Negation and Negative Concord in Romance

Henriëtte de Swart
UiL-OTS & French
Utrecht University
Henriette.deswart@let.uu.nl

Ivan A. Sag
Linguistics & CSLI
Stanford University
sag@csli.stanford.edu

March 2, 2000

Abstract

This paper addresses the two interpretations a combination of negative indefinites can get in concord languages like French, namely a concord reading which amounts to a single negation, or a double negation reading. We develop an analysis in a polyadic framework, in which a sequence of negative indefinites can be interpreted as an iteration of quantifiers or as absorption. The first option leads to a scopal relation, interpreted as double negation. The second option leads to the construction of a polyadic negative quantifier, which corresponds to the concord reading. Given that negation participates in negative concord, we develop an extension of the polyadic approach which can deal with non-variable binding operators. The contribution of negation in a concord context is semantically empty, which is taken to explain the cross-linguistic variation we find in the participation of negation in negative concord. The semantic analysis is incorporated into a grammatical analysis formulated in HPSG, which crucially relies on the assumption that quantifiers can be combined in more than one way upon retrieval from the quantifier store.

1 Approaches to Negative Concord

Negative concord is the general term for cases where multiple occurrences of morphologically negative constituents express a single negation. The phenomenon is exemplified by the French example in (1):[1]

[1] In spoken French, the *n(e)*-marking on the verb is optional; it is obligatory in written (formal) varieties. We return to the matter of *ne*-marking in section 4.

(1) Personne (n')a rien fait
No one NE-has nothing done

a. No one has done nothing (i.e. everyone did something) [DN]
$\neg\,\exists x\,\neg\,\exists y\ \mathrm{Do}(x,y)$

b. No one has done anything [NC]
$\neg\,\exists x\,\exists y\ \mathrm{Do}(x,y)$

(1) contains two negative quantifiers, and can be interpreted as expressing double negation (1a) or single negation (1b). The single negation reading is the concord interpretation in which two negative quantifiers 'merge' into one. We also find negative concord in other Romance languages (Spanish, Catalan, Italian), in West Flemish, African American Vernacular English, Polish, etc. This paper mainly discusses negative concord in French, but establishes some comparisons with other Romance languages.

Negative concord raises problems for the principle of semantic compositionality, which says that the meaning of a complex whole is a function of the meaning of its parts and the way they are put together. If we interpret the negative quantifiers in (1) in terms of first-order logic with negation and universal/existential quantification, we can derive the double negation reading, but this leaves the single negation reading (the 'concord' reading) unaccounted for. To deal with this problem, two types of analysis of negative concord have been proposed in the literature, which we may loosely call 'local' and 'global'.

The global approach preserves the negative character of both quantifiers, and translates them as negative indefinites. Zanuttini (1991) and Haegeman and Zanuttini (1996) define an operation of factorization which reinterprets a sequence of quantifiers $\forall x_1 \neg \ldots \forall x_n \neg$ as a new sequence $\forall x_1 \ldots x_n \neg$. According to May (1989), factorization fails to preserve compositionality, because part of the semantic contribution of the composing elements is simply erased. As an alternative, he defines an absorption operation which interprets a sequence of negative indefinites $\mathrm{NO}_{x_1} \ldots \mathrm{NO}_{x_n}$ as a polyadic quantifier complex $\mathrm{NO}_{x_1 \ldots x_n}$ (cf. also Van Benthem 1989, Keenan and Westerståhl 1997). May's analysis has also been criticized for its lack of compositionality (e.g. Corblin 1996). Note that absorption requires a mode of composition different from function application. If the only mode of composition we allow is function application, the only interpretation for a sequence of negative indefinites we obtain is the iteration of the monadic quantifiers $\mathrm{NO}_1 \ldots \mathrm{NO}_x$.

The local approach (e.g. Laka 1990, Ladusaw 1992) preserves strict compositionality by reinterpreting the concord item in such a way that function application yields the desired single negation interpretation. Typically, this is achieved by treating negative concord as a variety of negative polarity, which allows us to take the negative concord item to denote an existentially quantified NP, rather than a negative NP. According to Laka (1990), concord items are licensed by a possibly implicit negation operator. This assumption makes it impossible to explain why (2b) is a felicitous answer to the question in (2a), but (2c) cannot be used in this context:

(2) a. Qu'est-ce que tu as vu?
What have you seen?

b. Rien
Nothing

c. *Quoi que ce soit
Anything

Ladusaw (1992) overcomes the problems with Laka's analysis by assuming that negative concord items are negative polarity items that license themselves. In the absence of a trigger, *rien* thus licenses itself (2b), but *quoi que ce soit* (2c) does not. As pointed out by Corblin (1996), Ladusaw's analysis still suffers from too close an identification of concord items with negative polarity items. The [neg] feature contributed by each concord item is viewed as an agreement phenomenon: it is present multiple times, but only interpreted once. As a result, we only obtain the concord reading.

An important problem for an approach like Ladusaw's is the observation that a sentence like (1) is actually ambiguous between a double negation reading and a concord reading, but that the polarity approach to concord only derives the single negation reading. The reason that most researchers including Ladusaw (1992), Haegeman (1995), Deprez (1997) and others have ignored the double negation reading in Romance, is that sentences like (1) strongly prefer a concord reading. But examples which provide a better illustration of the double negation reading are provided by Corblin (1996), and given here in (3a) and (b):

(3) a. Personne n'est le fils de personne [ambiguous]
No one NE-is the son of no one
= No one is the son of anyone [NC]
= Everyone is the son of someone [DN]

b. Personne n'aime personne [ambiguous]
No one NE-loves no one
= No one loves anyone [NC]
= Everyone loves someone [DN]

c. Il ne va pas nulle part, il va à son travail [DN only]
He NE-goes not nowhere, he goes to his work

= He doesn't go nowhere, he goes to work = * He doesn't go anywhere, he goes to work

French speakers agree that sentences like (3a,b) have a double negation as well as a concord reading. But for (3c) (from Muller 1991: 259), double negation is the only option. The existence of double negation readings have led people to defend an analysis in terms of contextual ambiguity. Van der Wouden (1994: 98–103) argues that contextual ambiguity nicely reflects the two faces of concord items: they are negative if they are unembedded, but within the scope of a negative quantifier they shift towards existential quantifiers. Corblin (1996) adopts a similar approach, and formulates a construction rule for negative quantifiers in a DRT framework, which introduces a negation, and an indefinite in the scope of negation. If a new negative quantifier shows up when the construction rule has already applied, we can optionally just apply the second half of the rule. This is equivalent to a shift of the concord item to an existential quantifier. But there are two problems with the approach in terms of contextual ambiguity. One is that the ambiguity is not well motivated, and the other is that it does not seem appropriate to treat an ambiguity which clearly arises from the construction as a lexical ambiguity.

We are aware of only one argument that has been advanced in favor of the interpretation of expressions like *rien, personne, jamais* in terms of existential quantification. This involves modification by *presque* ('almost'), an adverb which combines with universal (4a), but not with existential quantifiers (4b):

(4) a. J'ai invité presque tous les étudiants.
I have invited almost all the students.

b. *J'ai invité presque quelques étudiants.
I have invited almost some students.

Accordingly, Van der Wouden and Zwarts (1993) take the contrast between (5a) and (5b) to indicate that the lower items in a concord chain are to be interpreted in terms of an existential quantifier, rather than a universal quantifier:

(5) a. Presque personne n'a rien dit. [ambiguous]
Almost no one NE-has nothing said.
= Almost no one said anything [NC]
= Nearly everyone said something [DN]

b. Personne n'a presque rien dit. [DN only]
No one NE-has almost nothing said.
= No one said almost nothing
= *No one said almost anything

However, it seems that the data are too weak to support this conclusion. As Vallduví (1994) points out, this leaves the felicity of the concord reading of the counterpart of (5b) in Catalan unaccounted for. Moreover, it turns out that modification of the lower concord item by *presque* is not always impossible in French, as examples like (6) demonstrate.[2]

(6) a. Un vieil écrivain nous a quittés sur la pointe des pieds sans que presque personne y prête attention.
An old writer has left us quietly without that almost no one paid attention to it.
= hardly without any attention

b. Je n'ai plus trouvé presque rien ridicule
I have no more found almost nothing ridiculous
= There was hardly anything I found ridiculous anymore

According to Muller (1991: 319), *presque* can modify an embedded concord item as long as we interpret the adverb as taking wide scope over the concord chain as a whole. This suggests that it is possible to interpret the data in (5) and (6) in such a way that they are compatible with an absorption analysis. Thus, modification of negative indefinites by *presque* does not provide evidence in favor of an interpretation of concord items in terms of existential quantifiers.

The second problem the contextual ambiguity approach faces is that it is difficult to formulate the conditions under which the negative and the indefinite interpretation show up as part of the lexical entry of the concord item. Consider in particular the problems which arise when we embed concord items under the sentential negation *pas*. The contrast between (7a) and (7b) suggests that *pas* is outside the concord system and provides the prime context to distinguish negative polarity items from concord items in French (compare Corblin 1996, Haegeman and Zanuttini 1996, de Swart 1998):

(7) a. Je n'ai pas vu quoi que ce soit [NPI: $\neg\exists$]
= I have not seen anything

b. Ce n'est pas rien [DN only]
It is not nothing
= It is quite something

c. Le enquêteurs n'ont pas fait le voyage pour rien [DN only]
The interviewers did not make the trip for nothing

[2](6a) is from Grevisse *Le bon usage*, section 726. (6b) is from S. de Beauvoir. *Mémoires d'une jeune fille rangée*, Poche p. 355, and is quoted by Muller (1991: 319).

(7a) has a single negation reading as expected, but (7b) and (c) only have a double negation reading.

