A Descriptive Characterization of Multicomponent Tree Adjoining Grammars

Laura Kallmeyer
TALaNa/Lattice, UFRL, University Paris 7
2 place Jussieu, Case 7003, 75005 Paris
`laura.kallmeyer@linguist.jussieu.fr`

Mots-clefs : Grammaires d'Arbres Adjoints, MCTAG, formalismes grammaticaux

Keywords: Tree Adjoining Grammars, MCTAG, grammar formalisms

Résumé Il a été montré que les Grammaires d'Arbres Adjoints Ensemblistes (Multicomponent Tree Adjoining Grammars, MCTAG) sont très utiles pour des applications TAL. Pourtant, la définition des MCTAG est problématique parce qu'elle fait référence au procès de dérivation même : une contrainte de simultanéité est imposée concernant la façon dont on ajoute les membres d'un même ensemble d'arbres. En regardant uniquement le résultat d'une dérivation, c'est-à-dire l'arbre dérivé et l'arbre de dérivation, cette simultanéité n'est plus visible. Par conséquent pour vérifier la contrainte de simultanéité, il faut toujours considérer l'ordre concret des pas de la dérivation. Afin d'éviter cela, nous proposons une caractérisation alternative de MCTAG qui permet une abstraction de l'ordre de dérivation : Les arbres générés par la grammaire sont caractérisés par les propriétés de leurs arbres de dérivation.

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. Looking only at the result of a derivation (i.e., the derived tree and the derivation tree), this simultaneity is no longer visible and therefore cannot be checked. I.e., this way of characterizing MCTAG does not allow to abstract away from the concrete order of derivation. Therefore, 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 the MCTAG licences.

1 Introduction

1.1 Tree Adjoining Grammars

Tree Adjoining Grammar (TAG, Joshi et al., 1975) is a tree-rewriting formalism. A TAG consists of a finite set of trees (*elementary* trees) with nonterminals and terminals as node labels (terminals only label leaf nodes). Starting from the elementary trees, larger trees are derived by

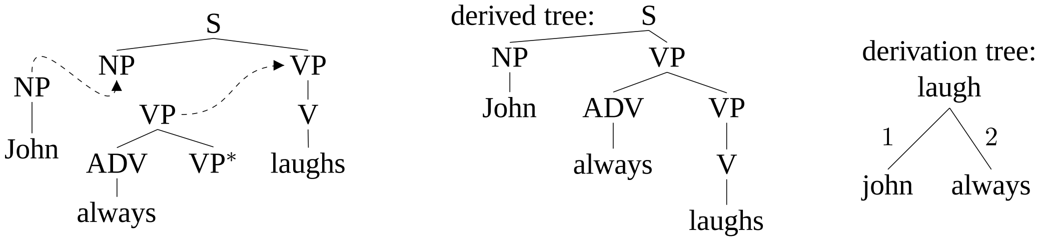

Figure 1: TAG derivation for *John always laughs*

substitution (replacing a leaf with a new tree) and *adjunction* (replacing an internal node with a new tree). In case of an adjunction, the new tree is a so-called *auxiliary* tree that has exactly one leaf marked as the foot node (marked with an asterisk). All other elementary trees are called *initial* trees. When adjoining an auxiliary tree β to a node μ, in the resulting tree, the subtree with root node μ from the old tree is put below the foot node of β. Each derivation starts with an initial tree. In the final derived tree, all leaves must have terminal labels. See for example Fig. 1 : Starting from the *laughs* tree, the tree for *John* is substituted for the NP leaf and the tree for *always* is adjoined at the VP node.

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. Each edge in the derivation tree stands for an adjunction or a substitution. The edges are labelled with Gorn addresses of the nodes where the substitutions/adjunctions take place.[1] E.g., in Fig. 1 the derivation tree indicates that the elementary tree for *John* is substituted for the node at address 1 and *always* is adjoined at node address 2.

