Composition Closure of ε-Free Linear Extended Top-Down Tree Transducers

Zoltán Fülöp[1,⋆] and Andreas Maletti[2,⋆⋆]

[1] Department of Foundations of Computer Science, University of Szeged
Árpád tér 2, H-6720 Szeged, Hungary
fulop@inf.u-szeged.hu
[2] Institute for Natural Language Processing, University of Stuttgart
Pfaffenwaldring 5b, 70569 Stuttgart, Germany
maletti@ims.uni-stuttgart.de

Abstract. The expressive power of compositions of linear extended top-down tree transducers with and without regular look-ahead is investigated. In particular, the restrictions of ε-freeness, strictness, and nondeletion are considered. The composition hierarchy is finite for all ε-free variants of these transducers except for ε-free nondeleting linear extended top-down tree transducers. The least number of transducers needed for the full expressive power of arbitrary compositions is presented.

1 Introduction

The top-down tree transducer is a simple formal model that encodes a tree transformation. It was introduced in [21,22] and intensively studied thereafter (see [13,14,12] for an overview). Roughly speaking, a top-down tree transducer processes the input tree symbol-by-symbol and specifies in its rules, how to translate an input symbol into an output tree fragment together with instructions on how to process the subtrees of the input symbol. This asymmetry between input (single symbol) and output (tree fragment) was removed in extended top-down tree transducers (xt), which were introduced and studied in [1,2]. In an xt the left-hand side of a rule now contains an input tree fragment, in which each variable can occur at most once as a placeholder for a subtree. In particular, the input tree fragment can even be just a variable, which matches every tree, and such rules are called ε-rules. We consider linear xt (l-xt), in which the right-hand side of each rule contains each variable at most once as well. Restricted variants of l-xt are used in most approaches to syntax-based machine translation [16,17].

We also add regular look-ahead [6] (i.e., the ability to check a regular property for the subtrees in an input tree fragment) to l-xt, so our most expressive model

⋆ Supported by the program TÁMOP-4.2.1/B-09/1/KONV-2010-0005 of the Hungarian National Development Agency.
⋆⋆ Supported by the German Research Foundation (DFG) grant MA/4959/1-1.

M.-P. Béal and O. Carton (Eds.): DLT 2013, LNCS 7907, pp. 239–251, 2013.

is the linear extended top-down tree transducer with regular look-ahead (l-xt$^{\mathrm{R}}$). Instead of variables in the left-hand side and a state-variable combination in the right-hand side of a rule, we only use states with the restriction that each state can occur at most once in the left-hand side and at most once in the right-hand side. Moreover, all states that occur in the right-hand side must also occur in the left-hand side. In this way, for each rule the states establish implicit links (a state links its occurrence in the right-hand side with its occurrence in the left-hand side), which form a bijection between a subset of the state occurrences in the left-hand side and all state occurrences in the right-hand side. The state occurrences (in the left-hand side) that do not participate in the bijection (i.e., those states that exclusively occur in the left-hand side) can restrict the acceptable subtrees at their position with the help of regular look-ahead. The implicit links in a rule are made explicit in a derivation, and a rule application expands (explicitly) linked state occurrences at the same time. Example 2 shows an l-xt$^{\mathrm{R}}$, for which we illustrate a few derivation steps in Fig. 1. We use l-XT$^{\mathrm{R}}$ and l-XT to denote the class of all tree transformations computed by l-xt$^{\mathrm{R}}$ and l-xt, respectively.

The expressive power of the various subclasses of l-XT$^{\mathrm{R}}$ is already well understood [15,11]. However, in practice complex systems are often specified with the help of compositions of tree transformations [20] because it is much easier to develop (or train) small components that manage a part of the overall transformation. Consequently, [17] and others declare that closure under composition is a very desirable property for classes of tree transformations (especially in the area of natural language processing). If $\mathcal{C}$ represents the class of all tree transformations computable by a device, then the fact that $\mathcal{C}$ is closed under composition means that we can replace any composition chain specified by several devices by just a single device, which enables an efficient modular development. Unfortunately, neither l-XT$^{\mathrm{R}}$ nor l-XT is closed under composition [2,3,15].

For a class $\mathcal{C}$ of tree transformations we obtain a composition hierarchy $\mathcal{C} \subseteq \mathcal{C}^2 \subseteq \mathcal{C}^3 \subseteq \cdots$, where $\mathcal{C}^n$ denotes the n-fold composition of $\mathcal{C}$. The class $\mathcal{C}$ might be closed under composition at a power n (i.e., $\mathcal{C}^n = \mathcal{C}^{n+1}$) or its hierarchy might be infinite (i.e., $\mathcal{C}^n \subsetneq \mathcal{C}^{n+1}$ for all n). The classes that are closed at a low power are also important in practice. We investigate the composition hierarchy of the classes l-XT$^{\mathrm{R}}$ and l-XT together with various subclasses determined by the properties: ε-freeness, strictness, and nondeletion, abbreviated by '$\not\varepsilon$', 's', and 'n', respectively. We use these symbols in front of l-XT$^{\mathrm{R}}$ and l-XT to obtain the class of all tree transformations computable by the corresponding restricted l-xt$^{\mathrm{R}}$ and l-xt, respectively. In this paper we consider in detail the closure of the classes $\not\varepsilon$l-XT$^{\mathrm{R}}$, $\not\varepsilon$l-XT, $\not\varepsilon$sl-XT$^{\mathrm{R}}$, and $\not\varepsilon$sl-XT under composition.

It is known that none of our considered classes is closed under composition [3]. In addition, it is known [3] that $\not\varepsilon$snl-XT = $\not\varepsilon$snl-XT$^{\mathrm{R}}$ is closed at power 2. We complete the picture by providing the least power at which the above classes are closed under composition in the following table.

Class	Least power of closedness	Proved in
$\not\in$sl-XT, $\not\in$sl-XT$^{\mathrm{R}}$	2	Theorem 14
$\not\in$l-XT	3 or 4 (4)	Theorem 17 (Conjecture)
$\not\in$l-XT$^{\mathrm{R}}$	3	Theorem 17
otherwise	∞	[9, Theorem 34]

