From Partial VP Fronting towards Spinal TT-MCTAG*

Timm Lichte
Emmy Noether Research Group
Collaborative Research Center 833
University of Tübingen
`timm.lichte@uni-tuebingen.de`

Abstract

In the face of partial fronting phenomena in German, we introduce *spinal TT-MCTAG*, a new MCTAG variant that integrates features of LTAG-spinal and TT-MCTAG. Using spinal TT-MCTAG we arrive at flat syntactic structures which make available a consistent account for the data.

1 Introduction

While the examination of coherent constructions in German has resulted in the design of TAG-extensions such as V-TAG (Rambow, 1994) and TT-MCTAG (Lichte, 2007), which can cope with a good deal of critical data, a remaining desideratum for both accounts is the analysis of embedded partial fronting of verbal heads, exemplified in (1).

(1) Zu reparieren versprochen hat ihn Peter.
to repair promised has it Peter
'Peter has promised to repair it.'

Here, the fronted material *zu reparieren versprochen* embeds the non-finite verb *zu reparieren*, whose remote complement *ihn* is on the other side of the finite verb *hat*.

To see the problem, consider the slightly simpler instance of partial VP fronting in (2), where no additional embedding of a verbal head takes place.

(2) Zu reparieren versprach ihn Peter.
to repair promised it Peter
'Peter promised to repair it.'

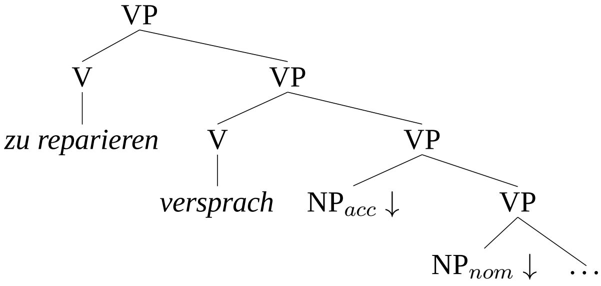


Figure 1: Right-branching derived tree for (2).

The intended derived tree for (2) would be the one in Fig. 1. In terms of TT-MCTAG, this would be derivable with the tree tuples in Fig. 2. A tree tuple consists of two components, namely a single elementary tree, called the *head tree*, and a set of auxiliary trees, called the *argument trees*. The usage of tree tuples is constrained in the following way: each argument tree either adjoins directly at the head tree, or indirectly under *node sharing*, i.e. in the derivation tree the head tree dominates an auxiliary tree γ and γ dominates the argument tree through a path of adjunctions at the root node.[1] Crucially, the tree tuples in Fig. 2 do not contain lexically anchored heads, which can be regarded as a downside, since it dissolves the encoding of the dependency relation. We refer to this desirable, yet dismissed property as the *head tree constraint*.

The schema of the tree tuples in Fig. 2 is reminiscent of the elementary tree sets that are used in the V-TAG approach in (Rambow, 1994), depicted in Fig. 3. Note that V-TAG basically is a non-local

---

[*] I am indebted to Laura Kallmeyer for helpful comments.

[1] See (Kallmeyer, 2009) for a formal explication.

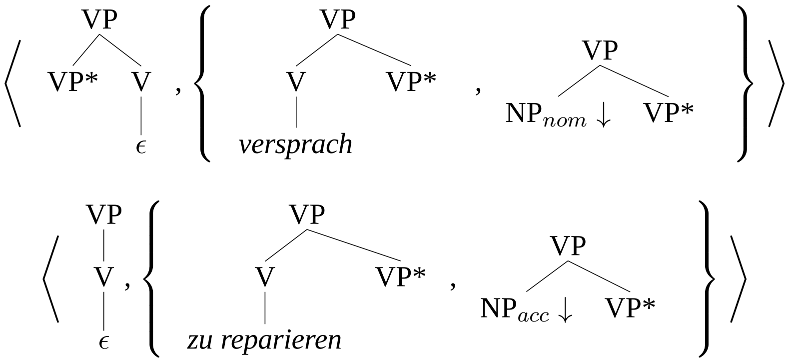


Figure 2: Tree tuples for the derived tree in Fig. 1.

