Multiple Context-Free Tree Grammars and Multi-Component Tree Adjoining Grammars

Joost Engelfriet[1] and Andreas Maletti[2]

[1] LIACS, Leiden University, P.O. Box 9512, 2300 RA Leiden, The Netherlands
j.engelfriet@liacs.leidenuniv.nl

[2] Universität Leipzig, P.O. Box 100 920, 04009 Leipzig, Germany
maletti@informatik.uni-leipzig.de

Abstract. Strong lexicalization is the process of turning a grammar generating trees into an equivalent one, in which all rules contain a terminal leaf. It is known that tree adjoining grammars cannot be strongly lexicalized, whereas the more powerful simple context-free tree grammars can. It is demonstrated that *multiple* simple context-free tree grammars are as expressive as *multi-component* tree adjoining grammars and that both allow strong lexicalization.

1 Introduction

In computational linguistics several grammar formalisms [7] have been proposed that generate semilinear superclasses of the context-free languages, are able to model cross-serial dependencies, but remain parsable in polynomial time. Among the most well known are the (set-local) multi-component tree adjoining grammar (MCTAG) [5,18], which is an extension of the tree adjoining grammar (TAG), and the multiple context-free (string) grammar (MCFG) [16], which was independently discovered as (string-based) linear context-free rewriting system (LCFRS) [17]. In both cases the ability to synchronously rewrite multiple components was added to a classical model (TAG and CFG). In the same spirit, the multiple context-free *tree* grammar (MCFTG) was introduced in [8, Section 5] as the context-free graph grammar in tree generating normal form of [1], but was implicitly envisioned as *tree-based* LCFRS already in [17].

We define the MCFTG as a straightforward generalization of both the MCFG and the classical simple (i.e., linear and nondeleting) context-free tree grammar (CFTG). Intuitively, an MCFTG G is a CFTG, in which several nonterminals are rewritten in one derivation step. Thus every rule of G is a sequence of rules of a CFTG, and the left-hand side nonterminals of these rules are rewritten synchronously. However, a sequence of nonterminals can only be rewritten if (earlier in the derivation) they were introduced explicitly as such by the application of a rule of G, which is called "locality" in [18,14]. Therefore, each rule of G must also specify the sequences of (occurrences of) nonterminals in its right-hand side that may later be rewritten. Although such derivations can easily be formalized, we prefer to define the semantics of G as a least fixed point (just as for an MCFG).

R. Klasing and M. Zeitoun (Eds.): FCT 2017, LNCS 10472, pp. 217–229, 2017.
DOI: 10.1007/978-3-662-55751-8_18

Two tree-generating grammars are *strongly* (resp. *weakly*) equivalent if they generate the same tree (resp. string) language, where the string language consists of the yields of the generated trees. It is not difficult to see that for every MCTAG there is a strongly equivalent MCFTG, just as for every TAG there is a strongly equivalent CFTG [5,13]. Our main contribution is that, vice versa, for every MCFTG there is a strongly equivalent MCTAG, generalizing the result of [9] that relates monadic CFTGs and non-strict TAGs. It also settles a problem stated in [18, Section 4.5]: "It would be interesting to investigate whether there exist LCFRS's with object level tree sets that cannot be produced by any MCTAG." We prove that such LCFRSs do not exist. It is proved in the cited section that MCTAGs are weakly equivalent to string-based LCFRSs, so MCFTGs are weakly equivalent to MCFGs.

Secondly, we consider lexicalized grammars [6] in which each rule contains a lexical item (i.e., a terminal symbol that appears in the yield of the generated tree). Lexicalized grammars are of importance because they are often more understandable and allow easier parsing (cf. the Introduction of [12]); moreover, a lexicalized grammar defines a so-called dependency structure on the lexical items of each generated string, allowing to investigate certain aspects of the grammatical structure of that string, see [10]. We investigate lexicalization, which is the process that transforms a grammar into an equivalent lexicalized one. Corresponding to the two notions of equivalence we obtain strong and weak lexicalization. Although TAGs can be weakly lexicalized [3], they cannot be strongly lexicalized, as unexpectedly shown in [11]. However, the more powerful CFTGs can be strongly lexicalized [12], and the used lexicalization procedure can easily be generalized to MCFTGs. Since our transformation of an MCFTG into an MCTAG preserves the property of being lexicalized, we obtain that MCTAGs can be strongly lexicalized in contrast to classical TAGs.

The *multiplicity* (or *fan-out*) of an MCFTG G is the maximal number of nonterminals that can be rewritten simultaneously in one derivation step. Our strong lexicalization of G preserves the multiplicity of G, but our transformation of G into a strongly equivalent MCTAG increases it polynomially, and so the same is true for the strong lexicalization of MCTAGs.

2 Preliminaries

The set $\{1, 2, 3, \dots\}$ of positive integers is denoted by $\mathbb{N}$, and $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. For all $k \in \mathbb{N}_0$ we write $[k]$ for $\{i \in \mathbb{N} \mid i \leq k\}$. The cardinality of a set A is $|A|$, and we let $A^* = \bigcup_{n\in\mathbb{N}_0} A^n$ and $A^+ = \bigcup_{n\in\mathbb{N}} A^n$, where A^n is the n-fold Cartesian product of A. Note that $A^0 = \{\varepsilon\}$, where ε is the empty sequence. If A is finite, then the elements of A and A^* are also called symbols and strings, respectively. The length $|w|$ of $w \in A^*$ is such that $w \in A^{|w|}$. For a sequence $w = (a_1, \dots, a_n) \in A^n$, the set $\mathrm{occ}(w) = \{a_1, \dots, a_n\}$ contains the elements of A that occur in w, and w is *repetition-free* if no element occurs more than once (i.e., $|\mathrm{occ}(w)| = n$). The concatenation $w \cdot v$ (or just wv) of w with a sequence $v = (b_1, \dots, b_m)$ is $(a_1, \dots, a_n, b_1, \dots, b_m)$. As usual, we let $w^0 = \varepsilon$ and $w^{n+1} = ww^n$ for every $n \in \mathbb{N}_0$. For a subset $B \subseteq A$, the *yield* of w with respect to B is the sequence $\mathrm{yd}_B(w)$ in B^* that is obtained from w by removing all symbols outside B. Formally, $\mathrm{yd}_B(\varepsilon) = \varepsilon$ and for all $v \in A^*$ we have $\mathrm{yd}_B(bv) = b \cdot \mathrm{yd}_B(v)$ for all $b \in B$ and $\mathrm{yd}_B(av) = \mathrm{yd}_B(v)$ for all $a \in A \setminus B$.