2 Notation

We denote the set of all nonnegative integers by $\mathbb{N}$. Every subset of $S \times T$ is a *relation* from S to T. Given relations $R_1 \subseteq S \times T$ and $R_2 \subseteq T \times U$, the *inverse* of R_1 and the *composition* of R_1 and R_2 are denoted by R_1^{-1} and $R_1 ; R_2$, respectively. These notions and notations are lifted to classes of relations in the usual manner. Moreover, the *powers* of a class $\mathcal{C}$ are defined by $\mathcal{C}^1 = \mathcal{C}$ and $\mathcal{C}^{n+1} = \mathcal{C}^n ; \mathcal{C}$ for $n \geq 1$. The *composition hierarchy* [resp. *composition closure*] *of* $\mathcal{C}$ is the family $(\mathcal{C}^n \mid n \geq 1)$ [resp. the class $\bigcup_{n \geq 1} \mathcal{C}^n$]. If $\mathcal{C}^{n+1} = \mathcal{C}^n$, then $\mathcal{C}$ *is closed under composition at power* n. A *ranked alphabet* is a finite set Σ, which is partitioned by $\Sigma = \bigcup_{k \in \mathbb{N}} \Sigma_k$ into subsets Σ_k containing the elements of rank k. We also write $\sigma^{(k)}$ to indicate that $\sigma \in \Sigma_k$. For the rest of this paper, Σ, Δ, and Γ will denote arbitrary ranked alphabets. For every set T, let $\Sigma(T) = \{\sigma(t_1, \ldots, t_k) \mid \sigma \in \Sigma_k, t_1, \ldots, t_k \in T\}$. Let S be a set with $S \cap \Sigma = \emptyset$. The set $T_\Sigma(S)$ of Σ*-trees with leaf labels* S is the smallest set U such that $S \subseteq U$ and $\Sigma(U) \subseteq U$. We write T_Σ for $T_\Sigma(\emptyset)$, and we use $\mathrm{pos}(t) \subseteq \mathbb{N}^*$ to denote the *positions* of $t \in T_\Sigma(S)$. For words $v, w \in \mathbb{N}^*$, we denote the longest common prefix of v and w by $\mathrm{lcp}(v, w)$. The positions $\mathrm{pos}(t)$ are partially ordered by the prefix order $\preceq$ on $\mathbb{N}^*$ [i.e., $v \preceq w$ if and only if $v = \mathrm{lcp}(v, w)$]. The size $|t|$ of the tree $t \in T_\Sigma(S)$ is $|\mathrm{pos}(t)|$. Let $t \in T_\Sigma(S)$ and $w \in \mathrm{pos}(t)$. We denote the *label* of t at w by $t(w)$, and the w*-rooted subtree* of t by $t|_w$. For every $U \subseteq S$, we let $\mathrm{pos}_U(t) = \{w \in \mathrm{pos}(t) \mid t(w) \in U\}$ and $\mathrm{pos}_s(t) = \mathrm{pos}_{\{s\}}(t)$ for every $s \in S$. The tree t is *linear* (resp. *nondeleting*) in U if $|\mathrm{pos}_u(t)| \leq 1$ (resp. $|\mathrm{pos}_u(t)| \geq 1$) for every $u \in U$. Moreover, $\mathrm{var}(t) = \{s \in S \mid \mathrm{pos}_s(t) \neq \emptyset\}$. We write $t[u]_w$ for the tree obtained from $t \in T_\Sigma(S)$ by replacing the subtree $t|_w$ at w by $u \in T_\Sigma(S)$.

For every $n \in \mathbb{N}$ we fix the set $X_n = \{x_1, \ldots, x_n\}$ of variables. Given $t \in T_\Sigma(X_n)$ and $t_1, \ldots, t_n \in T_\Sigma(S)$, we write $t[t_1, \ldots, t_n]$ for the tree obtained from t by replacing each occurrence of x_i by t_i for all $1 \leq i \leq n$. A *tree homomorphism from* Σ *to* Δ is a family of mappings $(h_k \mid k \in \mathbb{N})$ such that $h_k \colon \Sigma_k \to T_\Delta(X_k)$ for every $k \in \mathbb{N}$. Such a tree homomorphism is *linear* (resp. *nondeleting*) if for every $\sigma \in \Sigma_k$ the tree $h_k(\sigma)$ is linear (resp. nondeleting) in X_k. Moreover, it is *strict* [resp. *delabeling*] if $h_k \colon \Sigma_k \to \Delta(T_\Delta(X_k))$ [resp. $h_k \colon \Sigma_k \to X_k \cup \Delta(X_k)$] for every $k \in \mathbb{N}$. We abbreviate the above restrictions by 'l', 'n', 's', and 'd'. The tree homomorphism $(h_k \mid k \in \mathbb{N})$ induces a mapping $h \colon T_\Sigma(S) \to T_\Delta(S)$ defined in the usual way. We denote by H the class of all tree homomorphisms, and for any combination w of 'l', 'n', 's', and 'd' we denote by w-H the class of all w-tree homomorphisms. The set $\mathrm{Reg}(\Gamma)$ contains all *regular tree languages* $L \subseteq T_\Gamma$ [13,14] over the ranked alphabet Γ.

Finally, let $\mathrm{FTA}(\Gamma) = \{\mathrm{id}_L \mid L \in \mathrm{Reg}(\Gamma)\}$, where $\mathrm{id}_L = \{(t,t) \mid t \in L\}$, and let $\mathrm{FTA} = \bigcup_\Gamma \mathrm{FTA}(\Gamma)$ be the class of all partial identities induced by $\bigcup_\Gamma \mathrm{Reg}(\Gamma)$.

3 Linear Extended Top-Down Tree Transducers

Our main model is the linear extended top-down tree transducer [1,2,17,16] with regular look-ahead (l-xt$^{\mathrm{R}}$), which is based on the non-extended variant [21,22,6]. We will present it in a form that is closer to synchronous grammars [4].

Definition 1 (see [15, Section 2.2]). A *linear extended top-down tree transducer* with regular look-ahead (l-xt$^{\mathrm{R}}$) is a tuple $M = (Q, \Sigma, \Delta, I, R, c)$, where

- Q is a finite set of *states*, of which those in $I \subseteq Q$ are *initial*,
- Σ and Δ are ranked alphabets of *input* and *output symbols*,
- $R \subseteq T_\Sigma(Q) \times Q \times T_\Delta(Q)$ is a finite set of *rules* such that ℓ and r are linear in Q and $\mathrm{var}(r) \subseteq \mathrm{var}(\ell)$ for every $(\ell, q, r) \in R$, and
- $c\colon Q^{\mathrm{la}} \to \mathrm{Reg}(\Sigma)$ assigns regular look-ahead to each (potentially) deleted state, where $Q^{\mathrm{la}} = \{q' \in Q \mid \exists (\ell, q, r) \in R\colon q' \in \mathrm{var}(\ell), q' \notin \mathrm{var}(r)\}$.

Next, we recall some important syntactic properties of our model. To this end, let $M = (Q, \Sigma, \Delta, I, R, c)$ be an l-xt$^{\mathrm{R}}$ for the rest of the paper. It is

- a *linear extended tree transducer* [l-xt], if $c(q) = T_\Sigma$ for every $q \in Q^{\mathrm{la}}$,
- a *linear top-down tree transducer with regular look ahead* [l-t$^{\mathrm{R}}$] if $\ell \in \Sigma(Q)$ for every $(\ell, q, r) \in R$,
- a *linear top-down tree transducer* [l-t] if it is both an l-xt and an l-t$^{\mathrm{R}}$,
- *ε-free* [$\not\varepsilon$] (resp. *strict* [s]) if $\ell \notin Q$ (resp. $r \notin Q$) for every $(\ell, q, r) \in R$,
- a *delabeling* [d] if $\ell \in \Sigma(Q)$ and $r \in Q \cup \Delta(Q)$ for every $(\ell, q, r) \in R$,
- *nondeleting* [n] if $\mathrm{var}(r) = \mathrm{var}(\ell)$ for every $(\ell, q, r) \in R$ (i.e., $Q^{\mathrm{la}} = \emptyset$), and
- a *finite-state relabeling* [qr] if it is a nondeleting, strict delabeling l-t such that $\mathrm{pos}_p(\ell) = \mathrm{pos}_p(r)$ for every $(\ell, q, r) \in R$ and $p \in \mathrm{var}(r)$.

