Scope and Situation Binding in LTAG using Semantic Unification

Maribel Romero

romero@ling.upenn.edu

Laura Kallmeyer

lkallmey@linguist.jussieu.fr

Abstract

This paper develops a framework for TAG (Tree Adjoining Grammar) semantics that brings together ideas from different recent approaches. Then, within this framework, an analysis of scope is proposed that accounts for the different scopal properties of quantifiers, adverbs, raising verbs and attitude verbs. Finally, including situation variables in the semantics, different situation binding possibilities are derived for different types of quantificational elements.

1 Introduction

Quantificational Noun Phrases (NPs) differ from quantificational elements attached to the verbal spine (adverbs, raising verbs, attitudes verbs, etc.) in their scope possibilities. E.g., (1), with two quantificational NPs, is scopally ambiguous between the readings (1a) and (1b) (May (1985)). (2), in contrast, has only the surface reading (2a) and lacks the inverse reading (2b) (Cinque (1999)).[1] 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 (3) is entirely attributed to

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

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

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

Natural language predicates –nouns, adjectives, verbs, adverbs– are evaluated with respect to a given situation (a spatio-temporal "slice" of a world). When we spell out their situation variable, another difference arises between

---

[1] Note that in contrast to (2), (4) has both readings, (2a) and (2b). But (4) is a case of syntactic ambiguity, where two LTAG derivations yield the two scope orders respectively.

(4) John seems to laugh sometimes

NPs and elements attached to the verbal spine. The situation variable of NPs can be non-locally bound by a distant operator, whereas the situation variable of material on the verbal spine must be locally bound by the closest c-commanding situation binder (Percus (2000); Romero (2002)). To see this, consider sentence (5), a situation s consisting of yesterday's entire game and several subsituations s' corresponding to different rounds of that game.

(5) Mary thinks that in yesterday's game John sometimes beat the winner
$$\lambda s_0.\forall s[s \in Dox_m(s_0) \wedge \text{yest-game}(s) \rightarrow$$
$$\exists s'[s' \sqsubseteq s \wedge \text{beat}(j, \iota x[\text{winner}(x, s_0/s/\#s')], s')]]^2$$

(5) can be understood as having *the winner* translated as $\iota x[\text{winner}(x, s_0)]$ ('Of the actual overall winner x, Mary thinks that John beat x in some rounds') or as $\iota x[\text{winner}(x, s)]$ ('Mary thinks that in some rounds John beat whoever the overall winner was'), skipping over the local situation binder $\exists s'$. (The translation $\iota x[\text{winner}(x, s')]$ is in principle possible as well, but having s' for *the winner* and for *beat* is obviously pragmatically deviant.) In contrast, the situation variable of the verb *beat* must be locally bound by the $\exists s'$-quantifier of *sometimes*, yielding the subformula $\text{beat}(j, \iota x[...], s')$; if it could be distantly bound and represented as $\text{beat}(j, \iota x[...], s)$, the sentence (5) would be predicted to have the reading 'Mary thinks John beat the (/ a) sometimes winner', contrary to fact. Furthermore, the second situation within *sometimes* must be locally bound by the $\forall s$-quantification of *think*, yielding $\exists s'[s' \sqsubseteq s]$; if it could be distantly bound and represented as $\exists s'[s' \sqsubseteq s_0]$, the sentence could mean 'John sometimes beat the (/ a) person that Mary thinks is (/ may be) the winner', again contrary to fact.

Hence, the scope and situation of material on the verbal spine is determined by the immediate c-commanding intensional operator in the surface syntax. The scope and situation of an NP, instead, is not limited in this way. The question then arises whether NPs are free to scope anywhere and whether their situation variables can be bound at any distance. The answer is 'no' for scope and 'yes' for situation binding. On the one hand, the scope of an NP is generally assumed to be limited to the first finite clause containing the NP. E.g., inverse scope is possible across the non-finite clause in (6) but not across the finite clause in (7), nor can a quantifier scope outside the *if*-clause in (8).[3] On the other hand, the situation of *the winner* in (5) can

---

[2] $Dox_m(s_0)$ 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.

