A Declarative Characterization of Different Types of Multicomponent Tree Adjoining Grammars

Laura Kallmeyer

SFB 441, University of Tübingen,
Nauklerstr: 35, D-72074 Tübingen, Germany.

Abstract. Multicomponent Tree Adjoining Grammars (MCTAG) is a formalism that has been shown to be useful for many natural language applications. The definition of MCTAG however is problematic since it refers to the process of the derivation itself: a simultaneity constraint must be respected concerning the way the members of the elementary tree sets are added. This way of characterizing MCTAG does not allow to abstract away from the concrete order of derivation. In this paper, we propose an alternative definition of MCTAG that characterizes the trees in the tree language of an MCTAG via the properties of the derivation trees (in the underlying TAG) the MCTAG licences. This definition gives a better understanding of the formalism, it allows a more systematic comparison of different types of MCTAG, and, furthermore, it can be exploited for parsing.

1 TAG and MCTAG

Tree Adjoining Grammar (TAG, [Joshi and Schabes, 1997]) is a tree-rewriting formalism. A TAG consists of a finite set of trees (elementary trees). The nodes of these trees are labelled with nonterminals and terminals (terminals only label leaf nodes). Starting from the elementary trees, larger trees are derived by substitution (replacing a leaf with a new tree) and adjunction (replacing an internal node with a new tree). In case of an adjunction, the tree being adjoined has exactly one leaf that is marked as the foot node (marked with an asterisk). Such a tree is called an *auxiliary* tree. When adjoining it to a node n, in the resulting tree, the subtree with root n from the old tree is attached to the foot node of the auxiliary tree. Non-auxiliary elementary trees are called *initial* trees. A derivation starts with an initial tree. In a final derived tree, all leaves must have terminal labels. For a sample derivation see Fig. 1.

Definition 1 (Tree Adjoining Grammar)
A Tree Adjoining Grammar (TAG) is a tuple $G = \langle I, A, N, T\rangle$ *with*

- N *and* T *being disjoint finite sets, the nonterminals and terminals*
- I *being a finite set of initial trees with nonterminals* N *and terminals* T*, and*
- A *being a finite set of auxiliary trees with nonterminals* N *and terminals* T*.*

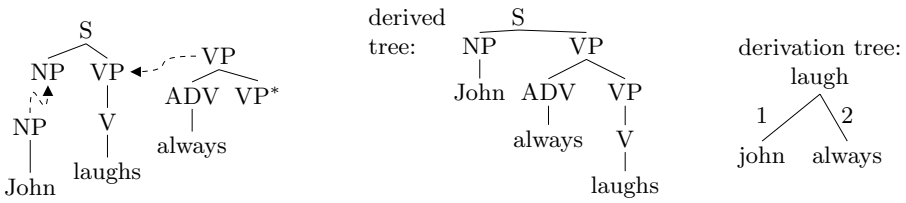


Fig. 1. TAG derivation for *J*ohn always laughs

Definition 2 (TAG derivation and tree language) *Let $G = \langle I, A, N, T\rangle$ be a TAG. Let γ and γ' be finite trees.*

- *$\gamma \Rightarrow \gamma'$ in G iff there is a node position p and a γ'_0 that is either elementary or derived from some elemenentary tree such that $\gamma' = \gamma[p, \gamma'_0]$.*[1]
  $\stackrel{*}{\Rightarrow}$ is the reflexive transitive closure of $\Rightarrow$.
- *The tree language of G is $L_T(G) := \{\gamma \mid$ there is an $\alpha \in I$ such that $\alpha \stackrel{*}{\Rightarrow} \gamma$ and all leaves in γ have terminal labels$\}$.*

TAG derivations are represented by derivation trees that record the history of how the elementary trees are put together. A derived tree is the result of carrying out the substitutions and adjunctions, i.e., the derivation tree describes uniquely the derived tree. Each edge in a derivation tree stands for an adjunction or a substitution. The edges are labelled with Gorn addresses.[2] E.g., the derivation tree in Fig. 1 indicates that the elementary tree for *John* is substituted for the node at address 1 and *always* is adjoined at node address 2.

