

Semantic Reconstruction in LTAG

Tatjana Scheffler

December 13, 2003

Abstract

In this paper, we show how so-called reconstruction effects can be modelled in a TAG semantics. We derive a lexical entry and semantic specification for *how many*, and show how it interacts compositionally with other scopal items in a question. The use of an underspecified semantics allows the compact representation of scope ambiguities. We demonstrate how this also enables us to obtain the correct readings in embedded questions. We briefly discuss the issue of weak island constraints, which eliminate one of the readings of an ambiguous embedded *how many* question.

Contents

1	Semantics of Lexicalized Tree Adjoining Grammar	2
1.1	Lexicalized Tree Adjoining Grammar	2
1.2	Semantics	3
1.2.1	Semantic Underspecification in TAG	3
1.2.2	Notation	4
1.2.3	Meaning of Questions	6
2	“how many” Questions	6
2.1	Scope Ambiguities	6
3	A TAG analysis	7
3.1	How many	8
3.2	Extraction from embedded sentences	10
3.3	Islands	11
4	Conclusions	12
	References	12

1 Semantics of Lexicalized Tree Adjoining Grammar

1.1 Lexicalized Tree Adjoining Grammar

Lexicalized Tree Adjoining Grammar (LTAG) is a tree rewriting system whose formal basis was first developed by Joshi et al. (1975). It has proven to be very useful for representing natural language grammar (see Kroch and Joshi, 1985).

An LTAG consists of a finite set of trees (the *elementary trees*) associated with lexical items. It has furthermore two composition operations, *substitution* and *adjunction* (or *adjoining*) for combining elementary trees into larger structures. The operations are depicted in figure 1.

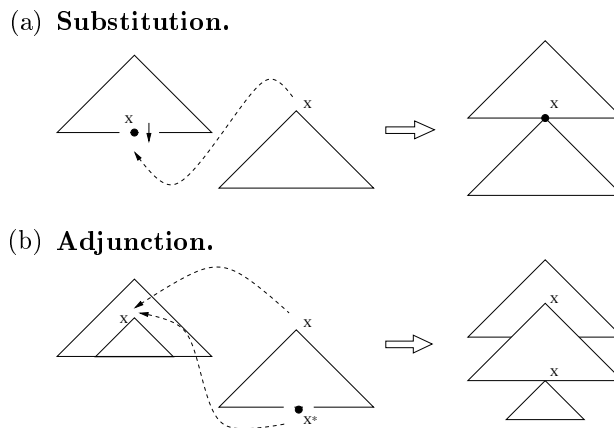


Figure 1: Substitution and Adjunction.

An LTAG analysis of a sentence is commonly represented not by the *derived tree*, i.e. the complete phrase structure tree that results from combining all the involved elementary trees. Instead, a *derivation tree* is given, which has as its vertices the names of elementary trees, and the combination operations (substitution or adjunction) between them as its edges. A TAG derivation and derivation tree¹ for the sentence “John always loves Mary.” is shown in figure 2.

A TAG has several desirable formal properties that make an elegant description of natural language possible. The two most important ones are (1) an extended domain of locality as compared to traditional phrase-structure grammars—in a TAG, the semantic and syntactic arguments of a lexical item are realized *locally* in that item’s elementary tree—, and (2), the factoring of

¹Full lines in the derivation trees denote the adjunction operation, dashed lines substitution. The numbers on the edges are the Gorn addresses (Gorn, 1967) of the adjunction resp. substitution targets in the outer tree.

