

Composition Closure of Linear Extended Top-down Tree Transducers

Joost Engelfriet · Zoltán Fülöp ·
Andreas Maletti

Abstract Linear extended top-down tree transducers (or synchronous tree-substitution grammars) are popular formal models of tree transformations that are extensively used in syntax-based statistical machine translation. The expressive power of compositions of such transducers with and without regular look-ahead is investigated. In particular, the restrictions of ε -freeness, strictness, and nondeletion are considered. The composition hierarchy turns out to be finite for all ε -free (all rules consume input) variants of these transducers except for the nondeleting ε -free transducers. The least number of transducers needed for the full expressive power of arbitrary compositions is presented. In all remaining cases (incl. the nondeleting ε -free transducers) the composition hierarchy does not collapse.

Keywords extended top-down tree transducer; composition hierarchy; bimorphism

This is a revised and extended version of [Z. FÜLÖP and A. MALETTI: *Composition closure of ε -free linear extended top-down tree transducers*. In Proc. 17th DLT, volume 7907 of LNCS, pages 239–251. Springer-Verlag, 2013].

This work was partially supported by the exchange project 55 657 of the German Academic Exchange Service (DAAD) and Hungarian Scholarship Board Office (MÖB). Z. Fülöp was partially supported by the Hungarian Scientific Research Fund (OTKA) grant K 108 448, and A. Maletti was partially supported by the German Research Foundation (DFG) grant MA / 4959 / 1-1.

J. Engelfriet
Leiden Institute of Advanced Computer Science, Leiden University, P.O. Box 9512,
2300 RA Leiden, The Netherlands. E-mail: j.engelfriet@liacs.leidenuniv.nl

Z. Fülöp
Department of Foundations of Computer Science, University of Szeged, Árpád tér 2,
H-6720 Szeged, Hungary. E-mail: fulop@inf.u-szeged.hu

A. Maletti
Institute of Computer Science, Universität Leipzig, Augustusplatz 10–11,
04109 Leipzig, Germany. E-mail: maletti@informatik.uni-leipzig.de

CR Subject Classification Theory of computation—Formal languages and automata theory—Tree languages

1 Introduction

Top-down tree transducers are simple formal models that encode tree transformations (i.e., relations between trees). They were introduced in [23, 24] and intensively studied thereafter (see [15, 16, 14] 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 and output (single symbol vs. tree fragment) was removed in extended top-down tree transducers (xt), which were introduced and studied in [1, 2]. In such a transducer the input side of a rule can now also contain a tree fragment, in which each variable can occur at most once as a placeholder for a subtree. In particular, the tree fragment can even be just a variable, which matches every input tree, and such rules are called ε -rules because they do not process any part of the input tree. In this contribution we only consider linear xt (l-xt), in which the output 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 [18, 19].

We also add regular look-ahead [7] (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 is the linear extended top-down tree transducer with regular look-ahead (l-xt^R). Contrary to most of the literature [7, 17] we present our models as synchronous grammars [4] because we sometimes use the auxiliary link structure in our proofs. Instead of variables in the input side and a state-variable combination in the output side of a rule, we immediately only use states with the restriction that all states that occur in the output side must also occur in the input side. Moreover, each state that occurs in both sides, must occur exactly once in the input side and exactly once in the output side, which corresponds to the classical linearity condition. In this way, for each rule the states establish links (a state links its occurrence in the output side with its occurrence in the input side), which form an injection from the state occurrences in the output side to the state occurrences in the input side. Regular look-ahead is specified only for the state occurrences (in the input side) that do not participate in the injection (i.e., those states that exclusively occur in the input side). A derivation of the grammar simultaneously generates an input tree and an output tree, which can contain states that are (possibly) linked by explicit links. A rule application expands two linked state occurrences at the same time, thus generating new input and output fragments with new (linked) state occurrences. Moreover, every unlinked state (in the input tree) is expanded into a tree from its regular look-ahead. Example 2 shows an l-xt^R, for which we illustrate a few derivation steps in Fig. 2. The tree transformation computed by the example l-xt^R is described in Example 8. In the following, we

use l-XTR and l-XT to denote the class of all tree transformations computed by l-xt^{R} and l-xt , respectively.

The expressive power of the various subclasses of l-XTR is already well understood [17, 13]. However, in practice complex systems are often specified with the help of compositions of tree transformations [22] because it is much easier to develop (or train) small components that manage a part of the overall transformation. Consequently, [19] 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 a class \mathcal{C} of tree transformations is closed under composition, then any composition chain $\tau_1; \dots; \tau_n$ of tree transformations τ_1, \dots, τ_n of \mathcal{C} can be replaced by a single tree transformation $\tau \in \mathcal{C}$. If \mathcal{C} represents the class of all tree transformations computable by a device, then closure 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-XTR nor l-XT is closed under composition [2, 3, 17].

In general, for a class \mathcal{C} of tree transformations (that contains the identity transformations) we obtain a composition hierarchy $\mathcal{C} \subseteq \mathcal{C}^2 \subseteq \mathcal{C}^3 \subseteq \dots$, where \mathcal{C}^n denotes the class of n -fold compositions of transformations from \mathcal{C} . The class \mathcal{C} might be closed under composition at power n (i.e., $\mathcal{C}^n = \mathcal{C}^{n+1}$) or its composition hierarchy might be infinite (i.e., $\mathcal{C}^n \subsetneq \mathcal{C}^{n+1}$ for all n). The former case yields that $\mathcal{C}^n = \mathcal{C}^m$ for all $m \geq n$, which means that the composition hierarchy of \mathcal{C} collapses at power n . In particular, \mathcal{C} is closed under composition if its composition hierarchy collapses at power 1. We note that in practice (e.g., in statistical machine translation) the classes that are closed under composition at a small power are also important because for such classes we can limit the length of composition chains [22]. In this contribution, we investigate the composition hierarchy of the classes l-XTR and l-XT together with their subclasses determined by any combination of the properties: ε -freeness, strictness, and nondeletion, which are abbreviated by ‘ ε ’, ‘s’, and ‘n’, respectively. Roughly speaking, ε -freeness requires that there are no ε -rules, strictness guarantees that the output side of each rule contains at least one output symbol, and nondeletion requires that for each rule exactly the same states occur in the input and output side. We use the property abbreviations in front of l-XTR and l-XT to obtain the class of all tree transformations computable by such restricted l-xt^{R} and l-xt , respectively. For instance, $\varepsilon\text{sl-XTR}$ denotes the class of all tree transformations computed by ε -free and strict l-xt^{R} .

It is known that none of our considered classes is closed under composition [3, Section 3.4]. In addition, it is known [3, Theorem 6.2] that the class $\varepsilon\text{snl-XT} = \varepsilon\text{snl-XTR}$ is closed at power 2. We complete the picture as follows. For each of the remaining classes, we either provide the least power at which the class is closed under composition or show that the composition hierarchy of the class is infinite (denoted by ∞). Our results (together with the mentioned existing result) are presented in Table 1.

Our contribution is organized as follows. Section 2 recalls the necessary concepts and introduces our notation. We continue in Section 3 with the for-

Class	Least power of closedness	Stated in
$\not\text{snl-XT} = \not\text{snl-XT}^{\text{R}}$	2	[3, Theorem 6.2]
$\not\text{sl-XT}^{\text{R}}, \not\text{sl-XT}$	2	Theorem 26
$\not\text{l-XT}^{\text{R}}$	3	Theorem 34
$\not\text{l-XT}$	4	Theorem 34
otherwise	∞	Theorem 45

Table 1 Characterization of the composition hierarchies.

mal introduction of our main model (l-xt^{R}) including its syntax and semantics and the restrictions that we consider later. In addition, we recall some known equalities between certain fundamental classes of tree transformations in preparation for our first main results. In Section 4 we give a power at which the classes $\not\text{sl-XT}$, $\not\text{sl-XT}^{\text{R}}$, $\not\text{l-XT}$, and $\not\text{l-XT}^{\text{R}}$ of tree transformations are closed under composition (see Table 1). This is completed in Section 5, where we conclude that these powers are minimal. In Section 6 we prove that the composition hierarchy of each of the remaining classes is infinite. Finally, we present the HASSE diagram of all the ε -free classes in Section 7.

2 Preliminaries

We denote the set of all nonnegative integers by \mathbb{N} . In the following, let S be a set. The *power set* of S is the set $\mathcal{P}(S) = \{S' \mid S' \subseteq S\}$ of all subsets of S . For an element s of S , we identify the singleton set $\{s\}$ with s , whenever convenient; this should not lead to confusion. The cardinality of S is denoted by $|S|$. The set of all *words* (finite sequences) over S is $S^* = \bigcup_{n \in \mathbb{N}} S^n$, where $S^0 = \{\varepsilon\}$ contains only the *empty word* ε . The *length* of a word $w \in S^*$ is the unique $n \in \mathbb{N}$ such that $w \in S^n$. We write $|w|$ for the length of w . The *concatenation* of two words $v, w \in S^*$ is denoted by $v.w$ or simply vw .

For sets S and T , 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 is the relation $R_1^{-1} = \{(t, s) \mid (s, t) \in R_1\}$, the *domain* of R_1 is

$$\text{dom}(R_1) = \{s \in S \mid \exists t \in T: (s, t) \in R_1\} ,$$

and the *composition* of R_1 and R_2 is the relation

$$R_1 ; R_2 = \{(s, u) \mid \exists t \in T: (s, t) \in R_1, (t, u) \in R_2\} \subseteq S \times U .$$

Given a relation $R \subseteq S \times S$, the *powers* of R are defined by $R^0 = \{(s, s) \mid s \in S\}$ and $R^{n+1} = R^n ; R$ for $n \in \mathbb{N}$. The reflexive and transitive closure of R is $R^* = \bigcup_{n \in \mathbb{N}} R^n$. These notions and notations are lifted to classes \mathcal{C}_1 and \mathcal{C}_2 of relations in the usual manner. Namely, we let $\mathcal{C}_1^{-1} = \{R_1^{-1} \mid R_1 \in \mathcal{C}_1\}$ and

$\mathcal{C}_1 ; \mathcal{C}_2 = \{R_1 ; R_2 \mid R_1 \in \mathcal{C}_1, R_2 \in \mathcal{C}_2\}$. 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 $n \geq 1$. Note that we do not consider the 0-th power for classes. The *composition hierarchy* [resp. *composition closure*] of \mathcal{C} is the family $(\mathcal{C}^n \mid n \geq 1)$ [resp. the class $\bigcup_{n \geq 1} \mathcal{C}^n$]. The classes \mathcal{C} of tree transformations that we will discuss always contain the identity relations. For such a class, $\mathcal{C}^n \subseteq \mathcal{C}^{n+1}$ for all $n \geq 1$. If $\mathcal{C}^n = \mathcal{C}^{n+1}$, then \mathcal{C} is *closed under composition at power n* . For $n = 1$ we shorten this to just \mathcal{C} is *closed under composition*. If \mathcal{C} is closed under composition at power n , then \mathcal{C}^n is the composition closure of \mathcal{C} .

An *alphabet* Σ is a nonempty and finite set, of which the elements are called *symbols*. The alphabet Σ is *ranked* if there additionally is a mapping $\text{rk} : \Sigma \rightarrow \mathbb{N}$ that assigns a rank to each symbol. We let

$$\Sigma_k = \{\sigma \in \Sigma \mid \text{rk}(\sigma) = k\}$$

for every $k \in \mathbb{N}$. Often the mapping ‘rk’ is obvious from the context, so we typically denote ranked alphabets by Σ alone. If it is not obvious, then we use the notation $\sigma^{(k)}$ to indicate that the symbol σ has rank k . For the rest of this paper, Σ , Δ , and Γ will denote arbitrary ranked alphabets if not specified otherwise.

For every set T , let $\Sigma(T) = \{\sigma(t_1, \dots, t_k) \mid k \in \mathbb{N}, \sigma \in \Sigma_k, t_1, \dots, t_k \in T\}$. Instead of $\sigma()$ with $\sigma \in \Sigma_0$ we will simply write σ . Let S be a set of “states” with $S \cap \Sigma = \emptyset$, to be used as additional leaf labels. The set $T_\Sigma(S)$ of Σ -trees with states in 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 any subset of $T_\Sigma(S)$ is a *tree language*. Given a unary symbol $\gamma \in \Sigma_1$ and a tree $t \in T_\Sigma(S)$, we write $\gamma^k(t)$ for the tree $\gamma(\dots\gamma(t)\dots)$, in which γ occurs k times on top of t .

The set $\text{pos}(t) \subseteq \mathbb{N}^*$ of *positions* of $t \in T_\Sigma(S)$ is inductively defined by $\text{pos}(s) = \{\varepsilon\}$ for every $s \in S$ and

$$\text{pos}(\sigma(t_1, \dots, t_k)) = \{\varepsilon\} \cup \bigcup_{i=1}^k \{iw \mid w \in \text{pos}(t_i)\}$$

for every $k \in \mathbb{N}$, $\sigma \in \Sigma_k$, and $t_1, \dots, t_k \in T_\Sigma(S)$. The positions of t are partially ordered by the *prefix order* \preceq on \mathbb{N}^* ; i.e., for words $w_1, w_2 \in \mathbb{N}^*$, we have $w_1 \preceq w_2$ if and only if there exists $w'_1 \in \mathbb{N}^*$ such that $w_1 w'_1 = w_2$. As usual we write $w_1 \prec w_2$ if w_1 is a proper prefix of w_2 ; i.e., $w_1 \preceq w_2$ and $w_1 \neq w_2$. For words $w_1, w_2 \in \mathbb{N}^*$, we denote the longest common prefix of w_1 and w_2 by $\text{lcp}(w_1, w_2)$. Note that $\text{lcp}(w_1, w_2) \in \text{pos}(t)$ for all $w_1, w_2 \in \text{pos}(t)$ because $\text{pos}(t)$ is prefix-closed. The *size* $|t|$ of a tree $t \in T_\Sigma(S)$ is $|\text{pos}(t)|$; i.e., the number of its positions. Its *height* $\text{ht}(t)$ is $\max\{|w| \mid w \in \text{pos}(t)\}$; i.e., the maximal length of its positions. Let $t, u \in T_\Sigma(S)$ and $w \in \text{pos}(t)$. The *label* of t at w is $t(w)$, the *subtree* of t rooted at w is $t|_w$, and the tree that is obtained from t by replacing the subtree $t|_w$ at w by u is denoted by $t[w \leftarrow u]$. Formally, $s(\varepsilon) = s|_\varepsilon = s$ and $s[\varepsilon \leftarrow u] = u$ for every $s \in S$, and for all $k \in \mathbb{N}$, $\sigma \in \Sigma_k$, and $t_1, \dots, t_k \in T_\Sigma(S)$ we have

(i) if $w = \varepsilon$, then

$$\begin{aligned} (\sigma(t_1, \dots, t_k))(w) &= \sigma \\ (\sigma(t_1, \dots, t_k))|_w &= \sigma(t_1, \dots, t_k) \\ (\sigma(t_1, \dots, t_k))[w \leftarrow u] &= u \end{aligned}$$

(ii) if $w = iv$ with $1 \leq i \leq k$ and $v \in \text{pos}(t_i)$, then

$$\begin{aligned} (\sigma(t_1, \dots, t_k))(w) &= t_i(v) \\ (\sigma(t_1, \dots, t_k))|_w &= t_i|_v \\ (\sigma(t_1, \dots, t_k))[w \leftarrow u] &= \sigma(t_1, \dots, t_{i-1}, t_i[v \leftarrow u], t_{i+1}, \dots, t_k) . \end{aligned}$$

For $1 \leq i \leq \text{rk}(t(w))$, the tree $t|_{w_i}$ is the i -th *direct subtree below* w in t . For every subset $\Delta \subseteq \Sigma \cup S$, we let $\text{pos}_\Delta(t) = \{w \in \text{pos}(t) \mid t(w) \in \Delta\}$. A tree $t \in T_\Sigma(S)$ is *linear* (resp. *nondeleting*) in a subset $Q \subseteq S$ of states if $|\text{pos}_q(t)| \leq 1$ (resp. $|\text{pos}_q(t)| \geq 1$) for every $q \in Q$. Moreover,

$$\text{states}(t) = \{s \in S \mid \text{pos}_s(t) \neq \emptyset\}$$

is the set of states that occur in t . For every selection $W \subseteq \text{pos}_S(t)$ of leaves and mapping $\theta: W \rightarrow \mathcal{P}(T_\Sigma(S))$ assigning a tree language to each selected leaf, we define the tree language

$$\begin{aligned} &t[w \leftarrow \theta(w) \mid w \in W] \\ &= \{t[w_1 \leftarrow u_1] \cdots [w_n \leftarrow u_n] \mid u_1 \in \theta(w_1), \dots, u_n \in \theta(w_n)\} \subseteq T_\Sigma(S) , \end{aligned}$$

where $W = \{w_1, \dots, w_n\}$. Similarly, given a selection $Q \subseteq S$ of states and a mapping $\theta: Q \rightarrow \mathcal{P}(T_\Sigma(S))$ assigning a tree language to each selected state, we define the tree language

$$t[q \leftarrow \theta(q) \mid q \in Q] = t[w \leftarrow \theta'(w) \mid w \in \text{pos}_Q(t)] ,$$

where $\theta': \text{pos}_Q(t) \rightarrow \mathcal{P}(T_\Sigma(S))$ is given by $\theta'(w) = \theta(t(w))$ for all $w \in \text{pos}_Q(t)$. The latter operation is also called *OI-substitution* [10] of θ in t . To simplify the notation, we fix the set $X = \{x_1, x_2, x_3, \dots\}$ of variables, which we assume to be disjoint with all ranked alphabets considered in the paper. For every $k \in \mathbb{N}$, we let $X_k = \{x_i \mid 1 \leq i \leq k\}$. Given $t \in T_\Sigma(X)$ and $\theta: X_k \rightarrow T_\Sigma(X)$, we simply write $t[\theta(x_1), \dots, \theta(x_k)]$ for $t[x \leftarrow \theta(x) \mid x \in X_k]$.

A *tree homomorphism* from Σ to Δ is a mapping $\varphi: \Sigma \rightarrow T_\Delta(X)$ such that $\varphi(\sigma) \in T_\Delta(X_k)$ for every $k \in \mathbb{N}$ and $\sigma \in \Sigma_k$. It is

- *linear* (resp. *nondeleting*) if for every $k \in \mathbb{N}$ and $\sigma \in \Sigma_k$ the tree $\varphi(\sigma)$ is linear (resp. nondeleting) in X_k , and
- *strict* (resp. *delabeling*) if $\varphi(\sigma) \notin X$ (resp. $\varphi(\sigma) \in X \cup \Delta(X)$) for every $\sigma \in \Sigma$.

We abbreviate the above restrictions by ‘l’, ‘n’, ‘s’, and ‘d’, respectively. The tree homomorphism φ induces a mapping $\varphi^*: T_\Sigma \rightarrow T_\Delta$ defined inductively by $\varphi^*(\sigma(t_1, \dots, t_k)) = \varphi(\sigma)[\varphi^*(t_1), \dots, \varphi^*(t_k)]$ for all $k \in \mathbb{N}$, $\sigma \in \Sigma_k$, and $t_1, \dots, t_k \in T_\Sigma$. As usual, we will from now on denote the induced mapping φ^* by φ , and we will also call it a tree homomorphism. We denote by \mathbf{H} the class of all tree homomorphisms, and for any combination w of ‘l’, ‘n’, ‘s’, and ‘d’ we denote by $w\text{-H}$ the class of all tree homomorphisms of type w . For instance, snl-H is the class of all strict, nondeleting and linear tree homomorphisms.

In the following, we need the class of *regular tree languages* [15,16] and some basic properties of that class. The set $\text{Reg}(\Sigma)$ contains all regular tree languages $T \subseteq T_\Sigma$ over the ranked alphabet Σ . A well-known folklore result states that $t[s \leftarrow \theta(s) \mid s \in S] \in \text{Reg}(\Sigma)$ for every finite S , tree $t \in T_\Sigma(S)$, and $\theta: S \rightarrow \text{Reg}(\Sigma)$.

A *bimorphism* is a triple $B = (\psi, T, \varphi)$ consisting of a regular tree language $T \in \text{Reg}(T)$, an input tree homomorphism $\psi: T_T \rightarrow T_\Sigma$, and an output tree homomorphism $\varphi: T_T \rightarrow T_\Delta$. The tree transformation $\tau(B) \subseteq T_\Sigma \times T_\Delta$ computed by the bimorphism B is the relation $\tau(B) = \{(\psi(t), \varphi(t)) \mid t \in T\}$, which will also be called a bimorphism. Given two combinations v and w of restrictions for tree homomorphisms, we let $\mathcal{B}(v, w)$ denote the class of all tree transformations computed by bimorphisms $B = (\psi, T, \varphi)$ such that ψ and φ are tree homomorphisms of type v and w , respectively.