Definition 3 (TAG derivation tree) *Let $G = \langle I, A, N, T\rangle$ be a TAG. Let γ be a tree derived as follows in G:*

$\gamma = \gamma_0[p_1, \gamma_1] \ldots [p_k, \gamma_k]$ where γ_0 is an instance of an elementary tree and the substitutions/adjunctions of the $\gamma_1, \ldots, \gamma_k$ are all the substitutions/adjunctions to γ_0 that are performed to derive γ.

Then the corresponding derivation tree has a root labelled with γ_0 that has k daugthers. The edges from γ_0 to these daughters are labelled with $p_1, \ldots, p_k$, and the daughters are the derivation trees for the derivations of $\gamma_1, \ldots, \gamma_k$.

A TAG extension that is useful for linguistic applications is multicomponent TAG (MCTAG, [Weir, 1988]). An MCTAG contains sets of elementary trees. In each derivation step, one of the tree sets is chosen and its trees are added simultaneously. Depending on the nodes to which the trees from the set attach, different kinds of MCTAGs are distinguished: if the nodes are required to be part of the same elementary tree, the MCTAG is *tree-local*, if they are required to be part of the same tree set, the grammar is *set-local* and otherwise it is *non-local*.

[1] For trees $\gamma, \gamma_1, \ldots, \gamma_n$ and pairwise different node positions $p_1, \ldots, p_n$ in γ, $\gamma[p_1, \gamma_1] \ldots [p_n, \gamma_n]$ denotes the result of subsequently substituting/adjoining the $\gamma_1, \ldots, \gamma_n$ to the nodes in γ with addresses $p_1, \ldots, p_n$ respectively.

[2] The root address is ϵ, and the jth child of a node with address p has address pj.

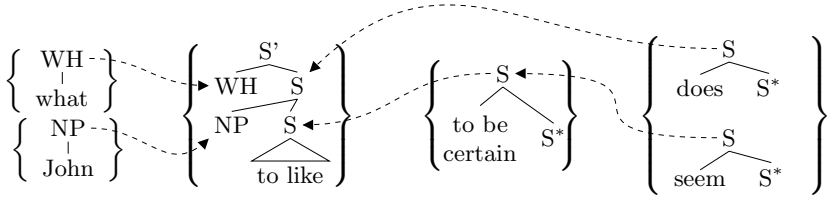


Fig. 2. MCTAG derivation of *what does John seem to be certain to like*

Definition 4 (MCTAG) *A multicomponent TAG (MCTAG) is a tuple* $G = \langle I, A, N, T, \mathcal{A}\rangle$ *such that:*

- $G_{TAG} := \langle I, A, N, T\rangle$ *is a TAG, and*
- $\mathcal{A}$ *is a partition of* $I \cup A$*, the set of elementary tree sets.*

$\gamma \Rightarrow \gamma'$ *is a multicomponent derivation step in* G *iff there is a* $\{\gamma_1, \ldots, \gamma_k\} \in \mathcal{A}$ *and there are pairwise different node addresses* $p_1, \ldots, p_k$ *in* γ *such that* $\gamma' = \gamma[p_1, \gamma_1] \ldots [p_k, \gamma_k]$.

A derivation starts from some $\alpha \in I$ being in a unary set. In the final derived tree, all leaves must be labelled by terminals.

In this paper, in section 2, we argue that the simultaneity requirement of the standard MCTAG definition is problematic and we propose an alternative declarative definition for non-local MCTAG. Section 3 extends this to different types of MCTAG, section 4 briefly sketches a parsing algorithm based on the declarative characterization from section 2, and section 5 concludes.

2 A declarative characterization of MCTAG

To illustrate the idea consider the derivation in Fig. 2. The *to be certain* tree adjoins to the lower S of the *like* tree, the WH and NP nodes of *like* are substituted for *what* and *John* respectively, and the trees for *does* and *seem* are adjoined simultaneously to the upper S node of *like* and the root node of *to be certain*. These last adjunctions cannot be performed before having added *to be certain* to *like*, otherwise the simultaneity requirement cannot be satisfied. Simultaneity imposes certain derivation orders even though different orders might lead to the same adjunctions and substitutions: E.g., one might as well start by adding *does* to *like* (at the higher S), then adjoin *to be certain* to *like* (at the lower S) and then adjoin *seem* to *certain*. This yields the same adjunctions and substitutions and the same derived tree. But the simultaneity requirement is not respected.