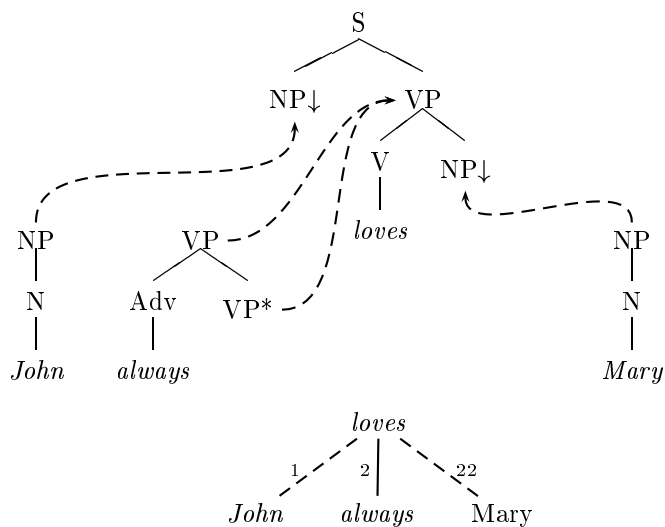


Figure 2: Derivation of the sentence *John always loves Mary*.

recursion into separate trees, the auxiliary trees. This second property, along with the operation of adjunction, allows for an elegant account for unbounded dependencies such as wh-questions.

1.2 Semantics

It is commonly argued that semantic composition in TAG should be done with respect to the derivation tree, not the derived tree. This is possible because semantic arguments of a lexical item (i.e., an *anchor*) are encapsulated in the elementary tree of that item. Thus, each elementary tree is associated with its appropriate semantic representation, and semantics of bigger chunks of the sentence are composed incrementally in parallel with the syntactic composition.

We choose a flat semantic representation with unification variables (similar to MRS, Copestake et al., 1999). Each elementary tree is assigned one semantic representation. Furthermore, variables in the semantics can be linked to nodes in the elementary tree. In this way, e.g. the variables whose values will be provided by the subject resp. the object of a transitive verb can be distinguished. See figure 3 for an example LTAG lexicon with semantic representations.

1.2.1 Semantic Underspecification in TAG

As laid out in (Kallmeyer and Joshi, 2003), semantic representations in TAG can be underspecified, to account for example for scope ambiguities.

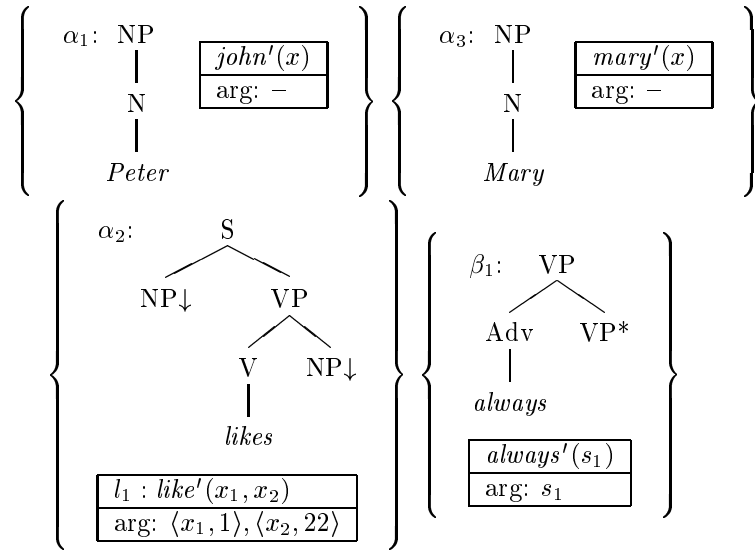


Figure 3: Example lexicon with semantics.

In addition to predications, the semantics also include propositional metavariables called *holes* ($h_1, h_2 \dots$). Holes are variables over propositional labels ($l_1, l_2 \dots$); here, they are used to provide underspecified representations of scope ambiguities. Semantic representations can contain constraints on the relative scope of holes and labels, providing an ambiguous semantics. At the end of a derivation, all possible *pluggings*, i.e. bijections between holes and labels, must be found to obtain the different possible scopings of the sentence.

1.2.2 Notation

In the remainder of this paper, we will use a notation for the derivation of TAG semantics that uses feature structures to keep track of the variable unifications. A simple derivation for the sentence “Every dog barks.” is shown in figure 4.

Note that the lexical item “every” is split into two separate elementary trees, as suggested for example in (Kallmeyer, 1999). One tree is substituted into the appropriate NP node and provides the predicate-argument information; the other tree is a degenerate auxiliary tree that consists only of a single S node, and which is used to obtain the correct scope constraints.