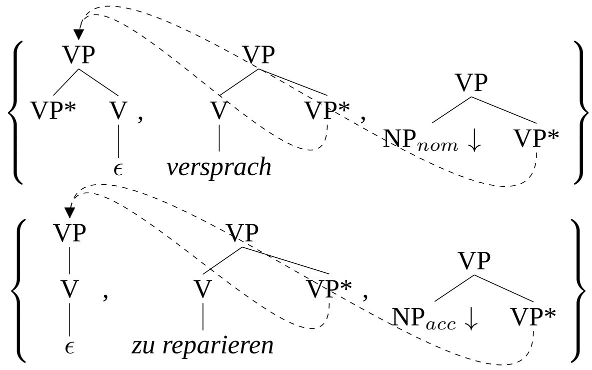


Figure 3: The corresponding V-TAG tree sets of the tree tuples in Fig. 2. Dominance links are expressed by dashed arrows.

MCTAG, where locality is recovered by dominance links (indicated as dashed arrows) and integrity constraints, that refer to the derived tree.

Both approaches essentially rely on the existence of a dominance relation between the elementary trees of the respective multicomponent structures, be it in the derivation tree for TT-MCTAG, or in the derived tree for V-TAG. While such a dominance relation can be found in Fig. 1, the intended derived structure for (1) essentially is the one in Fig. 4, which has a complex prefield constituent[2]. Here, no dominance relation of the embedded verbal head *zu reparieren* and its argument *ihn* can be established, and therefore, this structure cannot be derived in a linguistically appealing way no matter whether we choose TT-MCTAG or V-TAG.

As mentioned in (Lichte, 2007), the extension of node sharing to tree sharing could solve this dilemma in the case of TT-MCTAG. However, the exact complexity class being unknown, tree sharing seems to extend complexity somewhat in practice. Moreover, it is unclear, how such an extension would transfer to V-TAG. Note that, other than (Gerdes, 2004), we aim at an analysis which restricts itself immediately through the formalism that derives the syntactic structure.

2 Adapting the derived structures

Our strategy is to adapt the derived syntactic structure such that we obtain a dominance relation between the head and its argument both in the derivation tree and the derived tree. It has been already mentioned in (Lichte, 2007) that fronting phenomena no more pose a problem if the derived structure is left branching, such as in in Fig. 5. Both

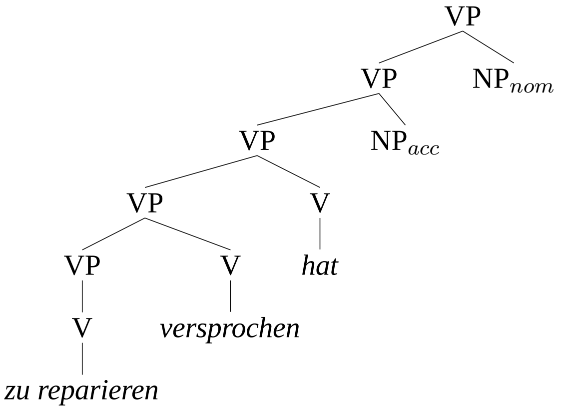


Figure 5: Left branching derived tree for (1).

---

[2] The prefield in German immediately precedes the finite verb in verb second configurations. In general, it is occupied by one single constituent.

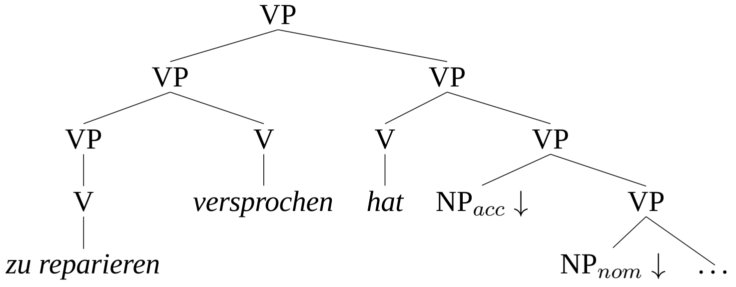


Figure 4: Left- and right-branching derived structure of the embedded partial VP fronting datum in (1).

the TT-MCTAG approach and the V-TAG approach then would suffice. This adaptation, however, is not desirable since, amongst others, the argument trees would also be required to be left-branching, leading to massive ambiguity in the lexicon due to the availability of right-branching and left-branching solutions.