[3] *Every professor* can have a de re reading over *said* in (7), but this is achieved using acquaintance relations rather than truly scoping mechanisms (Cresswell & von Stechow (1982)). What needs to be explained in terms of scope is the contrast in (6)-(7) for

be bound by λs_0 outside of its finite clause, and the situation of *most poor children* in (8) can be bound by λs_0 outside the *if*-clause (Farkas (1993)).

(6) a student wants to meet every professor $\exists > \forall, \quad \forall > \exists$

(7) a student said you met every professor $\exists > \forall, \quad * \forall > \exists$

(8) if most poor children were rich instead, this would be a better world
if $>$ most, * most $>$ if
$\lambda s_0. \forall s\ [s$ is-accessible-from $s_0\ \wedge\ most\ x[\mathsf{poor\text{-}children}(x, s_0), \mathsf{rich}(x, s)]\ \rightarrow$
$\mathsf{better}(\mathsf{this.world}, s)]$

In this paper, we present a general framework for LTAG semantics that allows us to compute semantic representations on the derivation tree.[4] Then, within this framework we propose a semantic analysis of quantifier scope accounting for the rigid scope of material on the verbal spine (adverbs, raising verbs, attitude verbs, etc.), and the underspecified scope of NPs within the first finite clause. Finally, we extend the formal machinery to the situation data, but leaving NP situation binding completely underspecified.

2 LTAG

Lexicalized Tree Adjoining Grammars (LTAG, Joshi & 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.

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.

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

---

the reading *every* $>$ *a*. Other potential counterexamples to finite clause boundness are analyzed as cases of illusive scope in Fox & Sauerland (1996) and Artstein (2003).

[4] This framework has been originally developed in Kallmeyer & Romero (2004).
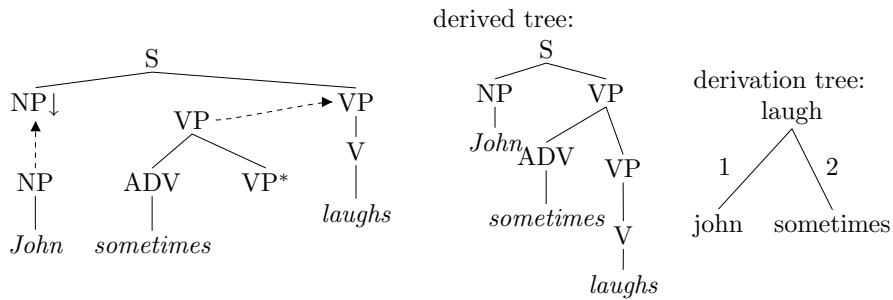
Figure 1: TAG derivation for (9)

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.[5] E.g., see the derivation of (9) 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.

(9) John sometimes laughs

3 LTAG Semantics

3.1 Semantic feature structures

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. This has been done for example in Kallmeyer & Joshi (2003). However, 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 missing links between lexical items that are needed for semantics can be obtained using some semantic feature unification. The idea is as follows: Each elementary tree in the derivation tree is linked to a semantic representation plus a semantic feature structure. Feature unification then provides the correct assignments for the variables in the semantic representations.[6]

---

[5]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 $p \cdot j$.

[6]For a detailed discussion of this approach and a comparison to other LTAG semantics frameworks such as Gardent & Kallmeyer (2003), see Kallmeyer & Romero (2004).

Semantic representations are as in Kallmeyer & Joshi (2003) except that they do not have argument variables: they consist of a set of formulas (typed λ-expressions with labels) and a set of scope constraints. A scope constraint is an expression $x \geq y$ where x and y are propositional labels or propositional variables (these last correspond to the holes in Kallmeyer & Joshi (2003)).[7] Each semantic representation is linked to a semantic feature structure.

