Scope and Situation Binding in LTAG using Semantic Unification

Laura Kallmeyer (`lk@sfs.uni-tuebingen.de`)
SFB 441, University of Tübingen,
Nauklerstr: 35, D-72074 Tübingen, Germany.

Maribel Romero (`romero@ling.upenn.edu`)
Department of Linguistics, 610, Williams Hall, University of Pennsylvania,
Philadelphia, PA, 19104-6305, USA.

Abstract. This paper sets up a framework for LTAG (Lexicalized Tree Adjoining Grammar) semantics that brings together ideas from different recent approaches addressing some shortcomings of LTAG semantics based on the derivation tree. The approach assigns underspecified semantic representations and semantic feature structure descriptions to elementary trees. Semantic computation is guided by the derivation tree and consists of adding feature value equations to the descriptions. A rigorous formal definition of the framework is given. Then, within this framework, an analysis of scope is proposed that accounts for the different scopal properties of quantifiers (including nested quantificational NPs), adverbs, raising verbs and attitude verbs. Furthermore, integrating situation variables in the semantics, different situation binding possibilities are derived for different types of quantificational elements.

Keywords: Tree Adjoining Grammar, Computational Semantics, quantifier scope, underspecified semantics, situation binding, feature logic

1. Introduction

1.1. Lexicalized Tree Adjoining Grammars (LTAG)

LTAG (Joshi and Schabes, 1997) is a tree-rewriting formalism. An LTAG consists of a finite set of *elementary* trees associated with lexical items. From these trees, larger trees are derived by substitution (replacing a leaf with a new tree) and adjunction (replacing an internal node with a new tree). In case of an adjunction, the new elementary tree has a special leaf node, the *foot node* (marked with an asterisk). Such a tree is called an *auxiliary* tree. When adjoining such a tree to a node μ, in the resulting tree, the subtree with root μ from the old tree is put below the foot node of the auxiliary tree. Non-auxiliary elementary trees are called *initial* trees. Each derivation starts with an initial tree.
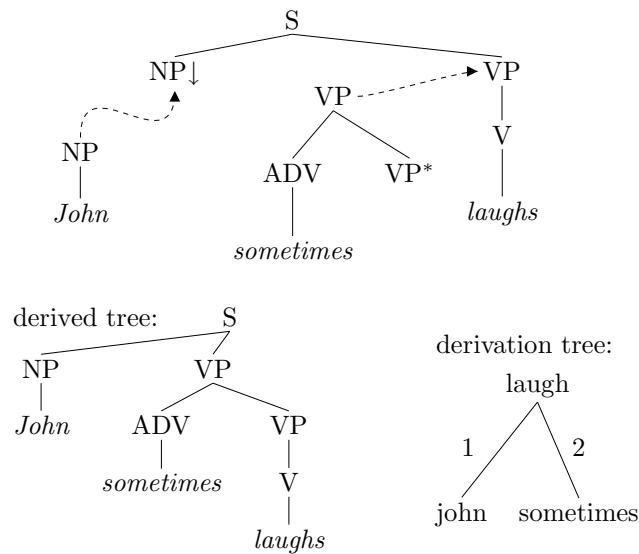
Figure 1. TAG derivation for (1)

The elementary trees of an LTAG represent extended projections of lexical items and encapsulate all syntactic/semantic arguments of the lexical anchor. They are minimal in the sense that only the arguments of the anchor are encapsulated, all recursion is factored away. These linguistic properties of elementary trees are formulated in the *Condition on Elementary Tree Minimality (CETM)* from Frank's (1992).

LTAG derivations are represented by derivation trees that record the history of how the elementary trees are put together. A derived tree is the result of carrying out the substitutions and adjoinings. Each edge in the derivation tree stands for an adjunction or a substitution. The edges are equipped with Gorn addresses of the nodes where the substitutions/adjunctions take place.[1] E.g., see the derivation of (1) in Fig. 1: Starting from the elementary tree of *laugh*, the tree for *John* is substituted for the node at position 1 and *sometimes* is adjoined at position 2.

(1) John sometimes laughs

The TAG formalism is in the class of so-called *mildly context-sensitive* grammar formalisms (Joshi, 1987). This means that it is more powerful than context-free grammars, while its expressive power is still sufficiently restricted to make it computationally tractable.

---

[1] The root has the address 0, the jth child of the root has address j and for all other nodes: the jth child of the node with address p has address $p \cdot j$.

TAGs are polynomially parsable and their formal properties have been investigated quite extensively in the literature (see among others Vijay-Shanker and Joshi, 1985, Vijay-Shanker, 1987, Weir, 1988). This is one of the reasons why TAG is a very attractive formalism for natural language processing.

1.2. LTAG SEMANTICS AND THE AIM OF THIS PAPER

The present paper is concerned with the semantics of LTAG. One question one has to ask in the beginning is: What are the semantic properties we require for TAG elementary trees? Concerning syntactic properties, the CETM mentioned in the previous section imposes certain minimality on elementary trees since they must encapsulate all and only the arguments of the lexical anchor, all recursion being factored away. As a consequence, whenever two items stand in a predicate-argument relation to each other, they cannot be part of the same elementary tree: the predicate contains a leaf for the attachment of the argument, but the argument itself is not part of the elementary tree of the predicate. Following the spirit of the CETM, in this paper we assume that whenever the semantic contributions of different lexical items can be clearly identified and separated from each other, i.e., whenever the semantic contribution of an expression can be decomposed into the semantic contributions of different lexical items, each of these lexical items has a separate elementary tree. We call this the *compositional minimality of elementary trees*.[2]

The aim of this paper is to develop an LTAG semantic framework that captures several important empirical properties of scope and of binding of situation variables in natural language. This general aim is structured in two goals, described below.

Because of the minimality of elementary trees and since derivation steps in TAG correspond to predicate-argument applications, it seems appropriate to base LTAG semantics on the derivation tree (Candito and Kahane, 1998, Joshi and Vijay-Shanker, 1999, Kallmeyer and Joshi, 2003). However, as we will see in section 2.1, it has been observed that in some cases this is problematic since the derivation tree does not provide enough information to correctly construct the desired semantic dependencies. The first goal of this paper is to bring together ideas from several recent approaches in order to develop a general framework for LTAG semantics that allows us to compute semantic representations on the derivation tree, overcoming

---

[2] For this paper we stay at the level of words, but actually a rigorous application of this condition of compositional minimality would even mean a further decomposition.

those problematic examples. The result will be a formally defined semantic framework for LTAG based on semantic feature structures and feature unification.

The second goal of this paper concerns quantificational elements. With regard to their scopal properties, one can roughly distinguish between two types of quantificational elements: on the one hand elements whose scope is determined by their surface position, and on the other hand elements that can scope higher than their surface position. The first class contains elements attached to the verbal spine (i.e., attached to some node on the path from the lexical anchor to the S node of a verb tree), such as adverbs, raising verbs, control verbs, attitude verbs, etc. To the second class belong quantificational N(oun) P(hrase)s. Further, even though the scope of quantificational NPs is not limited to their surface position, it must obey a locality condition: NP scope is limited to the minimal containing tensed clause when combining with a verb and limited to immediate scope over its host when nested in a quantificational NP. The question is how to derive these scopal properties of quantificational elements in a principled way.

In contrast to movement-based syntactic theories for long-distance dependencies (e.g., Chomsky, 1986, Chomsky, 1995 for *wh* dependencies, May, 1985 for scope), a fundamental property of TAG is that there is no syntactic movement and that everything is generated at its surface position. Furthermore, a 'moved' element (more precisely the slot, i.e., the substitution node for this element) is in the same elementary tree as the predicate it depends on. A *wh* long-distance dependency, for example, is obtained by assuming the slot for the *wh*-word to be part of the elementary tree of the verb it depends on, and then adjoining auxiliary trees for further embedding verbs between the *wh*-word and its verb. Locality constraints follow from the adjunction possibilities at the different nodes of the original verb tree (Kroch, 1987). Such an account of long dependencies without actual movement is possible in TAG because of the extended domain of locality of a TAG grammar, i.e., because the elementary trees of the grammar can describe arbitrarily large structures.

This approach to long dependencies without actual movement can also be exploited for quantifier scope: while the contribution to the predicate argument structure (e.g., the variable of a quantifier that is complement of a verb) stays lower, the part responsible for the boundness of scope is comparable to the *wh*-word in Kroch's (1987) and can attach higher. One possibility to obtain this separation of the contribution of an expression into two parts is to use multicomponent sets (Joshi, 1987, Weir, 1988) for quantifiers as proposed in Joshi and Vijay-Shanker's (1999) and Kallmeyer and Joshi's (2003). Another possibility

is to assume two different components in the semantic contribution of a quantifier while having only one elementary tree in the syntax. This is what we will do. Concerning the scope component of the semantics, its attachment, i.e., its scope, is limited by some upper scope boundary MAXS and this upper boundary is determined by the tree the quantifier attaches to. This is possible in TAG because of the extended domain of locality: No matter whether a quantifier is the subject or an object, the upper scope boundary is the same. Furthermore, it stays the same, even if other material is attached for example to the VP node. The upper scope boundary is always higher than the largest finite clause proposition containing the verb the quantifier depends on. In other words, similar to the *wh*-movement case in syntax, the proposition of the verb the quantifier depends on can be arbitrarily deeply embedded in the nuclear scope of the quantifier. I.e., the MAXS can be arbitrarily high even though it comes from the embedded verb.

Concerning the upper scope boundary MAXS, three different things can happen to this limit if a quantifier is an argument of some predicate P that is embedded under a higher predicate Q. First, the higher predicate can let the boundary pass, i.e., the MAXS of P and Q are the same and consequently quantifiers have the same scope possibilities no matter whether they are embedded only under the higher or under both predicates. The second possibility is that the higher predicate Q blocks scope, i.e., P is an island for scope. In this case, the MAXS boundaries determined by P and Q are different. In the last possibility, Q lets quantifiers from P pass but only one step further, in the sense that if a quantifier embedded under P takes scope over Q, then it takes immediate scope over Q. This last case occurs with nested quantifiers. We will give examples for the three cases in section 4.1. In order to derive underspecified representations for these three cases, we use constraints of the form $maxs \geq x$ where $maxs$ is the upper scope boundary and x is a variable for the nuclear scope of a quantifier. This differs from constraints $maxs \geq l$ where l is the label of the proposition of a quantifier that are usually used in underspecified semantics. This kind of constraints allows us to account not just for the two first cases above but also for the third, more complex case.

As part of this goal, the semantic proposal introduced for scope data will be extended to situation binding data. First, we will see that scope and situation binding are two different phenomena and do not always go together. Following Gallin (1975) and Cresswell (1990), the semantics of natural language requires direct quantification over world or situation variables (type s). Second, it will be shown that, as with scope, two types of elements need to be distinguished for situation binding: on the one hand elements whose situation must be locally

bound, and on the other hand elements whose situation does not need to be locally bound. To the first class belong elements attaching to the VP-spine. The second class corresponds to NPs. The split in two classes is the same for scope and situation binding. This will follow from the derivation tree and the general architecture of semantic features. But the constraints on NP scope and NP situation binding differ: MAXS limits NP scope, whereas situation binding is unlimited.
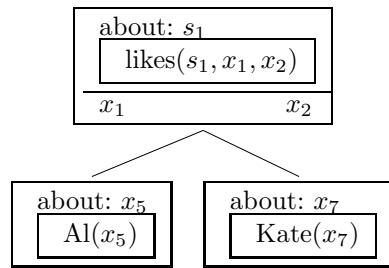
The structure of the paper is as follows. Section 2 presents previous approaches to LTAG semantics and the intuitive ideas of the LTAG semantics framework we are using. Section 3 then provides detailed formal definitions of the different components. Section 4 proposes an analysis of scope data accounting for the difference between quantificational NPs and quantificational material on the verbal spine. Section 5 integrates situations and situation binding into the framework. Finally, Section 6 compares our approach with some other approaches outside LTAG. Section 7 concludes.

2. A framework for LTAG Semantics

2.1. Previous approaches

Taking into account the semantic minimality of elementary trees and the fact that derivation steps in TAG correspond to predicate-argument applications, it seems appropriate to base LTAG semantics on the derivation tree. An early approach in this direction is the proposal to use synchronous TAG for the syntax-semantics interface (Shieber and Schabes, 1990, Shieber, 1994). The idea is to pair two TAGs, one for syntax and one for L(ogical) F(orm), and do derivations in parallel. In the latter formalization (Shieber, 1994), the two derivation trees are required to be isomorphic which amounts to doing semantics on the derivation tree without looking into the concrete rewriting process on the derived trees. For quantifiers, the approach uses multicomponent sets with a higher scope taking part and a lower part contributing the argument variable. For these multicomponent sets, one probably has to allow non-local derivations, otherwise the scope of quantifiers would be too restricted. In order to keep isomorphism of the derivation trees, Shieber (1994) needs to allow multiple adjunctions in the sense of Schabes and Shieber's (1994). The combination of Multi Component TAG (MCTAG) with (at least partly) non-local derivations and multiple adjunctions needs to be restricted
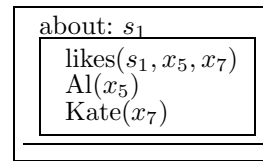
Derivation tree:

Resulting semantics:

$$\begin{array}{|l|}\hline \text{about: } s_1 \\ \hline \text{likes}(s_1, x_1, x_2) \\ \hline x_1 \qquad\qquad x_2 \\ \hline\end{array}$$

$$\begin{array}{|l|}\hline \text{about: } s_1 \\ \hline \text{likes}(s_1, x_5, x_7) \\ \text{Al}(x_5) \\ \text{Kate}(x_7) \\ \hline\end{array}$$

$$\begin{array}{|l|}\hline \text{about: } x_5 \\ \hline \text{Al}(x_5) \\ \hline\end{array} \qquad \begin{array}{|l|}\hline \text{about: } x_7 \\ \hline \text{Kate}(x_7) \\ \hline\end{array}$$

Figure 2. Derivation tree and resulting semantics for *Al likes Kate* à la Joshi and Vijay-Shanker (1999).

in some way, otherwise the formalism becomes too powerful.[3] Shieber (1994) does not investigate this since quantifiers are only mentioned in Shieber and Schabes' (1990). A difference with respect to the approach we will propose in this paper is that Shieber and Schabes generate LFs that are fully specified with respect to scope. However, the derivation tree can be considered as some kind of underspecified representation since the scope order of quantifiers attaching to the same S node is not fixed. But, crucially, the scope order between elements attaching to different nodes is fixed. In particular, a quantifier takes scope over adverbs attaching to the VP node. I.e., it is not clear how a synchronous TAG approach accounts for the *sometimes > every* scope order in (2):

(2) John sometimes kisses every girl

More recent approaches that more explicitly take the derivation tree to be the underlying structure for semantics are for example Candito and Kahane's (1998), Joshi and Vijay-Shanker's (1999) and Kallmeyer and Joshi's (2003). Consider the derivation tree and semantic computation of the sentence *Al likes Kate* in Fig. 2. The intuitive idea is that the semantic representation of each elementary tree 'is about' a variable. For example, the semantic representation of *likes* is about the (propositional or situation) variable s_1, and the representations of the NPs *Al* and *Kate* are about the individual variables x_5 and x_7 respectively. Furthermore, other variables in the semantic represention of a tree are linked to a particular leaf on that tree. For example, in the tree for *likes* x_1 is linked to the NP$_1$ position and x_2 is linked to the

---

[3] Non-local MCTAG are known to be NP-complete (Rambow and Satta, 1992). Furthermore, even in tree-local MCTAG, the possibility of multiple adjunctions increases the generative capacity, see Kallmeyer and Joshi's (2003) for an example.
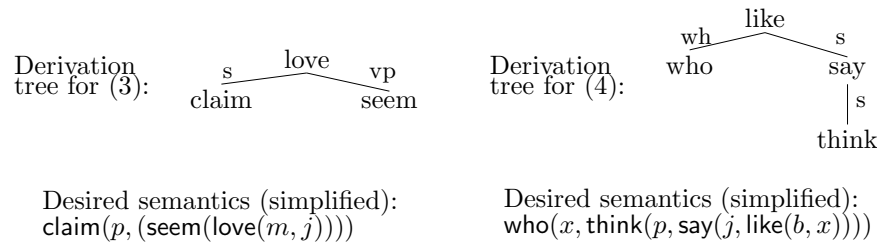
Derivation
tree for (3):

$\overset{\text{love}}{\underset{\text{claim}}{\text{s}}} \text{vp} \atop \text{seem}$

Derivation
tree for (4):

$\overset{\text{like}}{\underset{\text{who}}{\text{wh}}} \overset{\text{s}}{\text{say}} \atop \text{s} \atop \text{think}$

Desired semantics (simplified):
$\mathsf{claim}(p, (\mathsf{seem}(\mathsf{love}(m, j))))$

Desired semantics (simplified):
$\mathsf{who}(x, \mathsf{think}(p, \mathsf{say}(j, \mathsf{like}(b, x))))$

Figure 3. Problematic derivation trees for semantics

NP_2 position. The semantic composition is then performed following the derivation tree: for a tree γ_1 with leaf node p and a tree γ_2 attaching to γ_1 at p, the variable linked to p is identified with the 'about' variable of γ_2.

However, it has been observed that in some cases this simple semantic procedure is insufficient, since the derivation tree does not provide enough information to correctly construct the desired semantic dependencies (Rambow et al., 1995, Dras et al., 2004, Frank and van Genabith, 2001, Gardent and Kallmeyer, 2003). We will refer to this problem as "the missing link problem". The data that are, among others, claimed to be problematic for derivation tree based LTAG semantics are interactions of attitude verbs and raising verbs or adverbs, as in (3), and long-distance *wh*-movements, as in (4).[4]

(3) a. John Paul claims Mary seems to love
    b. Paul claims Mary apparently loves John

(4) Who does Paul think John said Bill liked?

In (3), *claim* and *seem* (or *apparently* resp.) adjoin to different nodes in the *love* tree, i.e., they are not linked in the derivation tree (see Fig. 3). But the propositional argument of *claim* is the *seems* (*apparently* resp.) proposition. I.e., the missing link one needs for semantics is a link between trees attaching to different nodes in the same tree. (4) is a different case: here, in the LTAG analysis, *who* is substituted into the *wh*-NP node of *like*, *say* is adjoined to the lower S node of *like* and *think* adjoins to *say*. Consequently, in the derivation tree, there is neither a link between *who* and *think* nor a link between *like* and *think*. But in the semantics, we want the *think* proposition to be the scopal argument of the *wh*-operator, i.e., a link between *who* and *think* must be established. This can be done via the semantics of *like* if we

---

[4] More recently (Forbes-Riley et al., 2005) it has been observed that similar problems occur when extending a derivation tree based LTAG semantics to discourse.

consider *like* to be the element that introduces the question operator. But at least some way to link *like* to *think* is needed. Here, the missing link is between trees γ_1 (here *like*) and γ_2 (here *think*) such that γ_2 adjoins to the root of a tree that (adjoins to the root of a tree that ...) attaches to some node μ in γ_1.

(3) is less hard than (4) since one can choose the semantics of *love* in such a way that the desired scope orders are obtained without direct link between the embedding attitude verb and the embedded raising verb (adverb resp.). A semantics in Kallmeyer and Joshi's (2003) framework is possible here. However, (4) still remains a serious problem.

Several proposals have been made to avoid the missing link problem that arises when doing semantics based on the derivation tree. We describe three of them in the following.

A proposal for computing semantics only on the derivation tree is to enrich the derivation tree with additional links as in Kallmeyer's (2002a, 2002b). In this approach, the derived tree need not be considered for computing semantics. The problem with this proposal is that sometimes it is not clear which link one has to follow in order to find the value for some semantic variable. Therefore additional rules for ordering the links for semantic computation are needed. The result is a rather complex machinery in order to obtain the dependencies needed for semantics.

Instead of using only the derivation tree for semantics, one could also use information from both, the derivation and the derived tree. Such an approach is pursued by Frank and van Genabith (2001).[5] This approach is actually quite close to what we will propose in this paper: elementary trees are associated with a meaning part and a glue part. The latter specifies how to combine the meaning parts. Furthermore, the single nodes in elementary trees are equipped with top and bottom features linking glue parts to these nodes. Substitutions and adjunctions yield equations between the features of the different nodes and thereby glue parts get identified. Finally, a deduction is performed using the resulting meaning parts and glue parts. An empirical shortcoming of this analysis is that it does not distinguish the scope possibilities of different modifiers attaching to the same clause. Otherwise, the overall approach resembles to ours except for the following aspects: Firstly, Frank and van Genabith do not procide a level of underspecified representations. Secondly, instead of using glue semantics, we prefer using established techniques from syntactic LTAG parsing for semantic computation.

---

[5] Fank and van Genabith actually claim that they compute LTAG semantics only on the derived tree. But this is not true since, similar to all other LTAG semantics approaches, they take elementary trees as a whole (i.e., nodes in the derivation tree) as the elements that are linked to semantic representations in the grammar.

Thereby we are able to guarantee that our formalism stays mildly context-sensitive.

More recently, Gardent and Kallmeyer (2003) propose to use the feature unification mechanism in the syntax, i.e., in the derived tree, in order to determine the values of semantic arguments. The underlying observation is that whenever a semantic link in the derivation tree is missing, it is either a) a link between trees attaching to different nodes in the same tree (see (3)), i.e., attaching to nodes that can share features inside an elementary tree, or b) a link between trees γ_1 and γ_2 such that γ_2 adjoins to the root of a tree that (adjoins to the root of a tree that ...) attaches to some node μ in γ_1 (see (4)). In this case, indirectly, the top of μ and the top of the root of γ_2 unify and thereby features can be shared. This approach works in the problematic cases and it has the advantage of using a well-defined operation, unification, for semantic computation. But it has the disadvantage of using the derived tree for semantics even though semantic representations are assigned to whole elementary trees (i.e., to nodes in the derivation tree) and not to nodes in the derived tree.[6]

Furthermore, the feature structures needed for semantics are slightly different from those used for syntax for the following two reasons:

– In feature-based TAG (FTAG, Vijay-Shanker and Joshi, 1988) one uses only a finite set of feature structure. This is crucial for showing that FTAG is equivalent to TAG.