A ranked alphabet is a finite set Σ with a ranking $\mathrm{rk}\colon \Sigma \to \mathbb{N}_0$. For every $k \in \mathbb{N}_0$ we let $\Sigma^{(k)} = \{\sigma \in \Sigma \mid \mathrm{rk}(\sigma) = k\}$ be the set of k-ary symbols, and let mrk_Σ be the minimal $k \in \mathbb{N}_0$ such that $\bigcup_{n=0}^{k} \Sigma^{(n)} = \Sigma$. In examples we introduce a symbol σ of rank k as $\sigma^{(k)}$. With every string $\bar{\sigma} = (\sigma_1, \ldots, \sigma_n) \in \Sigma^*$ we associate a multiple rank $\mathrm{rk}^*(\bar{\sigma}) = (\mathrm{rk}(\sigma_1), \ldots, \mathrm{rk}(\sigma_n)) \in \mathbb{N}_0^*$. We fix the countably infinite set $X = \{x_1, x_2, \ldots\}$ of variables and let $X_k = \{x_i \mid i \in [k]\}$ for every $k \in \mathbb{N}_0$. For every set $Z \subseteq X$ of variables, the set $T_\Sigma(Z)$ of *trees* over Σ and Z is the smallest set $T \subseteq (\Sigma \cup Z)^*$ such that $Z \subseteq T$ and $\sigma t_1 \cdots t_k \in T$ for all $k \in \mathbb{N}_0$, $\sigma \in \Sigma^{(k)}$, and $t_1, \ldots, t_k \in T$. As usual we also write the term $\sigma(t_1, \ldots, t_k)$ to denote $\sigma t_1 \cdots t_k$. We denote $T_\Sigma(X_0) = T_\Sigma(\emptyset)$ by T_Σ. The nodes of a tree are formalized as “positions”. The root is at position ε, and the position pi with $p \in \mathbb{N}^*$ and $i \in \mathbb{N}$ refers to the i-th child of the node at position p. Thus, the set $\mathrm{pos}(t) \subseteq \mathbb{N}^*$ of positions of a tree $t \in T_\Sigma(X)$ is defined by $\mathrm{pos}(x) = \{\varepsilon\}$ for $x \in X$ and $\mathrm{pos}(t) = \{\varepsilon\} \cup \{ip \mid i \in [k], p \in \mathrm{pos}(t_i)\}$ for $t = \sigma(t_1, \ldots, t_k)$. The label and subtree of t at $p \in \mathrm{pos}(t)$ are $t(p)$ and $t|_p$, respectively, so $x(\varepsilon) = x = x|_\varepsilon$, $t(\varepsilon) = \sigma$, $t|_\varepsilon = t$, $t(ip) = t_i(p)$, and $t|_{ip} = t_i|_p$.

A *forest* $t = (t_1, \ldots, t_m)$ is a sequence of trees $t_1, \ldots, t_m \in T_\Sigma(X)$. A single tree is a forest of length 1. The nodes of the forest t are addressed by positions from $(\mathbb{N} \cup \{\#\})^*$, where $\#$ is a special symbol. Intuitively, these positions are of the form $\#^{j-1}p$, in which $\#^{j-1}$ selects the tree t_j and $p \in \mathrm{pos}(t_j)$ is a position in t_j. Formally, $\mathrm{pos}(t) = \{\#^{j-1}p \mid j \in [m], p \in \mathrm{pos}(t_j)\}$. The label and subtree of t at position $\#^{j-1}p$ are $t(\#^{j-1}p) = t_j(p)$ and $t|_{\#^{j-1}p} = t_j|_p$, respectively. For every set $\Omega \subseteq \Sigma \cup X$, the set $\mathrm{pos}_\Omega(t) = \{p \in \mathrm{pos}(t) \mid t(p) \in \Omega\}$ contains the Ω-labeled positions of t. We let $\mathrm{occ}_\Omega(t) = \{t(p) \mid p \in \mathrm{pos}_\Omega(t)\}$ be the symbols of Ω that occur in t. The forest t is *uniquely Ω-labeled* if all symbols of Ω occur at most once in t; i.e., $|\mathrm{pos}_{\{\sigma\}}(t)| \leq 1$ for every $\sigma \in \Omega$. The set $P_\Sigma(X_k)$ of k-ary *patterns* is $P_\Sigma(X_k) = \{t \in T_\Sigma(X_k) \mid \forall x \in X_k\colon |\mathrm{pos}_{\{x\}}(t)| = 1\}$. The *rank* $\mathrm{rk}(t)$ of a k-ary pattern t is k. Clearly, $P_\Sigma(X_0) = T_\Sigma$. We let $P_\Sigma(X) = \bigcup_{k \in \mathbb{N}_0} P_\Sigma(X_k)$, and we associate the multiple rank $\mathrm{rk}^*(t) = (\mathrm{rk}(t_1), \ldots, \mathrm{rk}(t_m)) \in \mathbb{N}_0^*$ with every forest $t = (t_1, \ldots, t_m)$ of $P_\Sigma(X)^*$. For all $\theta\colon X \to T_\Sigma(X)$, the *first-order substitution* $t\theta$ is inductively defined by $x\theta = \theta(x)$, $t\theta = \sigma(t_1\theta, \ldots, t_k\theta)$, and $u\theta = (u_1\theta, \ldots, u_m\theta)$ for every $x \in X$, $t = \sigma(t_1, \ldots, t_k) \in T_\Sigma(X)$, and $u = (u_1, \ldots, u_m) \in T_\Sigma(X)^*$. Thus, each occurrence of a variable $x \in X$ is replaced by the tree $\theta(x)$. If there exists $n \in \mathbb{N}_0$ with $\theta(x_i) = x_i$ for all $i > n$, then we also write $t[\theta(x_1), \ldots, \theta(x_n)]$ instead of $t\theta$. In second-order substitution we replace nodes that are labeled by symbols of Σ. Let $\theta\colon \Sigma \to P_\Sigma(X)$ be such that $\mathrm{rk}(\theta(\sigma)) = \mathrm{rk}(\sigma)$ for all $\sigma \in \Sigma$. The *second-order substitution* $t\theta$ is inductively defined by $x\theta = x$, $t\theta = \theta(\sigma)[t_1\theta, \ldots, t_k\theta]$, and $u\theta = (u_1\theta, \ldots, u_m\theta)$ with x, t, and u as above. Intuitively, the second-order substitution $t\theta$ replaces each σ-labeled subtree of t by the tree $\theta(\sigma)$, into which the (recursively processed) direct subtrees are first-order substituted. If there exist distinct $\sigma_1, \ldots, \sigma_n \in \Sigma$

such that $\theta(\sigma) = \sigma(x_1, \ldots, x_k)$ for all $\sigma \in \Sigma^{(k)} \setminus \{\sigma_1, \ldots, \sigma_n\}$, we also write $t[(\sigma_1, \ldots, \sigma_n) \leftarrow (\theta(\sigma_1), \ldots, \theta(\sigma_n))]$ instead of $t\theta$. Finally, let $\mathcal{L} = \{\bar{\sigma}_1, \ldots, \bar{\sigma}_k\}$ be a subset of Σ^* such that $\bar{\sigma}_1 \cdots \bar{\sigma}_k$ is repetition-free. A (second-order) *substitution function* for $\mathcal{L}$ is a mapping $f \colon \mathcal{L} \to P_\Sigma(X)^*$ such that $\mathrm{rk}^*(f(\bar{\sigma})) = \mathrm{rk}^*(\bar{\sigma})$ for every $\bar{\sigma} \in \mathcal{L}$. For a forest $t \in P_\Sigma(X)^*$, the *simultaneous second-order substitution* $t[f]$ is defined by $t[f] = t[\bar{\sigma}_1 \cdots \bar{\sigma}_k \leftarrow f(\bar{\sigma}_1) \cdots f(\bar{\sigma}_k)]$. For a complete exposition of tree language theory, we refer the reader to [4].

3 Multiple Context-Free Tree Grammars

We define the (simple) multiple context-free tree grammar as a straightforward generalization of both the (simple) context-free tree grammar [15,2] and the multiple context-free (string) grammar [16,17]. We obtain essentially a tree-based linear context-free rewriting system.

