Negative Concord in Romanian as Polyadic Quantification

Gianina Iordăchioaia
Institut für Linguistik: Anglistik
Universität Stuttgart
gianina@ifla.uni-stuttgart.de

Frank Richter
IMS, Universität Stuttgart &
SfS, Universität Tübingen
fr@sfs.uni-tuebingen.de

1 Introduction

We present an analysis of the syntax and semantics of the core of Romanian Negative Concord (NC) constructions as polyadic quantification in Lexical Resource Semantics (LRS). Following a proposal by de Swart and Sag (2002) for French, we express the truth conditions associated with Romanian NC constructions by means of negative polyadic quantifiers. Going beyond de Swart and Sag's largely informal treatment of the logical representations for polyadic quantification in HPSG, we extend the logical representation language of Lexical Resource Semantics (LRS, Richter and Sailer (2004)) and modify the interface principles of LRS to accommodate polyadic quantification. Apart from the immediate benefit of a theory of Romanian NC, we obtain an interesting result for constraint-based approaches to model-theoretic semantics like LRS: Resumptive polyadic quantifiers, which are at the heart of this approach to NC, are a notorious problem for frameworks which use the lambda calculus in combination with a functional theory of types to define a compositional semantics for natural languages. LRS overcomes these fundamental logical limitations and is powerful enough to specify by standard HPSG devices a precise systematic relationship between a surface-oriented syntax and semantic representations with polyadic quantifiers.

2 Data

Sentential negation in Romanian is usually expressed by a verbal prefix, *nu* (Barbu (2004)). In the absence of other negative elements, it contributes semantic negation (1a). If in addition an n-word is present such as *niciun* in *niciun student (no student)* in (1b), only a NC reading is available, a double negation interpretation (DN) is not. The negation marker (NM) *nu* is obligatory. In constructions with two n-words, both a NC reading and a DN reading are available (1c).

(1) a. Un student **nu** a venit.
a student NM has come
'Some student didn't come.'

b. **Niciun** student *(**nu**) a venit.
no student NM has come
i. 'No student came.' (NC)
ii. # 'No student didn't come.' (DN)

c. **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)

The observations in (1b) and (1c) suggest that (a) n-words are exponents of semantic negation (as can be confirmed by other tests), and (b) the negative marker *nu* is semantically non-negative in the presence of n-words. This is confirmed by the test in (2) and (3): Negative functions are anti-additive: f is anti-additive iff for each X and Y, $f(X \vee Y) = f(X) \wedge f(Y)$. In the absence of n-constituents, *nu* receives an anti-additive interpretation (2):

(2) 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.'

If the disjunction that *nu* takes as argument contains n-constituents, anti-additivity disappears, and the two n-constituents are interpreted independently under the scope of negation (3):

(3) 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. ≠ Studenţii **nu** au citit **niciun** roman *şi* studenţii **nu** au citit
students-the NM have read no novel and students-the NM have read
nicio poezie.
no poem
≠ 'The students read no novel and the students read no poem.'

These data are consistent with an analysis that assumes that determiner n-words and negative NP constituents are quantifiers of Lindström type $\langle 1,1\rangle$ and $\langle 1\rangle$ (Lindström (1966)), respectively, and they may combine to form a polyadic quantifier (of type $\langle 1^n, n\rangle$ and $\langle n\rangle$) by resumption (Keenan and Westerståhl (1997)). The negative marker *nu* is analyzed as a negative quantifier of type $\langle 0\rangle$ that is absorbed under resumption with other negative polyadic quantifiers. The relevant technical details will be briefly outlined in our LRS implementation of polyadic quantification and resumption below.

3 LRS with Polyadic Quantifiers

For our analysis we will need a higher-order logical language with negative polyadic quantifiers. Here we briefly outline its crucial properties and indicate how to integrate it with LRS.

We assume a simple type theory with types e and t. Functional types are formed in the usual way. The syntax of the logical language provides function application, lambda abstraction, equality and negative polyadic quantifiers. By standard results this is enough to express the usual logical connectives and monadic quantifiers. In reference to the simple type theory, we call our family of languages Ty1. Var and $Const$ are a countably infinite supply of variables and constants of each type:

Definition 1 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* $v_{i,\tau} \in Var$, *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 $\tau \in Type$, *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* $\alpha_{t1}, \alpha_{t2}, ..., \alpha_{tn}, \beta_t \in Ty1$: $(NO(v_{i_1,\tau}, ..., v_{i_n,\tau})(\alpha_{t1}, ...\alpha_{tn})(\beta_t))_t \in Ty1$.

