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

Laura Kallmeyer[*] — **SinWon Yoon**[**]

[*] *SFB 441, University of Tübingen*
Nauklerstr. 35, D-72074 T übingen, Germany
laura.kallmeyer@linguist.jussieu.fr

[**] *UFRL, University Paris 7*
2 place Jussieu, Case 7003, 75251 Paris Cedex 05, France
swyoon@linguist.jussieu.fr

RÉSUMÉ. *Les Grammaires d'Arbres Adjoints (TAG) sont connues pour ne pas être assez puissantes pour traiter le brouillage d'arguments dans des langues à ordre des mots libre. Les variantes TAG proposées jusqu'à présent afin d'expliquer le brouillage n'ont pas donné entière satisfaction. Nous présentons ici une extension alternative de TAG, basée sur la notion de partage de noeuds. S'appuyant sur des données relatives à l'allemand et au coréen, il est démontré que cette extension de TAG est parfaitement en mesure d'analyser des données de brouillage d'arguments, même combiné à des extrapositions ou des topicalisations.*

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.*

MOTS-CLÉS : *Grammaires d'Arbres Adjoints, brouillage d'arguments, ordre des mots, allemand, coréen*

KEYWORDS: *Tree Adjoining Grammars, scrambling, word order, German, Korean*

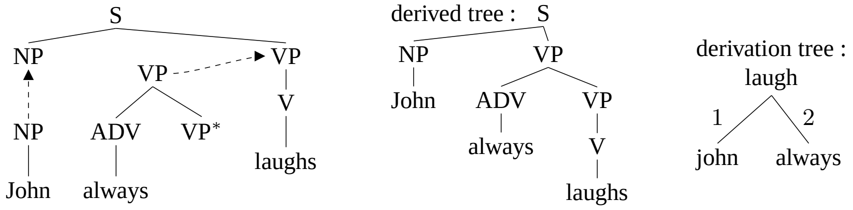

Figure 1. *TAG derivation for (1)*

1. LTAG and scrambling

1.1. *Lexicalized Tree Adjoining Grammars (LTAG)*

Tree Adjoining Grammar (TAG) is a tree-rewriting formalism originally defined by (Joshi *et al.*, 1975) ; for an introduction see (Joshi & Schabes, 1997). A TAG consists of a finite set of trees (elementary trees). The nodes of these trees are labelled with nonterminals and terminals (terminals only label leaf nodes). Starting from the elementary trees, larger trees are derived using two composition operations : substitution (replacing a leaf with a new tree) and adjunction (replacing an internal node with a new tree). In case of an adjunction, the tree being adjoined has exactly one leaf node that is identified as the foot node (marked with an asterisk). Such a tree is called an *auxiliary* tree. When adjoining an auxiliary tree to a node μ, in the resulting tree, the subtree with root node μ from the old tree is put below the foot node of the adjoined auxiliary tree. Elementary trees that are not auxiliary trees are called *initial* trees. Each derivation starts with an initial tree. In the final derived tree, all leaves must have terminal labels.

As an example see Fig. 1 for the derivation of (1). Here, the three elementary trees for *laughs*, *John* and *always* are combined : Starting from the elementary tree for *laughs*, the tree for *John* is substituted for the NP leaf and the tree for *always* is adjoined at the VP node.

(1) John always laughs

TAG derivations are represented by derivation trees that record the history of how the elementary trees are put together. A derived tree is the result of carrying out the substitutions and adjunctions. Each edge in the derivation tree stands for an adjunction or a substitution. The edges are labelled with Gorn addresses of the nodes where the substitutions/adjunctions take place : the root has the address ϵ, and the jth child of the node with address p has address pj. In Fig. 1 for example the derivation tree indicates that the elementary tree for *John* is substituted for the node at address 1 and *always* is adjoined at node address 2.

TAGs for natural languages are *lexicalized* (Schabes, 1990) which means that each elementary tree has a lexical anchor (usually unique but in some cases, there is more than one anchor). Furthermore, the elementary trees represent extended projections of lexical items (the anchors) and encapsulate all arguments of the lexical anchor, i.e., they contain slots (non-terminal leaves) for all arguments. Finally, elementary trees are minimal in the sense that only the arguments of the anchor are encapsulated, all recursion is factored away. This amounts to the *Condition on Elementary Tree Minimality* from (Frank, 1992).[1] The tree for *laughs* in Fig. 1 for example contains only a non-terminal leaf for the subject NP (a substitution node) but there is no slot for a VP adjunct. The adverb *always* is added by adjunction at an internal node.

In a TAG for natural languages, combining two elementary trees by substitution or adjunction corresponds to the application of a predicate to an argument. The derivation tree then reflects the predicate-argument structure of the sentence. This is why most approaches to semantics in TAG use the derivation tree as interface between syntax and semantics (see, e.g., (Candito & Kahane, 1998; Joshi & Vijay-Shanker, 1999; Kallmeyer & Joshi, 2003)). In this paper, we are not particularly concerned with semantics, but one of the goals of the paper is to obtain analyses with derivation trees representing the correct predicate-argument dependencies.

