

Compositional Semantics for Relative Clauses in Lexicalized Tree Adjoining Grammars

Chung-hye Han
Simon Fraser University

1. Introduction

This paper proposes a compositional semantics for relative clauses in Lexicalized Tree Adjoining Grammars (LTAG). As explicated in (Joshi and Vijay-Shanker, 1999; Joshi and Kallmeyer, 2000), in the phrase-structure based compositional semantics the meaning of a sentence is computed as a function of meaning of each node in the tree. On the other hand, in LTAG based compositional semantics, the meaning of a sentence is computed as a function of meaning of elementary trees put together to derive the sentence. This is because in LTAG, the elementary objects are lexicalized trees that encapsulate all syntactic/semantic arguments of the associated lexical item (i.e., the anchor). Each elementary tree is associated with a semantic representation, and given the history of how the elementary trees are put together to form a sentence, its semantics can be computed by combining the semantic representations of the elementary trees. In other words, semantics in LTAG can be defined to operate on bigger objects than in a phrase-structure based approach, without violating the principle of compositionality. One could naturally compose the full derived tree for the sentence at the end of the derivation process, and then compute the semantics on each node in the full derived tree. However, this has two major disadvantages: first, there is no correspondence between semantic composition and the syntactic operations of substitution and adjunction; and secondly, it is impossible to compute semantic interpretation incrementally and monotonically for partial derivations. This suggests that compositional semantics in TAG should be done on the derivation tree, not on the derived tree.

There are two ways of doing semantics on the derivation tree: (i) synchronous TAG as in (Abeillé, 1994), and (ii) flat semantics as in (Joshi and Vijay-Shanker, 1999; Joshi and Kallmeyer, 2000). In this paper, I pursue the flat semantics approach (also known as minimal recursion semantics), in which the main operation for semantic composition is the conjunction of the semantic representations associated with each elementary tree along with the unification of variables contributed by each semantic representation. Doing flat semantics on relative clauses is particularly interesting because it involves defining an alternative semantic role for the relative pronoun to the phrase-structure based approach, in which the relative pronoun has been argued to be an operator that turns the relative clause into a function of a predicate type (Heim and Kratzer, 1998). In addition, it involves defining a relationship between the head noun and the *wh* relative pronoun, which turns out to be non-trivial.

I will start the paper with an illustration of an LTAG-based compositional semantics for a simple sentence with an attributive adjective in section 2. This will allow us to understand how semantic composition in general and modification in particular work in LTAG semantics. In section 3, using a relative clause containing a genitive relative pronoun (e.g., *whose*), a case of pied-piping, I will first present a couple of approaches that do not work. This will allow us to clarify the necessary components for a proper analysis. I then propose my analysis of relative clauses that accounts for these components. Section 4 discusses how the proposed analysis can be generalized to relative clauses with a simple relative pronoun, adjunct relative clauses and relative clauses whose relative pronoun is deeply embedded in a recursive genitive NP. The discussion on recursive genitive NPs will lead to a slight modification of the proposed analysis. In general, I follow the English grammar developed in (The XTAG-Group, 2001) for the syntax of various constructions discussed in this paper (although in some cases, where convenient, I differ from the XTAG analysis to produce the appropriate semantics).

2. LTAG-based Compositional Semantics for a Simple Sentence with an Attributive Adjective

The elementary trees to generate the derived and the derivation tree for sentence in (1), and their corresponding semantic representations are given in Figure 1.

(1) John solved a difficult problem.

* I am indebted to Aravind Joshi for many discussions on this topic. I also wish to thank Tonia Bleam, Mark Dras, Maribel Romero, Anoop Sarkar, Alexander Williams and the members of the XTAG group at the University of Pennsylvania. All errors are mine.