Instead, we propose a flat derived structure for the complex partial fronting case, as sketched in Fig. 6, in which the NP-arguments are immediate daughters of the VP-root. Doing this allows for a unified account of fronting cases and cases of canonical word order.

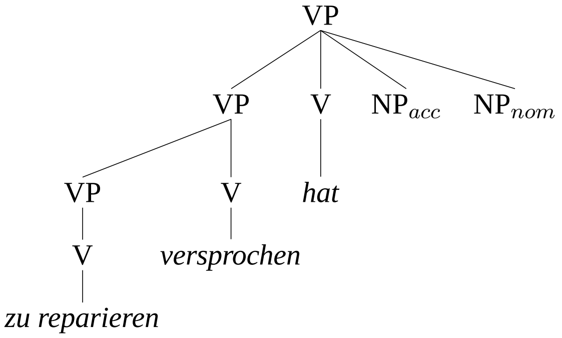


Figure 6: Flat derived tree for (1).

3 Spinal TT-MCTAG

Aiming at a flat derived structure such as in Fig. 6, we introduce a new TT-MCTAG variant that ties in with ideas recently laid out under the name LTAG-spinal in (Shen, 2006). In place of substitution, LTAG-spinal uses a rewriting operation called attachment, which is congruent with sister adjunction (Rambow et al., 1995; Chiang, 2003). Combining two trees γ_i and γ_j via attachment means that in the resulting tree one inner node v_i of γ_i dominates the root node v_j of γ_j, such that γ_j immediately precedes or follows the subtree dominated by v_i in γ_i. See Fig. 7 for an example from (Shen, 2006). Both arguments and modifiers are integrated by attachment, and thus elementary trees receive a "spinal" shape.

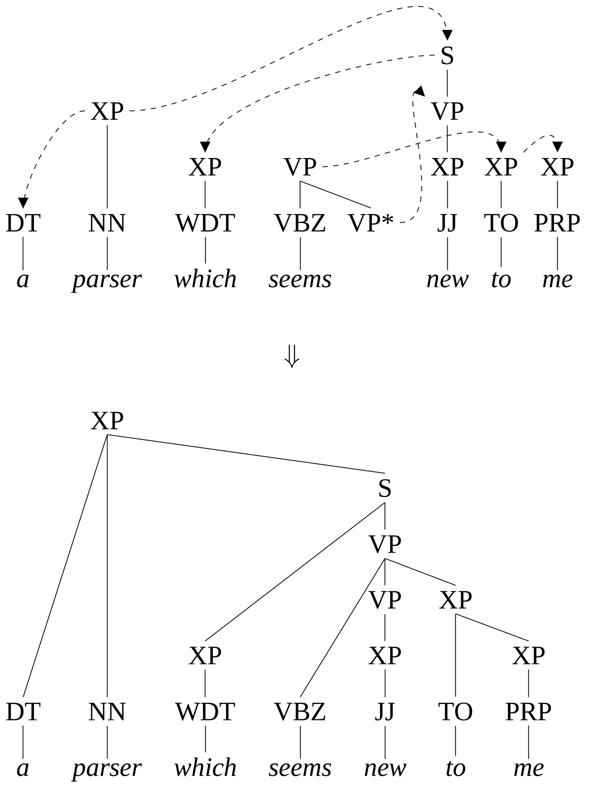
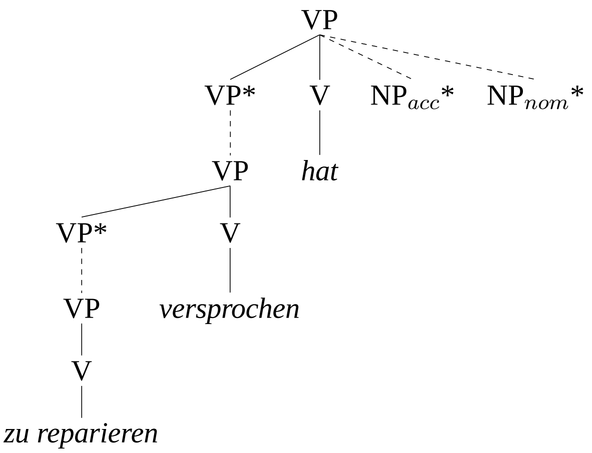


