Compositions of Extended Top-down Tree Transducers

Andreas Maletti

Institute of Theoretical Computer Science, Faculty of Computer Science
Technische Universität Dresden

maletti@tcs.inf.tu-dresden.de

Abstract. Unfortunately the class of transformations computed by nondeleting and linear extended top-down tree transducers [Graehl, Knight: *Training Tree Transducers.* HLT-NAACL 2004] is not closed under composition. It is shown that the class of transformations computed by nondeleting and linear bimorphisms actually coincides with the previously mentioned class. Moreover, every nondeleting and linear bimorphism with an ε-free input-homomorphism can straightforwardly be implemented by a multi bottom-up tree transducer [Fülöp, Kühnemann, Vogler: *A Bottom-up Characterization of Deterministic Top-down Tree Transducers with Regular Look-ahead.* Inf. Proc. Letters 91, 2004]. The class of transformations realized by the latter devices is shown to be closed under composition and is included in the composition of the class of transformations realized by top-down tree transducers with itself.

1 Introduction

Top-down tree transducers (for short: tdtts) were introduced in [1, 2] and intensively studied thereafter (see [3] for a survey). Those devices were originally motivated from syntax-directed semantics [4], but were later successfully applied to problems as diverse as: functional programming [5]; analysis of cryptographic protocols [6]; and decidability of the first-order theory of ground rewriting [7].

In particular, compositions of tdtts are considered in [8, 9]. In this paper we study compositions of extended tdtts, which were introduced in [10] and subsequently led to several improvements [11] in machine translation (see [12] for a survey). In fact, [12] explicitly mentions the closure of the class of transformations computed by extended tdtts under composition as an open problem of paramount importance in natural language processing. A partial solution is given in [13] where it is shown that the class of transformations computed by nondeleting and linear extended tdtts is not closed under composition. The proof shows that the non-closure is essentially due to the linearity property (i.e., copying extended tdtts can compute the transformation presented in [13]); thus the general problem remains open.

An extended tdtt essentially is a tdtt whose left-hand sides of rules offer not only shallow patterns of the form $\sigma(x_1, \ldots, x_k)$ for some k-ary symbol σ, but allow arbitrary patterns (without repeated variables) as left-hand sides. In this paper we will mostly consider nondeleting and linear extended tdtts, in which the right-hand side of a rule may not contain several occurrences of a variable and further must contain every variable that occurs in the left-hand side of that rule. Two example rules are shown in Fig. 1. The semantics of extended tdtts is given by a simple rewrite semantics. An instance of a left-hand side of a rule is replaced by the appropriately instantiated right-hand side of that rule. We start this rewriting process with $q(t)$ where q is an initial state and t is the input tree. An extended tdtt may thus transform an input tree t into an output tree u if there exists an initial state q such that $q(t)$ can be rewritten to u.

It is shown in [14] that synchronized tree substitution grammars [15] are as powerful (upto relabeling) as bimorphisms (see survey [16]) of type $(\mathrm{LC}, \mathrm{LC})$ [14]. We first show that nondeleting and linear extended tdtts are as powerful as bimorphisms of type $(\mathrm{LC}, \mathrm{LC})$, which thus shows that nondeleting and linear extended tdtts are as powerful as synchronized tree substitution grammars (modulo relabeling). It is already remarked in [10] that the two previously mentioned devices are similar. The problem of the closure under composition of the class of transformations computed by synchronized tree substitution grammars is open since their introduction in the 90s [15]. The bimorphism characterization [14] was proposed as a first step towards composition results, however no one seems to have followed this lead.

In this paper we approach the issue by a bottom-up device: multi bottom-up tree transducers [17] (for short: mbutts). We show that restricted nondeleting and linear extended tdtts can be simulated by nondeleting and linear mbutts. Then we show that the class of transformations computed by linear mbutts is closed under composition. This is surprising because nondeleting and linear mbutts can reproduce certain forms of (top-down and bottom-up) copying. Finally, we discuss how to implement mbutts in a top-down fashion, alas not as extended tdtts as this would be impossible in general because the class of transformations computed by nondeleting and linear extended tdtts is not closed under composition [13]. Thus we do not solve the problem as originally posed but present a suitable superclass of transformations which enjoys the much required closure under composition. Furthermore, we illustrate the power of compositions of extended tdtts and support the validation of composition algorithms (e.g., the implementation of compositions in TIBURON [18]).