For example, dl-t stands for "delabeling linear top-down tree transducer". We write $\ell \xrightarrow{q_1,\dots,q_k} r$ for the rules $(\ell, q_1, r), \dots, (\ell, q_k, r)$. For every $p \in Q$ and $(\ell, q, r) \in R$ we identify $\mathrm{pos}_p(\ell)$ and $\mathrm{pos}_p(r)$ with their unique element if the sets are non-empty. Finally, for every $q \in Q$, we let $R_q = \{\rho \in R \mid \rho = (\ell, q, r)\}$.

Example 2. Let us consider the dl-t$^{\mathrm{R}}$ $M_1 = (Q, \Sigma, \Sigma, \{\star\}, R, c)$ with the states $Q = \{\star, p, q, q^{\mathrm{la}}, \mathrm{id}, \mathrm{id}'\}$, the symbols $\Sigma = \{\sigma^{(2)}, \sigma_1^{(2)}, \sigma_2^{(2)}, \gamma^{(1)}, \alpha^{(0)}\}$, an the following rules in R:

$$\sigma_1(p,q) \xrightarrow{\star,p} \sigma_1(p,q) \qquad \sigma(q,\mathrm{id}) \xrightarrow{q} q \qquad \gamma(\mathrm{id}) \xrightarrow{\mathrm{id},\mathrm{id}'} \gamma(\mathrm{id})$$
$$\sigma_2(\mathrm{id},\mathrm{id}') \xrightarrow{p,q} \sigma_2(\mathrm{id},\mathrm{id}') \qquad \sigma(q^{\mathrm{la}},q) \xrightarrow{q} q \qquad \alpha \xrightarrow{\mathrm{id},\mathrm{id}'} \alpha \ .$$

Since $Q^{\mathrm{la}} = \{q^{\mathrm{la}}, \mathrm{id}\}$, we set $c(q^{\mathrm{la}}) = \{t \in T_\Sigma \mid \mathrm{pos}_{\sigma_2}(t) = \emptyset\}$ and $c(\mathrm{id}) = T_\Sigma$.

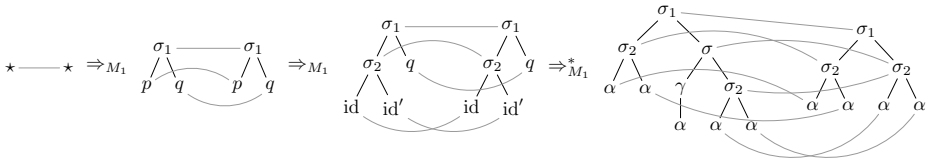

Fig. 1. Derivation using the dl-t$^{\mathrm{R}}$ M_1 of Example 2.

Next, we recall the semantics of the l-xt$^{\mathrm{R}}$ M, which is given by synchronous substitution. Let $\mathcal{L} = \{D \mid D \subseteq \mathbb{N}^* \times \mathbb{N}^*\}$ be the set of all *link structures*.

Definition 3 (see [10, Section 3]). A triple $\langle \xi, D, \zeta \rangle \in T_\Sigma(Q) \times \mathcal{L} \times T_\Delta(Q)$ is a *sentential form* (for M) if $v \in \mathrm{pos}(\xi)$ and $w \in \mathrm{pos}(\zeta)$ for every $(v, w) \in D$. For a set $\mathcal{S}$ of sentential forms we define $\mathrm{links}(\mathcal{S}) = \{D \mid \langle \xi, D, \zeta \rangle \in \mathcal{S}\}$. Let $\rho \in R$ be the rule (ℓ, q, r), and let $v, w \in \mathbb{N}^*$. The *explicit link structure* of ρ for the positions v and w is $\mathrm{links}_{v,w}(\rho) = \{(v.\,\mathrm{pos}_p(\ell), w.\,\mathrm{pos}_p(r)) \mid p \in \mathrm{var}(r)\}$.

Definition 4 (see [10, Section 3]). Given two sentential forms $\langle \xi, D, \zeta \rangle$ and $\langle \xi', D', \zeta' \rangle$, we write $\langle \xi, D, \zeta \rangle \Rightarrow_M \langle \xi', D', \zeta' \rangle$ if

- there are $\rho = (\ell, q, r) \in R$ and $(v, w) \in D$ with $v \in \mathrm{pos}_q(\xi)$ and $w \in \mathrm{pos}_q(\zeta)$ such that $\xi' = \xi[\ell]_v$, $\zeta' = \zeta[r]_w$, and $D' = D \cup \mathrm{links}_{v,w}(\rho)$, or
- there are $v \in \mathrm{pos}_Q(\xi)$ and $t \in c(\xi(v))$ with $w \notin \mathrm{pos}_Q(\zeta)$ for all $(v, w) \in D$ such that $\xi' = \xi[t]_v$, $\zeta' = \zeta$, and $D' = D$.

The l-xt$^{\mathrm{R}}$ M computes the dependencies

$$\mathrm{dep}(M) = \{(t, D, u) \in T_\Sigma \times \mathcal{L} \times T_\Delta \mid \exists q \in I \colon \langle q, \{(\varepsilon, \varepsilon)\}, q \rangle \Rightarrow_M^* (t, D, u)\} \ ,$$

where $\varepsilon \in \mathbb{N}^*$ is the empty word and $\Rightarrow_M^*$ is the reflexive, transitive closure of $\Rightarrow_M$. It also computes the link structures $\mathrm{links}(M) = \mathrm{links}(\mathrm{dep}(M))$ and the tree transformation $M = \{(t, u) \mid (t, D, u) \in \mathrm{dep}(M)\}$.

A few derivation steps using M_1 of Example 2 are illustrated in Fig. 1. Since every translation $(t, u) \in M$ is ultimately created by (at least) one successful derivation, we can inspect the links in the derivation process to obtain dependencies, which were called *contributions* in [7]. We use stem-capitalized versions of the abbreviations for the corresponding classes of computed tree transformations. For instance, dnl-XT is the class of all tree transformations computable by dnl-xt. The regular look-ahead is useless for nondeleting l-xt$^{\mathrm{R}}$ (because $Q^{\mathrm{la}} = \emptyset$), and thus nl-XT$^{\mathrm{R}}$ = nl-XT and similarly for the non-extended case and for all defined subclasses. Finally, we use the brackets '[' and ']' for optional use of the restrictions $\not\varepsilon$, 's', 'd', and 'n' that have to be consistently applied.

