J Log Lang Inf (2011) 20:343–359
DOI 10.1007/s10849-011-9133-1

Lexicalized Non-Local MCTAG with Dominance Links is NP-Complete

Lucas Champollion

Published online: 8 May 2011
© Springer Science+Business Media B.V. 2011

Abstract An NP-hardness proof for non-local Multicomponent Tree Adjoining Grammar (MCTAG) by Rambow and Satta (1st International Workshop on Tree Adjoining Grammers 1992), based on Dahlhaus and Warmuth (in J Comput Syst Sci 33:456–472, 1986), is extended to some linguistically relevant restrictions of that formalism. It is found that there are NP-hard grammars among non-local MCTAGs even if any or all of the following restrictions are imposed: (i) lexicalization: every tree in the grammar contains a terminal; (ii) dominance links: every tree set contains at most two trees, and in every such tree set, there is a link between the foot node of one tree and the root node of the other tree, indicating that the former node must dominate the latter in the derived tree. This is the version of MCTAG proposed in Becker et al. (Proceedings of the 5th conference of the European chapter of the Association for Computational Linguistics 1991) to account for German long-distance scrambling. This result restricts the field of possible candidates for an extension of Tree Adjoining Grammar that would be both mildly context-sensitive and linguistically adequate.

Keywords Tree Adjoining Grammar · MCTAG · NP-complete · Dominance links · Lexicalization · Mildly context-sensitive · Scrambling

1 Introduction

Much work at the intersection of generative syntax and formal language theory has been devoted to determining whether natural language is parsable in polynomial time. In this context, one of the most appealing features of the Tree Adjoining Grammar

L. Champollion (✉)
Tübinger Zentrum für Linguistik, Eberhard Karls Universität Tübingen, Nauklerstr. 35, 72074 Tübingen, Germany
e-mail: champoll@gmail.com

formalism (TAG) is precisely that it is polynomially parsable. TAG was introduced in Joshi et al. (1975); for a recent introduction to TAG, see Joshi and Schabes (1997). Intuitively, a TAG consists of a finite set of elementary trees labeled with terminals and nonterminals (terminals only label leaf nodes). The elementary trees are partitioned into two sets: *initial* trees and *auxiliary* trees. A derivation always starts with an initial tree and proceeds by combining elementary trees with it to derive larger trees. Trees can be combined through two operations, *substitution* and *adjunction*.

– Substitution is used to attach an initial tree α into a *substitution slot* of a host tree α'. Substitution slots are specially marked leaf nodes whose label must be identical with the root of α.
– Adjunction is used to attach an auxiliary tree α to a node n of an elementary tree α'. Auxiliary trees must have a *foot node*, a leaf node whose label is identical to the label of the root, conventionally marked with an asterisk. For adjunction to be allowed, n must carry the same label as the root and foot nodes of α. Adjunction is carried out by replacing the node n with the entire tree α. The foot node of α is then replaced by the subtree under n. It is possible for each node to specify the set of auxiliary trees (if any) that can be adjoined, and also whether adjunction at this node is obligatory.

When the fringe of a tree derived in this way contains only terminals, it represents a word in the language generated by the TAG.

As mentioned, languages generated by TAG can be parsed in polynomial time. However, it has been found early on (Kroch and Joshi 1987) that there are constructions in natural language syntax which cannot be given the right structural descriptions using standard TAG. Various ways have since then been proposed to extend TAG with just the right amount of additional generative power that is needed to describe natural language, while keeping efficient parsability. This paper restricts the field of candidates for such an extension of TAG by showing that some of the proposed extensions are in fact NP-complete. The proofs in this paper are mathematically straightforward. They are linguistically significant, however, because the extensions to which they apply have been argued to be necessary in order to give TAG sufficient power to model natural language syntax. This section describes the TAG extensions in question and the formal language theoretic context in which they have been proposed. Sections 2 through 5 present the formal proofs. Section 6 concludes, and places the results in a linguistic context.

Joshi (1985) proposed that the class of grammars that is needed to describe natural languages might be characterized as the class of *mildly context-sensitive grammars* (MCSG). This class includes those grammar formalisms which allow only a limited number of cross-serial dependencies, are parsable in polynomial time, and can only define languages that have the constant growth property. In this context, polynomial parsability is understood as referring to the fixed word recognition problem, where the grammar is not part of the input, as opposed to the universal recognition problem (Aravind Joshi, p.c.). Accordingly, this paper focuses on the fixed recognition problem; by contrast, the universal recognition problems of most TAG extensions discussed here are NP-hard. In a sense, the fixed recognition problem is the linguistically more interesting question: If we interpret formal complexity results as telling us something about how hard language is, then the fact that a given grammar formalism has an NP-hard universal recognition problem merely suggests that turning a grammar into a working parser can be a hard task if the grammar has been encoded in this formalism. By contrast, if the fixed recognition problem of a grammar formalism is also NP-hard, this suggests that either there are possible languages in which it can be a very hard task to process novel sentences, or this formalism is not adequate for describing natural language.

Among the TAG extensions that were first investigated, a promising candidate for a linguistically adequate MCSG seemed to be multicomponent TAG (MCTAG). MCTAGs were first discussed by Joshi et al. (1975) and Joshi (1985) and later defined precisely by Weir (1988). For the (rather lengthy) formal definitions of TAG and of the different MCTAGs, the reader is referred to Weir (1988); for an insightful and purely declarative alternative characterization, see Kallmeyer (2009). Intuitively, in an MCTAG, instead of auxiliary trees being single trees we have auxiliary sets, where a set consists of one or more (but still a fixed number of) auxiliary trees. Adjunction is defined as the simultaneous adjunction of all trees in a set to different nodes. Several variants of MCTAG have been defined. In a *tree-local* MCTAG, all trees from one set S must be simultaneously adjoined into the same elementary tree T. In a *set-local* MCTAG, all trees from one set S must be simultaneously adjoined into trees that all belong to the same set S_2. If we only require that trees from one set must be adjoined simultaneously, but drop the locality requirement, we obtain *non-local* MCTAG. As shown in Søgaard (2009), non-local MCTAG is context-sensitive, because there is a linear upper bound on the size of its derivation structures. However, it is not *mildly* context-sensitive, because its fixed recognition problem is NP-complete. This latter result is from Rambow and Satta (1992) and Rambow (1994) and is the basis for the work in this paper.[1]

