LTAG Semantics for Relative Clauses

Laura Kallmeyer
TALaNa/LaTTICe, University Paris 7,
2 place Jussieu, 75251 Paris Cedex 05, France.
laura.kallmeyer@linguist.jussieu.fr

Abstract

It has been shown that the standard syntactic analysis for relative clauses in LTAG (Lexicalized Tree Adjoining Grammar) is problematic for semantics since it does not allow to obtain the correct predicate-argument dependencies. Alternative proposals that try to keep as much of the standard analysis as possible are not entirely satisfying. Therefore, this paper presents a semantic analysis of relative clauses based on the alternative syntactic approach in Kahane et al. (2000) that considers the relative pronoun as the element that introduces the relative clause, i.e., as the element linked to the NP modified by the relative clause. The semantics used is the LTAG semantics proposed in Kallmeyer and Joshi (2002).

1 The data

The data considered in this paper are restrictive relative clauses showing some kind of long extraction. These are the data that are problematic for the standard LTAG analysis (as shown in Kahane et al. 2000; Han 2002a). (1)–(3) are examples of extractions out of complex NPs, (4) is a long-distance movement out of an embedded complement clause and the last examples concern extractions out of adjuncts:

(1) a. a problem whose solution is difficult (Han 2002a)

b. a problem whose solution's proof is difficult (Han 2002a)

(2) a girl to the mother of whom Peter talks (Kahane et al. 2000)

(3) a. a girl who Peter painted (a copy of) a picture of

b. a girl a picture of whom Peter painted a copy of

(4) a girl who Peter thinks John says Paul likes

(5) a. a house in which Mary lives

b. a house in whose courtyard Mary lives

c. a house which Mary lives in

The aim of this paper is to show that a semantic analysis of these data is possible within the LTAG syntax-semantics interface introduced in Kallmeyer and Joshi (2002) if the syntactic analysis proposed in Kahane et al. (2000) is adopted. Section 2 introduces LTAG and the syntax-semantics interface. The following section explains why the standard syntactic TAG analysis for relative clauses is problematic for semantics and it discusses some alternatives. Section 4, the main part of the paper, shows that an analysis considering the relative pronoun as the element introducing the relative clause as in Kahane et al. (2000) allows to obtain correct semantic analyses. In this section, the examples listed above are discussed in detail.

2 TAG and the syntax-semantics interface

See also Joshi et al. (2003) in the same volume for a more detailed introduction into LTAG semantics.

2.1 Lexicalized Tree Adjoining Grammars (LTAG)

An LTAG (Joshi and Schabes 1997) consists of a finite set of trees (elementary trees) associated with lexical items and of composition operations of substitution (replacing a leaf with a new tree) and adjunction (replacing an internal node with a new tree). The elementary trees represent extended projections of lexical items and encapsulate all syntactic/semantic arguments of the lexical anchor. They are minimal in the sense that only the arguments of the anchor are encapsulated, all recursion is factored away.

LTAG derivations are represented by derivation trees that record how the elementary trees are put together. A derived tree is the result of carrying out the substitutions and adjoinings. For a sample derivation see the TAG analysis of *John always loves Mary* in Fig. 1. The derivation starts with the elementary tree for *loves*. Therefore, the name of this tree is the root of the derivation tree. Its daughters are all the elementary trees immediately added (by subtitution of adjunction) to the tree of *loves*. The numbers are the node positions where the different trees are added: *John* is substituted for the node at position (1), *Mary* for the node at position (22) and *always* is adjoined to the node at position (2).

2.2 Compositional semantics with LTAG

Because of the localization of the arguments of a lexical item within elementary trees, TAG derivation trees express predicate argument dependencies. Therefore it is generally assumed that the proper way to define compositional semantics for LTAG is with respect to the derivation tree, rather than the derived tree (see, e.g., Shieber and Schabes 1990; Candito and Kahane 1998; Joshi and Vijay-Shanker 1999; Kallmeyer and Joshi 1999, 2002).

The overall idea is as follows. Each elementary tree is linked to a semantic representation. The way the semantic representations combine with each other

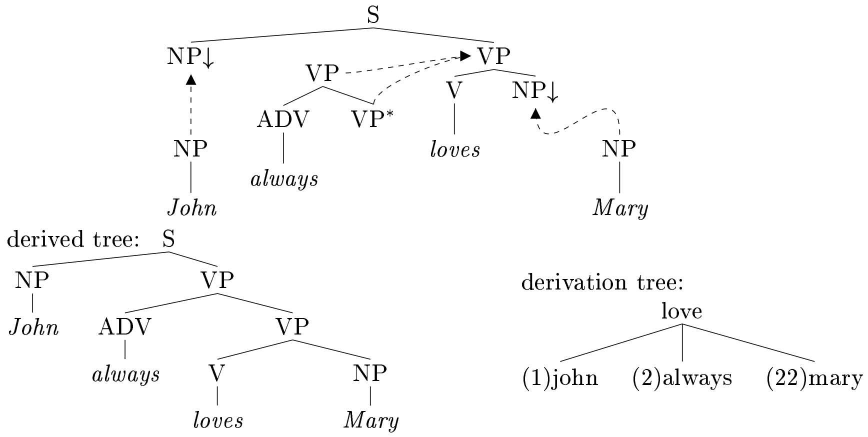


Figure 1: TAG derivation for *John always loves Mary*

depends on the derivation tree. Following Kallmeyer and Joshi (1999, 2002), in this paper, I adopt 'flat' semantic representations as in, for example, Minimal Recursion Semantics (MRS, Copestake et al., 1999). (6) shows the elementary semantic representations for *John always loves Mary.*

