On Multicomponent TAG Parsing*

Pierre Boullier

INRIA-Rocquencourt
Domaine de Voluceau
B.P. 105
78153 Le Chesnay Cedex, France
E-mail: Pierre.Boullier@inria.fr

Résumé

The notion of mild context-sensitivity is an attempt to express the formal power needed to define the syntax of natural languages. However, all incarnations of mildly context-sensitive formalisms are not equivalent. On the one hand, near the bottom of the hierarchy, we find tree adjoining grammars and, on the other hand, near the top of the hierarchy, we find multicomponent tree adjoining grammars. This paper proposes a polynomial parse time method for multicomponent tree adjoining grammars. This method uses range concatenation grammars as a high-level intermediate definition formalism. We show some upper bounds on the parse time of the set-local version of multicomponent tree adjoining grammar, and we introduce a hierarchy of restricted forms which can be parsed more efficiently. Our approach aims at giving both a new insight into the multicomponent adjunction mechanism and at providing a practical implementation schema.

1. Introduction

The notion of mild context-sensitivity is an attempt (see [Joshi, 1985] and [Weir, 1988]) to express the formal power needed to define the syntax of natural languages. However, all incarnations of mildly context-sensitive formalisms are not equivalent. On the one hand, near the bottom of the hierarchy, we find tree adjoining grammars (TAGs) [Vijay-Shanker, 1987] and some other weakly equivalent formalisms, linear indexed grammars [Gazdar, 1985], head grammars [Pollard, 1984] and combinatory categorial grammars [Steedman, 1987]) (see [Vijay-Shanker and Weir, 1994] for a proof of their equivalence). On the other hand, near the top of the hierarchy, we find multicomponent tree adjoining grammars (MC-TAGs) which are, in turn, equivalent to linear context-free rewriting systems (LCFRS) [Vijay-Shanker, Weir, and Joshi, 1987] and multiple context-free grammars (M-CFGs) [Seki, Matsumura, Fujii, and Kasami, 1991]. This paper proposes a polynomial parse time method for MC-TAGs, which uses range concatenation grammars (RCGs) as a high-level intermediate definition formalism.

*An extended version of this paper is in [Boullier, 1999a]

The concept of RCG is introduced in [Boullier, 1998]; it is a syntactic formalism which is a variant of the simple version of literal movement grammars, described in [Groenink, 1997], and which is also related to the framework of LFP developed by [Rounds, 1988]. This formalism is more powerful than LCFRS,[1] while staying computationally tractable: its sentences can be parsed in polynomial time. A minimal introduction to RCGs is in [Boullier, 1999c], figuring in these proceedings, and will not be repeated here. In [Boullier, 1999c], we show how any TAG, with unrestricted adjunction constraints, can be transformed into an equivalent RCG which can be parsed in $\mathcal{O}(n^6)$ time at worst. We assume that the reader is familiar with both the TAG formalism and with the first transformation algorithm of [Boullier, 1999c]. In this paper, we apply to MC-TAGs a generalization of this first method. This approach results in a polynomial upper bound for the parse time of unrestricted MC-TAGs. This bound can be improved if we restrict MC-TAGs to specific subclasses. Moreover, our approach gives a new insight into the multicomponent adjunction mechanism while providing a practical implementation schema.

2. Multicomponent TAG

An extension of TAGs was introduced in [Joshi, Levy, and Takahashi, 1975] and later refined in [Joshi, 1987] where the adjunction operation involves a set of auxiliary trees instead of a single auxiliary tree. In a MC-TAG, the elementary structures, both initial and auxiliary, instead of being two sets of single trees, consist of two finite sets of finite tree sets. In MC-TAGs, the adjunction operation of an auxiliary tree set is defined as the simultaneous adjunction of each of its component trees and accounts for a single step in the derivation process. This multicomponent adjunction (MCA) operation is defined as follows. All the trees of an auxiliary tree set can be adjoined into distinct nodes (addresses) in a single elementary (initial or auxiliary) tree set.[2] Of course, if the cardinality of each tree set is one, a MC-TAG is a TAG. If the maximum cardinality of the initial tree sets is k, we have a k-MC-TAG. In [Weir, 1988], the author has shown that the languages defined by MC-TAGs are equal to the languages defined by LCFRS. The equivalence also holds with M-CFLs.

We can think of two types of *locality* for MCAs, one type, named *tree locality*, requires that all trees in an auxiliary tree set adjoin to a unique tree of an elementary tree set; the other type, named *set locality*, requires that all trees in an auxiliary tree set adjoin to the same elementary tree set, not necessarily to a unique tree and not necessarily to all the trees in this elementary tree set. We choose to cover the set-local interpretation of the term since it is more general than the tree-local version and is the one equivalent to LCFRS.