1.2 Multicomponent TAG

Multicomponent TAG (MCTAG, Joshi, 1987; Weir, 1988) is a TAG extension useful for linguistic applications. An MCTAG contains sets of elementary trees. Starting with an initial tree, in each derivation step, all trees from one of the tree sets are added simultaneously. Depending on the nodes to which these trees attach, different kinds of MCTAGs are distinguished: if all nodes are required to be part of the same elementary tree, the MCTAG is *tree-local*; if all nodes are required to be part of the same tree set, the grammar is *set-local*; otherwise the grammar is *non-local*.[2] Consider for example the non-local MCTAG derivation in Fig. 2: the tree for *to be certain* adjoins to the lower S node of *like*, the WH and NP nodes of *like* are substituted for *what* and *John* respectively, and *does* and *seem* are adjoined simultaneously to the upper S node of *like* and the root node of *to be certain* respectively. These last two operations cannot be performed before having added *to be certain* to *like,* otherwise the simultaneity requirement is not satisfied.

Intuitively, the requirement of adding all elements of an elementary set simultaneously is easy to understand and this definition of MCTAG seems very clear. However, the simultaneity requirement imposes certain derivation orders even though a different order might lead to the same adjunctions and substitutions and to the same derived tree. E.g., in Fig. 2 one might as well

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

[2] Cases where MCTAGs have been argued to be useful are extractions out of complex NPs as in "which painting did you buy a copy of" where the two parts of the complex NP should be part of one elementary structure but cannot be part of the same elementary tree. For such examples Kroch and Joshi (1987) propose to use tree-local MCTAGs.

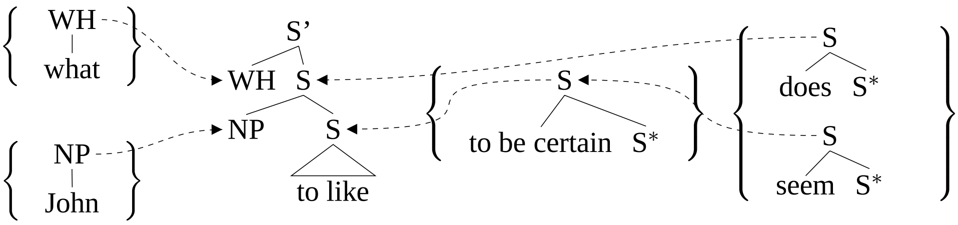

Figure 2: Derivation for *what does John seem to be certain to like* in an MCTAG G_M

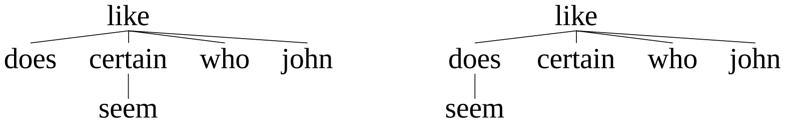

Figure 3: TAG derivation trees for the TAG underlying Fig. 2

start by adding *does* to *like* (at the higher S node), then adjoin *to be certain* to *like* (at the lower S node) and then adjoin *seem* to *to be certain*. This yields the same derived tree with the same adjunctions and substitutions. But the simultaneity requirement is not respected. Consequently, in order to check whether a given tree is part of the tree language, one has to check the possible derivations of this tree including the different derivation orders. In contrast to this, in a TAG it is sufficient to check whether there is a derivation tree yielding the tree in question. I.e., one can abstract away from the order of the derivations steps. E.g., in Fig. 1, no matter in which order *John* and *always* are added, the derivation tree and consequently the derived tree are the same.

For MCTAG as well one would like to abstract away from differences with respect to derivation order that do not make any difference concerning the substitutions and adjunctions that are performed. One way to achieve this is to consider an MCTAG as a TAG G with additional multicomponent tree sets (sets of initial and auxiliary trees from G) where certain derivation trees in G are disallowed since they do not satisfy certain constraints. E.g., the derivation trees in Fig. 3 are both possible in a TAG with the elementary trees from the MCTAG G_M in Fig. 2. The first derivation tree is the one for the derivation from Fig.2. Since we know that only *does* and *seem* are in one set and since *does* and *seem* are dominated by different daughters of *like* (namely *does* and *certain* respectively), this is a possible TAG derivation tree in G_M. The second derivation tree is possible in the underlying TAG but not in G_M: since *seem* adjoins into *does*, it is not possible to add *does* and *seem* simultaneously to different nodes in an already derived tree. With this characterization of MCTAG one gets rid of the problematic simultaneity requirement. Instead, one characterizes in a descriptive way the properties of the derivation trees licensed by the grammar. The advantage of this non-operational perspective is that one needs not to check all possible derivation orders with respect to the simultaneity constraint.

In section 2, standard definitions of TAG and MCTAG are given. Then, in section 3, an alternative descriptive characterization of MCTAG is proposed.