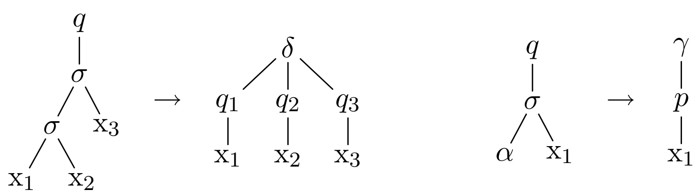

Fig. 1. Illustration of extended top-down tree transducer rules.

2 Preliminaries

We use $\mathbb{N}$ to denote the set of natural numbers including 0, and we use $\mathbb{N}_+$ to denote $\mathbb{N} \setminus \{0\}$. We fix a set $X = \{x_1, x_2, \dots\}$ of variables, and for every $k \in \mathbb{N}$ we let $X_k = \{x_i \mid 1 \leqslant i \leqslant k\}$. Since we need the restriction $1 \leqslant i \leqslant k$ often, we abbreviate $\{i \mid 1 \leqslant i \leqslant k\}$ by $[k]$. Alphabets and ranked alphabets are defined as usual. We use $\Sigma^{(k)}$ to denote the set of k-ary symbols of a ranked alphabet Σ and write rk_Σ for the rank function associated to Σ. The set of Σ-trees indexed by V is denoted by $T_\Sigma(V)$.

The set of variables occurring in a tree $t \in T_\Sigma(V)$ is denoted by $\mathrm{var}(t)$. We call t nondeleting (respectively, linear) in V if every $v \in V$ occurs at least (respectively, at most) once in t. The sets $\mathrm{pos}(t)$ and $\mathrm{Sub}(t)$ denote the set of positions of t and the set of subtrees of t, respectively, and are defined as usual. For every $w \in \mathrm{pos}(t)$ we write $t(w)$ for the symbol that occurs at position w in t. By $t|_w$ we denote the subtree of t that is rooted at w, and by $t[t']_w$ we denote the tree obtained from t by replacing the subtree rooted at w by t'.

For sets P and T, we write $P(T)$ to denote $\{p(t) \mid p \in P, t \in T\}$. Moreover, given a ranked alphabet Δ, we write $\Delta[T]$ for the set

$$\{\delta(t_1, \dots, t_k) \mid \delta \in \Delta^{(k)}, t_1, \dots, t_k \in T\} \enspace .$$

Finally, we write ; for function composition provided that the types are compatible; i.e., given $f\colon A \to B$ and $g\colon B \to C$ the expression $f \mathbin{;} g$ denotes the function from A to C such that $(f \mathbin{;} g)(a) = g(f(a))$ for every $a \in A$. A detailed introduction into tree language theory can be found, e.g., in [3]. There one will also find the definitions of Σ-algebra homomorphisms and recognizable tree languages.

3 Extended Top-down Tree Transducer and Bimorphism

In this section, we recall the notion of an extended top-down tree transducer [10]. Essentially extended top-down tree transducers have rules in which the left-hand side may contain arbitrary, not just shallow, patterns.

Definition 1 (cf. Section 4 in [10]). *An* extended top-down tree transducer *is a tuple* $M = (Q, \Sigma, \Delta, I, R)$ *such that*

- Q *is a finite set of* states*;*
- Σ *and* Δ *are* input *and* output ranked alphabet*;*
- $I \subseteq Q$ *is a set of* initial states*; and*
- $R \subseteq Q(T_\Sigma(X)) \times T_\Delta(Q(X))$ *is a finite set of* rules *such that* l *is linear in* X *and* $\mathrm{var}(r) \subseteq \mathrm{var}(l)$ *for every* $(l, r) \in R$.

We say that M *is* nondeleting *(respectively,* linear*), if* $\mathrm{var}(l) = \mathrm{var}(r)$ *(respectively,* r *is linear in* X*) for every* $(l, r) \in R$*. Moreover, we say that* M *is* a top-down tree transducer*, if for every* $(l, r) \in R$ *there exist* $k \in \mathbb{N}$*,* $q \in Q$*, and* $\sigma \in \Sigma_k$ *such that* $l = q(\sigma(x_1, \ldots, x_k))$*.*

Without loss of generality we commonly assume that for every rule $(l, r) \in R$ there exists $n \in \mathbb{N}$ such that $\mathrm{var}(l) = X_n$. Moreover, we commonly write $(l \to r)$ instead of (l, r) when handling rules. Finally, a top-down tree transducer is *deterministic*, if for every left-hand side l there exists at most one right-hand side r such that $l \to r$.

