LTAG Semantics for Questions

Maribel Romero
Department of Linguistics
610, Williams Hall
University of Pennsylvania
Philadelphia, PA, 19104-6305
U.S.A.
romero@ling.upenn.edu

Laura Kallmeyer
TALaNa-Lattice
UFRL, University Paris 7
2 place Jussieu, Case 7003,
75251 Paris Cedex 05
France
lkallmey@linguist.jussieu.fr

Olga Babko-Malaya
CIS Department
3330 Walnut St.
University of Pennsylvania
Philadelphia, PA 19104-6389
U.S.A.
malayao@linc.cis.upenn.edu

Abstract

This papers presents a compositional semantic analysis of interrogatives clauses in LTAG (Lexicalized Tree Adjoining Grammar) that captures the scopal properties of wh- and non-wh-quantificational elements. It is shown that the present approach derives the correct semantics for examples claimed to be problematic for LTAG semantic approaches based on the derivation tree. The paper further provides an LTAG semantics for embedded interrogatives.

1 Introduction.

Following (Karttunen, 1977), an interrogative clause Q expresses a function from possible situations (or worlds) to the set of true answers to that question Q in that situation. For example, the interrogative clause (1) has the meaning (2), where *who* contributes the ∃-quantification $\exists x[\mathsf{person}(x, s_0)]$. In a situation s_0 where Pat, Al, Kate and nobody else called, $[\![\mathtt{Q}(s_0)]\!]$ equals the set (3).

(1) who called?

(2) $\lambda s_0 \lambda p.p(s_0) \wedge$
$\exists x[\mathsf{person}(x, s_0) \wedge p = \lambda s.\mathsf{call}(x, s)]$

(3) $\{\ \lambda s.\mathsf{call}(\mathsf{pat}, s),\ \lambda s.\mathsf{call}(\mathsf{al}, s),\ \lambda s.\mathsf{call}(\mathsf{kate}, s)\ \}$

The aim of this paper is to develop a compositional semantic analysis of interrogative clauses in LTAG, with two goals: **(i)** the main goal is to capture the scopal properties of quantificational elements within the question, and **(ii)** the secondary goal is to achieve the correct semantics for interrogatives embedded under e.g. *know*.

The scope data concerning goal **(i)** are the following. When an interrogative clause contains a *wh*-element and a non-*wh* quantificational element, as in (4), the semantic contribution of *who* must be outside the proposition headed by λs, whereas the semantic contribution of *everybody* must be inside that proposition, as shown in (5).[1]

(4) (John knows) who likes everybody

(5) (John knows) $\lambda s_0 \lambda p.p(s_0)$
$\wedge\exists x[\mathsf{person}(x, s_0) \wedge p = \lambda s.\forall y[\mathsf{person}(y, s)$
$\rightarrow \mathsf{like}(x, y, s)]]$

Note that, when we have more than one wh-phrase and more than one non-wh-quantifier, the non-wh-quantifiers can yield difference scope configurations among themselves (and so can the wh-phrases among themselves, trivially). But all the wh-phrases must take scope above the λs proposition and all the non-wh-quantifiers must take scope below it. This is illustrated in (6), which has the readings (7)-(8), but not e.g. the readings (9)-(10).

(6) (John knows) who seemed to introduce who to everybody

(7) (John knows) $\lambda s_0 \lambda p.p(s_0)$
$\wedge\exists x\exists y[\mathsf{person}(x, s_0) \wedge \mathsf{person}(y, s_0)$
$\wedge\, p = \lambda s.\mathsf{seem}(\lambda s'.\forall z[\mathsf{person}(z, s')$
$\rightarrow \mathsf{introduce}(x, y, z, s')], s)]$

(8) (John knows) $\lambda s_0 \lambda p.p(s_0)$
$\wedge\exists x\exists y[\mathsf{person}(x, s_0) \wedge \mathsf{person}(y, s_0)$
$\wedge\ `p = \lambda s.\forall z[\mathsf{person}(z, s)$
$\rightarrow \mathsf{seem}(\lambda s'.\mathsf{introduce}(x, y, z, s')], s)]$

(9) (John knows) $\lambda s_0 \lambda p.p(s_0)$
$\wedge\exists x[\mathsf{person}(x, s_0) \wedge p = \lambda s.\exists y[\mathsf{person}(y, s)$
$\wedge\ \mathsf{seem}(\lambda s'.\forall z[\mathsf{person}(z, s')$
$\rightarrow \mathsf{introduce}(x, y, z, s')], s)]]$

(10) (John knows) $\lambda s_0 \lambda p.p(s_0)$
$\wedge\exists x\exists y[\mathsf{person}(x, s_0) \wedge \mathsf{person}(y, s_0)$
$\wedge\forall z[\mathsf{person}(z, s_0)$
$\rightarrow p = \lambda s.\mathsf{seem}(\lambda s'.\mathsf{introduce}(x, y, z, s')], s)]$

[1] We leave aside the so-called pair-list readings arising when *everybody* c-commands the trace of the *wh*-phrase and a special absorption operation takes place (Chierchia, 1993).

TAG+7: Seventh International Workshop on Tree Adjoining Grammar and Related Formalisms.
May 20-22, 2004, Vancouver, BC, CA.
Pages 186-193.