1.2. *Scrambling in LTAG*

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 (2). 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.

(2) ... dass $[\text{es}]_1$ der Mechaniker $[t_1$ zu reparieren$]$ verspricht
... that it the mechanic to repair promises
'... that the mechanic promises to repair it'

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.)

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 scram-

[1] This minimality is actually the reason why the substitution operation is needed ; formally TAGs without substitution and TAGs as introduced above have the same weak and strong generative capacity.

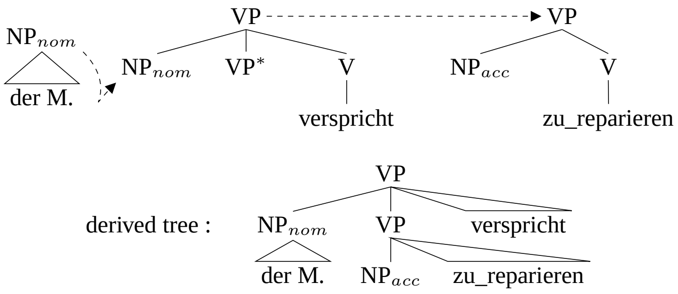

Figure 2. *Part of the TAG analysis of (2)*

bling 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 (2) 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. 2. 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*.[2] This leads to the third tree in Fig. 2. 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 (2), one could add a tree to the grammar 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)).

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. The Condition on Elementary Tree Minimality requires that (positions for) all of the arguments of the lexical anchor of an elementary tree be 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.

[2] The fact that *der Mechaniker* is at the same time logical subject of *reparieren* is accounted for in the semantics, see for example (Gardent & Kallmeyer, 2003; Romero & Kallmeyer, 2005).

One of the first proposals made was an analysis of German scrambling data using non-local multicomponent TAG (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.

The most interesting proposal for a TAG extension for scrambling is V-TAG (Rambow, 1994a; Rambow, 1994b; Rambow & Lee, 1994), a formalism that has nicer formal properties than non-local MCTAG. V-TAG also uses 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. (Rambow, 1994a) proposes detailed analyses of a large range of different word order phenomena in German using V-TAG and thereby shows the linguistic usefulness of V-TAG.

Even though V-TAG 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. Therefore, 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 us to put *integrity constraints* on certain nodes in order to make them act as barriers. This explicit marking of barriers is somewhat against the original appealing TAG idea that such constraints result from the Condition on Elementary Tree Minimality 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 combination possibilities of this structure. In other words, in local formalisms with an extended domain of locality such as TAG such constraints result from the form of the elementary structures and from the locality of the derivation operation. I.e., they follow from general properties of the grammar and they need not be stated explicitely. This is one of the aspects that make TAG so attractive from a linguistic point of view, and it gets lost in non-local TAG variants.

Other TAG variants that can be used for scrambling are D-tree substitution grammars (DSG, (Rambow *et al.*, 2001)) and Segmented Tree Adjoining Grammars (SegTAG, (Kulick, 2000)). 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.

SegTAG (Kulick, 2000) uses an operation on trees called *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 subtrees cannot be identified, and the completely different unification operation. Kulick suggests that SegTAGs are probably mildly context-sensitive but there is no actual proof of this. However, if this is the case, the generative power of the formalism is probably too limited to deal with scrambling in a general way. But it seems that the limit imposed by the grammar on the complexity of the scrambling data is fixed but arbitrarily high. (With increasing complexity the elementary trees however get larger and larger.) This means that one can probably define a SegTAG that can analyze scrambling up to some complexity level n for any $n > 0$. (A definition of what a complexity level is, is not given ; it is perhaps the depth of scrambling.) In this sense, as suggested by Kulick, a general treatment of scrambling might be possible.

Another formalism related to TAG but not conceived as a TAG variant that has been claimed to be able to deal with scrambling is Range Concatenation Grammar (RCG, (Boullier, 2000)). But the RCG scrambling analysis in (Boullier, 1999) assumes predicate-argument dependencies between nouns and verbs to be already known before parsing. However, these dependencies are exactly what one wants to find out when doing the analysis. With this information already given in advance, the analysis is of course easier. So (Boullier, 1999) does not present a general anaysis of scrambling.

All the TAG variants mentioned above 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. Therefore, in this paper, we propose to use tree-local MC-TAG with shared nodes (SN-MCTAG, (Kallmeyer, 2005)), more precisely restricted SN-MCTAG (RSN-MCTAG). This is a TAG-variant that a) can deal with scrambling and other word order variations, b) extends the generative capacity of TAG, i.e., the set of tree adjoining languages (containing the copy language) is a subset of the languages it generates, and c) is polynomially parsable if one imposes some additional restriction.

In section 2, tree-local MC-TAG with shared nodes (SN-MCTAG) and in particular restricted SN-MCTAG (RSN-MCTAG) are introduced. Sections 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)

SN-MCTAG were originally defined in (Kallmeyer, 2005) which includes a discussion of their formal properties.

To illustrate the idea of shared nodes, consider again example (2) from p. 2. 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. 2, 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. We call this locality *SN-tree-locality*.