Without loss of generality, we will prohibit (multicomponent) substitution in MC-TAGs, assuming that this operation can always be simulated by a MCA. In TAGs, the adjunction constraints can be defined by the adj function which gives, for every node η, the set of auxiliary trees which can be adjoined at η (for an optional adjunction, we have $nil \in adj(\eta)$). In MC-TAGs, for an elementary tree set, the choice of all possible MCAs will be defined by means of functions – the set of adjunction covers – introduced below.

If $\mathcal{N}_\tau$ denotes the set of nodes of an elementary tree set τ, in order for a MCA to take place, we first have to identify its *site*, the subset of $\mathcal{N}_\tau$ at which this MCA will occur. The identification of all sites first results in a partition $\{\mathcal{N}_1, \ldots, \mathcal{N}_j, \ldots, \mathcal{N}_m\}$ of $\mathcal{N}_\tau$ in which each

[1] In [Boullier, 1999b], we argue how this extra power can be used in natural language processing.

[2] If the MCA operation is allowed into derived tree sets instead of elementary tree sets, we have a *nonlocal* MC-TAG. This version of MC-TAGs has not yet been studied in details, and it is not known whether nonlocal MC-TAGs are polynomially parsable (see [Becker, Joshi and Rambow, 1991]).

site $\mathcal{N}_j$ is the set of nodes at which a single MCA of some tree set τ' can occur. Of course, if τ' is some auxiliary tree set β_i, we have $|\mathcal{N}_j| = |\beta_i|$ and if $\tau' = \{nil\}$, we have $|\mathcal{N}_j| = 1$. Second, for each node $\eta \in \mathcal{N}_j$, we must know which tree of β_i can be adjoined at η. Thus we assume that for each site $\mathcal{N}_j$, there is a bijective mapping ξ_j from $\mathcal{N}_j$ to β_i called *local adjunction cover*. We also define a function ξ, called *adjunction cover*, from $\mathcal{N}_\tau$ to $\cup_{\beta_i \in \mathcal{A}} \beta_i \cup \{nil\}$ s.t. $\forall j, 1 \leq j \leq m$, the restriction of ξ to $\mathcal{N}_j$ is the local adjunction cover ξ_j. For any MC-TAG, we assume that all MCA constraints of a given elementary tree set can be expressed by a finite set of adjunction covers.[3] Thus, associated with each elementary tree set τ, we assume that there is a finite set of adjunction covers. Each such adjunction cover ξ defines, on the one hand m local adjunction covers $\xi_1, \ldots, \xi_j, \ldots, \xi_m$, and, on the other hand, an m-partition $\Pi_\tau^\xi = \{\mathcal{N}_1, \ldots, \mathcal{N}_j, \ldots, \mathcal{N}_m\}$ of the adjunction nodes $\mathcal{N}_\tau$ of τ, s.t. each site $\mathcal{N}_j$ is the definition domain $dom(\xi_j)$ of ξ_j, and s.t. for each node $\eta \in \mathcal{N}_\tau$, $\xi(\eta) \in \cup_{\beta_i \in \mathcal{A}} \beta_i \cup \{nil\}$ is the tree that is adjoined at η.

Now, we are ready to show that for any set-local k-MC-TAG there is an equivalent simple $2k$-PRCG.

3. Set-Local MC-TAG to Simple PRCG

The transformation of a set-local MC-TAG into an equivalent simple PRCG is based upon a generalization of the first transformation algorithm from TAG to simple PRCG proposed in [Boullier, 1999c].

Without loss of generality, we assume that initial tree sets are singletons whose root nodes are all labeled by the start symbol S.