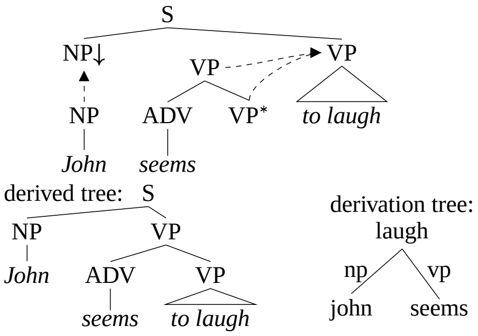

Figure 1: Derivation for (13)

With respect to goal **(ii)**, we need to construct a question meaning that will be able to combine with a question taking verb like *know*. In the end, a sentence like (11) must receive the truth-conditions in (12). The expression $Dox_j(s_0)$ in (12) stands for the set of doxastic alternatives of John in s_0, that is, for the set of possible situations s' that conform to John's beliefs in s_0. The formula (12) states that we are in a situation s_0 such that, for all of John's belief alternatives s' in s_0 and for all propositions p, $p \in [\![\text{who called}]\!](s')$ iff $p \in [\![\text{who called}]\!](s_0)$.

(11) John knows who called.

(12) $\lambda s_0.\forall s'_s \in Dox_j(s_0)\forall p_{<s,t>}$
$[\exists x[\mathsf{person}(x, s') \wedge p(s') \wedge p = \lambda s.\mathsf{call}(x, s)]$
$\leftrightarrow \exists x[\mathsf{person}(x, s_0) \wedge p(s_0) \wedge p = \lambda s.\mathsf{call}(x, s)]]$

2 Semantic unification

For LTAG semantics, we use the semantic unification framework described in (Kallmeyer and Romero, 2004) that is very close to (Gardent and Kallmeyer, 2003): We do compositional semantics on the derivation tree, i.e., each elementary tree has a semantic representation and the derivation tree indicates how to do semantic computation. Semantic representations are equipped with semantic feature structures. Semantic representations are sets of formulas (typed λ-expressions with labels) and scope constraints. A scope constraint is an expression $x \geq y$ where x and y are propositional labels or propositional variables. Semantic feature structures have features P for all node positions p that can occur in elementary trees.[2] The values of these features are feature structure that consist of a T and a B feature (top and bottom) whose values are feature structures with features I for individual variables, P for propositional labels and S for situations.

Semantic composition consists of unification: In the derivation tree, elementary trees are replaced by their semantic representations and their semantic feature structures. Then, for each edge from γ_1 to γ_2 with position p: 1. the T feature of position p in γ_1 and the T feature of the root of γ_2 are identified, and 2. if γ_2 is an auxiliary tree, then the B feature of the foot node of γ_2 and the B feature of position p in γ_1 are identified. Furthermore, for all γ occurring in the derivation tree and 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. Then, the union of all semantic representations is built which yields an underspecified representation. Finally, appropriate disambiguations must be found, i.e., assignments for the remaining propositional variables that respect the scope constraints in the sense of (Kallmeyer and Joshi, 2003). The disambiguated representations are interpreted conjunctively. As an example, Fig. 1 and 2 show the derivation and the semantics for (13).

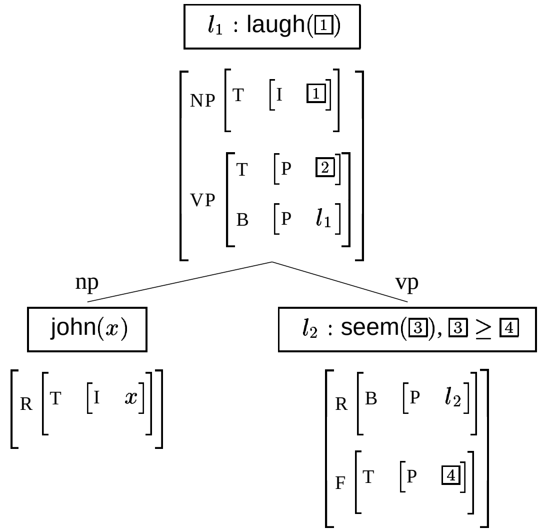

Figure 2: Semantics for (13)

(13) John seems to laugh

The feature identities because of unification are $\boxed{1} = x$, $\boxed{2} = l_2$, $\boxed{4} = l_1$which leads to (14). There is only one disambiguation, $\boxed{3} \rightarrow l_1$ which yields the semantics $\mathsf{john}(x) \wedge \mathsf{seem}(\mathsf{laugh}(x))$.[3]

(14) $\boxed{\begin{array}{l} l_1 : \mathsf{laugh}(x), \mathsf{john}(x), l_2 : \mathsf{seem}(\boxed{3}), \\ \boxed{3} \geq l_1 \end{array}}$

3 Scopal properties of wh-phrases

3.1 Quantificational NPs

Following previous approaches ((Kallmeyer and Joshi, 2003; Joshi et al., 2003) and also (Kallmeyer and Romero, 2004)), we assume that quantifiers as *everybody*

[2] For the sake of readability, we use names np, vp, ... for the node positions instead of the usual Gorn adresses.

[3] For simplification, in (14) situation variables are omitted.

in (15) have a multicomponent set containing an auxiliary tree that contributes the scope part and an initial tree that contributes the predicate argument part. Fig. 3 illustrates this approach.

