

A Direct Link between Tree-Adjoining and Context-free Tree Grammars

Kilian Gebhardt and Johannes Osterholzer

Faculty of Computer Science
Technische Universität Dresden
01062 Dresden, Germany

{kilian.gebhardt, johannes.osterholzer}@tu-dresden.de

Abstract

The tree languages of tree-adjoining grammars are precisely those of linear monadic context-free tree grammars. Unlike the original proof, we present a direct transformation of a tree-adjoining grammar into an equivalent linear monadic context-free tree grammar.

1 Introduction

Tree-adjoining grammars (tag) are tree-generating systems which enjoy widespread use in natural language processing to model syntactic phenomena which can not be adequately described by context-free grammars. In contrast, *context-free tree grammars (cftg)* were investigated in the field of formal tree languages, and there is arguably more insight on their tree languages than on those of tag: to name just a few results, they were characterized by least fixed points (Engelfriet and Schmidt, 1977), in Chomsky-Schützenberger-like theorems (Arnold and Dauchet, 1977; Kanazawa, 2013), or by push-down machines (Guessarian, 1983). Moreover, various syntactic restrictions of cftg have been investigated (Leguy, 1981; Fujiyoshi and Kasai, 2000).

Kepser and Rogers (2011) proved that the tree languages of tag are precisely those of linear monadic cftg (lm-cftg), and hence, the established results about cftg can also be applied to tag. However, this proof rests upon a number of intermediate normal forms. In particular, in the construction of a lm-cftg G from a tag H , three steps are involved: first, a *footed* cftg G' is read off from H , then G' is turned into a *spinal-formed* cftg G'' , and finally a result by Fujiyoshi and Kasai (2000) on spinal-formed cftg is applied to obtain G from G'' .

Therefore the relationship between H and its equivalent cftg G is not immediately apparent, as all involved constructions must be understood and reenacted to obtain H . Moreover, much of the struc-

ture of H is altered along the way to G . In the following, we describe an alternative direct construction of an equivalent linear monadic cftg G' from H . We argue that G' resembles H very closely.

After introducing some preliminary notions in Section 2, we will describe our construction and prove its correctness in Section 3. Section 4 illustrates the construction in an example, and contrasts it to the construction of Kepser and Rogers.

2 Preliminaries

The set of nonnegative integers is \mathbb{N} , and \mathbb{B} is the set of truth values $\{\mathbf{f}, \mathbf{t}\}$. The power set of a set A is 2^A . Denote, for $n \in \mathbb{N}$, the sets of variables $\{x_1, \dots, x_n\}$ and $\{y_1, \dots, y_n\}$ by X_n , resp. Y_n . An alphabet is a finite nonempty set.

We will briefly introduce the notation used for trees and their operations; for a thorough introduction to tree languages refer to Gécseg and Steinby (1984); see also Fülöp and Vogler (2009) for a notation that is closer to ours. Contrary to custom, we consider *unranked* trees, but the number of children of a node will be bounded by a constant χ . Thus, there is no impact on the power of the employed formalisms.

So assume some global constant $\chi \in \mathbb{N}$, and let the alphabet C consist of the three symbols \langle, \rangle , and \cdot . Let Σ be an alphabet, and let V be a set, both disjoint from C . The set $U_\Sigma(V)$ of *trees* (indexed by V) is the smallest set $U \subseteq (\Sigma \cup V \cup C)^*$ such that $V \subseteq U$ and for each $0 \leq k \leq \chi$, $\sigma \in \Sigma$, and $\xi_1, \dots, \xi_k \in U$, also $\sigma(\xi_1, \dots, \xi_k) \in U$. Given $W \subseteq V$, a tree $\xi \in U_\Sigma(V)$ is *linear (nondeleting)* in W if every element of W appears at most (at least) once in ξ . Henceforth, Σ denotes an alphabet; when not mentioned, the number k (of children of a node) will be quantified implicitly by $k \in \{0, \dots, \chi\}$. We write U_Σ for $U_\Sigma(\emptyset)$.

