Factorizing Complementation in a TT-MCTAG for German

Timm Lichte
Emmy-Nother-Nachwuchsgruppe
SFB 441
University of Tübingen
timm.lichte@uni-tuebingen.de

Laura Kallmeyer
Emmy-Nother-Nachwuchsgruppe
SFB 441
University of Tübingen
lk@sfs.uni-tuebingen.de

Abstract

TT-MCTAG lets one abstract away from the relative order of co-complements in the final derived tree, which is more appropriate than classic TAG when dealing with flexible word order in German. In this paper, we present the analyses for sentential complements, i.e., wh-extraction, that-complementation and bridging, and we work out the crucial differences between these and respective accounts in XTAG (for English) and V-TAG (for German).

1 Introduction

Classic TAG is known to offer rather limited (Becker et al., 1991) and unsatisfying ways to account for flexible word order in languages such as German. The descriptive overhead is immediately evident: Every possible relative order of co-complements of a verb, has to be covered by an extra elementary tree. To give an example from German, the verb *vergisst* (forgets) with two complements would receive two elementary trees in order to license the verb final configurations in (1), not mentioning the other extra elementary trees that are necessary for verb-second position.

(1) a. *dass Peter ihn heute vergisst*
b. *dass ihn Peter heute vergisst*
c. *dass ihn heute Peter vergisst*
d. *dass heute ihn Peter vergisst*
e. . . .
('that Peter forgets him/it today')

While classic TAG seems to be appropriate for dealing with fixed word order languages and structural case (i.e., rudimentary case inflection), it is somehow missing the point when applied to free word order languages with rich case inflection.

This work addresses the modelling of complementation in German by means of TT-MCTAG, a recently developed derivative of Multi-Component TAG (MCTAG), that uses tree tuples as elementary structures. In contrast to classic TAG, we are able to abstract away from the relative order of co-complements in the final derived tree. Consequently, the TT-MCTAG account of complementation does not seem to be available for strict word order languages such as English, if complement-argument linking is performed on the basis of prederivational, lexical structures.

Therefore, a part of this survey will deal with the comparison with XTAG (XTAG Research Group, 2001), a rich TAG for English. Focussing on wh-extraction, we can observe a trade-off between the extent of word order flexibility and the size of the lexicon. Another comparison is dedicated to V-TAG (Rambow, 1994), which follows a strategy similar to TT-MCTAG, but chooses a different path to constrain locality. The effects of this choice can be clearly observed with bridging constructions.

We thus restrict ourselves to sentential complements, namely wh-extraction, that-complementation and bridging. The assigned analyses are parts of an extensive grammar for German, GerTT (German TT-MCTAG), that is currently being implemented using TT-MCTAG.[1] A parser is also available as part of the TuLiPA framework (Parmentier et al., 2008).[2]