Figure 7: LTAG-spinal derivation example.

Spinal TT-MCTAG with attachment

If we supply TT-MCTAG with attachment analogously to LTAG-spinal, the result provides sufficient means to account for (1), as shown in Fig. 8. Other than with regular TT-MCTAG, arguments are realized by auxiliary trees with a single node. Furthermore, attachment takes over the role of substitution in that it defines islands for argument head dislocations, while node sharing still relies on root node adjunction.

Lexical partition of the derived tree:



Lexical entries:

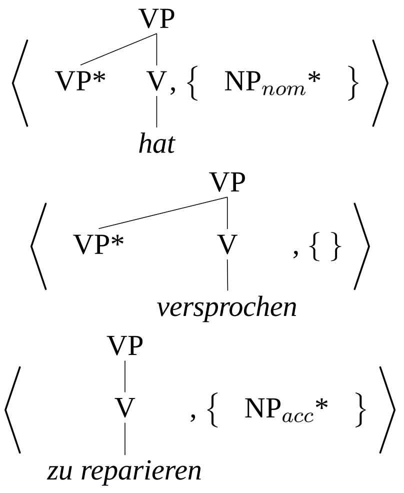


Figure 8: Derivation and lexical entries according to TT-MCTAG with attachment.

However, spinal TT-MCTAG with attachment is not without severe drawbacks. Since NPs can get attached to the head tree as unrestrictedly as modifiers, nothing so far prevents nominative NPs from attaching to the head tree in any number and any order. The way of licensing of nominal arguments by adjoining auxiliary trees from the argument set only requires the existence of proper NPs. One could apply some kind of downstream semantic filter, but we will explore a syntactic solution in the second version of spinal TT-MCTAG. More importantly, while embedded partial VP-fronting can be accounted for now, new gaps open concerning the coverage of other partial VP-fronting phenomena, such as in (3).

(3) Zu reparieren hat er ihn versprochen.
to repair has he it promised
'He has promised to repair it.'

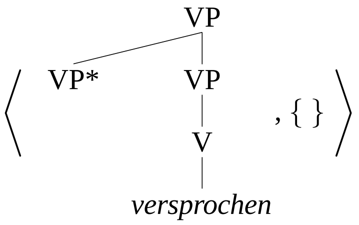


Figure 9: Tree tuple for *versprochen* in (3).