The standard constructs receive their usual interpretation. Here we only state the interpretation of the polyadic quantifiers:

Definition 2 The Semantics of Ty1 Terms *(clause for negative polyadic quantifiers only)*
For each term $\alpha_\tau \in Ty1$, for each model M and for each variable assignment $a \in Ass$, 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 $\alpha_{t1}, \alpha_{t2}, ..., \alpha_{tn}, \beta_t \in Ty1$:

$$[\![NO(v_{i_1,\tau}, ..., v_{i_n,\tau})(\alpha_{t1}, ..., \alpha_{tn})(\beta_t)]\!]^{M,a} = 1 \textit{ iff}$$
$$\textit{for every } d_{i_1}, d_{i_2}, ..., d_{i_n} \in D_{E,\tau},$$
$$[\![\alpha_{t1}]\!]^{M,a[v_{i_1,\tau}/d_{i_1}]} = 0 \textit{ or } [\![\alpha_{t2}]\!]^{M,a[v_{i_2,\tau}/d_{i_2}]} = 0 \textit{ or } \ldots$$
$$\textit{or } [\![\alpha_{tn}]\!]^{M,a[v_{i_n,\tau}/d_{i_n}]} = 0 \textit{ or } [\![\beta_t]\!]^{M,a[(v_{i_1},...,v_{i_n})/(d_{i_1},...,d_{i_n})]} = 0.$$

(4) shows the truth conditions that we obtain for the translation of the Romanian counterparts of *John didn't come* (4a) and *No teacher didn't give no book to no student*, where all NPs are n-constituents and form a ternary negative quantifier by resumption (4b):

(4) 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 $[\![give'(x, y, z)]\!]^{M,a[(x,y,z)/(d_1,d_2,d_3)]} = 0$

Minor adjustments suffice to integrate these logical representations in LRS. In the signature, the appropriateness of *gen-quantifier* of Richter and Kallmeyer (2007) is generalized to lists of variables (instead of single variables), and the restrictor of quantifiers now contains a list of expressions:

```
me  TYPE  type
    gen-quantifier  VAR  list
                    RESTR  list
                    SCOPE  me
```

The theory of well-formed logical expressions restricts polyadic generalized quantifiers to the form given in Definition 1. The relational restrictions in (5) guarantee that $\boxed{1}$ is a list of variables, they all have the same type $\boxed{3}$, the expressions in the restrictor list $\boxed{2}$ are of type t, and there are exactly as many restrictor expressions as variables on the two lists:

(5) $$\textit{gen-quantifier} \rightarrow \begin{bmatrix} \text{TYPE} & \textit{truth} \\ \text{VAR} & \boxed{1} \\ \text{RESTR} & \boxed{2} \\ \text{SCOPE} & \text{TYPE } \textit{truth} \end{bmatrix}$$
$$\wedge \texttt{ variable-list}(\boxed{1}) \wedge \texttt{ same-type-list}(\boxed{3},\boxed{1})$$
$$\wedge \texttt{ truth-list}(\boxed{2}) \wedge \texttt{ same-length}(\boxed{1}, \boxed{2})$$

The LRS Projection Principle (excont and incont percolation, inheritance of parts lists) remains unchanged. The clause of the Semantics Principle governing the combination of quantificational determiners with nominal heads is adjusted to polyadic quantifiers:

(6) The Semantics Principle, clause 1
1. if the non-head 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[1] of the list ϕ, and the incont value

[1] The symbol "$\triangleleft_\in$" is the infix notation of the new relation `subterm-of-member`, a generalized subterm relation. Note that v and ϕ are shorthand for a list of variables and a list of expressions in $Q(v, \phi, \psi)$. ψ is a single expression.

of the non-head daughter is identical to the EXCONT value of the head daughter:

$$\begin{bmatrix}\text{DTRS}|\ \text{SPR-DTR}|\text{SS}|\ \text{LOC}\begin{bmatrix}\text{CAT}|\ \text{HEAD} & det\\ \text{CONT}\ |\text{MAIN} & gen\text{-}quantifier\end{bmatrix}\end{bmatrix} \longrightarrow$$
$$\left(\begin{bmatrix}\text{DTRS} & \begin{bmatrix}\text{H-DTR}\ |\text{LF} & \begin{bmatrix}\text{EXCONT} & \boxed{1}\\ \text{INCONT} & \boxed{2}\end{bmatrix}\\ \text{SPR-DTR}|\text{LF} & \begin{bmatrix}\text{INCONT} & \boxed{1}\begin{bmatrix}gen\text{-}quantifier\\ \text{RESTR}\ \boxed{3}\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix} \wedge \boxed{2} \triangleleft_{\in} \boxed{3}\right)$$

Resumption will be implemented in LRS as identity of quantifiers contributed by lexical elements. Thus no special technical apparatus for the resumption operation has to be introduced in preparation of our analysis of negative concord in Romanian in the next section.