[1] http://www.sfb441.uni-tuebingen.de/emmy-noether-kallmeyer/gertt/

[2] http://sourcesup.cru.fr/tulipa/

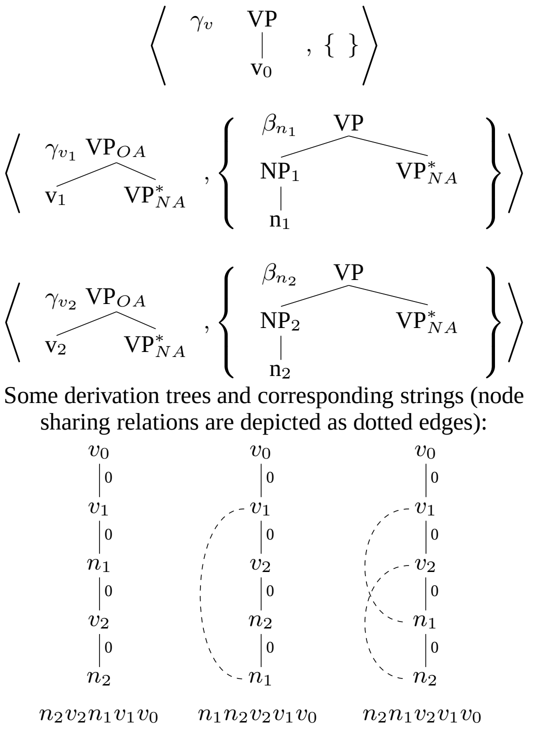

γ_v VP v_0 , { }
γ_{v_1} VP$_{OA}$ v_1 VP$^*_{NA}$, β_{n_1} VP NP$_1$ VP$^*_{NA}$ n_1
γ_{v_2} VP$_{OA}$ v_2 VP$^*_{NA}$, β_{n_2} VP NP$_2$ VP$^*_{NA}$ n_2
Some derivation trees and corresponding strings (node sharing relations are depicted as dotted edges):
v_0 0 v_1 0 n_1 0 v_2 0 n_2
v_0 0 v_1 0 v_2 0 n_2 0 n_1
v_0 0 v_1 0 v_2 0 n_1 0 n_2
$n_2v_2n_1v_1v_0$ $n_1n_2v_2v_1v_0$ $n_2n_1v_2v_1v_0$

Figure 1: Sample TT-MCTAG

2 k-TT-MCTAG

In TT-MCTAG, elementary structures are made of tuples of the form $\langle\gamma, \{\beta_1, ..., \beta_n\}\rangle$, where $\gamma, \beta_1, ..., \beta_n$ are elementary trees in terms of TAG (Joshi and Schabes, 1997). More precisely, γ is a lexicalized elementary tree while $\beta_1, ..., \beta_n$ are auxiliary trees. During derivation, the β-trees have to attach to the γ-tree, either directly or indirectly via node sharing (Kallmeyer, 2005). Roughly speaking, node sharing terms an extended locality, that allows β-trees to also adjoin at the roots of trees that either adjoin to γ themselves, or that are again in a node sharing relation to γ. In other words, an argument β must be linked by a chain of root adjunction to an elementary tree that adjoins to β's head γ.

As an example, consider the TT-MCTAG in Fig. 1. A derivation in this grammar necessarily starts with γ_v. We can adjoin arbitrarily many copies of γ_{v_1} or γ_{v_2}, always to the root of the already derived tree. Concerning the respective argument trees β_{n_1} and β_{n_2}, they must either adjoin immediately to the root of the correponding γ_{v_i} or their adjunction can be delayed. In this case they adjoin later to the root and we say that they stand in a node sharing relation to the corresponding γ_{v_i}. As a result we obtain all strings where an arbitrary sequence of v_i ($1 \leq i \leq 2$) precedes a v_0 and for each of the v_i ($1 \leq i \leq 2$), there is a unique correponding argument n_i in the string that precedes this v_i. In terms of dependencies, we obtain all permutations of the n_i, i.e., a language displaying everything from nested to cross-serial dependencies.

TT-MCTAG are further restricted, such that at each point of the derivation the number of pending β-trees is at most k. This subclass is also called k-TT-MCTAG. TT-MCTAG in general are NP-complete (Søgaard et al., 2007) while k-TT-MCTAG are mildly context-sensitive (Kallmeyer and Parmentier, 2008).

3 Principles of Complementation

3.1 Basic assumptions

The linguistic understanding of a tuple is that of a head (the γ-tree) and its subcategorization frame (the β-trees). More precisely, the β-trees contain a substitution node, where the complement is inserted. Another way to incorporate complements is to have a footnote in the head tree. This is exploited in, e.g., coherent constructions and bridging constructions. A TT-MCTAG account of scrambling and coherent constructions has been presented in Lichte (2007). Because of the nature of node sharing, subsitution establishes strong islands for movement, while adjunction widens the domain of locality.

In contrast to XTAG, we completely omit empty categories (e.g. traces, PRO) in syntactic description. This follows from rejecting a base word order for German, as well as dealing with argument raising and control only in the semantics.[3] As an example, consider the elementary tree tuples for (1) in Fig. 2 and the (TAG) derivation tree for (1)a.

In this derivation, none of the arguments adjoins immediately to their head *vergisst* but both stand in a node sharing relation to it.

Besides verb-final (V3) trees as in Fig. 2, there are also verb-second (V12) trees for finite verbs that contain two verbal positions: the left bracket (position of the verb) and the right bracket (sometimes containing, e.g., particles). See Fig. 3 for the *vergisst* tuple in verb-second sentences such as (2).