The figure shows the derivation tree for “Every dog barks”, but instead of the names of elementary trees, the nodes contain the semantic information associated with these elementary trees. Each tree is associated with a set of formulae and/or scope constraints. In addition, a feature structure specifies semantic top (T) and bottom (B) features associated with nodes in the syntactic

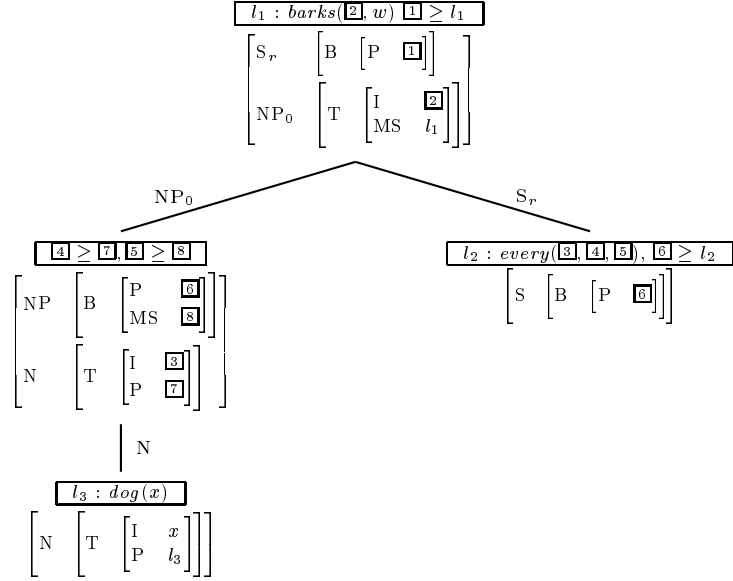


Figure 4: Semantic derivation of *Every dog barks*.

(i.e. elementary) trees. These feature structures store propositional (P, MS) and individual (I) variables to account for the appropriate variable unification effects. Boxes ($\boxed{1}$, $\boxed{2}$, ...) are coreference indices, and they can appear both within the feature structures as well as the formulae.

Unification follows the usual definitions for unification in Feature-based TAG syntax. Thus, the semantic derivation parallels the syntactic derivation. After carrying out all operations on the example derivation in figure 4, and after finalizing the semantic feature tree by unifying all corresponding top and bottom feature structures, we obtain the following semantic representation for our example sentence:

$$(1) \quad l_1 : \text{bark}(x, w), l_2 : \text{every}(x, \boxed{4}, \boxed{5}), l_3 : \text{dog}(x), \boxed{1} \geq l_1, \boxed{4} \geq l_3, \boxed{5} \geq l_1, \boxed{1} \geq l_2$$

The actual of the meaning(s) of a sentence can be obtained by finding pluggings from the coreference indices (*holes*) to labels that obey all explicit and implicit constraints². The only allowed plugging, and therefore the only reading of our simple example is:

$$(2) \quad \boxed{1} \rightarrow l_2, \boxed{4} \rightarrow l_3, \boxed{5} \rightarrow l_1 \quad \Rightarrow \quad l_2 : \text{every}(x, l_3 : \text{dog}(x), l_1 : \text{bark}(x, w))$$

²An implicit constraint is, for example, that no label can appear both in the restriction and the nuclear scope of a quantifier.

1.2.3 Meaning of Questions

We adopt a standard view towards the meaning of questions, which analyses a question denotation as a set of propositions, namely all those propositions that answer the question. See example 3 for a simple question, and its formal and intuitive semantic representations.

- (3) (a) Who did Kim talk to?
(b) $Q : \lambda w \lambda p.[p(w) = 1 \wedge \text{some}(x, \text{person}(x), p = \lambda w'. \text{talk_to}(k, x, w'))]$
(c) {Kim talked to Susan, Kim talked to Bill, Kim talked to John, ... }

2 “how many” Questions

In this paper, we deal with questions that are introduced by the question word *how many*. Broadly speaking, they ask for a number of objects that have some property. An example sentence is (4), given along with its semantic representation.