(6)

$l_1 : \mathsf{love}(x_1, x_2)$ $h_1 \geq l_1$	$\mathsf{john}(x)$	$l_2 : \mathsf{always}(h_2)$ $g_1 \geq l_2, h_2 \geq s_1$	$\mathsf{mary}(y)$
arg: $\langle x_1, (1)\rangle, \langle x_2, (22)\rangle$	arg: –	arg: g_1, s_1	arg: –

Roughly, a semantic representation consists of a conjunctively interpreted set of formulas (typed lambda-expressions), scope constraints and a set of argument variables. The formulas may contain labels and holes (metavariables for propositional labels). In the following, $l_1, l_2, \ldots$ are propositional labels, $h_1, h_2, \ldots$ are propositional holes, $s_1, s_2, \ldots$ are propositional and $x_1, x_2, \ldots$ individual argument variables (whose values must be propositional labels/free individual variables) and $g_1, g_2, \ldots$ are hole variables (special argument variables whose values must be holes). Argument variables may be linked to positions in the elementary tree, as it is the case for x_1 and x_2 in (6).

The use of holes is motivated by the desire to generate underspecified representations (as in, e.g., Bos, 1995) for scope ambiguities though this is not subject of this paper. After having constructed a semantic representation with holes and labels, disambiguation is done which consists of finding surjective total functions from holes to labels that respect the scope constraints.[1] In the semantic representation for *love*, there is for example a hole h_1 above the label l_1 (indicated by the constraint $h_1 \geq l_1$). Between h_1 and l_1, other labels and

[1]I am using surjections instead of bijections as in Bos (1995) since sometimes (e.g., in the semantic representations of relative pronouns) there are more holes than labels. But I am still assuming that each hole is mapped to a single label.

holes might come in (introduced for example by quantifiers or adverbs) or, if this is not the case, l_1 will be assigned to h_1 in the disambiguation(s).

When combining semantic representations, values are assigned to argument variables and the union of the semantic representations is built. (This corresponds more or less to a conjunction.) The values for the argument variables of a certain (elementary) semantic representation must come from semantic representations that are linked to it in the derivation tree.

The linking of argument variables and syntactic positions restricts the possible values as follows: In a substitution derivation step at a position p, only argument variables linked to p get values. In an adjunction step, only argument variables that are not linked to any positions can get values. In the case of a substitution, a new argument is inserted and therefore a value is assigned to an argument variable in the old semantic representation. However, in the case of an adjunction, a new modifier is applied and therefore a value is assigned to a variable in the semantic representation that is added.

The predicate always in (6) takes a propositional argument h_2 (its scope) that contains at least the proposition the adverb is added to (this is expressed by the constraint $h_2 \geq s_1$). Furthermore, the scope of he adverb is limited by the hole of the modified proposition (constraint $g_1 \geq l_2$). This prevents for example adverbs in embedded clauses from taking scope outside their clause.

The derivation tree in Fig. 1 indicates that the value of x_1 needs to come from the semantic representation of *John*, the one of x_2 from *Mary* and the values of g_1 and s_1 need to come from *love*. Consequently, $x_1 \to x, x_2 \to y, g_1 \to h_1$ and $s_1 \to l_1$. The result is (7).

(7)
$$\boxed{\begin{array}{l} l_1 : \mathsf{love}(x, y),\ \mathsf{john}(x),\ \mathsf{mary}(y),\ l_2 : \mathsf{always}(h_2) \\ h_1 \geq l_1, h_1 \geq l_2, h_2 \geq l_1 \\ \hline \text{arg: } - \end{array}}$$

According to (7), $h_1 \geq l_2$, $l_2 > h_2$ (because h_2 appears inside a formula labelled l_2) and $h_2 \geq l_1$. Consequently $h_1 \neq l_1$ and therefore the only possible disambiguation is $h_1 \to l_2, h_2 \to l_1$. This leads to the semantics $\mathsf{john}(x) \wedge \mathsf{mary}(y) \wedge \mathsf{always}(\mathsf{love}(x, y))$.

For a formalization of the syntax-semantics interface presented here see Kallmeyer and Joshi (2002), and see also Joshi et al. (2003) for some more details.

2.3 Separating scope and predicate argument information

A central aspect of (Kallmeyer and Joshi, 1999, 2002) is the idea that the contribution of a quantifier is separated into a scope and a predicate argument part: Quantifiers have a set of two elementary trees and multicomponent TAGs are used. An auxiliary tree consisting of a single node is linked to the scope part of the semantics, while an initial tree is linked to the predicate argument part. E.g., consider the analysis of *every dog barks* shown in Fig. 2. The corresponding elementary semantic representations are shown in (8).

