12 Tree Adjoining Grammar

Tree Adjoining Grammar (TAG) was developed by Aravind Joshi at the University of Pennsylvania in the USA (Joshi, Levy & Takahashi 1975). Several important dissertations in TAG have been supervised by Aravind Joshi and Anthony Kroch at the University of Pennsylvania (e. g. Rambow (1994)). Other research centers with a focus on TAG are Paris 7 (Anne Abeillé), Columbia University in the USA (Owen Rambow) and Düsseldorf, Germany (Laura Kallmeyer).

Rambow (1994) and Gerdes (2002b) are more detailed studies of German.[1]

TAG and its variants with relevant extensions are of interest because it is assumed that this grammatical formalism can – with regard to its expressive power – relatively accurately represent what humans do when they produce or comprehend natural language. The expressive power of Generalized Phrase Structure Grammar was deliberately constrained so that it corresponds to context-free phrase structure grammars (Type-2 languages) and it has in fact been demonstrated that this is not enough (Shieber 1985; Culy 1985).[2] Grammatical theories such as HPSG and CxG can generate/describe so-called type-0 languages and are thereby far above the level of complexity presently assumed for natural languages. The assumption is that this complexity lies somewhere between context-free and context-sensitive (Type 1) languages. This class is thus referred to as *mildly context sensitive*. Certain TAG-variants are inside of this language class and it is assumed that they can produce exactly those structures that occur in natural languages. For more on complexity, see Section 12.6.3 and Chapter 17.

There are various systems for the processing of TAG grammars (Doran, Hockey, Sarkar, Srinivas & Xia 2000; Parmentier, Kallmeyer, Maier, Lichte & Dellert 2008; Kallmeyer, Lichte, Maier, Parmentier, Dellert & Evang 2008). Smaller and larger TAG fragments have been developed for the following languages:

- Arabic (Fraj, Zribi & Ahmed 2008),
- German (Rambow 1994; Gerdes 2002a; Kallmeyer & Yoon 2004; Lichte 2007),
- English (XTAG Research Group 2001; Frank 2002; Kroch & Joshi 1987),
- French (Abeillé 1988; Candito 1996, 1998, 1999; Crabbé 2005),
- Italian (Candito 1998, 1999),

[1] Since my knowledge of French leaves something to be desired, I just refer to the literature in French here without being able to comment on the content.

[2] See Pullum (1986) for a historical overview of the complexity debate and G. Müller (2011a) for argumentation for the non-context-free nature of German, which follows parallel to Culy with regard to the N-P-N construction (see Section 21.10.4).

- Korean (Han, Yoon, Kim & Palmer 2000; Kallmeyer & Yoon 2004),
- Vietnamese (Le, Nguyen & Roussanaly 2008)

Candito (1996) has developed a system for the representation of meta grammars which allows the uniform specification of crosslinguistic generalizations. This system was used by some of the projects mentioned above for the derivation of grammars for specific languages. For instance Kinyon, Rambow, Scheffler, Yoon & Joshi (2006) derive the verb second languages from a common meta grammar. Among those grammars for verb second languages is a grammar of Yiddish for which there was no TAG grammar until 2006.

Resnik (1992) combines TAG with a statistics component.

12.1 General remarks on representational format

12.1.1 Representation of valence information

Figure 12.1 shows so-called elementary trees. These are present in the lexicon and can be combined to create larger trees: Nodes for the insertion of arguments are specially

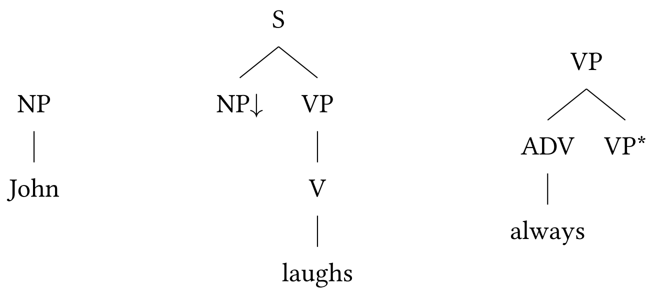

Figure 12.1: Elementary trees

marked (NP↓ in the tree for *laughs*). Nodes for the insertion of adjuncts into a tree are also marked (VP* in the tree for *always*). Grammars where elementary trees always contain at least one word are referred to as *Lexicalized Tree Adjoining Grammar* (LTAG).

12.1.2 Substitution

Figure 12.2 on the facing page shows the substitution of nodes. Other subtrees have to be inserted into substitution nodes such as the NP node in the tree for *laughs*. The tree for *John* is inserted there in the example derivation.

12.1.3 Adjunction

Figure 12.3 on the next page shows an example of how the adjunction tree for *always* can be used.

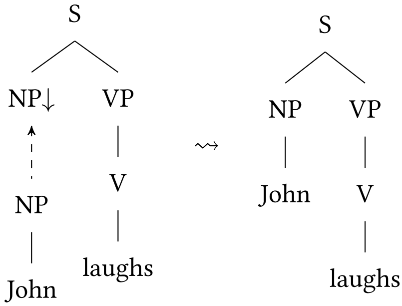

Figure 12.2: Substitution

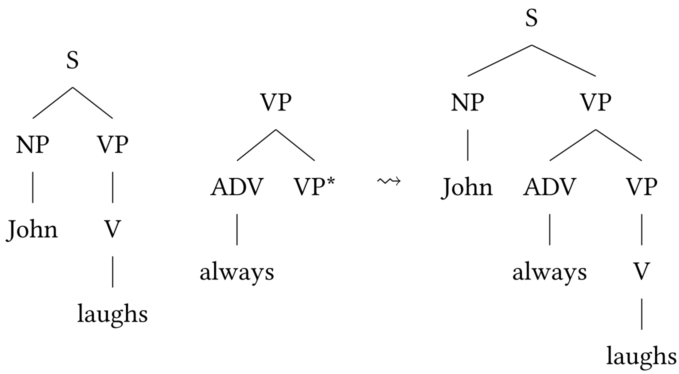

Figure 12.3: Adjunction

Adjunction trees can be inserted into other trees. Upon insertion, a node in a tree corresponding to the node marked by '*' in the adjunction tree is split up and the adjunction tree is entered in its place.

TAG differs considerably from the simple phrase structure grammars we encountered in Chapter 2 in that the trees extend over a larger domain: for example, there is an NP node in the tree for *laughs* that is not a sister of the verb. In a phrase structure grammar (and of course in GB and GPSG since these theories are more or less directly built on phrase structure grammars), it is only ever possible to describe subtrees with a depth of one level. For the tree for *laughs*, the relevant rules would be those in (1):

(1) S $\rightarrow$ NP VP
VP $\rightarrow$ V
V $\rightarrow$ laughs

In this context, it is common to speak of *locality domains*. The extension of the locality domain is of particular importance for the analysis of idioms (see Section 18.2).

TAG differs from other grammatical theories in that it is possible for structures to be broken up again. In this way, it is possible to use adjunction to insert any amount of material into a given tree and thereby cause originally adjacent constituents to end up being arbitrarily far away from each other in the final tree. As we will see in Section 12.5, this property is important for the analysis of long-distance dependencies without movement.

12.1.4 Semantics

There are different approaches to the syntax-semantics interface in TAG. One possibility is to assign a semantic representation to every node in the tree. The alternative is to assign each elementary tree exactly one semantic representation. The semantics construction does not make reference to syntactic structure, but rather the way the structure is combined. This kind of approach has been proposed by Candito & Kahane (1998) and then by Kallmeyer & Joshi (2003), who build on it. The basic mechanisms will be briefly presented in what follows.