Next, we relate the class l-XT$^{\mathrm{R}}$ to l-T$^{\mathrm{R}}$, which tells us how to emulate linear extended top-down tree transducers with regular look-ahead by linear top-down tree transducers with regular look-ahead. To illustrate the consistent application of optional restrictions, we observe that $\not\varepsilon$l-XT$^{\mathrm{R}}$ = snl-H^{-1} ; l-T$^{\mathrm{R}}$ and $\not\varepsilon$sdl-XT$^{\mathrm{R}}$ = snl-H^{-1}; $\not\varepsilon$sdl-T$^{\mathrm{R}}$ are instances of the first result of the next theorem.

Theorem 5 ([11, Lemma 4.1 and Corollary 4.1]).

$$\not\in[\mathrm{s}][\mathrm{d}][\mathrm{n}]\text{l-XT}^{\mathrm{R}} = \text{snl-H}^{-1} \,;\, [\mathrm{s}][\mathrm{d}][\mathrm{n}]\text{l-T}^{\mathrm{R}} \quad [\mathrm{s}][\mathrm{d}][\mathrm{n}]\text{l-XT}^{\mathrm{R}} = \text{nl-H}^{-1} \,;\, [\mathrm{s}][\mathrm{d}][\mathrm{n}]\text{l-T}^{\mathrm{R}}$$

4 Our Classes Are Closed at a Finite Power

In this section, we show that the classes $\not\in$l-XT$^{\mathrm{R}}$, $\not\in$l-XT, $\not\in$sl-XT$^{\mathrm{R}}$, and $\not\in$sl-XT are closed under composition at a finite power. We first recall a central result of [3]. Note that [3] expresses this result in terms of a class B of bimorphisms, but $\not\in$snl-XT = B by [2] and [18, Theorem 4].

Theorem 6 ([3, Theorem 6.2]). $\not\in\text{snl-XT} \subsetneq \not\in\text{snl-XT}^2 = \not\in\text{snl-XT}^n$ for $n \geq 3$.

Now we establish our first composition result, which is analogous to the classical composition result for linear top-down tree transducers with regular look-ahead [6, Theorem 2.11]. The only difference is that our first transducer has extended left-hand sides (i.e., it is an l-xt$^{\mathrm{R}}$ instead of just an l-t$^{\mathrm{R}}$).

Lemma 7. $[\not\in][\mathrm{s}][\mathrm{d}][\mathrm{n}]\text{l-XT}^{\mathrm{R}} \,;\, [\mathrm{s}][\mathrm{d}][\mathrm{n}]\text{l-T}^{\mathrm{R}} = [\not\in][\mathrm{s}][\mathrm{d}][\mathrm{n}]\text{l-XT}^{\mathrm{R}}$

Proof. Immediate, from Theorem 5 and the classical composition result for l-T$^{\mathrm{R}}$ in [6, Theorem 2.11][3], which states $[\mathrm{s}][\mathrm{d}][\mathrm{n}]\text{l-T}^{\mathrm{R}} \,;\, [\mathrm{s}][\mathrm{d}][\mathrm{n}]\text{l-T}^{\mathrm{R}} = [\mathrm{s}][\mathrm{d}][\mathrm{n}]\text{l-T}^{\mathrm{R}}$. □

Next, we present a decomposition that corresponds to property P of [5, Section II-2-2-3-2]. It demonstrates how to simulate an $\not\in$l-xt$^{\mathrm{R}}$ by a delabeling l-t$^{\mathrm{R}}$ and an $\not\in$snl-xt, for which we have the composition closure result in Theorem 6. We immediately combine the result with Lemma 7 to demonstrate, how we can shift an [s]dl-t$^{\mathrm{R}}$ from the back to the front.

Lemma 8. $\not\in[\mathrm{s}]\text{l-XT}^{\mathrm{R}} \,;\, [\mathrm{s}]\text{dl-T}^{\mathrm{R}} \subseteq \not\in[\mathrm{s}]\text{l-XT}^{\mathrm{R}} \subseteq [\mathrm{s}]\text{dl-T}^{\mathrm{R}} \,;\, \not\in\text{snl-XT}$

Proof. The first inclusion is due to Lemma 7. For the second inclusion, assume that M is ε-free. Moreover, let $m \in \mathbb{N}$ be such that $m \geq |\mathrm{var}(r)|$ for every $(\ell, q, r) \in R$. For every rule $\rho = (\ell, q, r) \in R$ and non-state position $w \in \mathrm{pos}_\Sigma(\ell)$, let $\mathrm{used}_\rho(w) = \{i \in \mathbb{N} \mid wi \in \mathrm{pos}(\ell), \mathrm{var}(\ell|_{wi}) \cap \mathrm{var}(r) \neq \emptyset\}$. We construct a dl-xt$^{\mathrm{R}}$ $M' = (Q', \Sigma, \Gamma, I', R', c')$ such that

- $Q' = \{\langle \rho, w\rangle \mid \rho = (\ell, q, r) \in R, w \in \mathrm{pos}(\ell)\}$ and $I' = \{\langle \rho, \varepsilon\rangle \mid q \in I, \rho \in R_q\}$,
- $\Gamma = \{\rho^{(|\mathrm{used}_\rho(\varepsilon)|)} \mid \rho \in R\} \cup \{@_i^{(i)} \mid 0 \leq i \leq m\}$,
- for every rule $\rho = (\ell, q, r) \in R$ and non-state position $w \in \mathrm{pos}_\Sigma(\ell)$, the rule

$$\ell(w)(\langle \rho, w1\rangle, \ldots, \langle \rho, wk\rangle) \xrightarrow{\langle \rho, w\rangle} \begin{cases} \langle \rho, wi_1\rangle & \text{if } r \in Q \\ \rho(\langle \rho, wi_1\rangle, \ldots, \langle \rho, wi_n\rangle) & \text{if } r \notin Q, w = \varepsilon \\ @_n(\langle \rho, wi_1\rangle, \ldots, \langle \rho, wi_n\rangle) & \text{otherwise,} \end{cases}$$

  is in R', where $\ell(w) \in \Sigma_k$ and $\{i_1, \ldots, i_n\} = \mathrm{used}_\rho(w)$ with $i_1 < \cdots < i_n$,

[3] The abbreviation 'd' has a completely different meaning in [6].

- for every rule $\rho = (\ell, q, r) \in R$, (non-deleted) state position $w \in \mathrm{pos}_{\mathrm{var}(r)}(\ell)$, and rule $\rho' \in R_{\ell(w)}$, the rule $\langle \rho', \varepsilon \rangle \xrightarrow{\langle \rho, w \rangle} \langle \rho', \varepsilon \rangle$ is in R', and
- $c'(\langle \rho, w \rangle) = \ell|_w[q \leftarrow c(q) \mid q \in \mathrm{var}(\ell|_w)]$ for every potentially deleted state $\langle \rho, w \rangle \in \{\langle \rho, w \rangle \in Q \mid \mathrm{used}_\rho(w) = \emptyset\}$, where $\leftarrow$ denotes the standard OI-substitution [8].