In the sequel, we abbreviate top-down tree transducer to tdtt. The semantics of those devices is given by a straightforward rewrite semantics. We identify an instance of the left-hand side in a sentential form and replace this instance by a corresponding (according to the rules of the tree transducer) instantiated right-hand side.

Definition 2 (cf. Section 4 in [10]). *Let* $M = (Q, \Sigma, \Delta, I, R)$ *be an extended tdtt. The relation* $\Rightarrow_M \subseteq T_\Delta(Q(T_\Sigma))^2$ *is defined by* $\xi \Rightarrow_M \xi'$ *iff*

- *there exists a position* $w \in \mathrm{pos}(\xi)$*;*
- *there exists a rule* $(l \to r) \in R$*; and*
- *there exists a substitution* $\theta \colon X \to T_\Sigma$

such that $l\theta = \xi|_w$ *and* $\xi' = \xi[r\theta]_w$*. The* tree transformation computed by M*, denoted by* $\|M\| \subseteq T_\Sigma \times T_\Delta$*, is defined by*

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

Our first result will relate nondeleting and linear extended tdtt and particular bimorphisms. To this end, let us recall the bimorphism approach to tree transformations. A homomorphism $h\colon T_\Sigma \to T_\Delta$ is called *nondeleting* (respectively, *linear*), if $h(\sigma)$ is nondeleting (respectively, linear) in X_k for every $k \in \mathbb{N}$ and $\sigma \in \Sigma^{(k)}$. A bimorphism just consists of a recognizable tree language and two homomorphisms.

Definition 3. *Let Σ, Γ, and Δ be ranked alphabets. A* bimorphism *is a triple $B = (\varphi, L, \psi)$ such that*

- $\varphi\colon T_\Gamma \to T_\Sigma$ *is a homomorphism (the* input homomorphism*);*
- $L \subseteq T_\Gamma$ *is a recognizable tree language; and*
- $\psi\colon T_\Gamma \to T_\Delta$ *is a homomorphism (the* output homomorphism*).*

The tree transformation induced by B*, denoted by $\|B\|$, is defined by*

$$\|B\| = \{(\varphi(s), \psi(s)) \in T_\Sigma \times T_\Delta \mid s \in L\} \enspace .$$

We call a bimorphism (φ, L, ψ) nondeleting (respectively, linear), if φ and ψ are nondeleting (respectively, linear). The class of tree transformations computable by nondeleting (note that the term "complete" is used instead of "nondeleting" in [14]) and linear bimorphisms is denoted by $\mathrm{B(LC, LC)}$, and the class of tree transformations computed by nondeleting and linear extended tdtts is denoted by nl–XTOP. The following straightforward lemma shows that the power of nondeleting and linear extended tdtts and nondeleting and linear bimorphisms coincides.

Lemma 4. $\mathrm{B(LC, LC)} = \text{nl–XTOP}$.

We already remarked that the class of transformations computed by nondeleting and linear extended tdtts is not closed under composition. This immediately yields the following corollary.

Corollary 5 (see [13]). $\mathrm{B(LC, LC)}$ *is not closed under composition.*

4 Multi Bottom-up Tree Transducer

In this section, we recall multi bottom-up tree transducers from [17]. We slightly adapt the model by omitting a special root symbol. In [17] this symbol is required so that the bottom-up device may deterministically identify the root of the input tree. However, in natural language processing deterministic tree transducers have only very limited applications [12], so we will not deal with deterministic devices.