In the literature on TAG, a distinction is made between derived trees and derivation trees. Derived trees correspond to constituent structure (the trees for *John laughs* and *John always laughs* in Figures 12.2 and 12.3). The derivation tree contains the derivational history, that is, information about how the elementary trees were combined. The elements in a derivation tree represent predicate-argument dependencies, which is why it is possible to derive a semantic derivation tree from them. This will be shown on the basis of the sentence in (2):

(2) Max likes Anouk.

The elementary tree for (2) and the derived tree are given in Figure 12.4. The nodes in

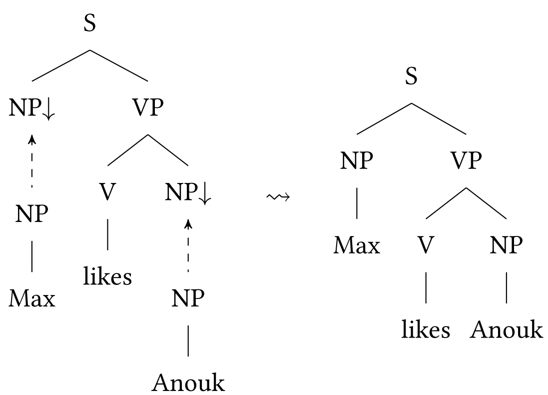

Figure 12.4: Elementary trees and derived tree for *Max likes Anouk.*

trees are numbered from top to bottom and from left to right. The result of this numbering of nodes for *likes* is shown in Figure 12.5 on the next page. The top-most node in the tree for *likes* is S and has the position 0. Beneath S, there is an NP and a VP node. These nodes are again numbered starting at 0. NP has the position 0 and VP the position 1. The VP node has in turn two daughters: V and the object NP. V receives number 0 and the object NP 1. This makes it possible to combine these numbers and then it is possible to unambiguously access individual elements in the tree. The position for the subject NP is 00 since this is a daughter of S and occurs in first position. The object NP has the numeric sequence 011 since it is below S (0), in the VP (the second daughter of S = 1) and occurs in second position (the second daughter of VP = 1).

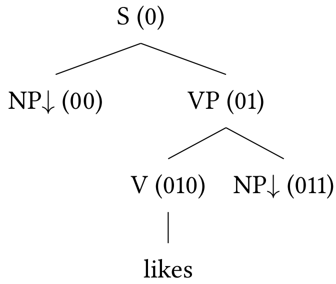

Figure 12.5: Node positions in the elementary tree for *likes*

With these tree positions, the derivation tree for (2) can be represented as in Figure 12.6. The derivation tree expresses the fact that the elementary tree for *likes* was

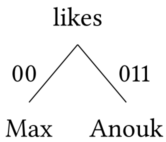

Figure 12.6: Derivation tree for *Max likes Anouk.*

combined with two arguments that were inserted into the substitution positions 00 and 011. The derivation tree also contains information about what exactly was placed into these nodes.

Kallmeyer & Joshi (2003) use a variant of *Minimal Recursion Semantics* as their semantic representational formalism (Copestake, Flickinger, Pollard & Sag 2005). I will use a considerably simplified representation here, as I did in the HPSG chapter. For the elementary trees *Max*, *likes* and *Anouk*, we can assume the semantic representations in (3).

(3) Semantic representations for elementary trees:

max(x)
arg: –

like(x_1, x_2)
arg: ⟨ x_1, 00 ⟩, ⟨ x_2, 011 ⟩

anouk(y)
arg: –

In a substitution operation, a variable is assigned a value. If, for example, the elementary tree for *Max* is inserted into the subject position of the tree for *likes*, then x_1 is identified with x. In the same way, x_2 is identified with y if the tree for *Anouk* is inserted into the object position. The result of these combinations is the representation in (4):

(4) Combination of the meaning of elementary trees:

like(x, y) max(x) anouk(y)
arg: –

Kallmeyer & Joshi (2003) show how an extension of TAG, Multi-Component LTAG, can handle quantifier scope and discuss complex cases with embedded verbs. Interested readers are referred to the original article.

12.2 Local reordering

In TAG, there is a family of trees for each word. In order to account for ordering variants, one can assume that there are six trees corresponding to a ditransitive verb and that each of these corresponds to a different ordering of the arguments. Trees are connected to one another via lexical rules. This lexical rule-based analysis is parallel to the one developed by Uszkoreit (1986b) in Categorial Grammar.

Alternatively, one could assume a format for TAG structures similar to what we referred to as the ID/LP format in the chapter on GPSG. Joshi (1987b) defines an elementary structure as a pair that consists of a dominance structure and linearization constraints. Unlike GPSG, linearization rules do not hold for all dominance rules but rather for a particular dominance structure. This is parallel to what we saw in Section 10.6.3 on Embodied-CxG. Figure 12.7 shows a dominance tree with numbered nodes. If we com-

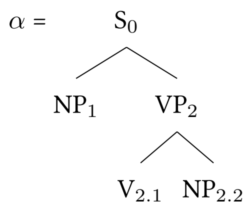

Figure 12.7: Dominance structure with numbered nodes

bine this dominance structure with the linearization rules in (5), we arrive at the exact order that we would get with ordinary phrase structure rules, namely NP_1 V NP_2.

(5) $LP^{\alpha}_{1} = \{ 1 < 2, 2.1 < 2.2 \}$

If one specifies the linearization restrictions as in (6), all the orders in (7) are permitted, since the empty set means that we do not state any restrictions at all.

(6) $LP^{\alpha}_{2} = \{ \}$

(7)
a. NP_1 V NP_2
b. NP_2 V NP_1
c. NP_1 NP_2 V
d. NP_2 NP_1 V
e. V NP_1 NP_2
f. V NP_2 NP_1

This means that it is possible to derive all orders that were derived in GPSG with flat sentence rules despite the fact that there is a constituent in the tree that consists of NP and VP. Since the dominance rules include a larger locality domain, such grammars are called LD/LP grammars (local dominance/linear precedence) rather than ID/LP grammars (immediate dominance/linear precedence) (Joshi, Shanker & Weir 1990).

Simple variants of TAG such as those presented in Section 12.1 cannot deal with reordering if the arguments of different verbs are scrambled as in (8).

(8) weil ihm das Buch jemand zu lesen versprochen hat[3]
because him.DAT the.ACC book somebody.NOM to read promised has
'because somebody promised him to read the book'

In (8), *das Buch* 'the book' is the object of *zu lesen* 'to read', and *ihm* 'him' and *jemand* 'somebody' are dependent on *versprochen* bzw. *hat*, respectively. These cases can be analyzed by LD/LP-TAG developed by Joshi (1987b) and Free Order TAG (FO-TAG) (Becker, Joshi & Rambow 1991: 21) since both of these TAG variants allow for crossing edges.

Since certain restrictions cannot be expressed in FO-TAG (Rambow 1994: 48–50), so-called Multi-Component TAG was developed. Joshi, Becker & Rambow (2000) illustrate the problem that simple LTAG grammars have with sentences such as (8) using examples such as (9):[4]

(9) a. … daß der Detektiv dem Klienten [den Verdächtigen des
that the.NOM detective the.DAT client the.ACC suspect theGEN
Verbrechens zu überführen] versprach
crime to indict promised
'that the detective promised the client to indict the suspect of the crime'

[3] For more on these kind of examples, see Bech (1955).