To obtain the desired dl-t$^{\mathrm{R}}$ we simply eliminate the ε-rules using standard methods.[4] Intuitively speaking, the transducer M' processes the input and deletes subtrees that are not necessary for further processing. Moreover, it executes nonstrict rules of M and marks the positions in the input where a strict rule application would be possible. It remains to construct the l-xt M''. Let $m'' \geq |\ell|$ for all $(\ell, q, r) \in R$, and let $M'' = (\{\star\}, \Gamma, \Delta, \{\star\}, R'')$ such that R'' contains all valid rules $\rho(t_1, \ldots, t_k) \xrightarrow{\star} r[q \leftarrow \star \mid q \in Q]$ of a strict nondeleting l-xt with $\rho = (\ell, q, r) \in R$, $\mathrm{pos}_R(t_i) = \emptyset$, and $|t_i| \leq m''$ for every $1 \leq i \leq k$, where k is the rank of ρ. □

Example 9. Let $\rho = \sigma(p, \sigma(\alpha, q)) \xrightarrow{q} \sigma(\alpha, \sigma(q, \alpha))$ be a rule with non-trivial look-ahead $c(p) = L$. We illustrate the construction of M' (in Lemma 8):

$$\sigma(\langle \rho, 1 \rangle, \langle \rho, 2 \rangle) \xrightarrow{\langle \rho, \varepsilon \rangle} \rho(\langle \rho, 2 \rangle) \qquad \alpha \xrightarrow{\langle \rho, 21 \rangle} @_0$$
$$\sigma(\langle \rho, 21 \rangle, \langle \rho, 22 \rangle) \xrightarrow{\langle \rho, 2 \rangle} @_1(\langle \rho, 22 \rangle) \qquad \langle \rho', \varepsilon \rangle \xrightarrow{\langle \rho, 22 \rangle} \langle \rho', \varepsilon \rangle$$

for all rules $\rho' \in R_q$. Moreover, the look-ahead c' of M' is such that $c'(\langle \rho, 1 \rangle) = L$ and $c'(\langle \rho, 21 \rangle) = \{\alpha\}$.

Theorem 10. $(\not\varepsilon\text{[s]l-XT}^{\mathrm{R}})^n \subseteq \text{[s]dl-T}^{\mathrm{R}} \,;\, \not\varepsilon\text{snl-XT}^2 \subseteq (\not\varepsilon\text{[s]l-XT}^{\mathrm{R}})^3$ for $n \geq 1$.

Proof. The second inclusion is trivial. We prove the first inclusion by induction over n. For $n = 1$, it follows from Lemma 8, and in the induction step, we obtain

$$\begin{aligned}(\not\varepsilon\text{[s]l-XT}^{\mathrm{R}})^{n+1} &\subseteq \not\varepsilon\text{[s]l-XT}^{\mathrm{R}} \,;\, \text{[s]dl-T}^{\mathrm{R}} \,;\, \not\varepsilon\text{snl-XT}^2 \\ &\subseteq \text{[s]dl-T}^{\mathrm{R}} \,;\, \not\varepsilon\text{snl-XT}^3 = \text{[s]dl-T}^{\mathrm{R}} \,;\, \not\varepsilon\text{snl-XT}^2\end{aligned}$$

by the induction hypothesis, then Lemma 8, and lastly Theorem 6. □

It is known [6, Theorem 2.6] that we can simulate every l-t$^{\mathrm{R}}$ (with look-ahead) by a composition of two l-t (without look-ahead). This allows us to conclude that the class $\not\varepsilon$l-XT is closed under composition at the fourth power.

Corollary 11. $\not\varepsilon\text{[s]l-XT}^n \subseteq \text{QR} \,;\, \text{[s]dl-T} \,;\, \not\varepsilon\text{snl-XT}^2 \subseteq \not\varepsilon\text{[s]l-XT}^4$ for every $n \geq 1$.

Proof. The second inclusion is trivial, and for the first inclusion we use Theorem 10 and [s]dl-T$^{\mathrm{R}} \subseteq$ QR ; [s]dl-T. □

[4] Note that due to the ε-freeness of M, we have $w \neq \varepsilon$ in the ε-rules of the fourth item. Since these rules are the only constructed ε-rules, we cannot chain two ε-rules.

In the rest of the section, we will show that the (strict) classes $\not\varepsilon$sl-XT$^{\mathrm{R}}$ and $\not\varepsilon$sl-XT are closed under composition already at the second power. This time, the main lemma demonstrates how to shift a strict delabeling linear homomorphism from the front to the back again creating a nondeleting transducer (cf. Lemma 8).

Lemma 12. sdl-H ; $\not\varepsilon$sl-XT $\subseteq$ $\not\varepsilon$sl-XT $\subseteq$ $\not\varepsilon$snl-XT ; sdl-H

Proof. For the first inclusion, let $d\colon T_\Gamma \to T_\Sigma$ be a strict delabeling linear tree homomorphism. Moreover, assume that M is a strict and ε-free l-xt, and let $m \in \mathbb{N}$ be such that $m \geq |\ell|$ for every $(\ell, q, r) \in R$. We construct the l-xt $M' = (Q', \Gamma, \Delta, I, R', c')$ with $Q' = Q \cup \{1, \ldots, m\}$ such that for every rule $(\ell, q, r) \in R$ we have each valid rule (ℓ', q, r) in R' where $\ell' \in d^{-1}(\ell)$ and $|\mathrm{pos}_\Gamma(\ell')| = |\mathrm{pos}_\Sigma(\ell)|$. Recall that d also defines a tree transformation $d\colon T_\Gamma(Q') \to T_\Sigma(Q')$, which acts as an identity on states; i.e., $d(q') = q'$ for every $q' \in Q'$. Moreover, $c'(q') = T_\Gamma$ for all $q' \in (Q')^{\mathrm{la}}$. Finally, we observe that M' is strict because it has the same right-hand sides as M, and it is ε-free because h is strict. For the second inclusion,

$$\not\varepsilon\text{sl-XT} \subseteq \text{snl-H}^{-1} ; \text{FTA} ; \text{sl-H} \subseteq \text{snl-H}^{-1} ; \text{FTA} ; \text{snl-H} ; \text{sdl-H} \subseteq \not\varepsilon\text{snl-XT} ; \text{sdl-H} \ ,$$

where the first and the last inclusion are by [18, Theorem 4] and the second inclusion is due to [5, Section I-2-1-3-5]. □

In contrast to Theorem 10 and Corollary 11, look-ahead does not increase the power of closedness in the strict case. In fact, the next theorem shows that $(\not\varepsilon\text{sl-XT}^{\mathrm{R}})^n = \not\varepsilon\text{sl-XT}^n$ for all $n \geq 2$.