  For semantics, at least theoretically, one needs an infinite number of feature structures since for example all individual variables from the terms used in the semantic representations (this is countably infinite) can occur as feature values.

– In FTAG, the syntactic feature structures are part of the syntactic representations and they are usually considered as partial feature structures, not as feature structure descriptions.

  In contrast to this, the features needed for semantics only serve to compute assignments for variables in the semantic representations. After semantic computation they are no longer relevant; they are not part of the semantic representations that finally get interpreted.

---

[6] A similar approach is Stone and Doran's (1997) where each elementary tree has a flat semantic representation, the semantic representations are conjoined when combining them and variable assignments are done by unification in the feature structures on the derived tree. But there is no underspecification, and the approach is less explicit than Gardent and Kallmeyer's (2003).

For these two reasons we think it more adequate to separate the semantic features from the syntactic ones and, furthermore, to use feature structure descriptions in the semantics. Semantic computation is then defined not as feature unification but as conjunction of feature structure descriptions and equations between feature values.

In this paper, we propose an approach that distinguishes between syntax with feature structures linked to nodes in the derived tree on the one hand and semantics with semantic representations and semantic feature structure descriptions linked to nodes in the derivation tree on the other hand. Formally, this means just extracting the semantic features used by Gardent and Kallmeyer (2003) from the derived trees and putting them in a semantic feature structure description linked to the semantic representation of the tree in question. Of course one still has to link semantic features to specific node positions in the elementary tree, e.g., in order to make sure that syntactic argument positions get correctly linked to the corresponding semantic arguments. Thus, our contribution to the solution of the missing link problem consists of placing those semantic features in semantic feature structure descriptions rather than in the derived tree and defining formally how semantic composition operates on these new feature structure descriptions. Furthermore, the choice to define separate semantic feature structure descriptions linked to elementary trees (not to single nodes in elementary trees) allows the introduction of global semantic features. These global features, similar to the "about" variables in Fig. 2, encode general semantic properties of an elementary tree.

2.2. LTAG SEMANTICS WITH SEMANTIC UNIFICATION

In our approach, each elementary tree in the TAG is linked to a pair consisting of a semantic representation and a semantic feature structure description. The latter are used to compute (via conjunction and additional equations) assignments for variables in the representations.

2.2.1. *Semantic representations and semantic feature structures*
As in Kallmeyer and Joshi's (2003), we use flat semantic representations in the style of MRS (Minimal Recursion Semantics, Copestake et al., 1999): semantic representations consist of a set of typed labelled formulas and a set of scope constraints. A scope constraint is an expression $x \geq y$ where x and y are propositional labels or propositional meta-variables (these last correspond to the holes in Kallmeyer and Joshi's (2003)).

The formulas in a semantic representation contain meta-variables –depicted as a boxed arabic numbers, e.g. $\boxed{1}$– of type e (individuals), s
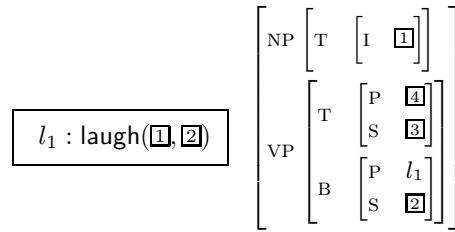
$$l_1 : \mathsf{laugh}(\boxed{1}, \boxed{2}) \qquad \begin{bmatrix} \text{NP} & \begin{bmatrix} \text{T} & \begin{bmatrix} \text{I} & \boxed{1} \end{bmatrix} \end{bmatrix} \\[2ex] \text{VP} & \begin{bmatrix} \text{T} & \begin{bmatrix} \text{P} & \boxed{4} \\ \text{S} & \boxed{3} \end{bmatrix} \\[2ex] \text{B} & \begin{bmatrix} \text{P} & l_1 \\ \text{S} & \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 4. Semantic representation and semantic feature structure of *laughs*

(situations) and $\langle s, t \rangle$ (propositions). Each semantic representation is linked to a semantic feature structure description. The meta-variables from the formulas can occur in these descriptions and to some of them values are assigned via feature equation. As an example see the semantic representation and the semantic feature structure of *laughs* in Fig. 4. The fact that the meta-variable of the first argument of laugh appears in the top (T) feature of the subject NP node position NP indicates for example that this argument will be obtained from the semantics of the tree substituted at the subject node. The second argument of laugh is a situation linked to the bottom (B) feature of the VP node, and the label of the laugh proposition, l_1, is linked to the bottom of the VP node as well. This signifies that the proposition l_1 is the minimal proposition corresponding to this node. If for example an adverb adjoins at the VP node, l_1 is embedded under that adverb and the value of the situation $\boxed{2}$ of l_1 is provided by that adverb.

Semantic feature structures are typed. The feature structure contains features 0 (the root position), 1, 2, ..., 11, 12, ... for all node positions that can occur in elementary trees (finite for each TAG).[7] The values of these features are structures containing two features T ('top') and B ('bottom'). Inside the T and B features, there are atomic features I, S and P whose values are individual variables, propositional labels and situation variables respectively.

2.2.2. *Semantic composition*

Semantic composition consists of conjoining feature structure descriptions while adding further feature value equations. It corresponds to the feature unifications in the syntax that are performed during substitutions and adjunctions and the final top-bottom unifications in the derived tree. In the derivation tree, elementary trees are replaced by their semantic representations plus the corresponding semantic feature

---

[7] For the sake of readability, we use names np, vp, ... for the node positions, somtimes with subscripts r for *root* and f for *foot*, instead of the Gorn adresses.

$l_1 : \mathsf{laugh}(\boxed{1}, \boxed{2})$

$$\left[\begin{array}{l} \text{NP} \left[\text{T} \left[\text{I} \quad \boxed{1}\right]\right] \\[2ex] \text{VP} \left[\begin{array}{ll} \text{T} & \left[\begin{array}{ll} \text{P} & \boxed{4} \\ \text{S} & \boxed{3} \end{array}\right] \\[3ex] \text{B} & \left[\begin{array}{ll} \text{P} & l_1 \\ \text{S} & \boxed{2} \end{array}\right] \end{array}\right] \end{array}\right]$$

np   vp

$l_3 : \mathsf{john}(x)$

$$\left[\text{NP} \left[\text{B} \left[\text{I} \quad x\right]\right]\right]$$

$l_2 : \mathsf{some}(s, s \text{ is part of } \boxed{5}, \boxed{6}), \boxed{6} \geq \boxed{7}$

$$\left[\begin{array}{ll} \text{VP}_r & \left[\text{B} \left[\begin{array}{ll} \text{P} & l_2 \\ \text{S} & \boxed{5} \end{array}\right]\right] \\[3ex] \text{VP}_f & \left[\text{T} \left[\begin{array}{ll} \text{P} & \boxed{7} \\ \text{S} & s \end{array}\right]\right] \end{array}\right]$$
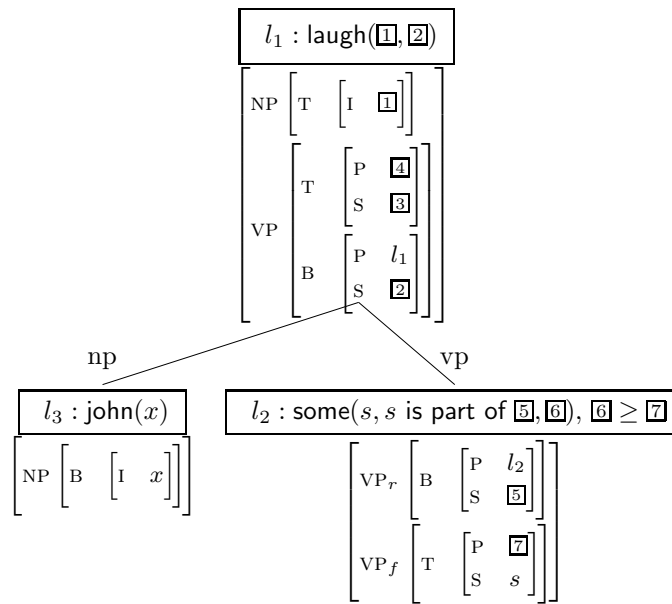
Figure 5. Semantic representations for (1) *John sometimes laughs*

structures. Then, for each edge in the derivation tree from γ_1 to γ_2 with position p:

- The top feature of position p in γ_1 and the top feature of the root position in γ_2, i.e., the features $\gamma_1.p.\text{T}$ and $\gamma_2.0.\text{T}$ are identified,
- and if γ_2 is an auxiliary tree, then the bottom feature of the foot node of γ_2 and the bottom feature of position p in γ_1, i.e., (if f is the position of the foot node in γ_2) the features $\gamma_1.p.\text{B}$ and $\gamma_2.f.\text{B}$ are identified.

Furthermore, for all γ in the derivation tree and for all positions p in γ such that there is no edge from γ to some other tree with position p: the T and B features of $\gamma.p$ are identified.

As an example consider the analysis of (1): Fig. 5 shows the derivation tree with the semantic representations and the semantic feature structure descriptions of the three elementary trees involved in the derivation. The formula $\mathsf{john}(x)$ is interpreted as meaning "there is a unique individual *John* and x is this individual". *Sometimes* is an existential quantification (some) over some situation s, where s is part of the situation $\boxed{5}$ of the *sometimes* proposition (here $\boxed{5}$ will default to the actual situation s_0).

The different feature value identifications lead to the identities marked in Fig. 6 with dotted lines. The top of the subject NP of *laughs* is identified with the top of the root NP of *John* (substitution) and
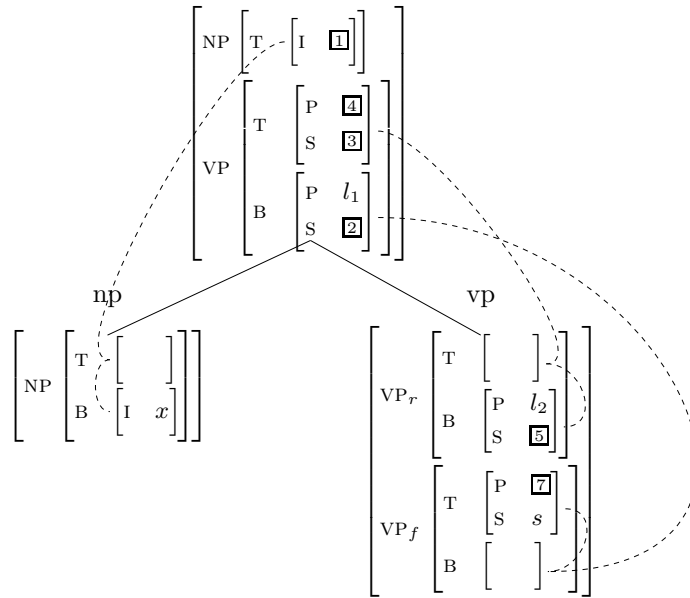
Figure 6. Semantic identifications for (1)

with the bottom of the root of *John* (final top-bottom unification). Consequently $\boxed{1} = x$. The bottom of the VP in *laughs* is identified with bottom and top of the foot VP_f of *sometimes* (adjunction and final top-bottom unification), yielding $\boxed{7} = l_1$ and $\boxed{2} = s$. Finally, the top of the VP in *laughs* is identified with the top and bottom of the root VP_r of *sometimes* (again, adjunction and final top-bottom unification), with the result $\boxed{4} = l_2$ and $\boxed{3} = \boxed{5}$.

The assignment obtained from the feature value equations is then applied to the semantic representation and the union of the representations is built. In our example this leads to (5):

$$(5) \quad \boxed{l_1 : \mathsf{laugh}(x,s), l_2 : \mathsf{some}(s, s \text{ is part of } \boxed{5}, \boxed{6}), l_3 : \mathsf{john}(x), \boxed{6} \geq l_1}$$

2.2.3. *Disambiguation*

The semantic representation obtained so far in this way is usually underspecified and cannot be interpreted yet. First, appropriate disambiguations must be found. These are assignments for the remaining meta-variables, i.e., functions that

- assign propositional labels to propositional meta-variables respecting the scope constraints, and

- assign situation variables to situation meta-variables so that s_0 (referring to the actual situation) is assigned to some meta-variable and every situation variable except s_0 is bound.

The disambiguated representation is then interpreted conjunctively.

(5) has only one disambiguation: Since $\boxed{6}$ cannot possibly equal l_2 ($\boxed{6}$ must be in the scope of l_2) and $\boxed{6}$ cannot possibly equal l_3 (otherwise there would be no meta-variable left below $\boxed{6}$ to be equated with l_1 in order to satisfy the constraint $\boxed{6} \geq l_1$), $\boxed{6} \to l_1$. Furthermore, $\boxed{5} \to s_0$, where s_0 is the actual situation, since $\boxed{5}$ is not in the scope of any situation binder. This leads to (6).[8]

(6) $\mathsf{john}(x) \wedge \mathsf{some}(s, s \text{ is part of } s_0, \mathsf{laugh}(x, s))$

3. Formal definition of the framework

In the following we give formal definitions of the framework we explained in an intuitive way in the preceding section.

3.1. SEMANTIC REPRESENTATIONS

The semantic representations we use resemble those defined in Kallmeyer and Joshi's (2003) (except that they do not contain an argument list): they consist of a set of labelled propositional formulas and a set of scope constraints.

The formulas in our semantic representations are typed. Types are defined in the usual recursive way, starting from basic types e, s and t for individuals, situations and truth values. For each type, there is not only a set of constants of this type and a set of variables but also a set of labels and, furthermore, a set of meta-variables. For any type T, C_T is the set of constants, V_T the set of variables, L_T the set of labels and M_T the set of meta-variables of type T. (In this paper we actually need only variables of types e and s, meta-variables of types e, s and $\langle s, t \rangle$, and labels of type $\langle s, t \rangle$.)

A scope constraint is an expression $x \geq y$ where x and y are propositional labels or propositional meta-variables.

Labels are useful in order to refer to formulas, for example in order to refer to a propositional formula inside a scope constraint. For the

---

[8] In previous work (Kallmeyer and Romero, 2004) we discussed some alternative ways of obtaining scope constraints (instead of putting them explicitly into the semantic representations). However, these alternatives all showed some disadvantages that made us prefer the architecture presented above.

examples we treat in this paper, we do not need labels of other than propositional type. Labels and also meta-variables are assumed to refer uniquely to one occurrence of a formula.

Propositional meta-variables can be considered as holes in the sense of Bos (1995), i.e., as variables where some propositional formula must be plugged in. Their values are specified either during semantic unification or, in case of underspecification, by a final disambiguation mapping (a plugging in the sense of Bos). The same will hold for situation meta-variables: they get their value during semantic unification or, if underspecified, from the final disambiguation mapping.[9]

DEFINITION 1 (Terms with labels and meta-variables).
1. *For each type T, each $c_T \in C_T$, each $v_T \in V_T$ and each $m_T \in M_T$ is an unlabelled term of type T.*
2. *For each type T, each unlabelled term τ of type T and each label $l \in L_T$, $(l : \tau)$ is a labelled term of type T.*
3. *For all types T_1, T_2 and each (possibly labelled) terms τ_1 of type $\langle T_1, T_2 \rangle$ and τ_2 of type T_1, $\tau_1(\tau_2)$ is an unlabelled term of type T_2.*
4. *For all types T_1, T_2, each term τ of type T_2 and each $x \in V_{T_1} \cup M_{T_1}$, $\lambda x.\tau$ is an unlabelled term of type $\langle T_1, T_2 \rangle$.[10]*
5. *Nothing else is a term.*

Brackets will be omitted in cases where the structure of a formula is still unambiguously given.

A semantic representation is a set of such terms together with a set of constraints on scope order, i.e. subordination constraints.

DEFINITION 2 (Semantic representation).
A semantic representation is a pair $\langle \mathcal{T}, \mathcal{C} \rangle$ such that:
- *$\mathcal{T}$ is a set of labelled terms.*
- *$\mathcal{C}$ is a set of constraints $x \geq y$ with $x, y \in L_{\langle s,t \rangle} \cup M_{\langle s,t \rangle}$.*

The typing of the meta-variables is needed in order to guarantee that only meta-variables standing for propositions occur in $\mathcal{C}$.

---

[9]   The same actually happens with individual meta-variables: they get their value during the semantic unification or in the final disambiguation mapping. An example of the latter is *his* in (7). *his* is underspecified and its value will be either the x of *every boy* or the y of *every man* or some individual z salient in the discourse. We will not consider examples like (7) in this paper.

(7) Every man$_1$ thinks every boy$_2$ likes his$_{1/2/3}$ cat.

[10] In this paper, we do not need lambda abstraction. But later it might be necessary, for questions for example.

The constraints in $\mathcal{C}$ restrict the possible scope orders. Besides these constraints, also the terms in $\mathcal{T}$ contain information about possible scope orders. A meta-variable or label that is in the scope of some x occurring in some term labelled l cannot have scope over l. And, x_1, x_2 that are in the scopes of y_1, y_2 respectively such that y_1 and y_2 are different arguments of a predicate, do not stand in any scope relation. This is to avoid for example that something ends up at the same time in the restriction and the nuclear scope of a quantifier. It actually means that the terms obtained after disambiguation are trees. Furthermore, scope order is transitive. The ordering relation on meta-variables and labels specified in such a way by $\mathcal{C}$ and $\mathcal{T}$, is called *subordination*. Its definition is more or less taken from Joshi et al. (2003).[11]

DEFINITION 3 (Subordination).

Let $\sigma = \langle \mathcal{T}, \mathcal{C} \rangle$ be a semantic representation with propositional meta-variables M_σ and propositional labels L_σ.

The subordination relation of σ, $\leq_\sigma \subseteq (M_\sigma \cup L_\sigma) \times (M_\sigma \cup L_\sigma)$ is defined as the smallest set $\leq_\sigma$ such that:

1. *for all $k \in M_\sigma \cup L_\sigma$: $k \leq_\sigma k$,*
2. *for all k, k' with $k' \geq k \in \mathcal{C}$: $k \leq_\sigma k'$,*
3. *for all $l \in L_\sigma$ and $k \in M_\sigma \cup L_\sigma$ such that there is a $l : \tau \in \mathcal{T}$, and k occurs in τ: it is the case that $k \leq_\sigma l$ and $l \not\leq_\sigma k$,*
4. *for all $k_1, k_2 \in M_\sigma \cup L_\sigma$ that are different arguments of the same predicate in some term $\tau \in \mathcal{T}$: there is no $k \in M_\sigma \cup L_\sigma$ such that $k \leq_\sigma k_1$ and $k \leq_\sigma k_2$, and*
5. *for all k, k', k'': if $k \leq_\sigma k'$ and $k' \leq_\sigma k''$, then $k \leq_\sigma k''$.*

If such a set does not exist, $\leq_\sigma$ is undefined.

---

[11] With this definition, $x \leq_\sigma y$ is intended to signify that x is in the scope of y. But it actually means that x is a subformula of y. So even cases such as the relation between a propositional argument and its matrix proposition are included in the $\leq_\sigma$ definition (e.g., from $l_1 : \mathsf{think}(x, l_2 : \mathsf{like}(y, z))$ would follow $l_2 \leq_\sigma l_1$), even though these cases are rather predicate-argument relations than scope relations. Some authors (e.g., Kahane, 2005) try to separate between predicate-argument relations and scope relations by saying that the first ones are those following from the syntax. However, we think this distinction not very clear and we prefer to define subordination as a subformula relation. We restrict it to propositional type expressions in this paper, but an extension to other types is of course possible if it is ever needed.

3.2. Semantic feature structure descriptions

Each semantic representation is linked to a semantic feature structure or rather a feature structure description. In TAG, the feature structures used for the syntax (Vijay-Shanker and Joshi, 1988) are usually considered being objects with a unification operation defined on them. However, as already mentioned above, the status of our semantic feature structures is different. We do not need the structures as part of the meaning in the sense of being part of the semantic representation that is interpreted with respect to a model and that describes the meaning of a sentence. We only need them to put semantic representations together; they are a kind of glue. Therefore we think it more appropriate to define the mechanism of combining the semantic feature structure descriptions in a purely syntactic way.

The difference between the usages of syntactic features and semantic features in LTAG is another reason to prefer separating the two from each other (in contrast to Gardent and Kallmeyer, 2003).

Our semantic feature structures (and also the corresponding terms) are typed. We will call the feature structure types *fs-types* to distinguish them from the types of the terms in the semantic representations. The fs-type of the whole feature structure is *sem*. The fs-types, their attributes and the fs-types of the values of these attributes are specified by a signature:

DEFINITION 4 (Signature of semantic feature structures).
A signature of a semantic feature structure is a tuple $\Sigma = \langle \mathcal{A}, \mathcal{T}_{fs}, A, t \rangle$ *such that*
- $\mathcal{A}$ *is a finite set of attributes (features),*
- $\mathcal{T}_{fs}$ *is a finite set of feature structure types (fs-types for short),*
- $A : \mathcal{T}_{fs} \to \mathcal{P}(\mathcal{A})$ *is a function specifying the set of attributes for each type,[12] and*
- $t : \mathcal{A} \to \mathcal{T}_{fs}$ *is a function specifying for each attribute the fs-type of its value.*

A fs-type T with $A(T) = \emptyset$ is called an atomic *fs-type.*

The fs-types we are using do not have a hierarchical structure. In other words, there are no sub-types as it is the case in many applications of typed feature structures (for example in HPSG, Pollard and Sag, 1994).