Semantic feature structures are typed feature structures, the type of the whole feature structure is *sem*. A feature structure of type *sem* consists of features 0 (the root position), 1, 2, ..., 11, 12, ... for all node positions that can occur in elementary trees (finite for each TAG), the values of these features are of type *tb* (for 'top-bottom'). A feature structure of type *tb* consists of a T and a B feature (top and bottom) whose values are feature structures of type *bindings*. A feature structure of type *bindings* consists of a feature I whose values are individual variables, a feature P whose values are propositional labels, and a feature S whose values are situation variables.

3.2   Semantic unification

Semantic composition consists only of feature unification: in the derivation tree, elementary trees are replaced by their semantic representations plus the corresponding semantic feature 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 feature structures $\gamma_1.p$.T and $\gamma_2.0$.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 feature structures $\gamma_1.p$.B and $\gamma_2.f$.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.

By these unifications, some of the variables in the semantic representations get values. After having performed the unifications, the union of all semantic representations is built yielding an underspecified representation.

As an example consider the analysis of (9): Fig. 2 shows the semantic representations and the semantic feature structures of the three elementary trees involved. The different unifications lead to the identifications of *laugh*.1.T with *john*.0.T (substitution), of *laugh*.2.T with *sometimes*.0.T and *laugh*.2.B with *sometimes*.2.B (adjunction), and of *john*.0.T with *john*.0.B, *sometimes*.0.T with *sometimes*.0.B and *sometimes*.2.T with *sometimes*.2.B

$$
\begin{bmatrix} 0 & \mathrm{B} & \begin{bmatrix} \mathrm{I} & x \end{bmatrix} \end{bmatrix}
\qquad
l_1 : \mathsf{laugh}(\boxed{1}, \boxed{2})
\qquad
\begin{bmatrix}
1 & \begin{bmatrix} \mathrm{T} & \begin{bmatrix} \mathrm{I} & \boxed{1} \end{bmatrix} \end{bmatrix} \\[1em]
2 & \begin{bmatrix}
\mathrm{T} & \begin{bmatrix} \mathrm{P} & \boxed{4} \\ \mathrm{S} & \boxed{3} \end{bmatrix} \\[1em]
\mathrm{B} & \begin{bmatrix} \mathrm{P} & l_1 \\ \mathrm{S} & \boxed{2} \end{bmatrix}
\end{bmatrix}
\end{bmatrix}
\qquad
\begin{bmatrix}
0 & \mathrm{B} & \begin{bmatrix} \mathrm{P} & l_2 \\ \mathrm{S} & \boxed{5} \end{bmatrix} \\[1em]
2 & \mathrm{T} & \begin{bmatrix} \mathrm{P} & \boxed{7} \\ \mathrm{S} & s \end{bmatrix}
\end{bmatrix}
$$

Tree edges labelled **1** and **2**, with leaf nodes $\mathsf{john}(x)$ and $l_2 : \mathsf{some}(s, s \sqsubseteq \boxed{5}, \boxed{6}),\ \boxed{6} \geq \boxed{7}$.
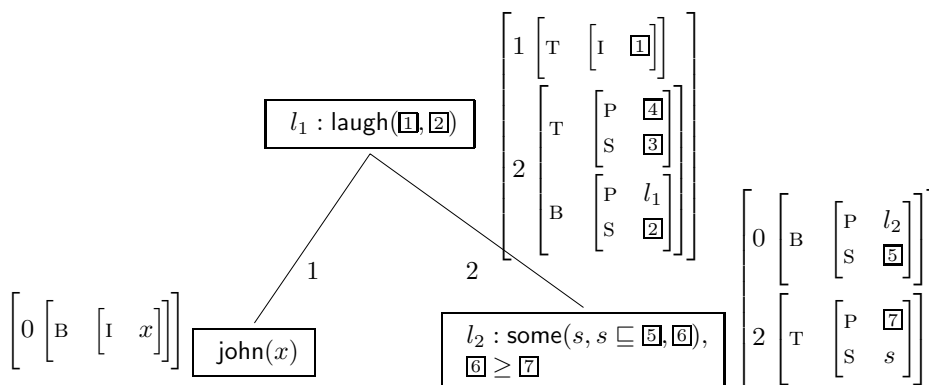
Figure 2: Semantic representations for (9)

