Extended Multi Bottom-Up Tree Transducers

Joost Engelfriet[1], Eric Lilin[2], and Andreas Maletti[3,⋆]

[1] Leiden Institute of Advanced Computer Science
Leiden University, P.O. Box 9512, 2300 RA Leiden, The Netherlands
`engelfri@liacs.nl`

[2] Université des Sciences et Technologies de Lille
UFR IEEA 59655, Villeneuve d'Ascq, France
`eric.lilin@lifl.fr`

[3] International Computer Science Institute
1947 Center Street, Suite 600, Berkeley, CA 94704, USA
`maletti@icsi.berkeley.edu`

Abstract. Extended multi bottom-up tree transducers are defined and investigated. They are an extension of multi bottom-up tree transducers by arbitrary, not just shallow, left-hand sides of rules; this includes rules that do not consume input. It is shown that such transducers can compute any transformation that is computed by a linear extended top-down tree transducer. Moreover, the classical composition results for bottom-up tree transducers are generalized to extended multi bottom-up tree transducers. Finally, a characterization in terms of extended top-down tree transducers is presented.

1 Introduction

In the field of natural language processing, Knight [1, 2] proposed the following criteria that any reasonable formal tree-to-tree model of syntax-based machine translation [3] should fulfil:

(a) It should be a genuine generalization of finite-state transducers [4]; this includes the use of epsilon rules, i.e., rules that do not consume any part of the input tree.
(b) It should be efficiently trainable.
(c) It should be able to handle rotations (on the tree level).
(d) Its induced class of transformations should be closed under composition.

Graehl and Knight [5] proposed the linear and nondeleting extended (top-down) tree transducer (ln-xtt) [6, 7] as a suitable formal model. It fulfils (a)–(c) but fails to fulfil (d). Further models were proposed but, to the authors' knowledge, they all fail at least one criterion. Table 1 shows some important models and their properties.

⋆ Author was supported by a fellowship within the Postdoc-Programme of the German Academic Exchange Service (DAAD).

Table 1. Overview of formal models with respect to desired criteria. "x" marks fulfilment; "–" marks failure to fulfil. A question mark shows that this remains open though we conjecture fulfilment.

Model \ Criterion	(a)	(b)	(c)	(d)
Linear and nondeleting top-down tree transducer [8, 9]	–	x	–	x
Quasi-alphabetic tree bimorphism [10]	–	?	–	x
Synchronous context-free grammar [11]	x	x	–	x
Synchronous tree substitution grammar [12]	x	x	x	–
Synchronous tree adjoining grammar [13–15]	x	x	x	–
Linear and complete tree bimorphism [15]	x	x	x	–
Linear and nondeleting extended top-down tree transducer [5–7]	x	x	x	–
Linear multi bottom-up tree transducer [16–18]	–	?	x	x
Linear extended multi bottom-up tree transducer [*this paper*]	x	?	x	x

We propose a formal model that satisfies criteria (a), (c), and (d), and has more expressive power than the ln-xtt. The device is called linear extended multi bottom-up tree transducer, and it is as powerful as the linear model of [16–18] enhanced by epsilon rules (as shown in Theorem 5). In this paper we formally define and investigate the extended multi bottom-up tree transducer (xmbutt) and various restrictions (e.g., *linear*, *nondeleting*, and *deterministic*). Note that we consider the xmbutt in general, not just its linear restriction.

We start with normal forms for xmbutts. First, we construct for every xmbutt an equivalent nondeleting xmbutt (see Theorem 3). This can be achieved by guessing the required translations. Though the construction preserves linearity, it obviously destroys determinism. Next, we present a *one-symbol normal form* for xmbutts (see Theorem 5): each rule of the xmbutt either consumes one input symbol (without producing output), or produces one output symbol (without consuming input). This normal form preserves all three restrictions above.

Our main result (Theorem 13) states that the class of transformations computed by xmbutts is closed under pre-composition with transformations computed by linear xmbutts and under post-composition with those computed by deterministic xmbutts. In particular, we also obtain that the classes of transformations computed by linear and/or deterministic xmbutts are closed under composition. These results are analogous to classical results (see [19, Theorems 4.5 and 4.6] and [20, Corollary 7]) for bottom-up tree transducers [21, 19] and thus show the "bottom-up" nature of xmbutts. Also, they generalize the composition results of [18, Theorem 11]. As in [18], our proof essentially uses the principle set forth in [20, Theorem 6], but the one-symbol normal form allows us to present a very simple composition construction for xmbutts and verify that it is correct, provided that the first input transducer is linear or the second is deterministic. We observe here that the "extension" of a tree transducer model (or even just the addition of epsilon rules) can, in general, destroy closure under composition, as can be seen from the linear and nondeleting top-down tree transducer. This seems to be due to the non-existence of a one-symbol normal form in the top-down case.

We verify that linear xmbutts have sufficient power for syntax-based machine translation. This is because, as mentioned before (and shown in Theorem 8), they can simulate all ln-xtts. Thus, we have a lower bound to the power of linear xmbutts. In fact, even the composition closure of the class of transformations computed by ln-xtts is strictly contained in the class of transformations computed by linear xmbutts. Finally, we also present exact characterizations (Theorems 7 and 14): xmbutts are as powerful as compositions of an ln-xtt with a deterministic top-down tree transducer. In the linear case the latter transducer has the so-called single-use property [22–26], and similar results hold in the deterministic case. Thus, the composition of two extended top-down tree transducers forms an upper bound to the power of the linear xmbutt. As a side-result we obtain that linear xmbutts admit a semantics based on recognizable rule tree languages. This suggests that linear xmbutts also satisfy criterion (b).

2 Preliminaries

Let A, B, C be sets. A relation from A to B is a subset of $A \times B$. Let $\tau_1 \subseteq A \times B$ and $\tau_2 \subseteq B \times C$. The composition of τ_1 and τ_2 is the relation $\tau_1 \mathbin{;} \tau_2$ given by $\tau_1 \mathbin{;} \tau_2 = \{(a, c) \mid \exists b \in B \colon (a, b) \in \tau_1, (b, c) \in \tau_2\}$. This composition is lifted to classes of relations in the usual manner.