Definition 1. A *(simple) multiple context-free tree grammar* (MCFTG) is a system $G = (N, \mathcal{N}, \Sigma, I, R)$ such that

- N is a ranked alphabet of *nonterminals*,
- $\mathcal{N} \subseteq N^+$ is a finite set of *big nonterminals*, which are nonempty repetition-free nonterminal sequences with $\mathrm{occ}(A) \neq \mathrm{occ}(A')$ for all distinct $A, A' \in \mathcal{N}$,
- Σ is a ranked alphabet of *terminals* such that $\Sigma \cap N = \emptyset$ and $\mathrm{mrk}_\Sigma \geq 1$,
- $I \subseteq \mathcal{N} \cap N^{(0)}$ is the set of *initial (big) nonterminals*, and
- R is a finite set of *rules* of the form $A \to (u, \mathcal{L})$, where $A \in \mathcal{N}$ is a big nonterminal, $u \in P_{N \cup \Sigma}(X)^+$ is a uniquely N-labeled forest (of patterns) such that $\mathrm{rk}^*(u) = \mathrm{rk}^*(A)$, and $\mathcal{L} \subseteq \mathcal{N}$ is a set of big nonterminals, called *links*, such that $\{\mathrm{occ}(B) \mid B \in \mathcal{L}\}$ is a partition of $\mathrm{occ}_N(u)$; i.e., $\mathrm{occ}(B) \cap \mathrm{occ}(B') = \emptyset$ for all distinct $B, B' \in \mathcal{L}$ and $\mathrm{occ}_N(u) = \bigcup_{B \in \mathcal{L}} \mathrm{occ}(B)$.

The *multiplicity* (or *fan-out*) of G, denoted by $\mu(G)$, is the maximal length of its big nonterminals. The *width* of G, denoted by $\omega(G)$, is the maximal rank of its nonterminals. A (simple) context-free tree grammar (CFTG) is an MCFTG of multiplicity 1. □

For a given rule $\rho = A \to (u, \mathcal{L})$, its left-hand side is A, its right-hand side is u, and its set of links is $\mathcal{L}$. Since $\mathrm{rk}^*(A) = \mathrm{rk}^*(u)$, the rule ρ is of the form

$$(A_1, \ldots, A_n) \to \big((u_1, \ldots, u_n), \{B_1, \ldots, B_k\}\big) \ ,$$

where $n \in \mathbb{N}$, $A_i \in N$, $u_i \in P_{N \cup \Sigma}(X_{\mathrm{rk}(A_i)})$, $k \in \mathbb{N}_0$, and $B_j \in \mathcal{N}$ for all $i \in [n]$ and $j \in [k]$. Intuitively, the application of the rule ρ consists of the simultaneous application of the n rules $A_i(x_1, \ldots, x_{\mathrm{rk}(A_i)}) \to u_i$ of an ordinary CFTG to occurrences of the nonterminals $A_1, \ldots, A_n$, and the introduction of all the nonterminals that occur in the big nonterminals $B_1, \ldots, B_k$. Every $B_j = (C_1, \ldots, C_m) \in N^+$ can be viewed as a link between the (unique) positions of u with labels $C_1, \ldots, C_m$ as well as a link between the corresponding positions after the application of ρ. The rule ρ can only be applied to positions

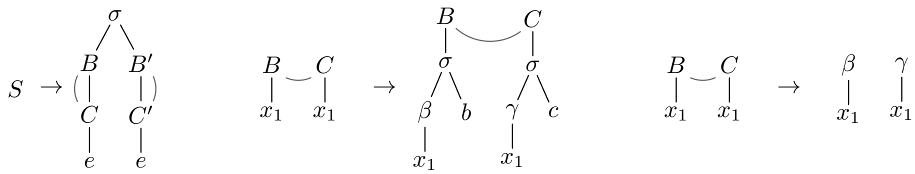

Fig. 1. The first three rules of Example 3.

with labels $A_1, \ldots, A_n$ that are joined by such a link. Thus, rule applications are "local" in the sense that a rule can rewrite only nonterminals that were previously introduced together in a single step of the derivation, just as for the local unordered scattered context grammar of [14], which is equivalent to the multiple context-free (string) grammar. Instead of defining derivation steps between trees in $T_{N\cup\Sigma}$, it is technically more convenient to define the generation of trees recursively. In an ordinary CFTG, a nonterminal A of rank k can be viewed as a generator of trees in $P_\Sigma(X_k)$ using derivations that start with $A(x_1, \ldots, x_k)$. In the same fashion, a big nonterminal A of an MCFTG generates forests in $P_\Sigma(X)^+$ of the same multiple rank as A.

Definition 2. Let $G = (N, \mathcal{N}, \Sigma, I, R)$ be an MCFTG. For every big nonterminal $A \in \mathcal{N}$ we recursively define the set $L(G, A) \subseteq P_\Sigma(X)^+$ of *forests generated* by A as follows. For every rule $\rho = A \to (u, \mathcal{L}) \in R$ and every substitution function $f\colon \mathcal{L} \to P_\Sigma(X)^+$ for $\mathcal{L}$ such that $f(B) \in L(G, B)$ for every $B \in \mathcal{L}$, the forest $u[f]$ is in $L(G, A)$. The *tree language* $L(G)$ *generated by* G is defined by $L(G) = \bigcup_{S\in I} L(G, S) \subseteq T_\Sigma$. □

Note that $u[f]$ is a simultaneous second-order substitution (see Section 2). Since $\mathrm{rk}^*(f(B)) = \mathrm{rk}^*(B)$ for all $B \in \mathcal{L}$, we have $\mathrm{rk}^*(t) = \mathrm{rk}^*(A)$ for every forest $t \in L(G, A)$. Two MCFTGs G and G' are (strongly) equivalent if $L(G) = L(G')$.

Example 3. We consider the MCFTG $G = (N, \mathcal{N}, \Sigma, \{S\}, R)$ with nonterminals $N = \{S^{(0)}, B^{(1)}, C^{(1)}, B'^{(1)}, C'^{(1)}\}$, big nonterminals $\mathcal{N} = \{S, (B, C), (B', C')\}$, terminals $\Sigma = \{\sigma^{(2)}, \beta^{(1)}, \gamma^{(1)}, b^{(0)}, c^{(0)}, e^{(0)}\}$, and the rules R (see Fig. 1):

$$S \to \sigma\big(B(C(e)), B'(C'(e))\big)$$
$$(B, C) \to \big(B(\sigma(\beta(x_1), b)),\, C(\sigma(\gamma(x_1), c))\big) \qquad (B, C) \to \big(\beta(x_1), \gamma(x_1)\big)$$
$$(B', C') \to \big(B(\sigma(\beta(x_1), b)),\, C(\sigma(\gamma(x_1), c))\big) \qquad (B', C') \to \big(\beta(x_1), \gamma(x_1)\big)\ ,$$

where we write a rule $A \to (u, \mathcal{L})$ as $A \to u$. In this example, and the next, the sets $\mathcal{L}$ of links are unique. Here they are $\{(B, C), (B', C')\}$ for the first rule, $\{(B, C)\}$ for the second and fourth rule, and $\emptyset$ for the third and fifth rule. Since the rules for (B, C) and (B', C') have the same right-hand sides and links, they are *aliases*. They represent essentially the same big nonterminal, but must be different because they occur together in the right-hand side of the first rule. It is easy to see that $L(G, (B, C)) = L(G, (B', C'))$ consists of all forests (t_m, u_m)

with $m \in \mathbb{N}_0$, where $t_m = \beta(\sigma\beta)^m x_1 b^m$ and $u_m = \gamma(\sigma\gamma)^m x_1 c^m$. Note that we here use string notation, thus, e.g., $u_2 = \gamma(\sigma\gamma)^2 x_1 c^2$ is the tree $\gamma\sigma\gamma\sigma\gamma x_1 cc$ which can be written as the term $\gamma(\sigma(\gamma(\sigma(\gamma(x_1), c)), c))$. Hence $L(G)$ consists of all trees $\sigma(t_m[u_m[e]], t_n[u_n[e]]) = \sigma\beta(\sigma\beta)^m\gamma(\sigma\gamma)^m ec^m b^m \beta(\sigma\beta)^n\gamma(\sigma\gamma)^n ec^n b^n$ with $m, n \in \mathbb{N}_0$. □