Let $\xi, \zeta \in U_\Sigma(V)$ for some set V . We denote the set of *positions of* ξ by $\text{pos}(\xi) \subseteq \mathbb{N}^*$. Let $w \in \text{pos}(\xi)$. The result of *replacing* the subtree of ξ at w

by ζ is written $\xi[\zeta]_w$. Finally, let $v_1, \dots, v_k \in V$ be pairwise distinct, and $\eta_1, \dots, \eta_k \in U_\Sigma(V)$. Denote by $\xi[v_1/\eta_1, \dots, v_k/\eta_k]$ the result of *substituting* every occurrence of v_i in ξ with η_i , for $1 \leq i \leq k$.

We recall one-state top-down tree transducers (td-tt₁), cf. Engelfriet (1975). Let Δ be an alphabet. A *td-tt₁* ρ from U_Σ to U_Δ is given by a finite set of rules of the form $\ast(\sigma(x_1, \dots, x_k)) \rightarrow \xi$, where \ast is the sole state of ρ , $\sigma \in \Sigma$, and $\xi \in U_\Delta(\ast(X_k))$, with $\ast(X_k) = \{\ast(x_i) \mid x_i \in X_k\}$. The tree transformation of ρ is given by term rewriting, it is a function $U_\Sigma \rightarrow 2^{U_\Delta}$ (or a partial function $U_\Sigma \rightarrow U_\Delta$ if ρ is deterministic), and also denoted by ρ .

A *context-free tree grammar (cftg)* over Σ is a tuple $G = (N, \Sigma, S, P)$ such that Σ and N are disjoint alphabets, $S \in N$, and P is a finite set of productions of the form $A(y_1, \dots, y_k) \rightarrow \xi$ for some $A \in N$, and $\xi \in U_{N \cup \Sigma}(Y_k)$. Such a production is *linear (nondeleting)* if ξ is linear (nondeleting) in Y_k . We call a cftg G *linear* (resp. *nondeleting*) if every production of G is so, and G is *monadic* if the only variable occurring in its productions is y_1 . In this case, y_1 is written as y . A *lm-cftg* is a linear and monadic cftg. Let $G = (N, \Sigma, S, P)$ be a cftg. Given $\zeta_1, \zeta_2 \in U_{N \cup \Sigma}$, we write $\zeta_1 \Rightarrow_G \zeta_2$ if there are $A(y_1, \dots, y_k) \rightarrow \xi$ in P , $\xi_1, \dots, \xi_k \in U_\Sigma$, and $w \in \text{pos}(\zeta_1)$ such that $\zeta_1|_w = A(\xi_1, \dots, \xi_k)$ and $\zeta_2 = \zeta_1[\xi[y_1/\xi_1, \dots, y_k/\xi_k]]_w$. This corresponds to the second-order substitution of the occurrence of A at w by ξ . The *tree language of G* , denoted $L(G)$, is the set $\{\xi \in U_\Sigma \mid S \Rightarrow_G^* \xi\}$.

Let Z be a finite set. The set of *tag labels* is $\Lambda = \{\langle \sigma, V, c, f \rangle \mid \sigma \in \Sigma, V \subseteq Z, c, f \in \mathbb{B}\}$. This notation is similar to the one of Kepser and Rogers (2011): in $\langle \sigma, V, c, f \rangle$, σ is the *label*, V the *selective adjunction constraint*, and c the *obligatory adjunction constraint*. The information whether $\langle \sigma, V, c, f \rangle$ labels a foot node is stored in f . It is a *foot label* if $f = \mathbf{t}$, and we say that it is *operative* (resp. *terminable*) if $V \neq \emptyset$ (resp. $c = \mathbf{f}$). Let U denote the set of all $\xi \in U_\Lambda$ with no occurrence of a foot label. Moreover, let U_\ast be the set of all $\xi \in U_\Lambda$ with exactly one occurrence of a foot label, such that this occurrence is a leaf of ξ . In this situation ξ can be decomposed into a unique $\tilde{\xi} \in U_\Lambda(X_1)$ with no occurrence of a foot label and exactly one occurrence of x_1 , as well as a $\langle \sigma, V, c, \mathbf{t} \rangle \in \Lambda$ such that $\xi = \tilde{\xi}[\langle \sigma, V, c, \mathbf{t} \rangle]$. We will use this notation in the sequel.