The nonnegative integers are denoted by $\mathbb{N}$ and $\{i \mid 1 \leq i \leq k\}$ is denoted by $[k]$. A ranked set is a set Σ of symbols with a relation $\mathrm{rk} \subseteq \Sigma \times \mathbb{N}$ such that $\{k \mid (\sigma, k) \in \mathrm{rk}\}$ is finite for every $\sigma \in \Sigma$. Commonly, we denote the ranked set only by Σ and the set of k-ary symbols of Σ by $\Sigma^{(k)} = \{\sigma \in \Sigma \mid (\sigma, k) \in \mathrm{rk}\}$. We also denote that $\sigma \in \Sigma^{(k)}$ by writing $\sigma^{(k)}$. Given two ranked sets Σ and Δ with associated rank relations rk_Σ and rk_Δ, respectively, the set $\Sigma \cup \Delta$ is associated the rank relation $\mathrm{rk}_\Sigma \cup \mathrm{rk}_\Delta$. A ranked set Σ is uniquely-ranked if for every $\sigma \in \Sigma$ there exists exactly one k such that $(\sigma, k) \in \mathrm{rk}$. For uniquely-ranked sets, we denote this k simply by $\mathrm{rk}(\sigma)$. An alphabet is a finite set, and a ranked alphabet is a ranked set Σ such that Σ is an alphabet.

Let Σ be a ranked set. The set of Σ-trees, denoted by T_Σ, is the smallest set T such that $\sigma(t_1, \ldots, t_k) \in T$ for every $k \in \mathbb{N}$, $\sigma \in \Sigma^{(k)}$, and $t_1, \ldots, t_k \in T$. We write α instead of $\alpha()$ if $\alpha \in \Sigma^{(0)}$. Let $\Gamma \subseteq \Sigma$ and $H \subseteq T_\Sigma$. By $\Gamma(H)$ we denote $\{\gamma(t_1, \ldots, t_k) \mid \gamma \in \Gamma^{(k)}, t_1, \ldots, t_k \in H\}$. Now, let Δ be a ranked set. We denote by $T_\Delta(H)$ the smallest set $T \subseteq T_{\Sigma \cup \Delta}$ such that $H \subseteq T$ and $\Delta(T) \subseteq T$.

Let $t \in T_\Sigma$. The set of positions of t, denoted by $\mathrm{pos}(t)$, is defined by $\mathrm{pos}(\sigma(t_1, \ldots, t_k)) = \{\varepsilon\} \cup \{iw \mid i \in [k], w \in \mathrm{pos}(t_i)\}$ for every $\sigma \in \Sigma^{(k)}$ and $t_1, \ldots, t_k \in T_\Sigma$. Note that we denote the empty string by ε and that $\mathrm{pos}(t) \subseteq \mathbb{N}^*$. Let $w \in \mathrm{pos}(t)$ and $u \in T_\Sigma$. The subtree of t that is rooted in w is denoted by $t|_w$, the symbol of t at w is denoted by $t(w)$, and the tree obtained from t by replacing the subtree rooted at w by u is denoted by $t[u]_w$. For every $\Gamma \subseteq \Sigma$ and $\sigma \in \Sigma$, let $\mathrm{pos}_\Gamma(t) = \{w \in \mathrm{pos}(t) \mid t(w) \in \Gamma\}$ and $\mathrm{pos}_\sigma(t) = \mathrm{pos}_{\{\sigma\}}(t)$.

Let $X = \{x_i \mid i \geq 1\}$ be a set of formal variables, each considered to have the unique rank 0. A tree $t \in T_\Sigma(X)$ is linear (respectively, nondeleting) in $V \subseteq X$ if $\mathrm{card}(\mathrm{pos}_v(t)) \leq 1$ (respectively, $\mathrm{card}(\mathrm{pos}_v(t)) \geq 1$) for every $v \in V$. The set of variables of t is $\mathrm{var}(t) = \{v \in X \mid \mathrm{pos}_v(t) \neq \emptyset\}$ and the sequence of variables is given by $\mathrm{yield}_X : T_\Sigma(X) \to X^*$ with $\mathrm{yield}_X(v) = v$ for every $v \in X$ and $\mathrm{yield}_X(\sigma(t_1, \ldots, t_k)) = \mathrm{yield}_X(t_1) \cdots \mathrm{yield}_X(t_k)$ for every $\sigma \in \Sigma^{(k)}$ and $t_1, \ldots, t_k \in T_\Sigma(X)$. A tree $t \in T_\Sigma(X)$ is normalized if $\mathrm{yield}_X(t) = x_1 \cdots x_m$ for some $m \in \mathbb{N}$. Every mapping $\theta : X \to T_\Sigma(X)$ is a substitution. We define the application of θ to a tree in $T_\Sigma(X)$ inductively by $v\theta = \theta(v)$ for every $v \in X$ and $\sigma(t_1, \ldots, t_k)\theta = \sigma(t_1\theta, \ldots, t_k\theta)$ for every $\sigma \in \Sigma^{(k)}$ and $t_1, \ldots, t_k \in T_\Sigma(X)$.

An *extended* (top-down) *tree transducer* (xtt, or *transducteur généralisé descendant*) [6, 7] is a tuple $M = (Q, \Sigma, \Delta, I, R)$ where Q is a uniquely-ranked alphabet such that $Q = Q^{(1)}$, Σ and Δ are ranked alphabets that are both disjoint with $Q \cup X$, $I \subseteq Q$, and R is a finite set of rules of the form $l \to r$ with $l \in Q(T_\Sigma(X))$ linear in X, and $r \in T_\Delta(Q(\mathrm{var}(l)))$. The xtt M is linear (respectively, nondeleting) if r is linear (respectively, nondeleting) in $\mathrm{var}(l)$ for every $l \to r \in R$. The semantics of the xtt is given by term rewriting. Let $\xi, \zeta \in T_\Delta(Q(T_\Sigma))$. We write $\xi \Rightarrow_M \zeta$ if there exist a rule $l \to r \in R$, a position $w \in \mathrm{pos}(\xi)$, and a substitution $\theta : X \to T_\Sigma$ such that $\xi|_w = l\theta$ and $\zeta = \xi[r\theta]_w$. The tree transformation computed by M is the relation

$$\tau_M = \{(t, u) \in T_\Sigma \times T_\Delta \mid \exists q \in I : q(t) \Rightarrow_M^* u\} \enspace .$$

The class of all tree transformations computed by xtts is denoted by XTOP. We use ‘l’ and ‘n’ to restrict to linear and nondeleting devices, respectively. Thus, ln-XTOP denotes the class of all tree transformations computable by linear and nondeleting xtts.