4 Lexicalization

For a given terminal alphabet Σ we fix a subset $\Delta \subseteq \Sigma^{(0)}$ of *lexical* symbols. We say that an MCFTG G is *lexicalized* if each rule contains at least one lexical symbol; i.e., if $\mathrm{pos}_\Delta(u) \neq \emptyset$ for every rule $A \to (u, \mathcal{L})$ of G. Clearly, if G is lexicalized, then $L(G)$ has finite ambiguity, in the following sense. Let the yield $\mathrm{yd}(t)$ of a tree $t \in T_\Sigma$ be the string of lexical symbols that label its leaves, from left to right. So, $\mathrm{yd}(t) = \mathrm{yd}_\Delta(t) \in \Delta^*$ (as defined in Section 2). We say that a tree language $L \subseteq T_\Sigma$ has *finite ambiguity* if $\{t \in L \mid \mathrm{yd}(t) = w\}$ is finite for every $w \in \Delta^+$ and $\{t \in L \mid \mathrm{yd}(t) = \varepsilon\} = \emptyset$. We can lexicalize MCFTGs, which means that for each MCFTG G of which $L(G)$ has finite ambiguity, we can construct an equivalent lexicalized MCFTG. This is called strong lexicalization [6,11] because we require strong equivalence.

Theorem 4. *For each MCFTG G such that $L(G)$ has finite ambiguity there is an equivalent lexicalized MCFTG G' with $\mu(G') = \mu(G)$ and $\omega(G') = \omega(G) + 1$.*

The construction is essentially the same as the one in [12] for CFTGs. First, all nonlexicalized rules of rank 0 and rank 1 are removed, where the rank of a rule $A \to (u, \mathcal{L})$ is $|\mathcal{L}|$. This is similar to the removal of rules $A \to \varepsilon$ and $A \to B$ from a context-free grammar. Since $L(G)$ has finite ambiguity, such rules can only generate finitely many trees. Second, all rules of rank 0 with exactly one lexical symbol are removed. That can be done by applying all such rules to the other rules, in all possible ways. Finally, we guess a lexical symbol for every application of a nonlexicalized rule and put the guessed symbol in a new argument of a nonterminal (thus turning the rule into a lexicalized one). It is passed from nonterminal to nonterminal until a rule of rank 0 is applied, where we exchange the same lexical symbol for the new argument. The resulting rule is still lexicalized because we made sure that rules of rank 0 contain at least two lexical symbols. Lexical symbols that are guessed for distinct rule applications are transported to distinct applications of rules of rank 0.

5 MCFTG and MCTAG

Next we prove that MCTAGs have the same tree generating power as MCFTGs. It is shown in [9, Section 4] that "non-strict" TAGs have the same tree generating power as "footed" CFTGs. Since the translation from one formalism to the other is straightforward, we avoid the introduction of the formal machinery that is needed to define MCTAGs in the usual way. Rather we first define non-strict

MCTAGs to be footed MCFTGs, which generalize footed CFTGs in an obvious way. After that we define (strict) MCTAGs as a special type of non-strict MCTAGs. The main result of [9] is that non-strict TAGs have the same tree generating power as monadic CFTGs, where a CFTG G is monadic if $\omega(G) \leq 1$. Our result shows that the monadic restriction is not needed in the multi case.

According to [9], a CFTG is footed if for every rule $A(x_1, \dots, x_k) \to u$ with $k \in \mathbb{N}$ there is a unique position of u with exactly k children that are labeled $x_1, \dots, x_k$ from left to right.

Definition 5. Let Ω be a ranked alphabet. A pattern $t \in P_\Omega(X_k)$ is *footed* if either $k = 0$, or $k \in \mathbb{N}$ and there exists $p \in \mathrm{pos}_\Omega(t)$, called the *foot node* of t, such that $t|_p = \sigma(x_1, \dots, x_k)$ for some $\sigma \in \Omega^{(k)}$. Let $G = (N, \mathcal{N}, \Sigma, I, R)$ be an MCFTG. A rule $A \to ((u_1, \dots, u_n), \mathcal{L}) \in R$ is footed if u_i is footed for every $i \in [n]$. The MCFTG G is footed if every rule in R is footed. □

Note that, by definition, every tree $t \in T_\Omega = P_\Omega(X_0)$ is footed. For a footed MCFTG G, it is straightforward to show that the trees $t_1, \dots, t_n$ are footed for every forest $(t_1, \dots, t_n) \in L(G, A)$. This implies that $\omega(G) \leq \mathrm{mrk}_\Sigma$.

Based on the close relationship between non-strict TAGs and footed CFTGs as shown in [9, Section 4], we here define a non-strict TAG to be a footed CFTG and, similarly, a non-strict MCTAG to be a footed MCFTG. To convince the reader familiar with TAGs of this definition, we add some more terminology. Let $A \to (u, \mathcal{L})$ be a rule with $A = (A_1, \dots, A_n)$ and $u = (u_1, \dots, u_n)$. If the rule is initial (i.e., $A \in I$), then the right-hand side u together with the set $\mathcal{L}$ of links is called an *initial forest*, and otherwise it is called an *auxiliary forest*. Application of the rule consists of adjunctions and substitutions. The replacement of the nonterminal A_i by u_i is called an *adjunction* if $\mathrm{rk}(A_i) \geq 1$ and a *substitution* otherwise. An occurrence of a nonterminal $C \in N$ in u with $\mathrm{rk}(C) \geq 1$ has an obligatory adjunction (OA) constraint, whereas an occurrence of a terminal $\sigma \in \Sigma$ in u with $\mathrm{rk}(\sigma) \geq 1$ has a null adjunction (NA) constraint. In the same manner we handle obligatory and null substitution (OS and NS) constraints. Each big nonterminal $B \in \mathcal{L}$ can be viewed as a selective adjunction/substitution (SA/SS) constraint, which restricts the auxiliary forests that can be adjoined/substituted for B to the right-hand sides of the rules with left-hand side B.

Given a footed pattern $t \in P_{N \cup \Sigma}(X_k)$ with $k \geq 1$, we define $\mathrm{rlab}(t) = t(\varepsilon)$ and $\mathrm{flab}(t) = t(p)$, where p is the foot node of t. Thus, $\mathrm{rlab}(t)$ and $\mathrm{flab}(t)$ are the labels of the root and the foot node of t, respectively. For $k = 0$, we let $\mathrm{rlab}(t) = t(\varepsilon)$ and $\mathrm{flab}(t) = t(\varepsilon)$ for technical convenience.

Definition 6. Let Ω, Σ be ranked alphabets and $\varphi \colon \Omega \to \Sigma$ be a fixed mapping. A pattern $t \in P_\Omega(X_k)$ is *adjoining* if it is footed and $\varphi(\mathrm{rlab}(t)) = \varphi(\mathrm{flab}(t))$. □

Definition 7. A *(strict and set-local) multi-component tree adjoining grammar* (MCTAG) is an MCFTG $G = (N, \mathcal{N}, \Sigma, I, R)$, for which there exists a rank-preserving mapping $\varphi \colon (N \cup \Sigma) \to \Sigma$ such that $\varphi(\sigma) = \sigma$ for every $\sigma \in \Sigma$, and moreover, for every rule $(A_1, \dots, A_n) \to ((u_1, \dots, u_n), \mathcal{L}) \in R$ and every $i \in [n]$, u_i is an adjoining pattern and $\varphi(\mathrm{rlab}(u_i)) = \varphi(A_i)$.