We will write signatures using an avm (attribute value matrix) notation. The signature Σ_{sem} of the semantic feature structures used here is

---

[12] For a set X, $\mathcal{P}(X)$ is the powerset of X, i.e., $\mathcal{P}(X) := \{Y \mid Y \subseteq X\}$.

$$sem \begin{bmatrix} \epsilon & tb \\ 1 & tb \\ \ldots & \end{bmatrix} \qquad tb \begin{bmatrix} \text{T} & bindings \\ \text{B} & bindings \end{bmatrix} \qquad bindings \begin{bmatrix} \text{I} & var_e \\ \text{P} & var_{\langle s,t \rangle} \\ \text{S} & var_s \end{bmatrix}$$

Figure 7. Semantic signature Σ_{sem}

shown in Fig. 7.[13] Here, the attributes of the feature structure of type *sem* are all node positions that can occur in elementary trees (finite for each TAG). The intuition behind the different fs-types in our signature Σ_{sem} is the following: a semantic feature structure links individuals, situations and propositions to syntactic positions, i.e., to nodes in the (syntactic) elementary tree. Each node has a top and a bottom feature structure. If no substitution or adjunction occurs at a node, top and bottom get identified. Otherwise, they can be separated.

The feature structure descriptions linked to the semantic representations are simple first order formulas with attributes and with constants for values of atomic type. Such first order formulas for attribute-value structures are introduced by Johnson (1988, 1990) except that, in contrast to Johnson's logic for feature structures, our logic is typed. Therefore we do not need a symbol $\bot$ for undefined values. We simply avoid computing such values by typing our feature terms and applying attributes only to terms of appropriate fs-types.

DEFINITION 5 (Semantic feature structure descriptions).

Let $\langle \mathcal{A}, \mathcal{T}_{fs}, A, t \rangle$ be a signature. Let $\mathcal{C}_T$ be a set of fs-constants for each atomic fs-type T. For non-atomic fs-types T, $\mathcal{C}_T := \emptyset$. Let $\mathcal{V}_T$ be a set of fs-variables for each fs-type T.

1. x is an fs-term *of fs-type T iff*

- *either $x \in \mathcal{V}_T \cup \mathcal{C}_T$, or*
- *there is a fs-term u of some fs-type T_1 and an attribute $a \in \mathcal{A}$ such that: $a \in A(T_1)$ (i.e., a is defined for fs-terms of type T_1), $t(a) = T$ (i.e., the fs-type of the value of a is T) and $x = a(u)$.*

---

[13] This avm notation means that the signature is $\langle \mathcal{A}, \mathcal{T}_{fs}, A, t \rangle$ with
- $\mathcal{A} := \{\epsilon, 1, \ldots, p, \text{T}, \text{B}, \text{I}, \text{P}, \text{S}\}$, where $\epsilon, 1, \ldots, p$ are the Gorn addresses of nodes in elementary trees of the grammar. This is a finite set for each TAG.
- $\mathcal{T}_{fs} := \{sem, tb, bindings, var_e, var_{\langle s,t \rangle}, var_s\}$,
- $A(sem) := \{0, 1, \ldots\}$, $A(tb) := \{\text{T}, \text{B}\}$, $A(bindings) := \{\text{I}, \text{P}, \text{S}\}$, $A(var_e) := A(var_{\langle s,t \rangle}) := A(var_s) := \emptyset$, and
- $t(0) := t(1) := \cdots := tb, t(\text{T}) := t(\text{B}) := bindings, t(\text{I}) := var_e, t(\text{P}) := var_{\langle s,t \rangle}, t(\text{S}) := var_s$.

Feature structure description:

$\mathrm{I}(\mathrm{T}(\mathrm{NP}(\boxed{0}))) = \boxed{1}\wedge$

$\mathrm{P}(\mathrm{T}(\mathrm{VP}(\boxed{0}))) = \boxed{4}\wedge$

$\mathrm{S}(\mathrm{T}(\mathrm{VP}(\boxed{0}))) = \boxed{3}\wedge$

$\mathrm{P}(\mathrm{B}(\mathrm{VP}(\boxed{0}))) = l_1\wedge$

$\mathrm{S}(\mathrm{B}(\mathrm{VP}(\boxed{0}))) = \boxed{2}$

Corresponding avm:

$$\boxed{0}\begin{bmatrix} \text{NP} & \begin{bmatrix} \text{T} & \begin{bmatrix} \text{I} & \boxed{1} \end{bmatrix} \end{bmatrix} \\ \text{VP} & \begin{bmatrix} \text{T} & \begin{bmatrix} \text{P} & \boxed{4} \\ \text{S} & \boxed{3} \end{bmatrix} \\ \text{B} & \begin{bmatrix} \text{P} & l_1 \\ \text{S} & \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 8. Feature structure description in avm notation

2. δ is a feature structure description *iff*

δ is a conjunction of $n \geq 1$ formulas of the form $y = z$ where y and z are fs-terms of the same fs-type.

We will use fs-variables $\boxed{0}, \boxed{1}, \ldots$. Feature structure descriptions will be notated in the usual avm notation. As an example see the feature structure description in Fig. 8. In this example, all conjuncts have a complex fs-term of the form $a(u)$ equated with a simple fs-variable or fs-constant $x \in \mathcal{V}_T \cup \mathcal{C}_T$.

In our case, the atomic constants of fs-type var_e are all individual variables from the logic used in our semantic representations, i.e., all $x \in V_e$. The atomic constants of fs-type $var_{\langle s,t\rangle}$ are all propositional labels, i.e., all $x \in L_{\langle s,t\rangle}$. And the atomic constants of fs-type var_s are all situation variables, i.e., all $x \in V_s$.

Because of the signature we are using, our feature structures are finite directed acyclic graphs. We assume satisfiability of the feature structure descriptions presented above to be defined in the usual model-theoretic way with the variables being interpreted as existentially bound (i.e., the description in Fig. 8 is actually to be read as $\exists \boxed{0}, \boxed{1}, \boxed{2}, \boxed{3}, \boxed{4}(\ldots)$). In contrast to most feature structure logics used in computational linguistics (see, e.g., Blackburn and Spaan, 1993, Johnson, 1994), the logic we are using is very restricted since we need neither negation nor disjunction nor universal quantification.[14]

Now we have to link the semantic feature descriptions to the semantic representations. This is done by using the meta-variables from the semantic representations as variables in the feature structure descriptions. More precisely, for a type T, $M_T = \mathcal{V}_{var_T}$. Furthermore, as

---

[14] If we limit the sets of fs-constants for atomic fs-types such that they become finite (this can be done for every practical application), and if the denotations of these constants are finite as well, our signature guarantees that the set of possible feature structures (i.e., the set of possible models) is finite.

mentioned above, propositional labels and the individual and situation variables from the semantic representations now become the atomic fs-constants in the feature structure descriptions. More precisely, $\mathcal{C}_{var_e} = V_e$, $\mathcal{C}_{var_{\langle s,t \rangle}} = L_{\langle s,t \rangle}$ and $\mathcal{C}_{var_s} = V_s$.

DEFINITION 6 (Elementary semantic entry).
   An elementary semantic entry *is a pair $\langle \sigma, \delta \rangle$ such that*

- *σ is a semantic representation with labels L_T, variables V_T and meta-variables M_T for all types T, and*
- *δ is a feature structure description with*

  - *signature Σ_{sem},*
  - *fs-constants $\mathcal{C}_{var_e} = V_e, \mathcal{C}_{var_{\langle s,t \rangle}} = L_{\langle s,t \rangle}, \mathcal{C}_{var_s} = V_s$*
  - *and fs-variables $\mathcal{V}_{var_T} = M_T$ for the types $T \in \{e, s, \langle s, t \rangle\}$.*

  such that δ contains exactly one fs-variable of fs-type sem called the top *of δ, and all other fs-variables in δ are of atomic fs-type.*

As an example consider the elementary semantic entry in Fig. 4, p. 12.[15]

3.3. SEMANTIC FEATURE IDENTIFICATION

Semantic composition consists only of feature unification or rather, since our objects are feature structure descriptions, of feature identification. These identifications correspond to the feature unifications in an FTAG in the syntax that are performed during substitutions and adjunctions and the final top-bottom unifications in the derived tree.

The syntax-semantics interface links each elementary tree to an elementary semantic entry. Semantic identifications are then done on the derivation tree. We assume that each time a new elementary semantic entry is chosen from the grammar, it contains fresh instances of labels, individual and situation variables and meta-variables. This way, the sets of labels and variables occurring in different nodes of the derivation tree are pairwise disjoint.

The derivation tree is a structure $\langle \mathcal{N}, \mathcal{E} \rangle$ that consists of a set of nodes $\mathcal{N}$ labelled with instances of elementary semantic entries and a set of directed edges $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$ labelled with Gorn addresses $p \in \mathbb{N}^*$. Each edge $\langle n_1, n_2 \rangle$ links mother node n_1 to daughter node n_2. Its label is a position p in the elementary tree of n_1.

---

[15] The top –e.g. ⓪ in Fig. 8– will often be left out in the avm notation, as in Fig. 4, since we know that a top is always present in feature structure descriptions being part of elementary semantic entries.

DEFINITION 7 (Semantic feature identification).

Let $D = \langle \mathcal{N}, \mathcal{E} \rangle$ be a derivation tree with L being the set of elementary semantic entries labelling the nodes in $\mathcal{N}$.

Then the result of the semantic feature identification over D is the following description δ_D: δ_D is a conjunction such that

- *each conjunct occurring in one of the feature structure descriptions in L occurs in δ_D*

- *for each edge $\langle n_1, n_2 \rangle \in \mathcal{E}$ with label p, with $\langle \sigma_1, \delta_1 \rangle$ label of n_1, $\langle \sigma_2, \delta_2 \rangle$ label of n_2, and t_1 and t_2 being the tops of δ_1 and δ_2 respectively:*

  $\mathrm{T}(p(t_1)) = \mathrm{T}(0(t_2))$ *is a conjunct in δ_D, and if the elementary tree corresponding to $\langle \sigma_2, \delta_2 \rangle$ is an auxiliary tree with p_f being the position of its foot node, then $\mathrm{B}(p(t_1)) = \mathrm{B}(p_f(t_2))$ is a conjunct in δ_D as well.*

- *For all $n \in \mathcal{N}$ with label $\langle \sigma, \delta \rangle$ and t top of δ such that γ is the corresponding elementary tree: for all node positions p in γ such that there is no edge $\langle n', n \rangle \in \mathcal{E}$ with label p:*

  $\mathrm{T}(p(t)) = \mathrm{B}(p(t))$ *is a conjunct in δ_D.*

- *These are all conjuncts in δ_D.*

As an example consider again the analysis of (1). Fig. 9 shows the derivation tree with the elementary semantic entries of the three elementary trees. The semantic feature identification have already been depicted in Fig. 6. The semantic feature identifications for the derivation tree in Fig. 9 lead to the feature value identities $\mathrm{T}(\mathrm{NP}(\boxed{0})) = \mathrm{T}(\mathrm{NP}(\boxed{5}))$ (substitution of *john* at position *np*), $\mathrm{T}(\mathrm{VP}(\boxed{0})) = \mathrm{T}(\mathrm{VP}_r(\boxed{6}))$ and $\mathrm{B}(\mathrm{VP}(\boxed{0})) = \mathrm{B}(\mathrm{VP}_f(\boxed{6}))$ (adjunction of *sometimes* at position *vp*) and $\mathrm{T}(\mathrm{NP}(\boxed{5})) = \mathrm{B}(\mathrm{NP}(\boxed{5}))$, $\mathrm{T}(\mathrm{VP}_r(\boxed{6})) = \mathrm{B}(\mathrm{VP}_r(\boxed{6}))$ and $\mathrm{T}(\mathrm{VP}_f(\boxed{6})) = \mathrm{B}(\mathrm{VP}_f(\boxed{6}))$ (final top-bottom unification). To see what this means, consider for example the substitution of *john* at position *np* in *laugh*: The identity $\mathrm{T}(\mathrm{NP}(\boxed{0})) = \mathrm{T}(\mathrm{NP}(\boxed{5}))$ means that the top feature of the position *np* in *laugh* is identified with the top feature of the root of *john*. This last feature is completely unspecified, therefore this identification does not give us anything new. Then, since no adjunction took place at the root of *john*, its top and bottom features also get identified ($\mathrm{T}(\mathrm{NP}(\boxed{5})) = \mathrm{B}(\mathrm{NP}(\boxed{5}))$). With these two identifications together we have an equation of the top feature of the position *np* in *laugh* and the bottom feature of the root of *john*. This means in particular identification of the two I attributes, and consequently leads to $\boxed{1} = x$.

$$l_1 : \mathsf{laugh}(\boxed{1}, \boxed{2})$$

$$\boxed{0}\begin{bmatrix} \mathrm{NP} & \begin{bmatrix} \mathrm{T} & \begin{bmatrix} \mathrm{I} & \boxed{1} \end{bmatrix} \end{bmatrix} \\[1em] \mathrm{VP} & \begin{bmatrix} \mathrm{T} & \begin{bmatrix} \mathrm{P} & \boxed{4} \\ \mathrm{S} & \boxed{3} \end{bmatrix} \\[1em] \mathrm{B} & \begin{bmatrix} \mathrm{P} & l_1 \\ \mathrm{S} & \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

np — vp

$$l_3 : \mathsf{john}(x)$$

$$\boxed{5}\begin{bmatrix} \mathrm{NP} & \begin{bmatrix} \mathrm{B} & \begin{bmatrix} \mathrm{I} & x \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$l_2 : \mathsf{some}(s, s \sqsubseteq \boxed{7}, \boxed{8}), \boxed{8} \geq \boxed{9}$$

$$\boxed{6}\begin{bmatrix} \mathrm{VP}_r & \begin{bmatrix} \mathrm{B} & \begin{bmatrix} \mathrm{P} & l_2 \\ \mathrm{S} & \boxed{7} \end{bmatrix} \end{bmatrix} \\[1em] \mathrm{VP}_f & \begin{bmatrix} \mathrm{T} & \begin{bmatrix} \mathrm{P} & \boxed{9} \\ \mathrm{S} & s \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 9. Semantic representations for (1) *John sometimes laughs*

Equality between fs-terms is reflexive, symmetric and transitive, and it extends to the different attributes allowed for the type of the term. This permits us to derive further conjuncts using the rules in (8).

(8) a. $\dfrac{}{x = x}$ *refl*   where x is an fs-term

    b. $\dfrac{x = y}{y = x}$ *symm*   where x and y are fs-terms

    c. $\dfrac{x = y \wedge y = z}{x = z}$ *trans*   where x, y and z are fs-terms

    d. $\dfrac{x = y}{a(x) = a(y)}$ *attr*   where x, y, $a(x)$ and $a(y)$ are fs-terms

If the resulting description δ is satisfiable, we can continue computing an assignment function from δ. In order to check for satisfiability, we have to check whether for all fs-constants c_1, c_2 with $\delta \vdash c_1 = c_2$, c_1 is indeed equal to c_2. Then, from δ an assignment function g can

be obtained for some of the meta-variables occurring in the semantic representations. We call the assignment a *meta-assignment function*.

We assume the meta-variables to be alphabetically ordered.

DEFINITION 8.

Let D be a derivation tree, δ_D the result of the semantic feature identification over D.

If δ_D is satisfiable, then the meta-assignment function *of δ_D, g_D is defined as a function from the set of all fs-variables to the set of all fs-constants and fs-variables such that:*

- *for all fs-variables $\boxed{m}$ such that there is a fs-constant c with $\delta_D \vdash \boxed{m} = c$: $g(\boxed{m}) = c$,*

- *for all fs-variables $\boxed{n_1}$ such that there is no fs-constant c with $\delta_D \vdash \boxed{n_1} = c$: if $\boxed{n_2}$ is the alphabetically first fs-variable such that $\delta_D \vdash \boxed{n_1} = \boxed{n_2}$, then $g(\boxed{n_1}) = \boxed{n_2}$.*

Let us go back to the example in Fig. 9. The feature identifications lead to the assignment g with $g(\boxed{1}) = x$, $g(\boxed{4}) = l_2$, $g(\boxed{3}) = g(\boxed{7}) = \boxed{3}$, $g(\boxed{9}) = l_1$ and $g(\boxed{2}) = s$.

In order to obtain the semantic representation for a derivation tree, the union of the semantic representations in the derivation tree is built, and then the assignment function obtained from the feature structure descriptions in the derivation tree is applied to it. More precisely, for all fs-variables $\boxed{m}$ and all $x \in V_e \cup V_s$ such that $g(\boxed{m}) = x$: all occurrences of $\boxed{m}$ are replaced with x. And for all fs-variables $\boxed{m}$ and all $l \in L_{\langle s,t \rangle}$ that label a term τ such that $g(\boxed{m}) = l$: all occurrences of $\boxed{m}$ in constraints in $\mathcal{C}$ are replaced with l; furthermore, if there are also occurrences of $\boxed{m}$ in $\mathcal{T}$, then these are replaced with $l : \tau$ and the original $l : \tau$ is deleted from the set.

In our example, this results in the semantic representation (9):

$$(9) \quad \boxed{l_1 : \mathsf{laugh}(x, s), l_2 : \mathsf{some}(s, s \text{ is part of } \boxed{3}, \boxed{8}), l_3 : \mathsf{john}(x), \boxed{8} \geq l_1}$$

3.4. Disambiguation

The semantic representation obtained from a derivation tree is usually underspecified and cannot be interpreted yet; first appropriate disambiguations must be found. These are assignments for the remaining variables, i.e., functions that assign propositional labels to propositional meta-variables, respecting the scope constraints, and that assign situations to situation meta-variables. A function f_p respects the scope

constraints if, after having applied it to a semantic representation σ, the subordination relation $\leq_{fp(\sigma)}$ of the result $f_p(\sigma)$ is a partial order (see Def. 3, p. 17 for the definition of subordination).[16]

DEFINITION 9 (Disambiguation function).

Let σ be a semantic representation with propositional meta-variables $M_{\langle s,t \rangle}(\sigma)$ and situation meta-variables $M_s(\sigma)$, and with propositional labels $L_{\langle s,t \rangle}(\sigma)$ and situation variables $V_s(\sigma)$. Let $s_0 \in V_s$ be a special situation, the actual situation. A pair $\delta = \langle f_p, f_s \rangle$ is a disambiguation for σ in s_0 iff

- *$f_p : M_{\langle s,t \rangle}(\sigma) \to L_{\langle s,t \rangle}(\sigma)$ is a total function such that $\leq_{f_p(\sigma)}$ is defined and $\langle L_{\langle s,t \rangle}(\sigma), \leq_{f_p(\sigma)} \rangle$ is a partially ordered set (poset).*
- *$f_s : M_s(\sigma) \to V_s(\sigma)$ is a total function such that*

  - *there is at least one $x \in M_s(\sigma)$ with $f_s(x) = s_0$, and*
  - *after application of f_p and f_s, all $s \in V_s(\sigma) \setminus \{s_0\}$ in the terms of σ must be bound by a quantifier.*

Note that the definition of f_p means in particular that each label occurs at most once in a term. Later, in the definition of *normal* semantic representations (Def. 10, p. 28) we even require for the semantic representations we are using that each label occurs at most once in $\mathcal{T}$.

A disambiguated representation is then interpreted conjunctively.

3.5. GLOBAL FEATURES

So far, we followed Gardent and Kallmeyer (2003) and assumed that semantic features are linked to specific node positions in the elementary trees. In some cases, these links are necessary. The features P and S for example are linked to specific nodes since they vary for the different VP nodes and the S node.

However, some semantic features are rather linked to the elementary tree as a whole and not to single node positions. We will call these semantic features "global features". An example is the feature I in the NP tree. The intuition is that the whole NP tree introduces a unique individual x. Another example for a global feature is the minimal proposition contributed by a verb. In Fig. 9, the minimal proposition of *laugh* is l_1, no matter which modifiers adjoin at the VP and S nodes. This minimal proposition is important because it determines the minimal

---

[16] $f_p(\sigma)$ is the semantic representation one obtains from σ as follows: for all propositional meta-variables $\boxed{m}$ with $f_p(\boxed{m}) = l$ where $l : \tau$ is the unique term with label l a) replace all occurrences of $\boxed{m}$ with $l : \tau$, and b) delete the original occurrence of $l : \tau$.

$$sem \begin{bmatrix} \textsc{global} & global \\ 0 & node \\ 1 & node \\ \dots & \end{bmatrix} \qquad node \begin{bmatrix} \textsc{global} & global \\ \textsc{t} & bindings \\ \textsc{b} & bindings \end{bmatrix}$$

$$bindings \begin{bmatrix} \textsc{p} & var_{\langle s,t \rangle} \\ \textsc{s} & var_s \end{bmatrix} \qquad global \begin{bmatrix} \textsc{i} & var_e \\ \textsc{maxs} & var_{\langle s,t \rangle} \\ \textsc{mins} & var_{\langle s,t \rangle} \end{bmatrix}$$

Figure 10. New semantic signature Σ_{sem} including global features

nuclear scope of any quantifier attached to *laugh*: In (10), the subject quantifier can scope under or over the adverb. But it must take at least the minimal proposition of *laugh* in its scope.

(10) Every girl sometimes laughs

For this minimal proposition, we assume a global feature MINS.

