Tree-local MCTAG with Shared Nodes: Word Order Variation in German and Korean

Laura Kallmeyer **SinWon Yoon**
UFRL, University Paris 7
2 place Jussieu, Case 7003, 75251 Paris Cedex 05
{laura.kallmeyer,swyoon}@linguist.jussieu.fr

Abstract

Tree Adjoining Grammars (TAG) are known not to be powerful enough to deal with scrambling in free word order languages. The TAG-variants proposed so far in order to account for scrambling are not entirely satisfying. Therefore, an alternative extension of TAG is introduced based on the notion of node sharing. Considering data from German and Korean, it is shown that this TAG-extension can adequately analyse scrambling data, also in combination with extraposition and topicalization.

1 Introduction

1.1 LTAG and scrambling

Lexicalized Tree Adjoining Grammars (LTAG, (Joshi and Schabes, 1997)) is a tree-rewriting formalism. An LTAG consists of a finite set of trees (elementary trees) associated with lexical items. Larger trees are derived by substitution (replacing a leaf with a new tree) and adjunction (replacing an internal node with a new tree). LTAG elementary trees represent extended projections of lexical items and encapsulate all syntactic 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.

Roughly, *scrambling* is the permutation of elements (arguments and adjuncts) of a sentence (we use the term *scrambling* in a purely descriptive sense without implying any theory of movement). A special case is *long-distance* scrambling where arguments or adjuncts of an embedded infinitive are 'moved' out of the embedded VP. This occurs for instance in languages such as German, Hindi, Japanese and Korean. These languages are therefore often said to have a free word order. Consider for example the German sentence (1). In (1), the accusative NP *es* is an argument of the embedded infinitive *zu reparieren* but it precedes *der Mechaniker*, the subject of the main verb *verspricht* and it is not part of the embedded VP. It has been argued that in German there is no bound on the number of scrambled elements and no bound on the depth of scrambling (i.e., in terms of movement, the number of VP borders crossed by the moved element). (See for example (Rambow, 1994a; Meurers, 2000; Müller, 2002) for descriptions of scrambling data.)

(1) ... dass [es]$_1$ der Mechaniker [t_1 zu reparieren] verspricht
... that it the mechanic to repair promises
'... that the mechanic promises to repair it'

As shown in (Becker et al., 1991), TAG are not powerful enough to describe scrambling in German in an adequate way. By this we mean that a TAG analysis of scrambling with the correct predicate-argument structure is not possible, i.e., an analysis with each argument attaching to the verb it depends on.

Let us consider the analysis of (1) in order to get an idea of why scrambling poses a problem for TAG. If we leave aside the complementizer *dass*, elementary trees for *verspricht* and *reparieren* might look as shown in Fig. 1. In the derivation, the *verspricht*-tree adjoins to the root of the *reparieren*-tree and the NP *der Mechaniker* is substituted for the subject node of *verspricht*.[1] This leads to the third tree in Fig. 1. When adding *es*, there is a problem: it should be added to *reparieren* since it is one of its arguments. But at the same time, it should precede *Mechaniker*, i.e., it must be adjoined either to the root or to the NP$_{nom}$ node in the derived tree. The root node belongs to *verspricht* and the NP$_{nom}$ node belongs to *Mechaniker*. Consequently, an adjunction to one of them would not give the desired predicate-argument structure. If it was only for (1), one could add a tree to the grammar

[1] The fact that *der Mechaniker* is at the same time logical subject of *reparieren* is accounted for in the semantics, see for example (Gardent and Kallmeyer, 2003).

TAG+7: Seventh International Workshop on Tree Adjoining Grammar and Related Formalisms.
May 20-22, 2004, Vancouver, BC, CA.
Pages 40-47.

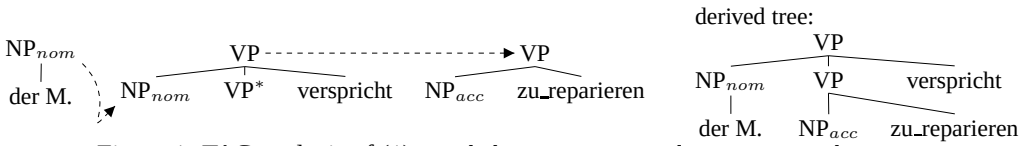

Figure 1: TAG analysis of (1) *dass* $[es]_1$ *der Mechaniker* $[t_1$ *zu reparieren*$]$ *verspricht*

for *reparieren* with a scrambled NP that allows adjunction of *verspricht* between the NP and the verb. But as soon as there are several scrambled elements that are arguments of different verbs, this does not work any longer. In general, it has been shown (Joshi et al., 2000) that adopting specific elementary trees it is possible to deal with a part of the difficult data: TAG can describe scrambling up to depth 2 (two crossed VP borders). But this is not sufficient. Even though examples of scrambling of depth > 2 are rare, they can occur (see Kulick, 2000).