A *tree adjoining grammar* (TAG) is an MCTAG of multiplicity 1. □

The MCFTG G of Example 3 is an MCTAG with respect to the mapping φ such that $\varphi(S) = \sigma$, $\varphi(B) = \varphi(B') = \beta$, and $\varphi(C) = \varphi(C') = \gamma$.

Each nonterminal C with $\varphi(C) = \sigma$ can be viewed as the terminal symbol σ together with some information that is relevant to SA and SS constraints. The requirements in Definition 7 mean that the root and foot node of u_i represent the same terminal symbol as A_i. Thus, intuitively, adjunction always replaces a (constrained) terminal symbol by a tree with that same symbol as root label and foot node label. Thus, if $(t_1, \ldots, t_n) \in L(G, (A_1, \ldots, A_n))$ then $\mathrm{rlab}(t_i) = \mathrm{flab}(t_i) = \varphi(A_i)$ for every $i \in [n]$. Our MCTAGs and TAGs are slightly more general than the usual ones, because the roots of the generated trees need not have the same label; in other words, the underlying syntax may have more than one "sentence symbol" $\varphi(S)$ with $S \in I$. We view this as an irrelevant technicality.

Let MCFTL and MCTAL denote the classes of tree languages generated by MCFTGs and MCTAGs, respectively. We now prove that MCTAL = MCFTL. By definition, we have MCTAL $\subseteq$ MCFTL. The next theorem shows that for every MCFTG G there is an equivalent MCTAG, which is also lexicalized if G is lexicalized. Roughly speaking, the transformation of an MCFTG into an MCTAG is realized by decomposing each tree u_i in the right-hand side of a rule $A \to (u, \mathcal{L})$ with $A = (A_1, \ldots, A_n)$ and $u = (u_1, \ldots, u_n)$ into a bounded number of parts, to replace u_i in u by the sequence of these parts, and to replace A_i in A by a corresponding sequence of new nonterminals that simultaneously generate these parts.

Theorem 8. *For every MCFTG G with terminal alphabet Σ there is an equivalent MCTAG G' such that $\mu(G') \leq \mu(G) \cdot \mathrm{mrk}_\Sigma \cdot |\Sigma| \cdot (2 \cdot \omega(G) - 1)$ if $\omega(G) \neq 0$, and $\mu(G') = \mu(G)$ otherwise. Moreover, if G is lexicalized, then so is G'.*

Proof. The basic fact used in this proof is that, for any ranked alphabet Ω and mapping $\varphi \colon \Omega \to \Sigma$, every tree $u \in T_\Omega(X)$ with $u \notin X$ and $\mathrm{pos}_X(u) \neq \emptyset$ can be decomposed into at most $\mathrm{mrk}_\Omega \cdot |\Sigma| \cdot (2k - 1)$ adjoining patterns, where k is the number $|\mathrm{pos}_X(u)|$ of occurrences of variables in u. This decomposition can be obtained inductively as follows. Let $p \in \mathrm{pos}_\Omega(u)$ be the longest position such that $\varphi(u(p)) = \varphi(u(\varepsilon))$ and $|\mathrm{pos}_X(u|_p)| = |\mathrm{pos}_X(u)|$. Then there are an adjoining pattern $u_\varepsilon \in P_\Omega(X_m)$ and trees $u_1, \ldots, u_m \in T_\Omega(X)$ such that $m = \mathrm{rk}(u(p)) \geq 1$, $u = u_\varepsilon[u_1, \ldots, u_m]$, and p is the foot node of u_ε. In other words, u is decomposed as $u_\varepsilon[u_1, \ldots, u_m]$ where u_ε is an adjoining pattern. For every $i \in [m]$ with $u_i \notin X$, either $u_i \in T_\Omega$ and so u_i is an adjoining pattern of rank 0, or $\mathrm{pos}_X(u_i) \neq \emptyset$, in which case the tree u_i can be decomposed further. It should be clear that, in this inductive process, there are at most $|\Sigma| \cdot (2k-1)$ such positions p. The factor mrk_Ω is due to the adjoining patterns of rank 0. As an example, let $\Omega = \{\sigma^{(2)}, \tau^{(2)}, \beta^{(1)}, a^{(0)}, b^{(0)}\}$ and let φ be the identity on Ω. Then the tree $u = \sigma(a, \sigma(v, \sigma(x_3, b)))$ with $v = \sigma(a, \tau(a, \sigma(a, \tau(x_1, \beta(\beta(x_2))))))$ is decomposed as $u = u_\varepsilon[u_1[u_{11}, u_{12}[x_1, u_{122}[x_2]]], u_2[x_3, u_{22}]]$ into the adjoining patterns $u_\varepsilon = \sigma(a, \sigma(x_1, x_2))$, $u_1 = \sigma(a, \tau(a, \sigma(x_1, x_2)))$, $u_{11} = a$, $u_{12} = \tau(x_1, x_2)$, $u_{122} = \beta(\beta(x_1))$, $u_2 = \sigma(x_1, x_2)$, and $u_{22} = b$. Using new symbols C_p^α such that $\alpha \in \Omega$, $p \in \mathbb{N}^*$, and $\mathrm{rk}(C_p^\alpha) = \mathrm{rk}(\alpha)$, we can also express this decomposition as $u = K[\gamma]$, where K is the tree $C_\varepsilon^\sigma(C_1^\sigma(C_{11}^a, C_{12}^\tau(x_1, C_{122}^\beta(x_2))), C_2^\sigma(x_3, C_{22}^b))$, which can be viewed as the skeleton of the decomposition, and γ is the substitution function such that $\gamma(C_p^\alpha) = u_p$. Note that the superscript α of C_p^α is equal to $\varphi(\mathrm{rlab}(u_p))$. This decomposition is formalized below and applied to (variants of) the trees in the right-hand sides of the rules of G.

Let $G = (N, \mathcal{N}, \Sigma, I, R)$ be an MCFTG. Provided that $\omega(G) \neq 0$, then we have $\mathrm{mrk}_\Sigma \cdot (2 \cdot \omega(G) - 1) \geq 1$ because $\mathrm{mrk}_\Sigma \geq 1$ by Definition 1. By straightforward constructions we may assume that G is "permutation-free" and "nonerasing". This means that if $(A_1, \ldots, A_n) \to ((u_1, \ldots, u_n), \mathcal{L})$ is a rule in R, then the pattern u_i is in $PF_{N \cup \Sigma}(X_{\mathrm{rk}(A_i)}) \setminus X$ for every $i \in [n]$, where $PF_\Omega(X_k)$ denotes the set of permutation-free k-ary patterns over Ω; i.e., patterns $t \in P_\Omega(X_k)$ such that $\mathrm{yd}_X(t) = x_1 \cdots x_k$. The nonerasing requirement that $u_i \notin X$ is only relevant when $\mathrm{rk}(A_i) = 1$, meaning that $u_i \neq x_1$.