A (*non-strict*) *tree-adjointing grammar (nstag)*¹

is a tuple $H = (\Sigma, Z, E, I, \nu)$ such that Z is a finite set (of *names*), E is a finite subset of U_\ast (of *elementary trees*), I is a finite subset of U (of *initial trees*), and $\nu: E \rightarrow Z$ is a bijection.

Let $\zeta_1, \zeta_2 \in U$, and H be an nstag. We write $\zeta_1 \Rightarrow_H \zeta_2$ if there are $w \in \text{pos}(\zeta_1)$ and $e = \tilde{e}[\langle \tau, V', c', \mathbf{t} \rangle] \in E$ such that ζ_2 is the result of *adjoining* the elementary tree e at the node w of ζ_1 – i.e., we have $\zeta_1|_w = \langle \sigma, V, c, f \rangle(\xi_1, \dots, \xi_k)$ for some $\langle \sigma, V, c, f \rangle \in \Lambda$ with $\nu(e) \in V$, and $\zeta_2 = \zeta_1[\tilde{e}[\langle \tau, V', c', \mathbf{f} \rangle](\xi_1, \dots, \xi_k)]_w$. The *tree language of H* , denoted by $L(H)$, is the set of all $\xi \in U_\Sigma$ such that there are an $i \in I$ and a $\zeta \in U$ that contains only terminable nodes, with $i \Rightarrow_G^* \zeta$, and $\xi = \pi_1^A(\zeta)$, the node-wise projection of ζ to the node label.

3 Main Construction

We will now describe our method to construct an equivalent lm-cftg G from a given nstag H . As noted by Kepser and Rogers, the main obstacle in simulating nstag by lm-cftg is the fact that in a derivation step of H , an arbitrary number of trees can be adjoined to the foot node of an elementary tree, while lm-cftg only allow the substitution of *one* tree for the variable y in the right-hand side of a production. Consider the derivation of H that is portrayed in Fig. 1. This derivation takes place at an adjunction site labeled F with child trees ξ_1, \dots, ξ_k . In each step of the derivation, an adjunction site is chosen, and an elementary tree is adjoined at that site. In particular, if the adjunction operations up to that point contributed foot nodes that are operative, the respective adjunction site may again be the direct parent node of the trees ξ_1, \dots, ξ_k . In a successful derivation, however, eventually some elementary tree must contribute a terminable foot node, say some $\sigma \in \Lambda$ as in Fig. 1, which is not subject to any further adjunction operations, and becomes the parent node of ξ_1, \dots, ξ_k .

In our construction of G , we model the above by *guessing* for every adjunction site F of a sentential form ζ the terminal σ the children ξ_1, \dots, ξ_k of F will eventually end up with as a parent label. We then replace the k trees ξ_1, \dots, ξ_k in ζ by the tree $\sigma(\xi_1, \dots, \xi_k)$. Of course, in the ensuing derivation of G , this guess must be checked for correctness, so we encode the guess into a respective nonterminal (F, σ) of G , and propagate the information

operation are omitted, cf. Kepser and Rogers (2011). Our definition of nstag differs from *ibid.* in allowing initial trees without foot node. Clearly this difference is purely syntactical.