1.2 TAG variants proposed for scrambling

The problem of long-distance scrambling and TAG is the fact that the trees representing the syntax of scrambled German subordinate clauses do not have the simple nested structure that ordinary TAG generates. In TAG, according to the Condition on Elementary Tree Minimality (CETM, (Frank, 1992)) (positions for) all of the arguments of the lexical anchor of an elementary tree are included in that tree. But in the scrambled tree the arguments of several verbs are interleaved freely. All TAG extensions that have been proposed to accommodate this interleaving involve factoring the elementary structures into multiple components and inserting these components at multiple positions in the course of the derivation.

One of the first proposals made was an analysis of German scrambling data using non-local MCTAG with additional dominance constraints (Becker et al., 1991). However, the formal properties of non-local MCTAG are not well understood and it is assumed that the formalism is not polynomially parsable. Therefore this approach is no longer pursued but it has influenced the different subsequent proposals.

An alternative formalism for scrambling is V-TAG (Rambow, 1994a; Rambow, 1994b; Rambow and Lee, 1994), a formalism that has nicer formal properties than non-local MCTAG. V-TAG also use multicomponent sets (so-called *vectors*) for scrambled elements, in this it is a variant of MCTAG. Additionally, there are dominance links between the trees of one vector. In contrast to MCTAG, the trees of a vector are not required to be added simultaneously. The lexicalized V-TAGs that are of interest for natural languages are polynomially parsable. Even though the formalism does not pose the problems of non-local MCTAG in terms of parsing complexity, it is still a non-local formalism in the sense that, as long as the dominance links are respected, arbitrary nodes can be chosen to attach the single components of a vector. This makes the formalism harder to understand than local TAG-variants since one needs a more gobal picture of what is going on in a derivation. Furthermore, in order to formulate certain locality restrictions (e.g., for wh-movement and also for scrambling), one needs an additional means to put constraints on what can interleave with the different trees of a vector or in other words constraints on how far a dominance link can be stretched. V-TAG allows to put *integrity constraints* on certain nodes that disallow these nodes to occur between two trees linked by a dominance link. This has the effect that these nodes act as barriers. This explicit marking of barriers is somewhat against the original appealing TAG idea that such constraints result from the CETM which imposes the position of the moved element and the verb it depends on to be in the same elementary structure, and from the further possibilities to combine this structure. In other words, in local formalisms with an extended domain of locality such as TAG or tree-local and set-local MCTAG such constraints result from the form of the elementary structures and the locality of the derivation.

D-tree substitution grammars (DSG, Rambow, Vijay-Shanker, and Weir, 2001) are another TAG-variant one could use for scrambling. DSG are a description-based formalism, i.e., the objects a DSG deals with are tree descriptions. A problem with DSG is that the expressive power of the formalism is probably too limited to deal with all natural language phenomena: according to (Rambow et al., 2001) it 'does not appear to be possible for DSG to generate the copy language". This means that the formalism is probably not able to describe cross-serial dependencies in Swiss German. Furthermore, DSG is non-local and therefore, as in the case of V-TAG, additional constraints (so-called *path constraints*) have to be put on material interleaving with the different parts of an elementary structure.

Another TAG-variant proposed in order to deal with scrambling are Segmented Tree Adjoining Grammars (SegTAG, Kulick, 2000). SegTAG can generate the copy language and therefore describe cross-serial dependencies. But the formalism uses a rather complex operation on trees, segmented adjunction, that consists partly of a standard TAG adjunction and partly of a kind of tree merging or tree unification. In this operation, two different things get mixed up, the more or less resource-sensitive adjoining operation of standard TAG where sub-

trees cannot be identified,[2] and the completely different unification operation. Furthermore, the formal properties of SegTAG are not clear. Kulick suggests that SegTAGs are probably in the class of LCFRS but there is no actual proof of this. However, if SegTAG is in LCFRS, the generative power of the formalism is probably too limited to deal with scrambling in a general way. In order to treat scrambling up to a certain depth, Kulick therefore allows certain extensions of SegTAG.

All these TAG variants are interesting with respect to scrambling and they give a lot of insight into what kind of structures are needed for scrambling. But, as explained above, none of them is entirely satisfying. The most convincing one is V-TAG since this formalism can deal with scrambling, it is polynomially parsable and the set of languages it generates contains the set TAL of all tree adjoining languages (in particular the copy language). But, as already mentioned, V-TAG has the inconvenient of being a non-local formalism. For the reasons explained above, it is desireable to find a local TAG extension for scrambling (as opposed to the non-locality of derivations in V-TAG, DSG and non-local MCTAG) such that locality constraints for movements follow only from the form of the elementary structures and from the local character of derivations. This paper proposes a local TAG-variant that can deal with scrambling, at least with an arbitrarily large set of scrambling phenomena, that is polynomially parsable and that properly extends TAG in the sense that TAL is a proper subset of the languages it generates.