[3] This is linguistically supported, e.g., by Sag and Fodor (1994) and Culicover and Wilkins (1986).

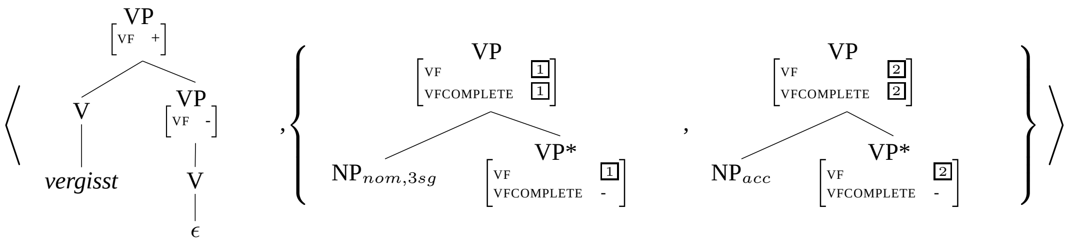


Figure 3: V12 tree tuple for *vergisst* as in (2)

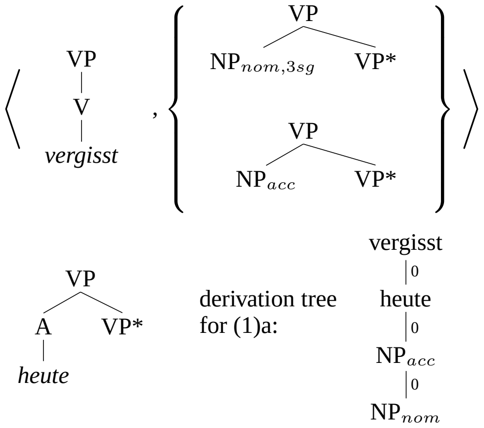


Figure 2: Tree tuples and derivation tree for (1)

(2) a. *Peter vergisst ihn*
Peter forgets him
b. *ihn vergisst Peter*
him forgets Peter

A feature VF for *vorfeld* indicates whether a VP node dominates the left bracket and therefore belongs to the vorfeld. If this is the case, then we must adjoin exactly one tree to this VP node since the vorfeld is always filled by exactly one constituent. This is guaranteed by the feature VFCOMPLETE that indicates whether the vorfeld is already filled. A vorfeld-adjoining argument tree switches this feature from $-$ to $+$.

3.2 Raising, auxiliaries and control

In our grammar, raising verbs and auxiliaries do not have a subject argument tree. Instead, the subject comes with the embedded infinitival. In this, we follow the choices of the XTAG grammar. Control verbs, however, have a subject. The argument identity between the controller and the subject of the embedded infinitive is established via a special feature that is then used within semantic computation.

Because of the difference between raising and control, we have to deal with verbs embedding an infinitive with subject (raising, auxiliaries, ECM verbs) and verbs embedding an infinitive without subject. This is more complicated than in XTAG since the presence of a subject cannot be seen from the verb tree, the subject argument tree being a separate auxiliary tree. Therefore we need a feature SUBJ that indicates whether a verb has a subject. Furthermore, the infinitive can have different forms, captured by the feature STAT for status: It can be a bare infinitive (STAT 1) an infinitive with *zu* (STAT 2) or a participle (STAT 3).

