretation of two negative quantifiers in an utterance. What we are interested in is the whole PP including the negative determiner. Given the condition [COMPS $\langle\rangle$], P alone will not meet the conditions in the antecedent of the DN PRINCIPLE in (278).

[13] See the sort hierarchy in Section 2.3.1.

[14] This principle is formulated to account for DN readings in simple sentences, so no functor (i.e. propositional attitude verb) should intervene between the two negative quantifiers. See Section 5.7 for a discussion of complex sentences.

[15] Note that the EXCONT PRINCIPLE will ensure that the two negative quantifiers contributed by $\boxed{1}$ and $\boxed{2}$ be subterms of the EXC value of the utterance, so we don't need to specify this once again in the principle in (278).

In conclusion, the two signs $\boxed{1}$ and $\boxed{2}$ can only be Dets, NPs and PPs containing a negative determiner. Note that the formulation of the DN PRINCIPLE is flexible enough to allow any projection above the two negative quantifiers (simple Det, NP, or PP) to contribute its *lrs* to the FOC/ BKGR value of the utterance. The only condition is for those *lrs*'s to contain the respective negative quantifiers as their EXC value ($\boxed{8}\left[\text{EXC } \boxed{3}\right] \in \boxed{6} \wedge \boxed{5}\left[\text{EXC } \boxed{4}\right] \in \boxed{7}$).

DN readings in LRS Now that we formulated the DN PRINCIPLE, let us go back to the two DN readings in (267) and see how they interact with the principle. For the reading in (267a) to be available, the quantifier *niciun student* 'no student' must take wide scope with respect to *nicio carte* 'no book'. In view of the DN PRINCIPLE this means that *niciun* must bear a non-empty FOC value and *nicio carte* must carry BKGR information. This coincides with the INF-STR described in the tree in FIGURE 5.6 representing the sentence in (268b). So we can conclude that the reading in (267a) is compatible with the DN PRINCIPLE. Note that the principle also allows the whole NP *niciun student* to be in focus and/ or the determiner *nicio* alone to carry background information. This depends on the linguistic context. In (279) the whole NP *niciun student* counts as focus and only the negative determiner *nicio* as background.

(279) a. A: Ion nu a citit **niciun roman** de pe listă.
John NM has read no novel from list
'John didn't read any novel on the list.'

b. B: **NIciun stuDENT** nu a citit **nicio carte** de pe listă.
no student NM has read no book from list
'No student read no book on the list.'

The DN reading in (267b) presents the second scope order of the two negative quantifiers: the direct object quantifier takes wide scope over the subject quantifier. According to the DN PRINCIPLE, this means that the determiner *nicio* or the NP *nicio carte* must bear focus and the determiner *niciun* or the NP *niciun student* must be background information. This accounts for (269), where the NP *niciun student* is background and the NP *nicio carte* is focus.

In conclusion, both DN readings can be obtained in the combinatorics of FIGURE 5.5 depending on the way the information structure conditions are distributed between the two negative quantifiers. At this point, our LRS analysis provides us with both a DN and a NC interpretation for sentence (264a).

As a final remark, the availability of DN readings in Romanian indicates that the NEGATION COMPLEXITY CONSTRAINT of Richter and Sailer (2004) should be reformulated for Romanian to allow two negations to occur as components of the EXCONT of a sign:

(280) THE NEGATION COMPLEXITY CONSTRAINT (for Romanian)
For each *sign*, there may be at most two negations that are components of the EXCONT value and have the MAIN value as a component of their nuclear scope.

The INF-STR conditions on NC Let us now consider the effects of the DN PRINCIPLE on the information structure of the NC reading. The formulation in (278) does not exclude the possibility of a NC interpretation meeting the INF-STR conditions of a DN reading as described in the consequent of (278). The entailment there applies if the utterance gets a DN interpretation. But if the antecedent is false (in our case, the utterance is interpreted as NC and not DN), the consequent may still be true (according to the truth conditions of logical implication). This means that our principle allows the possibility of a NC interpretation even if one n-word is in focus and the other one represents background information. We have to exclude this possibility from our grammar.

What we have to ensure for NC readings is that they do not occur in structures where the INF-STR conditions that favor DN are available. This is obtained by means of the constraint below:

(281) THE INF-STR CONSTRAINT ON NC

$$\forall\boxed{1}\ \forall\boxed{2}\ \forall\boxed{3}\ \forall\boxed{4}\ \forall\boxed{5}\ \forall\boxed{6}\ \forall\boxed{7}\ \forall\boxed{8}$$

$$\left(\begin{array}{l}\left(\begin{array}{l}\begin{bmatrix}\textit{u-sign} & \\ \text{INF-STR} & \begin{bmatrix}\text{FOC} & \boxed{6}\\ \text{BKGR} & \boxed{7}\end{bmatrix}\end{bmatrix} \wedge \boxed{1}\begin{bmatrix}\text{SS}|\ \text{LOC}|\ \text{CAT} & \begin{bmatrix}\text{HEAD}\ \textit{det} \vee \textit{noun} \vee \textit{prep}\\ \text{VAL}\begin{bmatrix}\text{SPR}\ \langle\rangle\\ \text{COMPS}\ \langle\rangle\end{bmatrix}\end{bmatrix}\\ \text{LF}\ \boxed{3}\begin{bmatrix}\text{EXC}\ \boxed{8}\ no(v,\alpha,\beta)\end{bmatrix} & \end{bmatrix}\\ \wedge\ \boxed{2}\begin{bmatrix}\text{SS}|\ \text{LOC}|\ \text{CAT} & \begin{bmatrix}\text{HEAD}\ \textit{det} \vee \textit{noun} \vee \textit{prep}\\ \text{VAL}\begin{bmatrix}\text{SPR}\ \langle\rangle\\ \text{COMPS}\ \langle\rangle\end{bmatrix}\end{bmatrix}\\ \text{LF}\ \boxed{4}\begin{bmatrix}\text{EXC}\ \boxed{5}\ no(w,\phi,\psi)\end{bmatrix} & \end{bmatrix} \wedge \boxed{1} \neq \boxed{2} \wedge \boxed{3} \in \boxed{6} \wedge \boxed{4} \in \boxed{7}\end{array}\right)\\ \rightarrow \boxed{5} \neq \boxed{8}\end{array}\right)$$

The antecedent of the constraint in (281) introduces the utterance and the two different signs contributing negative quantifiers, and describes the information structure conditions for a DN interpretation: one sign contributes its LF value on the FOC list of the utterance ($\boxed{3} \in \boxed{6}$), the other contributes its LF value on the BKGR list of the utterance ($\boxed{4} \in \boxed{7}$). The consequent of the constraint specifies that the two negative quantifiers must be different from each other ($\boxed{5} \neq \boxed{8}$). Since the NC reading can only be obtained if the two negative quantifiers get identified, the constraint ensures that the information structure conditions for DN do not occur with a NC interpretation.

Like in (278), the specifications SPR $\langle\rangle$ and COMPS $\langle\rangle$ for both $\boxed{1}$ and $\boxed{2}$ make sure that the two objects are either the maximal projections NP or PP or simple Dets, so they always include the negative determiner.

Note that $\boxed{1}$ and $\boxed{2}$ could be different projections containing the same negative quantifier: if the former is a negative determiner and the latter the NP containing it, $\boxed{1} \neq \boxed{2}$ as required by the antecedent in (281), but the negative quantifier under the two EXC values is the same, i.e. $\boxed{5} = \boxed{8}$ (contra the consequent of (281)). This case seems to indicate that our constraint is too strong. Note, however, that the antecedent in (281) also requires that $\boxed{1}$ contribute its *lrs* $\boxed{3}$ to the FOC list $\boxed{6}$ of the utterance and $\boxed{2}$ contribute its *lrs* $\boxed{4}$ to the BKGR list $\boxed{7}$. In our example, this means that the determiner $\boxed{1}$ contributes focus, while the NP projection $\boxed{2}$ above it contributes background information to the utterance. This is impossible, if we consider that focus and background information exclude each other: it cannot be the case that a node in a tree is in focus, while a higher projection containing it is in the background, or the other way around. This fact is not accounted for by the present simplified analysis, but a complete theory of information structure would ensure this. Given such a theory, the situation described above, where $\boxed{1}$ is a Det and $\boxed{2}$ is the NP projection above it, would not be subject to the INF-STR CONSTRAINT ON NC in (281), as it could not satisfy the INF-STR condition in the antecedent ($\boxed{3} \in \boxed{6} \wedge \boxed{4} \in \boxed{7}$).

Thus the two signs $\boxed{1}$ and $\boxed{2}$ have to be distinct NP/ PP arguments of a verb or simply Dets of such distinct arguments. If one of them contributes focus and the other background information, the constraint in (281) rules out the NC interpretation for the two negative quantifiers they carry.

5.4.3 Interaction with non-negative quantifiers

In Section 3.5, we discussed the scope properties of Romanian negative quantifiers when they interact with non-negative quantifiers. In this section I show how these properties can be accounted for in LRS. In particular, I will account for the contrast between the NC reading and the DN reading in sentence (282a). As I will show, the unavailability of the DN reading in (282b) and (282c) does not have to do with the semantics and the combinatorics of the negative quantifiers, but rather with the linear order and information structure conditions on quantifier scope in Romanian.

(282) **Niciun** student nu a citit **frecvent** **nicio** carte.
no student NM has read frequently no book

a. NO (student) $>$ FREQUENTLY $>$ NO (book) #NC / DN

b. NO (student) $-$ NO (book) $>$ FREQUENTLY NC / #DN

c. FREQUENTLY $>$ NO (student) $-$ NO (book) NC / #DN

Consider the lexical entry in (283) for the adverb *frecvent* 'frequently':[16]

(283) *frecvent* ('frequently')

$$\left[\begin{array}{l} \textit{word} \\ \text{PHON} \ \left\langle \texttt{frecvent} \right\rangle \\ \text{SS} \mid \text{LOC} \left[\begin{array}{l} \text{CAT} \mid \text{HEAD} \left[\begin{array}{l} \textit{adv} \\ \text{MOD} \ \text{V}\left[\text{LOC} \mid \text{CONT} \mid \text{MAIN} \ \boxed{3a}\right] \end{array}\right] \\ \text{CONT} \mid \text{MAIN} \ \boxed{7a}\ freq' \end{array}\right] \\ \text{LF} \left[\begin{array}{ll} \textit{lrs} & \\ \text{EXC} & \textit{me} \\ \text{INC} & \boxed{7}\ freq'(\boxed{8}) \\ \text{PARTS} & \left\langle \boxed{7}, \boxed{7a} \right\rangle \end{array}\right] \end{array}\right] \wedge \boxed{3a} \triangleleft \boxed{8}$$

As an adverb, *frecvent* must modify a verb with a specified semantic MAIN value which in our case is the MAIN value $read'$ of the verb *nu a citit* tagged $\boxed{3a}$ in FIGURE 5.5. Its local semantic contribution is the constant $freq'$ and the INCONT value in the non-local semantics is $freq'(\boxed{8})$. The lexical entry of the adverb also enforces that the MAIN value of the modified verb be a subterm of the argument of $freq'$ (i.e. $\boxed{3a} \triangleleft \boxed{8}$).

Given the other lexical specifications in (265), we can now derive the LRS-structure of the sentence (282) as in FIGURE 5.7. For interpretation we consider only the case when the subject negative quantifier takes scope over the object negative quantifier. For the adverb *frecvent* we consider all three scope possibilities: widest scope, intervention between the negative quantifiers and lowest scope. This gives us the three scope combinations in (284) with their respective interpretations in LRS.

(284) a. $\boxed{0} = \boxed{1} = no(v, \alpha, \beta) \wedge \boxed{8} = \boxed{6} = no(w, \phi, \psi)$

i. DN reading for (282a):
$no(x, student'(x), freq'(no(y, book'(y), read'(x, y))))$

[16] The semantics I give for *frequently* is informal and is based on the example for the quantificational adverb *always* in Richter and Kallmeyer (2007).

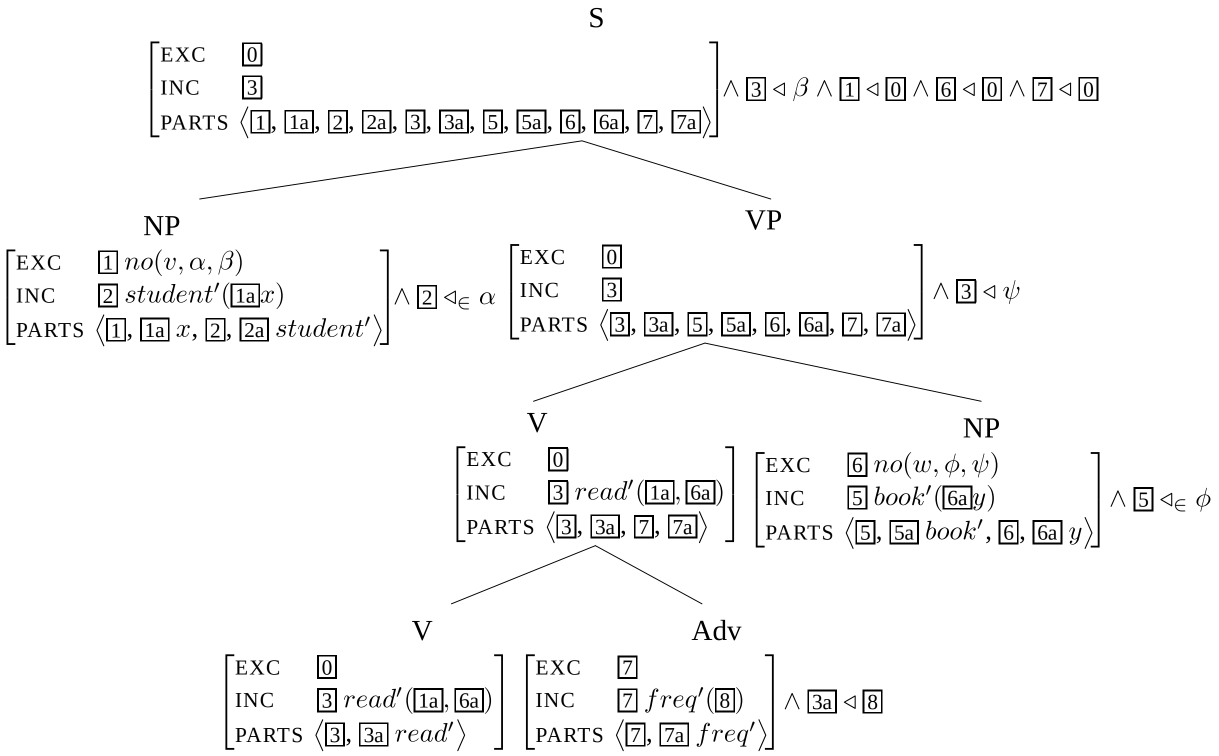


Figure 5.7: LRS analysis of *Niciun student nu a citit frecvent nicio carte* (without the NM)

ii. NC reading for (282a):
#

b. $\boxed{0} = \boxed{1} = no(v, \alpha, \beta) \wedge \boxed{8} = \boxed{3} = read(x, y)$

i. DN reading for (282b):
$\boxed{6} = \beta$: $no(x, student'(x), no(y, book'(y), freq'(read'(x, y))))$

ii. NC reading for (282b):
$\boxed{6} = \boxed{1} = \boxed{0}$: $no((x, y), (student'(x), book(y)), freq'(read(x, y)))$

c. $\boxed{0} = \boxed{7} = freq'(\boxed{8})$

i. DN reading for (282c):
$\boxed{8} = \boxed{1} \wedge \boxed{6} \triangleleft \beta$: $freq'(no(x, student'(x), no(y, book(y), read(x, y))))$

ii. NC reading for (282c):
$\boxed{8} = \boxed{1} = \boxed{6}$: $freq'(no((x, y), (student'(x), book(y)), read(x, y)))$

The scope interaction in (284a) represents the reading in (282a): the operator $freq'$ intervenes between the two negative quantifiers. In the LRS tree this means that the subject negative quantifier takes widest scope (i.e. $\boxed{0} = \boxed{1}$), while the object negative quantifier takes narrow scope with respect to the adverb (i.e. $\boxed{8} = \boxed{6}$). The LRS constraints in FIGURE 5.7 make only the DN reading available for this scope interaction. Given the two identities $\boxed{0} = \boxed{1}$ and $\boxed{8} = \boxed{6}$, the NC reading cannot be obtained, since the two negative quantifiers cannot be identified: $\boxed{1} \neq \boxed{6}$. If we enforce this identity, we obtain $\boxed{0} = \boxed{1} = \boxed{8} = \boxed{6}$ which entails that the EXCONT value of the sentence is in the scope of $freq'$ (i.e. $\boxed{7}$

= $freq'(\boxed{0})$). This violates the constraint $\boxed{7} \triangleleft \boxed{0}$ imposed by the second clause of the LRS EXCONT PRINCIPLE on the utterance in FIGURE 5.7. In conclusion, the present LRS analysis correctly predicts that the NC reading is unavailable for two negative quantifiers if a non-negative operator intervenes between them (see (282a)).

The scope interaction in (284b) is the representation of (282b): the operator $freq'$ takes lowest scope. In LRS this interaction is obtained if $freq'$ only takes scope over the INC value of the verb (i.e. $\boxed{8}$ = $\boxed{3}$, so $\boxed{7}$ = $freq'(read(x, y))$). Two interpretations are associated with this order of the operators, depending on the interaction between the two negative quantifiers: DN and NC. The DN reading obtains if the object negative quantifier appears as the nuclear scope of the subject negative quantifier ($\boxed{6}$ = β). The NC reading occurs if we identify the two negative quantifiers and thus obtain: $\boxed{6}$ = $\boxed{1}$ = $\boxed{0}$. Both readings are predicted to occur in the LRS analysis.

Similar predictions are made with respect to the scope interaction in (282c), given for LRS in (284c). Here the operator $freq'$ takes widest scope, so $\boxed{0}$ = $\boxed{7}$ = $freq'(\boxed{8})$. The value of $\boxed{8}$ is determined by the scope interaction between the two negative quantifiers: the subject can outscope the object and thus give a DN reading (see 284ci), or the two can get identified giving rise to the NC reading (284cii).

As can be observed in (284b) and (284c), the LRS analysis predicts that if the non-negative operator does not intervene between the two negative quantifiers, then they should have the same scope interpretations as in sentence (264a), where there is no other operator besides the negative quantifiers. However, the data in (282b) and (282c) seem to indicate that the present analysis overgenerates, since the DN reading is not available in either of the two orders of the three operators. Restricting the interpretation of such structures is possible in LRS. Nevertheless, I will show below that this should not be done in the LRS analysis, since the unavailability of the DN reading in these cases does not have to do with the semantics of the quantifiers and their scope interaction. Rather it can be explained by the interplay between the linear order of the quantifiers and the information structure of the sentence in determining the relative scope of the operators.

Let us concentrate on the DN interpretation in (284bi). If we constrain its unavailability in the LRS analysis, this interpretation will be ruled out for every sentence in which two negative quantifiers and a non-negative operator exhibit this scope interaction. This result contravenes the fact that such a reading is possible in the contexts in (285) and (286):

(285) a. Speaker A: Ion nu-şi vizitează **NIciun** coleg **frecvent**.
John NM-CL visits no colleague frequently
NO > FREQUENTLY 'No colleague is such that John visits him frequently.'

b. Speaker B: **NIciun** stuDENT nu-şi vizitează NI**ciun** coleg **frecvent**.
no student NM-CL visits no colleague frequently
NO > NO > FREQUENTLY: 'For no student is it the case that he has no colleague whom he visits frequently. = Every student has a colleague whom he visits frequently.'

(286) a. A: Mircea Cărtărescu nu a recomandat **NIciunui** student **multe** cărţi.
M.C. NM has recommended no student many books
NO > MANY: 'No student is such that M.C. recommended him many books.'

b. B: **NIciun** scriiTOR nu a recomandat NI**ciunui** student **multe** cărţi.
no writer NM has recommended no student many books
NO > NO > MANY: 'For no writer is it the case that there is no student to whom he recommended many books. = For every writer there is a student to whom he recommended many books.'

Given the primarily NC nature of Romanian and the complexity of the scope interaction between three operators, the sentences in (285b) and (286b) are difficult to interpret. But the DN reading is available and this contrasts with (282b). The only difference between (285b)/ (286b) and (282b) is the linear order of the operators in the sentence: the non-negative operator does not intervene between the negative ones in the former, but it does in the latter.

In view of the discussion in Section 3.5 we can explain these effects by appealing to the influence of the surface linear order on the scope of quantifiers in Romanian. In (285b)/ (286b), the operators are interpreted in their surface linear order, so what makes the interpretation difficult is the interaction between the special information structure requirements of the DN reading and the difficulty to process the scope interaction between three independent operators. In (282b) we have an additional complication: the surface linear order is different from the order of the scope interaction. In particular, the object negative quantifier follows the adverb *frecvent*. As discussed in Section 3.5, for the negative quantifier to take scope over the other operator in this case, it has to be emphasized which means that it will bear focus. But a DN reading requires the low scope negative quantifier to carry background information which correlates with its being deaccented. These two conditions obviously exclude each other, which explains why DN is unavailable in (282b).

A similar explanation can be offered for the unavailability of the DN reading in (282c). For the subject negative quantifier preceding *frecvent* to be able to take narrow scope, it must be deaccented. But this contravenes the information structure condition for DN readings which requires that the wide scope negative quantifier carry focus information and thus be accented (see the discussion in the previous section).

An exhaustive theory of quantifier scope in Romanian would have to take into account this interplay between the surface linear order, the scope interaction between the operators, and the information structure. Together with the LRS analysis here, this would account for the interpretations available for sentence (282). The present LRS account allows us to derive the NC reading and to rule out the DN reading in (282a) and predicts that the semantics of the operators involved is compatible with both a NC and a DN reading in (282b) and (282c).

5.5 The analysis of the NM

In the first part of this chapter the analysis of NC in Romanian was limited to the interpretation of n-words. In what follows, I concentrate on the syntax-semantics of the NM *nu* and its role in NC.

There are three issues that need to be clarified with respect to the NM: 1) its negative semantics outside NC (287a); 2) its lack of negation in the presence of n-words (287b) and 3) its obligatoriness in NC constructions (287c). These three properties are illustrated by the data below:

(287) a. *Sentential negation*

Un student **nu** a venit.
a student NM has come

'Some student didn't come.'

b. *Semantic absorption with n-words*

Niciun student **nu** a venit.
no student NM has come

i. 'No student came.' (NC)
ii. # 'No student didn't come.' (#DN)

c. *Obligatoriness with n-words*

Niciun student *(**nu**) a citit cartea.
no student (NM) has read book-the

'No student read the book.'

In Section 5.5.1 I summarize the semantic behavior of *nu* as discussed in Chapter 3 and I draw the conclusion that it contributes negation in the absence of n-words, but not when n-words are present. The syntactic ambiguity of *nu* between nu_{NM} and nu_{Adv} – pointed out by Barbu (2004) and presented here in Section 5.5.2 – leads to the question whether we can posit lexical ambiguity of *nu* between nu_{NM}, specific to NC and always non-negative and the negative nu_{Adv}. This would mean that (287a) contains nu_{Adv}, different from the non-negative nu_{NM} in (287b). But in Section 5.5.3 data concerning the licensing of NPIs and PPIs indicate that nu_{NM} is also possible in contexts like (287a), so it can contribute sentential negation, and thus the generalization that the NM contributes negation outside NC but not with n-words must be maintained and accounted for.

In Section 5.5.4, I propose a generalized quantifier analysis of nu_{NM} as NO^0 and I provide a way to integrate it in the resumption analysis of NC. The LRS analysis in Section 5.5.5 accounts for the lack of DN readings with nu_{NM} and n-words. In Section 5.5.6, I account for the obligatoriness of the NM in NC by means of a NC constraint which enforces the presence of the NM on the verbal head of an utterance whose semantics contains a negative quantifier.

5.5.1 The semantic behavior of *nu*

In this section I review the semantic behavior of *nu* both outside NC and within NC contexts. As we will see, *nu* is clearly negative outside NC. With n-words it usually does not visibly contribute negation, since, as indicated before, it does not trigger DN with an n-word. So *nu*'s negation always concords with that of n-words in NC constructions.

A discussion on marginal sentences where forcing the andi-additivity of *nu* over n-words results in a DN reading leads to the conclusion that in this case, it is not the NM *nu*, but a homonymous modifier *nu* that triggers DN with n-words. We thus have to distinguish between two syntactic instances of *nu*: nu_{NM} and nu_{Adv}. This will be addressed in the next section.

Outside NC *Nu* is the common marker for negation in Romanian. Its presence turns an affirmative sentence into a negative one, without any further help:

(288) a. Studenţii au citit romanul.
students-the have read novel-the

'The students have read the novel.'

b. Studenţii **nu** au citit romanul.
students-the NM have read novel-the

'The students did not read the novel.'

The negative properties of *nu* are verified by the antimorphicity test presented under (104c) in Chapter 3, p. 75. As can be seen in (289) and (290), the interpretation of *nu* taking scope over a disjunction is equivalent to the conjunction of the negated disjuncts (indicating anti-additivity and thus, meeting the first condition of antimorphicity), and *nu* taking scope over a conjunction is equivalent to the disjunction of the negated conjuncts, thus fulfilling the second condition in (104c):[17]

(289) *Anti-additivity*

a. Studenţii **nu** au citit romane *sau* poezii.
students-the NM have read novels or poems
'The students haven't read novels or poems.'

b. = Studenţii **nu** au citit romane *şi* studenţii **nu** au citit poezii.
students-the NM have read novels and students-the NM have read poems
= 'The students haven't read novels and the students haven't read poems.'

(290) *Antimorphicity*

a. Studenţii **nu** au citit romane *şi* poezii.
students-the NM have read novels and poems
'The students haven't read novels and poems.'

b. = Studenţii **nu** au citit romane *sau* studenţii **nu** au citit poezii.
students-the NM have read novels or students-the NM have read poems
= 'The students haven't read novels or the students haven't read poems.'

Within NC Considering the two facts above, the negative semantics of *nu* is obvious outside NC. But when n-words are involved, the situation becomes less clear. We saw in Section 3.4.2 that in denial contexts two n-words get a DN interpretation. In the same kind of context, *nu* was unable to yield DN in combination with an n-word (see (171) vs. (172), p. 95). (291) and (292) below illustrate the same contrast between the n-words (repeating the example in (268)) and *nu*:

(291) a. Speaker A: Un student nu a citit **nicio carte**.
one/a student NM has read no book
'One/A student read no book.'

b. Speaker B: **NICIun stuDENT** nu a citit **nicio carte**.
no student NM has read no book
'No student read no book. (= Every student read some book.)' (DN/ # NC)

(292) a. Speaker A: Un student **nu** a citit cartea.
one/a student NM has read book-the
'One/A student didn't read the book.'

b. Speaker B: **NICIun stuDENT nu** a citit cartea.
no student NM has read book-the
i. # 'No student didn't read the book. (Every student read the book.)' (# DN)
ii. 'No student read the book.' (NC)

The interpretation of (292b) contrasts with that of (291b) because DN is not available, although the sentence contains two negative elements: *nu* and *niciun student*, and a previous negative context favorable to DN.

[17] Recall that there is one more reading available for (289a) and (290a), where the disjunction/ conjunction occurs between two sentences, of which the latter is elliptical. This reading was the only possible one in case the disjunction involves two n-words: see (132), p. 83.

At the same time, as indicated in Chapter 3, p. 83, the anti-additive properties of *nu* – illustrated for instance in (289) above – are not available once n-words are involved. If in (289) the bare nouns are replaced by n-words, the semantic identity between the negated disjunction in (289a) and the conjunction of negated disjuncts in (289b) does not hold anymore. The only natural interpretation of the disjunction of the n-words in (293a) is the disjunction of two negative propositions in (293c).

(293) a. Studenţii **nu** au citit **niciun** roman *sau* **nicio** poezie.
students-the NM have read no novel or no poem
'The students read no novel or no poem.'

b. *Anti-additivity*

$\neq$ Studenţii **nu** au citit **niciun** roman *şi* studenţii **nu** au citit **nicio** poezie.
students-the NM have read no novel and students-the NM have read no poem

$\neq$ 'The students read no novel and the students read no poem.'

c. *Ellipsis*

= Studenţii **nu** au citit **niciun** roman *sau* studenţii **nu** au citit **nicio** poezie.
students-the NM have read no novel or students-the NM have read no poem

= 'The students read no novel or the students read no poem.'

In Chapter 3, I referred to the reading in (293c) as elliptical, because the disjunction seems to coordinate two negative propositions, and thus outscopes the NM. At the same time, I noted that an anti-additive interpretation of the sentence in (293a) is marginally possible, but in this case *nu* bears stress and denies the disjunction between the two n-words, so the interpretation ends up being affirmative (i.e. DN):

(294) a. ?? Studenţii **NU** au citit **niciun** roman *sau* **nicio** poezie.
students-the not have read no novel or no poem
'The students read no novel or no poem.'

b. *Anti-additivity*

= **Nu** e adevărat că studenţii **nu** au citit **niciun** roman *şi* **nu** e adevărat că studenţii **nu** au citit **nicio** poezie. (= Studenţii au citit romane *şi* studenţii au citit poezii.)
NM is true that students-the NM have read no novel and NM is true that students-the NM have read no poem students-the have read novels and students-the have read poems

'It is not true that the students read no novel and it is not true that the students read no poem. (= The students read novels and the students read poems.)'

(294a) is only marginally grammatical, for reasons that have to do with the licensing of the two n-words. As we will see in Section 5.5.2, *nu* in (294a) is syntactically different from the NM *nu* (cf. Barbu (2004)): it is an adverb. This explains the reduced grammaticality of (294a) in comparison to (293a). In the latter we have the NM licensing the n-words, so the sentence is grammatical. In the former, the n-words are not licensed by a NM, because *nu* is an adverb. Since the anti-additivity test in (293a) shows the NM *nu* not to be anti-additive, it means that its negative semantics is not attested in NC contexts.

The data in (292) and (293), compared (291) and (289), respectively, indicate that the NM does not contribute negation in the presence of n-words: it does not yield DN in combination with a negative quantifier, and it does not have anti-additive properties in NC. But the NM does not always lack negative content, given the properties illustrated in (288) and (289)-(290). Alternatively, we could claim that *nu* is lexically ambiguous: negative when it appears alone, and non-negative in NC. As already mentioned and as we will see in Section 5.5.2, there is evidence for a syntactic ambiguity of *nu* between the NM status and adverbial status. Barbu (2004) argues that the former is a verbal affix, while the latter is an adverb and functions as a modifier. If one could show that the contexts in (288) and (289)-(290) contain the adverb *nu* and not the NM *nu*, then lexical ambiguity would be a path to follow. But as we will see below, the NM *nu* is not excluded in (288) and (289)-(290), so it does contribute negative semantics in these contexts. Let us first address the syntactic properties and the ambiguity of *nu*.

5.5.2 The syntactic status of the NM

The discussion on the semantic contribution of *nu* led to the hypothesis that it may be lexically ambiguous. This would explain its contradictory semantic behavior, expressing negation in some contexts but not in others. In this section I address the syntactic ambiguity of *nu*, as noted in Barbu (2004), and I investigate the possibility of relating this to its semantic ambiguity documented above. In the end, we will see that the two ambiguities do not fully overlap, so we cannot conclude that the syntactic item *nu* which acts as a NM is always non-negative, as the NC constructions seem to suggest.

The second aim of this section is to determine the syntactic status of the NM *nu*, an important piece for the HPSG analysis of the NM in NC constructions which will be developed in Section 5.5.6. The conclusion will be that the NM is a verbal affix, so in HPSG terms, it attaches to the verb by a lexical rule.

The ambiguity of *nu* In reply to Monachesi (2000)'s analysis of the Romanian negation *nu* as a full lexical item playing the role of a VP modifier, Barbu (2004) argues for the ambiguity of *nu* between a modifier and a verbal affix. Her distinction correlates with the difference between an adverb *nu* and the negative marker *nu*. The former *nu* will be marked in the following examples as "nu_{Adv}" and the latter as "nu_{NM}".

The important difference between the two items is the ability to license n-words which characterizes only nu_{NM} (see (296) vs. (295)). This correlates with nu_{NM}'s occurrence exclusively within the "verbal complex"[18], while modifier nu_{Adv} easily modifies constituents of any category (e.g. NPs, PPs, CPs in (295)):

(295) a. Ştiu asta **nu**$_{Adv}$ [$_{PP}$ de la Ion/ *de la **niciun** student], ci din ziar.
know this not from John/ from no student, but from newspaper
'I know this not from John, but from newspapers.'

b. **Nu**$_{Adv}$ [$_{NP}$ Ion/ ***nimeni**] este vinovatul aici.
not John/ nobody is guilty-the here
'John is not the guilty one here.'

[18]For Romance languages including Romanian, the term has been mostly employed in Barbu (1999), Monachesi (2000), Abeillé and Godard (2000), Abeillé and Godard (2003), among others. Here it will be used to cover all the elements (lexical and grammatical affixes) which function as a single lexical unit together with the verb.

c. A spus **nu**$_{Adv}$ [$_{CP}$ că va veni (***niciodată**)], ci că va incerca să vină.
has said not that will come (never), but that will try SJ come
'She did not say that she would come, but that she would try to come.'

(296) **Nu**$_{NM}$ ştie **nimeni** asta.
NM knows nobody this
'Nobody knows this.'

The syntactic distinction between the two homonymous *nu*'s is further supported by *nu*$_{Adv}$ replacing negative adverbs like *nicidecum* ('not at all') or *în niciun caz* ('by no means') in (297a), which is not possible for *nu*$_{NM}$ (297b):[19]

(297) a. Ştiu asta **nu**$_{Adv}$/ **nicidecum**/ **în niciun caz** [de la Ion/ # de la **nimeni**], ci din ziar.
know this not/ not at all/ by no means from John/ from nobody, but from newspaper

b. **Nu**$_{NM}$/ ***nicidecum**/ ***în niciun caz** ştie **nimeni** asta.
NM/ not at all/ by no means knows nobody this

The ungrammaticality of (297b) casts doubt on the idea that *nu*$_{NM}$ may be a VP modifier like *nu*$_{Adv}$. In addition to this, *nu*$_{NM}$ is in complementary distribution with the prefix *ne-* in non-finite verb forms: while *nu*$_{NM}$ is used with the infinitive, *ne-* appears with the present/ past participle and the supine:

(298) a. *infinitive*
a **nu**$_{NM}$ şti **nimic**
to NM know nothing

b. *present participle*
neştiind **nimic**
un-knowing nothing

c. *past participle*
neştiut de **nimeni**
un-known by nobody

d. *supine*
de **ne**conceput de către **nimeni**
of un-conceived-Sup by nobody
'inconceivable by anybody'

The data in (297) and (298) indicate that a syntactic distinction needs to be made between *nu*$_{Adv}$, a full lexical item syntactically acting as a modifier, and *nu*$_{NM}$, an affixal item, part of the verbal complex.

Following Barbu (2004), further arguments can be brought to support this generalization: as an affix, *nu*$_{NM}$ in (299a) and (300a) follows the infinitive marker *a* and the supine marker *să*, while *nu*$_{Adv}$, as a modifier, precedes them in (299b) and (300b):

[19]The n-word *nimeni* in (297a) may be grammatical, but with an interpretation different from the NC reading triggered by *nu*$_{NM}$ in (297b). In (297a) the effect would be DN between *nu*$_{Adv}$ and *de la nimeni*. So the n-word cannot be licensed by *nu*$_{Adv}$ the way it is typically licensed by the NM in (297b).

(299) a. [*A* **nu**$_{NM}$ spune **nimic**] este strategia optimă.
to NM tell nothing is strategy best
'The best strategy is not to tell anything.'

b. **Nu**$_{Adv}$ [*a* spune minciuni/ ***nimic**] este strategia optimă.
not to tell lies/ nothing is strategy best
'Telling lies is not the best strategy.'

(300) a. Ţi-am cerut [*să* **nu**$_{NM}$ spui **nimic**].
you-have asked SJ NM tell nothing
'I asked you not to tell anything.'

b. Ţi-am cerut **nu**$_{Adv}$ [*să* spui minciuni/ ***nimic**], ci să spui adevărul.
you-have asked not SJ tell lies/ *nothing, but SJ tell truth
'I asked you not to tell lies, but to tell the truth.'

While *nu*$_{Adv}$ modifies the whole VP and thus precedes it, *nu*$_{NM}$ intermingles with other components of the verbal form (like the infinitive and the subjunctive marker), thus emerging as a morpho-syntactic part of the verbal complex.