In section 2, tree-local MC-TAG with shared nodes (SN-MCTAG) and in particular restricted SN-MCTAG (RSN-MCTAG) are introduced. Section 3 to 5 show the analyses of different word order variations using this formalism, namely scrambling, extraposition and topicalization, considering data from German and Korean.

2 Tree-local MCTAG with shared nodes (SN-MCTAG)

To illustrate the idea of shared nodes, consider again example (1). In standard TAG, nodes to which new elementary trees are adjoined or substituted disappear, i.e., they are replaced by the new elementary tree. E.g., after the derivation steps shown in Fig. 1, the root node of the *reparieren* tree does not exist any longer. It is replaced by the *verspricht* tree and its daughters have become daughters of the foot node of the *verspricht* tree. I.e., the root node of the derived tree is considered being part of only the *verspricht* tree. Therefore, an adjunction at that node is an adjunction at the *verspricht* tree. However, this standard TAG view is not completely justified: in the derived tree, the root node and the lower VP node might as well be considered as belonging to *reparieren* since they are results of identifying the root node of *reparieren* with the root and the foot node of *verspricht*.[3] Therefore, we propose that the two nodes in question belong to both, *verspricht* and *reparieren*. In other words, these nodes are shared by the two elementary trees. Consequently, they can be used to add new elementary trees to *verspricht* and (in contrast to standard TAG) also to *reparieren*.

We use a multicomponent TAG (MCTAG, Joshi, 1987; Weir, 1988). This means that the elements of the grammar are sets of elementary trees. In each derivation step, one of these sets is chosen and the trees in this set are added simultaneously (by adjunction or substitution) to different nodes in the already derived tree. We assume tree-locality, i.e., the nodes to which the trees of such a set are added must all belong to the same elementary tree. Standard tree-local MCTAGs are strongly equivalent to TAG but they allow to generate a richer set of derivation structures. In combination with shared nodes, tree-local multicomponent derivation extends the weak generative power of the grammar.

Let us go back to (1). Assume the tree set on the left of Fig. 2 for *es*. Adopting the idea of shared nodes, this tree set can be added to *reparieren* using the root of the already derived tree for adjunction of the first tree and the NP_{acc} node for substitution of the second tree. The operation is tree-local since both nodes are part of the *reparieren* tree.

In general, the notion of shared nodes means the following: When substituting an elementary tree α into an elementary tree γ, in the resulting tree, the root node of the subtree α is considered being part of α and of γ. When adjoining an elementary β at a node that is part of the elementary trees $\gamma_1, \ldots, \gamma_n$, then in the resulting tree, the root and foot node of β are both considered being part of $\gamma_1, \ldots, \gamma_n$ and β. Consequently, if an elementary γ' is added to an elementary γ and if there is then a sequence of adjunctions at root or foot nodes starting from γ', then each of these adjunctions can be considered as an adjunction at γ since it takes place at a node shared by γ, γ' and all the subsequently adjoined trees. In Fig. 2 for example the *es*-tree is adjoined to the root of a tree that was adjoined to *reparieren*. Therefore this adjunction can be

[2] More precisely, only the root of the new elementary tree and eventually (i.e., in case of an adjunction) the foot node get identified with the node the new tree attaches to. But there is no unification of whole subtrees.

[3] Actually, in a Feature-Structure Based TAG (FTAG, (Vijay-Shanker and Joshi, 1988)), the top feature structure of the root of the derived tree is the unification of the top of the root of *verspricht* and the top of the root of *reparieren*. The bottom feature structure of the lower VP node is the unification of the bottom of the foot of *verspricht* and the bottom of the root of *reparieren*. In this sense, the root of the *reparieren* tree gets split into two parts. The upper part merges with the root node of the *verspricht* tree and the lower part merges with the foot node of the *verspricht* tree.

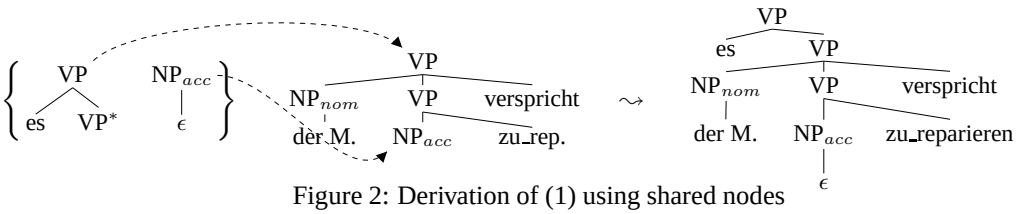

Figure 2: Derivation of (1) using shared nodes

considered being an adjunction at *reparieren*. An adjunction at a node where other trees already have been added (e.g., this adjunction of *es* to the root of *reparieren*) is called a *secondary* adjunction while a first adjunction at a node is called a *primary* adjunction.