An xtt $M = (Q, \Sigma, \Delta, I, R)$ is a *top-down tree transducer* [8, 9] if for every rule $l \to r \in R$ there exist $q \in Q$ and $\sigma \in \Sigma^{(k)}$ such that $l = q(\sigma(x_1, \ldots, x_k))$. The top-down tree transducer M is *deterministic* if (i) $\mathrm{card}(I) = 1$ and (ii) for every $l \in Q(\Sigma(X))$ there exists at most one r such that $l \to r \in R$. Finally, M is *single-use* [22–25] if for every $q(v) \in Q(X)$, $k \in \mathbb{N}$, and $\sigma \in \Sigma^{(k)}$ there exist at most one $l \to r \in R$ and $w \in \mathrm{pos}(r)$ such that $l(1) = \sigma$ and $r|_w = q(v)$. We use TOP and $\mathrm{TOP}_{\mathrm{su}}$ to denote the classes of transformations computed by top-down tree transducers and single-use top-down tree transducers, respectively. We also use the prefixes ‘l’, ‘n’, and ‘d’ to restrict to linear, nondeleting, and deterministic devices, respectively.

Finally, we recall top-down tree transducers with regular look-ahead [27]. We use the standard notion of a *recognizable* (or *regular*) tree language [28, 29], and we let $\mathrm{Rec}(\Sigma) = \{L \subseteq T_\Sigma \mid L \text{ recognizable}\}$. A *top-down tree transducer with regular look-ahead* is a pair $\langle M, c\rangle$ such that $M = (Q, \Sigma, \Delta, I, R)$ is a top-down tree transducer and $c : R \to \mathrm{Rec}(\Sigma)$. We say that such a transducer $\langle M, c\rangle$ is *deterministic* if (i) $\mathrm{card}(I) = 1$ and (ii) for every $l \in Q(\Sigma(X))$ and $t \in T_\Sigma$ there exists at most one r such that $l \to r \in R$, $l(1) = t(\varepsilon)$, and $t \in c(l \to r)$. Similarly, $\langle M, c\rangle$ is *single-use* [26, Definition 5.5] if for every $q(v) \in Q(X)$ and $t \in T_\Sigma$ there exist at most one $l \to r \in R$ and $w \in \mathrm{pos}(r)$ such that $l(1) = t(\varepsilon)$, $t \in c(l \to r)$, and $r|_w = q(v)$. The semantics of $\langle M, c\rangle$ is defined in the same manner as for xtt with the additional restriction that $l\theta|_1 \in c(l \to r)$. We use $\mathrm{TOP}^{\mathrm{R}}$ and $\mathrm{TOP}^{\mathrm{R}}_{\mathrm{su}}$ to denote the classes of transformations computed by top-down tree transducers with regular look-ahead and single-use top-down tree transducers with regular look-ahead, respectively. We use the prefix 'd' in the usual manner. For further information on tree languages and tree transducers, we refer to [28, 29].

3 Extended Multi Bottom-up Tree Transducers

In this section, we define *S-transducteurs ascendants généralisés* [30], which are a generalization of *S-transducteurs ascendants* (STA) [30, 31]. We choose to call them *extended multi bottom-up tree transducers* here in line with [16–18], where 'multi' refers to the fact that states may have ranks different from one.

Definition 1. *An* extended multi bottom-up tree transducer *(xmbutt) is a tuple* $(Q, \Sigma, \Delta, F, R)$ *where*

- Q *is a uniquely-ranked alphabet of* states*, disjoint with* $\Sigma \cup \Delta \cup X$*;*
- Σ *and* Δ *are ranked alphabets of* input *and* output symbols*, respectively, which are both disjoint with* X*;*
- $F \subseteq Q \setminus Q^{(0)}$ *is a set of* final states*; and*
- R *is a finite set of* rules *of the form* $l \to r$ *where* $l \in T_\Sigma(Q(X))$ *is linear in* X *and* $r \in Q(T_\Delta(\mathrm{var}(l)))$*.*

A rule $l \to r \in R$ *is an* epsilon rule *if* $l \in Q(X)$*; otherwise it is* input-consuming*. The sets of epsilon and input-consuming rules are denoted by* R^ε *and* R^Σ*, respectively.*

An xmbutt $M = (Q, \Sigma, \Delta, F, R)$ is a *multi bottom-up tree transducer* (mbutt) [respectively, an STA] if $l \in \Sigma(Q(X))$ [respectively, $l \in \Sigma(Q(X)) \cup Q(X)$] for every $l \to r \in R$. Linearity and nondeletion of xmbutts are defined in the natural manner. The xmbutt M is *linear* if r is linear in $\mathrm{var}(l)$ for every rule $l \to r \in R$. Moreover, M is *nondeleting* if (i) $F \subseteq Q^{(1)}$ and (ii) r is nondeleting in $\mathrm{var}(l)$ for every $l \to r \in R$. Finally, M is *deterministic* if (i) there do not exist two distinct rules $l_1 \to r_1 \in R$ and $l_2 \to r_2 \in R$, a substitution $\theta \colon X \to X$, and $w \in \mathrm{pos}(l_2)$ such that $l_1\theta = l_2|_w$, and (ii) there does not exist an epsilon rule $l \to r \in R$ such that $l(\varepsilon) \in F$. Let us now present a rewrite semantics. In the rest of this section, let $M = (Q, \Sigma, \Delta, F, R)$ be an xmbutt.