Note however that *pas* does trigger negative concord in other cases:

(8) a. Il ne veut pas que personne soit lésé. [restricted varieties][NC]
He NE wants not that no one is-SUBJ wronged.
= He does not want anyone to be wronged

b. S'il y a quelque chose, il fera pas d'cadeau à personne. [restricted varieties][NC]
If there is something, he will not give a present to no one
= If there is something, he will not grant anyone a favor

c. Je n'ai pas donné le moindre renseignement à personne. [NC]
I NE have not given the least information to no one
= I have not given the least information to anyone

d. Il y a pas personne en ville. [Québécois][NC]
There is not no one in town
= There is no one/not anyone in town

e. Jan pa we pèsòn [Haitian creole][NC]
Jan not see no one
= Jan does not see anyone

The examples in (8a,b) are from Muller (1991: 261, 263), who points out that a concord item can be embedded under *pas* if it is in an indirect argument or in an embedded clause. The felicity of the concord reading of these sentences is subject to dialectal variation. However, as pointed out by Richter and Sailer (1999), speakers of standard French accept the concord reading of (8c), an example they attribute to F. Corblin (p.c.). We will come back to this issue in section 4.1 below.

These observations suggest that we do not want to rule out embedding of concord items under negation, but that we want to impose syntactic anti-locality restrictions on the relation between *pas* and the concord item. The claim that the relation between negation and negative concord is subject to syntactic, in addition to semantic constraints receives further support from the fact that in older stages of the language (e.g. middle French) concord items were easily licensed by *pas*. This is still the case in modern Québécois French as illustrated by (8d) (from Muller 1991: 262) and in the French-based Haitian creole, as exemplified by (8e) (from Depréz 1997).

The interaction with the syntax suggests that we should treat the constraints on the interpretation of the concord item in the syntax-semantics interface, rather than in the lexical semantics of the concord item. If a 'local' approach is not well equipped to deal with ambiguities which clearly arise from the construction as a whole, then we might be better off with a 'global' approach after all. As pointed out above, May (1989) and Van Benthem (1989) develop a polyadic quantifier approach in which a sequence of negative indefinites is interpreted as a complex negative quantifier. May and Van Benthem only treat a few isolated examples of English. The aim of this paper is to use the polyadic approach to develop a serious model of interpretation of negation and negative concord in Romance.

2 A Polyadic Analysis of Negative Concord

The polyadic quantifier approach has been developed as an extension of the generalized quantifier framework developed by Lindström (1966), Barwise and Cooper (1981), Van Benthem (1986) and others to sentences involving multiple quantifiers. In the generalized quantifier framework, NPs are analyzed as expressions of type $\langle\langle e,t,\rangle,t\rangle$, and determiners are expressions of type $\langle\langle e,t\rangle,\langle\langle e,t\rangle,t\rangle\rangle$ which denote relations between sets. These type assignements correspond with the Lindström characterization of NPs as quantifiers of type $\langle 1\rangle$ and determiners as quantifiers of type $\langle 1,1\rangle$. As Keenan and Westerståhl (1997) point out, the type assignment $\langle\langle e,t\rangle,t\rangle$ is too specific, as it focusses on NPs in subject position which combine with a VP of type $\langle e,t\rangle$ to produce a proposition of type t. In a more general perspective, NPs should be regarded as argument reducing expressions: they reduce the number of argument positions of the verb by one. Thus NPs map intransitive verbs onto propositions, transitive verbs onto intransitive verbs, ditransitive verbs onto transitive verbs, etc. This does not affect their Lindström type: they remain monadic quantifiers of type $\langle 1\rangle$. In this more general perspective, the question arises how a sequence of NPs in a transitive, ditransitive, etc. sentence is interpreted in the generalized quantifier framework.

Assume we have a sentence with a sequence of NPs (two or more). If we just combine expressions by function application, we obtain an iteration of NPs, corresponding which a scopal order between the NPs:

- Function application allows for a sequence of NPs $\langle \mathrm{NP}_1 \ldots \mathrm{NP}_n\rangle$ to be interpreted as Iteration($\mathrm{NP}_1 \ldots \mathrm{NP}_n$), that is, the iteration of the monadic quantifiers $\mathrm{NP}_1 \ldots \mathrm{NP}_n$ applied to the n-ary relation R_n.

For a sentence like (9a), iteration of the monadic quantifiers leads to the quantificational structure in (9b), which gets the interpretation in (9c):

(9) a. Some student read no book

b. SOME (STUDENT, $\{x|$ NO (BOOK, $\{y|x$ READ $y\}$)$\}$)

c. STUDENT $\cap$ $\{x|$ BOOK $\cap$ $\{y|x$ READ $y\} = \emptyset\} \neq \emptyset$

d. [SOME^{STUDENT}, NO^{BOOK}] (READ)

In words: the intersection of the set of students with the set of individuals x such that the intersection of the set of books with the set of things x read is empty is non-empty. Instead of using the structure in (9b), we could also represent the sentence by means of the structure in (9d). (9d) reflects the insight from generalized quantifier theory that the determiner complex denotes a relation between two common nouns and a two-place predicate. The Lindström type of the quantifier complex in (9d) is $\langle 1,1,2 \rangle$. In the polyadic quantifier framework we can ask questions like how many polyadic quantifiers are there? are they all reducible to an iteration of monadic quantifiers (as in (9)) or not? which ways of putting together sequences of NPs do we find in natural language?

Some examples of combinations of quantifiers which are not reducible to an iteration of monadic quantifiers are given in (10):

(10) a. Every student likes himself.
[EVERY^{STUDENT}, SELF] (LIKE)
$\forall x$ Like(x, x)

b. Every student bought a different book.
[EVERY^{STUDENT}, DIFFERENT^{BOOK}] (BUY)
BOOK $\cap$ READ$_a$ $\neq$ BOOK $\cap$ READ$_b$ for all $a, b \in$ STUDENT such that a $\neq$ b

c. Five hundred companies own three thousand computers.
[FIVE HUNDRED^{COMPANIES}, THREE THOUSAND^{COMPUTERS}] (OWN)
$\|$ COMPANY $\cap$ OWN$^{Computer}{}_y\| = 500 \wedge$
$\|$ COMPUTER $\cap$ OWN$^{Company}{}_x\| = 3000$

d. No one loves no one.
[NO^{ONE}, NO^{ONE}] (LOVE)
NO$_{x,y}$ LOVE
$= \neg\, \exists x\, \exists y$ Love(x,y)

The reflexive pronoun *himself* in (10a) is crucially dependent on the quantificational NP *every student*, because it is bound by the subject. The dependency relation makes it impossible to reduce the quantifier complex to an iteration of monadic quantifiers (Keenan 1987). Similar considerations apply to (10b): we have to match the pairs of students and books in order to verify that thebook x read is different from the book y read (Keenan 1987). In (10) it is the cumulative reading of the sentence in which a total of five hundred companies owns a total of three thousand computers which is unreducible, because the two NPs are mutually dependent (Keenan and Westerståhl 1997). May (1989) and Van Benthem (1989) treat the reading of (10d) in which the love relation is empty (the 'no love world') as a case of resumptive quantification. Although the negative quantifier occurs twice, it is interpreted only once, as a negative quantifier complex which binds two variables. As shown by (10a) and (d), unreducibility is not the same as not having a translation in first-order logic. The meaning of these sentences is easy to capture, but it is difficult to build up the meaning of the whole from the meanings of the parts in a systematic way.

If we treat (10d) as an example of resumption, we need to formulate a rule for absorption of two (or more) quantifiers into one bigger quantifier complex. The general rule for absorption can be formulated as follows (cf. Keenan and Westerståhl 1997):

- Absorption of a sequence of k $\langle 1,1\rangle$ quantifiers leads to the construction of a resumptive quantifier of type $\langle k,k\rangle$.

- If Q is an NP of type $\langle 1,1\rangle$, the k-ary resumption of Q, $\mathrm{Res}^k(Q)$, is defined for $R, S \subseteq E^k$ by: $\mathrm{Res}^k(Q)^R_E(S) = Q^R_{E^k}(S)$.

R and S are k-ary relations corresponding to the restrictor and the scope of the resumptive quantifier. In the cases discussed in this paper, R is provided by the k-ary cartesian product of the CN denotation of the monadic quantifier and S is provided by the k-place verb.

The phenomenon of negative concord that we find in Romance languages and elsewhere can be viewed as a generalization of the resumption interpretation procedure illustrated in (9d). As observed by Muller (1991), Ladusaw (1992) and others, negative concord is strictly restricted to anti-additive expressions like *personne*, 'no one', *rien* 'nothing', *jamais* 'never', at the exclusion of simply downward entailing expressions like *peu* 'few', *rarement* 'seldom', etc. May (1989) already observed that the construction of a resumptive quantifier requires a sequence of the 'same' NPs. In the domain of negative concord, the common feature is anti-additivity, as defined by Zwarts (1986):

- Anti-additivity An NP is anti-additive iff $\mathrm{NP}(\mathrm{VP}_1 \cup \mathrm{VP}_2) \Leftrightarrow \mathrm{NPVP}_1 \cap \mathrm{NPVP}_2$

The set of anti-additive NPs is thus a subset of the set of monotone decreasing NPs. The rule for Absorption of negative quantifiers restricted to anti-additive NPs can be formulated as follows:

- Absorption of a sequence of k $\langle 1,1\rangle$ anti-additive quantifiers leads to the construction of a resumptive negative quantifier of type $\langle k,k\rangle$.

We define the resumption of a set of anti-additive quantifiers $\{NO_1, \ldots, NO_n\}$ as shown in (11):

(11) Resumption:

$$\text{Res}(\{NO^{R_1}_{\sigma_1}, \ldots, NO^{R_n}_{\sigma_n}\}) = \{NO^{R_1 \cup \ldots \cup R_n}_{\sigma_1 \cup \ldots \cup \sigma_n}\}$$

Here the σ_i are sets of variables bound by each quantifier. $R_1 \ldots R_n$ give the restriction on the variable as provided by the common noun.