Definition 6 (cf. Section 3 in [17]). *A* multi bottom-up tree transducer *is a tuple* $(Q, \Sigma, \Delta, F, R)$ *such that*

- Q *is a ranked alphabet (the* states*) with* $Q^{(0)} = \emptyset$*;*
- Σ *and* Δ *are ranked alphabets (the* input *and* output *symbols);*
- $F \subseteq Q^{(1)}$ *(the set of* final states*); and*
- R *is a finite set (the* rules*) in which every element is of the form*

$$\sigma(q_1(x_{1,1}, \ldots, x_{1,n_1}), \ldots, q_k(x_{k,1}, \ldots, x_{k,n_k})) \to q(t_1, \ldots, t_n)$$

with $k \in \mathbb{N}$*,* $\sigma \in \Sigma^{(k)}$*,* $n, n_1, \ldots, n_k \in \mathbb{N}_+$*,* $q \in Q^{(n)}$*,* $q_i \in Q^{(n_i)}$ *for every* $i \in [k]$*, and* $t_1, \ldots, t_n \in T_\Delta(Y)$ *with* $Y = \{x_{i,j} \mid i \in [k], j \in [n_i]\}$*.*

We say that M *is* nondeleting *(respectively,* linear*), if for every rule* $(l \to r) \in R$ *every variable that occurs in* l *occurs at least (respectively, at most) once in* r*.*

We abbreviate $\sigma(q_1(x_{1,1}, \ldots, x_{1,n_1}), \ldots, q_k(x_{k,1}, \ldots, x_{k,n_k}))$ simply to $\sigma(q_1, \ldots, q_k)$. Moreover, we abbreviate multi bottom-up tree transducer to mbutt. The semantics of mbutts is given by a rewrite semantics (the set V of variables is only needed for the composition construction).

Definition 7. *Let* $M = (Q, \Sigma, \Delta, F, R)$ *be a mbutt and* V *a set. The relation* $\Rightarrow_M \subseteq T_\Sigma(Q[T_\Delta(V)])^2$ *is defined for every* $\xi, \xi' \in T_\Sigma(Q[T_\Delta(V)])$ *by* $\xi \Rightarrow_M \xi'$ *iff*

- *there exists a position* $w \in \mathrm{pos}(\xi)$*;*
- *there exists a rule* $(l \to r) \in R$*; and*
- *there exists a substitution* $\theta \colon X \to T_\Delta(V)$

such that $\xi|_w = l\theta$ *and* $\xi' = \xi[r\theta]_w$*. The tree transformation* computed by M*, denoted by* $\|M\| \subseteq T_\Sigma \times T_\Delta$*, is defined by*

$$\|M\| = \{(t, u) \in T_\Sigma \times T_\Delta \mid \exists q \in F \colon t \Rightarrow_M^* q(u)\}.$$

It might be somewhat surprising that we start our investigation of mbutts with a composition result (similar to the composition result for deterministic mbutts in [19]). We will later relate mbutts to extended tdtts and bimorphisms using the composition result established next. First let us define the composition of two mbutts. The general idea is the classic one: take the cross-product of the sets of states and simulate the second transducer on the right-hand sides of the first transducer. However, a k-ary state of the first transducer has k prepared (partial) output trees. Thus we also need to process those k trees with the second transducer, which gives states of the form $(q, p_1 \cdots p_k)$. This idea was already used in the composition construction for deterministic mbutts in [19].

Definition 8 (cf. [19]). *Let M_1 be the mbutt $(Q_1, \Sigma, \Gamma, F_1, R_1)$ and M_2 be the mbutt $(Q_2, \Gamma, \Delta, F_2, R_2)$. The* composition of M_1 and M_2, *denoted by $M_1 \,;M_2$, is the mbutt $M_1 \,;M_2 = (Q, \Sigma, \Delta, F_1 \times F_2, R)$ where*

- $Q^{(k)} = \{(q, q_1 \cdots q_n) \in Q_1 \times Q_2^n \mid n = \mathrm{rk}_{Q_1}(q), k = \sum_{i=1}^n \mathrm{rk}_{Q_2}(q_i)\}$ *for every $k \in \mathbb{N}$;*
- *R is given as follows.*

Let $k \in \mathbb{N}$, $\sigma \in \Sigma^{(k)}$, $n, n_1, \ldots, n_k \in \mathbb{N}_+$, $q \in Q_1^{(n)}$, and $q_i \in Q_1^{(n_i)}$ for every $i \in [k]$. Moreover, let $w_i \in Q_2^{n_i}$ for every $i \in [k]$ and $k_1, \ldots, k_n \in \mathbb{N}_+$, $q'_j \in Q_2^{(k_j)}$ for every $j \in [n]$, and $t'_{j,j'} \in T_\Delta(X)$ for every $j \in [n]$ and $j' \in [k_j]$. The set R contains the rule