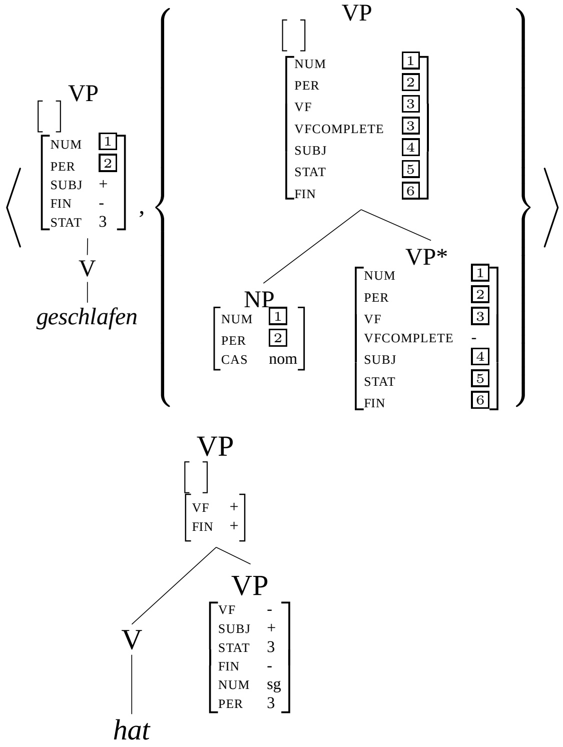

VP [ ] [NUM 1, PER 2, SUBJ +, FIN -, STAT 3]
V
geschlafen
VP [ ] [NUM 1, PER 2, VF 3, VFCOMPLETE 3, SUBJ 4, STAT 5, FIN 6]
NP [NUM 1, PER 2, CAS nom]
VP* [NUM 1, PER 2, VF 3, VFCOMPLETE -, SUBJ 4, STAT 5, FIN 6]
VP [ ] [VF +, FIN +]
V
hat
VP [VF -, SUBJ +, STAT 3, FIN -, NUM sg, PER 3]

Figure 4: Analysis of auxiliaries

(3) *Peter hat geschlafen*
Peter has slept

As an example, Fig. 4 shows the trees for *hat* and *geschlafen* in (3). In V2 auxiliary constructions such as (3), the left bracket is contributed by a separate auxiliary tree instead of being fixed within the tuple of the main verb. We must make sure that the auxiliary is recognized as left bracket and that there is exactly one element occupying the vorfeld, i.e., preceding the left bracket. This can be done by setting the feature VF $-$ at the foot node and $+$ at the root. The feature VFCOMPLETE on the argument trees works then exactly as in the case where the left bracket comes with the main verb.

A further issue to take into account is the agreement between subject and verb. Since we have a free word order language, we do not know where on the verbal spine the subject comes in. Therefore we need to percolate the subject agreement feature along the entire verbal spine to be unified with the auxiliary verb agreement features.

3.3 PP and sentential argument trees

Concerning the morphological form an argument can take (a NP, a PP or a sentential argument), we do not distinguish between these at the level of the category of the argument slot. Rather, their specific properties are treated within appropriate features (e.g., CASE). This can be achieved by assigning to the mopho-syntactic category (CAT) of argument slots either an underspecified value or a disjunction of category labels.[4] As a result, the same tree-family can be used for all verbs taking the same number of arguments. The selection of a preposition for one of the arguments is done via the case feature.

Furthermore, in our grammar, the family of a verb does not contain extra tree tuples for wh-extraction. Instead, the wh-element has a nominal category and can be substituted into a nominal argument tree. This accounts for the facts that wh-elements distribute similarly to non-wh NPs, see (4).

(4) a. *Peter hat wen heute gesehen*
Peter has whom_WH today seen
b. *wen hat Peter wann gesehen*
whom_WH has Peter when seen

The underspecification of argument categories and the fact that wh-extraction does not require special tree truples considerably decreases the number of verb families that are needed compared to grammars such as XTAG. From our experience with implementing the grammar we have the impression that this is an advantage for the grammar writer.

The choice to treat sentential and nominal arguments alike means in particular that sentential complements are added by substitution and therefore constitute islands for scrambling. However, an exceptional case are bridge verbs (see next section).

4 Sentential Complements

We present the analysis of sentential complements for German, that have a finite verb in clause-final position (V3). Nonfinite sentential complementation is ignored throughout the paper.

In German, V3 sentences serve as source for subordinate clauses, that are marked by certain elements in sentence-initial position, e.g., a wh-pronoun, a relative pronoun, or a complementizer. To model this fact, we introduce the feature S-TYPE, which indicates the sentence type via a complex value. Fig. 5 presents the schema of S-TYPE and its specification in the tree tuple of *vergisst*. Note that marking is enforced by the top-bottom mismatch of MARKED in the root node of the head tree.

4.1 Free relatives and embedded questions

Free relatives and embedded questions consist of V3 sentences that start with a relative pronoun or a wh-pronoun, respectively. Examples are given in (5).