(15) everybody laughs

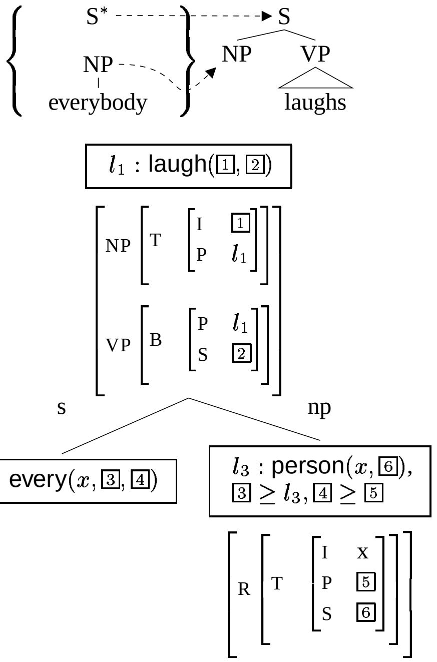

Figure 3: Analysis of (15)

The analysis in Fig. 3 leads to the feature identities $\boxed{1} = x, \boxed{5} = l_1$. As a result one obtains (16). There is one disambiguation, $\boxed{3} \to l_3, \boxed{4} \to l_1$, that yields the semantics $\mathsf{every}(x, \mathsf{person}(x, \boxed{6}), \mathsf{laugh}(x, \boxed{2}))$.

(16) $$\boxed{\begin{array}{l} l_1 : \mathsf{laugh}(x, \boxed{2}), l_2 : \mathsf{every}(x, \boxed{3}, \boxed{4}), \\ l_3 : \mathsf{person}(x, \boxed{6}), \boxed{3} \geq l_3, \boxed{4} \geq l_1 \end{array}}$$

Following (Percus, 2000), situation variables in verbs must be locally bound, and situation variables in NPs can be non-locally bound by any situation binder in the sentence (e.g. by *know* in (4)). In the current example (15), the situation variable $\boxed{2}$ in the verb *laugh* and the situation variable $\boxed{6}$ in *everybody* will default to s_0 (the situation of the whole proposition), since there is no situation binder in the formula. This yields the final semantics $\mathsf{every}(x, \mathsf{person}(x, s_0), \mathsf{laugh}(x, s_0))$.

3.2 Wh-phrases as quantifiers

Consider again example (4) *who likes everybody?* and its Karttunen-style semantics in (5), repeated as (17) below. To achieve this result in LTAG, we propose the derivation and the semantics in Fig. 4. The crucial ingredients are as follows.

(17) (John knows) $\lambda s_0 \lambda p.p(s_0)$
$\wedge \exists x[\mathsf{person}(x, s_0) \wedge p = \lambda s.\forall y[\mathsf{person}(y, s/s_0)$
$\to \mathsf{like}(x, y, s)]]$

The semantic representation for the interrogative elementary tree of *like* must include all the semantic information in (5) except for $\exists x[\mathsf{person}(x, s_0)]$ –coming from *who*– and $\forall y[\mathsf{person}(y, s/s_0)]$ –coming from *everybody*. Since the wh- and non-wh-quantificational elements must have scope over different portions of the formula, the semantic representation of the interrogative tree for *like* is split into several separate subformulae, each with its own label and with constraints guaranteeing the correct scopal configuration among them. First, it contains the formula $l_1 : \mathsf{like}(\boxed{1}, \boxed{2}, \boxed{3})$, shared by all the family trees for *like*. Second, it contributes the formula $l_2 : p = \lambda s.\boxed{7}$, which will take scope over l_1, given that $\boxed{7} = \boxed{4}$ and that $\boxed{6} = l_1$ (by identification of T and B features in positions S and VP respectively) and given the scope constraint $\boxed{4} \geq \boxed{6}$. Finally, the interrogative tree for *like* contributes the expression $q_3 : \lambda p.\boxed{5}$, with scope over l_2 due to the scope constraint $\boxed{5} \geq l_2$. (Note that q_3 is not a propositional formula and hence cannot be interpreted as conjoined with the rest. See section 4 and footnote 6 on this issue.)

What we need to achieve with respect to scope is that all quantificational NPs take scope under $\boxed{7}$ and over l_1, and that all wh-phrases take scope under $\boxed{5}$ and over l_2 We propose a multi-component analysis of *wh*-phrases parallel to that of quantificational NPs, with the only difference that the scope part of a wh-quantifier adjoins to S' whereas the scope part of a non-wh-quantifier adjoins to S, as shown in Fig. 3. This parallel treatment is appropriate since the scope of wh-quantifiers is not strictly related to their surface positions, e.g., in situ wh phrases can take wide scope. We then define a "scope window" for wh- and non-wh-quantificational NPs by using two semantic features linked to the two parts of the multi-component: MAXS is linked to the S* or S'* part and gives the upper limit of the scope window, and P is linked to the NP-part and determines the lower limit of the scope window. In the case of *everybody* in Fig. 4, the value of MAXS is $\boxed{13}$, then $\boxed{13} = \boxed{4}$ (by adjunction to S in *like* tree), and finally $\boxed{4} = \boxed{7}$ (by T/B unification in S of *likes*). The value of *everybody*'s lower limit P is $\boxed{16}$, and $\boxed{16} = l_1$ (by substitution into position NP in *like* tree). This gives us the desired result $\boxed{7} \geq l_6 \geq l_1$, where l_6 introduces the $\forall$-quantification corresponding to *everybody*.[4] The case of *who* is parallel. Its MAXS feature, in the S'* part, has the value $\boxed{8}$, and $\boxed{8} = \boxed{5}$ (by adjunction to S'). Its lower limit feature P, in the NP part, has the value $\boxed{11}$, and $\boxed{11} = l_2$ (by substituion into position WH of *like* tree). This yields the desired scope $\boxed{5} \geq l_4 \geq l_2$, where l_4 corresponds to

[4] See also (Kallmeyer and Romero, 2004) for further motivation of the MAXS feature for quantifiers.