Theorem 13. $(\not\varepsilon\text{sl-XT}^{\mathrm{R}})^n \subseteq \not\varepsilon\text{snl-XT} ; \not\varepsilon\text{sl-XT} \subseteq \not\varepsilon\text{sl-XT}^2$ for every $n \geq 1$.

Proof. Again, the second inclusion is trivial. For the first inclusion, we first prove that $\not\varepsilon$s[n]l-XT$^{\mathrm{R}}$; $\not\varepsilon$sl-XT$^{\mathrm{R}}$ = $\not\varepsilon$s[n]l-XT$^{\mathrm{R}}$; $\not\varepsilon$sl-XT, which we call (†), as follows:

$$\not\varepsilon\text{s[n]l-XT}^{\mathrm{R}} ; \not\varepsilon\text{sl-XT}^{\mathrm{R}} \subseteq \not\varepsilon\text{s[n]l-XT}^{\mathrm{R}} ; \text{QR} ; \not\varepsilon\text{sl-XT} \subseteq \not\varepsilon\text{s[n]l-XT}^{\mathrm{R}} ; \not\varepsilon\text{sl-XT} \ ,$$

where we used [6, Theorem 2.6] in the first step and Lemma 7 in the second step.[5] Now we prove the first inclusion of our main statement by induction on n. The induction basis ($n = 1$) follows from $\not\varepsilon$sl-XT$^{\mathrm{R}} \subseteq$ QR ; $\not\varepsilon$sl-XT [6, Theorem 2.6], and the induction step is proved as follows

$$\begin{aligned}(\not\varepsilon\text{sl-XT}^{\mathrm{R}})^{n+1} &\subseteq (\not\varepsilon\text{sl-XT}^{\mathrm{R}})^n ; \not\varepsilon\text{sl-XT} \subseteq \not\varepsilon\text{snl-XT} ; \not\varepsilon\text{sl-XT}^2 \subseteq \not\varepsilon\text{snl-XT}^3 ; \text{sdl-H} \\ &\subseteq \not\varepsilon\text{snl-XT}^2 ; \text{sdl-H} \qquad \subseteq \not\varepsilon\text{snl-XT} ; \not\varepsilon\text{sl-XT}^{\mathrm{R}} \subseteq \not\varepsilon\text{snl-XT} ; \not\varepsilon\text{sl-XT}\end{aligned}$$

using, in sequence, statement (†), the induction hypothesis, Lemma 12 twice, Theorem 6, Lemma 7, and statement (†) again. □

[5] The converse inclusion is trivial.

5 Least Power of Closedness

In this section, we will determine the least power at which the class is closed under composition for the classes $\not\varepsilon$l-XT$^{\mathrm{R}}$, $\not\varepsilon$sl-XT$^{\mathrm{R}}$, and $\not\varepsilon$sl-XT. In addition, we conjecture the least power for the class $\not\varepsilon$l-XT.

Theorem 14. For every $n \geq 3$

$$\not\varepsilon\text{sl-XT} \subsetneq \not\varepsilon\text{sl-XT}^{\mathrm{R}} \subsetneq \not\varepsilon\text{sl-XT}^2 = (\not\varepsilon\text{sl-XT}^{\mathrm{R}})^2 = \not\varepsilon\text{sl-XT}^n = (\not\varepsilon\text{sl-XT}^{\mathrm{R}})^n \ .$$

Proof. Theorem 13 proves the final three equalities. The first inclusion is trivial and strictness follows from the proof of [15, Lemma 4.3]. The second inclusion is also trivial (given the previous equalities) and the strictness follows from [18, Theorem 4] and [3, Section 3.4], which show that class $\not\varepsilon$sl-XT$^{\mathrm{R}}$ is not closed under composition at power 1.[6] □

Definition 15 ([19, Definitions 8 and 10]). A set $\mathcal{D} \subseteq \mathcal{L}$ of link structures

- is *input hierarchical*[7] if for every $D \in \mathcal{D}$ and $(v_1, w_1), (v_2, w_2) \in D$ we have (i) if $v_1 \prec v_2$, then $w_1 \preceq w_2$, and (ii) if $v_1 = v_2$, then $w_1 \preceq w_2$ or $w_2 \preceq w_1$.
- has *bounded distance in the input* if there exists an integer $k \in \mathbb{N}$ such that for every $D \in \mathcal{D}$ and all $(v, w), (vv'', w'') \in D$ there exists $(vv', w') \in D$ with $v' \prec v''$ and $|v'| \leq k$.

Moreover, $\mathcal{D}$ is *output hierarchical* (resp. has *bounded distance in the output*) if $\mathcal{D}^{-1}$ is input hierarchical (resp. has bounded distance in the input). If $\mathcal{D}$ fulfills both versions of the property, then we just call it *hierarchical* or *bounded distance*.

Corollary 16 (of Def. 4). links(M) is hierarchical with bounded distance.

We will consider the problem whether a tree transformation can be computed by two l-xt$^{\mathrm{R}}$. For this we specify certain links that are intuitively clear and necessary between nodes of input-output tree pairs. Then we consider whether this specification can be implemented by two l-xt$^{\mathrm{R}}$. Often we cannot identify the nodes of a link exactly. In such cases, we use splines with inverted arrow heads, which indicate that there is a link to some position of the subtree pointed to.

Theorem 17. For every $n \geq 4$,

$$\begin{aligned}\not\varepsilon\text{l-XT} \subsetneq \not\varepsilon\text{l-XT}^{\mathrm{R}} \subsetneq \not\varepsilon\text{l-XT}^2 \subseteq (\not\varepsilon\text{l-XT}^{\mathrm{R}})^2 \subsetneq \not\varepsilon\text{l-XT}^3 \subseteq (\not\varepsilon\text{l-XT}^{\mathrm{R}})^3 \\ = \not\varepsilon\text{l-XT}^4 = (\not\varepsilon\text{l-XT}^{\mathrm{R}})^n = \not\varepsilon\text{l-XT}^{n+1} \ .\end{aligned}$$

Proof. We have $(\not\varepsilon\text{l-XT}^{\mathrm{R}})^n \subseteq \not\varepsilon\text{l-XT}^{n+1}$ for all $n \geq 1$ by repeated application of Lemma 8. The equalities follow from Theorem 10, so we only have to prove strictness. The first inclusion is strict by [15, Lemma 4.3] and the strictness of the second inclusion follows from that of the fourth. Finally, we prove the

[6] In fact, Theorem 17 reproves this statement.
[7] This notion is called *strictly input hierarchical* in [19].