(final top and bottom unification). This gives the identities $\boxed{1} = x$, $\boxed{4} = l_2$, $\boxed{3} = \boxed{5}$, $\boxed{7} = l_1$ and $\boxed{2} = s$, which results in the semantic representation (10):

(10) $\quad \boxed{l_1 : \mathsf{laugh}(x, s), \mathsf{john}(x), l_2 : \mathsf{some}(s, s \sqsubseteq \boxed{5}, \boxed{6}), \boxed{6} \geq l_1}$

Finally, appropriate disambiguations must be found, i.e., functions that assign propositional labels to the remaining propositional variables, respecting the scope constraints in (Kallmeyer & Joshi (2003)), and that assign situations to situation variables. The disambiguated representation is interpreted conjunctively. (10) has one disambiguation, $\boxed{6} \to l_1$, $\boxed{5} \to s_0$ (s_0 is by default the actual situation), leading to $\mathsf{john}(x) \wedge \mathsf{some}(s, s \sqsubseteq s_0, \mathsf{laugh}(x, s))$.

3.3 Computational aspects

The computation of the underspecified semantic representation via feature unification on the derivation tree uses the same mechanisms as ordinary LTAG parsing in a feature-structure based TAG (FTAG, Vijay-Shanker & Joshi (1988)). 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. 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.

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 $x \leq y$ is NP-complete. But the semantic

representations we actually use are such that they can be formulated as so-called *normal dominance constraints* (Koller et al. (2003)). For this type of constraints an efficient polynomial solver has been developed.

4 Scope semantics in LTAG[8]

4.1 Quantifiers

(11) everybody laughs

To account for the flexible scope of NPs, we follow previous LTAG semantics approaches (Joshi & Vijay-Shanker (1999); Kallmeyer & Joshi (2003)) and assume that quantifiers as *everybody* have a multicomponent set containing an auxiliary tree that contributes the scope part and an initial tree that contributes the predicate argument part. When adding such a quantifier, both trees must be added simultaneously. E.g., consider the derivation for (11) in Fig. 3. The two trees need not be added to the same elementary tree, i.e., the derivation is not tree-local. But the way we use non-local multicomponent composition does not increase the generative power because we allow non-local adjunction only for scope trees (trees with one node: S*).[9]

Concerning quantifier scope, 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. The first constraint must result from the combination of the lower part of the quantifier (the NP tree) and the tree to which it attaches.[10] We introduce a feature MINP ('minimal proposition') to pass the proposition of a tree to a quantifier NP tree. The second constraint must result from the adjunction of the scope part of the quantifier. We use a feature MAXP ('maximal proposition', including a finite verb plus all its quantifiers, adverbs etc. but excluding the next higher finite verb) that passes an upper limit for scope from a verb tree to an adjoining scope tree. E.g., see Fig. 3 for the analysis of (11). It leads to the unifications: $\boxed{6} = \boxed{2}$ (adjunction of scope part), $\boxed{1} = x$ and $\boxed{7} = l_1$ (substitution of predicate-argument part), and $\boxed{3} = l_1$ (final top-bottom unification). The result is (12) which has one disambiguation: $\boxed{2} \to l_2, \boxed{4} \to l_3, \boxed{5} \to l_1$.[11]

---

[8]We omit situation variables in this section, to be reintroduced in section 5.

[9]Furthermore, we allow only derivations that can be viewed as local adopting flexible composition as in Joshi et al. (2003).