Let us go back to (2). Assume the tree set on the left of Fig. 3 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 SN-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 $\gamma_1, \dots, \gamma_n$, then in the resulting tree, the root and foot node of β are both considered being part of $\gamma_1, \dots, \gamma_n$ 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. 3 for example the *es*-tree is adjoined to the root of a tree that was adjoined to *reparieren*. Therefore this adjunction can be

[3] Actually, in a Feature-Structure Based TAG (FTAG, (Vijay-Shanker & 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. (However, the use of feature structures in TAG or MCTAG does not increase the generative capacity of the formalism since the MCTAG locality constraints do not depend on the feature structures.)

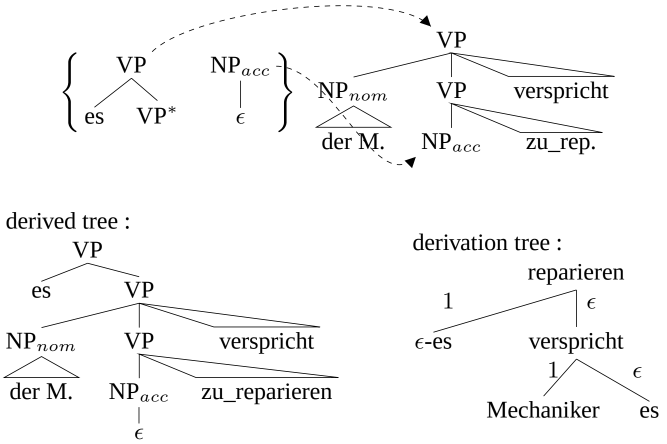

Figure 3. *Derivation of (2) using shared nodes*

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.

One way to define SN-MCTAG (see (Kallmeyer, 2005)) is a definition referring to the standard TAG derivation tree in the following way : Define the grammar as an MCTAG and then allow only derivation trees[4] that satisfy the following SN-tree-locality condition : for each instance $\{\gamma_1, \dots, \gamma_k\}$ of an elementary tree set in the derivation tree there is a γ such that each of the γ_i is either a daughter of γ or it is linked to one of the daughters of γ by a chain of adjunctions at root or foot nodes.

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, (Kallmeyer, 2005) defines 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. In other words, at least one of the γ_i in the definition mentioned above must be an actual daughter of γ. E.g., in Fig. 3, *es* is secondarily adjoined to *reparieren* while the second element of the tree set, the tree ϵ-es, is primarily added (substituted) to *reparieren*, i.e., in the corresponding derivation tree in Fig. 3 it is a daughter of *reparieren*.

[4] Here we do not mean the MCTAG derivation tree defined in (Weir, 1988) but the TAG derivation tree in the TAG underlying the MCTAG. I.e., if $G = \langle I, A, N, T, \mathcal{A} \rangle$ is the MCTAG, then $G_{TAG} = \langle I, A, N, T \rangle$ is the underlying TAG and each derivation in G is at the same time a derivation in G_{TAG} with a corresponding derivation tree.

Obviously, all tree adjoining languages can be generated by RSN-MCTAGs since a TAG is an MCTAG with unary multicomponent sets. In (Kallmeyer, 2005) RSN-MCTAGs of specific arities are defined and it is shown that for each RSN-MCTAG of a specific arity n, an equivalent LCFRS (linear context-free rewriting system, (Weir, 1988)) can be constructed. LCFRSs are mildly context-sensitive and in particular polynomially parsable and therefore, this also holds for RSN-MCTAGs of fixed arity. These RSN-MCTAG perhaps cannot analyze all scrambling phenomena but, if the arity is appropriately chosen, they can analyze a sufficiently large set. The crucial point is that the arity is a complexity level n that is variable.

With scrambling analyses that respect the Condition on Elementary Tree Minimality, the linguistic signification of restricting the arity of the grammar to some n is that the lexical material containing a verb, all its arguments (including arguments and adjuncts of these arguments etc.) and all its adjuncts cannot be separated into more than n discontinuous substrings in the whole sentence. E.g., an RSN-MCTAG of arity 2 with elementary tree sets similar to those proposed above for scrambling would not be able to analyze (3) : In (3) the VP *das Fahrrad zu reparieren zu versuchen* is separated into three discontinuous substrings.

(3) ... dass $[$das Fahrrad$]_1$ er $[t_1$ zu reparieren$]_2$ dem Kunden $[t_2$ zu versuchen$]$
... that the bike he to repair the costumer to try
verspricht
promises
‘ ... that he promises the customer to try to repair the bike’

Depending on the complexity of the data one expects to encounter the arity of the grammar can be chosen arbitraily high. In this sense RSN-MCTAG can deal with scrambling in a general way.

The arity of the grammar excludes certain derivations ; the grammar itself however looks like a general RSN-MCTAG. Therefore, when developing the grammar, the problem of fixing the arity can be left aside. Based on empirical studies, the arity can be determined later. In this paper, we will therefore not deal with the determination of the arity. We will just develop a general RSN-MCTAG.