- (4) How many students did Kate interview?
 $Q : \lambda w \lambda p.[p(w) = 1 \wedge \text{some}(n, n \in \mathbb{N},$
 $p = \lambda w'. \text{some}(y, \text{student}^*(y, w') \wedge |y| = n, \text{interview}(k, y, w')))]$

2.1 Scope Ambiguities

The *how many* phrase interacts with other scopal predicates, yielding sentence (5) ambiguous:

- (5) How many students should Kate interview?
(a) $Q : \lambda w \lambda p.[p(w) = 1 \wedge \text{some}(n, n \in \mathbb{N},$
 $p = \lambda w'. \text{should}(\text{some}(y, \text{student}^*(y, w') \wedge |y| = n, \text{interview}(k, y, w')))]$
(b) $Q : \lambda w \lambda p.[p(w) = 1 \wedge \text{some}(n, n \in \mathbb{N},$
 $p = \lambda w'. \text{some}(y, \text{student}^*(y, w') \wedge |y| = n, \text{should}(\text{interview}(k, y, w')))]$

The first meaning might be intended when Kate is known to make a representative survey among students, and the speaker wants to know how many students (no matter who they are) have to be interviewed in order for Kate to be able to make valid judgments. Meaning (b) is more salient if Kate has been assigned to ask certain students (e.g., Bill, Bob, and Susan), and the speaker wants to know how big the group of people whom Kate has to interview is exactly.

3 A TAG analysis

In this section, we give MC-TAG elementary trees and appropriate semantic representations that show how to derive the meaning of *how many* sentences in TAG. We will proceed step-by-step, as outlined by the following example questions:

- (6) Who did Mary see?
 $Q : \lambda w \lambda p.[p(w) = 1 \wedge \text{some}(y, \text{person}(y), p = \lambda w'.[\text{see}(x, y, w') \wedge \text{mary}(x)])]$
- (7) Which students did Mary see?
 $Q : \lambda w \lambda p.[p(w) = 1 \wedge \text{some}(y, \text{student}^*(y), p = \lambda w'.[\text{see}(x, y, w') \wedge \text{mary}(x)])]$
- (8) How many students did Mary see?
 $Q : \lambda w \lambda p.[p(w) = 1 \wedge \text{some}(n, n \in \mathbb{N}, p = \lambda w'.\text{some}(y, \text{student}^*(y, w') \wedge |y| = n, \text{see}(x, y, w') \wedge \text{mary}(x)))]]$
- (9) How many students should Mary see?
- (a) $Q : \lambda w \lambda p.[p(w) = 1 \wedge \text{some}(n, n \in \mathbb{N}, p = \lambda w'.\text{should}(\text{some}(y, \text{student}^*(y, w') \wedge |y| = n, \text{see}(x, y, w') \wedge \text{mary}(x)))))]]$
- (b) $Q : \lambda w \lambda p.[p(w) = 1 \wedge \text{some}(n, n \in \mathbb{N}, p = \lambda w'.\text{some}(y, \text{student}^*(y, w') \wedge |y| = n, \text{should}(\text{see}(x, y, w') \wedge \text{mary}(x)))))]]$
- (10) How many students do you think Mary should see?
- (a) $Q : \lambda w \lambda p.[p(w) = 1 \wedge \text{some}(n, n \in \mathbb{N}, p = \lambda w'.\text{think}(u, \text{should}(\text{some}(y, \text{student}^*(y, w') \wedge |y| = n, \text{see}(x, y, w') \wedge \text{mary}(x)))))]]$
- (b) $Q : \lambda w \lambda p.[p(w) = 1 \wedge \text{some}(n, n \in \mathbb{N}, p = \lambda w'.\text{think}(u, \text{some}(y, \text{student}^*(y, w') \wedge |y| = n, \text{should}(\text{see}(x, y, w') \wedge \text{mary}(x)))))]]$
- (11) How many students do you wonder whether Mary should see?
- (b) $Q : \lambda w \lambda p.[p(w) = 1 \wedge \text{some}(n, n \in \mathbb{N}, p = \lambda w'.\text{wonder}(u, \text{some}(y, \text{student}^*(y, w') \wedge |y| = n, \text{should}(\text{see}(x, y, w') \wedge \text{mary}(x)))))]]$

In (11), the weak island effect (Ross, 1967) can be observed: Only the non-reconstructed reading (where Mary should see specific students) is possible for this sentence.