¹A tag is non-strict if certain restrictions on its adjoining

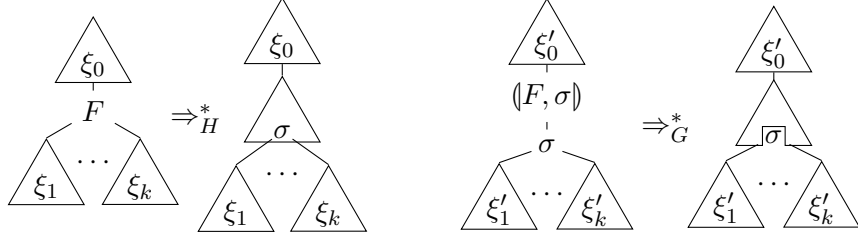


Figure 1: A derivation in H and its simulation in G .

during the course of the derivation. If σ is eventually adjoined as foot node, the guess was correct. As σ was already produced, we “cut out” the foot node of such elementary trees (as in Fig. 1) in the corresponding production of G , replacing it by y .

Theorem 1. *For every nstag H , there is a lm-cftg G such that $L(G) = L(H)$.*

Proof. Let $H = (\Sigma, Z, E, I, \nu)$ be an nstag, and let $N = \{S\} \cup \{(\sigma, V, c, \tau) \mid \sigma, \tau \in \Sigma, V \subseteq Z, c \in \mathbb{B}\}$. Let $\rho: U_\Lambda \rightarrow 2^{U_{N \cup \Sigma}}$ be a (nondeterministic) td-tt₁ with the rules $\ast(\langle \sigma, V, c, \mathbf{f} \rangle(x_1, \dots, x_k)) \rightarrow \langle \sigma, V, c, \delta \rangle(\delta(\ast(x_1), \dots, \ast(x_k)))$ for every $V \subseteq Z$, $\sigma, \delta \in \Sigma$, and $c \in \mathbb{B}$. Further, for every $\tau \in \Sigma$, let $\rho_\tau: U_\Lambda \rightarrow 2^{U_{N \cup \Sigma}(Y_1)}$ be a td-tt₁ that has every rule of ρ , as well as the rule $\ast(\langle \sigma, V, c, \mathbf{t} \rangle) \rightarrow \langle \sigma, V, c, \tau \rangle(y)$ for every $V \subseteq Z$, $\sigma \in \Sigma$, and $c \in \mathbb{B}$.

Construct the cftg $G = (N, \Sigma, S, P)$, where (i) for every $i \in I$ and $i' \in \rho(i)$, the production $S \rightarrow i'$ is in P ; (ii) for every $\langle \sigma, V, c, \mathbf{f} \rangle \in \Lambda$, $\tau \in \Sigma$, $e \in \nu^{-1}(V)$, and $e' \in \rho_\tau(e)$, we have that $\langle \sigma, V, c, \tau \rangle(y) \rightarrow e'$ is in P ; and (iii) for every $\sigma \in \Sigma$ and $V \subseteq Z$, P contains $\langle \sigma, V, \mathbf{f}, \sigma \rangle(y) \rightarrow y$.

We will demonstrate that $L(G) = L(H)$.