We define $G' = (N', \mathcal{N}', \Sigma, I', R')$. The set N' of nonterminals consists of all quadruples $\langle C, \sigma, m, p\rangle$ with $C \in N$, $\sigma \in \Sigma$, $m \in \{0, \mathrm{rk}(\sigma)\}$, and $p \in \mathbb{N}^*$ such that $|p| \leq |\Sigma| \cdot \omega(G)$. The rank of $\langle C, \sigma, m, p\rangle$ is m. The set of initial nonterminals is $I' = \{\langle S, \sigma, 0, \varepsilon\rangle \mid S \in I, \sigma \in \Sigma\}$. We will define $\mathcal{N}'$ and R' in such a way that G' is an MCTAG with respect to the mapping $\varphi\colon (N' \cup \Sigma) \to \Sigma$ such that $\varphi(\langle C, \sigma, m, p\rangle) = \varphi(\sigma) = \sigma$. For every nonterminal $C \in N$, a *skeleton* of C is a pattern $K \in PF_{N'}(X_{\mathrm{rk}(C)}) \setminus X$ such that

(1) for every $p \in \mathrm{pos}_{N'}(K)$ there exist a symbol $\sigma \in \Sigma$ and $m \in \{0, \mathrm{rk}(\sigma)\}$ such that $K(p) = \langle C, \sigma, m, p\rangle$;
(2) for all $p, q \in \mathrm{pos}_{N'}(K)$, if position q is a proper descendant of position p, then $\varphi(K(q)) \neq \varphi(K(p))$ or $|\mathrm{pos}_X(K|_q)| < |\mathrm{pos}_X(K|_p)|$;
(3) for every $p \in \mathrm{pos}_{N'}(K)$, if $K|_p \in T_{N'}$ then $\mathrm{rk}(K(p)) = 0$.

For such a skeleton K, we let $\mathrm{seq}(K) = \mathrm{yd}_{N'}(K)$, which is in $(N')^+$. There are only finitely many skeletons K of C because $|\mathrm{pos}_{N'}(K)| \leq \mathrm{mrk}_\Sigma \cdot |\Sigma| \cdot (2k - 1)$, if $k = \mathrm{rk}(C) \geq 1$. If $\mathrm{rk}(C) = 0$, then each skeleton of C is of the form $\langle C, \sigma, 0, \varepsilon\rangle$ with $\sigma \in \Sigma$. Note that K can be reconstructed from $\mathrm{seq}(K)$ because K is permutation-free. In the example above, the tree K is a skeleton of C, provided C_p^α denotes $\langle C, \alpha, \mathrm{rk}(\alpha), p\rangle$, and $\mathrm{seq}(K) = (C_\varepsilon^\sigma, C_1^\sigma, C_{11}^a, C_{12}^\tau, C_{122}^\beta, C_2^\sigma, C_{22}^b)$.

We apply the above basic fact to patterns over $N' \cup \Sigma$. Let K be a skeleton of $C \in N$. A substitution function γ for $\mathrm{occ}_{N'}(K)$ is *adjoining* if, for every $C' \in \mathrm{occ}_{N'}(K)$, the pattern $\gamma(C') \in P_{N' \cup \Sigma}(X)$ is adjoining and we have $\varphi(\mathrm{rlab}(\gamma(C'))) = \varphi(C')$. We say that the pair $\langle K, \gamma\rangle$ is an *adjoining C-decomposition* of the tree $K[\gamma]$. By a straightforward induction, following the above basic fact, we can prove that every pattern u over $N' \cup \Sigma$ has an adjoining C-decomposition $\mathrm{dec}_C(u)$. More precisely, for every $C \in N$ and every $u \in PF_{N' \cup \Sigma}(X_{\mathrm{rk}(C)}) \setminus X$ there is a pair $\mathrm{dec}_C(u) = \langle K, \gamma\rangle$ such that K is a skeleton of C, γ is an adjoining substitution function for $\mathrm{occ}_{N'}(K)$, and $K[\gamma] = u$.

A *skeleton function* for $A \in \mathcal{N}$ is a substitution function κ for $\mathrm{occ}(A)$ that assigns a skeleton $\kappa(C)$ of C to every nonterminal $C \in \mathrm{occ}(A)$. The string homomorphism h_κ from $\mathrm{occ}(A)^*$ to $(N')^*$ is defined by $h_\kappa(C) = \mathrm{seq}(\kappa(C))$ for every $C \in \mathrm{occ}(A)$. We define the set $\mathcal{N}'$ of big nonterminals to be the set of all $h_\kappa(A)$, where $A \in \mathcal{N}$ and κ is a skeleton function for A.

We finally define the set R' of rules of G'. Let $\rho = A \to (u, \mathcal{L})$ be a rule in R such that $A = (A_1, \ldots, A_n)$, $u = (u_1, \ldots, u_n)$, and $\mathcal{L} = \{B_1, \ldots, B_k\}$. Moreover, let $\overline{\kappa} = (\kappa_1, \ldots, \kappa_k)$, where κ_j is a skeleton function for B_j for every $j \in [k]$. Intuitively, $\overline{\kappa}$ guesses for every nonterminal C that occurs in $B_1, \ldots, B_k$ the skeleton of an adjoining C-decomposition of the tree generated by C. Let f be the substitution function for $\mathrm{occ}_N(u)$ such that $f = \bigcup_{j\in[k]} \kappa_j$; i.e., $f(C) = \kappa_j(C)$ if $C \in \mathrm{occ}(B_j)$. Obviously, $u_i[f] \in PF_{N'\cup\Sigma}(X_{\mathrm{rk}(A_i)}) \setminus X$ for every $i \in [n]$. For every $i \in [n]$, let $u'_i = u_i[f]$ and let $\mathrm{dec}_{A_i}(u'_i) = \langle K_i, \gamma_i\rangle$ (the adjoining A_i-decomposition of u'_i); moreover, if $\mathrm{seq}(K_i) = (C'_1, \ldots, C'_\ell)$ with $C'_1, \ldots, C'_\ell \in N'$, then let $v'_i = (\gamma_j(C'_1), \ldots, \gamma_j(C'_\ell))$. Then we construct the rule

$$\langle \rho, \overline{\kappa}\rangle \;=\; \mathrm{seq}(K_1)\cdots\mathrm{seq}(K_n) \to (v'_1 \cdots v'_n, \mathcal{L}')$$

with $\mathcal{L}' = \{h_{\kappa_1}(B_1), \ldots, h_{\kappa_k}(B_k)\}$ in R'. We also define the skeleton function $\kappa_{\rho,\overline{\kappa}}$ for A by $\kappa_{\rho,\overline{\kappa}}(A_i) = K_i$ for every $i \in [n]$. Intuitively, K_i is the skeleton of an adjoining A_i-decomposition of the tree generated by A_i, resulting from the skeletons guessed by $\overline{\kappa}$.

It should be clear that G' is an MCTAG with respect to φ. Moreover, since the right-hand sides of the rules ρ and $\langle\rho, \overline{\kappa}\rangle$ contain the same terminal symbols, G' is lexicalized if G is lexicalized. The intuition underlying the correctness of G' is that for every A_i, the skeleton K_i generates the same terminal tree in G' as A_i generates in G, provided that the skeleton $\kappa_j(C)$ generates the same terminal tree in G' as C generates in G for every $j \in [k]$ and $C \in \mathrm{occ}(B_j)$. □

Example 9. We consider the footed CFTG $G_1 = (N_1, \mathcal{N}_1, \Sigma, \{S\}, R_1)$ such that $N_1 = \mathcal{N}_1 = \{S^{(0)}, A^{(1)}, A'^{(1)}\}$, $\Sigma = \{\tau^{(3)}, \ell^{(1)}, r^{(1)}, a^{(0)}, b^{(0)}, e^{(0)}\}$, and R_1 contains the rules $S \to \ell AA're$, $A \to \ell AA'rx_1$, and $A \to \ell\ell\tau(a, b, rrx_1)$, plus the two rules for the alias A' of A. Note that, for the sake of readability, we omit here and in what follows the parentheses around the arguments of unary symbols; e.g., the right-hand side of the third rule is $\ell(\ell(\tau(a, b, r(r(x_1))))) = \ell\ell\tau abrrx_1$. Let $\Delta = \{a, b\}$ be the set of lexical symbols. Clearly, $L(G_1)$ has finite ambiguity. However, there is no equivalent lexicalized footed CFTG. In fact, G_1 is a variant of the TAG of [11], for which there is no (strongly) equivalent lexicalized TAG. Thus, since we defined non-strict TAGs to be footed CFTGs, G_1 is a non-strict TAG that cannot be lexicalized (as non-strict TAG). We will construct an equivalent lexicalized MCTAG for G_1.