The definition of the class of mildly context-sensitive grammars in Joshi (1985) was left informal: in particular, the requirement that only a limited number of cross-serial dependencies be allowed was not formally defined. Vijay-Shanker et al. (1987) and Weir (1988) proposed the class of linear context-free-rewrite systems (LCFRS) as a formal characterization of the MCSG class. LCFRS are equally powerful to set-local MCTAG, in the sense that for each set-local MCTAG, there is a strongly equivalent LCFRS, and for each LCFRS, there is a weakly equivalent set-local MCTAG. Becker et al. (1992) and Rambow (1994) argue that long-distance scrambling in German puts natural language even beyond the power of LCFRS. Provided that LCFRS is indeed adequate as a characterization of the MCSG class, this implies that natural language is not mildly context-sensitive, contra Joshi (1985). The result also implies that a number of equivalent or less powerful formalisms, such as head grammars (Pollard 1984) and combinatory categorial grammars (Steedman 1988), are too weak to represent natural language, since these formalisms can be classified as LCFRS (Joshi et al. 1991).[2]

[1] The *universal* recognition problem for non-local MCTAG is NP-complete as well, as shown in Søgaard (2009).

[2] However, there is some reason to believe that German scrambling is in fact more restricted than described in Becker et al. (1991) and that scrambling might therefore not be beyond LCFRS after all (see Sect. 6).

Despite these results, one can still hope to find a language class that is adequate for natural language and has the property of being parsable in polynomial time. This is so because LCFRS do not include all languages that are polynomially parsable.[3] In particular, restricted variants of non-local MCTAG might be able to describe German scrambling data and still be polynomially parsable. Thus, Becker et al. (1991) propose to deal with German scrambling by non-local MCTAG with dominance links (MCTAG-DL). In this modification of non-local MCTAG, an additional requirement is added: an ordered pair may be specified between any two nodes of different trees in the same tree set. In the final derived tree, the first node must dominate the other (though not necessarily immediately). In the restriction studied in this paper, the foot node of one of the components of an auxiliary set has to dominate the root node of the other component in the same auxiliary set. (This also means that there are no more than two trees in each auxiliary set.) Under an alternative definition, dominance links are an optional feature that may or may not be present in the grammar. In that sense, every non-local MCTAG is a MCTAG-DL, and therefore MCTAG-DL is of course NP-hard. Here, however, I only consider MCTAG-DL in which dominance links are obligatorily present in each auxiliary set.

MCTAG-DL are widely used in grammar modeling but are formally not well understood. As (Rambow, 1994, p. 59) writes, "[w]hile any linguistic application of any of the MCTAG systems will use dominance links, they have not been studied formally"; he conjectures that they do not appear to decrease weak generative power. Linguistically, dominance links are often used to enforce c-command relationships between displaced constituents and their traces. The first linguistic application of MCTAG with dominance links goes back to Kroch and Joshi (1987), where they are used for the analysis of extraposition in English. These authors, unlike Becker et al. (1991), impose the additional constraint of tree-locality. Kallmeyer (2009) suspects that dominance links in connection with tree-locality or set-locality can be simulated by choosing appropriate node labels. If this is so, then dominance links in connection with locality constraints do not change the generative power of the grammar in these cases. This means that the restrictions of MCTAG used by Kroch and Joshi (1987) do not reach beyond LCRFS, and are therefore not expressive enough for natural language in general, if Becker et al. (1992) and Rambow (1994) are right. For this reason, I concentrate on *non-local* MCTAG-DL in this paper.

While non-local multi-component rewriting systems tend to be NP-complete (see Rambow 1994, p. 62 for an overview), there are exceptions.[4] However, in this paper it is shown that the fixed and (therefore) the universal recognition problem for non-local MCTAG-DL are in fact NP-hard. As mentioned above, the fixed and universal recognition problem for non-local MCTAG *without* dominance links are already known to be NP-complete. Therefore, a conjecture by Rambow (1994) that

[3] For example, the positive version of Range Concatenation Grammars covers exactly the class of polynomially recognizable languages, but it is more powerful than LCFRS because its languages are not semilinear (Boullier 1998).

[4] One such exception is Rambow's non-local V-TAG, which is like non-local MCTAG-DL except that elements of a tree set need not be used simultaneously in the derivation. Lexicalized V-TAG with *integrity constraints* (node diacritics that prevent dominance links from going through them) is polynomially parsable. See Sect. 6 for discussion.

dominance links do not decrease the weak generative power of MCTAG is corroborated. Non-local MCTAG-DL is beyond LCFRS; it is NP-complete, and therefore by definition not mildly context-sensitive. This is the main result of this paper.

It is generally accepted that only the *lexicalized* variants of TAGs are suitable candidates for encoding natural language. Schabes (1990) defines a lexicalized grammar as a grammar in which every elementary structure is associated with a lexical item, and every lexical item is associated with a finite set of elementary structures. From a theoretical perspective, lexicalization is justified by the assumption that grammatical structure is projected from (i.e. listed in) the lexicon. From a practical perspective, the interest stems from the considerable importance of word-based corpora in natural language processing (Rambow et al. 2001).

While standard TAGs are closed under lexicalization (Schabes 1990), it is not known whether this also applies to non-local MCTAG. So it would be conceivable that *lexicalized* non-local MCTAG are mildly context-sensitive. However, it is shown below that the fixed recognition problem for lexicalized non-local MCTAG is in fact NP-complete. Moreover, even if both restrictions (dominance links and lexicalization) are applied to non-local MCTAG at the same time, it still remains NP-complete.

2 Non-Local MCTAG is NP-Hard

This section presents a detailed proof of the NP-hardness of standard non-local MCTAG with adjunction constraints (MCTAG from now on). This is essentially the proof that was reported by Dahlhaus and Warmuth (1986) for scattered context grammars (SCG), a grammar class defined in Greibach and Hopcroft (1969). A scattered context grammar is a rewriting system similar to a context-free grammar, except that several nonterminals can be rewritten in parallel. Each production in a scattered context grammar specifies a sequence of nonterminals that must be present in the input string in a specific order, but they do not have to be adjacent to each other. If the order requirement is dropped, we obtain *unordered* scattered context grammars (USCG). Since USCG are to context-free grammar (CFG) what multicomponent TAG are to TAG, we can think of USCG as "multicomponent CFG": Each USCG production can be represented as a set of CFG productions that must be applied simultaneously.