In contrast to this, in a TAG it is sufficient to check whether there is a derivation tree that yields the tree in question; one can abstract away from the order of the derivations steps. E.g., the derivation order in Fig. 1 does not matter for the resulting derivation tree and derived tree.

For MCTAG as well one would like to abstract away from derivation orders that do not yield different substitutions and adjunctions. One way to achieve this

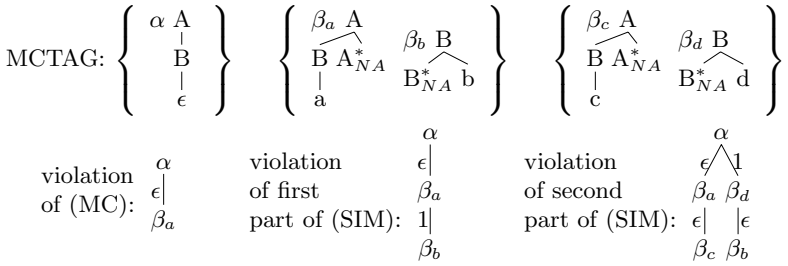
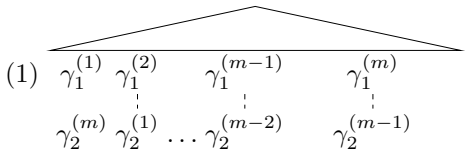


Fig. 3. Sample derivations not licensed in a non-local MCTAG

is to consider an MCTAG as a TAG G where the elementary trees are grouped into pairwise disjoint sets. Because of this grouping, only some of the derivation trees in G are licensed by the MCTAG. Namely those that satisfy the constraints following from the multicomponent sets and the simultaneity requirement. This is the idea we will pursue in this paper.

Each MCTAG derivation step is a sequence of substitutions and/or adjunctions. Consequently, each derivation in an MCTAG G with $G = \langle I, A, N, T, \mathcal{A} \rangle$ corresponds to a derivation in the underlying TAG $G_{TAG} := \langle I, A, N, T \rangle$. Let us define the *TAG derivation tree* of such a multicomponent derivation as the corresponding derivation tree in G_{TAG}. We can then define different variants of MCTAG by putting different constraints on this derivation tree.

In general, the TAG derivation trees for MCTAG derivations must have three properties (see Fig. 3 for sample derivation trees not satisfying these properties): Firstly, if an instance of an elementary tree set is used, then all trees from this set must occur in the derivation tree. This is the multicomponent condition that we will call (MC). Secondly, one tree from an elementary tree set cannot be substituted or adjoined into another tree from the same set. This is the first part of the simultaneity condition (SIM). Thirdly, two tree sets cannot be interleaved, i.e., they must be added one after the other. Consequently, derivation trees as (1) are not allowed with $\gamma_1^{(i)}$ and $\gamma_2^{(i)}$ being in the same tree set Γ_i because (1) indicates that there is a cycle in the order of adding the tree sets: Γ_2 must be added before Γ_1, Γ_3 before Γ_2 etc., Γ_m before Γ_{m-1} and Γ_1 before Γ_m. By transitive closure, Γ_1 must be added before Γ_m and Γ_m before Γ_1. This is the second part of the condition (SIM).



For each derivation tree D, let $\mathcal{P}_D$ be the parent relation and $\mathcal{D}_D$ the dominance relation on the nodes in D. In other words, for two nodes n_1, n_2 in D, $\langle n_1, n_2 \rangle \in \mathcal{P}_D$ means that n_1 is the mother of n_2 and $\langle n_1, n_2 \rangle \in \mathcal{D}_D$ means that n_1 dominates n_2.