Definition 2. *Let* Σ' *and* Δ' *be ranked alphabets disjoint with* Q*. Moreover, let* $\xi, \zeta \in T_{\Sigma\cup\Sigma'}(Q(T_{\Delta\cup\Delta'}))$*, and* $l \to r \in R$*. We write* $\xi \Rightarrow_M^{l\to r} \zeta$ *if there exist* $w \in \mathrm{pos}(\xi)$ *and* $\theta \colon X \to T_{\Delta\cup\Delta'}$ *such that* $\xi|_w = l\theta$ *and* $\zeta = \xi[r\theta]_w$*, and we write* $\xi \Rightarrow_M \zeta$ *if there exists* $\rho \in R$ *such that* $\xi \Rightarrow_M^\rho \zeta$*. The* tree transformation computed by M *is* $\tau_M = \{(t, \xi|_1) \in T_\Sigma \times T_\Delta \mid \xi \in F(T_\Delta), t \Rightarrow_M^* \xi\}$*.*

The xmbutt M' is *equivalent* to M if $\tau_{M'} = \tau_M$. We denote by XMBOT the class of tree transformations computed by xmbutts. We use the prefixes 'l', 'n', and 'd' to restrict to linear, nondeleting, and deterministic devices, respectively. For example, l-XMBOT denotes the class of all tree transformations computed by linear xmbutts.

If M is deterministic and $t \in T_\Sigma$, then there exists at most one $\xi \in Q(T_\Delta)$ such that (i) $t \Rightarrow_M^* \xi$ and (ii) there exists no ζ such that $\xi \Rightarrow_M \zeta$. Hence, τ_M is a partial function, if M is deterministic. Moreover, if M is a deterministic mbutt and $t \in T_\Sigma$, then there exists at most one $\xi \in Q(T_\Delta)$ such that $t \Rightarrow_M^* \xi$. A deterministic mbutt M is *total* if for every $t \in T_\Sigma$ there exists $\xi \in Q(T_\Delta)$ such that $t \Rightarrow_M^* \xi$ (an equivalent static definition is easy to formulate).

Our first result shows that every xmbutt is equivalent to a nondeleting one. Unfortunately, the construction does not preserve determinism.

Theorem 3. XMBOT = n-XMBOT *and* l-XMBOT = ln-XMBOT.

Proof. For the xmbutt $M = (Q, \Sigma, \Delta, F, R)$ we construct an equivalent nondeleting xmbutt $M' = (Q', \Sigma, \Delta, F', R')$. The idea is that M' simulates M but guesses at each moment which subtrees of the states will be deleted in the remainder of M's computation. The set Q' of states of M' consists of all pairs $\langle q, J\rangle$ with $q \in Q^{(k)}$ and $J \subseteq [k]$, and the rank of $\langle q, J\rangle$ is $\mathrm{card}(J)$; moreover, $F' = \{\langle q, \{1\}\rangle \mid q \in F\}$. The rules of M' are constructed such that $t \Rightarrow_{M'}^* \langle q, J\rangle(u_{i_1}, \ldots, u_{i_m})$, where $J = \{i_1, \ldots, i_m\}$ and $i_1 < \cdots < i_m$, if and only if there exist $u_i \in T_\Delta$ for every $i \in [k] \setminus J$ such that $t \Rightarrow_M^* q(u_1, \ldots, u_k)$. □

The following normal form will be at the heart of our composition construction in the next section. It says that exactly one input or output symbol occurs in every rule (such rules will be called *one-symbol rules*).

Definition 4. *The xmbutt M is in* one-symbol normal form *if for every rule $l \to r \in R$ we have* $\mathrm{card}(\mathrm{pos}_\Sigma(l)) + \mathrm{card}(\mathrm{pos}_\Delta(r)) = 1$.

Theorem 5. *For every xmbutt M there exists an equivalent xmbutt N in one-symbol normal form. Moreover, if M is linear (respectively, nondeleting, deterministic), then so is N.*

Proof. Let us assume, without loss of generality, that all left-hand sides of rules of M are normalized. First we take care of the left-hand sides of rules and decompose rules with more than one input symbol in the left-hand side into several rules, cf. [30, Proposition II.B.5]. Take a uniquely-ranked set P and a bijection $f \colon T_\Sigma(Q(X)) \to P$ such that (i) $Q \subseteq P$, (ii) $f(q(x_1, \ldots, x_n)) = q$ for every $q \in Q^{(n)}$, and (iii) $\mathrm{rk}(f(l)) = \mathrm{card}(\mathrm{var}(l))$ for every $l \in T_\Sigma(Q(X))$. In fact, we will only use $f(l)$ for normalized $l \in T_\Sigma(Q(X))$.

Let $l \to r \in R$ be input-consuming such that $l \notin \Sigma(P(X))$. Suppose that $l = \sigma(l_1, \ldots, l_k)$ for some $\sigma \in \Sigma^{(k)}$ and $l_1, \ldots, l_k \in T_\Sigma(Q(X))$. Moreover, let $\theta_1, \ldots, \theta_k \colon X \to X$ be bijections such that $l_i\theta_i$ is normalized. Finally, for every $i \in [k]$ let $p_i = f(l_i\theta_i)$ and $r_i = p_i(x_1, \ldots, x_m)$ where $m = \mathrm{rk}(p_i)$. We construct the xmbutt $M_1 = (Q \cup \{p_1, \ldots, p_k\}, \Sigma, \Delta, F, (R \setminus \{l \to r\}) \cup R_{1,1} \cup R_{1,2})$ where $R_{1,1} = \{l_i\theta_i \to r_i \mid i \in [k], l_i \notin Q(X)\}$ and $R_{1,2} = \{l' \to r\}$, in which l' is the unique normalized tree of $\{\sigma\}(P(X))$ such that $l'(i) = p_i$ for every $i \in [k]$ (note that if $l_i \in Q(X)$, then $p_i = l_i(\varepsilon)$ and so $l'|_i = l_i$). Repeated application of this construction (keeping P and the mapping f fixed) eventually yields an equivalent xmbutt $M' = (Q', \Sigma, \Delta, F, R')$ such that $l \in \Sigma(P(X))$ for each input-consuming rule $l \to r \in R'$.