In the polyadic quantifier framework, we thus have more than one mode of composition. Good old function composition leads to iteration, but other ways of interpreting a sequence of quantifiers lead to cumulative and absorbed interpretations. For a sequence of negative quantifiers, two interpretations are particularly relevant. If two negative quantifiers enter a scopal relation as an iteration of monadic quantifiers, we end up with a double negation reading (12a). If two negative quantifiers undergo absorption, they create a resumptive polyadic quantifier, which corresponds to a concord reading (12b):

(12) a. Personne n'aime personne [DN]
No one is such that they love no one
$[\text{NO}^{\{Human(x)\}}_{\{x\}}, \text{NO}^{\{Human(y)\}}_{\{y\}}]\ (\text{Love}(x,y))$
$= \neg\exists x\, \neg\exists y\ \text{Love}(x,y)$

b. Personne n'aime personne [NC]
No one loves anyone
$\text{NO}^{\{Human(x), Human(y)\}}_{\{x,y\}}\ (\text{Love}(x,y))$
$= \neg\exists x\, \exists y\ \text{Love}(x,y)$

Iteration of negative quantifiers is restricted to sequences of two or sometimes three quantifiers. This seems to be due to processing constraints (cf. Corblin 1996 and Corblin and Derzhansky 1997). Resumption is not limited to a sequence of two or three quantifiers. A sequence of three or four monadic quantifiers lead to tryadic or quadratic quantifiers:

(13) a. Personne ne dit rien à personne
No one NE says nothing to no one

b. $\text{NO}^{\{Human(x), Thing(y), Human(z)\}}_{\{x,y,z\}}\ (\text{Say}(x,y,z))$
$= \neg\exists x\, \exists y \exists z\ \text{Say}(x,y,z)$

(14) a. Personne ne dit jamais rien à personne
No one NE says never nothing to no one

b. $\text{NO}^{\{Human(x), Thing(y), Human(z), Time(e)\}}_{\{x,y,z,e\}}\ (\text{Say}(x,y,z,e))$
$= \neg\exists x\, \exists y \exists z \exists e\ \text{Say}(x,y,z,e)$

Note that the resumptive quantifier is reducible: the polyadic quantifiers in (12b), (13b) and (14b) are equivalent to the iteration of monadic quantifiers in (15a), (b) and (c), respectively:

(15) a. $[\text{NO}_{\{x\}}^{\{Human(x)\}}, \text{SOME}_{\{y\}}^{\{Human(y)\}}]\ (\text{Love}(x,y))$
$= \neg\exists x\, \exists y\ \text{Love}(x,y)$

b. $[\text{NO}_{\{x\}}^{\{Human(x)\}}, \text{SOME}_{\{y\}}^{\{Thing(y)\}}, \text{SOME}_{\{z\}}^{\{Human(z)\}}]\ (\text{Say}(x,y,z))$
$= \neg\exists x\, \exists y \exists z\ \text{Say}(x,y,z)$

c. $[\text{NO}_{\{x\}}^{\{Human(x)\}}, \text{SOME}_{\{y\}}^{\{Thing(y)\}}, \text{SOME}_{\{z\}}^{\{Human(z)\}}, \text{SOME}_{\{e\}}^{\{Time(e)\}}]$
$(\text{Say}(x,y,z,e))$
$= \neg\exists x\, \exists y \exists z \exists e\ \text{Say}(x,y,z,e)$

The equivalence between the formulas in (12b) and (15a), (13b) and (15b), (14b) and (15c) need not come as a surprise given the observation that the pairs of sentences in (16a,b) have the same truth conditions (under the negative concord interpretation of the sequence of negative qantifiers of course):[3]

(16) Personne n'aime personne ⇔
Personne n'aime qui que ce soit

(17) a. Personne ne dit rien à personne ⇔

b. Personne ne dit quoi que ce soit à personne ⇔

c. Personne ne dit rien a qui que ce soit ⇔

d. Personne ne dit quoi que ce soit à qui que ce soit

The observation that a concord interpretation of a sequence of negative quantifiers leads to the same interpretation as a negative quantifier followed by one or more existential quantifiers inspired the treatment of concord items as negative polarity items in the first place (cf. section 1 above). The fact that the truth conditions of negative sentences in concord languages can be captured in terms of iteration of monadic quantifiers can be construed as an argument against the polyadic analysis of negative concord and in favor of the local approach. However, it is unclear how strong this argument is in view of the following two observations. First, there are other unreducible polyadic constructions which can be translated into first-order

[3] Compare Muller 1991: 316–317 for the data. According to Richter and Sailer 1999, not everyone finds (17b) fully grammatical. This might be due to the choice of the negative polarity item: speakers who find (17b) somewhat less felicitous accept an axample like (8c) above, which has the exact same structure.

logic, such as consturctions in which reflexive pronouns are bound to a quantificational subject (compare 10a above). So first-order definability is not decisive. Furthermore, it has been argued more generally that language does go beyond the Frege boundary, and polyadic quantification is taken to be at issue in other complex constructions (compare May 1989, Van Benthem 1989, Westerståhl 1989, Keenan 1987, 1992, Moltmann 1995, and Keenan and Westerståhl 1997 for discussion). This means that the definitions of absorption and resumptive quantification have independent motivation in the sense that they are embedded in a larger framework, and are not introduced as ad hoc mechanisms to account for a small set of data (compare the rejection of neg-factorization in section 1 above). Note also that an approach in terms of iteration of monadic quantifiers is only possible if we interpret embedded concord items in terms of existential quantification, an analysis for which we have not found independent evidence, as we saw in section 1. Finally, we also find resumptive readings in languages like English, which is not a concord language. May (1989) argues that the assumption that *no one* translates as an existential quantifier is an unacceptable solution for English negative quantifiers, because it does not respect the lexical semantics of the quantifier. A treatment of negative concord as resumptive quantification makes it possible to develop a unified analysis of the English and Romance examples.

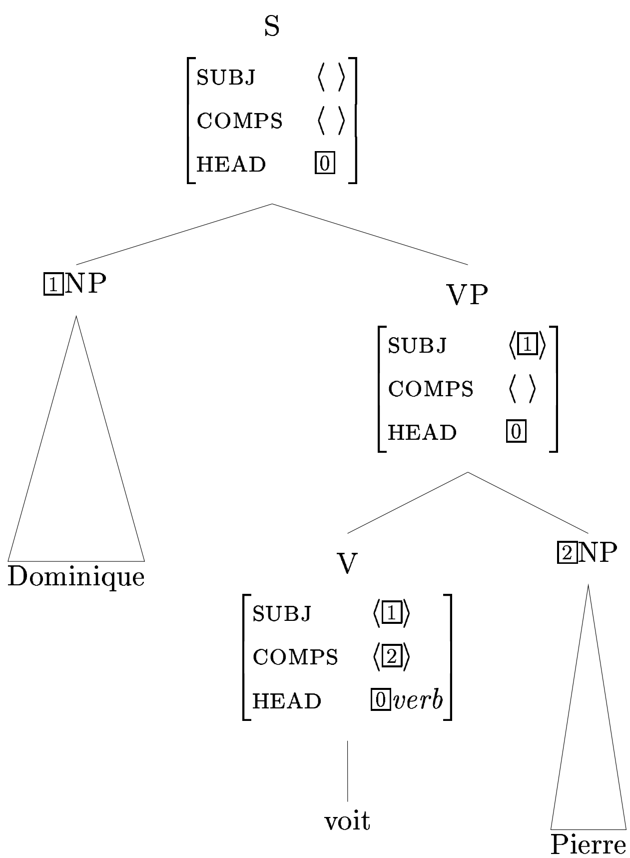
We would like to explain the synchronic situation of negation and negative concord as a result of a coherent diachronic development. Concord items like *rien, personne*, and even the negation operator *pas* started out as indefinites indicating minimal amounts. They became polarity items when they were used as a reinforcement of the negation *ne*. Over the centuries, *ne* grew weaker and became incapable of being the sole expression of negation. As a result, the original indefinites ended up becoming real negatives, and *pas* became the negative marker. Because the iteration of monadic quantifiers in (17c) is equivalent to the polyadic quantifier (17b), it became possible to view each of the support items as negative items in and of themselves. Once the concord items were analyzed as real negative quantifiers, it became possible to combine them in ways other than just by absorption. This opened the way to the interpretation of a sequence of negative indefinites as an iteration of monadic quantifiers, which made it possible to derive a double negation reading. The role of *pas* is particularly striking in this respect. As pointed out in section 1 above, *pas* participated in the negative concord system up until the 17th century (middle French). The position of *pas* in the current system of negation in French will be discussed in section 4.2 below.

Summing up, the main difference between a concord language like French and a double negation language like English is that the former has a preference for absorption over iteration of negative indefinites, whereas the latter has a preference for iteration over absorption of negative indefinites. This preference is a result of the different diachronic developments of the two languages. Also, it is really just a matter of preference: in principle, both interpretations are available for both types of languages.

3 A Lexical Analysis

We incorporate the semantic analysis just sketched into a grammatical analysis formulated in Head-Driven Phrase Structure Grammar (HPSG; see Pollard and Sag 1994, Sag and Wasow 1999). We assume that the finite verb is the head of its clause, which is projected by 'cancelling' elements (as in Categorial Grammar) from the values of its valence features, i.e. the COMPS and SUBJ lists sketched in (18):

(18)



The value for the feature HEAD (reflecting part of speech and related information) is percolated from verb to VP to S via a separate constraint: the Head Feature Principle, familiar from work in X-Bar Theory.

Recent work in HPSG (e.g. Davis 1996, Manning and Sag 1998) has explained a variety of problems of linking and binding in terms of a further lexical feature – ARGUMENT STRUCTURE (ARG-ST), which, barring extraction, 'cliticization', or 'pro drop', is aligned with the valence features as shown in (19):

(19) $\begin{bmatrix} \text{SUBJ} & \boxed{a} \\ \text{COMPS} & \boxed{b} \\ \text{ARG-ST} & \boxed{a} \oplus \boxed{b} \end{bmatrix}$

Thus the subject and complements of a verb are included in its (syntactic) argument structure. The feature ARG-ST will provide the platform on which we will place the semantic analysis of the previous section.

First, we need to consider the general treatment of quantificational semantics in a constraint-based theory like HPSG:

- All quantifiers 'start out' in storage (Cooper 1983, Pollard and Sag 1994).
- Quantifiers are retrieved from storage at the lexical level, e.g. by verbs other than raising verbs (Pollard and Yoo 1998).
- This retrieval is effected by a constraint that relates the STOREs of a verb's arguments and the verb's semantic content.

The implementation of these ideas proceeds in terms of a semantic content structured as shown in (20)

(20) $\begin{bmatrix} \text{CONTENT} & \begin{bmatrix} \text{QUANTS} & \langle \ldots \rangle \\ \text{NUCLEUS} & [\ ] \end{bmatrix} \end{bmatrix}$

Here the QUANT(IFIER)S feature takes a (possibly empty) list of generalized quantifiers as its value and NUCLEUS takes a (nonquantified) parametric expression (an open sentence) as its value. The meaning thus represented is the iteration of the generalized quantifiers applied to the nucleus (i.e. a classical scoping of the quantifiers).