A further difference between the two *nu*'s concerns the ability to act as a pro-form in ellipsis. This characterizes *nu*$_{Adv}$, but not *nu*$_{NM}$. In (301), the second disjunct can be completely replaced by *nu*$_{Adv}$ or the negative adverb *nicidecum*. In the same context *nu*$_{NM}$ is ungrammatical, because as an affix, it is dependent on its verbal host:

(301) Ion să plece, însă Maria [să **nu**$_{NM}$ plece].
John SJ leave, but Maria SJ NM leave
'John should leave, but Maria shouldn't leave.'

a. Ion să plece, însă Maria [**nu**$_{Adv}$/ **nicidecum**].
John SJ leave but Maria not/ not at all
'John should leave, but Maria should not.'

b. * Ion să plece, însă Maria [să **nu**$_{NM}$].
John SJ leave, but Maria SJ NM

Finally, this distinction correlates with morpho-phonological differences. In most of the examples above (e.g. (295), (297a), (298b), (299b)), *nu*$_{Adv}$ is employed with a contrastive role: to emphasize that something is not the case, and that something else holds instead. As a consequence, *nu*$_{Adv}$ always bears stress, and it never reduces phonologically to *n-*. By contrast, *nu*$_{NM}$ does reduce to *n-* when it is followed for instance by an auxiliary beginning with the vowel *a*:

(302) a. Ion **NU**$_{Adv}$/ ***N-** [a alungat], ci a omorît ţînţarii.
John not has chased-away, but has killed mosquitoes-the
'John didn't chase the mosquitoes away, but killed them.'

b. Ion **nu**$_{NM}$/ **n-** a alungat **niciun** ţînţar.
John NM has chased-away no mosquito
'John didn't chase any mosquito away.'

This discussion on the ambiguity of *nu* allows a better description of the properties of the NM, the unambiguous *nu* involved in NC constructions: it only appears within the verbal complex, and there are well-defined constraints on its exact position[20] (see for instance (299a) and (300a)). It cannot be substituted by lexical adverbs (297b), on the contrary, it is in complementary distribution with the prefix *ne-* (298). Finally, it cannot appear without its verbal host (301b), and it reduces to *n-* if the following phonological context favors it (302b). As a consequence of these properties, I conclude with Barbu (2004) that the NM is an affix. In the HPSG account in Section 5.5.5, the NM will be attached to the verb by means of a lexical rule.

DIGRESSION **Terminological clarifications** Given the diverse literature discussing various kinds of negation (see Klima (1964), Horn (1989), McCawley (1991), Kim and Sag (2002), to mention only a few for English), some terminological clarifications are in order. It is necessary to determine to what extent the distinction made here with respect to Romanian *nu* may relate to some of the notions already proposed in the literature.

The discussion on (302) concerning *nu*$_{Adv}$'s usual contrastive role is reminiscent of McCawley's (1991) notion of *contrastive negation*, standing for *not X but Y* constructions. But although at first sight modifier *nu* could be thought of as *contrastive negation*, this does not completely hold. One reason is that the notion of contrastive negation in McCawley's view refers to the whole construction, not only to *not [X]*, and thus presupposes the existence of two constituents that fill one syntactic position. Even if this is most often the case with modifier *nu* in Romanian, the second part (*but Y*) is not obligatory (see (295b) and (299b)). Another argument against the correlation between *nu*$_{Adv}$ and contrastive negation is that the latter also covers English cases like (303a) (McCawley (1991), p. 190) which in Romanian may be expressed by means of *nu*$_{NM}$, as (303b) indicates:

(303) a. John didn't drink coffee but tea.

b. Să **nu**$_{NM}$ bei cafea, ci lapte.
SJ NM drink coffee, but milk
'You shouldn't drink coffee, but milk.'

In conclusion, contrastive negation in English does not fully correspond to modifier *nu*$_{Adv}$ in Romanian, and syntactically it involves a more complex structure.

The data in (295) and (297) concerning the syntax of *nu*$_{Adv}$ may lead to associating it with what Kim and Sag (2002) describe as *constituent negation not* in English, and *ne-pas* in French. Although such a generalization is not groundless, given the diversity of the constituents that *nu*$_{Adv}$ can modify, it should be noted that Kim and Sag's distinction is basically established between *finite* and *constituent* negation and this does not correlate with the distinction between *nu*$_{NM}$ and *nu*$_{Adv}$ in Romanian. First of all, sentential negation in English/ French, unlike the Romanian NM *nu*$_{NM}$, is a full lexical item[21], optionally selected by the verb as a complement.[22] But most importantly, *constituent negation* in Kim and Sag (2002) is used to cover modification of non-finite constructions, like in (304a), which in Romanian can easily employ the *nu*$_{NM}$ as well as the modifier *nu*$_{Adv}$, but with different interpretations (see (304b) and (304c)):

(304) a. **Not** [speaking English] is a disadvantage.

b. A **nu**$_{NM}$ vorbi **nicio** limbă străină este un dezavantaj.
to not speak no language foreign is a disadvantage
'Not speaking any foreign language is a disadvantage.'

[20]More on the position of the NM will be given in Section 5.5.6, when formulating the NM Lexical Rule.
[21]See Kim and Sag (2002)'s arguments (p. 24) against positing a morphological unit between the verb and *not*/ *pas*.
[22]For more details, see also Abeillé and Godard (1997) and Sag (to appear).

c. **NU**$_{Adv}$ [a vorbi o/ ***nicio** limbă străină] este avantajul cel mai mare, ci a
not to speak a/ *no foreign language is advantage the biggest, but to
vorbi limba ţării în care eşti.
speak language country in which are

'Not speaking a foreign language, but speaking the language of the country where you are is the biggest advantage.'

In conclusion, *constituent negation* of Kim and Sag (2002) for English and French cuts across both cases of the Romanian *nu*: modifier *nu*$_{Adv}$ and NM *nu*$_{NM}$. Still, the syntactic analysis of modifier *nu* should be similar to that of constituent negation *not*.

5.5.3 Is *nu*$_{NM}$ non-negative?

In Section 5.5.1 we concluded that the NM *nu* does not contribute negation in the NC context (e.g. (287b) repeated below as (305)). But *nu* is undoubtedly negative in non-NC contexts like that of sentential negation (e.g. (287a) repeated below as (306)). In view of the syntactic ambiguity of *nu* described in the previous section, the question now is whether the syntactic ambiguity of *nu* correlates with its semantic ambiguity illustrated in (305) and (306).

(305) *Semantic absorption with n-words*

Niciun student **nu**$_{NM}$ a venit.
no student NM has come

i. 'No student came.' (NC)
ii. # 'No student didn't come.' (DN)

(306) *Sentential negation*

Ion **nu**$_{NM/Adv}$ a citit cartea.
John NM/ not has read book-the

'John didn't read the book.'

Given that the licensing of n-words is an important criterion for the syntactic distinction between *nu*$_{NM}$ and *nu*$_{Adv}$, in (305) we can only have *nu*$_{NM}$. But *nu* in (306) could be both *nu*$_{NM}$ and *nu*$_{Adv}$. The latter would bear stress and would modify the VP constituent, the former could be phonologically reduced to *n-*.

If we can determine that actually only *nu*$_{Adv}$ is possible in (306) and that *nu*$_{NM}$ is excluded, we can argue for a lexical ambiguity of *nu* that has both syntactic and semantic effects. This would mean that only *nu*$_{Adv}$ is semantically negative and plays the role of sentential negation and that *nu*$_{NM}$ is restricted to NC contexts and is non-negative.

The semantics of *nu*$_{Adv}$ The negative semantics of *nu*$_{Adv}$ cannot be doubted, since it is the only negative element in the contexts where it appears: unlike *nu*$_{NM}$, it does not license n-words. In some special contexts where *nu*$_{Adv}$ modifies an n-word, it creates a contrast, so it triggers a DN interpretation, as in (307):

(307) Ion a făcut **NU**$_{Adv}$ [**nimic**], ci chiar foarte multe pentru această petrecere.
John has done not nothing but quite very many for this party

'John did not do nothing for this party, he did quite a lot.'

Coming back to the anti-additivity of *nu* with respect to n-words in (294) repeated below as (308), we may now conclude that it is *nu*$_{Adv}$ that triggers anti-additivity and DN with respect to the disjunction of the two n-words:

(308) a. ?? Studenţii **NU**$_{Adv}$ [au citit **niciun** roman *sau* **nicio** poezie].
students-the not have read no novel or no poem
'The students read no novel or no poem.'

b. = **Nu** e adevărat că studenţii **nu** au citit **niciun** roman *şi* **nu** e
NM is true that students-the NM have read no novel and NM is
adevărat că studenţii **nu** au citit **nicio** poezie. (= Studenţii au citit
true that students-the NM have read no poem students-the have read
romane *şi* studenţii au citit poezii.)
novels and students-the have read poems
'It is not true that the students read no novel and it is not true that the students read no poem. (= The students read novels and the students read poems.)'

In (308a), *nu*$_{Adv}$ modifies the whole VP *au citit niciun roman sau nicio poezie* including the disjunction of the two n-words. This way we can also explain the marginality of the sentence which is due to the absence of a NM licensing the two n-words.

In conclusion, for the sentence in (293a) discussed above, if *nu* is the syntactic item *nu*$_{NM}$, then anti-additivity with respect to n-words is never available and the only interpretation is the one with ellipsis (293c). This fact apparently brings support for our assumption that *nu*$_{NM}$ may never contribute negation, so it would be excluded in (306). However, as we will see below, evidence from NPI licensing indicates that *nu*$_{NM}$ is not excluded in those contexts where we have sentential negation, so *nu*$_{NM}$ is allowed in (306) and it can contribute negation. Consequently, it is only the presence of n-words that prevents *nu*$_{NM}$ from contributing negation in NC.

The negative semantics of *nu*$_{NM}$ In order to check if *nu*$_{NM}$ can appear in contexts without n-words and contribute negation at the same time, we have to find something that disambiguates between *nu*$_{Adv}$ and *nu*$_{NM}$ in the absence of n-words. In the previous section we saw that when *nu* appears after the subjunctive/ infinitive particle it is *nu*$_{NM}$ (see (299a) and (300a)). In such contexts, *nu*$_{NM}$ can also occur without an n-word, so it expresses negation, as the data below indicate:

(309) a. [A **nu**$_{NM}$ spune adevărul] este strategia optimă.
to NM tell truth-the is strategy-the optimal
'The optimal strategy is not to tell the truth.'

b. Ţi-am cerut [să **nu**$_{NM}$ spui adevărul].
you-have asked SJ NM tell truth-the
'I asked you not to tell the truth.'

Another test that confirms the negative semantics of *nu*$_{NM}$ in the absence of n-words is NPI licensing. An NPI of medium strength like *prea* 'NPI really' (cf. van der Wouden (1997), Zwarts (1998)) can only be licensed by sentential negation with or without n-words.[23] In all the other contexts it is used as an intensifier corresponding to the English 'too'. As an NPI, it appears within the verbal complex, qualifying as a 'semi-adverb' in Ciompec's (1985) terms. Otherwise, it usually precedes adjectives or adverbs:

[23] See also the description of the licensing conditions for *prea* in CoDII (2008).

(310) a. Această campanie **nu** *prea*$_{npi}$ şi-a atins scopul.
this campaign not/NM really itself-has touched aim-the
'This campaign has not really reached its target.'

b. **Nicio** campanie **nu** *prea*$_{npi}$ îşi mai atinge scopul.
no campaign NM really itself anymore touch aim-the
'Campaigns don't really reach their targets anymore.'

c. Această campanie a fost *prea* agresivă.
this campaign has been too aggressive
'This campaign has been too aggressive.'

The NPI *prea* also has a PPI equivalent, *cam* 'somewhat, pretty' (see also Avram (1986, pp. 205–206)). *Cam* is grammatical in positive contexts, but it is excluded with n-words, sentential negation and even in downward entailing contexts like the scope of *few*:

(311) a. Această campanie (***nu**) şi-a *cam* atins scopul.
this campaign NM/not itself-has pretty touched aim-the
'This campaign has pretty much reached its target.'

b. * **Nicio** campanie **nu** şi-a *cam* atins scopul.
no campaign NM itself-has pretty touched aim-the

c. * **Puţini** studenţi au *cam* tras chiulul.
few students have pretty skipped classes

Nu$_{Adv}$ apparently cannot license a strong NPI like *prea* occurring in the constituent that it modifies, and it is grammatical with the PPI *cam*, so it does not interact with the licensing conditions for NPIs and PPIs (see (312a)). By contrast, *nu*$_{NM}$ licenses the NPI *prea* and is ungrammatical with the PPI *cam* in a similar context (312b), so it interacts with polarity items:

(312) a. A început **NU**$_{Adv}$ [să *cam*/ **prea* tragă chiulul], ci să lipsească săptămîni în şir de la şcoală.
has started not SJ pretty/ really skip classes, but SJ miss weeks in row from school
'He has not started to pretty much skip classes, but to miss classes weeks in a row.'

b. A început să **nu**$_{NM}$ **cam*/ *prea* (mai) tragă chiulul.
has started SJ NM pretty/ really anymore skip classes
'He started not to really skip classes (anymore).'

Another NPI - PPI pair, also mentioned by Avram (1986), is that of *decît* 'but' vs. *numai* 'only'. Like *prea* vs. *cam*, *nu*$_{Adv}$ is compatible with the PPI *numai*, but cannot license the NPI *decît*, while *nu*$_{NM}$ licenses the latter, but is incompatible with the former:

(313) a. A început **NU**$_{Adv}$ [să mănînce *numai*/ **decît* produse lactate], ci să evite grăsimile.
has started not SJ eat only/ but products milk-based, but SJ avoid fats
'He has not started to eat only dairy products, but to avoid fat.'

b. A început să **nu**$_{NM}$ mănînce **numai/ decît* produse lactate.
has started SJ NM eat only/ but products milk-based
'He started not to eat anything else but dairy products.'

In view of the data in (312) and (313), *nu* in (310a) and (311a) can only be *nu*$_{NM}$. This means that, despite the syntactic ambiguity of *nu*, sentential negation contexts most often involve *nu*$_{NM}$, so there is no doubt that the latter has negative semantics.

In conclusion, we cannot argue for a lexical ambiguity of *nu* between non-negative *nu*$_{NM}$ and negative *nu*$_{Adv}$. *Nu*$_{NM}$ is negative in contexts where no n-words are present.

5.5.4 The semantic analysis of the NM

We now have to explain why the negation of the NM is lost when n-words are present. We need account for the lack of DN readings between the NM and n-words (305) and the lack of anti-additivity over n-words (293).

First of all, note that DN readings are not always excluded between the NM and an n-word. If the NM appears in a negative question and the answer is an n-word, the interpretation of the answer is DN and not NC: see (314). Thus it is only in NC constructions that the NM does not yield DN readings. This means that in NC the negation of the NM must concord with that of the n-word(s).

(314) a. Cine să **nu** vină?
who SJ NM come
'Who should not come?'

b. **Nimeni**. (Să vină toţi.)
nobody SJ come all
i. 'Nobody should not come. (= Everybody should come.)' (DN)
ii. # 'Nobody should come.' (NC)

This observation is also supported by the lack of anti-additivity effects over n-words in (293) repeated below:

(315) a. Studenţii **nu** au citit **niciun** roman *sau* **nicio** poezie.
students-the NM have read no novel or no poem
'The students read no novel or no poem.'

b. *Anti-additivity*
$\neq$ Studenţii **nu** au citit **niciun** roman *şi* studenţii **nu** au citit **nicio** poezie.
students-the NM have read no novel and students-the NM have read no poem
$\neq$ 'The students read no novel and the students read no poem.'

c. *Ellipsis*
$=$ Studenţii **nu** au citit **niciun** roman *sau* studenţii **nu** au citit **nicio** poezie.
students-the NM have read no novel or students-the NM have read no poem
$=$ 'The students read no novel or the students read no poem.'

The NM does exhibit anti-additivity in the absence of n-words. The fact that the subjunctive marker precedes *nu* (316) disambiguates it to nu_{NM}:

(316) a. Să **nu** citeşti revistele acestea *sau* ziarele acelea.
SJ NM read magazines-the these or newspapers-the those
'You should not read these magazines or those newspapers.'

b. *Anti-additivity*
= Să **nu** citeşti revistele acestea *şi* să **nu** citeşti ziarele acelea.
SJ NM read magazines-the these and SJ NM read newspapers-the those
= 'You should not read the magazines and you should not read the newspapers.'

c. *Ellipsis*
=? Să **nu** citeşti revistele acestea *sau* [să **nu** citeşti] ziarele acelea.
SJ NM read magazines-the these or SJ NM read newspapers-the those
= 'You should not read these magazines or you should not read those newspapers.'

As we see in (316c), the elliptical reading for the disjunction is always available, so it does not only occur with n-words like in (315c). With the n-words in (315), the anti-additive reading is blocked by the presence of the n-word.

In conclusion, the NM carries negation, but within NC its semantics obligatorily concords with that of the n-word.

The NM within polyadic quantification To express the negative semantics of the NM, I follow de Swart and Sag (2002) and represent it as a generalized negative quantifier NO, similar to the one carried by n-words. The difference is that the quantifier expressed by the NM is a propositional operator, so it does not bind any variable, has no restriction, and takes a proposition (i.e. a truth value) to a truth value. In Lindström's classification summarized in Section 2.1, TABLE 2.3, p. 17 it takes a 0-ary relation to a truth value, so it is a type $\langle 0 \rangle$ quantifier. Following CONVENTION 4.1, we abbreviate it as NO^0. The semantics of NO^0 can be derived from the general semantics of NO^k given in DEFINITION 4.2, repeated below:

Definition 4.2 (p. 112) The semantics of NO^k
For a domain E*, for every* $\text{A}_1, \text{A}_2, \ldots, \text{A}_k \subseteq \text{E}$, $\text{R} \subseteq \text{E}^k$*:*

$$[\![\text{NO}^k]\!](\text{A}_1 \times \text{A}_2 \times \ldots \times \text{A}_k, \text{R}) = 1 \text{ iff } (\text{A}_1 \times \text{A}_2 \times \ldots \times \text{A}_k) \cap \text{R} = \emptyset$$

The polyadic quantifier NO^k denotes the empty intersection between the k-ary Cartesian product on the domain E and another k-ary relation R. This means that the quantifier NO^0 denotes the empty intersection between the 0-ary Cartesian product on the domain E and another 0-ary relation R. Since the 0-ary Cartesian product on the domain E is E^0, the singleton set $\{()\}$, we can derive the semantics of NO^0 as below:[24]

Definition 5.9 The semantics of NO^0
$[\![NO^0]\!](\text{E}^0, \text{R}) = 1 \textit{ iff } \text{E}^0 \cap \text{R} = \emptyset$
$\Leftrightarrow [\![NO^0]\!](\text{E}^0, \text{R}) = 1 \textit{ iff } \{()\} \cap \text{R} = \emptyset$
$\Leftrightarrow [\![NO^0]\!](\text{E}^0, \text{R}) = 1 \textit{ iff } \text{R} = \{\} = \emptyset$

[24] See also the discussion with respect to TABLE 2.2 in Section 2.1, p. 15.

Given the fact that {} = 0/ false and {()}= 1/ true, the semantics of NO^0 is the same as that of the logical negation: it is true only of false propositions.

Consider the semantics of the NM together with that of the two n-words in sentence (199a), p. 111, repeated in (317). There are three NO quantifiers that undergo resumption: $\mathrm{NO}_{\mathrm{E}}^{\mathrm{STUDENT}}$ and $\mathrm{NO}_{\mathrm{E}}^{\mathrm{BOOK}}$ of type $\langle 1,1\rangle$, corresponding to the two n-words, and $\mathrm{NO}_{\mathrm{E}^0}$ of type $\langle 0\rangle$, the NM. The resumption mechanism available in Section 2.1 cannot account for resumption of quantifiers of different types. DEFINITION 2.15 refers to *k*-ary resumption of type $\langle 1,1\rangle$ quantifiers. To allow type $\langle 0\rangle$ quantifiers to undergo resumption with $\langle 1,1\rangle$ quantifiers we define resumption for both types of quantifiers:[25]

(317) **Niciun** student **nu** a citit **nicio** carte.
no student NM has read no book
'No student read any book.'

Definition 5.10 Resumption of type $\langle 1,1\rangle$ and type $\langle 0\rangle$ quantifiers
For a domain E, $\mathrm{A}_1, \mathrm{A}_2, \ldots, \mathrm{A}_k \subseteq \mathrm{E}$, $\mathrm{R} \subseteq \mathrm{E}^k$, *resumption of a k-sequence of type* $\langle 1,1\rangle$ *quantifiers* Q *and an l-sequence of type* $\langle 0\rangle$ *quantifiers* Q *is given by:*
$Res^k(Q)_{\mathrm{E}}^{\mathrm{A}_1,\mathrm{A}_2,\ldots,\mathrm{A}_k}(\mathrm{R}) = \mathrm{Q}_{\mathrm{E}^k}^{\mathrm{A}_1\times\mathrm{A}_2\times\ldots\times\mathrm{A}_k}(\mathrm{R})$

The formula by which resumption of k type $\langle 1,1\rangle$ quantifiers and l type $\langle 0\rangle$ quantifiers is derived is identical to that of k-ary resumption in DEFINITION 2.15, p. 32. So type $\langle 0\rangle$ quantifiers add nothing to the complexity of a resumption of type $\langle 1,1\rangle$ quantifiers. This is expected if we recall that the restriction and the nuclear scope of $\langle 0\rangle$ quantifiers are subsets of E^0 (see DEFINITION 5.9). The l-Cartesian product of E^0 is E^0 and the Cartesian product of E^k and E^0 is E^k (LEMMA 5.1), so the presence of $\langle 0\rangle$ quantifiers in a resumption does not change the type of the polyadic quantifier.

Lemma 5.1

i. l*-Cartesian product of* E^0: $\underbrace{\mathrm{E}^0 \times \mathrm{E}^0 \times \ldots \times \mathrm{E}^0}_{l\text{-times}} = \mathrm{E}^{(0+0+\ldots+0)} = \mathrm{E}^0$

ii. Cartesian product of E^k *and* E^0: $\underbrace{\mathrm{E} \times \mathrm{E} \times \ldots \times \mathrm{E}}_{k\text{-times}} \times \mathrm{E}^0 = \mathrm{E}^{(k+0)} = \mathrm{E}^k$

5.5.5 The LRS-analysis

We now express the semantics of the NM in LRS terms, so that we can integrate it in the LRS analysis of Romanian NC developed in Section 5.4. We first have to enrich the lexical specification of negated verbs with the negative quantifier NO^0, contributed by the NM.[26] Let us account for the sentential negation context given in (287a) and repeated below as (318). The lexical specification for the verb *nu a venit* ('NM has come') containing the negative quantifier is given in (319).[27]

(318) *Sentential negation*

[25]See de Swart and Sag (2002, p. 399) for a more general definition of resumption as applying to a sequence of k type $\langle n,n\rangle$ quantifiers and l type $\langle m,m\rangle$ quantifiers.

[26]The lexical rule by which a verb becomes negative will be discussed in Section 5.5.6.

[27]As will become clear in Section 5.5.6, the lexical specification in (319) is not a lexical entry, but a description of the output of the lexical rule in (327) that attaches the NM to a verb.

Un student **nu** a venit.
a student NM has come

'Some student didn't come.'

(319) *nu a venit* ('NM has come', with the semantic contribution of the NM)

$$\left[\begin{array}{ll} \multicolumn{2}{l}{\textit{word}} \\ \text{PHON} & \left\langle \texttt{nu a venit} \right\rangle \\ \text{SS} \mid \text{LOC} & \left[\begin{array}{ll} \text{CAT}|\text{VAL} \mid \text{SUBJ} & \left\langle \text{NP}_{\boxed{1a}} \right\rangle \\ \text{CONT} & \left[\begin{array}{ll} \text{INDEX} \mid \text{VAR} & \textit{no-var} \\ \text{MAIN} & \boxed{3a}\, come' \end{array}\right] \end{array}\right] \\ \text{LF} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{lrs}} \\ \text{EXC} & \boxed{0} \\ \text{INC} & \boxed{3}\, come'(\boxed{1a}) \\ \text{PARTS} & \left\langle \boxed{3}, \boxed{3a}, \boxed{7}\, no(u, \gamma, \delta) \right\rangle \end{array}\right] \end{array}\right] \wedge \boxed{7} \triangleleft \boxed{0} \wedge \boxed{3} \triangleleft \delta$$

The negative quantifier NO^0 appears on the PARTS list of the verb. In order to ensure that the NM negates the verb, we constrain the INC value of the verb to be a subterm of the nuclear scope of NO^0, i.e. $\boxed{3} \triangleleft \delta$. The negative quantifier contributed by the NM has an empty list of variables, so I assume that the local semantics of verbs does not contribute a variable (contra Sailer (2004)).[28] Within the present HPSG grammar, I extend the value of the attribute SS | LOC | CONT | INDEX | VAR from *variable* to a more general sort *var(iable)-value* with two subsorts: *variable* for NPs and *no-var(iable)* for verbs.[29] The relevant piece of the type hierarchy is given in (320):

(320)

var(iable)-value

no-var(iable) *variable*

If we consider that the NP *un student* ('a student') is similar to the NP *niciun student* ('no student') described in FIGURE 5.4 with the difference that it carries an existential quantifier and not a negative one, we represent the sentence (318) as in FIGURE 5.8. There are two possible interpretations for this sentence depending on the way we disambiguate the value of $\boxed{0}$ in FIGURE 5.8:

(321) a. $some(x, student'(x), no((), (), come'(x)))$
$\boxed{0} = \boxed{1} \wedge \boxed{3} = \delta \wedge \beta = \boxed{7}$

b. $no((), (), some(x, student'(x), come'(x)))$
$\boxed{0} = \boxed{7} \wedge \boxed{3} = \beta \wedge \delta = \boxed{1}$

[28]This choice is meant to fit the grammar we have defined so far, in which quantifiers only bind individual variables of type e. Introducing an event variable for verbs would modify the semantic type of the quantifiers. While this may turn out to be useful especially for an exhaustive account of NC including adverbial n-words functioning as modifiers, I restrict the discussion here to nominal n-words functioning as arguments and to generalized quantifiers binding variables of type e.

[29]A similar specification is suggested in Sailer (2004, p. 206) for the defective PHI value of verbs which is of sort *no-phi*.

$$\text{S}: \begin{bmatrix} \text{EXCONT} & \boxed{0} \\ \text{INCONT} & \boxed{3} \\ \text{PARTS} & \left\langle \boxed{1}, \boxed{1a}, \boxed{2}, \boxed{2a}, \boxed{3}, \boxed{3a}, \boxed{7} \right\rangle \end{bmatrix} \wedge \boxed{3} \triangleleft \beta \wedge \boxed{1} \triangleleft \boxed{0} \wedge \boxed{7} \triangleleft \boxed{0}$$

$$\text{NP}: \begin{bmatrix} \text{EXCONT} & \boxed{1}\, some(v, \alpha, \beta) \\ \text{INCONT} & \boxed{2}\, student'(\boxed{1a}x) \\ \text{PARTS} & \left\langle \boxed{1}, \boxed{1a}\, x, \boxed{2}, \boxed{2a}\, student' \right\rangle \end{bmatrix} \wedge \boxed{2} \triangleleft_{\in} \alpha \qquad \text{V}: \begin{bmatrix} \text{EXCONT} & \boxed{0} \\ \text{INCONT} & \boxed{3}\, come'(\boxed{1a}) \\ \text{PARTS} & \left\langle \boxed{3}, \boxed{3a}\, come', \boxed{7}\, no(u, \gamma, \delta) \right\rangle \end{bmatrix}$$

Figure 5.8: LRS analysis of (318) *Un student nu a venit*

In the first reading the existential quantifier has wide scope over negation, in the second one the order is reversed. This means that in the former case it is the no quantifier that appears in the nuclear scope of $some$, in the latter, $some$ is in the nuclear scope of no. In Romanian, only the first reading is available for a sentence like (318), i.e. existential quantifiers co-occurring with negation must outscope it. This has to do with the interaction between existential quantifiers and negation, which we discussed in Section 3.5.1. Thus the reading in (321b) should be excluded by language-specific principles which determine the possible scope interpretations of two quantificational operators. But in principle the reading is available in FIGURE 5.8.

Let us now consider the analysis of a NC sentence as the one in (287b), repeated below:

(322) *Semantic absorption within NC*

Niciun student **nu** a venit.
no student NM has come

i. 'No student came.' (NC)
ii. # 'No student didn't come.' (# DN)

$$\text{S}: \begin{bmatrix} \text{EXCONT} & \boxed{0} \\ \text{INCONT} & \boxed{3} \\ \text{PARTS} & \left\langle \boxed{1}, \boxed{1a}, \boxed{2}, \boxed{2a}, \boxed{3}, \boxed{3a}, \boxed{7} \right\rangle \end{bmatrix} \wedge \boxed{3} \triangleleft \beta \wedge \boxed{1} \triangleleft \boxed{0} \wedge \boxed{7} \triangleleft \boxed{0}$$

$$\text{NP}: \begin{bmatrix} \text{EXCONT} & \boxed{1}\, no(v, \alpha, \beta) \\ \text{INCONT} & \boxed{2}\, student'(\boxed{1a}x) \\ \text{PARTS} & \left\langle \boxed{1}, \boxed{1a}\, x, \boxed{2}, \boxed{2a}\, student' \right\rangle \end{bmatrix} \wedge \boxed{2} \triangleleft_{\in} \alpha \qquad \text{V}: \begin{bmatrix} \text{EXCONT} & \boxed{0} \\ \text{INCONT} & \boxed{3}\, come'(\boxed{1a}) \\ \text{PARTS} & \left\langle \boxed{3}, \boxed{3a}\, come', \boxed{7}\, no(u, \gamma, \delta) \right\rangle \end{bmatrix}$$

Figure 5.9: LRS analysis of (322) *Niciun student nu a venit*

The LRS representation of the sentence is given in FIGURE 5.9. It differs from the previous representation in that the NP contributes a negative quantifier. Since now we have two generalized quantifiers of the same sort (i.e. no), we can also identify them (like in the case of the two n-words in FIGURE 5.5) and thus obtain the NC reading as the third possible interpretation (see (323c)).

(323) a. $no(x, student'(x), no((), (), come'(x)))$

$\boxed{0} = \boxed{1} \wedge \boxed{3} = \delta \wedge \beta = \boxed{7}$

b. $no((), (), no(x, student'(x), come'(x)))$

$\boxed{0} = \boxed{7} \wedge \boxed{3} = \beta \wedge \delta = \boxed{1}$

c. $no(x, student'(x), come'(x))$

$\boxed{0} = \boxed{1} = \boxed{7}$

Next we have to ensure that the NM does not contribute its negation independently of other negative quantifiers in a sentence, so it cannot trigger DN. By accounting for this, we eliminate the possibility for the sentence in (322) to get either of the two interpretations in (323a) and (323b) which are available in the LRS analysis at this point. As previously discussed, DN in Romanian appears with two n-words but not between the NM and an n-word. The only difference between the NM and an n-word in our analysis is the number of variables that are bound by the negative quantifier: while the NM contributes no variable, an n-word contributes one. This difference can be used to exclude DN readings with the NM but not with n-words as in the principle below, which is a reformulation of the NEG CRITERION of Richter and Sailer (2004) for Romanian:

(324) THE NEG CRITERION for Romanian

For every finite verb, if there is a NO^0 in the external content of the verb that has scope over the verb's MAIN value, then any other negative quantifier in the verb's external content that also has scope over the verb's MAIN value must be on the verb's PARTS list.

$$\forall\boxed{0}\,\forall\boxed{1}\,\forall\boxed{2}\,\forall\boxed{3}\left(\begin{array}{l} \begin{bmatrix} \textit{word} \\ \text{SS}\,|\,\text{LOC} \begin{bmatrix} \text{CAT}\,|\,\text{HEAD} \begin{bmatrix}\textit{verb}\\ \text{VFORM}\ \textit{fin}\end{bmatrix} \\ \text{CONT}\,|\,\text{MAIN}\ \boxed{3}\end{bmatrix} \\ \text{LF}\,|\,\text{EXC}\ \boxed{0}\end{bmatrix} \wedge \boxed{1}\, no((),(),\delta) \triangleleft \boxed{0} \wedge \boxed{2}\, no(v,\alpha,\beta)\triangleleft\boxed{0}\wedge\boxed{3}\triangleleft\delta\wedge\boxed{3}\triangleleft\beta \\ \rightarrow \\ \exists\boxed{4}\left(\left[\text{LF}\,|\,\text{PARTS}\ \boxed{4}\right]\wedge\boxed{2}\in\boxed{4}\right) \end{array}\right)$$

The principle in (324) says that once a NO^0 quantifier takes scope over the MAIN value of a verb, any negative quantifier taking scope over the MAIN value of the verb must be on the verb's PARTS list. Since the PARTS list of the verb is lexically specified, only lexically contributed quantifiers can appear on it. The effect of the principle is that the external content of a verb which contains a NO^0 cannot contain any other negative quantifier and this excludes DN interpretations with a NO^0. The only situation where an external content value can contain a NO^0 is the one in which the negative marker contributes the only negation in the sentence (e.g. (318)).

The NEG CRITERION for Romanian is weaker than the one for Polish in Richter and Sailer (2004), because the presence of a NO^0 is presupposed in the antecedent. For this reason, it does not account for the obligatoriness of the NM with n-words. That will be ensured in the syntactic analysis of the NM in Section 5.5.6. In conclusion, by means of the NEG CRITERION for Romanian we can restrict the interpretation of the sentence in (322) to the NC reading in (323c).

An important issue concerning the value of $\boxed{0}$ in FIGURE 5.8 and 5.9 has to do with the way we determine the list of variables for each quantifier. While the constraint in the lexical entry of the determiner *un/ niciun* constrains its variable x to be part of the list of variables v in the generalized quantifier $some/no$, nothing enforces the list of variables for the type $\langle 0 \rangle$ quantifier no to be empty. This is something that should be ensured by the general LRS theory which should provide an appropriate mechanism for handling the variables bound by quantifiers. For a discussion of this matter the reader is referred to the digression in Section 5.6.

5.5.6 The syntactic analysis of the NM in NC

Our analysis of NC treats the NM as part of NC with no supplementary semantic contribution to that conveyed by n-words. The assumption that the NM carries a negative quantifier accounts for its anti-additive and antimorphic properties and for the fact that it triggers sentential negation.

But there are two important facts that the semantic theory developed here does not cover. It does not prevent a second occurrence of the NM in sentence (326a)[30,31] and it does not account for the obligatoriness of the NM with n-words (326b):

(326) a. *Impossibility to repeat the NM*:

Studenţii (***nu**$_{NM}$) **nu**$_{NM}$ au citit **niciun** roman/ romanul.
students NM NM have read no novel/ novel-the

'The students didn't read any novel/ the novel.'

b. *Obligatoriness with n-words*:

Niciun student *(**nu**) a venit.
no student NM has come

'No student came.'

I will account for these two facts within an HPSG syntax-semantics interface for NC constructions. First, a verb will be able to undergo the lexical rule for NM-attachment only once. This is possible if lexical verbs come with a head feature [NEG –] which is turned into [NEG +] when they undergo the lexical rule. Second, I will formulate a NC constraint which requires that the head of an utterance whose EXCONT value contains a negative quantifier be marked as [NEG +]. This means that it must have a NM.

The NM Lexical Rule Given the affixal status of the NM established in Section 5.5.2, the mechanism by which a verb becomes negative should be part of the lexicon.[32] In HPSG this can be done by means of a lexical rule, in a way similar to the treatment of other verbal affixes like clitics (cf. Miller

[30] Note that DEFINITION 5.10, p. 205, in particular, makes the prediction that further occurrences of the NM should not change the interpretation of a sentence.

[31] Two occurrences of *nu* may be allowed if the first one is *nu*$_{Adv}$ and only the second one is *nu*$_{NM}$ like in (325). The interpretation is double negation:

(325) Studenţii **NU**$_{Adv}$ [**nu**$_{NM}$ au citit **niciun** roman], ci chiar le-au citit pe toate.
students not NM have read no novel but even them-have read all

'It is not the case that the students didn't read any novel, to the contrary they read them all.'

As already suggested by the fact that its negation does not concord with that of n-words (see Section 5.5.3), we have to assume that *nu*$_{Adv}$'s negative semantics cannot be expressed by a negative quantifier, but most likely by the logical operator "¬". This way its presence will not interfere with the conditions on n-words licensing. This is the desired result, given that *nu*$_{Adv}$ has no role in NC.

[32] A similar conclusion was reached in Kupść and Przepiórkowski (2002) for the Polish NM.

and Sag (1997), Abeillé et al. (1998) for French, Monachesi (1996), Monachesi (1999) for Italian, Monachesi (1998) for Romanian).

There are three facts that have to be taken into account by the lexical rule which attaches the NM to a verb: 1) the impossibility of the NM to attach twice to the same verb (see (326a)); 2) the presence of a negative quantifier NO^0 contributed by the NM (see sentential negation); 3) the linear order of the NM with respect to other components of the verbal complex.