3. Scrambling

As already mentioned above, in many SOV languages, such as German, Hindi, Japanese and Korean, constituents (arguments or adjuncts) display a larger freedom in term of ordering in clauses. E.g., any permutation of the five constituents (5 ! = 120) in (4), taken from (Rambow, 1994a) is grammatically well-formed. (See (Uszkoreit, 1987) for a description of word order in German and (Lee, 1993) for Korean.)

(4) ... dass [eine hiesige Firma] [meinem Onkel] [die Möbel] [vor drei Tagen]
... that a local company$_{nom}$ my oncle$_{dat}$ the furniture$_{acc}$ tree days ago

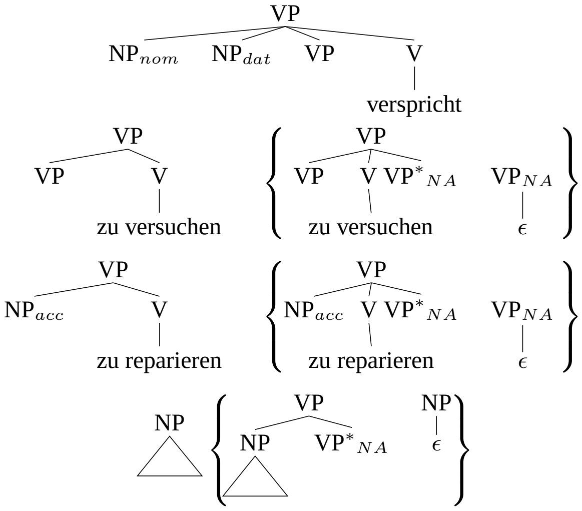

Figure 4. *Elementary trees for word order variations of (6)*

[ohne Voranmeldung] zugestellt hat
without advance warning delivered has
'... that a local company has delivered the furniture to my uncle three days ago without advance warning'

The constituents of a lower clause can even occur in the upper clause, (so-called *long distance* scrambling). E.g., in the German sentence (2), repeated as (5a), and in the Korean sentence (5b) the arguments *es* and *jatongcha-lul* of the embedded verb move into the upper clause.

(5) a. ... dass es_1 der Mechaniker [t_1 zu reparieren] verspricht

b. jatongcha-lul_1 keu-ka [t_1 surihakess-tako] yaksokhaessta
the car_{acc} he_{nom} [t_1 repair-to] promises
'He promises to repair the car'

Generally, in both languages, the characteristics of scrambling can be summarized in a double unboundedness :

i) there is no bound on the distance over which each element can scramble (In (6b), the most deeply embedded NP_{acc} *das Auto* is scrambled into the upper clause.), and

ii) there is no bound on the number of elements that can scramble in one sentence and, furthermore, more than one element can scramble out of a clause (In (7b), two NPs are scrambled out of the embedded clause, and the original order among them need not be retained.).[5]

[5] In Korean, the same types of constructions as in (6b) and (7b) also can be observed.

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

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

(7) a. ... dass der Detektiv dem Klienten [den Verdächtigen des Verbrechens
... that the detective$_{nom}$ the client$_{dat}$ the suspect$_{acc}$ the crime$_{gen}$
zu überführen] versprach
to indict promised
'... that the detective promised the client to indict the suspect of the crime'

b. ... dass des Verbrechens$_2$ dem Klienten den Verdächtigen$_1$ der Detektiv [t_1 t_2 zu überführen] versprach

Even though doubly unbounded, scrambling satisfies some constraints : it always goes to the left, so all scrambling word order variations have to be such that all arguments precede the verbs they depend on. Furthermore, another constraint, at least in German, is that scrambling cannot proceed out of tensed clauses as illustrated in (8) :

(8) a. Peter sagt, dass er [das Auto zu reparieren] versucht
Peter says that he the car to repair tries
'Peter says that he tries to repair the car'

b. *Peter [das Auto]$_1$ sagt, dass er [t_1 zu reparieren] versucht

In the following we will show how MCTAG with Shared Nodes allows to deal with long distance scrambling, and how the different aspects in scrambling can be handled. The elementary trees we use for word order variations of (6a) are shown in Fig. 4. We propose to use single trees for non-scrambled elements and tree sets for scrambled elements.[6] If an element is scrambled, an initial tree with the empty word fills the substitution node while the lexical material is adjoined at a VP node of the verb it depends on.

Consider (6b) where the most deeply embedded NP$_{acc}$ *das Auto* is scrambled into the upper clause. The analysis with the RSN-MCTAG for (6b) is sketched in the following : For *das Auto*, the tree set is used. Further, we also use tree sets for the NP$_{dat}$

[6] The subscript NA ('null adjunction') indicates that adjunctions at that node are disallowed. TAG actually allows for each internal node to specify the set of auxiliary trees that can be adjoined using so-called *adjunction constraints* and, furthermore, to specify whether adjunction at that node is obligatory or not. This is an important feature of TAG since it influences the generative capacity of the formalism : $\{a^nb^nc^nd^n \mid n \geq 0\}$ for example is a language that can be generated by a TAG with adjunction constraints but not by a TAG without adjunction constraints (Joshi, 1985).