To show that $L(H) \subseteq L(G)$, we first prove that for every $n \in \mathbb{N}$, $i \in I$, $\zeta \in U$, and $\zeta' \in \rho(\zeta)$, whenever $i \Rightarrow_H^n \zeta$, then also $S \Rightarrow_G^* \zeta'$. The proof is by induction on n . If $n = 0$, then $\zeta = i$, thus by definition $S \Rightarrow_G \zeta'$. So assume there is $\eta \in U$ such that $i \Rightarrow_H^n \eta \Rightarrow_H \zeta$. Thus there are $\kappa \in U_\Lambda(X_1)$ linear and nondeleting in X_1 , $\zeta_1, \dots, \zeta_k \in U$, some $\langle \sigma, V, c, \mathbf{f} \rangle \in \Lambda$, and $e = \tilde{e}[\langle \delta, V', c', \mathbf{t} \rangle] \in E$ such that $\nu(e) \in V$, $\eta = \kappa[\langle \sigma, V, c, \mathbf{f} \rangle(\zeta_1, \dots, \zeta_k)]$, and $\zeta = \kappa[\tilde{e}[\langle \delta, V', c', \mathbf{f} \rangle(\zeta_1, \dots, \zeta_k)]]$. As $\zeta' \in \rho(\zeta)$, there must be some $\kappa' \in \rho(\kappa)$ and $\zeta'_i \in \rho(\zeta_i)$ for $1 \leq i \leq k$, $\tilde{e}' \in \rho(\tilde{e})$, and $\tau \in \Sigma$ such that $\zeta' = \kappa'[\tilde{e}'[\langle \delta, V', c', \tau \rangle(\tau(\zeta'_1), \dots, \zeta'_k)]]$. Thus also $\eta' \in \rho(\eta)$, where $\eta' = \kappa'[\langle \sigma, V, c, \tau \rangle(\tau(\zeta'_1), \dots, \zeta'_k))]$. By the induction hypothesis, $S \Rightarrow_G^* \eta'$. Furthermore, observe that $\tilde{e}'[\langle \sigma, V', c', \tau \rangle(y)] \in \rho_\tau(e)$, hence by construction of G , there is the production $\langle \sigma, V, c, \tau \rangle(y) \rightarrow e'[\langle \sigma, V', c', \tau \rangle(y)]$, and

therefore $\eta' \Rightarrow_G \zeta'$.

Now, let $\xi \in L(H)$. Thus there are $i \in I$ and $\zeta \in U$ such that $i \Rightarrow_H^* \zeta$, $\pi_1^4(\zeta) = \xi$, and all nodes of ζ are terminable. Consider the tree $\mu(\zeta)$, where $\mu: U_\Lambda \rightarrow U_{N \cup \Sigma}$ is a partial deterministic td-tt₁ with rules $\ast(\langle \sigma, V, \mathbf{f}, \mathbf{f} \rangle(x_1, \dots, x_k)) \rightarrow \langle \sigma, V, \mathbf{f}, \sigma \rangle(\sigma(\ast(x_1), \dots, \ast(x_k)))$ for every $\sigma \in \Sigma$, and $V \subseteq Z$. It is easy to see that $\mu(\zeta) \in \rho(\zeta)$, thus $S \Rightarrow_G^* \mu(\zeta)$. Moreover, $\mu(\zeta) \Rightarrow_G^* \xi$, by productions of type (iii). Hence $\xi \in L(G)$.