In cases where a global property is attributed to an elementary tree as a whole, one could of course simply link the global feature to all nodes in the elementary tree. But this does not reflect their global character. Therefore we propose that semantic feature structures contain additional global features besides the different features p, where p is a node position.

We assume that global features are features

- that intuitively are not linked to a single node of an elementary tree but to the tree as a whole, and
- whose values are unique for the whole semantic feature structure (i.e., unique for the whole elementary tree).

Global features are analogous to the "about" variable proposed in Joshi and Vijay-Shanker's (1999).

The global features of an elementary tree are listed in a feature GLOBAL of type *global*. Besides this, we allow elementary trees, i.e., nodes in the derivation tree to look into the global features of the elementary trees they are immediately linked to. In other words, they can access the global features of their mother and their daughters in the derivation tree.

In order to allow for this, the feature structure of a specific node n in an elementary tree can also contain a feature GLOBAL. Our signature is therefore modified as shown in Fig. 10. (The global feature MAXS will be explained later.)

Depending on the nature of the node n in an elementary tree that has a feature GLOBAL, this feature is identified with the GLOBAL of

the mother or of a daughter in the derivation tree. If n is an argument slot (a substitution node/a foot node), its feature GLOBAL is identified with the feature GLOBAL of the argument tree (the substituted initial tree/the tree the auxiliary tree adjoins to).[17] If n is the root node of an initial tree, its feature GLOBAL is identified with the feature GLOBAL of the tree this initial tree attaches to. In all other cases n's feature GLOBAL is identified with the GLOBAL of any auxiliary tree adjoining at n:

For a derivation tree D, for each edge $\langle n_1, n_2 \rangle \in \mathcal{E}$ labelled with position p where $\langle \sigma_1, \delta_1 \rangle$ and $\langle \sigma_2, \delta_2 \rangle$ are the labels of n_1 and n_2 respectively, t_1 and t_2 the top variables of δ_1 and δ_2 and γ_1 and γ_2 the corresponding trees:

- if γ_2 is an initial tree, then the conjuncts $\mathrm{GLOBAL}(t_2) = \mathrm{GLOBAL}(p(t_1))$ and $\mathrm{GLOBAL}(t_1) = \mathrm{GLOBAL}(0(t_2))$ are added to the result of the semantic feature identification over D.
- if γ_2 is an auxiliary tree with foot node position f then the conjuncts $\mathrm{GLOBAL}(t_1) = \mathrm{GLOBAL}(f(t_2))$ and $\mathrm{GLOBAL}(t_2) = \mathrm{GLOBAL}(p(t_1))$ are added to the result of the semantic feature identification over D.

One can express a lot of things with global features. However, we still need the features linked to specific nodes in elementary trees and, furthermore, we need the top bottom feature identifications parallel to those in the syntax. Otherwise we would again have the missing link problem. Since the problems identified so far as missing link problems all concern scope or propositional arguments, we probably only need non-global features P and S.

As already explained, the approach proposed in Gardent and Kallmeyer's (2003) can be directly transformed into the framework presented here. The other direction is also true since global features can be reformulated as features on all nodes of an elementary tree. (The identification of global features between elementary trees can be done using appropriate features in the top features of substitution nodes and of roots of initial trees and also in the bottom features of internal nodes and of foot nodes.)

An advantage of our approach is however that semantic feature structures are linked to whole elementary trees and therefore they offer the possibility to define global features for elementary trees. We think this important because it allows to capture the intuition from Joshi and Vijay-Shanker (1999) that the semantic representation of an elementary tree as a whole is 'about' something.[18]

---

[17] This is parallel to the argument variables in Joshi and Vijay-Shanker's (1999) and Kallmeyer and Joshi's (2003).

[18] A further difference is that we do not use explicit holes $h_1, h_2, \ldots$ besides propositional variables. Instead, the propositional variables that remain after having

3.6. COMPUTATIONAL ASPECTS

The computation of the underspecified semantic representation via feature identification on the derivation tree uses the same mechanisms as ordinary LTAG parsing in a feature-structure based TAG (FTAG, Vijay-Shanker and Joshi, 1988).[19] The only difference is that the set of possible feature values is not finite in general (e.g., possible values for features of propositional type are $l_1, l_2, l_3, \ldots$). However, in practical applications, the feature value set always can be limited.[20] Then the complexity of syntactic FTAG parsing (which is $O(n^6)$) and FTAG parsing including semantics is the same except for some constant factors.[21]

The second aspect to consider is the complexity of the disambiguation, i.e., the computation of the different readings yielded by an underspecified representation. In general, the complexity of disambiguating expressions with scope constraints of the form $x \geq y$ is NP-complete (Koller et al., 1998). But the semantic representations we actually use are close to so-called *normal dominance constraints* (Koller et al., 2003, Fuchss et al., 2004). For this type of constraints an efficient polynomial solver has been developed. We define *normal semantic representations* (that correspond directly to normal dominance constraints):

DEFINITION 10.
    A semantic representation $\langle \mathcal{T}, \mathcal{C} \rangle$ is normal *iff*

1. *no $l \in L_{\langle s,t \rangle}$ is in argument position in $\mathcal{T}$ and no $x \in M_{\langle s,t \rangle}$ is in labelling position in $\mathcal{T}$.*
2. *every $x \in L_{\langle s,t \rangle} \cup M_{\langle s,t \rangle}$ occurring in $\mathcal{T} \cup \mathcal{C}$ occurs exactly once in $\mathcal{T}$.*
3. *for each constraint $x \geq y$ in $\mathcal{C}$: $x \in M_{\langle s,t \rangle}$ and $y \in L_{\langle s,t \rangle}$.*

---

performed all feature identifications are understood as being holes in the sense of previous LTAG semantics approaches. This simplifies the formal framework considerably.

[19] The global features could be expressed within the top and bottom feature structures of the nodes, so they do not add anything to the computational complexity either.

[20] We could for example restrict ourselves to the first 100 propositional labels, the first 100 inidividual variables and the first 100 situation variables. If we choose a limit that is sufficiently high, we can still parse any sentence that actually occurs as input to our system.

[21] The addition of global features does not increase the complexity either since, similar to the unifications linked to substitution and adjunction, unification is only performed between the feature structures of mothers and daughters in the derivation tree.

The restriction that each label occurs at most once in $\mathcal{T}$ is motivated by the intuition that labels point at (occurrences of) formulas and this should be unique.

The semantic representations we are using are not normal since they do not satisfy the third condition. We need scope constraints of the forms $l \geq x$ and $y \geq x$ with $l \in L_{\langle s,t \rangle}$ and $x, y \in M_{\langle s,t \rangle}$. So far, such constraints are not allowed in the normal dominance constraint solver. But, (Alexander Koller, personal communication) if certain restrictions are respected, they probably can be integrated into the solver while keeping the solving polynomial.[22]

The situation disambiguation is less problematic than the scope constraint solving. Once the scope is fixed the situation disambiguation f_s must be chosen in such a way that all situation variables except s_0 are bound. This can be done in a single top-down traversal of the term trees.

4. Scope in LTAG

4.1. THE DATA

Quantificational NPs differ from quantificational elements attached to the verbal spine (adverbs, raising verbs, attitudes verbs, etc.) in their scope possibilities. E.g., (14), with two quantificational NPs, is scopally ambiguous between the surface reading (14a) and the inverse reading (14b) (May 1985, among many others). (15) and (16), in contrast, have only the surface reading (a) and lack the reading (b) (Cinque 1999, among others). Given the flexible scope of NPs and the rigid scope of (ad)verbal attachments, the ambiguity resulting from combining an NP with an (ad)verbal element as in (17) is entirely attributed to the NP.[23]

---

[22] These restrictions are such that

1. For each $y \in M_{\langle s,t \rangle}$ there is at most one $l \in L_{\langle s,t \rangle}$ with a constraint of the form $l \geq y$, and

2. when considering the terms in the semantic representation as trees and the scope constraints as dominance edges, then the following holds: build a graph by taking the dominance edges of the form $x \leq l$ and reversing all edges in the term trees. Then, in the resulting graph, there must be no cycles (prohibits for example $l_1 : \ldots \boxed{1} \ldots, l_2 : \ldots \boxed{2} \ldots$ and $l_1 \geq \boxed{2}, l_2 \geq \boxed{1}$).

[23] We will assume fix surface-syntax scope for VP-spine material regardless of what node(s) on the spine are targeted and what type of material is attached. Some alleged counterexamples in the literature are (11) and (12) (Bouma et al., 1998, Kallmeyer and Joshi, 2003). Note, however, that the scopal ambiguity in (11) may

(14) Exactly one student admires every professor
  a. $\exists!x[\mathsf{st}(x) \land \forall y[\mathsf{prof}(y) \rightarrow \mathsf{adm}(x,y)]]$
  b. $\forall y[\mathsf{prof}(y) \rightarrow \exists!x[\mathsf{st}(x) \land \mathsf{adm}(x,y)]]$

(15) John seems to sometimes laugh.
  a. $\mathsf{seem}(\mathsf{sometimes}(\mathsf{laugh}(j)))$
  b. * $\mathsf{sometimes}(\mathsf{seem}(j, \mathsf{laugh}(j)))$

(16) John wants Mary to sometimes laugh.
  a. $\mathsf{want}(j, \mathsf{sometimes}(\mathsf{laugh}(m)))$
  b. * $\mathsf{sometimes}(\mathsf{want}(j, \mathsf{laugh}(m)))$

(17) John seems to have visited everybody
  seem $> \forall$, $\forall >$ seem

Hence, the scope of a quantificational element attached to the verbal spine is fully determined by the surface syntax whereas the scope of a quantificational NP is not limited in this way. An NP argument of a verb can scope anywhere within the clause headed by that verb. Nevertheless, the scope of an NP is not completely unbounded beyond that minimal clause. The interesting question is what grammatical environments allow for NPs to scope beyond the minimal clause; that is, what limits the otherwise free scope of an NP, and in what way it limits it. In descriptive terms, three things may happen to the upper

---

be attributed to the different attachment possibilities in surface syntax: *rarely* may c-command the *because* clause in the derived tree, or vice-versa. As for (12), the speakers consulted can only get the inverse reading *allegedly > usually* if *allegedly* is read between commas, i.e. as a parenthetical. Parentheticals and other expressive items like *obviously* are special in that they cannot be semantically embedded under other operators (Kratzer, 1999, Potts, 2002, von Fintel and Iatridou, 2003), witness the contrast in (13a,b). In (13a), the cause of John's being upset may be just the obviousness of Mary's indifference, making the sentence compatible with the described scenario. In (13b), the reason has to be Mary's indifference per se, rendering the sentence false in the same scenario. That is, the semantic contribution of *obviously* is not embedded under *because* but is simply a separate comment of the speaker. This special semantic behavior of parentheticals and expressive items seems to be the source of the putative inverse reading in (12). We leave the analysis of these items outside the scope of this paper.

(11) Sandy rarely visited a friend because of El Niño.

(12) Usually, Pat allegedly drives a cadillac.

(13) Scenario: John and Mary made the deal that they would pretend to be in love with each other. In reality, they do not care for each other's love.
  a. John is upset because it is obvious that Mary doesn't love him.
  b. John is upset because Mary(,) obviously(,) doesn't love him.

boundary of the scoping possibilities of an NP –i.e., to MAXS in our analysis– when that NP is the argument of some predicate P embedded under some higher predicate Q: (i) Q blocks NP-scope; (ii) Q lets the NP-scope pass imposing no limitations; and (iii) Q lets NP-scope pass imposing some limitations. We present the empirical pattern for each in turn.[24]

Operators blocking NP-scope in case (i) are attitude verbs like *think, say, wish*, etc. Their crucial property is that they select for a tensed clause as their complement. The empirical generalization standardly assumed is this: the scope of an NP functioning as argument of an embedded verb is limited to the first finite clause containing the NP. For example, inverse scope is not possible across a finite clause in (18), as the reading *every > a* is not allowed.[25]

(18) A student said you met every professor
   a > every,   * every > a

In contrast, inverse scope is available across a non-finite clause in (19), making the reading *every > a* available. Verbs selecting for a non-finite complement clause –control verbs like *want, try* or *tell* and ECM verbs– let NP-scope rise above them freely, thus constituting case (ii).

(19) A student wants to meet every professor
   a > every,   every > a

Finally, in case (iii), when an NP_2 is nested inside an NP_1, the embedded NP_2 is allowed to scope above NP_1, but only immediately above NP_1. Consider the NP *every city* nested within the NP headed by *someone* in (21). Of the five in principle possible readings in (22), the reading (e) where the quantifier *two* scopally intervenes between *every* and *some* is excluded (Larson, 1987) (Sauerland, 2000) (**?**) (**?**).[26]

---

[24] These three possibilities are somewhat reminiscent of the three types of higher operators Q determining presupposition projection (Karttunen, 1973): plugs (presuppositions do not project beyond the argument of Q), holes (presuppositions project beyond the argument of Q), and filters (presupposition project beyond the argument of Q where they originate but only a bit further, to another argument of Q).

[25] Some alleged counterexamples to finite clause boundness are analyzed nowadays as cases of illusive scope (Fox and Sauerland, 1996, Artstein, 2003).

[26] Readings (b') and (b") below –which are like (22b) but with ∀ and 2 scoping over each other– are ruled out on independent grounds because of the logical architecture of the sentence (Hobbs and Shieber, 1987). Syntactically, *every* is generated as part of the N' of *some*, whereas *two* is outside it. This means that, when the two quantifiers remain under the scope of *some*, *every* must lie within the restrictor of *some* and *two* within the nuclear scope. That is, when *some* scopes over both

(21) Two policemen spy on someone from every city.

(22)
a.	$2 > \exists > \forall$	b.	$\exists > \forall,2$
c.	$2 > \forall > \exists$	d.	$\forall > \exists > 2$

e. $* \forall > 2 > \exists$

Note that the nested NP *every city* is allowed to scope over its host *somebody*, as in readings (c) and (d). But, when it does scope over its host, the quantifier *two* cannot scopally intervene between the two quantifiers, as the unavailability of reading (e) shows. The question is how to rule out reading (e).

In sum, the empirical generalizations to account for are the following:

- For quantificational elements on the VP spine, scope is fully determined by the surface syntax: they take scope exactly where they appear.

- For quantificational NPs filling substitution slots, scope is not fully determined by the surface syntax: their scope is underspecified, but limited to the minimal containing tensed clause when combining with a verb and limited to immediate scope over its host when nested in a quantificational NP.

In the following subsections, we are concerned with scope and we will not mention situations. Situations will be reintroduced in section 5.

4.2. ADVERBS AND RAISING VERBS

We have already seen the analysis of adverbs (see *sometimes* in Fig. 5). Raising verbs are analyzed similarly. Both, raising verbs and adverbs, are in a sense inserted between the top and bottom P values of the node to which they adjoin. They scope over the lower proposition. A second adverb/raising verb adjoined to the root of a first one has therefore scope over the first one. By unification, the proposition introduced by the topmost adverb/raising verb is the P value of the root of the verb tree.

As examples for the interaction of two modifiers that are raising verbs or adverbs consider (23) and (24). In (23) the two modifiers adjoin both at the VP node while in (24) one of them adjoins at the S node, the other one at the VP node. This last example is an instance of the

---

quantifiers, the two quantifiers are part of two different arguments of *some* and thus have to remain unordered with respect to each other, as in (22b).

(20) b'. $* \exists > \forall > 2$
    b''. $* \exists > 2 > \forall$
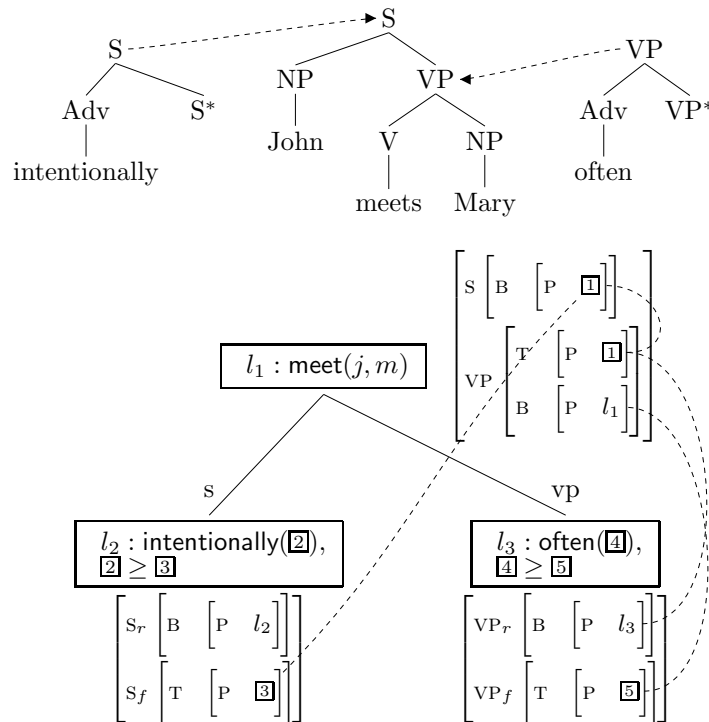
Figure 11. Modifiers at S and VP: missing link problem

missing link problem: *intentionally* takes *often* into its scope but it is linked only to *meets* in the derivation tree.

(23) John seems to sometimes laugh

(24) Intentionally, John often meets Mary

Fig. 11 shows the analysis of (24). When adjoining the VP modifier (together with the final top-bottom unification), we obtain $\boxed{5} = l_1$ and $\boxed{1} = l_3$, and the upper proposition l_3 is then passed up to the S node. When adjoining the S modifier, we obtain $\boxed{3} = l_3$. The topmost propositional label in the end is l_2. The result is the semantic representation (25):

(25)
$$
\begin{array}{l}
l_1 : \mathsf{meet}(j,m),\ l_2 : \mathsf{intentionally}(\boxed{2}),\ l_3 : \mathsf{often}(\boxed{4}) \\
\boxed{2} \geq l_3,\ \boxed{4} \geq l_1
\end{array}
$$

The articulation of the VP-spine with propositional labels allows S modifiers to have scope over VP modifiers.[27]

---

[27] The idea of articulating the VP spine to solve the missing link problem was first proposed by Frank and van Genabith (2001) to distinguish between verbs attaching

4.3. Quantificational NPs

The semantic contribution of a quantificational determiner like *every* is a formula of shape $l : \mathsf{every}(x, \boxed{4}, \boxed{5})$ together with some constraints on its restrictor $\boxed{4}$ and on its nuclear scope $\boxed{5}$. In the case of the NP *every man*, the final constraints for the restrictor $\boxed{4}$ will be determined by a separate proposition contributed by the noun *man*. In the case of the NP *everybody*, the final constraints for $\boxed{4}$ will come from a proposition directly contributed by *everybody* itself. To see an example of the latter case, consider (26) and its derivation tree in Fig. 12 with the elementary semantic entries. The semantic representation for *everybody* contains a proposition with the quantifier (label l_2) and a proposition (l_3) that contributes to the restriction of the quantifier. This last relation is expressed by the constraint $\boxed{4} \geq l_3$.

(26) Everybody laughs

Our main concern, however, is how to determine the final constrains for the nuclear scope $\boxed{5}$. In contrast to several previous LTAG semantics approaches (Joshi and Vijay-Shanker, 1999, Kallmeyer and Joshi, 2003) we do not assume two semantic parts of a quantifier like *every* linked to two different trees. In Joshi and Vijay-Shanker's (1999) and Kallmeyer and Joshi's (2003), quantifiers have a multicomponent tree set containing an auxiliary tree that contributes the (nuclear) scope part and an initial tree that contributes the predicate argument part. This complicates the formal machinery unnecessary. So far we do not see any example of a construction where the separation of the scope part in the syntax is really needed. Even the inverse linking cases with complex NPs where an analysis using the scope parts was proposed (Joshi et al., 2003) can be accounted for without separate scope parts. We will show that in section 4.6.

Concerning the nuclear scope of quantificational NPs, two things must be guaranteed: 1. the proposition to which a quantifier attaches must be in its nuclear scope and 2. a quantifier cannot scope higher than the next finite clause. In other words, there is a minimal nuclear scope and a maximal nuclear scope for each quantifier.