(5) a. *den heute Peter bestohlen hat*
whom_REL today Peter stolen_from has
b. *wen heute Peter bestohlen hat*
whom_WH today Peter stolen_from has
('from whom Peter has stolen')

The corresponding constructions in English are commonly said to involve *wh-extraction*. Note that, in contrast to English, German lacks do-support and preposition stranding altogether. The analyses of free relatives and embedded questions in Fig. 6 only differ with respect to the terminal and the MARKING value in the elementary trees of

[4] Both strategies are supported by the metagrammar framework XMG (Duchier et al., 2004), but not yet by the TuLiPA parser (Parmentier et al., 2008). Therefore, in our current implementation of the grammar, only NP and PP arguments are treated uniformly, both of them having the category NP.

$$\begin{bmatrix} \textsc{s-type} & \begin{bmatrix} \textsc{config} & (\text{V12}|\text{V3}) \\ \textsc{marked} & (+|\text{-}) \\ \textsc{marking} & (\text{dass}|\text{ob}|\text{wh}|\text{rel}|\ldots) \\ \textsc{comp} & (+|\text{-}) \end{bmatrix} \end{bmatrix}$$

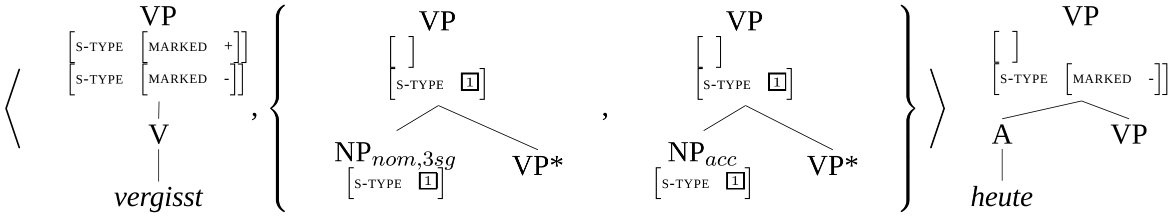


Figure 5: Use of the features STYPE and MARKED

the respective pronouns. Both substitute into a regular complement slot, and both have the MARKED feature set to $+$, which suffices to resolve the feature conflict in the VP projection.

4.2 Complementized sentences

Complementized sentences consist of V3 sentences that have a complementizer in initial position, e.g., *dass* (that), *ob* (whether), and *wenn* (if). An example is given in (6).

(6) a. *dass ihn heute Peter vergisst*
that him today Peter forgets
b. **dass dass ihn heute Peter vergisst*
c. **dass ihn vergisst heute Peter*

Two pitfalls have to be avoided: stacked complementizers as in (6)b, and V12 configurations as in (6)c. Considering Fig. 7, the first pitfall is avoided by using the feature COMP, that indicates whether complementation already took place. To account for the second one, the feature CONFIG specifies the topological configuration of the underlying sentence.

4.3 Bridge verbs

Bridge verbs allow for the extraction of constituents from the complementized sentential complement, see (7).

(7) a. *Wen glaubst du, dass Peter heute*
Whom think_2SG you, that Peter today
vergisst?
forgets
b. ?*Wer glaubst du, dass ihn heute*
Who think_2SG you, that him today
vergisst?
forgets
c. **Wen magst du, dass Peter heute*
Whom like_2SG you, that Peter today
vergisst?
forgets
d. **Du glaubst wen, dass Peter heute*
You think_2SG whom, that Peter today
vergisst?
forgets

In order to derive the example sentence in (7), the tree tuple from Fig. 8 has to be attached to some derived tree such as in Fig.7, but where the accusative object is still pending. Due to the adjunction of the bridge verb, the pending complement is able to adjoin at its root node via node sharing. The VF feature makes sure that only one pending complement can attach higher.

The long extraction of the subject in (7)b is claimed to be ungrammatical in English (*that-trace effect*). If this would also hold for German (which is rejected by several authors, see Featherston (2003)), we would have to introduce further features indicating the type of the complement. As it is now, the bridge verb is agnostic towards the material that is adjoined at its root. The contrast between bridge verbs and non-bridge verbs in (7)c could be explained by the absence of bridging tree tuples for non-bridge verbs. Long-distance extraction to a non-initial position, as in (7)d, is ruled out since the lower-right VP node is no root and therefore not shared.

