Factoring Predicate Argument and Scope Semantics: Underspecified Semantics with LTAG*

Laura Kallmeyer
University of Tübingen

Aravind Joshi
University of Pennsylvania

Abstract

This paper proposes a compositional semantics for lexicalized tree-adjoining grammar (LTAG). Tree-local multicomponent derivations allow separation of the semantic contribution of a lexical item into one component contributing to the predicate argument structure and a second component contributing to scope semantics. Based on this idea a syntax-semantics interface is presented where the compositional semantics depends only on the derivation structure. It is shown that the derivation structure (and indirectly the locality of derivations) allows an appropriate amount of underspecification. This is illustrated by investigating underspecified representations for quantifier scope ambiguitites and related phenomena such as adjunct scope and island constraints.

1 Introduction: Multicomponent LTAG

A LTAG consists of a finite set of trees (elementary trees) associated with lexical items and composition operations of substitution (replacing a leaf with a new tree) and adjoining (replacing an internal node with a new tree). The elementary trees represent extended projections of lexical items and encapsulate syntactic/semantic arguments of the lexical anchor. They are minimal in the sense that all and only the syntactic/semantic arguments are encapsulated and further, all recursion is factored away. This factoring of recursion is what leads to the trees being extended projections. The elementary trees of LTAG are therefore said to possess an extended domain of locality.

In our approach we use a LTAG varient called multicomponent TAG (MC-TAG). A MC-TAG consists of elementary sets of trees. The locality of composition in LTAG is extended to MC-TAG as follows. Basically, when two multicomponent tree sets are combined, the components of one set combine with only one of the components of the other set. This formalism, called tree-local MC-TAG, is known to be equivalent to LTAG, thus the use of MC-TAG does not take us beyond the power of LTAG. We use tree-local MC-TAG with at most two components in each set. The key idea is that one of the components of a tree set contributes to the predicate argument aspects of semantics and the other component contributes to the scope semantics. This allows us to obtain derivation trees that provide the right kind of underspecification for scope semantics.

2 Derivation trees and semantic dependencies

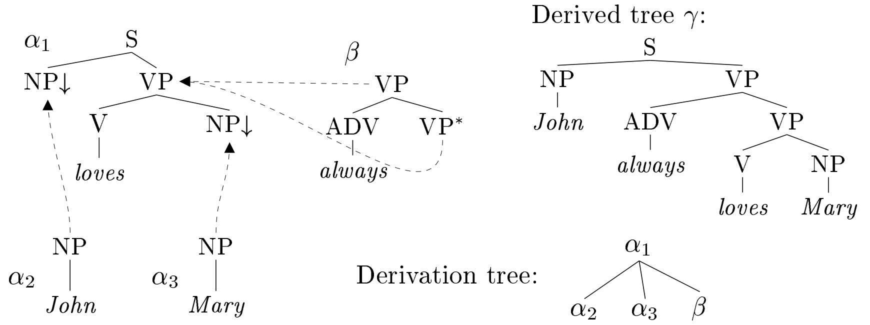
LTAG derivations are represented by derivation trees that record the history of how the elementary trees are put together. A derived tree is the result of carrying out the substitutions and adjoinings.

(1) John always loves Mary.

*This work was done during a visit of Laura Kallmeyer at the Institute for Research in Cognitive Science (IRCS), University of Pennsylvania. A longer version of the paper will appear as technical report at IRCS.

The elementary trees for (1) are shown in (2) together with the derived tree γ and the derivation tree. γ is generated by adding α_2 and α_3 by substitution in α_1 and adjoining β to α_1. This is reflected by the derivation tree: An edge to an intitial tree $\alpha, \alpha_1, \ldots$ stands for a substitution and an edge to an auxiliary tree $\beta, \beta_1, \ldots$ for an adjunction.

(2)



Because of the localization of the arguments of a lexical item within elementary trees the proper way to define compositional semantics for LTAG is with respect to the derivation tree rather than the derived tree. We assume that each elementary tree is related to a semantic representation. The derivation tree indicates how to combine the semantic representations, where the direction of a semantic composition depends on the specific syntactic operation: In case of a substitution an argument is added to the semantic representation, and when adjoining a tree the new semantic representation is applied to the old one. This contrasts with traditional approaches where each node in the syntactic structure is associated with a semantic representation. Although this insight has been present from the beginning of the work on LTAG (Shieber & Schabes 1990) a systematic formulation was begun only recently by Joshi and Vijay-Shanker (1999). One of their goals was to investigate the role of underspecification in compositional semantics; they suggested that LTAG derivation trees provide just the right amount of underspecification necessary for scope semantics. Their discussion was preliminary, however.