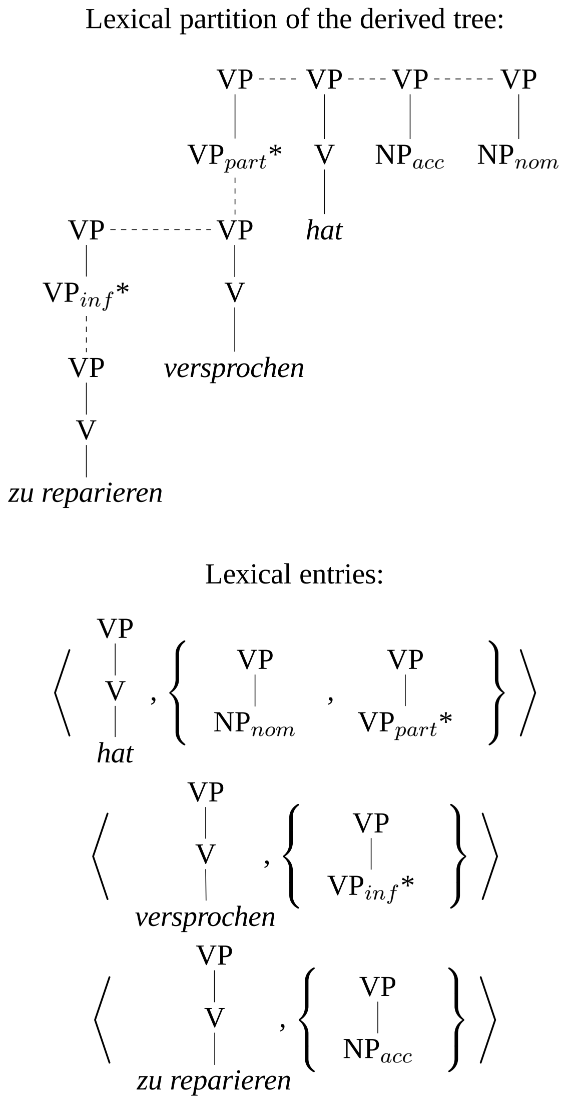
Other than in (1) and (2), the verb *versprochen* and the head of its verbal argument *zu reparieren* are not adjacent, but separated by the finite auxiliary *hat* and one argument from each of the full verbs. Since the tree of *versprochen* would still have to adjoin to the tree of *zu reparieren* in order to allow for the dislocation of its argument *ihn*, the tree tuple for *versprochen* would look as in Fig. 9, including an additional lower VP-node. This lower VP-node would be essential for providing a landing site for the wrapped material, i.e. *hat*, *er* and *ihn*. The result would be, however, that the argument tree of *zu reparieren* (that adjoins into the tree of *ihn*) would not be able to attach at the root node of the tree of *versprochen* and the node sharing relation between *zu reparieren* and its argument would be lost.

This problem is not at all new, but echos the situation of the original TT-MCTAG account as described above. And again, neglecting the head tree constraint would help. Alternatively, one could think of modifying the current version of spinal TT-MCTAG with attachment (e.g., by reactivating substitution). But instead of this, I will introduce a second version of spinal TT-MCTAG, that successfully circumvents this concession.

Spinal TT-MCTAG with fusion

Instead of attachment, we make use of a similar but novel rewriting operation that we refer to as *fusion*. The fusion operation is the amalgamation of single nodes rather than the drawing of a new edge. More formally spoken: If two nodes v_i, v_j of trees γ_i, γ_j are fused, in the resulting tree (i) they are replaced by a node v', for which it holds that all in-going edges of v_i, v_j now point to v' and (ii) the subtrees dominated by v_i and v_j are immediately adjacent and dominated by v' in the resulting tree. We restrict fusion to pairs of nodes, of which at least one node is the root node of the respective tree, in order to maintain the tree shape of the derived tree. Furthermore, it holds that the categorial labels of the fused nodes must be identical.

An important split then is between fusion of root nodes and fusion of a root and a non-root node: the former one, but not the latter one, is non-embedding in that the affected trees are equivalent in the derivation process. In that respect, fusion at inner nodes bears more similarity to attachement and multiple adjunction (Schabes and Shieber, 1994). Fusion in general, however, integrates both arguments and modifiers. The derived tree in Fig. 6 is then the result of the derivation and the lexical entries in Fig. 10. Note that adjunction only applies to the root node of target trees. The division of labor is the following: adjunction extends locality, while fusion at an inner node parallels substitution and defines islands of locality. Hence, the argument set of tree tuples consists of spinal trees that have non-terminal leaves (i.e. the argument slots) and that either are initial or auxiliary trees. To give an example, the NP-slots in the argument sets of the tuples in Fig. 10 constitute islands, whereas the VP-slots do not. The derivational meaning of tree tuples is then the following: The argument trees are (directly or indirectly) fused with the head tree, otherwise the argument trees stand in a node sharing relation to the head tree based on the derivation tree.


