Lexicalized non-local MCTAG with dominance links is NP-complete[1]

Lucas Champollion
Dept. of Linguistics
University of Pennsylvania
USA
champoll@ling.upenn.edu

Abstract

An NP-hardness proof for nonlocal MCTAG by Rambow and Satta (1992), based on Dahlhaus and Warmuth (1986), is extended to some restrictions of that formalism. It is found that there are NP-hard grammars among nonlocal MCTAGs even if the following restrictions are imposed: every tree in every tree set has a lexical anchor; every tree set may contain at most two trees; in every such tree set, there is a dominance link between the foot node of one tree and the root node of the other tree and this dominance link must be obeyed in the derived tree. This is the version of MCTAG used in Becker, Joshi, and Rambow (1991). The lexicalization restriction makes the grammar class NP-complete.

1 Introduction

In trying to model the syntax of natural language within the Tree Adjoining Grammar (TAG) formalism (Joshi, Levy, and Takahashi, 1975), it has been found early on (Kroch and Joshi, 1987) that there are constructions in natural language which cannot be given the right structural descriptions using standard TAG. Various extensions to standard TAG have been investigated to answer the question of how much additional generative power is needed to describe natural language.

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), which include formalisms that are semilinear, allow only a limited number of crossing dependencies, and are parsable in polynomial time. Among the

[1]I am grateful to Joan Chen-Main, Laura Kallmeyer, Timm Lichte, Wolfgang Maier, Alexander Perekrestenko, the Penn CLUNCH and XTAG groups, and especially to Aravind K. Joshi for helpful discussion and encouragement.

TAG extensions investigated, a promising candidate for a linguistically adequate MCSG seemed to be set-local multicomponent TAG (MCTAG), which is more powerful than TAG but is still mildly context-sensitive. 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. 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 this requirement is dropped altogether, we obtain *non-local* MCTAG.) MCTAGs were first discussed by Joshi (1985) and later defined precisely by Weir (1988). For lack of space, this paper omits the formal definitions of MCTAGs; the reader is referred to Weir (1988).

In contrast, Becker, Joshi, and Rambow (1991) argue that long-distance scrambling in German is even beyond the power of LCFRS, a formalism which was introduced in Weir (1988) and which remains the best known formal characterization of the only roughly defined 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. This means that if one accepts Becker, Joshi, and Rambow (1991)'s argument, then set-local MCTAG, as well as a number of equivalent or less powerful formalisms such as head grammars (Pollard, 1984) and combinatory categorial grammars (Steedman, 1988) that can be classified as LCFRS (Joshi, Vijay-Shanker, and Weir, 1991), are no longer in the game.[2]

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.[3] This excludes, for example, nonlocal MCTAG, because there are nonlocal MCTAGs that generate languages for which the word recognition problem is NP-complete.

[2]However, there is some reason to believe that German scrambling is in fact more restricted than described in Becker, Joshi, and Rambow (1991) and that scrambling might not be beyond LCFRS after all (see section 6). For a polynomially parsable TAG variant claimed to assign the right structural descriptions to German scrambling, see Kallmeyer (2005b); cf. also Lichte (to appear).

[3]LCFRS do not include all languages that are polynomially parsable. 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)

This result is from Rambow and Satta (1992) and Rambow (1994) and is the basis for the work in this paper.

One of the first proposals to deal with the German scrambling data used nonlocal MCTAG with dominance links (MCTAG-DL) (Becker, Joshi, and Rambow, 1991). In this modification of nonlocal MCTAG, an additional requirement is added: in the final derived tree, 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.)[4]

MCTAG-DL have already been used by Kroch and Joshi (1987) for the analysis of extraposition in English. But unlike Becker, Joshi, and Rambow (1991), they impose the additional constraint of tree-locality. Dominance links in connection with tree-locality or set-locality can be simulated by choosing appropriate node labels (Kallmeyer, 2005a). Therefore, dominance links do not increase the generative power of the grammar in this case. For this reason, I am only interested in *nonlocal* MCTAG-DL in this paper.

While nonlocal multi-component rewriting systems tend to be NP-complete (see Rambow (1994), p. 62 for an overview), there are exceptions.[5] However, in this paper it is shown that the word recognition problem for MCTAG-DL is in fact NP-hard. Therefore, if as is generally assumed, $P \neq NP$, then MCTAG-DL is not a LCFRS and not mildly context-sensitive. The conjecture by Rambow (1994) that dominance links do not decrease the weak generative power of MCTAG is therefore confirmed. 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

[4]Under an alternative definition, dominance links are an optional feature that may or may not be present in the grammar. In that sense, every nonlocal MCTAG is a MCTAG-DL, and therefore MCTAG-DL is of course NP-hard. In this paper, however, I only consider MCTAG-DL in which dominance links are obligatorily present in each auxiliary set.