(3)

σ_{α_1}	σ_{α_2}	σ_{α_3}	σ_{β}
$l_1 : loves(x_1, x_2)$	$john(x)$	$mary(y)$	$l_2 : always(s_1)$
arg: x_1, x_2	arg: –	arg: –	arg: s_1

(3) shows the semantic representations linked to the elementary trees in (2). We use 'flat' semantic representations (as in, for example, Minimal Recursion Semantics MRS, Copestake et al. 1997) consisting of a conjunctively interpreted set of formulas (typed lambda-expressions) and a set of argument variables. The formulas may have propositional labels $l_1, l_2, \ldots$. Roughly, the application of one semantic representation σ to another σ' consists of assigning values (of appropriate type) from σ' to some of the arguments in σ and then building the union of σ and σ'. In (3), σ_{α_1} is applied to σ_{α_2} assigning x to x_1 and to σ_{α_3} assigning y to x_2. σ_β is applied to σ_{α_1} with l_1 assigned to s_1. The result is (4):

(4)

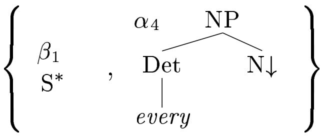
$l_1 : loves(x,y)$, $john(x)$, $mary(y)$, $l_2 : always(l_1)$
arg: –

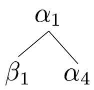
3 Scope information and underspecification

In order to describe underspecified representations for scope ambiguities, we adopt ideas from Hole Semantics (Bos 1995) and enrich the semantic representations with propositional metavariables $h_1, h_2, \ldots$ called holes. A partial order on holes and propositional labels describes the scope structure of a semantic representation. A disambiguation function maps holes to propositional labels in such a way that the scope constraints are respected.

(5) Every student loves some course.

Consider (5) for example. We suppose scope components of quantifiers to be syntactically empty, they are auxiliary trees containing one single node. (6) shows the elementary tree set for *every*, together with the derivation tree for adding this quantifier to *loves*.

(6) 

Derivation tree: 

(7) σ_{α_1} $\begin{array}{|l|}\hline l_1 : loves(x_1, x_2) \\ l_1 \leq h_1 \\ \hline \text{arg: } x_1, x_2 \\ \hline \end{array}$ σ_{β_1} $\begin{array}{|l|}\hline l_2 : every(x, h_2, h_3) \\ s_1 \leq h_3 \\ \hline \text{arg: } s_1 \\ \hline \end{array}$ σ_{α_4} $\begin{array}{|l|}\hline l_3 : p_1(x) \\ l_3 \leq h_2 \\ \hline \text{arg: } p_1 \\ \hline \end{array}$

(7) shows the (revised) semantic representations of α_1 from (2), β_1 and α_4. The constraints $s_1 \leq h_3$ and $l_3 \leq h_2$ in σ_{β_1} and σ_{α_4} separate restriction and body of *every*. The auxiliary tree in the tree set of a quantifier contributes to scope semantics: it introduces slots (h_2 and h_3 in the case of *every*) for the scope of the quantifier, i.e. its restriction and body. The NP part of the tree set contributes to the predicate-argument semantics: it is inserted as a syntactic argument and it contributes (a part of) the restriction of the quantifier. The argument p_1 in σ_{α_4} stands for the predicate denoted by the noun in the NP that will be added by substitution. This separation between scope information and contribution to the predicate argument structure is partly inspired by Muskens (1998) and Muskens & Krahmer (1998).

To make sure that in a substitution step the corresponding argument variables are chosen in the semantic representation, each substitution node is linked to at least one argument variable. In (7) the subject NP of α_1 is linked to x_1 and the object NP to x_2. The N substitution node in α_4 is linked to p_1.

The derivation tree in (6) indicates that σ_{β_1} is applied to σ_{α_1} assigning l_1 to s_1, and σ_{α_1} is applied to σ_{α_4} assigning x to x_1. This leads to (8):

(8) $\begin{array}{|l|}\hline l_1 : loves(x, x_2),\ l_2 : every(x, h_2, h_3),\ l_3 : p_1(x) \\ l_1 \leq h_1,\ l_1 \leq h_3,\ l_3 \leq h_2 \\ \hline \text{arg: } x_2, p_1 \\ \hline \end{array}$

Similarly, semantic representations for *some* are added, where the scope component is also adjoined to the root of α_1. Adding then *student* and *course* gives (9):

(9) $\begin{array}{|l|}\hline l_2 : every(x, h_2, h_3),\ l_4 : some(y, h_4, h_5), \\ l_1 : loves(x, y),\ l_3 : student(x),\ l_5 : course(y) \\ l_3 \leq h_2, l_1 \leq h_3,\ l_5 \leq h_4, l_1 \leq h_5,\ l_1 \leq h_1 \\ \hline \text{arg: } - \\ \hline \end{array}$

With the constraints in (9), $loves(x, y)$ is in the scope of both quantifiers, $student(x)$ in the scope of *every* and $course(y)$ in the scope of *some*. The scope relation between *every* and *some* is unspecified. Thus this approach generates underspecified representations for scope ambiguities.