Lexical partition of the derived tree:

Lexical entries:

Figure 10: Derivation and lexical entries according to spinal TT-MCTAG with fusion.

Other than the proposal with attachment, it is now possible to underspecify the relative position of the head anchor and the verbal complement. The derivation of the partial fronting case in (3), therefore, does not require concessions such as the violation against the head tree constraint. In fact, it does not even require further lexical variation such that the tree tuples in Fig. 10 suffice also to this end.

This shift to the fusion operation, however, has significant effects on the nature of the derivation tree and thus on the notion of node sharing. Moreover, it is necessary to define a regulation method for fusion which differs from usual feature-unification-based approaches. Both issues are covered separately in the next two sections.

4 The new face of the derivation tree

Since fusion at the root node is understood as being inherently non-embedding, it is indicated with chains as nodes in the derivation tree. Edges are then used for the representation of fusion to some non-root node, and for the representation of adjunction. In other words, edges are dominance relations, whereas nodes represent precedence relations. Hence, the derivation in Fig. 10 receives the derivation tree in Fig. 11. Note that the edge label indicates adjunction (A) or fusion (F) followed by the tree label of the embedding tree. Fusion labels furthermore contain the Gorn address of the embedding tree.[3] Other than in TAG derivation trees, auxiliary trees dominate their target since adjunction is only necessary at the root node.

Such derivation trees can be defined in the following way: A **spinal TT-MCTAG derivation tree** is a tuple $D = \langle C, V, E \rangle$ with labelling functions $l_E : E \rightarrow L_E$ and $l_V : V \rightarrow L_V$, where V is the set of nodes, C is the set of chains[4] with $C = V \times 2^{V \times V}$ and E is the set of edges with $E = C \times C$. It holds that E is a tree over C. For each $v \in V$ there is exactly one $\zeta \in C$ with $\zeta = \langle V_\zeta, E_\zeta \rangle$, such that $v \in V_\zeta$.

The idea of node sharing is to constrain the path between the head and its argument in a derivation tree. Elementary trees, however, now correspond to nodes of chains. This can be accounted for in the following way: Given a spinal TT-MCTAG derivation tree $D = \langle C, V, E \rangle$, a **path** P between nodes v_i, v_j in chains $\zeta_i, \zeta_j \in C$ is a subset of E, such that $\zeta_i \overset{*}{\rightarrow}_P \zeta_j$.

Therefore, the path from the argument NP_{acc} to its head `zu_reparieren` in Fig. 11 is the edge label sequence A.VP$_{part}$, A.VP$_{inf}$.

5 Adapting the node sharing relation

Having explained paths in such derivation trees, we can now specify the node sharing relation that is essential for the derivational meaning of tree tuples: Given two nodes $v_i, v_j \in V$in a spinal TT-MCTAG derivation tree D, v_i is in the node sharing relation to v_j , iff all edges in the path P from v_i to v_j according to D have the label A.TID, with TID being a tree label. This excludes edges with label F.TID.p, $p > 0$. A node sharing relation of this kind holds for the nodes with label NP_{acc} and `zu_reparieren` in the derivation tree in Fig. 11. Note that, contrary to the original definition of node sharing, the argument now dominates the head in the derivation tree.

Finally, we can explicate, what a well-formed derivation tree for a spinal TT-MCTAG G is: Given a spinal TT-MCTAG derivation tree $D = \langle C, V, E \rangle$, if $v_1, \ldots, v_n \in V$ are pairwise different nodes for which it holds that $l_V(v_i) = \gamma$ for $1 \leq i \leq n$ with γ being the head tree of a tree tuple $\langle \gamma, A \rangle$ in G, then for each $\gamma' \in A$, there are pairwise different nodes $u_1, \ldots, u_n \in V$ with $l_V(u_i) = \gamma'$ for $1 \leq i \leq n$. Furthermore, u_i and v_i are members of a chain $\zeta \in C$, or u_i is a member of $\zeta_u \in C$ and v_i is a member of $\zeta_v \in C$ and $\zeta_v \rightarrow \zeta_u$, or u_i is in a node sharing relation to v_i.