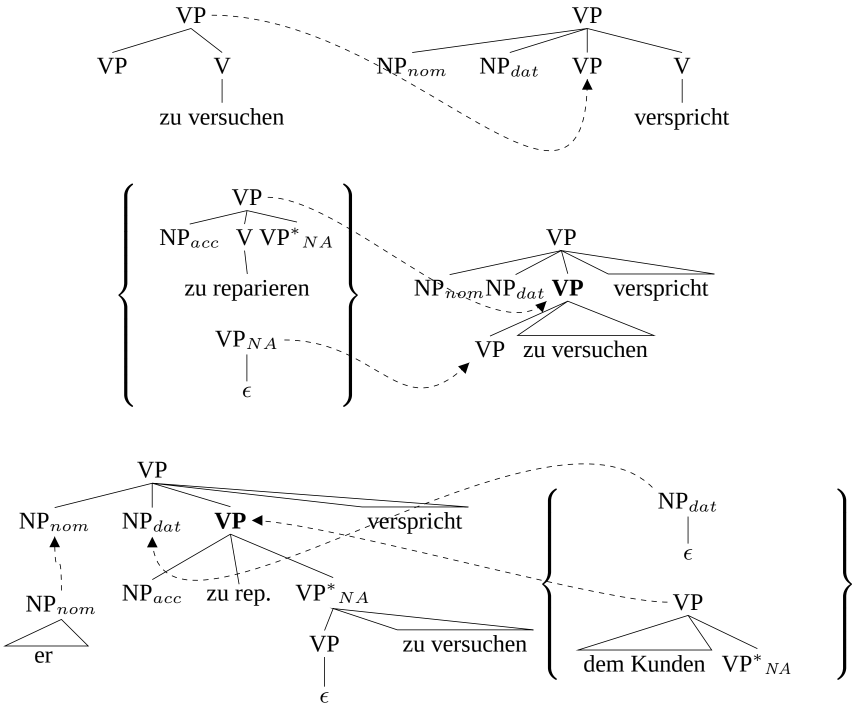

Figure 5. *Derivation for (6b)*

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. 5 shows the different derivation steps for (6b). 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.

With the analysis proposed here, the only condition for adding a tree set or a single tree is that the verb it depends on has already been added since the tree of this verb provides the substitution node for the initial tree. Furthermore, the lexical material is always left of the foot node. One obtains therefore that all arguments precede their verbs.

Additionally, 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. In this way, a barrier effect is obtained without stipulating explicitely a barrier as it is done in V-TAG. Instead, this locality of scrambling is a consequence of the form of the elementary trees and of the SN-tree-locality of the derivations.

As already mentioned, in German, scrambling can never proceed out of tensed clauses. However in Korean, scrambling out of a tensed clause is possible, e.g., in (9) the argument *jatongcha-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.[7]

(9) jatongcha-lul$_1$ keu-ka [kokaek-i t_1 kuiphaess-tako] malhaessta.
the car$_{acc}$ he$_{nom}$ [the customer$_{nom}$ t_1 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 (10), the *reparieren* VP occurs behind the finite verb *verspricht*. The same goes for the Korean extraposition (11).

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

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

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. 6. 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 (10), we use the single trees shown in Fig. 4. The derivation for (10) is as sketched in Fig. 6. The empty VP fills the argument slot while the auxiliary tree for extraposition of *reparieren* adjoins to the higher VP node.

[7] One aspect we did not consider in this paper but that definitely needs to be spelled out is the fact that in both 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.

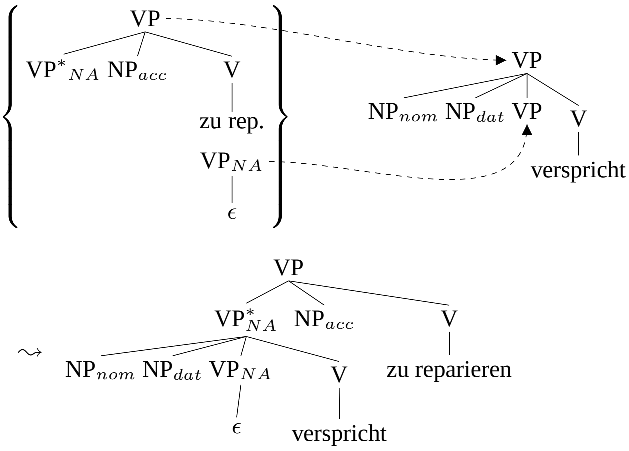

Figure 6. *Derivation for (10)*

German and Korean allow both extraposition of complete VPs. But the following differences between the two languages can be observed : In German, infinitives without their arguments can be extraposed (so-called *third construction*, see (12)a), which is not possible in Korean (see (13)a). In Korean however, arguments of embedded verbs can be extraposed while leaving their verb behind (see (13)b), which is not possible in German (see (12)b).[8]

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

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

To account for the difference between (12a) and (13a), we disallow the adjunction of scrambled elements at the root nodes of Korean auxiliary extraposition trees.

For (13b), in Korean, we propose additional tree sets for extraposed NPs (see Fig. 7). They are similar to the tree sets for scrambled NPs in Fig. 4, except that the foot node is on the left. Such tree sets do not exist in German.