Next, we remove all epsilon rules $l \to r \in R'$ such that $r \in Q'(X)$ in the standard way. Finally, we decompose the right-hand sides. Let $M'' = (S, \Sigma, \Delta, F, R'')$ be the xmbutt obtained so far and $l \to r \in R''$ a rule that is not yet a one-symbol rule. Let $r = s(u_1, \ldots, u_{i-1}, \delta(u'_1, \ldots, u'_k), u_{i+1}, \ldots, u_n)$ for some $s \in S^{(n)}$, $i \in [n]$, $\delta \in \Delta^{(k)}$, and $u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_n, u'_1, \ldots, u'_k \in T_\Delta(X)$. Also, let $q \notin S$ be a new state of rank $k + n - 1$. We construct the xmbutt $M''_1 = (S \cup \{q\}, \Sigma, \Delta, F, R''_1)$ with $R''_1 = (R'' \setminus \{l \to r\}) \cup R''_{1,1}$ where $R''_{1,1}$ contains the two rules:

- $l \to q(u_1, \ldots, u_{i-1}, u'_1, \ldots, u'_k, u_{i+1}, \ldots, u_n)$ and
- $q(x_1, \ldots, x_{k+n-1}) \to s(x_1, \ldots, x_{i-1}, \delta(x_i, \ldots, x_{i+k-1}), x_{i+k}, \ldots, x_{k+n-1})$.

Repeated application of the procedure yields the desired xmbutt N. □

Example 6. Let $(Q, \Sigma, \Gamma, Q, R)$ be the xmbutt with $Q = \{q^{(1)}\}$, $\Sigma = \{\sigma^{(1)}, \alpha^{(0)}\}$, $\Gamma = \{\gamma^{(2)}, \alpha^{(0)}\}$, and $R = \{\sigma(\alpha) \to q(\alpha),\ \sigma(q(x_1)) \to q(\gamma(x_1, \alpha))\}$. Clearly, it is linear, nondeleting, and deterministic. Applying the procedure of Theorem 5 we obtain the states $q^{(1)}, q_1^{(0)}, q_2^{(0)}, q_3^{(2)}, q_4^{(1)}$ and the rules $\alpha \to q_1,\ \sigma(q_1) \to q_2$, $q_2 \to q(\alpha),\ \sigma(q(x_1)) \to q_4(x_1),\ q_4(x_1) \to q_3(x_1, \alpha),\ q_3(x_1, x_2) \to q(\gamma(x_1, x_2))$.

In the deterministic case we have an additional normal form: the deterministic mbutt. This allows us to characterize the classes d-XMBOT and ld-XMBOT in terms of top-down tree transducers, using the result of [16].

Theorem 7. *For every deterministic xmbutt M there exists an equivalent total deterministic mbutt N. Moreover, if M is linear, then so is N. Consequently,* d-XMBOT = d-TOP$^{\mathrm{R}}$ *and* ld-XMBOT = d-TOP$^{\mathrm{R}}_{\mathrm{su}}$.

Proof. Applying the first construction in the proof of Theorem 5 and then removing all epsilon rules in the usual way, we obtain an equivalent deterministic mbutt N. Obviously, by introducing a dummy state of rank 0, N can be made total. To obtain a deterministic multi bottom-up tree transducer of [16] we also have to add a special root symbol and add rules that, while consuming the special root symbol, project on the first argument of a final state. It is proved in [16] that such deterministic multi bottom-up tree transducers have the same power as deterministic top-down tree transducers with regular look-ahead. The second equality was already suggested in the Conclusion of [17]. We prove it by reconsidering (a minor variation of) the proofs of [16, Lemmata 4.1 and 4.2]. If the mbutt is linear, then the corresponding top-down tree transducer with regular look-ahead will be single-use and vice versa. □

Finally, we verify that l-XMBOT is suitably powerful for applications in machine translation. We do this by showing that all transformations of l-XTOP are also in l-XMBOT. This shows that xmbutts can handle rotations [2].

Theorem 8. l-XTOP ⊂ l-XMBOT.

Proof. The inclusion can be proved in a similar manner as l-TOP ⊆ l-BOT [19, Theorem 2.8], where l-BOT denotes the class of transformations computed by linear bottom-up tree transducers [21, 19]. Every transformation of l-XTOP preserves recognizability [18, Theorem 4], but there is a linear (deterministic) mbutt that computes the transformation $\{(\sigma(t), \delta(t, t)) \mid t \in T_{\Sigma \setminus \{\sigma\}}\}$ where $\sigma \in \Sigma^{(1)}$ and $\delta \in \Delta^{(2)}$. Hence, not every transformation of l-XMBOT preserves recognizability. □

4 Composition Construction

In this section, we investigate compositions of tree transformations computed by xmbutts. Let us first recall the classical composition results for bottom-up tree transducers [19, 20]. Let M and N be bottom-up tree transducers. If M is linear or N deterministic, then the composition of the transformations computed by M and N can be computed by a bottom-up tree transducer. As a special case, the classes of transformations computed by linear, linear and nondeleting, and deterministic bottom-up tree transducers are closed under composition.

In our setting, let M and N be xmbutts. We will prove that if M is linear or N is deterministic, then there is an xmbutt $M ; N$ that computes $\tau_M ; \tau_N$. In particular, we prove that l-XMBOT, d-XMBOT, and ld-XMBOT are closed under composition. The closure of l-XMBOT was first presented in [30, Propositions II.B.5 and II.B.7]. The closure of d-XMBOT is also immediate from Theorem 7 and [27, Theorem 2.11]; in [31, Proposition 2.5] it was shown for a different notion of determinism. The closure of ld-XMBOT is to be expected from Theorem 7 and the fact that the single-use restriction was introduced in [22, 23] to guarantee the closure under composition of attribute grammar transformations (see [25, Theorem 3]).