[4] The authors use *versprochen hat* 'has promised' rather than *versprach* 'promised', which sounds better but does not correspond to the trees they use.

b. … daß des Verbrechens$_k$ der Detektiv den Verdächtigen$_j$ dem
that the.GEN crime the.NOM detective the.ACC suspect the.DAT
Klienten [$\_{j}$ $\_{k}$ zu überführen] versprach
client to indict promised

In LTAG, the elementary trees for the relevant verbs look as shown in Figure 12.8. The

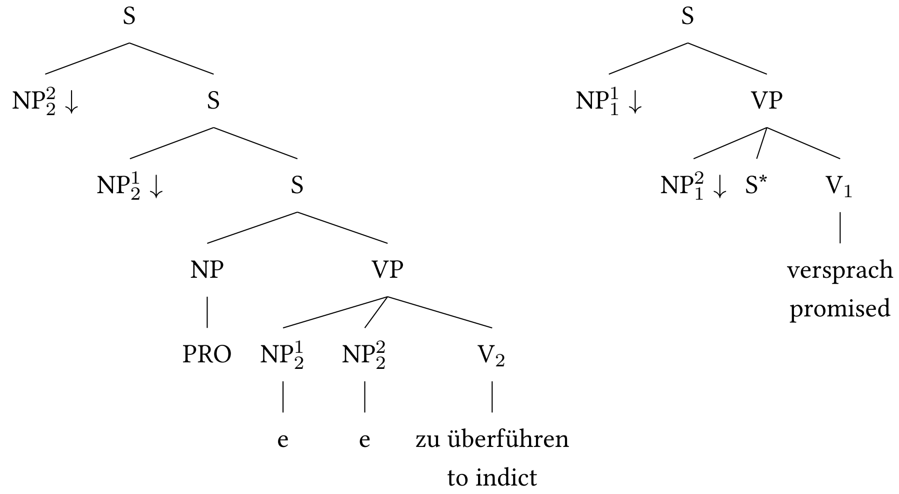

Figure 12.8: Elementary trees of an infinitive and a control verb

verbs are numbered according to their level of embedding. The NP arguments of a verb bear the same index as that verb and each has a superscript number that distinguishes it from the other arguments. The trees are very similar to those in GB. In particular, it is assumed that the subject occurs outside the VP. For non-finite verbs, it is assumed that the subject is realized by PRO. PRO is, like *e*, a phonologically empty pronominal category that also comes from GB. The left tree in Figure 12.8 contains traces in the normal position of the arguments and the relevant NP slots in higher trees positions. An interesting difference to other theories is that these traces only exist in the tree. They are not represented as individual entries in the lexicon as the lexicon only contains words and the corresponding trees.

The tree for *versprach* 'promised' can be inserted into any S node in the tree for *zu überführen* 'to indict' and results in trees such as those in the Figures 12.9 and 12.10. In Figure 12.9, the tree for *versprach* is inserted directly above the PRO NP and in Figure 12.10 above NP_2^1.

It is clear that it is not possible to derive a tree in this way where an argument of *überführen* 'to indict' occurs between the arguments of *versprach* 'promised'. Joshi, Becker & Rambow (2000) therefore suggest an extension of the LTAG formalism. In MC-TAG, the grammar does not consist of elementary trees but rather finite sets of elementary

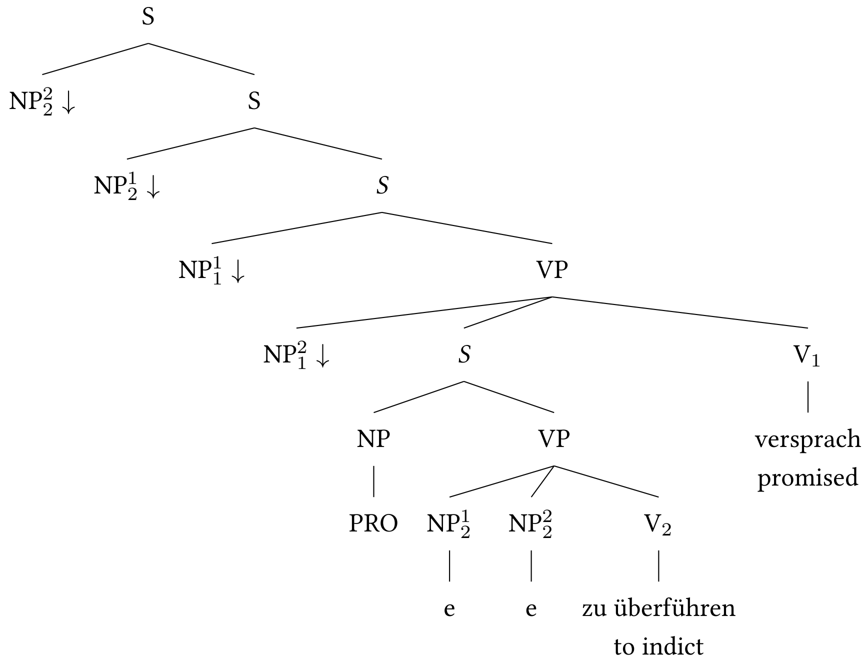

Figure 12.9: Analysis of the order NP^2_2 NP^1_2 NP^1_1 NP^2_1 V_2V_1: adjunction to the lowest S node

trees. In every derivational step, a set is selected and the elements of that set are simultaneously added to the tree. Figure 12.11 on the next page shows an elementary tree for *versprach* 'promised' consisting of multiple components. This tree contains a trace of NP^1_1 that was moved to the left. The bottom-left S node and the top-right S node are connected by a dashed line that indicates the dominance relation. However, immediate dominance is not required. Therefore, it is possible to insert the two subtrees into another tree separately from each other and thereby analyze the order in Figure 12.12 on page 421, for example.

Other variants of TAG that allow for other constituent orders are V-TAG (Rambow 1994) and TT-MC-TAG (Lichte 2007).

12.3 Verb position

The position of the verb can be analyzed in a parallel way to the GPSG analysis: the verb can be realized in initial or in final position in a given linearization domain. Since the verb position has an effect on the clause type and hence on semantics, a lexical rule-based analysis would be also viable: a tree with the finite verb in initial position is licensed by

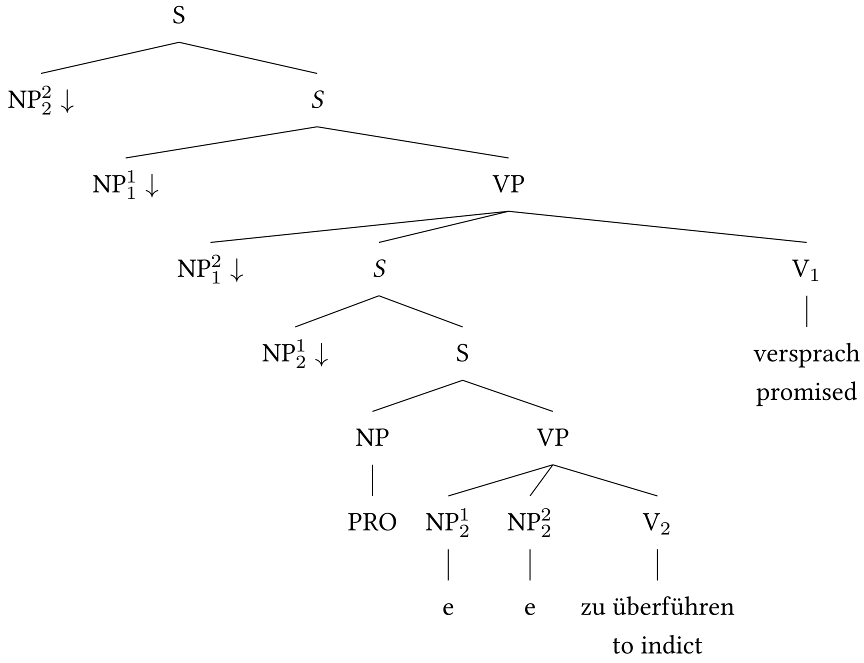

Figure 12.10: Analysis of the order NP^2_2 NP^1_1 NP^2_1 NP^1_2 V_2V_1: adjunction to the S node between NP^2_2 and NP^1_2