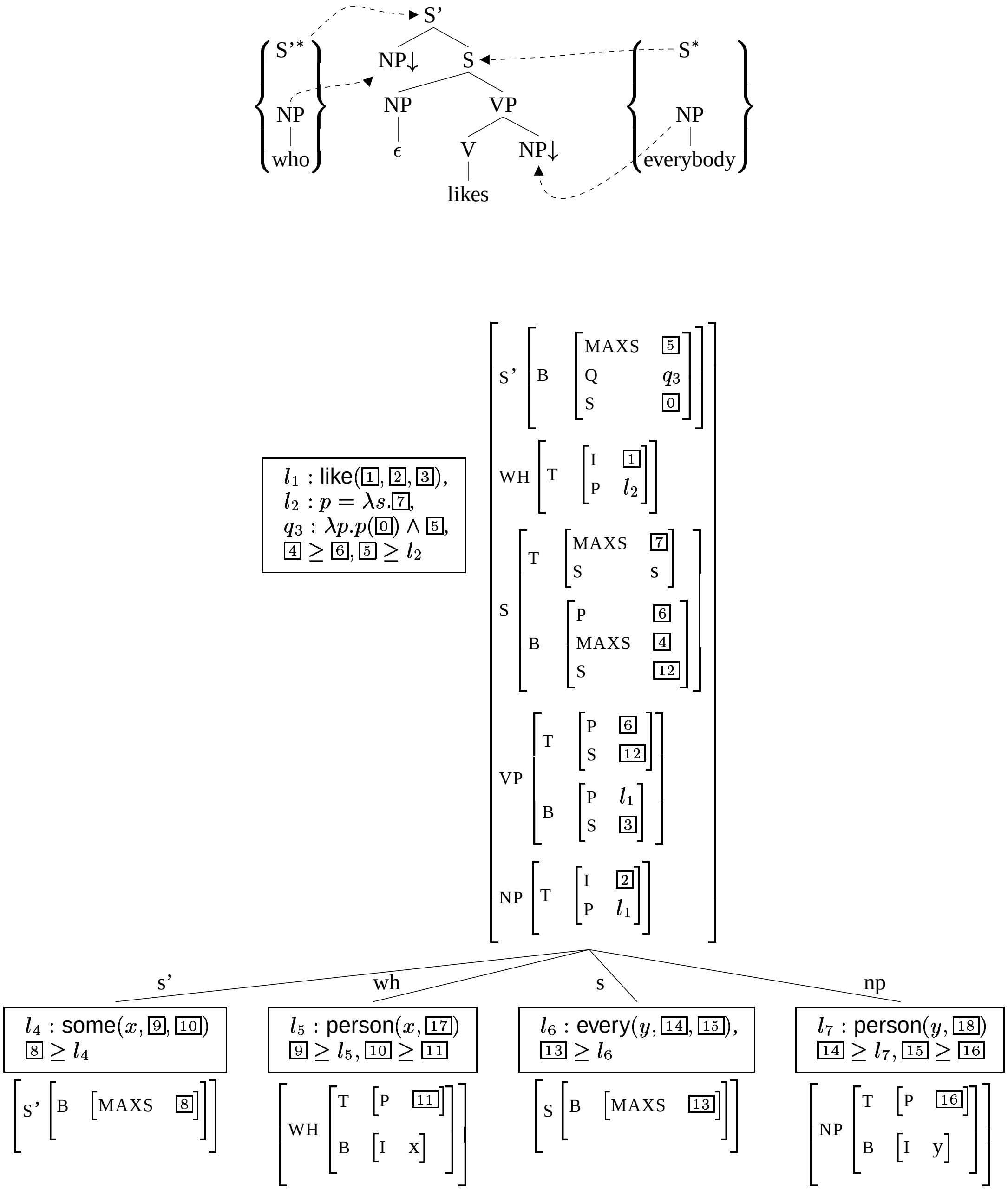

Figure 4: Derivation and derivation tree with semantics for (4) *who likes everybody*

the ∃-quantification of *who*. Hence, by defining an upper limit feature MAXS and a lower limit feature P for wh- and non-wh-quantifiers, we can obtain the right scopal configurations.

The semantic representation one obtains for (4) is (18):

$$(18)\quad \boxed{\begin{array}{l} l_1 : \mathsf{like}(x,y,s), \\ l_2 : p = \lambda s.\boxed{7}, \quad q_3 : \lambda p.p(\boxed{0}) \wedge \boxed{5}, \\ l_4 : \mathsf{some}(x,\boxed{9},\boxed{10}), l_5 : \mathsf{person}(x,\boxed{17}), \\ l_6 : \mathsf{every}(y,\boxed{14},\boxed{15}), l_7 : \mathsf{person}(y,\boxed{18}), \\ \boxed{7} \geq l_1, \boxed{5} \geq l_2 \\ \boxed{5} \geq l_4, \boxed{9} \geq l_5, \boxed{10} \geq l_2 \\ \boxed{7} \geq l_6, \boxed{14} \geq l_7, \boxed{15} \geq l_1 \end{array}}$$

As intended, (18) allows only one disambiguation, namely $\boxed{5} \rightarrow l_4$, $\boxed{9} \rightarrow l_5$, $\boxed{10} \rightarrow l_2$, $\boxed{7} \rightarrow l_6$, $\boxed{14} \rightarrow l_7$, $\boxed{15} \rightarrow l_1$. The situation indices $\boxed{0}$ and $\boxed{17}$ default to s_0 and the value of $\boxed{18}$ remains underspecified (it could be s_0 or s). This leads to $q_3 : \lambda p.p(s_0) \wedge \mathsf{some}(x, \mathsf{person}(x, s_0), p = \lambda s.\mathsf{every}(y, \mathsf{person}(y, s/s_0), \mathsf{like}(x, y, s)))$.