$$\sigma((q_1, w_1), \ldots, (q_k, w_k)) \to (q, q'_1 \cdots q'_n)(t'_{1,1}, \ldots, t'_{1,k_1}, \ldots, t'_{n,1}, \ldots, t'_{n,k_n})$$

if and only if

- *there exists a rule $(\sigma(q_1, \ldots, q_k) \to q(t_1, \ldots, t_n)) \in R_1$;*
- *for every $j \in [k]$ let $w_j = p_{j,1} \cdots p_{j,n_j}$ for some $m_{j,1}, \ldots, m_{j,n_j} \in \mathbb{N}_+$ and $p_{j,j'} \in Q_2^{(m_{j,j'})}$ for every $j' \in [n_j]$; and*
- *for every $i \in [n]$*

$$t_i[p_{j,j'}(\boldsymbol{x}_{\langle jj' \rangle})]_{j \in [k], j' \in [n_j]} \Rightarrow^*_{M_2} q'_i(t'_{i,1}, \ldots, t'_{i,k_i})$$

where $\boldsymbol{x}_{\langle jj' \rangle} = (x_{j, m_{j,1} + \cdots + m_{j,j'-1} + 1}, \ldots, x_{j, m_{j,1} + \cdots + m_{j,j'}})$.

For the next lemma we need a new concept. Let $M = (Q, \Sigma, \Delta, F, R)$ be a mbutt. We call M *total*, if for every $k \in \mathbb{N}$, $\sigma \in \Sigma^{(k)}$, and states $q_1, \ldots, q_k \in Q$ there exists r such that $(\sigma(q_1, \ldots, q_k) \to r) \in R$. The classic construction shows that for every mbutt a (semantically) equivalent total mbutt can be constructed. The following lemma states the central property which is required to show the correctness of the construction of Definition 8 for restricted input mbutts. The restrictions are that the first transducer is linear and the second total.

Lemma 9. *Let $M_1 = (Q_1, \Sigma, \Gamma, F_1, R_1)$ and $M_2 = (Q_2, \Gamma, \Delta, F_2, R_2)$ be mbutts such that M_1 is linear and M_2 is total, and let $M = M_1 \,;M_2$. Let $t \in T_\Sigma$, $n \in \mathbb{N}_+$, $q \in Q_1^{(n)}$, $m_1, \ldots, m_n \in \mathbb{N}_+$, and $\boldsymbol{q}_i \in Q_2^{(m_i)}$ and $\boldsymbol{u}_i \in T_\Delta^{m_i}$ for every $i \in [n]$.*

$$t \Rightarrow^*_M (q, \boldsymbol{q})(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n)$$
$$\iff \exists \boldsymbol{v} \in T_\Gamma^n, \forall i \in [n] \colon \Big(t \Rightarrow^*_{M_1} q(\boldsymbol{v}) \wedge \boldsymbol{v}_i \Rightarrow^*_{M_2} \boldsymbol{q}_i(\boldsymbol{u}_i)\Big)$$

Theorem 10. *Let M_1 and M_2 be mbutts such that M_1 is linear and M_2 is total, and let $M = M_1 \,; M_2$. Then $\|M\| = \|M_1\| \,; \|M_2\|$.*

The following corollary summarizes the composition results obtained. It is easy to check that the nondeletion and linearity conditions are preserved in the construction of Definition 8. By MBOT we denote the class of tree transformations computable by mbutts. We use the prefixes n and l for nondeletion and linearity, respectively; i.e., the class nl–MBOT comprises all tree transformations computable by nondeleting and linear mbutts. We write [n] or [l] for an optional nondeletion or linearity restriction. In statements such an optional restriction $[r]$ can be replaced (consistently) by either the empty word ε or r to obtain a valid statement.

Corollary 11 (of Theorem 10).

$$\text{[n]l–MBOT} \,; \text{[n][l]–MBOT} = \text{[n][l]–MBOT} \enspace .$$

5 Bimorphisms and Multi Bottom-up Tree Transducers

As promised we return to the issue of relating mbutts to bimorphisms. We immediately observe that $\mathrm{B(LC, LC)}$ exhibits a strong symmetry because it is closed under inverses; i.e., with every $\rho \in \mathrm{B(LC, LC)}$ also $\rho^{-1} \in \mathrm{B(LC, LC)}$. Let B be a bimorphism computing a tree transformation $\|B\| \subseteq T_\Sigma \times T_\Delta$. It is easily verifiable that, in general, it is possible that there exists a tree t such that $\|B\| \cap (\{t\} \times T_\Delta)$ is infinite. However, for an mbutt M the set $\|M\| \cap (\{t\} \times T_\Delta)$ is always finite. This observation remains valid if we restrict ourselves to nondeleting and linear bimorphisms and mbutts.