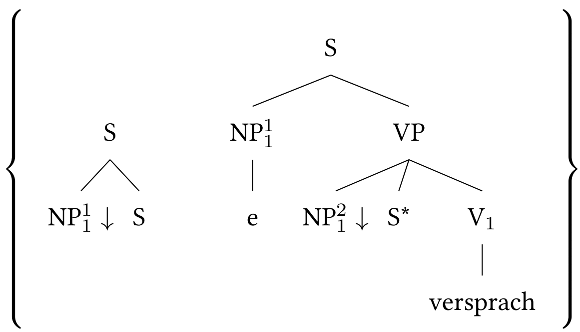

Figure 12.11: Elementary tree for *versprach* consisting of multiple components

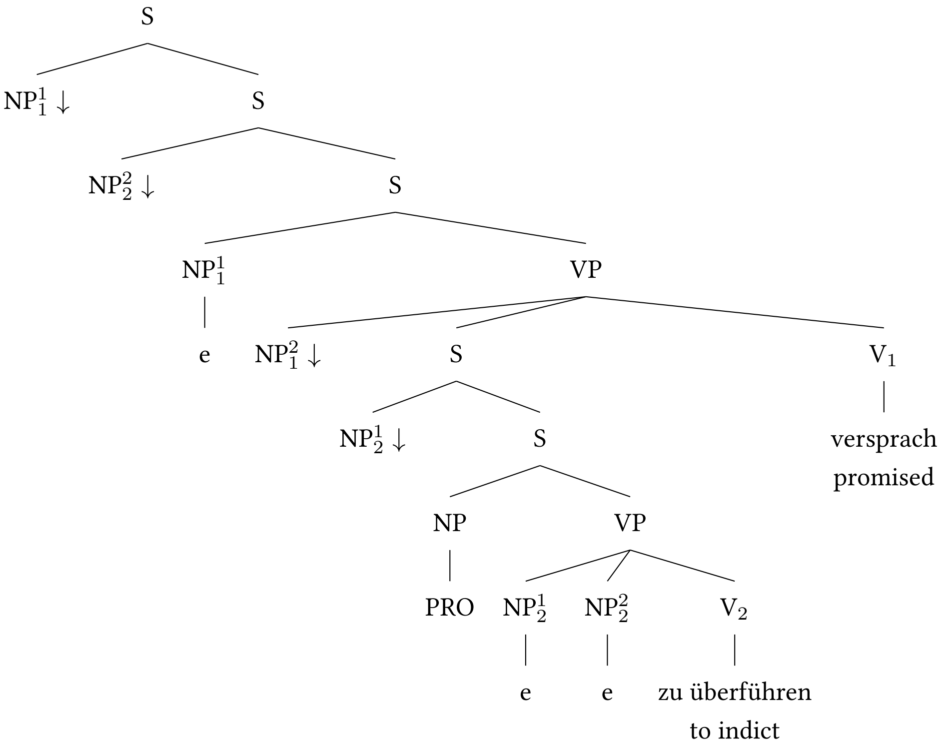

Figure 12.12: Analysis of the order NP^1_1 NP^2_2 NP^2_1 NP^1_2 V_2V_1: adjunction to the S node between NP^2_2 and NP^1_2

a lexical rule that takes a tree with the verb in final position as input. This would be similar to the analyses in GB, Minimalism, and HPSG.

12.4 Passive

There is a possible analysis for the passive that is analogous to the transformations in Transformational Grammar: one assumes lexical rules that create a lexical item with a passive tree for every lexical item with an active tree (Kroch & Joshi 1985: 50–51).

Kroch & Joshi (1985: 55) propose an alternative to this transformation-like approach that more adequately handles so-called raising constructions. Their analysis assumes that arguments of verbs are represented in subcategorization lists. Verbs are entered into trees that match their subcategorization list. Kroch and Joshi formulate a lexical rule that corresponds to the HPSG lexical rule that was discussed on page 281, that is, an accusative object is explicitly mentioned in the input of the lexical rule. Kroch and Joshi then suggest a complex analysis of the impersonal passive which uses a semantic null role for a non-realized object of intransitive verbs (p. 56). Such an analysis with abstract auxiliary entities can be avoided easily: one can instead use the HPSG analysis going back to Haider (1986a), which was presented in Section 9.2.

There are also proposals in TAG that use inheritance to deal with valence changing processes in general and the passive in particular (Candito (1996) and Kinyon, Rambow, Scheffler, Yoon & Joshi (2006) following Candito). As we saw in Section 10.2 of the Chapter on Construction Grammar, inheritance is not a suitable descriptive tool for valence changing processes. This is because these kinds of processes interact syntactically and semantically in a number of ways and can also be applied multiple times (Müller 2006, 2007c; 2007b: Section 7.5.2; 2013c; 2014a). See also Section 21.4 of this book.

12.5 Long-distance dependencies

The analysis of long-distance dependencies in TAG is handled with the standard apparatus: simple trees are inserted into the middle of other trees. Figure 12.13 on the next page shows an example of the analysis of (10):

(10) Who$_i$ did John tell Sam that Bill likes $\__i$?

The tree for *WH COMP NP likes* $\__i$ belongs to the tree family of *likes* and is therefore present in the lexicon. The tree for *tell* is adjoined to this tree, that is, this tree is inserted in the middle of the tree for *who that Bill likes* $\__i$. Such an insertion operation can be applied multiple times so that sentences such as (11) where *who* is moved across multiple sentence boundaries can be analyzed:

(11) Who$_i$ did John tell Sam that Mary said that Bill likes $\__i$?

There is another important detail: although the tree for (12) has the category S, (12) is not a grammatical sentence of English.

(12) * who that Bill likes

This has to be captured somehow. In TAG, the marking OA ensures that a tree counts as incomplete. If a tree contains a node with marking OA, then an obligatory adjunction operation must take place at the relevant position.

12.6 New developments and theoretical variants

In Section 12.2, we introduced Multi-Component-TAG. There are a large number of TAG variants with different formal properties. Rambow (1994) gives an overview of the variants existing in 1994. In the following, we will discuss two interesting variants of TAG: Feature Structure-Based TAG = FTAG (Vijay-Shanker & Joshi 1988) and Vector-TAG (V-TAG, Rambow (1994)).