We can account for the first property by introducing a head feature NEG(ation) on verbs. Its value will be "+" for verbs which are marked with negation and "–" for verbs which are not marked. The lexical entries of verbs are [NEG –], which means that verbs are affirmative in form. It should be noted that this feature refers only to syntactic negation, so semantically negative verbs like *nega* "deny" will be also [NEG –] in their lexical entry. The NM LEXICAL RULE that attaches a NM to the verb will turn the verb into [NEG +]. The attribute NEG is only used for verbs, so it characterizes *verb* objects as the value of HEAD.

Taking all these facts into account, we formulate the NM LEXICAL RULE below:

(327) THE NM LEXICAL RULE (NMLR)

$$\left[\begin{array}{ll} \multicolumn{2}{l}{\textit{word}} \\ \text{PHON} & \boxed{X} \\ \text{SS}|\text{LOC}|\text{CAT} & \left[\text{HEAD} \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{verb}} \\ \text{VFORM} & \textit{fin} \vee \textit{inf} \\ \text{NEG} & - \end{array}\right]\right] \\ \text{LF} & \left[\begin{array}{ll} \text{EXCONT} & \boxed{0} \\ \text{INCONT} & \boxed{1} \\ \text{PARTS} & \boxed{2} \end{array}\right] \end{array}\right] \longmapsto \left[\begin{array}{ll} \text{PHON} & Neg(\boxed{X}) \\ \text{SS}\,|\,\text{LOC}\,|\text{CAT}\,|\text{HEAD} & \left[\text{NEG} \;\; +\right] \\ \text{LF}\,|\text{PARTS} & \boxed{2} \oplus \left\langle \boxed{3}\, no(u,\gamma,\delta) \right\rangle \end{array}\right] \wedge \boxed{1} \triangleleft \delta \wedge \boxed{3} \triangleleft \boxed{0}$$

In Section 5.5.2 we saw that the NM *nu* attaches both to finite verbs and to the infinitive, so the specification [VFORM *fin* $\vee$ *inf*] is meant to select only these verb forms as the input of the NMLR.

The negative quantifier contributed by the NM must appear on the PARTS list of the verb having undergone the NM LEXICAL RULE, so that eventually, via the LRS PROJECTION PRINCIPLE, it can become a part of the semantics of the whole utterance. Moreover, this quantifier must be constrained to be a subterm of the EXCONT value of the verb ($\boxed{3} \triangleleft \boxed{0}$). This is needed for complex sentences where the negative quantifier lexically contributed by the NM of the embedded verb should be prevented from taking scope in the matrix clause (see Section 5.7 for discussion). The lexical rule also enforces the INCONT value of the verb to be a subterm of the expression representing the scope of the negative quantifier contributed by the NM ($\boxed{1} \triangleleft \delta$). This way the verb is always interpreted in the scope of the negation contributed by the NM.

The Neg function specified in (328) describes the phonology of the verb after the NM is attached including the environment following *nu* which favors its reduction to *n-*. This includes the clitic *o* "her"[33] and auxiliary verbs starting with the vowels *a* or *o*. All these items are collected in $\text{H}_{a/o}$, the set of phonological hosts which start with a or o. The Neg function attaches nu to a stem which does

[33]Since the NM precedes the clitics in the verbal complex, the lexical rule by which clitics attach to a verb should be enforced to apply only to non-negated, i.e. [NEG –], verbs. This is compatible for instance with the "Complement Cliticization Lexical Rule" in Monachesi (1998, p. 109).

not start with any of the elements in $\mathrm{H}_{a/o}$ and attaches nu or n to a verb form which starts with one of these elements:

(328) *The Neg Function*

$$Neg(\boxed{X}) = \left\langle \texttt{nu} \right\rangle \oplus \boxed{X} \text{ in case } \boxed{X} \notin \mathrm{H}_{a/o} \text{ and}$$

$$Neg(\boxed{X}) = \left(\left\langle \texttt{n} \right\rangle \oplus \boxed{X}\right) \vee \left(\left\langle \texttt{nu} \right\rangle \oplus \boxed{X}\right) \text{ in case } \boxed{X} \in \mathrm{H}_{a/o}$$

$$\text{where } \mathrm{H}_{a/o} = \left\{\left\langle \texttt{am} \right\rangle, \left\langle \texttt{aș} \right\rangle, \left\langle \texttt{ai} \right\rangle, \left\langle \texttt{ar} \right\rangle, \left\langle \texttt{a} \right\rangle, \left\langle \texttt{ați} \right\rangle, \left\langle \texttt{o} \right\rangle, \left\langle \texttt{oi} \right\rangle, \left\langle \texttt{om} \right\rangle, \left\langle \texttt{oți} \right\rangle, \left\langle \texttt{or} \right\rangle\right\}$$

The NC Constraint In Chapter 3 we concluded that the NM does not play a semantic role in the licensing of n-words. Its obligatory co-occurrence with n-words was explained as a condition of syntactic licensing: the presence of an n-word requires the presence of the NM on the verb (see (326b)).

Given the feature NEG, which indicates whether a verb has a NM, we can now enforce the presence of the NM on finite verbs taking n-words as arguments. An n-word contributes a negative quantifier so the EXC value of the sentence in which the n-word appears will contain that negative quantifier as well (see the EXCONT PRINCIPLE). As the verbal head of a sentence has an EXC value token-identical with that of the sentence itself, the negative quantifier contributed by an n-word is also a subexpression of the verb's EXC value. Thus we can formulate the NC licensing condition directly on the verb at the *word*-level: if a verb's EXC value contains a negative quantifier taking scope over its MAIN value, that verb must be [NEG +]. The effect is that a finite verb is not allowed to have a negative quantifier in its semantics, unless it also carries a feature [NEG +] on its head. All verbs are lexically specified [NEG –], so an n-word can only co-occur with a verb that has undergone the NMLR. We can formulate this restriction as the NC CONSTRAINT below. The possibility for an embedded n-word to be syntactically licensed by the NM on the matrix verb will be addressed in Section 5.7.

(329) THE NC CONSTRAINT (NCC)

$$\forall\boxed{0}\ \forall\boxed{1}\ \forall\boxed{2}$$

$$\left(\begin{bmatrix} \textit{word} \\ \text{SS}\,|\,\text{LOC} \begin{bmatrix} \text{CAT}\,|\,\text{HEAD} \begin{bmatrix} \textit{verb} \\ \text{VFORM} \ \textit{fin} \end{bmatrix} \\ \text{CONT}\,|\,\text{MAIN} \ \boxed{1} \end{bmatrix} \\ \text{LF}\,|\,\text{EXCONT} \ \boxed{0} \end{bmatrix} \wedge \boxed{2}\, no(v,\alpha,\beta) \triangleleft \boxed{0} \wedge \boxed{1} \triangleleft \beta \right.$$
$$\longrightarrow$$
$$\left. \begin{bmatrix} \text{SS}\,|\,\text{LOC}\,|\,\text{CAT}\,|\,\text{HEAD} \begin{bmatrix} \text{NEG} \ + \end{bmatrix} \end{bmatrix} \right)$$

Note that sentences with the NM contributing sentential negation obey the NCC, since their semantics contains a negative quantifier NO^0 and their heads are marked as [NEG +]. But utterances that contain at least one n-word and no NM are [NEG –], so they are ruled out by the NCC.

Two remarks are in order here. First, note that the NCC in (329) rules out sentence (307) repeated below as (330), as the n-word *nimic* occurs in the absence of the NM on the verb. This is due to the presence of the modifier nu_{Adv} which creates an island for the licensing of the n-word. Other syntactic islands for NC are relative and adverbial clauses (see Section 3.3.3). But in those cases, I showed that the n-word cannot be licensed by the main clause NM: the presence of a finite verb in the relative/ adverbial clause imposes licensing of the n-word by a clausemate NM on that finite verb. Unlike with relative and adverbial clauses, the n-word in (307) does not need to establish a NC relationship with the finite verb, as it is separated from the latter by the modifier *nu*$_{Adv}$: the syntactic island in which the n-word appears does not contain a finite verb and consequently, the n-word is free to occur unlicensed. This example is comparable with other contexts lacking a finite verb where n-words can appear without NM licensing as for instance (163) to (165), p. 93.

(330)	Ion	a	făcut	**NU**$_{Adv}$	[**nimic**],	ci	chiar	foarte	multe	pentru	această	petrecere.
	John	has	done	not	nothing	but	quite	very	many	for	this	party

‘John did not do nothing for this party, he did quite a lot.’

In this thesis, I have not investigated the n-word licensing conditions in details, so a full analysis of the syntactic islands that prevent the typical licensing conditions for n-words is not aimed at here. For (330) one would need to introduce a disjunction in the consequent of the constraint in (329) that also allows an n-word to be modified by *nu*$_{Adv}$ instead of being licensed by a [NEG +] finite verb. Further syntactic islands would need to be accommodated in the NC CONSTRAINT as well.

A second remark concerns the way the NCC is formulated. In (329) I relate the licensing of n-words to the syntactic specification [NEG +], as we independently need this attribute to account for the impossibility of a verb to acquire two NMs. But the NCC could also be formulated entirely in the semantics without making use of [NEG +]. To do this, the consequent in (329) would have to guarantee that the PARTS list of the finite verb contains a negative quantifier. This negative quantifier could only come from the verb so it would be an element of the PARTS list only with the condition of the verb having undergone the NM LEXICAL RULE in (327). In this case the verb would also be [NEG +], so the effects of the NCC are the same, independently of whether we formulate it with [NEG +] or with a negative quantifier having to be available on the PARTS list of the verb.

5.6 Digression. A discussion on variables

There is a technical issue that needs to be clarified in the present account. It concerns the way variables can be handled in LRS and how this influences the analysis. It was in part raised in Section 5.4.1 with respect to the order of the variables in a resumptive quantifier and also came up in relation to the empty list of variables that are bound by the negative quantifier contributed by the NM in FIGURE 5.8, p. 207.

Let us have a look at what the present LRS theory does with respect to variables. The logical foundations of LRS tacitly assume quantifiers to be monadic, that is, they bind only one variable. The literature (see for instance Richter and Sailer (2004) and Richter and Kallmeyer (2007)) only addresses this kind of quantifiers. By introducing polyadic and type $\langle 0 \rangle$ quantifiers, several technical problems may occur, among which I note the following: 1) spurious ambiguities, 2) variables contributed by different quantifiers getting identified or the same variable being bound by two operators 3) impossibility to determine the empty list of variables for $\langle 0 \rangle$ quantifiers. Some of these problems are already taken care of, some others need to be solved in the logical foundations of LRS.

Spurious ambiguities can easily occur due to the VAR and RESTR values of sort *list*. Note that our grammar does not fix the order of the variables and of the corresponding restrictions in a resumptive quantifier. In (266b), p. 180, where two n-words occur, the resumptive quantifier could also appear in three other variants: “$no((x,y),(book(y),student(x)),\ read(x,y))$”, “$no((y,x),(book(y),$ $student(x)),read(x,y))$” and “$no((y,x),(student(x),book(y)),\ read(x,y))$”. The variables may in principle appear in any order, independently of the position/ syntactic role of the NP quantifier that contributed them. Similarly, the order of the restrictions is undetermined, and may even be different from that of the corresponding variables. In a grammar implementation this indeterminacy would trigger much undesired ambiguity. In our grammar, this should be harmless as long as the truth conditions of the utterance are not affected.

The parallelism between the order of the variables and that of the corresponding restrictions could easily be ensured either globally, in the SEMANTICS PRINCIPLE, or locally, in the lexical entries of the determiners. But since this adjustment would have no effect on the interpretation of the LRS structures and would complicate our grammar, we keep the order correspondence undetermined. The important concern with respect to variables and their restrictions is that a variable contributed by one determiner must be restricted by the predicate contributed by the corresponding N and not by some other predicate. As apparent in the lexical entry (265a), p. 177, our grammar does take care of this.

The order of the variables in a resumptive quantifier could be constrained to correspond to the linear order of the NPs or to the syntactic structure of the sentence. If it were related to the linear order, this could be analyzed in a linearization account following Reape (1994) and Kathol (1995). If it were related to the kind of syntactic structure (i.e. objects of sort *head-struc*), the variable contributed by the subject daughter could, for instance, be taken to appear first on any VAR list of a resumptive quantifier, the one contributed by the direct object as second and so on.

Concerning the second possible problem mentioned above, variables contributed by different words are taken to be different in LRS, so every new quantifier brings in a new variable. Thus it will never be the case that by identifying two monadic quantifiers we end up with only one still monadic quantifier or that two independent negative quantifiers end up binding the same variable.

The third problem is related to the previous one and concerns the way we determine that the VAR/ RESTR value of the quantifier contributed by the NM in *nu a venit* in FIGURE 5.8 is the empty list. The lexical entry leaves the list unspecified and there is no variable that must be a member of this list. Fixing the value of VAR to *elist* may not be very important for the structure in FIGURE 5.8. But it is important in FIGURE 5.9, where in order to exclude the DN readings between an n-word and the NM, we have to make use of the only difference between the quantifiers they contribute: the number of variables they must bind. This difference is essential for the NEG CRITERION for Romanian given in (324). So a mechanism is needed in LRS to ensure at the utterance level that a quantifier can only bind those variables that are contributed by the words involved in the structure and if no variable is enforced to appear on one VAR list, this list must be empty.

5.7 Locality conditions on NC

In Chapter 3 I concluded, among others, that the role of the NM in NC is to mark the scope of the negative quantifier contributed by an n-word. I showed that the scope of a negative quantifier appearing in an embedded subjunctive clause is the matrix clause if the matrix verb has a NM (331a), or the embedded clause if the embedded verb has a NM (331b):

(331) a. Ion **nu** i-a cerut Mariei să citească **nicio** carte.
John NM CL-has asked Mary SJ read no book
'There is no book that John asked Mary to read.'

b. Ion i-a cerut Mariei să **nu** citească **nicio** carte.
John CL-has asked Mary SJ NM read no book
'John asked Mary not to read any book.'

So far I have only considered simple sentences in this chapter. In this section I will briefly show how we can account for the scope properties of the negative quantifiers contributed by n-words with respect to the position of the NM in complex sentences.

Note that a fully developed analysis of contexts like (331) necessitates a logical language that deals with intensionality. The representation language $Ty1$ that I defined in Section 5.1 for this grammar can only describe extensional contexts, since it does not include a type for worlds. For the limited purposes of this section, I will continue using our language without worlds, since the goal here is not to offer an account of intensionality, but to simply show how the syntax-semantics of NC provided here can account for the licensing conditions between n-words and the NM over subjunctive clause boundaries.

5.7.1 Licensing of embedded n-words

Let us now have a look at how we can analyze the sentence (331a) with the present apparatus.[34] I take the lexical specifications in (332) for *Ion* 'John', *Mariei* 'Mary', *nu i-a cerut* 'NM CL-has asked'[35], *să* (subjunctive marker) and *citească* 'read'. The lexical specification for proper names follow Richter and Kallmeyer (2007), the one for verbs taking a sentential complement follows Sailer (2006), adapted to the extensional fragment here. For *nicio carte* 'no book' I follow the example in (265), p. 177. The lexical entry for the marker *să* follows Pollard and Sag (1994) to which I add the semantic specification under LF. For Romanian, we will assume that *să* is a subsort of the sort *marked* that we discussed in Section 2.3, repeated in (333) below. The LRS SEMANTICS PRINCIPLE enforces a marker to identify its INC value with that of the head, in our case, $\boxed{3}$. The INCONT PRINCIPLE enforces $\boxed{3}$ to be also an element of the PARTS list and since there is no further semantic contribution from the marker, its EXC value will be $\boxed{3}$ as well. We thus obtain the structure in FIGURE 5.10, p. 217.

(332) a. Lexical entry for *Ion*:

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \texttt{ion} \rangle \\ \text{SS}\,|\,\text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix}\text{HEAD} & \textit{noun}\\ \text{VAL} & \begin{bmatrix}\text{SPR} & \langle\rangle \\ \text{COMPS} & \langle\rangle\end{bmatrix}\end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{INDEX}\,|\,\text{VAR} & \boxed{1}\, john' \\ \text{MAIN} & \boxed{1}\, john'\end{bmatrix}\end{bmatrix} \\ \text{LF} & \begin{bmatrix} \textit{lrs} & \\ \text{EXCONT} & \textit{me} \\ \text{INCONT} & \boxed{1}\, john' \\ \text{PARTS} & \langle \boxed{1}\, john' \rangle \end{bmatrix}\end{bmatrix}$$

b. Lexical entry for *Mariei*:

[34] The analysis for (331b) does not raise any issues concerning the licensing of the n-word, which is as in simple sentences.

[35] I consider the dative clitic *i* 'her' to be part of the lexical specification of the verb as the output of a lexical rule (see Monachesi (1998)) similarly to the NM *nu*. In this sentence the verb still combines with the indirect object *Mariei* 'Mary', so the presence of the clitic does not affect the valence of the verb. See also footnote 33.

$$
\begin{bmatrix}
\textit{word} & \\
\text{PHON} & \langle \texttt{mariei} \rangle \\
\text{SS} \mid \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\rangle \\ \text{COMPS} & \langle\rangle \end{bmatrix} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{INDEX} \mid \text{VAR} & \boxed{4}\, mary' \\ \text{MAIN} & \boxed{4}\, mary' \end{bmatrix} \end{bmatrix} \\
\text{LF} & \begin{bmatrix} \textit{lrs} & \\ \text{EXCONT} & \textit{me} \\ \text{INCONT} & \boxed{4} \\ \text{PARTS} & \langle \boxed{4} \rangle \end{bmatrix}
\end{bmatrix}
$$

c. Lexical rule output for *nu i-a cerut*:

$$
\begin{bmatrix}
\textit{word} & \\
\text{PHON} & \langle \texttt{nu i-a cerut} \rangle \\
\text{SS} \mid \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \boxed{9} \begin{bmatrix} \textit{verb} & \\ \text{NEG} & + \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SUBJ} & \left\langle \text{NP}_{\boxed{1}} \right\rangle \\ \text{COMPS} & \left\langle \text{NP}_{\boxed{4}}, \text{VP} \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{VAL} \mid \text{SUBJ} \left\langle \text{NP}_{\boxed{4}} \right\rangle \\ \text{MARKING} \;\; \textit{să} \end{bmatrix} \\ \text{CONT} \mid \text{MAIN} & \boxed{3a} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{INDEX} \mid \text{VAR} & \textit{no-var} \\ \text{MAIN} & \boxed{2a}\, ask' \end{bmatrix} \end{bmatrix} \\
\text{LF} & \begin{bmatrix} \textit{lrs} & \\ \text{EXCONT} & \boxed{10} \\ \text{INCONT} & \boxed{2}\, ask'(\boxed{1}, \boxed{4}, \eta) \\ \text{PARTS} & \left\langle \boxed{2}, \boxed{2a}\, ask', \boxed{7}\, no(v, \alpha, \beta) \right\rangle \end{bmatrix}
\end{bmatrix}
$$

$$
\wedge\ \boxed{3a} \triangleleft \eta \wedge \boxed{2} \triangleleft \beta \wedge \boxed{7} \triangleleft \boxed{10}
$$

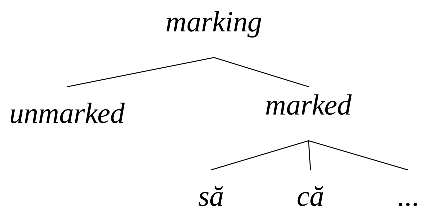
d. Lexical entry for *să*:

$$
\begin{bmatrix}
\textit{word} & \\
\text{PHON} & \langle \texttt{să} \rangle \\
\text{SS} \mid \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{marker} & \\ \text{SPEC} & \boxed{12}\ \text{VP} \begin{bmatrix} \text{LOC} \mid \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{VFORM} & \textit{subj} \end{bmatrix} \\ \text{MARKING} & \textit{unmarked} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SUBJ} & \langle\rangle \\ \text{SPR} & \langle\rangle \\ \text{COMPS} & \langle\rangle \end{bmatrix} \\ \text{MARKING} & \boxed{14}\ \textit{să} \end{bmatrix} \\
\text{LF} & \begin{bmatrix} \text{EXC} & \boxed{3} \\ \text{INC} & \boxed{3} \\ \text{PARTS} & \langle \boxed{3} \rangle \end{bmatrix}
\end{bmatrix}
$$

e. Lexical entry for *citească*:

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \texttt{citească} \rangle \\ \text{SS } \boxed{12} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD } \boxed{8} & \begin{bmatrix} \textit{verb} & \\ \text{NEG} & - \\ \text{VFORM} & \textit{subj} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SUBJ} & \left\langle \text{NP}_{\boxed{4}} \right\rangle \\ \text{COMPS} & \left\langle \text{NP}_{\boxed{6a}} \right\rangle \end{bmatrix} \\ \text{MARKING} & \textit{unmarked} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{INDEX}|\text{ VAR} & \textit{no-var} \\ \text{MAIN} & \boxed{3a}\, read' \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{LF} & \begin{bmatrix} \textit{lrs} & \\ \text{EXCONT} & \boxed{0} \\ \text{INCONT} & \boxed{3}\, read'(\boxed{4}, \boxed{6a}) \\ \text{PARTS} & \langle \boxed{3}, \boxed{3a} \rangle \end{bmatrix} \end{bmatrix} \wedge \boxed{3} \triangleleft \boxed{0}$$

(333)



At the lowest level in the tree the quantifier NP carries the constraint that its INC value $\boxed{5}$ be a component of a member of the restriction list ϕ of the negative quantifier. This is imposed by the first clause of the SEMANTICS PRINCIPLE. Moreover, at the embedded VP level the second clause of the same principle enforces the INC value $\boxed{3}$ of the verb to be a subterm of the nuclear scope ψ of the quantifier. This VP will then be marked by *să* and the marked VP will become a complement to the matrix VP *nu i-a cerut Mariei* 'NM CL-has asked Mary'. Since the embedded VP is now a non-head daughter, the first clause of the EXCONT PRINCIPLE constrains its EXCONT value $\boxed{0}$ to be a member of its PARTS list $\boxed{13}$. Given the third clause of the LRS PROJECTION PRINCIPLE, which constrains the PARTS list of a mother node to collect all the PARTS elements of the daughters, it means that $\boxed{0}$ will be either identical to $\boxed{3}$, the INCONT value of the verb *citească*, or to $\boxed{6}$, the EXCONT value of the quantifier *nicio carte*. So we will either have $\boxed{0} = \boxed{3}$ or $\boxed{0} = \boxed{6}$. At the S level the second clause of the EXCONT PRINCIPLE enforces the EXCONT value $\boxed{6}$ of the negative quantifier contributed by the n-word to be a subterm of the EXCONT value $\boxed{10}$ of the whole sentence.

Interpretation Given these constraints together with the ones carried by the lexical specifications in (332), the value of $\boxed{10}$ can be determined by fixing the value of the metavariables η, ψ, and β which depend on the scope interaction between the two negative quantifiers $\boxed{6}$ and $\boxed{7}$.

We know from (332c) that the matrix verb's MAIN value must be in the nuclear scope β of the negative quantifier $\boxed{7}$. So it is only the negative quantifier $\boxed{6}$ that can take narrow or wide scope with respect to ask'. If it has narrow scope, we get the interpretation in (334a) which is ruled out by the NCC, because $no(y, book'(y), read'(mary', y))$ does not outscope the MAIN value ask' and the verb *citească* whose MAIN value it outscopes is not [NEG +].

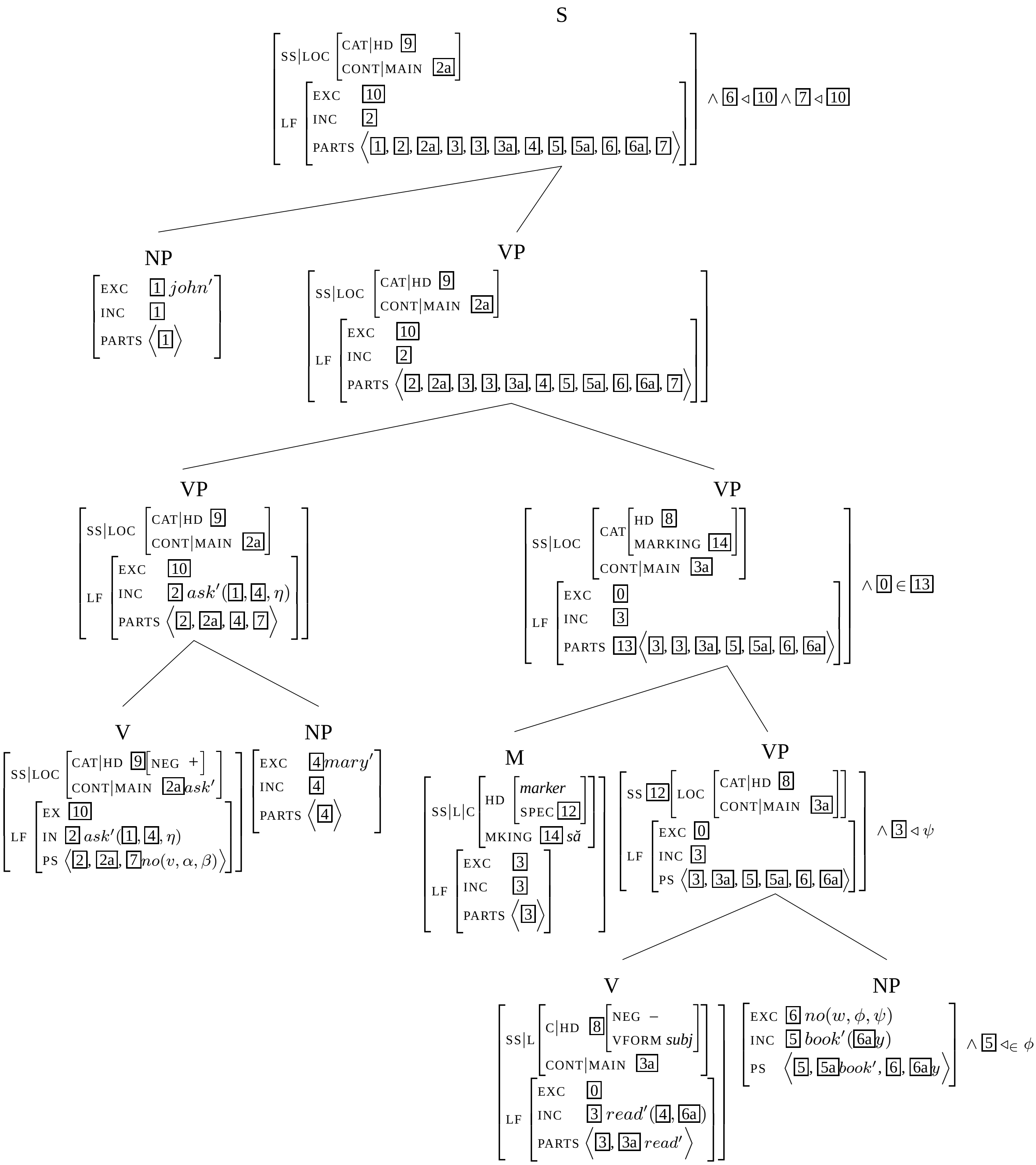


Figure 5.10: LRS analysis of (331a) *Ion nu i-a cerut Mariei să citească nicio carte*

If the negative quantifier $\boxed{6}$ takes scope over ask', independently of whether it outscopes $\boxed{7}$ (334c) or not (334b), the interpretation violates the NEG CRITERION for Romanian which disallows a $\langle 0 \rangle$ quantifier to cooccur with another negative quantifier that takes scope over the same verb's MAIN value. The only possible interpretation is thus the one in which the two negative quantifiers concord (334d) and this also gives us the right reading for (331a). This means that our analysis makes the right predictions for the licensing of embedded n-words by a NM on the matrix verb.

(334) Interpretation for (331a) (the value of $\boxed{10}$ in FIGURE 5.10)

a. * $no((), (), ask'(john', mary', no(y, book'(y), read'(mary', y))))$
for $\boxed{10}$ = $\boxed{7}$, η = $\boxed{6}$

b. * $no((), (), no(y, book'(y), ask'(john', mary', read'(mary', y))))$
for $\boxed{10}$ = $\boxed{7}$, β = $\boxed{6}$

c. * $no(y, book'(y), no((), (), ask'(john', mary', read'(mary', y))))$
for $\boxed{10}$ = $\boxed{6}$, ψ = $\boxed{6}$

d. $no(y, book'(y), ask'(john', mary', read'(mary', y)))$
for $\boxed{10}$ = $\boxed{7}$ = $\boxed{6}$, $\eta = \boxed{3}$, $\beta = \psi = \boxed{2}$

5.7.2 Scope ambiguity related to NC licensing

Another construction that is worth considering in this discussion is one where both verbs carry a NM. If this is the case, the embedded negative quantifier may take scope in the matrix or in the embedded clause. The overall interpretation will involve two negations each time:[36]

(335) Ion **nu** i-a cerut Mariei să **nu** citească **nicio** carte.
John NM CL-has asked Mary SJ NM read no book

i. ‘There is no book that John asked Mary not to read.’
ii. ‘John didn't ask Mary not to read any book.’

But let us first consider the simple case in (336) with both negated verbs and no n-word to see if our analysis makes the right predictions about the scope interaction between two type $\langle 0 \rangle$ negative quantifiers.

(336) Ion **nu** i-a cerut Mariei să **nu** citească *Nostalgia*.
John NM CL-has asked Mary SJ NM read nostalgia-the

John didn't ask Mary not to read *The Nostalgia*.’

If we take (337) as the lexical specification for *nu citească* and consider the book title *Nostalgia* a proper name with a lexical entry similar to the one for *Ion*, we obtain the tree structure in FIGURE 5.11, p. 220 for (336).

[36]If two n-words are present in the embedded clause, the prediction here is that we should get several readings: both n-words interpreted in the main clause, or in the embedded clause, or one of them in the matrix and the other in the embedded clause. It is unlikely that native speakers would be able to obtain all these readings, given the difficulty one usually has in processing several negations in language. It would be an interesting issue for future research to see what constraints are at play in such situations and which readings are preferred.

(337) Lexical rule output for *nu citească*:

$$\begin{bmatrix} \textit{word} \\ \text{PHON}\ \ \langle \texttt{nu citească} \rangle \\ \text{SS} \mid \text{LOC}\ \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD}\ \boxed{8} \begin{bmatrix} \textit{verb} & \\ \text{NEG} & + \\ \text{VFORM} & \textit{subj} \end{bmatrix} \\ \text{VAL} \begin{bmatrix} \text{SUBJ} & \left\langle \text{NP}_{\boxed{4}} \right\rangle \\ \text{COMPS} & \left\langle \text{NP}_{\boxed{6a}} \right\rangle \end{bmatrix} \\ \text{MARKING}\ \ \textit{unmarked} \end{bmatrix} \\ \text{CONT} \begin{bmatrix} \text{INDEX} \mid \text{VAR} & \textit{no-var} \\ \text{MAIN} & \boxed{3a}\, read' \end{bmatrix} \end{bmatrix} \\ \text{LF}\ \begin{bmatrix} \textit{lrs} & \\ \text{EXCONT} & \boxed{0} \\ \text{INCONT} & \boxed{3}\, read'(\boxed{4}, \boxed{15}) \\ \text{PARTS} & \left\langle \boxed{3}, \boxed{3a}, \boxed{11} no(u, \gamma, \delta) \right\rangle \end{bmatrix} \end{bmatrix} \wedge \boxed{3} \triangleleft \boxed{0} \wedge \boxed{3} \triangleleft \delta \wedge \boxed{11} \triangleleft \boxed{0}$$

In FIGURE 5.11 the embedded verb is negative so it carries a negative quantifier $\boxed{11}$ on its PARTS list. According to the EXCONT PRINCIPLE $\boxed{11}$ must also be a subterm of the EXCONT value of the whole sentence ($\boxed{11} \triangleleft \boxed{10}$). The constraint $\boxed{0} \in \boxed{13}$ at the level of the marked embedded VP is now equivalent to $\boxed{0} = \boxed{11}$, since $\boxed{11}$ is the expression with the widest scope among the ones on the PARTS list $\boxed{13}$ ($\boxed{3}$ is a subterm of the nuclear scope of $\boxed{11}$).

Interpretation We again have two operators, so the interpretation of the sentence depends on the scope interaction between them. However, the situation is different from the one in FIGURE 5.10 where one operator was contributed by an NP. While empirical tests indicate that such operators can take scope in the matrix clause (see (331a) and (335)), this does not apply to the $\langle 0 \rangle$ quantifier contributed by the NM whose scope is limited to the embedded clause (see (336) and (338)). This issue is partially taken care of by the NM LEXICAL RULE in (327) which constrains the $\langle 0 \rangle$ negative quantifier to be a subterm of the EXCONT value of the verb ($\boxed{11} \triangleleft \boxed{0}$). But note that at this point nothing prevents the EXCONT value $\boxed{0}$ of the embedded verb in FIGURE 5.11 to be identified with the EXCONT value $\boxed{10}$ of the matrix clause. In this case, a NM on the embedded verb would also be able to negate the affirmative matrix verb in a sentence like (338), which is contrary to fact.

(338) Ion i-a cerut Mariei să **nu** citească *Nostalgia*.
John CL-has asked Mary SJ NM read nostalgia-the

i. 'John asked Mary not to read *The Nostalgia*.'
ii. # 'John didn't ask Mary to read *The Nostalgia*.'

This means that we should restrict the EXCONT value of an embedded verb to stay in the scope of the matrix verb. So for propositional attitude verbs like *ask* we will introduce a fourth clause of the LRS SEMANTICS PRINCIPLE that specifies this condition as in (339) below:[37]

(339) THE SEMANTICS PRINCIPLE

4. if the head-daughter of a phrase has a MAIN value which takes a propositional argument

[37]Notice that (339) is only an informal description of this principle. A proper specification would require intensional operators, which are not provided by our language.

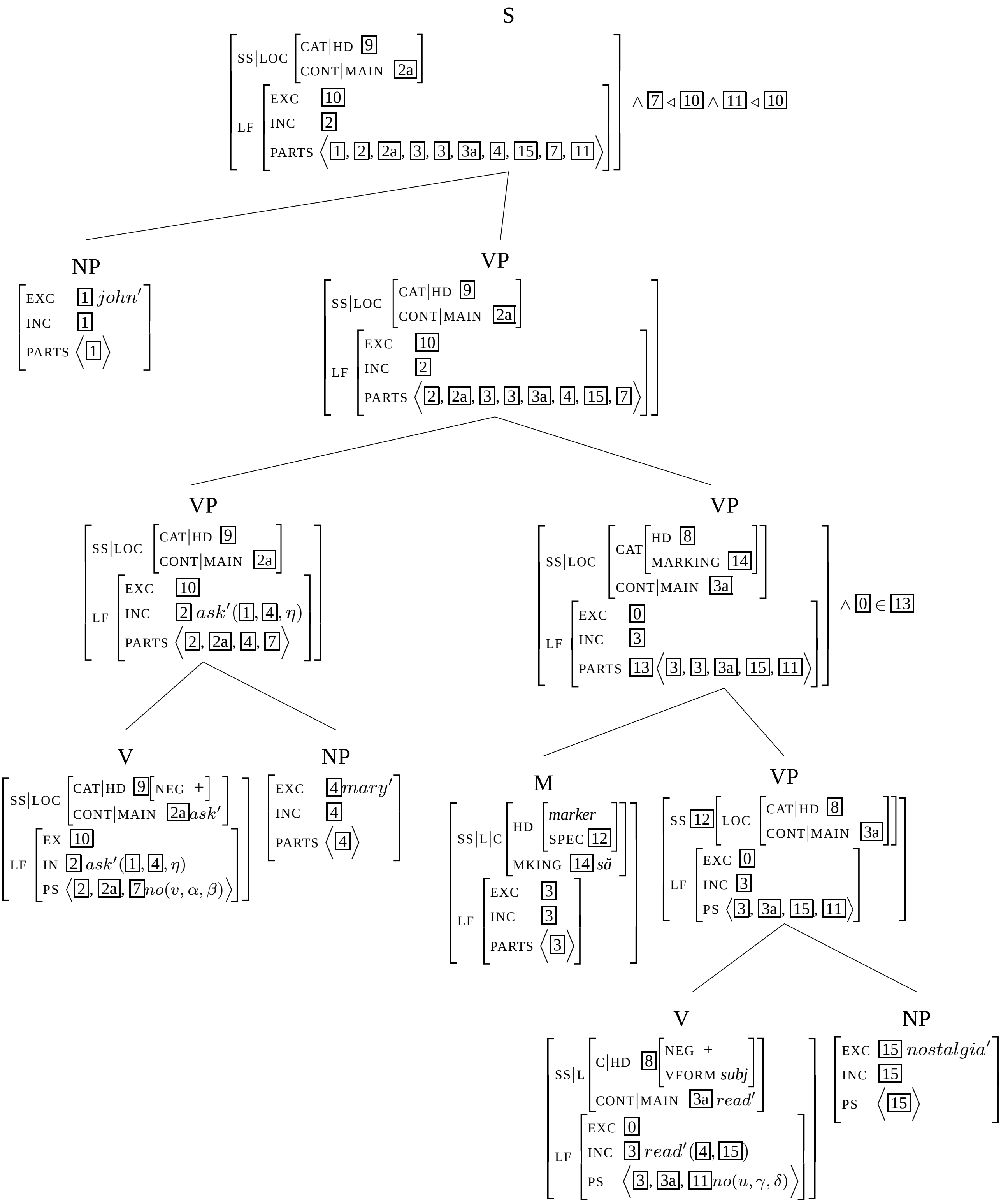


Figure 5.11: LRS analysis of (336) *Ion nu i-a cerut Mariei să nu citească* Nostalgia

η and the non-head-daughter is a propositional complement, then the EXCONT value of the complement must be a subterm of η.