3 Linear extended top-down tree transducers

Our main model is the linear extended top-down tree transducer [1,2,19,18] with regular look-ahead (l-xt^R), which is based on the classical linear top-down tree transducer without [23,24] and with regular look-ahead [7]. We will present it as a synchronous grammar [4] because we will use an auxiliary structure, called the links, in later proofs. In synchronous grammars, occurrences of equal states in the left- and right-hand side of a rule (representing the input and output side, respectively) are (implicitly) linked and these links are made explicit in a derivation. Each derivation step replaces such a pair of linked state occurrences (at the same time) by the left- and right-hand side of a rule for that state. In a rule of an l-xt^R, the (implicit) links form an injection from the state occurrences in the right-hand side to the state occurrences in the left-hand side. Thus, some states might exclusively occur in the left-hand side. Such states can be used to implement regular look-ahead, which restricts the subtrees that are acceptable at these occurrences. It should be clear (see [17, Theorem 4.4]) that there is no need to have regular look-ahead for the other states in the left-hand side, as that can be incorporated into the (nondeterministic) state behavior of the transducer.

Definition 1 [17, Section 2.2] *A linear extended top-down tree transducer with regular look-ahead (l-xt^R) is a tuple $M = (Q, \Sigma, \Delta, Q_0, R, c)$, where*

- Q is a finite set of *states* and $Q_0 \subseteq Q$ is a set of *initial states*,

- Σ and Δ are ranked alphabets of *input* and *output symbols* that are both disjoint with Q ,
- $R \subseteq T_\Sigma(Q) \times Q \times T_\Delta(Q)$ is a finite set of *rules* such that for every $(\ell, q, r) \in R$
 - $\text{states}(r) \subseteq \text{states}(\ell)$, i.e., all states that occur in r must occur in ℓ , and
 - ℓ and r are linear in $\text{states}(r)$,
- $c: Q^{\text{la}} \rightarrow \text{Reg}(\Sigma)$ is a mapping that assigns *regular look-ahead* to each (potentially) deleted state, where $Q^{\text{la}} = \bigcup_{(\ell, q, r) \in R} (\text{states}(\ell) \setminus \text{states}(r))$. Formally, the set Q^{la} depends on R (or M), but we prefer the simpler notation and hope that it does not lead to confusion. \square

For a rule $(\ell, q, r) \in R$ we say that ℓ and r are its left- and right-hand side. In contrast to other definitions [17, 13], we do not allow the same state to occur several times in the right-hand side. However, with the help of a simple renaming, each traditional linear extended top-down tree transducer can be written in our slightly more restrictive format. Next, we recall some important syntactic properties of our model. To this end, let $M = (Q, \Sigma, \Delta, Q_0, R, c)$ be an l-xt^R in the following. It is

- a *linear extended top-down tree transducer* (without look-ahead) [l-xt], if $c(q) = T_\Sigma$ for every $q \in Q^{\text{la}}$,
- a *linear top-down tree transducer with regular look ahead* [l-t^R] if $\ell \in \Sigma(Q)$ for every $(\ell, q, r) \in R$,
- a *linear top-down tree transducer* (without look ahead) [l-t] if it is both an l-xt and an l-t^R,
- ε -free (resp. *strict*) if $\ell \notin Q$ (resp. $r \notin Q$) for every $(\ell, q, r) \in R$,
- *delabeling* if $\ell \in \Sigma(Q)$ and $r \in Q \cup \Delta(Q)$ for every $(\ell, q, r) \in R$,
- *nondeleting* if $\text{states}(r) = \text{states}(\ell)$ for every $(\ell, q, r) \in R$ (i.e., $Q^{\text{la}} = \emptyset$), and
- a *finite-state relabeling* [qr] if every rule of R is of the form

$$(\sigma(q_1, \dots, q_k), q, \delta(q_1, \dots, q_k))$$

with $k \in \mathbb{N}$, $\sigma \in \Sigma_k$, $\delta \in \Delta_k$, and $q, q_1, \dots, q_k \in Q$.

Since the look-ahead component c is trivial for all l-xt, we simply omit it from their representation. We note that every nondeleting l-xt^R is an l-xt. Moreover, all l-t^R are automatically ε -free. Note also that every qr [finite-state relabeling] is a strict nondeleting delabeling l-t. For clearness' sake, we sometimes write rules as $\ell \xrightarrow{q} r$ instead of (ℓ, q, r) and, to simplify the notation in examples and illustrations, we write $\ell \xrightarrow{q_1, \dots, q_k} r$ as a shorthand for the k rules $\ell \xrightarrow{q_1} r, \dots, \ell \xrightarrow{q_k} r$. Note that for every $(\ell \xrightarrow{q} r) \in R$ the trees ℓ and r are linear in $\text{states}(r)$. Hence for every state $p \in \text{states}(r)$ the sets $\text{pos}_p(\ell)$ and $\text{pos}_p(r)$ are singletons that we identify with their unique element.

Example 2 Let us consider the l-xt^R $M_1 = (Q, \Sigma, \Delta, Q_0, R, c)$ given by

- $Q = \{\star, p, q, q^{\text{la}}, \text{id}, \text{id}'\}$ and $Q_0 = \{\star\}$,
- $\Sigma = \{\sigma^{(2)}, \gamma_1^{(1)}, \gamma_2^{(1)}\} \cup \Delta$ and $\Delta = \{\sigma_1^{(2)}, \sigma_2^{(2)}, \gamma^{(1)}, \alpha^{(0)}\}$,

– R consists of the following rules

$$\begin{array}{lll} \sigma_1(p, q) \xrightarrow{*p} \sigma_1(p, q) & \sigma_2(\text{id}, \text{id}') \xrightarrow{p; q} \sigma_2(\text{id}, \text{id}') & \gamma_1(p) \xrightarrow{p} p \\ \sigma(q, q^{\text{la}}) \xrightarrow{q} q & \sigma(q^{\text{la}}, q) \xrightarrow{q} q & \gamma_2(q) \xrightarrow{q} q \\ \gamma(\text{id}) \xrightarrow{\text{id}, \text{id}'} \gamma(\text{id}) & \alpha \xrightarrow{\text{id}, \text{id}'} \alpha & \end{array}$$

– $c: Q^{\text{la}} \rightarrow \text{Reg}(\Sigma)$ is given by $c(q^{\text{la}}) = T_{\{\gamma_2, \alpha\}}$ because $Q^{\text{la}} = \{q^{\text{la}}\}$.

Obviously, $c(q^{\text{la}})$ is a regular tree language. Additionally, we note that the state id' is essentially just a renaming of the state id (and both realize the identity on $T_{\{\gamma, \alpha\}}$). The l-xt^R M_1 is an ε -free, delabeling, linear top-down tree transducer with regular look-ahead. It is not strict and not nondeleting. \square

Next, we recall the semantics of the l-xt^R $M = (Q, \Sigma, \Delta, Q_0, R, c)$, which is (mostly) given by synchronous substitution. Formally, a *link* is just an element $(v, w) \in \mathbb{N}^* \times \mathbb{N}^*$. While the links in a rule are implicit and established due to occurrences of equal states, we need an explicit representation of the links in the sentential forms computed by M . These links together with the trees into which they point will form a dependency that is used in proofs later on. Our derivation relation is thus defined over structures consisting of an input tree, an output tree, and a set of links relating positions of those trees. Let us formalize this notion, which we call form.

Definition 3 [12, Section 3] A *form* (over Q, Σ , and Δ) is a triple $\langle \xi, L, \zeta \rangle$ consisting of an input tree $\xi \in T_\Sigma(Q)$, an output tree $\zeta \in T_\Delta(Q)$, and a set $L \subseteq \text{pos}(\xi) \times \text{pos}(\zeta)$ of links relating positions in the two trees. \square

Next, we formalize the links in a rule $\rho \in R$. These links are added to the links of a form whenever the rule ρ is applied in the derivation process. Since these links are relative to the positions at which the rule is applied, two parameters $v, w \in \mathbb{N}^*$ indicate those two positions.

Definition 4 Let $(\ell \xrightarrow{q} r) \in R$ and $v, w \in \mathbb{N}^*$. The set of *links* of $\ell \xrightarrow{q} r$ for the positions v and w is

$$\text{links}_{v, w}(\ell \xrightarrow{q} r) = \{(v, \text{pos}_p(\ell)), (w, \text{pos}_p(r)) \mid p \in \text{states}(r)\} . \quad \square$$

Example 5 Let us compute two such sets of links. Whenever it is clear that the relevant positions are in $\{1, \dots, 9\}^*$, we write positions without separating dots; e.g., 211 stands for the position 2.1.1 of length 3.

$$\begin{aligned} \text{links}_{1, 21}(\sigma_1(p, q) \xrightarrow{*} \sigma_1(p, q)) &= \{(11, 211), (12, 212)\} \\ \text{links}_{1, 21}(\sigma(q^{\text{la}}, q) \xrightarrow{q} q) &= \{(12, 21)\} \end{aligned}$$

We use grayed splines to indicate links in illustrations. The rules ρ_1 and ρ_2 above and their links, which are those of $\text{links}_{\varepsilon, \varepsilon}(\rho_1) = \{(1, 1), (2, 2)\}$ and $\text{links}_{\varepsilon, \varepsilon}(\rho_2) = \{(2, \varepsilon)\}$, are displayed in Fig. 1. \square



Fig. 1 Illustration of two rules with their implicit links.

The derivation process is started with a simple form $\langle q_0, \{(\varepsilon, \varepsilon)\}, q_0 \rangle$ consisting of an initial state $q_0 \in Q_0$ as input and output tree and the trivial link relating both occurrences of q_0 (i.e., the roots of the trees). The current form can evolve in two ways. Either (i) we apply a rule $(\ell, q, r) \in R$ to a pair (v, w) of linked occurrences of the state q or (ii) we apply the look-ahead. In the former case, such a rule application replaces the linked occurrences of q in the input and output tree by the left- and right-hand side of the rule to obtain the new input and output trees, respectively. The links of the derived form are obtained by adding the links of the rule (ℓ, q, r) for v and w to the current links. Since we are interested in the links used during the derivation, we preserve all links [in particular also the link (v, w) just used] and never remove a link. In the latter case, in which we want to apply the look-ahead, we require an occurrence of a state q at position v of the input tree that does not take part in any link with an occurrence of q in the output tree. It turns out that such a state q must be in Q^{la} , and we can replace that occurrence of q by any tree of the regular look-ahead tree language $c(q)$. Note that such replacements are independent, so different occurrences of q can be replaced by different look-ahead trees of $c(q)$. We can (potentially) continue these replacements until the form is an element of $T_\Sigma \times \mathcal{P}(\mathbb{N}^* \times \mathbb{N}^*) \times T_\Delta$.

Definition 6 [12, Section 3] Given two forms $\langle \xi, L, \zeta \rangle$ and $\langle \xi', L', \zeta' \rangle$ over Q , Σ , and Δ , we write $\langle \xi, L, \zeta \rangle \Rightarrow_M \langle \xi', L', \zeta' \rangle$ if one of the following two conditions holds:

- there exist a rule $(\ell \xrightarrow{q} r) \in R$ and a link $(v, w) \in L \cap (\text{pos}_q(\xi) \times \text{pos}_q(\zeta))$ such that

$$\xi' = \xi[v \leftarrow \ell] \quad \zeta' = \zeta[w \leftarrow r] \quad \text{and} \quad L' = L \cup \text{links}_{v,w}(\ell \xrightarrow{q} r) ,$$

- there exist a state $q \in Q^{\text{la}}$, a position $v \in \text{pos}_q(\xi)$ with $w \notin \text{pos}_q(\zeta)$ for all links $(v, w) \in L$, and a tree $t \in c(q)$ such that

$$\xi' = \xi[v \leftarrow t] \quad \zeta' = \zeta \quad \text{and} \quad L' = L .$$

A form $\langle \xi, L, \zeta \rangle$ is a *sentential form* (of M) if $\langle q_0, \{(\varepsilon, \varepsilon)\}, q_0 \rangle \Rightarrow_M^* \langle \xi, L, \zeta \rangle$ holds for some $q_0 \in Q_0$. The set of all sentential forms is denoted by $\mathcal{SF}(M)$. \square

A few derivation steps using the l-xt^R M_1 of Example 2 are illustrated in Fig. 2. Next, we define the tree transformation computed by an l-xt^R.

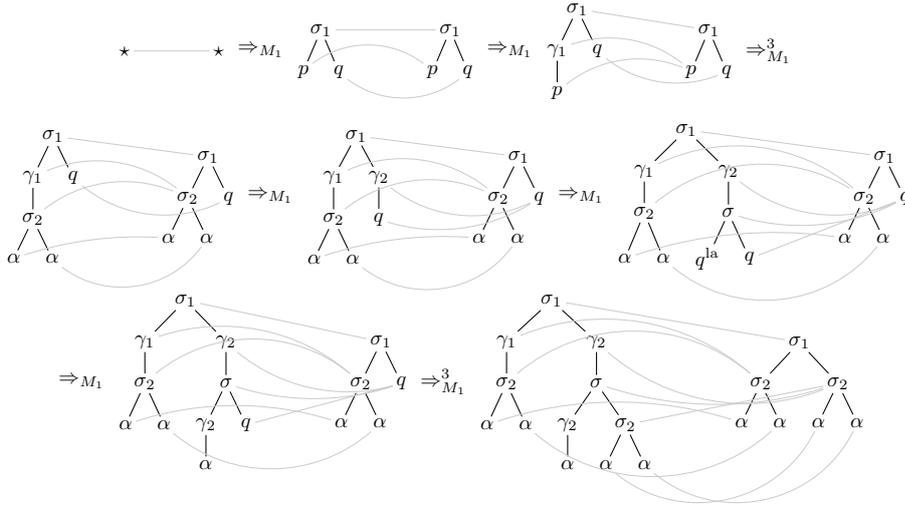


Fig. 2 Derivation using the l-xt^R M_1 of Example 2.

Definition 7 The l-xt^R M computes the set $\mathcal{D}(M)$ of *dependencies*, which are the sentential forms with state-free input and output trees. Hence

$$\mathcal{D}(M) = \{ \langle t, L, u \rangle \in \mathcal{SF}(M) \mid t \in T_\Sigma, u \in T_\Delta \} .$$

Moreover, it computes the *tree transformation* $\tau(M)$, which is given by

$$\tau(M) = \{ (t, u) \mid \langle t, L, u \rangle \in \mathcal{D}(M) \text{ for some } L \subseteq \mathbb{N}^* \times \mathbb{N}^* \} .$$

Two l-xt^R M_1 and M_2 are *equivalent* if $\tau(M_1) = \tau(M_2)$. \square

Example 8 Let M_1 be the l-xt^R of Example 2. Then

$$\langle \sigma_1(\gamma_1(\sigma_2(\alpha, \alpha)), \gamma_2(\sigma(\gamma_2(\alpha), \sigma_2(\alpha, \alpha)))), L, \sigma_1(\sigma_2(\alpha, \alpha), \sigma_2(\alpha, \alpha)) \rangle \in \mathcal{D}(M_1)$$

where

$$L = \{ (\varepsilon, \varepsilon), (1, 1), (11, 1), (111, 11), (112, 12) \} \cup \{ (2, 2), (21, 2), (212, 2), (2121, 21), (2122, 22) \} ,$$

which corresponds to the final sentential form of the derivation displayed in Fig. 2. To describe the tree transformation computed by M_1 in general, we first need some terminology. A tree $t \in T_\Sigma$ is “special” if there exist a tree $c \in T_{\{\sigma, \gamma_2, \alpha\}}(X_1)$ and two trees $t_1, t_2 \in T_{\{\gamma, \alpha\}}$ such that (i) $t = c[\sigma_2(t_1, t_2)]$, (ii) c is linear and nondeleting in X_1 , and (iii) for all $w \in \text{pos}(c)$ we have $c(w) = \sigma$ only if $w \prec \text{pos}_{x_1}(c)$. For such a special tree, the subtree $\sigma_2(t_1, t_2)$ is the “anchor” of t . Furthermore, the “left spine” of a tree $t \in T_\Sigma$ is the set $\text{pos}(t) \cap \{1\}^*$ of positions. For every $i \in \{1, 2\}$ and position v on the left spine, if $t(v) = \sigma_i$, then the subtree $t|_{v_2}$ is a “ σ_i -rib” of t .

The domain of $\tau(M_1)$ consists of all trees $t \in T_\Sigma$ such that (i) the sequence of labels of (the positions on) the left spine of t (from root to leaf) is in $\sigma_1\{\sigma_1, \gamma_1\}^*\sigma_2\gamma^*\alpha$, (ii) each σ_1 -rib of t is special, and (iii) the unique σ_2 -rib of t is in $T_{\{\gamma, \alpha\}}$. Such a tree t is only related to u in the transformation $\tau(M_1)$, where u is obtained from t by (i) removing all γ_1 -symbols on the left spine and (ii) replacing each σ_1 -rib by its anchor. Consequently, $\tau(M_1)$ is actually a partial function. \square

Since every pair $(t, u) \in \tau(M)$ is ultimately created by (at least) one successful derivation, leading to a dependency $\langle t, L, u \rangle$, we can inspect the links in L , which associate subtrees of t with subtrees of u . Roughly speaking, the links establish which parts of the output tree u were generated due to a particular part of the input tree t . Variants of this correspondence are called *contribution* in [9] and *origin* in [20]. Occasionally, we are not interested in the links. In those cases we also write $q \Rightarrow_M^* (t, u)$ provided that $\langle q, \{(\varepsilon, \varepsilon)\}, q \rangle \Rightarrow_M^* \langle t, L, u \rangle$ for some $L \subseteq \mathbb{N}^* \times \mathbb{N}^*$. The next, basic lemma expresses the fact that the replacements in the derivations of an l-xt^R are context-free.

Lemma 9 (context-freeness) For every state $q \in Q$, input tree $t \in T_\Sigma$ and output tree $u \in T_\Delta$, we have $q \Rightarrow_M^* (t, u)$ if and only if there exists a rule $(\ell, q, r) \in R$ with $\text{pos}(\ell) \subseteq \text{pos}(t)$ and $\text{pos}(r) \subseteq \text{pos}(u)$ such that

- $t(v) = \ell(v)$ for all $v \in \text{pos}_\Sigma(\ell)$ and $u(w) = r(w)$ for all $w \in \text{pos}_\Delta(r)$,
- $\ell(v) \Rightarrow_M^* (t|_v, u|_w)$ for every $(v, w) \in \text{links}_{\varepsilon, \varepsilon}(\ell \xrightarrow{q} r)$, and
- $t|_v \in c(\ell(v))$ for all $v \in \text{pos}(\ell)$ with $\ell(v) \in \text{states}(\ell) \setminus \text{states}(r)$. \square

This lemma can be used in proofs by induction on the length of a derivation because the derivations $\ell(v) \Rightarrow_M^* (t|_v, u|_w)$ are shorter than the derivation $q \Rightarrow_M^* (t, u)$.

Notation 10 To allow concise statements, we introduce the following short-hands, which mirror those already defined for tree homomorphisms:

$$\not\prec = \varepsilon\text{-free} \quad \text{s} = \text{strict} \quad \text{n} = \text{nondeleting} \quad \text{d} = \text{delabeling} .$$

We use these abbreviations in conjunction with l-xt^R to restrict to transducers with the indicated properties. For example, snl-xt stands for “strict and nondeleting linear extended top-down tree transducer” (without look-ahead). We use the same abbreviations with the stem (i.e., the material behind the hyphen) in capital letters for the corresponding classes of computed tree transformations. For instance, snl-XT stands for the class of all tree transformations computable by snl-xt, and QR denotes the class of all tree transformations computable by qr. We already remarked that every nondeleting l-xt^R is an l-xt, so we have nl-XT^R = nl-XT and similarly for the non-extended case and for all defined subclasses. To write such statements concisely, we also use sets of restrictions containing ‘ $\not\prec$ ’, ‘s’, ‘n’, and ‘d’ in front of the (potentially already restricted) stems. For instance, for every $y \subseteq \{\not\prec, \text{s}, \text{n}, \text{d}\}$, we denote by $y\text{l-XT}^{\text{R}}$ the class of all tree transformations computed by l-xt^R that obey all restrictions in y . In particular, $\emptyset\text{l-XT}^{\text{R}} = \text{l-XT}^{\text{R}}$. In this manner we can simply state $y\text{nl-XT}^{\text{R}} = y\text{nl-XT}$ for all $y \subseteq \{\not\prec, \text{s}, \text{d}\}$. \square

We observe that $yl\text{-H} \subseteq yl\text{-T}$ for every $y \subseteq \{s, n, d\}$; i.e., every linear tree homomorphism is a linear top-down tree transformation (with the same properties: ‘s’, ‘n’, ‘d’). In fact, if $\varphi: T_\Sigma \rightarrow T_\Delta$ is a linear tree homomorphism, then an equivalent l-t M_φ has the set $Q = X_m \cup \{\star\}$ of states, where m is the maximal rank of an element of Σ , the initial state \star , and all rules $(\sigma(x_1, \dots, x_k), q, \varphi(\sigma))$ with $\sigma \in \Sigma_k$, $k \in \mathbb{N}$, and $q \in Q$. It should be clear that $\tau(M_\varphi) = \varphi$.

Next, we recall some results that relate l-xt^R to bimorphisms. In [2] the class $\mathcal{B}(\text{snl}, \text{snl})$ is denoted by BI, and in [21] the class $\mathcal{B}(\text{snl}, \text{nl})$ is denoted by B(LCE, LC).