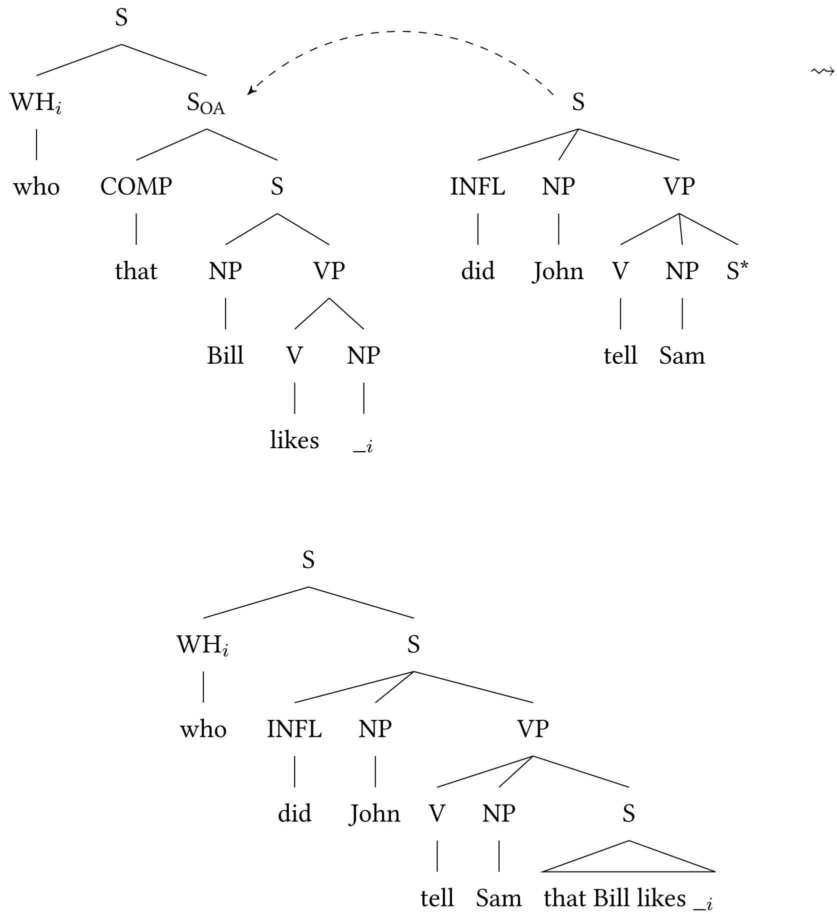

Figure 12.13: Analysis of long-distance dependencies in TAG

12.6.1 FTAG

In FTAG, nodes are not atomic (N, NP, VP or S), but instead consist of feature descriptions. With the exception of substitution nodes, each node has a top structure and a bottom structure. The top structure says something about what kind of properties a given tree has inside a larger structure, and the bottom structure says something about the properties of the structure below the node. Substitution nodes only have a top structure. Figure 12.14 on the following page shows an example tree for *laughs*. A noun phrase can be combined with the tree for *laughs* in Figure 12.14. Its top structure is identified with the NP node in the tree for *laughs*. The result of this combination is shown in Figure 12.15 on page 425.

In a complete tree, all top structures are identified with the corresponding bottom

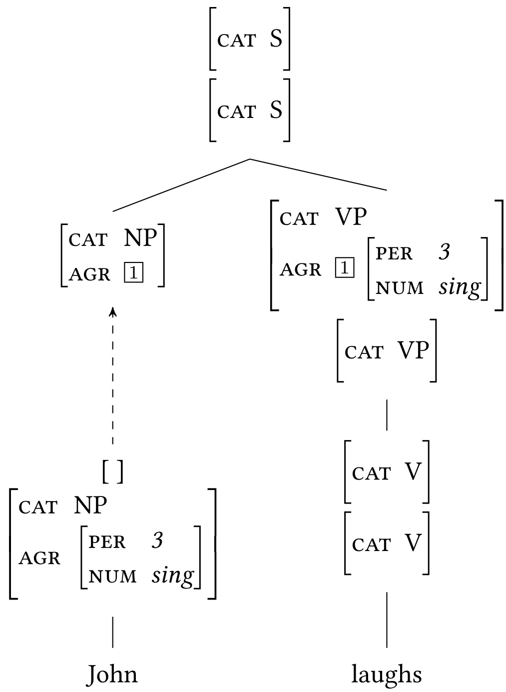

Figure 12.14: Elementary trees for *John* and *laughs* in FTAG

structures. This way, only sentences where the subject is in third person singular can be analyzed with the given tree for *laughs*, that is, those in which the verb's agreement features match those of the subject.

For adjunction, the top structure of the tree that is being inserted must be unifiable with the top structure of the adjunction site, and the bottom structure of the node marked '*' in the inserted tree (the so-called foot node) must be unifiable with the adjunction site.

The elementary trees discussed so far only consisted of nodes where the top part matched the bottom part. FTAG allows for an interesting variant of specifying nodes that makes adjunction obligatory in order for the entire derivation to be well-formed. Figure 12.16 on page 426 shows a tree for *laughing* that contains two VP nodes with incompatible MODE values. In order for this subtree to be used in a complete structure, another tree has to be added so that the two parts of the VP node are separated. This happens by means of an auxiliary tree as shown in Figure 12.16. The highest VP node of the auxiliary tree is unified with the upper VP node of *laughing*. The node of the auxiliary tree marked with '*' is unified with the lower VP node of *laughing*. The result of this is given in Figure 12.17 on page 427.

If a tree is used as a final derivation, the top structures are identified with the bottom structures. Thus, the AGR value of the highest VP node is identified with that of the lower

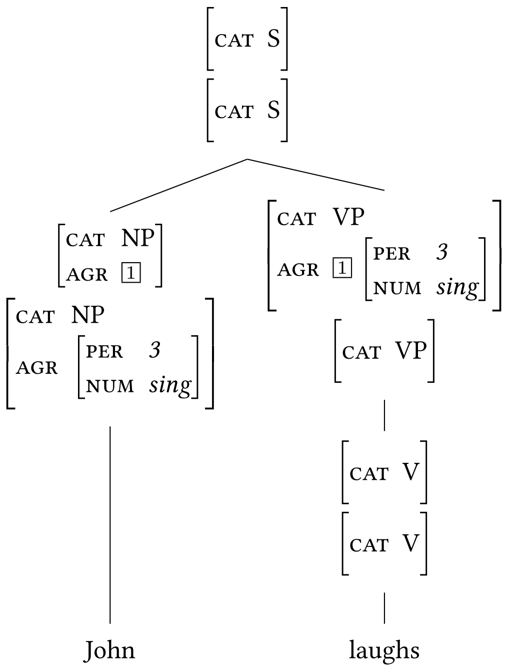

Figure 12.15: Combination of the trees for *John* and *laughs* in FTAG

one in the tree in Figure 12.17. As such, only NPs that have the same AGR value as the auxiliary can be inserted into the NP slot.

This example shows that, instead of the marking for obligatory adjunction that we saw in the section on long-distance dependencies, the same effect can be achieved by using incompatible feature specifications on the top and bottom structures. If there are incompatible top and bottom structures in a tree, then it cannot be a final derivation tree and therefore this means that at least one adjunction operation must still take place in order to yield a well-formed tree.

12.6.2 V-TAG

V-TAG is a variant of TAG proposed by Owen Rambow (1994) that also assumes feature structures on nodes. In addition, like MC-TAG, it assumes that elementary trees consist of multiple components. Figure 12.18 on page 427 shows the elementary lexical set for the ditransitive verb *geben* 'give'. The lexicon set consists of a tree for the verb, an empty element of the category VP and three trees where a VP has been combined with an NP. As in MC-TAG, dominance relations are also indicated. The dominance constraints in Figure 12.18 ensure that all lower VP nodes dominate the highest VP node of the tree further to the right. The order of the arguments of the verb as well as the position of the