4 The Analysis of Romanian NC

The analysis of simple negated sentences without n-constituents follows immediately from the lexical analysis of verbs with the negative marker prefix *nu*, which we derive by lexical rule (not shown here, but see Przepiórkowski and Kupść (1997) for a comparable analysis of the Polish negative marker, and Ionescu (1999) for similar assumptions about Romanian). For the verb in (1a) we get:

(7) *nu a venit* ('NM has come', derived by Lexical Rule)

$$\begin{bmatrix}word\\ \text{PHON} & \left\langle \texttt{nu, a, venit}\right\rangle\\ \text{SS}\ |\ \text{LOC} & \begin{bmatrix}\text{CAT} & \begin{bmatrix}\text{HEAD}\ |\text{NEG} & +\\ \text{VAL}\ |\text{SUBJ} & \left\langle \text{NP}_{\boxed{1a}}\right\rangle\end{bmatrix}\\ \text{CONT} & \begin{bmatrix}\text{INDEX}\ |\text{VAR} & no\text{-}var\\ \text{MAIN} & \boxed{3a}\ come'\end{bmatrix}\end{bmatrix}\\ \text{LF} & \begin{bmatrix}\text{EXC} & \boxed{0}\\ \text{INC} & \boxed{3}\ come'(\boxed{1a})\\ \text{PARTS} & \left\langle \boxed{3}, \boxed{3a}, \boxed{7}\ no(u,\gamma,\delta)\right\rangle\end{bmatrix}\end{bmatrix} \ \&\ \boxed{3}\triangleleft\boxed{0}\ \ \&\ \boxed{3}\triangleleft\delta\ \ \&\ \boxed{7}\triangleleft\boxed{0}$$

With standard LRS mechanisms in combination with a language-specific constraint that excludes the existential quantifier originating from *un student* from occurring in the immediate scope of negation, we obtain $some(x, student'(x), no((), (), come'(x)))$ as the truth condition for (1a).

For the analysis of NC (1b), we adapt the NEG CRITERION of Richter and Sailer (2004) to Romanian and the polyadic quantifier approach:

(8) THE NEG CRITERION for Romanian
For every finite verb, if there is a type $\langle 0\rangle$ *no* quantifier 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.

$$\forall\boxed{0}\ \forall\boxed{1}\ \forall\boxed{2}\ \forall\boxed{3}$$
$$\left(\begin{array}{l}\begin{bmatrix}word\\ \text{SS}\ |\text{LOC} & \begin{bmatrix}\text{CAT}\ |\text{HEAD} & \begin{bmatrix}verb\\ \text{VFORM} & fin\end{bmatrix}\\ \text{CONT}\ |\text{MAIN} & \boxed{3}\end{bmatrix}\\ \text{LF}\ |\text{EXC}\ \boxed{0}\end{bmatrix} \wedge \boxed{1}\ no((),(),\delta)\triangleleft\boxed{0} \wedge \boxed{2}\ no(v,\alpha,\beta)\triangleleft\boxed{0} \wedge \boxed{3}\triangleleft\delta \wedge \boxed{3}\triangleleft\beta\\ \rightarrow\\ \exists\boxed{4}\left(\begin{bmatrix}\text{LF}\ |\text{PARTS} & \boxed{4}\end{bmatrix} \wedge \boxed{2}\in\boxed{4}\right)\end{array}\right)$$

The obligatoriness of the negative marker in negative concord constructions is an immediate consequence of the NEGATIVE CONCORD CONSTRAINT of Romanian (9). If a sentential negation ($\boxed{2}$) outscopes a verb (within the verb's EXCONT), the verb must be negatively marked, which in turn can only be the case if it is licensed as output of the negation lexical rule.

(9) THE NC CONSTRAINT (NCC)

$$\forall\boxed{0}\ \forall\boxed{1}\ \forall\boxed{2}\left(\begin{array}{l} \begin{bmatrix} \mathit{word} \\ \text{SS}\,|\text{LOC} \begin{bmatrix} \text{CAT}\,|\text{HEAD} \begin{bmatrix} \mathit{verb} \\ \text{VFORM}\ \ \mathit{fin} \end{bmatrix} \\ \text{CONT}\,|\text{MAIN}\ \boxed{1} \end{bmatrix} \\ \text{LF}\,|\text{EXCONT}\ \ \boxed{0} \end{bmatrix} \wedge \boxed{2}\ no(v,\alpha,\beta) \triangleleft \boxed{0} \wedge \boxed{1} \triangleleft \beta \\ \rightarrow \\ \begin{bmatrix} \text{SS}\,|\text{LOC}\,|\text{CAT}\,|\text{HEAD}\ \begin{bmatrix}\text{NEG}\ \ +\end{bmatrix} \end{bmatrix} \end{array}\right)$$

Assuming that n-words introduce on their PARTS list a negative quantifier of unspecified type $\langle 1^n, n\rangle$ with exactly one new variable, these two principles suffice to guarantee the correct analysis of (1b) and (1c), shown in (10) and (11), respectively.