[5]An example of this is Rambow's nonlocal 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* (that block dominance links going through them) is polynomial and used by Rambow to model scrambling. However, unlike most other TAG variants it does not reduce island constraints to first principles, but stipulates them as integrity constraints; see Kallmeyer (2005b) for discussion.

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, Vijay-Shanker, and Weir, 2001)

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

2 Nonlocal MCTAG is NP-hard

This section presents a detailed proof of the NP-hardness of standard nonlocal MCTAG with adjunction constraints (MCTAG from now on). This is essentially the proof that was reported by Dahlhaus and Warmuth (1986) for scattered grammars. 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.

I now present a polynomial reduction from the NP-complete problem *3-Partition* to a specific 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 nonlocal MCTAG G_1 that has the property that

$\langle n_1, \ldots, n_{3k}, B\rangle$ is an 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 (see Figure 2), which is closely based on the growing scattered grammar G in Dahlhaus and Warmuth (1986), section 5. (The productions of G are displayed in Figure 2 as well.) 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, 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 $xa^{n_1}xa^{n_2}\ldots xa^{n_{3k}}yb^{m_1}ya^{m_2}\ldots yb^{m_k}$. In addition, all the strings it produces each contain an equal number of a's and b's, because each tree or 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 w.l.o.g. that $\beta_{create-triple}$ is always used as early as possible, all derivations allowed by G_1 follow the same general pattern:

`step 1` Initialize the derivation by α_{start}.

`step 2` Create k triples by using $\beta_{create-triple}$ as many times as needed.

`step 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.

`step 4` Optionally use $\beta_{fill-triple}$ to add an equal number of a's and b's to the left and right.

`step 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 `step 3` through `step 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, 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 unordered scattered grammar (USCG) (Dahlhaus and Warmuth, 1986). The productions of this USCG are reproduced in Figure 1, along with a sample derivation. A corresponding derivation is also available in G_1. For ease of reference, each rule is also reproduced in Figure 2 next to the tree that corresponds to it.

I now give the formal NP-hardness proof.[6] 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 saturated non-terminals, 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 $s_t \Rightarrow s$.[7] As usual, we write $\overset{*}{\Rightarrow}$ for

[6] From Dahlhaus and Warmuth (1986), with a few extensions.

[7] 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.

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

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

Figure 1: Sample derivation of the 3-partition instance: $\langle 4, 1, 3, 2, 5, 3; B = 9\rangle$ and productions of the USCG that corresponds to G_1.

the reflexive and transitive closure of $\Rightarrow$. Obviously, for all $w \in \Sigma^*$, G derives w iff $G \stackrel{*}{\Rightarrow} w$.

Clearly $G_1 \stackrel{*}{\Rightarrow} X(Y\hat{Y}\hat{Y})^k$. 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 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 (i.e. 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 $G_1 \stackrel{*}{\Rightarrow} w$, 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:

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

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 Figure 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 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. □

4 Restriction to lexicalized grammars

Here I modify the grammar G_1 to get a lexicalized grammar G_3 (see Figure 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 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. Lexicalized MCTAG with dominance links is NP-hard. □

5 NP-completeness

While the previous sections have shown that the languages G_1, G_2, and G_3 are NP-hard, it has not yet been established that they are NP-complete. This section accomplishes this by showing their membership in NP. All lexicalized MCTAGs are also shown NP-complete.

Note that the NP-hardness of some fixed member G_f of a grammar class $\mathscr{G}$ (say MCTAG) implies that the universal recognition problem for $\mathscr{G}$ (that is the problem of deciding for an *arbitrary* grammar $G \in \mathscr{G}$ and word w if $w \in L(G)$) is also NP-hard (because there is a trivial polynomial reduction from the fixed-recognition problem to the universal recognition problem). Therefore, the proof by Rambow and Satta (1992) presented above implies that the universal recognition problem for MCTAG is NP-hard. However, the same is not true for NP-completeness, because there is no guarantee that the universal recognition problem for $\mathscr{G}$ is in NP. It can not even be concluded from G_f being NP-complete that the word recognition problem for all members of $\mathscr{G}$ is at most NP-complete. This means that some fixed nonlocal MCTAGs, for example the ones that are used to model natural language syntax, might be exponential. So, it is important to stress that Rambow and Satta (1992)'s result does not mean that nonlocal MCTAGs as a class are at most NP-complete (neither in the sense of the universal recognition problem nor in the sense that each MCTAG grammar generates an at most NP-complete language), even though this is how the result is usually cited.