Minimal scope comes from the tree to which the quantifier attaches. In (26), for example, the minimal nuclear scope is the *laugh* proposition (label l_1). This minimal proposition is a global feature of the verb tree since it is invariant no matter how much embedding material attaches

---

to S and VP-modifiers. However, in their analysis, an S-modifying adverb and a VP-modifying adverb can ambiguously scope over each other, contrary to fact. See footnote 22.
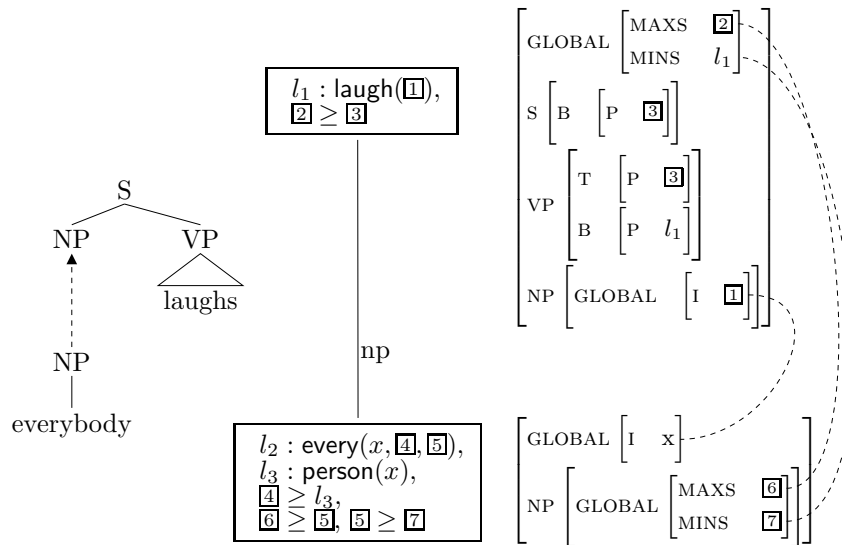
$l_1 : \mathsf{laugh}(\boxed{1})$,
$\boxed{2} \geq \boxed{3}$

S
NP   VP
laughs

NP

NP

everybody

np

$l_2 : \mathsf{every}(x, \boxed{4}, \boxed{5})$,
$l_3 : \mathsf{person}(x)$,
$\boxed{4} \geq l_3$,
$\boxed{6} \geq \boxed{5}$, $\boxed{5} \geq \boxed{7}$

$$\begin{bmatrix} \text{GLOBAL} & \begin{bmatrix} \text{MAXS} & \boxed{2} \\ \text{MINS} & l_1 \end{bmatrix} \\ \text{S} & \begin{bmatrix} \text{B} & \begin{bmatrix} \text{P} & \boxed{3} \end{bmatrix} \end{bmatrix} \\ \text{VP} & \begin{bmatrix} \text{T} & \begin{bmatrix} \text{P} & \boxed{3} \end{bmatrix} \\ \text{B} & \begin{bmatrix} \text{P} & l_1 \end{bmatrix} \end{bmatrix} \\ \text{NP} & \begin{bmatrix} \text{GLOBAL} & \begin{bmatrix} \text{I} & \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix} \text{GLOBAL} & \begin{bmatrix} \text{I} & \text{x} \end{bmatrix} \\ \text{NP} & \begin{bmatrix} \text{GLOBAL} & \begin{bmatrix} \text{MAXS} & \boxed{6} \\ \text{MINS} & \boxed{7} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 12. Analysis of (26)

to the verb tree. I.e., even if further adverbs, raising verbs or other operators attach to the verbal spine, the quantifier can be inside their scope. In other words the minimal proposition in the scope of the quantifier is still the proposition l_1. Therefore we propose a global feature MINS provided by the verbal tree that determines the minimal scope of any attaching quantifier.

Similarly the maximal nuclear scope of an attaching quantifier is also determined by the verbal tree, namely by its largest embedding finite clause (including all adverbs, quantifiers, etc. occurring in that clause). This is also a global property. Therefore we propose a second global feature, MAXS, provided by a verbal tree that determines the maximal scope of any attaching quantificational NP. The MAXS proposition includes all operators attaching to the VP spine that do not embed a finite clause. Therefore it embeds the proposition linked to the bottom feature of the S node (constraint $\boxed{2} \geq \boxed{3}$ in our example).[28]

The quantifier asks for the global MAXS and MINS features of the tree it attaches to (see the features $\boxed{6}$ and $\boxed{7}$ on the root node position NP in Fig. 12). The scope constraints of the quantifier delimit the nuclear scope (here $\boxed{5}$) by these two values (constraints $\boxed{6} \geq \boxed{5}$, $\boxed{5} \geq \boxed{7}$ in our example).

---

[28] Attitude verbs, since they embed a finite clause, will embed the proposition of the S node and also the MAXS proposition. We will see that in section 4.5.

Note that in contrast to previous proposals using the same kind of semantic representations (Kallmeyer and Joshi, 2003, Gardent and Kallmeyer, 2003, Kallmeyer and Romero, 2004) the constraint for the maximal nuclear scope concerns the nuclear scope itself ($\boxed{6} \geq \boxed{5}$) and not the quantifier label ($\boxed{6} \geq l_2$). For simple examples this amounts to the same. But we will see in section 4.6 that for inverse linking examples with nested quantifiers this makes a crucial difference.

In Fig. 12 we obtain the following identifications: $\boxed{6} = \boxed{2}$ and $\boxed{7} = l_1$ (equation of the global features of the verb with the global request on the root of the NP tree), $\boxed{1} = x$ (equation of global features of NP tree and the global request at the substitution node in the *laugh* tree) and $\boxed{3} = l_1$ (final top-bottom equation). The result is (27), which has just one disambiguation: $\boxed{2} \to l_2, \boxed{4} \to l_3, \boxed{5} \to l_1$.

$$(27) \quad \boxed{\begin{array}{l} l_1 : \mathsf{laugh}(x),\ l_2 : \mathsf{every}(x, \boxed{4}, \boxed{5}),\ l_3 : \mathsf{person}(x) \\ \boxed{2} \geq l_1, \boxed{4} \geq l_3, \boxed{2} \geq \boxed{5}, \boxed{5} \geq l_1 \end{array}}$$

Note that the analysis of quantifiers uses only global features. Features linked to specific nodes are needed for articulating the VP spine which is necessary to solve the missing link problem.

With our quantifier analysis, we obtain underspecified semantic representations for ambiguous examples such as (28a). The semantics is shown in (28b). (28b) has two disambiguations that lead to wide scope of *some* and *every* respectively.

(28) a. Someone likes everybody

$$\text{b.} \quad \boxed{\begin{array}{l} l_1 : \mathsf{like}(x,y),\ l_2 : \mathsf{some}(x, \boxed{4}, \boxed{5}),\ l_3 : \mathsf{person}(x), \\ l_4 : \mathsf{every}(y, \boxed{8}, \boxed{9}),\ l_5 : \mathsf{person}(y) \\ \boxed{2} \geq l_1, \boxed{4} \geq l_3, \boxed{2} \geq \boxed{5},\ \boxed{5} \geq l_1, \boxed{8} \geq l_5, \boxed{2} \geq \boxed{9}, \boxed{9} \geq l_1 \end{array}}$$

(29) Someone apparently laughs

(29) (and similarly, (17), p. 30) allows narrow and wide scope of the quantificational NP, since the scope constraints for the quantifier are independent from the adverb/raising verb.

4.4. CONTROL VERBS

With respect to scope, control verbs are completely parallel to S-adverbs, e.g., *intentionally* in (24). They attach to the VP spine and they take everything that is on the spine below the attachment site into their scope. To see how this is achieved, see for example Fig. 13 for the analysis of (30). The propositional argument of the control verb (here
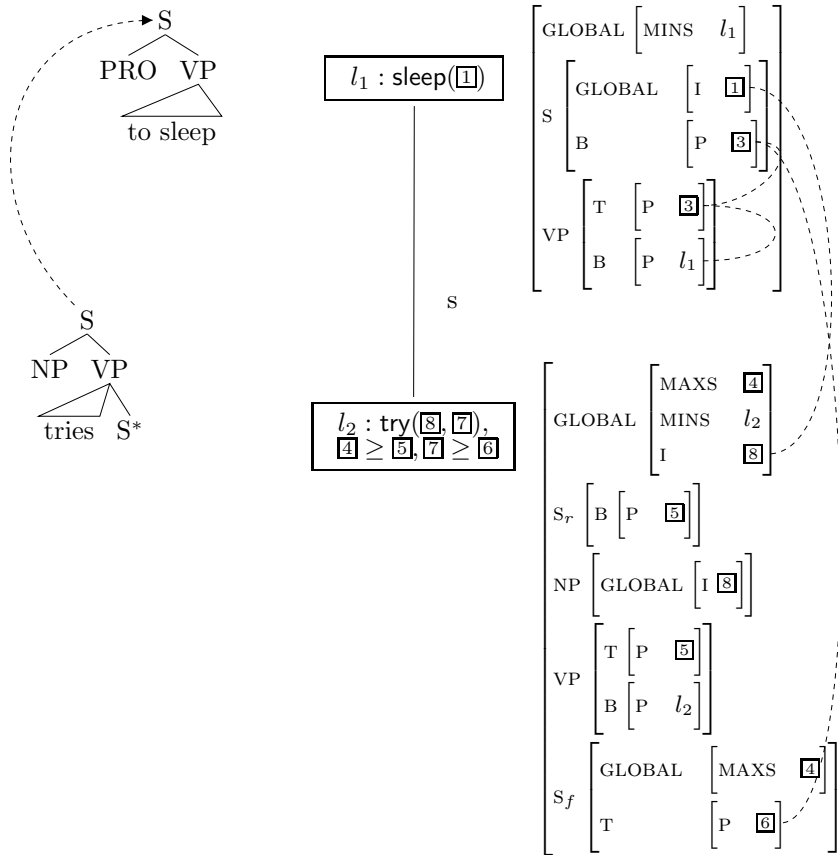
S
PRO VP
to sleep

$l_1 : \mathsf{sleep}(\boxed{1})$

$$\begin{bmatrix} \text{GLOBAL} & \begin{bmatrix} \text{MINS} & l_1 \end{bmatrix} \\ \text{S} & \begin{bmatrix} \text{GLOBAL} & \begin{bmatrix} \text{I} & \boxed{1} \end{bmatrix} \\ \text{B} & \begin{bmatrix} \text{P} & \boxed{3} \end{bmatrix} \end{bmatrix} \\ \text{VP} & \begin{bmatrix} \text{T} & \begin{bmatrix} \text{P} & \boxed{3} \end{bmatrix} \\ \text{B} & \begin{bmatrix} \text{P} & l_1 \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

s

S
NP VP
tries S*

$l_2 : \mathsf{try}(\boxed{8}, \boxed{7}),$
$\boxed{4} \geq \boxed{5}, \boxed{7} \geq \boxed{6}$

$$\begin{bmatrix} \text{GLOBAL} & \begin{bmatrix} \text{MAXS} & \boxed{4} \\ \text{MINS} & l_2 \\ \text{I} & \boxed{8} \end{bmatrix} \\ \text{S}_r & \begin{bmatrix} \text{B} & \begin{bmatrix} \text{P} & \boxed{5} \end{bmatrix} \end{bmatrix} \\ \text{NP} & \begin{bmatrix} \text{GLOBAL} & \begin{bmatrix} \text{I} & \boxed{8} \end{bmatrix} \end{bmatrix} \\ \text{VP} & \begin{bmatrix} \text{T} & \begin{bmatrix} \text{P} & \boxed{5} \end{bmatrix} \\ \text{B} & \begin{bmatrix} \text{P} & l_2 \end{bmatrix} \end{bmatrix} \\ \text{S}_f & \begin{bmatrix} \text{GLOBAL} & \begin{bmatrix} \text{MAXS} & \boxed{4} \end{bmatrix} \\ \text{T} & \begin{bmatrix} \text{P} & \boxed{6} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 13. Analysis of (30) *John tries to sleep*

$\boxed{7}$) scopes over the proposition found in the bottom feature structure (proposition $\boxed{3}$) of the attachment site (constraint $\boxed{7} \geq \boxed{6}$ and $\boxed{6}$ will be identified with $\boxed{3}$).

(30) John tries to sleep

Concerning MAXS, infinitives do not have a MAXS feature since they do not introduce a finite clause proposition. More precisely, their global MAXS will be determined by an attaching embedding finite verb. Therefore, the global MAXS of the control verb (here $\boxed{4}$) is provided at the foot node and consequently passed to the embedded infinitive and from there to any potential quantificational NP.

Control verbs provide an individual argument for the embedded infinitive. In (30), since we have a subject control verb, the subject of *tries* (here $\boxed{8}$) will be passed as the subject of *sleep* (here $\boxed{1}$). Since
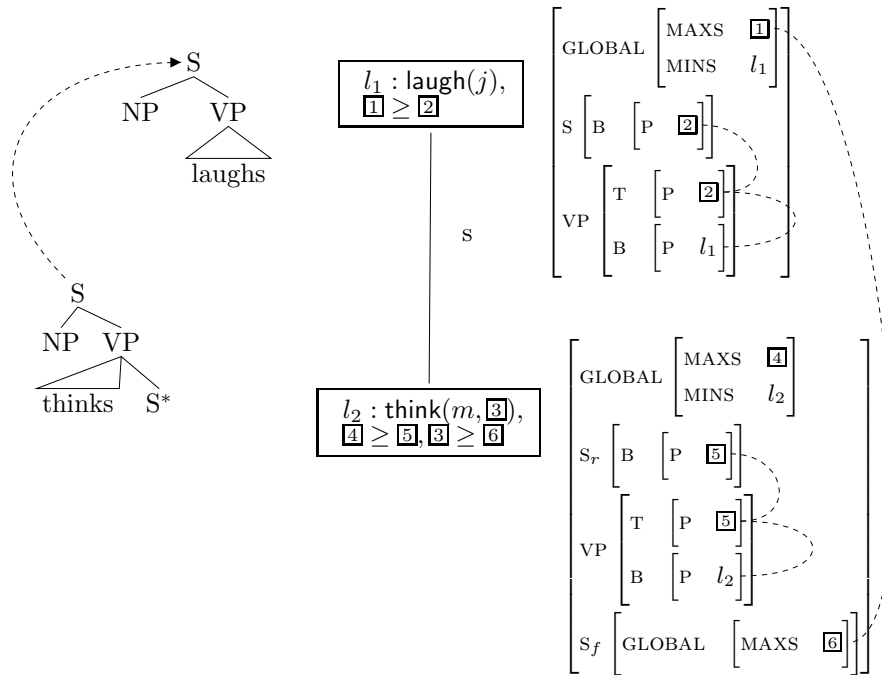
Figure 14. Analysis of (32)

the control verb provides this individual argument, we take it to be its global feature I. The infinitive requests for this global feature at the S node where the embedding finite verb will adjoin and identifies it with its individual argument.

In the case of an object control verb as in (31) the object is provided to the infinitive as argument.

(31) John forces Mary to come

4.5. ATTITUDE VERBS

Syntactically, attitude verbs resemble control verbs, i.e., their elementary trees look similar. See for example the elementary trees for (32) in Fig. 14.

(32) Mary thinks John laughs

The analysis of attitude verbs such as *thinks* in (32) is shown in Fig. 14. Attitude verbs, in contrast to control verbs, take a finite clause

as their argument. This means that the embedded finite verb specifies a MAXS and that the attitude verb will be able to operate over it and thus block scope. The difference between the two types of verbs is modelled in the following way: whereas the complement of a control verb must scope over the relevant P value of the embedded verb ($\boxed{7} \geq \boxed{6}$ in Fig. 13), the complement of an attitude verb must scope over the MAXS of the embedded verb ($\boxed{3} \geq \boxed{6}$ in Fig. 14). In Fig. 14, the foot node of the *thinks* tree is the argument position for the embedded sentence. At this node we place a request for the embedded MAXS (here $\boxed{6}$). The adjunction then leads to the unification $\boxed{6} = \boxed{1}$.

This analysis means that quantifiers attaching to the lower verb can scope over *tries* but not over *thinks*. E.g., in (33), wide scope of *everybody* is excluded.

(33) Mary thinks John likes everybody

Now we have seen enough to be able to analyze another missing link example from section 2, namely (34) where we have an attitude verb and, additionally, a raising verb or adverb in the embedded sentence.

(34) a. Mary claims John seems to laugh
     b. Mary claims John apparently laughs

In general, by unification, in a finite clause, the proposition introduced by the topmost adverb/raising verb is the P value of the root of the verb tree which is below the MAXS proposition. Since attitude verbs as *claim* embed the MAXS of their argument, they necessarily take scope over any embedded raising verb/adverb. The analysis of (34b) is shown in Fig. 15.

4.6. Nested quantificational NPs

Now we will consider examples such as (21), repeated in (35) with three quantificational NPs where two of them are nested.

(35) Two policemen spy on someone from every city.

As mentioned above, (35) has only four possible scope orders out of five:

(36) a. $2 > \exists > \forall$ b. $\exists > \forall, 2$
      c. $2 > \forall > \exists$ d. $\forall > \exists > 2$ e. $* \forall > 2 > \exists$
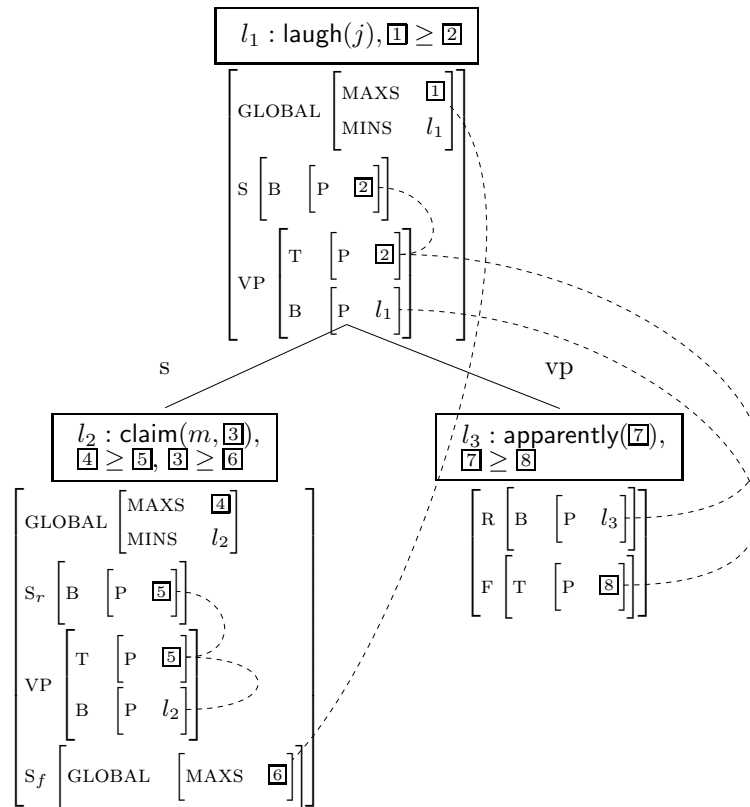
$$l_1 : \mathsf{laugh}(j), \boxed{1} \geq \boxed{2}$$

$$\begin{bmatrix} \text{GLOBAL} & \begin{bmatrix} \text{MAXS} & \boxed{1} \\ \text{MINS} & l_1 \end{bmatrix} \\ \text{S} & \begin{bmatrix} \text{B} & \begin{bmatrix} \text{P} & \boxed{2} \end{bmatrix} \end{bmatrix} \\ \text{VP} & \begin{bmatrix} \text{T} & \begin{bmatrix} \text{P} & \boxed{2} \end{bmatrix} \\ \text{B} & \begin{bmatrix} \text{P} & l_1 \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

s   vp

$$l_2 : \mathsf{claim}(m, \boxed{3}),\ \boxed{4} \geq \boxed{5}, \boxed{3} \geq \boxed{6}$$

$$\begin{bmatrix} \text{GLOBAL} & \begin{bmatrix} \text{MAXS} & \boxed{4} \\ \text{MINS} & l_2 \end{bmatrix} \\ \text{S}_r & \begin{bmatrix} \text{B} & \begin{bmatrix} \text{P} & \boxed{5} \end{bmatrix} \end{bmatrix} \\ \text{VP} & \begin{bmatrix} \text{T} & \begin{bmatrix} \text{P} & \boxed{5} \end{bmatrix} \\ \text{B} & \begin{bmatrix} \text{P} & l_2 \end{bmatrix} \end{bmatrix} \\ \text{S}_f & \begin{bmatrix} \text{GLOBAL} & \begin{bmatrix} \text{MAXS} & \boxed{6} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$l_3 : \mathsf{apparently}(\boxed{7}),\ \boxed{7} \geq \boxed{8}$$

$$\begin{bmatrix} \text{R} & \begin{bmatrix} \text{B} & \begin{bmatrix} \text{P} & l_3 \end{bmatrix} \end{bmatrix} \\ \text{F} & \begin{bmatrix} \text{T} & \begin{bmatrix} \text{P} & \boxed{8} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 15. Analysis of (34b)

The question is how to derive an underspecified representation that allows for the readings (36a-d) and excludes reading (36e).

Joshi et al. (2003) propose to derive the specific scope constraints for such constructions –namely, the fact that the third quantificational NP cannot scopally intervene between the host and nested quantifiers– from the way the scope parts of the different quantifiers combine. This account is not possible if the scope part is not separated from the NP tree carrying the lexical material.

In the following we will show that the desired constraints can be obtained without multicomponents if we use the scope constraint language we have so far.[29]

---

[29] Readings (b') and (b") from footnote 25 –* $\exists > \forall > 2$ and * $\exists > 2 > \forall$ respectively– are excluded on independent grounds, since they do not yield tree-shaped terms: Assume that the three quantifiers are $l_2 : 2(x, \boxed{3}, \boxed{4})$, $l_4 : \mathsf{some}(y, \boxed{7}, \boxed{8})$ and $l_8 : \mathsf{every}(z, \boxed{13}, \boxed{14})$. Then for the scope order $\mathsf{some} > 2$ and $\mathsf{some} > \mathsf{every}$ we have the following: 2 is in the nuclear scope of some, i.e., $\boxed{8} \geq l_2$ because of spy being in the nuclear scope of both, some and 2. The proposition coming with *from*
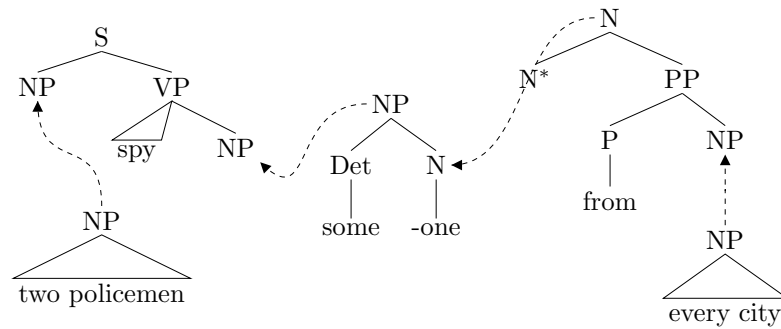
Figure 16. Derivation for (35)

To exclude the reading every $> 2 >$ some, the idea, in a nutshell, is the follwoing. We derive a constraint saying that the maximal nuclear scope of every is the some proposition. This means that every can rise and take scope over some, but, if it does so, then it has to take *immediate* scope over some.