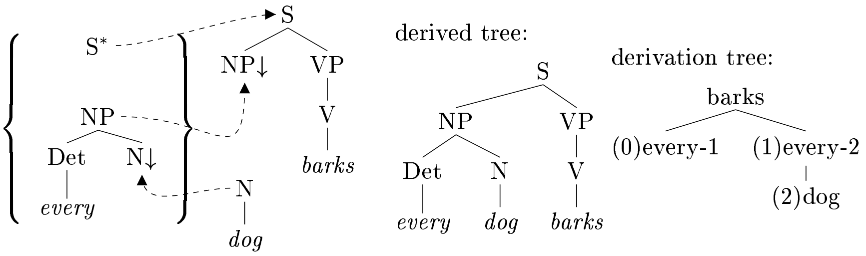


Figure 2: Syntactic analysis of *every dog barks*

(8) $\begin{array}{|l|}\hline l_1 : \mathsf{bark}(x_1) \\ h_1 \geq l_1 \\ \hline \text{arg: } x_1 \\ \hline \end{array}$ $\begin{array}{|l|}\hline l_2 : \mathsf{every}(x, h_2, h_3) \\ h_3 \geq s_1 \\ \hline \text{arg: } s_1 \\ \hline \end{array}$ $\begin{array}{|l|}\hline l_3 : p_1(x) \\ h_2 \geq l_3 \\ \hline \text{arg: } p_1 \\ \hline \end{array}$ $\begin{array}{|l|}\hline q_1 : \mathsf{dog} \\ \hline \text{arg: } - \\ \hline \end{array}$

The scope part of the quantifier (second representation in (8)) introduces a proposition containing the quantifier, its variable and two holes for its restrictive and nuclear scope. The proposition this semantic representation is applied to (variable s_1) is in the nuclear scope of the quantifier ($h_3 \geq s_1$). The predicate argument part (third representation in (8)) introduces a proposition $p_1(x)$ where p_1 will be the noun predicate dog. This proposition is in the restrictive scope of the quantifier ($h_2 \geq l_3$). The values for the argument variables are $x_1 \to x, s_1 \to l_1, p_1 \to q_1$ which leads to (9).

(9) $\begin{array}{|l|}\hline l_1 : \mathsf{bark}(x),\, l_2 : \mathsf{every}(x, h_2, h_3),\, l_3 : \mathsf{dog}(x),\, h_1 \geq l_1,\, h_3 \geq l_1,\, h_2 \geq l_3 \\ \hline \text{arg: } - \\ \hline \end{array}$

The only disambiguation is $h_1 \to l_2, h_2 \to l_3, h_3 \to l_1$ which leads to the semantics $\mathsf{every}(x, \mathsf{dog}(x), \mathsf{bark}(x))$.

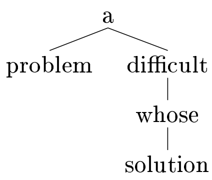
It is not necessarily the case that the hole of the verb (here h_1) is above or equal the label of the quantifier (here l_2). Quantifiers inside complement clauses for example can take scope outside the complement clause they are embedded in. In this case they end up higher than the hole of the embedded verb. In contrast to this, the scope of adverbs is bound by the hole of the clause they adjoin to (see the constraint $g_1 \geq l_2$ in the semantics of *always* in (6)).

To account for cases with more than one quantifier, a restricted use of multiple adjunctions (for the scope parts) is necessary.

3 Previous LTAG approaches to relative clauses

3.1 The standard analysis

In standard TAG (XTAG Research Group 1998), relative clauses are NP-modifiers, i.e., they are adjoined to the root node of the modified NP, and the anchor of a relative clause is its verb. Assuming that NPs are treated as DPs in



$l_1 : \mathsf{a}(x, \mathsf{problem}(x) \wedge l_3, \ldots)$

$l_2 : \mathsf{difficult}(y)$

$l_3 : \mathsf{rel}(y, \mathsf{solution}(y, x), l_2)$

Figure 3: Derivation tree of standard analysis of (1)a. and desired semantics

the sense that they consist of an initial tree for the determiner with a substitution node for the noun, the standard analysis of (1)a. *a problem whose solution is difficult* for example leads to the derivation tree in Fig.3. The semantics one wants to obtain is sketched on the right of the tree.

The relative pronoun introduces a variable (y) that is an argument of $\mathsf{difficult}$ while linking y to the variable x corresponding to the modified NP, i.e., to the variable introduced by the determiner. The problem is that the derivation tree does not show a link between the relative pronoun and the determiner of the modified NP. Therefore, in a semantics strictly based on the derivation tree, the relative pronoun does not have access to the NP variable.

Alternatively, one might assume that the noun introduces a variable (here, x could be introduced by *problem*). But the problem would be the same: the anchor of the relative clause (here *difficult*) would intervene between the NP variable and the relative pronoun in the derivation tree.