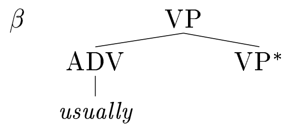
Since LTAG parsing is polynomial it follows that the construction of the underspecified representaion in the derivation tree is also computable in polynomial time.

4 Adjunct Scope

(10) Pat allegedly usually drives a cadillac.

(10) is an example of adjunct scope taken from Bouma et al. 1998. As pointed out by Bouma et al., in (10) *usually* must be in the scope of *allegedly*. Considering only cases where both adverbs are VP-modifiers, there are three scope orders: *allegedly* must have scope over *usually*, and the quantifier *a cadillac* can either have wide scope or be between the two adverbs or it can have narrow scope.
(11) is a natural elementary representation for VP-modifiers as *usually*:

(11)



$l_1 : usually(h_1)$
$s \leq h_1$
arg: s

Schabes & Shieber (1994) would argue that in (10), both adverbs are adjoined to the VP-node of *drives*, i.e. they would prefer multiple adjunction in this case. They propose to consider the scope constraints for (10) as a consequence of the specific syntactic derivation order. However, one of our underlying assumptions was that the compositional semantics depends only on the derivation tree. In particular, it should be independent from syntactic derivation order. Therefore, contrary to Schabes & Shieber, we assume that for tree sets containing single auxiliary trees, multiple adjunctions of several such trees at one and the same node are not allowed. The difference between adverbs modifying the whole VP and adverbs modifying only an embedded adverb is accounted for by adjoining in the first case at the VP-node and in the second case at the node with label ADV (with a different semantic representation). The restriction that several adverbial modifiers cannot be adjoined at the same node reflects our assumption that operators adjoined at the same node should be equivalent with respect to their scoping possibilities.
In the preceding section we have seen that we need multiple adjunction at single nodes since the scope parts of the quantifiers in (5) were adjoined to the same node. These scope components are lexically empty and in this case multiple adjoining does not increase the generative power of the grammar. However, if tree-local multicomponent derivations are combined with an unrestricted use of multiple adjunctions, the power of the formalism is beyond LTAG. Thus our restriction of not allowing multiple adjunctions at the same node in the case of adverbs is formally motivated also and not just from the linguistic considerations.
If multiple adjunction at the VP-node of *drives* is not allowed in (10), the only possible derivation is to adjoin *usually* to the VP-node of *drives*, and then to adjoin *allegedly* to *usually*. With this derivation, the desired restriction is obtained since the argument of *allegedly* is the label of $usually(h_1)$.

5 Island constraints

Island constraints for quantifier scope hold independently from specific quantifiers. In particular relative clauses are widely accepted to be islands for quantifier scope in the sense that quantifiers inside relative clauses cannot outscope the quantifier of the relativized NP (see Rodman 1976, Reyle 1993, Muskens 1995, Kallmeyer 1999).

(12) a. Every representative of most of the companies saw this sample.
b. Every person who represents most of the companies saw this sample.

In (12)a. *most of the companies* can have wide scope, whereas in (12)b., wide scope of the embedded quantifier *most of the companies* is not possible. The relative clause in (12)b. is an island for quantifier scope.

We claim that the difference between (12)a. and (12)b. follows from different kinds of derivations: In (12)a., the tree anchored by *representative* and *of* is an initial tree, whereas the relative clause tree with anchor *represents* in (12)b. is an auxiliary tree. This suggests that auxiliary trees constitute island whereas initial trees do not. In the dependency structure expressed by a derivation tree, auxiliary trees also mark some kind of islands in the following sense: Suppose that the edges in a derivation tree are directed from predicates to arguments. For substitutions we have downwards dependencies whereas for adjunctions we have upwards dependencies. Then, with an auxiliary tree the chain of downwards dependencies is interrupted and a new dependency tree begins. This observation suggests that islands follow not just from a technical difference between two tree operations but rather that quantifiers can rise to higher trees in the derivation structure as long as there is a downwards dependency relation. Based on this observation, island constraints can be read off the derivation structure as follows: Let the top of a semantic representation be defined as its topmost element with respect to subordination. (Subordination is the scope order given by the formulas and constraints in a semantic representation.) On the one hand, everything inside an auxiliary tree is "blocked" by the next higher tree: the top of the semantic representation σ_β of an auxiliary β must be below the top of the semantic representation of the tree to which β is adjoined. On the other hand, as long as there are only arguments added by substitution below an auxiliary β, everything inside these arguments can rise up to the top of σ_β, i.e. the tops of these arguments must be below the top of σ_β.