The proof in Dahlhaus and Warmuth (1986) shows NP-completeness for a language which is generated by a particular SCG as well as by an equivalent USCG. It was noted by Rambow and Satta (1992) and Rambow (1994) that the proof carries over to certain MCTAGs in principle, but they do not actually perform the construction of the NP-hard grammar. I flesh out the proof that they had in mind in detail here, as we are going to need it later. The main intuition behind the construction is that just like a TAG can simulate a CFG, a non-local MCTAG can simulate a USCG. The property of non-local MCTAG and USCG that underlies this proof is the following: We can introduce pairs of terminals into the derivation at two different (indeed arbitrarily distant) places in the tree, but we must introduce them at the same time. This allows us to build a grammar that counts up to the same arbitrary number in two places of the derivation. In the final string, each of these numbers is expressed as a block of identical terminals. In designing our grammar, we may either choose to delimit these blocks from each other by special separator symbols, or simulate addition by leaving out these separators. In this case, since the string contains no record of the derivation, a recognizer only sees the sum and not the summands, and must in effect guess which summands have been chosen.

I now present a polynomial reduction from the NP-complete problem *3-Partition* to a specific non-local MCTAG.

3-Partition.

Instance. A set of $3k$ natural numbers n_i, and a bound B.
Question. Can the numbers be partitioned into k subsets of cardinality 3, each of which sums to B?

An instance of 3-Partition can be described as the sequence $\langle n_1, \ldots, n_{3k}, B \rangle$, or equivalently the string $xa^{n_1}xa^{n_2}\ldots xa^{n_{3k}}(yb^B)^k$ where a, b, x, y are arbitrary symbols. (In this string, x and y are only used as separators. It will be seen later why the end of the string was chosen to be repeated k times.) I will provide below a non-local MCTAG G_1 that has the property that $\langle n_1, \ldots, n_{3k}, B \rangle$ is a positive instance of 3-Partition if and only if the string $xa^{n_1}xa^{n_2}\ldots xa^{n_{3k}}(yb^B)^k$ is accepted by G_1.

3-Partition is strongly NP-complete, which means that it remains NP-complete even if the numbers n_i are encoded in unary (Garey and Johnson 1979). Since the length of the string given above is polynomial in the length of a unary encoding of the instance, any instance of 3-Partition can be transformed into an instance of the word problem of G in polynomial time.

I now exhibit the MCTAG G_1, which is a translation of the scattered context grammar G in Dahlhaus and Warmuth (1986), Sect. 5. G_1 is displayed Fig. 1; all figures are found at the end of this paper. (The productions of G are displayed in Fig. 1 as well for comparison.)

To simplify the construction, assume that 3-Partition is restricted in the way that there are at least three numbers n_i (i.e. that $k \geq 1$) and that each of the numbers n_i is greater or equal to two. As usual in the TAG literature, I indicate obligatory adjunction sites with OA and null-adjunction sites with NA. Foot nodes are always null-adjunction sites and are therefore not explicitly marked as such. There are no substitution sites in G_1.

G_1 produces only strings of the form $xa^{n_1}xa^{n_2}\ldots xa^{n_{3k}}yb^{m_1}yb^{m_2}\ldots yb^{m_k}$. In addition, all the strings it produces each contain an equal number of a's and b's, because each tree set that is adjoined adds an equal number of a's and b's to the derivation.

To get an idea of how the grammar works, note that all terminals are introduced to the left of the spine of their auxiliary tree, so whatever is introduced towards the top of the derived tree will appear towards the left of the string. In all derived trees, any of X and $\overline{X}$ will always dominate any of Y, $\overline{Y}$ and $\hat{Y}$, and any of x and a will c-command and precede any of y and b.

At all times there is at most one of $\{X, \overline{X}\}$ in the derivation. Assuming without loss of generalization that $\beta_{create-triple}$ is always used as early as possible, all derivations allowed by G_1 follow the same general pattern:

$G_1 = (NT, \Sigma, S, I, A)$ where

$$\begin{aligned} NT &= \{X, \overline{X}, Y, \overline{Y}, \hat{Y}\} \\ \Sigma &= \{a, b, x, y\} \\ I &= \{\alpha_{start}\} \\ A &= \{\beta_{create-triple}, \beta_{consume-y}, \beta_{consume-\hat{y}}, \beta_{fill-triple}, \beta_{close-triple}, \beta_{end}\} \end{aligned}$$

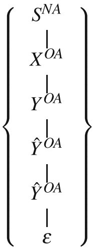
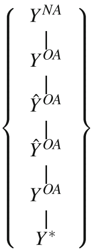
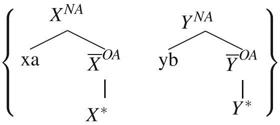
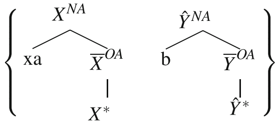
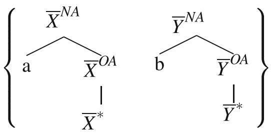
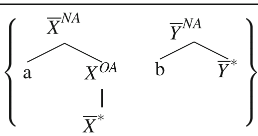
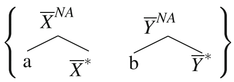
Label	Tree set	Corresponding USCG production
α_{start}		$S \rightarrow XY\hat{Y}\hat{Y}$
$\beta_{create-triple}$		$Y \rightarrow Y\hat{Y}\hat{Y}Y$
$\beta_{consume-y}$		$X \rightarrow xa\overline{X},\ Y \rightarrow yb\overline{Y}$
$\beta_{consume-\hat{y}}$		$X \rightarrow xa\overline{X},\ \hat{Y} \rightarrow b\overline{Y}$
$\beta_{fill-triple}$		$\overline{X} \rightarrow a\overline{X},\ \overline{Y} \rightarrow b\overline{Y}$
$\beta_{close-triple}$		$\overline{X} \rightarrow aX,\ \overline{Y} \rightarrow b$
β_{end}		$\overline{X} \rightarrow a,\ \overline{Y} \rightarrow b$

Fig. 1 The MCTAG G_1 with its corresponding USCG productions

1 Initialize the derivation by α_{start}.
2 Create k triples by using $\beta_{create-triple}$ as many times as needed.
3 Pick the X and some Y (resp. $\hat{Y}$) and use $\beta_{consume-y}$ (resp. $\beta_{consume-\hat{y}}$) to generate xa on the left and yb (resp. b) on the right. This introduces $\overline{X}$ on the left and $\overline{Y}$ on the right.
4 Optionally use $\beta_{fill-triple}$ to add an equal number of a's and b's to the left and right.
5 Finally replace $\overline{X}$ by a and $\overline{Y}$ by b. Either $\beta_{close-triple}$ or β_{end} can be used for this. The only difference consists in whether another X is introduced. But there is no real choice here: If there are any Y's or $\hat{Y}$'s left on the right, they need to be consumed by introducing an X on the left and then going through steps 3 through 5 again with that X. If not, no X can be introduced or the derivation would get stuck.