As for TAGs, every node η in each individual tree τ is annotated. If η is a terminal (leaf) node, it is decorated by a single symbol which is its label (a terminal symbol or ε). If η is an nonterminal (adjunction) node,[4] it is decorated by two symbols: a left decoration L_η and a right decoration R_η. These symbols, called LR-variables, are RCG variables which are supposed to capture the left (resp. right) terminal yield of all the derived auxiliary trees that can be adjoined at η. Afterwards, these decorations are all gathered into a *decoration string* σ_τ during a top-down left to right traversal of τ; L-variables are gathered during the top-down traversal of non-leaf nodes, while R-variables are gathered during the bottom-up traversal. Moreover, if τ is an auxiliary tree, we have $\sigma_\tau = \sigma_\tau^l \sigma_\tau^r$, where σ_τ^l (resp. σ_τ^r), called *left* (resp. *right*) decoration string, denotes the part of σ_τ which has been gathered before (resp. after) the traversal of the foot node of τ. Afterwards, for every elementary tree set, the decoration strings of each tree are concatenated into a single decoration string, assuming some ordering on the component trees. For an initial tree set $\{\alpha\}$, σ_α denotes the decoration string of that singleton. For an auxiliary tree set β_i whose elements are the trees $\beta_{i1}, \ldots, \beta_{ij}, \ldots, \beta_{ip_i}, p_i = |\beta_i|$, its decoration string σ_{β_i} is the concatenation of the decoration strings $\sigma_{\beta_{ij}} = \sigma^l_{\beta_{ij}} \sigma^r_{\beta_{ij}}$ of its component trees and has thus the form $\sigma_{\beta_i} = \sigma^l_{\beta_{i1}} \sigma^r_{\beta_{i1}} \ldots \sigma^l_{\beta_{ij}} \sigma^r_{\beta_{ij}} \ldots \sigma^l_{\beta_{ip_i}} \sigma^r_{\beta_{ip_i}}$. Recall that $\sigma^l_{\beta_{ij}}$ has the form $L_{r_{\beta_{ij}}} \ldots L_{f_{\beta_{ij}}}$, while $\sigma^r_{\beta_{ij}}$ has the form $R_{f_{\beta_{ij}}} \ldots R_{r_{\beta_{ij}}}$ if the root and the foot nodes of β_{ij} are respectively denoted by $r_{\beta_{ij}}$ and $f_{\beta_{ij}}$.

Now, we can state our generation algorithm. For every elementary tree set τ, and for every adjunction cover ξ, we associate a unique clause $\psi_0 \rightarrow \psi_1 \ldots \psi_j \ldots \psi_m$, constructed as follows:

[3] The way such a knowledge is acquired from a grammar, either by a direct specification or by some computation from more locally defined adjunction constraints, such as the adj function, lies outside the scope of this paper.

[4] Recall that substitutions are prohibited.

- If τ is an initial tree set $\{\alpha\}$, we have $\psi_0 = S(\gamma_\alpha^\xi)$, where γ_α^ξ is a string built from its decoration string σ_α, in replacing each LR-variable L_η or R_η, such that $\xi(\eta) = nil$, by the empty string (LR-variables associated with nil adjunction constraints are erased).

- If τ is an auxiliary tree set $\beta_i = \{\beta_{i1}, \ldots, \beta_{ij}, \ldots, \beta_{ip_i}\}$, we have $\psi_0 = \beta_i(\gamma_{\beta_{i1}}^{\xi,l}, \gamma_{\beta_{i1}}^{\xi,r}, \ldots, \gamma_{\beta_{ij}}^{\xi,l}, \gamma_{\beta_{ij}}^{\xi,r}, \ldots, \gamma_{\beta_{ip_i}}^{\xi,l}, \gamma_{\beta_{ip_i}}^{\xi,r})$ where $\gamma_{\beta_{ij}}^{\xi,l}$ and $\gamma_{\beta_{ij}}^{\xi,r}$ are respectively built from $\sigma_{\beta_{ij}}^l$ and $\sigma_{\beta_{ij}}^r$, the left and right decoration strings of the auxiliary tree β_{ij}, in replacing each LR-variable L_η or R_η, such that $\xi(\eta) = nil$, by the empty string. Note that the arity of the predicate name β_i is twice the cardinality of the auxiliary tree set β_i, since, as for TAGs, the description of each individual auxiliary tree takes two arguments.

- Its RHS is produced analogously, whether we consider initial or auxiliary tree sets. Let $\xi_1, \ldots, \xi_j, \ldots, \xi_m$ be the local adjunction covers of ξ and $\Pi_\tau^\xi = \{\mathcal{N}_1, \ldots, \mathcal{N}_j, \ldots, \mathcal{N}_m\}$, $\mathcal{N}_j = dom(\xi_j)$ be the corresponding partition of $\mathcal{N}_\tau$. For each local adjunction cover ξ_j whose codomain $codom(\xi_j)$ is not $\{nil\}$, if $\beta_l = codom(\xi_j) = \{\beta_{l1}, \ldots, \beta_{lh}, \ldots, \beta_{lk}\}$, we generate the predicate call $\psi_j = \beta_l(L_1, R_1, \ldots, L_h, R_h, \ldots, L_k, R_k)$ where $L_h = L_\eta$ and $R_h = R_\eta$ if $\beta_{lh} = \xi_j(\eta)$, $\eta \in \mathcal{N}_j$. If $codom(\xi_j) = \{nil\}$, we have $\psi_j = \varepsilon$.