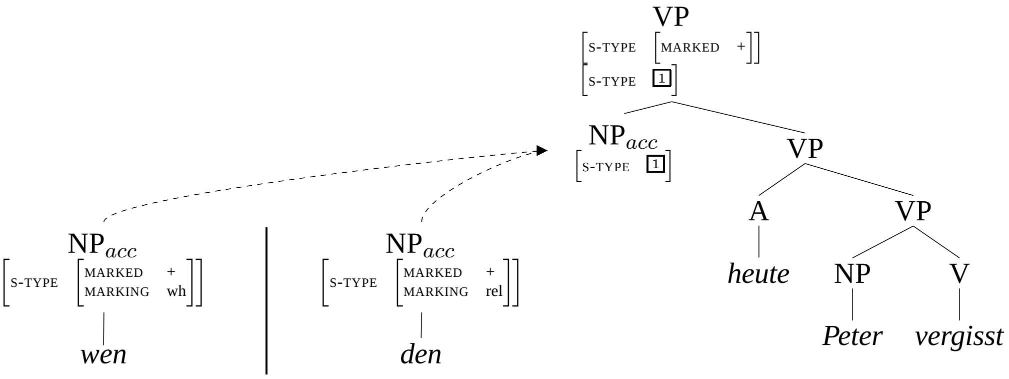


Figure 6: Wh-pronouns and relative pronouns

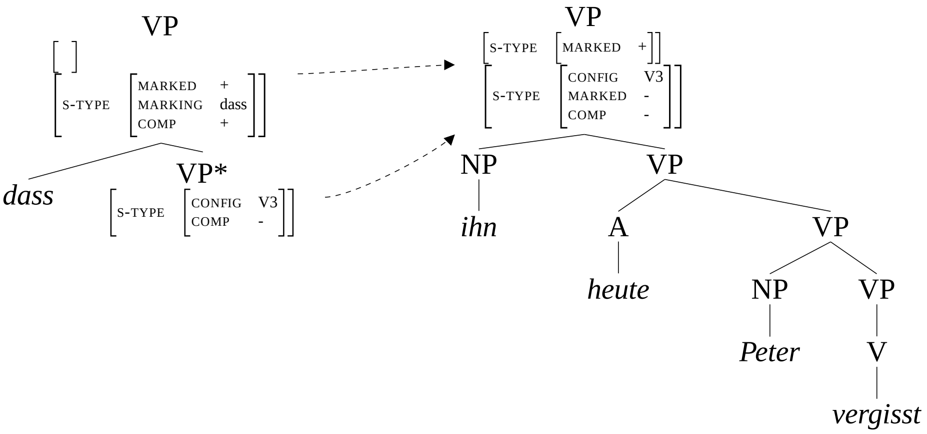


Figure 7: Complementizers

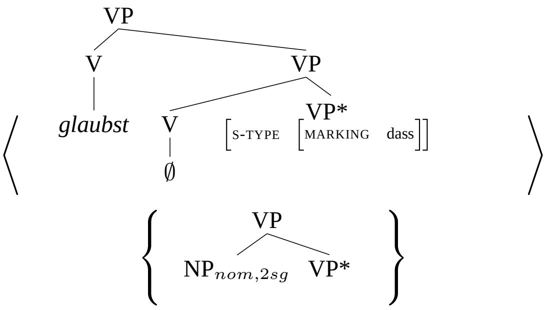


Figure 8: Bridge verbs

5 Extraction in XTAG

The principal discrepancy between XTAG and our grammar is the way of encoding the relative order between complements: using TAG, XTAG determines the relative order of complements in elementary trees. Consequently, deviations from the canonical order of complements have to be explicitly anticipated by providing extra trees in the grammar. This can be prominently observed with wh-extraction phenomena, where potentially every complement can be extracted. Thus, focussing merely on wh-extraction, a verb with n complements receives $n + 1$ elementary trees in XTAG, such as the one for object extraction in Fig. 9.[5] In our grammar based on TT-MCTAG, however, there is exactly one tree tuple for each verb and its subcategorization frame.