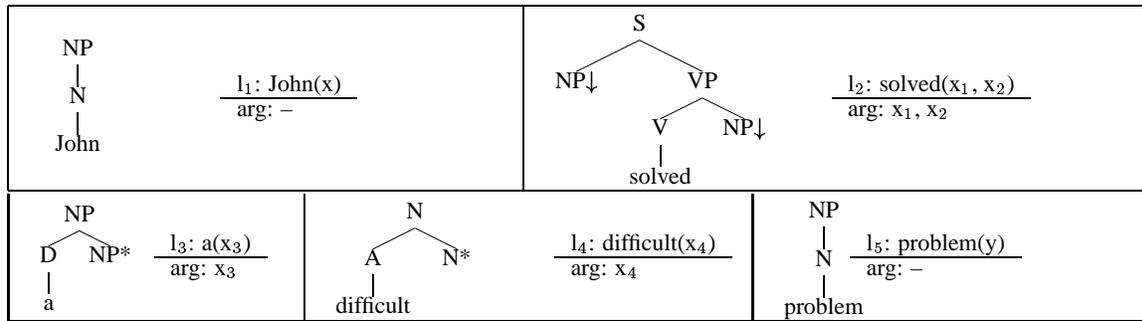


Figure 1: Elementary Trees and Semantic Representations

The symbols l_i label each semantic representation. The elementary tree anchoring *solved* contains two substitution sites, each corresponding to subject and object argument. This is associated with a semantic representation with the predicate *solved* and two argument variables. The *Arg* slot contains two variables x_1 and x_2 , indicating that they each must be unified with variables contributed by the subject and the object in the semantic composition. The elementary trees anchoring *a* and *difficult* are adjunction trees. They are each associated with a semantic representation with the predicate corresponding to the anchor and one argument variable. The *Arg* slot contains a variable which must be unified with a variable that is contributed by the adjoining noun (or NP). The elementary trees anchoring *John* and *problem* are associated with semantic representations that each contributes a variable. The argument slot is empty, reflecting the fact that the elementary tree is an initial tree with no substitution sites. The derivation tree and the semantic composition for (1) are given in Figure 2.

Assuming a bottom-up semantic composition, first the semantics for *a* and *problem* combine, unifying the argument variable of *a* with the variable contributed by *problem*. Further, the semantics for *difficult* and *problem* combine, unifying the argument variable of *difficult* with the variable contributed by *problem*. And then the semantics for *John* and *problem* combine with the semantics for *solved*. This results in unifying the argument variables of *solved* with the variables contributed by *John* and *problem*. The final semantics for (1) is a conjunction of each labeled semantic representations in Figure 2.



Figure 2: Derivation Tree and Semantic Composition

3. LTAG-based Compositional Semantics for Relative Clauses

The example in (2) will be used throughout this section to illustrate the analysis for an LTAG-based semantics for relative clauses.

- (2) A problem whose solution is difficult

It will be shown that the main source of the problem is that in a relative clause there's actually two variables that must be kept track of: a variable corresponding to the gap in the relative clause and the variable corresponding to the *wh* relative pronoun. In order to get the correct predicate/argument relation and the semantics that a relative clause is a modifier of a head noun, the variable for the relative pronoun (*whose*) must unify with the head noun (*problem*), and the variable for the gap must come from the head of the pied-piped structure (*solution*). In the simple case with no pied-piping, the two variables are the same. But as soon as the relative pronoun occurs in a pied-piped structure, the two variables are not the same, and since the *wh* is embedded, its variable cannot directly unify with the variable from the head noun, creating a locality problem. In this section, I will first present a couple of approaches that do not work in subsections 3.1 and 3.2 to illustrate the issues just described and motivate the analysis proposed in subsection 3.3.

3.1. Trial 1

As a first try, let's consider the elementary trees and their corresponding semantic representations in Figure 3. The relative clause tree has a substitution site designated as NP[WH]. I am using this notation for convenience to represent the assumption that relative clause trees are encoded with a [WH] feature that requires a phrase dominating a relative pronoun to be substituted into this position in the course of derivation.