This also holds for the derivation tree in Fig. 11.

6 The regulation of fusion

Substitution and adjunction is usually regulated by using some kind of feature unification, also referred to as top-bottom unification. This has to be adapted in the case of attachment, since attachment, other than substitution and adjunction, needs to be regulated also with respect to the direction of attachment.[5] Fusion, on the other side, does not seem to be compatible with a feature unification account due to its non-embedding nature. Instead, we propose and briefly sketch a novel regulation method, where node labels refer to recursive transition networks (RTN,(Woods, 1970)). RTNs are named finite state automata where transitions may additionally depend on successful calls of further RTNs. Other than regular finite state automata, RTNs are weakly equivalent with CFGs. We use RTNs in the following way: a categorial label of a node in a elementary tree, say VP$_{fin}$, does not stand for a set of features, but maps onto an RTN as depicted in Fig. 12, such that fusion effects state transitions rather than feature unifications. This implies a strict order on the application of fusion from the left to the right. While the non-terminals AP, NP and VP point to respective RTNs, the POS-labels V$_{fin}$ and PART(ICLE) can

---

[3] The Gorn address of the root node is ε while the Gorn address of the ith daughter of a node with Gorn address p is $p \cdot i$.

[4] Chains are trees where the nodes have out-degree and in-degree of at most 1.

[5] C.f. sister adjunction constraints (SAC) from (Rambow et al., 1995).

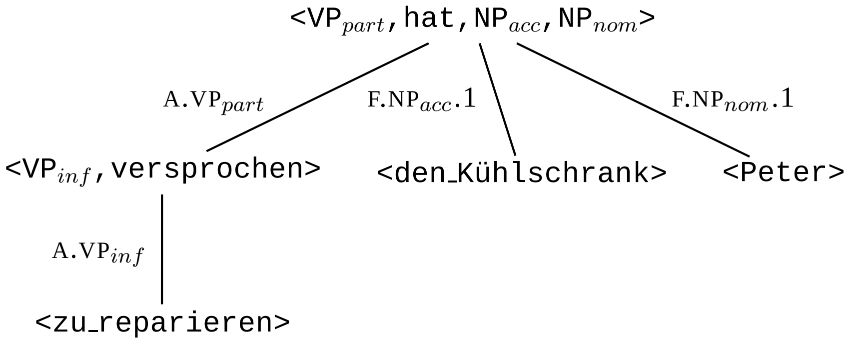


Figure 11: Derivation tree for (1) according to the spinal TT-MCTAG in Fig. 10.

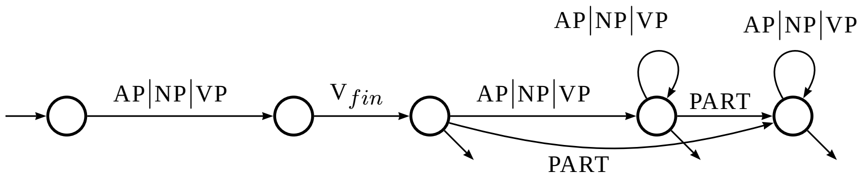


Figure 12: A recursive transition network for the label VP_{fin}, i.e. a finite clause.

be regarded as terminal symbols. Note that the provided prototype of a VP_{fin}-RTN straightforwardly accounts for the prefield conditions for German - the conditions being that the prefield, i.e. the preverbal position, must be occupied and there is exactly one constituent that occupies it.

7 The generative power of spinal TT-MCTAG

From a linguistic point of view, one central ingredient of mild context-sensitivity certainly is the potential for the analysis of cross-serial dependencies. While German usually serves as an exemplar of a center embedding language, it also allows for cross-serial dependencies (to some degree) due to the flexible order of the nominal arguments. This can be observed, e.g., in (4).