Lemma 1 *Let $G = \langle I, A, N, T, \mathcal{A}\rangle$ be an MCTAG, $G_{TAG} := \langle I, A, N, T\rangle$. Let D be a derivation tree in G_{TAG} with the corresponding derived tree t being in $L(G_{TAG})$. Let L be the set of node labels in D.*

D is a possible TAG derivation tree in G with $t \in L(G)$ iff

- **(MC)** *There are k pairwise disjoint instances $\Gamma_1, \ldots, \Gamma_k$ of elementary tree sets ($k \geq 1$) such that $L = \bigcup_{i=1}^{k} \Gamma_i$.*
- **(SIM)** *D satisfies the following:*
  - *(a) For all Γ_i, $1 \leq i \leq k$ and for all $\gamma_1, \gamma_2 \in \Gamma_i$ being labels of nodes n_1, n_2 in D: if $n_1 \neq n_2$, then $\langle n_1, n_2\rangle \notin \mathcal{D}_D$.*
  - *(b) For all pairwise different $\Gamma'_1, \Gamma'_2, \ldots, \Gamma'_m \in \{\Gamma_1, \ldots, \Gamma_k\}$ ($m \geq 2$) with $N_1, \ldots, N_m$ being the sets of nodes labelled with elements from $\Gamma'_1, \Gamma'_2, \ldots, \Gamma'_m$ respectively:*
    There are no $n_1^{(i)}, n_2^{(i)} \in N_i$ ($1 \leq i \leq m$) such that $\langle n_1^{(1)}, n_2^{(m)}\rangle \in \mathcal{D}_D$ and $\langle n_1^{(i)}, n_2^{(i-1)}\rangle \in \mathcal{D}_D$ for $2 \leq i \leq m$.

The proof is given in [Kallmeyer, 2005]. The lemma gives us a way to characterize non-local MCTAG via the properties of the TAG derivation trees the grammar licenses.

3 Different types of MCTAG

We can define tree-local and set-local derivations by imposing further conditions.

Definition 5 *Let $G = \langle I, A, N, T, \mathcal{A}\rangle$ be an MCTAG. Let D be a TAG derivation tree for some $t \in L(\langle I, A, N, T\rangle)$.*

- *D is tree-local iff*
  (TL) *for all nodes $n_1, \ldots, n_m$ in D, $m > 1$, with labels from the same tree set instance: there is a node n_0 such that $\langle n_0, n_1\rangle, \ldots \langle n_0, n_m\rangle \in \mathcal{P}_D$.*
- *D is set-local iff*
  (SL) *for all nodes $n_1, \ldots, n_m$ in D, $m > 1$, with labels from the same elementary tree set instance Γ: there are $n'_1, \ldots, n'_m$ with labels from another elementary tree set instance Γ' such that $\langle n'_i, n_i\rangle \in \mathcal{P}_D$ for $1 \leq i \leq m$.*

Obviously, an MCTAG is *tree-local*/*set-local* iff the TAG derivation trees licensed by it satisfy (MC), (SIM) and (TL)/(SL). Actually, if (TL) or (SL) is required, (SIM) is not needed since it holds anyway:

Lemma 2 *Let $G = \langle I, A, N, T, \mathcal{A}\rangle$ be an MCTAG, D a TAG derivation tree for a $t \in L(\langle I, A, N, T\rangle)$ satisfying (MC). If D satisfies (SL) then it satisfies (SIM).*

The proof is given in the appendix. Note that a derivation tree that satisfies even (TL) trivially satisfies (SL) and therefore (SIM).

Other restrictions sometimes imposed on MCTAG are so-called *dominance links* (e.g., non-local MCTAG with dominance links [Becker et al., 1991] and

(2) ... dass [es]$_1$ der Mechaniker [t_1 zu reparieren] verspricht
... that it the mechanic to repair promises
'... that the mechanic promises to repair it'

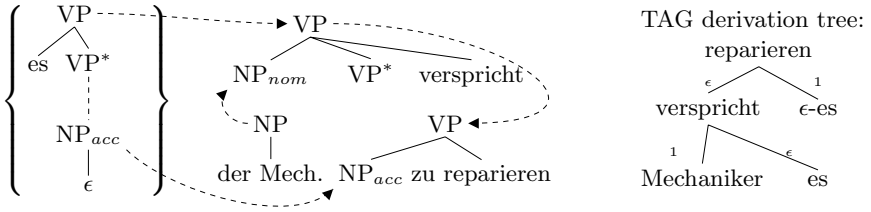


Fig. 4. MCTAG with dominance constraints: derivation of (2)