To show that $L(G) \subseteq L(H)$, we prove that for every $n \in \mathbb{N}$ and $\zeta' \in U_{N \cup \Sigma}$, whenever $S \Rightarrow_G^{n+1} \zeta'$ using only productions of type (i) or type (ii), there exist $i \in I$ and $\zeta \in U$ such that $i \Rightarrow_H^n \zeta$ and $\zeta' \in \rho(\zeta)$. The proof is by induction on n . If $n = 0$, then $S \rightarrow \zeta'$ is in P . By construction there is an $i \in I$ such that $\zeta' \in \rho(i)$, and thus $i \Rightarrow_H^0 \zeta$ with $\zeta = i$. So assume that there is some $\eta' \in U_{N \cup \Sigma}$ such that $S \Rightarrow_G^{n+1} \eta' \Rightarrow_G \zeta'$ using only productions of type (i) and (ii). As the last applied rule must be of type (ii), there are $\kappa' \in U_{N \cup \Sigma}(X_1)$ linear and non-deleting in X_1 , $\sigma, \tau \in \Sigma$, $V \subseteq Z$, $\bar{\zeta} \in U_{N \cup \Sigma}$, and $e' \in U_{N \cup \Sigma}(Y_1)$ such that the production $\langle \sigma, V, c, \tau \rangle(y) \rightarrow e'$ is in P , $\eta' = \kappa'[\langle \sigma, V, c, \tau \rangle(\bar{\zeta})]$, and $\zeta' = \kappa'[e'[\bar{\zeta}]]$. By induction, there exist $i \in I$ and $\eta \in U$ such that $i \Rightarrow_H^n \eta$ and $\eta' \in \rho(\eta)$. Thus, there are $\kappa \in U(X_1)$ linear and nondeleting in X_1 , $\zeta_1, \dots, \zeta_k \in U$ and $\zeta'_1 \in \rho(\zeta_1), \dots, \zeta'_k \in \rho(\zeta_k)$ such that $\eta = \kappa[\langle \sigma, V, c, \mathbf{f} \rangle(\zeta_1, \dots, \zeta_k)]$, $\kappa' \in \rho(\kappa)$, and $\bar{\zeta} = \tau(\zeta'_1, \dots, \zeta'_k)$. By construction, there is $e = \tilde{e}[\langle \delta, V', c', \mathbf{t} \rangle] \in E$ such that $e' \in \rho_\tau(e)$ and $\nu(e) \in V$. Then also $\eta \Rightarrow_H \zeta$, where $\zeta = \kappa[\tilde{e}[\langle \delta, V', c', \mathbf{f} \rangle(\zeta_1, \dots, \zeta_k)]]$. Thus, $i \Rightarrow_H^n \eta \Rightarrow \zeta$. As $e' \in \rho_\tau(e)$, there is $\tilde{e}' \in \rho(\tilde{e})$ such that $e' = \tilde{e}'[\langle \delta, V', c', \tau \rangle(y)]$ and $\zeta' = \kappa'[\tilde{e}'[\langle \delta, V', c', \tau \rangle(\tau(\zeta'_1), \dots, \zeta'_k))]]$, thus $\zeta' \in \rho(\zeta)$.

Let $\xi \in L(G)$, i.e. $\xi \in U_\Sigma$ and $S \Rightarrow_G^+ \xi$. Since G is linear and nondeleting, we may assume that there is $\zeta' \in U_{N \cup \Sigma}$ such that $S \Rightarrow_G^+ \zeta'$ using only productions of type (i) and (ii) and $\zeta' \Rightarrow_G^* \xi$ using only productions of type (iii). Thus, every nontermi-

nal (σ, V, c, τ) occurring in ζ' needs to be such that $\sigma = \tau$ and $c = \mathbf{f}$. As shown above, there are $i \in I$ and $\zeta \in U$ such that $i \Rightarrow_H^* \zeta$ and $\zeta' \in \rho(\zeta)$. In fact, $\zeta' = \mu(\zeta)$ and each node of ζ is terminable, with μ as before. Obviously, $\xi = \pi_1^4(\zeta) \in L(H)$. \square

4 Illustration

We will now illustrate our construction by a small example, and contrast its result to the outcome of the constructions applied by Kepser and Rogers.

Let $Z = \{1, 2\}$, $\Sigma = \{a, b\}$, and consider the set of initial trees $I = \{\langle b, \{1, 2\}, \mathbf{f}, \mathbf{f} \rangle\}$ and the set of elementary trees E that is displayed in Fig. 2.

The construction of Kepser and Rogers involves the following steps: (a) A tag H is transformed into an equivalent footed cftg G' . Then (b) G' is transformed into an equivalent spinal-formed cftg G'' , (c) for which a normal form exists that is linear and monadic, yielding the cftg G .² Regarding (a), footed cftg are the counterparts of tag in the world of cftg: a cftg is footed if it is linear and each of its productions' right-hand sides ξ has a unique node whose children are y_1, \dots, y_k , the variables occurring in ξ . In fact, in the construction of G' from H , these variables are just inserted below the foot nodes of its elementary trees. Every tag label becomes a nonterminal symbol.