Proposition 11 ([2] and [21, Theorems 17 and 4])

$$\not\exists \text{nl-X}T = \mathcal{B}(\text{snl}, \text{snl}) \quad \not\exists \text{nl-X}T = \mathcal{B}(\text{snl}, \text{nl}) \quad \text{l-X}T^R = \mathcal{B}(\text{nl}, \text{l}) \quad \square$$

Thus, every tree transformation in l-XT^R is the composition of an inverse tree homomorphism, the identity on a regular tree language, and a tree homomorphism. We will need a similar, but simpler result that tells us how to emulate an l-xt^R by the composition of an inverse homomorphism and an l-t^R.

Proposition 12 [13, Lemma 4.1 and Corollary 4.1] For every $y \subseteq \{s, n\}$

$$y \not\exists \text{l-X}T^R = \text{snl-H}^{-1}; y \text{l-T}^R \quad \text{and} \quad y \text{l-X}T^R = \text{nl-H}^{-1}; y \text{l-T}^R .$$

PROOF. We prove both inclusions starting with (\supseteq). Let φ be a nondeleting and linear tree homomorphism from Γ to Σ , and let $M = (Q, \Gamma, \Delta, Q_0, R, c)$ be an l-t^R. We construct the l-xt^R $M' = (Q, \Sigma, \Delta, Q_0, R', c')$ such that for each rule $(\gamma(q_1, \dots, q_k), q, r)$ in R (with $k \in \mathbb{N}$, $\gamma \in \Gamma_k$, and $q_1, \dots, q_k \in Q$) the rule $(\varphi(\gamma)[q_1, \dots, q_k], q, r)$ is in R' . No further rules are in R' . Note that since φ is nondeleting, we have $\text{states}(\varphi(\gamma)[q_1, \dots, q_k]) = \{q_1, \dots, q_k\}$ and thus Q^{la} is the same for M' and M . For every $q \in Q^{\text{la}}$ we set $c'(q) = \varphi(c(q))$, which is regular because, as is well known, the class of regular tree languages is closed under linear tree homomorphisms. Using Lemma 9, it is straightforward to show that $q \Rightarrow_{M'}^* (t, u)$ if and only if there exists $s \in T_\Gamma$ with $t = \varphi(s)$ and $q \Rightarrow_M^* (s, u)$. Thus $\tau(M') = \{(\varphi(s), u) \mid (s, u) \in \tau(M)\} = \varphi^{-1}; \tau(M)$.

For the remaining inclusion (\subseteq), let $M = (Q, \Sigma, \Delta, Q_0, R, c)$ be an l-xt^R. We turn R into a ranked alphabet such that $\text{rk}(\ell \xrightarrow{q} r) = |\text{pos}_Q(\ell)|$ for every $(\ell \xrightarrow{q} r) \in R$. Using this ranked alphabet R we now construct the l-t^R $M' = (Q, R \cup \Sigma, \Delta, Q_0, R', c)$ and the nondeleting and linear tree homomorphism φ from $R \cup \Sigma$ to Σ as follows. For every $k \in \mathbb{N}$ and rule $\rho = (\ell, q, r)$ in R_k with $\text{pos}_Q(\ell) = \{v_1, \dots, v_k\}$ the rule $\rho(\ell(v_1), \dots, \ell(v_k)) \xrightarrow{q} r$ is in R' and $\varphi(\rho) = \ell[v_i \leftarrow x_i \mid 1 \leq i \leq k]$. No further rules are in R' . Additionally, $\varphi(\sigma) = \sigma(x_1, \dots, x_k)$ for every $k \in \mathbb{N}$ and $\sigma \in \Sigma_k$, which yields that $\varphi(t) = t$ for every $t \in T_\Sigma$. This latter part is needed for the look-ahead c . Clearly, if we apply the construction in the above proof of the first inclusion to φ and M' , then we reobtain M . Hence $\varphi^{-1}; \tau(M') = \tau(M)$. \square

We use this proposition to establish our first composition result, which extends the classical composition result of [7] for linear top-down tree transducers with regular look-ahead. The only difference is that our first transducer has extended left-hand sides (i.e., it is an $l\text{-xt}^R$ instead of just an $l\text{-t}^R$).

Lemma 13 (composition on the right) For every $y \subseteq \{\not\prec, s, n\}$

$$yl\text{-XT}^R ; yl\text{-T}^R \subseteq yl\text{-XT}^R .$$

PROOF. Immediate from Proposition 12 and the composition closure result $(l\text{-T}^R)^2 \subseteq l\text{-T}^R$ for linear top-down tree transducers with regular look-ahead; it is straightforward to check that the proof of this result in [7, Theorem 2.11] preserves the properties ‘s’ and ‘n’. \square

We conclude this section by discussing two results on regular look-ahead. First, we recall that when deletion is allowed, regular look-ahead adds expressive power.

Proposition 14 [17, Lemma 4.3] $\not\prec sl\text{-XT}^R \not\subseteq l\text{-XT}$

PROOF. The counter-example presented in the proof of [17, Lemma 4.3], which shows $l\text{-T}^R \not\subseteq l\text{-XT}$, is in $sl\text{-T}^R$. \square

Second, we recall from [7, Theorem 2.6] that an $l\text{-t}^R$ (with look-ahead) can be decomposed into two $l\text{-t}$ (without look-ahead), of which the first is a finite-state relabeling. This result can easily be generalized to extended top-down tree transducers and their compositions.

Lemma 15 (look-ahead decomposition)

$$(yl\text{-XT}^R)^n \subseteq QR ; yl\text{-XT}^n \subseteq yl\text{-XT}^{n+1}$$

for every $n \geq 1$ and $y \subseteq \{\not\prec, s, n\}$.

PROOF. The second inclusion is immediate because $QR \subseteq snl\text{-T} \subseteq \not\prec snl\text{-XT}$. We prove the first inclusion by induction on n . For $n = 1$ an obvious generalization of the construction in the proof of [7, Theorem 2.6], which preserves ‘s’ and ‘n’, can be used. For $n \geq 1$, we have

$$\begin{aligned} (yl\text{-XT}^R)^n ; yl\text{-XT}^R &\subseteq (yl\text{-XT}^R)^n ; QR ; yl\text{-XT} \\ &\subseteq (yl\text{-XT}^R)^n ; yl\text{-XT} \\ &\subseteq QR ; yl\text{-XT}^{n+1} , \end{aligned}$$

where the case $n = 1$ is used in the first step, Lemma 13 in the second step, and the induction hypothesis in the last step. \square

Lemma 15 implies that

$$(yl\text{-XT}^R)^n \subseteq yl\text{-XT}^{n+1} \subseteq (yl\text{-XT}^R)^{n+1}$$

for every $n \geq 1$ and $y \subseteq \{\not\prec, s, n\}$, so the classes $yl\text{-XT}^R$ and $yl\text{-XT}$ have the same composition closure. However, this closure is potentially achieved at different powers.

4 Four classes that are closed at a finite power

In this section, we show that the four classes $\not\text{sl-XT}^{\text{R}}$, $\not\text{sl-XT}$, $\not\text{l-XT}^{\text{R}}$, and $\not\text{l-XT}$ are closed under composition at a finite power. We first recall a central result of [3], which shows that none of them is closed under composition.

Proposition 16 [3, Section 3.4] $\not\text{sl-XT}^2 \not\subseteq \text{l-XT}^{\text{R}}$ □

We note that [3] states the even stronger result that the class $\mathcal{B}(\text{snl}, \text{snl})^2$ is not contained in the class of all bimorphisms, which implies the above result by Proposition 11. In [3] the class $\mathcal{B}(\text{snl}, \text{snl})$ is denoted by $\mathcal{B}(s, c)$. The proof of Theorem 31 in Section 5 implies Proposition 16 for $\not\text{l-XT}^{\text{R}}$ instead of l-XT^{R} , which is all we need; for the implication see non-inclusion (ii) in the proof of Theorem 47.

Next we recall another central result of [3]: the (very restricted) class $\not\text{snl-XT}$ is not closed under composition (by the previous proposition), but is closed under composition at power 2.

Proposition 17 [3, Theorem 6.2] $\not\text{snl-XT} \subsetneq \not\text{snl-XT}^2 = \not\text{snl-XT}^n$ for every $n \geq 3$. □

As we will show now, the (strict) classes $\not\text{sl-XT}^{\text{R}}$ and $\not\text{sl-XT}$ are also closed under composition already at the second power. We start with a lemma that decomposes an $\not\text{sl-xt}$ into two transducers of which one is an $\not\text{snl-xt}$, for which we have the composition closure result of Proposition 17. For the benefit of Section 6, we make ε -freeness optional in the next two lemmas.

Lemma 18 (decomposition on the right) $y\text{sl-XT} \subseteq y\text{snl-XT} ; \text{sdl-H}$ for every $y \subseteq \{\not\}$.

PROOF. This is proved for strict tree homomorphisms in [5, Section I-2-1-3-5]. The proof can be generalized to $y\text{sl-xt}$, as follows. Let $M = (Q, \Sigma, \Delta, Q_0, R)$ be an sl-xt . Clearly, we may assume a separation of the states into deleted and non-deleted states. More precisely, we assume $m \geq \max\{\text{rk}(\sigma) \mid \sigma \in \Sigma\}$ such that $Q = Q_1 \cup \{1, \dots, m\}$ with $Q_1 \cap \mathbb{N} = \emptyset$ and for every rule $(\ell, q, r) \in R$ the following three conditions hold: (i) $q \in Q_1$ and $\text{states}(r) \subseteq Q_1$, (ii) ℓ is linear in Q , and (iii) $\text{states}(\ell) \setminus \text{states}(r) \subseteq \{1, \dots, m\}$. Let Δ' be the ranked alphabet $\{\delta_n \mid \delta \in \Delta, 0 \leq n \leq m\}$ with $\text{rk}(\delta_n) = \text{rk}(\delta) + n$. In addition, let $\bar{\Sigma} = \{\bar{\sigma} \mid \sigma \in \Sigma\}$ be the ranked alphabet with $\text{rk}(\bar{\sigma}) = \text{rk}(\sigma)$. We suppose that Δ , Δ' , and $\bar{\Sigma}$ are pairwise disjoint. As intermediate alphabet we take $\Gamma = \Delta \cup \Delta' \cup \bar{\Sigma}$, and let $\alpha \in \Delta_0$ be an arbitrary nullary output symbol. Now we first construct the strict delabeling tree homomorphism φ from Γ to Δ such that (i) $\varphi(\delta_n) = \varphi(\delta) = \delta(x_1, \dots, x_k)$ for every $\delta \in \Delta_k$ and $0 \leq n \leq m$ and (ii) $\varphi(\bar{\sigma}) = \alpha$ for every $\sigma \in \Sigma$. Thus, φ turns every δ_n into δ and deletes its last n arguments.

For every rule $(\ell, q, r) \in R$ there exist $k \in \mathbb{N}$, $\delta \in \Delta$, and $r_1, \dots, r_k \in T_{\Delta}(Q)$ such that $r = \delta(r_1, \dots, r_k)$ because M is strict. We construct the nondeleting sl-xt $M' = (Q, \Sigma, \Gamma, Q_0, R')$ such that R' contains the rule

$$(\ell, q, \delta_n(r_1, \dots, r_k, i_1, \dots, i_n))$$

for every rule $(\ell, q, \delta(r_1, \dots, r_k)) \in R$, where $\text{states}(\ell) \setminus \text{states}(r) = \{i_1, \dots, i_n\}$ with $i_1 < \dots < i_n$. This is a proper rule because ℓ is linear in Q . Moreover, R' contains the rules $(\sigma(1, \dots, k), i, \bar{\sigma}(1, \dots, k))$ for all $k \in \mathbb{N}$, $\sigma \in \Sigma_k$, and $i \in \{1, \dots, m\}$. The set R' contains no further rules. Thus, M' simulates M but attaches the subtrees that are deleted by M to the root of the right-hand side of each rule. It is straightforward to show that $\tau(M'); \varphi = \tau(M)$. Clearly, if M is ε -free, then so is M' . \square

The next lemma is our second composition result, which is more restricted than the first, which is Lemma 13, but sufficiently powerful in combination with Lemma 18.

Lemma 19 (composition on the left) $\text{sdl-H}; \text{ysl-XT} \subseteq \text{ysl-XT}$ for every $y \subseteq \{\emptyset\}$.

PROOF. Let $\varphi: T_\Sigma \rightarrow T_\Gamma$ be a strict delabeling linear tree homomorphism, and let $M = (Q, \Gamma, \Delta, Q_0, R)$ be a strict l-xt. Moreover, let $Q' = Q \cup \{\star\}$ for a new state $\star \notin Q$. We extend φ to a tree transformation $\varphi': T_\Sigma(Q') \rightarrow T_\Gamma(Q')$ such that $\varphi'(q') = q'$ for every $q' \in Q'$ and

$$\varphi'(\sigma(t_1, \dots, t_k)) = \varphi(\sigma)[\varphi'(t_1), \dots, \varphi'(t_k)]$$

for every $k \in \mathbb{N}$, $\sigma \in \Sigma_k$, and $t_1, \dots, t_k \in T_\Sigma(Q')$. We construct the l-xt $M' = (Q', \Sigma, \Delta, Q_0, R')$ such that for each rule $(\ell, q, r) \in R$ we have all rules (ℓ', q, r) in R' for which (i) $\ell' \in T_\Sigma(Q')$ is linear in Q , (ii) $\varphi'(\ell') = \ell$, and (iii) $|\text{pos}_\Sigma(\ell')| = |\text{pos}_\Gamma(\ell)|$. No further rules are in R' .

Let us quickly consider a small example. Suppose that R contains the rule $\gamma(\alpha, \gamma'(q)) \xrightarrow{q} \delta(q)$ and we have (i) $\varphi(\sigma) = \gamma(x_3, x_2)$, (ii) $\varphi(\sigma') = \gamma'(x_1)$, and (iii) $\varphi(\alpha) = \alpha$ with $\{\alpha^{(0)}, (\sigma')^{(2)}, \sigma^{(3)}\} \subseteq \Sigma$. Then R' contains the rule $\sigma(\star, \sigma'(q, \star), \alpha) \xrightarrow{q} \delta(q)$.

It should be clear that $\tau(M') = \varphi; \tau(M)$. Finally, we observe that M' is strict because it has the same right-hand sides of rules as M , and it is ε -free if M is ε -free because φ is strict. \square

The previous two lemmas are now used to prove that $\not\text{sl-XT}^R$ and $\not\text{sl-XT}$ are closed under composition at power 2.

Theorem 20 $(\not\text{sl-XT}^R)^n \subseteq \not\text{sl-XT}; \not\text{sl-XT} \subseteq \not\text{sl-XT}^2$ for every $n \geq 1$.

PROOF. The second inclusion is trivial because $\not\text{sl-XT} \subseteq \not\text{sl-XT}$. For the first inclusion, we first prove that $\not\text{sl-XT}^n \subseteq \not\text{sl-XT}^n; \text{sdl-H}$. The idea of this inclusion is that the first $\not\text{sl-xt}$ splits off a tree homomorphism of type ‘sdl’ on the right (using Lemma 18), which is then absorbed on the left by the second $\not\text{sl-xt}$ (using Lemma 19). This auxiliary statement is proved by induction. For $n = 1$, we have to prove $\not\text{sl-XT} \subseteq \not\text{sl-XT}; \text{sdl-H}$, which is the statement of Lemma 18. For $n \geq 1$ we obtain that

$$\not\text{sl-XT}^{n+1} = \not\text{sl-XT}; \not\text{sl-XT}^n \subseteq \not\text{sl-XT}; \text{sdl-H}; \not\text{sl-XT}^n$$

$$\subseteq \not\text{sl-XT} ; \not\text{sl-XT}^n \subseteq \not\text{sl-XT}^{n+1} ; \text{sdl-H}$$

where we used Lemma 18 in the second step, Lemma 19 in the third step, and the induction hypothesis in the last step. From Lemma 15 and the above inclusion we now conclude that

$$(\not\text{sl-XT}^{\text{R}})^n \subseteq \text{QR} ; \not\text{sl-XT}^n \subseteq \text{QR} ; \not\text{sl-XT}^n ; \text{sdl-H} .$$

Since $\text{QR} \subseteq \not\text{sl-XT}$, this implies that

$$(\not\text{sl-XT}^{\text{R}})^n \subseteq \not\text{sl-XT}^{n+1} ; \text{sdl-H} \subseteq \not\text{sl-XT}^2 ; \text{sdl-H} ,$$

where the second inclusion is due to Proposition 17. Since $\text{sdl-H} \subseteq \text{sl-T}$ we can apply Lemma 13 to obtain that

$$\not\text{sl-XT}^2 ; \text{sdl-H} \subseteq \not\text{sl-XT}^2 ; \text{sl-T} \subseteq \not\text{sl-XT} ; \not\text{sl-XT}^{\text{R}} .$$

Applying Lemmas 15 and 13 once more, we obtain

$$\not\text{sl-XT} ; \not\text{sl-XT}^{\text{R}} \subseteq \not\text{sl-XT} ; \text{QR} ; \not\text{sl-XT} \subseteq \not\text{sl-XT}^{\text{R}} ; \not\text{sl-XT} .$$

Since $\not\text{sl-XT}^{\text{R}} = \not\text{sl-XT}$ we have proved the statement. \square

Up to now, we have shown that the (strict) classes $\not\text{sl-XT}^{\text{R}}$ and $\not\text{sl-XT}$ are closed under composition at the second power. In the rest of this section, we will show that the classes $\not\text{sl-XT}^{\text{R}}$ and $\not\text{sl-XT}$ are closed under composition at the third and fourth power, respectively. We start with a normal form for $\not\text{sl-xt}^{\text{R}}$, in which every rule that violates the strictness condition is simulated by a chain of rules for a (non-extended) l-t^{R} .

Lemma 21 (non-strict normal form) For every $\not\text{sl-xt}^{\text{R}} (Q, \Sigma, \Delta, Q_0, R, c)$ there exists an equivalent $\not\text{sl-xt}^{\text{R}} (Q', \Sigma, \Delta, Q_0, R', c')$ such that $\ell \in \Sigma(Q')$ for every rule $(\ell, q, r) \in R'$ with $r \in Q'$.

PROOF. Let $M = (Q, \Sigma, \Delta, Q_0, R, c)$ and $M' = (Q', \Sigma, \Delta, Q_0, R', c')$. Every (non-strict) rule $\rho = (\ell, q, r)$ in R with $r \in Q$ can be simulated by a finite set R'_ρ of l-t^{R} rules as follows. We consider new states of the form $\langle \rho, v \rangle$ where $v \in \text{pos}(\ell) \setminus \{\varepsilon, \text{pos}_r(\ell)\}$. Moreover, we let $\langle \rho, \varepsilon \rangle = q$ and $\langle \rho, \text{pos}_r(\ell) \rangle = r$. For every position $v \in \text{pos}(\ell)$ such that $v \prec \text{pos}_r(\ell)$, we have the following rule in R'_ρ :

$$\ell(v)(\langle \rho, v1 \rangle, \dots, \langle \rho, vk \rangle) \xrightarrow{\langle \rho, v \rangle} \langle \rho, vi \rangle ,$$

where $k = \text{rk}(\ell(v))$ and $i \in \mathbb{N}$ is the unique integer such that $vi \preceq \text{pos}_r(\ell)$. The look-ahead for every new state $\langle \rho, vj \rangle$ with $j \neq i$ is defined by

$$c'(\langle \rho, vj \rangle) = (\ell|_{vj})[q' \leftarrow c(q') \mid q' \in \text{states}(\ell|_{vj})] .$$

We note that $\text{states}(\ell|_{vj}) \subseteq Q^{\text{la}}$. The tree language $c'(\langle \rho, vj \rangle)$ is regular by the folklore result stating that OI-substitution preserves regularity, which we mentioned at the end of Section 2. Recall that in OI-substitution, different occurrences of the same state q' can be replaced by different trees of $c(q')$.

The rules in R'_ρ simulate the rule ρ by consuming the left-hand side ℓ position by position, following the path from the root to the unique occurrence of r . Thus, we define the set Q' of states of M' to consist of Q together with all the mentioned new states. The set R' of rules consists of all strict rules in R together with the rules in R'_ρ for all non-strict rules ρ in R . The look-ahead c' equals c on the states in Q^{la} , and is defined as above for the new states. Then $\tau(M') = \tau(M)$ and M' satisfies the requirements. \square

Example 22 We illustrate the construction on the example rule

$$\rho = \sigma(\sigma(p, p), \sigma(\alpha, r)) \xrightarrow{q} r ,$$

for which $p, q, r \in Q$ and the relevant look-ahead is $c(p) = T$. Corresponding to this rule, M' has the following two rules in R'_ρ :

$$\sigma(\langle \rho, 1 \rangle, \langle \rho, 2 \rangle) \xrightarrow{q} \langle \rho, 2 \rangle \quad \text{and} \quad \sigma(\langle \rho, 21 \rangle, r) \xrightarrow{\langle \rho, 2 \rangle} r$$

because $q = \langle \rho, \varepsilon \rangle$ and $r = \langle \rho, 22 \rangle$. Moreover, we have

$$c'(\langle \rho, 1 \rangle) = \{\sigma(t_1, t_2) \mid t_1, t_2 \in T\}$$

and $c'(\langle \rho, 21 \rangle) = \{\alpha\}$ for the look-ahead c' of M' . \square

The next lemma is similar to Lemma 18, in that it demonstrates how to decompose an $\not\exists\text{-xt}^{\text{R}}$ into a delabeling l-t^{R} and an $\not\exists\text{sl-xt}^{\text{R}}$, for which we now have the composition closure result of Theorem 20. The proof is, however, more complicated than the one of Lemma 18. Since the delabeling property is not essential in the following, we actually state a weaker variant.