Vector MCTAG with Dominance Links, V-TAG, [Rambow, 1994]). A dominance link is a constraint $f \geq n$ where f is a foot node in some β, n an internal node in some γ, and β and γ belong to the same elementary tree set. It indicates that in the final derived tree, the node f (or the root of anything adjoined to f) must dominate n (or anything adjoined to n). For an example see Fig. 4. *Es* and its trace are in the same tree set. The dominance link between them signifies that in the derived tree *es* must c-command the trace. The crucial property of the TAG derivation tree is that *es* is adjoined to the spine[3] of *verspricht* which is adjoined to *reparieren* at a node that dominates the substitution site of the trace ϵ-*es*. Dominance constraints are restrictions on the derived trees. We can formulate a corresponding constraint (Dom) for the possible TAG derivation trees.

We first define a relation $\mathcal{SD}_D$ of *spine-dominance* on the nodes of a derivation tree D: A node n_1 spine-dominates a node n_2 (notation $\langle n_1, n_2 \rangle \in \mathcal{SD}_D$) if all edge labels on the path that links them are spine addresses.[4]

For each chain of adjunctions γ_1 adjoining γ_0, γ_2 to γ_1, γ_3 to γ_2 etc. up to some γ_n, in the derived tree the following holds: the foot node of γ_n dominates a node k from γ_0 iff this node is below the node γ_1 adjoins to and γ_1 spine-dominates γ_n. This is the following lemma (see the appendix for the proof).

Lemma 3 *Let D be a TAG derivation tree, $n_0, \ldots, n_m$ nodes in D with labels $\gamma_0, \ldots, \gamma_m$. Let $\langle n_{i-1}, n_i \rangle \in \mathcal{P}_D$ with edge label p_i for all $1 \leq i \leq m$. Let γ_m be an auxiliary tree such that nothing adjoins to the foot node f of γ_m. Furthermore, let p be a node address in γ_0 pointing at a node k.*

In the corresponding derived tree the foot node f of γ_m dominates k (or anything adjoined or substituted at k) iff $\langle n_1, n_m \rangle \in \mathcal{SD}_D$ and p_1 prefix of p.

Now we define (Dom). For $f \geq k$ with f foot node of β and k node in γ, (Dom) distinguishes three caes: (a) if k is part of a γ' that is attached to β, then γ' must be adjoined to f in order to satisfy the dominance link. (b) and (c) (see

[3] The spine is the path from the root to the foot node.

[4] A spine address is a node address on the spine of an auxiliary tree, i.e., a prefix of the foot node address.

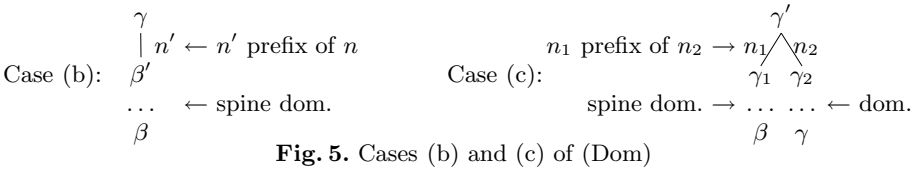


Fig. 5. Cases (b) and (c) of (Dom)

Fig. 5) correpond to the configuration of Lemma 3. With (b), if $\langle \gamma, \beta \rangle \in \mathcal{D}_D$ then the foot node f of β dominates the foot node of β', and the foot node of β' dominates n. With (c), if $\langle \gamma, \beta \rangle \notin \mathcal{D}_D$ then f dominates the foot node of γ_1 that dominates everything attached to k_2.