As to (b), we briefly sketch the concept of a *spinal-formed* cftg, which can be found in Fujiyoshi and Kasai (2000). To every nonterminal symbol A , we associate a nonnegative integer $h(A)$, called the *head* of A , such that $h(A) \leq \chi$. Such an alphabet is said to be *head-pointing*. In every production $A(y_1, \dots, y_k) \rightarrow \xi$ of a spinal-formed cftg, the variable $y_{h(A)}$ must occur exactly once in ξ . The unique path from the root of ξ to $y_{h(A)}$ is called the *spine* (of ξ). There is the additional restriction for spinal-formed cftg that each variable in $Y_k \setminus \{y_{h(A)}\}$ occurs at least once in ξ , and only as the child of some node on the spine. Further, if some nonterminal B is located on the spine, then its $h(B)$ -th successor must be on the spine, too. Thus, to transform G' into a spinal-formed cftg G'' , we create copies of each nonterminal (and its productions) with different heads. Afterwards, the nonterminals in the right-hand sides of productions only need to be replaced by the respective copy.

Pertaining to (c), Fujiyoshi and Kasai (2000) present a method to transform any spinal-formed cftg into a linear and monadic normal form. After this transformation, the right-hand side of a production is either a non-branching tree of monadic nonterminals with leaf y , a single terminal, or of height 2. The crucial substeps involve replacing all subtrees beside the spine by new nullary nonterminals, guessing an assignment for each variable except the head variable, and finally, collapsing the leaves beside the spine into their neighboring nonterminals on the spine. The nondeterministic choice for the variables is propagated by extending the grammar's nonterminals. Applying (a)–(c) (and some simplifications) yields the lm-cftg G , whose productions are displayed in Fig. 3.

In contrast, our construction vertically expands every node of some initial or elementary tree by guessing a nonterminal and a terminal as described above. Thus, the overall structure of the trees is maintained. The useful productions of the grammar are depicted in Fig. 4. Obviously, our procedure can be refined by translating tag nodes that are non-operative and terminable just into terminals, thus avoiding useless nonterminals and productions.

²Formal definitions of these notions are omitted for brevity. We hope the comparison of the respective results stands for itself.

$$E = \left\{ \begin{array}{l} \text{1:} \quad \begin{array}{c} \langle a, \emptyset, \mathbf{f}, \mathbf{f} \rangle \\ \swarrow \quad \downarrow \\ \langle a, \emptyset, \mathbf{f}, \mathbf{f} \rangle \quad \langle a, \{1, 2\}, \mathbf{f}, \mathbf{f} \rangle \\ \quad \quad \quad \downarrow \\ \langle b, \emptyset, \mathbf{f}, \mathbf{t} \rangle \quad \langle a, \emptyset, \mathbf{f}, \mathbf{f} \rangle \end{array} \\ \text{2:} \quad \begin{array}{c} \langle a, \emptyset, \mathbf{f}, \mathbf{f} \rangle \\ \swarrow \quad \downarrow \\ \langle b, \emptyset, \mathbf{f}, \mathbf{f} \rangle \quad \langle a, \{1, 2\}, \mathbf{f}, \mathbf{f} \rangle \\ \quad \quad \quad \downarrow \\ \langle b, \emptyset, \mathbf{f}, \mathbf{t} \rangle \quad \langle b, \emptyset, \mathbf{f}, \mathbf{f} \rangle \end{array} \end{array} \right\}$$

Figure 2: Elementary trees and their names indicated to their left.

$$\begin{array}{c} \langle a, ay \rangle \quad \langle a, by \rangle \\ \downarrow \quad \downarrow \\ \langle (a, Z, \mathbf{f}), ya \rangle + \langle (a, Z, \mathbf{f}), yb \rangle + b \quad \langle \alpha, \beta y \rangle (y) \rightarrow \beta \swarrow \downarrow \begin{array}{c} \alpha \\ y \end{array} \langle \alpha, y\beta \rangle (y) \rightarrow \begin{array}{c} \alpha \\ y \end{array} \searrow \beta \\ \downarrow \quad \downarrow \\ \langle a, ay \rangle \quad \langle a, by \rangle \\ \downarrow \quad \downarrow \\ \langle (a, Z, \mathbf{f}), y\gamma \rangle (y) \rightarrow \langle (a, Z, \mathbf{f}), ya \rangle + \langle (a, Z, \mathbf{f}), yb \rangle + \langle a, y\gamma \rangle \\ \downarrow \quad \downarrow \quad \downarrow \\ \langle b, y\gamma \rangle \quad \langle b, y\gamma \rangle \quad y \end{array}$$