This way, the grammar produces a sequence of blocks of a's followed by a sequence of blocks of b's. The sizes of the blocks of a's correspond to the numbers n_i. While X is deriving xa^{n_i} followed by X, either some Y derives yb^{n_i} or some $\hat{Y}$ derives b^{n_i}. There is a block of b's for each n_i, but the blocks of b's are permuted and grouped in threes. While the grammar produces more words than the ones that correspond to solutions of 3-Partition, those words in which each group of three sums to B are exactly the ones that correspond to some solution.

The behavior of G_1 can be mimicked by a "multicomponent CFG", i.e. an USCG (Dahlhaus and Warmuth 1986). The productions of this USCG are reproduced in Fig. 2, along with a sample derivation. A corresponding derivation is also available in G_1. For ease of reference, each rule is also reproduced in Fig. 1 next to the tree that corresponds to it.

I now give the formal NP-hardness proof.[5] Suppose we are given a solution of the instance of 3-Partition, i.e. disjoint sets $A_1, \ldots, A_k$, each of which contains 3 n_i's that add to B. It will be shown that the word $w = xa^{n_1}xa^{n_2}\ldots xa^{n_{3k}}(yb^B)^k$ that describes the instance of 3-Partition is in $L(G_1)$.

For any derived MCTAG tree t, do a left-to-right preorder traversal of t concatenating all the node labels and skipping any non-terminals at which adjunction can not or can no longer take place, and call the resulting string the *unsaturated yield of* t. Define a relation "$\Rightarrow$" ("is rewritten to") as holding between two strings s_1 and s_2 wrt. an MCTAG G iff there exist trees t_1, t_2 with unsaturated yields s_1, s_2 such that t_2 can be obtained from t_1 in a single (possibly multicomponent) substitution or adjunction step. We write $G \Rightarrow s$ iff G contains an initial tree t rooted in the start symbol of G such that there is a string s_t that is the unsaturated yield of t and $s_t \Rightarrow s$.[6] As usual, we write $\overset{*}{\Rightarrow}$ for the reflexive and transitive closure of $\Rightarrow$. Obviously, for all $w \in \Sigma^*$, G derives w iff $G \overset{*}{\Rightarrow} w$.

Clearly $G_1 \overset{*}{\Rightarrow} X(Y\hat{Y}\hat{Y})^k$. Associate each set $A_q, 1 \leq q \leq k$, with the qth group $Y\hat{Y}\hat{Y}$ and associate each of the three elements of the set with one of the three

5 From Dahlhaus and Warmuth (1986), with a few extensions.

6 This notion is intended to capture the close relationship between an MCTAG G_1 and its corresponding USCG. At any point in the derivation, the unsaturated yield of an unfinished derived MCTAG tree will be identical with the string that the USCG is rewriting.

USCG $G = (NT, \Sigma, P, S)$ where

$$NT = \{X, \overline{X}, Y, \overline{Y}, \hat{Y}\}$$
$$\Sigma = \{a, b, x, y\}$$
$$P = \{\text{start, create-triple, consume-}y\text{, consume-}\hat{y}\text{, fill-triple, close-triple, end}\}$$

Label	Production
start	$S \to XY\hat{Y}\hat{Y}$
create-triple	$Y \to Y\hat{Y}\hat{Y}Y$
consume-y	$X \to xa\overline{X},\ \ Y \to yb\overline{Y}$
consume-$\hat{y}$	$X \to xa\overline{X},\ \ \hat{Y} \to b\overline{Y}$
fill-triple	$\overline{X} \to a\overline{X},\ \ \overline{Y} \to b\overline{Y}$
close-triple	$\overline{X} \to aX,\ \ \overline{Y} \to b$
end	$\overline{X} \to a,\ \ \overline{Y} \to b$

`step 1`	start	X	$Y\hat{Y}\hat{Y}$	
`step 2`	create-triple	X	$Y\hat{Y}\hat{Y}$	$Y\hat{Y}\hat{Y}$
`step 3`	consume-y	$xa\overline{X}$	$Y\hat{Y}\hat{Y}$	$yb\overline{Y}\hat{Y}\hat{Y}$
`step 4`	fill-triple	$xaa\overline{X}$	$Y\hat{Y}\hat{Y}$	$ybb\overline{Y}\hat{Y}\hat{Y}$
`step 4`	fill-triple	$xaaa\overline{X}$	$Y\hat{Y}\hat{Y}$	$ybbb\overline{Y}\hat{Y}\hat{Y}$
`step 5`	close-triple	$xaaaaX$	$Y\hat{Y}\hat{Y}$	$ybbbb\hat{Y}\hat{Y}$
`step 3`	consume-$\hat{y}$	$xaaaa\ xaa\overline{X}$	$Yb\overline{Y}\hat{Y}$	$ybbbb\hat{Y}\hat{Y}$
`step 5`	close-triple	$xaaaa\ xaaX$	$Ybb\hat{Y}$	$ybbbb\hat{Y}\hat{Y}$
`step 3`	consume-$\hat{y}$	$xaaaa\ xaa\ xa\overline{X}$	$Ybb\hat{Y}$	$ybbbb\hat{Y}b\overline{Y}$
`step 4`	fill-triple	$xaaaa\ xaa\ xaa\overline{X}$	$Ybb\hat{Y}$	$ybbbb\hat{Y}bb\overline{Y}$
`step 5`	close-triple	$xaaaa\ xaa\ xaaaX$	$Ybb\hat{Y}$	$ybbbb\hat{Y}bbb$
`step 3`	consume-$\hat{y}$	$xaaaa\ xaa\ xaaaa\ xa\overline{X}$	$Ybb\hat{Y}$	$ybbbbb\overline{Y}bbb$
`step 5`	close-triple	$xaaaa\ xaa\ xaaaa\ xaaX$	$Ybb\hat{Y}$	$ybbbbbbbbb$
`step 3`	consume-y	$xaaaa\ xaa\ xaaaa\ xaa\ xa\overline{X}$	$yb\overline{Y}bb\hat{Y}$	$ybbbbbbbbb$
`step 4`	fill-triple	$xaaaa\ xaa\ xaaaa\ xaa\ xaa\overline{X}$	$ybb\overline{Y}bb\hat{Y}$	$ybbbbbbbbb$
`step 4`	fill-triple	$xaaaa\ xaa\ xaaaa\ xaa\ xaaa\overline{X}$	$ybbb\overline{Y}bb\hat{Y}$	$ybbbbbbbbb$
`step 4`	fill-triple	$xaaaa\ xaa\ xaaaa\ xaa\ xaaaa\overline{X}$	$ybbbb\overline{Y}bb\hat{Y}$	$ybbbbbbbbb$
`step 5`	close-triple	$xaaaa\ xaa\ xaaaa\ xaa\ xaaaaaX$	$ybbbbbbb\hat{Y}$	$ybbbbbbbbb$
`step 3`	consume-$\hat{y}$	$xaaaa\ xaa\ xaaaa\ xaa\ xaaaaa\ xa\overline{X}$	$ybbbbbbbb\overline{Y}$	$ybbbbbbbbb$
`step 5`	end	$xaaaa\ xaa\ xaaaa\ xaa\ xaaaaa\ xaa$	$ybbbbbbbbb$	$ybbbbbbbbb$