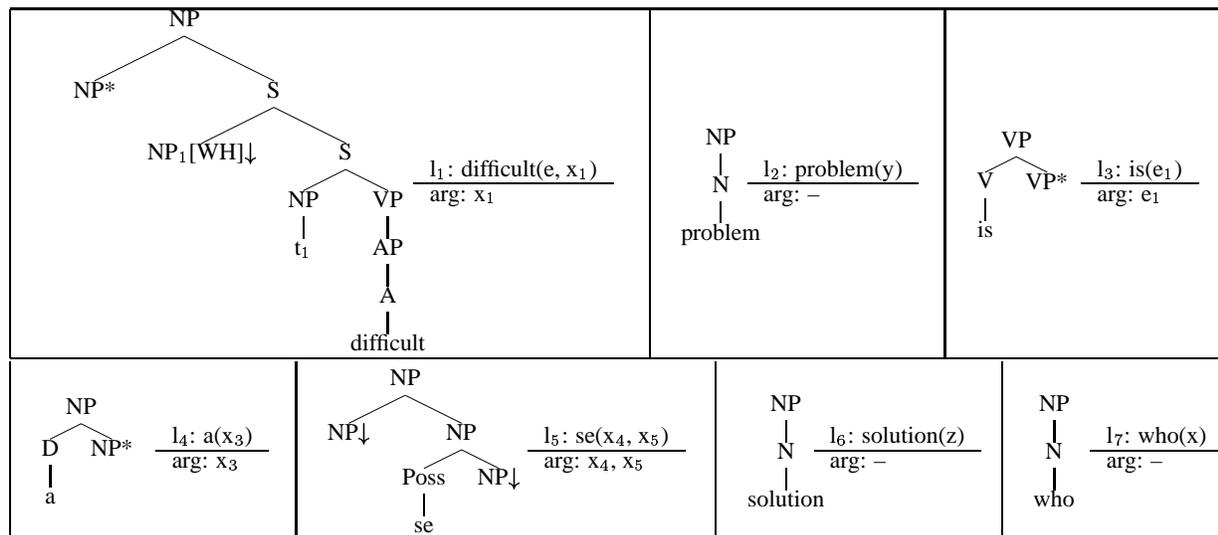


Figure 3: Elementary Trees and Semantic Representations

The semantics for *a*, *problem*, *solution* and *who* are straightforward. I have defined the semantics of the auxiliary verb *is* to be a proposition composed of a predicate and an event argument. This event argument is needed to unify with the event argument contributed by the adjoining verb (or adjective). Consequently, the semantics for the relative clause tree anchoring *difficult* is defined to contribute an event variable. Further, the semantics for the relative clause tree is defined to require an argument variable which must unify with the variable contributed by the head noun (i.e., *problem*). Moreover, assuming that *se* in *whose* is equivalent to genitive 's, *se* anchors an elementary tree with two substitution sites for the possessor and the possessee. This corresponds to a semantic representation with the predicate *se* and two argument variables that must unify with the variables contributed by the substituting NPs. The derivation tree for (2) and the semantic composition under this approach are given in Figure 4.¹



Figure 4: Derivation Tree and Semantic Composition

The problem with this approach is that it derives the incorrect meaning that it is the *problem* that is difficult, not the *solution*. Another problem, which is related to the first problem, is that there is no way to define the relationship between the relative pronoun and the head noun.

1. Notice that I am assuming multiple modification analysis given in (Schabes and Shieber, 1994).

3.2. Trial 2

This thus takes us to the second approach. In this approach, I define an operator called LINK that enforces the unification of the variables contributed by the *wh* relative pronoun and the head noun.

The LINK operation does the similar job as predicate modification in phrase-structure based compositional semantics, as defined in (3) (Heim and Kratzer, 1998). When applied to a relative clause and its head noun, which are both predicate types, the predicate modification ensures that both of them are predicates over the same variable. This in turn effectively derives the interpretation of the relative clause as a modifier of the head noun. The LINK operation is intended to perform the same function.

(3) Predicate Modification

If α has the form α $\begin{matrix} \beta \\ \gamma \end{matrix}$, and $[[\beta]]^s$ and $[[\gamma]]^s$ are both in $D_{\langle e,t \rangle}$, then $[[\alpha]]^s = \lambda x_e [[\beta]]^s(x) \wedge [[\gamma]]^s(x)$.

The semantic representations under the second approach are given in Figure 5. The only difference between the first and the second approaches is in the semantics for the relative clause elementary tree anchoring *difficult*. Here, x_1 stands for the variable for the *wh* relative pronoun, and x_2 stands for the variable for the head noun. We can think of [WH] feature encoded in the relative clause tree to be responsible for contributing the variable for the relative pronoun. This approach again derives wrong semantics for (2): the *problem* incorrectly ends up being *difficult*, as shown in Figure 6.

$\frac{l_1: [\text{difficult}(e,x_1) \wedge \text{LINK}(x_1,x_2)]}{\text{arg: } x_1, x_2}$	$\frac{l_2: \text{problem}(y)}{\text{arg: } -}$	$\frac{l_3: \text{is}(e_1)}{\text{arg: } e_1}$
$\frac{l_4: a(x_3)}{\text{arg: } x_3}$	$\frac{l_5: \text{se}(x_4,x_5)}{\text{arg: } x_4, x_5}$	$\frac{l_6: \text{solution}(z)}{\text{arg: } -}$
		$\frac{l_7: \text{who}(x)}{\text{arg: } -}$

Figure 5: Semantic Representations

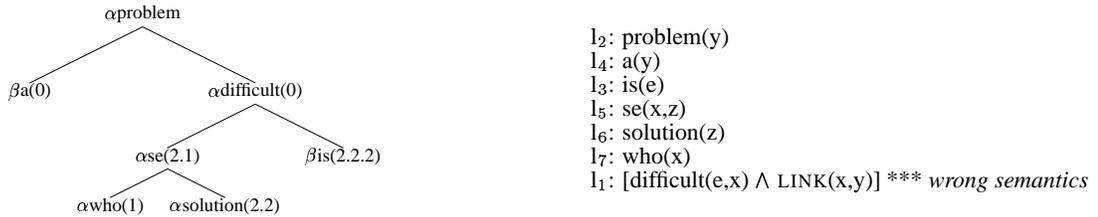


Figure 6: Derivation Tree and Semantic Composition

Changing the semantics for the relative clause as in Figure 7 will not help. *Difficult* is a predicate over the variable for the head noun, and so it will again derive the incorrect interpretation that the *problem*, and not the *solution*, is difficult.