3.3 Multiple wh-questions

A more complex example is (6) *who seemed to introduce who to everybody*, where two wh-quantifiers (one of them in situ) interact with a raising verb and a non-wh-quantifier. In order to treat in situ wh-quantifiers correctly, it must be possible to obtain the minimal scope of wh-quantifiers from any NP substitution node. Therefore, in NP substitution nodes we have to provide both, the minimal scope of wh-quantifiers and the minimal scope of non-wh-quantifiers. In the case of *like* in Fig. 4 for example, the minimal scope of *who* is l_2 while the minimal scope of *everybody* is l_1. We will use the feature WP for the first and the feature P for the second. For example, at the object substitution node in the tree for *introduce* in Fig. 5 we put a P value (as before) and additionally a WP value in case a wh-quantifier is added.

The derivation of (6) *who seemed to introduce who to everybody* and its semantic analysis are shown in Fig. 5. The raising verb in (6) adjoins to the VP node. This means that its label l_8 will become the value of the top P feature $\boxed{6}$ of the VP node, which is below the MAXS feature $\boxed{4}$ for non-wh-quantifiers (see the constraint $\boxed{4} \geq \boxed{6}$ in the semantics of *introduce* in Fig. 5). The scope trees of the wh-quantifiers adjoin both to the S' node, i.e., their scopes are limited by the MAXS value $\boxed{5}$ of the root. And, because of the WP features, both wh-quantifiers take scope over the proposition l_2 containing $\boxed{7}$, equated in turn with the non-wh MAXS value $\boxed{4}$ ($\boxed{7}$= $\boxed{4}$ by T/B unification in S of *introduce*). Consequently, we obtain the following scope orders: the two wh-quantifiers have both scope over *seem* and *everybody*, but the scope order of the raising verb and the non-wh-quantifier is unspecified.

3.4 Long-distance wh-dependencies

In long-distance wh-dependencies as (19) one also wants to obtain an interpretation where the wh-quantifier takes scope over all verbs in the sentence while providing the argument of the most embedded verb. Such examples have always been claimed to be problematic for derivation tree based LTAG semantics approaches (see (Kallmeyer and Romero, 2004) and the literature cited there).

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

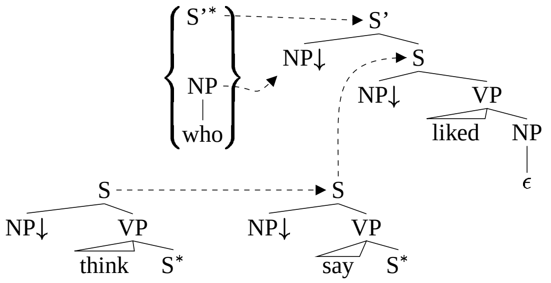

Figure 6: Derivation of (19)

The syntactic analysis of (19) (see (Kroch, 1987)) is shown in Fig. 6, and the combination of *like*, *say* and *think* in the semantics is shown in Fig. 7. Each of the attitude verbs takes the bottom MAXS proposition of the S node as its argument and it gives a larger proposition with a new (higher) bottom MAXS value. In the end, the highest of these MAXS values is unified with the top MAXS of the S node (i.e., with $\boxed{7}$). Therefore, all attitude verbs are embedded under the top MAXS value of the S node of *like* which is in the scope of any wh-quantifier added to *like*. In this way the correct scope analyses for wh-quantifiers in long-distance dependencies are obtained. The initial NP tree of such a quantifier is of course as before substituted for the corresponding argument position in *like* which leads to the correct predicate argument dependencies.

3.5 Comparison with other approaches to the scope of wh-phrases

The Karttunen-style semantic tradition ((Lahiri, 1991), (Chierchia, 1993), among many others), within the Montagovian Formal Semantics framework, draws the distinction between wh-scope and non-wh-scope by basing the semantics on the derived tree and using different semantic types for the relevant nodes. The S node has the propositional type <s,t>, and the semantics of non-wh-quantificational elements operates on functions of that type. The S' node (or, more specifically, the C' node) has the type <s,<<s,t>,t>>> corresponding to functions from situations to sets of propositions, and wh-quantifiers must combine with functions of such type. This derives