To prove that every MCTAG grammar is at most NP-complete, it would be necessary to show that a nondeterministic Turing machine can always guess the derivation of a word w in at most $|w|^k$ steps, for some fixed k. Call the subclass of MCTAG grammars for which this is the case *MCTAG-NP*. I leave open the question whether MCTAG-NP = MCTAG. I show now that G_1 and G_2 are MCTAG-NPs and therefore NP-complete.

Proof. It has been shown above that G_1 and G_2 are strongly equivalent, so the proof only needs to be carried out once. Every auxiliary tree set in G_1 except the unary set $\beta_{create-triple}$ intro-

duces 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 is at most NP-complete.

Proof. By definition, every derivation step introduces terminals. So it always takes at most $|w|$ steps to derive w. □

Corollary. G_3 is NP-complete. □

6 Conclusion and linguistic implications

Unless P=NP, lexicalized MCTAG with dominance links cannot be parsed in polynomial time and is therefore outside LCFRS. The conjecture by Rambow (1994) that dominance links do not decrease the weak generative power of MCTAG is therefore confirmed. The proposal by Becker, Joshi, and Rambow (1991) to model German scrambling by nonlocal MCTAG-DL is undermined.

However, there exist alternative views on the complexity of scrambling. Becker, Joshi, and Rambow (1991) had assumed that any number n of verbal arguments can be scrambled at once and that all scrambling orders are possible (the "double unboundedness" of Rambow (1994)). This is hard to check empirically, as sentences involving four or more scrambled arguments are usually very hard to judge. There are exceptions, though: Some special patterns (such as no permutation or an end-around permutation of arguments, for example) are much easier to judge positively for all n (Aravind Joshi, p.c.). Certain polynomially parsable TAG variants exist that do not derive all scrambling orderings for large n, but do derive these special patterns on which we do get clear empirical judgments (Chen-Main and Joshi, 2007).

Thus, it may be that the only data that would discriminate between polynomial-time and NP-complete variants of TAG grammar (if Rambow's V-TAG is disregarded – see fn.5) is unavailable for judgments. This suggests that depending on which grammar formalisms we allow into the competition, the question whether natural language is in P might turn out to be empirically untestable.

References

Becker, Tilman, Aravind K. Joshi, and Owen Rambow. 1991. Long-distance scrambling and Tree Adjoining Grammars. In *EACL*, pages 21–26.

Boullier, Pierre. 1998. A generalization of mildly context-sensitive formalisms. In *Proceedings of the fourth international workshop on Tree Adjoining Grammars and Related Frameworks (TAG+4)*, pages 17–20. University of Pennsylvania.

Chen-Main, Joan and Aravind K. Joshi. 2007. Some observations on a "graphical" model-theoretical approach and generative models. In *ESSLLI 2007 workshop on Model Theoretic Syntax*.

Dahlhaus and Warmuth. 1986. Membership for Growing Context-Sensitive Grammars is polynomial. *JCSS: Journal of Computer and System Sciences*, 33.

Garey, Michael R. and David S. Johnson. 1979. *Computers and Intractability*. W. H. Freeman and Company, San Francisco.

Joshi, Aravind K. 1985. Tree Adjoining Grammars: How much context-sensitivity is required to provide reasonable structural descriptions? In L. Karttunen D. R. Dowty and A. M. Zwicky, editors, *Natural Language Parsing*. Cambridge University Press, Cambridge.

Joshi, Aravind K., Leon S. Levy, and Masako Takahashi. 1975. Tree Adjunct Grammars. *Journal of Computer and System Sciences*, 10(1):136–163, February.

Joshi, Aravind K., K. Vijay-Shanker, and D. Weir. 1991. The convergence of mildly context-sensitive grammatical formalisms. In P. Sells, S. Shieber, and T. Wasow, editors, *Foundational Issues in Natural Language Processing*, pages 31–81, Cambridge, MA. MIT Press.

Kallmeyer, Laura. 2005a. A descriptive characterization of Multicomponent Tree Adjoining Grammars. Presentation to XTAG meeting, 13 October 2005.