In German, even arguments of embedded VPs can be left behind as in

(14) 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 infitives where deeper embedded infinitives can be added. The corresponding tree sets for German are shown in Fig. 8.

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

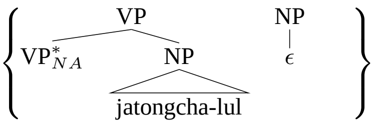

Figure 7. *Tree set for Korean extraposition of NPs*

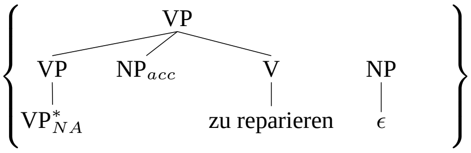

Figure 8. *Trees for German extraposition*

Certain combinations of scrambling and extraposition are not possible : (Rambow, 1994a) examines word order variations of (15) and concludes that some orders are ungrammatical.

(15) Weil niemand [[das Fahrrad zu reparieren] zu versuchen] verspricht
Because nobody the bike to repair to try promises
'Because nobody promises to try to repair the bike'

First, if the order of the verbs is *zu versuchen zu reparieren verspricht*, then *niemand* cannot occur between *zu versuchen* and *zu reparieren*. I.e., the orders in (16) are not possible. Furthermore, if the order of the verbs is *zu versuchen verspricht zu reparieren*, then *das Fahrrad* cannot occur between *zu versuchen* and *verspricht*. I.e., the orders in (17) are not possible.

(16) a. *Weil zu versuchen das Fahrrad niemand zu reparieren verspricht
b. *Weil das Fahrrad zu versuchen niemand zu reparieren verspricht
c. *Weil zu versuchen niemand das Fahrrad zu reparieren verspricht

(17) a. *Weil niemand zu versuchen das Fahrrad verspricht zu reparieren
b. *Weil zu versuchen niemand das Fahrrad verspricht zu reparieren
c. *Weil zu versuchen das Fahrrad niemand verspricht zu reparieren

In the following, we will show that with the analysis proposed here these orders can in fact be excluded.

First let us examine the examples in (16). In order to obtain the order *versuchen reparieren verspricht*, one has to use the extraposition tree set for *reparieren* in order to place it to the right of *versuchen*. Furthermore, we assume an additional modification of the extraposition auxiliary trees, namely that the label of root and foot nodes is underspecified such that adjunction at VP and also V nodes is allowed. For *versuchen*,

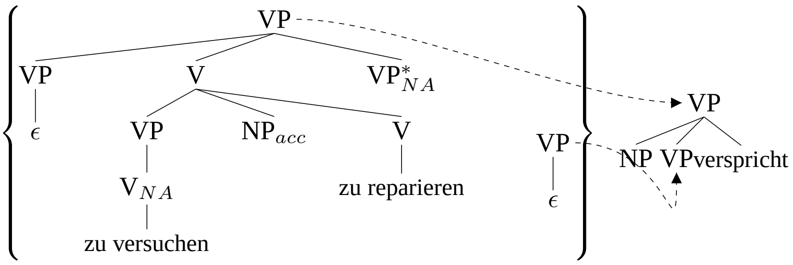

Figure 9. *Extraposition of* reparieren *past* versuchen

we assume that in the single tree, adjunction of extraposition auxiliary trees is not allowed. This can be obtained using an appropriate feature. Consequently we have to use the scrambling tree set, adjoining the *reparieren* auxiliary tree at the V node. This means that one obtains the derived tree set on the left of Fig. 9 for *zu versuchen zu reparieren* and combines this with the initial tree for *verspricht*. Concerning *niemand*, it can either be added on top of everything which places it before *zu versuchen* or it can be added between the *verspricht* tree and the auxiliary tree derived for *zu versuchen zu reparieren*. This places it between *reparieren* and *verspricht*. The same order is obtained if the single tree is used for *niemand*. Consequently, the orders in (16) with *niemand* between *versuchen* and *reparieren* are excluded.

Consider now the examples in (17). Since *reparieren* is extraposed past *verspricht*, we have to use the scrambling tree set for *versuchen* and the extraposition tree set for *reparieren*. The auxiliary tree for *versuchen* adjoins to the root of *verspricht* and then the auxiliary tree for *reparieren* adjoins on top of it. This is sketched in Fig. 10. In the resulting derived tree, only the bold VP nodes are possible adjunction sites for an auxiliary tree for *Fahrrad*. Consequently, it is impossible to put *Fahrrad* between *versuchen* and *verspricht* and the orders in (17) cannot be derived.

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., *jatongcha-nun* in (18a.) : an element marked by -*nun*(-*un*) can also appear in sentence medial position e.g., *jatongcha-nun* in (18b.). 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 the specifier position of the CP (see for example (Suh, 2002)).