Lemma 23 (decomposition on the left) $\not\exists\text{-XT}^{\text{R}} \subseteq \text{l-T}^{\text{R}} ; \not\exists\text{sl-XT}^{\text{R}}$

PROOF. Let $M = (Q, \Sigma, \Delta, Q_0, R, c)$ be an ε -free l-xt^{R} such that $\ell \in \Sigma(Q)$ for every rule $(\ell, q, r) \in R$ with $r \in Q$. We can assume this normal form without loss of generality by Lemma 21. Additionally, we may assume that $|Q| \geq m$, where $m = \max\{\text{rk}(\sigma) \mid \sigma \in \Sigma\}$. We will construct an l-t^{R} M_1 and a strict $\not\exists\text{-xt}^{\text{R}}$ M_2 such that $\tau(M_1) ; \tau(M_2) = \tau(M)$. Intuitively speaking, the transducer M_1 processes the input by nondeterministically executing a number of non-strict rules of M . Whenever it executes two consecutive non-strict rules, M_1 simulates the state behavior of M . Moreover, M_1 marks the positions in the (processed) input where it has applied a sequence of consecutive non-strict rules by indicating the corresponding state transition of M . The transducer M_2 then uses these markings to execute the missing strict rules of M .

As intermediate ranked alphabet we use $\Gamma = \Sigma \cup (\Sigma \times Q \times Q)$, where each triple $\langle \sigma, q', q \rangle \in \Sigma \times Q \times Q$ has the same rank as σ . We fix m pairwise different states $p_1, \dots, p_m \in Q$. We construct the l-t^{R} $M_1 = (Q_1, \Sigma, \Gamma, \{p_1\}, R_1, c_1)$ with states $Q_1 = Q \cup (Q \times Q)$ and the set R_1 of rules consists of:

- (i) the rule $\sigma(p_1, \dots, p_k) \xrightarrow{p} \sigma(p_1, \dots, p_k)$ for every $k \in \mathbb{N}$, $\sigma \in \Sigma_k$, and $p \in Q$,

(ii) the two rules

$$\begin{aligned} \sigma(q_1, \dots, q_{i-1}, \langle q, q_i \rangle, q_{i+1}, \dots, q_k) &\xrightarrow{p} \langle q, q_i \rangle \\ \sigma(q_1, \dots, q_{i-1}, \langle q', q_i \rangle, q_{i+1}, \dots, q_k) &\xrightarrow{\langle q', q \rangle} \langle q', q_i \rangle \end{aligned}$$

for every non-strict rule $\sigma(q_1, \dots, q_k) \xrightarrow{q} q_i$ in R with $1 \leq i \leq k$ and every $p, q' \in Q$, and

(iii) the rule $\sigma(p_1, \dots, p_k) \xrightarrow{\langle q', q \rangle} \langle \sigma, q', q \rangle(p_1, \dots, p_k)$ for every $k \in \mathbb{N}$, $\sigma \in \Sigma_k$, and $q', q \in Q$.

We note that $Q_1^{\text{la}} \subseteq Q^{\text{la}}$. The look-ahead mapping $c_1: Q_1^{\text{la}} \rightarrow \text{Reg}(\Sigma)$ is given by $c_1(q) = c(q)$ for every $q \in Q_1^{\text{la}}$. Actually, M_1 is delabeling.

Next, we construct the $\not\text{sl-xt}^{\text{R}}$ $M_2 = (Q_2, \Gamma, \Delta, Q_0, R' \cup R_2, c_2)$ with the set $Q_2 = Q$ of states, the set $R' = \{(\ell, q, r) \in R \mid r \notin Q\}$ of strict rules of M , and the set

$$R_2 = \{ \langle \sigma, q', q \rangle(\ell_1, \dots, \ell_k) \xrightarrow{q'} r \mid q' \in Q, (\sigma(\ell_1, \dots, \ell_k) \xrightarrow{q} r) \in R' \} .$$

Again, $Q_2^{\text{la}} \subseteq Q^{\text{la}}$, where Q_2^{la} contains the look-ahead states of M_2 , so we just set the look-ahead mapping $c_2: Q_2^{\text{la}} \rightarrow \text{Reg}(\Gamma)$ to $c_2(q) = c(q)$ for every $q \in Q_2^{\text{la}}$.

Intuitively, it should be clear that $\tau(M_1); \tau(M_2) = \tau(M)$. Whenever M_2 arrives in state q' at an input position with label $\langle \sigma, q', q \rangle$, it knows that M_1 has applied a sequence of non-strict rules of M that led from state q' to state q , and thus M_2 can continue acting as if it is already in state q . Formally, it can be proved that, for every state $q \in Q$, input tree $t \in T_\Sigma$, and output tree $u \in T_\Delta$, we have $q \Rightarrow_M^*(t, u)$ if and only if there exists $s \in T_\Gamma$ such that $p_1 \Rightarrow_{M_1}^*(t, s)$ and $q \Rightarrow_{M_2}^*(s, u)$. The proof is by induction on the length of the derivations using Lemma 9. It uses several elementary properties of the derivations of M_1 and M_2 such as (for all $p, q, q', q'' \in Q$):

- if $p_1 \Rightarrow_{M_1}^*(t, s)$, then $p \Rightarrow_{M_1}^*(t, s)$,
- if $p_1 \Rightarrow_{M_1}^*(t, \sigma(s_1, \dots, s_k))$, then $\langle q', q \rangle \Rightarrow_{M_1}^*(t, \langle \sigma, q', q \rangle(s_1, \dots, s_k))$,
- $p_1 \Rightarrow_{M_1}^*(t, \langle \sigma, q', q \rangle(s_1, \dots, s_k))$ if and only if $\langle q'', q' \rangle \Rightarrow_{M_1}^*(t, \langle \sigma, q'', q \rangle(s_1, \dots, s_k))$, and
- $q \Rightarrow_{M_2}^*(\sigma(s_1, \dots, s_k), u)$ if and only if $q' \Rightarrow_{M_2}^*(\langle \sigma, q', q \rangle(s_1, \dots, s_k), u)$. \square

Lemmas 23 and 13 now enable us to prove that the class $\not\text{1-XT}^{\text{R}}$ is closed under composition at power 3. The proof is similar to, but easier than, the one of Theorem 20.

Theorem 24 $(\not\text{1-XT}^{\text{R}})^n \subseteq \text{l-T}^{\text{R}}; \not\text{sl-XT}^2 \subseteq (\not\text{1-XT}^{\text{R}})^3$ for every $n \geq 1$.

PROOF. Again, the second inclusion is trivial because l-T^{R} and $\not\text{sl-XT}$ are subclasses of $\not\text{1-XT}^{\text{R}}$. Similar to the proof of Theorem 20, the idea of the first inclusion is that the last $\not\text{1-xt}^{\text{R}}$ splits off an l-t^{R} on the left (using

Class	Closed at power	Stated in
$\not\leq\text{sl-XT}^{\text{R}}, \not\leq\text{sl-XT}$	2	Theorem 20
$\not\leq\text{l-XT}^{\text{R}}$	3	Theorem 24
$\not\leq\text{l-XT}$	4	Corollary 25

Table 2 Summary of the results of Section 4.

Lemma 23), which is then absorbed on the right by the penultimate $\not\leq\text{l-xt}^{\text{R}}$ (using Lemma 13). Formally we prove by induction on n that

$$(\not\leq\text{l-XT}^{\text{R}})^n \subseteq \text{l-T}^{\text{R}} ; (\not\leq\text{sl-XT}^{\text{R}})^n ,$$

which suffices by Theorem 20. For $n = 1$ we obtain $\not\leq\text{l-XT}^{\text{R}} \subseteq \text{l-T}^{\text{R}} ; \not\leq\text{sl-XT}^{\text{R}}$, which is stated in Lemma 23. In the induction step for $n \geq 1$, we obtain

$$\begin{aligned} (\not\leq\text{l-XT}^{\text{R}})^{n+1} &= (\not\leq\text{l-XT}^{\text{R}})^n ; \not\leq\text{l-XT}^{\text{R}} \subseteq (\not\leq\text{l-XT}^{\text{R}})^n ; \text{l-T}^{\text{R}} ; \not\leq\text{sl-XT}^{\text{R}} \\ &\subseteq (\not\leq\text{l-XT}^{\text{R}})^n ; \not\leq\text{sl-XT}^{\text{R}} \subseteq \text{l-T}^{\text{R}} ; (\not\leq\text{sl-XT}^{\text{R}})^{n+1} , \end{aligned}$$

where we use Lemma 23 in the second step, Lemma 13 in the third step, and the induction hypothesis in the last step. \square

It is immediate from Theorem 24 and Lemma 15 that the class $\not\leq\text{l-XT}$ is closed under composition at power 4. Thus, in contrast to Theorem 20, look-ahead influences the power of closedness in the non-strict case, as will be proved in the next section.

Corollary 25 $\not\leq\text{l-XT}^n \subseteq \not\leq\text{l-XT}^4$ for every $n \geq 1$. \square

A summary of our results concerning the powers at which the considered classes are closed under composition is provided in Table 2. In the next section, we will demonstrate that these powers are indeed the least ones with this property.

5 Least power of closedness

In this section, we will determine the least power at which the composition closure is achieved for the classes $\not\leq\text{l-XT}^{\text{R}}$, $\not\leq\text{sl-XT}^{\text{R}}$, $\not\leq\text{sl-XT}$, and $\not\leq\text{l-XT}$, which are all computed by certain ε -free l-xt^{R} . For the strict classes the least power is 2, as stated in the next theorem. In the remainder of this section we consider the non-strict classes.

Theorem 26

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

for every $n \geq 3$.

PROOF. The first inclusion is trivial and its strictness follows from Proposition 14. The second inclusion is also trivial and its strictness follows from Proposition 16, which shows that the class sl-XT^R is not closed under composition. The three equalities are proved in Theorem 20. \square

In the following, we will use the computed dependencies in $\mathcal{D}(M)$, for which we recall some important properties from [11]. Let $L \subseteq \mathbb{N}^* \times \mathbb{N}^*$ be a set of links [e.g., the set L of links in a dependency $\langle t, L, u \rangle \in \mathcal{D}(M)$]. The elements of $\text{dom}(L)$ are also called *link origins* of L . For the next definition, proposition and lemma, let $M = (Q, \Sigma, \Delta, Q_0, R, c)$ be the considered ε -free l-xt^R.

Definition 27 [11, Definitions 4 and 5] A set $L \subseteq \mathbb{N}^* \times \mathbb{N}^*$ of links is

- *strictly input hierarchical* if for all links $(v_1, w_1), (v_2, w_2) \in L$
 - $v_1 \prec v_2$ implies $w_1 \preceq w_2$ and
 - $v_1 = v_2$ implies $w_1 \preceq w_2$ or $w_2 \preceq w_1$,
- *input link-distance bounded by $b \in \mathbb{N}$* if for all link origins $v_1, v_2 \in \text{dom}(L)$ with $v_1 \prec v_2$ and $|v_2| - |v_1| > b$ there exists a link origin $v \in \text{dom}(L)$ such that $v_1 \prec v \prec v_2$ and $|v| - |v_1| \leq b$.

The set $\mathcal{D}(M)$ of dependencies has those properties if for each dependency $\langle t, L, u \rangle \in \mathcal{D}(M)$ the set L of links has them. We also say that $\mathcal{D}(M)$ is *input link-distance bounded* if there exists an integer $b \in \mathbb{N}$ such that it is input link-distance bounded by b . \square

We assume that the corresponding properties are defined for the output side, using L^{-1} instead of L . For example, L is *strictly output hierarchical* if L^{-1} is strictly input hierarchical. The set $\mathcal{D}(M)$ computed by the ε -free l-xt^R M always has these properties as shown in [11].

Proposition 28 [11, Corollary 1 and Theorem 2] The set $\mathcal{D}(M)$ of dependencies is strictly input and output hierarchical, and it is input and output link-distance bounded. \square

These properties should be intuitively clear. They are discussed in more detail in [11]. Roughly speaking, the set L of links of a sentential form of M is strictly input and output hierarchical because links cannot cross each other. In addition, if b is the maximal height of the left-hand (resp. right-hand) side of a rule of M , then L is obviously input (resp. output) link-distance bounded by b . Next, we observe some simple consequences of Proposition 28, which we will use later. Whenever we mention ‘(in)comparable’ in the following, we refer to the partial prefix order \preceq .

Lemma 29 Let $\langle t, L, u \rangle \in \mathcal{D}(M)$ be a dependency, and let $\mathcal{D}(M)$ be input link-distance bounded by b .

- (i) For all links $(v, w), (v', w') \in L$, v and v' are incomparable if and only if w and w' are incomparable.
- (ii) For all positions $v_1, v_2 \in \text{pos}(t)$ and link origins $v_0, v_3 \in \text{dom}(L)$ with $v_0 \preceq v_1 \prec v_2 \preceq v_3$ and $|v_2| - |v_1| > b$, there exists a link origin $v \in \text{dom}(L)$ such that $v_1 \prec v \prec v_2$ and $|v| - |v_1| \leq b$.



Fig. 3 Links with inverted arrows.

PROOF. We start with the if-direction in the first item. Without loss of generality, suppose that $v \preceq v'$. Then by the definition of strictly input hierarchical, we know that w and w' are comparable. The other direction is similarly true by the definition of strictly output hierarchical. We prove the second item by induction on $|v_3| - |v_0|$ as follows. Since $v_0 \preceq v_1 \prec v_2 \preceq v_3$ and $|v_3| - |v_0| \geq |v_2| - |v_1| > b$, there exists a link origin $v \in \text{dom}(L)$ such that $v_0 \prec v \prec v_3$ and $|v| - |v_0| \leq b$. Consequently, $v \prec v_2$. Now we distinguish two cases: (a) If $v_1 \prec v$, then $v_1 \prec v \prec v_2$ and $|v| - |v_1| \leq |v| - |v_0| \leq b$ proving the second item. (b) Otherwise, we have $v_0 \prec v \preceq v_1 \prec v_2 \preceq v_3$ with $v, v_3 \in \text{dom}(L)$ and $|v_2| - |v_1| > b$. Since $|v_3| - |v| < |v_3| - |v_0|$, we can apply the induction hypothesis to $v \preceq v_1 \prec v_2 \preceq v_3$ to prove the statement. \square

In the proofs of Theorems 31 and 33 we will see applications of these properties and the following linking theorem, which we also recall from [11].

Proposition 30 [11, Theorem 4] Let Ω and Ψ be ranked alphabets with $\Psi_0 \neq \emptyset$ and $\Psi_1 \neq \emptyset$. Let $k, n \geq 1$, and let M_1, \dots, M_k be \mathcal{A} -xt^R such that

$$\{(c'[t_1, \dots, t_n], c''[t_1, \dots, t_n]) \mid t_1, \dots, t_n \in T_\Psi\} \subseteq \tau(M_1); \dots; \tau(M_k),$$

where $c', c'' \in T_\Omega(X_n)$ are linear and nondeleting in X_n . There exist trees $t_1, \dots, t_n \in T_\Psi$, dependencies

$$\langle u_0, L_1, u_1 \rangle \in \mathcal{D}(M_1), \langle u_1, L_2, u_2 \rangle \in \mathcal{D}(M_2), \dots, \langle u_{k-1}, L_k, u_k \rangle \in \mathcal{D}(M_k)$$

with $u_0 = c'[t_1, \dots, t_n]$ and $u_k = c''[t_1, \dots, t_n]$, and a family $(v_{ij}, w_{ij}) \in L_j$ of links for $1 \leq i \leq n$ and $1 \leq j \leq k$, such that for all $1 \leq i \leq n$:

- (i) $\text{pos}_{x_i}(c'') \preceq w_{ik}$,
- (ii) $v_{i(j+1)} \preceq w_{ij}$ for all $1 \leq j \leq k-1$, and
- (iii) $\text{pos}_{x_i}(c') \preceq v_{i1}$. \square

Intuitively, the items mean that (i) position w_{ik} is in the subtree t_i of the output tree $u_k = c''[t_1, \dots, t_n]$, (ii) position w_{ij} has prefix $v_{i(j+1)}$ in the intermediate tree u_j , and (iii) position v_{i1} is in the subtree t_i of the input tree $u_0 = c'[t_1, \dots, t_n]$.

To show that an integer $k > 1$ is the least power at which the closure under composition is achieved for a class \mathcal{C} , we present a tree transformation $\tau \in \mathcal{C}^k$ that is not in \mathcal{C}^{k-1} . Roughly speaking, this is achieved by deducing certain links given the tree transformation with the help of Proposition 30. These links are necessary in the dependency for the determined input-output tree pairs. Thus, we obtain a partial specification of several dependencies in the sense that we know some of its links, but not necessarily all of them. Then we consider

whether these partial specifications can be implemented by a composition of $\not\{l\text{-xt}}^{\text{R}}$. It can be seen from Proposition 30 that we will often not be able to identify both positions of a link exactly, but rather determine that one of its positions has a certain other prominent position as prefix. In such cases, we graphically display the link using a spline with an inverted arrow head that points to the subtree rooted at that prominent position (instead of to the actual position). For example, the splines in Fig. 3 indicate that a position of t on the left (resp. u on the right) is linked to position 2 on the right (resp. on the left).

We now prove that 3 is the least power at which the class $\not\{l\text{-XT}}^{\text{R}}$ is closed under composition.

Theorem 31 $(\not\{l\text{-XT}}^{\text{R}})^2 \subsetneq \not\{l\text{-XT}}^3$

PROOF. The inclusion follows from Lemma 15. To prove the strictness, let $M'_1 = (Q', \Sigma, \Delta, \{\star\}, R')$ be the $\not\{nl\text{-xt}}$ that is obtained from the $\not\{l\text{-xt}}^{\text{R}}$ M_1 of Example 2 by removing the state q^{la} and all rules for the input symbol σ ; i.e., the rules $\sigma(q, q^{\text{la}}) \xrightarrow{q} q$ and $\sigma(q^{\text{la}}, q) \xrightarrow{q} q$. Thus, $\tau(M'_1)$ is the restriction of $\tau(M_1)$ to input trees that do not contain any occurrence of σ . In addition, we use the two bimorphisms $B_2, B_3 \in \mathcal{B}(\text{snl}, \text{snl})$ of [5, Section II-2-2-3-1], where strictness is denoted by ‘e’ and nondeletion by ‘c’. These bimorphisms are similar to the two bimorphisms that are used in [3, Section 3.4] to prove Proposition 16. By Proposition 11, $\mathcal{B}(\text{snl}, \text{snl}) = \not\{snl\text{-XT}}$, hence B_2 and B_3 can also be defined by $\not\{snl\text{-xt}}$ M_2 and M_3 , respectively. For convenience, we present M_2 and M_3 explicitly before we show that $\tau = \tau(M'_1); \tau(M_2); \tau(M_3)$ cannot be computed by a composition of two $\not\{l\text{-xt}}^{\text{R}}$.

Let $M_2 = (Q_2, \Delta, \Gamma, \{\star\}, R_2)$ be the $\not\{snl\text{-xt}}$ with $Q_2 = \{\star, \text{id}, \text{id}'\}$, the ranked alphabet $\Gamma = \{\sigma^{(2)}, \gamma^{(1)}, \alpha^{(0)}\}$, and the set R_2 consisting of the rules

$$\begin{array}{ll} \sigma_1(\star, \sigma_2(\text{id}, \text{id}')) \xrightarrow{\star} \sigma(\sigma(\star, \text{id}), \text{id}') & \gamma(\text{id}) \xrightarrow{\text{id}, \text{id}'} \gamma(\text{id}) \\ \sigma_2(\text{id}, \text{id}') \xrightarrow{\star} \sigma(\text{id}, \text{id}') & \alpha \xrightarrow{\text{id}, \text{id}'} \alpha . \end{array}$$

Moreover, let $M_3 = (Q_3, \Gamma, \Delta, \{\star\}, R_3)$ be the $\not\{snl\text{-xt}}$ with $Q_3 = \{\star, p, \text{id}, \text{id}'\}$ and the set R_3 consisting of the rules

$$\begin{array}{ll} \sigma(p, \text{id}) \xrightarrow{\star} \sigma_1(p, \text{id}) & \gamma(\text{id}) \xrightarrow{p, \text{id}, \text{id}'} \gamma(\text{id}) \\ \sigma(\sigma(p, \text{id}), \text{id}') \xrightarrow{p} \sigma_1(p, \sigma_2(\text{id}, \text{id}')) & \alpha \xrightarrow{p, \text{id}, \text{id}'} \alpha . \end{array}$$

Note that both $\tau(M_2)$ and $\tau(M_3)$ are partial functions.