Definition 6 *Let $G = \langle I, A, N, T, \mathcal{A} \rangle$ be an MCTAG, Dom_G a set of dominance links for G. Let D be a TAG derivation tree for some $t \in L(\langle I, A, N, T \rangle)$ that satisfies (MC). D respects the dominance links in Dom_G iff the following holds:*

(Dom) *for each pair β, γ from the same elementary tree set instance with β labelling n_β, γ labelling n_γ in D such that there is a dominance constraint $f \geq k$ with f foot node of β, k an internal node in γ:*

(a) if $\langle n_\beta, n_\gamma \rangle \in \mathcal{D}_D$, then there is a node n with $\langle n_\beta, n \rangle \in \mathcal{P}_D$ with the foot node position in β as edge label, and $\langle n, n_\gamma \rangle \in \mathcal{D}_D$

(b) if $\langle n_\gamma, n_\beta \rangle \in \mathcal{D}_D$, then there is a β' labelling $n_{\beta'}$ and a node address k' in γ that is a prefix of k such that $\langle n_\gamma, n_{\beta'} \rangle \in \mathcal{P}_D$ with edge label k' and $n_{\beta'}$ spine-dominates n_β, and

(c) if $\langle n_\gamma, n_\beta \rangle \notin \mathcal{D}_D$ and $n_{\gamma'}$ (with label γ') is the lowest node dominating both in D, then there are $n_{\gamma_1}, n_{\gamma_2}$ and positions k_1, k_2 in γ' such that k_1 prefix of k_2 with $\langle n_{\gamma'}, n_{\gamma_1} \rangle \in \mathcal{P}_D$ with edge label k_1, $\langle n_\gamma, n_{\gamma_2} \rangle \in \mathcal{P}_D$ with edge label k_2 and $\langle n_{\gamma_1}, n_\beta \rangle \in \mathcal{SD}_D$ and $\langle n_{\gamma_2}, n_\gamma \rangle \in \mathcal{D}_D$.

In MCTAG with dominance links the TAG derivation trees satisfy (MC), (SIM) and (Dom) while in V-TAG they satisfy only (MC) and (Dom). (With Lemma 3, this is immediate.) Fig. 6 summarizes the different formalisms.[5]

Local MCTAG variants:

	(TL)	(SL)
(MC)	tree-local MCTAG	set-local MCTAG
(MC), (SIM)	tree-local MCTAG	set-local MCTAG

Non-local MCTAG variants:

	(Dom)	–
(MC)	Vector MCTAG with Dominance Links (V-TAG)	Vector MCTAG
(MC), (SIM)	non-local MCTAG with dominance links	non-local MCTAG

Fig. 6. Summary of different MCTAG variants

[5] Vector-MCTAG ([Rambow, 1994]) are like non-local MCTAG but without the simultaneity requirement.

4 Parsing

The proposed declarative definition of non-local MCTAG can be exploited for parsing. It allows us to consider the underlying TAG derivation tree and check whether this derivation tree satisfies (MC) and (SIM). In other words, parsing can be done in two steps: 1. parsing in the underlying TAG G_{TAG} and 2. check of (MC) and (SIM) for all derivation trees obtained in the first step. For reasons of space, we will only briefly sketch this idea.

Concerning the parsing in G_{TAG}, the difficulty is that G_{TAG} is not lexicalized: In the MCTAG there might be sets of elementary trees with only some trees containing lexical items, other trees containing the empty word or only nodes with non-terminal symbols. However, each of these non-lexicalized trees occurs only in combination with a lexicalized tree. Therefore, we can still guarantee that the set of elementary tree instances used for parsing a sentence is limited to $c \cdot n$ where c is a constant and n is the length of the input string. Consequently, polynomial parsing with G_{TAG} is possible. We assume that the output is a derivation foest, i.e., a compact representation of the derivation trees.

Once we have this derivation forest, in order to determine the MCTAG derivations, we have to extract the single derivation trees while deciding which elementary tree instances belong to the same set instance. In general, this cannot be done in polynomial time. The question whether an additional condition on the TAG derivation trees can be checked in polynomial time therefore depends closely on the question whether, for checking this condition, one needs to compute the grouping of tree instances into set instances.

In order to check (MC), we only need to make sure that all trees from a $\Gamma \in \mathcal{A}$ occur the same number of times. It is actually possible to check (MC) in polynomial time. Among others, this tells us that lexicalized Vector MCTAG are polynomially parsable, a result already conjectured in [Rambow, 1994].