Let $\varphi \colon T_\Gamma \to T_\Sigma$ be a homomorphism. We call φ *nonerasing*, if $\varphi(\gamma) \notin X$ for every $\gamma \in \Gamma$. Correspondingly, we call an extended tdtt $M = (Q, \Sigma, \Delta, I, R)$ *input-consuming*, if $l \notin Q(X)$ for every $(l \to r) \in R$. We use the stem $\mathrm{XTOP_{ic}}$ (with the usual prefixes) for classes of transformations computable by input-consuming extended tdtt. Moreover, we use $\mathrm{B(LC_{ne}, LC)}$ for the class of transformations computable by nondeleting and linear bimorphisms whose input homomorphism is nonerasing.

Corollary 12 (of Lemma 4). $\mathrm{B(LC_{ne}, LC)} = \text{nl–}\mathrm{XTOP_{ic}}$.

Now we are ready to show that nondeleting and linear bimorphisms whose input homomorphism is nonerasing can be implemented by nondeleting and linear mbutts. For this we present the bimorphism as a composition of three tree transformations and show that each can be simulated by a mbutt. The composition result in Corollary 11 then yields the desired result.

Lemma 13. *Let Σ and Δ be ranked alphabets and $h\colon T_\Sigma \to T_\Delta$ be a nondeleting and linear homomorphism.*

(i) $h \in \text{nl–MBOT}$*; and*
(ii) if h *is nonerasing, then* $h^{-1} \in \text{nl–MBOT}$.

Proof. The tree transformation h can trivially be realized by a nondeleting and linear tdtt and thus also by a nondeleting and linear mbutt.

The inverse homomorphism is more difficult. We construct a nondeleting and linear mbutt $M = (Q', \Delta, \Sigma, F, R)$ as follows:

- $Q^{(k)} = \{(\sigma, w) \mid \sigma \in \Sigma, w \in \text{pos}(h(\sigma)), k = |\text{var}(h(\sigma)|_w)|\}$ for every $k \in \mathbb{N}$;
- $Q' = Q \cup \{\star^{(1)}\}$;
- $F = \{\star\}$; and
- the set R of rules is given as follows.

Let $k \in \mathbb{N}$ and $\delta \in \Delta^{(k)}$. Moreover, let $n_1, \ldots, n_k \in \mathbb{N}_+$ and $q_1, \ldots, q_k \in Q$ be such that $q_i \in Q^{(i)}$ for every $i \in [k]$. Finally, let $u \in T_\Sigma(Q(X))$, and let $\boldsymbol{x}_i = (x_{i,1}, \ldots, x_{i,n_i})$ for every $i \in [k]$. We have that

$$\Big(\delta(q_1(\boldsymbol{x}_1), \ldots, q_k(\boldsymbol{x}_k)) \to u\Big) \in R$$

if and only if there exists $n \in \mathbb{N}$, $\sigma \in \Sigma^{(n)}$, and $w \in \text{pos}(h(\sigma))$ such that

(i) $h(\sigma)(w) = \delta$;
(ii) for every $i \in [k]$

$$q_i = \begin{cases} (\sigma, wi) & \text{if } h(\sigma)(wi) \notin X \\ \star & \text{otherwise} \end{cases}$$

(iii) and

$$u = \begin{cases} (\sigma, w)(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k) & \text{if } w \neq \varepsilon \\ \star(\sigma(\boldsymbol{y})) & \text{otherwise} \end{cases}$$

where $\boldsymbol{y}$ contains the variables of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ sorted in the order induced by $h(\sigma)$; i.e., the $x_{i,j}$ that corresponds to x_1 in $h(\sigma)$ comes first, then the variable corresponding to x_2 in $h(\sigma)$, etc.

Note that by conditions (ii) and (iii) and the nondeletion and linearity of h we have $n_1 + \cdots + n_k = n$. It is easily checked that M is indeed nondeleting and linear. It is intuitively clear that $\|M\| = h^{-1}$. □

From this lemma we can easily conclude that every input-consuming nondeleting and linear extended tdtt can be simulated by a nondeleting and linear mbutt.