Figures 5 and 6 respectively show the syntactic and semantic derivations for the sentence *Which students did Mary see?*

³The two component parts of the lexical entry for *which* are marked by a dotted box.

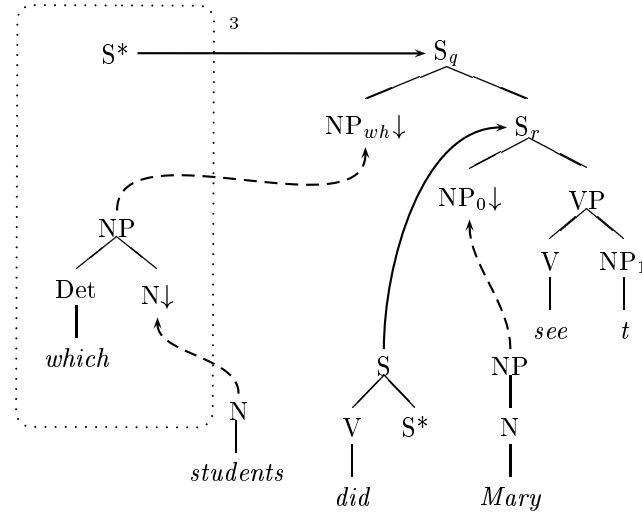
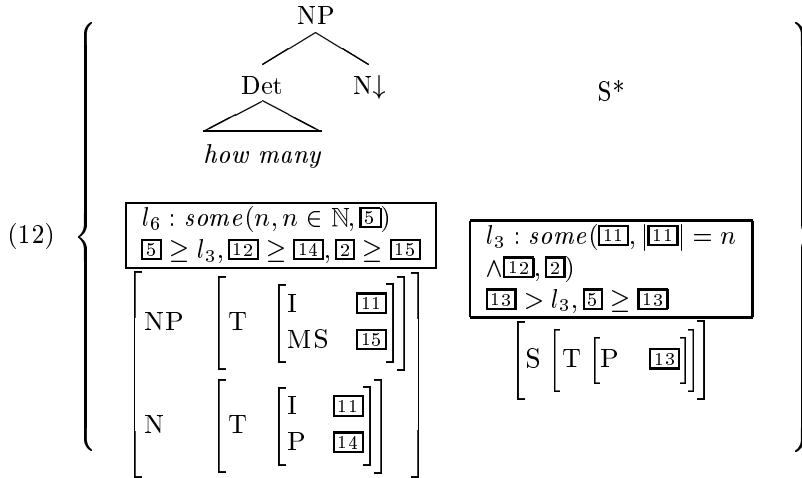


Figure 5: Syntactic derivation of *Which students did Mary see?*

3.1 How many

We propose the following lexical entry for *how many*:



This semantic representation ensures that the predicate substituted into it ($\boxed{14}$) ends up in the restriction of the quantifier, while its individual variable ($\boxed{11}$) is inherited up to the NP node (and will later be unified with a position in the verbal semantics).