From Theorem 4, or rather from [12], we obtain an equivalent lexicalized CFTG G_2 with $\omega(G_2) = 2$. It has the new nonterminals $B^{(2)}$ and $B'^{(2)}$, where B' is an alias of B. Its rules are

$$\begin{array}{lll} \rho_1\colon\ S \to \ell AB(b, re) & \rho_2\colon\ A \to \ell AB(b, rx_1) & \rho_4\colon\ B \to \ell B(x_1, B'(b, rx_2)) \\ & \rho_3\colon\ A \to \ell\ell\tau(a, b, rrx_1) & \rho_5\colon\ B \to \ell\ell\tau(a, x_1, rrx_2) \end{array}$$

plus the rules ρ'_4 and ρ'_5 for B'. Clearly, the tree $B(b, x_1)$ generates the same terminal trees as $A(x_1)$.

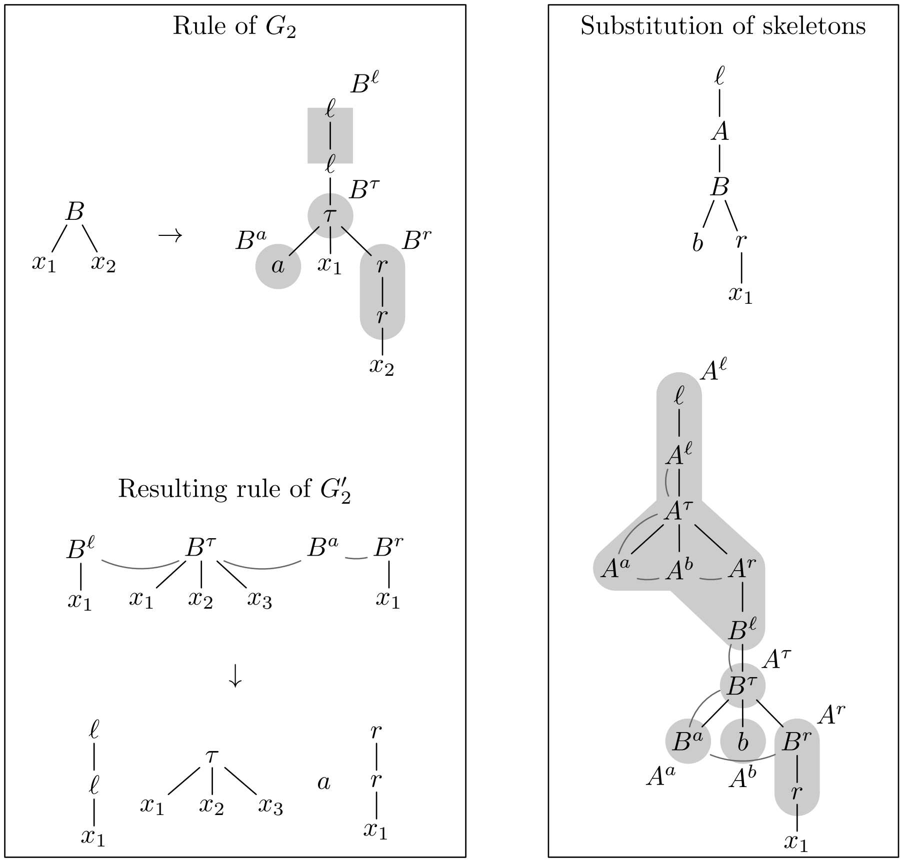

Fig. 2. Left part: The adjoining decomposition $\langle K_5, \gamma_5\rangle$ of the right-hand side of rule ρ_5, with the resulting rule $\tilde{\rho}_5$. Right part: Substitution of the skeletons K_3 of A and K_5 of B into the right-hand side of rule ρ_2, with the adjoining decomposition $\langle K_3, \gamma_2\rangle$.

We now turn G_2 into an equivalent lexicalized MCTAG G'_2 using the construction in the proof of Theorem 8. For the rule $\rho_5 = B \to u_5$ and $\overline{\kappa} = \varepsilon$, we obtain the adjoining B-decomposition $\mathrm{dec}_B(u_5) = \langle K_5, \gamma_5\rangle$, in which we have $K_5 = B^\ell(B^\tau(B^a, x_1, B^r(x_2)))$, $\gamma_5(B^\ell) = \ell\ell x_1$, $\gamma_5(B^\tau) = \tau(x_1, x_2, x_3)$, $\gamma_5(B^a) = a$, and $\gamma_5(B^r) = rrx_1$, where $B^\ell = \langle B, \ell, 1, \varepsilon\rangle$, $B^\tau = \langle B, \tau, 3, 1\rangle$, $B^a = \langle B, a, 0, 11\rangle$, and $B^r = \langle B, r, 1, 13\rangle$. The resulting rule $\tilde{\rho}_5 = \langle \rho_5, \varepsilon\rangle$ is

$$\tilde{\rho}_5\colon \quad \bar{B} \to (\ell\ell x_1,\, \tau(x_1, x_2, x_3),\, a,\, rrx_1)\ ,$$

where $\bar{B} = \mathrm{seq}(K_5) = (B^\ell, B^\tau, B^a, B^r)$, and the corresponding skeleton function for B is $\kappa_5 = \kappa_{\rho_5,\varepsilon}$ such that $\kappa_5(B) = K_5$. The construction of this rule is illustrated in the left part of Fig. 2. Of course we obtain similar primed results for B'. For the rule $\rho_4 = B \to u_4$ and $\overline{\kappa} = (\kappa_5, \kappa'_5)$, we substitute K_5 for B and K'_5 for B' in u_4 and obtain the tree $u'_4 = \ell B^\ell B^\tau(B^a, x_1, B^r B'^\ell B'^\tau(B'^a, b, B'^r r x_2))$

with the adjoining B-decomposition $\mathrm{dec}_B(u'_4) = \langle K_4, \gamma_4\rangle$, where K_4 equals K_5, and γ_4 is defined by $\gamma_4(B^\ell) = \ell B^\ell x_1$, $\gamma_4(B^\tau) = B^\tau(x_1, x_2, x_3)$, $\gamma_4(B^a) = B^a$, and $\gamma_4(B^r) = B^r B'^\ell B'^\tau(B'^a, b, B'^r r x_1)$. The resulting rule $\tilde{\rho}_4 = \langle \rho_4, (\kappa_5, \kappa'_5)\rangle$ is

$$\tilde{\rho}_4\colon \quad \bar{B} \to (\ell B^\ell x_1,\ B^\tau(x_1, x_2, x_3),\ B^a,\ B^r B'^\ell B'^\tau(B'^a, b, B'^r r x_1)) \enspace .$$