For (SIM), in contrast to (MC), we have to group the tree instances into set instances. This can be done building the different derivation trees while traversing the derivation forest in the order of a simultaneous multicomponent derivation. This way both, (SIM) and (MC) are checked in once. I.e., we construct entire derivation trees by simulating the MCTAG derivations on the derivation forest. The number of possible derivation trees can explode since it includes the computation of the instances of elementary tree sets. This explosion of the number of possible multicomponent derivation trees was to be expected since the parsing of non-local MCTAG (i.e., MCTAG satisfying (MC) and (SIM)) is NP-complete, even in the lexicalized case [Rambow and Satta, 1992,Champollion, 2006].[6]

To summarize, based on our declarative MCTAG definition, parsing of lexicalized MCTAG could be separated into TAG-parsing of the underlying TAG followed by checking of the constraints (MC) and (SIM). As long as (SIM) is not considered, parsing is polynomial.

[6] Note, however, that this explosion is only the worst case. In a lexicalized MCTAG it depends on the number of times a terminal occurs in the input. In natural languages it is rather rare that words (except some functional operators) occur more than once or twice in a sentence.

5 Conclusion

TAG derivation trees abstract away from the concrete order of derivation steps. A similar abstraction is not possible with the classical MCTAG definition since the simultaneity constraint refers to the process of the derivation itself. In this paper, we propose an alternative declarative definition of MCTAG that characterizes the trees in the tree language via the properties of the TAG derivation trees the MCTAG licences. In this way, in MCTAG like in TAG, the TAG derivation tree can be considered being the central structure of the formalism and the desired abstraction can be obtained. This declarative MCTAG definition enables us to see more clearly the differences between classical MCTAG and variants of them where some of the constraints for the TAG derivation trees need not hold while others are added. Furthermore, this way of defining MCTAG can also be exploited for parsing: The parsing of the underlying TAG can be separated from the constraint checking on the derivation forest given by the TAG parser.

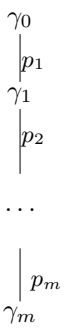
Appendix: Proofs

Proof of Lemma 2: Assume that D as in Lemma 2 satisifes (SL). We assume that D does not satisfy (SIM) and show that this yields a contradiction. If D does not satisfy (SIM), then

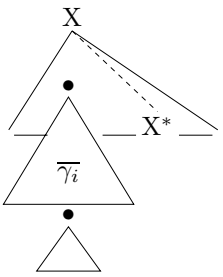
- either there are nodes n_0, n_0' in D, $n_0 \neq n_0'$ with labels from the same tree set instance and with $\langle n_0, n_0' \rangle \in \mathcal{D}_\mathcal{D}$.
  Then with (SL), there must be n_1 and n_1' with $\langle n_1, n_0 \rangle, \langle n_1', n_0' \rangle \in \mathcal{P}_\mathcal{D}$ (and therefore $n_1 \neq n_1'$) and $\langle n_1, n_1' \rangle \in \mathcal{D}_\mathcal{D}$ such that n_1 and n_1' have labels from the same tree set instance.
  By induction, with (SL), for all $i \geq 0$ there must be n_{i+1} and n_{i+1}' with $\langle n_{i+1}, n_i \rangle, \langle n_{i+1}', n_i' \rangle \in \mathcal{P}_\mathcal{D}$, $\langle n_{i+1}, n_{i+1}' \rangle \in \mathcal{D}_\mathcal{D}$ and $n_{i+1} \neq n_{i+1}'$ such that n_{i+1} and n_{i+1}' have labels from the same tree set instance.
  Contradiction since the set of nodes in D is finite.
- or there are pairwise disjoint sets of nodes $N_1, \ldots N_m$ ($m \geq 2$) in D with labels from the elementary tree set instances $\Gamma_1, \Gamma_2, \ldots, \Gamma_m$ respectively such that: There are $n_1^{(i)}, n_2^{(i)} \in N_i$ ($1 \leq i \leq m$) such that $\langle n_1^{(1)}, n_2^{(m)} \rangle \in \mathcal{D}_D$ and $\langle n_1^{(i)}, n_2^{(i-1)} \rangle \in \mathcal{D}_D$ for $2 \leq i \leq m$.
  Because of (SL), we can show: whenever there are two different set instances Γ and Γ' that are labels of the node sets N and N' respectively such that there are $n \in N, n' \in N'$ with $\langle n, n' \rangle \in \mathcal{D}_\mathcal{D}$, then for all $n' \in N'$ there is a $n \in N$ with $\langle n, n' \rangle \in \mathcal{D}_\mathcal{D}$. (This follows immediately from (SL) by induction on the length of the path from n to n'.)
  For the $N_1, \ldots N_m$ we consequently obtain: for each $n_m \in N_m$ there is a $n_1 \in N_1$ with $\langle n_1, n_m \rangle \in \mathcal{D}_\mathcal{D}$ and (by induction) for each $n_1 \in N_1$ there is a $n_m \in N_m$ with $\langle n_m, n_1 \rangle \in \mathcal{D}_\mathcal{D}$. Contradiction since (with $N_1 \cap N_m = \emptyset$) this would imply that dominance is not antisymmetric.