We present a proof by contradiction, so we assume that $\tau = \tau(N_1); \tau(N_2)$ for some $\not\{l\text{-xt}}^{\text{R}}$ N_1 and N_2 . By Proposition 28 there exist $a_1, a_2, b_1, b_2 \geq 1$ such that $\mathcal{D}(N_1)$ and $\mathcal{D}(N_2)$ are strictly input and output hierarchical, input link-distance bounded by a_1 and a_2 , respectively, and output link-distance

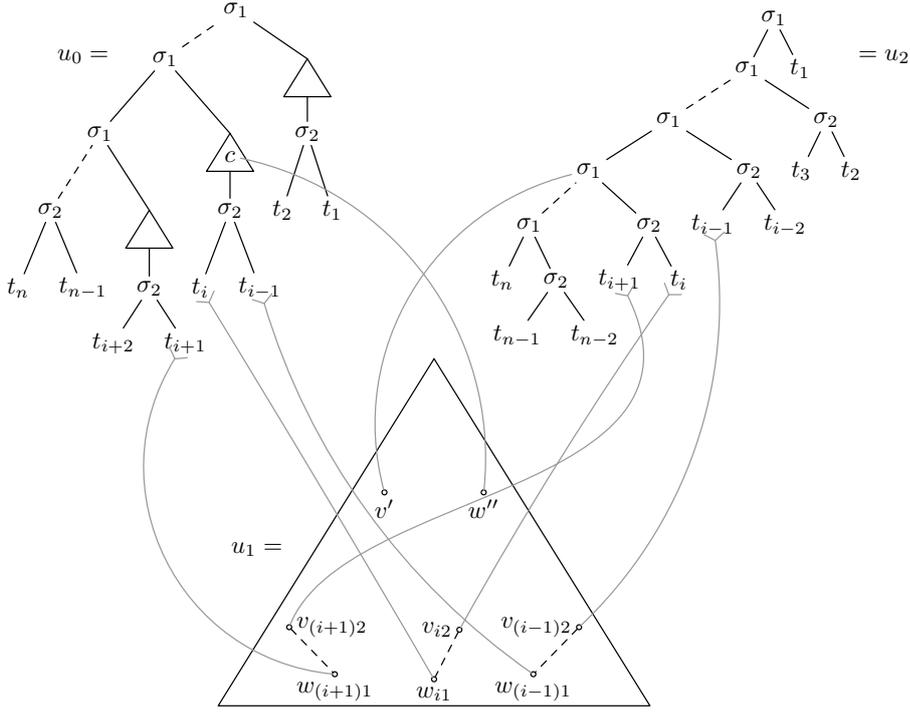


Fig. 4 Illustration of the relevant part of the specification used in the proof of Theorem 31.

bounded by b_1 and b_2 , respectively. Let $n = 2 \cdot \max(a_1, a_2, b_1, b_2) + 2$. We select the trees

$$\begin{aligned} c &= \gamma_2^n(x_1) , \\ c' &= \sigma_1(\sigma_1(\cdots \sigma_1(\sigma_2(x_n, x_{n-1}), c[\sigma_2(x_{n-2}, x_{n-3})]) \cdots , c[\sigma_2(x_4, x_3)]), \\ &\quad c[\sigma_2(x_2, x_1)]) , \text{ and} \\ c'' &= \sigma_1(\sigma_1(\cdots \sigma_1(x_n, \sigma_2(x_{n-1}, x_{n-2})) \cdots , \sigma_2(x_3, x_2)), x_1) , \end{aligned}$$

of which c' and c'' are linear and nondeleting in X_n (see Fig. 4). To be completely formal, c' and c'' are defined inductively as follows: First, $c' = c'_1$ with $c'_{n-1} = \sigma_2(x_n, x_{n-1})$ and $c'_{i-1} = \sigma_1(c'_{i+1}, c[\sigma_2(x_i, x_{i-1})])$ for every even integer $2 \leq i \leq n-2$. Second, we let $c'' = \sigma_1(c''_2, x_1)$ with $c''_n = x_n$ and $c''_{i-2} = \sigma_1(c''_i, \sigma_2(x_{i-1}, x_{i-2}))$ for every even $4 \leq i \leq n$.

It is straightforward to check that

$$(c'[t_1, \dots, t_n], c''[t_1, \dots, t_n]) \in \tau$$

for all $t_1, \dots, t_n \in T_\Psi$ with $\Psi = \{\gamma^{(1)}, \alpha^{(0)}\}$. Note that, according to Example 8, every σ_1 -rib $c[\sigma_2(t_i, t_{i-1})] = \gamma_2^n(\sigma_2(t_i, t_{i-1}))$ is transformed into $\sigma_2(t_i, t_{i-1})$ by $\tau(M'_1)$. Consequently, we can apply Proposition 30 to obtain that there exist trees $t_1, \dots, t_n \in T_\Psi$, dependencies $(c'[t_1, \dots, t_n], L_1, u_1) \in \mathcal{D}(N_1)$ and

$\langle u_1, L_2, c''[t_1, \dots, t_n] \rangle \in \mathcal{D}(N_2)$, and links $(v_{i1}, w_{i1}) \in L_1$ and $(v_{i2}, w_{i2}) \in L_2$ for $1 \leq i \leq n$ such that

- $\text{pos}_{x_i}(c'') \preceq w_{i2}$,
- $v_{i2} \preceq w_{i1}$, and
- $\text{pos}_{x_i}(c') \preceq v_{i1}$.

The splines with the inverted arrow heads indicate some of those links in Fig. 4.

Now, let us consider the obtained (partial) dependencies, which are depicted in Fig. 4. We clearly have $(\varepsilon, \varepsilon), (v_{n2}, w_{n2}) \in L_2$ and

$$1^{\frac{n}{2}} = \text{pos}_{x_n}(c'') \preceq w_{n2} .$$

Thus $|w_{n2}| \geq \frac{n}{2} > b_2$. Since $\mathcal{D}(N_2)$ is output link-distance bounded by b_2 , there exists a link $(v', w') \in L_2$ with $\varepsilon \prec w' \prec w_{n2}$ and $|w'| \leq b_2$. Consequently, the position w' has label σ_1 in $u_2 = c''[t_1, \dots, t_n]$ as indicated in Fig. 4. Formally, $w' = 1^m$ for some $1 \leq m \leq b_2 \leq \frac{n}{2} - 1$. Let $i = 2m$, which yields $2 \leq i \leq n - 2$. Then $w' \prec w_{(i+1)2}$ and $w' \prec w_{i2}$ because $w' \prec \text{pos}_{x_{i+1}}(c'') \preceq w_{(i+1)2}$ and $w' \prec \text{pos}_{x_i}(c'') \preceq w_{i2}$. Since $\mathcal{D}(N_2)$ is strictly output hierarchical, we can conclude that $v' \preceq v_{(i+1)2} \preceq w_{(i+1)1}$ and $v' \preceq v_{i2} \preceq w_{i1}$. Additionally, w' and $\text{pos}_{x_{i-1}}(c'')$ are incomparable and $\text{pos}_{x_{i-1}}(c'') \preceq w_{(i-1)2}$, so also the positions w' and $w_{(i-1)2}$ are incomparable (see Lemma 32(i) for a proof of this and similarly straightforward arguments). Consequently, Lemma 29(i) shows that v' and $v_{(i-1)2}$ are also incomparable. Using $v_{(i-1)2} \preceq w_{(i-1)1}$, we obtain that v' and $w_{(i-1)1}$ are incomparable, and in particular that $v' \not\preceq w_{(i-1)1}$.

Next, we inspect the input tree $u_0 = c'[t_1, \dots, t_n]$ and the links $(\varepsilon, \varepsilon), (v_{i1}, w_{i1})$, and $(v_{(i-1)1}, w_{(i-1)1})$ in L_1 . We already know that $\text{pos}_{x_i}(c') \preceq v_{i1}$ and $\text{pos}_{x_{i-1}}(c') \preceq v_{(i-1)1}$. Let

$$V = \{v \in 1^*2\mathbb{N}^* \mid v \prec \text{pos}_{x_i}(c'), c'(v) \neq \sigma_2\}$$

be the set of positions of c' (and hence of u_0) that are in an occurrence of the tree c and are prefixes of $\text{pos}_{x_i}(c')$. Since $|V| = n > a_1$, it follows from Lemma 29(ii) that there exists a link $(v'', w'') \in L_1$ such that $v'' \in V$, which also yields $v'' \prec v_{i1}$ and $v'' \prec v_{(i-1)1}$. Since $\mathcal{D}(N_1)$ is strictly input hierarchical, we obtain that $w'' \preceq w_{i1}$ and $w'' \preceq w_{(i-1)1}$. Since v'' and $v_{(i+1)1}$ are incomparable, Lemma 29(i) implies that w'' and $w_{(i+1)1}$ are incomparable, and in particular that $w'' \not\preceq w_{(i+1)1}$.

Summing up, we have

$$\begin{array}{lll} v' \preceq w_{(i+1)1} & v' \preceq w_{i1} & v' \not\preceq w_{(i-1)1} \quad (\dagger) \\ w'' \not\preceq w_{(i+1)1} & w'' \preceq w_{i1} & w'' \preceq w_{(i-1)1} \quad (\ddagger) \end{array}$$

Since $v' \preceq w_{i1}$ and $w'' \preceq w_{i1}$, we must have $v' \preceq w''$ or $w'' \preceq v'$. In the former case, we obtain $v' \preceq w'' \preceq w_{(i-1)1}$ contradicting the last statement of (\dagger) . Similarly, in the second case, we obtain $w'' \preceq v' \preceq w_{(i+1)1}$ contradicting the first statement of (\ddagger) . Since both cases are contradictory, the assumption that we can compute τ with two $\not\preceq$ -xt^R is wrong. \square

Fortunately, we can reuse the ideas used in the proof of Theorem 31 to conclude that 4 is the least power at which the class $\not\mathcal{A}$ -XT is closed under composition. The slightly more elaborate proof first establishes that a deleting rule, which is a rule $\ell \xrightarrow{q} r$ such that $\text{states}(r) \subsetneq \text{states}(\ell)$, must be used at a certain position and then employs the classical cut-and-paste technique to establish that this deletion (without look-ahead) enables undesired translations.

We will use some well-known elementary properties of the prefix order, which we state in the next lemma.

Lemma 32 Let $v, v_1, v_2, v'_1, v'_2 \in \mathbb{N}^*$ be positions with $v_1 \preceq v'_1$ and $v_2 \preceq v'_2$.

- (i) If v_1 and v_2 are incomparable, then so are v'_1 and v'_2 .
- (ii) If v_1 and v_2 are incomparable, $v \preceq v'_1$, and $v \preceq v'_2$, then $v \preceq v_1$ and $v \preceq v_2$.
- (iii) If v_1 and v'_2 are incomparable and v'_1 and v_2 are incomparable, then v_1 and v_2 are incomparable.

PROOF. If v_1 and v_2 are incomparable, then $\text{lcp}(v_1, v_2)$ is a proper prefix of both v_1 and v_2 . Hence $\text{lcp}(v'_1, v'_2) = \text{lcp}(v_1, v_2)$, which implies the first two items. For the third item we note that if $v_1 \preceq v_2$ then $v_1 \preceq v'_2$, and symmetrically, if $v_2 \preceq v_1$ then $v_2 \preceq v'_1$. \square

Theorem 33 $\not\mathcal{A}$ -XT³ \subsetneq ($\not\mathcal{A}$ -XT^R)³

PROOF. Since the inclusion is trivial, it remains to prove its strictness. Let M_1 be the $\not\mathcal{A}$ -xt^R of Example 2, and let M_2 and M_3 be the bimorphisms defined as $\not\mathcal{A}$ -xt in the proof of Theorem 31. We will show that the tree transformation $\tau = \tau(M_1); \tau(M_2); \tau(M_3)$ cannot be computed by a composition of three $\not\mathcal{A}$ -xt.

We again present a proof by contradiction, hence we assume that

$$\tau = \tau(N_1); \tau(N_2); \tau(N_3)$$

for some $\not\mathcal{A}$ -xt $N_1 = (P, \Sigma, \Delta_1, P_0, R_1)$, N_2 , and N_3 . By Proposition 28 there exist integers $a_1, a_2, a_3, b_1, b_2, b_3 \geq 1$ such that $\mathcal{D}(N_1)$, $\mathcal{D}(N_2)$, and $\mathcal{D}(N_3)$ are strictly input and output hierarchical, input link-distance bounded by a_1 , a_2 , and a_3 , respectively, and output link-distance bounded by b_1 , b_2 , and b_3 , respectively. As before, let $n = 2 \cdot \max(a_1, a_2, a_3, b_1, b_2, b_3) + 2$. Moreover, let $m \in \mathbb{N}$ be such that $m > \text{ht}(\ell)$ for all rules $(\ell, p, r) \in R_1$. This time, we select the trees

$$\begin{aligned} s &= \gamma_2^m(\alpha) , \\ c &= \sigma(s, \sigma(s, \dots \sigma(s, x_1) \dots)) \text{ with } n^2 \text{ occurrences of } \sigma , \\ c' &= \sigma_1(\sigma_1(\dots \sigma_1(\sigma_2(x_n, x_{n-1}), c[\sigma_2(x_{n-2}, x_{n-3})]) \dots , c[\sigma_2(x_4, x_3)]), \\ &\quad c[\sigma_2(x_2, x_1)]]) , \text{ and} \\ c'' &= \sigma_1(\sigma_1(\dots \sigma_1(x_n, \sigma_2(x_{n-1}, x_{n-2})) \dots , \sigma_2(x_3, x_2)), x_1) . \end{aligned}$$

We note that c' and c'' are the same as in the proof of Theorem 31 (see Fig. 4), except that we selected a more complicated tree c ; thus, c' and c'' are again linear and nondeleting in X_n , and can be defined formally as in that

proof. Clearly $(c'[t_1, \dots, t_n], c''[t_1, \dots, t_n]) \in \tau$ for all $t_1, \dots, t_n \in T_{\Psi}$ with $\Psi = \{\gamma^{(1)}, \alpha^{(0)}\}$. This time, every σ_1 -rib $c[\sigma_2(t_i, t_{i-1})]$ is of the form

$$\sigma(\gamma_2^m(\alpha), \sigma(\gamma_2^m(\alpha), \dots \sigma(\gamma_2^m(\alpha), \sigma_2(t_i, t_{i-1})) \dots)) .$$

It is transformed into $\sigma_2(t_i, t_{i-1})$ by $\tau(M_1)$ as before (see Example 8). So we can apply Proposition 30 once again to obtain that there exist $t_1, \dots, t_n \in T_{\Psi}$, dependencies

$$\begin{aligned} \langle c'[t_1, \dots, t_n], L_1, u_1 \rangle &\in \mathcal{D}(N_1) , & \langle u_1, L_2, u_2 \rangle &\in \mathcal{D}(N_2) \quad \text{and} \\ \langle u_2, L_3, c''[t_1, \dots, t_n] \rangle &\in \mathcal{D}(N_3) , \end{aligned}$$

and links $(v_{i1}, w_{i1}) \in L_1$, $(v_{i2}, w_{i2}) \in L_2$, and $(v_{i3}, w_{i3}) \in L_3$ for $1 \leq i \leq n$ such that

- $\text{pos}_{x_i}(c'') \preceq w_{i3}$,
- $v_{i3} \preceq w_{i2}$ and $v_{i2} \preceq w_{i1}$, and
- $\text{pos}_{x_i}(c') \preceq v_{i1}$.

We first observe that for every $j \in \{1, 2, 3\}$, the positions v_{1j}, \dots, v_{nj} are pairwise incomparable (as also shown in the proof of Proposition 30 in [11, Theorem 4]). In fact, since $\text{pos}_{x_1}(c''), \dots, \text{pos}_{x_n}(c'')$ are pairwise incomparable, so are w_{13}, \dots, w_{n3} by the first item above and Lemma 32(i). Hence the corresponding link origins v_{13}, \dots, v_{n3} are pairwise incomparable by Lemma 29(i). This implies that w_{12}, \dots, w_{n2} are pairwise incomparable by the second item above, and hence so are the corresponding link origins v_{12}, \dots, v_{n2} using again Lemma 29(i). This argument can be repeated once more to show the observation.

We now start the analysis of the given dependencies in the same way as in the proof of Theorem 31 by considering the output tree $u_3 = c''[t_1, \dots, t_n]$. Entirely similar to that proof, we obtain a position $v' \in \text{pos}(u_2)$ such that $v' \preceq w_{(i+1)2}$, $v' \preceq w_{i2}$, and $v' \not\preceq w_{(i-1)2}$.

Next we move to the input tree $u_0 = c'[t_1, \dots, t_n]$, where the analysis will be slightly different. As before, we consider the links $(\varepsilon, \varepsilon)$, (v_{i1}, w_{i1}) , and $(v_{(i-1)1}, w_{(i-1)1})$ in L_1 , for which we already know that $\text{pos}_{x_i}(c') \preceq v_{i1}$ and $\text{pos}_{x_{i-1}}(c') \preceq v_{(i-1)1}$. Let

$$V = \{v \in 1^*2\mathbb{N}^* \mid v \prec \text{pos}_{x_i}(c'), c'(v) \neq \sigma_2\} .$$

Clearly, $|V| = n^2 > n \cdot a_1$. Thus, since $\mathcal{D}(N_1)$ is input link-distance bounded by a_1 , the set $V' = \{v \in V \mid \exists w \in \mathbb{N}^*: (v, w) \in L_1\}$ of link origins in V contains at least n elements by Lemma 29(ii). Let $W' = \{w \mid \exists v \in V': (v, w) \in L_1\}$ be the set of corresponding link targets. Since the elements of V' are pairwise comparable, the elements of W' are also pairwise comparable by Lemma 29(i). For every $w \in W'$, we have $w \preceq w_{i1}$ and $w \preceq w_{(i-1)1}$ because $v \prec v_{i1}$ and $v \prec v_{(i-1)1}$ for every $v \in V'$ and $\mathcal{D}(N_1)$ is strictly input hierarchical. Additionally, for every $w \in W'$, the positions w and $w_{(i+1)1}$ are incomparable because v and $v_{(i+1)1}$ are incomparable for every $v \in V'$. Since v_{i2} and $v_{(i-1)2}$ are incomparable by the above observation, and $v_{i2} \preceq w_{i1}$ and $v_{(i-1)2} \preceq w_{(i-1)1}$,

we obtain from Lemma 32(ii) that $w \preceq v_{i2}$ and $w \preceq v_{(i-1)2}$ for every $w \in W'$. Moreover, for every $w \in W'$, since $v_{(i+1)2} \preceq w_{(i+1)1}$, $w \preceq v_{i2}$, $v_{(i+1)2}$ and v_{i2} are incomparable, and w and $w_{(i+1)1}$ are incomparable, Lemma 32(iii) shows that w and $v_{(i+1)2}$ are incomparable.

Now we distinguish two cases. First, let us assume that $|W'| \geq n$. In this case, we can continue to derive a contradiction in much the same way as in the proof of Theorem 31. Since the positions in W' are pairwise comparable, there are positions $w_{\min}, w_{\max} \in W'$ of minimal and maximal length, respectively, with $w_{\min} \prec w_{\max}$. Clearly, $|w_{\max}| - |w_{\min}| \geq n - 1 > a_2$. Since $(\varepsilon, \varepsilon), (v_{i2}, w_{i2}) \in L_2$ and $w_{\min} \preceq v_{i2}$, there must be a link $(v'', w'') \in L_2$ such that $w_{\min} \prec v'' \prec w_{\max}$ by Lemma 29(ii). This implies that $v'' \prec v_{i2}$, $v'' \prec v_{(i-1)2}$, and v'' and $v_{(i+1)2}$ are incomparable. Since $\mathcal{D}(N_2)$ is strictly input hierarchical, we obtain that $w'' \preceq w_{i2}$, $w'' \preceq w_{(i-1)2}$, and from Lemma 29(i) we obtain that w'' and $w_{(i+1)2}$ are incomparable, which takes us to the situation

$$\begin{array}{lll} v' \preceq w_{(i+1)2} & v' \preceq w_{i2} & v' \not\preceq w_{(i-1)2} \\ w'' \not\preceq w_{(i+1)2} & w'' \preceq w_{i2} & w'' \preceq w_{(i-1)2} \end{array} ,$$

which is the analogue of (\dagger) and (\ddagger) [in the proof of Theorem 31] and thus contradictory for the same reasons.