Fig. 2 Above, the USCG that corresponds to G_1. Below, a sample derivation of the 3-partition instance $\langle 4, 2, 3, 2, 5, 2;\ B = 9\rangle$. The step numbers refer to the pattern described in Sect. 2

symbols Y, $\hat{Y}$, and $\hat{Y}$, respectively, in the group. The association within each group is arbitrary. The derivation $X(Y\hat{Y}\hat{Y})^k \stackrel{*}{\Rightarrow} w$ is organized in $3k$ phases. In the jth phase, for $1 \leq j < 3k$, X is rewritten to $xa^{n_j}X$ and in parallel the Y-symbol (resp. $\hat{Y}$-symbol) that is associated with n_j is rewritten to yb^{n_j} (resp. b^{n_j}). In the $3k$th phase X

is rewritten to $xa^{n_{3k}}$ and in parallel the Y-symbol (resp. $\hat{Y}$-symbol) that is associated with n_{3k} is rewritten to $yb^{n_{3k}}$ (resp. $b^{n_{3k}}$). Since the numbers of A_q add to B, each group $Y\hat{Y}\hat{Y}$ derives yb^B.

For the opposite direction, we need to prove that each $w = xa^{n_1}xa^{n_2}\ldots xa^{n_{3k}}(yb^B)^k,\ w \in L(G_1)$, describes a solution of the instance of 3-Partition. Assume now that $G_1 \overset{*}{\Rightarrow} w$, where $w = xa^{n_1}xa^{n_2}\ldots xa^{n_{3k}}(yb^B)^k$. Normalize the derivation by adjoining all instances of $\beta_{create-triple}$ as early as possible within the derivation of w. The normalized derivation has the form:

$$G_1 \overset{*}{\Rightarrow} X(Y\hat{Y}\hat{Y})^k \overset{*}{\Rightarrow} w$$

The symbol X is rewritten to $\overline{X}$ and after a number of steps to X again. More exactly, X produces $xa^{n_i}X$ at the jth phase, for $1 \leq j < 3k$, and $xa^{n_{3k}}$ in the last phase. Furthermore, in the ith phase, for $1 \leq i \leq 3k$, a particular Y (resp. $\hat{Y}$) is rewritten to yb^{n_i} (resp. b^{n_i}). Observe that each non-terminal Y is responsible for a terminal y in w and the Y's produce exactly B b's. Each group thus corresponds to a different set of three numbers that adds to B and there are k such sets. □

3 Restriction to Dominance Links

I now restrict the above proof to MCTAG-DL. This is done by modifying the grammar G_1 to produce a strongly equivalent MCTAG-DL G_2. Since the two grammars have the same language, it follows that MCTAG-DL is also NP-hard.

Proof Call any element of $\{X, \overline{X}\}$ an *X-like symbol* and any element of $\{Y, \overline{Y}, \hat{Y}\}$ a *Y-like symbol*. Observe that in the tree α_{start} in G_1, and vacuously in all the other trees of the grammar, any X-like symbol dominates any Y-like symbol. Call any elementary or derived tree with this property an *X-over-Y tree*.

Add dominance links between the X-like foot nodes and the Y-like root nodes of the trees in each multicomponent set of G_1. Call the grammar obtained this way G_2 (see Fig. 3). A derived tree that violates any of these dominance links would have a Y-like root node dominate an X-like foot node and would therefore not be X-over-Y. In other words, the dominance links will never rule out an X-over-Y tree.

In every tree set in G_1, the tree with the X-like foot node contains only X-like non-terminals and the tree with the Y-like root node contains only Y-like non-terminals. Therefore, if the tree set is adjoined to a derived tree that is already X-over-Y, the resulting derived tree will also be X-over-Y. Moreover, adjoining the single auxiliary tree $\beta_{create-triple}$ to an X-over-Y derived tree always produces an X-over-Y derived tree.

By induction, it follows that all the derived trees produced by G_1 or G_2 are X-over-Y. Hence the dominance links that have been added to G_1 can never be violated. Therefore G_1 and G_2 are strongly equivalent. □

$G_2 = (NT, \Sigma, S, I, A)$ where

$$\begin{aligned} NT &= \{X, \overline{X}, Y, \overline{Y}, \hat{Y}\} \\ \Sigma &= \{a, b, x, y\} \\ I &= \{\alpha_{start}\} \\ A &= \{\beta_{create-triple}, \beta_{consume-y}, \beta_{consume-\hat{y}}, \beta_{fill-triple}, \beta_{close-triple}, \beta_{end}\} \end{aligned}$$