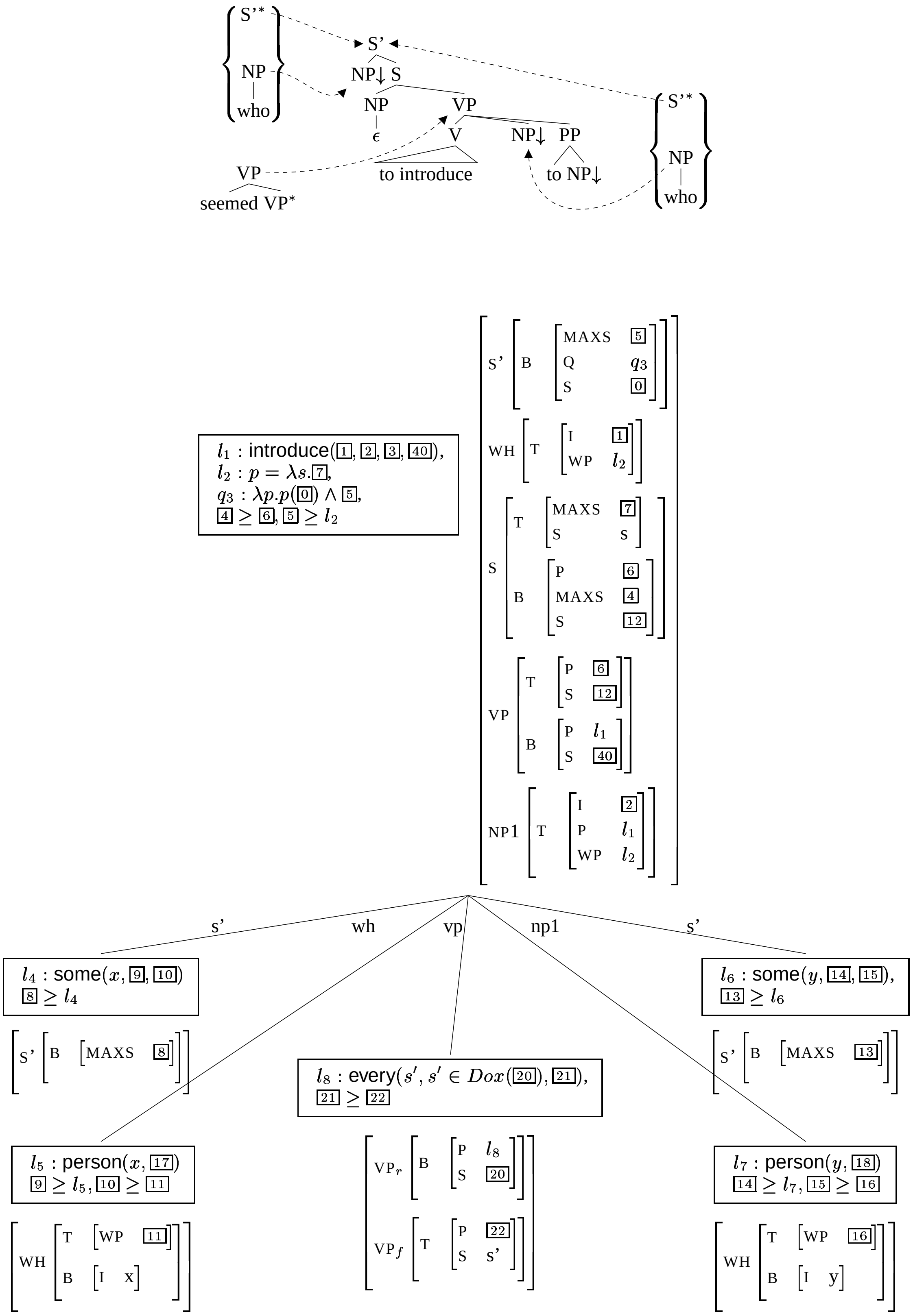

Figure 5: Abriged derivation and derivation tree with semantics (without quantifier *everybody*) for (6) *who seemed to introduce who to everybody*

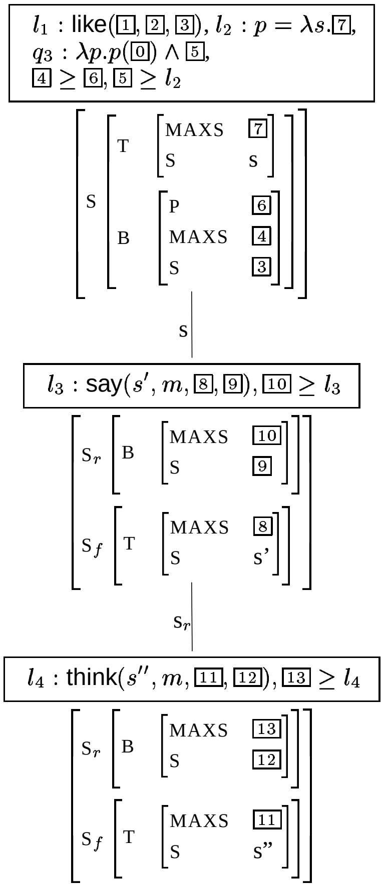
l_1 : like($\boxed{1}$, $\boxed{2}$, $\boxed{3}$), $l_2 : p = \lambda s.\boxed{7}$,
$q_3 : \lambda p.p(\boxed{0}) \wedge \boxed{5}$,
$\boxed{4} \geq \boxed{6}, \boxed{5} \geq l_2$
S
T
MAXS $\boxed{7}$
S s
B
P $\boxed{6}$
MAXS $\boxed{4}$
S $\boxed{3}$
s
l_3 : say(s', m, $\boxed{8}$, $\boxed{9}$), $\boxed{10} \geq l_3$
S_r
B
MAXS $\boxed{10}$
S $\boxed{9}$
S_f
T
MAXS $\boxed{8}$
S s'
s_r
l_4 : think(s'', m, $\boxed{11}$, $\boxed{12}$), $\boxed{13} \geq l_4$
S_r
B
MAXS $\boxed{13}$
S $\boxed{12}$
S_f
T
MAXS $\boxed{11}$
S s"

Figure 7: Abriged derivation tree and semantics for (19)

the effect that all wh-quantifiers must scope over all the non-wh-quantifiers.