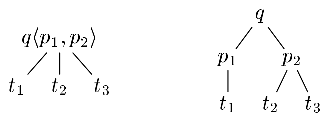

Fig. 1. Tree homomorphism φ where $q \in Q^{(2)}$, $p_1 \in P^{(1)}$, and $p_2 \in P^{(2)}$.

Let us prepare the composition construction. Let $M = (Q, \Sigma, \Gamma, F_M, R_M)$ and $N = (P, \Gamma, \Delta, F_N, R_N)$ be xmbutts such that Q, P, and $\Sigma \cup \Gamma \cup \Delta$ are pairwise disjoint. We define the uniquely-ranked alphabet

$$Q\langle P\rangle = \{q\langle p_1, \dots, p_n\rangle \mid q \in Q^{(n)}, p_1, \dots, p_n \in P\}$$

such that $\mathrm{rk}(q\langle p_1, \dots, p_n\rangle) = \sum_{i=1}^{n} \mathrm{rk}(p_i)$ for every $q \in Q^{(n)}$ and $p_1, \dots, p_n \in P$. Let $\Pi = \Sigma \cup \Gamma \cup \Delta \cup X$. We define the mapping $\varphi \colon T_{\Pi \cup Q\langle P\rangle} \to T_{\Pi \cup Q \cup P}$ such that for every $q\langle p_1, \dots, p_n\rangle \in Q\langle P\rangle^{(k)}$, $\pi \in \Pi^{(k)}$, and $t_1, \dots, t_k \in T_{\Pi \cup Q\langle P\rangle}$

$$\begin{aligned}\varphi(q\langle p_1, \dots, p_n\rangle(t_1, \dots, t_k)) &= q(p_1(\varphi(t_1), \dots, \varphi(t_l)), \dots, p_n(\varphi(t_m), \dots, \varphi(t_k)))\\ \varphi(\pi(t_1, \dots, t_k)) &= \pi(\varphi(t_1), \dots, \varphi(t_k))\end{aligned}$$

where $l = \mathrm{rk}(p_1)$ and $m = k - \mathrm{rk}(p_n) + 1$. Thus, we group the subtrees below the corresponding state p_i (see Fig. 1). Note that φ is a linear and nondeleting tree homomorphism, which acts as a bijection from $T_\Sigma(Q\langle P\rangle(T_\Delta(X)))$ to $T_\Sigma(Q(P(T_\Delta(X))))$. In the sequel, we will identify t with $\varphi(t)$ for all trees $t \in T_\Sigma(Q\langle P\rangle(T_\Delta(X)))$.

Definition 9. *Let $M = (Q, \Sigma, \Gamma, F_M, R_M)$ be an xmbutt in one-symbol normal form and $N = (P, \Gamma, \Delta, F_N, R_N)$ an STA. Moreover, let* $\mathrm{LHS}(\Sigma)$ *and* $\mathrm{LHS}(\varepsilon)$ *be the sets of normalized trees of $\Sigma(Q\langle P\rangle(X))$ and $Q\langle P\rangle(X)$, respectively. The* composition *$M\,;N = (Q\langle P\rangle, \Sigma, \Delta, F_M\langle F_N\rangle, R)$ of M and N is the STA with $R = R_1 \cup R_2 \cup R_3$ where:*

$$\begin{aligned}
R_1 &= \{l \to r \mid l \in \mathrm{LHS}(\Sigma) \text{ and } \exists \rho \in R_M^{\Sigma}\colon l \Rightarrow_M^{\rho} r\},\\
R_2 &= \{l \to r \mid l \in \mathrm{LHS}(\varepsilon) \text{ and } \exists \rho \in R_N^{\varepsilon}\colon l \Rightarrow_N^{\rho} r\}, \text{ and}\\
R_3 &= \{l \to r \mid l \in \mathrm{LHS}(\varepsilon) \text{ and } \exists \rho_1 \in R_M^{\varepsilon}, \rho_2 \in R_N^{\Gamma}\colon l\ (\Rightarrow_M^{\rho_1}\,;\Rightarrow_N^{\rho_2})\ r\}.
\end{aligned}$$

To illustrate the implicit use of φ, let us show the “official” definition of R_1:

$$R_1 = \{l \to r \mid l \in \mathrm{LHS}(\Sigma), r \in Q\langle P\rangle(T_\Delta(X)), \text{ and } \exists \rho \in R_M^{\Sigma}\colon \varphi(l) \Rightarrow_M^{\rho} \varphi(r)\} \ .$$

The construction preserves linearity; moreover, it preserves determinism if N is an mbutt. In the rest of this section we investigate when $\tau_{M;N} = \tau_M\,;\tau_N$, but we first illustrate the construction on our small running example.

Example 10. Let M be the xmbutt of Example 6 in one-symbol normal form, and let $N = (\{g^{(1)}, h^{(1)}\}, \Gamma, \Delta, \{g\}, R_N)$ be the STA with $\Delta = \Gamma \cup \{\delta^{(1)}\}$ and

$$R_N = \{\alpha \to h(\alpha),\ h(x_1) \to h(\delta(x_1)),\ \gamma(h(x_1), h(x_2)) \to g(\gamma(x_1, x_2))\} \ .$$

Clearly, N computes $\{(\gamma(\alpha,\alpha), \gamma(\delta^i(\alpha), \delta^j(\alpha)) \mid i, j \in \mathbb{N}\}$, and hence $\tau_M\,;\tau_N$ is $\{(\sigma(\sigma(\alpha)), \gamma(\delta^i(\alpha), \delta^j(\alpha)) \mid i, j \in \mathbb{N}\}$. The states of $M\,;N$ will be

$$\{q\langle g\rangle^{(1)}, q\langle h\rangle^{(1)}, q_1\langle\rangle^{(0)}, q_2\langle\rangle^{(0)}, q_3\langle g, g\rangle^{(2)}, \ldots, q_3\langle h, h\rangle^{(2)}, q_4\langle g\rangle^{(1)}, q_4\langle h\rangle^{(1)}\} \ ,$$

of which only $q\langle g\rangle$ is final. We present some relevant rules only [left in official form $l \to r$; right in alternative notation $\varphi(l) \to \varphi(r)$].