With this clause, at the highest matrix VP level in FIGURE 5.11 we have $\boxed{0} \triangleleft \eta$ which now prevents identifying $\boxed{0}$ with $\boxed{10}$. Note, however, that the lexical constraint on $\langle 0 \rangle$ negative quantifiers to be subterms of the EXCONT of the verb they negate (here, $\boxed{11} \triangleleft \boxed{0}$) is still needed in order to rule out the case where the negative quantifier $\boxed{11}$ takes scope in the matrix clause independently of the condition $\boxed{0} \triangleleft \eta$. As we will see below, a negative quantifier contributed by an n-word still has this possibility, because it does not have to stay within the EXCONT of the verbal head.

With all these specifications in our grammar, the only expression we obtain for the EXCONT value $\boxed{10}$ is the one in (340) which gives us the right interpretation for (336).

(340) Interpretation for (336) (the value of $\boxed{10}$ in FIGURE 5.11)

$$no((),(),ask'(john',mary',no((),(),read'(mary',nostalgia'))))$$

We are now at the point where we can proceed with the analysis for the ambiguous sentence in (335). The corresponding tree structure is given in FIGURE 5.12. It differs from FIGURE 5.10 only in the embedded verb which is [NEG +] and carries a negative quantifier $\boxed{11}$ on its PARTS list. This now introduces the possibility of interpreting the embedded negative quantifier $\boxed{6}$ in the lower clause, thus giving rise to the reading in (335ii). If we compare this structure to the one in FIGURE 5.11, the difference is that we have the negative quantifier $\boxed{6}$ instead of the proper name $\boxed{15}$ as the direct object of the embedded verb. At the S level, the EXCONT PRINCIPLE enforces this negative quantifier to be a subterm of the EXCONT value $\boxed{10}$ of the matrix clause.

Interpretation The first reading of (335) is similar to the interpretation we derived in FIGURE 5.10. That means we can obtain it if the negative quantifier $\boxed{6}$ concords with the negative quantifier $\boxed{7}$ contributed by the matrix NM. In this case, the embedded NM simply negates the embedded verb, so the latter's EXCONT value $\boxed{0}$ is identified with $\boxed{11}$. This interpretation is given in (341a). To get the second reading of (335), the negative quantifier $\boxed{6}$ must concord with the embedded NM $\boxed{11}$ ($\boxed{6}$ = $\boxed{11}$). For this, the EXCONT value $\boxed{0}$ which is a member of the PARTS list $\boxed{13}$, gets identified with the quantifier $\boxed{6}$ at the level of the marked embedded VP. We obtain the interpretation in (341b). Any other possible scope interactions between the three negative quantifiers $\boxed{6}$, $\boxed{7}$ and $\boxed{11}$ are ruled out by the analysis, as already shown with respect to FIGURE 5.10.

(341) Interpretation for (335) (the value of $\boxed{10}$ in FIGURE 5.12)

a. $no(y,book'(y),ask'(john',mary',no((),(),read'(mary',y))))$
for $\boxed{10}$ = $\boxed{7}$ = $\boxed{6}$, $\boxed{0}$ = $\boxed{11}$ = η, $\psi = \beta$

b. $no((),(),ask'(john',mary',no(y,book'(y),read'(mary',y))))$
for $\boxed{10}$ = $\boxed{7}$, $\boxed{0}$ = $\boxed{6}$ = η = $\boxed{11}$, $\psi = \delta$

In conclusion, the negative quantifier contributed by the n-word must concord either with the negative quantifier of the matrix NM (341a) or with that of the embedded NM (341b) and this gives us the right readings for (335). Note that the NCC is satisfied in (341), since both verbs are [NEG +] and can thus license the n-word.

In this section, I showed how the locality of n-word licensing can be accounted for in the LRS-analysis developed here. Other locality conditions concerning the scope of negative quantifiers which

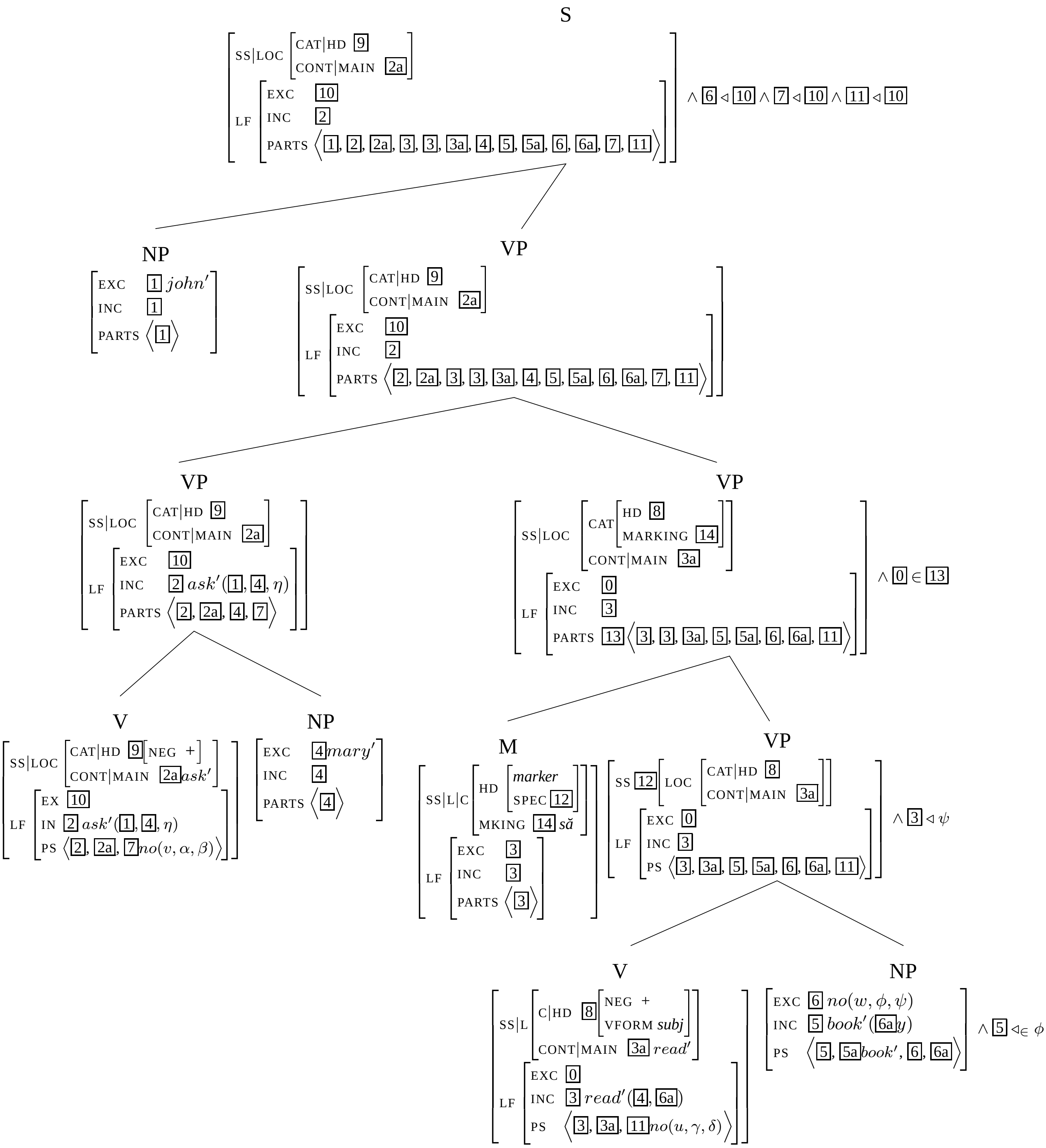


Figure 5.12: LRS analysis of (335) *Ion nu i-a cerut Mariei să nu citească nicio carte*

coincide with the conditions on quantifier scope in general (see Section 3.3.3) can be accounted for in a general theory of quantifier scope, which is not attempted in this thesis.

5.8 Conclusion

To summarize the results of this chapter, I proposed a core analysis of the syntax-semantics of NC constructions in Romanian. This analysis takes into account the negative semantics of n-words and the NM, and the scope properties of negative quantifiers in NC and DN readings, as documented in Chapters 3 and 4. The syntax-semantics interface is provided by HPSG and the semantic platform that allows the integration of polyadic quantifiers is LRS.

To express resumptive negative quantifiers in LRS, I first defined them in the representation language $Ty1$. For this language I then defined the RSRL grammar Γ_{Ty1} which allows the use of $Ty1$ expressions as semantic representations in HPSG. The LRS SEMANTICS PRINCIPLE had to be reformulated to cover resumptive quantifiers and a new clause was added that deals with propositional attitude verbs. With these minimal extensions to LRS and a few constraints required in part by NC constructions in general (NEG CRITERION, NM LEXICAL RULE, NEGATION COMPLEXITY CONSTRAINT for Romanian), in part by language-specific properties of Romanian NC (DN PRINCIPLE, INF-STR CONSTRAINT ON NC, NC CONSTRAINT) we can thus account for the core properties of NC in Romanian.

I showed how the present analysis accounts for the following: 1) NC readings in simple sentences with two or more n-words, 2) DN readings for sentences with two n-words, 3) the scope interaction between two negative quantifiers and one non-negative quantifier and its effects on the interpretation of the sentence as NC or DN, 4) the ability of the NM to negate a sentence on its own, 5) the lack of a DN reading between a NM and an n-word, 6) the ungrammaticality of n-words in finite sentences without a NM and 7) the obligatory disambiguation of the scope of an embedded n-word depending on whether the NM is on the embedded or the matrix verb. This covers the main properties of NC constructions in Romanian described in Chapter 3.

Chapter 6

Comparison to other approaches

In this thesis I have argued for a treatment of n-words as negative quantifiers and offered an account of negative concord as a resumptive negative quantifier. In this chapter I will compare relevant aspects of the analysis proposed here with other approaches in the literature. I will first discuss the so-called 'NPI approaches'[1], where n-words are considered non-negative, and then the 'NQ approaches', where n-words are analyzed as negative.

NPI approaches attribute a non-negative semantics to n-words in order to avoid the compositionality problem that NC would otherwise raise (see Laka (1990), Ladusaw (1992), Progovac (1994), Acquaviva (1997), Déprez (1997), Giannakidou (1998), Richter and Sailer (1999*b*), Przepiórkowski and Kupść (1999), Zeijlstra (2004), Giannakidou (2006), Penka (2007), among many others). This contrasts with the present analysis. Some of their arguments were rejected for Romanian n-words in Chapter 3. In Section 6.1 I will discuss the other empirical tests used in this literature.

The NQ approaches claim that n-words are negative quantifiers, so they typically offer a solution for the compositionality problem (see Zanuttini (1991), Haegeman and Zanuttini (1991), Haegeman (1995), Haegeman and Zanuttini (1996), de Swart and Sag (2002), Richter and Sailer (2004)). In Section 6.2 I will compare these approaches to the one I have developed in this thesis.

6.1 The NPI approaches to NC

The NPI approaches to NC use a wide variety of empirical tests intended to clarify the semantic status of n-words. In this section I only mention those tests that were not used in Chapter 3 and may challenge the present assumption that Romanian n-words are negative quantifiers. I first address the range of properties that Giannakidou (2006) uses to determine whether n-words are most like existential, universal, or negative quantifiers (Section 6.1.1). I then discuss some independent issues on Romanian NC and n-words mentioned in Ionescu (2004) and Isac (2004). The former raises doubts concerning the semantic status of the NM in an analysis of NC as resumption, the latter discusses the effect of Focus on the quantificational behavior of n-words (Section 6.1.2). In the end I address the split scope readings of n-words on the basis of which Penka (2007) argues that n-words are crosslinguistically indefinites (Section 6.1.3).

[1]Under 'NPI approaches', I subsume all analyses that consider n-words non-negative, independently of whether they are argued to be NPIs, indefinites or non-negative quantifiers.

6.1.1 General tests for n-words: Giannakidou (2006)

The crosslinguistic study in Giannakidou (2006) is intended to establish the semantic status of n-words in NC languages as diverse as Romance, Slavic, Greek, and Hungarian. Most of the tests are also discussed at length in Richter and Sailer (1999*b*) for Polish.

First, Giannakidou rejects both the indefinite and the unambiguous negative quantifier treatment of n-words in NC languages. As we will see, this is due to a specific theoretical perspective on negative quantifiers and indefinites which we can easily reconcile with the view in this thesis. Second, she argues for a lexical ambiguity approach by showing that n-words in NC languages do not display a uniform semantic behavior. I will show that my treatment of Romanian n-words as negative quantifiers is compatible with their apparently non-uniform behavior, if we take into account that negative quantifiers as 'weak quantifiers' (Milsark (1974)) exhibit both 'strong' and 'weak' quantifier properties (see also Déprez (1997)).

Background assumptions in Giannakidou (2006) Giannakidou makes a clear distinction between true negative quantifiers in DN languages (e.g. Germanic in (342)) and n-words in NC languages (Romance, Slavic in (343), see also our discussion in Section 3.1.1). In her view, n-words in NC languages cannot be pure negative quantifiers, given NC. However, in non-strict NC languages like French, Italian, and Spanish, they may be ambiguous between negative quantifiers and a kind of NPIs. The negative status of n-words in these languages is taken to be motivated by the fact that preverbal n-words can contribute negation alone and license other n-words (see *nessuno* and *nadie* in (344)).

(342) Negative quantifiers in DN languages:

a. Frank heeft **niet niemand** gezien.
Frank has not nobody seen
'It is not the case that Frank didn't see anybody.'
'Frank didn't see anybody.' (Dutch)

b. Frank did **not** see **nobody**. (English)

(343) N-words as NPIs in NC languages:

a. Gianni *(**non**) ha visto **niente**.
John not has seen nothing
'John didn't see anything.' (Italian)

b. Milan *(**ne**) vidi **nista**.
Milan not sees nothing
'Milan cannot see anything.' (Serbian/Croatian)

(344) N-words as negative quantifiers in NC languages:

a. **Nessuno** ha letto **niente**/ il libro.
nobody has read nothing/ the book
'Nobody read anything/the book.' (Italian)

b. **Nadie** dijo **nada**/ eso.
nobody said nothing/ this
'Nobody said anything/ this.' (Spanish)

Giannakidou (2006) rejects the indefinite hypothesis, because n-words lack the main property of indefinites, namely, the quantificational variability exemplified in (345) (see also Lewis (1975), Heim

(1982) and subsequent literature). Unlike indefinites, n-words cannot be bound by a quantificational adverb, they remain existentially closed in the VP under the NM, as the Greek examples in (346) show:

(345) Sometimes/ Usually, if a cat falls from the fifth floor, it survives.
'Some/ Most cats that fall from the fifth floor survive.'

(346) a. Sixna/ Pu ke pu, otan o Janis ine thimomenos, **dhen** milai me KANENAN[2].
'Usually/ Sometimes, when John is upset, he talks to nobody.'

b. Usually(s) [John is upset in s][$\neg\exists x(person(x,s) \wedge talk(John,x,s))$]
Sometimes(s) [John is upset in s][$\neg\exists x(person(x,s) \wedge talk(John,x,s))$]

In each of the two representations in (346b) given for (346a), the quantificational adverb binds the situation *s*, but not the n-word. If n-words are indefinites, then they must be of a special kind which can only be existentially bound under negation. In this case, Giannakidou (2006) concludes, one should consider them existential quantifiers.

In Chapter 3 we saw that the Ladusaw (1992) tradition of treating n-words as Heimian indefinites relies precisely on the idea that they have to be existentially bound by negation. The distinction that Giannakidou makes between existential quantifiers and indefinites is mostly terminological, so the arguments she uses to indicate the existential quantifier status of n-words are also relevant for a treatment as indefinites.

Given the two possible representations of a negative statement with quantifiers in (347), Giannakidou proposes that n-words in NC should be either existential or universal quantifier NPIs:

(347) *Logical representations for negative statements*:

a. $\forall x$ [P(x) $\rightarrow \neg$ Q(x)] (Universal negation)

b. $\neg\exists x$ [P(x) $\wedge$ Q(x)] (Existential negation)

Sometimes n-words may exhibit properties of both types of quantifiers within one language: this is the case in Greek where 'emphatic' n-words behave like universal and 'non-emphatic' ones like existential quantifiers. She therefore builds an account of n-words as mainly ambiguous and she identifies the properties that are typical of one behavior or the other.

Crosslinguistic tests Let us now have a look at the inventory of properties that Giannakidou uses in order to determine the semantic status of n-words across languages. Giannakidou (2006) identifies three semantic classes which n-words may belong to or be ambiguous between: existential quantifiers, universal quantifiers, and negative quantifiers. The corresponding properties are enumerated in (348), (349) and (350), respectively.

(348) **Existential n-words**

a. are licensed freely long distance in complement clauses;

b. can be licensed in syntactic islands, e.g. relative clauses and adjunct clauses;

c. cannot be modified by *almost*;

[2] I follow Giannakidou (2006) in spelling the Greek 'emphatic' n-words in uppercase letters. 'Emphatic' n-words in Greek are said to have a different behavior from non-emphatic ones, so they usually receive a separate account (see also Giannakidou (1998)).

d. need not express existential commitment, i.e. we can interpret them with an empty restriction;

e. can bind donkey pronouns;

f. can be used as predicate nominals.

(349) **Universal n-words**

a. are licensed only by local negation; long distance licensing may be allowed only through an infinitival or subjunctive clause;

b. can be modified by *almost*;

c. can be used as topic in topicalization structures; in these cases, they may be coindexed with (clitic) pronouns;

d. express existential commitment, i.e. we tend to interpret them with a non-empty restriction;

e. cannot bind donkey pronouns;

f. cannot be used as predicate nominals.

(350) **Negative n-words**

a. receive negative meaning and exclude sentential negation in the preverbal position;

b. receive negative meaning and exclude sentential negation when they cooccur with another n-word (negative spread); the first n-word is usually in preverbal position;

c. are licensed only by local negation; long distance licensing may be allowed only through an infinitival or subjunctive clause;

d. can be modified by *almost*;

e. can be used as topic in topicalization structures; in these cases, they may be coindexed with (clitic) pronouns;

f. cannot bind donkey pronouns;

g. usually cannot be used as predicate nominals.

Comparing the three classes, the properties of negative n-words are similar to those of universal n-words, if we exclude (350a) and (350b), which make particular reference to their negative content. I start by discussing the first two categories of n-words and show that Romanian n-words have more in common with universals than with existentials, a conclusion that Giannakidou (2006) reaches as well. This is clearly indicated by the tests involving locality and *almost*-modification. The tests for existential commitment and dynamic binding are less clear in this respect, as n-words present an inconsistent behavior. I will attribute this to the dual nature of negative quantifiers as 'weak quantifiers' in Milsark's (1974) terminology. From this perspective, it is not surprising that they exhibit a variable behavior in contexts that are compatible with both universal and existential quantifiers. This is independent of their negative semantics, which I argued for in Section 3.4.

Locality We saw in Section 3.3.3 that in terms of locality the licensing of n-words resembles the scope properties of universal quantifiers in Romanian. The relevant data are repeated below.[3]

[3]Note that *a vrea* 'to want' is not a Neg-Raising verb in Romanian, so the grammaticality of (352) is not due to Neg-Raising. See also the other examples in Section 5.7 with n-word licensing across the subjunctive.

(351) Un student a încercat să citească *fiecare* carte.
a student has tried SJ read every book
'A student tried to read every book.'

a. $\exists > \forall$: A (certain) student tried to read every book.
b. $\forall > \exists$: For every book there is a student who tried to read it.

(352) Ion **nu** a încercat să citească **nicio** carte.
John NM has tried SJ read no book
'John didn't try to read any book.'

(353) a. Un student a zis că a citit *fiecare* carte.
a student has said that has read every book
'A student said that he read every book.'

i. $\exists > \forall$: A (certain) student said that he read every book.
ii. # $\forall > \exists$: For every book there is a student who said that he read it.

b. * **Nu** a zis că a citit **nicio** carte.
NM has said that has read no book

(351) and (353a) show that an embedded universal quantifier can outscope an existential quantifier from a subjunctive clause (reading (351b)), but not over a 'that'-clause (353a-ii). Similarly, an n-word can be licensed by a matrix NM in a subjunctive (352), but not in a 'that'-clause (353b). The data in (353) contrast with the ones in (354) where an existential quantifier can easily outscope a universal quantifier over a 'that'-boundary (reading (354b)).

(354) Fiecare student a zis că a citit *o* carte.
every student has said that has read a book
'Every student said that s/he read a book.'

a. $\forall > \exists$: For every student there is a book such that the former said that he read the latter.
b. $\exists > \forall$: There is a (certain) book such that every student said that he read it.

Relative and adjunct clauses are well-known barriers for quantifier scope, so they provide another test for the status of n-words: if n-words can be licensed over such barriers, they are existential quantifiers (see (348b)), if they cannot, then they are universal quantifiers (349a). In Greek, Giannakidou (2006) shows that emphatic n-words behave like universal quantifiers, while non-emphatic ones behave like existentials (355).

(355) a. **Dhen** prodhosa mistika [pu eksethesan *kanenan*/ **KANENAN*].
NM betrayed secrets that exposed nobody
'I didn't reveal secrets that exposed anybody.'

b. **Dhen** milisa [epidhi ithela na prosvalo *kanenan*/ **KANENAN*].
NM talked because wanted SJ offend nobody
'I didn't talk because I wanted to offend anybody (but because I had to).'

Romanian n-words pattern with Greek emphatic n-words and with universal quantifiers, since they cannot be licensed in relative and adjunct clauses:

(356) a. * **Nu** am dezvăluit secrete [care au expus pe **nimeni**].
NM have disclosed secrets that have exposed PE nobody
'I didn't disclose secrets that exposed anybody.'

b. * **Nu** am vorbit [pentru că mi-a cerut **nimeni**].
NM have talked because me-has asked nobody
'I didn't talk because anybody asked me to.'

As indicated by the data in (357), universal quantifiers embedded in relative (357a) and adjunct clauses (357b) indeed cannot outscope an existential in the matrix clause:

(357) a. Un student a dezvăluit secrete [care (l-)au compromis (pe) *fiecare* profesor].
a student has disclosed secrets that him-have discredited (PE) every teacher

i. $\exists > \forall$: A (certain) student disclosed secrets that discredited every teacher.

ii. # $\forall > \exists$: For every teacher there is some student, such that the latter disclosed secrets that discredited the former.

b. Un student a vorbit [pentru că i-a cerut *fiecare* profesor].
a student has talked because him-has asked every teacher

i. $\exists > \forall$: A (certain) student talked because every teacher asked him to.

ii. # $\forall > \exists$: For every teacher there is some student, such that the latter talked because the former asked him to.

Almost-**modifiers** *Almost*-modification given in (348c)/ (349b) and illustrated in (358) further indicates the similarity between Romanian n-words and universal quantifiers and the contrast with existential quantifiers. Unlike existential NPIs, Romanian n-words can be modified by *almost* (see (359)). As I argued, contra Penka (2006), in Section 3.3.4, the *almost*-modification test is relevant and clearly indicates that n-words are not existential quantifiers at least in Romanian, where *almost* does not seem to be a PPI as Penka (2007) suggests.

(358) a. *Almost* everybody came.

b. * *Almost* somebody came.

(359) a. **Nu** am putut vedea *aproape* **nicio** casă în întuneric.
NM have could see almost no house in darkness
'We could see almost no house in the dark.'

b. * **Nimeni nu** a putut vedea *aproape* vreo casă în întuneric.
nobody NM has could see almost any house in darkness

Topicalization Let us now concentrate on property (349c), that is, that universal n-words can be used as topic in topicalization structures and be coindexed with a clitic. Giannakidou (2006) relates topicalization to Heim's (1982) notion of 'familiarity' within file change semantics. If a quantifier carries an index that has already been introduced in the files of the previous discourse, then it is familiar. Universal quantifiers relate to familiar discourse referents, so they are expected to appear as topics and to be doubled by clitics, as confirmed by the Greek data below (see also Cinque (1990), Rizzi (1997), Giannakidou (2000)). However, for this the universal quantifier must also have a rich descriptive content: bare quantifiers are ungrammatical in such contexts (360b).

(360) a. *Kathe* dhema to paredhosa ston paralipti tu
every parcel it delivered in-the recipient its
'As for every parcel, I delivered it to its recipient.'

b. * *Kathena*, ton idha.
everybody him saw

Universal n-words should exhibit a similar behavior and Giannakidou (2006) shows that Greek emphatic n-words do (361).[4]

(361) a. [KANENAN fititi]$_i$ dhen (ton$_i$) idha na erxete stin ora tu.
no student not him saw SJ come on time his
'I saw no student arriving on time.'

b. * KANENAN$_i$ dhen ton$_i$ idha.
nobody not him saw

Like Greek emphatics, Romanian n-words can undergo topicalization and clitic doubling if enough descriptive content is provided (see Dobrovie-Sorin (1994) and Cornilescu (2002) for discussion):

(362) a. [Pe **niciun** student]$_i$ nu l$_i$-am văzut venind la timp.
PE no student not him-have seen coming on time
'I saw no student arriving on time.'

b. * Pe **nimeni**$_i$ nu l$_i$-am văzut.
nobody not him-have seen

Existential commitment (348d)/ (349d) Horn (1997) argues that universal quantifiers bring about an existence inference, so their restriction cannot be interpreted as empty. Giannakidou (2006) uses this idea as a further test to distinguish between universal and existential n-words, since existential quantifiers in general need not trigger an existence inference. She shows that Greek emphatic n-words bear an existential commitment, while the non-emphatic ones do not:

(363) a. # I Cleo **dhen** idhe kathe/ KANENA monokero.
the Cleo NM saw every/ no unicorn
'Cleo didn't see every unicorn./ Cleo saw no unicorns.'

b. I Cleo **dhen** idhe enan/ *kanena* monokero.
the Cleo NM saw a/ no unicorn
'Cleo didn't see a/ any unicorn.'

In (363a), the universal *kathe* and the emphatic n-word *kanena* make the sentence sound odd, since they suggest the existence of unicorns in the actual world. But the sentence in (363b) involving an existential quantifier or a non-emphatic n-word is fine and can be continued with something like 'because unicorns don't exist'.

Romanian n-words are ambiguous with respect to this test. They do not necessarily trigger existential commitment, so they seem to pattern with existential quantifiers. However, there are contexts where an existential commitment is present. I will exemplify this with clitic doubling.

First, sentence (364) sounds fine in Romanian under the interpretation that John saw zero unicorns because there are no unicorns:

[4] The same is argued for Italian n-words which are concluded to be ambiguous between negative and existential quantifiers.

(364) Ion **nu** a văzut **niciun** unicorn (pentru că nu există unicorni).
John NM has seen no unicorn (because NM exist unicorns)
'John didn't see any unicorn (because there are no unicorns).'

Sentence (364) seems to indicate that Romanian n-words resemble existential quantifiers, although the previous tests pointed to a clear similarity with universal quantifiers. The context in (364) allows a quantifier with or without an empty restriction and the continuation 'because there are no unicorns' cancels a possible existence inference. But in contexts where the existential commitment is forced, n-words are still grammatical and sound odd with a continuation that cancels the existence inference.

We mentioned that clitic doubling is possible with an n-word. Clitics are known to require a discourse-linked/ specific and/ or familiar reading of the NPs they double (see Dobrovie-Sorin (1994)). Thus Romanian n-words in clitic doubling contexts do indeed trigger an existential commitment like the Greek emphatic n-words: the continuation 'because he has no students in his class' makes (365) sound contradictory, since the clitic doubled n-word suggests the existence of a set of students which is then denied:

(365) Ion **nu** l$_i$-a văzut pe [**niciun** student]$_i$ venind la timp (# pentru că nu are studenţi deloc la curs).
John NM has seen PE no student comming at time (because NM has students at all at class
'John didn't see any of the students coming on time (because he has no students in his class).'

Opaque contexts also provide evidence for the presence of an existential commitment with n-words. An indefinite occuring as the direct object of verbs like 'seek' usually gives rise to two readings: *de re* (366a) and *de dicto* (366b). In the former reading the existential quantifier is assumed to take widest scope, in the latter it takes narrow scope with respect to the property that the opaque verb requires. The de re reading thus presupposes a non-empty restriction for the quantifier.

(366) Ion caută *o* secretară.
John seeks a secretary

a. There is a certain secretary and John seeks her. (de re)
b. John is involved in a search for a secretary. (de dicto)

The possibility of a de re interpretation for the n-word in (367) indicates existential commitment:

(367) Ion **nu** caută **nicio** secretară.
John NM seeks no secretary

a. There is no secretary such that John seeks her. (de re)
b. ? It is not the case that John is involved in a search for a secretary. (split scope)
c. # John is involved in a search for no secretary. (de dicto)

However, note that the availability of the de re reading is not an argument for the universal quantifier status of n-words and against the existential one, since this reading can easily be expressed with an existential quantifier, too, and existential quantifiers do not exclude existential commitment. The problem that the universal quantifier assumption raises for opaque contexts is that it cannot account for the other possible reading in (367b), as pointed out by Richter and Sailer (1999*b*).[5] Universal quantifiers usually only get a de re reading in opaque contexts (368).

(368) Ion caută *fiecare* secretară.
John NM seeks every secretary
'Every secretary is such that John seeks her'.

For now, I note that the reading (367b) is rather marked in Romanian, which I indicate with the '?' symbol. The neutral construction expressing the same meaning would have a bare noun instead of the n-word. I will come back to this issue in Section 6.1.3.

Thus we can conclude that Romanian n-words do not necessarily trigger existential commitment. At the end of this section I will show that this is compatible with an analysis of n-words as negative quantifiers.

Donkey pronouns and dynamic binding The test in (348e) and (349e) is used in Giannakidou (1998, 2006) as a further criterion to determine the status of n-words and is also discussed and partially refined in Richter and Sailer (1999*b*). Existential quantifiers usually bind donkey pronouns, but universal quantifiers cannot, a contrast that occurs in Romanian as well (369). Thus the impossibility to bind the pronoun in (370) should indicate that n-words behave like universal quantifiers.

(369) Studenţii care au cumpărat [*o*/ **fiecare* carte]$_i$, s-o$_i$ aducă cu ei.
students-the who have bought a/ every book SJ.-it bring with them
'The students who bought [a/ *every book]$_i$ should bring it$_i$ with them.'

(370) * Studenţii care **nu** au cumpărat [**nicio** carte]$_i$, s-o$_i$ aducă cu ei.
students-the who NM have bought no book SJ.-it bring with them
'The students who didn't buy [any book]$_i$ should bring it$_i$ with them.'

But Richter and Sailer (1999*b*) suggests that the ungrammaticality of sentences like (370) may be due to the presence of negation which blocks anaphoric binding. This seems to be the case with existential quantifiers as well:

(371) a. * The students that didn't buy [*any*/ *some* book]$_i$ should show it$_i$ now.
(Giannakidou (2006))

b. * Studenţii care **nu** au cumpărat [**o** carte]$_i$, s-o$_i$ aducă cu ei.
students-the who NM have bought a book SJ.-it bring with them
'The students who didn't buy some book$_i$ should bring it$_i$ with them.'

Instead Richter and Sailer provide another context with dynamic binding across negation (372) following an example in Roberts (1989) attributed to Barbara Partee. They show for Polish that universal quantifiers still cannot bind the anaphora in that context, so the contrast seems to indicate that n-words pattern with existential quantifiers. In (373) I give the relevant examples for Romanian.

(372) Either there's no bathroom in this house or it's in a funny place.

[5] Richter and Sailer (1999*b*) call this reading 'de dicto'. Following the discussion on the readings that n-words get with modal verbs (see Jacobs (1980, 1991), de Swart (2000), among others) I will call this reading 'split scope' and distinguish it from the unavailable de dicto reading in (367c). See also the discussion in Section 6.1.3.

(373) a. În casa asta ori nu există **nicio** baie, ori au construit-*o* într-un loc ciudat.
in house this either NM exists no bathroom, either have built-it in-a place strange
'Either there is no bathroom in this house, or they built it in a strange place.'

b. În casa asta ori nu există *baie*, ori au construit-*o* într-un loc ciudat.
in house this either NM exists bathroom, either have built-it in-a place strange
'Either there doesn't exist a bathroom in this house, or they built it in a strange place.'

c. * Ori *fiecare* cîine de pe strada asta **nu** mai latră, ori *l*-au alungat tunetele.
either every dog in street this NM more barks either it-have scared-away thunders
'Either every dog in this street doesn't bark anymore, or the thunders scared it away.'

In (373a) the n-word *nicio baie* can bind the anaphor *o* just like the bare noun *baie* in (373b). The universal quantifier *fiecare cîine* cannot bind the pronoun *l* in (373c). This seems to indicate that Romanian n-words behave like existential quantifiers.

N-words as weak quantifiers Notice, however, that the context in (373a) is similar to existential 'there'-contexts in English which only allow 'weak' readings of weak quantifiers. Unambigously 'strong' NPs like universal quantifiers and definite NPs are ungrammatical (see Milsark (1974)):

(374) a. There is *a/ no/ *every/ *the* bathroom in this house.

b. There are *two/ many/ no/ *all/ *the* bathrooms in this house.

The proposal advanced in this thesis is that n-words are negative quantifiers, so in Milsark's classification they pattern with weak quantifiers and are expected to be grammatical in existential sentences under their weak reading (374). But weak quantifiers are also known to exhibit a 'strong' reading with individual-level predicates like in (375a), which allow universal quantifiers (Diesing (1992), Kratzer (1995)). Romanian bare nouns, which always take narrow scope and never get a strong interpretation, are excluded in such contexts (375a), although they are grammatical in existential ones (375b):

(375) a. [*Fiecare* student/ *niciun* student/ *studentul*/ *mulţi* studenţi/ *trei* studenţi/ **student*] (nu) e/ sînt inteligent(i).
every student/ no student/ student-the/ many students/ three students/ student (NM) is/ are intelligent
'[Every student/ no student/ student-the/ many students/ three students/ a student] is/ are intelligent.'

b. În casa asta (nu) există [*nicio* baie/ *multe* băi/ *trei* băi/ baie/ *baia/ **fiecare baie*].
in house this (NM) exists no bathroom/ many bathrooms/ three bathrooms/ bathroom/ bathroom-the/ every bathroom
'There is/ are [no bathroom/ many bathrooms/ three bathrooms/ a bathroom/ *the bathroom/ *every bathroom] in this house.'

As indicated by the data in (375), Romanian n-words can appear both in contexts that favor a strong reading (375a) and in those that favor a weak reading (375b). In the first case they pattern with universal quantifiers, in the second with existential quantifiers. Going back to the dynamic binding data in (373) we can conclude that the n-word in (373a) can bind the anaphor *o* because it is in an existential context and receives a weak reading. This suggests that in a context that requires a strong NP an n-word should not be able to bind an anaphor. This prediction is borne out as indicated by (376):

(376) a. * Ori **niciun** student$_i$ din grupa asta **nu** e inteligent, ori l$_i$-am buimăcit
either no student in group this NM is intelligent, either him-have confused
cu exemplele mele întortocheate.
with examples mine crooked
'Either [no student]$_i$ in this group is intelligent, or I confused [him/ her]$_i$ with my crooked examples.'

b. * Ori **niciun** cîine$_i$ de pe strada asta **nu** mai latră, ori l$_i$-au alungat
either no dog in street this NM more barks either it-have scared-away
tunetele.
thunders.
'Either [no dog]$_i$ in this street barks anymore, or the thunders scared it$_i$ away.'

Romanian n-words as negative quantifiers If n-words as negative quantifiers are also weak quantifiers, we can now explain their behavior with respect to dynamic binding. The dynamic binding data are only compatible with n-words being negative quantifiers or ambiguous between universal and existential quantifiers, as in the case of Greek emphatic and non-emphatic n-words. In Romanian no independent distinction can be made that would correspond to the emphatic vs. non-emphatic contrast in Greek, so their behavior can only be related to their negative quantifier status.

The tests we discussed before: locality, *almost*-modification and topicalization in (350c), (350d), (350e) are compatible with an analysis of Romanian n-words as negative quantifiers. (350f) should be modified, since even English n-words, which are negative quantifiers in Giannakidou's view, can bind pronouns in existential contexts (see (372) above) and fail to do so in 'strong' contexts like (377):

(377) * Either [*no* dog]$_i$ in that street barks at all, or *it*$_i$ is very quiet.