In the remaining case, we have $|W'| < n$. Together with $|V'| \geq n$, we obtain by the pigeonhole principle that several input positions of V' are linked in L_1 to the same output position \bar{w} of W' . We choose $(\bar{v}, \bar{w}) \in L_1$ such that $\bar{v} \in V'$ and $|\bar{v}|$ is minimal. Consequently, a rule $(\ell, p, r) \in R_1$ with a state $r \in P$ as right-hand side must have been applied at position \bar{v} of $u_0 = c'[t_1, \dots, t_n]$. Since $\bar{v} \in V$, the subtree $t|_{\bar{v}}$ is of the form $\sigma(s, \sigma(s, \dots \sigma(s, \sigma_2(t_i, t_{i-1})) \dots))$, where $s = \gamma_2^m(\alpha)$. Hence, since N_1 is ε -free, the root of the left-hand side ℓ has label σ . Moreover, $\ell|_1 = \gamma_2^k(p')$ for some $0 \leq k < m$ and $p' \in P$. By the choice of \bar{v} , the state r occurs in $\ell|_2$ and so the state p' is deleted [i.e., $p' \notin \text{states}(r)$] in this rule. Therefore, the subtree $u_0|_{\bar{v}.1^{k+1}} = \gamma_2^{m-k}(\alpha)$ has been created using the second item of Definition 6. Since N_1 is an $\not\preceq$ -xt, its look-ahead mapping is trivial, and thus any tree can be created instead of $u_0|_{\bar{v}.1^{k+1}}$; e.g., the tree $\sigma_2(\alpha, \alpha)$. This shows that also $\langle u'_0, L_1, u_1 \rangle \in \mathcal{D}(N_1)$, where $u'_0 = u_0[\bar{v}.1^{k+1} \leftarrow \sigma_2(\alpha, \alpha)]$, and so $(u'_0, c''[t_1, \dots, t_n]) \in \tau(N_1); \tau(N_2); \tau(N_3)$. However, since $u'_0|_{\bar{v}}$ is of the form

$$\sigma(\gamma_2^k(\sigma_2(\alpha, \alpha)), \sigma(s, \dots \sigma(s, \sigma_2(t_i, t_{i-1})) \dots)) ,$$

the σ_1 -rib $u'_0|_{1^{h2}}$ of u'_0 with $1^{h2} \preceq \bar{v}$ (i.e., $h = \frac{i}{2} - 1$) has two occurrences of σ_2 . Hence u'_0 is not in the domain of $\tau(M_1)$ [see Example 8]. This implies that u'_0 is not in the domain of $\tau = \tau(M_1); \tau(M_2); \tau(M_3)$, but

$$(u'_0, c''[t_1, \dots, t_n]) \in \tau(N_1); \tau(N_2); \tau(N_3) = \tau ,$$

which is a contradiction.

Since both cases are contradictory, τ cannot be computed by a composition of three $\not\preceq$ -xt. \square

Class	Least power of closedness	Stated in
$\mathcal{B}(1,1)$	4	[5, Section II-2-2-3-3]
$\not\#sl\text{-XT} = \not\#sl\text{-XT}^R$	2	[3, Theorem 6.2]
$\not\#sl\text{-XT}^R, \not\#sl\text{-XT}$	2	Theorem 26
$\not\#l\text{-XT}^R$	3	Theorems 31 and 34
$\not\#l\text{-XT}$	4	Theorems 33 and 34

Table 3 Summary of the results of Section 5.

Thus, we have shown that the least power, at which the composition closure is achieved for the classes $\not\#l\text{-XT}^R$ and $\not\#l\text{-XT}$, is 3 and 4, respectively. This is stated in the next theorem.

Theorem 34 For every $n \geq 4$,

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

PROOF. We have $(\not\#l\text{-XT}^R)^n \subseteq \not\#l\text{-XT}^{n+1}$ for all $n \geq 1$ by Lemma 15. The equalities follow from Theorem 24. The fourth and fifth inclusions are strict by Theorems 31 and 33, respectively. The strictness of the second and third inclusion follows from that of the fourth and fifth, respectively. The strictness of the first inclusion is a consequence of Proposition 14; it also follows from that of the third. \square

In Table 3 we summarize the main results of this and the previous section, which allow us to present the least power at which the closure of the considered composition hierarchies is achieved. For the sake of completeness, we also present the corresponding results for the classes $\not\#sl\text{-XT}$ and $\mathcal{B}(1,1)$ that were obtained in [3,5]. Recall that $\mathcal{B}(1,1)$ is the class of all tree transformations computable by bimorphisms, in which both tree homomorphisms are linear.

6 Infinite composition hierarchies

To complete the picture, we will need one further result showing the infiniteness of the composition hierarchy for a large number of classes. In order to obtain a result that is as general as possible, we use bimorphisms [3] instead of $l\text{-xt}^R$ in this section; cf. Proposition 11. We conclude several results for various tree transducer classes from the result for bimorphisms.

To handle bimorphisms properly, we need to define links for tree homomorphisms. As observed after Notation 10, every linear tree homomorphism $\varphi: T_\Sigma \rightarrow T_\Delta$ can be viewed as a linear top-down tree transducer M_φ . In particular, for every $t \in T_\Sigma$ there is a (unique) set $L_\varphi(t) \subseteq \text{pos}(t) \times \text{pos}(\varphi(t))$

of links such that $\langle t, L_\varphi(t), \varphi(t) \rangle \in \mathcal{D}(M_\varphi)$. We now generalize this notion to arbitrary tree homomorphisms.

Definition 35 Let $\varphi: T_\Sigma \rightarrow T_\Delta$ be a tree homomorphism and $t \in T_\Sigma$. The set of t -links of φ , denoted by $L_\varphi(t)$, is the smallest subset of $\text{pos}(t) \times \text{pos}(\varphi(t))$ such that

- $(\varepsilon, \varepsilon) \in L_\varphi(t)$ and
- $(vi, ww') \in L_\varphi(t)$ for all links $(v, w) \in L_\varphi(t)$, integers $1 \leq i \leq \text{rk}(t(v))$, and positions $w' \in \text{pos}_{x_i}(\varphi(t(v)))$. \square

Intuitively, $(v, w) \in L_\varphi(t)$ means that φ translates the subtree of t rooted at v into the subtree of $\varphi(t)$ rooted at w . Note that for a given position v there can be several such positions w (which are, of course, pairwise incomparable), since φ is not necessarily linear, or there may be no such w , since φ is not necessarily nondeleting. We will need the following elementary properties of $L_\varphi(t)$.

Lemma 36 Let $\varphi: T_\Sigma \rightarrow T_\Delta$ be a tree homomorphism, and let $t \in T_\Sigma$, $u = \varphi(t)$, and $L = L_\varphi(t)$.

- (i) If $(v, w) \in L$, then $\varphi(t|_v) = u|_w$.
- (ii) If $(v, w) \in L$, then $L_\varphi(t|_v) = \{(v', w') \mid (vv', ww') \in L\}$.
- (iii) If φ is nondeleting, then for all $(v_1, w_1) \in L$ and all $v_1 \preceq v \in \text{pos}(t)$ there exists a position $w_1 \preceq w$ such that $(v, w) \in L$.
- (iv) For all links $(v_1, w_1), (v_2, w_2) \in L$ with $v_1 \preceq v_2$ and $w_1 \preceq w_2$, and all $v_1 \preceq v \preceq v_2$ there exists a unique position $w_1 \preceq w \preceq w_2$ such that $(v, w) \in L$.
- (v) For all $(v_1, w_1) \in L$ and all $w_1 \preceq w \in \text{pos}(u)$ there exist unique positions $v, w', w'' \in \mathbb{N}^*$ such that $v_1 \preceq v$, $w_1 \preceq w'$, $w = w'w''$, $(v, w') \in L$, and $w'' \in \text{pos}_\Delta(\varphi(t(v)))$.

PROOF. The proofs of statements (i) and (ii) are straightforward, and hence left to the reader. It is also straightforward to prove the following three statements, which are the special case of statements (iii)–(v), in which we have $(v_1, w_1) = (\varepsilon, \varepsilon)$. We also leave their proofs to the reader.

- (iii)' If φ is nondeleting, then $\text{dom}(L) = \text{pos}(t)$.
- (iv)' For all $(v_2, w_2) \in L$ and all $v \preceq v_2$ there exists a unique position $w \preceq w_2$ such that $(v, w) \in L$.
- (v)' For every $w \in \text{pos}(u)$ there exist unique positions $v, w', w'' \in \mathbb{N}^*$ such that $w = w'w''$, $(v, w') \in L$, and $w'' \in \text{pos}_\Delta(\varphi(t(v)))$.

Each non-primed statement can now easily be obtained from the corresponding primed statement with the help of (i) and (ii). We start with statement (iii). Let $(v_1, w_1) \in L$ and $v_1 \preceq v \in \text{pos}(t)$. Since $v_1 \preceq v$, let \hat{v} be such that $v_1\hat{v} = v$. Obviously $\hat{v} \in \text{pos}(t|_{v_1})$, and consequently, by statement (iii)', there exists \hat{w} such that $(\hat{v}, \hat{w}) \in L_\varphi(t|_{v_1})$. Together with $(v_1, w_1) \in L$ and statement (ii) we conclude that $(v_1\hat{v}, w_1\hat{w}) \in L$. Thus, $(v, w) \in L$ where $w_1 \preceq w = w_1\hat{w}$.

For statement (iv), let $(v_1, w_1), (v_2, w_2) \in L$ with $v_1 \preceq v_2$ and $w_1 \preceq w_2$, and let $v_1 \preceq v \preceq v_2$. Since $w_1 \preceq w_2$, let \hat{w}_2 be such that $w_1\hat{w}_2 = w_2$. Similarly, since

$v_1 \preceq v \preceq v_2$, let $\hat{v} \preceq \hat{v}_2$ such that $v_1\hat{v} = v$ and $v_1\hat{v}_2 = v_2$. Since $(v_1, w_1) \in L$, statement (ii) implies that $(\hat{v}_2, \hat{w}_2) \in L_\varphi(t|_{v_1})$. Thus, since also $\hat{v} \preceq \hat{v}_2$, statement (iv)' implies that there exists $\hat{w} \preceq \hat{w}_2$ such that $(\hat{v}, \hat{w}) \in L_\varphi(t|_{v_1})$. Using statement (ii) again, we have $(v_1\hat{v}, w_1\hat{w}) \in L$. Hence the requirements are fulfilled by $w = w_1\hat{w}$; note that $w_1 \preceq w_1\hat{w} \preceq w_1\hat{w}_2 = w_2$. The uniqueness of w follows immediately from the uniqueness condition in statement (iv)'.

Finally, for statement (v), let $(v_1, w_1) \in L$ and $w_1 \preceq w \in \text{pos}(u)$. By statement (i) we have $\varphi(t|_{v_1}) = u|_{w_1}$. Since $w_1 \preceq w$, let \hat{w} be such that $w_1\hat{w} = w$. Obviously, $\hat{w} \in \text{pos}(u|_{w_1})$. By statement (v)' applied to \hat{w} , there exist $\hat{v}, \hat{w}', \hat{w}''$ such that $\hat{w} = \hat{w}'\hat{w}''$, $(\hat{v}, \hat{w}') \in L_\varphi(t|_{v_1})$, and $\hat{w}'' \in \text{pos}_\Delta(\varphi(t|_{v_1}(\hat{v})))$. Since $(v_1, w_1) \in L$ we can use statement (ii) applied to $(\hat{v}, \hat{w}') \in L_\varphi(t|_{v_1})$ to conclude that $(v_1\hat{v}, w_1\hat{w}') \in L$. Hence the requirements are fulfilled by $v = v_1\hat{v}$, $w' = w_1\hat{w}'$, and $w'' = \hat{w}''$. The uniqueness of v , w' , and w'' follows immediately from the uniqueness condition in statement (v)'.

The unique position $v \in \text{pos}(t)$ corresponding to the position $w \in \text{pos}(u)$ in Lemma 36(v) is informally called the position in t that *creates* the symbol $u(w)$ at w . Since item (v) holds in particular for $(v_1, w_1) = (\varepsilon, \varepsilon)$, that position does not depend on the link $(v_1, w_1) \in L$. Similarly, the unique position w in item (iv) does not depend on the link (v_1, w_1) .

We now turn to the proof of the infiniteness of the composition hierarchies. The main auxiliary notion used in that proof is the assignment of levels to positions in a tree. Let $t \in T_\Sigma$. Since the *branching positions* of t (i.e., those that are labeled by symbols of rank at least 2) will play an essential role, we define the set of branching positions of t , the set of branching positions of t together with two different successor indices, and the set of branching positions along a given path, as follows:

$$\begin{aligned} \text{br}_t &= \{v \in \text{pos}(t) \mid t(v) \notin \Sigma_0 \cup \Sigma_1\} \\ \text{bri}_t &= \{\langle v, i, j \rangle \mid v \in \text{br}_t, 1 \leq i, j \leq \text{rk}(t(v)), i \neq j\} \end{aligned}$$

and for every $v_1, v_2 \in \text{pos}(t)$ with $v_1 \preceq v_2$ we let

$$\text{br}_t(v_1, v_2) = \{v \in \text{br}_t \mid v_1 \preceq v \preceq v_2\} .$$

Let $\ell \geq 2$ be arbitrary (called *distance* in the sequel). We inductively define the sets $\text{PI}_n^\ell(t) \subseteq \text{pos}(t) \times \mathbb{N} \times \mathbb{N}$ of *special positions of level n and distance ℓ with successor indices* and the sets $\text{P}_n^\ell(t) \subseteq \text{pos}(t)$ for the same special positions without successor indices for every $n \in \mathbb{N}$ as follows:

$$\begin{aligned} \text{PI}_0^\ell(t) &= \text{bri}_t \\ \text{P}_0^\ell(t) &= \text{br}_t = \{v \mid \exists i, j: \langle v, i, j \rangle \in \text{PI}_0^\ell(t)\} \\ \text{PI}_{n+1}^\ell(t) &= \{\langle v, i, j \rangle \in \text{bri}_t \mid \exists v_1 \in \text{P}_n^\ell(t): v_1 \preceq v, |\text{br}_t(v_1, v) \cap \text{P}_n^\ell(t)| \geq \ell^{n+1}, \\ &\quad \exists v_2 \in \text{P}_n^\ell(t): v_2 \preceq v, |\text{br}_t(v, v_2) \cap \text{P}_n^\ell(t)| \geq \ell^{n+1}\} \\ \text{P}_{n+1}^\ell(t) &= \{v \mid \exists i, j: \langle v, i, j \rangle \in \text{PI}_{n+1}^\ell(t)\} . \end{aligned}$$

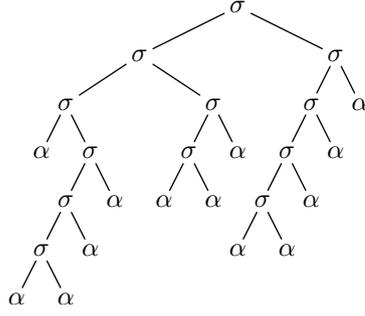


Fig. 5 Tree used in Example 37.

Intuitively, each branching position is a special position of level 0 (for any distance ℓ) and a branching position v is a special position of level $n + 1$ if there are two paths in different direct subtrees below v that both have at least ℓ^{n+1} special positions of level n along the path. Clearly, $\text{PI}_{n+1}^\ell(t) \subseteq \text{PI}_n^\ell(t)$ and $\text{P}_{n+1}^\ell(t) \subseteq \text{P}_n^\ell(t)$ for all $n \in \mathbb{N}$. Note that in the definition of $\text{PI}_{n+1}^\ell(t)$, the condition that $v_1, v_2 \in \text{P}_n^\ell(t)$ is superfluous, but technically convenient.

Example 37 Let t be the tree depicted in Fig. 5. Then

$$\text{P}_0^2(t) = \{\varepsilon, 1, 11, 112, 1121, 11211, 12, 121, 2, 21, 211, 2111\} = \text{br}_t$$

$$\text{P}_1^2(t) = \{\varepsilon, 1\}$$

$$\text{P}_2^2(t) = \emptyset .$$

□

Lemma 38 Let $t \in T_\Sigma$ and $\ell, n \in \mathbb{N}$ with $\ell \geq 2$. Moreover, let $v, v' \in \mathbb{N}^*$ and $i, j \in \mathbb{N}$.

(i) $\langle v', i, j \rangle \in \text{PI}_n^\ell(t|_v)$ if and only if $\langle vv', i, j \rangle \in \text{PI}_n^\ell(t)$, and
 $v' \in \text{P}_n^\ell(t|_v)$ if and only if $vv' \in \text{P}_n^\ell(t)$.

(ii) If $v, viv' \in \text{P}_n^\ell(t)$, then there exists $m \in \mathbb{N}$ such that $\langle v, i, m \rangle \in \text{PI}_n^\ell(t)$.

PROOF. We prove the items individually. We start with (i), which is obvious because whether or not $\langle v, i, j \rangle$ is in $\text{PI}_n^\ell(t)$ only depends on the positions of which v is a prefix. Statement (ii) is also trivial for $n = 0$, hence we only prove it for $n + 1$. Let $v, viv' \in \text{P}_{n+1}^\ell(t)$. Since $v \in \text{P}_{n+1}^\ell(t)$ there exist integers i_1, i_2 such that $\langle v, i_1, i_2 \rangle \in \text{PI}_{n+1}^\ell(t)$. If $i \in \{i_1, i_2\}$, then the statement is obviously true. In the remaining case, let $i \notin \{i_1, i_2\}$. There exists a position $v_2 \in \text{P}_n^\ell(t)$ such that $vi_2 \preceq v_2$ and $|\text{br}_t(vi_2, v_2) \cap \text{P}_n^\ell(t)| \geq \ell^{n+1}$. Since $viv' \in \text{P}_n^\ell(t)$, there exist $i' \in \mathbb{N}$ and $v_1 \in \text{P}_n^\ell(t)$ such that $viv'i' \preceq v_1$ and $|\text{br}_t(viv'i', v_1) \cap \text{P}_n^\ell(t)| \geq \ell^{n+1}$. Hence $vi \preceq v_1$ and $|\text{br}_t(vi, v_1) \cap \text{P}_n^\ell(t)| \geq \ell^{n+1}$, which shows that $\langle v, i, i_2 \rangle \in \text{PI}_{n+1}^\ell(t)$. □

We now prove that a nondeleting tree homomorphism preserves the maximal level of the special positions of a tree.

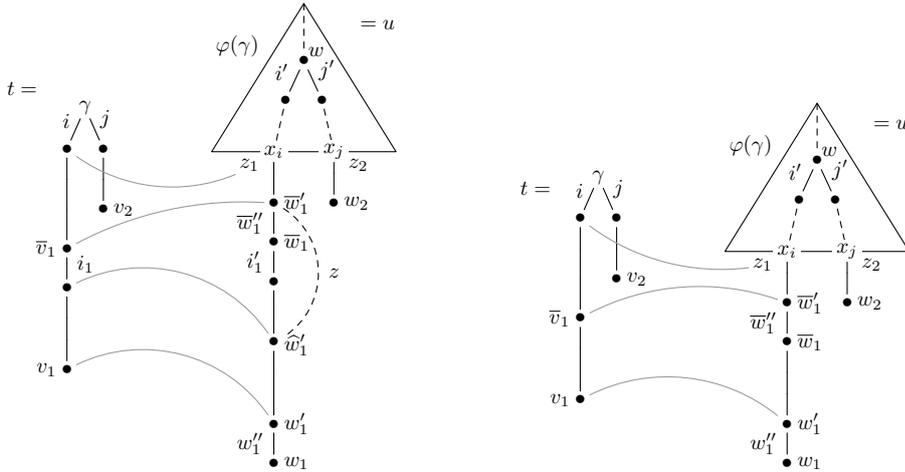


Fig. 6 Illustration of the trees and positions discussed in the proof of Lemma 39 (left) and Lemma 40 (right).

Lemma 39 Let $\varphi: T_\Gamma \rightarrow T_\Delta$ be a nondeleting tree homomorphism, and let $t = \gamma(t_1, \dots, t_k)$ for some $k \in \mathbb{N}$, $\gamma \in \Gamma_k$, and $t_1, \dots, t_k \in T_\Gamma$. Moreover, let $\ell, n, i, j \in \mathbb{N}$ be such that $\ell \geq 2$ and $\langle \varepsilon, i, j \rangle \in \text{PI}_n^\ell(t)$. Then for every $z_1 \in \text{pos}_{x_i}(\varphi(\gamma))$ and $z_2 \in \text{pos}_{x_j}(\varphi(\gamma))$ there exists $\langle w, i', j' \rangle \in \text{PI}_n^\ell(\varphi(t))$ such that $w \in \text{pos}(\varphi(\gamma))$ and $wi' \preceq z_1$ and $wj' \preceq z_2$.

PROOF. Let $u = \varphi(t) = \varphi(\gamma)[\varphi(t_1), \dots, \varphi(t_k)]$ and $L = L_\varphi(t)$. We prove the statement by induction on n . In the induction base, we have $n = 0$ and $\langle \varepsilon, i, j \rangle \in \text{PI}_0^\ell(t) = \text{bri}_t$. Consider $z_1 \in \text{pos}_{x_i}(\varphi(\gamma))$ and $z_2 \in \text{pos}_{x_j}(\varphi(\gamma))$, which are occurrences of the variables $x_i \neq x_j$ in $\varphi(\gamma)$. Let $w = \text{lcp}(z_1, z_2)$ be their longest common prefix. Since $x_i \neq x_j$, we have $w \prec z_1$ and $w \prec z_2$, so let $i', j' \in \mathbb{N}$ be the unique (and necessarily distinct) integers such that $wi' \preceq z_1$ and $wj' \preceq z_2$. Clearly, $w \in \text{pos}(\varphi(\gamma))$ and $\langle w, i', j' \rangle \in \text{bri}_u = \text{PI}_0^\ell(u)$. This completes the induction base.