The feature STORE takes as value a (possibly empty) set of generalized quantifiers. Thus every syntactic expression has some value for the feature STORE – the set of those quantifiers occurring within that expression that are not already retrieved. Given this, a verb's ARG-ST list includes information about all the unscoped quantifiers within all its syntactic arguments. The argument structure for (21a), for example, is as shown in (21b).

(21) a. Everyone loves someone.

b. $\begin{bmatrix} \text{ARG-ST} & \left\langle \begin{bmatrix} \text{STORE} & \left\{ \text{EVERY}^{\{\mathbf{person}(i)\}}_{\{i\}} \right\} \end{bmatrix}, \begin{bmatrix} \text{STORE} & \left\{ \text{SOME}^{\{\mathbf{person}(j)\}}_{\{j\}} \right\} \end{bmatrix} \right\rangle \end{bmatrix}$

$$\text{c.}\quad \left[\text{CONTENT}\quad \begin{bmatrix}\text{QUANTS} & \left\langle \text{EVERY}^{\{\textbf{person}(i)\}}_{\{i\}}\ ,\ \text{SOME}^{\{\textbf{person}(j)\}}_{\{j\}} \right\rangle \\ \text{NUCL} & \mathit{Love}(i,j)\end{bmatrix}\right]$$

$$\text{d.}\quad \left[\text{CONTENT}\quad \begin{bmatrix}\text{QUANTS} & \left\langle \text{SOME}^{\{\textbf{person}(j)\}}_{\{j\}},\ \text{EVERY}^{\{\textbf{person}(i)\}}_{\{i\}} \right\rangle \\ \text{NUCL} & \mathit{Love}(i,j)\end{bmatrix}\right]$$

Lexical retrieval, as defined by Pollard and Yoo, allows both the scopings in (21c,d) as the semantic content of the verb. That is, the proper set of scopings is allowed if we constrain (non-raising) verbs as follows:

(22) Lexical Quantifier Retrieval (first formulation):

$$\mathit{verb} \Rightarrow \begin{bmatrix}\text{ARG-ST} & \left\langle \left[\text{STORE}\quad \Sigma_1\right], \ldots, \left[\text{STORE}\quad \Sigma_1\right] \right\rangle \\ \text{CONTENT} & \left[\text{QUANTS}\quad \texttt{iteration}(\Sigma_1 \cup \ldots \cup \Sigma_n)\right]\end{bmatrix}$$

This lexical retrieval approach to quantifier scoping, which differs in key respects from Cooper's treatment in terms of phrasal retrieval, may seem unfamiliar at first, but, as Pollard and Yoo show, it provides a straighforward account of examples like (23a,b), where a subject quantifier may take narrow scope with respect to a syntactically lower element (the verb *appears* in (23a); the adverb in (23b)):

(23) a. A unicorn appears to be approaching.

b. A unicorn is always approaching.

The raising verbs *appears* and *is* identify their subject argument with that of their VP complement. Hence in both cases the verb *approaching* can retrieve the existential quantifier and incorporate it into its semantics (in accordance with Lexical Quantifier Retrieval in (22)). This produces the correct result that the subject quantifier, though syntactically superior, has narrow scope with respect to *appear* or *always* in (23).

It is straighforward to modify Lexical Quantifier Retrieval so as to allow resumption of quantifiers. Given that we limit ourselves in this paper to resumptive readings of negative quantifiers, we will only define resumption for sets of anti-additive quantifiers. Suppose we define a relation that maps a set of quantifiers to an iteration of quantifiers in such a way as to allow any subset of anti-additive quantifiers to undergo resumption. First, we generalize our treatment of variables so that every generalized quantifier specifies not just its variable, but a set of variables. The `retrieve` relation can then be defined as follows:

(24) `Retrieve`:

Given a set of generalized quantifiers Σ and a partition of Σ into two sets Σ_1 and Σ_2, where $\Sigma_2 = \{NO^{R_1}_{\sigma_1}, \ldots, NO^{R_n}_{\sigma_n}\}$, then

$\mathtt{retrieve}(\Sigma) =_{\text{def}} \mathtt{iteration}(\Sigma_1 \cup \text{Res}(\Sigma_2))$

This relation then serves to constrain a verb's QUANTS value in the following revision of Lexical Quantifier Retrieval:

(25) Lexical Quantifier Retrieval (second formulation)

$$verb \;\Rightarrow\; \begin{bmatrix} \text{ARG-ST} & \left\langle \begin{bmatrix}\text{STORE} & \Sigma_1\end{bmatrix}, \ldots, \begin{bmatrix}\text{STORE} & \Sigma_n\end{bmatrix} \right\rangle \\ \text{CONTENT} & \begin{bmatrix}\text{QUANTS} & \mathtt{retrieve}(\Sigma_1 \cup \ldots \cup \Sigma_n)\end{bmatrix} \end{bmatrix}$$

The analysis works as follows. The verb's arguments merge their stored quantifiers into a set, and the verb's QUANTS value is a list of quantifiers determined from this set by the relation `retrieve`.[4] `Retrieve` does not fix which subset of quantifiers will undergo resumption. It is always possible to let Σ_2 be the empty set, for example. In this case, the result will be an iteration of quantifiers, even if these are anti-additive, as in (26b), with the corresponding semantics in (26c).[5]

(26) a. Personne n'aime personne. [DN]
No one is such that they love no one

b.
$$\begin{bmatrix} \text{PHONOLOGY} & \langle \textit{n'aime} \rangle \\ \text{ARG-ST} & \left\langle \begin{bmatrix}\text{STORE} & \{NO^{\{Person(x)\}}_{\{x\}}\}\end{bmatrix}, \begin{bmatrix}\text{STORE} & \{NO^{\{Person(y)\}}_{\{y\}}\}\end{bmatrix} \right\rangle \\ \text{CONTENT} & \begin{bmatrix} \text{QUANTS} & \left\langle \text{NO}^{\{Person(x)\}}_{\{x\}}, \text{NO}^{\{Person(y)\}}_{\{y\}}\} \right\rangle \\ \text{NUCLEUS} & Love(x,y) \end{bmatrix} \end{bmatrix}$$

c. $\neg\exists x\, \neg\exists y$ Love(x, y)

Alternatively, if we let Σ_2 be the doubleton set containing both anti-additive quantifiers (and Σ_1 be the empty set), then `retrieve` allows the following resumptive interpretation for the verb, and hence for the same sentence:

[4] Note that the enumeration is meant to allow for the empty ARG-ST list, in which case the verb's QUANTS list is empty.

[5] Note that monotone decreasing quantifiers do not allow inverse scope readings, as observed by Liu (1990). A discussion of the constraints on iteration of quantifiers is beyond the scope of this paper, cf. Szabolsci 1997 and Beghelli and Stowell 1997 for a proposal.

(27) a. Personne n'aime personne. [NC]

b.
$$\begin{bmatrix} \text{PHONOLOGY} & \langle \textit{n'aime} \rangle \\ \text{ARG-ST} & \left\langle \begin{bmatrix}\text{STORE} & \{NO^{\{Person(x)\}}_{\{x\}}\}\end{bmatrix}, \begin{bmatrix}\text{STORE} & \{NO^{\{Person(y)\}}_{\{y\}}\}\end{bmatrix} \right\rangle \\ \text{CONTENT} & \begin{bmatrix} \text{QUANTS} & \left\langle \mathrm{NO}^{\{Person(x),Person(y)\}}_{\{x,y\}} \right\rangle \\ \text{NUCLEUS} & Love(x,y) \end{bmatrix} \end{bmatrix}$$

c. $\neg\exists x\, \exists y$ Love(x, y)

This analysis is quite general. For example, it extends to cases of the sort we saw earlier, where the stored quantifier is properly contained within an argument of the verb:

(28) a. Personne ne dit ça [à personne] [ambiguous]
No one NE says that to no one
= No one says that to anyone [NC]
= No one says that to no one [DN]

b. Personne n'aime [le fils de personne] [ambiguous]
No one NE-loves the son of no one
= No one loves the son of anyone [NC]
= Everyone loves the son of someone [DN]

This follows if we simply assume that quantifiers are passed up from the STORE of nouns like *personne* to the STORE of NPs and PPs that contain them.

Second, we follow recent work by Przepiórkowski (1999) and Bouma et al. (to appear), who show that there is much to be gained (e.g. a general account of the case marking systems of languages like Korean, Finnish, Polish, etc. and a uniform treatment of *wh*-extraction phenomena) by extending ARG-ST lists to include various adverbials that are usually analyzed as modifiers. In this way, such adverbials contribute their stored quantifiers to the verb's content in exactly the same way that subjects and complements do. Thus, without introducing any further devices, our account immediately extends to cases like (29).

(29) Personne n'a jamais dit ça.
No-one NE-has never said that [ambiguous]
= No one has ever said that [NC]
= No one has never said that [DN]

Finally, note that if we treat adverbs like *presque* as quantifiers of type 0, i.e. as having an empty restriction set and as binding the empty set (of variables),[6] then we have an immediate account for the observation noted above regarding examples like (30):

(30) a. Presque personne n'a rien dit. [ambiguous]
Almost no one NE-has nothing said.
= Almost no one said anything [NC]
= Nearly everyone said something [DN]

b. Personne n'a presque rien dit.
[DN or NC such that *presque* takes widest scope]
No one NE-has almost nothing said.
= No one said almost nothing [DN]
= Almost no one said anything [NC]
= *No one said almost anything

On our treatment, the amalgamation of the stores of the verb's arguments is here a tripleton set containing $\mathrm{ALMOST}^{\emptyset}_{\emptyset}$, $\mathrm{NO}^{\{Person(x)\}}_{\{x\}}$, and $\mathrm{NO}^{\{Thing(y)\}}_{\{y\}}$. The anti-additive quantifiers can be iterated or they may undergo resumption. If they are iterated, then ALMOST is free to scope wider or narrower. And if the two NOs undergo resumption, then ALMOST must take wide scope with respect to the resumptive quantifier as a whole (compare our discussion of the examples in (6)). The only permissible scopings are those shown in (31):[7]

(31)

a. $\langle ALMOST^{\emptyset}_{\emptyset}, \mathrm{NO}^{\{Person(x)\}}_{\{x\}}, \mathrm{NO}^{\{Thing(y)\}}_{\{y\}}\rangle$

b. $\langle NO^{\{Person(x)\}}_{\{x\}}, \mathrm{ALMOST}^{\emptyset}_{\emptyset}, \mathrm{NO}^{\{Thing(y)\}}_{\{y\}}\rangle$

c. $\langle ALMOST^{\emptyset}_{\emptyset}, \mathrm{NO}^{\{Thing(y)\}}_{\{y\}}, \mathrm{NO}^{\{Person(x)\}}_{\{x\}}\rangle$

[6] We return to this type of analysis in the next section.