Concerning formal properties, SN-MCTAG is hard to compare to other local TAG-related formalisms since arbitrarily many trees can be added by secondary adjunction to a single elementary tree. Therefore, we define a restricted version, *restricted SN-MCTAG (RSN-MCTAG)* that limits the number of secondary adjunctions to an elementary tree by allowing secondary adjunction only in combination with at least one simultaneous primary adjunction or substitution. E.g., in Fig. 2, *es* is secondarily adjoined to *reparieren* while the second element of the tree set is primarily added (substituted) to *reparieren*.

Obviously, all tree adjoining languages can be generated by RSN-MCTAGs since a TAG is an MCTAG with unary multicomponent sets. It can be shown that for each RSN-MCTAG of a specific type, an equivalent simple Range Concatenation Grammars (RCG, (Boullier, 1998; Boullier, 1999)) and therefore an equivalent LCFRSs (linear context-free rewriting systems, (Weir, 1988)) can be constructed. LCFRSs are mildly context-sensitive and in particular polynomially parsable and therefore, this also holds for these specific RSN-MCTAGs. For a formal definition of SN-MCTAG and RSN-MCTAG and a sketch of the proof of the mildly context-sensitivity see (Kallmeyer, 2004). The additional restriction imposed on RSN-MCTAG in order to obtain the equivalence to LCFRS puts a limit on the complexity of the scrambling data one can analyze. This limit however is variable in the sense that an arbitrarily large limit can be chosen. Consequently, based on empirical studies, the limit can be chosen such that all scrambling data are covered that are assumed to occur in real texts. In this respect, RSN-MCTAG differs crucially from TAG where the limit is fixed (scrambling up to depth 2 can be described and nothing more). In this sense one can say that RSN-MCTAG can analyze scrambling in general since it can anlyze any arbitrarily large finite set of scrambling data.

There are mainly two crucial differences between SN-MCTAG and V-TAG: firstly, in V-TAG the adjunctions of auxiliary trees from the same set are not required to be simutaneously. In this respect, V-TAG differs from standard MCTAG in general. Secondly, V-TAG is non-local in the sense of non-local MCTAG while RSN-MCTAG is local, even though the locality is not based on the parent relation in the TAG derivation tree as it is the case in standard local MCTAG. As a consequence of the locality, in contrast to other TAG variants for scrambling, we do not need dominance links in RSN-MCTAG. The locality condition put on the derivation sufficiently constrains the possibilities for attaching the trees from elementary tree sets: different trees from a tree set attach to different nodes of the same elemenary tree, so the dominance relations between these different nodes are crucial for the dominance relation between the different trees from the tree set. Because of this dominance links are not necessary. This is different of course for non-local TAG-variants such as V-TAG or DSG where one can in principle attach the different components of an elementary structure at arbitrary nodes in the derived tree.

3 Scrambling

In many SOV languages, such as German, Hindi, Japanese and Korean, constituents (argument or adjunct) display a larger freedom in term of ordering in clauses. This phenomenon is called *scrambling*. (See (Uszkoreit, 1987) for a description of word order in German and (Lee, 1993) for Korean.) The constituents of the lower clause can even occur in the upper clause, (so-called *long distance* scrambling). E.g., the arguments *es* and *jadoncha-lul* of the embedded verb move into the upper clause in German (1), repeated as (2)a., and in the Korean sentence (2)b.

(2) a. ... dass es$_1$ der Mechaniker [t_1 zu reparieren] verspricht

b. jadoncha-lul$_1$ keu-ka [t_1 surihakess-tako]
the car$_{acc}$ he$_{nom}$ [t_1 repair-to]
yaksokhaessta
promises
'He promises to repair the car'

Generally, in both languages, it is assumed that there is no bound on the number of elements that can scramble in one sentence, and there is no bound on the distance over which each element can scramble. In the following we will show how RSN-MCTAG allows to deal with long distance scrambling. Elementary trees for word order variations of (3) are shown in Fig. 3. We propose

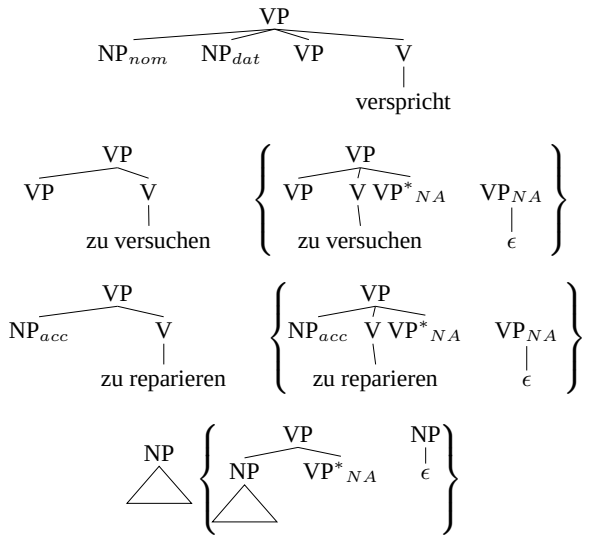

Figure 3: Elementary trees for word order variations of (3) *... dass er dem Kunden* [[*das Auto zu reparieren*] *zu versuchen*] *verspricht*

single trees for non-scrambled elements, and tree sets for scrambled elements.