The analysis of (35) is shown in Fig. 16 and Fig. 17. The new MAXS value for the maximal scope of embedded quantifiers is the global MAXS of the *from* tree. It is equated with the label of the quantifier to which *from* attaches. Consequently, any quantifier attaching to *from* (here every) takes the label of this quantifier as its limit for maximal scope. Quantifiers now provide their labels as potential global MAXS at their N nodes in case something adjoins here. Modifiers such as *from* receive this value then as the global MAXS that limits the nuclear scope of further embedded quantifiers. In our example, the maximal nuclear scope of *every* (here $\boxed{15}$) will be equated with the global MAXS of *from* (here $\boxed{12}$), which in turn is equated with l_4. For *every*, we consequently obtain $l_4 \geq \boxed{14}$.

The result of the semantic unifications in Fig. 17 is the semantic representation (37):

---

must of course be in the restriction of some and it must be in the nuclear scope of every. Consequently, if every is in the scope of some, it must be in its restriction, i.e., $\boxed{7} \geq l_8$. Then, if 2 > every, with $\boxed{8} \geq l_2$ (2 in the nuclear scope of some), every would have to be at the same time in the nuclear scope of some, i.e., $\boxed{8} \geq l_8$. This would violate the assumed principle that terms are tree-shaped, i.e., that nothing is part of two different arguments of the same predicate (here every cannot be part of $\boxed{7}$ and $\boxed{8}$ at the same time). Similarly, if every > 2, then 2 would be at the same time in the restriction and the nuclear scope of some, leading to the same problem.

42

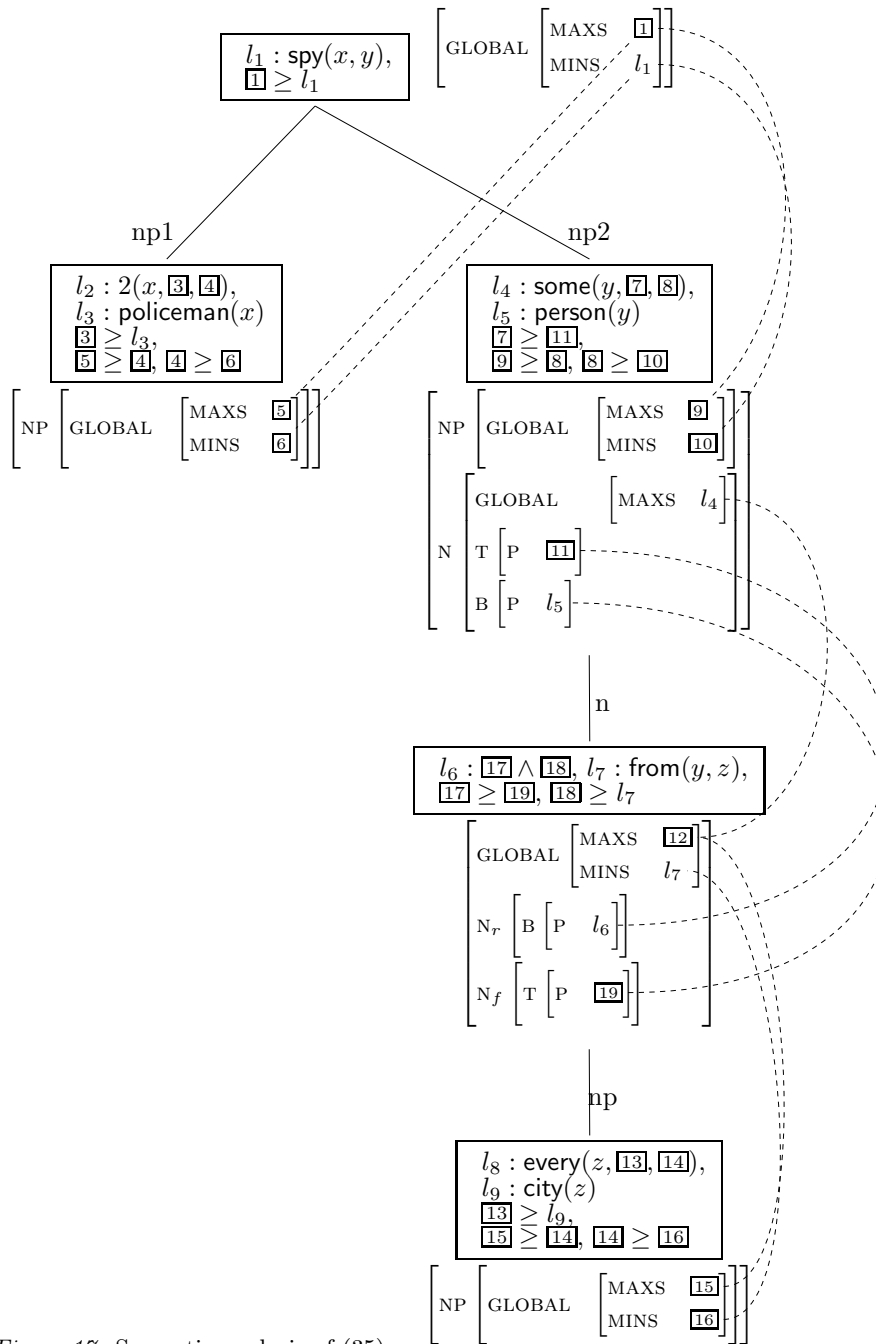$l_1 : \mathsf{spy}(x, y),$
$\boxed{1} \geq l_1$

$\left[\text{GLOBAL} \begin{bmatrix} \text{MAXS} & \boxed{1} \\ \text{MINS} & l_1 \end{bmatrix} \right]$

np1

$l_2 : 2(x, \boxed{3}, \boxed{4}),$
$l_3 : \mathsf{policeman}(x)$
$\boxed{3} \geq l_3,$
$\boxed{5} \geq \boxed{4}, \boxed{4} \geq \boxed{6}$

np2

$l_4 : \mathsf{some}(y, \boxed{7}, \boxed{8}),$
$l_5 : \mathsf{person}(y)$
$\boxed{7} \geq \boxed{11},$
$\boxed{9} \geq \boxed{8}, \boxed{8} \geq \boxed{10}$

$\left[\text{NP} \left[\text{GLOBAL} \begin{bmatrix} \text{MAXS} & \boxed{5} \\ \text{MINS} & \boxed{6} \end{bmatrix} \right] \right]$

$\left[\text{NP} \left[\text{GLOBAL} \begin{bmatrix} \text{MAXS} & \boxed{9} \\ \text{MINS} & \boxed{10} \end{bmatrix} \right] \right.$

$\left. \text{N} \begin{bmatrix} \text{GLOBAL} & \begin{bmatrix} \text{MAXS} & l_4 \end{bmatrix} \\ \text{T} & \begin{bmatrix} \text{P} & \boxed{11} \end{bmatrix} \\ \text{B} & \begin{bmatrix} \text{P} & l_5 \end{bmatrix} \end{bmatrix} \right]$

n

$l_6 : \boxed{17} \wedge \boxed{18}, l_7 : \mathsf{from}(y, z),$
$\boxed{17} \geq \boxed{19}, \boxed{18} \geq l_7$

$\left[\begin{array}{l} \text{GLOBAL} \begin{bmatrix} \text{MAXS} & \boxed{12} \\ \text{MINS} & l_7 \end{bmatrix} \\ \text{N}_r \begin{bmatrix} \text{B} \begin{bmatrix} \text{P} & l_6 \end{bmatrix} \end{bmatrix} \\ \text{N}_f \begin{bmatrix} \text{T} \begin{bmatrix} \text{P} & \boxed{19} \end{bmatrix} \end{bmatrix} \end{array} \right]$

np

$l_8 : \mathsf{every}(z, \boxed{13}, \boxed{14}),$
$l_9 : \mathsf{city}(z)$
$\boxed{13} \geq l_9,$
$\boxed{15} \geq \boxed{14}, \boxed{14} \geq \boxed{16}$

$\left[\text{NP} \left[\text{GLOBAL} \begin{bmatrix} \text{MAXS} & \boxed{15} \\ \text{MINS} & \boxed{16} \end{bmatrix} \right] \right]$

Figure 17. Semantic analysis of (35)

(37)
$$l_1 : \mathsf{spy}(x,y),\ l_2 : 2(x,\boxed{3},\boxed{4}),\ l_3 : \mathsf{policeman}(x)$$
$$l_4 : \mathsf{some}(y,\boxed{7},\boxed{8}),\ l_6 : \boxed{17} \wedge \boxed{18},\ l_5 : \mathsf{person}(y),\ l_7 : \mathsf{from}(y,z)$$
$$l_8 : \mathsf{every}(z,\boxed{13},\boxed{14}),\ l_9 : \mathsf{city}(z)$$
$$\boxed{1} \geq l_1,\ \boxed{3} \geq l_3,\ \boxed{1} \geq \boxed{4},\ \boxed{4} \geq l_1,\ \boxed{7} \geq l_6,\ \boxed{1} \geq \boxed{8},\ \boxed{8} \geq l_1$$
$$\boxed{17} \geq l_5,\ \boxed{18} \geq l_7,\ \boxed{13} \geq l_9,\ l_4 \geq \boxed{14},\ \boxed{14} \geq l_7$$

As desired the scope order $\mathsf{every} > 2 > \mathsf{some}$ is not possible: Assume it was possible, i.e., $l_8 > l_2 > l_4$. Then with $\boxed{14} \geq l_7$ and $l_8 > l_4 > \boxed{7} \geq l_6 > \boxed{18} \geq l_7$, it follows that $\boxed{14} \geq l_2$ and $\boxed{14} \geq l_4$. Consequently the disambiguation δ would be such that $\boxed{13} \to l_9, \boxed{14} \to l_2, \boxed{3} \to l_3, \boxed{4} \to l_4, \boxed{7} \to l_6, \boxed{8} \to l_1, \boxed{17} \to l_5, \boxed{18} \to l_7$. But then $\delta(\boxed{14}) = l_2 > \delta(\boxed{4}) = l_4$. This contradicts the constraint $l_4 \geq \boxed{14}$.

In order to obtain the underspecified representation (37), we have to allow scope constraints that do not correspond to normal dominance constraints since we need a constraint of the form $l \geq x$ where l is a label, x a meta-variable. As already mentioned, with some restrictions on these constraints, they could probably be integrated into the constraint solver for dominance constraints so that polynomial constraint solving is still possible. It is an important result that it is enough to have only a slight extension of normal dominance constraints in order to capture the scope relations in nested quantifiers. In particular, the original constraint language allowing only subordination constraints does not need to be extended in the style of the quantifier sets proposed by Joshi et al. (2003).[30]

---

[30] Following Larson (1987), the main concern in the literature on nested NPs has been to rule out the reading (21e) where another quantifier scopally intervenes between the host and the nested quantifier. However, more recently, Sauerland (2000) has presented data suggesting that intervention is possible if the extra quantifier is not an NP itself but a verbal element (e.g. *want*), as in (38). We presently do not have an account for why this difference results (but see Joshi et al. 2003). However, note that the relaxation of scope constraints with verbal intervenors seems to be a wider phenomenon independent of the nested NPs at hand. In (39), e.g., *someone* cannot scope under the verbal conjunction, but it can scope under it if some intensional verbal element intervenes, as in (40).

(38) John wants to meet someone from every city you do (want to meet someone from).

(39) Someone from NY jumped and ran.
   $*\ \exists x[from(x,ny) \wedge jumped(x)]\ \wedge\ \exists y[from(y,ny) \wedge ran(y)]$

(40) Someone from NY is likely to win the lottery but unlikely to win this bet.
   $likely(\exists x[from(x,ny) \wedge win(x,l)])\ \wedge\ unlikely(\exists y[from(y,ny) \wedge win(y,b)])$

5. Situation Binding in LTAG

5.1. THEORETICAL BACKGROUND

Independently of LTAG or any other grammar formalism, in order to represent the semantics of natural language, a formal language is needed that allows us to evaluate linguistic expressions not only at the actual situation or world but also at other logically possible worlds. In this subsection, we will describe two formal languages that have been traditionally used for this purpose in the literature, namely Modal Predicate Logic (ModPrL) –without situation terms– and Two-Sorted Type Theory (Ty2) –with situation terms and quantification over them. Then we will summarize Cresswell's (1990) argument in favor of Ty2. Because of this argument, the formal language assumed in the present paper and defined in section 3.1 follows Ty2 rather than ModPrL, that is, it includes situation variables and direct quantification over them.

A simple example of intensional language is Modal Predicate Logic. ModPrL is like standard Predicate Logic with the addition of the syn-categorematic symbols $\square$ and $\lozenge$. For a model M of the sort described in (41) and an assignment function g from variables to individuals in M, the interpretation of terms and formulas is defined in (42) and (43) (see Gamut, 1991: vol. II, ch 3.).

(41) A model M for a modal predicate logic language L consists of:
  i.   a non empty set of possible worlds W
  ii.   an accessibility relation R on W [31]
  iii. a domain function assigning a domain of individuals D_w to each w $\in$ W
  iv.   an interpretation function I such that:
    a.   I assigns an entity I(c) to each constant c of L, and
    b.   for every world w $\in$ W, I assigns a subset $I_w(P)$ of $(D_w)^n$ to each n-ary predicate letter P of L.

---

[31] R can stand for the epistemic (*Epi*, e.g. with *know*), deontic (*Deo*, e.g. with *have to*), doxastic (*Dox*, e.g. with *believe*), etc. accessibility relation. wRw' is read as 'w' is accessible from w'. E.g., $wEpiw'$ is read as 'w' is epistemically accessible from w', that is, 'for all we know in the actual world w, w' could equal w'.

(42) Interpretation of terms:[32] $[\![t]\!]^{M,w,g}$

    1.   If t is a constant,

        $[\![t]\!]^{M,w,g} = \mathrm{I}(t)$ if $\mathrm{I}(t) \in \mathrm{D}_w$ (and undefined otherwise).

    2.   If t is a variable,

        $[\![t]\!]^{M,w,g} = \mathrm{g}(t)$ if $\mathrm{g}(t) \in \mathrm{D}_w$ (and undefined otherwise).

(43) Interpretation of formulas: $[\![\phi]\!]^{M,w,g}$

    3. $[\![P(t_1, ..., t_n)]\!]^{M,w,g} = 1$

$$\text{iff } \langle [\![t_1]\!]^{M,w,g}, ..., [\![t_n]\!]^{M,w,g} \rangle \in [\![P]\!]^{M,w,g}$$

    4. $[\![\neg\phi]\!]^{M,w,g} = 1$ iff $[\![\phi]\!]^{M,w,g} = 0$

    5.1. $[\![\phi \wedge \psi]\!]^{M,w,g} = 1$ iff $[\![\phi]\!]^{M,w,g} = [\![\psi]\!]^{M,w,g} = 1$

    5.2. ...

    6.1. $[\![\forall x[\phi]]\!]^{M,w,g} = 1$ iff for all $d \in \mathrm{D}_w$: $[\![\phi]\!]^{M,w,g^{d/x}} = 1$

    6.2. $[\![\exists x[\phi]]\!]^{M,w,g} = 1$ iff for some $d \in \mathrm{D}_w$: $[\![\phi]\!]^{M,w,g^{d/x}} = 1$

    7.1. $[\![\square[\phi]]\!]^{M,w,g} = 1$ iff for all w' such that $w\mathrm{R}w'$: $[\![\phi]\!]^{M,w',g} = 1$

    7.2. $[\![\diamondsuit[\phi]]\!]^{M,w,g} = 1$ iff for some w' such that $w\mathrm{R}w'$: $[\![\phi]\!]^{M,w',g} = 1$

To see an example, the sentence (44) is translated into ModPrL as in (45) for its reading 'It must be the case that someone or other from New York wins the lottery'.

(44) Someone from NY must have won the lottery.

(45) $\square[\exists x[\mathsf{person}(x) \wedge \mathsf{from}(x, ny) \wedge \mathsf{win}(x)]$

An alternative to intensional languages like ModPrL is the so-called Two-Sorted Type Theory (Ty2) (Gallin, 1975:58-63). Ty2 is characterized by having two sorts of basic individuals rather than one: besides regular individuals (type e), we add possible situations or worlds as a basic type (type s). This means that a Ty2 (first-order) formal language will include variables over worlds, and that these world variables will serve as arguments of predicates and will combine with the syncategorematic symbols $\forall$ and $\exists$. The interpretation is defined in (47)-(48). A Ty2 version of ModPrL assigns the translation (49) to the aforementioned reading of (44) –with *must* interpreted at world w_0– with direct quantification over world variables.

---

[32] Here we slightly deviate from Gamut (1991). In the end, though, in both cases the condition that $\mathrm{I}(t)$ and $\mathrm{g}(t) \in \mathrm{D}_w$ ends up as a presupposition rather than as part of the assertion.

(46) A model M for the Ty2 version of modal predicate logic language L consists of:

    i.   a non empty set of individuals D_e,

    ii.   a non-empty set of possible worlds D_s

    iii.   an accesibility relation R on D_s

    iv.   an interpretation function I such that:

        a.   I assigns an entity I(c) to each constant c of L, and

        b.   I assigns a subset I(P) of $D_{e,1} \times ... \times D_{e,n} \times D_s$ to each n-ary predicate letter P of L.

(47) Interpretation of terms: $[\![t]\!]^{M,g}$

    1. If t is a constant, $[\![t]\!]^{M,g} = I(t)$.

    2. If t is a variable, $[\![t]\!]^{M,g} = g(t)$.

(48) Interpretation of formulas: $[\![\phi]\!]^{M,g}$

    3. $[\![P(t_1, ..., t_n, w)]\!]^{M,g} = 1$

                          iff $\langle [\![t_1]\!]^{M,g}, ..., [\![t_n]\!]^{M,g}, [\![w]\!]^{M,g} \rangle \in [\![P]\!]^{M,g}$

    4. $[\![\neg\phi]\!]^{M,g} = 1$ iff $[\![\phi]\!]^{M,g} = 0$

    5.1. $[\![\phi \wedge \psi]\!]^{M,g} = 1$ iff $[\![\phi]\!]^{M,g} = [\![\psi]\!]^{M,g} = 1$

    5.2. ...

    For any variable v of type e or s:

    6.1. $[\![\,\forall v[\phi]\,]\!]^{M,g} = 1$ iff for all $d \in D_e \cup D_w$: $[\![\phi]\!]^{M,g^{d/v}} = 1$

    6.2. $[\![\,\exists v[\phi]\,]\!]^{M,g} = 1$ iff for some $d \in D_e \cup D_w$: $[\![\phi]\!]^{M,g^{d/v}} = 1$

(49) $\forall w' [\ w_0 R w' \rightarrow \exists x[\mathsf{person}(x, w') \wedge \mathsf{from}(x, ny, w') \wedge \mathsf{win}(x, w')]\ ]$

Note that a Ty2 language has more expressive power than ModPrL, as the former, but not the latter, will allow for a predicate within the scope of a modal to be evaluated at a higher world w. For (44) e.g., Ty2 will allow for the truth conditions in (50) –where the complex predicate *-one from NY* is in the scope of the modal but evaluated at the actual world w_0–, but standard ModPrL will not generate this reading.

(50) $\forall w' [\ w_0 R w' \rightarrow \exists x[\mathsf{person}(x, w_0) \wedge \mathsf{from}(x, ny, w_0) \wedge \mathsf{win}(x, w')]\ ]$

This means that the scope and the situation binding of a predicate (e.g. a noun, adjective, verb, preposition, etc.) are two different issues in Ty2 but necessarily go together in standard ModPrL: in the former, a predicate within the scope of several intensional operators can be evaluated at the world w introduced by any of these operators; in the

latter, it will necessarily be evaluated at the world introduced by the closest of these operators. The crucial question is, then, whether scope and situation binding of a predicate go together in natural language –as in ModPrL– or not –as in Ty2.

The empirical observation for natural language is that not all the predicates under the scope of a given intensional operator (e.g. *must*, *if*, *to rumour*) are evaluated at the w' introduced by that operator. For example, in the most salient reading of (51), *every* and thus its restrictor *poor child* scope under *if*, but still *poor child* denotes the set of actual poor children rather than the set of poor children in the hypothetical world.[33] In a similar fashion, in (52), *every* scopes under the second *if* but its restrictor *poor child in the neighborhood* is evaluated at the world introduced by *rumour*.

(51) If every poor child was rich instead, this would be a happy world.

(52) I know there aren't any poor children in this neighborhood. But, if there were poor children in this neighborhood, people would rumour that if every poor child in the neighborhood was rich instead, the mayor would be re-elected.

To capture all the possible world binding readings in natural language, we would need a switch operator **actually**$_n$ for every world binder in the sentence. Cresswell (1990) proves that a formal language with such switch operators has the expressive power of explicit quantification over world variables, that is, the power of a Ty2 language (Cresswell ,1990:ch 4). Ty2 gives us the translation of (51) in (53).

(53) $\forall w\ [\ w_0 R w\ \wedge \forall x[\text{poor-child}(x, w_0) \to \text{rich}(x, w)]$
$\to \text{happy-world}(\text{this}, w)\ ]$

Hence, following Gallin (1975) and Cresswell (1990), our formal language defined in section 3.1 follows Ty2: each predicate has a world argument and we explicitly quantify over world variables.

5.2. THE DATA

The punchline from the previous subsection is that scope and world or situation binding are two separate phenomena in natural language. As a consequence, a formal language with the expressive power of Ty2 is

---

[33] If *poor child* in (51) was evaluated in the hypothetical world w' introduced by *if*, w' would have the contradictory property of being such that all the children that are poor in w' are rich in w'. This is clearly not the intended reading.

needed. Section 4 was concerned with handling scope data. Now that
we have motivated the Ty2-like formal language we use, the present
section tackles situation binding.