[7] Technically, there are even more permissible scopings, e.g. (i)–(iii):

(i) $\langle NO^{\{Person(x)\}}_{\{x\}}, \mathrm{NO}^{\{Thing(y)\}}_{\{y\}}, \mathrm{ALMOST}^{\emptyset}_{\emptyset}\rangle$

(ii) $\langle NO^{\{Thing(y)\}}_{\{y\}}, \mathrm{NO}^{\{Person(x)\}}_{\{x\}}, \mathrm{ALMOST}^{\emptyset}_{\emptyset}\rangle$

(iii) $\langle NO^{\{Thing(y),Person(x)\}}_{\{y,x\}}, ALMOST^{\emptyset}_{\emptyset}\rangle$

However, these scopings are ruled out independently by a constraint on *almost* which requires it to be a modifier of something. As a result, it cannot take narrowest scope in a sequence of quantifiers.

d. $\langle NO_{\{y\}}^{\{Thing(y)\}}, ALMOST_{\emptyset}^{\emptyset}, \mathrm{NO}_{\{x\}}^{\{Person(x)\}}\rangle$

e. $\langle ALMOST_{\emptyset}^{\emptyset}, \mathrm{NO}_{\{y,x\}}^{\{Thing(y),Person(x)\}}\rangle$

The anti-additive quantifiers can be iterated, but there will be no way for ALMOST to 'break up' the resumptive quantifier in (31e). Thus it follows that there is no concord interpretation like the starred one in (30b).

4 The Role of Negation in Negative Concord

In the studies on resumption within the polyadic quantifier approach (May 1989, Van Benthem 1989, Keenan and Westerståhl 1997), there is not much discussion of constraints on the types of quantifiers that undergo absorption. For *wh*-expressions, absorption seems to be restricted to variable binding operators like *who*, *which N*, etc. (cf. May 1989).[8] Non-variable binding operators like *if*, or *whether* do not participate in this process. Negative concord is different in that not only negative quantifiers, but also non-variable binding negative operators such as sentential negation (section 4.1) and negative prepositions (section 4.2) participate in negative concord.

4.1 Embedding under sentential negation

In order to treat a mixture of negation and negative quantifiers in a polyadic approach, we need to determine the type of negation. Given that the type of a quantifier reflects the number of variables it binds, we can take non-variable binding operators such as negation to be quantifiers with adicity zero, or quantifiers of type $\langle 0\rangle$, as we did for *almost* at the end of the previous section.

- Non-variable binding operators such as negation are treated as quantifiers of type $\langle 0\rangle$.

We can now extend our absorption rule to allow resumption of quantifiers of different types. We can define a mixed rule as follows:

- Absorption of a sequence of k type $\langle n,n\rangle$ quantifiers and l type $\langle m,m\rangle$ quantifiers from the N-store leads to the construction of a resumptive quantifier of type $\langle (k\times n)+(l\times m),(k\times n)+(l\times m)\rangle$.

The variant of this rule which we need in the combination of sentential negation and concord items is the following:

[8] These may in fact not be variable-binding operators in the same sense as generalized quantifiers. See, for example, Ginzburg and Sag to appear.

- Absorption of a sequence of k type $\langle 1,1 \rangle$ quantifiers and l type $\langle 0 \rangle$ quantifiers leads to the construction of a resumptive quantifier of type $\langle k,k \rangle$.

Adding zero does not add anything, so the result of combining negation and a sequence of negative quantifiers is that negation does not affect the type of the polyadic quantifier. This means that negation is semantically empty in a concord context.

At first, the semantic emptiness of negation might seem like a somewhat odd result, given the familiar 'local' approaches to negative concord in which sentential negation plays an important role as the licensor of the concord item (cf. Laka 1990, Ladusaw 1992, Przepiórkowski and Kupść 1997 and others). However, as Ladusaw (1992: footnotes 10 and 11) admits, that is in fact an important problem for the local approach to negative concord. The participation of sentential negation is subject to considerable cross-linguistic variation, which means that we have to assume that each language has its own set of licensing conditions on concord items. Given that licensing conditions on negative polarity are by and large the same across languages, and that variation only obtains within strict limits, this is not a very attractive result. The polyadic approach provides a more principled explanation of the cross-linguistic variation we find: given that negation is semantically empty in concord contexts, languages are free to include or exclude simple negation from the concord system.

As far as French is concerned, we observed already that the combination of *pas* and a concord item usually leads to a double negation reading, compare (7b) above, repeated here as (32a). However, examples like (8a–c), repeated as (32b–d), illustrate that in certain varieties and certain constructions, concord readings are available as well:

(32) a. Il ne va pas nulle part, il va à son travail [DN only]
He does not go nowhere, he goes to work

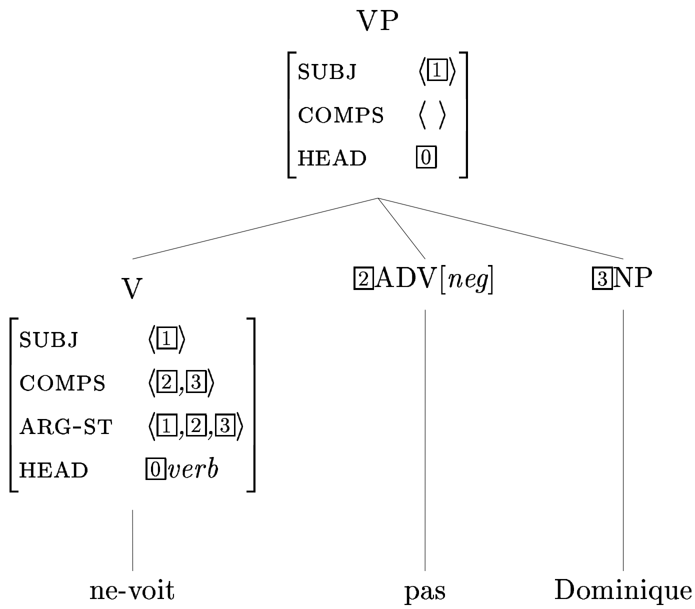
b. Il ne veut pas que personne soit lésé. [restricted varieties][NC]
He NE wants not that no one is-SUBJ wronged.
= He does not want anyone to be wronged

c. S'il y a quelque chose,
il fera pas d'cadeau à personne. [restricted varieties][NC]
If there is something, he will not give a present to no one
= If there is something, he will not grant anyone a favor

d. Je n'ai pas donné le moindre renseignement à personne. [NC]
I NE have not given the least information to no one
= I have not given the least information to anyone

To accommodate such variation, we build on the work of Kim and Sag (1995) and Warner (1993, 2000), who argue that French *pas* and English *not* are not modifiers of the finite verb, but rather are selected as a complement. The *pas*-selecting finite verbal forms can either be derived by lexical rule (Kim and Sag) or else treated via lexical types (Warner). In either analysis, the argument structure of such verbs is augmented by a negative adverb in the second position. This allows finite verbs to project phrases like the following.

(33)



In this head-complement structure (like the one ilustrated earlier), the non-head daughters correspond to the members of the head daughter's COMPS list and the mother's COMPS value is the empty list. The head daughter's HEAD value matches that of its mother, in keeping with the Head Feature Principle.

We will assume that *pas* is lexically represented as in (34).

(34) $\begin{bmatrix} \text{HEAD} & adv \\ \text{CONT} & \boxed{1}NO^{\emptyset}_{\emptyset} \\ \text{STORE} & \{\boxed{1}\} \end{bmatrix}$

Hence, because *pas* is part of the verb's ARG-ST list, its negative quantifier will automatically be included in the verb's QUANTS list by the Lexical Quantifier Principle formulated earlier. This provides the basis for treating the many Romance varieties where simple negation participates in resumption: because the verb amalgamates the stored 0-place quantifier of the negative adverb, this negative quantifier functions like any other—it may participate in resumption (giving the NC reading) or not (giving the DN reading). This is how things work in Québéquois French, and in certain dialects of French as illustrated in (7a–c, e).

We assume that *pas* does not participate in the system of concord of standard French, because it is not needed semantically. Our treatment of standard French involves a parochial constraint requiring that *pas*'s stored quantifier must be in the verb's QUANTS list. This constraint can be incorporated into the lexical rule analysis (as a condition on the rule output) or simply added to the type constraint already required for *pas*-selecting finite verbs. Warner's (2000) system has a type *not-arg*, for example, whose French analog *pas-arg* could be constrained as follows .

(35) *pas-arg* ⇒

$$\begin{bmatrix} \text{CONT}|\text{QUANTS} & \langle \ldots, \boxed{1}, \ldots \rangle \\ \text{ARG-ST} & \left\langle \text{NP}, \begin{matrix}\text{ADV}\\ \begin{bmatrix}\text{CONT} & \boxed{1}\end{bmatrix}\end{matrix} \right\rangle \end{bmatrix}$$

The effect of (35) is that the 0-place quantifier that serves as the content of *pas* cannot participate in resumption, for it must be preserved intact in the QUANTS list of any verb that selects *pas* as a complement (this is precisely the class denoted by the type *pas-arg*). Note further that this constraint is independently motivated, as something must ensure that the negative quantifier introduced by *pas* never stays in STORE to be retrieved higher in the tree, a possibility that is available for other negative quantifiers, as we will see below.