(3) ... dass er dem Kunden [[das Auto zu reparieren]
... that he$_{nom}$ the customer$_{dat}$ the car$_{acc}$ to repair
zu versuchen] verspricht
to try promises
'... that he promises the customer to try to repair the car'

(4) ... dass er das Auto$_1$ dem Kunden [[t_1 zu reparieren] zu versuchen] verspricht

Consider (4) where the most deeply embedded NP$_{acc}$ *das Auto* is scrambled into the upper clause. For *das Auto*, the tree set is used. Further, we also use tree sets for the NP$_{dat}$ *dem Kunden* which intervenes between the scrambled argument and its clause, and for the VP clause *reparieren* of witch argument is scrambled out over a clause of depth ≥ 2. For the non-scrambled NP$_{nom}$ *er*, and for the non-scrambled VP *versuchen*, single trees are used. Fig. 4 shows the different derivation steps for (4). First, *verspricht* and *versuchen* are combined by substitution. In the resulting derived tree (on the right on top of the figure), the bold VP node is now shared by *verspricht* and *versuchen*. Then the auxiliary tree in the tree set for *reparieren* adjoins to the shared node. This is a primary adjunction at *versuchen*. The initial tree is substituted for the VP leaf of *versuchen*. The former root node of the *reparieren* auxiliary tree, i.e., the bold VP node in the tree in the middle of the bottom of the figure, is now shared by *verspricht*, *versuchen* and *reparieren*. The next *secondary* adjunctions can occur at this new shared node: *dem Kunden* is added as sketched in the figure, and then *das Auto* is added in the same way. The tree for *er* is added into the substitution slot in the *verspricht* tree.

Note that a scrambled elements always adjoins to a VP node and the scrambled element is to the left of the foot node. Therefore it precedes everything that is below or on the right of the VP node to which it adjoins. Consequently, given the form of the verbal elementary trees in Fig. 3 where the verb is always below or right of all VP nodes allowing adjunction, the order $x\ v$ for an x being a nominal or a verbal argument of v is always respected.

Since all scrambled elements attach to a VP node in the elementary tree of the verb they depend on, they cannot attach to the VP of a higher finite verb that embeds the sentence in which the scrambling occurs. Therefore, this analysis correctly predicts that scrambling can never proceed out of tensed clauses. In other words, a barrier effect is obtained without posing any explicit barrier as it is done in V-TAG. Instead, the locality of scrambling is a consequence of the form of the elementary trees and of the locality of the derivations.

In contrast to German, Korean allows scrambling out of a tensed clause. For example, in (5) the argument *jadoncha-lul* is scrambled out of a tensed clause. This difference can be captured by using in Korean the node label S instead of VP for the root and the foot node in the auxiliary trees for scrambling.[4]

(5) jadoncha-lul$_1$ keu-ka [kokaek-i t_1
the car$_{acc}$ he$_{nom}$ [the customer$_{nom}$ t_1
kuiphaess-tako] malhaessta.
buy-that] said
'He said that the customer bought the car'

4 Extraposition

In German and Korean, clausal arguments can optionally appear behind the finite verb. This is called *extraposition*. E.g., in (6), the *reparieren* VP occurs behind the finite verb *verspricht*. The same goes for the Korean extraposition (7).

(6) ... dass er$_{nom}$ dem Kunden$_{dat}$ t_1 verspricht, [das Auto$_{acc}$ zu reparieren]$_1$
'... that he promises the customer to repair the car'

(7) keu-ka$_{nom}$ kokaek-ekey$_{dat}$ t_1 yaksokhassta, [jadoncha-lul$_{acc}$ surihakess -tako]$_1$
'He promises the customer to repair the car'

[4] One aspect we did not consider in this paper but that definitely needs to be spelled out is the fact that in both languages, German and Korean, not all verbs allow scrambling to the same degree. In German, this is related to the difference between obligatorily and optionally coherent verbs (see (Meurers, 2000; Müller, 2002)). These facts probably can be modelled using specific features that control the scrambling possibilities of a verb.