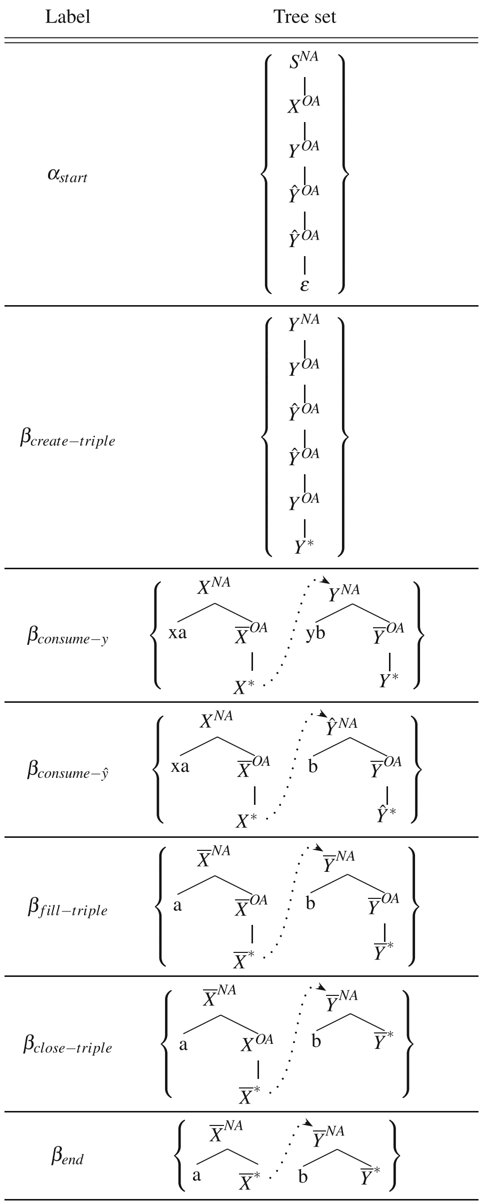


Fig. 3 The MCTAG with dominance links G_2. (Identical to G_1 except for the dominance links, which are indicated as dotted lines)

4 Restriction to Lexicalized Grammars

Here I modify the grammar G_1 to get a lexicalized grammar G_3 (see Fig. 4) that accepts a slightly different language than G_1 does. It is shown that this language is NP-hard as well.

Proof G_3 only differs from G_1 in the two trees α_{start} and $\beta_{create-triple}$, each of which has been added a new "dummy" terminal symbol #. Since the terminals in the other trees are always located to the left of the spine, the new symbols amass at the end of the word. Thus each word $w \in L(G_1)$ can be uniquely related to some word $w' \in L(G_3)$ which is identical to w except for k+1 dummy terminals at the end of w', where k is the number of times that $\beta_{create-triple}$ has been used in the derivation. (The additional dummy terminal comes from α_{start}.) Since k is also the number of sets of three numbers of an instance of 3-Partition, there is a straightforward polynomial time transformation between that instance and the corresponding word of L_3. □

Since both restrictions just presented can be applied to G_1 at the same time and do not interact, there obtains:

Corollary 1 *Lexicalized MCTAG with dominance links is NP-hard.*

5 NP-Completeness

The previous sections have shown that the *fixed* recognition problem for the languages generated by G_1, G_2 and G_3 are NP-hard. The *universal* recognition problem for non-local MCTAG is NP-complete, as shown in Søgaard (2009). This entails that the languages considered here have NP-complete fixed recognition problems.

The NP-completeness of these grammars can also be shown directly by a simple argument. It has been shown above that G_1 and G_2 are strongly equivalent, so the proof only needs to be carried out once for both of them. Every auxiliary tree set in G_1 except the unary set $\beta_{create-triple}$ introduces terminals into the derivation. So for any word w, the length of w is an upper bound on the amount of times each of these tree sets can have occurred in the derivation. The initial tree α_{start} is always used exactly once. Observe that the unary set $\beta_{create-triple}$ is used exactly k times where k is the amount of blocks of b's contained in w. So the number of steps to derive w can be guessed in linear time by a nondeterministic Turing machine. The same argument can be applied to show that each lexicalized MCTAG, such as G_3, is at most NP-complete. By definition, every derivation step introduces terminals. So it always takes at most $|w|$ steps to derive w.

6 Conclusion and Linguistic Implications

This paper establishes that the fixed recognition problem of non-local MCTAG with dominance links is NP-complete and is therefore outside LCFRS, a class of polynomially parsable formalisms that encodes the notion of mild context-sensitivity. As for non-local MCTAG *without* dominance links, the combined results in

$G_3 = (NT, \Sigma, S, I, A)$ where

$$
\begin{aligned}
NT &= \{X, \overline{X}, Y, \overline{Y}, \hat{Y}\} \\
\Sigma &= \{a, b, x, y, \#\} \\
I &= \{\alpha_{start}\} \\
A &= \{\beta_{create-triple}, \beta_{consume-y}, \beta_{consume-\hat{y}}, \beta_{fill-triple}, \beta_{close-triple}, \beta_{end}\}
\end{aligned}
$$

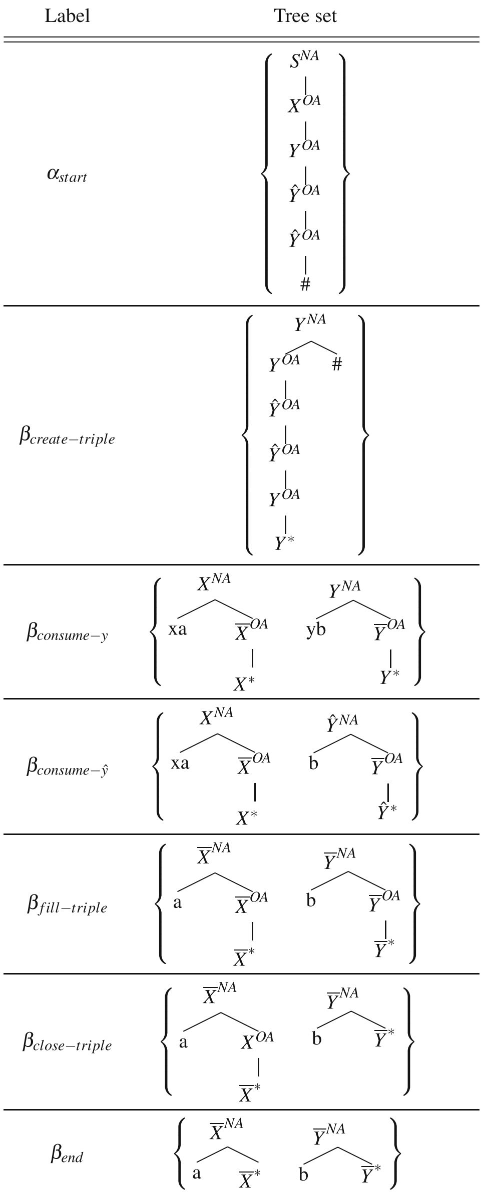


Fig. 4 The lexicalized MCTAG G_3. (Identical to G_1 except that new terminals have been added to α_{start} and to $\beta_{create-triple}$)

Rambow and Satta (1992) and Søgaard (2009) entail that the fixed recognition problem is also NP-complete. (Rambow and Satta (1992) show that it is NP-hard; Søgaard (2009) shows that it is in NP.) The conjecture by Rambow (1994) that dominance links do not decrease the weak generative power of MCTAG is therefore corroborated. All this remains the case even if only lexicalized grammars are considered. This result undermines the proposal by Becker et al. (1991) to model German scrambling by non-local MCTAG-DL, since we cannot adopt MCTAG-DL if we want to model language with a mildly context sensitive formalism, one of the primary motivations for the linguistic study of TAG and its variants.