This problem is even worse with extractions out of more embedded structures as in (1)b. or out of discontinuous constituents as in (3).

3.2 Alternative approaches

Han (2002a) proposes to use a variable passing mechanism in order to obtain a link between two elementary trees that are not direct neighbours in the derivation tree. This means however a certain lack of locality in semantics and, as a result, parsing would be harder. Furthermore, Han (2002a) considers only extraction out of genitive NPs as in (1) and it is not clear how the other examples mentioned in this paper can be treated in her approach.

In order to avoid the additional variable passing mechanism, Han (2002b) discusses alternative elementary trees for relative clauses: The elementary trees contain the verb of the relative clause and the relative pronoun. However, this analysis contradicts the principle of semantic minimality, one of the guiding linguistic principles of LTAG.

In order to obtain a derivation tree that reflects predicate-argument dependencies, Kahane et al. (2000) propose an alternative syntactic analysis for relative clauses that respects the principle of semantic minimality since there are separate elementary trees for the verb and the relative pronoun. This analysis considers the relative pronoun as the element that introduces the relative clause or, in other words, that links the relative clause to the modified NP.

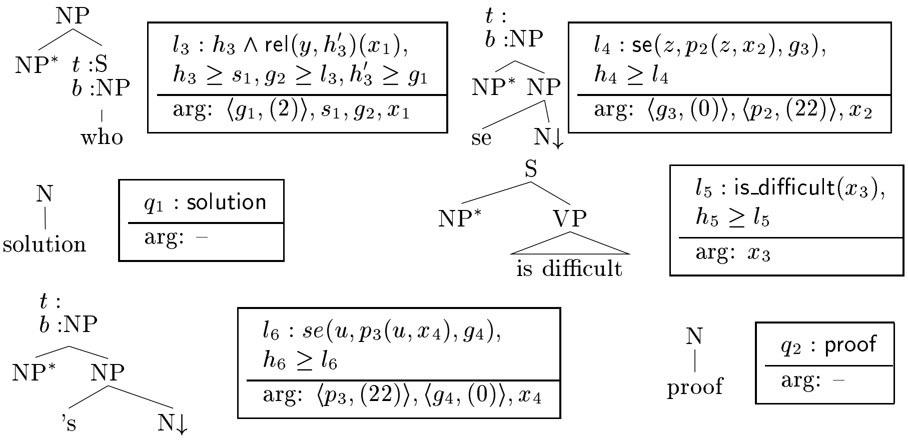


Figure 4: elementary trees and semantic representations for (1)a. and b.

The elementary tree of a relative pronoun is an NP-modifier tree to which the elementary tree of the verb of the relative clause is adjoined.

In this paper, I will adopt the syntactic analysis of Kahane et al. (2000) and I will integrate it into the syntax-semantic interface sketched above. In the following, I will present analyses of the data listed in Section 1.

4 The analysis

The semantic representation of the quantifier *a* (its lower part) is shown in (10). This is linked to the NP tree to which later the relative pronoun will be adjoined. The value of p_1 is the head noun of the modified NP (in the case of (1) this is $\mathsf{problem}$). h_1 is the hole that marks the restrictive scope of the quantifier.

(10)

$l_2 : p_1(x),\ h_1 \geq l_2$
arg: p_1

4.1 Extractions out of complex NPs

The elementary trees and semantic representations for (1)a. and b. are shown in Fig. 4 and their derivation trees can be found in Fig. 5. I assume a feature structure based TAG (FTAG, Vijay-Shanker and Joshi 1988) and $t : \ldots$ and $b : \ldots$ are the top and bottom feature structures of a node.

The semantics of the relative pronoun is the conjunction (intersection) of the noun predicate applied to the NP variable (here $\mathsf{problem}(x)$) which will be below h_3 and the predicate given by the relative clause applied to x. This last predicate is $\mathsf{rel}(y, h'_3)$ where semantically, rel is a λ abstraction, i.e., the interpretation of $\mathsf{rel}(y, h'_3)$ is the interpretation of $\lambda y(h'_3)$. Instead of using directly $\lambda y(h'_3)$ I prefer however to use $\mathsf{rel}(y, h'_3)$ in order to be able to treat y syntactically as a free variable much in the same way as the variables introduced by quantifiers. The fact that y syntactically is a free variable makes it available as possible value for argument variables when doing semantic composition.

The contribution of *whose* is separated into two parts: the relative pronoun *who* and the genitive *se* that introduces a new variable z. z is restricted by some relation p_2 (which will be $\mathsf{solution}$) that links it to the variable y of the relative pronoun. Furthermore, some proposition g_3 (e.g., $\mathsf{is\_difficult}(z)$ in (1)a.) holds for z. This captures the parallelism between the *se* of *whose* and the genitive *'s* in *solution's proof* which has the same semantics.

In the following, I adopt a slight modification of the possibilities to link argument variables to node positions: besides the possibility to link argument variables to substitution nodes, one can also link argument variables to internal nodes (i.e., nodes where adjunctions might take place). In this case, the variable receives its value from the elementary tree adjoined at that internal node. This is why the value of g_1 (linked to position (2) in the *who* tree) must come from the *se* tree adjoined at that position.