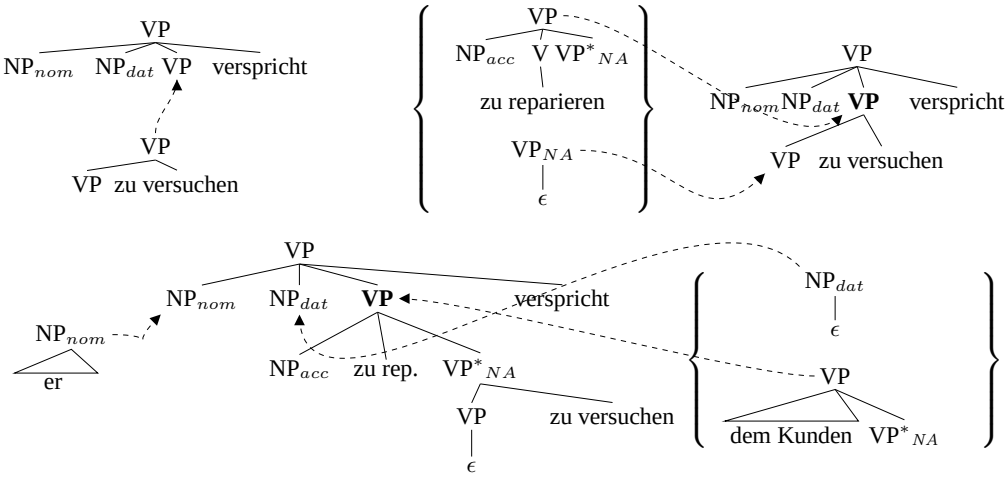

Figure 4: Derivation for (4) ... *dass er das Auto dem Kunden zu reparieren zu versuchen verspricht*

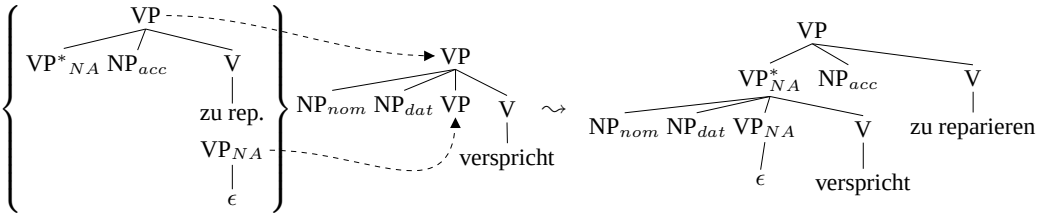

Figure 5: Derivation for (6) ... *dass er dem Kunden verspricht, das Auto zu reparieren*

Extraposition is doubly unbounded, as it is the case for *scrambling*. In order to analyze *extraposition*, we propose tree sets as the one for *reparieren* in Fig. 5. They resemble to those for *scrambling* except that the foot node is on the left because the extraposed material goes to the right of the finite verb. For the NP arguments in (6), we use the single trees shown in Fig. 3. The derivation for (6) is as sketched in Fig. 5.

The following differences between German and Korean are observed: both languages allow extraposition of complete VPs. Furthermore, in German, infinitives without their arguments can be extraposed (so-called *third construction*, see (8)a), which is not possible in Korean (see (9)a). In Korean however, arguments of embedded verbs can be extraposed while leaving their verb behind (see (9)b), which is not possible in German (see (8)b).[5]

(8) a. ... dass er es t_1 verspricht, [zu reparieren]$_1$

b. $*$... dass er [t_1 zu reparieren] verspricht, [es]$_1$

(9) a. $*$keu-ka$_{nom}$ jadoncha-lul$_{acc}$ t_1 yaksokhassta, [surihakess-tako]$_1$

b. keu-ka$_{nom}$ [t_1 surihakess-tako] yaksokhassta, [jadoncha-lul$_{acc}$]$_1$

[5]For this reason, Korean extraposition is often called *right-forward scrambling*.

To account for the difference between (8a) and (9a), we disallow the adjunction of scrambled elements at the root nodes of Korean auxiliary extraposition trees.[6] For (9b), in Korean, we propose additional tree sets for extraposed NPs. They are similar to the tree sets for scrambled NPs in Fig. 3, except that the foot node is on the left. Such tree sets do not exist in German.

5 Topicalization

Korean *topicalization* is realized with the topic marker *-nun*(*-un*). The topicalized constituent has to appear in the beginning of clauses, e.g., *jadoncha-nun* in (10a.): an element marked by *-nun*(*-un*) can also appear in sentence medial position e.g., *jadoncha-nun* in (10b.). It is perceived, in Korean, that an element with *-nun*(*-un*) in sentence initial position receives the theme reading, i.e., *topicalization*, and the counterpart in sentence medial position the contrastive reading. To describe *topicalization* movement, a topic argument may be inserted into the verbal projection tree at [Spec, CP] (see, e.g., (Suh, 2002)).

[6]In German, even arguments of embedded VPs can be left behind as in ... *dass er [es]$_1$ verspricht, [[t_1 zu reparieren] zu versuchen]*. For such cases, we propose an additional VP node on the spine of extraposed infinitives where deeper embedded infinitives can be added. For reason of space, we will not go into the details here.