□

Proof of Lemma 3: The configuration in D is as shown on the right.



First we show the $\Leftarrow$ part of the iff:
Assume $\langle n_1, n_m \rangle \in \mathcal{SD}_D$ and p_1 a prefix of p.
Then by induction the auxiliary tree $\overline{\gamma_1}$ derived from γ_1 is such that the foot node f of γ_m either dominates the foot node of $\overline{\gamma_1}$ or it is even the foot node of $\overline{\gamma_1}$ (in this last case, all $p_i, 1 \leq i \leq m$ are foot node positions). In both cases, adjoining $\overline{\gamma_1}$ to γ_0 at position p_1 (p_1 prefix of p, the address of k) leads to a tree where f dominates k.
Now we show the $\Rightarrow$ direction of the iff:
Assume that the left side of the iff holds. Then assume that the right side does not hold and show that this gives a contradiction.
First assume that p_1 is not a prefix of p. In this case
(i) either p prefix of p_1 and $p \neq p_1 \Rightarrow k$ strictly dominates anything adjoined to position p_1 in γ_1, including the foot node f of γ_m. Contradiction.
(ii) or p no prefix of $p_1 \Rightarrow$ (since p_1 no prefix of p either) no dominance relation between k and the material adjoined to position p_1 in γ_1. Contradiction.
Then assume $\langle n_1, n_m \rangle \notin \mathcal{SD}_D$. $\Rightarrow$ there is a $i \in \{1, \ldots, m\}$ such that p_i is not a spine address. Consequently, the auxiliary tree derived from γ_{i-1} is either as shown on the right or symmetric to this but with the foot node on the left of $\overline{\gamma_i}$, the tree derived from γ_i.
Adjoining this tree to a tree $\overline{\gamma_0}$ derived from γ_0 by the first $i-1$ adjunctions yields a tree where no node from $\overline{\gamma_i}$ dominates a node from $\overline{\gamma_0}$. In particular, f does not dominate k.



□

References

[Becker et al., 1991] Becker, T., Joshi, A. K., and Rambow, O. (1991). Long-distance scrambling and tree adjoining grammars. In *Proceedings of ACL-Europe*.

[Champollion, 2006] Champollion, L. (2006). Lexicalized non-local mctag with dominance links is np-complete. Ms, University of Pennsylvania.

[Joshi and Schabes, 1997] Joshi, A. K. and Schabes, Y. (1997). Tree-Adjoning Grammars. In Rozenberg, G. and Salomaa, A., editors, *Handbook of Formal Languages*, pages 69–123. Springer, Berlin.

[Kallmeyer, 2005] Kallmeyer, L. (2005). Tree-local multicomponent tree adjoining grammars with shared nodes. *Computational Linguistics*, 31(2):187–225.

[Rambow, 1994] Rambow, O. (1994). *Formal and Computational Aspects of Natural Language Syntax*. PhD thesis, University of Pennsylvania.

[Rambow and Satta, 1992] Rambow, O. and Satta, G. (1992). Formal Properties of Non-Locality. In *Proceedings of the TAG+ Workshop*, Philadelphia.

[Telljohann et al., 2003] Telljohann, H., Hinrichs, E. W., and Kübler, S. (2003). *Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z)*. Seminar für Sprachwissenschaft, Universität Tübingen, Germany.

[Weir, 1988] Weir, D. J. (1988). *Characterizing mildly context-sensitive grammar formalisms*. PhD thesis, University of Pennsylvania.