Furthermore, the constraints $\boxed{5} \geq \boxed{13}$, $\boxed{13} > l_3$, and $\boxed{2} \geq \boxed{15}$ specify that the minimal scope of the verbal tree must be included in both quantifiers introduced

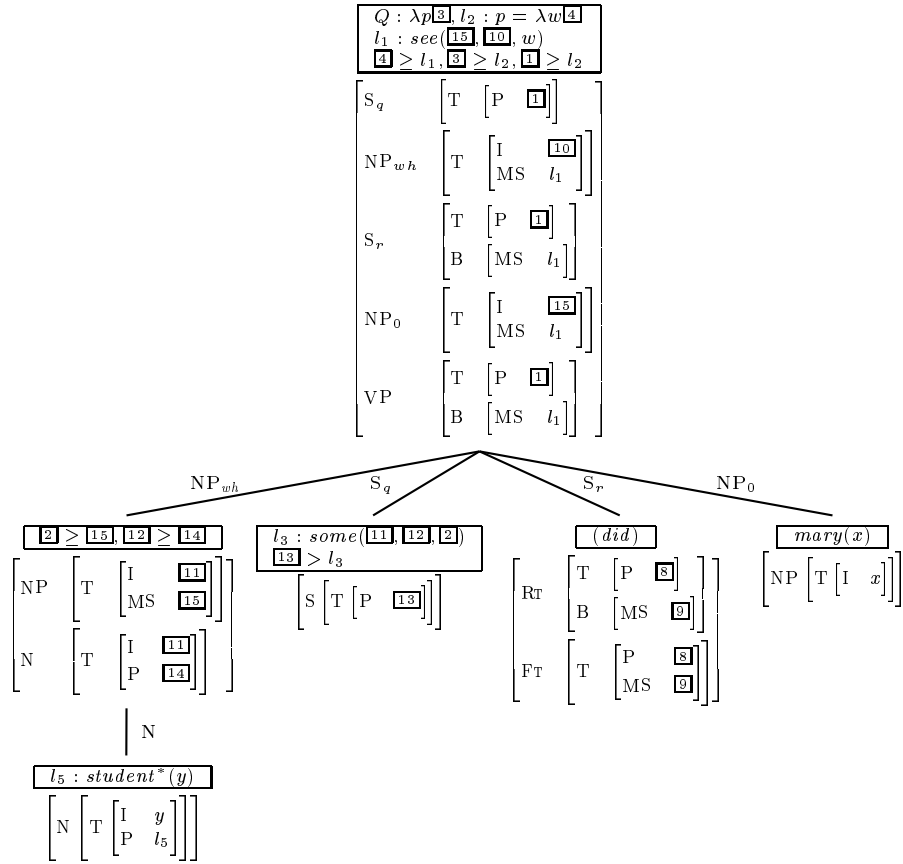


Figure 6: Semantic derivation of *Which students did Mary see?*

by *how many*, while the outer scope ($\boxed{13}$) dominates one quantifier (this accounts for the reconstruction effect), but is included in the other one (the question word itself must stay on top of the tree and not be reconstructed).

Figure 7 shows the semantic derivation tree for sentence (9).