Kallmeyer, Laura. 2005b. Tree-local Multicomponent Tree-Adjoining Grammars with Shared Nodes. *Computational Linguistics*, 31(2):187–225.

Kroch, Anthony S. and Aravind K. Joshi. 1987. Analyzing extraposition in a Tree Adjoining Grammar. In Geoffrey Huck and Almerindo Ojeda, editors, *Discontinuous constituency*, Syntax and semantics 20. Academic Press, New York, pages 107–149.

Lichte, Timm. to appear. An MCTAG with tuples for coherent constructions in German. In Gerald Penn, editor, *Proceedings of the Formal Grammar 2007 Conference*. CSLI Online Publications.

Pollard, C. 1984. *Generalised Context-free Grammars, Head Grammars and natural language*. Ph.D. thesis, Department of Linguistics, Stanford University, Palo Alto, CA.

Rambow, Owen and Giorgio Satta. 1992. Formal properties of non-locality. In *1^{st} Int. Workshop on Tree Adjoining Grammars. 1992.*

Rambow, Owen, K. Vijay-Shanker, and David J. Weir. 2001. D-Tree Substitution Grammars. *Computational Linguistics*, 27(1):87–121.

Rambow, Owen C. 1994. *Formal and computational aspects of natural language syntax*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA, USA.

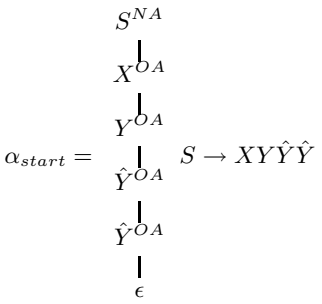
Schabes, Yves. 1990. *Mathematical and computational aspects of lexicalized grammars*. Ph.D. thesis, University of Pennsylvania.

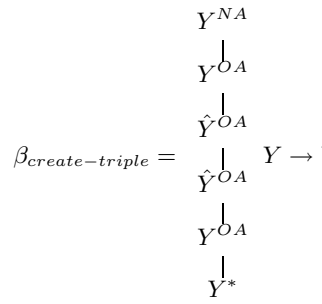
Steedman, Mark. 1988. Combinators and grammars. In R. Oehrle, E. Bach, and D. Wheeler, editors, *Categorial Grammars and Natural Language Structures*. Dordrecht: Reidel.

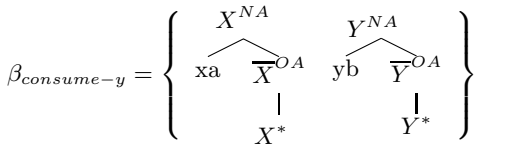
Weir, David Jeremy. 1988. *Characterizing mildly context-sensitive grammar formalisms*. Ph.D. thesis, University of Pennsylvania.

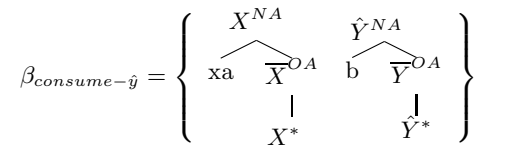
$G_2 = (NT, \Sigma, S, I, A)$ where

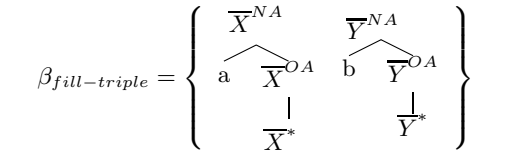
$$\begin{aligned} NT &= \{X, \overline{X}, Y, \overline{Y}, \hat{Y}\} \\ \Sigma &= \{a, b, x, y\} \\ I &= \{\alpha_{start}\} \\ A &= \{\beta_{create-triple}, \beta_{consume-y}, \beta_{consume-\hat{y}}, \beta_{fill-triple}, \beta_{close-triple}, \beta_{end}\} \end{aligned}$$

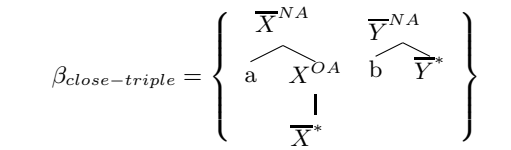
$\alpha_{start} =$  $S \rightarrow XY\hat{Y}\hat{Y}$

$\beta_{create-triple} =$  $Y \rightarrow Y\hat{Y}\hat{Y}Y$

$\beta_{consume-y} =$  $X \rightarrow xa\overline{X}, Y \rightarrow yb\overline{Y}$