$\frac{l_1: [\text{difficult}(e,x_2) \wedge \text{LINK}(x_1,x_2)]}{\text{arg: } x_1, x_2}$
--

Figure 7: Semantic Representation for a Relative Clause Anchoring *difficult*

3.3. Trial 3: A proposal

In order to derive the correct semantics for (2), *difficult* must be a predicate over a variable associated with *solution*. As a way of ensuring this, I define three argument variables for relative clauses: one for the *wh* relative pronoun, another for the head noun, and another for the head of NP[WH]. The semantics under this approach is given in Figure 8.

$\frac{l_1: [\text{difficult}(e, x_0) \wedge \text{LINK}(x_1, x_2)]}{\text{arg: } x_0, x_1, x_2}$	$\frac{l_2: \text{problem}(y)}{\text{arg: } -}$	$\frac{l_3: \text{is}(e_1)}{\text{arg: } e_1}$
$\frac{l_4: a(x_3)}{\text{arg: } x_3}$	$\frac{l_5: \text{se}(x_4, x_5)}{\text{arg: } x_4, x_5}$	$\frac{l_6: \text{solution}(z)}{\text{arg: } -}$
		$\frac{l_7: \text{who}(x)}{\text{arg: } -}$

Figure 8: Semantic Representations

In the semantics for the relative clause anchoring *difficult*, three argument variables are defined: x_0 must unify with the variable contributed by the head of NP[WH] (i.e., *solution*), x_1 must unify with the variable contributed by the *wh*-word (i.e., *who*), and x_2 must unify with the variable contributed by the head noun (i.e., *problem*). The semantic composition under this approach is given in Figure 9. This semantics correctly derives the meaning of the relative clause in (2): the *solution* is difficult, and this *solution* is in a possession relation with *problem*, as forced by the LINK operation unifying the variables for *who* and *problem*.

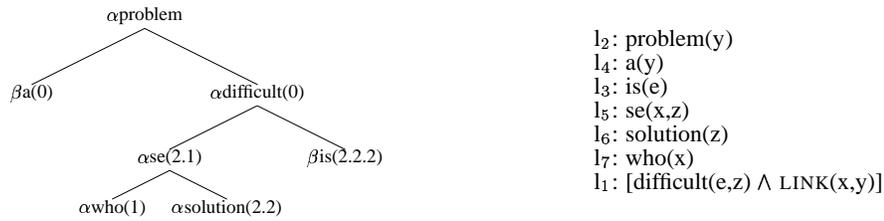


Figure 9: Derivation Tree and Semantic Composition

4. Generalizing

In this section, we will see how the proposed analysis can be generalized to relative clauses whose relative pronoun is the head of NP[WH], adjunct relative clauses, and relative clauses containing a recursive genitive NP[WH].

4.1. Relative clauses whose relative pronoun is the head of NP[WH]

The proposed approach straightforwardly extends to the simple case where the relative clause contains a relative pronoun which is the head of the NP[WH], as in (4).

(4) The solution which is difficult

The semantics for the elementary trees are given in Figure 10, and the derivation tree and the corresponding semantic composition are given in Figure 11. In this case, since the head of NP[WH] is the relative pronoun itself, x_0 for the relative pronoun and x_1 for the head of NP[WH] in the relative clause tree semantics both unify with the variable x from the relative pronoun tree. By the LINK operation, x is unified with the variable from *solution*, giving us the correct interpretation that it is the solution that is difficult.

$\frac{l_1: [\text{difficult}(e, x_0) \wedge \text{LINK}(x_1, x_2)]}{\text{arg: } x_0, x_1, x_2}$	$\frac{l_2: \text{solution}(y)}{\text{arg: } -}$
$\frac{l_3: \text{is}(e_1)}{\text{arg: } e_1}$	$\frac{l_4: a(x_3)}{\text{arg: } x_3}$
	$\frac{l_5: \text{which}(x)}{\text{arg: } -}$

Figure 10: Semantic Representations

4.2. Adjunct relative clauses

We now discuss how the proposed analysis can be extended to handle the semantics of adjunct relative clauses as in (5). We will consider two possible approaches: (i) an approach based on the assumption that the adjunct



Figure 11: Derivation Tree and Semantic Composition

phrase *in which* substitutes into the relative clause tree; and (ii) an approach based on the assumption that the adjunct phrase adjoins onto the relative clause tree.

- (5) The place in which John lives is expensive.

4.2.1. Substitution approach

Under the substitution approach, the elementary tree for the adjunct relative clause anchoring *lives* has two substitution sites: one for the subject NP and the other for the PP that will contain the relative pronoun in the course of the derivation. The corresponding semantic representation is given in the first box in Figure 12. Here, *lives* takes an event argument variable (*e*) and a variable (*x₀*) for the subject. Further, the variable for the *wh* relative pronoun (*x₁*) and the variable for the head noun (*x₂*) are forced to unify by the LINK operation as before. The derivation requires a PP initial tree anchoring *in*. The semantics for this tree is given in *l₅* in Figure 12: *in* is a predicate taking an event variable, and another variable for the substituting NP. Substituting this PP into the relative clause tree will allow the event variable from the PP tree to unify with the event variable from the relative clause tree. This will have the interpretive effect that the PP is modifying the verb *lives*.