[10]This is particularly clear in examples with quantificational NPs that are embedded in other quantificational NPs, as considered in Joshi et al. (2003).

Figure 3: Analysis of (11)

$$(12) \quad \boxed{\begin{array}{l} l_1 : \mathsf{laugh}(x),\ l_2 : \mathsf{every}(x, \boxed{4}, \boxed{5}),\ l_3 : \mathsf{person}(x) \\ \boxed{2} \geq l_1, \boxed{2} \geq l_2, \boxed{4} \geq l_3, \boxed{5} \geq l_1 \end{array}}$$

4.2 Attitude verbs

The analysis of attitude verbs such as *thinks* in (14) is shown in Fig. 4. The propositional argument of *think* (variable $\boxed{3}$) is the MAXP value of the embedded verb. The adjunction leads to the unification $\boxed{3} = \boxed{1}$.

(14) Mary thinks John laughs
(15) Mary thinks John likes everybody

In (15) wide scope of *everybody* should be disallowed. If its scope part attaches to *likes*, then the scope is blocked by the MAXP of *likes*. Consequently, *everybody* cannot have scope over *thinks* because *thinks* takes the MAXP proposition of *likes* as its argument. However, we have to make sure that the scope part of *everybody* cannot attach higher, i.e., to *thinks*.

---

[11]For ambiguous examples as (13a) we obtain underspecified representations. The semantics in (13b) has two disambiguations, wide scope of *some* and *every* respectively.

(13) a. someone likes everybody

b. $\boxed{\begin{array}{l} l_1 : \mathsf{like}(x, y),\ l_2 : \mathsf{some}(x, \boxed{4}, \boxed{5}),\ l_3 : \mathsf{person}(x), \\ l_4 : \mathsf{every}(y, \boxed{8}, \boxed{9}),\ l_5 : \mathsf{person}(y) \\ \boxed{1} \geq l_1,\ \boxed{1} \geq l_2,\ \boxed{4} \geq l_3,\ \boxed{5} \geq l_1,\ \boxed{1} \geq l_4,\ \boxed{8} \geq l_5,\ \boxed{9} \geq l_1 \end{array}}$
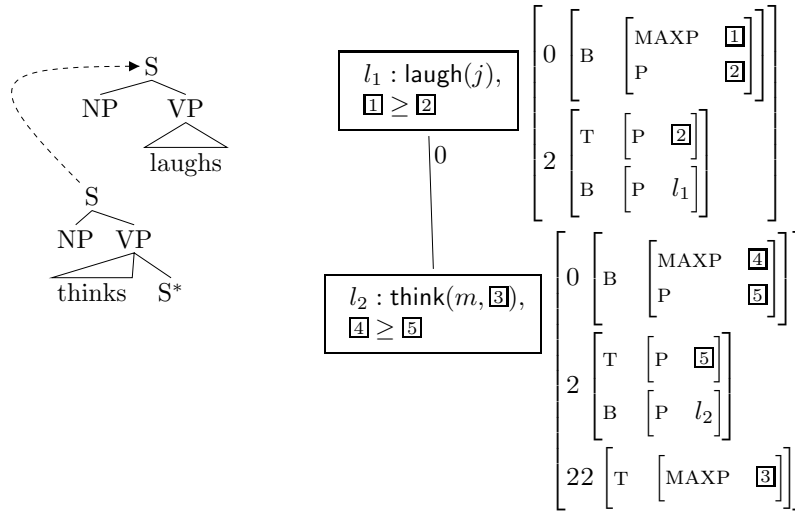
8

Figure 4: Analysis of (14)

In general, we allow scope parts to adjoin higher. But, following Joshi et al. (2003), the compositions must be such that one or more already derived trees or tree sets attach (by substitution or adjunction) to one single elementary tree. If only the NP tree of *everybody* attaches to *like*, a correct derivation for (15) is not possible: Either the scope part of *everybody* adjoins to *thinks*. Then the derived *like* tree also must be added to *thinks* since it is part of the same derived tree set, i.e., *thinks* would have a substitution node instead of a foot node. This however is problematic for the analysis of long-distance dependencies in LTAG. Or *thinks* attaches to *like* simulaneously with the lower *everybody* part. But then the scope part has to find some other node than the S node of *thinks* in order to attach to it. There is no other S node besides those coming from *thinks* or *like*, so this possibility does not work either. Consequently, one has to adjoin the scope part to *like*.