$\beta_{consume-\hat{y}} =$  $X \rightarrow xa\overline{X}, \hat{Y} \rightarrow b\overline{Y}$

$\beta_{fill-triple} =$  $\overline{X} \rightarrow a\overline{X}, \overline{Y} \rightarrow b\overline{Y}$

$\beta_{close-triple} =$  $\overline{X} \rightarrow aX, \overline{Y} \rightarrow b$

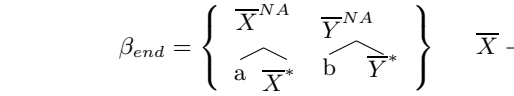
$\beta_{end} =$  $\overline{X} \rightarrow a, \overline{Y} \rightarrow b$

Figure 2: The MCTAG G_1 with its corresponding USCG rules.

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

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

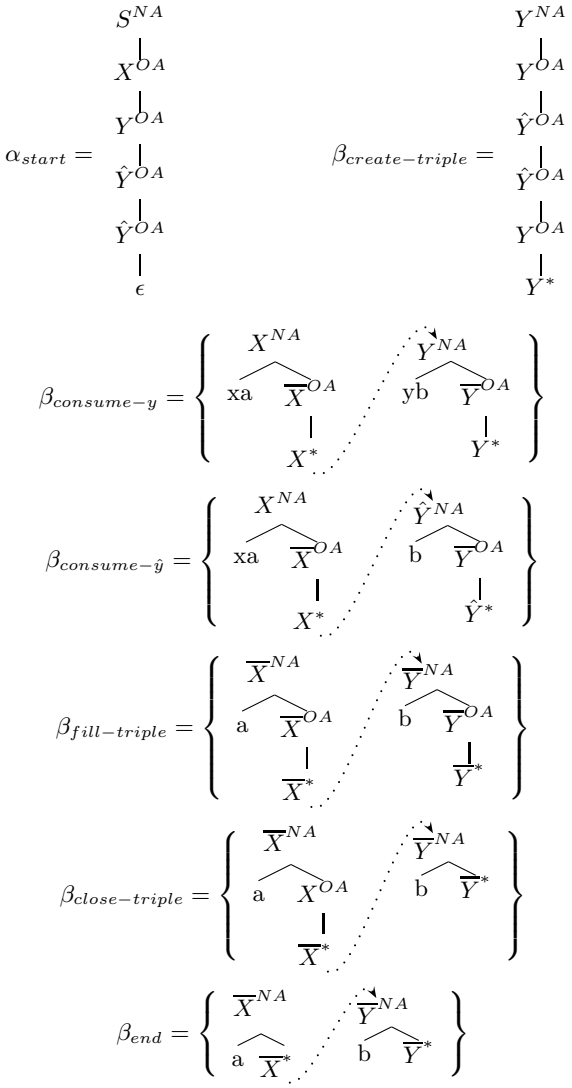
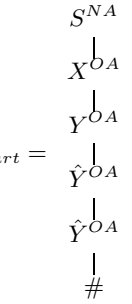


Figure 3: The MCTAG with dominance links G_2. (Identical to G_1 except for the dominance links.)

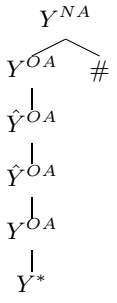
$G_3 = (NT, \Sigma, S, I, A)$ where

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

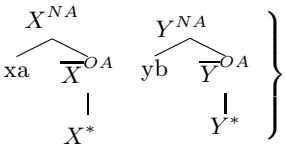
$\alpha_{start} =$



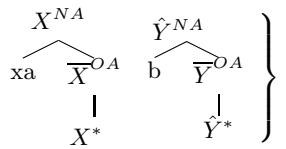
$\beta_{create-triple} =$



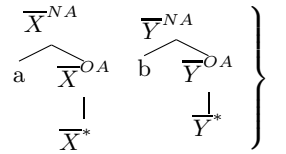
$\beta_{consume-y} =$



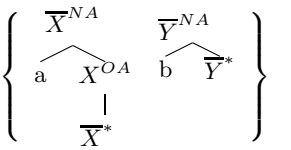
$\beta_{consume-\hat{y}} =$



$\beta_{fill-triple} =$



$\beta_{close-triple} =$



$\beta_{end} =$

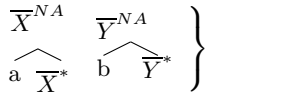


Figure 4: The lexicalized MCTAG G_3. (Identical to G_1 except that new terminals have been added to α_{start} and to $\beta_{create-triple}$.)