Nesson and Shieber (2007) consider a tree-local MCTAG account to reduce the set of extraction trees in XTAG by introducing tree sets that contain the extracted complement and its trace in separate trees. This, however, only moves the inherent ambiguity to the representation of the nouns, which does not seem to be more preferable to us.

6 Comparison to V-TAG

TT-MCTAG's nearest relative certainly is V-TAG from Rambow (1994), also designed for flexible word order phenomena in German. Superficially,

[5] We ignore preposition stranding here, since it does not exist in German. Furthermore, we deal with sentential complements in terms of direct objects.

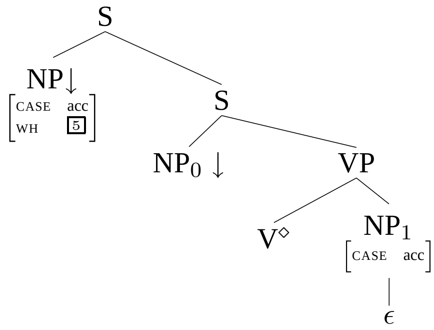
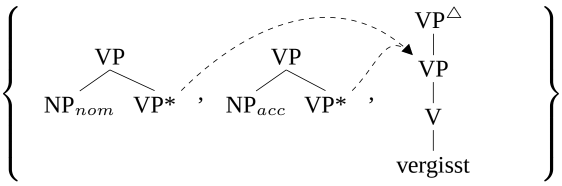


Figure 9: Feature reduced XTAG tree for object extraction (Fig. 15.1 therein)

their elementary structures look quite similar, as Fig. 10 shows. Technically, however, the limitation of non-locality is accomplished in different ways: where TT-MCTAG refers to the derivation tree using the notion of shared nodes, V-TAG makes use of *dominance links* and *integrity constraints* in the derived tree.



(dotted arrows = dominance links, △ = integrity constraint)

Figure 10: V-TAG tree set for *vergisst* ('forgets')

Most of the presented analyses for sentential complements can be easily mapped onto V-TAG variants, while preserving the idea of factorizing complementation. There is, however, one crucial exception: The analysis for bridging constructions cannot be borrowed directly, since, within V-TAG, it is not possible to express that the VP root node is accessible for a complement of the sentential complement, while the lower VP node is not accessible. Hence, in order to exclude (7)d while keeping an analysis that factors arguments into separate auxiliary trees, one needs different argument trees for complements that might be scrambled and complements that are extracted. The latter might, e.g., be forced to adjoin to a node with VF = +, as shown in Fig.11. However, the necessary removing of the integrity constraint on the VP root of the verbal tree would allow a movement of the non-extracted complement into the mittelfeld of the bridge verb as in (8). This is something that cannot happen with TT-MCTAG since the mittelfeld node of the bridge verb is not accessible via node sharing for arguments of the embedded verb.

(8)**Wen glaubst Peter du, dass heute*
Whom think_2SG Peter you, that today
vergisst?
forgets

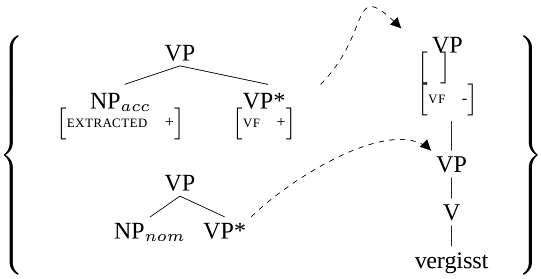


Figure 11: Possible V-TAG tree set for extraction with factored complements

Of course, in order to analyse bridge verbs with extraction in V-TAG, there is always the possibility to have the extracted argument and its verbal head in the same elementary tree; only the (possibly scrambled) other complements are in separate argument trees. Then the integrity constraint on the upper VP node can be maintained and examples (7) and (8) are analysed correctly.