The derivation tree and the corresponding semantic composition for (5) are given in Figure 13. We correctly end up with the interpretation that the place is expensive, and John lives in this place.

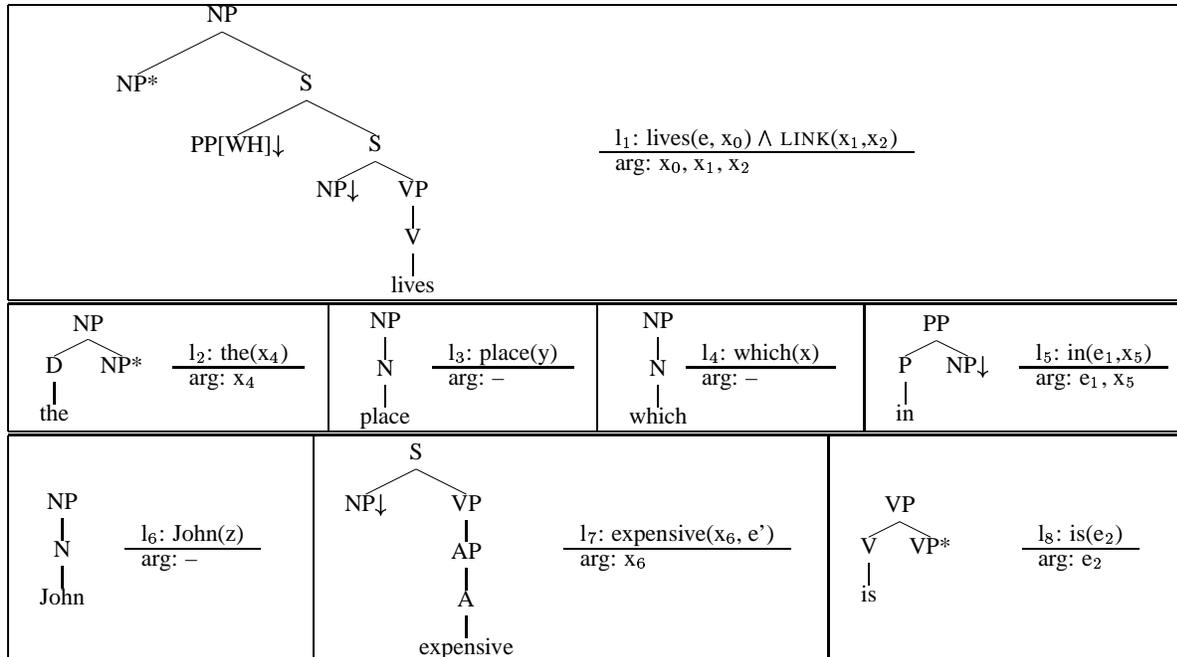


Figure 12: Elementary Trees and Semantic Representations

4.2.2. Adjunction approach

Under the assumption that adjunct phrase *in which* is adjoined to the relative clause tree, the elementary tree and the corresponding semantic representation for the adjunct relative clause tree anchoring *lives* can be specified

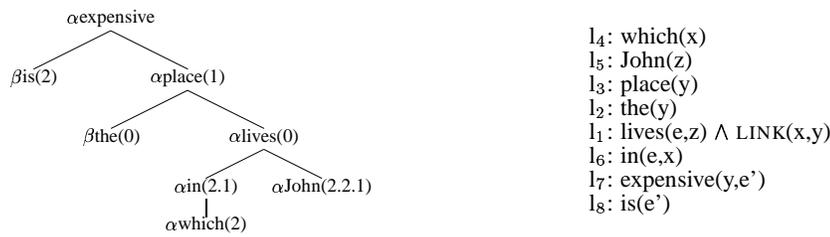


Figure 13: Derivation Tree and Semantic Composition

as in the first box in Figure 14. The semantics for the adjunct relative clause is as before: *lives* takes an event argument variable (e) and a variable (x_0) which will unify with the subject, and the variable for the head noun (x_2) is forced to be unified with the variable for the *wh* relative pronoun (x_1). Although there is no syntactic position designated for a relative pronoun in the relative clause tree, we can motivate a variable for it with the assumption that the tree is encoded with a [WH] feature that requires a relative pronoun containing phrase to be adjoined onto the S node. Further, the derivation under the adjunction approach requires an S-rooted auxiliary tree anchoring *in*, which has an NP node that will be substituted with a *wh* relative pronoun. Its semantics is represented in l_6 in Figure 14: e_1 is an event argument variable that will unify with the event variable from the adjoining S, and x_5 will unify with the variable from the substituting NP. All other elementary trees and their semantics necessary for the derivation are as same as in Figure 12.