A comparable approach using semantic features is developed in (Ginzburg and Sag, 2000), who make an ontological distinction between states-of-affairs (SOAs) and propositions. A verb introduces a SOA, which is the original building block from which later one builds propositions, questions, outcomes and facts. The idea is that a non-wh-quantifier has a SOA as its nuclear scope, and a wh-phrase has a proposition as its nuclear scope. Hence, wh-phrases necessarily have wider scope than non-wh-quantifiers in their clause.

The present approach provides an account of the scopal properties of wh- and non-wh-quantifiers within a 'flat' semantics framework in the style of MRS (Copestake et al., 1999) without invoking finer ontological distinctions. The semantic contribution of each elementary and auxiliary tree is a set of formulae (type t, the extensional version of propositions). Such a flat approach simplifies the design of algorithms for semantic computation as explained in (Copestake et al., 1999). Since the semantic material that will end up in the nuclear scope of a wh- and non-wh-quantifier is invariably introduced as a formula, no type distinction can be made to which the scopal properties of wh- and non-wh-quantifiers could relate. Furthermore, no ontological distinction between state-of-affairs and propositions is used to make scope follow from selectional properties. Instead, the present account proposes to define appropriate scope windows using the features MAXS, P and WP and feature unification.[5]

4 Embedded interrogatives

We have seen that the elementary tree for verbs includes formulae with situation arguments, e.g. l_1 : laugh($\boxed{1}$, $\boxed{2}$) in Fig. 3 and l_1 : introduce($\boxed{1}$, $\boxed{2}$, $\boxed{3}$, $\boxed{40}$) in Fig. 5. When no operator binds that variable, it defaults to the utterance situation s_0, as we saw for $\boxed{2}$ in *laugh* in (16), section 3.1. Otherwise, the situation variable must be bound by some operator, using feature unification: e.g., $\boxed{40}$ is bound by the $\forall s'$-quantifiction introduced by *seems* in Fig. 5 ($\boxed{40} = s'$ by adjunction of *seems* to VP).

In the case of $q_3 : \lambda p.p(\boxed{0}) \wedge \boxed{5}$ in an interrogative verb tree, we also have a situation variable $\boxed{0}$ that, if unbound, will default to s_0, as noted for (18). The issue is how this situation variable becomes bound when the interrogative clause is embedded under, e.g., *know*. Note that, in the final semantics for *John knows who called* in (12), repeated as (20) below, the semantic contribution of the embedded interrogative has to be used twice, once evaluated for the doxastic situation s' and once for the utterance situation s_0. But, if we take $q_3 : \lambda p.p(\boxed{0}) \wedge \boxed{5}$ in any of the derivations above and we simply perform feature unification to the extent that $\boxed{0} = s'$, q_3 will invariably amount to $\lambda p.p(s') \wedge \boxed{5}$ all the times it is used. The question is, thus, how to achieve the effect that $\boxed{0}$ is replaced by s' in one occurrence of the formula and by s_0 in another.

(20) $\lambda s_0.\forall s'_s \in Dox_j(s_0)\forall p_{<s,t>}$
$[\exists x[\text{person}(x, s') \wedge p(s') \wedge p = \lambda s.\text{call}(x, s)]$
$\leftrightarrow \exists x[\text{person}(x, s_0) \wedge p(s_0) \wedge p = \lambda s.\text{call}(x, s)]]$

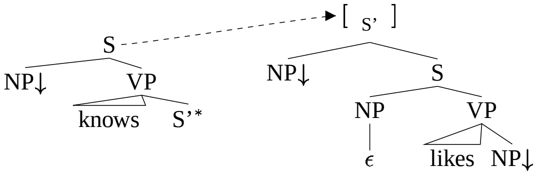

Figure 8: Derivation of (4)

Our analysis of (4) *John knows who likes everybody* is given in Fig. 8 and Fig. 9. To obtain the desired effect, we propose that the semantics of the verb tree for *know* includes a $\lambda s''$ that will bind $\boxed{0}$ in both occurrences of

[5] A third approach treats wh-phrases, along with indefinites, as open formulae whose variable is bound by an unselective binder (Berman, 1991). As we treat indefinites as contributing their own quantificational force, we do the same for wh-phrases.

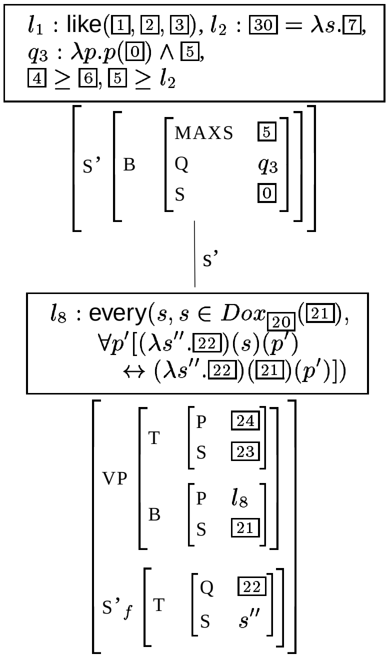
l_1 : like($\boxed{1}$, $\boxed{2}$, $\boxed{3}$), l_2 : $\boxed{30} = \lambda s.\boxed{7}$,
$q_3 : \lambda p.p(\boxed{0}) \wedge \boxed{5}$,
$\boxed{4} \geq \boxed{6}$, $\boxed{5} \geq l_2$
S' B MAXS $\boxed{5}$ Q q_3 S $\boxed{0}$
s'
l_8 : every($s, s \in Dox_{\boxed{20}}(\boxed{21})$,
$\forall p'[(\lambda s''.\boxed{22})(s)(p')$
$\leftrightarrow (\lambda s''.\boxed{22})(\boxed{21})(p')])$
VP T P $\boxed{24}$ S $\boxed{23}$
B P l_8 S $\boxed{21}$
S'$_f$ T Q $\boxed{22}$ S s''