A Romance variety might have this constraint or not. In the former case, e.g. in standard French, any number of negative quantifiers other than *pas* may be absorbed in the retrieval that determines a verb's QUANTS list. This predicts, for example, that double *personne* examples like (26) will be ambiguous, as analyzed above. But because simple negation cannot undergo resumption, only a double negation reading is possible for examples like (32a). This analysis also excludes NC-readings of examples like (32b-c) in standard French, as desired.

We need to postulate one exception to the *pas*-rule for standard French in order to account for examples like (32d). The construction shows that *pas* is required in order to license the negative polarity item *le moindre renseignement*. Modulo the exceptions discussed in de Swart (1998), a negative polarity item needs to be licensed by a c-commanding licensor with the appropriate semantic licensing properties. We know that *personne* has the appropriate semantic properties to function as a licensor for *le moindre renseignement*. However, *personne* does not c-command the negative polarity item. The presence of *pas* in a concord chain is thus licensed in order to save the well-formedness of the construction by fulfilling the requirements of the negative polarity item.

Of course there is considerable variation regarding negation systems, not only among varieties of French, but also across the Romance languages quite generally. In Catalan, for example, the use of negation becomes optional in a concord context.

So we don't have a rule similar to (35). As a result, both (36a) and (36b) are grammatical, and they have the same meaning (from Vallduví 1994):

(36) a. Cap d'ells vindrà [Catalan]
None of them comes

b. Cap d'ells no vindrà pas
None of them not comes
= None of them comes

In other Romance languages, negation is required when the negative quantifier is in a postverbal position, but is optional or even excluded when the negative quantifier is in preverbal position, e.g. Italian, Spanish:

(37) a. No funciona nada [Spanish]
Not works nothing
= Nothing works

b. Nada funciona
Nothing works
= Nothing works

(38) a. Gianni *(non) dice niente a nessuno [Italian]
Gianni *(not) says nothing to no one
= Gianni does not tell anyone anything

b. Nessuno (*non) legge niente.
No one (*not) reads nothing
= No one reads anything

Ladusaw (1992) argues that negation functions as a scope marker in these languages. A VP-internal (i.e. postverbal) negative quantifier in Spanish or Italian cannot take sentential scope in the absence of a negation marker, but obviously, this problem vanishes for negative quantifiers in a VP-external (i.e. preverbal) position. Haegeman (1995) develops a similar account in a transformational framework, claiming that Italian and Spanish obey the so-called NEG-criterion at S-structure.

Evidence that *ne*-marking is reanalyzed as a scope marker can also be found in French. Kayne (1981) argues that the contrast between (39a) and (39b) involves the scope of the negation:

(39) a. Je ne demande qu'ils arrêtent personne
I NE ask that they arrest no one
= I don't ask them to arrest anyone

b. Je demande qu'ils n'arrêtent personne
I ask that-they NE-arrest no one
= I ask that they arrest no one

The occurrence of *ne* in the embedded clause indicates the upper limit of the semantic scope of negation. In order to give negation scope over the sentence as a whole, we can 'insert' *ne* in the main clause.

To accommodate these observations, we may assume a lexical constraint on French *ne*-forms:

(40) *ne-verb* $\Rightarrow$ $\left[\text{CONT}|\text{QUANTS} \quad \langle ...,\textit{neg-quant},... \rangle\right]$

This guaranties that any *ne*-prefixed verb form in standard French must retrieve at least one negative quantifier into its QUANTS list. This constraint immediately accounts for contrasts like those in (39) and expresses a generalization true of all *ne*-verbs in our analysis. Although *ne* functioned originally as the marker of sentential negation, it has lost its negative force over time, and has in fact become optional in spoken French. The real bearer of simple negation in modern French is *pas*.

In order to flesh out this analysis of examples like (39a), where a negative quantifier (*personne*) is scoped in the higher clause, we must revise our lexical retrieval constraint to permit a quantifier to remain in store, instead of being retrieved and scoped at the lowest level of structure. Again, this is independently motivated, as quantifiers in general are not strictly clause-bounded in their scope, as is well known. Hence we may reformulate our lexical retrieval constraint as follows:[9]

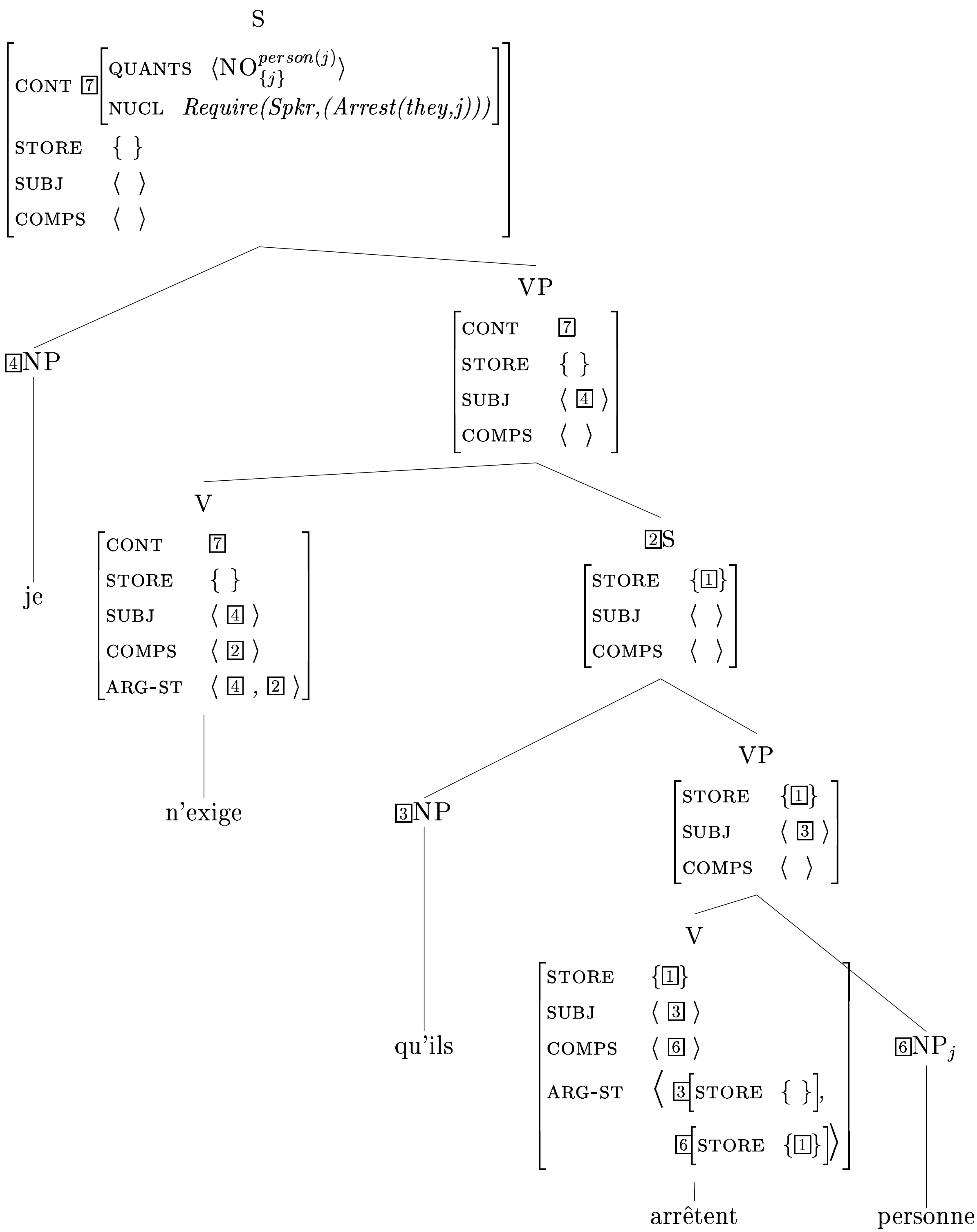
(41) Lexical Quantifier Retrieval (third formulation):

$$\textit{verb} \Rightarrow \begin{bmatrix} \text{ARG-ST} & \left\langle \left[\text{STORE} \quad \Sigma_1\right], \ldots, \left[\text{STORE} \quad \Sigma_n\right] \right\rangle \\ \text{CONTENT} & \left[\text{QUANTS} \quad \texttt{retrieve}((\Sigma_1 \cup \ldots \cup \Sigma_n) \dot{-} \Sigma)\right] \\ \text{STORE} & \Sigma \end{bmatrix}$$

What (41) does is make retrieval optional. That is, the quantifiers stored in a verb's arguments may be retrieved into the verb's QUANTS list (possibly undergoing resumption), but the stored quantifiers that are not in the verb's QUANTS list are in the verb's STORE set. The verb's STORE value will then be passed up to the VP and the S that it projects, in accordance with the Head Feature Principle, if this is generalized in the fashion proposed by Ginzburg and Sag (to appear). Alternatively, a separate STORE inheritance principle is required, as proposed by Pollard and Sag (1994). The resulting treatment of (39a) is sketched in (42).

[9]We use '$\dot{-}$' to designate a relation of contained set difference that is identical to the familiar notion of set difference ($\Sigma_1 - \Sigma_2$ = the set of all elements in Σ_1 that are not in Σ_2), except that $\Sigma_1 \dot{-} \Sigma_2$ is defined only if Σ_2 is a subset of Σ_1.