The derivation tree and the corresponding semantic composition for (5) are given in Figure 15. This results in the correct interpretation that the place is expensive and John lives in that place.

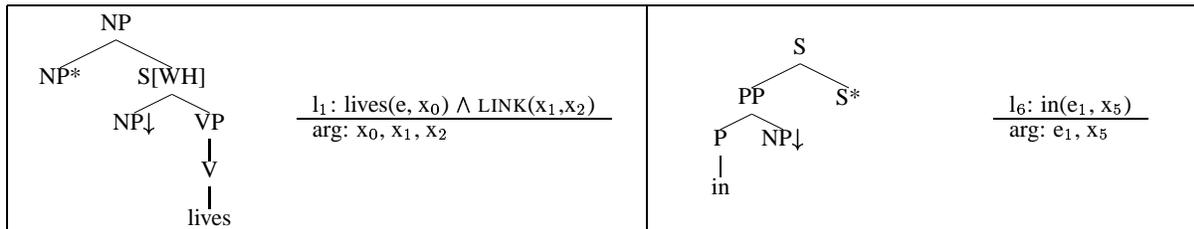


Figure 14: Elementary Trees and Semantic Representations

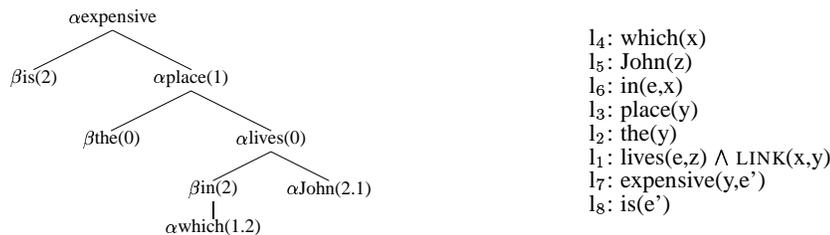


Figure 15: Derivation Tree and Semantic Composition

At the current stage of understanding, the adjunction approach seems to be preferable to the substitution approach. This is because under the substitution approach, adjunct PPs enter into the derivation through substitution. However, in all other cases, while adjunct PPs are represented with auxiliary trees that enter into the derivation through adjunction, argument PPs are represented with initial trees and enter into the derivation through substitution. The adjunction approach allows us to maintain this dichotomy between arguments and adjuncts.

4.3. Relative clauses containing a recursive genitive NP[WH]

In the derivation of relative clauses with a recursive genitive NP[WH] as in (6), each genitive contributes an elementary tree with two substitution sites. They are each associated with the semantic representation in which *se*

or 's is a predicate requiring an argument variable for the possessor and an argument variable for the possessee. These are represented as l_5 and l_9 in Figure 16.

Let's see what happens if we use the semantics given in Figure 16 and the derivation tree given in (17) to do the compositional semantics for (6). The semantics for the relative clause tree and other elementary trees are similar to the the ones we used in section 3.3. But now we have a problem. Although the resulting interpretation gets the right predicate/argument relation between *difficult* and *proof*, and the possession relation between *solution* and *proof* and *who* and *solution*, the variable for *who* cannot be unified with x_1 in the relative clause semantics. This is because *who* is deeply embedded and so its variable cannot pass all the way up to the relative clause semantics. Thus, there is no way to enforce the unification between the variable from *who* and x_1 , and the meaning that *problem* is the possessor of *solution* is lost and the meaning that the relative clause is the modifier of the head noun cannot be represented.

(6) A problem whose solution's proof is difficult

$l_1: [\text{difficult}(e, x_0) \wedge \text{LINK}(x_1, x_2)]$ arg: x_0, x_1, x_2	$l_2: \text{problem}(y)$ arg: -	$l_3: \text{is}(e_1)$ arg: e_1
$l_4: a(x_3)$ arg: x_3	$l_5: 's(x_4, x_5)$ arg: x_4, x_5	$l_6: \text{solution}(z)$ arg: -
$l_7: \text{who}(x)$ arg: -	$l_8: \text{proof}(v)$ arg: -	$l_9: \text{se}(x_6, x_7)$ arg: x_6, x_7

Figure 16: Semantic Representations

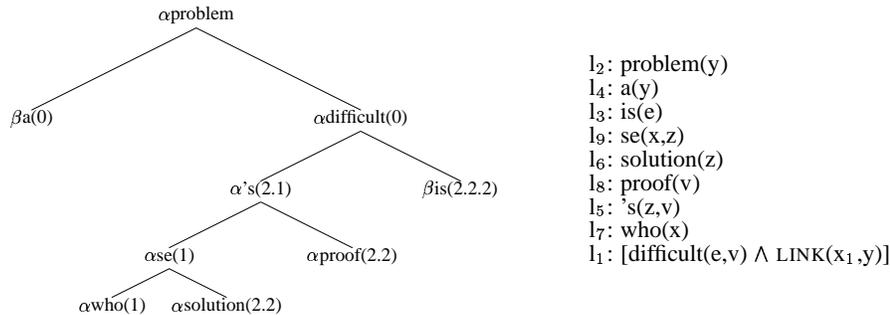


Figure 17: Derivation Tree and Semantic Composition

Here, I will sketch two possible approaches to address this problem: one is to exploit feature unification (Vijay-Shanker and Joshi, 1991), and the other is to use set-local multi-component TAG (MC-TAG) (Weir, 1988).