In the induction step, let $\langle \varepsilon, i, j \rangle \in \text{PI}_{n+1}^\ell(t)$, and suppose that $v_1 \in \text{P}_n^\ell(t)$ and $v_2 \in \text{P}_n^\ell(t)$ are the required special positions of level n such that $i \preceq v_1$ and $j \preceq v_2$ and

$$|\text{br}_t(i, v_1) \cap \text{P}_n^\ell(t)| \geq \ell^{n+1} \quad \text{and} \quad |\text{br}_t(j, v_2) \cap \text{P}_n^\ell(t)| \geq \ell^{n+1} .$$

Now, we follow a similar approach as in the induction base. Figure 6 illustrates the used positions and their relations. Consider positions $z_1 \in \text{pos}_{x_i}(\varphi(\gamma))$ and $z_2 \in \text{pos}_{x_j}(\varphi(\gamma))$. As before, we let

$$w = \text{lcp}(z_1, z_2) \in \text{pos}(\varphi(\gamma))$$

be their longest common prefix, and let $wi' \preceq z_1$ and $wj' \preceq z_2$. Clearly, $i' \neq j'$ and so $\langle w, i', j' \rangle \in \text{bri}_u$. It remains to show that $\langle w, i', j' \rangle \in \text{PI}_{n+1}^\ell(u)$.

By Definition 35, we have $(\varepsilon, \varepsilon) \in L$ and $(i, z_1) \in L$. Since φ is nondeleting and $(i, z_1) \in L$ and $i \preceq v_1$, it follows from Lemma 36(iii) that there exists w'_1 such that $z_1 \preceq w'_1$ and $(v_1, w'_1) \in L$. Thus, Lemma 36(i) shows that $u|_{w'_1} = \varphi(t|_{v_1})$. By assumption we have $v_1 \in P_n^\ell(t)$, which yields $\varepsilon \in P_n^\ell(t|_{v_1})$ by Lemma 38(i); i.e., $\langle \varepsilon, i'', j'' \rangle \in \text{PI}_n^\ell(t|_{v_1})$ for some i'', j'' . Since φ is nondeleting, the sets $\text{pos}_{x_{i''}}(\varphi(t|_{v_1}))$ and $\text{pos}_{x_{j''}}(\varphi(t|_{v_1}))$ are nonempty. Consequently, the induction hypothesis implies the existence of $w''_1 \in P_n^\ell(\varphi(t|_{v_1})) = P_n^\ell(u|_{w'_1})$. Hence $w'_1 w''_1 \in P_n^\ell(u)$ by Lemma 38(i). Let $w_1 = w'_1 w''_1$, and let w_2 be determined in an analogous way. We claim that w_1 and w_2 are the special positions of level n that are required to show that $\langle w, i', j' \rangle \in \text{PI}_{n+1}^\ell(u)$. We will only verify the condition

$$|\text{br}_u(wi', w_1) \cap P_n^\ell(u)| \geq \ell^{n+1} \quad (\dagger)$$

because the proof for w_2 works analogously. Due to $wi' \preceq z_1$, we obtain that $w_1 \in \text{br}_u(wi', w_1) \cap P_n^\ell(u)$.

Let $\bar{v}_1 \in \text{br}_t(i, v_1) \cap P_n^\ell(t)$ be any position of level n along the path from i to v_1 such that $\bar{v}_1 \prec v_1$. Hence there exists a unique integer $i_1 \in \mathbb{N}$ such that $\bar{v}_1 i_1 \preceq v_1$. Since $(i, z_1), (v_1, w'_1) \in L$ together with $i \preceq \bar{v}_1 i_1 \preceq v_1$ we can use Lemma 36(iv) to conclude that there exists $z_1 \preceq \hat{w}'_1 \preceq w'_1$ such that $(\bar{v}_1 i_1, \hat{w}'_1) \in L$. Applied once more to $i \preceq \bar{v}_1 \preceq \bar{v}_1 i_1$ and the links $(i, z_1), (\bar{v}_1 i_1, \hat{w}'_1) \in L$, there exists $z_1 \preceq \bar{w}'_1 \preceq \hat{w}'_1$ with $(\bar{v}_1, \bar{w}'_1) \in L$. Let $z \in \mathbb{N}^*$ be such that $\bar{w}'_1 z = \hat{w}'_1$. By Definition 35 we have $z \in \text{pos}_{x_{i_1}}(\varphi(t|_{\bar{v}_1}))$. Since $\bar{v}_1, v_1 \in P_n^\ell(t)$, we conclude from Lemma 38(ii) that there exists $j_1 \in \mathbb{N}$ such that $\langle \bar{v}_1, i_1, j_1 \rangle \in \text{PI}_n^\ell(t)$. Hence $\langle \varepsilon, i_1, j_1 \rangle \in \text{PI}_n^\ell(t|_{\bar{v}_1})$ by Lemma 38(i) and $u|_{\bar{w}'_1} = \varphi(t|_{\bar{v}_1})$ by Lemma 36(i). Now we can apply the induction hypothesis to obtain that there exists $\langle \bar{w}''_1, i'_1, j'_1 \rangle \in \text{PI}_n^\ell(\varphi(t|_{\bar{v}_1}))$ such that $\bar{w}''_1 i'_1 \preceq z$. Hence $\langle \bar{w}''_1, i'_1, j'_1 \rangle \in \text{PI}_n^\ell(u|_{\bar{w}'_1})$ and so $\langle \bar{w}'_1 \bar{w}''_1, i'_1, j'_1 \rangle \in \text{PI}_n^\ell(u)$ by Lemma 38(i). Consequently, $\bar{w}_1 \in P_n^\ell(u)$, where $\bar{w}_1 = \bar{w}'_1 \bar{w}''_1$. In addition, $wi' \preceq \bar{w}_1 \prec w_1$ because

$$wi' \preceq z_1 \preceq \bar{w}'_1 \preceq \bar{w}_1 \quad \text{and} \quad \bar{w}_1 \prec \bar{w}'_1 \bar{w}''_1 i'_1 \preceq \bar{w}'_1 z = \hat{w}'_1 \preceq w'_1 \preceq w_1 .$$

In other words, we have shown that $\bar{w}_1 \in \text{br}_u(wi', w_1) \cap P_n^\ell(u)$. Moreover, since $\bar{w}''_1 i'_1 \preceq z$, we have that $\bar{w}''_1 \in \text{pos}_\Delta(\varphi(t|_{\bar{v}_1}))$. Since also $(\bar{v}_1, \bar{w}'_1) \in L$, we can say that \bar{v}_1 is the position in t that creates the symbol $u(\bar{w}_1) = u(\bar{w}'_1 \bar{w}''_1)$ at \bar{w}_1 . Hence, the uniqueness condition in Lemma 36(v) guarantees that for each selection of \bar{v}_1 we obtain a different position $\bar{w}_1 \in \text{br}_u(wi', w_1) \cap P_n^\ell(u)$. This verifies (\dagger) because $w_1 \in \text{br}_u(wi', w_1) \cap P_n^\ell(u)$ and there are at least $\ell^{n+1} - 1$ possible selections of \bar{v}_1 (and each position \bar{w}_1 differs from w_1 because $\bar{w}_1 \prec w_1$ as shown above). \square

The next lemma shows that an inverse linear tree homomorphism reduces the maximal level of the special positions of a tree by at most 1 (for a sufficiently large distance ℓ).

Lemma 40 Let $\psi: T_\Gamma \rightarrow T_\Sigma$ be a linear tree homomorphism. Moreover, let $t \in T_\Gamma$ and $\ell, n \in \mathbb{N}$ be such that $\ell > \text{ht}(\psi(\gamma'))$ for all symbols $\gamma' \in \Gamma$. If there exists $w \in P_{n+1}^\ell(\psi(t))$ with $w \in \text{pos}_\Sigma(\psi(t(\varepsilon)))$, then $\varepsilon \in P_n^\ell(t)$.

PROOF. The proof is similar to the one of Lemma 39. Let $t = \gamma(t_1, \dots, t_k)$ with $k \in \mathbb{N}$, $\gamma \in \Gamma_k$, and $t_1, \dots, t_k \in T_\Gamma$, and let $u = \psi(t) = \varphi(\gamma)[\varphi(t_1), \dots, \varphi(t_k)]$. Moreover, let $\langle w, i', j' \rangle \in \text{PI}_{n+1}^\ell(u)$ with $w \in \text{pos}_\Sigma(\psi(\gamma))$. By the definition of $\text{PI}_{n+1}^\ell(u)$, there exist positions $w_1 \in P_n^\ell(u)$ and $w_2 \in P_n^\ell(u)$ such that $wi' \preceq w_1$ and $wj' \preceq w_2$ and

$$|\text{br}_u(wi', w_1) \cap P_n^\ell(u)| \geq \ell^{n+1} \quad \text{and} \quad |\text{br}_u(wj', w_2) \cap P_n^\ell(u)| \geq \ell^{n+1} .$$

The paths in u from w to w_1 and from w to w_2 contain strictly more than ℓ^{n+1} positions, so they are longer than any path in $\psi(\gamma)$. Together with $w \in \text{pos}_\Sigma(\psi(\gamma))$ we conclude that there must exist $1 \leq i, j \leq k$ and positions $z_1 \in \text{pos}_{x_i}(\psi(\gamma))$ and $z_2 \in \text{pos}_{x_j}(\psi(\gamma))$ such that $wi' \preceq z_1 \preceq w_1$ and $wj' \preceq z_2 \preceq w_2$. Since ψ is linear and $i' \neq j'$, we have $i \neq j$, which yields $\langle \varepsilon, i, j \rangle \in \text{bri}_t$. It remains to prove that $\langle \varepsilon, i, j \rangle \in \text{PI}_n^\ell(t)$, which we prove by induction on n . In the induction base we have $n = 0$ and thus $\langle \varepsilon, i, j \rangle \in \text{bri}_t = \text{PI}_0^\ell(t)$.

We proceed with the induction step. Again, Figure 6 illustrates the used positions and their relations. Clearly, we have $(i, z_1) \in L$. Let v_1 be the position of t_i that creates the symbol $u(w_1)$ at w_1 . More precisely, by Lemma 36(v), there exist unique positions v_1, w'_1, w''_1 such that $i \preceq v_1$, $z_1 \preceq w'_1$, $w_1 = w'_1 w''_1$, $(v_1, w'_1) \in L$, and $w''_1 \in \text{pos}_\Sigma(\psi(t(v_1)))$. Similarly, let $v_2 \in \text{pos}(t)$ be the position that creates the symbol $u(w_2)$ at w_2 . We claim that the property required to prove that $\langle \varepsilon, i, j \rangle \in \text{PI}_n^\ell(t)$, and hence $\varepsilon \in P_n^\ell(t)$, holds for $\text{br}_t(i, v_1)$ and $\text{br}_t(j, v_2)$, i.e.,

$$|\text{br}_t(i, v_1) \cap P_{n-1}^\ell(t)| \geq \ell^n \quad \text{and} \quad |\text{br}_t(j, v_2) \cap P_{n-1}^\ell(t)| \geq \ell^n .$$

We only prove this property for v_1 because the proof for v_2 is analogous. Since $w_1 = w'_1 w''_1 \in P_n^\ell(u)$, it follows from Lemma 38(i) that $w''_1 \in P_n^\ell(u|_{w'_1})$. Moreover, $(v_1, w'_1) \in L$ and Lemma 36(i) yield that $u|_{w'_1} = \psi(t|_{v_1})$ and thus $P_n^\ell(u|_{w'_1}) = P_n^\ell(\psi(t|_{v_1}))$. Together with $w''_1 \in \text{pos}_\Sigma(\psi(t(v_1)))$, we can conclude that $\varepsilon \in P_{n-1}^\ell(t|_{v_1})$ from the induction hypothesis, and hence $v_1 \in P_{n-1}^\ell(t)$ by Lemma 38(i).

Next, we consider any position $\bar{w}_1 \in \text{br}_u(z_1, w_1) \cap P_n^\ell(u)$. We follow the same approach as in the beginning of the induction step. Let \bar{v}_1 be the position of t_i that creates the symbol $u(\bar{w}_1)$ at \bar{w}_1 . More precisely, we apply Lemma 36(v) to \bar{w}_1 to obtain that there exist positions $\bar{v}_1, \bar{w}'_1, \bar{w}''_1$ such that $i \preceq \bar{v}_1$, $z_1 \preceq \bar{w}'_1$, $\bar{w}_1 = \bar{w}'_1 \bar{w}''_1$, $(\bar{v}_1, \bar{w}'_1) \in L$, and $\bar{w}''_1 \in \text{pos}_\Sigma(\psi(t(\bar{v}_1)))$. By the same reasoning as in the previous paragraph, we obtain that $\bar{v}_1 \in P_{n-1}^\ell(t)$. Also, since $\bar{w}_1 \preceq w_1$, we clearly have that $\bar{w}'_1 \preceq w'_1$ because w'_1 (resp. \bar{w}'_1) is the first position on the path from w_1 (resp. \bar{w}_1) to ε that occurs in a link of L . Now note that L is strictly output hierarchical by Proposition 28 because $\langle t, L, u \rangle \in \mathcal{D}(M_\psi)$, where M_ψ is the l-t defined after Notation 10. Hence

$\bar{v}_1 \preceq v_1$ because either $\bar{w}'_1 \prec w'_1$, which directly yields $\bar{v}_1 \preceq v_1$, or $\bar{w}'_1 = w'_1$, which yields $\bar{v}_1 = v_1$ because of the uniqueness of \bar{v}_1 . Thus we have shown that

$$\bar{v}_1 \in \text{br}_t(i, v_1) \cap \text{P}_{n-1}^\ell(t) .$$

If two different selections of \bar{w}_1 correspond to the same position \bar{v}_1 , then (since $(\bar{v}_1, \bar{w}'_1), (v_1, w'_1) \in L$ with $\bar{v}_1 \preceq v_1$ and $\bar{w}'_1 \preceq w'_1$) they also correspond to the same \bar{w}'_1 by the uniqueness condition in Lemma 36(iv), and hence, since $\bar{w}''_1 \in \text{pos}_\Sigma(\psi(t(\bar{v}_1)))$, their distance is at most $\text{ht}(\psi(t(\bar{v}_1))) \leq \ell - 1$. In summary, a single position \bar{v}_1 can create the symbols of at most ℓ positions of $\text{br}_u(z_1, w_1)$. Since there are at most $\ell - 2$ positions between w and z_1 we have

$$|\text{br}_u(z_1, w_1) \cap \text{P}_n^\ell(u)| \geq |\text{br}_u(wi', w_1) \cap \text{P}_n^\ell(u)| - \ell + 2 \geq \ell^{n+1} - \ell + 2 .$$

Consequently, $|\text{br}_t(i, v_1) \cap \text{P}_{n-1}^\ell(t)| \geq \ell^n$ as required since

$$|\text{br}_t(i, v_1) \cap \text{P}_{n-1}^\ell(t)| \leq \ell^n - 1$$

would imply $|\text{br}_u(z_1, w_1) \cap \text{P}_n^\ell(u)| \leq \ell(\ell^n - 1) < \ell^{n+1} - \ell + 2$. This completes the induction step and the proof. \square

Next, we combine the previous two lemmas into the main result of this section that will be used to prove the infinity of several composition hierarchies. We show that a bimorphism in $\mathcal{B}(1, n)$ can reduce the maximal level of the special positions by at most 1 (for a sufficiently large distance ℓ).

Theorem 41 Let $B = (\psi, T, \varphi)$ be a bimorphism such that $\psi: T_\Gamma \rightarrow T_\Sigma$ is linear and $\varphi: T_\Gamma \rightarrow T_\Delta$ is nondeleting. Moreover, let $(s, u) \in \tau(B)$, and let $\ell \in \mathbb{N}$ be such that $\ell > \text{ht}(\psi(\gamma))$ for every $\gamma \in \Gamma$. For every $n \in \mathbb{N}$, if $\text{P}_{n+1}^\ell(s) \neq \emptyset$, then $\text{P}_n^\ell(u) \neq \emptyset$.

PROOF. Since $(s, u) \in \tau(B)$, there exists $t \in T$ such that $\psi(t) = s$ and $\varphi(t) = u$. By assumption, we have that $\text{P}_{n+1}^\ell(\psi(t)) \neq \emptyset$, so let $w \in \text{P}_{n+1}^\ell(\psi(t))$. By Lemma 36(v) there exist v, w', w'' such that $w = w'w''$, $(v, w') \in L_\psi(t)$, and $w'' \in \text{pos}_\Sigma(\psi(t(v)))$. Moreover, $\psi(t)|_{w'} = \psi(t|_v)$ by Lemma 36(i). Since $w'w'' \in \text{P}_{n+1}^\ell(\psi(t))$, Lemma 38(i) implies that

$$w'' \in \text{P}_{n+1}^\ell(\psi(t)|_{w'}) = \text{P}_{n+1}^\ell(\psi(t|_v)) .$$

Hence, by Lemma 40, $\varepsilon \in \text{P}_n^\ell(t|_v)$. Since φ is nondeleting, $\text{pos}_{x_i}(\varphi(t(v)))$ is nonempty for every $1 \leq i \leq \text{rk}(t(v))$. Consequently, Lemma 39 implies that $\text{P}_n^\ell(\varphi(t|_v)) \neq \emptyset$. By Lemma 36(iii) there exists \bar{w} such that $(v, \bar{w}) \in L_\varphi(t)$, and moreover, $\varphi(t|_v) = u|_{\bar{w}}$ by Lemma 36(i). Hence $\text{P}_n^\ell(u|_{\bar{w}}) = \text{P}_n^\ell(\varphi(t|_v)) \neq \emptyset$, which proves that $\text{P}_n^\ell(u) \neq \emptyset$ by Lemma 38(i), as desired. \square

Now we can simply chain Theorem 41 to show that an n -fold composition of tree transformations in $\mathcal{B}(1, n)$ can decrease the maximal level by at most n (for a suitable distance ℓ).

Corollary 42 (of Theorem 41) Let $n \geq 1$, and for every $1 \leq i \leq n$ let $B_i = (\psi_i, T_i, \varphi_i)$ be a bimorphism such that ψ_i is linear and φ_i is nondeleting. Moreover, let $\varphi_i: T_{\Gamma_i} \rightarrow T_{\Delta_i}$ and $\psi_{i+1}: T_{\Gamma_{i+1}} \rightarrow T_{\Delta_i}$ for every $1 \leq i < n$. Finally, let $\ell \in \mathbb{N}$ be such that $\ell > \text{ht}(\psi_i(\gamma))$ for every $1 \leq i \leq n$ and $\gamma \in \Gamma_i$, and let $(t, u) \in \tau(B_1); \dots; \tau(B_n)$. If $P_{n+1}^\ell(t) \neq \emptyset$, then $P_1^\ell(u) \neq \emptyset$.

It remains to demonstrate a tree transformation that can be computed by $n+1$ $\not\leq$ nl-xt and that reduces the maximal level of special positions from $n+1$ to 0. Clearly, this tree transformation cannot be computed by an n -fold composition of tree transformations from $\mathcal{B}(1, n)$ because the output tree should contain a special position of level 1 by Corollary 42. We make sure that the assumptions of Corollary 42 are satisfied.

Example 43 Let $M = (Q, \Sigma, \Sigma, \{\star\}, R)$ be the $\not\leq$ nl-xt with

- $Q = \{\star, q\}$ and $\Sigma = \{\sigma^{(2)}, \alpha^{(0)}\}$, and
- the set R consisting of the following rules

$$\sigma(\star, \alpha) \xrightarrow{\star, q} \star \quad \sigma(\star, q) \xrightarrow{\star, q} \sigma(\star, q) \quad \alpha \xrightarrow{\star} \alpha .$$

It is easy to see that $\tau(M)$ is a total function. Intuitively, for an input tree t , it removes all positions v and $v2$ of t such that $t(v) = \sigma$ and $t(v2) = \alpha$. Figure 7 shows the repeated application of $\tau(M)$, where one application is indicated by \mapsto . Assuming that each dashed line contains at least three more positions, it is easy to check that, for distance $\ell = 2$, the root of the first tree has level 2 (because positions 1, 11, 111, 1111, 2, 21, 211, and 2111 all have level 1). The penultimate tree, which is obtained from the first tree by the application of $\tau(M)^2$, only has special positions of level 0. \square

We use the $\not\leq$ nl-xt M of Example 43, and show that n transformations from $\mathcal{B}(1, n)$ cannot compute the tree transformation $\tau(M)^{n+1}$.

Lemma 44 $\not\leq$ nl-XT $^{n+1} \not\subseteq \mathcal{B}(1, n)^n$ for every $n \geq 1$.

PROOF. Let $\Sigma = \{\sigma^{(2)}, \alpha^{(0)}\}$. The powers of a tree $c \in T_\Sigma(\{x_1\})$ are defined by $c^1 = c$ and $c^{k+1} = c[c^k]$ for every $k \geq 1$. Let $T_{-1} = \{\alpha\}$. For every $n \in \mathbb{N}$, we define the tree languages $C_n \subseteq T_\Sigma(\{x_1\})$ and $T_n \subseteq T_\Sigma$ inductively by $C_n = \{\sigma(x_1, t)^k \mid t \in T_{n-1}, k \geq 1\}$ and $T_n = \{c[\alpha] \mid c \in C_n\}$.