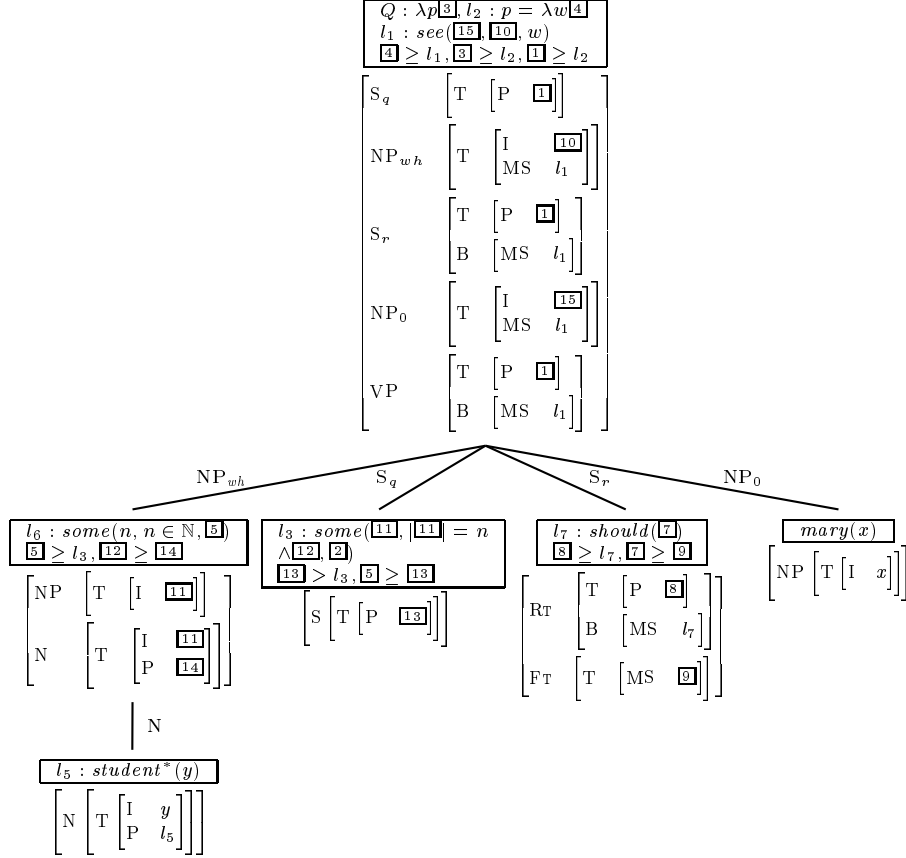


Figure 7: Semantic derivation tree for *How many students should Mary see?*

3.2 Extraction from embedded sentences

In TAG, control and raising predicates anchor auxiliary trees that adjoin into their embedded sentences. Figure 8 shows the lexical entry for the verb *think*⁴. Thus, the semantic derivation for sentence (10) is very similar to the non-embedded sentence (9). The only difference is the additional combination of

⁴For simplicity, we have already combined *think* with *do* and *you* in this figure. So for all practical purposes, this would not be a lexical entry for any broad TAG-grammar, although nothing in the theory prohibits such lexical items.

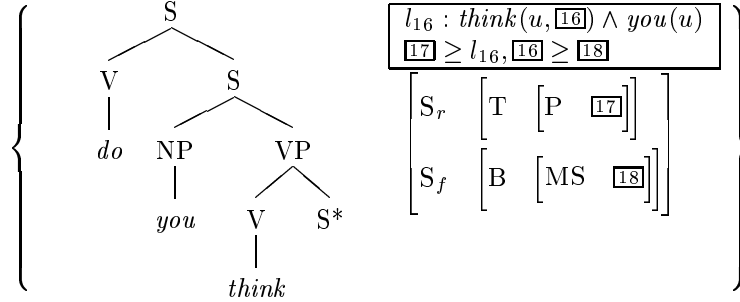


Figure 8: Lexical entry for *think*.

the semantic representation as shown in figure 7 with the semantic formulae and feature structure shown in figure 8. This yields the following semantics for the complete sentence *How many students do you think Mary should see?*:

$$(13) \quad \boxed{\begin{array}{l} Q : \lambda p \mathbf{3}, l_2 : p = \lambda w \mathbf{4}, l_1 : \text{see}(x, y, w), l_6 : \text{some}(n, n \in \mathbb{N}, \mathbf{5}), \\ l_5 : \text{student}^*(y), l_3 : \text{some}(y, |y| = n \wedge \mathbf{12}, \mathbf{2}), l_7 : \text{should}(\mathbf{7}), \\ \text{mary}(x), l_{16} : \text{think}(u, \mathbf{16}), \text{you}(u) \end{array}} \\ \boxed{\begin{array}{l} \mathbf{4} \geq l_1, \mathbf{3} \geq l_2, \mathbf{1} \geq l_2, \mathbf{5} \geq l_3, \mathbf{12} \geq l_5, \mathbf{1} > l_3, \mathbf{5} \geq \mathbf{1}, \mathbf{1} \geq l_7, \\ \mathbf{7} \geq l_1, \mathbf{1} \geq l_{16}, \mathbf{16} \geq l_1 \end{array}}$$

There are two possible pluggings of labels into holes (i.e., the coreference boxes). They are shown below, and they represent the two respective readings shown in example (10).