(42)

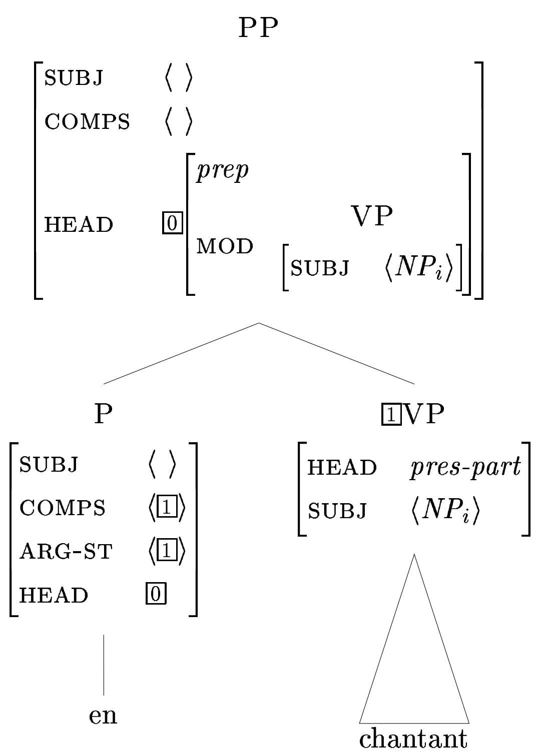

S
CONT 7 QUANTS ⟨NO person(j) {j}⟩
NUCL Require(Spkr,(Arrest(they,j)))
STORE { }
SUBJ ⟨ ⟩
COMPS ⟨ ⟩
4NP
je
VP
CONT 7
STORE { }
SUBJ ⟨ 4 ⟩
COMPS ⟨ ⟩
V
CONT 7
STORE { }
SUBJ ⟨ 4 ⟩
COMPS ⟨ 2 ⟩
ARG-ST ⟨ 4 , 2 ⟩
n'exige
2S
STORE {1}
SUBJ ⟨ ⟩
COMPS ⟨ ⟩
3NP
qu'ils
VP
STORE {1}
SUBJ ⟨ 3 ⟩
COMPS ⟨ ⟩
V
STORE {1}
SUBJ ⟨ 3 ⟩
COMPS ⟨ 6 ⟩
ARG-ST ⟨ 3[STORE { }], 6[STORE {1}] ⟩
arrêtent
6NPj
personne

The inheritance of stored quantifiers is subject to a variety of constraints that we cannot do justice to here. For present purposes, it will suffice to assume that subjunctive verb forms freely allow negative quantifiers (other than the 0-place *pas*) to be passed up to a higher scope domain and that present indicative verb forms are constrained to have an empty STORE. This makes the (possibly too) strong claim that finite verbs are absolute scope islands. Intermediate positions, where the STORE value of a finite verb is restricted, but not required to be empty, are certainly possible.

4.2 Embedding under Negative Prepositions

One advantage of the extension of our rule of resumption to include non-variable binding operators is that we can integrate embedding of NCIs under *sans* 'without' in our system. We take *sans* to be the negative counterpart of *en*, which combines with a present participle to build an intersective modifier, as shown in (43):

(43)



We will assume that there is an intersective modification construction (a head-adjunct structure) that combines PPs like (43) with VPs to produce VPs like the one bracketed in (44).[10]

[10] It is not essential that we adopt this analysis of intersective modification or the conjunctive semantics for it proposed below. Our proposal is compatible, for example, with the modification

(44) Anne [est partie *en chantant*].

If we assume, for simplicity, that the intersective modification construction simply conjoins the semantics of the modifier and the modified VP, then the lexical entry for the preposition *en* can be formulated as in (45).

(45) *en*

$$\left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll} \text{MOD} & \begin{array}{c} \text{VP} \\ \left[\text{SUBJ} \quad \langle NP_i \rangle\right] \end{array} \end{array}\right] \\ & \\ \text{ARG-ST} & \left\langle \begin{array}{c} \text{VP} \\ \left[\begin{array}{ll} \text{SUBJ} & \langle NP_i \rangle \\ \text{HEAD} & \textit{pres-part} \\ \text{CONT} & \boxed{1} \end{array}\right] \end{array} \right\rangle \\ & \\ \text{CONT} & \boxed{1} \end{array}\right]$$

Note that this lexical entry (presumably derived by principles of more general scope) fixes the basic semantics properly by (1) identifying the preposition's content with that of the participial VP and (2) coindexing the SUBJ element of the participial VP with that of the VP that the projected PP will modify (its MOD value). The PP's semantics is identical to that of the participial VP. Hence assuming simple conjunction as the semantics of the intersective modifier construction, we have the following semantics (ignoring tense, among other things) for (44):

(46) *Leave(Anne) & Sing(Anne)*

Along similar lines, we treat *sans* as a preposition which combines with an infinitival VP complement to build an intersective modifier. The semantic contribution of *sans* then reduces to simple negation, allowing the interpretation of (47) to be analogous to (46):

(47) a. Anne est partie sans chanter
Anne has left without singing.

b. *Leave(Anne) &* $NO^{\emptyset}_{\emptyset}$*(Sing(Anne))*

Note that in our earlier treatment of quantifiers and negation, the lexical head (always a verb until now) provides the locus for the stored quantifiers in the head's arguments to be assigned scope. What is unusual about *sans* is that it introduces

analysis proposed by Kasper (1999), which distinguishes the internal and external content of the modifier.

a 0-place negative quantifier (simple negation) into the semantics—this is intrinsic, not inherited from any element within the argument of *sans*. To accommodate such lexical contribution to scope, we will have to revise lexical quantifier retrieval one last time. First, we introduce a feature LEXICAL-QUANTIFIER (LEX-QUANT), whose value will be empty for most words, but in the case of *sans* will be a singleton set containing the 0-place negative quantifier. *sans* also acts as a scope island and hence its lexical entry will require an empty STORE, as shown in (48).

(48) *sans*

$$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{MOD} & \begin{array}{c} \text{VP} \\ \begin{bmatrix} \text{SUBJ} & \langle NP_i \rangle \end{bmatrix} \end{array} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \begin{array}{c} \text{VP} \\ \begin{bmatrix} \text{SUBJ} & \langle NP_i \rangle \\ \text{HEAD} & \textit{infin} \\ \text{CONT} & \begin{bmatrix} \text{QUANTS} & \langle\ \rangle \\ \text{NUCL} & \boxed{1} \end{bmatrix} \end{bmatrix} \end{array} \right\rangle \\ \text{LEX-QUANT} & \{NO^{\emptyset}_{\emptyset}\} \\ \text{CONT} & \begin{bmatrix} \text{NUCL} & \boxed{1} \end{bmatrix} \\ \text{STORE} & \emptyset \end{bmatrix}$$

Two other properties of (48) are noteworthy: (1) the content of *sans* takes its nucleus from its infinitival VP argument and (2) that argument is required to have no quantifiers on its QUANTS list. The latter constraint prevents unwanted spurious ambiguity that would otherwise arise, given that *sans* allows quantifier retrieval.

We may now generalize lexical quantifier retrieval to apply to all words whose content is propositional (technically, a *state-of-affairs* (*soa*) in our system). We do this as follows:

(49) Lexical Quantifier Retrieval (final formulation):

$$\textit{soa-wd} \Rightarrow \begin{bmatrix} \text{ARG-ST} & \left\langle \begin{bmatrix} \text{STORE} & \Sigma_1 \end{bmatrix}, \ldots, \begin{bmatrix} \text{STORE} & \Sigma_n \end{bmatrix} \right\rangle \\ \text{LEX-QUANT} & \Sigma_0 \\ \text{CONTENT} & \begin{bmatrix} \text{QUANTS} & \mathtt{retrieve}((\Sigma_0 \cup \ldots \cup \Sigma_n) \dot{-} \Sigma) \end{bmatrix} \\ \text{STORE} & \Sigma \end{bmatrix}$$

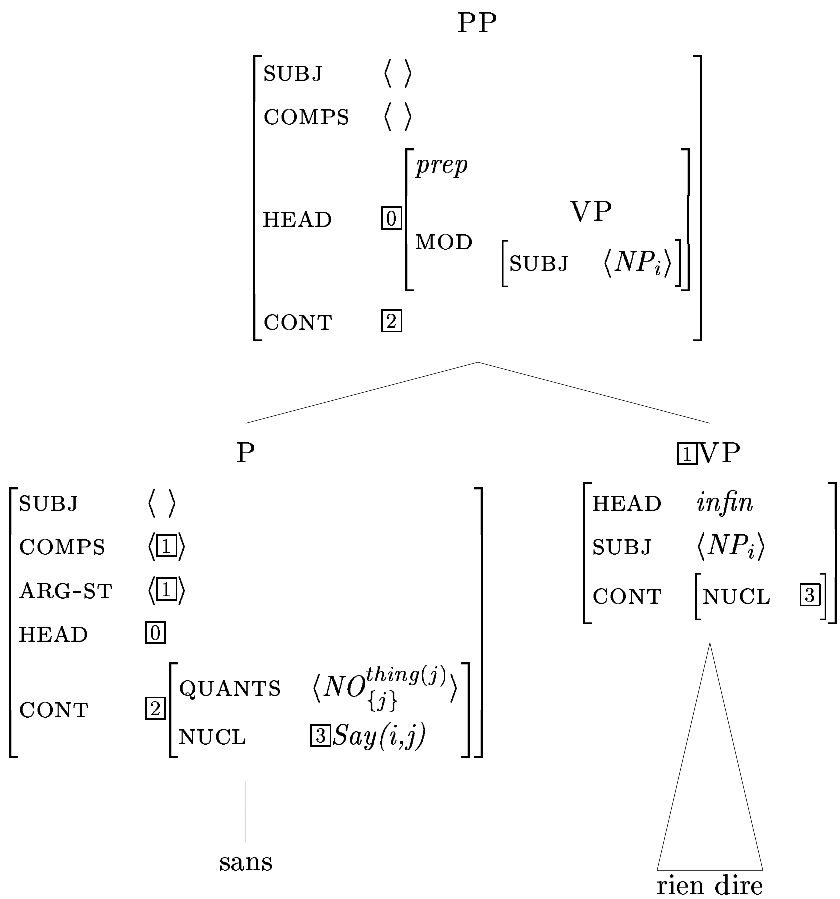
This is a minor revision that allows a word like *sans* to contribute its own quantification to the pool of quantifiers that can be retrieved into its QUANTS list. Note that since *sans* has an empty STORE value, its lexical quantifier cannot be passed up to be retrieved at a higher level of structure.

This analysis immediately predicts the possibility of a concord reading for examples like (50).

(50) Anne est partie sans rien dire
Anne has left without nothing say
= Anne has left without saying anything

That is, given that *rien*'s quantifier can be in the store of the infinitival VP (in fact it must be, since *sans* requires that the QUANTS list of that VP be empty), the lexical 0-place quantifier of *sans* and *rien*'s quantifier can undergo resumption. The result is simply a monadic quantifier, as illustrated in (51).