However, there exist alternative views on the scrambling facts and how to interpret them in the context of formal language theory. Like any formal proof of a property of a natural language, the proof by Becker et al. (1992) that puts German scrambling outside LCFRS relies on specific empirical assumptions: in this case, that there is no bound on the number of verbal arguments can be scrambled at once; that there is no bound on the level of embedding (i.e., the number of verbs over which each argument can scramble); and that scrambled arguments can appear in any permutation. These assumptions are hard to check, because sentences involving four or more scrambled arguments are usually very hard to judge. Only certain special patterns are much easier to judge positively for large numbers of scrambled arguments, such as when the order of the scrambled arguments is exactly identical to the order of their verbs, or exactly opposite to that order (Aravind Joshi, p.c.) Moreover, some native speakers are reluctant to accept sentences with scrambling across more than two levels of embedding. In order for the argument in Becker et al. (1992) to go through, this reluctance must be interpreted as a performance issue similarly to center embedding beyond two levels in English. But as Joshi et al. (2002) point out, it is equally possible to interpret that reluctance as indicating a restriction on speakers' competence, the property which formal grammars attempt to model. As they show, even tree-local MCTAG would be sufficient to handle scrambling in this case.

Against this uncertain empirical background, Chen-Main and Joshi (2007, 2008) compare a number of MCTAG variants based on which orderings of scrambled arguments they can derive, given certain linguistic assumptions on the shape of the elementary trees. These variants consist in extending tree-local MCTAG with various formal devices that were not discussed in this paper, specifically, flexible composition (Joshi and Kallmeyer 2003) and multiple adjoining (Schabes and Shieber 1994). Chiang and Scheffler (2008) show that extending tree-local MCTAG with (their formalization of) flexible composition does not increase its weak generative capacity. However, this does not entail membership in LCFRS in the sense of Becker et al. (1992) because the notion used there is not weak generative capacity but "derivational generative capacity", or the ability to derive sets of derivation structures. In the case of scrambled sentences, the structures in question are sentences in which the scrambled arguments are coindexed with their verbs. Further work may reveal whether the proof by Becker et al. (1992) extends to some of the more restricted languages generated by the extensions of MCTAG which Chen-Main and Joshi discuss.

Depending on the outcome of these investigations, we may find ourselves in the uncomfortable position where the only data that would discriminate between polynomial-time and NP-complete variants of TAG is unavailable for judgments because the sentences involved are too complex to process. In such a case, depending on which grammar formalisms we are willing to consider, the question whether natural language is polynomially parsable might very well turn out to be empirically untestable.

Looming in the background is the question of which grammar formalisms we allow into the competition in the first place. This question is itself thorny, since it cannot be dissociated from possibly subjective theoretical considerations. For example, one of the theoretically attractive properties that local variants of MCTAG share with TAG itself is that a domain of locality can be formulated over elementary trees that includes a verb and all its arguments, something which is not possible in context-free grammars because the VP node intervenes between subject and object arguments (Frank 2002; Joshi 2004a,b). With respect to scrambling and other long-distance dependencies, this extended domain of locality entails that constituents can only be displaced if they substitute or adjoin into the same elementary tree (or tree set) as if they were not displaced. In local extensions of TAG, the principle is a reflection of the locality constraint of the formalism itself. In contrast, nonlocal variants of TAG, such as the ones considered in this paper, are generally considered theoretically unattractive because locality constraints must be enforced by additional stipulations.

One example is V-TAG, which was briefly mentioned in Footnote 4. V-TAG is obtained from nonlocal MCTAG with dominance links when we no longer require members of a tree set to be introduced into the derivation simultaneously. It was proposed by Rambow (1994) to model scrambling precisely because it is polynomially parsable. However, unlike local TAG variants, V-TAG stipulates constraints on long-distance dependencies as integrity constraints, that is, node diacritics that act as barriers to movement by preventing dominance links from going through them. Despite its attractive parsing complexity, Kallmeyer (2005) rejects V-TAG along with other nonlocal MCTAG variants because locality constraints are not derived from the locality of the derivation operation. But adopting a TAG version whose locality constraint is too strict will wrongly rule out grammatical derivations (Kulick 2000). One set of examples are the scrambling orders discussed by Chen-Main and Joshi (2007, 2008) and mentioned above. Another example is extraction from weak islands, a phenomenon known as long movement (see Frank 2002 for discussion).

The question that determines whether grammar formalisms are considered theoretically attractive is whether the linguistic notion of locality is general and language-independent enough that it can be derived from abstract principles of the formalism, or so specific that it must be encoded by stipulations such as integrity constraints. These notions are arguably subjective to a certain extent. Unfortunately, they cannot be fully dissociated from the quest for a linguistically adequate and yet mildly context-sensitive formalism.

This should not be a reason for discouragement, though. Kallmeyer (2005) and Lichte (2007) propose TAG variants designed specifically to assign the right structural descriptions to German scrambling while maintaining a relaxed notion of locality in the formalism. The universal recognition problem for both these variants is NP-hard (Søgaard et al. 2007); most recently, however, Kallmeyer and Satta (2009) have shown that the fixed recognition problem for TT-MCTAG, the variant proposed in Lichte (2007), is polynomial. So the NP-completeness results presented in this paper are far from dashing the hope that some mildly context-sensitive variant of TAG will ultimately be found adequate for capturing the complexities of natural language.

Acknowledgments I am grateful to Joan Chen-Main, Laura Kallmeyer, Timm Lichte, Wolfgang Maier, Alexander Perekrestenko, Anders Søgaard, the Penn CLUNCH and XTAG groups, and to the audience and organizers of the 10th Mathematics of Language conference. I am especially grateful to Aravind K. Joshi for helpful discussion and continuous encouragement. Thanks to the University of Pennsylvania and to the Palo Alto Research Center (PARC) for financial support.

References

Becker, T., Joshi, A. K., & Rambow, O. (1991). Long-distance scrambling and Tree Adjoining Grammars. In *Proceedings of the 5th conference of the European chapter of the Association for Computational Linguistics*, Association for Computational Linguistics (pp. 21–26).

Becker, T., Rambow, O., & Niv, M. (1992). *The derivational power of formal systems, or scrambling is beyond LCFRS*. Technical report IRCS 92 38, Institute for Research in Cognitive Science, University of Pennsylvania, Philadelphia, PA, USA.