$$(a) \quad \begin{array}{l} \mathbf{1} = \mathbf{4} \\ \mathbf{2} \rightarrow l_1 \\ \mathbf{3} \rightarrow l_6 \\ \mathbf{4} \rightarrow l_{16} \\ \mathbf{5} \rightarrow l_2 \\ \mathbf{7} \rightarrow l_3 \\ \mathbf{12} \rightarrow l_5 \\ \mathbf{16} \rightarrow l_7 \end{array} \quad (b) \quad \begin{array}{l} \mathbf{1} = \mathbf{4} \\ \mathbf{2} \rightarrow l_7 \\ \mathbf{3} \rightarrow l_6 \\ \mathbf{4} \rightarrow l_{16} \\ \mathbf{5} \rightarrow l_2 \\ \mathbf{7} \rightarrow l_1 \\ \mathbf{12} \rightarrow l_5 \\ \mathbf{16} \rightarrow l_3 \end{array}$$

3.3 Islands

The status of weak islands is not completely clear. Many studies suggest that the factor that prohibits one of the possible interpretations in sentences such as (11), and which is traditionally attributed to the failure of *students* to reconstruct across a weak island barrier (Cresti, 1995, see), is really a pragmatic rather than syntactic or semantic phenomenon.

For the time being, we will follow this line here, and construct the meaning of sentence (11), *How many students do you wonder whether Mary should see?*,

in analogy to (10), *How many students do you think Mary should see?*. We expect a pragmatic process to rule out the infelicitous reading.

4 Conclusions

In this paper we showed that using recently developed frameworks for representing semantics in LTAG, we can account for ambiguities that arise in *how many* questions in an elegant way. The use of holes for underspecification and the feature unification process as employed also in the syntactic composition in TAG together allow the reconstruction of the restrictor deeper in the question. This is possible because TAG provides an extended domain of locality, and because Multi-Component TAGs and constraints for scopal underspecification provide a flexible treatment of scopal phenomena (which was already noted in the literature for quantifier scope and inverse linking phenomena, among others; see (Joshi et al., 2003)).

Here, we gave a lexical entry and semantic specification for *how many* that allows exactly the right ambiguities in simple and embedded questions. We showed how this lexical entry interacts compositionally with other scopal elements in questions.

An account for weak island constraints (that eliminate one of the readings due to an effect usually attributed to the failure of reconstruction of the *how many* restrictor) is left for future work. We propose that weak island barriers in these contexts may actually be a pragmatic effect that should not affect our semantic analysis.

References

- Ann Copestake, Dan Flickinger, Ivan A. Sag, and Carl Pollard. Minimal Recursion Semantics. An Introduction. Draft, Stanford University, 1999.
- Diana Cresti. Extraction and reconstruction. *Natural Language Semantics*, 3: 79–122, 1995.
- Saul Gorn. Explicit definitions and linguistic dominoes. In J. Hart and S. Takasu, editors, *Systems and Computer Science*, pages 77–115. University of Toronto Press, Toronto, CA, 1967.
- A. Joshi, L. Levy, and M. Takahashi. Tree adjunct grammars. *Journal of the Computer and System Sciences*, 10:136–163, 1975.
- Aravind K. Joshi, Laura Kallmeyer, and Maribel Romero. Flexible composition in LTAG, quantifier scope and inverse linking. In *Proceedings of the 5th IWCS*, pages 179–194, Tilburg, NL, 2003.
- Laura Kallmeyer. *Tree Description Grammars and Underspecified Representations*. PhD thesis, Universität Tübingen, 1999. Technical Report IRCS-99-08 at the Institute for Research in Cognitive Science, Philadelphia.

Laura Kallmeyer and Aravind K. Joshi. Factoring predicate argument and scope semantics: Underspecified semantics with LTAG. *Research on Language and Computation*, 1:3–58, 2003.

Anthony Kroch and Aravind K. Joshi. The linguistic relevance of Tree Adjoining Grammar. Technical Report MS-CS-85-16, Department of Computer and Information Sciences, University of Pennsylvania, 1985.

John Robert Ross. *Constraints on variables in Syntax*. PhD thesis, MIT, 1967.