We can easily check that this process only builds simple clauses and that the maximum arity of its predicates is $2k$, if we start from a k-MC-TAG.[5]

If we apply, to our case, the general formula which gives the degree d of the polynomial parse time for a simple RCG i.e. $d = \max_{c_j \in P}(k_j + v_j)$ where c_j is the j^{th} clause, v_j is the number of RCG variables within c_j, and k_j is the arity of its LHS predicate. For a k-MC-TAG, we have $k_j \leq 2k$ and, if v is the maximum number of nonterminal nodes hosting a non nil adjunction constraint, we have $v_j \leq 2v$. Thus, any k-MC-TAG can be parsed at worst in $\mathcal{O}(n^{2(k+v)})$ time.

4. MC-TAG parsing optimization

We first verify that this complexity result depends both on k, the number of trees in an elementary tree set, and on v, the number of nonterminal nodes in that elementary tree set.

This relation towards k is due to the fact that, at one time, unavoidably, an auxiliary tree set τ of cardinality k, defines $2k$ discontinuous ranges, since each auxiliary tree defines both a left yield and a right yield. Naturally, the role of these $2k$ ranges is played, within the RCG framework, by the $2k$ arguments of the predicate definition associated with τ.

On the other side, the dependence towards v is less obvious since we know that, in TAG, we can get a parse time, the famous $\mathcal{O}(n^6)$, which does not depend on the form of its elementary trees. In particular, we have shown in [Boullier, 1999c] that, within the RCG framework, this constant parse time is reached because the decoration strings are well balanced (Dyck) strings in which the LR-variables play the role of parentheses. Thus we can wonder whether an analogous property holds for MC-TAGs and, in particular, whether decoration strings of elementary tree sets are extended Dyck strings in which the parentheses are the LR-variables associated with the MCA sites. Unfortunately, the answer is no. This comes from the fact that, in the general case, within a tree set, its MCA nodes can be so completely interlaced that it is not possible to isolate one MCA site, without isolating the others.

[5] We will not address the correctness of the previous algorithm and we assume that it generates a PRCG which is equivalent to the original MC-TAG.

Therefore, in [Boullier, 1999a], we propose a method which tries to replace each clause, generated by the algorithm in Section 3, by a sequence of equivalent clauses in which the number of variables (and hence the number of free bounds) has decreased. The basic idea of the method is to successively partition $\mathcal{N}_\tau$ into subsets s.t., on the one hand, each MCA site, as defined by the current adjunction cover, entirely lies within a unique subset, and, on the other hand, the number of subsets minimizes the number of free bounds in its corresponding clause. Of course, such a method may failed. However, doing so, we define a hierarchy of MC-TAG subclasses, the c-split forms, whose parse time complexity is $\mathcal{O}(n^{4k+c})$ at worst. We note that TAGs are 1-MC-TAGs in 2-split form.

5. Conclusion

We have shown that any set-local MC-TAG with unrestricted multicomponent adjunction constraints can be translated into an equivalent simple PRCG. This PRCG, in turn, as any other RCG, can be parsed in polynomial time. However, in the general case, the degree of this polynomial depends both on the maximum number k of elementary trees in a tree set and on the maximum number v of adjunction nodes in a tree set: we show that a MC-TAG can be parsed at worst in $\mathcal{O}(n^{2(k+v)})$ time. In order to release from the v parameter, we define an optimized generation algorithm which tries to minimize some value c leading to a grammar in the so-called c-split form. For k-MC-TAGs in c-split form, the parse time of its equivalent simple PRCG is $\mathcal{O}(n^{4k+c})$. This c parameter depends on the way multiple component adjunctions are interlaced within tree sets.

If we assume that, in order to be (linguistically) interesting, a MC-TAG must be at least as powerful as a TAG, the first subclass, in the split form hierarchy, corresponds to $c = 2$, for we have noticed that TAGs and 1-MC-TAGs in 2-split form are equivalent. Of course, in this case, both parsers work in $\mathcal{O}(n^6)$ time at worst. However, the linguistic relevance of the split form hierarchy still has to be demonstrated though it can easily be shown that multiple agreements of degree k (i.e. $\{a_1^n b_1^n \dots a_k^n b_k^n \mid n \geq 1, k \geq 2\}$) can be defined by a $(k-1)$-MC-TAG in 0-split form and that duplication of degree k (i.e. $\{w^k \mid w \in \{a,b\}^*, k \geq 2\}$) can also be defined by a $(k-1)$-MC-TAG in 0-split form.[6] Since RCGs can be implemented very efficiently, this approach can open the way to practical implementation of MC-TAGs. Moreover, we think that the view of MC-TAGs as a particular case of RCGs helps to understand the multiple component adjunction mechanism.