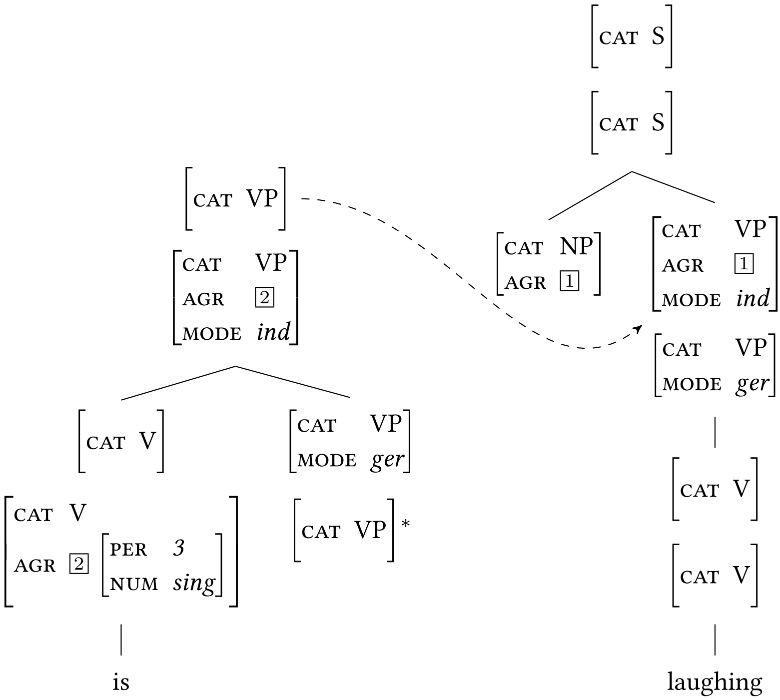

Figure 12.16: Obligatory adjunction in FTAG

verb is not given. The only thing required is that lower VP in the NP trees and lower VP in the *geben* tree dominate the empty VP node. With this lexicon set, it is possible to derive all permutations of the arguments. Rambow also shows how such lexical entries can be used to analyze sentences with verbal complexes. Figure 12.19 on page 428 shows a verbal complex formed from *zu reparieren* 'to repair' and *versprochen* 'promised' and the relevant dominance constraints. Both of the first NP trees have to dominate *versprochen* and the third and fourth NP tree have to dominate *zu reparieren*. The order of the NP trees is not restricted and thus all permutations of NPs can be derived.

The interesting thing here is that this approach is similar to the one proposed by Berman (1996: Section 2.1.3) in LFG (see Section 7.4): in Berman's analysis, the verb projects directly to form a VP and the arguments are then adjoined.

A difference to other analyses discussed in this book is that there is always an empty element in the derived trees regardless of verb position.

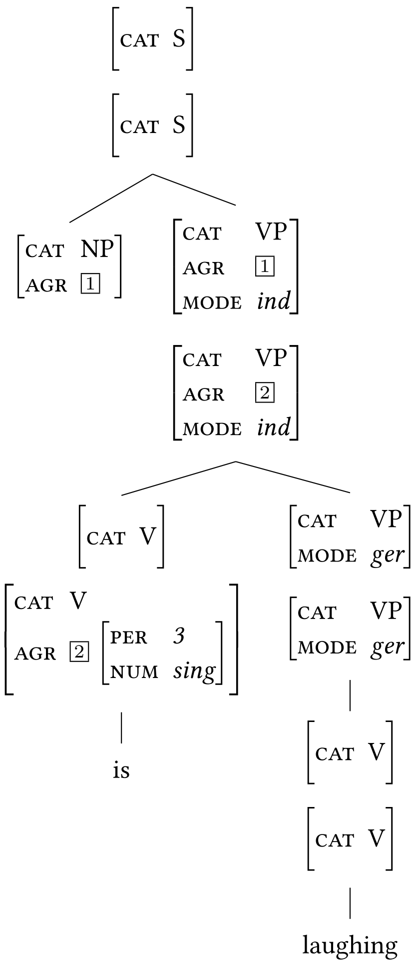

Figure 12.17: Result of obligatory adjunction in FTAG

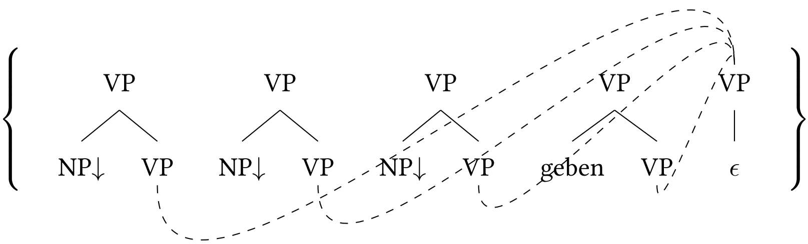

Figure 12.18: Lexicon set for *geben* ‘to give’ in V-TAG according to Rambow (1994: 6)

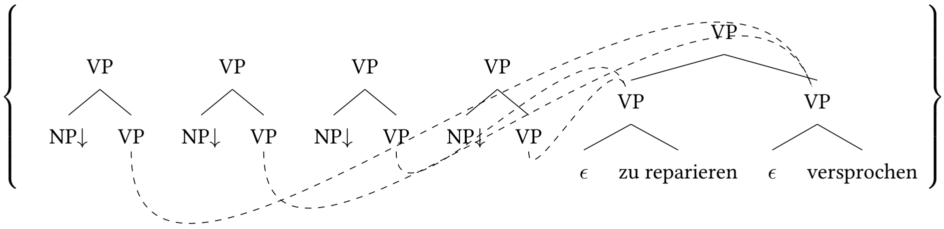

Figure 12.19: Analysis of the verbal complex *zu reparieren versprochen* in V-TAG

12.6.3 The competence-performance distinction and the generative capacity of tree-local MC-LTAG

In many of the theories discussed in this book, a distinction is made between competence and performance (Chomsky 1965: Section I.1). Competence theories are supposed to describe linguistic knowledge, whereas a performance theory should explain how linguistic knowledge is used and why we make mistakes during speech production and comprehension, etc. See Chapter 15 for further discussion.

Joshi, Becker & Rambow (2000) discuss examples of center self embedding of relative clauses as those in (13b), and follow Chomsky & Miller (1963: 286) in the assumption that the fact that this kind of embedding is only possible up to three levels should not be described by grammar, but is rather due to processing problems with the hearer independent of their principle abilities with regard to grammar.

(13) a. dass der Hund bellt, der die Katze jagt, die die Maus gefangen hat
that the dog barks that the cat chases that the mouse caught has
'that the dog that chases the cat that caught the mouse barks'

b. dass der Hund, [$_1$ der die Katze, [$_2$ die die Maus gefangen hat, $_2$] jagt
that the dog that the cat that the mouse caught has caught
$_1$] bellt
barks

What is interesting in this context is that it is possible to construct examples of center embedding so that they are easier to process for the hearer. In this way, it is possible to increase the number of center embeddings possible to process by one and therefore to show that all grammars that formulate a restriction that there may be at most two center-embedded relative clauses are incorrect. The following example from Hans Uszkoreit is easier to process since all embedded relative clauses are isolated and the verbs are separated by material from the higher clause.

(14) Die Bänke, [$_1$ auf denen damals die Alten des Dorfes, [$_2$ die allen
the benches on which back.then the old.people of.the village that all
Kindern, [$_3$ die vorbeikamen $_3$], freundliche Blicke zuwarfen $_2$], lange Stunden
children that came.by friendly glances gave long hours

schweigend nebeneinander saßen$_1$], mussten im letzten Jahr einem Parkplatz
silent next.to.each.other sat must in.the last year a car.park
weichen.
give.way.to

'The benches on which the older residents of the village, who used to give friendly glances to all the children who came by, used to sit silently next to one another had to give way to a car park last year.'

For other factors that play a role in processing, see Gibson (1998).

Joshi et al. (2000) discuss verbal complexes with reordered arguments. The general pattern that they discuss has the form shown in (15):

(15) $\sigma(\mathrm{NP}_1\ \mathrm{NP}_2 \dots \mathrm{NP}_n)\ \mathrm{V}_n \mathrm{V}_{n-1} \dots \mathrm{V}_1$

Here, σ stands for any permutation of noun phrases and V_1 is the finite verb. The authors investigate the properties of Lexicalized Tree Adjoining Grammar (LTAG) with regard to this pattern and notice that LTAG cannot analyze the order in (16) if the semantics is supposed to come out correctly.