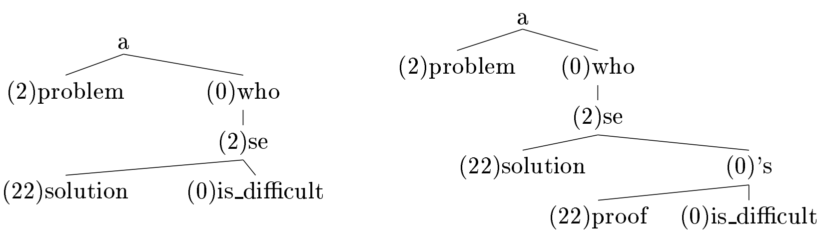


Figure 5: derivation trees of (1)a. and b.

For (1)a., the assignments are $x_1 \rightarrow x, g_1 \rightarrow h_4, s_1 \rightarrow l_2, g_2 \rightarrow h_1, p_2 \rightarrow q_1, g_3 \rightarrow h_5, x_2 \rightarrow y, x_3 \rightarrow z$. This leads to (11).

(11)

$l_2 : \mathsf{problem}(x)$, $l_3 : h_3 \wedge \mathsf{rel}(y, h_3')(x)$, $l_4 : \mathsf{se}(z, \mathsf{solution}(z, y), h_5)$, $l_5 : \mathsf{is\_difficult}(y)$ $h_1 \geq l_2$, $h_3 \geq l_2, h_1 \geq l_3$, $h_3' \geq h_4$, $h_4 \geq l_4$, $h_5 \geq l_5$
arg: –

According to this representation, $h_1 \geq l_3 > h_3 \geq l_2$ and $h_1 \geq l_3 > h_3' \geq h_4 \geq l_4 > h_5 \geq l_5$. Furthermore, $h_3 \not\geq h_3'$ and $h_3' \not\geq h_3$ since the two holes occur in different arguments of a binary predicate (of $\wedge$). Consequently, there is only one possible disambiguation, namely $h_1 \rightarrow l_3, h_3 \rightarrow l_2, h_3' \rightarrow l_4, h_4 \rightarrow l_4, h_5 \rightarrow l_5$. This leads to the semantics $\mathsf{problem}(x) \wedge \mathsf{rel}(y, \mathsf{se}(z, \mathsf{solution}(z, y), \mathsf{is\_difficult}(z)))(x)$ for the restriction of the quantifier a.

I leave it to the reader to verify that the analysis of (1)b. works as well.

Note that *solution* and *proof* were analyzed as two-place predicates, i.e., the problem/solution were considered being an argument of solution/proof respectively. Cases of unary predicate as *whose house* need a slightly different analysis but do not pose a problem.

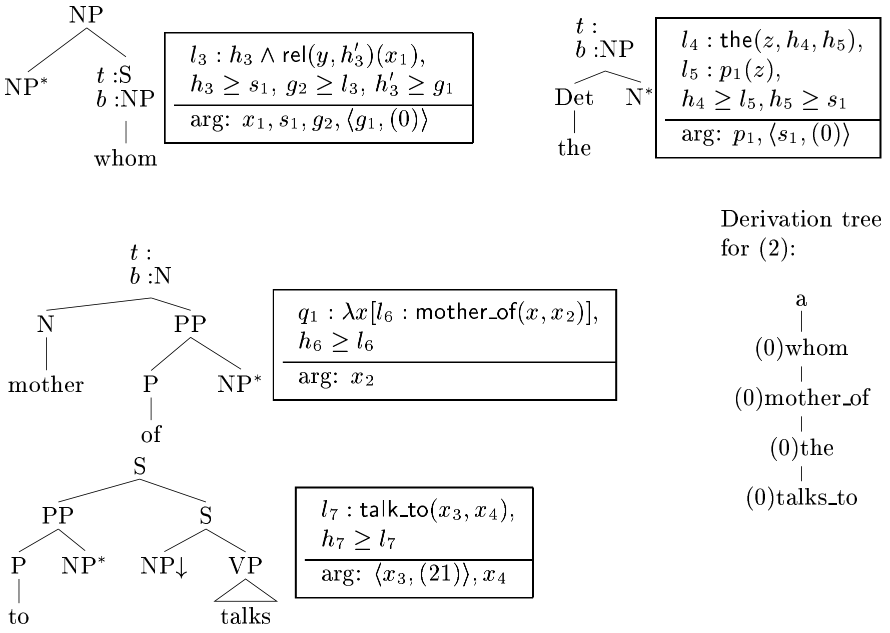


Figure 6: elementary trees, semantic representations and derivation tree for (2)

Next, consider example (2) *a girl to the mother of whom Peter talks.* The derivation is shown in Fig. 6. The assignments are $x_1 \to x, s_1 \to l_2, g_2 \to h_1, g_1 \to h_6, x_2 \to y, p_1 \to q_1, s_1 \to l_7, x_4 \to z$. This leads to the (12) which shows the correct predicate argument relations.