Figure 3: The lm-cftg obtained through the construction of Kepser and Rogers, where $\alpha, \beta, \gamma \in \{a, b\}$.

$$\begin{array}{c} S \rightarrow \begin{array}{c} \langle b, Z, \mathbf{f}, b \rangle \\ \downarrow \\ b \end{array} + \begin{array}{c} \langle b, Z, \mathbf{f}, a \rangle \\ \downarrow \\ a \end{array} \quad \langle \beta, V, \mathbf{f}, \beta \rangle (y) \rightarrow y \\ \downarrow \quad \downarrow \\ \langle a, \emptyset, \mathbf{f}, a \rangle \quad \langle a, \emptyset, \mathbf{f}, a \rangle \\ \downarrow \quad \downarrow \\ \langle \beta, \emptyset, \mathbf{f}, \beta \rangle \quad \langle a, Z, \mathbf{f}, b \rangle \quad \langle \beta, \emptyset, \mathbf{f}, \beta \rangle \quad \langle a, Z, \mathbf{f}, a \rangle \\ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ \langle \alpha, Z, \mathbf{f}, \gamma \rangle (y) \rightarrow \begin{array}{c} \langle b, \emptyset, \mathbf{f}, \beta \rangle \\ \downarrow \\ \langle b, \emptyset, \mathbf{f}, \gamma \rangle \end{array} \quad \langle \beta, \emptyset, \mathbf{f}, \beta \rangle \quad \langle b, \emptyset, \mathbf{f}, \gamma \rangle \quad \langle \beta, \emptyset, \mathbf{f}, \beta \rangle \\ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ y \quad \beta \quad y \quad \beta \end{array}$$

Figure 4: The lm-cftg obtained through our construction, where $\alpha, \beta, \gamma \in \{a, b\}$ and $V \subseteq Z$.

References

- A. Arnold and M. Dauchet. 1977. Un Théorème de Chomsky-Schützenberger dans le Cas des Forêts Algébriques. *Calcolo*, 14(2):161–184.
- J. Engelfriet. 1975. Bottom-Up and Top-Down Tree Transformations—A Comparison. *Mathematical Systems Theory*, 9(2):198–231.
- J. Engelfriet and E. M. Schmidt. 1977 and 1978. IO and OI. *Journal of Computer and System Sciences*, 15(3):328–353, 16(1):67–99.
- A. Fujiyoshi and T. Kasai. 2000. Spinal-Formed Context-Free Tree Grammars. *Theory of Computing Systems*, 33(1):59–83.
- Z. Fülöp and H. Vogler. 2009. Weighted Tree Automata and Tree Transducers. In M. Droste, W. Kuich, and H. Vogler, eds., *Handbook of Weighted Automata*, ch. 9, pp. 313–404. Springer.
- F. Gécseg and M. Steinby. 1984. *Tree Automata*. Akadémiai Kiadó.
- I. Guessarian. Pushdown Tree Automata. 1983. *Mathematical Systems Theory*, 16(1):237–263.
- M. Kanazawa. Multi-Dimensional Trees and a Chomsky-Schützenberger-Weir Representation Theorem for Simple Context-Free Tree Grammars. To appear in *Journal of Logic and Computation*. Published online June 30, 2014.
- S. Kepser and J. Rogers. 2011. The Equivalence of Tree Adjoining Grammars and Monadic Linear Context-Free Tree Grammars. *Journal of Logic, Language and Information*, 20(3):361–384.
- B. Leguy. 1981. Reducing Algebraic Tree Grammars. *Proc. of Fundamentals of Computation Theory*, LNCS, vol. 117, pp. 226–233.