4.3 Raising verbs and adverbs

We analyze raising verbs similar to adverbs (see *sometimes* in Fig. 2). They are inserted between the top and bottom P values of the node to which they adjoin, scoping over the lower proposition. A second adverb/raising verb adjoined to the root of a first one has then 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 which is below the MAXP proposition. I.e., in (16), the attitude verb *claim* scopes over the adverb/raising verb. (17) (and similar, (3)) however allows narrow and wide scope of the NP, since the scope of the NP is independent from the adverb/raising verb.
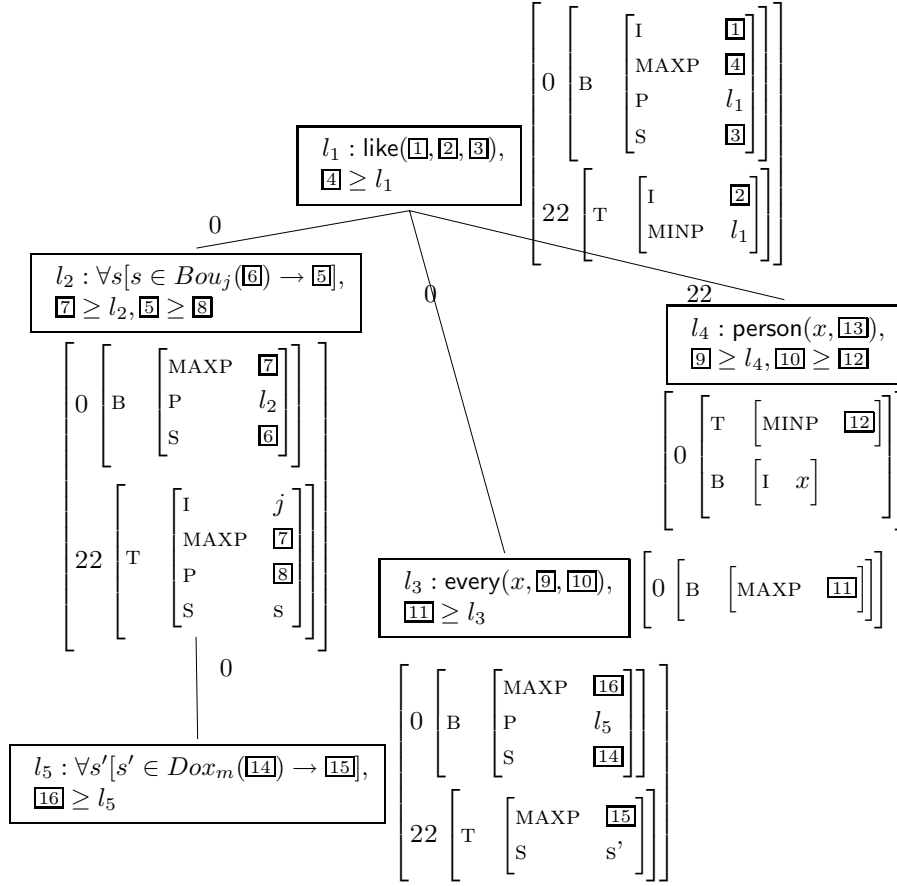
Figure 5: Analysis of (18) *Mary thinks John tries to like everybody*

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

(17) someone apparently laughs

4.4 Control verbs

Syntactically, control verbs like *try* resemble attitude verbs, i.e., their elementary trees look similar. But control verbs do not block scope. Therefore, in our analysis in Fig. 5, *try* (label l_2) embeds the P value of the verb it attaches to under its argument (giving ⑧ $= l_1$), similar to raising verbs. *Try* further passes up the MAXP value of the embedded verb (④ $=$ ⑦), and it passes the controller (j) to the embedded verb (①).