The empirical generalizations for situation binding in natural lan-
guage are the following:

- Noun Phrases: The situation variable of an NP can be non-
  locally bound by a distant operator (Cresswell, 1990) (Farkas, 1997)
  (Percus, 2000) or remain free.
- Quantificational elements attached to the VP-spine: The situation
  variable of material attached to the VP spine must be locally bound
  by the closest situation binder scoping over it (Percus, 2000). If there
  is no situation binder scoping over it at all, then it remains free.

When a situation variable remains free, it defaults to the actual
situation s_0.

Data supporting the first generalization are (51)-(52) in the previous
subsection. To see the different situation binding possibilities for NPs
and VP-spine material, consider now sentence (54).

(54) Mary thinks that, in yesterday's game, John sometimes beat the
     winner.

Consider, furthermore, a scenario with s_0 as the actual situation, s as
the situation consisting of yesterday's entire game according to Mary's
beliefs and several subsituations s' corresponding to different rounds of
that game in Mary's beliefs. With this scenario in mind, (54) can be
understood as having *the winner* translated as $\iota x[\mathsf{winner}(x, s_0)]$ ('Of the
actual overall winner x, Mary thinks that John beat x in some rounds')
or as $\iota x[\mathsf{winner}(x, s)]$ ('Mary thinks that in some rounds John beat
whoever the overall winner of the game in Mary's beliefs is'), skipping
in both cases over the local situation binder $\exists s'$. This is represented in
(55). (The translation $\iota x[\mathsf{winner}(x, s')]$ is in principle possible as well,
but having s' for *the winner* and for *beat* is obviously pragmatically
deviant.)

(55) $\forall s \,[\, s \in Dox_m(s_0) \land \mathsf{game}(s) \rightarrow$
     $\exists s'[s' \sqsubseteq s \land \mathsf{beat}(j, \iota x[\mathsf{winner}(x, s_0/s/\#s')], s')] \,]$
     'Mary thinks that in some rounds John beat the actual overall
     winner of the game / whoever the overall winner of the game is in
     Mary's belief worlds.'

In contrast, the situation variable of the verb *beat* must be locally
bound by the $\exists s'$-quantifier of *sometimes*, yielding the subformula
$\mathsf{beat}(j, \iota x[...], s')$ that we see in (55). If it could be distantly bound

and represented as $\mathsf{beat}(j, \iota x[...], s)$, as in (56), the sentence (54) would be predicted to have a reading 'Mary thinks John beat a sometimes winner', contrary to fact.

(56) $\forall s$ [$s \in Dox_m(s_0) \wedge \mathsf{game}(s) \rightarrow$
$\exists s'[s' \sqsubseteq s \wedge \mathsf{beat}(j, \iota x[\mathsf{winner}(x, s')], s)]$]
'Mary thinks John beat in the overall game the winner of some round.'

Furthermore, the (non-quantified) situation within *sometimes* must be locally bound by the $\forall s$-quantification of *think*, yielding the subformula $\exists s'[s' \sqsubseteq s]$ we see in (55). If it could be distantly bound and represented as $\exists s'[s' \sqsubseteq s_0]$, as in (57), the sentence would be able to roughly mean 'For every person that Mary thinks may be the overall winner, John sometimes beat that person', again contrary to fact.[34]

(57) $\forall s$ [$s \in Dox_m(s_0) \wedge \mathsf{game}(s) \rightarrow$
$\exists s'[s' \sqsubseteq s_0 \wedge \mathsf{beat}(j, \iota x[\mathsf{winner}(x, s)], s')]$]
'For every person x that Mary thinks may be the overall winner, John beat x in some round.'

In sum, for NPs, situation binding is not determined by their surface syntax: their situation can be bound at any distance and, if unbound, it defaults to s_0. For quantificational elements on the VP-spine, situation binding is fully determined by the surface syntax: their situation variable is bound by the closest situation binder and, if no such binder exists, it defaults to s_0.

5.3. The analysis

In the same way that situation or world variables were added as an extra argument to each predicate in Ty2, we add situation variables s_0, s, s', s'', etc., situation meta-variables, and the feature s to our LTAG semantic representations and semantic feature structures. Other arguments –e.g agents, patients, goals, etc.– are linked to leaf nodes in the elementary tree, as they get their values from the substitutions that will be perfomed at those nodes. The situation argument is linked to the VP-spine, as it can get its value from quantificational elements adjoined to VP or to S. Since the situation argument of a verb can be accessed from these two nodes but it can possibly only be bound once, it will be given two meta-variables in the semantic feature structure:

---

[34] The expression $Dox_m(s_0)$ stands for the set of doxastic alternatives of Mary in s_0, i.e. the set of possible situations s that conform to Mary's beliefs in s_0. $\sqsubseteq$ denotes the part-of relation between situations, as defined in Kratzer's (1989).

a 'lower' one for VP-adjunctions and a 'higher' one for S-adjunctions, the latter being coindexed with the situation meta-variable in the top part of the VP-node. This is illustrated in the feature structure for *win* in Fig. 18 (see [2] and [3]).

The feature unification mechanism used for scope derives the situation data as well. On the one hand, material attached to the verbal spine will unify its situation variables univocally with the situation variables at the attachment positions. On the other hand, NPs will remain underspecified with respect to situation binding. Since we saw that the binder of an NP's situation can be outside the closest finite clause, we impose no minimality or maximality constraints on NP situations, hence leaving NP situation binding completely underspecified.

(58) Mary thinks every person sometimes won.

This is illustrated in (58), where an attitude verb, a VP-adverb and an embedded quantifier interact for scope and for situation binding. The derivation is in Fig. 18. VP-attachment of *sometimes* -label l_2- guarantees that it will quantify over the 'lower' situation variable [2] of *win*. Note that [2] is the situation argument in the formula $l_1 : \mathsf{win}([1], [2])$ in the semantic representation. S-attachment of *think* secures that *thinks* will quantify over the 'higher' situation variable [3] in the feature structure of *win*. Since by top-bottom unification [3] will be equated with the situation meta-variable [6] of *sometimes*, *think* will end up quantifying over the situation variable of *sometimes*. (If no adjunction had been perfomed at VP, by top-bottom unification, [3] would have been equated with [2] and thus *thinks* would have directly quantified over the situation of *win*.) As for the NP *every person* (or *everyone from NY*), no restrictions are placed on the situation variable [13] of its restrictor proposition $l_3 : \mathsf{person}(x, [13])$: [13] can be bound by any situation binder scoping over it and, if unbound, it will default to s_0.

One obtains the feature value identities $[1] = x, [2] = s'$, $[8] = l_1$, $[4] = l_2$, $[3] = [6]$, $[3] = s$, $[12] = l_1$, and $[11] = [0] = [17]$, which leads to (59):

$$
(59) \quad
\begin{array}{l}
l_1 : \mathsf{win}(x, s'), \quad l_2 : \mathsf{some}(s', s' \sqsubseteq s, [5]), \quad l_3 : \mathsf{every}(x, [9], [10]), \\
l_4 : \mathsf{person}(x, [13]), \quad l_5 : \mathsf{every}(s, s \in Dox_m([14]), [15]) \\
[16] \geq l_5, \quad [15] \geq [0], \quad [0] \geq l_2, \quad [5] \geq l_1, \quad [9] \geq l_4, \quad [10] \geq l_1, \quad [0] \geq [10]
\end{array}
$$

In (59), the maximal scope of the quantificational NP is blocked by the argument of *thinks*: the constraints $[15] \geq [0]$ and $[0] \geq [10]$ guarantee that the argument [15] of *thinks* will include the nuclear scope [10] of *every* and thus scope over the entire $l_3 : \mathsf{every}(x, [9], [10])$. But the order of the quantificational NP and the adverb is unspecified. The situation

$$\begin{bmatrix} \text{GLOBAL} & \begin{bmatrix} \text{MAXS} & \boxed{0} \\ \text{MINS} & l_1 \end{bmatrix} \\ \text{S} & \begin{bmatrix} \text{B} & \begin{bmatrix} \text{P} & \boxed{4} \\ \text{S} & \boxed{3} \end{bmatrix} \end{bmatrix} \\ \text{VP} & \begin{bmatrix} \text{T} & \begin{bmatrix} \text{P} & \boxed{4} \\ \text{S} & \boxed{3} \end{bmatrix} \\ \text{B} & \begin{bmatrix} \text{P} & l_1 \\ \text{S} & \boxed{2} \end{bmatrix} \end{bmatrix} \\ \text{NP1} & \begin{bmatrix} \text{GLOBAL} & \begin{bmatrix} \text{I} & \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\boxed{l_1 : \text{win}(\boxed{1}, \boxed{2}) \\ \boxed{0} \geq \boxed{4}}$$

s   vp   np1

$$\boxed{\begin{array}{l} l_5 : \text{every}(s, s \in Dox_m(\boxed{14}), \boxed{15}), \\ \boxed{16} \geq l_5, \boxed{15} \geq \boxed{17} \end{array}}$$

$$\begin{bmatrix} \text{GLOBAL} & \begin{bmatrix} \text{MAXS} & \boxed{16} \end{bmatrix} \\ \text{S}_r & \begin{bmatrix} \text{B} & \begin{bmatrix} \text{P} & l_5 \\ \text{S} & \boxed{14} \end{bmatrix} \end{bmatrix} \\ \text{S}_f & \begin{bmatrix} \text{GLOBAL} & \begin{bmatrix} \text{MAXS} & \boxed{17} \end{bmatrix} \\ \text{T} & \begin{bmatrix} \text{S} & s \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\boxed{\begin{array}{l} l_3 : \text{every}(x, \boxed{9}, \boxed{10}), \\ l_4 : \text{person}(x, \boxed{13}), \\ \boxed{9} \geq l_4, \boxed{10} \geq \boxed{12}, \\ \boxed{11} \geq \boxed{10} \end{array}}$$

$$\begin{bmatrix} \text{GLOBAL} & \begin{bmatrix} \text{I} & x \end{bmatrix} \\ \text{NP2} & \begin{bmatrix} \text{GLOBAL} & \begin{bmatrix} \text{MAXS} & \boxed{11} \\ \text{MINS} & \boxed{12} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\boxed{\begin{array}{l} l_2 : \text{some}(s', s' \sqsubseteq \boxed{6}, \boxed{5}), \\ \boxed{5} \geq \boxed{8} \end{array}}$$

$$\begin{bmatrix} \text{VP}_r & \begin{bmatrix} \text{B} & \begin{bmatrix} \text{P} & l_2 \\ \text{S} & \boxed{6} \end{bmatrix} \end{bmatrix} \\ \text{VP}_f & \begin{bmatrix} \text{T} & \begin{bmatrix} \text{P} & \boxed{8} \\ \text{S} & s' \end{bmatrix} \end{bmatrix} \end{bmatrix}$$
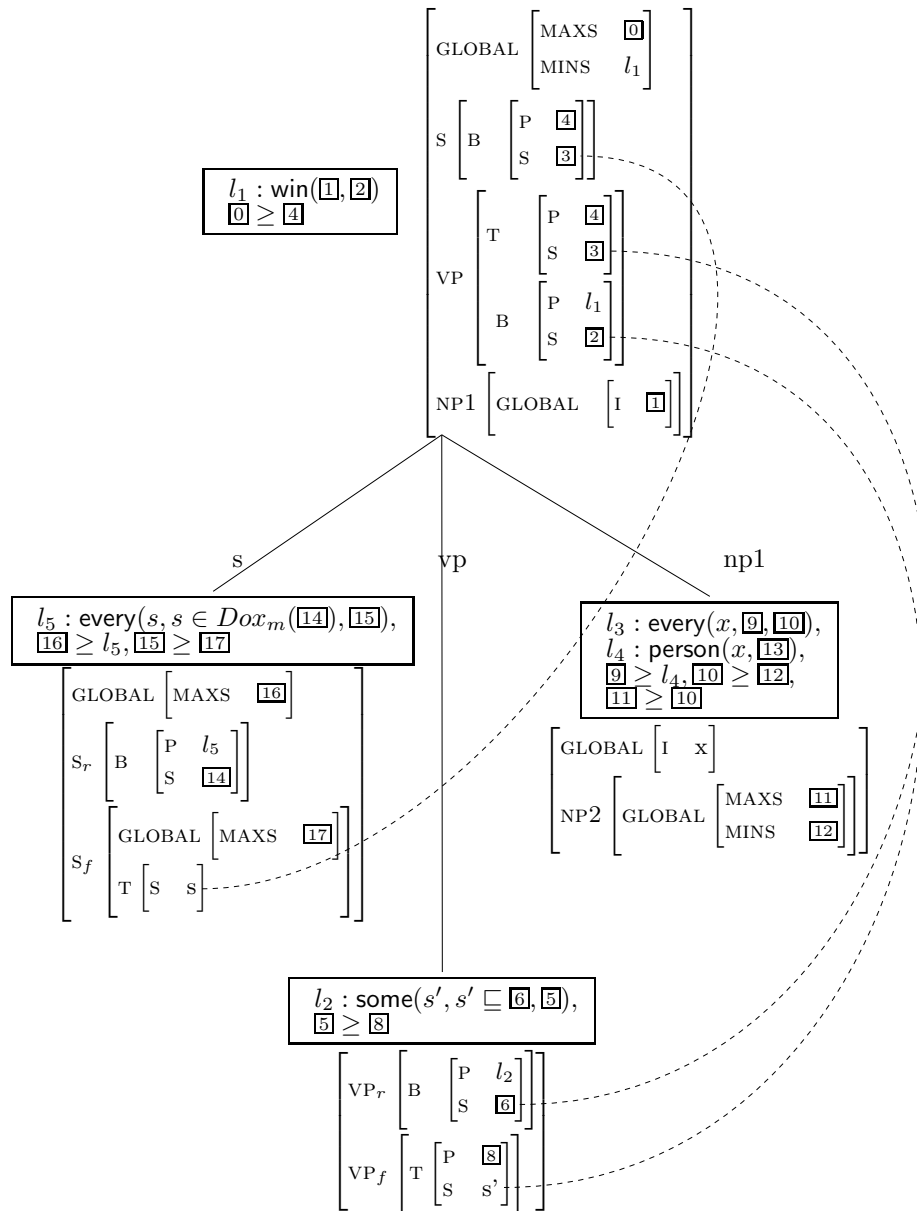
Figure 18. Analysis of (58) *Mary thinks everybody sometimes won* with s feature identifications.

$\boxed{14}$ in the l_5 of *thinks* will default to s_0, since there is no possible binder above it and it will thus remain unbound. The situation variable $\boxed{13}$ in l_4 corresponding to the noun in *every person* is underspecified. It could be resolved to s_0, to s, or –if *every person* is in the scope of *sometimes*, as in (60b)– also to s'. Consequently, one obtains the readings in (60):

(60) a. $\mathsf{every}(s,\ s \in Dox_m(s_0),$
$\qquad \mathsf{every}(x,\ \mathsf{person}(x, \boxed{13}),\ \mathsf{some}(s', s' \sqsubseteq s, \mathsf{win}(x, s'))))$
b. $\mathsf{every}(s,\ s \in Dox_m(s_0),$
$\qquad \mathsf{some}(s',\ s' \sqsubseteq s,\ \mathsf{every}(x, \mathsf{person}(x, \boxed{13}), \mathsf{win}(x, s'))))$

6. Comparison to other approaches outside LTAG

Besides LTAG, substantial work on quantifier scope has been done in several other grammar formalisms. We will point out some of them and briefly describe how they differ from the approach proposed in this paper.

In the Generative Transformational Grammar framework, long distance dependencies are captured by movement operations like Quantifier Raising (QR) (May, 1977) and *wh*-movement (Chomsky, 1986). These movement operations unambiguously determine the scope of the moved quantificational element. For example, for (61a), the longer movement of the NP *every professor* in (61b) univocally leads to the scope *every > refused*, whereas the shorter movement in (61c) results in the scope relation *refused > every*.[35] More recently, movement has been remodeled using Copy Theory in Chomsky's (1995) Minimalist Program: movement amounts to leaving copies of the "moved" element in several syntactic positions. Only part of those copies is interpreted: the interpreted copy of the determiner gives the scope, while the interpreted copy of NP-part stays in situ. This is illustrated in (61b',c'). This way of syntactically splitting the purely quantificational part from the NP-part in Minimalism is reminiscent of multicomponents in TAG (Joshi and Vijay-Shanker, 1999, Kallmeyer and Joshi, 2003) and of the distinction between MAXS and the rest of the semantics of the quantifier in this paper.

(61) a. John refused to meet every professor
b. [$_S$ [every professor]$_1$ [$_S$ John refused to meet t$_1$]]
c. [$_S$ John refused [$_S$ [every professor]$_1$ to meet t$_1$]]

---

[35] See May (1985), Aoun and Li (1993), and Kitahara (1996) among others for a variant where QR still leads to ambiguity.

b'. [$_S$ every ~~professor~~ [$_S$ John refused to meet ~~every~~ professor]]

c'. [$_S$ John refused [$_S$ every ~~professor~~ to meet ~~every~~ professor]]

Some important differences between Transformational Grammar and the present LTAG proposal are the following. First, it is not clear how to build the desired locality conditions of quantifier dependencies into the definition of QR and islands for QR. It is intuitive to define QR as being able to cross certain boundaries (namely, non-finite clause boundaries, in case (ii)) and not others (finite clause boundaries, in case (i)). But it is not clear why, after being allowed to cross a given boundary (a host NP, in case (iii)), the quantifier should be required to stop rising. That is, if *every city* is allowed to rise outside the host NP and attach to S_2 in (64a), it is not clear why the quantifier is not able to attach to S_1 in (64b).[36]

(64) Two politicians spy on someone from every city.

   a. [$_S$ [two politicians]$_1$ [$_S$ [every city]$_3$ [$_S$ [someone from t_3]$_2$ [$_S$ t_1 spy t_2]]]]

   b. [$_S$ [every city]$_3$ [$_S$ [two politicians]$_1$ [$_S$ [someone from t_3]$_2$ [$_S$ t_1 spy t_2]]]]

In contrast, in the proposed LTAG framework, the maximal scope window is not determined by movement and islands to that movement, but by what MAXS is equated with. If the MAXS of the quantificational NP is identified with the MAXS of the higher predicate Q, this means Q lets NP-scope pass. If the MAXS of NP is equated with something inside the complement of Q, then Q blocks NP-scope. If the MAXS of NP is equated with the label of Q itself, this guarantees that the NP can scope immediately over Q but not any higher. In sum, in our approach, locality conditions on NP-scope correlate with the feature identifications that an embedding operator Q can possibly perform.

Second, despite some resemblance between Copy Theory and our separation between MAXS and the rest, the two approaches differ in

---

[36] In fact, some treatments of inverse linking in the Transformational Grammar framework involve QRing the nested NP not to the S node but to the edge of the host NP, as in (62). But then something else needs to be done for the nested NP *every city* in (63) to be able to scope over the rest of the sentence and bind the pronoun *its*. See May (1985), Heim and Kratzer (1998), Büring (2001) for some possibilities.

(62) [$_S$ ([two politicians]$_1$) [$_{NP}$ [every city]$_3$ [someone from t_3]]$_2$ ([two politicians]$_1$) t_1 spy t_2]

(63) Someone from every city$_i$ hates its$_i$ subway system.

important respects. One difference is that the scope part determines the exact scope in Copy Theory but MAXS gives the maximal possible scope in LTAG. Another difference is that special semantics is needed to interpret split quantificational phrases in Copy Theory. For example, if we interpret the structure (66b) as in (66c), the truth conditions are too weak: any x that is not a professor would make the subformula under "For most x" true, and thus, if there are more non-professors than professors, the entire (66c) becomes automatically true regardless of whether John has seen most professors or not. Special semantics have been developed for the split structures proposed in Copy Theory.[37] In the proposed LTAG approach, instead, the semantic contribution of the NP-part always ends up in the restrictor of the quantifier and never in its nuclear scope, regardless of what happens with MAXS, and thus the interpretive problem in (66) does not arise.

(66) a. John hasn't seen (yet) most professors.
    b. [most$_x$ ~~professors~~ [John hasn't seen ~~most~~ professors$_x$]]
    c. "For most x: $\neg[\mathsf{see}(j,x) \wedge \mathsf{professor}(x)]$"

Third and finally, recall that the scope and the situation variable of a given NP do not need to go together. Once a Ty2 language with situation variables is assumed, nothing predicts the asymmetry between situation variables on the verbal spine and situation variables in NPs if we do Montague-style semantics on the derived tree. As argued in Percus' (2000), a situation binder –e.g. the λ-abstractor under *if* in (51)– will simply look for a situation variable in its sister node –the entire [*every poor child was rich*]– without knowing the internal structure of that sister, that is, without knowing what situation variable belongs to the verbal predicate and which to the NP. In fact, Percus suggests that the appropriate binding conditions for situation variables in Transformational Grammar need to be stated *in the syntax*. In the present LTAG proposal, constraints on situation binding follow from the semantic S features and the architecture of the derivation tree. Situation binders like *think*, *sometimes* or *if* attach to the verb tree

---

[37] Besides Copy Theory, split structures have been proposed because of examples like (65) (Reinhart, 1992), in which the *wh*-phrase *which phantasies about himself* is understood with matrix scope but its restrictor contains a variable bound within the embedded clause. See Reinhart (1992) and Romero (1998) for choice functions for *wh*-phrases, Rullmann and Beck (1998) for a presuppositional treatment of *wh*-phrases, and Fox (1999) for quantificational NPs.