(10) a. jadoncha-nun$_1$ keu-ka [t_1 kuiphakess-tako]
the car$_{top}$ he$_{nom}$ [t_1 buy-to]
yaksokhassta.
promises
'As for the car, he promises to buy (it)'

b. keu-ka jadoncha-nun kuiphakess-tako yaksokhassta.
'He promises to buy the car'

German *topicalization* is more strict. German exhibits the verb second effect (V2), i.e., the finite verb (main verb or auxiliary) occupies the second position in the clause. This divides the clause into two parts: the part before the finite verb, the *Vorfeld* (VF), and the part between the finite verb and non-finite verb, the *Mittlefeld* (MF). The VF must contain exactly one constituent. This constituent is considered having moved into the VF. This movement is called *topicalization*. E.g., in (11) the auxiliary verb *hat* appears in second position, the NP$_{acc}$ *das Buch* that moved from the MF into the first position is topicalized.

(11) das Buch$_2$ hat ihm$_1$ niemand [t_1 t_2 zu geben] versucht.
the book has him nobody [t_1 t_2 to give] tried.
'Nobody has tried to give him the book.'

In both languages, *topicalization* concerns exactly one element, and the element has to appear in the beginning of the clause, while scrambling and extraposition can occur for more than one element. I.e., no operation to add constituents in front of topicalized element is accepted. Furthermore, in German matrix clauses, topicalization is obligatory. We capture these restrictions by certain features. The last step in a derivation for a sentence exhibiting *topicalization* is the adjunction of the topicalized constituent. The feature of the final derived root node becomes $\left[\begin{smallmatrix}\text{CP}\\ \text{CP}\end{smallmatrix}\right]$. It prevents adding other constituents at the root.[7]

Topicalization and *scrambling* can occur simultaneously as in (11) where *ihm* is long-distance scrambled and *das Buch* is long-distance topicalized. Fig. 6 shows the derivation for (11): Starting with the initial tree for *versucht*, the auxiliary tree for *geben* is adjoined at the root node with top category CP and bottom category VP (we assume here feature structures as labels with different top and bottom features), and simultaneously the initial VP tree is added into the lower VP. After this, the $\left[\begin{smallmatrix}\text{CP}\\ \text{VP}\end{smallmatrix}\right]$ root node is shared by *versucht* and *geben*. Then, *niemand* and *ihm* are subsequently added. This gives the tree on the left of the bottom of the figure. Next, *hat* is adjoined at the root which leads to a $\left[\begin{smallmatrix}\text{CP}\\ \text{C'}\end{smallmatrix}\right]$ root node shared (among others) by *geben* and *versucht*. Finally, the topicalized element is adjoined to the root node.

For topicalized elements in Korean, we propose the same kind of tree set as for German topicalized elements, except that the category of the foot node is unspecified. This does not fix the position of the topicalized element between CP and C'(as in German).

6 Conclusion

Since TAG are not powerful enough to describe scrambling data in free word order languages, alternative formalisms are needed. The proposals made so far in the litereature are not entirely satisfying. Therefore, we developed a new TAG extension, restricted MCTAG with shared nodes (RSN-MCTAG). The basic idea is that, after having performed an adjunction or substitution at some node, this node does not disappear (as in standard TAG) but instead, in the resulting derived tree, the node is shared between the old tree and the newly added tree. Consequently, further adjunctions at that node can be considered being adjunctions at either of the trees. In combination with tree-local multicomponent derivation, this modification of the TAG derivation gives sufficient additional power to analyse the difficult scrambling data.

Considering data from German and Korean, we showed that RSN-MCTAG can adequately analyse scrambling data, also in combination with extraposition and topicalization. The analyses proposed in the paper treat long-distance scrambling, long-distance extraposition and long-distance topicalization and they take into account the differences German and Korean exhibit with respect to these phenomena.

Acknowledgments

For helpful comments and fruitful discussions of the subject of this paper, we would like to thank Anne Abeillé, David Chiang, Aravind Joshi and Seth Kulick. Furthermore, we are grateful to three anonymous reviewers for their valuable suggestions for improving the paper.

References

Tilman Becker, Aravind K. Joshi, and Owen Rambow. 1991. Long-distance scrambling and tree adjoining grammars. In *Proceedings of ACL-Europe*.

Pierre Boullier. 1998. A Proposal for a Natural Language Processing Syntactic Backbone. Technical Report 3342, INRIA.