(4) dass den Kühlschrank ihm Peter reparieren half
that the fridge him Peter repair helped
'that Peter helped him to repair the fridge'

This kind of cross-serial dependency guided by case is derivable in both versions of a spinal TT-MCTAG. The order of the NP sequence and the verbal complex is basically independent. In Dutch, however, where the mapping of verbs and arguments depends on their relative order, this does not suffice. The assumed generalization is that the ith noun can be only the subject of the ith verb, but counting so far is not supported by spinal TT-MCTAG.

(5) dat Jan Piet Marie zag helpen swemmen
that Jan Piet Marie saw help swim
'that Jan saw Piet helping Marie to swim'

8 Conclusion

Certain partial VP fronting phenomena in German seem to pose an intractable problem for currently available MCTAG variants for German, i.e. V-TAG and TT-MCTAG. This paper therefore proposed to aim at flatter derived structures and investigated ways to modify TT-MCTAG, in order to generate them. Ideas for two novel variants of TT-MCTAG, spinal TT-MCTAG with attachment and spinal TT-MCTAG with fusion, were sketched, which both offer means to account for the data in question. It turned out that spinal TT-MCTAG with fusion performs better, since it is straightforwardly applicable to other phenomena of flexible word order without violating the head tree constraint, contrary to spinal TT-MCTAG with attachment. Another major advantage is that the number of lexical entries considerably reduces due to the spinal shape of the head tree. In return, the shape of the derivation tree had to be modified, replacing atomic node labels by chains, which correspond to the non-embedding nature of the fusion operation. For the regulation of fusion, we proposed to use recursive transition networks instead of feature unification. These modifications due to fusion certainly are far-reaching, but we think that they are far from being mere technical repairs. RTNs, for example, offer interesting means to express syntactic generalizations.

Certainly, the current paper does not present a complete picture of the proposal, and there are many aspects, e.g. complexity issues and the regulation by RTNs, that have to be worked out in further research.

References

Manuel Bodirsky, Marco Kuhlmann, and Mathias Möhl. 2005. Well-nested drawings as models of syntactic structure. In *In 10th Conference on Formal Grammar and 9th Meeting on Mathematics of Language (FG-MOL05)*, Edinburgh.

David Chiang. 2003. Statistical parsing with an automatically extracted tree adjoining grammar. In Rens Bod, Remko Scha, and Khalil Sima'an, editors, *Data Oriented Parsing*, pages 299–316. CSLI Publications.

Kim Gerdes. 2004. Tree Unification Grammar. In Lawrence S. Moss and Richard T. Oehrle, editors, *Electronic Notes in Theoretical Computer Science*, volume 53. Elsevier. Proceedings of the joint meeting of the 6th Conference on Formal Grammar and the 7th Conference on Mathematics of Language.

Laura Kallmeyer. 2009. A declarative characterization of different types of multicomponent tree adjoining grammar. *Research on Language and Computation*, 7:55–99.

Marco Kuhlmann and Mathias Möhl. 2006. Extended cross-serial dependencies in Tree Adjoining Grammars. In *Proceedings of the 8th International Workshop on Tree Adjoining Grammar and Related Formalisms*, pages 121–126, Sydney.

Timm Lichte. 2007. An MCTAG with tuples for coherent constructions in German. In *Proceedings of the 12th Conference on Formal Grammar*. Dublin, Ireland, 4-5 August 2007.

Owen Rambow, K. Vijay-Shanker, and David Weir. 1995. D-tree grammars. In *Proceedings of the 33rd Annual Conference of the Association for Computational Linguistics*, Cambridge, MA.

Owen Rambow. 1994. *Formal and Computational Aspects of Natural Language Syntax*. Ph.D. thesis, University of Pennsylvania, Philadelphia. IRCS Report 94-08.

Yves Schabes and Stuart Shieber. 1994. An alternative conception of tree-adjoining derivation. *Computational Linguistic*, 20(1):91–124.

Libin Shen. 2006. *Statistical LTAG Parsing*. Ph.D. thesis, University of Pennsylvania.

William A. Woods. 1970. Transition network grammars for natural language analysis. *Commun. ACM*, 13:591–606, October.