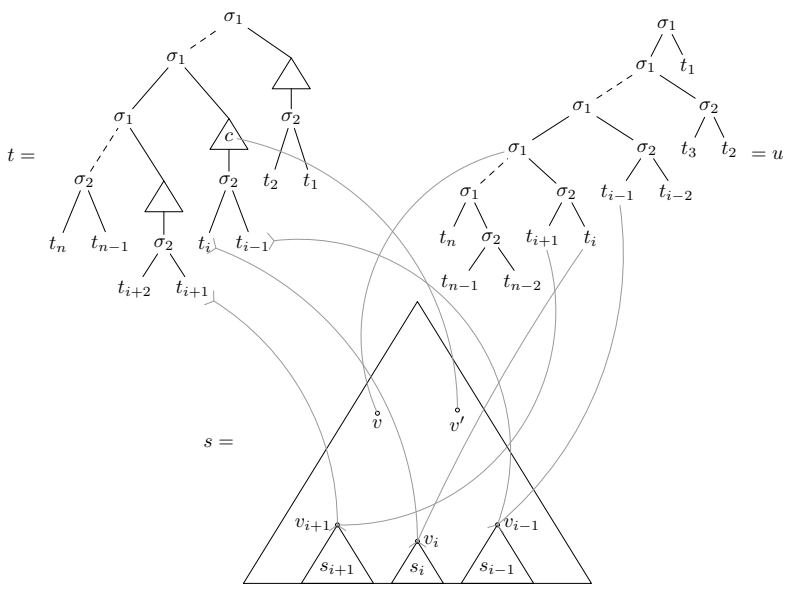

Fig. 2. The relevant part of the specification used in the proof of Theorem 17.

strictness of the fourth inclusion. For this, recall the l-t$^{\mathrm{R}}$ M_1 of Example 2. In addition, we use the two bimorphisms B_2 and B_3 of [5, Section II-2-2-3-1], which are in the class B mentioned before Theorem 6, and hence can also be defined by some $\not\varepsilon$snl-xt M_2 and M_3, respectively. For convenience, we present M_2 and M_3 explicitly before we show that $\tau = M_1 ; M_2 ; M_3$ cannot be computed by a composition of two $\not\varepsilon$l-xt$^{\mathrm{R}}$.

Let $M_2 = (\{\star, \mathrm{id}, \mathrm{id}'\}, \Sigma, \Sigma, \{\star\}, R_2)$ be the $\not\varepsilon$snl-xt with the rules

$$\sigma_1(\star, \sigma_2(\mathrm{id}, \mathrm{id}')) \xrightarrow{\star} \sigma(\sigma(\star, \mathrm{id}), \mathrm{id}') \qquad \sigma_2(\mathrm{id}, \mathrm{id}') \xrightarrow{\star} \sigma(\mathrm{id}, \mathrm{id}')$$
$$\gamma(\mathrm{id}) \xrightarrow{\mathrm{id},\mathrm{id}'} \gamma(\mathrm{id}) \qquad \alpha \xrightarrow{\mathrm{id},\mathrm{id}'} \alpha \ .$$

Moreover, let $M_3 = (\{\star, p, \mathrm{id}, \mathrm{id}'\}, \Sigma, \Sigma, \{\star\}, R_3)$ be the $\not\varepsilon$snl-xt with the rules

$$\sigma(p, \mathrm{id}) \xrightarrow{\star} \sigma_1(p, \mathrm{id}) \qquad \sigma(\sigma(p, \mathrm{id}), \mathrm{id}') \xrightarrow{p} \sigma_1(p, \sigma_2(\mathrm{id}, \mathrm{id}'))$$
$$\gamma(\mathrm{id}) \xrightarrow{p} \gamma(\mathrm{id}) \qquad \gamma(\mathrm{id}) \xrightarrow{\mathrm{id},\mathrm{id}'} \gamma(\mathrm{id}) \qquad \alpha \xrightarrow{p,\mathrm{id},\mathrm{id}'} \alpha \ .$$

We present a proof by contradiction, hence we assume that $\tau = N_1 ; N_2$ for some $\not\varepsilon$l-xt$^{\mathrm{R}}$ $N_1 = (P_1, \Sigma, \Delta, I_1, R_1')$ and $N_2 = (P_2, \Delta, \Sigma, I_2, R_2')$. Using a standard construction, we can construct an ε-cycle free l-xt$^{\mathrm{R}}$ $N_2' = (P_2', \Delta, \Sigma, I_2', R_2'')$ such that $N_2' = N_2$ and each rule $\ell \xrightarrow{p} r \in R_2''$ that contains γ in its right-hand side r obeys $r = \gamma(p)$ with $p \in P_2'$. With the help of Corollary 16, we can further conclude that links(N_1) and links(N_2') are hierarchical with bounded distance k_1 and k_2, respectively. Moreover, let

$$m \geq \max\{k_1, k_2, |\ell|, |r| \mid \ell \xrightarrow{p} r \in R_1' \cup R_2''\} \ .$$

Clearly, all $(t, u) \in \tau$ have the shape shown in Fig. 2. Next, we will make an assumption, derive the contradiction, and then prove the assumption. Suppose that there exists $(t, u) \in \tau$ such that (see Fig. 2 for the named subtrees)

- the left σ_1-spine of t is longer than m,
- for all trees $c' \in T_\Sigma(\{x_1\})$ indicated by small triangles in t (like c in Fig. 2) the only element of $\mathrm{pos}_{x_1}(c')$ is longer than m, and
- for all $(t, D_1, s) \in \mathrm{dep}(N_1)$, $(s, D_2, u) \in \mathrm{dep}(N_2')$, and $1 \leq j \leq n$ we have
  - there exists $(v_j, w_j) \in D_2$ such that w_j is the root of t_j in u, and
  - there exists $(y_j, v_j') \in D_1$ such that y_j is a position inside t_j and $v_j \preceq v_j'$.

Since the left σ_1-spine in u is longer than k_2 and there are links at the root (i.e., $(\varepsilon, \varepsilon) \in D_2$) and at w_n, there must be a linking point at position $w \in \mathrm{pos}_{\sigma_1}(u)$ along the left σ_1-spine with $w \neq \varepsilon$, which links to position v in the intermediate tree s (i.e., $(v, w) \in D_2$). Let $u|_w = \sigma_1(u', \sigma_2(t_{i+1}, t_i))$ for some $2 \leq i \leq n-2$. By our assumption, there exist links $(v_{i+1}, w_{i+1}), (v_i, w_i), (v_{i-1}, w_{i-1}) \in D_2$. Since D_2 is hierarchical and w_{i+1} and w_i are below w in u, we know that v_{i+1} and v_i are below v in s (i.e., $v \preceq v_{i+1}, v_i$), whereas $v \not\preceq v_{i-1}$. Next, we locate t_i in the input tree t. By the general shape of t, the subtree t_i occurs in a subtree $\sigma_1(t', c[\sigma_2(t_i, t_{i-1})])$ for some tree $c \in T_\Sigma(\{x_1\})$ with exactly one occurrence of x_1. We know that c is suitably large, which forces a linking point y inside c in addition to those in t_{i+1}, t_i, and t_{i-1}, which exist by the assumption. Note that y is a proper prefix of the root position of the subtree $\sigma_2(t_i, t_{i-1})$. Let $(y, v') \in D_1$ be the link linking c to s, which dominates the links $(y_i, v_i'), (y_{i-1}, v_{i-1}') \in D_1$ linking t_i and t_{i-1} to s, respectively. Thus, $v' \preceq v_i', v_{i-1}'$ and $v' \not\preceq v_{i+1}'$ because $y \not\preceq y_{i+1}$. Obviously, $v' \not\preceq v_{i+1}$, and moreover, $v' \preceq v_i, v_{i-1}$ because otherwise the positions v_{i+1}, v_i, v_{i-1} would not be incomparable, which is required because $\mathrm{links}(N_2')$ is hierarchical. We have either $\mathrm{lcp}(v_{i+1}, v_i) \preceq \mathrm{lcp}(v_i, v_{i-1})$ or $\mathrm{lcp}(v_i, v_{i-1}) \preceq \mathrm{lcp}(v_{i+1}, v_i)$. We either get $v \preceq \mathrm{lcp}(v_{i+1}, v_i) \preceq \mathrm{lcp}(v_i, v_{i-1}) \preceq v_{i-1}$ or $v' \preceq \mathrm{lcp}(v_i, v_{i-1}) \preceq \mathrm{lcp}(v_{i+1}, v_i) \preceq v_{i+1}$, which are both contradictions.