(10) $no(x, student'(x), come'(x))$

(11) a. $no(x, student'(x), no(y, book'(y), read'(x, y)))$ (DN)

b. $no((x, y),(student'(x), book'(y)), read'(x, y))$ (NC)

In (1b), the verb and the n-word each contribute a negative polyadic quantifier. The verb does not contribute a variable for the quantifier, whereas the negative determiner does. If the two negative quantifiers were not identical, they would be subject to the NEG CRITERION, because the quantifier contributed by the verb would have an empty variable list, i.e. it would be of type $\langle 0\rangle$. But then the quantifier contributed by the n-word would have to be on the PARTS list of the verb. This cannot be the case, since the verb only contributes one negative quantifier. Therefore the quantifiers contributed by the n-word and by the verb must be identical, with one variable on the VAR list, resulting in (10). The NEG CRITERION has nothing to say about this case, because there is no type $\langle 0\rangle$ quantifier in the formula. Since an n-word always contributes a negative quantifier, the NCC guarantees that the verb must have the negation prefix and contribute a negative quantifier in the presence of an n-word.

In sentences with more than one n-word such as (1c), the negative quantifier contributed by the verb must undergo resumption with at least one of the two quantifiers contributed by the n-words for the reasons just described. If one n-word does not undergo resumption with the NM and the other n-word, we obtain the DN reading as in (11a). However, there is also the possibility that all the negative quantifier contributions in the sentence are identified. The number of variables contributed by the individual n-words determines the type of the resumptive quantifier. For (1c) with two n-words, each contributing one variable, the second available alternative is resumption of all three negative quantifiers and leads to a quantifier of type $\langle 1^2, 2\rangle$ for the NC reading, shown in (11b).

In the talk, we will also show how our analysis treats cases in which NC crosses clause boundaries of embedded subjunctive clauses. In these constructions, a negated matrix verb may license n-words in an embedded subjunctive clause. The matrix negation then enters into a negative polyadic quantifier with the embedded n-words. Our analysis of the syntax-semantics interface will provide an account of the conditions when this is possible.

5 Conclusion

The present analysis of NC in Romanian applies the approach that was pioneered by an analysis of French in de Swart and Sag (2002). Our theory considerably extends de Swart and Sag's proposal by explicitly integrating a higher-order logic with polyadic quantification in HPSG. We expect that the formulation of the polyadic quantifier approach to NC in LRS will make it possible to unify this line of research with the typological approach to NC in Polish, French and German presented in Richter and Sailer (2006). Last but not least, adding polyadic quantification to LRS opens the door to exploring a whole range of new semantic phenomena in HPSG such as cumulative and *same/different* (unreducible) polyadic quantifiers (Keenan (1992), Keenan and Westerståhl (1997)). Since our constraint-based syntax-semantics interface supports the integration of polyadic quantifiers, HPSG theories can take full advantage of them. This brings within reach an explicit specification of the syntax and semantics of constructions that require unreducible polyadic quantifiers for an adequate rendering of their truth conditions and have, for that reason, turned out to be problematic in other grammar frameworks.

References

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*, pp. 68–82. Bucharest U. Press.

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.

Keenan, Edward L. (1992). Beyond the Frege Boundary. *Linguistics and Philosophy 15*, 199–221.

Keenan, Edward L. and Westerståhl, Dag (1997). Generalized Quantifiers in Linguistics and Logic. In J. van Benthem and A. ter Meulen (Eds.), *Handbook of Language and Logic*, pp. 837–893. Amsterdam: Elsevier Science.

Lindström, Per (1966). First Order Predicate Logic with Generalized Quantifiers. *Theoria 32*, 186–195.

Przepiórkowski, Adam and Kupść, Anna (1997). Negative Concord in Polish. Technical report, Institute of Computer Science, Polish Academy of Sciences.

Richter, Frank and Kallmeyer, Laura (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*, Volume 7 of *CST Working Papers, København, Denmark*, pp. 31–83.

Richter, Frank and Sailer, Manfred (2004). Basic Concepts of Lexical Resource Semantics. In *ESSLLI 2003 – Course Material I*, Volume 5 of *Collegium Logicum*. Kurt Gödel Society Wien.

Richter, Frank and Sailer, Manfred (2006). Modeling Typological Markedness in Semantics: The Case of Negative Concord. In S. Müller (Ed.), *Proceedings of the 13th International Conference on Head-Driven Phrase Structure Grammar*, pp. 305–325. CSLI Publications.

de Swart, Henriëtte and Sag, Ivan A. (2002). Negation and Negative Concord in Romance. *Linguistics and Philosophy 25*, 373–417.