2 Standard definitions of TAG and MCTAG

We assume that the definitions of initial and auxiliary trees and the definitions of substitution and adjunction are already known.[3] a TAG (see, e.g., Vijay-Shanker, 1987) is a tuple $G =$

[3] For formal definitions of initial and auxiliary trees with certain alphabets of nonterminal and terminal symbols and also for formal definitions of the operations substitution and adjunction see for example Kallmeyer (1999).

$\langle I, A, N, T\rangle$ with N abd T being finite sets of nonterminals and terminals, and I and A being finites sets of initial and auxiliary trees with nonterminals N and terminals T.

In a TAG $G = \langle I, A, N, T\rangle$, a *derivation step* is defined as follows: Let γ and γ' be finite trees. $\gamma \Rightarrow \gamma'$ in G iff there is a node position p and a tree $\gamma_0 \in I \cup A$[4] such that $\gamma' = \gamma[p, \gamma_0]$.[5] $\overset{*}{\Rightarrow}$ is the reflexive transitive closure of $\Rightarrow$. The *tree language* of G is then $L_T(G) := \{\gamma \mid$ there is an $\alpha \in I$ such that $\alpha \overset{*}{\Rightarrow} \gamma$ and all leaves in γ have terminal labels$\}$.

Each node address p in a derived tree points at a node belonging to some elementary tree γ_e. In γ_e this node has some address p_e. In the following we assume that the address p in a derivation step $\gamma \Rightarrow \gamma'$ of the node where the adjunction/substitution takes place is the corresponding tuple $\langle p_e, \gamma_e\rangle$. This is possible since each node in a derived tree in TAG belongs uniquely to one of the elementary trees used in the course of the derivation. E.g., the address of the ADV node in the derived tree in Fig. 1 is $\langle 1, always\rangle$. Using these addresses we can define derivation trees: A *derivation tree* is a tuple $\langle \mathcal{N}, \mathcal{E}\rangle$ of nodes and edges. $\mathcal{N}$ is a finite set of instances of elementary trees and $\mathcal{E} \subset \mathcal{N} \times \mathcal{N} \times \mathbb{N}^*$ where $\mathbb{N}^*$ is the set of Gorn addresses. (The edges are directed from the mother node to the daughter.)[6] For a TAG $G = \langle I, A, N, T\rangle$ and a derivation $\gamma_0 \Rightarrow \gamma_1 \Rightarrow \gamma_2 \cdots \Rightarrow \gamma_n$ in G, the derivation tree $\langle \mathcal{N}, \mathcal{E}\rangle$ is then as follows: $\gamma_0 \in \mathcal{N}$, and for all derivation steps $\gamma_i \Rightarrow \gamma_{i+1}$, $0 \leq i < n$ in the derivation such that there is a node position $\langle p_e, \gamma_e\rangle$ and a tree $\gamma \in I \cup A$ with $\gamma_{i+1} = \gamma_i[\langle p_e, \gamma_e\rangle, \gamma]$: $\gamma \in \mathcal{N}$ and $\langle \gamma_e, \gamma, p_e\rangle \in \mathcal{E}$. These are all nodes and edges. In a derivation tree $D = \langle \mathcal{N}, \mathcal{E}\rangle$, the *parent* relation is the relation between mothers and daughters, $\mathcal{P}_D := \{\langle n_1, n_2\rangle \mid$ there is a $p \in \mathbb{N}^*$ such that $\langle n_1, n_2, p\rangle \in \mathcal{E}\}$. The *dominance* relation is its reflexive transitive closure, $\mathcal{D}_D := \{\langle n_1, n_2\rangle \mid n_1, n_2 \in \mathcal{N}$ and either $n_1 = n_2$ or there is a n_3 such that $\langle n_1, n_3\rangle \in \mathcal{P}_D$ and $\langle n_3, n_2\rangle \in \mathcal{D}_D\}$.