Predicate nominals Let us now concentrate on the other tests in Giannakidou's classification: the usage as predicate nominals and negative content.

In principle, the occurrence of n-words in a predicative position indicates their existential quantifier status. But Giannakidou shows that even n-words that clearly behave like existential quantifiers are sometimes ungrammatical as predicate nominals. She concludes that this test has more to do with the way predicate nominals can be expressed in a language than with the semantic status of n-words in that language.

Giannakidou argues that n-words in some Romance languages must be ambiguous between existential quantifiers and negative quantifiers, since they occur in contexts without a negative marker and get an NPI interpretation: see (378a) and (378b-i):

(378) a. È venuto **nessuno**?
is come nobody
'Has anyone come?' (Italian)

b. Est-ce que tu as vu **personne**?
is-it that you have seen nobody
i. 'Did you see anybody?'
ii. 'Is it true that you saw nobody?' (French)

The data above lead us to expect that Italian and French n-words should also be grammatical as predicate nominals. But despite their ability to act as existential quantifiers, they still are excluded in predicative contexts ((379a), (379b)). In Romanian, they are typically ungrammatical (379c), but see Section 6.1.3 for more discussion:

(379) a. * **Non** è **nessun** dottore.
NM is no doctor (Italian)

b. * Il n'est **aucun** docteur.
he NM'is no doctor (French)

c. * Maria **nu** e **niciun** doctor.
Maria NM is no doctor (Romanian)

While the grammaticality of n-words as predicate nominals should indicate their existential quantifier status, they may be ungrammatical even in languages where there is independent evidence for n-words being existential quantifiers (378). Thus I conclude with Giannakidou that this test is irrelevant for the semantic status of n-words.

Negative content The properties in (350a) and (350b) repeated in (380) below are taken by Giannakidou to indicate the negative quantifier status of n-words. They are formulated to describe the negative spread data in non-strict NC languages (see also Section 3.1.2). The Italian and Spanish data in (344) slightly modified in (381) below show that n-words in these languages contribute negation alone, exclude the presence of the NM when they appear in preverbal position[6], and can license other n-words.

(380) Negative n-words
a. receive negative meaning and exclude sentential negation in the preverbal position;
b. receive negative meaning and exclude sentential negation when they cooccur with another n-word (negative spread); the first n-word is usually in preverbal position.

(381) a. **Nessuno** (*non) ha letto **niente**/ il libro.
nobody (NM) has read nothing/ the book
'Nobody read anything/ the book.' (Italian)

b. **Nadie** (*no) dijo **nada**/ eso.
nobody (NM) said nothing/ this
'Nobody said anything/ this.' (Spanish)

French and Portuguese n-words behave similarly, so in Giannakidou's classification they are negative quantifiers in these contexts. Italian, French and Spanish also use n-words in typical NPI contexts of the kind in (378) and Giannakidou argues that they are existential quantifiers in these constructions. By contrast, Portuguese uses a special paradigm of NPIs:

[6]Zanuttini (1991, Ch. 4.3.1 & p. 151) argues for Italian that the NM is not always excluded and may trigger a DN reading, especially with topicalized n-words.

(382) a. Telefonou [***ninguém**/ **alguém**]?
'Did you call anybody?'

b. Se vem [***ninguém**/ **alguém**] estamos perdidos.
'If anybody comes, we are lost.'

The data above lead Giannakidou to conclude that Portuguese n-words must be unambiguously negative quantifiers, so the NC constructions in this language should be accounted for by a mechanism similar to resumption.

Like Portuguese, Romanian also has a special paradigm of NPIs, so n-words cannot be used in contexts without sentential negation (see also Section 3.2.3):

(383) a. A sunat [***nimeni**/ **cineva**]?
has called nobody/ anybody
'Has anybody called?'

b. Dacă vine [***nimeni**/ **cineva**], sîntem pierduţi.
if comes nobody/ anybody are lost
'If anybody comes, we are lost.'

Giannakidou suggests that Romanian n-words should be universal quantifiers because they seem to behave like Greek emphatic n-words. As we saw, the dynamic binding data indicate that they are negative quantifiers, which is also compatible with the other tests we discussed. The only thing that prevents us from classifying Romanian n-words as negative quantifiers within Giannakidou's system is their obligatory cooccurence with the NM in finite sentences. Even in preverbal position, Romanian n-words require the NM on the finite verb (384):

(384) **Niciun** student *(**nu**) a citit **nicio** carte.
no student NM has read no book
'No student read any book.'

But despite the strict-NC character of Romanian, I showed in Section 3.4.1 that there are contexts where n-words appear alone and express negation: fragmentary answers, gapping, comparative, and past participial constructions. The relevant examples are repeated below:

(385) *Fragmentary answers*:

a. Speaker A: Ce a cumpărat?
what has bought
'What did he buy?'

b. Speaker B: **Nimic**.
nothing
'Nothing.'

(386) *Gapping*:

Maria *tot* *mai* citeşte, dar Ion (**niciodată**) **nimic**.
Maria still still reads, but John never nothing

'Maria still reads, but John never does.'

(387) *Comparative constructions*:

Ion e înalt ca **nimeni** altul de la el din clasă.
John is tall like nobody else from him from class

'John is taller than everybody else in his class. (Nobody in John's class is as tall as he is.)'

(388) *Past participial constructions*:

Acest articol, de **nimeni** citat, a rămas uitat.
this article by nobody cited has remained forgotten

'This article, which hasn't been cited by anybody, was forgotten.'

Moreoveor, in Section 3.4.2 I showed that two cooccurring n-words in Romanian can yield DN readings in denial contexts (389) and in some constructions that make the the NC reading pragmatically strange ((390), Fălăuş (2007)). N-words in fragmentary answers to negative questions are interpreted as DN as well ((391), Section 3.4.1). The examples are repeated below:

(389) *Denial*:

a. Speaker A: Un student nu a citit **nicio carte**.
one/a student NM has read no book

'One/A student read no book.'

b. Speaker B: **NICIun stuDENT** nu a citit **nicio carte**.
no student NM has read no book

'No student read no book. (= Every student read some book.)'

(390) *Pragmatically strange NC reading*:

Nimeni nu moare **niciodată**.
nobody NM dies never

a. # Nobody ever dies. (NC)

b. Nobody never dies. (Everybody dies one day.) (DN)

(391) *Fragmentary answers to negative questions*:

a. Speaker A: Ce **nu** a cumpărat?
what NM has bought

'What didn't he buy?'

b. Speaker B: **Nimic** (a cumpărat tot).
nothing (has bought everything)

'Nothing (he bought everything).'

True indicators of the negative content The data in (385) – (391) clearly show that Romanian n-words carry negation. They do not match the description in (380a) and (380b), because it is formulated to accommodate n-words in non-strict NC languages as negative quantifiers. In Giannakidou's view 'sentential negation' (the NM in our terms) is the only contributor of negative meaning in strict NC languages like Romanian. However, we saw in Section 5.5.4 that the Romanian NM does not contribute negation in NC constructions. Besides, even proponents of NPI approaches to NC have argued that the NM in non-strict NC languages differs from the one in strict NC languages precisely in contributing negation in NC constructions. In Zeijlstra (2004, Ch. 8), for instance, the NM in Italian is argued to have an 'interpretable' Neg feature, in contrast to the Romanian NM, which has an 'uninterpretable' Neg feature.

Thus the way Giannakidou (2006) describes the negative content of negative quantifiers is too narrow and excludes the possibility of n-words being negative in some strict NC languages as well. To overcome this drawback, I replace the two descriptions in (380) with the ones in (392). In contrast to the original ones, they characterize the negative content of n-words independently of the language.

(392) a. can express negation alone;

b. can yield DN in the presence of another expressor of negation.

In conclusion, I have shown that the assumption in this thesis that n-words are negative quantifiers is compatible with the inventory of properties listed in Giannakidou (2006), if we take into account that n-words as negative quantifiers exhibit both 'weak' and 'strong' quantifier properties (Milsark (1974)), and if we consider the properties in (392) to appropriately describe the negative semantics of n-words crosslinguistically.

6.1.2 NPI approaches to Romanian n-words

Earlier accounts of Romanian NC take it for granted that n-words are negative quantifiers (Isac (1998)) or existential quantifiers (Ionescu (1999)). The debate on the semantic status of n-words in Romanian is recent (Barbu (2003), Ionescu (2004), and Isac (2004)). Barbu (2003) and Isac (2004) argue on independent grounds that n-words are indefinites, while Ionescu (2004) claims that they are existential quantifiers. Most of the tests that are used in these approaches follow the ones collected in Richter and Sailer (1999*b*) and Giannakidou (2006) and have already been addressed here. I showed that they are compatible with the treatment of n-words as negative quantifiers.

In this section I first discuss the doubts that Ionescu (2004) raises with respect to the semantic status of the NM in an analysis of NC as resumption. Then, I address Isac's (2004) arguments in support of the claim that Romanian n-words lack quantificational force.

Ionescu (2004) follows the NC analysis for Polish in Przepiórkowski and Kupść (1999) and proposes an account of Romanian NC where n-words are existential quantifiers. He admits that data like (387) and (388) indicate that n-words can also be negative quantifiers, but chooses not to apply the NC analysis in de Swart and Sag (2002) to Romanian NC for reasons that have to do with the semantic contribution of the NM.

Let us first summarize the main points of the present analysis of the NM and then address the comments in Ionescu (2004). In Section 5.5 I argued that the NM does not contribute negation in the presence of n-words, as it does not trigger DN readings (393a). I also showed that the NM does carry negation, since it contributes sentential negation in the absence of n-words (393b) and it also licenses NPIs of medium strength (like *prea*) and disallows PPIs (like *cam*) in (393c).

(393) a. *Semantic absorption with n-words*

Niciun student **nu** a venit.
no student NM has come

i. 'No student came.' (NC)
ii. # 'No student didn't come.' (DN)

b. *Sentential negation*
Un student **nu** a venit.
a student NM has come
'Some student didn't come.'

c. *Licensing of strong NPIs*
A început să **nu** *prea*/ **cam* (mai) tragă chiulul.
has started SJ NM really/ pretty anymore skip classes
'He started pretty much not to skip classes (anymore).'

To account for the negative content of the NM, I followed de Swart and Sag (2002) and assumed that it is a type $\langle 0 \rangle$ negative quantifier (the type of propositional operators in Lindström's (1966) classification). In NC constructions, this quantifier undergoes resumption with the other type $\langle 1, 1 \rangle$ negative quantifiers, as they all carry the same operator NO. To account for the lack of DN readings with n-words, I introduced the NEG CRITERION for Romanian which excludes the cooccurrence of a $\langle 0 \rangle$ negative quantifier with another negative quantifier in the logical representation of an utterance.

This analysis suggests that the NM contributes nothing to the complexity of the resumptive quantifier that is built by n-words: a sentence with two n-words and a NM builds a type $\langle 1, 1, 2 \rangle$ negative quantifier, just like a sentence with two n-words and no NM. Our analysis, however, enforces the NM to always cooccur with n-words for syntactic reasons, thus accounting for its obligatoriness in NC constructions in Romanian, a strict NC language.

Ionescu (2004, pp. 92–93) argues that by considering n-words to be negative quantifiers one is led to conclude that "in NC environments, the negative marker loses its semantic function and becomes *expletive*". This cannot be right, since clear instances of expletive negation in Romanian are incompatible with n-words.

As we have seen above, the analysis I propose here does not require treating the NM as losing its semantic negation. Expletive negation I assume is a different use of *nu* (*nu*$_{expl}$ below) that is most likely triggered by the specific lexical items that require *nu*$_{expl}$'s insertion: for instance verbs like *a se teme* 'to fear' or uses of *pînă* 'until' exemplified in (394):

(394) a. Mă tem să **nu**$_{expl}$ mă vadă **vreunul**/ ***nimeni**.
me fear SJ not$_{expl}$ me see anyone/ *nobody
'I fear that somebody might see me.'

b. Să plecăm pînă **nu**$_{expl}$ ne prinde **vreunul**/ ***nimeni**.
SJ leave until not$_{expl}$ us catches anybody/ nobody
'Let's go before somebody catches us.'

It is true that weak NPIs like *vreunul* 'anyone' in (394) can be licensed in these contexts, but it is the lexical item requiring expletive negation that licenses these NPIs: note for instance that weak NPIs are also licensed in parallel contexts without expletive negation (395). Moreover, stronger NPIs like *prea* cannot be licensed either with (396) or without *nu*$_{expl}$ (395):

(395) a. Mă tem că mă (**prea*) vede **vreunul**.
me fear that me (*really) sees anyone
'I fear that somebody might see me.'

b. Să plecăm pînă să ne (**prea*) prindă **vreunul**.
SJ leave until SJ us (*really) catches anybody/ nobody
'Let's go before somebody catches us.'

(396) a. Mă tem să **nu**$_{expl}$ mă (**prea*) vadă **vreunul**.
me fear SJ not$_{expl}$ me (*really) see anyone
'I fear that somebody might see me.'

b. Să plecăm pînă **nu**$_{expl}$ ne (**prea*) prinde **vreunul**.
SJ leave until not$_{expl}$ us (*really) catches anybody
'Let's go before somebody catches us.'

The data above clearly show that the NM *nu* is semantically different from expletive *nu*, as it has negative content, unlike the latter. Thus the present analysis of NC does not predict that the NM *nu* is semantically similar to *nu*$_{expl}$.

Isac (2004) We now turn to the NPI analysis of n-words in Isac (2004). Isac's goal is to account for the contrast between Romanian and other Romance languages that correlates with the strict NC vs. non-strict NC language distinction, exemplified above and repeated in (397) for convenience:

(397) a. **Nessuno** (*non) ha letto **niente**/ il libro.
nobody (NM) has read nothing/ the book
'Nobody read anything/ the book.' (Italian)

b. **Nadie** (*no) dijo **nada**/ eso.
nobody (NM) said nothing/ this
'Nobody said anything/ this.' (Spanish)

c. **Niciun** student *(**nu**) a citit **nicio** carte/ cartea.
no student NM has read no book/ book-the
'No student read any book/ the book.' (Romanian)

Isac starts with the assumption that true negative quantifiers have both a [neg](ative) and a [qu](antificational) feature (so they are [+neg,+qu]), NPIs are [-neg,-qu], non-negative quantifiers are [-neg,+qu] and n-words are indefinites specified as [+neg,-qu]. Only a [qu] feature can trigger (quantifier) raising to a position from where [neg] can take sentential scope (the case of negative quantifiers in DN languages). Since n-words in Romance lack a [qu] feature, they do not raise and the [neg] feature cannot take sentential scope. Isac (2004) argues that the preverbal n-words in (397) are in a syntactic Focus position where they also acquire the quantificational feature of Focus.

The difference between the two groups of languages is claimed to lie in the way the [qu] feature of Focus and the [neg] feature are realized. In languages like Spanish and Italian, both features appear on the head of FocusP as [pol](arity) and [foc](us) features. Importantly, the NM in these languages is merged under Focus, checks the [pol] feature as negative and sometimes also the [foc] feature if this is not checked by non-negative focused constituents which raise to Spec FocusP. When an n-word raises to Spec FocusP, it obligatorily checks both the [pol] feature as negative and the [foc] feature, so merging the NM with the [neg] feature becomes superfluous and yields ungrammaticality. For Romanian Isac (2004) argues that the NM with the [neg] feature is realized as the head of a PolarityP, while the [foc] feature appears on the head of FocusP. Given the two independent projections, the NM does not check the [foc] feature, which can thus be checked by a preverbal n-word.

This approach relies heavily on theoretical claims independent of negation and negative concord, so I will not go into a detailed discussion of its pros and cons. I concentrate on the claim it makes with respect to what it means for an n-word to be a negative quantifier and why Romanian n-words cannot be negative quantifiers. In what follows I will first show that the so-called quantificational feature attributed by focus cannot be made responsible for the negative quantifier status of n-words. Then I will show that the tests that Isac uses to argue for the lack of a quantificational feature in n-words are inconclusive, since other quantifiers in Romanian, and negative quantifiers in English exhibit a similar behavior.

First, Isac assumes that what prevents n-words from behaving like negative quantifiers is the lack of a [qu] feature. It then follows that two n-words that carry a [qu] feature (due to a particular context) should trigger DN readings like true negative quantifiers. An argument in support of this is provided in Isac (2004) on the basis of (398). This sentence is argued to receive a DN reading "only if both n-words are under stress". Whatever 'stress' is taken to be, it is unlikely to be the same as accent, since I argued in Sections 3.4.2 and 5.4.2 that one of the two n-words receiving a DN interpretation is in Focus while the other counts as background and is deaccented. In Isac (2004) 'stress' is said to attribute a quantificational status to n-words, so it includes Focus. However, it remains unclear how the deaccented n-word becomes quantificational and how the assumed notion of 'stress' can achieve this.

(398) **Nimeni** nu iubeşte pe **nimeni**.
nobody NM loves PE nobody
'Nobody loves anybody.' (NC)
'Nobody loves nobody. (Everybody loves somebody.)' (DN)

Thus, this argument only goes through if one posits a [qu] feature of deaccented material provided by the context. For now, we do not have any independent support for this.

Moreover, n-words carrying Focus are taken in Isac (2004) to implicitly carry a [foc]/[qu] feature. In her view, n-words in fragmentary answers are focused so they carry this feature. This leads us to expect that two n-words in a fragmentary answer should have a DN reading, as both of them carry both a [neg] and a [foc] feature. But (399) indicates that this is contrary to fact.

(399) A: Cine ce a citit?
who what has read
'Who read what?'

B: **Nimeni nimic**.
nobody nothing
'Nobody read anything.' (NC)
'Nobody read nothing.' (DN)

The data in (399) raise doubts as to the determinative role of Focus in the quantificational behavior of n-words, the thesis advanced in Isac (2004).

Another argument that Isac (2004) uses to support her claim concerns the apparent non-uniform behavior of preverbal and postverbal n-words in comparison to bona fide quantifiers. Isac argues that preverbal n-words are quantificational because of their Focus position, while the postverbal ones are non-quantificational. This is claimed to be indicated by the possibility of the preverbal n-word to take wide scope over the quantifier *mai mult de doi* in (400a), and the impossibility of the postverbal n-word to take wide scope over the quantifier *cel puţin doi* in (400b):

(400) a. **Niciun** copil n-a văzut *mai mult de doi* hoţi.
no child NM-has seen more than two thieves
NO $>$ MORE THAN TWO: 'No child saw more than two thieves.'

b. *Cel puţin doi* copii n-au văzut **niciun** hoţ.
at least two children NM-have seen no thief
AT LEAST TWO $>$ NO: 'At least two children saw no thief.'

Isac (2004) argues that (400b) only has one scope interpretation, but remains silent about whether a wide scope reading for *mai mult de doi* is available in (400a).

Recall that quantifier scope in Romanian is greatly influenced by the linear order of the quantifiers (Section 3.5), so the scope preference in (400) is expected. In Isac's reasoning, *mai mult de doi* is a true quantifier so it should have quantificational force even in the object position and thus easily take wide scope over the preverbal n-word. According to my intuitions, a wide scope reading is slightly easier to obtain for *mai mult de doi* in (400a) than for the postverbal n-word *niciun* in (400b).[7] But this does not indicate that the n-word in postverbal position is less quantificational than a non-negative quantifier, as one would expect in Isac's analysis, since the same scope preference can be observed in English, a DN language where n-words are assumed in Isac (2004) to always be negative quantifiers:

(401) a. **No** child saw *more than two* thieves.
i. NO $>$ MORE THAN TWO
ii. ? MORE THAN TWO $>$ NO

b. *At least two* children saw **no** thief.
i. AT LEAST TWO $>$ NO
ii. ?? NO $>$ AT LEAST TWO

I conclude here that the evidence for an account in terms of a quantificational feature making n-words behave like negative quantifiers is not decisive. It would be if there were an explanation for the way the deaccented n-word in (398) can receive such a feature and for the lack of DN in (399). Moreover, the fact that wide scope readings for postverbal n-words are harder to obtain does not indicate that Romanian n-words cannot be negative quantifiers; negative quantifiers in English exhibit a similar behavior.

6.1.3 Split scope readings of n-words

Next I address the split scope readings of n-words that Penka (2007) takes to be crucial evidence for the indefinite status of n-words even in DN languages like German. Such readings are sometimes also available for Romanian n-words. In this section we will see that in some contexts cardinal quantifiers exhibit split readings as well. This suggests that the split readings of n-words are one instance of a more general phenomenon. An account for this phenomenon would also cover negative quantifiers, thus one wouldn't need to assume that n-words are indefinites.

The German data Split scope readings of n-words have been discussed for German and Dutch in Bech (1955/1957), Jacobs (1980, 1991), Geurts (1996), de Swart (2000) and Penka and Stechow (2001). For sentence (402) Penka (2007) gives the three possible interpretations below:

(402) Bei der Prüfung muss **kein** Professor anwesend sein.
at the exam must no professor present be

a. $\neg >$ MUST $> \exists$: 'It is not required that there be a professor present.' (split scope)

b. $\neg\exists >$ MUST: 'There is no professor who is required to be present.' (de re)

[7] See also the discussion in Section 3.5.2.

c. ?? MUST $>$ $\neg\exists$: 'It is required that there be no professor present.' (de dicto)

The split scope interpretation is said to be the most natural one of the three, while the de dicto reading is the least available one.

For the following, we assume a representation of a negative quantifier as an existential outscoped by the negative operator ($\neg\exists$) instead of the special operator NO employed in this thesis. This allows a clearer representation for the split scope reading. The de re reading is normally obtained if the negative quantifier takes scope over the modal operator, the de dicto one if the negative quantifier takes narrow scope with respect to the modal. As Penka (2007, pp. 87–88) shows, under the assumption that the negative operator and the existential quantifier make up a unit, there is no way to derive the split scope reading where the negation takes wide scope over the modal, and the existential quantifier is outscoped by the modal. For this reason, Penka argues that the meaning of *kein* cannot always be that of a negative quantifier.

The solution she proposes is to treat *kein* as a 'free variable' (i.e. Heimian) indefinite that has to be syntactically licensed by an abstract operator that contributes the semantic negation. This operator can adjoin to the VP-level of the embedded verb allowing for the de dicto interpretation, or to the VP-level of the modal for the de re and split readings. In the split scope reading the indefinite *kein* is existentially bound by the modal which thus intervenes between the negation and the existential quantifier as required. The approach in Penka (2007) actually extends Zeijlstra's (2004) analysis of n-words in NC languages to n-words in DN languages. She claims that n-words are crosslinguistically indefinites and natural language does not have any lexical items instantiating negative quantifiers.

Several other contexts have been shown to exhibit split scope readings: opaque verbs (403a), predicative contexts (403b), topic-focus accent constructions (403c), and idiomatic expressions (403d):

(403) a. Peter sucht **kein** Einhorn.
Peter seeks no unicorn
i. $\neg > \text{SEEK} > \exists$: 'Peter doesn't try to find a unicorn.' (split scope)
ii. $\neg\exists > \text{SEEK}$: 'There is no unicorn Peter tries to find.' (de re)

b. Jim wurde **kein** Rockstar.
Jim became no rock-star
'Jim didn't become a rock-star.'

c. ALLE$_{top}$ Ärzte haben **KEIN**$_{foc}$ Auto.
all doctors have no car
'It is not the case that all doctors have a car.'

d. Peter hat **keine** Schraube locker.
Peter has no screw loose
'Peter doesn't have a screw loose.'

For the following discussion I concentrate on contexts with modals, opaque verbs and predicative n-words. For the topic-focus accent constructions we need a theory of information structure which would take us too far afield. The idiomatic expressions cannot make a case for the syntax-semantics of n-words, as they might receive a special lexical entry as a whole.

Split scope readings with Romanian n-words Split readings of n-words are found not only in DN languages, but also in NC languages like Romanian.[8] In what follows I will show that Romanian

[8] Split scope readings of Polish n-words are discussed in Richter and Sailer (1999*b*).

has other standard means to express the interpretations for which German uses split readings. As a consequence, these readings are colloquial and contextually restricted.

Split readings of Romanian n-words can be found in contexts similar to those in German:[9]

(404) La examen **nu** trebuie să fie **niciun** profesor prezent.
at exam NM must SJ be no professor present

a. $\neg\exists >$ MUST: 'No (particular) professor must be present. (de re)

b. MUST $> \neg\exists$: 'It is required that no professor be present.' (de dicto)

c. ? $\neg >$ MUST $> \exists$: 'It is not required that there be a professor present.' (split scope)

(405) Ion **nu** caută **nicio** secretară.
John NM seeks no secretary

a. $\neg\exists >$ SEEK: 'No (particular) secretary is such that John seeks her.' (de re)

b. # SEEK $> \neg\exists$: 'John is trying to not find a secretary.' (de dicto)

c. ? $\neg >$ SEEK $> \exists$: 'John is not trying to find a secretary.' (split scope)

(406) ?? Ion **nu** a ajuns **niciun** doctor.
John NM has become no doctor
'John didn't become a doctor.'

For the Romanian sentence in (404) the de re and de dicto readings are equally available[10], while for (405) the de dicto reading is excluded. In both cases the split scope reading is informal and usually appears in colloquial speech.[11]

Split readings seem to involve a property interpretation required by the context in which they appear (Penka (2007, Ch. 3)). Romanian n-words do not easily express properties. Sentence (406) with an n-word in predicative position is highly marked precisely because this position requires a property (Partee (1987)). The most natural context where split readings of Romanian n-words appear is that of denial ((407), (408), Section 3.4.2) or contrastive negation ((409), McCawley (1991)):

(407) A: La examen trebuie să fie un profesor prezent.
at exam must SJ be a professor present
MUST $> \exists$: 'It is required that a professor be present.' (de dicto)

B: Vorbeşti prostii. La examen **nu** trebuie să fie **niciun** profesor prezent.
speak nonsense at exam NM must SJ be no professor present
'You're speaking nonsense. It is not required that a professor be present.' (split scope)

(408) A: Am înţeles că Ion caută (o) secretară.
have understood that John seeks (a) secretary
SEEK $> \exists$: 'I hear that John is trying to find a secretary.' (de dicto)

B: Nu e adevărat. Ion **nu** caută **nicio** secretară.
NM is true John NM seeks no secretary

[9]I represent negative quantifiers as existentials outscoped by the negative operator, to allow a clear notational distinction between the de re and the split scope reading.

[10]The modal *trebuie* 'must' also acts like a Neg-raising verb in Romanian which explains why the de dicto reading is fully natural.

[11]A slight dialectal difference may also be at play. Linguists from the south tend to allow these readings more easily than the ones from north-east. But colloquial speech uses them just as frequently.

'That's not true. John is not trying to find a secretary.' (split scope)

(409) A: Am înţeles că Ion a ajuns doctor.
have understood that John has become doctor
'I hear that John has become a doctor.'

B: N-a ajuns **niciun** doctor (e un simplu asistent).
NM-has become no doctor (is a simple assistant)
'He hasn't become a doctor (he's a simple medical assistant).'

To express the denial of the de dicto reading of an indefinite in the scope of a modal (407) or an opaque verb (408), an n-word can replace the indefinite thus yielding the split scope reading. In similar contexts, an n-word can also appear in a predicative position (409). However, the natural way to express these readings in Romanian is by employing an indefinite or a bare noun under negation:

(410) a. La examen **nu** trebuie să fie *un* profesor/ profesori prezent(i).
at exam NM must SJ be a professor/ professors present
$\neg > \text{MUST} > \exists$: 'It is not required that a professor/ professors be present.

b. Ion **nu** caută (*o*) secretară.
John NM seeks (a) secretary
$\neg > \text{SEEK} > \exists$: 'John is not trying to find a secretary.'

c. **N**-a ajuns (*un) doctor (e un simplu asistent).
NM-has become (a) doctor (is a simple assistant)
'He hasn't become a doctor (he's a simple medical assistant).'

The sentences in (410) can also appear in denial contexts like (407) – (409), but unlike the latter, they are not restricted to denial. They can also be used to neutrally convey the negation of a de dicto reading.

The existence of split readings in Romanian, a strict NC language, may be a good argument to analyze n-words as indefinites and the NM as the only contributor of negation. However, as we saw in Section 5.5.1, the NM does not contribute independent negation in NC. Moreover, if n-words were pure indefinites in the split reading contexts, the sentences in (407) – (409) should be fully equivalent to the ones in (410), which is not the case. By contrast, the use of split scope *kein* is the only way to express the negation of a de dicto reading under a modal in German: pure indefinites are usually disallowed to cooccur with sentential negation *nicht* (see Kratzer (1995, pp. 144–147)). This is an important factor in determining the split readings of *kein* in German.

A related phenomenon In what follows I present some observations that cast doubt on taking split scope readings of n-words as evidence for a general treatment of n-words as indefinites. I mentioned before that the contexts where these readings occur in German require a property interpretation for the NP. This suggests that other quantifiers may also exhibit split scope readings in such contexts, i.e. the property would be interpreted in situ, while the quantificational operator would be interpreted across an intervening operator. In this section I will show that the so-called 'event readings' of cardinal quantifiers (Krifka (1990) and Doetjes and Honcoop (1997)) with modals also require a split scope interpretation of the quantifier.

Krifka (1990) observes that cardinal quantifiers can sometimes quantify over the number of events rather than the number of objects/ individuals involved in the event. He uses (411) to illustrate this:

(411) *Four thousand* ships passed through the lock last year.

a. 4000 ships are such that each of them passed through the lock. (object reading)

b. There were 4000 events in which a ship passed through the lock. (event reading)

If reading (411a) is true, (411b) is true as well. But the contexts in which (411b) is true are not always contexts in which (411a) is true. For example, if a ship passed through the lock more than once last year, it is still true that there were four thousand different events (411b), but not that there were four thousand different ships (411a).

The event reading of cardinal quantifiers can be observed in Romanian sentences as well. From the perspective of a person who guards the lock, one can easily understand sentence (412) as (412b):[12]

(412) *Patru mii* de vapoare au trecut prin ecluză anul trecut.
four thousands of ships have passed through lock year last

a. 4000 ships are such that each of them passed through the lock. (object reading)

b. There were 4000 events in which a ship passed through the lock. (event reading)

In sentences with a modal verb, one can obtain an event reading with the modal where the cardinal quantifier counts the number of situations in which the modality holds. In this case, we obtain a reading similar to the split scope reading of negative quantifiers, as the cardinal quantifier is understood as split between the cardinal operator and an indefinite.

Imagine the following scenario: John is a personnel recruiter and interviews applicants for various companies. He may interview one and the same applicant more than once (for different companies or different jobs). He has a certain number of interviews he does per day, but every now and then an emergency occurs. He never does an emergency interview unless he really has to (e.g. something about an obligatory interview requires doing an emergency interview first). In this context, we can understand sentence (413a) with a split scope reading where there are four hundred situations in which John had to additionally interview somebody besides his normal amount of work. Similarly, we can understand (413b) in a scenario where John was assigned to recruit a secretary forty times.

(413) a. Ion a trebuit sa intervieveze *patru sute* de aplicanţi peste normă anul trecut.
John has must-ed SJ interview four hundreds of applicants over quota year last

$400 > \text{MUST} > \exists$: 'There were 400 times when John had to additionally interview an applicant last year.'

b. Ion a căutat patruzeci de secretare anul trecut.
John has sought forty of secretaries year last

$40 > \text{SEEK} > \exists$: 'There were forty times when John tried to find a secretary last year.'

[12] The postverbal position of the cardinal quantifier makes the event reading more natural than the object reading.

The data in (413)[13] indicate that we need a theory of quantifier scope that allows the operator to be interpreted higher than the restriction of a quantifier.[14] If we adopt the treatment of negative quantifiers with split readings as indefinites licensed by an abstract negative operator (Penka (2007)), we have to assume a similar mechanism for split readings of cardinal quantifiers as well. This may ultimately require an infinite inventory of abstract quantificational operators, an undesirable consequence for linguistic theory.

Split scope in LRS The split scope readings of negative quantifiers should be understood in the larger context of what kinds of quantifiers can split their scope and when. Doetjes and Honcoop (1997) argue that only weak quantifiers in their weak reading receive the event reading discussed above. Our assumption that n-words are negative quantifiers is fully compatible with this idea.

LRS is well-suited to account for split readings in general, as it employs discontinuous semantic representations that allow flexibility in operator scope interaction. Richter and Sailer (2004) give an account of split scope readings of n-words in Polish where a modal takes scope between the negative operator and the existential quantifier that make up the negative quantifier expressed by the n-word. Similarly, Richter and Sailer (2008) offer an account of epistemic modals that take scope between negation and universal quantifiers like 'not every'.

The analysis of Romanian NC proposed in this thesis does not employ an intensional language, so it is hard to envision a solution for the split scope readings. Moreover, I represented negative quantifiers by the special operator NO which at first sight disallows a split between the negative operator and an existential quantifier (or a property). But one could allow NO to be separated from the property contributed by the common noun if, instead of enforcing the latter to be a subexpression of the restriction of the quantifier (SEMANTICS PRINCIPLE, clause 1, p. 169), we also allow it to be a subterm of the nuclear scope of the quantifier.[15] This way the restriction list of a quantifier could be empty and the NP quantifier could be identified with the type $\langle 0 \rangle$ NO contributed by the NM. A new representation of negative quantifiers in LRS would also be needed. They should not take a variable argument anymore, but only a restriction and a nuclear scope. If we want to split the quantificational operator from its restriction, the variable should appear with the restriction and not with the operator which acts like a propositional operator. Thus instead of $e^n(t^n(t(t)))$, quantifiers would be of type

[13] A first look at German indicates that split readings of cardinal quantifiers are available in this language as well (Florian Schäfer, p.c.):

(414) a. Hans musste im letzten Jahr 400 Kandidaten ausserhalb der Reihe interviewen.
John must-ed in last year 400 applicants additionally interview
$400 > \text{MUST} > \exists$: 'There were 400 times when John had to additionally interview an applicant last year.'

b. Hans hat 40 Sekretärinnen gesucht im letzten Jahr.
John has 40 secretaries sought in last year
$40 > \text{SEEK} > \exists$: 'There were forty times when John tried to find a secretary last year.'

[14] These readings could be viewed as the inverse of quantifier floating as Richard Larson (p.c.) remarks: the quantifier 'floats' in the semantics, but keeps its determiner position in the syntax.

[15] A somewhat similar point is made in Ebert et al. (2007) where it is shown that in German the proportional reading of some quantifiers maps a verb second restrictive relative clause in the nuclear scope instead of the restriction of the quantifier. So what appears as the restriction of a quantifier in the syntax is interpreted in the nuclear scope. From this viewpoint, the situation is similar to that of split readings, where the common noun of a quantificational NP is interpreted in the nuclear scope. The case described in Ebert et al. (2007) is, however, a slightly different issue since the matrix clause, which should be the nuclear scope, is interpreted as the restriction: that is, the nuclear scope and the restriction interchange. To account for split readings of n-words in our analysis, we need to free the quantifier of its restriction entirely.

$(et)^n((e^nt)t)$.[16]

In conclusion, an account for split readings of n-words is possible in the LRS analysis of Romanian NC here, but we first need a better understanding of the contexts where these readings occur and the implications they have for a theory of quantifier scope. This will be needed for a general and accurate formulation of the SEMANTICS PRINCIPLE in LRS.

6.1.4 Conclusion

In this section I discussed empirical issues that have been taken by NPI approaches as evidence against the negative quantifier status of n-words and an NQ approach to NC. I showed that none of these remains a challenge if one looks at the behavior of n-words from the wider perspective of how weak quantifiers behave in general. At the theoretical level, LRS is powerful enough to account for the split readings of n-words, which are hard to analyze in a compositional semantics.

6.2 The NQ approaches to NC

Having discussed the issues NPI approaches raise with respect to n-words and NC, we will now have a look at how NQ approaches other than the one in this thesis account for them. The challenge is to solve the compositionality problem that the negative quantifier status of n-words raises. This is not a trivial matter, so the NQ approaches are by far not as numerous as the NPI approaches. Among such accounts that also give a syntax-semantics for NC we can distinguish three groups: 1) those that make use of the NEG-Criterion (Haegeman and Zanuttini (1991, 1996), Zanuttini (1991), Haegeman (1995)), 2) those based on polyadic quantification (de Swart and Sag (2002), de Swart (2010)), and 3) those employing underspecification mechanisms (Richter and Sailer (2004)).

There is little to say about a comparison between the analysis of NC in this thesis and the one in Richter and Sailer (2004), which I presented in Section 5.3.2. Both make use of the possibility offered by HPSG to identify several negations in order to obtain the NC reading. The difference lies in the representation of negative quantifiers: while I use a polyadic NO quantifier, Richter and Sailer employ the traditional representation with a negative operator outscoping an existential quantifier. Identifying several negations results in identifying the entire NO quantifiers, or only the negative operators (cf. Section 5.3.2). The advantage that my analysis brings is that of opening the possibility to integrate polyadic quantifiers in LRS and thus accounting for other instances of polyadic quantification. For the analysis of NC itself, the results are similar.