Since the skeleton function $\kappa_{\rho_4,(\kappa_5,\kappa'_5)}$ for B is again κ_5, these are all the necessary rules of G'_2 with left-hand side $\bar{B}$, and similarly for $\bar{B}' = (B'^\ell, B'^\tau, B'^a, B'^r)$. We now turn to rules ρ_3 and ρ_2. The only skeleton needed for A is the tree

$$K_3 = \kappa_{\rho_3,\varepsilon}(A) = A^\ell A^\tau(A^a, A^b, A^r x_1) \enspace ,$$

where $A^\ell = \langle A, \ell, 1, \varepsilon\rangle$, $A^\tau = \langle A, \tau, 3, 1\rangle$, $A^a = \langle A, a, 0, 11\rangle$, $A^b = \langle A, b, 0, 12\rangle$, and $A^r = \langle A, r, 1, 13\rangle$. The rule $\tilde{\rho}_3 = \langle \rho_3, \varepsilon\rangle$ is

$$\tilde{\rho}_3\colon \quad \bar{A} \to (\ell\ell x_1,\ \tau(x_1, x_2, x_3),\ a,\ b,\ rrx_1) \enspace ,$$

where $\bar{A} = \mathrm{seq}(K_3) = (A^\ell, A^\tau, A^a, A^b, A^r)$. Substituting K_3 for A and K_5 for B in the right-hand side $u_2 = \ell AB(b, rx_1)$ of ρ_2, we obtain

$$u'_2 = \ell A^\ell A^\tau(A^a, A^b, A^r B^\ell B^\tau(B^a, b, B^r r x_1)) \enspace ,$$

which has the decomposition $\mathrm{dec}_A(u'_2) = \langle K_3, \gamma_2\rangle$ shown in the right part of Fig. 2, where $\gamma_2(A^\ell) = \ell A^\ell A^\tau(A^a, A^b, A^r B^\ell x_1)$, $\gamma_2(A^\tau) = B^\tau(x_1, x_2, x_3)$, $\gamma_2(A^a) = B^a$, $\gamma_2(A^b) = b$, and $\gamma_2(A^r) = B^r r x_1$. The rule $\tilde{\rho}_2 = \langle \rho_2, (\kappa_{\rho_3,\varepsilon}, \kappa_5)\rangle$ is

$$\tilde{\rho}_2\colon \quad \bar{A} \to (\ell A^\ell A^\tau(A^a, A^b, A^r B^\ell x_1),\ B^\tau(x_1, x_2, x_3),\ B^a,\ b,\ B^r r x_1) \enspace .$$

Finally, we consider rule ρ_1. The only skeleton needed for S is $S^\ell = \langle S, \ell, 0, \varepsilon\rangle$, which is the unique initial nonterminal of G'. Substituting K_3 for A and K_5 for B in the right-hand side of ρ_1, we obtain $u'_2[e]$ and the rule $\tilde{\rho}_1 = \langle \rho_1, (\kappa_{\rho_3,\varepsilon}, \kappa_5)\rangle$:

$$\tilde{\rho}_1\colon \quad S^\ell \to \ell A^\ell A^\tau(A^a, A^b, A^r B^\ell B^\tau(B^a, b, B^r r e)) \enspace .$$

Thus, G'_2 has the rules $\{\tilde{\rho}_1, \tilde{\rho}_2, \tilde{\rho}_3, \tilde{\rho}_4, \tilde{\rho}_5, \tilde{\rho}'_4, \tilde{\rho}'_5\}$. Clearly, the tree K_3 generates the same terminal trees as $A(x_1)$ and the tree K_5 generates the same terminal trees as $B(x_1, x_2)$, and hence the tree $B^\ell B^\tau(B^a, b, B^r(x_1))$ also generates the same terminal trees as $A(x_1)$. It is easy to check that G'_2 is a lexicalized MCTAG with respect to the mapping φ such that $\varphi(C^x) = x$ for every $C \in \{S, A, B, B'\}$ and every $x \in \{\tau, \ell, r, a, b\}$. The multiplicity of G'_2 is $\mu(G'_2) = 5$. □

It follows from Theorems 4 and 8 that MCTAGs can be strongly lexicalized as opposed to TAGs.

Corollary 10. *For every finitely ambiguous MCTAG G with terminal alphabet Σ there is an equivalent lexicalized MCTAG G' such that*

$$\mu(G') \leq \mu(G) \cdot \mathrm{mrk}_\Sigma \cdot |\Sigma| \cdot (2 \cdot \omega(G) + 1) \enspace . \qquad \square$$

We do not know whether the multiplicity bounds in Theorem 8 and Corollary 10 are optimal.

Acknowledgment. We are grateful to the reviewers for their constructive comments.

References

1. Engelfriet, J., Maneth, S.: Tree languages generated by context-free graph grammars. In: Proc. 8th Int. Workshop Theory and Application of Graph Transformation. LNCS, vol. 1764, pp. 15–29 (2000)
2. Engelfriet, J., Schmidt, E.M.: IO and OI I. J. Comput. System Sci. 15(3), 328–353 (1977)
3. Fujiyoshi, A.: Epsilon-free grammars and lexicalized grammars that generate the class of the mildly context-sensitive languages. In: Proc. 7th Int. Workshop Tree Adjoining Grammar and Related Formalisms. pp. 16–23 (2005)
4. Gécseg, F., Steinby, M.: Tree languages. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, vol. 3, chap. 1, pp. 1–68. Springer (1997)
5. Joshi, A.K., Levy, L.S., Takahashi, M.: Tree adjunct grammars. J. Comput. System Sci. 10(1), 136–163 (1975)
6. Joshi, A.K., Schabes, Y.: Tree-adjoining grammars and lexicalized grammars. In: Nivat, M., Podelski, A. (eds.) Tree Automata and Languages, pp. 409–431. North-Holland (1992)
7. Kallmeyer, L.: Parsing Beyond Context-Free Grammars. Cognitive Technologies, Springer (2010)
8. Kanazawa, M.: Second-order abstract categorial grammars as hyperedge replacement grammars. J. Log. Lang. Inf. 19(2), 137–161 (2010)
9. Kepser, S., Rogers, J.: The equivalence of tree adjoining grammars and monadic linear context-free tree grammars. J. Log. Lang. Inf. 20(3), 361–384 (2011)
10. Kuhlmann, M.: Dependency Structures and Lexicalized Grammars: An Algebraic Approach, LNAI, vol. 6270. Springer (2010)
11. Kuhlmann, M., Satta, G.: Tree-adjoining grammars are not closed under strong lexicalization. Comput. Linguist. 38(3), 617–629 (2012)
12. Maletti, A., Engelfriet, J.: Strong lexicalization of tree adjoining grammars. In: Proc. 50th Ann. Meeting Association for Computational Linguistics. pp. 506–515. ACL (2012)
13. Mönnich, U.: Adjunction as substitution: An algebraic formulation of regular, context-free and tree adjoining languages. In: Proc. 3rd Int. Conf. Formal Grammar. pp. 169–178. Université de Provence, France (1997)
14. Rambow, O., Satta, G.: Independent parallelism in finite copying parallel rewriting systems. Theoret. Comput. Sci. 223(1–2), 87–120 (1999)
15. Rounds, W.C.: Context-free grammars on trees. In: Proc. 1st ACM Symp. Theory of Computation. pp. 143–148. ACM (1969)
16. Seki, H., Matsumura, T., Fujii, M., Kasami, T.: On multiple context-free grammars. Theoret. Comput. Sci. 88(2), 191–229 (1991)
17. Vijay-Shanker, K., Weir, D.J., Joshi, A.K.: Characterizing structural descriptions produced by various grammatical formalisms. In: Proc. 25th Ann. Meeting Association for Computational Linguistics. pp. 104–111. ACL (1987)
18. Weir, D.J.: Characterizing Mildly Context-Sensitive Grammar Formalisms. Ph.D. thesis, University of Pennsylvania (1988)