Références

[Becker, Joshi and Rambow, 1991] Becker T., Joshi A. and Rambow O. (1991). Long distance scrambling and tree adjoining grammars. In *Proceedings of the fifth Conference of the European Chapter of the Association for Computational Linguistics (EACL'91)*, pages 21–26.

[Boullier, 1998] Boullier P. (1998). Proposal for a Natural Language Processing Syntactic Backbone. In *Research Report No 3342* at `http://www.inria.fr/RRRT/RR-3342.html`, INRIA-Rocquencourt, France, Jan. 1998, 41 pages.

[Boullier, 1999a] Boullier P. (1999). On TAG and Multicomponent TAG parsing. In *Research Report No 3668* at `http://www.inria.fr/RRRT/RR-3668.html`, INRIA-Rocquencourt, France, Apr. 1999, 39 pages.

[6] Recall that cross agreements of degree k (i.e. $\{a_1^{n_1} \dots a_k^{n_k} b_1^{n_1} \dots b_k^{n_k} \mid n_i \geq 1\}$) is a TAL.

[Boullier, 1999b] Boullier P. (1999). Chinese Numbers, MIX, Scrambling, and Range Concatenation Grammars In *Proceedings of the 9th Conference of the Europwean Chapter of the Association for Computational Linguistics (EACL'99)*, Bergen, Norway, June 8–12.

[Boullier, 1999c] Boullier P. (1999). On TAG Parsing In $6^{ème}$ *conférence annuelle sur le Traitement Automatique des Langues Naturelles (TALN'99)*, Cargèse, Corsica, France, July 12-17.

[Gazdar, 1985] Gazdar G. (1985). Applicability of Indexed Grammars to Natural Languages. In *Technical Report CSLI-85-34*, Center for Study of Language and Information, 1985.

[Groenink, 1997] Groenink A. (1997). SURFACE WITHOUT STRUCTURE Word order and tractability issues in natural language analysis. PhD thesis, Utrecht University, The Netherlands, Nov. 1977, 250 pages.

[Joshi, Levy, and Takahashi, 1975] Joshi A., Levy L. and Takahashi M. (1975). Tree adjunct grammars. In *Journal of Computer ans System Sciences*, 10, pages 136–163.

[Joshi, 1985] Joshi A. (1985). How much context-sensitivity is necessary for characterizing structural descriptions — Tree Adjoining Grammars. In *Natural Language Processing — Theoritical, Computational and Psychological Perspective*, D. Dowty, L. Karttunen, and A. Zwicky, editors, Cambridge University Press, New-York, NY.

[Joshi, 1987] Joshi A. (1987). An Introduction to Tree Adjoining Grammars. In *Mathematics of Language*, Manaster-Ramer, A., editors, John Benjamins, Amsterdam, pages 87–114.

[Pollard, 1984] Pollard C. (1984). Generalized Phrase Structure Grammars, Head Grammars and Natural Language. PhD thesis, Stanford University.

[Rounds, 1988] Rounds W. (1988). LFP: A Logic for Linguistic Descriptions and an Analysis of its Complexity. In *ACL Computational Linguistics*, Vol. 14, No. 4, pages 1–9.

[Seki, Matsumura, Fujii, and Kasami, 1991] Seki H., Matsumura T., Fujii M. and Kasami T. (1991). On multiple context-free grammars. In *Theoretical Computer Science*, Elsevier Science, 88, pages 191–229.

[Steedman, 1987] Steedman M. (1987). Combinatory grammars and parasitic gaps. In *Natural language and Linguistic Theory*, 1987.

[Vijay-Shanker, 1987] Vijay-Shanker K. (1987). A study of tree adjoining grammars. *PhD thesis*, University of Pennsylvania, Philadelphia, PA.

[Vijay-Shanker, Weir, and Joshi, 1987] Vijay-Shanker K., Weir D. and Joshi A. (1987). Characterizing Structural Descriptions Produced by Various Grammatical Formalisms. In *Proceedings of the 25th Meeting of the Association for Computational Linguistics (ACL'87)*, Stanford University, CA, pages 104–111.

[Vijay-Shanker and Weir, 1994] Vijay-Shanker K. and Weir D. (1994). The equivalence of four extensions of context-free grammars. In *Math. Systems Theory*, Vol. 27. pages 511–546

[Weir, 1988] Weir D. (1988). Characterizing Mildly Context-Sensitive Grammar Formalisms. In *PhD thesis*, University of Pennsylvania, Philadelphia, PA.