Finally, we define multicomponent TAG (MCTAG, Joshi, 1987; Weir, 1988): 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 $\mathcal{A} \subseteq P(I \cup A)$ is a set of subsets of $I \cup A$, the set of elementary tree sets.[7] $\gamma \Rightarrow \gamma'$ is a *multicomponent derivation step* in G iff there is an instance $\{\gamma_1, \ldots, \gamma_n\}$ of an elementary tree set in $\mathcal{A}$ and there are pairwise different node addresses $p_1, \ldots, p_n$ such that $\gamma' = \gamma[p_1, \gamma_1] \ldots [p_n, \gamma_n]$ where $\gamma[p_1, \gamma_1] \ldots [p_n, \gamma_n]$ is the result of adding the γ_i ($1 \leq i \leq n$) at node positions p_i in γ. As in TAG, a derivation starts from an initial tree and in the final derived tree, all leaves must be labelled by terminals.

In each MCTAG derivation step, the trees from a new elementary tree set are added to the already derived tree. Since they are added to pairwise different nodes, one can as well add them one after the other, i.e., each multicomponent derivation in an MCTAG $G = \langle I, A, N, T, \mathcal{A}\rangle$ corresponds to a derivation in the 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}.[8]

[4] To be precise, this must be an occurrence of an elementary tree. Henceforth, whenever we use an elementary tree in a derivation we actually mean an occurrence of this elementary tree.

[5] As usual, we use the following notations for substitution and adjunction. For trees γ and γ' and for node positions p, $\gamma[p, \gamma']$ is defined as follows: If γ' is (derived from) an initial tree with root label $X \in N$ and the node at position p in γ is a substitution node with label X, then $\gamma[p, \gamma']$ is the tree one obtains by substitution of γ' into γ at node position p. If γ' is (derived from) an auxiliary tree with root label $X \in N$and if the node at position p in γ is an internal node with label X, then $\gamma[p, \gamma']$ is the tree one obtains by adjunction of γ' to γ at node position p. Otherwise $\gamma[p, \gamma']$ is undefined.

[6] Linear precedence is not needed in a derivation tree since it does not influence the result of the derivation.

[7] $P(X)$ is the set of subsets of some set X.

[8] This TAG derivation tree is not the MCTAG derivation tree defined in Weir (1988). The nodes of Weir's MCTAG derivation trees are labelled by sequences of elementary trees (i.e., by elementary tree sets) and each edge stands for simultaneous adjunctions/substitutions of all elements of such a set.

3 A descriptive characterization of MCTAG

The TAG derivation trees for MCTAG derivations have certain properties resulting from the requirement that the elements of elementary tree sets must be added simultaneously: Firstly, if an elementary tree set is used, then all trees from this set must occur in the derivation tree. Secondly, one tree from an elementary tree set cannot be substituted or adjoined into another tree from the same set. Thirdly, different tree sets cannot be interleaved. More concretely there cannot be n tree sets such a tree from the first is added to a tree from the second, a tree from the second to a tree from the third etc. (which amounts to adding first the nth tree set, then the $(n-1)$th etc.), while at the same time a tree from the nth set is added to a tree from the first set. For non-local MCTAG, these are all constraints the TAG derivation tree needs to satisfy.

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

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

- **(MC1)** *The root of D is an instance of an initial tree $\alpha \in I$ and all other nodes are instances of trees from tree sets in $\mathcal{A}$ such that for all instances Γ of elementary tree sets from $\mathcal{A}$ and for all $\gamma_1, \gamma_2 \in \Gamma$: if $\gamma_1 \in \mathcal{N}$, then $\gamma_2 \in \mathcal{N}$.*
- **(MC2)** *For all instances Γ of elementary tree sets from $\mathcal{A}$ and for all $\gamma_1, \gamma_2 \in \Gamma$, $\gamma_1 \neq \gamma_2$: $\langle \gamma_1, \gamma_2\rangle \notin \mathcal{D}_D$.*
- **(MC3)** *For all pairwise different instances $\Gamma_1, \Gamma_2, \dots, \Gamma_n$, $n \geq 2$ of elementary tree sets from $\mathcal{A}$: there are no $\gamma_1^{(i)}, \gamma_2^{(i)} \in \Gamma_i$, $1 \leq i \leq n$ such that $\langle \gamma_1^{(1)}, \gamma_2^{(n)}\rangle \in \mathcal{D}_D$ and $\langle \gamma_1^{(i)}, \gamma_2^{(i-1)}\rangle \in \mathcal{D}_D$ for $2 \leq i \leq n$.*

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 and thereby to get rid of the original simultaneity requirement: The corresponding properties are now captured in the three constraints (MC1)–(MC3). Since these constraints need to hold only for the TAG derivation trees that correspond to derived trees in the tree language, sub-derivation trees need not satisfy them. In other words, γ_1 and γ_2 from the same tree set can be added at different moments of the derivation as long as the final TAG derivation tree satisfies (MC1)–(MC3).