Pierre Boullier. 1999. On TAG Parsing. In *TALN 99, 6^e conférence annuelle sur le Traitement Automatique*

[7] We also pursued an alternative analysis, namely putting the slot for the topicalized element (a substitution node) and the verb it depends on in the same initial tree. I.e., the topicalized element is added by substitution while scrambled or extraposed elements are added by adjunction. This is a more obvious way to capture the restrictions for *topicalization*. Unfortunately, this approach does not work with some combinations of topicalization and scrambling as for example *[es]$_1$ hat er [t_1 zu reparieren]$_2$ dem Kunden [t_2 zu versuchen] versprochen.*

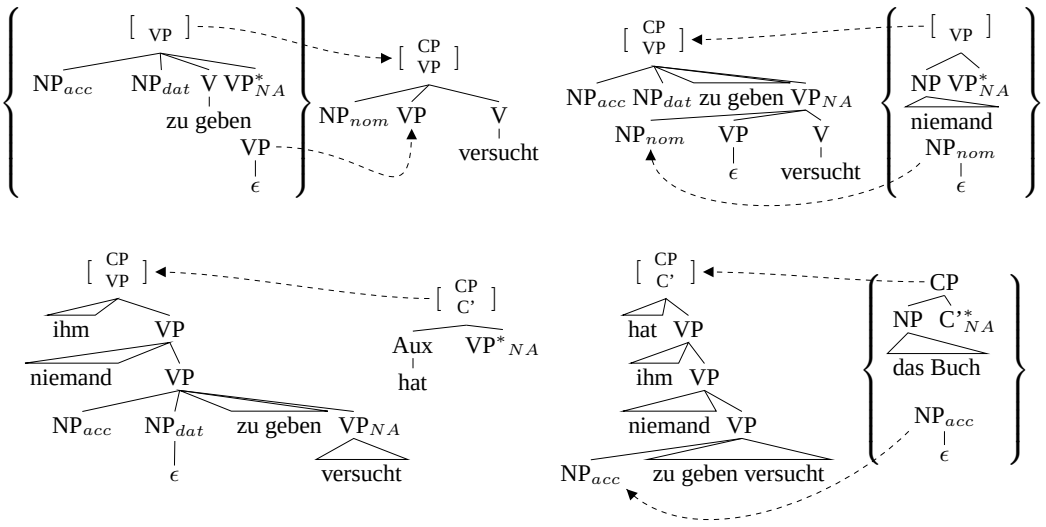

Figure 6: Derivation for (11) *... das Buch hat ihm niemand zu geben versucht*

des Langues Naturelles, pages 75–84, Cargèse, Corse, July.

Robert Frank. 1992. *Syntactic Locality and Tree Adjoining Grammar: Grammatical, Acquisition and Processing Perspectives*. Ph.D. thesis, University of Pennsylvania.

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

Aravind K. Joshi and Yves Schabes. 1997. Tree-Adjoining Grammars. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, pages 69–123. Springer, Berlin.

Aravind K. Joshi, Tilman Becker, and Owen Rambow. 2000. Complexity of scrambling: A new twist to the competence/performance distinction. In Anne Abeillé and Owen Rambow, editors, *Tree Adjoining Grammars: Formalisms, Linguistic Analyses and Processing*. CSLI.

Aravind K. Joshi. 1987. An introduction to Tree Adjoining Grammars. In A. Manaster-Ramer, editor, *Mathematics of Language*, pages 87–114. John Benjamins, Amsterdam.

Laura Kallmeyer. 2004. Tree-local multicomponent tree adjoining grammars with shared nodes. Unpublished manuscript, University of Paris 7. Under revision for resubmission, April.

Seth Norman Kulick. 2000. *Constraining Non-local Dependencies in Tree Adjoining Grammar: Computational and Linguistic Perspectives*. Ph.D. thesis, University of Pennsylvania.

Young-Suk Lee. 1993. *Scrambling as Case-driven Obligatory Movement*. Ph.D. thesis, University of Pennsylvania. Published as technical report IRCS-93-06.

Walt Detmar Meurers. 2000. *Lexical Generalizations in the Syntax of German Non-Finite Constructions*. Ph.D. thesis, Universität Tübingen.

Stefan Müller. 2002. *Complex Predicates: Verbal Complexes, Resultative Constructions, and Particle Verbs in German*. CSLI Stanford.

Owen Rambow and Young-Suk Lee. 1994. Word order variation and Tree-Adjoining Grammars. *Computational Intelligence*, 10(4):386–400.

Owen Rambow, K. Vijay-Shanker, and David Weir. 2001. D-Tree Substitution Grammars. *Computational Linguistics*.

Owen Rambow. 1994a. *Formal and Computational Aspects of Natural Language Syntax*. Ph.D. thesis, University of Pennsylvania.

Owen Rambow. 1994b. Multiset-Valued Linear Index Grammars: Imposing dominance constraints on derivations. In *Proceedings of ACL*.

Chung-Mok Suh. 2002. Topicalization and Focusing in Korean. In *The Twelfth International Conference on Korean Linguistics*, pages 511–522.

Hans Uszkoreit. 1987. *Word Order and Constituent Syntax in German*. CSLI Stanford.

K. Vijay-Shanker and Aravind K. Joshi. 1988. Feature structures based tree adjoining grammar. In *Proceedings of COLING*, pages 714–719, Budapest.

David J. Weir. 1988. *Characterizing mildly context-sensitive grammar formalisms*. Ph.D. thesis, University of Pennsylvania.