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

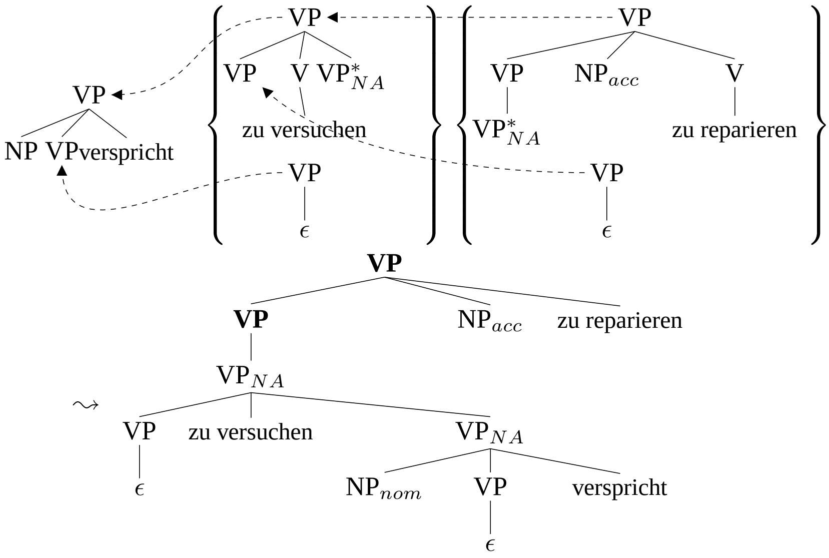

Figure 10. *Extraposition of reparieren past verspricht*

b. keu-ka jatongcha-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 (19) the auxiliary verb *hat* appears in second position, the NP_{acc} *das Buch* that moved from the MF into the first position is topicalized.

(19) das $Buch_2$ hat ihm_1 niemand [t_1 t_2 zu geben] versucht.
the $book_{acc}$ has him_{dat} 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.[9] The last step in a

[9] We use feature structures exactly as in FTAG : each internal node or foot node has a top and a bottom feature structure, substitution nodes have top feature structures. In a substitution

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.

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 (20).

(20) [das Auto]$_1$ hat er [t_1 zu reparieren]$_2$ dem Kunden [t_2 zu versuchen]
the car has he to repair the customer to try

versprochen.
promised
'he has promised the customer to try to repair the car'

Topicalization and *scrambling* can occur simultaneously as in (19) where *ihm* is long-distance scrambled and *das Buch* is long-distance topicalized. Fig. 11 shows the derivation for (19) : 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, 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 top 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 proposed to use 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

operation the top of the substitution node is unified with the top of the root of the tree that is added. In an adjunction of a tree β to a node μ, the top of μ is unified with the top of the root of β and the bottom of μ is unified with the bottom of the foot node of β. As in the case of standard TAG, the use of feature structures instead of simple non-terminals does not increase the generative capacity of the formalism as long as only a finite number of feature structures is possible.

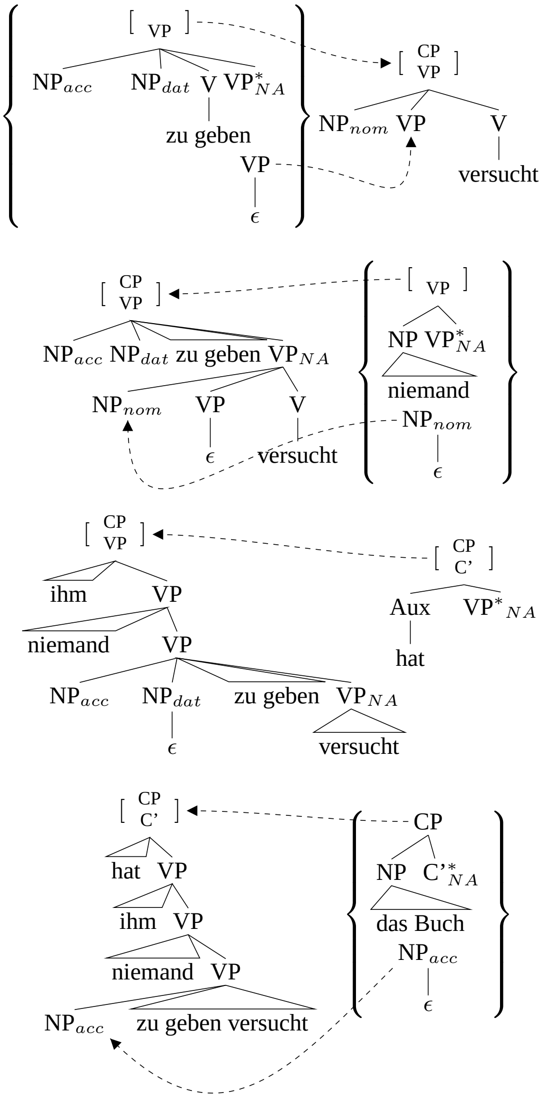

Figure 11. *Derivation for (19)*

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 extension of TAG gives sufficient additional power to analyse the difficult scrambling data.

As shown in (Kallmeyer, 2005), under certain restrictions, RSN-MCTAG belong to the class of mildly context-sensitive grammar formalisms and are therefore polynomially parsable. The restrictions one has to impose limit the complexity of the scrambling data one can deal with. But, depending on empirical investigations, the limit can be chosen arbitrarily high, and, furthermore, it does not influence the form of the elementary trees (i.e., the grammar itself). It only excludes certain derivations. In other words, one can develop a RSN-MCTAG as sketched in this paper, and then one can fix an appropriate complexity limit for the data one wants to analyse.