Theorem 14. $\text{nl–XTOP}_{\text{ic}} \subseteq \text{nl–MBOT}$.

Proof. By Corollary 12 we have $\mathrm{B}(\mathrm{LC}_{\mathrm{ne}}, \mathrm{LC}) = \text{nl–XTOP}_{\text{ic}}$. This yields that for every input-consuming nondeleting and linear extended tdtt $M = (Q, \Sigma, \Delta, I, R)$ there exists a nondeleting and linear bimorphism $B = (\varphi, L, \psi)$ with φ nonerasing such that $\|B\| = \|M\|$.

$$\begin{aligned} \|B\| &= \{(\varphi(t), \psi(t)) \mid t \in L\} \\ &= \{(t, u) \mid \exists s \in T_\Gamma \colon (t, s) \in \varphi^{-1}, (s, s) \in \mathrm{id}_L, (s, u) \in \psi\} \\ &= \varphi^{-1} \,;\, \mathrm{id}_L \,;\, \psi \end{aligned}$$

By Lemma 13 and Corollary 11 this shows that $\|M\| \in \text{nl–MBOT}$ because id_L can be implemented by a nondeleting and linear mbutt. □

It follows from [13] that $\text{nl–XTOP}_{\text{ic}}$ is not closed under composition. Thus we immediately obtain that $\text{nl–XTOP}_{\text{ic}} \subset \text{nl–MBOT}$ because nl–MBOT is closed under composition (see Corollary 11). Thus we identified a suitable superclass which possesses the much required closure under composition. Nondeleting mbutts are usually quite efficient because they visit each node of the input tree at most once and each constructed output tree is used in the final output tree. However, they are also more difficult to implement than, e.g., tdtts. Let us investigate the input-consuming restriction for extended tdtts. It might seem like a harsh restriction to disallow rules of the form $q(x) \to u$. Such rules are quite useful in practice and used, e.g., for spontaneous insertion in machine translation systems.

Lemma 15. *Let $M = (Q, \Sigma, \Delta, I, R)$ be a nondeleting and linear extended tdtt such that $\|M\| \cap (\{t\} \times T_\Delta)$ is finite for every $t \in T_\Sigma$. Then there exists an input-consuming nondeleting and linear extended tdtt M' such that $\|M'\| = \|M\|$.*

We showed that only if the spontaneous insertations are unbounded then we potentially cannot model it by a input-consuming extended tdtt.

Finally, let us consider how mbutts relate to extended tdtts. An important result in this respect can be found in [17]. It is shown there that every linear deterministic mbutt can be simulated by a deterministic tdtt with regular look-ahead [20]. Here we present a slightly different construction which however yields the mentioned result of [17] as a corollary. Our construction is a faithful generalization of the decomposition of bottom-up tree transducers of [9]. We denote by QREL and d–TOP the classes of tree transformations computable by stateful relabelings (i.e., tdtts with rules of the form $q(\sigma(x_1, \ldots, x_k)) \to \delta(q_1(x_1), \ldots, q_k(x_k))$ where σ and δ are k-ary) and deterministic tdtts, respectively.

Theorem 16 (cf. Lemma 4.1 of [17]). nl–MBOT $\subseteq$ QREL ; d–TOP

Proof. The stateful relabeling annotates the input tree with the transitions applied by a run of the mbutt. It thus takes care of the nondeterminism. The deterministic tdtt then executes the annotated transitions using a state for each parameter position. □

It can thus be shown that compositions of input-consuming, nondeleting and linear tdtts can be simulated by a composition of a stateful relabeling and a deterministic tdtt. This is our main theorem for compositions of extended tdtts.

Theorem 17. *For every* $n \in \mathbb{N}_+$

$$\text{nl-XTOP}_{\text{ic}}^n \subseteq \text{nl–MBOT} \subseteq \text{QREL} \,;\, \text{d–TOP} \enspace .$$

Proof. We have the inclusions

$$\text{nl-XTOP}_{\text{ic}}^n \;\subseteq\; \text{nl–MBOT}^n \;\subseteq\; \text{nl–MBOT} \;\subseteq\; \text{QREL} \,;\, \text{d–TOP}$$

by Theorem 14, Corollary 11, and Theorem 16, respectively. □

6 Conclusions and Open Problems

We have identified a class, namely nl–MBOT, that is closed under compositions and contains all transformations that can be computed by input-consuming, nondeleting and linear extended tdtt. We further showed that compositions of input-consuming, nondeleting and linear extended tdtt can be implemented by a single composition of a stateful relabeling and a deterministic tdtt.