(51)



This construction is productive, as is shown by the resumptive quantifiers built up in contexts like (52):

(52) a. Il est parti sans rien dire à personne.
He has left without nothing say to no one
= He left without saying anything to anyone

b. Il part sans jamais rien dire à personne.
He leaves without never nothing say to no one
= He leaves without ever saying anything to anyone

Our treatment of *sans* thus integrates all such cases into the general analysis of negative concord. We do not need to treat *sans* as illustrating a negative polarity use of concord items (as proposed, for example, by Muller 1991: 264–265).

Our account of negative concord also explains certain other facts discussed in the literature, usually accounted for by appeal to 'Σ-sequences'. For example, concord items embedded under *sans* can participate in a 'negative chain', but embedding *sans* under concord items higher in the tree always leads to a double negation reading. Compare the previous examples with (53):

(53) a. Personne n'est parti sans rien dire [DN]
No one NE-has left without nothing say
= No one left without saying nothing

b. Je n'ai invité aucun étudiant qui n'a rien fait [DN]
I NE-have invited no student who NE has nothing done
= I haven't invited any student who did not do anything

In our analysis, the negation introduced by *sans* takes scope over the modifier only. From the empty store constraint in the lexical entry of *sans* it follows that this 0-place negative quantifier can never be in the STORE at the higher level of structure, where it would have to be, were it to be retrieved via resumption into the QUANTS list of the higher verb. The same is true of the relative clause in (53b), as the finite indicative verb within the relative clause must also have an empty STORE, as noted earlier.

5 Conclusion

The interpretation of negation and negative concord in a polyadic quantifier framework has a number of advantages over treatments of negative concord in terms of negative polarity. Within the framework developed in this paper, we can derive the double negaton and the concord readings as different ways of building up the polyadic quantifier, rather than postulating a contextual ambiguity which lacks independent motivation. This approach has the additional advantage of reducing the difference between concord and non-concord languages to a preference for one mode of composition or the other. Furthermore, the extension of the resumptive polyadic approach to include non-variable binding operators allows us to treat negation as semantically empty in concord contexts. The observation that the negation marker can take up a syntactic function (e.g. as a scope marker) in the absence of a semantic role explains why languages vary in the participation of negation in negative concord.

The embedding of the polyadic quantifier analysis in an HPSG grammar makes it possible to treat iteration and resumption as two variants of the rule which governs retrieval of quantifiers from the store where they are entered. The lexical treatment of negative quantifiers allows us to formulate specific rules for type $\langle 0 \rangle$ quantifiers which capture the restrictions on the interpretation and the scope of *pas* and *sans* in standard French. Of course these lexical rules are subject to dialectal and cross-linguistic variation, which circumvents an important problem the local approach faces, namely the observation that sentential negation does not behave in the same way across concord languages.

Finally, the HPSG implementation of the polyadic approach developed in this paper opens new ways to talk about the famous Jesperson cycle, i.e. the observation that certain existential expressions become negative polarity items and finally shift entirely to negative quantifiers in the development of a language. This diachronic phenomenon can be captured in terms of changes in the interpretation of a sequence of quantifiers from iteration (of multiple existential quantifiers within the scope of negation) to absorption (of negative quantifiers) and back to iteration (of negative quantifiers).

References

Barwise, Jon and Cooper, Robin. 1981. Generalized quantifiers and natural language. *Linguistics and Philosophy* 4: 159–219.

Beghelli, Filippo and Stowell, Tim. 1997. Distributivity and negation: the syntax of *each* and *every*. In *Ways of Scope Taking*, ed. by Anna Szabolcsi. Dordrecht: Kluwer Academic Press. Pp. 71–197.

van Benthem, Johan. 1986. *Essays in Logical Semantics.*, Dordrecht: Reidel.

van Benthem, Johan. 1989. Polyadic quantifiers. *Linguistics and Philosophy* 12: 437–464.

Bouma, Gosse, Robert Malouf, and Ivan A. Sag. to appear. Satisfying constraints on extraction and adjunction, parts I and II. *Natural Language and Linguistic Theory.*

Cooper, Robin. 1983. *Quantification and Syntactic Theory.* Dordrecht: Reidel.

Corblin, Francis. 1996. Multiple negation processing in natural language. *Theoria* 17: 214–259.

Corblin, Francis and Derzhansky, Ivan. 1997. Multiple negation, optional arguments and the reification of eventualities. In *Empirical Issues in Formal Syntax and Semantics*, ed. by Francis Corblin, Danièle Godard and Jean-Marie Marandin. Bern: Peter Lang.

Davis, Anthony. 1996: *Lexical Semantics and Linking in the Hierarchical Lexicon.* Doctoral dissertation, Stanford University.

Depréz, Viviane. 1997. Two types of negative concord. *Probus* 9: 103–143.

Ginzburg, Jonathan, and Ivan A. Sag. to appear. *English Interrogative Constructions.* Stanford: CSLI Publications.

Grevisse, Maurice. 1993. *Le Bon Usage* (13th revised edition). Paris: Duculot.

Haegeman, Liliane. 1995. *The Syntax of Negation.* Cambridge: Cambridge University Press.

Haegeman, Liliane and Zanuttini, Rafaella. 1996. Negative concord in West Flemish. In *Parameters and Functional Heads*, ed. by Andrea Belletti and Luigi Rizzi. Oxford: Oxford University Press. Pp. 117–179.

Higginbotham, James and May, Robert. 1981. Questions, quantifiers and crossing. *The Linguistic Review* 1: 41–79.

Kasper, Robert. 1999. Semantics of recursive modification. *Journal of Linguistics.* [CHECK THIS]

Kayne, Richard. 1981. ECP-extensions. *Linguistic Inquiry* 12: 349–371.

Keenan, Edward 1987. Unreducible n-ary quantifiers in natural language. In *Generalized Quantifiers*, ed. by Peter Gärdenfors. Dordrecht: Reidel. Pp. 109–150.

Keenan, Edward. 1992. Beyond the Frege boundary. *Linguistics and Philosophy* 15: 199–221.

Keenan, Edward and Dag Westerståhl. 1997. Generalized quantifiers in linguistics and logic. In *Handbook of Logic and Language*, ed. by Johan van Benthem and Alice ter Meulen. Amsterdam: Elsevier. Pp. 837–893.

Kim, Jongbok, and Ivan A. Sag. 1995. The parametric variation of English and French negation. In *Proceedings of the Fourteenth Annual Meeting of the West Coast Conference on Formal Linguistics.* Stanford: SLA, CSLI Publications. Pp. 303–317.

Ladusaw, William. 1992. Expressing negation. *Proceedings of SALT 2.* Columbus: The Ohio State University. Pp. 237–259.

Laka Mugarza, Itziar. 1990 *Negation in Syntax.* Doctoral dissertation, MIT.

Laka Mugarza, Itziar. 1993. Negative fronting in Romance. In *Linguistic perspectives on the Romance*, ed. by William Ashby. Amsterdam: John Benjamins. Pp. 315–333.

Lindström, P. 1966. First order predicate logic with generalized quantifiers. *Theoria* 35: 1–11.

Liu, F.-H. 1990. *Scope dependency in English and Chinese.* Doctoral dissertation, University of California at Los Angeles.

Manning, Christopher, and Ivan A. Sag. 1998. Argument Structure, Valence, and Binding. *Nordic Journal of Linguistics* 21: 107–144.

May, Robert. 1989. Interpreting logical form. *Linguistics and Philosophy* 12: 387–435.

Moltmann, Friederike. 1995. Exception sentences and polyadic quantification. *Linguistics and Philosophy* 18: 223–280.

Muller, Claude. 1991. *La Négation en Français.* Genève: Droz.

Pollard, Carl and Ivan A. Sag. 1994. *Head-driven Phrase Structure Grammar.* Chicago: University of Chicago Press and Stanford: CSLI Publications.

Pollard, Carl, and Eun Jung Yoo. 1998. A unified theory of scope for quantifiers and *wh*-phrases. *Journal of Linguistics.*

Przepiórkowski, Adam. 1999. *Case Assignment and the Complement-Adjunct Dichotomy: A Non-Configurational Constraint-Based Approach.* Doctoral dissertation, Universität Tübingen, Germany.

Richter, Frank, and Manfred Sailer. 1999. A lexicalist collocation analysis of sentential negation and negative concord in French. *Arbeitspapiere des SFB 340.* Universität Tübingen, Germany.

Przepiórkowski, Adam and Kupść, Anna. 1997. Negative concord in Polish. unpublished ms., Warsau.

Sag, Ivan A., and Thomas Wasow. 1999. *Syntactic Theory: A formal introduction.* Stanford: CSLI Publications. Distributed by Cambridge University Press.

de Swart, Henriëtte. 1998. La position de *ni* dans le système de la négation. In *Recherches de Linguistique Française et Romane, vol. 17*, ed. by Denis Delfitto, Jan Schroten, and Henriëtte de Swart. Utrecht: UiL-OTS/Romance languages. Pp. 67–80.

de Swart, Henriëtte. 1998. Licensing of negative polarity items under inverse scope. *Lingua* 105:175–200.

Szabolcsi, Anna. 1997. Strategies for scope taking. In *Ways of Scope Taking*, ed. by Anna Szabolcsi. Dordrecht: Kluwer Academic Press. Pp. 109–154.

Vallduví, Enric. 1994. Polarity items, n-words and minimizers in Catalan and Spanish. *Probus* 6: 263–294.

Westerståhl, Dag. 1989. Quantifiers in formal and natural languages. In *Handbook of Philosophical Logic, Vol. 4*, ed. by Dov Gabbay and Frantz Guenthner. Dordrecht: Reidel. Pp. 1–131.

Van der Wouden, Ton and Frans Zwarts. 1993. A semantic analysis of negative concord. *Proceedings of SALT 3.* Ithaca: Cornell University, Dept. of Linguistics.

van der Wouden, Ton. 1994. *Negative Contexts.* Doctoral dissertation, University of Groningen.

Warner, Anthony. 1993. *English Auxiliaries: Structure and History* Cambridge and New York: Cambridge University Press.

Warner, Anthony. 2000. English auxiliaries without lexical rules. In *Syntax and Semantics 32: The Nature and Function of Syntactic Categories*, ed. by Robert Borsley. San Diego and London: Academic Press. Pp. 167–220.

Zanuttini, Rafaella. 1991. *Syntactic Properties of Sentential Negation.* Doctoral dissertation, University of Pennsylvania.