Boullier, P. (1998). A generalization of mildly context-sensitive formalisms. In *Proceedings of the 4th international workshop on Tree Adjoining Grammars and related frameworks (TAG+4)*. University of Pennsylvania, Philadelphia, PA, USA (pp. 17–20).

Chen-Main, J., & Joshi, A. K. (2007). Some observations on a "graphical" model-theoretical approach and generative models. In *ESSLLI 2007 workshop on model theoretic syntax*.

Chen-Main, J., & Joshi, A. K. (2008). Flexible composition, multiple adjoining and word order variation. In *Proceedings of the 9th international workshop on Tree Adjoining Grammars and related formalisms (TAG+9)*. Tübingen, Germany.

Chiang, D., & Scheffler, T. (2008). Flexible composition and delayed tree-locality. In *Proceedings of the 9th international workshop on Tree Adjoining Grammars and related formalisms (TAG+9)*. Tübingen, Germany.

Dahlhaus, E., & Warmuth, M. K. (1986). Membership for Growing context-sensitive grammars is polynomial. *Journal of Computer and System Sciences, 33*(3), 456–472.

Frank, R. (2002). *Phrase structure composition and syntactic dependencies*. Cambridge, MA, USA: MIT Press.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability*. San Francisco, CA, USA: W. H. Freeman and Company.

Greibach, S. A., & Hopcroft, J. E. (1969). Scattered context grammars. *Journal of Computer and System Sciences, 3*, 233–247.

Joshi, A., & Schabes, Y. (1997). Tree-adjoining grammars. In A. Salomaa & G. Rozenberg (Eds.), *Handbook of formal languages, vol 3: Beyond words* (pp. 69–123). Heidelberg, Germany: Springer.

Joshi, A. K. (1985). Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions?. In D. R. Dowty, L. Karttunen, & A. M. Zwicky (Eds.), *Natural language parsing: Psychological, computational, and theoretical perspectives*. Cambridge, UK: Cambridge University Press.

Joshi, A. K. (2004a). Domains of locality. *Data & Knowledge Engineering, 50*(3), 277–289.

Joshi, A. K. (2004b). Starting with complex primitives pays off: Complicate locally, simplify globally. *Cognitive Science, 28*(5), 637–668.

Joshi, A. K., & Kallmeyer, L. (2003). Flexible composition in LTAG: Quantifier scope and inverse linking. In H. Bunt & R. Muskens (Eds.), *Computing meaning, vol. 3*. Dordrecht: Kluwer.

Joshi, A. K., Levy, L. S., & Takahashi, M. (1975). Tree adjunct grammars. *Journal of Computer and System Sciences, 10*(1), 136–163.

Joshi, A. K., Vijay-Shanker, K., & Weir, D. (1991). The convergence of mildly context-sensitive grammatical formalisms. In P. Sells, S. Shieber, & T. Wasow (Eds.), *Foundational issues in natural language processing* (pp. 31–81). Cambridge, MA, USA: MIT Press.

Joshi, A. K., Becker, T., & Rambow, O. (2002). Complexity of scrambling: A new twist to the competence-performance distinction. In A. Abeillé & O. Rambow (Eds.), *Tree-adjoining grammars* (pp. 167–181). Stanford, CA, USA: CSLI Publications.

Kallmeyer, L. (2005). Tree-local multicomponent tree-adjoining grammars with shared nodes. *Computational Linguistics, 31*(2), 187–225.
Kallmeyer, L. (2009). A declarative characterization of different types of multicomponent Tree Adjoining Grammars. *Research on Language and Computation, 7*(1), 55–99.
Kallmeyer, L., & Satta, G. (2009). A polynomial-time parsing algorithm for TT-MCTAG. In *Proceedings of the 47th annual meeting of the Association for Computational Linguistics (ACL09)*, Singapore.
Kroch, A. S., & Joshi, A. K. (1987). Analyzing extraposition in a tree adjoining grammar. In G. Huck & A. Ojeda (Eds.), *Discontinuous constituency, syntax and semantics 20* (pp. 107–149). New York, NY, USA: Academic Press.
Kulick, S. (2000). *Constraining non-local dependencies in tree adjoining grammar: Computational and linguistic perspectives*. PhD thesis, University of Pennsylvania.
Lichte, T. (2007). An MCTAG with tuples for coherent constructions in German. In G. Penn (Ed.), *Proceedings of the Formal Grammar 2007 conference*. CSLI Online Publications.
Pollard, C. (1984). *Generalised context-free grammars, head grammars and natural language*. PhD thesis, Department of Linguistics, Stanford University, Palo Alto, CA, USA.
Rambow, O. (1994). *Formal and computational aspects of natural language syntax*. PhD thesis, University of Pennsylvania, Philadelphia, PA, USA.
Rambow, O., & Satta, G. (1992). Formal properties of non-locality. In *Proceedings of 1st international workshop on Tree Adjoining Grammars*. 1992.
Rambow, O., Vijay-Shanker, K., & Weir, D. J. (2001). D-Tree substitution grammars. *Computational Linguistics, 27*(1), 87–121.
Schabes, Y. (1990). *Mathematical and computational aspects of lexicalized grammars*. PhD thesis, University of Pennsylvania, Philadelphia, PA, USA.
Schabes, Y., & Shieber, S. (1994). An alternative conception of tree-adjoining derivation. *Computational Linguistics, 20*(1), 91–124.
Søgaard, A. (2009). Verifying context-sensitive treebanks and heuristic parses in polynomial time. In K. Jokinen & E. Bick (Eds.), *Proceedings of the 17th Nordic conference on computational linguistics (NODALIDA 2009)*.
Søgaard, A., Lichte, T., & Maier, W. (2007). The complexity of linguistically motivated extensions of Tree-Adjoining Grammar. In *Proceedings of Recent Advances in Natural Language Processing 2007*.
Steedman, M. (1988). Combinators and grammars. In R. Oehrle, E. Bach, & D. Wheeler (Eds.), *Categorial grammars and natural language structures*. Dordrecht: Reidel.
Vijay-Shanker, K., Weir, D. J., & Joshi, A. K. (1987). Characterizing structural descriptions produced by various grammatical formalisms. In *Proceedings of the 25th meeting of the Association for Computational Linguistics*.
Weir, D. J. (1988). *Characterizing mildly context-sensitive grammar formalisms*. PhD thesis, University of Pennsylvania, Philadelphia, PA, USA.