In this section I discuss the central ideas of the approaches in 1) and 2) concerning the solution for the NC interpretation of a sentence. I first consider the NEG-Criterion approaches in Section 6.2.1 and then some issues related to de Swart and Sag (2002) in Section 6.2.2.

6.2.1 The NEG-Criterion

Zanuttini (1991) offers an NQ analysis for n-words in Italian based on the data discussed in Section 6.1.1. Following Pollock (1989), she assumes that sentential negation projects a NegP.[17] In negative sentences like (415) negative quantifiers raise to Spec NegP and enter a configuration of Spec-Head agreement with Neg^0 at LF. For non-negative contexts like (416), Zanuttini argues that it is the C head that hosts negative features and the negative quantifier thus moves to Spec CP at LF to enter a Spec-Head agreement relation with C^0:

[16] This type was assumed for negative quantifiers in our discussion in Chapter 4.

[17] Zanuttini (1991) actually argues for two possible NegPs, but this is irrelevant for our discussion.

(415) a. Mario **non** a visto **nessuno**.
Mario NM has seen noone
'Mario didn't see anybody.'

b. **Nessuno** a visto Mario.
noone has seen Mario
'Nobody saw Mario.'

c. **Non** a telefonato **nessuno**?
NM has called noone
'Hasn't anybody called?'

(416) a. Ha telefonato **nessuno**?
has called noone
'Has anybody called?'

b. Mi domando se verrà **nessuno**.
me ask if will-come noone
'I wonder whether anyone will come.'

The second clause of the NEG-Criterion in (417) (from (Haegeman and Zanuttini (1996))) ensures that negative quantifiers (i.e. 'Negative phrases') move to enter a Spec-Head configuration with the heads carrying negative features. This is the syntactic mechanism allowing for NC constructions.

(417) The NEG-Criterion

a. Each X^0[NEG] must be in a Spec-head configuration with a Negative phrase.

b. Each Negative phrase must be in a Spec-head configuration with an X^0[NEG].

On the basis of the *almost*-modification test discussed in Section 3.3.4 and Section 6.1.1, Zanuttini (1991) argues that negative quantifiers are universal quantifiers outscoping negation. The semantic mechanism by which a NC interpretation is obtained involves two operations: a process of *Absorption* and one of *Negation Factorization*.

Absorption was defined in Higginbotham and May (1981) for multiple wh-questions. In the case of negation, the universal quantifier component of two or more negative quantifiers undergoing Absorption result in one universal quantifier binding two or more variables.

The negative component of a negative quantifier goes through a process of Factorization, by which consecutive instances of negation following the universal quantifiers are factored out to convey a single negative operator (418a). More refined versions of the theory (Haegeman and Zanuttini (1996)) assume one more Factorization operation to also include the NM in combination with n-words ((418b):

(418) a. $[\forall x \neg][\forall y \neg] \rightarrow [\forall x, y]\neg$

b. $[\forall x \neg][\neg] \rightarrow [\forall x]\neg$

Issues raised by NEG-Criterion proposals There are three claims made by these proposals that have been subject to criticism in the literature: 1) the assumption that n-words are negative quantifiers (see for instance Déprez (1997), Penka (2006)), 2) the parallelism between NC and multiple Wh-questions supported by the similarity between the NEG-Criterion and the WH-Criterion of Rizzi (1991) (see Acquaviva (1997) and Giannakidou (1998)), and 3) the (non)compositionality of the Factorization operation (May (1989) and de Swart and Sag (2002)).

I have argued in favor of 1) at various points in this thesis, so I will not address it again. The NEG-Criterion is a theory-specific syntactic mechanism which is not relevant for a comparison to the analysis in this thesis. In particular, the GB/ Minimalist idea that negative sentences have a Neg functional projection is not adopted by constraint-based theories like HPSG. Recall that the NM in Romanian cannot contribute negation independently of a cooccurring n-word, which I formulated as the NEG-CRITERION for Romanian in (324), p. 208. This constraint, although surface-oriented, has effects comparable to those of the NEG-Criterion of Haegeman and Zanuttini.

The third point of criticism concerns the solution for the compositionality problem that the NEG-Criterion proposals offer. May (1989) argues that Factorization fails to respect compositionality, as parts of the semantic contribution of the elements involved in the operation are simply erased (see (418)). To solve this, May (1989) proposes to replace Absorption and Factorization by the resumption mechanism of polyadic quantifiers (van Benthem (1989), Keenan and Westerståhl (1997), Peters and Westerståhl (2006)). As I showed in Chapter 4, resumption is not compositional either, so it seems that lack of compositionality is the price to pay if we start with the assumption that several negative quantifiers can be interpreted as NC.

However, even though both the Absorption and Factorization mechanism in Zanuttini (1991) and Haegeman and Zanuttini (1991, 1996), and polyadic resumption assumed here are non-compositional, it does make a difference for linguistic theory whether we use one or the other. To see this difference, we have to consider why we need compositionality. Intuitively, the motivation for the principle of compositionality in linguistics is the necessity to provide a systematic mapping between the syntax and semantics of the parts of a complex expression in relation to the whole. We need an abstract mechanism by which we can derive the meaning of any complex expression from the meanings of its parts and in a way that is consistent with their syntax. For that, the mechanism must at least be mathematically precise and not make wrong predictions about the language.

The Absorption and Factorization mechanism has hardly received a precise formulation and has consequently also been given up by one of its first advocates (Higginbotham and May (1981) vs. May (1989)). By contrast, polyadic quantifiers are given a precise description in Lindström's (1966) mathematical classification of generalized quantifiers (see Section 2.1). Moreover, Keenan (1992, 1996), Keenan and Westerståhl (1997), and Peters and Westerståhl (2006) discuss several cases of natural language quantification which can only be accounted for as polyadic quantification. Polyadic quantifiers thus have the two minimal properties of the kind of mechanism that we need to describe natural language; however, it is only a semantic mechanism where nothing is said about the syntax. The advantage of LRS is that it provides us with a syntax-semantics interface where we can integrate polyadic quantifiers. LRS uses underspecified representations in close correspondence with the constituent structure of a surface-oriented syntax, by means of which we can identify the semantic contribution of several syntactic units into a resumptive quantifier. This mechanism allows a systematic syntax-semantics for polyadic quantifiers like the one provided for Romanian NC in Chapter 5.

6.2.2 A resumption-based alternative to LRS?

The treatment of NC as resumption of negative quantifiers is integrated in a syntax-semantics in de Swart and Sag (2002). Without going into details, let me briefly summarize their analysis of the ambiguous French sentence in (419) (see also Section 4.3.1). Following Pollard and Sag (1994), de Swart and Sag (2002) make use of Cooper storage (Cooper (1983)) to underspecify quantifier scope. In the HPSG syntax-semantics the two interpretations of (419) are obtained by lexical retrieval in the lexical entry of the verb *n'aime*. The DN reading is obtained by means of a quantifier retrieval operation called *iteration* (419b), NC by an operation called *resumption* (419d).

(419) **Personne** n'aime **personne**.
nobody NM-loves nobody

a. $It([\![\text{NO}]\!]^{[\![\text{PERSON}]\!]}, [\![\text{NO}]\!]^{[\![\text{PERSON}]\!]})([\![\text{LOVE}]\!]) = 1$
$\overset{D:2.8}{\Longleftrightarrow} [\![\text{NO}]\!]^{[\![\text{PERSON}]\!]} \circ [\![\text{NO}]\!]^{[\![\text{PERSON}]\!]} = 1$
$\overset{L:2.1}{\Longleftrightarrow} [\![\text{PERSON}]\!] \cap \{x \in \text{E} | [\![\text{PERSON}]\!] \cap \{y \in \text{E} | (x, y) \in [\![\text{LOVE}]\!]\} = \emptyset\} = \emptyset$ (DN)

b. DN in HPSG:

$$\begin{bmatrix} \text{PHON} & \left\langle \textit{n'aime} \right\rangle \\ \text{SS}|\text{LOC}|\text{CONT} & \begin{bmatrix} \text{QUANTS} & \left\langle \text{NO}_{\text{E}}^{\text{PERSON}}, \text{NO}_{\text{E}}^{\text{PERSON}} \right\rangle \\ \text{NUCL} & \text{LOVE} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \left[\text{STORE } \left\{\text{NO}_{\text{E}}^{\text{PERSON}}\right\}\right], \left[\text{STORE } \left\{\text{NO}_{\text{E}}^{\text{PERSON}}\right\}\right] \right\rangle \\ \text{STORE} & \{\} \end{bmatrix}$$

c. $Res^2([\![\text{NO}]\!])_{\text{E}}^{[\![\text{PERSON}]\!], [\![\text{PERSON}]\!]}([\![\text{LOVE}]\!]) = 1$
$\overset{D:2.16}{\Longleftrightarrow} \text{NO}_{\text{E}^2}^{[\![\text{PERSON}]\!] \times [\![\text{PERSON}]\!]}([\![\text{LOVE}]\!]) = 1$
$\overset{D:4.2}{\Longleftrightarrow} ([\![\text{PERSON}]\!] \times [\![\text{PERSON}]\!]) \cap [\![\text{LOVE}]\!] = \emptyset$ (NC)

d. NC in HPSG:

$$\begin{bmatrix} \text{PHON} & \left\langle \textit{n'aime} \right\rangle \\ \text{SS}|\text{LOC}|\text{CONT} & \begin{bmatrix} \text{QUANTS} & \left\langle \text{NO}_{\text{E}^2}^{\text{PERSON},\text{PERSON}} \right\rangle \\ \text{NUCL} & \text{LOVE} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \left[\text{STORE } \left\{\text{NO}_{\text{E}}^{\text{PERSON}}\right\}\right], \left[\text{STORE } \left\{\text{NO}_{\text{E}}^{\text{PERSON}}\right\}\right] \right\rangle \\ \text{STORE} & \{\} \end{bmatrix}$$

In Section 4.3.3 I showed that resumption and iteration as polyadic lifts defined in Keenan and Westerståhl (1997) cannot be given a compositional syntax-semantics with lambda-calculus and a functional type theory, the combinatorics usually assumed in compositional grammars. There I used a type shifting mechanism to derive the scope interaction between quantifiers. de Swart and Sag (2002) developed their analysis of NC as polyadic quantification by means of Cooper storage, which employs an underspecified representation of quantifier scope. Cooper storage allows flexible quantifier scope interaction, so one may now wonder whether a precise syntax-semantics with Cooper storage would allow us to integrate polyadic quantifiers in a compositional grammar.

In this section I will investigate this possibility and I will show that this is not possible: Cooper storage keeps the combinatorics with lambda calculus and functional types which, as I showed in Section 4.3.3, prevents us from formulating a syntax-semantics for resumption.

Cooper storage The 'storage' mechanism proposed in Cooper (1983) is designed to deal with quantifier scope ambiguities at the semantic level, independently of the syntax, and thus avoids supplementary grammar rules like 'Quantifying-in' necessary in Montague's (1973) approach (see Blackburn and Bos (2005, Ch. 3) for details).

Cooper associates each node of a syntactic tree with a 'store' containing a core semantic representation followed by all the quantifiers that appear on the lower nodes in the tree. At the sentence level the store of quantifiers is used to generate all the possible interpretations for that sentence. The

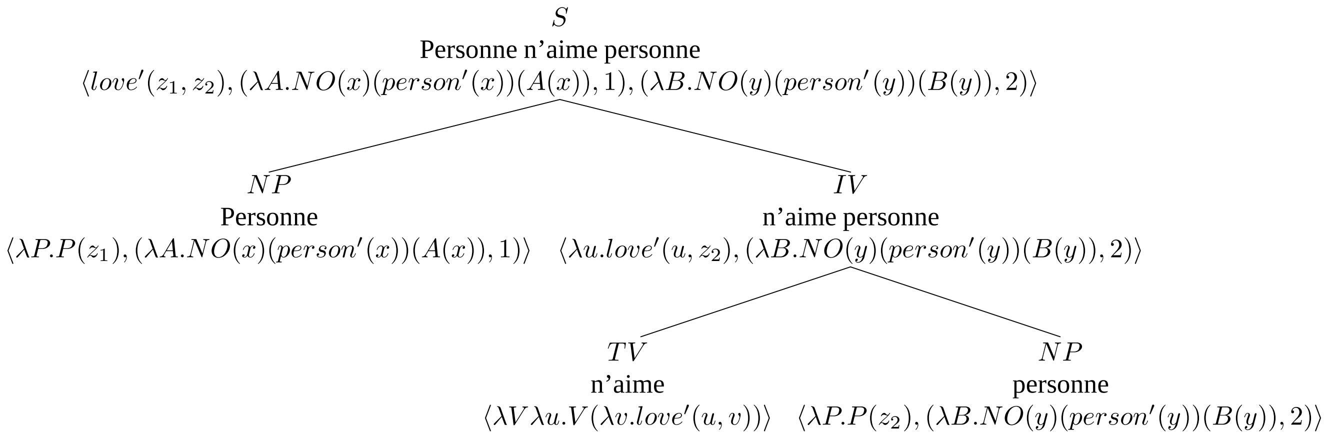


Figure 6.1: Syntactic tree with stores for sentence (419).

order in which the quantifiers are 'retrieved' from the store and combined with the core representation generates different scope possibilities between the quantifiers.

A store is an n-place sequence (within angle brackets) where the first item is a lambda expression giving the core semantic representation of a linguistic expression. Subsequent elements (if any) are pairs (β, i), where β is the semantic representation of a quantified NP and i is an index. The storage mechanism allows a quantified NP to store its semantic representation β with an index i and contribute the expression $\lambda P_{et}.P(z_{i,e})$ for the combinatorics of the sentence (DEFINITION 6.1). For the VP and the two NPs in sentence (419), we have the (storage) semantic representations in (420). I keep the discussion here within the limits of the $Ty1$ logical language defined in Section 5.1.

Definition 6.1 Cooper storage
For every $P \in Ty1_{et}$, $z \in Ty1_e$, $\phi, \beta, \beta' \in Ty1_{(et)t}$, $i, j, k \in \mathbb{N}^+$, if the store $\langle \phi, (\beta, j), (\beta', k)\rangle$ is a semantic representation for a quantified NP, then the store $\langle \lambda P.P(z_i), (\phi, i), (\beta, j), (\beta', k)\rangle$ is also a representation for that NP.

(420) a. (subject) *personne*$_{NP} \rightsquigarrow \lambda A_{et}.NO(x)(person'(x))(A(x))$
$\overset{D:6.1}{\Longleftrightarrow}$ (subject) *personne*$_{NP} \rightsquigarrow \langle \lambda P_{et}.P(z_{e,1}), (\lambda A_{et}.NO(x)(person'(x))(A(x)), 1)\rangle$

b. (object) *personne*$_{NP} \rightsquigarrow \lambda B_{et}.NO(y)(person'(y))(B(y))$
$\overset{D:6.1}{\Longleftrightarrow}$ (object) *personne*$_{NP} \rightsquigarrow \langle \lambda P_{et}.P(z_{e,2}), (\lambda B_{et}.NO(y)(person'(y))(B(y)), 2)\rangle$

c. *n'aime*$_{TV} \rightsquigarrow \langle \lambda X_{(et)t} \lambda u_e.X(\lambda v_e.love'(u, v))\rangle$

The subject *personne* stores its semantic representation $\lambda A_{et}.NO(x)(person'(x))(A(x))$ with the index 1 and contributes the expression $\lambda P_{et}.P(z_{e,1})$ to the combinatorics of the sentence. Similarly, the object *personne* stores its semantic representation under the index 2. The sentence is generated as in FIGURE 6.1.[18]

The interpretation of the sentence will be obtained by successively *retrieving* each of the quantifiers in the store representation $\langle love'(z_1, z_2), (\lambda A.NO(x)(person'(x))(A(x)), 1), (\lambda B.NO(y)$

[18]The core semantic representations at the IV and S level are obtained by λ-application and β-reduction. See Section 4.3.2.4 for a similar example with a detailed description of how this is done step by step.

$(person'(y))(B(y)), 2)\rangle$ on the S node. To do this, we λ-abstract over each of the variables z_1 and z_2 within the core representation and apply the corresponding quantifier to the lambda expression that we obtain (DEFINITION 6.2). The quantifier that is retrieved last will take wide scope. For instance, if we choose to first retrieve the quantifier contributed by the object *personne* and then the subject quantifier, we obtain the expression in (421), where the subject *personne* has wide scope. For the other scope interaction, we first retrieve the subject quantifier and then the object.

Definition 6.2 Cooper retrieval
Let σ_1 and σ_2 be possibly empty sequences of quantifier-index pairs. For every $z \in Ty1_e$, $\beta \in Ty1_{(et)t}$, $\phi \in Ty1_t$, $i \in \mathbb{N}^+$, if the store $\langle \phi, \sigma_1, (\beta, i), \sigma_2 \rangle$ is associated with an expression of category S, then the store $\langle \beta(\lambda z_i.\phi), \sigma_1, \sigma_2 \rangle$ is also associated with this expression.

(421) Quantifier retrieval for the S node in FIGURE 6.1:

a. Object quantifier retrieval:

$$\langle love'(z_1, z_2), (\lambda A.NO(x)(person'(x))(A(x)), 1), (\lambda B.NO(y)(person'(y))(B(y)), 2)\rangle$$
$$\overset{D:6.2}{\Longleftrightarrow} \langle [\lambda B.NO(y)(person'(y))(B(y))](\lambda z_2.love'(z_1, z_2)), (\lambda A.NO(x)(person'(x))(A(x)), 1)\rangle$$
$$\overset{\lambda-app}{\Longleftrightarrow} \langle NO(y)(person'(y))([\lambda z_2.love'(z_1, z_2)](y)), (\lambda A.NO(x)(person'(x))(A(x)), 1)\rangle$$
$$\overset{\beta-red}{\Longleftrightarrow} \langle NO(y)(person'(y))(love'(z_1, y)), (\lambda A.NO(x)(person'(x))(A(x)), 1)\rangle$$

b. Subject quantifier retrieval:

$$\langle NO(y)(person'(y))(love'(z_1, y)), (\lambda A.NO(x)(person'(x))(A(x)), 1)\rangle$$
$$\overset{D:6.2}{\Longleftrightarrow} \langle [\lambda A.NO(x)(person'(x))(A(x))](\lambda z_1.NO(y)(person'(y))(love'(z_1, y)))\rangle$$
$$\overset{\lambda-app}{\Longleftrightarrow} \langle NO(x)(person'(x))([\lambda z_1.NO(y)(person'(y))(love'(z_1, y))](x))\rangle$$
$$\overset{\beta-red}{\Longleftrightarrow} \langle NO(x)(person'(x))(NO(y)(person'(y))(love'(x, y)))\rangle$$

The result in (421) indicates that we can obtain the DN reading of (419) by means of Cooper storage. The inverse scope DN reading is also possible and would need to be ruled out by the grammar if it is not available for the sentence. At any rate, the result is the same as in the example in Section 4.3.2.4 where we did not employ Cooper storage, but only type shifting mechanisms. The question to ask now is: can Cooper storage help us to give a compositional syntax-semantics for polyadic quantifiers?

Polyadic quantifiers with Cooper storage? The HPSG analysis in de Swart and Sag (2002), where quantifiers are retrieved by means of Cooper storage, suggests that we should be able to get both the iteration and the resumption interpretations of the sentence in (419) by simply giving a refined definition to *retrieval*. In their terms, retrieval could be done as *iteration* or *resumption*.

De Swart and Sag do not make this proposal precise, so it is hard to guess how exactly they would do the retrieval. However, note that in order to get the resumptive reading for the expression under S in FIGURE 6.1, we need to retrieve both negative quantifiers at once. For this, we would need to λ-abstract both variables in $love'(z_1, z_2)$. Further we need to turn the two monadic quantifiers $\lambda A.NO(x)(person'(x))(A(x))$ and $\lambda B.NO(y)(person'(y))(B(y))$ into a binary quantifier

$\lambda V_{e(et)}.NO(x,y)(person'(x), person'(y))(V(x,y))$ that could be retrieved by applying to the binary relation $\lambda z_2 \lambda z_1.love'(z_1, z_2)$.

However, I showed in Section 4.3.3.3 that this cannot be done. There is no way to define an operation between any two unary relations in a domain E that would give us the desired correspondence to all the binary relations in E^2 (Henk Barendregt, p.c.). In Section 4.3.3.3, I showed that the cardinality of the domain E^2 of binary relations is usually different from that of the Cartesian product $E \times E$. This prevents us from expressing a direct correspondence between two monadic quantifiers and a binary one.[19]

In conclusion, Cooper storage cannot offer us a way to implement a syntax-semantics for resumption. Although it is a means to underspecify quantifier scope without appeal to syntax, it does make use of the typical compositional combinatorics which doesn't allow us to express polyadic quantifiers. The same problem would arise with other storage mechanisms that use lambda calculus in a functional type theory, as for instance the Keller storage (Keller (1988), see also Blackburn and Bos (2005, Ch. 3)). In Chapter 5 we were only able to formulate a syntax-semantics for NC as resumption, because LRS, the semantic framework employed, uses underspecified representations that can be identified and replaces the rigid compositional combinatorics with one based on the constituent structure fed by a surface-oriented syntax.

6.3 Concluding remarks

In this chapter I compared the analysis of NC in this thesis with alternative accounts in the literature. If one adheres to strict compositionality, one must take an NPI approach to NC. In this case one must, however, consider the large amount of counterevidence to the assumption that n-words are non-negative. In Section 6.1 I showed that the arguments brought against the negative quantifier status of n-words do not go through.

There are two observations about the NPI approaches that, to my mind, make them undesirable for the treatment of NC. First, they transfer the semantic problem raised by NC to the syntax, which is empirically unmotivated and theoretically unsatisfactory. Second, these theories altogether fail to provide us with a coherent story about n-words and the source of NC. This is because they sometimes make contradictory claims, despite common traits that qualify them all as NPI approaches. The second problem is a consequence of the first one: once one admits that n-words are negative and it is the task of the semantics to account for NC, we know exactly what we are after and we can learn with each analysis what step needs to be taken next. If one tries to argue that n-words are not negative, a whole range of possibilities suddenly open. Every other approach tries a new option and it is hard to identify its contribution to the original semantic problem, as the core solution is always a (new) syntactic one.

To understand the two problems, note first that NPI approaches try to build a one-to-one correspondence between the syntax and the semantics, for the sake of compositionality. This puts a great amount of weight on certain issues which are irrelevant from the semantic perspective, as for instance the question whether the negative quantifiers are universal quantifiers outscoping negation, or existential quantifiers outscoped by negation. From a semantic point of view, if the truth conditions are the same, this debate is immaterial. We don't have to represent negative quantifiers by an existential/ universal quantifier and a negative operator, we can assign them any symbol, as long as we associate the right semantics with it. Related to this, note that the idea of universal negative quantifiers in Zanuttini (1991) seems to have syntactic motivation as well: universal quantifiers are typically known to un-

[19] Note that this issue is independent of the compositionality problem raised by the incompatibility between the syntax of polyadic lifts and that of natural language that I discussed in Section 4.3.3.2.

dergo quantifier raising in generative grammar. Thus, by arguing that n-words are universal negative quantifiers, Zanuttini gets quantifier raising for free.

Moreover, NPI approaches try to account for the difference between n-words and NPIs/ indefinites on the basis of some syntactic mechanism that ensures that the right elements enter a NC constellation. A syntactic feature, which supposedly has nothing to do with the meaning of n-words, accounts for exactly those properties of n-words that the NQ approaches take to be indicative of their negative quantifier status. Thus negative concord as a semantic issue is turned into a purely syntactic issue. The distinction between the syntax and the semantics of n-words and NC eventually becomes unclear even for the proponents of NPI approaches, as claims in one approach sometimes contradict fundamental claims in others.

Consider, for instance, the uninterpretable Neg feature that n-words carry in some NPI approaches. Most of these approaches (e.g. Zeijlstra (2004), Penka (2007)) argue for this feature, although n-words are said to be semantically non-negative and evidence is brought for this idea. At the other extreme, Watanabe (2004) provides empirical evidence for treating n-words in Japanese as negative quantifiers, but accounts for NC readings in the syntax by means of an uninterpretable Neg feature. So the semantic task of accounting for NC is again transferred to the syntax. Furthermore, Déprez (1997) argues that French n-words are zero numerals. This means that they have the same semantics as negative quantifiers, but they are still indefinites in Déprez (1997) (presuppositional indefinites in Diesing (1992)). It is hard to see how one would work out the details of a syntax-semantics in this approach. I assume it would ultimately be an NQ approach, given that the empirical claim has much in common with the one in this thesis.

It seems to me that, beyond trying to argue against such analyses, we are in need of a common ground for discussion, which they fail to provide at the moment.

NQ approaches, on the other hand, have one problem to deal with: compositionality. In this thesis, I showed that resumptive quantifiers can straightforwardly account for NC in Romanian and similar proposals have been made for other languages as well (van Benthem (1989), May (1989), Keenan and Westerståhl (1997), de Swart and Sag (2002)). In Chapter 4 we saw that there is no way to define polyadic quantifiers in a compositional grammar. This is also the case for the Cooper storage mechanism (Section 6.2.2). Moreover, I argued in Section 6.2.1 that not all 'non-compositional' analyses are theoretically equally motivated and precise. We are in search of a systematic syntax-semantics for complex linguistic expressions. If the limits of compositionality, as understood at this point, are too tight for us to express the syntax-semantics of natural language, we are most likely in need of a reformulation of the notion of compositionality.

Chapter 7

Conclusion and perspectives

The contribution of this thesis can be regarded as both theoretical and applicative. First, it is a demonstration of how a syntax-semantics for negative concord can be built in general if we start with the assumption that n-words are negative quantifiers. Second, it applies this syntax-semantics to Romanian, for which it thus offers an extensive analysis of the core behavior of n-words and the defining properties of negative concord.

This enterprise has three main aspects that cut across the two (general vs. language-specific) dimensions: 1) an empirical one concerning the semantic status of n-words (Chapter 3 and Section 6.1), 2) the semantic mechanism for negative concord in relation to the principle of compositionality (Chapter 4), and 3) a systematic syntax-semantics for negative concord (Chapter 5 and Section 6.2).

In Section 7.1 I summarize the results and in Section 7.2 I discuss some general implications of the present analysis for linguistic theory and issues that remain open for future research.

7.1 Summary of results

Considering the empirical aspect, it is argued in Chapter 3 that Romanian n-words carry semantic negation so they should be treated as negative quantifiers. The claim that n-words are negative polarity items, put forth by NPI approaches to negative concord, is shown to be incompatible with the properties of negative concord and to make wrong predictions about the behavior of n-words in Romanian. In particular, I show that unlike NPIs, n-words do not need a semantic licenser, as they have anti-additive (negative) semantics themselves. Moreover, the negative marker, the only possible licenser for n-words, does not exhibit anti-additivity in combination with n-words, while it does with NPIs. It is also argued that *almost*-modification and the locality conditions on negative concord licensing point to a similarity between n-words and true quantifiers. The empirical tests brought by NPI approaches to determine the semantic status of n-words turn out to be compatible with the claim here that n-words are negative quantifiers (Section 6.1).

Two further important arguments are brought to support the negative semantics of n-words: the negative contribution in non-NC contexts (fragmentary answers, gapping, comparative, and past participial constructions) and the availability of a double negation interpretation for two n-words (Section 3.4). An NQ approach can straightforwardly account for these semantic facts, while NPI approaches usually offer a syntactic solution by appealing to covert negative licensers for which no independent evidence is available.

An investigation of the scope properties of two n-words reveals a close resemblance between negative concord and cumulative readings of cardinal quantifiers (Section 3.5). As cumulative readings can easily be analyzed with polyadic quantification (Section 2.1), this similarity is taken as indicative of the appropriateness of polyadic quantifiers in accounting for negative concord readings. This is the line of reasoning that I pursue in proposing a semantic account for negative concord and double negation with polyadic quantifiers.

The second aspect of this thesis concerns the semantic analysis of negative concord and double negation with polyadic quantifiers, and the investigation of the compositional status of iteration and resumption as polyadic lifts (Chapter 4). Double negation is accounted for as iteration and negative concord as resumption of negative quantifiers. Despite the reducibility of the resumptive negative quantifier NO^2 to the iteration $\mathrm{NO} \circ \mathrm{SOME}$, it is shown that only the former can account for the uniform semantics of n-words documented in Chapter 3 and for their idiosyncratic scope properties in negative concord.

The compositional status of polyadic lifts is investigated in a small compositional fragment of Romanian (Section 4.3). I show that the semantics of resumption disregards the syntactic parts involved in the operation and for this reason it cannot be made compositional with the logical syntax. Iteration can be given a compositional syntax-semantics in the logical language, but compositionality fails at the interface with the natural language syntax: the logical syntax of iteration as a mode of composition requires putting together two negative determiners in one step and natural language syntax does not have a surface constituent structure equivalent to this operation. It is also shown that the reason why polyadic lifts cannot be turned into modes of composition has to do with the expressive power of binary quantifiers which is higher than that of a combination of two monadic quantifiers in a functional type theory with λ-calculus, the combinatorics assumed in compositional grammars.

To give a syntax-semantics for polyadic quantifiers one must use a different combinatorics to build complex linguistic expressions from simple ones. Lexical Resource Semantics is a framework that offers the appropriate combinatorics. For this reason it is employed in Chapter 5 with the aim of developing a syntax-semantics for Romanian negative concord as resumptive quantification. LRS keeps the tradition of a logical representation language with functional types for semantics. But unlike the compositional grammar in Chapter 4 LRS can also employ underspecified representations. This is because LRS gives up the traditional techniques of combining the syntactic parts in a functional type theory with λ-calculus and replaces them with a constraint-based combinatorics that respects the surface constituent structure of the natural language. This shift allows an encoding of generalized quantifiers in LRS as resumptive quantifiers of an underspecified complexity. In an HPSG syntax-semantics interface this permits an account of Romanian negative concord in which two lexical negative quantifiers identify their lists of variables, restrictions, and the nuclear scope and give rise to one binary resumptive quantifier. By means of this resumptive negative quantifier, we obtain the resumptive semantics without any appeal to a supplementary mode of composition.

Thus I present a syntax-semantics interface for the negative concord reading of two negative quantifiers with possible extension to n quantifiers and for the locality conditions in the licensing relation between n-words and the negative marker. Double negation readings receive an analysis which integrates the necessary information structure conditions. With these results, the present account makes the right predictions about the availability of the negative concord and the double negation reading as related to the scope interaction between negative and non-negative quantifiers.

In comparison to the NQ approaches using negation factorization, the present resumption-based approach to negative concord is argued to be theoretically superior, as resumptive quantifiers can be given a systematic syntax-semantics and have a precise mathematical status in an extended theory of generalized quantifiers (Section 6.2). Considering other options of integrating resumptive quantifiers in a syntax-semantics, it is shown that employing Cooper storage to underspecify quantifier scope is not helpful in overcoming the compositionality problem with resumptive quantifiers. Cooper storage keeps the traditional combinatorics with functional types and λ-calculus, so it has the limitations of any other compositional grammar, and lacks the flexibility that LRS obtains by giving up this tradition.

In conclusion, the LRS account of negative concord in this thesis is the only one to date I am aware of that gives a systematic syntax-semantics for a linguistic phenomenon that has been argued to require the expressive power of polyadic quantifiers.

7.2 Perspectives for future research

For an account of negative concord, in this thesis I make the unconventional choice of treating n-words as negative quantifiers and offering a non-compositional, though systematic, syntax-semantics.

The starting point in this thesis is that n-words are negative quantifiers. This commits me to a particular kind of analysis, usually avoided in the literature, which, however, allows me to investigate the precise points where compositionality and the analysis of negative concord as a polyadic quantifier come in conflict. This is a broad theoretical problem and for this reason, this thesis is quite programmatic, so it cannot pretend to have exhausted all theoretical issues, or the entire empirical domain of Romanian negative concord. There are several stimulating questions that arise for further research, of which I mention a few below.

7.2.1 Empirical coverage

Adverbial n-words First, this thesis does not account for negative concord with modifiers. The analysis has been developed to cover argument n-words. Adverbial and prepositional modifiers, however, also participate in negative concord constructions and should be taken into account by an extended analysis. Negative quantifier adverbs like *niciodată* 'never', *nicăieri* 'nowhere', *nicidecum* 'nohow' in (422) are examples:

(422) a. **Niciun** student **nu** a venit **niciodată** tîrziu la ore.
no student NM has come never late to classes
'No student ever came late to classes.'

b. Ion **nu** va merge **nicidecum** singur **nicăieri**.
John NM will go nohow alone nowhere
'There is no way John will go anywhere alone.'

To integrate adverbial n-words in negative concord structures, the simplest assumption would be that they are negative quantifiers that take a variable. These quantifiers usually have a restriction referring to the time/ location/ manner of an event. Given the one-to-one correspondence between the number of variables and that of restriction predicates in the present grammar, the quantifiers should have as many variables as they have restrictions. What kind of a variable this should be and how it relates to the verb in their nuclear scope is to be determined by independent study on event modification. Our account would need to be extended to include such (event) variables of a possibly different type from e, the common type of nominal variables.

A good starting point for an account of adverbial n-words is the extensive study of quantificational adverbs like *always*, *often* as generalized quantifiers in de Swart (1993). A few adjustments would be necessary to deal with adverbs of manner and location besides the temporal ones and to make this analysis fit into our LRS account.

A treatment of adverbial n-words in the present analysis is also interesting from the point of view of polyadic quantifiers, where the literature usually focuses on NPs. For instance, it would be relevant for an account of resumptive quantification in English as in Peters and Westerståhl (2002). Peters and Westerstahl claim that quantificational adverbs in English can trigger resumptive readings in sentences like (423), but they do not give the quantifier representation of the adverb itself, they limit their attention to the two nominal restrictions contributed by the bare plurals *cats* and *dogs* as the restriction of the quantificational operator MOST.

(423) a. *Cats* **usually** dislike *dogs*.
b. $\mathrm{MOST}^2(\mathrm{CAT} \times \mathrm{DOG}, \mathrm{DISLIKE})$

Although this is not made precise, Peters and Westerståhl (2002) suggests a binary resumptive quantifier like in (423b) to account for the interpretation of (423a). That is, they do not take into account the time restriction of the quantifier *usually*. While this is irrelevant for their purposes, in our LRS grammar, the two nouns *cats* and *dogs* would not contribute the quantifier themselves, so they wouldn't be able to take part in the resumptive quantifier. We would need the adverbial quantifier to be represented and get identified with a possibly underspecified quantifier contributed by the two bare nouns to build the resumptive quantifier. Then the adverb would also contribute its time restriction to the complex restriction of the resumptive quantifier. This also implies that in (423b) we would have MOST^3 instead of MOST^2, as we would have three restrictions: CAT, DOG, and TIME.

Split readings A second question concerns the way the present analysis can integrate split readings of n-words (see (408) repeated in (424)). In Section 6.1.3 I showed that cardinal quantifiers can also get a split scope interpretation in their event readings (see (413b), repeated in (425)). Thus we are in need of an appropriate mechanism to deal with split readings of quantifiers independently of negative concord and negative quantifiers. In Section 6.1.3 I suggested that discontinuous representations in LRS should allow a natural treatment of these readings.

(424) A: I hear that John is trying to find a secretary.

B: Nu e adevărat. Ion nu caută **nicio** secretară.
NM is true John NM seeks no secretary

$\neg > \mathrm{SEEK} > \exists$: 'That's not true. John is not trying to find a secretary.'

(425) Ion a căutat **patruzeci** de secretare anul trecut.
John has sought forty of secretaries year last
$40 > \mathrm{SEEK} > \exists$: 'There were forty times when John tried to find a secretary last year.'

Accounting for split readings may, however, require assuming lexical ambiguity of n-words between a behavior like existential quantifiers (or simply, indefinites) in contexts with split readings and as negative quantifiers everywhere else. This would, of course, be an undesired solution. N-words have been argued to be indefinites licensed by an abstract negative operator (Penka (2007)), but this option would result in positing infinitely many such operators to account for the split readings of cardinal quantifiers. Before making generalizations about the semantic nature of n-words and cardinal quantifiers, I think a better understanding of the empirical phenomenon is necessary.

A comparison to floating quantifier constructions like (426) may turn out useful in this respect. In (426) the quantifier appears separated from its nominal restriction in the syntax (Dowty and Brodie (1984), Sportiche (1988), Fukushima (1993), Nakanishi (2005) and many others). In split readings, the split is not in the syntax, but in the semantics: the quantifier forms a DP with the noun, but the latter is interpreted in the nuclear scope rather than in the restriction, due to an intervening operator (see (424) and (425)).

(426) a. *The students* have **all** read the book.

b. (Dintre) *Studenţi* au venit **trei** ieri la curs.
(of) students have come three yesterday to class
'As for students, three (of them) came to class yesterday.' (Romanian)

c. *Cărţi* am adus **multe** astăzi.
books have brought many today
'As for books, I brought many today.' (Romanian)

d. *Fotos* wurden **keine** gemacht.
photos were no made
'As for photos, there weren't made any.' (German)

Nakanishi argues that floating quantifiers measure in the verbal domain using arguments like the fact that sentences like (426b) lack a collective reading, specific to quantifiers that measure in the nominal domain. The collective reading is usually available for the non-split *trei studenţi* together with the distributive one (427). In (426b) we only obtain a distributive reading for *trei*, which indicates that the quantifier refers to the number of events.