It remains an open problem to decide whether the composition of the transformations computed by two extended tdtts can be computed by just a single extended tdtt. In the relevant subcase where the two extended tdtts are input-consuming one can investigate how to implement (restricted) mbutts using just one extended tdtt.

Acknowledgement. The author is grateful to the anonymous referees for their insightful remarks on the draft version of the paper.

References

1. Doner, J.: Tree acceptors and some of their applications. J. Comput. System Sci. **4**(4) (1970) 406–451
2. Thatcher, J.W.: Generalized^2 sequential machine maps. J. Comput. System Sci. **4**(4) (1970) 339–367
3. Gécseg, F., Steinby, M.: Tree languages. In Rozenberg, G., Salomaa, A., eds.: Handbook of Formal Languages. Volume 3. Springer (1997) 1–68
4. Fülöp, Z., Vogler, H.: Syntax-directed Semantics—Formal Models Based on Tree Transducers. Monographs in Theoret. Computer Sci. Springer (1998)
5. Kühnemann, A.: Benefits of tree transducers for optimizing functional programs. In Arvind, V., Ramanujam, R., eds.: Proc. 18th Int. Conf. Found. Software Tech. & Theoret. Computer Sci. Volume 1530 of LNCS., Springer (1998) 146–157
6. Küsters, R., Wilke, T.: Automata-based analysis of recursive cryptographic protocols. In Diekert, V., Habib, M., eds.: Proc. 21st Annual Symp. Theoret. Aspects of Comput. Sci. Volume 2996 of LNCS., Springer (2004) 382–393
7. Dauchet, M., Tison, S.: The theory of ground rewrite systems is decidable. In: Proc. 5th Annual IEEE Symp. Logic in Comput. Sci., IEEE Computer Society (1990) 242–248
8. Baker, B.S.: Composition of top-down and bottom-up tree transductions. Inform. and Control **41**(2) (1979) 186–213
9. Engelfriet, J.: Bottom-up and top-down tree transformations—a comparison. Math. Systems Theory **9**(3) (1975) 198–231
10. Graehl, J., Knight, K.: Training tree transducers. In: Proc. HLT/NAACL, Association for Computational Linguists (2004) 105–112
11. Huang, B., Knight, K.: Relabeling syntax trees to improve syntax-based machine translation quality. In: Proc. HLT/NAACL, Association for Computational Linguistics (2006) 240–247
12. Knight, K., Graehl, J.: An overview of probabilistic tree transducers for natural language processing. In Gelbukh, A.F., ed.: Proc. 6th Int. Conf. Comput. Linguistics and Intel. Text Proc. Volume 3406 of LNCS., Springer (2005) 1–24
13. Knight, K., Graehl, J., Hopkins, M.: Extended top-down tree transducers. Manuscript (2006)
14. Shieber, S.M.: Synchronous grammars as tree transducers. In: Proc. 7th Int. Workshop Tree Adjoining Grammars and Related Formalisms. (2004) 88–95
15. Schabes, Y.: Mathematical and Computational Aspects of Lexicalized Grammars. PhD thesis, University of Pennsylvania (1990)
16. Dauchet, M., Tison, S.: Structural complexity of classes of tree languages. In Nivat, M., Podelski, A., eds.: Tree Automata and Languages. Elsevier Science (1992) 327–353
17. Fülöp, Z., Kühnemann, A., Vogler, H.: A bottom-up characterization of deterministic top-down tree transducers with regular look-ahead. Inform. Proc. Letters **91** (2004) 57–67
18. May, J., Knight, K.: Tiburon: A weighted tree automata toolkit. In Ibarra, O.H., Yen, H.C., eds.: Proc. 11th Int. Conf. Implementation and Application of Automata. Volume 4094 of LNCS., Springer (2006) 102–113
19. Kühnemann, A.: Composition of deterministic multi bottom-up tree transducers. Manuscript (2006)
20. Engelfriet, J.: Top-down tree transducers with regular look-ahead. Math. Systems Theory **10** (1977) 289–303