(65) a. Who knows which patient$_i$ has [which phantasies about himself$_i$]?
    b. [who$_x$ which$_y$ ~~phantasies about himself~~ [x knows which patient$_z$ has ~~which~~ phantasies$_y$ about z]]

and hence ask the verb for a situation variable. Together with the constraints on scope, this forces local binding of situations on the verbal spine. The same situation binders do not combine with the NP, and hence cannot ask the NP for a situation to bind. As a result, binding of NP situation variables by VP-spine material is not enforced and, thus, they remain underspecified.[38]

In the Combinatory Categorial Grammar (CCG) framework, Steedman (2003) has proposed an extensive theory of quantifier scope. CCG is a system where lexical entries encode syntactic and semantic types. Larger structures are built by applying different rules, allowing type shifting in the semantics. In contrast to our approach, quantifier scope depends on the order in which quantifiers and certain operations apply, and the interpretations one obtains are not underspecified with respect to scope. Steedman treats only universals *every, each* and *no* as true generalized quantifiers. All other quantificational determiners –e.g. *two, some, at most four*, etc.– are explained as Skolem terms. A Skolem term depends on any variable introduced by a true quantifier scoping over the Skolem term. But, crucially, the Skolem term can be applied at any time in the course of the derivation. I.e., if it is applied before the Skolem term ends up in the scope of some quantifier, then the Skolem term does not depend on that quantifier and therefore appears to have wide scope with respect to it. In a sentence containing only Skolem terms, a scope order effect can only be obtained through the distributivity operator, which introduces true universal quantification. As distributivity follows surface order, the subject is predicted to always have scope over the object in this case, as in (67):

(67) Some linguist knows at most two languages
     some > at most two; * *at most two > some*

While Steedman (2003) makes finer distinctions between quantificational determiners than the ones aimed for in the present paper, we believe that his account may prove too restrictive. By translating indefinites exclusively as Skolem terms and never as generalized quantifiers, he can account for the unique surface scope of (67). But it is not clear whether the model theory he sketches in his appendix can then derive the correct truth conditions for the *every > a* reading of (68a). Note that the truth conditions paraphrased in (68b) do not capture this reading: they yield TRUE even if no student that read a (any) paper by

---

[38] Underspecification of NP situation variables behaves exactly like underspecification of other NP arguments, e.g. *his* in *his cat*. Both can be bound locally, non-locally, or simply remain unbound and default to some salient situation (s_0) or individual. See footnote 9.

Frege passed the class, as long as for each student there is a paper by Frege that that student did not read.[39]

(68) a. Every student that read a paper by Frege passed the class.
b. There is a Skolem function $f_{paper-by-Frege}$ such that:
$$\forall x[\mathsf{student}(x) \wedge \mathsf{read}(x, f^x_{paper-by-Frege}) \rightarrow \mathsf{pass}(x)]$$

Furthermore, in contrast to our approach, Steedman assumes quantifier scope not to be finite clause bound. We have referred the reader to analyses of apparent counterexamples to finite clause boundedness in footnote 25, p. 31.[40]

Other computational semantics approaches that show some similarities with our framework are approaches within *Head-Driven Phrase Structure Grammar (HPSG)* (Pollard and Sag, 1994), namely *Minimal Recursion Semantics (MRS)* (Copestake et al., 1999) and more recently *Lexical Resource Semantics* (Richter and Sailer, 2004). Of course, HPSG differs crucially from TAG: The grammar is a set of constraints that describe parts (finite subtrees) of the whole structure (in TAG terms, the derived tree). This means that there is no extended locality in the sense of TAG that allows to localize long dependencies. The underlying structures are typed feature structures that can be thought of as phrase structure trees with feature structures labelling the nodes. These features structures allow structure sharing and reentrancy.

For semantics, Copestake et al. (1997) use flat semantic representations enriched with propositional meta-variables and scope constraints. However, the way they deal with quantifier scope is different from our proposal and it is not clear how they obtain the various constraints for quantifier scope mentioned in this paper. The approach by Richter and Sailer (2004) has a lot in common with our approach, despite the different formalism (see also Kallmeyer and Richter, 2006 for a comparison). Of course, since they do not have an extended domain of locality, they compute semantics on the derived tree, i.e., the constituency structure, using general semantic principles to obtain scope constraints. But they also use a Ty2 language with underspecification and, particularly, they have features EXCONT and INCONT that resemble very much to our MAXS and MINS: EXCONT constitutes an upper boundary for the scope

---

[39] This problem was noted in Chierchia's (2001) for certain choice function approaches to indefinites.

[40] Steedman (2003) also derives interesting scope constraints on coordinate structures like (69). We refer the reader to Babko-Malaya (2004) for an LTAG analysis of coordination within the semantic framework in the present paper.

(69) Every boy admires and every girl detests some saxophonist.

of a quantifier. However, it still needs to be investigated in which cases EXCONT might be blocked and in which cases it might be passed up. INCONT corresponds to the lowest proposition and it constitutes the minimal nuclear scope of attaching quantifiers. Both EXCONT and IN-CONT are passed up the constituency tree along the head projection line. We take the striking similarity between EXCONT and INCONT on the one hand and our MAXS and MINS on the other hand as an additional evidence for the adequacy of our proposal.

7. Conclusion

In this paper we have introduced an LTAG semantics framework based on the derivation tree. Besides underspecified semantic representations, semantic feature structure descriptions are assigned to elementary trees. As the semantic composition operation, we use conjunction of the feature descriptions plus additional feature value equations. These additional equations are parallel to the unifications performed for the syntactic features on the derived tree that are used in TAG.

Roughly, the semantic feature structure descriptions serve two purposes. On the one hand, they encode the contributions of the semantic representations to functional applications. I.e., they state which elments are contributed as possible arguments for other semantic expressions and which arguments need to be filled. They thereby simulate lambda abstraction and functional application. On the other hand, they also serve to model the scopal behaviour of different operators and to capture the different boundaries for scope.

Within this framework, we have proposed an account of the differences between quantificational NPs on the one hand and quantificational elements attached to the verbal spine (adverbs, raising verbs, attitude verbs, etc.) on the other hand with respect to their scope possibilities and also with respect to the way their situation variables are bound.

The main aspects of our analysis are the following: The scope of quantificational NPs is limited by an upper and a lower boundary constituting a kind of scope window. These boundaries are global features, i.e., they depend on elementary trees but not on single nodes in elementary trees. Within this window, the quantifier can scope freely. The binding of the situation variable of an NP does not depend on syntax and therefore remains unspecified during the syntactic derivation; in the disambiguation process, either it becomes a bound variable or it defaults to s_0. In contrast to this, quantificational elements on the verbal spine take scope where they attach making use of a local feature

specifying the proposition corresponding to the tree below a specific node in the constituent structure. Furthermore, their situation binder also depends on a local feature of the node they attach to, since their binder has to be the next higher binder on the verbal spine (if no binder is found, the situation defaults to s_0 in the disambiguation).

The combination of LTAG's extended domain of locality with a semantics using feature structure descriptions enables us to capture even apparently non-local scope relations within a mildly context-sensitive framework: The structures underlying the computation of syntax and semantics are the context-free derivation trees.

References

Abeillé, A.: 2002, *Une grammaire électronique du français*. Paris: CNRS Editions.

Aoun, J. and Y. A. Li: 1993, *Syntax of Scope*. MIT Press.

Artstein, R.: 2003, 'Quantificational arguments in temporal adjuncts clauses'. Ms. Technion University.

Babko-Malaya, O.: 2004, 'LTAG Semantics of NP-Coordination'. In: *Proceedings of TAG+7*. Vancouver.

Bech, S. and H. Rullmann: 1999, 'A Flexible Approach to Exhaustivity in Questions'. *Natural Language Semantics* **7**(3), 249–297.

Berman, S.: 1991, 'On the Semantics and Logical Form of Wh-Clauses'. Ph.D. thesis, University of Massachusetts at Amherst.

Blackburn, P. and E. Spaan: 1993, 'A Modal Perspective on the Computational Complexity of Attribute Value Grammar'. *Journal of Logic, Language and Information* **2**, 129–169.

Bos, J.: 1995, 'Predicate Logic Unplugged'. In: P. Dekker and M. Stokhof (eds.): *Proceedings of the 10th Amsterdam Colloquium*. pp. 133–142.

Bouma, G., R. Malouf, and I. Sag: 1998, 'Adjunct Scope'. Presented at the Workshop *Models of Underspecification and the Representation of Meaning*, 18–22 May 1998, Bad Teinach.

Büring, D.: 2001, 'A Situation Semantics for Binding out of DP'. In: R. Hastings, B. Jackson, and Z. Zvolenski (eds.): *Proceedings from Semantics and Linguistic Theory XI*. Ithaca, pp. 56–75, CLC.

Candito, M.-H. and S. Kahane: 1998, 'Can the TAG Derivation Tree represent a Semantic Graph? An Answer in the Light of Meaning-Text Theory'. In: *Fourth International Workshop on Tree Adjoining Grammars and Related Frameworks, IRCS Report 98–12*. University of Pennsylvania, Philadelphia, pp. 25–28.

Chierchia, G.: 1993, 'Questions with quantifiers'. *Natural Language Semantics* **1**, 181–234.

Chierchia, G.: 2001, 'A puzzle about indefinites'. In: C. Cecchetto, G. Chierchia, and M. Guasti (eds.): *Semantic Interfaces: reference, anaphora and aspect*. Stanford: CSLI.

Chomsky, N.: 1986, *Barriers*. Cambridge, Mass.: MIT Press.

Chomsky, N.: 1995, *The Minimalist Program*. MIT Press.

Cinque, G.: 1999, *Adverbs and functional heads : a cross-linguistic perspective*. NY: Oxford University Press.

Copestake, A., D. Flickinger, I. A. Sag, and C. Pollard: 1999, 'Minimal Recursion Semantics. An Introduction'. Manuscript, Stanford University.

Cresswell, J. M.: 1990, *Entities and Indices*. Dordrecht: Kluwer.

Dras, M., D. Chiang, and W. Schuler: 2004, 'On Relations of Constituency and Dependency Grammars'. *Journal of Language and Computation* **2**(2), 281–305.

Farkas, D.: 1997, 'Evaluation Indices and Scope'. In: A. Szabolsci (ed.): *Ways of scope taking*. Dordrecht: Kluwer.

Forbes-Riley, K., B. Webber, and A. K. Joshi: To appear, 'Computing Discourse Semantics: The Predicate-Argument Semantics of Discourse Connectives in D-LTAG'. *Journal of Semantics*.

Fox, D.: 1999, *Economy and Semantic Interpretation*. MIT Press.

Fox, D. and U. Sauerland: 1996, 'Illusive Scope of Universal Quantifiers'. In: *Proceedings of NELS 26*. Amherst.

Frank, A. and J. van Genabith: 2001, 'GlueTag. Linear Logic based Semantics for LTAG – and what it teaches us about LFG and LTAG'. In: M. Butt and T. H. King (eds.): *Proceedings of the LFG01 Conference*. Hong Kong.

Frank, R.: 1992, 'Syntactic Locality and Tree Adjoining Grammar: Grammatical, Acquisition and Processing Perspectives'. Ph.D. thesis, University of Pennsylvania.

Fuchss, R., A. Koller, J. Niehren, and S. Thater: 2004, 'Minimal Recursion Semantics as Dominance Constraints: Translation, Evaluation, and Analysis'. In: *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04)*. Barcelona, Spain.

Gallin, D.: 1975, *Intensional and Higher-Order Modal Logic with Applications to Montague Semantics*, North Holland mathematics studies 19. Amsterdam: North-Holland Publ. Co.

Gamut, L.T.F.: 1991, *Logic Language and Meaning*. University of Chicago Press.

Gardent, C. and L. Kallmeyer: 2003, 'Semantic Construction in FTAG'. In: *Proceedings of EACL 2003*. Budapest.

Ginzburg, J. and I. A. Sag: 2000, *Interrogative Investigations. The Form, Meaning, and Use of English Interrogatives*, No. 123 in CSLI Lecture Notes. CSLI.

Heim, I. and A. Kratzer: 1998, *Semantics in Generative Grammar*. Blackwell.

Hobbs, J. R. and S. M. Shieber: 1987, 'An Algorithm for Generating Quantifier Scopings'. *Computational Linguistics* **13**, 47–63.

Johnson, M.: 1988, *Attribute-value logic and the theory of grammar*, CSLI Lecture Notes Series. Chicago: University of Chicago Press.

Johnson, M.: 1990, 'Expressing disjunctive and negative feature constraints with classical first-order logic'. In: *Proceedings of ACL*. pp. 173–179.

Johnson, M.: 1994, 'Computing with Features and Formulae'. *Computational Linguistics* **20**(1), 1–25.

Joshi, A. K.: 1987, 'An introduction to Tree Adjoining Grammars'. In: A. Manaster-Ramer (ed.): *Mathematics of Language*. Amsterdam: John Benjamins, pp. 87–114.

Joshi, A. K., L. Kallmeyer, and M. Romero: 2003, 'Flexible Composition in LTAG: Quantifier Scope and Inverse Linking'. In: H. Bunt, I. van der Sluis, and R. Morante (eds.): *Proceedings of the Fifth International Workshop on Computational Semantics IWCS-5*. Tilburg, pp. 179–194.

Joshi, A. K. and Y. Schabes: 1997, 'Tree-Adjoning Grammars'. In: G. Rozenberg and A. Salomaa (eds.): *Handbook of Formal Languages*. Berlin: Springer, pp. 69–123.

Joshi, A. K. and K. Vijay-Shanker: 1999, 'Compositional Semantics with Lexicalized Tree-Adjoining Grammar (LTAG): How Much Underspecification is Necessary?'. In: H. C. Blunt and E. G. C. Thijsse (eds.): *Proceedings of the Third International Workshop on Computational Semantics (IWCS-3)*. Tilburg, pp. 131–145.

Kahane, S.: 2005, 'Structure des représentations logiques, polarisation et sous-spécification'. In: *Proceedings of TALN 2005*. Dourdan, France.

Kahane, S., M.-H. Candito, and Y. de Kercadio: 2000, 'An alternative description of extraction in TAG'. In: *Proceedings of TAG+5*. Paris, pp. 115–122.

Kallmeyer, L.: 2002a, 'Enriching the TAG derivation tree for semantics'. In: *Proceedings of KONVENS 2002*. Saarbrücken, pp. 67 – 74.

Kallmeyer, L.: 2002b, 'Using an Enriched TAG Derivation Structure as Basis for Semantics'. In: *Proceedings of TAG+6 Workshop*. Venice, pp. 127 – 136.

Kallmeyer, L.: 2003, 'LTAG Semantics for Relative Clauses'. In: H. Bunt, I. van der Sluis, and R. Morante (eds.): *Proceedings of the Fifth International Workshop on Computational Semantics IWCS-5*. Tilburg, pp. 195–210.

Kallmeyer, L. and A. K. Joshi: 2003, 'Factoring Predicate Argument and Scope Semantics: Underspecified Semantics with LTAG'. *Research on Language and Computation* **1**(1–2), 3–58.

Kallmeyer, L. and F. Richter: 2006, 'Constraint-based Computational Semantics: A Comparison between LTAG and LRS'. In: *Proceedings of The Eighth International Workshop on Tree Adjoining Grammar and Related Formalisms (TAG+8)*. Sydney, Australia.

Kallmeyer, L. and M. Romero: 2004, 'LTAG Semantics with Semantic Unification'. In: *Proceedings of TAG+7*. Vancouver.

Kallmeyer, L. and T. Scheffler: 2004, 'LTAG Analysis for Pied-Piping and Stranding of wh-Phrases'. In: *Proceedings of TAG+7*. Vancouver.

Karttunen, L.: 1973, 'Presupositions of Compound Sentences'. *Linguistic Inquiry* (4), 167–193.

Karttunen, L.: 1977, 'The syntax and semantics of questions'. *Linguistics and Philosophy* **1**, 3–44.

Kitahara, H.: 1996, 'Raising Quantifiers without Quantifier Raising'. In: W. A. et al. (ed.): *Minimal Ideas*. John Benjamins, pp. 189–198.

Koller, A., J. Niehren, and S. Thater: 2003, 'Bridging the Gap Between Underspecification Formalisms: Hole Semantics as Dominance Constraints'. In: *Meeting of the European Chapter of the Association of Computational Linguistics*. pp. 195–202.

Koller, A., J. Niehren, and R. Treinen: 1998, 'Dominance Constraints: Algorithms and Complexity'. In: *Proceedings of the Third International Conference on Logical Aspects of Computational Linguistics (LACL)*. Grenoble, France.

Kratzer, A.: 1989, 'An Investigation of the Lumps of Thought'. *Linguistics and Philosophy* **12**.

Kratzer, A.: 1999, 'Beyond 'ouch' and 'oops'. How descriptive and expressive meaning interact'. Ms. `http://semanticsarchive.net`.

Kroch, A.: 1989, 'Asymmetries in long-distance extraction in a Tree Adjoining Grammar'. In: Baltin and Kroch (eds.): *Alternative Conceptions of Phrase Structure*. University of Chicago.

Kroch, A. S.: 1987, 'Unbounded dependencies and subjacency in a Tree Adjoining Grammar'. In: A. Manaster-Ramer (ed.): *Mathematics of Language*. Amsterdam: John Benjamins, pp. 143–172.

Lahiri, U.: 1991, 'Embedded Interrogatives and Predicates That Embed Them'. Ph.D. thesis, MIT.

Larson, R.: 1987, 'Quantifying into NP'. Ms. MIT.

May, R.: 1977, 'The Grammar of Quantification'. Ph.D. thesis, MIT.

May, R.: 1985, *Logical Form. Its Structure and Derivation*, Linguistic Inquiry Monographs. Cambridge, Massachusetts / London, England: MIT Press.

Percus, O.: 2000, 'Constraints on some other variables in syntax'. *Natural Language Semantics* **8**, 173–229.

Pollard, C. and I. A. Sag: 1994, *Head-Driven Phrase Structure Grammar*, Studies in Contemporary Linguistics. Chicago, London: The University of Chicago Press.

Potts, C.: 2002, 'The syntax and semantics of 'as' parentheticals'. *Natural Language and Linguistic Theory* **20**, 623–689.

Rambow, O. and G. Satta: 1992, 'Formal Properties of Non-Locality'. In: *Proceedings of the TAG+ Workshop*. Philadelphia.

Rambow, O., K. Vijay-Shanker, and D. Weir: 1995, 'D-Tree Grammars'. In: *Proceedings of ACL*.

Reinhart, T.: 1992, 'Wh-in-situ: An apparent paradox'. In: P. D. et al. (ed.): *Proceedings of the Eight Amsterdam Colloquium*.

Richter, F. and M. Sailer: 2004, 'Basic Concepts of Lexical Resource Semantics'. To appear in the series of the Kurt Gödel Sociey, Vienna.

Rogers, J.: 2002, 'One More Perspective on Semantic Relations in TAG'. In: *Proceedings of TAG+6 Workshop*. Venice, pp. 118 – 126.

Romero, M.: 1998, 'Focus and reconstruction effects in wh-phrases'. Ph.D. thesis, UMass.

Romero, M.: 2002, 'Quantification over situations variables in LTAG: some constraints'. In: *Proceedings of TAG+6*. Venice, Italy.

Romero, M., L. Kallmeyer, and O. Babko-Malaya: 2004, 'LTAG Semantics for Questions'. In: *Proceedings of TAG+7*. Vancouver.

van Rooy, R.: 2003, 'Questioning to Resolve Decision Problems'. *Linguistics and Philosophy*.

Rullmann, H. and S. Beck: 1998, 'Presupposition Projection and the Interpretation of Which-Questions'. In: D. Strolovitch and A. Lawson (eds.): *Proceedings from Semantics and Linguistic Theory VIII (SALT VIII)*. pp. 215–232, CLC Publications.

Sauerland, U.: 2000, 'Syntactic Economy and Quantifier Raising'. Ms. University of Tübingen.

Schabes, Y. and S. M. Shieber: 1994, 'An Alternative Conception of Tree-Adjoining Derivation'. *Computational Linguistics* **20**(1), 91–124.

Shieber, S. M.: 1994, 'Restricting the weak-generative capacity of synchronous Tree-Adjoining Grammars'. *Computational Intelligence* **10**(4), 271–385.

Shieber, S. M. and Y. Schabes: 1990, 'Synchronous Tree-Adjoining Grammars'. In: *Proceedings of COLING*. pp. 253–258.

Steedman, M.: 2003, 'Scope Alternation and the Syntax-Semantics Interface'. Draft 4.1, University of Edinburgh.

Stone, M. and C. Doran: 1997, 'Sentence Planning as Description Using Tree-Adjoining Grammar'. In: *Proceedings of ACL*. pp. 192–205.

Vijay-Shanker, K.: 1987, 'A Study of Tree Adjoining Grammars'. Ph.D. thesis, University of Pennsylvania.

Vijay-Shanker, K. and A. K. Joshi: 1985, 'Some computational properties of Tree Adjoining Grammars'. In: *Proceedings of ACL*.

Vijay-Shanker, K. and A. K. Joshi: 1988, 'Feature Structures Based Tree Adjoining Grammar'. In: *Proceedings of COLING*. Budapest, pp. 714–719.

von Fintel, K. and S. Iatridou: 2003, 'Epistemic containment'. *Linguistic Inquiry* (34), 173–198.

Weir, D. J.: 1988, 'Characterizing mildly context-sensitive grammar formalisms'. Ph.D. thesis, University of Pennsylvania.

XTAG Research Group: 1998, 'A Lexicalized Tree Adjoining Grammar for English'. Technical Report 98–18, Institute for Research in Cognitive Science, Philadelphia.