(427) *Trei studenţi* au venit ieri la curs.
three students have come yesterday to class
'Three students came to class yesterday.'

In Section 6.1.3 we saw that split readings of cardinal quantifiers appear in their event reading. We thus expect the two phenomena to receive the same kind of semantics that accounts for their quantification over the event variable. If we could relate split readings of n-words to the same semantics, we would then be in the position to offer an account of split readings in general. Floating quantifiers would differ from split readings only in that they also exhibit syntactic effects. The literature on floating quantifiers offers various observations and tests that may shed some light on split readings, which we know very little about at the moment.

Apart from the empirical issues, an account for split readings requires a logic with a world type to deal with intensional constructions and this is an important next step for the research initiated in this thesis. Intensionality should be technically easy to integrate in the LRS analysis here, as previous LRS literature makes full use of it.

Negation and the use of the subjunctive On the empirical side of intensionality, there is one further issue that deserves special attention and is relevant at least for Romanian and Romance languages. It concerns subjunctive relative clauses modifying n-words and relates to a discussion introduced in Farkas (1985). According to Farkas, a subjunctive relative clause modifying the indefinite object of an intensional verb disambiguates the latter to a de dicto reading (428):

(428) Ion caută *o* secretară [care să ştie chineză].
John seeks a secretary that SJ know Chinese

a. 'John is trying to find a secretary that knows Chinese.' (de dicto)

b. # 'There is a secretary that knows Chinese and John seeks her.' (de re)

As noted in Ionescu (2004), Romanian n-words in intensional contexts can also be modified by subjunctive relative clauses, thus confirming their de dicto (or rather, split scope) reading:

(429) Ion **nu** caută **nicio** secretară [care să ştie chineză].
John NM seeks no secretary that SJ know Chinese
'It is not the case that John is involved in a search for a secretary that knows Chinese.'

However, subjunctive relative clauses seem to successfully modify n-words even when they are objects to extensional verbs (430a):

(430) a. Ion **nu** a întîlnit **nicio** secretară [care să ştie chineză].
John NM has met no secretary that SJ know Chinese
'John didn't meet any secretary that knows Chinese.'

b. Ion **nu** a întîlnit *o* secretară [care să ştie chineză].
John NM has met a secretary that SJ know Chinese
'John didn't meet a secretary that knows Chinese.'

c. * Ion a întîlnit *o* secretară [care să ştie chineză].
John has met a secretary that SJ know Chinese
'John met a secretary that knows Chinese.'

The n-word in (430a) can be replaced by the indefinite *o* 'a' as long as the sentential negation is present (430b). But the sentence becomes ungrammatical if there is no negation in the matrix clause (430c). So the subjunctive relative clause modifying n-words doesn't depend only on the intensional context, but also on the negative context.

Given that both (429) and (430a) contain a subjunctive relative clause, although the latter does not provide an intensional context, the question that arises is whether (429) also receives a de re reading. In that case, the subjunctive relative clause would be allowed by the negative context, like in (430a). But a de re reading does not seem to be available in (429), which confirms Farkas's (1985) claim. Now we have to ask what exactly it is that the negation in (430a) and an (even affirmative) intensional context ((428), (429)) have in common that allows the occurrence of subjunctive relative clauses. The answer will probably also have to do with the semantics of the subjunctive.

Another place where negation enables the use of the subjunctive otherwise disallowed in the corresponding affirmative context concerns propositional intensional verbs like *crede* 'believe'. Negating *crede* may turn a 'that' complement clause (431a) into a subjunctive complement (431d) which is ruled out in the affirmative (431b). In (431d) we have both a negative and an intensional context for the subjunctive clause.

(431) a. Ion crede *că are* cartea.
John believes that has book-the
'John believes that he has the book.'

b. * Ion crede *să aibă* cartea.
John believes SJ have book-the
'John believes to have the book.'

c. Ion **nu** crede *că are* cartea.
John NM believes that has book-the
'John doesn't believe that he has the book. '

d. Ion **nu** crede *să aibă* cartea.
John NM believes SJ have book-the
'John doesn't believe to have the book.'

Similar constructions seem to occur in other Romance languages and a close investigation might provide us with a better understanding of negation and its relation to the subjunctive and intensionality.

Non-finite and *fără* 'without' constructions Another empirical area that has not been addressed in this thesis concerns non-finite constructions (432) and contexts where n-words cooccur with *fără* 'without' (433). Two issues have to be mentioned here: a general one on how we would include such constructions in the grammar, and a more specific one regarding past participial constructions with preverbal n-words.

Regarding the first issue, it should be easy to account for the two constructions, if we assume that *ne* and *fără* carry a NO^0 quantifier. We would also modify the NEGATIVE CONCORD CONSTRAINT in (329), p. 211 to enforce the *ne*- prefix on the verb in non-finite clauses with n-words. At the same time the presence of *fără* should allow n-words to appear in *fără* constructions. Other possible differences from negative concord in finite utterances would have to be solved locally.

(432) Maria a plecat **ne**observată de **nimeni**.
Mary has left un-noticed by nobody
'Mary left without being noticed by anybody.'

(433) Ion a rezolvat problema *fără* **niciun** ajutor/ a cere ajutor **nimănui**/ să ceară ajutor **nimănui**.
John has solved problem-the without no help/ to ask help nobody-Dat/ SJ ask help nobody-Dat
'John solved the problem without any help/ asking anybody for help.'

The second issue is more intriguing, as it raises several empirical and theoretical questions for which we don't have an answer yet. As mentioned in Chapter 3, p. 94, a preverbal n-word can trigger negation alone without the prefix *ne*- on the verb (see (434a)). (434a) is to be compared to (434b) which shows that even in participial constructions n-words in postverbal position always require the presence of the NM *ne*- on the verb:

(434) a. articol de **nimeni** citat
article by nobody cited
'article which hasn't been cited by anybody'

b. Acest articol *(**ne**)citat de **niciun** critic este de fapt foarte interesant.
this article un-cited by no critic is in fact very interesting
'This article, which wasn't cited by any critic, is actually very interesting.'

To account for (434b) we need to assume a NEGATIVE CONCORD CONSTRAINT for non-finite clauses that enforces the presence of the prefix *ne*- on the verb if an n-word is present. This NEGATIVE CONCORD CONSTRAINT would then rule out (434a), because the verb does not have the prefix *ne*-.

An account for (434a) in our analysis shouldn't be too difficult from a technical point of view. Once we have the properties of the construction, an HPSG grammar offers enough flexibility for us to specify the NEGATIVE CONCORD CONSTRAINT in a weaker way as to correctly describe both (434b) and (434a). A sketch of such an account is given in Iordăchioaia (2004), for instance. What seems more intriguing about these constructions is identifying the source of the contrast between (434b) and (434a) and seeing what in the nature of participial constructions allows this variation. In what follows I give a few observations in this respect.

Note that these constructions are special also independently of negation, as typical arguments are usually disallowed to occur in preverbal position. Quantificational elements seem to be an exception: in (435) the quantifiers *adesea* 'often' and *de toată lumea* 'by everybody' can appear preverbally, but *de Ion* 'by John' cannot:

(435) un articol ***de Ion**/ **adesea**/ **de toată lumea** citat
an article by John/ often/ by all world cited
'an article which has (often) been cited by John/ by everybody'

These constructions have been related to the adjectival nature of past participles (Iordăchioaia (2004)), since a similar case occurs with adjectives, which can be negated by an adverbial n-word:

(436) o secretară **niciodată**/ **adesea** disponibilă
a secreatry never/ often available
'a secretary who is never/ often available'

Concerning the past participle, Parsons (1991, pp. 234–235) distinguishes between the target state and the resultant state of an event as expressed by the past participle: "If I throw a ball onto the roof, the target state of this event is the ball's being on the roof, a state that may or may not last for a long time"; the resultant state is "the state of my having thrown the ball onto the roof and (...) cannot cease holding at some later time". In the literature on the adjectival use of the past participle Kratzer (2000) and Alexiadou and Anagnostopoulou (2008) identify several semantic and syntactic tests that support this distinction.

The question that arises for us is whether the distinction between *target-state* and *resultant-state participle* is relevant for the behavior of the past participle in (434a). Kratzer showed that the two kinds of participles get a different interpretation with the negative prefix: target-state participles are interpreted as *contrary negation* (e.g. *un*happy, where *not* (*un*happy) $\neq$ happy), and resultant-state participles as *contradictory negation* (e.g. *non*-black, where *not* (*non*-black) $=$ black, cf. Horn (1989, Ch. 5)). If this distinction plays a role in the past participial constructions with preverbal n-words, we may also be able to determine the kind of negation that is involved in NC (see also (437)).

This brings us to another issue in the literature on negative concord introduced in Przepiórkowski (1999*b*): the relevance of the distinction in Situation Semantics (Cooper (1997)) between *eventuality negation* and *propositional negation*. For a proposition p, the negation of p is expressed as "It is the case that not-p" in the eventuality negation reading and as "It is not the case that p" in the propositional negation reading. Przepiórkowski argues that negative concord (at least in Polish and Italian) occurs only with eventuality negation. It remains to be seen how and whether we can express this distinction in model-theoretic semantics and, more importantly, if it has a crucial role to play in NC. An extensive investigation of the constructions in (434a) might give an answer to this as well, since the preverbal n-word does not seem to license other n-words (see (437), taken from Ionescu (1999)). If only one of the two kinds of negation enters NC, the preverbal n-word must contribute the other one:

(437) * carte de **nimeni** citită **niciodată**
book by nobody read never
'book that has never been read by anybody'

A further question is to what extent Romanian data like (434a) are related to those contexts in non-strict NC languages where a preverbal n-word contributes negation alone:

(438) a. **Nessuno** (*non) ha letto **niente**/ il libro.
nobody (NM) has read nothing/ the book
'Nobody read anything/ the book.' (Italian)

b. **Nadie** (*no) dijo **nada**/ eso.
nobody (NM) said nothing/ this
'Nobody said anything/ this.' (Spanish)

Note, however, that the parallelism between *nessuno* and *nadie* in (438) and *de nimeni* in (434a) is not complete, since the former can license other n-words, while the latter cannot (437).

From the point of view of the assumption here that n-words are negative quantifiers neither (438), nor (434a) are unexpected. Still, there seems to be something special about the preverbal position that gives an n-word more independence than the postverbal position and we need to identify the source of this contrast. Information structure seems to play an important role, as noted in Zanuttini (1991), Isac (2004), and Watanabe (2004), but it is probably not the only factor that influences the behavior of n-words in these two contexts.

7.2.2 Theoretical issues

Other polyadic quantifiers in LRS The LRS- and HPSG-based account of negative concord in this thesis is, as far as I am aware, the first to integrate polyadic quantifiers in a coherent syntax-semantics interface. This is achieved by underspecifying generalized quantifiers as n-ary resumptive quantifiers. The quantificational operator, however, is always the same. In Section 2.1 I mentioned several other instances of polyadic quantifiers where distinct quantificational operators together build a polyadic quantifier: iteration, "different/ same", and cumulative quantifiers. The semantic effects of iteration can be accounted for in our grammar, independently of the polyadic lift iteration. But there are natural language quantifiers like "different/ same" constructions and cumulative readings of cardinal quantifiers for which iteration cannot derive the right semantics. The polyadic quantifiers that correctly interpret them are also shown by Keenan to not be reducible to iteration.

Such quantifiers cannot receive a syntax-semantics in the grammar that I give in Chapter 5, as it does not allow building a generalized quantifier with more than one quantificational operator. The grammar can, however, be naturally extended to accommodate a new attribute OPERATOR that specifies the quantificational operator of a generalized quantifier. By further allowing the value of this attribute to be a list of operators, just like in the case of the VARIABLE and RESTRICTION attributes, we can derive any operator combinations for polyadic quantifiers. A precise formulation of the possible operator combinations will have to be ensured by supplementary LRS constraints. Though far from trivial, interpreting these quantifiers in our LRS account is only a matter of how we represent them in $Ty1$; their semantics is already provided by the polyadic quantifier literature.

Negative concord as a cumulative quantifier Another way to account for NC, one might think[1], is in terms of a cumulative quantifier, if we assume that n-words are zero cardinal quantifiers and thus have the same truth conditions as negative quantifiers. Moreover, cumulative quantifiers might pose fewer problems for compositionality than resumptive ones, given that their semantics takes into account the contribution of each monadic quantifier. This approach seems quite appealing, especially if we can make it compositional. It would also conform with the conclusion that we reached with respect to the semantics of n-words, while the general claim is similar to Déprez (1997).

[1]This suggestion comes from an anonymous semantics workshop reviewer.

There is, however, a piece of data that prevents us from taking up this option for now. If n-words are zero cardinal quantifiers and negative concord is an instance of a cumulative quantifier, we expect n-words to also get a cumulative reading when they cooccur with true zero quantifiers like in (439a):

(439) a. **Niciun** student **nu** a citit *zero* cărţi pentru examen.
no student NM has read zero books for exam
i. 'No student read zero books for the exam.' (DN)
ii. # 'No student read any book for the exam.' (NC)

b. *Patruzeci* de studenţi au citit *zero* cărţi pentru examen.
forty of students has read zero books for exam
i. 'Forty students read each zero books for the exam.' (iteration)
ii. 'A total of forty students read a total of zero books for the exam.' (cumulation)

The sentence in (439a) with an n-word and a zero quantifier cannot be interpreted as NC, it only receives a DN reading. The lack of a NC reading indicates two things: 1) the two quantifiers do not undergo resumption and 2) they cannot be interpreted cumulatively. The unavailability of resumption is a sign that *niciun* and *zero* contribute distinct monadic quantifiers. Recall from Section 2.1 that resumption requires that the monadic quantifiers carry the same operator. Cumulative readings of cardinal quantifiers impose no particular constraint on the operator of the monadic quantifiers (see the interpretation ii. in (439b)).[2] But this interpretation is not available for (439a) either. If it were available we would have the semantics of a conjunction of two zero quantifiers which is interpreted as negative (see Section 2.1), so we would have negative concord. By contrast, the DN reading of (439a) indicates that we only get a scopal interpretation of the two quantifiers which are composed by iteration.

In conclusion, negative concord cannot be accounted for as a cumulation of zero quantifiers, unless we introduce special conditions on n-words as cardinal quantifiers. For this, we do not have any independent motivation at the moment.

Compositionality The last theoretical implication of this thesis that I want to mention here is most likely also the most controversial and concerns compositionality. I showed that polyadic quantifiers as they are conceived of in the Extended Generalized Quantifier Theory cannot be reconciled with our notion of compositionality. This is due to the fact that compositionality in linguistics is inseparable from the technique of syntactically combining linguistic expressions by means of λ-calculus with a functional type theory. An HPSG syntax-semantics for resumptive quantifiers was possible in this thesis, because LRS gives up this kind of combinatorics.

The present analysis of negative concord acknowledges the negative semantics of n-words and their special behavior in concord constructions at the price of replacing the traditionally compositional combinatorics of a functional type theory with a new one guided by constituent structure. My choice raises the discussion of what compromise with respect to our previous understanding is more adequate in order to maintain *compositional transparency* in representing the syntax-semantics of natural language: is it preferable to give up the idea that n-words are negative, or the semantic combinatorics?

Dowty (2007) argues that to the extent that speakers utter and understand infinitely many sentences, natural language must be compositional and the task of the linguist is to define this property

[2]There is a clear difference in truth conditions between the readings i. and ii. in (439b). In a context where no student read an entire book but each of them read some different pages in one and the same book, such that when all the pages read by the forty students are put together they make up the entire book, the reading in i. is true, while the one in ii. is false. In this case, the total of forty students read one book altogether.

in a mathematically precise and empirically adequate way. In Dowty's understanding of compositionality, there cannot be counterexamples or challenges to compositionality itself, only to the way linguists represent it. The issue is not whether language is compositional, but where *transparent compositionality* stops and how compositionality functions from that point on. Dowty formulates three inter-related properties on the basis of which we should evaluate our linguistic hypotheses about compositionality: *compositional transparency*, *syntactic economy* and *structural semantic economy*. The first concerns the degree to which the semantic interpretation is immediately computable from the syntactic structure. The second requires the syntactic structures to be no more complicated than they need to be to reflect the compositional semantics. The third applies to the semantic operations and requires them to be no more complicated than they need to be in order to derive all the complete sentence meanings that natural language expresses. To put it briefly, given two ways of building a complex semantics as compositional, we would prefer the one on the basis of which natural language is more compositionally transparent and more economical in terms of syntax and combinatorics.

I have provided crucial evidence for the negative semantics of n-words, which indicates that assuming that n-words are not negative reduces compositional transparency. Moreover, the approaches that follow this assumption are forced to employ a non-economical syntax with abstract semantic operators that are not syntactically justified. Covert categories in general create opacity with respect to compositionality.

In my analysis, I attribute n-words a negative quantifier status, but I use a higher-level semantic operation and non-standard semantic combinatorics to ensure compositional transparency. A surface-oriented syntax as the one employed here is as economical as syntax can be and using a constituent structure-based combinatorics is obviously not less economical, if we are to consider semantic combinatorics part of the structural semantics in Dowty's view. The general rules for constituent structure formation are used as semantic structural rules. We do not only have economical semantic combinatorics, we reduce it to syntactic structures. What we need is a higher-level notion of quantification, where an operator can bind several variables. This addition, however, is also required by other natural language phenomena as argued in the generalized quantifier literature. What about compositional transparency? In the analysis I have presented negative concord is obtained by means of standard HPSG mechanisms that allow token-identity between variables of the same sort: a lexically contributed monadic quantifier can be identified with the polyadic quantifier contributed by several n-words at the sentential level. This means that the complexity of a lexically contributed quantifier is underspecified.

One might prefer a slightly less economical syntax with a more transparent semantic combinatorics that does not use underspecification like LRS with HPSG does, and employs the traditional semantic combinatorics instead. This would be a harmless compromise in favor of keeping our strong semantic tradition. However, I have shown that a Cooper storage mechanism, the only syntactic mechanism that has been proposed to accommodate polyadic quantifiers is not flexible enough to allow a systematic syntax-semantics for negative concord, for similar reasons as other accounts that use a functional type theory with lambda calculus. The present analysis thus remains the only one that accounts for negative concord in a way that is compatible with its syntactic and semantic conditions.

In conclusion, going back to Dowty's view that natural language *is* compositional and linguists must find the adequate way to define this property, it is important for linguistic theory that the mechanism we traditionally use to define compositionality is not allowed to prevent us from accounting for empirical phenomena that do not comply with the notion of compositionality that we have created. If this kind of situation occurs, it may be an indicator that we are in need of a refined mechanism, and we should continue to consider alternatives to our definition of compositionality.

I have brought evidence here that negative concord in Romanian indicates a deficiency in our traditional understanding of compositionality. Given that negative concord is the rule rather than the exception in the languages of the world (see Haspelmath (2005)), this shows that the mechanism we have at the moment needs reconsideration of alternatives one of which is the analysis presented here in Chapter 5.

Appendix A

Definitions for Section 5.2

(440) (Intended model I_{Ty1})
Extensions to Definition 3.2 of Sailer (2003, pp. 117-118 and p. 395)

We enrich the set of species given in Sailer (2003) with the additional species in Γ_{Ty1}: *elist*, *nelist*, *no*, *some*, and *every*, corresponding to (sequences of) $Ty1$ terms. Thus:

$\mathrm{S}_{Ty1}((\,))$ = *elist*,

for each $\tau \in Type$, for each $n \in \mathbb{N}^{+}$, for each $\alpha_{1,\tau}, \alpha_{2,\tau}, ..., \alpha_{n,\tau} \in Ty1$,

$\mathrm{S}_{Ty1}(\alpha_{1,\tau}, \alpha_{2,\tau}, ..., \alpha_{n,\tau})$= *nelist*,

for each $\tau \in Type$, for each $n \in \mathbb{N}^{0}$, for each $i_1, i_2, ..., i_n \in \mathbb{N}^{+}$,
for each $v_{i_1,\tau}, v_{i_2,\tau}, ..., v_{i_n,\tau} \in Var$, for each $\alpha_{t1}, \alpha_{t2}, ..., \alpha_{tn}, \beta_t \in Ty1$,

$\mathrm{S}_{Ty1}((NO(v_{i_1,\tau}, ..., v_{i_n,\tau})(\alpha_{t1}, ...\alpha_{tn})(\beta_t))_t)$= *no*,

for each $\tau \in Type$, for each $n \in \mathbb{N}^{+}$, for each $i_1, i_2, ..., i_n \in \mathbb{N}^{+}$,
for each $v_{i_1,\tau}, v_{i_2,\tau}, ..., v_{i_n,\tau} \in Var$, for each $\alpha_{t1}, \alpha_{t2}, ..., \alpha_{tn}, \beta_t \in Ty1$,

$\mathrm{S}_{Ty1}((SOME(v_{i_1,\tau}, ..., v_{i_n,\tau})(\alpha_{t1}, ...\alpha_{tn})(\beta_t))_t)$= *some*,

for each $\tau \in Type$, for each $n \in \mathbb{N}^{+}$, for each $i_1, i_2, ..., i_n \in \mathbb{N}^{+}$,
for each $v_{i_1,\tau}, v_{i_2,\tau}, ..., v_{i_n,\tau} \in Var$, for each $\alpha_{t1}, \alpha_{t2}, ..., \alpha_{tn}, \beta_t \in Ty1$,

$\mathrm{S}_{Ty1}((EVERY(v_{i_1,\tau}, ..., v_{i_n,\tau})(\alpha_{t1}, ...\alpha_{tn})(\beta_t))_t)$= *every*,

We enrich the set of attributes given in Sailer (2003) with those attributes introduced by *nelist* and *gen-quantifier* in Γ_{Ty1}. Thus:

for each $\tau \in Type$, for each $n \in \mathbb{N}^{+}$, for each $\alpha_{\tau 1}, \alpha_{\tau 2}, ..., \alpha_{\tau n} \in Ty1$,

A_{Ty1}(FIRST)$((\alpha_{\tau 1}, \alpha_{\tau 2}, ..., \alpha_{\tau n}))$ = $\alpha_{\tau 1}$,

A_{Ty1}(REST)$((\alpha_{\tau 1}, \alpha_{\tau 2}, ..., \alpha_{\tau n}))$ = $(\alpha_{\tau 2}, ..., \alpha_{\tau n})$,

for each $\tau \in Type$, for each $n \in \mathbb{N}^0$, for each $i_1, i_2, ..., i_n \in \mathbb{N}^+$,
for each $v_{i_1,\tau}, v_{i_2,\tau}, ..., v_{i_n,\tau} \in Var$, for each $\alpha_{t1}, \alpha_{t2}, ..., \alpha_{tn}, \beta_t \in Ty1$,
for each $NO(v_{i_1,\tau}, ..., v_{i_n,\tau})(\alpha_{t1}, ...\alpha_{tn})(\beta_t) \in Ty1$,

A_{Ty1}(VAR)($NO(v_{i_1,\tau}, ..., v_{i_n,\tau})(\alpha_{t1}, ...\alpha_{tn})(\beta_t)$)= $(v_{i_1,\tau}, ..., v_{i_n,\tau})$, and
A_{Ty1}(RESTR)($NO(v_{i_1,\tau}, ..., v_{i_n,\tau})(\alpha_{t1}, ...\alpha_{tn})(\beta_t)$)= $(\alpha_{t1}, ..., \alpha_{tn})$, and
A_{Ty1}(SCOPE)($NO(v_{i_1,\tau}, ..., v_{i_n,\tau})(\alpha_{t1}, ...\alpha_{tn})(\beta_t)$)= β_t,

for each $\tau \in Type$, for each $n \in \mathbb{N}^+$, for each $i_1, i_2, ..., i_n \in \mathbb{N}^+$,
for each $v_{i_1,\tau}, v_{i_2,\tau}, ..., v_{i_n,\tau} \in Var$, for each $\alpha_{t1}, \alpha_{t2}, ..., \alpha_{tn}, \beta_t \in Ty1$,
for each $SOME(v_{i_1,\tau}, ..., v_{i_n,\tau})(\alpha_{t1}, ...\alpha_{tn})(\beta_t)) \in Ty1$,

A_{Ty1}(VAR)($SOME(v_{i_1,\tau}, ..., v_{i_n,\tau})(\alpha_{t1}, ...\alpha_{tn})(\beta_t)$)= $(v_{i_1,\tau}, ..., v_{i_n,\tau})$, and
A_{Ty1}(RESTR)($SOME(v_{i_1,\tau}, ..., v_{i_n,\tau})(\alpha_{t1}, ...\alpha_{tn})(\beta_t)$)= $(\alpha_{t1}, ..., \alpha_{tn})$, and
A_{Ty1}(SCOPE)($SOME(v_{i_1,\tau}, ..., v_{i_n,\tau})(\alpha_{t1}, ...\alpha_{tn})(\beta_t)$)= β_t,

for each $\tau \in Type$, for each $n \in \mathbb{N}^+$, for each $i_1, i_2, ..., i_n \in \mathbb{N}^+$,
for each $v_{i_1,\tau}, v_{i_2,\tau}, ..., v_{i_n,\tau} \in Var$, for each $\alpha_{t1}, \alpha_{t2}, ..., \alpha_{tn}, \beta_t \in Ty1$,
for each $EVERY(v_{i_1,\tau}, ..., v_{i_n,\tau})(\alpha_{t1}, ...\alpha_{tn})(\beta_t)) \in Ty1$,

A_{Ty1}(VAR)($EVERY(v_{i_1,\tau}, ..., v_{i_n,\tau})(\alpha_{t1}, ...\alpha_{tn})(\beta_t)$)= $(v_{i_1,\tau}, ..., v_{i_n,\tau})$, and
A_{Ty1}(RESTR)($EVERY(v_{i_1,\tau}, ..., v_{i_n,\tau})(\alpha_{t1}, ...\alpha_{tn})(\beta_t)$)= $(\alpha_{t1}, ..., \alpha_{tn})$, and
A_{Ty1}(SCOPE)($EVERY(v_{i_1,\tau}, ..., v_{i_n,\tau})(\alpha_{t1}, ...\alpha_{tn})(\beta_t)$)= β_t.

(441) (Extension of *me* objects)
Extensions to Definition 3.9 of Sailer (2003, pp. 123–124 and p. 396)

We specify the extension of the additional $Ty1$ terms (i.e. generalized quantifiers) that do not appear in the grammar of $Ty2$ in Sailer (2003). Thus:

if S(*u*)$\sqsubseteq$ *no*,

such that there are $(v_1, v_2, ..., v_n), (\alpha_1, \alpha_2, ..., \alpha_n), t \in U_{Ty1}$, with
$(v_1, v_2, ..., v_n)$ = T_{I}(:VAR)(*u*),
$(\alpha_1, \alpha_2, ..., \alpha_n)$ = T_{I}(:RESTR)(*u*),
$t = T_{\mathrm{I}}$(:VAR FIRST TYPE)(*u*)

$\{\![u]\!\}^{M,A}$= 1 if

for each $d_1, d_2, ..., d_n \in D_{E,W,[t]}$
$\{\!\alpha_1\!\}^{M,A[v_1/d_1]}$= 0 or . . . ,
or $\{\!\alpha_n\!\}^{M,A[v_n/d_n]}$= 0,
or $\{\![T_{\mathrm{I}}(\text{:SCOPE})(u)]\!\}^{M,A[(v_1,...,v_n)/(d_1,...,d_n)]}$= 0,

else 0.

if S(*u*)$\sqsubseteq$ *some*,

such that there are $(v_1, v_2, ..., v_n), (\alpha_1, \alpha_2, ..., \alpha_n), t \in U_{Ty1}$, with
$(v_1, v_2, ..., v_n)$ = T_{I}(:VAR)(*u*),
$(\alpha_1, \alpha_2, ..., \alpha_n)$ = T_{I}(:RESTR)(*u*),
$t = T_{\text{I}}$(:VAR FIRST TYPE)(*u*)
$\{\![[u]]\!\}^{M,A}$= 1 if
there exist $d_1, d_2, ..., d_n \in D_{E,W,[t]}$
$\{\!|\alpha_1|\!\}^{M,A[v_1/d_1]}$= 1 and . . . ,
and $\{\!|\alpha_n|\!\}^{M,A[v_n/d_n]}$= 1
and $\{\![[T_{\text{I}}$(:SCOPE)(*u*)$]]\!\}^{M,A[(v_1,...,v_n)/(d_1,...,d_n)]}$= 1,
else 0,

if S(*u*)$\sqsubseteq$ *every*,

such that there are $(v_1, v_2, ..., v_n), (\alpha_1, \alpha_2, ..., \alpha_n), t \in U_{Ty1}$, with
$(v_1, v_2, ..., v_n)$ = T_{I}(:VAR)(*u*),
$(\alpha_1, \alpha_2, ..., \alpha_n)$ = T_{I}(:RESTR)(*u*),
$t = T_{\text{I}}$(:VAR FIRST TYPE)(*u*)
$\{\![[u]]\!\}^{M,A}$= 1 if
for each $d_1, d_2, ..., d_n \in D_{E,W,[t]}$
if $\{\!|\alpha_1|\!\}^{M,A[v_1/d_1]}$= 1 and . . . ,
and $\{\!|\alpha_n|\!\}^{M,A[v_n/d_n]}$= 1,
then $\{\![[T_{\text{I}}$(:SCOPE)(*u*)$]]\!\}^{M,A[(v_1,...,v_n)/(d_1,...,d_n)]}$= 1,
else 0,

(442) (*SR*)
Extensions to Definition 3.12 of Sailer (2003, pp. 125–126 and p. 396)

In the following we ensure that the $Ty1$ objects introduced in addition to Sailer (2003) and their equivalence classes $[u]$ of objects in Γ_{Ty1} have the same extension. Thus:

for each $u \in U_{Ty1}$, such that S(u) $\sqsubseteq$ *list*,
if S(u) $\sqsubseteq$ *elist*, then,
$SR([u]) =$ ( ),
if S(u) $\sqsubseteq$ *nelist*, then,
$SR([u]) = (SR([T_{\text{I}}(\text{:FIRST})(u)]), (SR([T_{\text{I}}(\text{:REST})(u)]))),$
for each $u \in U_{Ty1}$, such that S(u) $\sqsubseteq$ *me*,
if S(u) $\sqsubseteq$ *no*, then,
$SR([u]) = (NO(SR([T_{\text{I}}(\text{:VAR})(u)]))(SR([T_{\text{I}}(\text{:RESTR})(u)]))(SR([T_{\text{I}}(\text{:SCOPE})(u)]))),$
if S(u) $\sqsubseteq$ *some*, then,

$SR([u]) =$

$(SOME(SR([T_{\mathrm{I}}(\text{:VAR})(u)]))(SR([T_{\mathrm{I}}(\text{:RESTR})(u)]))(SR([T_{\mathrm{I}}(\text{:SCOPE})(u)])))$,

if S(u) $\sqsubseteq$ *every*, then,

$SR([u]) =$

$(EVERY(SR([T_{\mathrm{I}}(\text{:VAR})(u)]))(SR([T_{\mathrm{I}}(\text{:RESTR})(u)]))(SR([T_{\mathrm{I}}(\text{:SCOPE})(u)])))$.

(443) (“*”)

Extensions to Definition 3.16 in Sailer (2003, pp. 127–128 and pp. 396–397)

We ensure that the $Ty1$ notation of generalized quantifiers and sequences/ lists of $Ty1$ terms receive an appropriate AVM description when used in the grammar Γ_{Ty1}. Thus:

for each $l \in \mathcal{L}$,

if l = ( ), then

$l^* =$ *elist*,

if l = $(\alpha_{\tau 1}, \alpha_{\tau 2}, ..., \alpha_{\tau n})$, then

$$l^* = \begin{pmatrix} :\sim\textit{nelist} \\ \textbf{and } \alpha^*_{\tau 1}[\text{:FIRST/:}] \\ \textbf{and } (\alpha_{\tau 2}, ...\alpha_{\tau n})^*[\text{:REST/:}] \end{pmatrix}$$

for each $\alpha \in Ty1$,

if $\alpha = (NO(v_{i_1,\tau}, ..., v_{i_n,\tau})(\alpha_{t1}, ...\alpha_{tn})(\beta_t))_t$, then

$$\alpha^* = \begin{pmatrix} :\sim\textit{no} \\ \textbf{and } \text{:TYPE}\sim\textit{truth} \\ \textbf{and } (v_{i_1,\tau}, ..., v_{i_n,\tau})^*[\text{:VAR/:}] \\ \textbf{and } (\alpha_{t1}, ...\alpha_{tn})^*[\text{:RESTR/:}] \\ \textbf{and } \beta^*_t[\text{:SCOPE/:}] \end{pmatrix},$$

if $\alpha = (SOME(v_{i_1,\tau}, ..., v_{i_n,\tau})(\alpha_{t1}, ...\alpha_{tn})(\beta_t))_t$, then

$$\alpha^* = \begin{pmatrix} :\sim\textit{some} \\ \textbf{and } \text{:TYPE}\sim\textit{truth} \\ \textbf{and } (v_{i_1,\tau}, ..., v_{i_n,\tau})^*[\text{:VAR/:}] \\ \textbf{and } (\alpha_{t1}, ...\alpha_{tn})^*[\text{:RESTR/:}] \\ \textbf{and } \beta^*_t[\text{:SCOPE/:}] \end{pmatrix},$$

if $\alpha = (EVERY(v_{i_1,\tau}, ..., v_{i_n,\tau})(\alpha_{t1}, ...\alpha_{tn})(\beta_t))_t$, then

$$\alpha^* = \begin{pmatrix} :\sim\textit{every} \\ \textbf{and } \text{:TYPE}\sim\textit{truth} \\ \textbf{and } (v_{i_1,\tau}, ..., v_{i_n,\tau})^*[\text{:VAR/:}] \\ \textbf{and } (\alpha_{t1}, ...\alpha_{tn})^*[\text{:RESTR/:}] \\ \textbf{and } \beta^*_t[\text{:SCOPE/:}] \end{pmatrix}.$$

Bibliography

Abeillé, Anne and Danièle Godard (1997), The syntax of French negative adverbs, *in* D.Forget, P.Hirschbühler, F.Martineau and M.-L.Rivero, eds, 'Negation and Polarity: Syntax and Semantics', John Benjamins, Amsterdam, pp. 1–17.

Abeillé, Anne and Danièle Godard (2000), Varieties of *esse* in Romance languages, *in* D.Flickinger and A.Kathol, eds, 'Proceedings of the 7th International Conference on Head-Driven Phrase Structure Grammar', CSLI Publications, Stanford, pp. 2–22.

Abeillé, Anne and Danièle Godard (2003), Les prédicats complexes, *in* D.Godard, ed., 'Grammaire des Langues Romanes', CNRS, Paris, pp. 125–184.

Abeillé, Anne, Danièle Godard and Ivan A. Sag (1998), Two kinds of composition in French complex predicates, *in* E.Hinrichs, A.Kathol and T.Nakazawa, eds, 'Complex Predicates in Nonderivational Syntax', Academic Press, New York, pp. 1–41.

Acquaviva, Paolo (1997), *The Logical Form of Negation: A Study of Operator-Variable Structures in Syntax*, Garland, New York and London. Revised Ph.D. Dissertation, Scuola Normale Superiore di Pisa, 1993.

Ajdukiewicz, Kazimierz (1935), 'Die syntaktische Konexität', *Studia Philosophica* **1**, 1–27. Translated in S. McCall, ed. (1967), *Polish Logic: 1920-1939*, Oxford: Clarendon Press, pp. 207–231.

Alexiadou, Artemis and Elena Anagnostopoulou (2008), Structuring participles, *in* C.Chang and H.Haynie, eds, 'Proceedings of the 26th West Coast Conference on Formal Linguistics', Cascadilla Proceedings Project, Somerville, pp. 33–41.

Avram, Mioara (1986), *Gramatica pentru toţi (The Grammar for Everybody)*, Editura Academiei Române, Bucharest.

Bach, Emmon, Eloise Jelinek, Angelika Kratzer and Barbara H. Partee, eds (1995), *Quantification in Natural Languages*, Vol. 1–2, Kluwer, Dordrecht.

Barbu, Ana-Maria (1999), 'Complexul verbal (The verbal complex)', *Studii şi cercetări lingvistice* **1**, 39–84.

Barbu, Ana-Maria (2004), The negation NU: lexical or affixal item, *in* E.Ionescu, ed., ‘Understanding Romanian Negation. Syntactic and Semantic Approaches in a Declarative Perspective’, Bucharest University Press, pp. 68–82.