10

5 Situation semantics in LTAG

The same feature unifications that guarantee fix scope for elements on the verbal spine and underspecified scope for NPs can derive the situation binding data as well. On the one hand, material attached to the verbal spine (see *sometimes* in Fig. 2) will unify its situation variables univocally with the situation variables at the attachment positions. On the other, 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.

(18) a. Mary thinks John tries to like everybody

b.
$$
\begin{array}{|l|}
\hline
l_1 : \mathsf{like}(j, x, s),\ l_2 : \forall s[s \in Bou_j(s') \to \boxed{5}], \\
l_3 : \mathsf{every}(x, \boxed{9}, \boxed{10}),\ l_4 : \mathsf{person}(x, \boxed{13}),\ l_5 : \forall s'[s' \in Dox_m(\boxed{14}) \to \boxed{4}] \\
\boxed{4} \geq l_1,\ \boxed{4} \geq l_2,\ \boxed{5} \geq l_1,\ \boxed{4} \geq l_3,\ \boxed{9} \geq l_4,\ \boxed{10} \geq l_1,\ \boxed{16} \geq l_5 \\
\hline
\end{array}
$$

This is illustrated in (18a) where an attitude verb, a control verb and an embedded quantifier interact for scope and for situation binding. The (abridged) derivation in Fig. 5 is the only possible derivation, i.e., the scope part of *everybody* must adjoin to *like*. One obtains $\boxed{1} = j, \boxed{3} = s,\ \boxed{7} = \boxed{4}$, $\boxed{8} = l_1,\ \boxed{11} = \boxed{4},\ \boxed{12} = l_1, \boxed{2} = x,\ \boxed{15} = \boxed{7} = \boxed{4}$, and $\boxed{6} = s'$, which gives (18b). In (18b), both the maximal scope of the quantifier and the maximal scope of the control verb are blocked by the argument of *thinks* ($\boxed{4} \geq l_3$ and $\boxed{4} \geq l_2$) but the order of the quantifier and the control verb is unspecified. The situation $\boxed{14}$ will default to s_0 while $\boxed{13}$ is underspecified. One obtains the scope orders *thinks > every > try > like* and *thinks > try > every > like*.

6 Conclusion

In this paper we used an LTAG semantics framework based on the derivation tree with feature structure unification on the derivation tree as semantic composition operation. Within this framework, we proposed an account of the differences between quantificational NPs on the one hand and elements attached to the verbal spine (adverbs, raising verbs, attitude verbs, etc.) on the other hand, both with respect to their scope possibilities and with respect to the way their situation variables are bound.

References

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

Bos, J.: 1995. Predicate logic unplugged. *in* P. Dekker and M. Stokhof (eds), *Proceedings of the 10th Amsterdam Colloquium*. pp. 133–142.

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

Cresswell, M. and von Stechow, A.: 1982. De Re Belief Generalized. *Linguistics and Philosophy* **5**, 503–535.

Farkas, D.: 1993. Modal Anchoring and NP Scope. LRC. UC Santa Cruz.

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

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

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

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

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

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

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

Koller, A., Niehren, J. and Thater, S.: 2003. Bridging the gap between underspecification formalisms: Hole semantics as dominance constraints. *Meeting of the European Chapter of the Association of Computational Linguistics*.

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

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

Reyle, U.: 1993. Dealing with ambiguities by underspecification: Construction, representation and deduction. *Journal of Semantics* **10**, 123–179.

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

Vijay-Shanker, K. & Joshi, A. K.: 1988. Feature structures based tree adjoining grammar. *Proceedings of COLING*. Budapest. pp. 714–719.