Under the feature unification approach, we need to make the assumption that a *wh* feature is encoded in relative pronoun trees as well as in relative clause trees, and that these features are syntactically constrained to be the same. This syntactic constraint is instantiated as the semantic constraint that the variable for the *wh* relative pronoun in the semantics of the relative clause tree and the variable in the semantics of the relative pronoun tree be the same.

The semantics for the relative clause anchoring *difficult* now looks as in the first box in Figure 18. x_0 will unify with the variable from *proof*, the head of NP[WH], and x_2 will unify with the variable from the head noun *problem*. And x_w is the variable for the relative pronoun, which is motivated by the *wh* feature encoded in the relative clause tree. This feature is syntactically constrained to be the same as the feature on the relative pronoun tree. This means that semantically, the relative pronoun *who* contributes the same variable, x_w .

$l_1: [\text{difficult}(e, x_0) \wedge \text{LINK}(x_w, x_2)]$ arg: x_0, x_2	$l_7: \text{who}(x_w)$ arg: -
---	----------------------------------

Figure 18: Modified Semantics for Relative Clause and Relative Pronoun Trees

The semantic composition using these semantics will give us the correct interpretation, as shown in Figure 19: *problem* is the possessor of *solution*, *solution* is the possessor of *proof* and *proof* is *difficult*.

$l_2: \text{problem}(y)$
 $l_4: a(y)$
 $l_3: \text{is}(e)$
 $l_9: \text{se}(z, v)$
 $l_6: \text{solution}(z)$
 $l_8: \text{proof}(v)$
 $l_5: 's(x_w, z)$
 $l_7: \text{who}(x_w)$
 $l_1: \text{difficult}(e, v) \wedge \text{LINK}(y, x_w)$

Figure 19: Semantic Composition

Under the set-local approach, we need to assume three sets of trees as shown in Figure 20. One set contains an NP tree anchoring *who* and a degenerate auxiliary tree S*, another set contains a relative clause tree and an NP tree anchoring 's, and the other set contains NP trees anchoring *se* and *problem* respectively. The first set is for the relative pronoun and can be naturally motivated: the NP tree anchoring *who* corresponds to the contribution of *who* to the predicate/argument structure, and S* contributes to the scope of *who*. The other two sets, however, are not a linguistically natural set, although it will be shown that postulation of these sets are necessary in resolving our problem.

The syntactic derivation will proceed as follows: S* adjoins to S₁ in the relative clause tree, and NP anchoring *who* substitutes into the specifier of *se* tree. And *solution* tree substitutes into the complement of *se* tree, which will substitute into the specifier of 's tree. The complement of 's tree is substituted with *proof* tree. And then 's tree substitutes into NP[WH] node of the relative clause tree. The derivation tree is given in Figure 21