Barbu Mititelu, Verginica and Roxana Maftei Ciolăneanu (2004), The main aspects of the grammar of negation in Romanian, *in* E.Ionescu, ed., ‘Understanding Romanian Negation. Syntactic and Semantic Approaches in a Declarative Perspective’, Bucharest University Press, pp. 32–67.

Barbu, Verginica (2003), Some aspects of negation in Romanian. MA Thesis, University of Bucharest.

Barwise, Jon and John Perry (1983), *Situations and Attitudes*, MIT Press, Cambridge, Mass.

Barwise, Jon and Robin Cooper (1981), ‘Generalized quantifiers and natural language’, *Linguistics and Philosophy* **4**, 159–219.

Bateman, Nicoleta and Maria Polinsky (2006), Romanian as a two-gender language. Manuscript, University of California at San Diego; presented at the LSA meeting, Albuquerque, 2005.

Bech, Gunnar (1955/1957), *Studien über das deutsche verbum infinitum*, Det Kongelige Danske Akademie av Videnskaberne, Kobenhavn.

Ben-Shalom, Dorit (1994), A tree characterization of generalized quantifier reducibility, *in* M.Kanazawa and C. J.Pinón, eds, ‘Dynamics, Polarity and Quantification’, Vol. 48 of *CSLI Lecture Notes*, CSLI Publications, pp. 147–171.

Bîlbîie, Gabriela (2008), Towards a fragmentary analysis of gapping in Romanian. Talk at the 24th annual meeting of the Israel Association for Theoretical Linguistics, October 26–27.

Blackburn, Patrick and Johan Bos (2005), *Representation and Inference for Natural Language. A First Course in Computational Semantics*, CSLI Publications, Stanford.

Błaszczak, Joanna (1999), Investigations into the Interaction between the Indefinites and Negation, PhD thesis, Humboldt Universität Berlin. Published: Akademieverlag, Berlin, Studia grammatica 51, 2001.

Borsley, D. Robert (1987), Subjects and complements in HPSG, Technical Report CSLI-107-87, CSLI, Stanford.

Bouma, Gosse, Robert Malouf and Ivan A. Sag (2001), ‘Satisfying constraints on extraction and adjunction’, *Natural Language and Linguistic Theory* **19.1**, 1–65.

Bresnan, Joan, ed. (1982), *The Mental Representation of Grammatical Relations*, MIT Press, Cambridge, Mass.

Carpenter, Bob (1997), *Type-Logical Semantics*, MIT Press, Cambridge, Mass and London, UK.

Chomsky, Noam (1981), *Lectures on Government and Binding*, Foris Publications, Dordrecht.

Church, Alonzo (1940), 'A formulation of the simple theory of types', *Journal of Symbolic Logic* **5**, 56–68.

Cinque, Guglielmo (1990), *Types of A'-Dependencies*, Vol. 17 of *Linguistic Inquiry Monograph*, MIT Press, Cambridge, Mass.

Ciompec, Georgeta (1985), *Morfosintaxa adverbului românesc. Sincronie şi diacronie ('The morphosyntax of the Romanian adverb. Synchrony and diachrony')*, Editura Ştiinţifică şi Encliclopedică, Bucureşti.

Clark, Robin and Edward L. Keenan (1987), 'The absorption operator and universal grammar', *The Linguistic Review* **5**, 113–136.

CoDII (2008), 'The collection of distributionally idiosyncratic items', online database. Project A5, *Distributional Idiosyncrasies*, Collaborative Research Centre SFB 441, University of Tübingen. **URL:** *http://www.sfb441.uni-tuebingen.de/a5/codii/*

Cooper, Robin (1983), *Quantification and syntactic theory*, Reidel, Dordrecht.

Cooper, Robin (1997), Austinian propositions, Davidsonian events and perception complements, *in* J.Ginzburg et al., eds, 'The Tbilisi Symposium on Logic, Language and Computation: Selected Papers', CSLI Publications, Stanford, pp. 19–34.

Corblin, Francis (1995), 'Compositionality and complexity in multiple negation', *IGPL Bulletin* **3 (3-2)**, 449–473.

Corblin, Francis and Lucia M. Tovena (2001), On the multiple expression of negation in Romance, *in* Y.D'Hulst, J.Rooryck and J.Schroten, eds, 'Romance Languages and Linguistic Theory 1999', John Benjamins, Amsterdam, pp. 87–115.

Corblin, Francis, Viviane Déprez, Henriëtte de Swart and Lucia Tovena (2004), Negative concord, *in* F.Corblin and H.de Swart, eds, 'Handbook of French Semantics', CSLI Publications, Stanford, pp. 427–461.

Cornilescu, Alexandra (1997), 'The double subject constructions in Romanian. Notes on the syntax of the subject', *Revue Roumaine de Linguistique* **3-4**, 1–45. Reprinted as Cornilescu (2000*a*).

Cornilescu, Alexandra (2000*a*), The double subject constructions in Romanian, *in* V.Motapanyane, ed., 'Comparative Studies in Romanian Syntax', Elsevier, pp. 83–135.

Cornilescu, Alexandra (2000*b*), Notes on the interpretation of the prepositional accusative in Romanian, *in* 'Bucharest Working Papers in Linguistics', Vol. II: 1, University of Bucharest, Bucharest, pp. 91–106.

Cornilescu, Alexandra (2002), At the Romanian left periphery, *in* ‘Bucharest Working Papers in Linguistics’, Vol. 4, University of Bucharest, Bucharest, pp. 89–107.

Curry, Haskell B. (1930), ‘Grundlagen der kombinatorischen Logik’, *American Journal of Mathematics* **52**, 509–536, 789–834.

De Kuthy, Kordula (2002), *Discontinuous NPs in German: a case study of the interaction of syntax, semantics, and pragmatics*, CSLI Publications, Stanford.

De Kuthy, Kordula and Detmar Meurers (2006), Tutorial on the interface of syntax and information structure from an HPSG perspective. CoGETI (Scientific Network on Constraint-Based Grammar) Workshop, Göttingen.

de Swart, Henriëtte (1993), *Adverbs of Quantification: A Generalized Quantifier Approach*, Garland, New York and London.

de Swart, Henriëtte (1999), Negation and negative concord in a polyadic quantification framework, *in* J.Gerbrandy, M.Marx, M.de Rijke and Y.Venema, eds, ‘JFAK. Essays dedicated to Johan van Benthem on the occasion of his 50th birthday’, Amsterdam University Press, Amsterdam. http://www.illc.uva.nl/j50/contribs/deswart/index.html.

de Swart, Henriëtte (2000), Scope ambiguities with negative quantifiers, *in* K.von Heusinger and U.Egli, eds, ‘Reference and anaphoric relations’, Kluwer Academic Press, Dordrecht, pp. 118–142.

de Swart, Henriëtte (2010), *Expression and interpretation of negation. An OT typology*, Springer, Dordrecht.

de Swart, Henriëtte and Ivan A. Sag (2002), ‘Negation and negative concord in Romance’, *Linguistics and Philosophy* **25**, 373–417.

Dekker, Paul (2003), ‘Meanwhile, within the Frege boundary’, *Linguistics and Philosophy* **26**, 547–556.

den Besten, Hans (1986), Double negation and the genesis of Afrikaans, *in* P.Muysken and N.Smith, eds, ‘Substrata versus Universals in Creole Languages’, John Benjamins, Amsterdam, pp. 185–230.

Déprez, Viviane (1997), A non-unified analysis of negative concord, *in* D.Forget et al., eds, ‘Negation and Polarity: Syntax and Semantics’, John Benjamins, Amsterdam, pp. 53–74.

Diesing, Molly (1992), *Indefinites*, MIT Press, Cambridge, Mass.

Dobrovie-Sorin, Carmen (1994), *The Syntax of Romanian: Comparative Studies in Romance*, Mouton de Gruyter, Berlin. In Romanian translation, *Sintaxa limbii române*, Editura Univers, Bucureşti, 2000.

Doetjes, Jenny and Martin Honcoop (1997), The semantics of event-related readings: A case for pair-quantification, *in* A.Szabolsci, ed., 'Ways of scope taking', Kluwer Academic Publishers, Dordrecht, pp. 263–310.

Dowty, David (2007), Compositionality as an empirical problem, *in* C.Barker and P.Jacobson, eds, 'Direct Compositionality', Oxford University Press, Oxford, pp. 23–101.

Dowty, David and Belinda Brodie (1984), The semantics of "floated" quantifiers in a transformationless grammar, *in* 'Proceedings of the West Coast Conference on Formal Linguistics 3', CSLI Publications, Stanford, pp. 75–90.

Dowty, David R., Robert E. Wall and Stanley Peters (1981), *Introduction to Montague Semantics*, Reidel, Dordrecht.

Ebert, Christian, Cornelia Endriss and Hans-Martin Gärtner (2007), 'An information structural account of German integrated verb second clauses', *Research on Language & Computation. Special Issue on Semantic Compositionality* **5: 4**, 415–434.

Engdahl, Elisabet and Enruic Vallduví (1996), Information packaging in HPSG, *in* C.Grover and E.Vallduví, eds, 'Studies in HPSG', Vol. 12 of *Edinburgh Working Papers in Cognitive Science*, University of Edinburgh, Edinburgh.

Espinal, M. Teresa (2000), 'On the semantic status of n-words in Catalan and Spanish', *Lingua* **110**, 557–580.

Farkas, Donka (1985), *Intensional Descriptions and the Romance Subjunctive Mood*, Garland Publishers, New York.

Farkas, Donka F. (2002), Extreme non-specificity in Romanian, *in* C.Beyssade, R.Bok-Bennema, F.Drijkoningen and P.Monachesi, eds, 'Romance Languages and Linguistic Theory 2000', John Benjamins, Amsterdam, pp. 127–151.

Frege, Gottlob (1893), *Grundgesetze der Arithmetik*, Hermann Pohle, Jena.

Fălăuş, Anamaria (2007), Le paradoxe de la double négation dans une langue à concordance négative stricte, *in* F.Floricic, ed., 'La négation dans les langues romanes', John Benjamins, pp. 75–97.

Fukushima, Kazuhiko (1993), 'Model theoretic semantics for Japanese floating quantifiers and their scope properties', *Journal of East Asian Linguistics* **2**, 213–228.

GA (1966), *Gramatica limbii române (The Grammar of Romanian)*, second edn, Editura Academiei Române, Bucharest.

Gallin, Daniel (1975), *Intensional and Higher-Order Modal Logic*, North Holland, Amsterdam.

Gamut, L.T.F. (1991), *Logic, language, and meaning*, University of Chicago Press, Chicago.

Gazdar, Gerald, Ewan Klein, Geoffrey K. Pullum and Ivan A. Sag (1985), *Generalized Phrase Structure Grammar*, Blackwell and Harvard University Press, Oxford and Cambridge, Mass.

Geurts, Bart (1996), 'On no', *Journal of Semantics* **13**, 67–86.

Geurts, Bart (1998), 'The mechanisms of denial', *Language* **74**, 274–307.

Giannakidou, Anastasia (1998), *Polarity Sensitivity as (Non)Veridical Dependency*, Vol. 23 of *Linguistik Aktuell*, John Benjamins, Amsterdam.

Giannakidou, Anastasia (2000), 'Negative ... concord?', *Natural Language and Linguistic Theory* **18**, 457–523.

Giannakidou, Anastasia (2006), N-words and negative concord, *in* 'Linguistics Companion', Blackwell, Oxford, pp. 327–391.

Ginzburg, Jonathan and Ivan A. Sag (2000), *Interrogative Investigations. The Form, Meaning, and Use of English Interrogatives*, CSLI Publications, Stanford.

Göbbel, Edward (1995), On the syntax of topic and focus in Romanian, *in* I.Kohlhof, S.Winkler and H. B.Drubig, eds, 'Proceedings of the Göttingen Focus Workshop. 17. DGfS, March 1–3, 1995', Arbeitsbericht Nr. 69 des Sonderforschungsbereichs 340, Universität Tübingen, pp. 261–301.

Göbbel, Edward (2003), *Syntactic and focus-structural aspects of triadic constructions*, Peter Lang, Frankfurt am Main.

Groenendijk, Jereon and Martin Stokhof (1997), Questions, *in* J.van Benthem and A.ter Meulen, eds, 'Handbook of Logic and Language', Elsevier and MIT Press, Amsterdam and Cambridge, Mass, pp. 1055–1124.

Haegeman, Liliane (1995), *The Syntax of Negation*, Cambridge University Press, Cambridge.

Haegeman, Liliane and Raffaella Zanuttini (1991), 'Negative heads and the Neg criterion', *The Linguistic Review* **8**, 233–251.

Haegeman, Liliane and Raffaella Zanuttini (1996), Negative concord in West Flemish, *in* A.Belletti and L.Rizzi, eds, 'Parameters and Functional Heads: Essays in Comparative Syntax', Oxford University Press, New York and Oxford, pp. 117–180.

Halvorsen, Per-Kristian and William A. Ladusaw (1979), 'Montague's 'Universal Grammar': An introduction for the linguist', *Linguistics and Philosophy* **3**, 185–223.

Hamm, Fritz (1989), *Natürlich-sprachliche Quantoren: modelltheoretische Untersuchungen zu universellen semantischen Beschränkungen*, Niemeyer, Tübingen.

Haspelmath, Martin (2005), Negative indefinite pronouns and predicate negation, *in* M.Haspelmath, M.Dryer, D.Gil and B.Comrie, eds, 'The World Atlas of Language Structures', Oxford University Press, Oxford, pp. 466–469.

Heim, Irene (1982), The Semantics of Definite and Indefinite Noun Phrases, PhD thesis, University of Massachusetts, Amherst.

Hendriks, Herman (1993), *Studied Flexibility: Categories and Types in Syntax and Semantics*, ILLC Dissertation Series 1993: 5, Universiteit van Amsterdam.

Herburger, Elena (2001), 'The negative concord puzzle revisited', *Natural Language Semantics* **9**, 289–333.

Higginbotham, James (1995), The semantics of questions, *in* S.Lapin, ed., 'Handbook of Contemporary Semantic Theory', Blackwell, Oxford, pp. 361–384.

Higginbotham, James and Robert May (1981), 'Questions, quantifiers and crossing', *The Linguistic Review* **1**, 41–79.

Hindley, J. Roger and Johnathan P. Seldin (2008), *Lambda-calculus and combinators: an introduction*, Cambridge University Press, New York.

Höhle, Tilman N. (1999), An architecture for phonology, *in* R.Borsley and A.Przepiórkowski, eds, 'Slavic in HPSG', CSLI Publications, Stanford, pp. 261–301.

Horn, Laurence (1985), 'Metalinguistic negation and pragmatic ambiguity', *Language* **61**, 121–174.

Horn, Laurence (1989), *A Natural History of Negation*, University of Chicago Press, Chicago and London.

Horn, Laurence R. (1997), 'All John's children are as bald as the king of France: Existential import and the geometry of opposition', *Chicago Linguistics Society* **33**, 155–179.

Ionescu, Emil (1999), A quantification-based approach to negative concord in Romanian, *in* G.Kruijff and R.Oehrle, eds, 'Proceedings of Formal Grammar 1999', Utrecht, pp. 25–35.

Ionescu, Emil (2001), 'On the status of PE in the direct object construction in Romanian', *Romanian Journal of Information Science and Technology* **4**, 293–310.

Ionescu, Emil (2004), The semantic status of Romanian n-words in negative concord constructions, *in* E.Ionescu, ed., 'Understanding Romanian Negation. Syntactic and Semantic Approaches in a Declarative Perspective', Bucharest University Press, pp. 83–118.

Iordăchioaia, Gianina (2004), N-words as negative quantifiers in Romanian, *in* E.Ionescu, ed., 'Understanding Romanian Negation. Syntactic and Semantic Approaches in a Declarative Perspective', Bucharest University Press, pp. 119–150.

Isac, Daniela (1998), Sentence negation in English and Romanian. A syntactic and semantic approach, PhD thesis, University of Bucharest.

Isac, Daniela (2004), Focus on negative concord, *in* R.Bok-Bennema, B.Hollenbrandse, B.Kampers-Manhe and P.Sleeman, eds, 'Romance Languages and Linguistic Theory 2002: Selected papers from 'Going Romance', Groningen, 28–30 November 2002', pp. 119–140.

Jacobs, Joachim (1980), 'Lexical decomposition in Montague grammar', *Thoeretical Linguistics* **7**, 121–136.

Jacobs, Joachim (1991), Negation, *in* A.von Stechow and D.Wunderlich, eds, 'Semantics: An International Handbook', Mouton de Gruyter, Berlin, pp. 560–596.

Janssen, Theo M. V. (1986), *Foundations and applications of Montague grammar. Part 1: Philosophy, framework, computer science, Part 2: Applications to natural language*, Centrum voor Wiskunde en Informatica, Amsterdam. CWI tracts 19 and 28.

Janssen, Theo M.V. (1997), Compositionality, *in* J.van Benthem and A.ter Meulen, eds, 'Handbook of Logic and Language', Elsevier and MIT Press, Amsterdam and Cambridge, Mass, pp. 417–473.

Jespersen, Otto (1917), *Negation in English and other languages*, Hoest, Kopenhagen.

Karttunen, Lauri (1977), 'Syntax and semantics of questions', *Linguistics and Philosophy* **1**, 3–44.

Kathol, Andreas (1995), Linearization-Based German Syntax, PhD thesis, Ohio State University.

Keenan, Edward L. (1987), Unreducible n-ary quantifiers in natural language, *in* P.Gardenfors, ed., 'Generalized Quantifiers', Reidel, Dordrecht, pp. 109–150.

Keenan, Edward L. (1992), 'Beyond the Frege boundary', *Linguistics and Philosophy* **15**, 199–221.

Keenan, Edward L. (1996), Further beyond the Frege boundary, *in* J.van der Does and J.van Eijck, eds, 'Quantifiers, Logic, and Language', CSLI Publications, Stanford, pp. 179–201.

Keenan, Edward L. and Dag Westerståhl (1997), Generalized quantifiers in linguistics and logic, *in* J.van Benthem and A.ter Meulen, eds, 'Handbook of Language and Logic', Elsevier Science, Amsterdam, pp. 837–893.

Keller, William R. (1988), Nested Cooper storage: The proper treatment of quantification in ordinary noun phrases, *in* U.Reyle and C.Rohrer, eds, 'Natural Language Parsing and Linguistic Theories', Reidel, Dordrecht, pp. 432–447.

Kim, Jong-Bok and Ivan A. Sag (2002), 'Negation without head movement', *Natural Language and Linguistic Theory* **20.2**, 339–412.

Klima, Edward (1964), Negation in English, *in* J. A.Fodor and J. J.Katz, eds, 'The Structure of Language', Prentice Hall, Englewood Cliffs, pp. 246–323.

Kratzer, Angelika (1995), Stage-level and individual-level predicates, *in* G.Carlson and F.Pelletier, eds, 'The generic book', University of Chicago Press, Chicago, pp. 125–175.

Kratzer, Angelika (2000), Building statives, *in* L.Conathan et al., eds, 'Proceedings of the 26th Annual Meeting of the Berkeley Linguistics Society', Berkeley, pp. 385–399.

Krifka, Manfred (1990), 'Four thousand ships passed through the lock: Object-induced measure functions on events', *Linguistics and Philosophy* **13**, 487–520.

Kuhn, Jonas (1995), Information packaging in German – some motivation from HPSG-based translation. Manuscript: Centre for Cognitive Science, University of Edinburgh.

Kupść, Anna (2000), An HPSG Grammar of Polish Clitics, PhD thesis, Institute of Computer Science, Polish Academy of Sciences, and Université Paris 7.

Kupść, Anna and Adam Przepiórkowski (2002), Morphological aspects of verbal negation in Polish, *in* P.Kosta and J.Frasek, eds, 'Current Approaches to Formal Slavic Linguistics', Peter Lang, Frankfurt, pp. 337–346.

Labov, William (1972), 'Negative attraction and negative concord in English grammar', *Language* **48**(4), 773–818.

Ladusaw, William A. (1980), *Polarity Sensitivity as Inherent Scope Relations*, Garland Press, New York.

Ladusaw, William A. (1992), Expressing negation, *in* 'Proceedings of SALT 2', Columbus: The Ohio State University, pp. 237–259.

Laka, Miren Itziar (1990), Negation in Syntax: On the Nature of Functional Categories and Projections, PhD thesis, MIT.

Lewis, D. (1975), Adverbs of quantification, *in* E.Keenan, ed., 'Formal Semantics of Natural Language', Cambridge University Press, Cambridge, pp. 3–15.

Lindström, Per (1966), 'First order predicate logic with generalized quantifiers', *Theoria* **32**, 186–195.

Linebarger, Marcia (1980), The Grammar of Negative Polarity, PhD thesis, MIT.

Maekawa, Takafumi (2004), Constituency, word order and focus projection, *in* S.Müller, ed., 'Proceedings of the HPSG04 Conference', CSLI Publications, Centre for Computational Linguistics Katholieke Universiteit Leuven, pp. 168–188.

Manning, Christopher, Ivan A. Sag and Masayo Iida (1999), The lexical integrity of Japanese causatives, *in* R.Levine and G.Green, eds, 'Readings in Modern Phrase Structure Grammar', Cambridge Univeristy Press, pp. 39–79.

Martin-Löf, Per (1984), *Intuitionistic Type Theory*, Bibliopolis, Napoli. Lecture notes by Giovanni Sambin.

Mathieu, Eric (2001), 'On the nature of French n-words', *University College, London (UCL) Working Papers in Linguistics* **13**, 319–353.

May, Robert (1989), 'Interpreting logical form', *Linguistics and Philosophy* **12**, 387–435.

McCawley, James D. (1991), 'Contrastive negation and metalinguistic negation', *Chicago Linguistic Society. The Parasession on Negation* **27.2**, 189–206.

Merchant, Jason (2001), *The syntax of silence*, Oxford University Press, Oxford.

Meurers, Detmar (1999), Lexical Generalizations in the Syntax of German Non-Finite Constructions, PhD thesis, University of Tübingen. Published in 2000 as Volume 145 in Arbeitspapiere des SFB 340.

Miller, Philip H. and Ivan A. Sag (1997), 'French clitic movement without clitics or movement', *Natural Language and Linguistic Theory* **15**, 573–639.

Milsark, Gary (1974), Existential Sentences in English, PhD thesis, MIT.

Moltmann, Friederike (1995), 'Exception sentences and polyadic quantification', *Linguistics and Philosophy* **18**, 223–280.

Moltmann, Friederike (1996), Resumptive quantifiers in exception sentences, *in* M.Kanazawa et al., eds, 'Quantifiers, Deduction, and Context. Proceedings of the Workshop on Logic, Language and Computation', pp. 139–170.

Monachesi, Paola (1996), A grammar of Italian clitics, PhD thesis, Tilburg University.

Monachesi, Paola (1998), The morphosyntax of Romanian cliticization, *in* P.Coppen, H.van Halteren and L.Teunissen, eds, 'Proceedings of Computational Linguistics in The Netherlands 1997', Rodopi, Amsterdam and Atlanta, pp. 99–118.

Monachesi, Paola (1999), *A lexical approach to Italian cliticization*, CSLI Publications, Stanford.

Monachesi, Paola (2000), Clitic placement in the Romanian verbal complex, *in* B.Gerlach and J.Grijzenhout, eds, 'Clitics in phonology, morphology, and syntax', John Benjamins, Amsterdam, pp. 255–294.

Montague, Richard (1970), 'Universal grammar', *Theoria* **36**, 373–398. Reprinted in Thomason (1974), pp. 222–246.

Montague, Richard (1973), The proper treatment of quantification in ordinary English, *in* J.Hintikka, J.Moravcisk and P.Suppes, eds, 'Approaches to Natural Language', Reidel, Dordrecht. Reprinted in Thomason (1974), pp. 247–270.

Mostowski, Andrzej (1957), 'On a generalization of quantifiers', *Fundamenta Mathematica* **44**, 12–36.

Nakanishi, Kimiko (2005), Semantic properties of split topicalization in German, *in* C.Maienborn and A.Wölstein, eds, 'Event Arguments: Foundations and Applications', Max Niemeyer Verlag, Tübingen, pp. 331–356.

Niculescu, Alexandru (1965), *Individualitatea limbii române între limbile romanice (The inidividuality of Romanian among Romance languages)*, Editura Ştiinţifică, Bucureşti.

Parsons, Terence (1991), *Events in the Semantics of English: A Study in Subatomic Semantics*, MIT Press, Cambridge, Mass.

Partee, Barbara H. (1984), Compositionality, *in* F.Landman and F.Veltman, eds, 'Varieties of Formal Semantics: Proceedings of the 4th Amsterdam Colloquium, Sept. 1982', Foris Publications, Dordrecht, pp. 281–311.

Partee, Barbara H. (1987), Noun phrase interpretation and type-shifting principles, *in* J.Groenendijk et al., eds, 'Studies in Discourse Representation Theory and the Theory of Generalized Quantifiers', Foris Publications, Dordrecht, pp. 115–143.

Partee, Barbara H., Alice ter Meulen and Robert E. Wall (1990), *Mathematical Methods in Linguistics*, Kluwer, Dordrecht.

Penka, Doris (2006), Almost there: The meaning of 'almost', *in* C.Ebert and C.Endriss, eds, 'Proceedings of Sinn und Bedeutung', Vol. 10, ZAS, Berlin, pp. 89–107.

Penka, Doris (2007), Negative Indefinites, PhD thesis, Eberhard Karls Universität Tübingen.

Penka, Doris and Arnim von Stechow (2001), Negative Indefinita unter Modalverben, *in* R.Müller and M.Reis, eds, 'Modalität und Modalverben im Deutschen', Linguistische Berichte, Sonderheft 9.

Penn, Gerald (1998), Parametric types for typed attribute-value logic, *in* 'Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics (COLING/ ACL-98)', Vol. 2, pp. 1027–1033.

Penn, Gerald B. (2000), The Algebraic Structure of Attributed Type Signatures, PhD thesis, Carnegie Mellon University.

Penn, Gerald and Frank Richter (2004), Lexical resource semantics: From theory to implementation, *in* S.Müller, ed., 'Proceedings of the HPSG04 Conference', CSLI Publications, Stanford, pp. 423–443.

Peres, João (1997), Extending the notion of negative concord, *in* D.Forget et al., eds, 'Negation and Polarity', John Benjamins, Amsterdam.

Peters, Stanley and Dag Westerståhl (2002), Does English really have resumptive quantification? and do 'donkey' sentences really express it?, *in* D.Beaver, L. D.Casillas Martınez, B. Z.Clark and S.Kaufmanns, eds, 'The Construction of Meaning', CSLI Publications, pp. 181–195.

Peters, Stanley and Dag Westerståhl (2006), *Quantifiers in Language and Logic*, Oxford University Press, Oxford and New York.

Pollard, Carl and Ivan A. Sag (1994), *Head-driven Phrase Structure Grammar*, University of Chicago Press, Chicago.

Pollard, Carl J. and Eun Jung Yoo (1998), 'A unified theory of scope for quantifiers and *wh*-phrases', *Journal of Linguistics* **34**, 415–445.

Pollock, Jean-Yves (1989), 'Verb movement, UG and the structure of IP', *Linguistic Inquiry* **20 (3)**, 365–424.

Progovac, Ljiljana (1994), *Positive and Negative Polarity: A Binding Approach*, Cambridge University Press, Cambridge.

Przepiórkowski, Adam (1998), 'A Unified Theory of Scope' revisited: Quantifier retrieval without spurious ambiguities, *in* G.Bouma, G.-J.Kruijff and R.Oehrle, eds, 'Proceedings of FHCG'98 (Joint Conference on Formal Grammar, Head-driven Phrase Structure Grammar, and Categorial Grammar)', Saarbrücken, Germany, pp. 185–195.

Przepiórkowski, Adam (1999*a*), Negative polarity questions and Italian negative concord, *in* V.Kordoni, ed., 'Tübingen Studies in Head-Driven Phrase Structure Grammar', Arbeitspapiere des Sonderforschungsbereichs 340 Bericht Nr. 132, Seminar für Sprachwissenschaft, Universität Tübingen, pp. 353–400.

Przepiórkowski, Adam (1999*b*), On negative eventualities, negative concord, and negative *yes/no* questions, *in* T.Matthews and D.Strolovitch, eds, 'Proceeding of Semantics and Linguistic Theory 9', CLC Publications, Ithaca, NY, pp. 237–254.

Przepiórkowski, Adam and Anna Kupść (1997), Negative concord in Polish, Technical report, Institute of Computer Science, Polish Academy of Sciences.

Przepiórkowski, Adam and Anna Kupść (1999), Eventuality negation and negative concord in Polish and Italian, *in* R.Borsley and A.Przepiórkowski, eds, 'Slavic in HPSG', CSLI Publications, Stanford, pp. 211–246.

Ranta, Aarne (1991), 'Intuitionistic categorial grammar', *Linguistics and Philosophy* **14**, 203–239.

Ranta, Aarne (1994), *Type Theoretical Grammar*, Oxford University Press, Oxford.

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', CSLI Publications, Stanford, pp. 151–197.

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', Arbeitsbericht Nr. 95 des Sonderforschungsbereichs 340, Universität Tübingen, pp. 13–188.

Richter, Frank (2004*a*), Foundations of Lexical Resource Semantics. Habilitationsschrift, University of Tübingen.

Richter, Frank (2004*b*), A Mathematical Formalism for Linguistic Theories with an Application in Head-Driven Phrase Structure Grammar, PhD thesis (2000), Eberhard Karls Universität Tübingen.

Richter, Frank (2007), Closer to the truth: A new model theory for HPSG, *in* J.Rogers and S.Kepser, eds, 'Model-Theoretic Syntax at 10', pp. 99–108.

Richter, Frank and Laura Kallmeyer (2007), Feature logic-based semantic composition: A comparison between LRS and LTAG, *in* A.Søgaard and P.Haugereid, eds, 'Postproceedings of the 1st International Workshop on Typed Feature Structure Grammars', Vol. 7 of *CST Working Papers*, pp. 31–83.

Richter, Frank and Manfred Sailer (1999*a*), 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 Sonderforschungsbereichs 340 Bericht Nr. 132, Seminar für Sprachwissenschaft, Universität Tübingen, pp. 231–300.

Richter, Frank and Manfred Sailer (1999*b*), LF conditions on expressions of Ty2: An HPSG analysis of negative concord in Polish, *in* R.Borsley and A.Przepiórkowski, eds, 'Slavic in HPSG', CSLI Publications, Stanford, pp. 247–282.

Richter, Frank and Manfred Sailer (2004), Basic concepts of lexical resource semantics, *in* A.Beckmann and N.Preining, eds, 'ESSLLI 2003 – Course Material I', Vol. 5 of *Collegium Logicum*, Kurt Gödel Society Wien, pp. 87–143.

Richter, Frank and Manfred Sailer (2006), Modeling typological markedness in semantics: The case of negative concord, *in* S.Müller, ed., 'Proceedings of the HPSG06 Conference', CSLI Publications, Varna, pp. 305–325.

Richter, Frank and Manfred Sailer (2008), Simple trees with complex semantics: On epistemic modals and strong quantifiers, *in* M.Romero, ed., 'What syntax feeds semantics?', ESSLLI 2008, pp. 70–81. Workshop organized as part of the 20th European Summer School in Logic, Language and Information in Hamburg, Germany.

Rizzi, Luigi (1991), Residual verb second and the WH criterion. ms., Université de Genève, published as Rizzi (1996).

Rizzi, Luigi (1996), Residual verb second and the WH criterion, *in* A.Belletti and L.Rizzi, eds, 'Parameters and Functional Heads: Essays in Comparative Syntax', Oxford University Press, New York and Oxford.

Rizzi, Luigi (1997), The fine structure of the left periphery, *in* L.Haegeman, ed., 'Elements of Grammar', Kluwer, Dordrecht, pp. 281–337.

Roberts, Craige (1989), 'Modal subordination and pronominal anaphora in discourse', *Linguistics and Philosophy* **12**, 683–721.

Sag, Ivan A. (1997), 'English relative clause constructions', *Journal of Linguistics* **33**, 431–483.

Sag, Ivan A. (to appear), Rules and exceptions in the English auxiliary system. To appear in *Journal of Linguistics*.

Sag, Ivan A. and Carl Pollard (1987), *Information-based syntax and semantics*, CSLI Lecture Notes 13, Stanford.

Sag, Ivan A., Thomas Wasow and Emily M. Bender (2003), *Syntactic Theory: A formal introduction, Second Edition*, CSLI Publications, Stanford.

Sailer, Manfred (2003), *Combinatorial Semantics and Idiomatic Expressions in Head-Driven Phrase Structure Grammar*, Vol. 161, Arbeitspapiere des SFB 340, Universität Stuttgart and Universität Tübingen.

Sailer, Manfred (2004), Local semantics in head-driven phrase structure grammar, *in* O.Bonami and P. C.Hofherr, eds, 'Empirical Issues in Syntax and Semantics', Vol. 5, http://www.cssp.cnrs.fr/eiss5, pp. 197–214.

Sailer, Manfred (2006), Neg-raising in an underspecified semantics framework, *in* O.Bonami and P.Cabredo Hofherr, eds, 'Empirical Issues in Syntax and Semantics 6', Paris, pp. 375–403.

Schönfinkel, Moses (1924), 'Über die Bausteine der mathematischen Logik', *Mathematische Annalen* **92**, 305–316.

Seuren, Pieter A. M. (1988), 'Presupposition and negation', *Journal of Semantics* **6**, 175–226.

Sportiche, Dominique (1988), 'A theory of floating quantifiers and its corollaries for constituent structure', *Linguistic Inquiry* **19**, 425–449.

Sundholm, Göran (1989), 'Constructive generalized quantifiers', *Synthese* **79**, 1–12.

Suñer, Margarita (1995), 'Negative elements, island effects and resumptive *no*', *The Linguistic Review* **12**, 233–373.

Thomason, Richmond (1974), *Formal Philosophy: Selected Papers of Richard Montague*, Yale University Press, New Haven, Conn.

Tovena, Lucia M. (2003), Distributional restrictions on negative determiners, *in* K.Jaszczolt and K.Turner, eds, 'Meaning Through Language Contrast', John Benjamins, Amsterdam, pp. 3–28.

Tubau, Susagna (2008), *Negative concord in English and Romance: Syntax-Morphology interface conditions on the expression of negation*, LOT Publications, Utrecht.

van Benthem, Johan (1986*a*), *Essays in Logical Semantics*, Reidel, Dordrecht.

van Benthem, Johan (1986*b*), 'The relational theory of meaning', *Logique et analyse* **29**, 251–273.

van Benthem, Johan (1989), 'Polyadic quantifiers', *Linguistics and Philosophy* **12**, 437–464.

Van der Sandt, Rob (1991), 'Denial', *Chicago Linguistic Society. The Parasession on Negation* **27.2**, 331–344.

van der Wouden, Ton (1994), Polarity and 'illogical negation', *in* M.Kanazawa and C. J.Pinon, eds, 'Dynamics, Polarity, and Quantification', CSLI Publications, Stanford, pp. 17–45.

van der Wouden, Ton (1997), *Negative Contexts. Collocation, polarity and multiple negation*, Routledge, London and New York.

van Eijck, Jan (2005), 'Normal forms for characteristic functions on *n*-ary relations', *Journal of Logic and Computation* **15**(2), 85–98.

Watanabe, Akira (2004), 'The genesis of negative concord: Syntax and morphology of negative doubling', *Linguistic Inquiry* **35**, 559–612.

Westerståhl, Dag (1989), Quantifiers in formal and natural languages, *in* D.Gabbay and F.Guenthner, eds, 'Handbook of Philosophical Logic', Vol. IV, Reidel, Dordrecht, pp. 1–131.

Westerståhl, Dag (1994), Iterated quantifiers, *in* M.Kanazawa and C. J.Piñón, eds, ‘Dynamics, Polarity and Quantification’, CSLI Publications, Stanford, pp. 173–209.

Zanuttini, Raffaella (1991), Syntactic Properties of Sentential Negation. A Comparative Study of Romance Languages, PhD thesis, University of Pennsylvania.

Zeijlstra, Hedde H. (2002), What the Dutch Jespersen Cycle may reveal about negative concord, *in* A.Alexiadou, S.Fischer and M.Stavrou, eds, ‘Linguistics in Potsdam’, Vol. 19, pp. 183–206.

Zeijlstra, Hedde H. (2004), *Sentential Negation and Negative Concord*, LOT Publications, Utrecht.

Zimmermann, Thomas Ede (1990), Formal semantics. Lecture notes, University of Massachusetts at Amherst.

Zwarts, Frans (1983), Determiners: a relational perspective, *in* A.ter Meulen, ed., ‘Studies in Modeltheoretic Semantics’, Vol. 1 of *Groningen-Amsterdam Studies in Semantics*, Foris Publications, Dordrecht, pp. 37–62.

Zwarts, Frans (1998), Three types of polarity, *in* F.Hamm and E.Hinrichs, eds, ‘Plurality and Quantification’, Vol. 69 of *Studies in Linguistics and Philosophy*, Kluwer Academic Publishers, pp. 177–238.