(16) $\mathrm{NP}_2\ \mathrm{NP}_3\ \mathrm{NP}_1\ \mathrm{V}_3 \mathrm{V}_2 \mathrm{V}_1$

Since (17) is possible in German, LTAG is not sufficient to analyze all languages.

(17) dass ihm$_2$ das Buch$_3$ niemand$_1$ zu lesen$_3$ versprechen$_2$ darf$_1$
that him the book nobody to read promise be.allowed.to

'that nobody is allowed to promise him to read the book'

Therefore, they propose the extension of TAG discussed in Section 12.2; so-called *tree-local multi-component LTAG* (Tree-local MC-LTAG). They show that tree-local MC-TAG can analyze (17) but not (18) with the correct semantics. They claim that these orders are not possible in German and argue that in this case, unlike the relative clause examples, one has both options, that is, the unavailability of such patterns can be explained as a performance phenomenon or as a competence phenomenon.

(18) $\mathrm{NP}_2\ \mathrm{NP}_4\ \mathrm{NP}_3\ \mathrm{NP}_1\ \mathrm{V}_4 \mathrm{V}_3 \mathrm{V}_2 \mathrm{V}_1$

If we treat this as a performance phenomenon, then we are making reference to the complexity of the construction and the resulting processing problems for the hearer. The fact that these orders do not occur in corpora can be explained with reference to the principle of cooperativeness. Speakers normally want to be understood and therefore formulate their sentences in such a way that the hearer can understand them. Verbal complexes in German with more than four verbs are hardly ever found since it is possible to simplify very complex sentences with multiple verbs in the right sentence bracket by extraposing material and therefore avoiding ambiguity see Netter (1991: 5) and Müller (2007b: 262)).

The alternative to a performance explanation would involve using a grammatical formalism which is just powerful enough to allow embedding of two verbs and reordering of their arguments, but rules out embedding of three verbs and reordering of the arguments. Joshi et al. (2000) opt for this solution and therefore attribute the impossibility of the order of arguments in (18) to competence.

In HPSG (and also in Categorial Grammar and in some GB analyses), verbal complexes are analyzed by means of argument composition (Hinrichs & Nakazawa 1989a, 1994). Under this approach, a verbal complex behaves exactly like a simplex verb and the arguments of the verbs involved can be placed in any order. The grammar does not contain any restriction on the number of verbs that can be combined, nor any constraints that ban embedding below a certain level. In the following, I will show that many reorderings are ruled out by communication rules that apply even with cases of simple two-place verbs. The conclusion is that the impossibility of embedding four or more verbs should in fact be explained as a performance issue.

Before I present arguments against a competence-based exclusion of (18), I will make a more general comment: corpora cannot help us here since one does not find any instances of verbs with four or more embeddings. Bech (1955) provides an extensive collection of material, but had to construct the examples with four embedded verbs. Meurers (1999b: 94–95) gives constructed examples with five verbs that contain multiple auxiliaries or modal verbs. These examples are barely processable and are not relevant for the discussion here since the verbs in (18) have to select their own arguments. There are therefore not that many verbs left when constructing examples. It is possible to only use subject control verbs with an additional object (e. g. *versprechen* 'to promise'), object control verbs (e. g. *zwingen* 'to force') or AcI verbs (e. g. *sehen* 'to see' or *lassen* 'to let') to construct examples. When constructing examples, it is important make sure that all the nouns involved differ as much as possible with regard to their case and their selectional restrictions (e. g. animate/inanimate) since these are features that a hearer/reader could use to possibly assign reordered arguments to their heads. If we want to have patterns such as (18) with four NPs each with a different case, then we have to choose a verb that governs the genitive. There are only a very small number of such verbs in German. Although the example constructed by Joshi et al. (2000) in (9b) fulfills these requirements, it is still very marked. It therefore becomes clear that the possibility of finding a corresponding example in a newspaper article is extremely small. This is due to the fact that there are very few situations in which such an utterance would be imaginable. Additionally, all control verbs (with the exception of *helfen* 'to help') require an infinitive with *zu* 'to' and can also be realized incoherently, that is, with an extraposed infinitival complement without verbal complex formation. As mentioned above, a cooperative speaker/author would use a less complex construction and this reduces the probability that these kinds of sentences arise even further.

Notice that tree-local MC-LTAG does not constrain the number of verbs in a sentence. The formalism allows for an arbitrary number of verbs. It is therefore necessary to assume, as in other grammatical theories, that performance constraints are responsible for the fact that we never find examples of verbal complexes with five or more verbs. Tree-local MC-LAG makes predictions about the possibility of arguments to be reordered. I consider it wrong to make constraints regarding mobility of arguments dependent on the power of the grammatical formalism since the restrictions that one finds are independent of verbal complexes and can be found with simplex verbs taking just two arguments. The problem with reordering is that it still has to be possible to assign the noun phrases to the verbs they belong to. If this assignment leads to ambiguity that cannot be resolved by case, selectional restrictions, contextual knowledge or intonation, then the unmarked constituent order is chosen. Hoberg (1981: 68) shows this very nicely with examples similar to the following:[5]

(19) a. Hanna hat immer schon gewußt, daß das Kind sie verlassen will.
Hanna has always already known that the child she leave wants
'Hanna has always known that the child wants to leave her.'

b. # Hanna hat immer schon gewußt, daß sie das Kind verlassen will.
Hanna has always already known that she the child leave wants
Preferred reading: 'Hanna has always known that she wants to leave the child.'

c. Hanna hat immer schon gewußt, daß sie der Mann verlassen will.
Hanna has always already known that she the.NOM man leave wants.to
'Hanna has always known that the man wants to leave her.'

It is not possible to reorder (19a) to (19b) without creating a strong preference for another reading. This is due to the fact that neither *sie* 'she' nor *das Kind* 'the child' are unambiguously marked as nominative or accusative. (19b) therefore has to be interpreted as Hanna being the one that wants something, namely to leave the child. This reordering is possible, however, if at least one of the arguments is unambiguously marked for case as in (19c).

For noun phrases with feminine count nouns, the forms for nominative and accusative as well as genitive and dative are the same. For mass nouns, it is even worse. If they are used without an article, all cases are the same for feminine nouns (e. g. *Milch* 'milk') and also for masculines and neuters with exception of the genitive. In the following example from Wegener (1985: 45) it is hardly possible to switch the dative and accusative object, whereas this is possible if the nouns are used with articles as in (20c,d):

(20) a. Sie mischt Wein Wasser bei.
she mixes wine water into
'She mixes water into the wine.'

b. Sie mischt Wasser Wein bei.
she mixes water wine into
'She mixes wine into the water.'

[5] Instead of *das* 'the', Hoberg uses the possessive pronoun *ihr* 'her'. This makes the sentences more semantically plausible, but one then gets interference from the linearization requirements for bound pronouns. I have therefore replaced the pronouns with the definite article.

c. Sie mischt dem Wein das Wasser bei.
she mixes the.DAT wine the.ACC water into
'She mixes the water into the wine.'

d. Sie mischt das Wasser dem Wein bei.
she mixes the.ACC water the.DAT wine into
'She mixes the water into the wine.'

The two nouns can only be switched if the meaning of the sentence is clear from the context (e. g. through explicit negation of the opposite) and if the sentence carries a certain intonation.