Figure 9: Abriged derivation tree and semantics for (4)

q_3. This is achieved by adding the situation feature S $\boxed{0}$ at the S' position of interrogative *like*, which will unify with the feature S s'' at the foot of *know*. As a result, within l_8 of *know* we have the newly created expression $\lambda s'' \lambda p.p(s'') \wedge \boxed{5}$, arising from $\lambda s''.\boxed{22}$ and from $\boxed{22} = q_3$ by adjunction of *know* to the S' of *like*. Then, l_8 includes the new λ-expression twice: once it applies it to the doxastic sitation s, and once it combines it with the situation index $\boxed{21}$. Index $\boxed{21}$ (and $\boxed{17}$ below) is left unbound and will thus default to the situation s_0 of the whole proposition. Finally, by substitution of *John*, $\boxed{20}$ is identified with x. The result of the computation is given in (21).[6]

$$
(21)\quad \boxed{\begin{array}{l}
\text{john}(x),\ l_1 : \text{like}(x, y, s)\\
l_2 : p = \lambda s.\boxed{7},\ q_3 : \lambda p.p(s'') \wedge \boxed{5},\\
l_4 : \text{some}(x, \boxed{9}, \boxed{10}),\ l_5 : \text{person}(x, s_0),\\
l_6 : \text{every}(y, \boxed{14}, \boxed{15}),\ l_7 : \text{person}(y, \boxed{18})\\
l_8 : \text{every}(s, s \in Dox_x(s_0),\\
\quad \forall p'[(\lambda s''.q_3)(s)(p')\\
\quad\quad \leftrightarrow (\lambda s''.q_3)(s_0)(p')]),\\
\boxed{7} \geq l_1, \boxed{5} \geq l_2, \boxed{5} \geq l_4, \boxed{9} \geq l_5, \boxed{10} \geq l_2\\
\boxed{7} \geq l_6, \boxed{14} \geq l_7, \boxed{15} \geq l_1
\end{array}}
$$

5 Conclusion

In sum, we have proposed an account for the semantics of wh-questions in LTAG that captures the different scope properties of wh- and non-wh-quantifiers and that derives the adequate semantics for embedded interrogative clauses.

Acknowledgments

For many fruitful discussions we would like to thank Aravind K. Joshi and all members of the XTAG Group at the University of Pennsylvania. Furthermore, we are grateful to two anonymous reviewers for their useful comments.

References

Sigrid Beck and Hotze Rullmann. 1999. A Flexible Approach to Exhaustivity in Questions. *Natural Language Semantics*, (7):249–298.

Steven Berman. 1991. *The Semantics of Open Sentences*. Ph.D. thesis, UMass, Amherst.

Gennaro Chierchia. 1993. Questions with quantifiers. *Natural Language Semantics*, (1):181–234.

Ann Copestake, Dan Flickinger, Ivan A. Sag, and Carl Pollard. 1999. Minimal Recursion Semantics. An Introduction. Manuscript, Stanford University.

Claire Gardent and Laura Kallmeyer. 2003. Semantic Construction in FTAG. In *Proceedings of EACL 2003*, Budapest.

Jonathan Ginzburg and Ivan A. Sag. 2000. *Interrogative Investigations. The Form, Meaning, and Use of English Interrogatives*. CSLI.

Aravind K. Joshi, Laura Kallmeyer, and Maribel Romero. 2003. Flexible Composition in LTAG: Quantifier Scope and Inverse Linking. In Harry Bunt, Ielka van der Sluis, and Roser Morante, editors, *Proceedings of the Fifth International Workshop on Computational Semantics IWCS-5*, pages 179–194, Tilburg.

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

Laura Kallmeyer and Maribel Romero. 2004. LTAG Semantics with Semantic Unification. In *Proceedings of TAG+7*, Vancouver.

Lauri Karttunen. 1977. The syntax and semantics of questions. *Linguistics and Philosophy*, (1):3–44.

Anthony S. Kroch. 1987. Unbounded dependencies and subjacency in a Tree Adjoining Grammar. In A. Manaster-Ramer, editor, *Mathematics of Language*, pages 143–172. John Benjamins, Amsterdam.

Uptal Lahiri. 1991. *Embedded Interrogatives and Predicates That Embed Them*. Ph.D. thesis, MIT.

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

Robert van Rooy. 2003. Quetioning to Resolve Decision Problems. *Linguistics and Philosophy*, (26):727–763.

[6] In the case of direct questions Q, we can assume that their truth-conditional content amounts to the proposition expressed by *I want to know* Q. For weaker degrees of exhaustivity of direct and embedded questions compatible with the present approach, see (Beck and Rullmann, 1999) and (van Rooy, 2003).