$$\begin{array}{rr}
\alpha \to q_1\langle\rangle & \alpha \to q_1\\
\sigma(q_1\langle\rangle) \to q_2\langle\rangle & \sigma(q_1) \to q_2\\
q_2\langle\rangle \to q\langle h\rangle(\alpha) & q_2 \to q(h(\alpha))\\
q\langle h\rangle(x_1) \to q\langle h\rangle(\delta(x_1)) & q(h(x_1)) \to q(h(\delta(x_1)))\\
\sigma(q\langle h\rangle(x_1)) \to q_4\langle h\rangle(x_1) & \sigma(q(h(x_1))) \to q_4(h(x_1))\\
q_4\langle h\rangle(x_1) \to q_3\langle h, h\rangle(x_1, \alpha) & q_4(h(x_1)) \to q_3(h(x_1), h(\alpha))\\
q_3\langle h, h\rangle(x_1, x_2) \to q\langle g\rangle(\gamma(x_1, x_2)) & q_3(h(x_1), h(x_2)) \to q(g(\gamma(x_1, x_2)))
\end{array}$$

The first, second, and fifth rules are in R_1 (of Definition 9), the fourth rule is in R_2, and the remaining rules in R_3. □

Next, we will prove that $\tau_M\,;\tau_N$ is in XMBOT provided that (i) M is linear or (ii) N is deterministic. We can assume that M is in one-symbol normal form, by Theorem 5, and that it is nondeleting in case (i), by Theorem 3. We can also assume that N is an STA in case (i), by Theorem 5, and a total deterministic mbutt in case (ii), by Theorem 7. Thus, we meet the requirements of Definition 9 and henceforth assume its notation.

We start with a simple lemma. It shows that in a derivation that uses steps of M and N (like the derivations of $M\,;N$) we can always perform all steps of M first and only then perform the derivation steps of N. This already proves one direction needed for the correctness of the composition construction.

Lemma 11. *Let $t \in T_\Sigma$ and $\xi \in Q(P(T_\Delta))$. If $t \Rightarrow^* \xi$ where $\Rightarrow$ is $\Rightarrow_M \cup \Rightarrow_N$, then $t\ (\Rightarrow_M^* \ ; \Rightarrow_N^*)\ \xi$. In particular, $\tau_{M;N} \subseteq \tau_M \ ; \tau_N$.*

Proof. It obviously suffices to prove: For every $\xi, \zeta \in T_\Sigma(Q(T_\Gamma(P(T_\Delta))))$, if $\xi\ (\Rightarrow_N \ ; \Rightarrow_M)\ \zeta$, then $\xi\ (\Rightarrow_M \ ; \Rightarrow_N^*)\ \zeta$. Its proof is easy. $\square$

Next we prove that $\tau_M\,;\tau_N \subseteq \tau_{M;N}$ under the above assumptions on M and N, by a standard induction over the length of the derivation.

Lemma 12. *Let $t \in T_\Sigma$ and $\xi \in Q(P(T_\Delta))$ be such that $t\ (\Rightarrow_M^* \ ; \Rightarrow_N^*)\ \xi$. If (i) M is linear and nondeleting, or (ii) N is a total deterministic mbutt, then $t \Rightarrow_{M;N}^* \xi$. With $\xi \in F_M(F_N(T_\Delta))$ we obtain $\tau_M\,;\tau_N \subseteq \tau_{M;N}$.*

Theorem 13. *The three classes* l-XMBOT*,* d-XMBOT*, and* ld-XMBOT *are closed under composition. Moreover,*

$$\text{l-XMBOT} \ ; \ \text{XMBOT} \subseteq \text{XMBOT} \qquad \textit{and} \qquad \text{XMBOT} \ ; \ \text{d-XMBOT} \subseteq \text{XMBOT} \ .$$

Proof. The inequalities follow directly from Lemmata 11 and 12 using Theorems 3, 5, and 7 to establish the preconditions of Definition 9 and Lemma 12. The closure results follow from the fact that the composition construction preserves linearity and determinism. $\square$

5 Relation to Top-down Tree Transducers

Now, let us focus on an upper bound to the power of xmbutts. By [18, Theorem 14] every mbutt computes a transformation of ln-TOP ; d-TOP. Here we prove a similar result for xmbutts.

Theorem 14.

$$\text{l-XMBOT} = \text{ln-XTOP} \ ; \ \text{d-TOP}_{\text{su}} \qquad \textit{and} \qquad \text{XMBOT} = \text{ln-XTOP} \ ; \ \text{d-TOP} \ .$$

Proof. By Theorems 7, 8, and 13, the inclusions $\supseteq$ are immediate. For the decomposition results, we employ the standard idea of separating the input and output behavior of the given xmbutt M (cf. [19, Theorem 3.15]). For each input tree, the first xtt M_1 outputs “rule trees” that encode which rules could be applied. The second xtt M_2 then deterministically executes these rules and creates the output. Linearity of M implies that M_2 is single-use. More formally, if $t \Rightarrow_M^* q(u_1, \dots, u_m)$, then there is a “rule tree” $\tilde{t}$ such that $q(t) \Rightarrow_{M_1}^* \tilde{t}$ and $n(\tilde{t}) \Rightarrow_{M_2}^* u_n$ for every $n \in [m]$. $\square$

We note that the direction $\subseteq$ of all four equalities in Theorems 7 and 14 is proved in essentially the same manner. If we compare the deterministic (Theorem 7) to the nondeterministic case (Theorem 14), then in the former case there exists at most one successful "rule tree", which can be constructed by a deterministic, finite-state bottom-up relabeling [19, 27] and has the shape of the input tree. Thus, the deterministic top-down tree transducer can query its look-ahead for the rule that labels its current position in the successful "rule tree".