(13)

$l_1 : saw(x, x_2)$, $l_2 : every(x, h_2, h_3)$, $l_3 : person(x)$ $l_1 \leq h_1$, $l_1 \leq h_3$, $l_3 \leq h_2$, $l_2 \leq h_1$,
arg: x_2

We will illustrate this by showing a part of the analysis of (12)b. (13) is obtained by combinig the semantic representations for *saw*, *every* and *person*. Here $l_2 \leq h_1$ is an additional island constraint that has no effect in this case since *every* is added to the matrix clause. Next, the relative clause is adjoined to the NP-node taking x as an argument. After adding the semantic representations for *represents* and then for *who*, (14) is obtained, where $h_4 \leq h_2$ is an additional island constraint.

(14)

$l_1 : saw(x, x_2)$, $l_2 : every(x, h_2, h_3)$, $l_3 : person(x)$, $l_4 : represents(x, x_3)$ $l_1 \leq h_1$, $l_1 \leq h_3$, $l_3 \leq h_2$, $l_4 \leq h_4$, $l_2 \leq h_1$, $h_4 \leq h_2$
arg: x_2, x_3

Adding the quantifier *most* to *represents* gives (15). Here $l_5 \leq h_4$ and $h_4 \leq h_2$ ensure that *most* (label l_5) is in the restriction (and therefore the scope) of *every*.

(15)

$l_1 : saw(x, x_2)$, $l_2 : every(x, h_2, h_3)$, $l_3 : person(x)$, $l_4 : represents(x, y)$, $l_5 : most(y, h_5, h_6)$, $l_6 : p_1(y)$ $l_1 \leq h_1$, $l_1 \leq h_3$, $l_3 \leq h_2$, $l_4 \leq h_4$, $l_4 \leq h_6$, $l_6 \leq h_5$ $l_2 \leq h_1$, $h_4 \leq h_2$, $l_5 \leq h_4$, $h_6 \leq h_4$
arg: x_2, p_1

Note that the locality of the TAG is responsible for the fact that quantifier scope trees inside a relative clause cannot be adjoined to the matrix clause. So the locality of the grammar together with the island constraints read off the derivation tree provide just the amount of underspecification needed for quantifier scope.

6 Related work

Among recent approaches to underspecified semantics, in particular Muskens & Krahmer (1998) and Kallmeyer (1999) are closely related to our work. Both proposals also separate scope information from predicate argument semantics. Muskens and Krahmer however do not adopt any locality constraint and therefore their use of underspecification is too general. Kallmeyer uses tree descriptions and makes use of the locality of TAGs. But in order to control the amount of underspecification that comes with the use of descriptions, rather complex formal definitions are necessary. This problem is avoided in our approach where syntactic structures are represented by trees and underspecification is used only in a very limited way for scope relations between propositional formulas.

7 Conclusion

In this paper, we have presented a compositional semantics for LTAG based on the idea of factoring predicate argument and scope semantics. The framework proposed here provides just the right amount of underspecification adequate for the analysis of scope ambiguities.

References

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

Bouma, G., Malouf, R. & Sag, I.: 1998, Adjunct Scope. Workshop *Models of Underspecification and the Representation of Meaning*, May 1998, Bad Teinach.

Copestake, A., Flickinger, D. & Sag, I. A.: 1997, Minimal Recursion Semantics. An Introduction. Manuscript, Stanford University.

Joshi, A. K. & Vijay-Shanker, K.: 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 ot the Third International Workshop on Computational Semantics (IWCS-3)*, Tilburg, pp. 131–145.

Kallmeyer, L.: 1999, Synchronous Local TDGs and Scope Ambiguities, *in* G. Bouma, E. W. Hinrichs, G.-J. Kruijff and R. T. Oehrle (eds), *Constraints and Resources in Natural Language Syntax and Semantics*, CSLI, pp. 245 – 262.

Muskens, R.: 1995, Order-independence and underspecification, *in* J. Groenendijk (ed.), *Ellipsis, Underspecification, Events and More in Dynamic Semantics*, DYANA Report R2.2.C.

Muskens, R.: 1998, Underspecified semantics. Workshop *Models of Underspecification and the Representation of Meaning*, May 1998, Bad Teinach.

Muskens, R. & Krahmer, E.: 1998, Description Theory, LTAGs and Underspecified Semantics, *Fourth International Workshop on Tree Adjoining Grammars and Related Frameworks, IRCS Report 98–12*, University of Pennsylvania.

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

Rodman, R.: 1976, Scope phenomena, "movement transformations", and relative clauses, *in* B. H. Partee (ed.), *Montague Grammar*, Academic Press, pp. 165–176.

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

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