Let M be the $\not\leq$ nl-xt of Example 43. We have already remarked that $\tau(M): T_\Sigma \rightarrow T_\Sigma$ is a total function. It is easy to see that $\tau(M)(t_n) \in T_{n-1}$ for every $n \in \mathbb{N}$ and $t_n \in T_n$. Consequently, $\tau(M)^{n+1}(t_{n+1}) \in T_0$ for every $t_{n+1} \in T_{n+1}$ (see Fig. 7 that shows trees in T_2, T_1, T_0 , and T_{-1}). Obviously, $P_1^\ell(u) = \emptyset$ for every $u \in T_0$ and $\ell \geq 2$. Thus, with the help of Corollary 42, we can complete the proof by showing that for every $\ell \geq 2$ there exists $t \in T_{n+1}$ such that $P_{n+1}^\ell(t) \neq \emptyset$.

Let $\ell \geq 2$ be fixed. We now prove that for every $n \in \mathbb{N}$ there exists $t \in T_n$ such that $P_n^\ell(t) \neq \emptyset$ by induction on n . In fact, we prove the stronger statement that there exists $t \in T_n$ and $v \in P_n^\ell(t)$ such that $|\text{br}_t(\varepsilon, v) \cap P_n^\ell(t)| \geq \ell^{n+1}$. For $n = 0$, we select the tree $t = c^\ell[\alpha] \in T_0$, where $c = \sigma(x_1, \alpha)$, and the

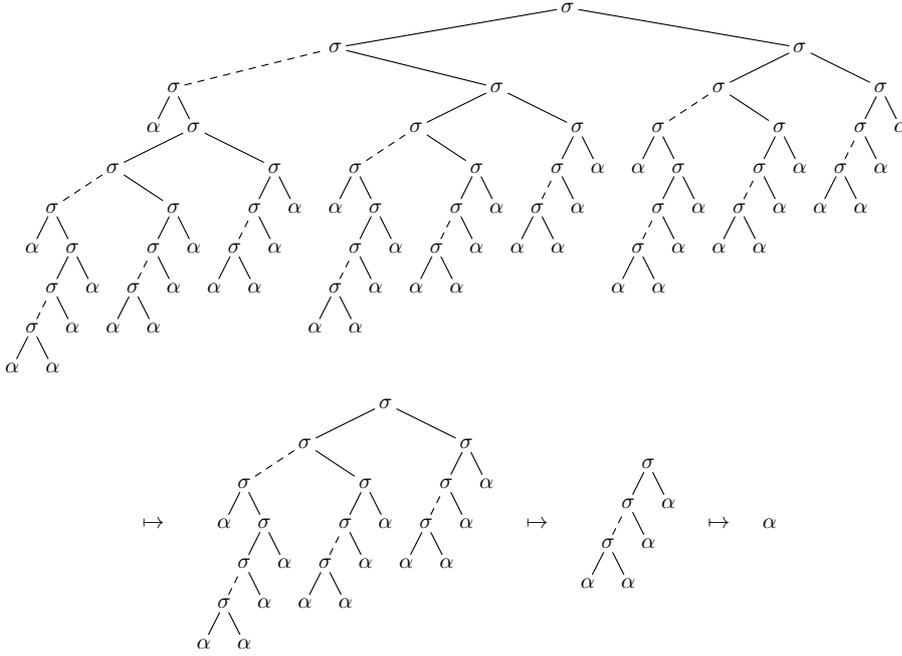


Fig. 7 Illustration of the repeated application of the tree transformation $\tau(M)$ of Example 43.

position $v = 1^{\ell-1}$. Since $P_0^\ell(t) = \text{br}_t(\varepsilon, v)$, this selection of t and v fulfills the requirements. In the induction step, there exist a tree $t \in T_n$ and $v \in P_n^\ell(t)$ such that $|\text{br}_t(\varepsilon, v) \cap P_n^\ell(t)| \geq \ell^{n+1}$. We consider the tree $t' = c^{(\ell^{n+2}+1)}[\alpha]$ with $c = \sigma(x_1, t)$ and the position $v' = 1^{\ell^{n+2}-1}$. Obviously, $t' \in T_{n+1}$ and $v'' \in P_{n+1}^\ell(t')$ for every $v'' \preceq v'$ because $\langle v'', 1, 2 \rangle \in \text{PI}_{n+1}^\ell(t')$ via the positions $v_1 = v''12v$ and $v_2 = v''2v$ using Lemma 38(i). This completes our induction and proof. \square

Now we are able to prove that the composition hierarchy of $\not\leq\text{nl-XT}$ and several other classes is infinite.

Theorem 45 For every $n \geq 1$ and $y \subseteq \{s, n\}$

$$\begin{aligned} \mathcal{B}(l, n)^n \subsetneq \mathcal{B}(l, n)^{n+1} & & y\text{-l-XT}^n \subsetneq y\text{-l-XT}^{n+1} \\ \not\leq\text{nl-XT}^n \subsetneq \not\leq\text{nl-XT}^{n+1} & & (y\text{-l-XT}^R)^n \subsetneq (y\text{-l-XT}^R)^{n+1} . \end{aligned}$$

PROOF. Since all inclusions are trivial, we only need to prove their strictness. By Proposition 11 we have $\not\leq\text{nl-XT} = \mathcal{B}(\text{snl}, \text{nl})$, hence $\not\leq\text{nl-XT}^n \subseteq \mathcal{B}(l, n)^n$ and $\not\leq\text{nl-XT}^{n+1} \subseteq \mathcal{B}(l, n)^{n+1}$. Together with Lemma 44 these two statements imply the strictness of the two inclusions on the left. To prove the strictness of the other two inclusions, we prove that $\text{snl-XT}^{n+1} \not\subseteq (l\text{-XT}^R)^n$. Using simple symmetry, we observe that $\text{snl-XT} = \not\leq\text{nl-XT}^{-1}$, which together with

the symmetric version of Lemma 44 yields $\text{snl-XT}^{n+1} \not\subseteq \mathcal{B}(n,1)^n$. Furthermore, $\text{l-XT}^{\text{R}} = \mathcal{B}(nl,1)$ by Proposition 11, which yields $(\text{l-XT}^{\text{R}})^n \subseteq \mathcal{B}(n,1)^n$. Together with $\text{snl-XT}^{n+1} \not\subseteq \mathcal{B}(n,1)^n$ we obtain $\text{snl-XT}^{n+1} \not\subseteq (\text{l-XT}^{\text{R}})^n$ as desired. \square

For the classes $yl\text{-XT}$ and $yl\text{-XT}^{\text{R}}$ with $y \subseteq \{s\}$ we can make more precise statements, which are similar to those in Theorems 26 and 34.

Theorem 46 For every $n \geq 2$,

$$\begin{aligned} \text{sl-XT} \subsetneq \text{sl-XT}^{\text{R}} \subsetneq \text{sl-XT}^n &= (\text{sl-XT}^{\text{R}})^n \subsetneq \text{sl-XT}^{n+1} \\ \text{l-XT} \subsetneq \text{l-XT}^{\text{R}} \subsetneq \text{l-XT}^n &\subseteq (\text{l-XT}^{\text{R}})^n \subsetneq \text{l-XT}^{n+1} \end{aligned}$$

PROOF. The inclusions from left to right are trivial or follow from Lemma 15. The first strict inclusion on each line follows from Proposition 14. The other strict inclusions follow from $\text{snl-XT}^{m+1} \not\subseteq (\text{l-XT}^{\text{R}})^m$, which was shown in the proof of Theorem 45 for every $m \geq 1$.

It remains to prove that $(\text{sl-XT}^{\text{R}})^n \subseteq \text{sl-XT}^n$. Clearly, it suffices to prove this for $n = 2$. We first observe that $\text{QR} ; \text{snl-XT} \subseteq \text{snl-XT}$. In fact, since $\text{snl-XT}^{-1} = \not\text{nl-XT}$ (as mentioned in the proof of Theorem 45) and, obviously, $\text{QR}^{-1} = \text{QR}$, we obtain that

$$\text{QR} ; \text{snl-XT} = (\not\text{nl-XT} ; \text{QR})^{-1} \subseteq (\not\text{nl-XT})^{-1} = \text{snl-XT} ,$$

where the inclusion follows from Lemma 13. Thus,

$$\begin{aligned} (\text{sl-XT}^{\text{R}})^2 &\subseteq \text{QR} ; \text{sl-XT}^2 \subseteq \text{QR} ; \text{snl-XT} ; \text{sdl-H} ; \text{sl-XT} \\ &\subseteq \text{QR} ; \text{snl-XT} ; \text{sl-XT} \subseteq \text{snl-XT} ; \text{sl-XT} , \end{aligned}$$

where the first step is by Lemma 15, the second step by Lemma 18, the third step by Lemma 19 and the last step by the above observation. \square

The authors do not know whether, but guess that $\text{l-XT}^n \subsetneq (\text{l-XT}^{\text{R}})^n$ for all $n \geq 2$. Table 4 summarizes the main results of this section. For the sake of completeness, we mention some additional results from the literature, where T stands for the class of all tree transformations computable by top-down tree transducers [6], and $\not\text{-XT}$ stands for the class of tree transformations computable by ε -free extended top-down tree transducers [17]. The result $\not\text{-XT} \subseteq \text{T}^2$ mentioned in Table 4 can be concluded from [17, Theorem 4.8].

7 Hasse diagram for the ε -free classes

Finally, let us compare the six classes of Theorem 34 with the three classes of Theorem 26 and the two classes of Proposition 17. Additionally, we consider the composition hierarchy for the class $\not\text{nl-XT}$, for which we established the infiniteness in Theorem 45. Thus, we compare all ε -free classes considered in this paper.

Class with infinite composition hierarchy	Stated in
$\mathcal{B}(l, n)$, $\not\mathcal{A}\text{-XT}$, $y\text{-XT}$, $y\text{-XT}^R$	Theorem 45
T	[8, Theorem 3.14]
$\not\mathcal{A}\text{-XT}$	$\text{T} \subseteq \not\mathcal{A}\text{-XT} \subseteq \text{T}^2$ and ($\text{T}^n \mid n \geq 1$) is infinite
$\mathcal{B}(n, n)$	[1] and [5, Section II-2-2-3-4]

Table 4 Summary of the results of Section 6, where $y \subseteq \{s, n\}$.

Theorem 47 Figure 8 is the HASSE diagram of the displayed classes of tree transformations for all $n \geq 4$.

PROOF. The equalities are proved in Theorems 20 and 34, and all the inclusions are trivial or hold by either Lemma 15 or Corollary 25. The strictness of the vertical inclusions is proven in Proposition 17 and Theorems 26, 34, and 45. For the remaining strictness and incomparability results (with respect to \subseteq) we have to prove the following six results.

- (i) $\not\mathcal{A}\text{-XT}^R \not\subseteq \not\mathcal{A}\text{-XT}$: This is a consequence of Proposition 14.
- (ii) $\not\mathcal{A}\text{-XT}^2 \not\subseteq \not\mathcal{A}\text{-XT}^R$: This follows from Proposition 16. It is also a consequence of the proof of Theorem 31 as follows. Consider the $\not\mathcal{A}\text{-xt}$ M'_1 and the $\not\mathcal{A}\text{-xt}$ M_2 and M_3 in that proof. If $\tau(M_2); \tau(M_3) \in \not\mathcal{A}\text{-XT}^R$, then $\tau(M'_1); \tau(M_2); \tau(M_3) \in (\not\mathcal{A}\text{-XT}^R)^2$, contradicting the proof of Theorem 31.
- (iii) $\not\mathcal{A}\text{-XT} \not\subseteq \not\mathcal{A}\text{-XT}^2$: Let M'_1 , M_2 , and M_3 be as in the proof of Theorem 31. Note that $\tau(M'_1) \in \not\mathcal{A}\text{-XT}$ and $\tau(M_2), \tau(M_3) \in \not\mathcal{A}\text{-XT}$. Now suppose that $\tau(M'_1) \in \not\mathcal{A}\text{-XT}^2$. Then $\tau(M'_1); \tau(M_2); \tau(M_3)$ is in

$$\not\mathcal{A}\text{-XT}^4 = \not\mathcal{A}\text{-XT}^2 \subseteq (\not\mathcal{A}\text{-XT}^R)^2,$$

where the first equality is due to Theorem 20. However, this contradicts the proof of Theorem 31.

- (iv) $\not\mathcal{A}\text{-XT} \not\subseteq \bigcup_{k \geq 1} \not\mathcal{A}\text{-XT}^k$: The translation $\tau = \{(t, \alpha) \mid t \in T_\Sigma\}$ with $\Sigma = \{\sigma^{(2)}, \alpha^{(0)}\}$ can obviously be computed by an $\not\mathcal{A}\text{-xt}$ with the rules $\sigma(q, q') \xrightarrow{q_0} \alpha$ and $\alpha \xrightarrow{q_0} \alpha$, but $\tau \notin \not\mathcal{A}\text{-XT}^k$ for all $k \geq 1$ by Corollary 42 because there exists a tree $t \in T_\Sigma$ such that $P_{k+1}^\ell(t) \neq \emptyset$ as demonstrated in the proof of Lemma 44.
- (v) $\not\mathcal{A}\text{-XT}^3 \not\subseteq (\not\mathcal{A}\text{-XT}^R)^2$: This follows from the proof of Theorem 31 because the $\not\mathcal{A}\text{-xt}$ M'_1 , M_2 , and M_3 in that proof are nondeleting.
- (vi) $\not\mathcal{A}\text{-XT}^4 \not\subseteq \not\mathcal{A}\text{-XT}^3$: This result follows from the proof of Theorem 33, so let M_1 , M_2 , and M_3 be the $\not\mathcal{A}\text{-xt}^R$ of that proof. We note that $\tau(M_2), \tau(M_3) \in \not\mathcal{A}\text{-XT}$. It is easy to show that $\tau(M_1) \in \not\mathcal{A}\text{-XT}^2$, which can be achieved by the decomposition $\tau(M_1) = \tau(N_1); \tau(N_2)$, where N_1 is obtained from M_1 by replacing the two rules involving q^{la} by $\sigma(q, q^{\text{la}}) \xrightarrow{q} \sigma(q, q^{\text{la}})$ and $\sigma(q^{\text{la}}, q) \xrightarrow{q} \sigma(q^{\text{la}}, q)$ and adding the

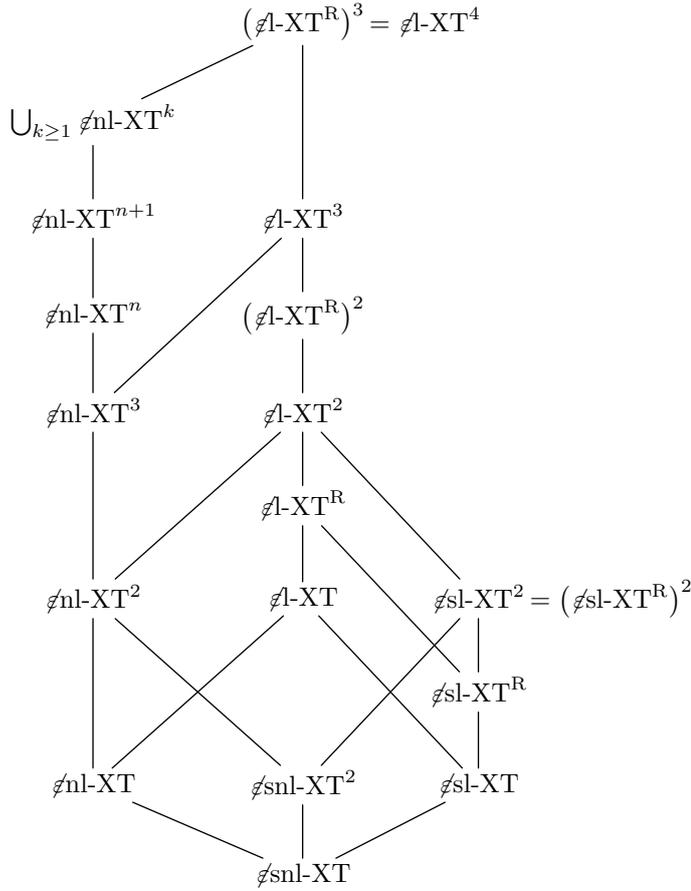


Fig. 8 HASSE diagram of the discussed classes of tree transformations for all $n \geq 4$.

two rules $\gamma_2(q^{la}) \xrightarrow{q^{la}} q^{la}$ and $\alpha \xrightarrow{q^{la}} \alpha$. Then N_1 is nondeleting. Similarly, we obtain N_2 from M_1 by replacing the two rules involving q^{la} by $\sigma(q, \alpha) \xrightarrow{q} q$ and $\sigma(\alpha, q) \xrightarrow{q} q$ (and removing the two rules $\gamma_1(p) \xrightarrow{p} p$ and $\gamma_2(q) \xrightarrow{q} q$ because the symbols γ_1 and γ_2 have already been removed by N_1). Note that also N_2 is nondeleting. The decomposition yields that $\tau(M_1); \tau(M_2); \tau(M_3)$ is in $\phi nl-XT^4$. However, as demonstrated in the proof of Theorem 33, we have that $\tau(M_1); \tau(M_2); \tau(M_3)$ is not in $\phi l-XT^3$. \square

The authors did not attempt to present a HASSE diagram that contains all the classes (including the non- ε -free classes) discussed in this paper, but consider this a worthwhile effort.

Conclusion

Linear extended top-down tree transducers (with or without regular look-ahead) are formal models of syntax-based statistical machine translation. They have several good properties [19]. In particular, most of them can be presented as bimorphisms in the sense of [3], which yields that a result of [3] implies that ε -free, strict, and nondeleting l-xt are not closed under composition and that their composition hierarchy collapses at power 2. We extended their investigation to the composition hierarchy of the classes obtained by dropping some of the restrictions ε -freeness, strictness, and nondeletion. We showed in Theorem 34 that the composition hierarchy of ε -free l-xt^R collapses at power 3 and that of ε -free l-xt collapses at power 4. In fact, the powers 3 and 4 are the least powers with that property. To complete the picture, we showed in Theorem 45 that the composition hierarchies of l-xt, l-xt^R, and ε -free and nondeleting l-xt are infinite. Finally, we presented the HASSE-diagram of the powers of the considered ε -free classes in Theorem 47. In the future, the authors would like to investigate the composition hierarchy of weighted linear extended top-down tree transducers.

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. ACL (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.: Bottom-up and top-down tree transformations — a comparison. Math. Systems Theory **9**(3), 198–231 (1975)
7. Engelfriet, J.: Top-down tree transducers with regular look-ahead. Math. Systems Theory **10**(1), 289–303 (1977)
8. Engelfriet, J.: Three hierarchies of transducers. Math. Systems Theory **15**(2), 95–125 (1982)
9. Engelfriet, J., Maneth, S.: Macro tree translations of linear size increase are MSO definable. SIAM J. Comput. **32**(4), 950–1006 (2003)
10. Engelfriet, J., Schmidt, E.M.: IO and OI I. J. Comput. System Sci. **15**(3), 328–353 (1977)
11. Fülöp, Z., Maletti, A.: Linking theorems for tree transducers (2014). Submitted manuscript; available at: <http://www.inf.u-szeged.hu/~fulop/publ/linking.pdf>
12. Fülöp, Z., Maletti, A., Vogler, H.: Preservation of recognizability for synchronous tree substitution grammars. In: ATANLP, pp. 1–9. ACL (2010)
13. Fülöp, Z., Maletti, A., Vogler, H.: Weighted extended tree transducers. Fundam. Inform. **111**(2), 163–202 (2011)
14. Fülöp, Z., Vogler, H.: Syntax-Directed Semantics — Formal Models Based on Tree Transducers. EATCS Monographs on Theoret. Comput. Sci. Springer (1998)
15. Gécseg, F., Steinby, M.: Tree Automata. Akadémiai Kiadó, Budapest (1984)
16. Gécseg, F., Steinby, M.: Tree languages. In: G. Rozenberg, A. Salomaa (eds.) Handbook of Formal Languages, vol. 3, chap. 1, pp. 1–68. Springer (1997)

17. Graehl, J., Hopkins, M., Knight, K., Maletti, A.: The power of extended top-down tree transducers. *SIAM J. Comput.* **39**(2), 410–430 (2009)
18. Graehl, J., Knight, K., May, J.: Training tree transducers. *Comput. Linguist.* **34**(3), 391–427 (2008)
19. Knight, K., Graehl, J.: An overview of probabilistic tree transducers for natural language processing. In: *CICLing, LNCS*, vol. 3406, pp. 1–24. Springer (2005)
20. Lemay, A., Maneth, S., Niehren, J.: A learning algorithm for top-down XML transformations. In: *PODS*, pp. 285–296. ACM (2010)
21. Maletti, A.: Compositions of extended top-down tree transducers. *Inform. and Comput.* **206**(9–10), 1187–1196 (2008)
22. May, J., Knight, K., Vogler, H.: Efficient inference through cascades of weighted tree transducers. In: *ACL*, pp. 1058–1066. ACL (2010)
23. Rounds, W.C.: Mappings and grammars on trees. *Math. Systems Theory* **4**(3), 257–287 (1970)
24. Thatcher, J.W.: Generalized² sequential machine maps. *J. Comput. System Sci.* **4**(4), 339–367 (1970)