It remains to show the assumption. Obviously, the first two items can be satisfied simply by a proper selection of $(t, u) \in \tau$. For every $1 \leq j \leq n$, we know that there exists a link $(v_j, w_j) \in D_2$ to the root of t_j in u due to the special shape of the right-hand sides of N_2'. We note that all $v_1, \ldots, v_n$ are pairwise incomparable. Moreover, we observe that there is a linear height (and size) relation between input and output trees related by a link, which is true for all ε-cycle free l-xt. Consequently, there is a linear height relation between $s_j = s|_{v_j}$ and $t_j = u|_{w_j}$. Thus by selecting each t_j suitably tall, we can enforce that each s_j is taller than m, which yields that there is a link $(y_j, v_j') \in D_1$ such that $v_j \preceq v_j'$. Exploiting the linear height (and size) relation between linked subtrees again, we can additionally show that (i) y_j is a position inside t_j in t, in which case we are done, or (ii) y_j is a prefix of the root position of t_j in t. In the latter case, the size of t_j can be chosen such that there is also a link (y_j', v_j'') with $y_j \prec y_j'$ and $v_j' \preceq v_j''$. Moreover, this can be iterated until y_j' points to a position inside t_j. A detailed proof of these statements can be found in [9]. □

We conjecture that $\not\varepsilon$l-$\mathrm{XT}^3 \subsetneq \not\varepsilon$l-$\mathrm{XT}^4 = \not\varepsilon$l-$\mathrm{XT}^n$ for every $n \geq 4$. The inclusion is trivial and the equality follows from Corollary 11. For the strictness,

the proof of Theorem 17 essentially shows that in the first step we must delete the contexts indicated by triangles (such as c) in Fig. 2 because otherwise we can apply the method used in the proof to derive a contradiction (it relies on the existence of a linking point inside such a context c). Thus, in essence we must first implement a variant of the $\not\in$l-xt$^{\mathrm{R}}$ M_1 of Example 2. It is a simple exercise to show that the deletion of the excess material cannot be done by a single l-xt as it cannot reliably determine the left-most occurrence of σ_2 without the look-ahead. Thus, if we only have l-xt to achieve the transformation, then we already need a composition of two l-xt to perform the required deletion.

For the sake of completeness we mention the following. In the full version [9] of this paper we prove that the composition hierarchy is infinite for all other combinations of '$\not\in$', 's', and 'n'.

Theorem 18 ([9, Theorem 34]). The composition hierarchy of the classes $\not\in$nl-XT, [s][n]l-XT$^{\mathrm{R}}$, and [s][n]l-XT is infinite.

Acknowledgment The authors are indebted to an anonymous referee for his valuable report.

References

1. Arnold, A., Dauchet, M.: Transductions inversibles de forêts. Thèse 3ème cycle M. Dauchet, Université de Lille (1975)
2. Arnold, A., Dauchet, M.: Bi-transductions de forêts. In: ICALP. pp. 74–86. Edinburgh University Press (1976)
3. Arnold, A., Dauchet, M.: Morphismes et bimorphismes d'arbres. Theoret. Comput. Sci. 20(1), 33–93 (1982)
4. Chiang, D.: An introduction to synchronous grammars. In: ACL. Association for Computational Linguistics (2006), part of a tutorial given with K. Knight
5. Dauchet, M.: Transductions de forêts — Bimorphismes de magmoïdes. Première thèse, Université de Lille (1977)
6. Engelfriet, J.: Top-down tree transducers with regular look-ahead. Math. Systems Theory 10(1), 289–303 (1977)
7. Engelfriet, J., Maneth, S.: Macro tree translations of linear size increase are MSO definable. SIAM J. Comput. 32(4), 950–1006 (2003)
8. Engelfriet, J., Schmidt, E.M.: IO and OI I. J. Comput. System Sci. 15(3), 328–353 (1977)
9. Fülöp, Z., Maletti, A.: Composition closure of linear extended top-down tree transducers (2013), manuscript available at: `http://arxiv.org/abs/1301.1514`
10. Fülöp, Z., Maletti, A., Vogler, H.: Preservation of recognizability for synchronous tree substitution grammars. In: ATANLP. pp. 1–9. Association for Computational Linguistics (2010)
11. Fülöp, Z., Maletti, A., Vogler, H.: Weighted extended tree transducers. Fundam. Inform. 111(2), 163–202 (2011)
12. Fülöp, Z., Vogler, H.: Syntax-Directed Semantics — Formal Models Based on Tree Transducers. Springer (1998)
13. Gécseg, F., Steinby, M.: Tree Automata. Akadémiai Kiadó, Budapest (1984)

14. Gécseg, F., Steinby, M.: Tree languages. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, vol. 3, chap. 1, pp. 1–68. Springer (1997)
15. Graehl, J., Hopkins, M., Knight, K., Maletti, A.: The power of extended top-down tree transducers. SIAM J. Comput. 39(2), 410–430 (2009)
16. Graehl, J., Knight, K., May, J.: Training tree transducers. Comput. Linguist. 34(3), 391–427 (2008)
17. Knight, K., Graehl, J.: An overview of probabilistic tree transducers for natural language processing. In: CICLing. LNCS, vol. 3406, pp. 1–24. Springer (2005)
18. Maletti, A.: Compositions of extended top-down tree transducers. Inform. and Comput. 206(9–10), 1187–1196 (2008)
19. Maletti, A.: Tree transformations and dependencies. In: MOL. LNAI, vol. 6878, pp. 1–20. Springer (2011)
20. May, J., Knight, K., Vogler, H.: Efficient inference through cascades of weighted tree transducers. In: ACL. pp. 1058–1066. Association for Computational Linguistics (2010)
21. Rounds, W.C.: Mappings and grammars on trees. Math. Systems Theory 4(3), 257–287 (1970)
22. Thatcher, J.W.: Generalized^2 sequential machine maps. J. Comput. System Sci. 4(4), 339–367 (1970)