The problem with verbal complexes is now that with four noun phrases, two of them almost always have the same case if one does not wish to resort to the few verbs governing the genitive. A not particularly nice-sounding example of morphologically unambiguously marked case is (21):

(21) weil er den Mann dem Jungen des Freundes gedenken
because he.NOM the.ACC man the.DAT boy of.the.GEN friend remember
helfen lassen will
help let wants
'because he wants to let the man help the boy remember his friend'

Another strategy is to choose verbs that select animate and inanimate objects so that animacy of the arguments can aid interpretation. I have constructed such an example where the most deeply embedded predicate is not a verb but rather an adjective. The predicate *leer fischen* 'to fish empty' is a resultative construction that should be analyzed parallel to verbal complexes (Müller 2002a: Chapter 5).

(22) weil niemand$_1$ [den Mann]$_2$ [der Frau]$_3$ [diesen Teich]$_4$ leer$_4$
because nobody.NOM the.ACC man the.DAT woman this.ACC pond empty
fischen$_3$ helfen$_2$ sah$_1$
fish help saw
'because nobody saw the man help the woman fish the pond empty'

If one reads the sentences with the relevant pauses, it is comprehensible. Case is unambiguously marked on the animate noun phrases and our word knowledge helps us to interpret *diesen Teich* 'this pond' as the argument of *leer* 'empty'.

The sentence in (22) would correctly be analyzed by an appropriately written tree-local MC-LTAG and also by argument composition analyses for verbal complexes and resultative constructions. The sentence in (23) is a variant of (22) that corresponds exactly to the pattern of (18):

(23) weil [der Frau]$_2$ [diesen Teich]$_4$ [den Mann]$_3$ niemand$_1$ leer$_4$
because the.DAT woman this.ACC pond the.ACC man nobody.NOM empty
fischen$_3$ helfen$_2$ sah$_1$
fish help saw
'because nobody saw the man help the woman fish the pond empty'

(23) is more marked than (22), but this is always the case with local reordering (Gisbert Fanselow, p. c. 2006). This sentence should not be ruled out by the grammar. Its markedness is more due to the same factors that were responsible for the markedness of reordering of arguments of simplex verbs. Tree-local MC-LTAG can not correctly analyze sentences such as (23), which shows that this TAG variant is not sufficient for analyzing natural language.

There are varying opinions among TAG researchers as to what should be counted as competence and what should be counted as performance. For instance, Rambow (1994: 15) argues that one should not exclude reorderings that cannot be processed by means of competence grammar or the grammatical formalism. In Chapter 6, he presents a theory of performance that can explain why the reordering of arguments of various verbs in the middle field is harder to process. One should therefore opt for TAG variants such as V-TAG or TT-MC-TAG (Lichte 2007) that are powerful enough to analyze the diverse reorderings and then also use a performance model that makes it possible to explain the gradual differences in acceptability.

An alternative to looking for a grammatical formalism with minimal expressive power is to not restrict the grammatical formalism at all with regard to its expressive power and instead develop as restrictive linguistic theories as possible. For further discussion of this point, see Chapter 17.

12.7 Summary and classification

In sum, we have seen the following: LTAG is lexicalized, that is, there is at least one lexical element in every tree. There are not any trees that correspond to the rule S → NP, VP since no words are mentioned in this rule. Instead, there are always complex trees that contain the subject NP and the VP. Inside the VP, there can be as much structure as is necessary to ensure that the verb is contained in the tree. As well as the head, elementary trees in LTAG always contain the arguments of the head. For transitive verbs, this means that both the subject and the object have to be components of the elementary tree. This is also true of the trees used to analyze long-distance dependencies. As shown in Figure 12.13, the object must be part of the tree. The fact that the object can be separated from the verb by multiple sentence boundaries is not represented in the elementary tree, that is, recursive parts of grammar are not contained in elementary trees. The relevant effects are achieved by adjunction, that is, by insertion of material into elementary trees. The elementary tree for extraction in Figure 12.13 differs from the elementary tree for *likes* given in Figure 12.4 for the use in normal SVO clauses. Every minimal construction, in which *likes* can occur (subject extraction, topicalization, subject relative clauses, object relative clauses, passive, …) needs its own elementary tree (Kallmeyer & Joshi 2003: 10). The different elementary trees can be connected using lexical rules. These lexical rules map a particular tree treated as underlying to other trees. In this way, it is possible to derive a passive tree from an active tree. These lexical rules are parallel to transformations in Transformational Grammar, however, one should bear in mind that there is always a lexical element in the tree, which makes the entire grammar more restrictive than grammars with free transformations.

An interesting difference to GB and variants of LFG, CG, and HPSG that assume empty elements is that the variants of TAG presented here[6] do not contain empty elements in the lexicon. They can be used in trees but trees are listed as a whole in the lexicon.

Elementary trees can be of any size, which makes TAG interesting for the analysis of idioms (see Section 18.2). Since recursion is factored out, trees can contain elements that appear very far away from each other in the derived tree (extended domains of locality).

Kasper, Kiefer, Netter & Shanker (1995) show that it is possible to transfer HPSG grammars that fulfill certain requirements into TAG grammars. This is interesting as in this way one arrives at a grammar whose complexity behavior is known. Whereas HPSG grammars are generally in the type-0 area, TAG grammars can, depending on the variant, fall into the realm of type-2 languages (context-free) or even in the larger set of the mildly context-sensitive grammars. Yoshinaga, Miyao, Torisawa & Tsujii (2001) have developed a procedure for translating FB-LTAG grammars into HPSG grammars.

Comprehension questions

1. How are long-distance dependencies analyzed in TAG? Does one need empty elements for this?

2. Is it possible to analyze the reordering of arguments of multiple verbs using standard TAG processes?

Exercises

1. Analyze the following string in LTAG:

    (24) der dem König treue Diener
    the the.DAT king loyal servant
    'the servant loyal to the king'

Further reading

Some important articles are Joshi, Levy & Takahashi (1975), Joshi (1987a), and Joshi & Schabes (1997). Many works discuss formal properties of TAG and are therefore not particularly accessible for linguistically interested readers. Kroch & Joshi (1985) give a good overview of linguistic analyses. An overview of linguistic and computational linguistic works in TAG can be found in the volume edited by Abeillé and Rambow from 2000. Rambow (1994) compares his TAG variant (V-TAG) to Karttunen's *Radical*

[6] See Rambow (1994) and Kallmeyer (2005: 194), however, for TAG analyses with an empty element in the lexicon.

Lexicalism approach, Uszkoreit's GPSG, Combinatorial Categorial Grammar, HPSG and Dependency Grammar.

Shieber & Johnson (1993) discuss psycholinguistically plausible processing models and show that it is possible to do incremental parsing with TAG. They also present a further variant of TAG: synchronous TAG. In this TAG variant, there is a syntactic tree and a semantic tree connected to it. When building syntactic structure, the semantic structure is always built in parallel. This structure built in parallel corresponds to the level of Logical Form derived from S-Structure using transformations in GB.

Rambow (1994: Chapter 6) presents an automaton-based performance theory. He applies it to German and shows that the processing difficulties that arise when reordering arguments of multiple verbs can be explained.

Kallmeyer & Romero (2008) show how it is possible to derive MRS representations directly via a derivation tree using FTAG. In each top node, there is a reference to the semantic content of the entire structure and each bottom node makes reference to the semantic content below the node. In this way, it becomes possible to insert an adjective (e. g. *mutmaßlichen* 'suspected') into an NP tree *alle Mörder* 'all murderers' so that the adjective has scope over the nominal part of the NP (*Mörder* 'murderers'): for adjunction of the adjective to the N node, the adjective can access the semantic content of the noun. The top node of *mutmaßlichen* is then the top node of the combination *mutmaßlichen Mörder* 'suspected murderers' and this ensures that the meaning of *mutmaßlichen Mörder* is correctly embedded under the universal quantifier.

Part II

General discussion