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.

7. Bibliographie

Becker T., Joshi A. K. & Rambow O. (1991). Long-distance scrambling and tree adjoining grammars. In *Proceedings of ACL-Europe*.

Boullier P. (1999). Chinese numbers, mix, scrambling, and range concatenation grammars. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL'99)*, p. 53–60, Bergen, Norway.

Boullier P. (2000). Range Concatenation Grammars. In *Proceedings of the Sixth International Workshop on Parsing Technologies (IWPT2000)*, p. 53–64, Trento, Italy.

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

Frank R. (1992). *Syntactic Locality and Tree Adjoining Grammar : Grammatical, Acquisition and Processing Perspectives*. PhD thesis, University of Pennsylvania.

Gardent C. & Kallmeyer L. (2003). Semantic Construction in FTAG. In *Proceedings of EACL 2003*, Budapest.

Joshi A. K. (1985). Tree adjoining grammars : How much contextsensitivity is required ro provide reasonable structural descriptions ? In D. Dowty, L. Karttunen & A. Zwicky, RÃ©dacteurs, *Narural Language Parsing*, p. 206–250. Cambridge University Press.

Joshi A. K. (1987). An introduction to Tree Adjoining Grammars. In A. Manaster-Ramer, RÃ©dacteur, *Mathematics of Language*, p. 87–114. Amsterdam : John Benjamins.

Joshi A. K., Becker T. & Rambow O. (2000). Complexity of scrambling : A new twist to the competence/performance distinction. In A. Abeillé & O. Rambow, RÃ©dacteurs, *Tree Adjoining Grammars : Formalisms, Linguistic Analyses and Processing*. CSLI.

Joshi A. K., Levy L. S. & Takahashi M. (1975). Tree Adjunct Grammars. *Journal of Computer and System Science*, **10**, 136–163.

Joshi A. K. & Schabes Y. (1997). Tree-Adjoning Grammars. In G. Rozenberg & A. Salomaa, RÃ©dacteurs, *Handbook of Formal Languages*, p. 69–123. Berlin : Springer.

Joshi A. K. & Vijay-Shanker K. (1999). Compositional Semantics with Lexicalized

Tree-Adjoining Grammar (LTAG) : How Much Underspecification is Necessary ? In H. C. BLUNT & E. G. C. THIJSSE, RÃ©dacteurs, *Proceedings of the Third International Workshop on Computational Semantics (IWCS-3)*, p. 131–145, Tilburg.

KALLMEYER L. (2005). Tree-local multicomponent tree adjoining grammars with shared nodes. *Computational Linguistics*, **31**(2), 187–225.

KALLMEYER L. & JOSHI A. K. (2003). Factoring Predicate Argument and Scope Semantics : Underspecified Semantics with LTAG. *Research on Language and Computation*, **1**(1–2), 3–58.

KULICK S. N. (2000). *Constraining Non-local Dependencies in Tree Adjoining Grammar : Computational and Linguistic Perspectives*. PhD thesis, University of Pennsylvania.

LEE Y.-S. (1993). *Scrambling as Case-driven Obligatory Movement*. PhD thesis, University of Pennsylvania. Published as technical report IRCS-93-06.

MEURERS W. D. (2000). *Lexical Generalizations in the Syntax of German Non-Finite Constructions*. PhD thesis, Universität Tübingen.

MÜLLER S. (2002). *Complex Predicates : Verbal Complexes, Resultative Constructions, and Particle Verbs in German*. CSLI Stanford.

RAMBOW O. (1994a). *Formal and Computational Aspects of Natural Language Syntax*. PhD thesis, University of Pennsylvania.

RAMBOW O. (1994b). Multiset-Valued Linear Index Grammars : Imposing dominance constraints on derivations. In *Proceedings of ACL*.

RAMBOW O. & LEE Y.-S. (1994). Word order variation and Tree-Adjoining Grammars. *Computational Intelligence*, **10**(4), 386–400.

RAMBOW O., VIJAY-SHANKER K. & WEIR D. (2001). D-Tree Substitution Grammars. *Computational Linguistics*.

ROMERO M. & KALLMEYER L. (2005). Scope and Situation Binding in LTAG using Semantic Unification. In *Proceedings of IWCS-6*, Tilburg.

SCHABES Y. (1990). *Mathematical and Computational Aspects of Lexicalized Grammars*. PhD thesis, University of Pennsylvania.

SUH C.-M. (2002). Topicalization and Focusing in Korean. In *The Twelfth International Conference on Korean Linguistics*, p. 511–522.

USZKOREIT H. (1987). *Word Order and Constituent Structure in German*. CSLI Stanford.

VIJAY-SHANKER K. & JOSHI A. K. (1988). Feature structures based tree adjoining grammar. In *Proceedings of COLING*, p. 714–719, Budapest.

WEIR D. J. (1988). *Characterizing mildly context-sensitive grammar formalisms*. PhD thesis, University of Pennsylvania.