By Theorem 14, the power of xmbutts is limited by two extended top-down tree transducers. In particular, the first equation shows in a precise way how much stronger l-XMBOT is with respect to ln-XTOP. But Theorem 14 also shows that we can separate nondeterminism and state checking (performed by the linear and nondeleting xtt) from evaluation (performed by the deterministic top-down tree transducer). Since linear xtts preserve recognizability, the *rule trees* mentioned in the proof of Theorem 14 form a recognizable tree language. Formally, the set $\{u \in T_R \mid \exists t \in T_\Sigma \colon (t, u) \in \tau_{M_1}\}$ is recognizable, which shows that the set of rule trees of an xmbutt is recognizable. This is a strong indication toward the existence of efficient training algorithms.

Conclusion and Open Problems We have shown that xmbutts are suitably powerful to compute any transformation that can be computed by linear extended top-down tree transducers (see Theorem 8). Moreover, we generalized the main composition results of [19, 20] for bottom-up tree transducers to xmbutts (see Theorem 13). In particular, we showed that l-XMBOT and d-XMBOT are closed under composition. Finally, we characterized XMBOT as the composition of ln-XTOP and d-TOP (see Theorem 14), which shows that, analogously to bottom-up tree transducers, nondeterminism and evaluation can be separated.

Since linear xmbutts do not necessarily preserve recognizability whereas linear xtts do, it is clear that even the composition closure of l-XTOP is strictly contained in l-XMBOT. This raises two questions: (a) Which xmbutts can be transformed into an xtt? (b) Can we characterize the composition closure of l-XTOP?

References

1. Knight, K.: Criteria for reasonable syntax-based translation models. Personal communication (2007)
2. Knight, K., Graehl, J.: An overview of probabilistic tree transducers for natural language processing. In: CICLing. Volume 3406 of LNCS, Springer (2005) 1–24
3. DeNeefe, S., Knight, K., Wang, W., Marcu, D.: What can syntax-based MT learn from phrase-based MT? In: EMNLP & CoNLL. (2007) 755–763
4. Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation. Addison Wesley (1979)
5. Graehl, J., Knight, K.: Training tree transducers. In: HLT-NAACL. (2004) 105–112
6. Arnold, A., Dauchet, M.: Transductions inversibles de forêts. Thèse 3ème cycle M. Dauchet, Université de Lille (1975)
7. Arnold, A., Dauchet, M.: Bi-transductions de forêts. In: ICALP. Edinburgh University Press (1976) 74–86

8. Rounds, W.C.: Mappings and grammars on trees. Math. Systems Theory **4**(3) (1970) 257–287
9. Thatcher, J.W.: Generalized^2 sequential machine maps. J. Comput. System Sci. **4**(4) (1970) 339–367
10. Steinby, M., Tîrnăucă, C.I.: Syntax-directed translations and quasi-alphabetic tree bimorphisms. In: CIAA. Volume 4783 of LNCS, Springer (2007) 265–276
11. Aho, A.V., Ullman, J.D.: Syntax directed translations and the pushdown assembler. J. Comput. System Sci. **3**(1) (1969) 37–56
12. Shabes, Y.: Mathematical and Computational Aspects of Lexicalized Grammars. PhD thesis, University of Pennsylvania (1990)
13. Shieber, S.M., Shabes, Y.: Synchronous tree-adjoining grammars. In: COLING. (1990) 1–6
14. Shieber, S.M.: Unifying synchronous tree adjoining grammars and tree transducers via bimorphisms. In: EACL. The Association for Computer Linguistics (2006) 377–384
15. Arnold, A., Dauchet, M.: Morphismes et bimorphismes d'arbres. Theoret. Comput. Sci. **20** (1982) 33–93
16. Fülöp, Z., Kühnemann, A., Vogler, H.: A bottom-up characterization of deterministic top-down tree transducers with regular look-ahead. Inf. Process. Lett. **91**(2) (2004) 57–67
17. Fülöp, Z., Kühnemann, A., Vogler, H.: Linear deterministic multi bottom-up tree transducers. Theoret. Comput. Sci. **347**(1–2) (2005) 276–287
18. Maletti, A.: Compositions of extended top-down tree transducers. Inform. and Comput. (2008) to appear.
19. Engelfriet, J.: Bottom-up and top-down tree transformations: A comparison. Math. Systems Theory **9**(3) (1975) 198–231
20. Baker, B.S.: Composition of top-down and bottom-up tree transductions. Inform. and Control **41**(2) (1979) 186–213
21. Thatcher, J.W.: Tree automata: An informal survey. In: Currents in the Theory of Computing. Prentice Hall (1973) 143–172
22. Ganzinger, H.: Increasing modularity and language-independency in automatically generated compilers. Sci. Comput. Prog. **3**(3) (1983) 223–278
23. Giegerich, R.: Composition and evaluation of attribute coupled grammars. Acta Inform. **25**(4) (1988) 355–423
24. Kühnemann, A.: Berechnungsstärken von Teilklassen primitiv-rekursiver Programmschemata. PhD thesis, Technische Universität Dresden (1997)
25. Kühnemann, A.: Benefits of tree transducers for optimizing functional programs. In: FSTTCS. Volume 1530 of LNCS, Springer (1998) 146–157
26. Engelfriet, J., Maneth, S.: Macro tree transducers, attribute grammars, and MSO definable tree translations. Inform. and Comput. **154**(1) (1999) 34–91
27. Engelfriet, J.: Top-down tree transducers with regular look-ahead. Math. Systems Theory **10**(1) (1977) 289–303
28. Gécseg, F., Steinby, M.: Tree Automata. Akadémiai Kiadó, Budapest (1984)
29. Gécseg, F., Steinby, M.: Tree languages. In: Handbook of Formal Languages. Volume 3. Springer (1997) 1–68
30. Lilin, E.: Une généralisation des transducteurs d'états finis d'arbres: les S-transducteurs. Thèse 3ème cycle, Université de Lille (1978)
31. Lilin, E.: Propriétés de clôture d'une extension de transducteurs d'arbres déterministes. In: CAAP. Volume 112 of LNCS, Springer (1981) 280–289