In general one can say that formalisms such as V-TAG (and also DSG (Rambow et al., 2001)) have to model locality constraints explicitely since the derivation itself in these formalisms is not constrained by any locality requirement. As a result, an analysis that factors complementation the way we propose it within TT-MCTAG seems less easily available. Furthermore, the fact that locality constraints follow from the TT-MCTAG formalism and need not be explicitely stipulated is in our view an advantage of this formalism.[6]

7 Discussion

As already mentioned, a key idea of our grammar is the factorization of argument slots in separate auxiliary trees. As a result, we need considerably less elementary tree sets per family than standard TAG. Furthermore, since we treat prepositional, sentential and nomial arguments alike, the number of tree set families reduces as well. From our current experience with the development of the grammar, we feel that this is an advantage for grammar

[6] We might of course encounter cases where the TT-MCTAG locality is too restrictive.

implementation. Concerning parsing, we have to take into account all possible combinations of the trees in our tuples. In this respect, the factorization of course only shifts the task of building constituent structures for subcategorization frames to a different part of the system.

k-TT-MCTAG is mildly context-sensitive and, furthermore, we suspect that it is a proper subclass of set-local MCTAG. Recently, Chen-Main and Joshi (2007) discussed the fact that in actual analyses, only a very small part of the possibilities provided by multicomponent TAG extensions (e.g., tree-local and set-local MCTAG) is used. Consequently, the proposed MCTAGs don't corre-pond to the actual need for linguistic descriptions. We hope that k-TT-MCTAG with its rather strong locality might be a further step towards the identification of the class of grammar formalisms suitable for natural language processing.

Acknowledgments

The work presented here was funded by the Emmy-Noether program of the DFG.

We would like to thank three anonymous reviewers whose comments helped to considerably improve the paper. Furthermore, we are indebted to the TuLiPA-team, in particular Yannick Parmentier, Wolfgang Maier and Johannes Dellert. Without the TuLiPA parser, the development of our German TT-MCTAG would not have been possible.

References

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

Chen-Main, Joan and Aravind Joshi. 2007. Multicomponent Tree Adjoining Grammars, Dependency Graph Models, and Linguistic Analyses. In *ACL 2007 Workshop on Deep Linguistic Processing*, pages 1–8, Prague, Czech Republic, June. Association for Computational Linguistics.

Culicover, Peter W. and Wendy Wilkins. 1986. Control, PRO, and the projection principle. *Language*, 62(1):120–153.

Duchier, Denys, Joseph Le Roux, and Yannick Parmentier. 2004. The Metagrammar Compiler: An NLP Application with a Multi-paradigm Architecture. In *Second International Mozart/Oz Conference (MOZ'2004)*.

Featherston, Sam. 2003. That-trace in German. *Lingua*, 1091:1–26.

Joshi, Aravind K. and Yves Schabes. 1997. Tree-adjoining grammars. In Rozenberg, G. and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 69–124. Springer, Berlin, New York.

Kallmeyer, Laura and Yannick Parmentier. 2008. On the relation between Multicomponent Tree Adjoining Grammars with Tree Tuples (TT-MCTAG) and Range Concatenation Grammars (RCG). In *Proceedings of the 2nd International Conference on Language and Automata Theory and Applications LATA*, Tarragona, Spain, March.

Kallmeyer, Laura. 2005. Tree-local multicomponent tree adjoining grammars with shared nodes. *Computational Linguistics*, 31:2:187–225.

Lichte, Timm. 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.

Nesson, Rebecca and Stuart Shieber. 2007. Extraction phenomena in synchronous TAG syntax and semantics. In Wu, Dekai and David Chiang, editors, *Proceedings of the Workshop on Syntax and Structure in Statistical Translation*. Rochester, New York, 26 April 2007.

Parmentier, Yannick, Laura Kallmeyer, Wolfgang Maier, Timm Lichte, and Johannes Dellert. 2008. TuLiPA: A syntax-semantics parsing environment for mildly context-sensitive formalisms. In *Proceedings of TAG+9*.

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

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

Sag, Ivan A. and Janet Dean Fodor. 1994. Extraction without traces. In *Proceedings of the Thirteenth Annual Meeting of the West Coast Conference on Formal Linguistics*, pages 365–384, Stanford. CSLI Publications.

Søgaard, Anders, Timm Lichte, and Wolfgang Maier. 2007. The complexity of linguistically motivated extensions of tree-adjoining grammar. In *Recent Advances in Natural Language Processing 2007*, Borovets, Bulgaria.

XTAG Research Group. 2001. A Lexicalized Tree Adjoining Grammar for English. Technical report, Institute for Research in Cognitive Science, University of Pennsylvania, Philadelphia, PA.