(12)
$$
\begin{array}{|l|}
\hline
l_2 : \mathsf{girl}(x),\ l_3 : h_3 \wedge \mathsf{rel}(y, h_3')(x),\ l_4 : \mathsf{the}(z, h_4, h_5), \\
l_5 : l_6 : \mathsf{mother\_of}(z, y),\ l_7 : \mathsf{talk\_to}(x_3, z), \\
h_1 \geq l_2, h_3 \geq l_2, h_1 \geq l_3, h_3' \geq h_6, h_4 \geq l_5, h_5 \geq l_7, h_6 \geq l_6, h_7 \geq l_7 \\
\hline
\text{arg: } \langle x_3, (21) \rangle \\
\hline
\end{array}
$$

Another issue, however, is the position of the definite article the that is treated as a kind of quantifier. (I leave aside the fact that it carries a presupposition.) There are two possibilities: either $l_4 : \mathsf{the}(z, h_4, h_5)$ is part of the relative clause (i.e., below h_3') and in this case, $l_7 : \mathsf{talk\_to}(x_3, z)$ is also part of the relative clause since it is in the scope of the (below h_5). Or the takes scope outside the relative clause, and in this case $l_3 : h_3 \wedge \mathsf{rel}(y, h_3)(x)$ is part of the restriction while $l_7 : \mathsf{talk\_to}(x_3, z)$ is part of the nuclear scope of the. Consequently, in this case, $l_7 : \mathsf{talk\_to}(x_3, z)$ is outside the scope of the relative pronoun. In other words, one obtains either $\mathsf{girl}(x) \wedge \mathsf{rel}(y, \mathsf{the}(z, \mathsf{mother\text{-}of}(z, y), \mathsf{talk\text{-}to}(x_3, z)))(x)$ or $\mathsf{girl}(x) \wedge \mathsf{the}(z, \mathsf{rel}(y, \mathsf{mother\text{-}of}(z, y))(x), \mathsf{talk\text{-}to}(x_3, z))$. Since rel is a lambda abstractor, both semantics are equivalent. After β-reduction they both lead to $\mathsf{girl}(x) \wedge \mathsf{the}(z, \mathsf{mother\text{-}of}(z, x), \mathsf{talk\text{-}to}(x_3, z))$

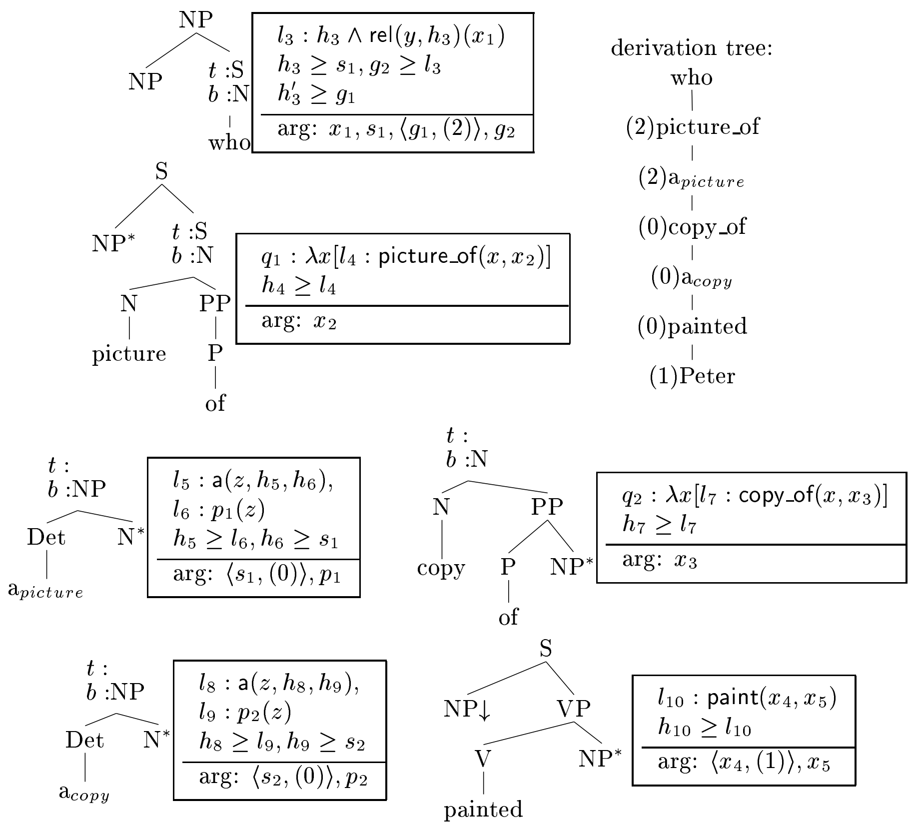


Figure 7: elementary trees and semantic representations for (3)a.