We can now define tree-local and set-local TAG derivation trees by imposing further conditions: Let $G = \langle I, A, N, T, \mathcal{A}\rangle$ be an MCTAG. Let $D = \langle \mathcal{N}, \mathcal{E}\rangle$ be a TAG derivation tree for some $t \in L(\langle I, A, N, T\rangle)$. D is a *multicomponent* derivation tree iff it satisfies (MC1)–(MC3). D is *tree-local* iff for all instances $\{\gamma_1, \dots, \gamma_n\}$ of elementary tree sets with $\gamma_1, \dots, \gamma_n \in \mathcal{N}$: there is one γ such that $\langle \gamma, \gamma_1\rangle, \dots \langle \gamma, \gamma_n\rangle \in \mathcal{P}_D$. D is *set-local* iff for all instances $\{\gamma_1, \dots, \gamma_n\}$ of elementary tree sets with $\gamma_1, \dots, \gamma_n \in \mathcal{N}$: there is an instance Γ of an elementary tree set such that for all $1 \leq i \leq n$ there is a $t_i \in \Gamma$ with $\langle t_i, \gamma_i\rangle \in \mathcal{P}_D$.

The following lemma is immediate.

Lemma 2 *Let G be an MCTAG.*

- *G is a* tree-local MCTAG *iff the set of trees generated by G, $L_T(G)$, is defined as the set of those trees that can be derived with a tree-local multicomponent TAG derivation tree in G.*
- *G is a* set-local MCTAG *iff the set of trees generated by G, $L_T(G)$, is defined as the set of those trees that can be derived with a set-local multicomponent TAG derivation tree in G.*

4 Conclusion

MCTAG is an extension of TAG that has been shown to be useful for many natural language applications. Therefore a profound understanding of the mathematical properties of the formalism is indispensable. In a TAG, the central structure of a derivation, the derivation tree abstracts away from the order of derivation steps as long as the result of the derivation is the same: in the derivation tree, the adjunction/substitution operations corresponding to different daughters of the same node can be performed in any order without influencing the derived tree one obtains. Consequently, the derivation trees are unordered with respect to linear precedence.

This way of abstracting away from the concrete order of derivation steps is not possible with the classical MCTAG definition. The definition 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. Looking only at the result a derivation (i.e., the derived tree and the derivation tree), this simultaneity is no longer visible and therefore cannot be checked. I.e., this way of characterizing MCTAG does not allow to abstract away from the concrete order of derivation. Therefore, 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 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.

Apart from the fact that this descriptive characterization of MCTAG helps to understand the mathematical properties of the grammar formalism, it probably also has an impact on parsing. Parsing can be done independently from concrete derivations since the simultaneity constraint need not be checked. Only the outcoming derivation trees need to be checked for well-formedness in the sense of (MC1)–(MC3). However, we do not pursue this further here and we leave the subject for future research.

Références

Joshi, A. K.: 1987. An introduction to Tree Adjoining Grammars. *in* A. Manaster-Ramer (ed.), *Mathematics of Language*. John Benjamins. Amsterdam. pp. 87–114.

Joshi, A. K., Levy, L. S. and Takahashi, M. 1975. Tree Adjunct Grammars. *Journal of Computer and System Science* **10**, 136–163.

Kallmeyer, L.: 1999. *Tree Description Grammars and Underspecified Representations*. PhD thesis. Universität Tübingen. Technical Report IRCS-99-08 at the Institute for Research in Cognitive Science, Philadelphia.

Kallmeyer, L. 2005. Tree-local multicomponent tree adjoining grammars with shared nodes. *Computational Linguistics*. To appear.

Kroch, A. S. and Joshi, A. K.: 1987. Analyzing extraposition in a tree adjoining grammar. *in* G. J. Huck and A. E. Ojeda (eds), *Syntax and Semantics: Discontinuous Constituency*. Academic Press, Inc.. pp. 107–149.

Vijay-Shanker, K.: 1987. *A Study of Tree Adjoining Grammars*. PhD thesis. University of Pennsylvania.

Weir, D. J.: 1988. *Characterizing mildly context-sensitive grammar formalisms*. PhD thesis. University of Pennsylvania.