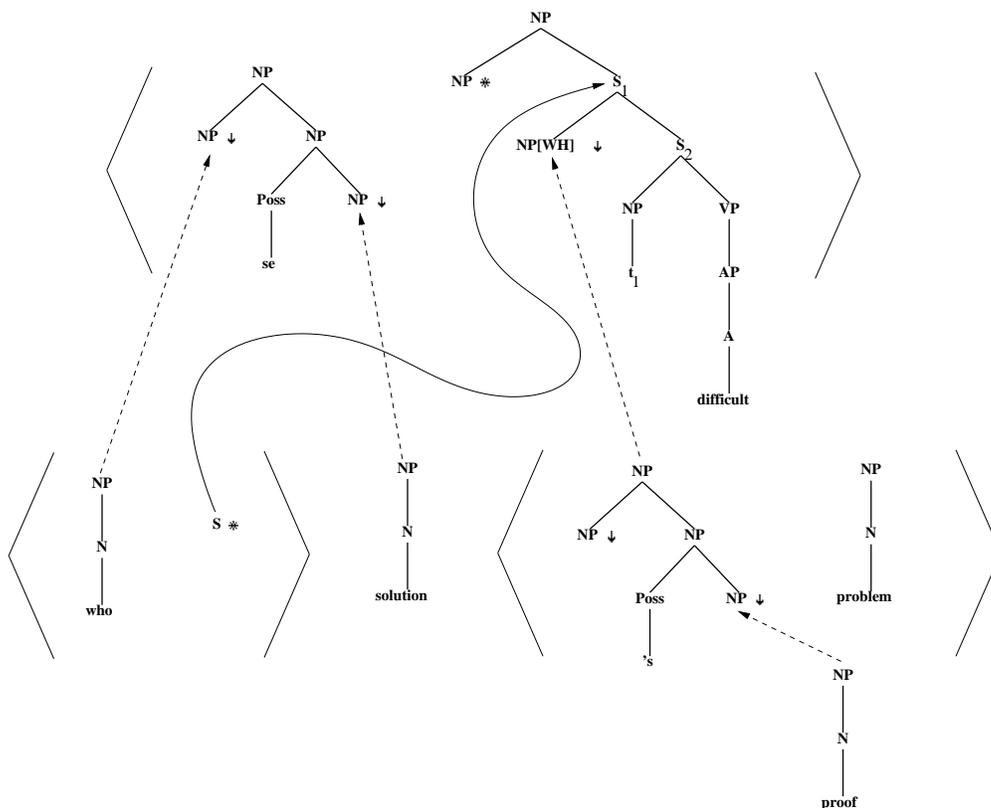


Figure 20: Derivation in Set-Local MC-TAG

The only new thing we need to do for semantics is to redefine the semantics for *who*, as in Figure 22, and the rest will look exactly the same as in Figure 16. The semantics in $l_{7[1]}$ is for the elementary tree anchoring *who*, and the semantics in $l_{7[2]}$ is for the degenerate S* tree. The variable from $l_{7[1]}$ will unify with the variable for the

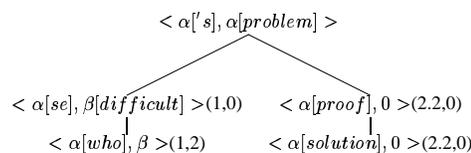


Figure 21: Derivation tree



Figure 22: Modified Semantics for Relative Pronoun Trees

possessor in l_9 , and the variable from $l_{7[2]}$ will unify with the variable for the *wh* relative pronoun in l_1 . This has the desirable result that *who* is the possessor of *solution* and that the relative clause is the modifier of the head noun *proof*.

While both feature unification and set-local approaches give us the correct semantics, there are problems with both. In feature unification approach, the variable for *who* ends up being LINKED to the variable for *problem*, not through a direct variable unification, but because the *wh* features encoded in the relative clause elementary tree and in the relative pronoun tree are stipulated to translate to the same variable, x_w . In the set-local approach, variable unification in semantics works without resorting to any stipulation, but the cost to syntax is too much. From an implementational point of view, it seems that feature unification approach is preferable, given its relative simplicity.

5. Conclusion

I have shown that an LTAG-based compositional semantics for relative clauses can be done by defining three argument variables for the semantics of relative clause elementary trees: one for the *wh* relative pronoun, one for the head of NP[WH] and the other for the head noun. I have introduced an operator, LINK, that forces variable unification between the *wh* relative pronoun and the head noun. We have seen that the proposed analysis handles relative clauses with a simple relative pronoun as well as those with a relative pronoun in pied-piping structure, and adjunct relative clauses. I have also pointed out a potential problem in variable unification in relative clauses with a deeply embedded relative pronoun, and suggested two possible ways of addressing this problem: exploiting feature unification and using set-local MC-TAG. All this ensures the unification between the variables from the head noun and the relative pronoun, no matter how deeply embedded the relative pronoun is, deriving the desirable predicate/argument relations and the interpretation that the relative clause is a modifier of the head noun. It remains to be seen how the proposed analysis can be extended to relative clauses with long distance relativization (e.g., *the solution which John said Mary thinks is difficult*).

References

- Abeillé, Ann. 1994. Syntax or semantics? Handling nonlocal dependencies with MCTAGs or synchronous TAGs. *Computational Intelligence*.
- Heim, Irene and Angelika Kratzer. 1998. *Semantics in Generative Grammar*. Blackwell Publishers.
- Joshi, Aravind K. and Laura Kallmeyer. 2000. Factoring predicate argument and scope semantics: underspecified semantics with LTAG. Ms. University of Pennsylvania and UFRL, University Paris 7.
- Joshi, Aravind K. and K. Vijay-Shanker. 1999. Compositional semantics with lexicalized tree-adjoining grammar (LTAG): How much underspecification is necessary? In *Proceedings of the 3rd International Workshop on Computational Semantics*, pages 131–145, The Netherlands.
- Schabes, Yves and Stuart M. Shieber. 1994. An Alternative Conception of Tree-Adjoining Derivation. *Computational Linguistics*, 20(1):91–124.
- The XTAG-Group. 2001. A Lexicalized Tree Adjoining Grammar for English. Technical Report IRCS 01-03, University of Pennsylvania.
- Vijay-Shanker, K. and Aravind Joshi. 1991. Unification Based Tree Adjoining Grammars. In *Unification-based Grammars*. MIT Press, Cambridge, Massachusetts.
- Weir, D. 1988. *Characterizing mildly context-sensitive grammar formalisms*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA, August.