Next, consider (3)a. *a girl who Peter painted a copy of a picture of.* The derivation is shown in Fig. 7. It leads to the following assignments: $x_1 \to x, s_1 \to l_2, g_2 \to h_1, g_1 \to h_4, x_2 \to y, s_1 \to l_7, p_1 \to q_1, x_3 \to z, s_2 \to l_{10}, p_2 \to q_2, x_4 \to v$ (assuming that v is the variable of *Peter*), $x_5 \to u$. After β-reduction, the result is (13). Once β-reduction is done, l_6 and l_4 and also l_9 and l_7 label the same propositions.

(13)
$$\begin{array}{|l|}\hline l_2 : \mathsf{girl}(x),\ l_3 : h_3 \wedge \mathsf{rel}(y, h'_3)(x),\ l_5 : \mathsf{a}(z, h_5, h_6),\ l_6 : l_4 : \mathsf{picture\_of}(z, y), \\ l_8 : \mathsf{a}(u, h_8, h_9),\ l_9 : l_7 : \mathsf{copy\_of}(u, z),\ l_{10} : \mathsf{paint}(v, u),\ \mathsf{Peter}(v) \\ h_1 \geq l_2, h_3 \geq l_2, h_1 \geq l_3, h'_3 \geq h_4, h_4 \geq l_4, h_5 \geq l_6, h_6 \geq l_7, h_7 \geq l_7, \\ h_8 \geq l_9, h_9 \geq l_{10}, h_{10} \geq l_{10} \\ \hline \text{arg: } - \\ \hline \end{array}$$

According to this, the object u that is painted is the copy of some z that is the picture of some y, and y is the modified girl x. These are the correct predicate argument dependencies.

Here again, as in (12), the quantifiers occurring inside the relative clause can have wide scope and in this case, the propositions in their nuclear scopes are outside the scope of the relative pronoun as well.

(3)b. is very similar to (3)a. For (3)b., the elementary tree and the semantics of *picture of* would look like the one for *copy of* in Fig. 7. (3)c. is of the same form as (3)a. and can be treated in the same way.

4.2 Long-distance extraction out of embedded complement clauses

The derivation tree of (4) *a girl who Peter thinks John says Paul likes* is shown in Fig. 8. Obviously, the variable introduced by the relative pronoun can be used as value of an argument in the representation of *likes* since both are directly linked in the derivation tree. I.e., the correct predicate argument dependencies can be obtained. I omit details of the analysis.

However, here again, a part of the relative clause can be outside the scope of the relative operator: $\mathsf{say}(\mathsf{John}, \mathsf{like}(\mathsf{Paul}, y))$ must be in the scope of rel. But $\mathsf{think}(\mathsf{Peter}, \ldots)$ can have wide or narrow scope with respect to the relative operator. (Of course, it always has scope over its complement clause $\mathsf{say}(\ldots)$.)

(0)who
|
(2)likes
|
(2)says
|
(0)thinks

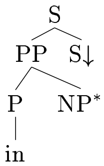
Figure 8: Derivation tree for (4)

4.3 Extraction out of adjuncts

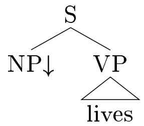
Next, we will analyse the examples (5) where the relative pronoun is part of an adjunct PP. These examples are obviously particularly problematic for an approach as in Han (2002b) where one single elementary tree contains the verb of the relative clause and the relative pronoun.

Consider only (5)b. *a house in whose courtyard Mary lives*. Fig. 9 shows the elementary trees for *in* and *likes* and their semantic representations. The combination of *a*, *house* and *whose* is like in the analysis of (1)a. (see Fig. 4 and 5). The tree for *in* adjoins to the root of the tree for *se* and then the *lives* tree substitutes into the *in* tree. The same elementary tree for *in* can be used for (5)a. For (5)c., the preposition would be on the right of the S substitution node. But the semantic representation would be the same. The tree and semantic representation for *lives* is the one of a simple matrix clause.

Here, *lives* introduces an event e that is modified by the *in* PP. x_4 becomes the variable introduced by *Mary*. The other assignments are $x_1 \rightarrow x, s_1 \rightarrow$



$$\begin{array}{|l|}\hline l_5 : \mathsf{in}(e_1, x_3), \\ h_5 \geq l_5, h_5 \geq g_4 \\ \hline \text{arg: } x_3, \langle e_1, (2)\rangle, \langle g_4, (2)\rangle \\ \hline \end{array}$$



$$\begin{array}{|l|}\hline l_6 : \mathsf{lives}(e, x_4), \\ h_6 \geq l_6 \\ \hline \text{arg: } \langle x_4, (1)\rangle \\ \hline \end{array}$$

Figure 9: elementary trees and semantic representations for (5)b.

$l_2, g_1 \to h_4, g_2 \to h_1, p_2 \to \mathsf{courtyard}, g_3 \to h_5, x_2 \to y, x_3 \to z, g_4 \to h_6$. This leads to

$$(14)\quad \begin{array}{|l|}\hline \mathsf{Mary}(u),\ l_2 : \mathsf{house}(x),\ l_3 : h_3 \wedge \mathsf{rel}(y, h_3')(x), \\ l_4 : \mathsf{se}(z, \mathsf{courtyard}(z, y), h_5),\ l_5 : \mathsf{in}(e, z),\ l_6 : \mathsf{lives}(e, u), \\ h_1 \geq l_2, h_3 \geq l_2, h_3' \geq h_4, h_1 \geq l_3, h_4 \geq l_4, h_5 \geq l_5, h_5 \geq h_6, h_6 \geq l_6 \\ \hline \text{arg: } - \\ \hline \end{array}$$

5 Conclusion

In this paper it has been shown that a semantic analysis of relative clauses in LTAG is possible using the Kahane et al. (2000) approach. This approach considers the relative pronoun to be the element that introduces the relative clause. Kahane et al. proposed this analysis in order to obtain correct predicate-argument dependencies for relative clauses but they did not propose a semantic analysis. In this paper, their proposal was pursued further and a concrete semantic analysis was worked out using the framework in (Kallmeyer and Joshi 1999, 2002).

The only issue is that sometimes parts of the syntactic relative clause end up outside of the scope of the relative operator. This occurs only in cases where these parts do not contain the variable introduced by the relative pronoun. But since the relative pronoun corresponds to a λ abstractor, i.e., $\mathsf{rel}(y, \phi)(x)$ is equivalent to $\phi[y/x]$ (ϕ with all occurrences of y replaced by x), the occurrence outside the relative scope is not problematic: for a ϕ_2 that does not contain any occurrences of y, $\mathsf{rel}(y, \phi_1 \wedge \phi_2)(x)$ is equivalent to $\mathsf{rel}(y, \phi_1)(x) \wedge \phi_2$.

Things are different of course for interrogatives. Here, placing a proposition outside or inside the scope of the wh-operator leads to different interpretations. Therefore one needs to find some way to ensure that everything that is syntactically part of the interrogative is semantically in the scope of the wh-element, i.e., part of the denotated question. One possibility to do this is to use an enriched derivation structure as basis for semantics as proposed in Kallmeyer (2002a,b).

A subject I did not say anything about are island constraints; it is often claimed that relative clauses are islands for quantifier scope. In the analysis proposed in this paper, this is not the case, i.e., quantifiers can take scope outside the relative clause they occur in. This might be appropriate since it is not clear so far whether relative clauses are really scope islands, sometimes it is also claimed that they are not. However, if relative clauses are scope islands, then perhaps a flexible composition approach similar to Joshi et al. (2003) might be used to obtain the desired constraints.

Acknowledgments

For valuable discussions of the problems this paper addresses and for many helpful comments I would like to thank Laurence Danlos, Chung-Hye Han, Aravind K. Joshi, Sylvain Kahane and Maribel Romero. I am also grateful to three anonymous reviewers whose comments helped considerably to clarify the paper.

References

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

Candito, M.-H. and Kahane, S.: 1998. Can the TAG Derivation Tree represent a Semantic Graph? an Answer in the Light of Meaning-Text Theory. *Fourth International Workshop on Tree Adjoining Grammars and Related Frameworks, IRCS Report 98–12*. University of Pennsylvania, Philadelphia. pp. 25–28.

Copestake, A., Flickinger, D., Sag, I. A. and Pollard, C.: 1999. Minimal Recursion Semantics. An Introduction. Manuscript, Stanford University.

Han, C.-H.: 2002a. Compositional Semantics for Relative Clauses in Lexicalized Tree Adjoining Grammars. *Proceedings of TAG+6 Workshop*. Venice. pp. 1 – 10.

Han, C.-H.: 2002b. Compositional Semantics for Relative Clauses in Lexicalized Tree Adjoining Grammars. Slides, talk at TAG+6 Workshop, Venice.

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

Joshi, A. K. and 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.

Joshi, A. K., Kallmeyer, L. and Romero, M.: 2003. Flexible Composition in LTAG. Quantifier Scope and Inverse Linking. *Proceedings of International Workshop on Computational Semantics (IWCS)*. Tilburg.

Kahane, S., Candito, M.-H. and de Kercadio, Y.: 2000. An alternative description of extraction in TAG. *Proceedings of TAG+5*. Paris. pp. 115–122.

Kallmeyer, L.: 2002a. Enriching the tag derivation tree for semantics. *Proceedings of KONVENS 2002*. Saarbrücken. pp. 67 – 74.

Kallmeyer, L.: 2002b. Using an Enriched TAG Derivation Structure as Basis for Semantics. *Proceedings of TAG+6 Workshop*. Venice. pp. 127 – 136.

Kallmeyer, L. and Joshi, A. K.: 1999. Factoring Predicate Argument and Scope Semantics: Underspecified Semantics with LTAG. *in* P. Dekker (ed.), *12th Amsterdam Colloquium. Proceedings*. Amsterdam. pp. 169–174.

Kallmeyer, L. and Joshi, A. K. 2002. Factoring Predicate Argument and Scope Semantics: Underspecified Semantics with LTAG. *Research on Language and Computation*. To appear.

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

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

XTAG Research Group: 1998. A Lexicalized Tree Adjoining Grammar for English. *Technical Report 98–18*. Institute for Research in Cognitive Science. Philadelphia.