



Composition Closure of Linear Weighted Extended Top-Down Tree Transducers

Zoltán Fülöp¹ and Andreas Maletti²(✉)

¹ Department of Foundations of Computer Science, University of Szeged,
Árpád tér 2, Szeged 6720, Hungary

fulop@inf.u-szeged.hu

² Department of Mathematics and Computer Science, Universität Leipzig,
PO Box 100 920, 04009 Leipzig, Germany

maletti@informatik.uni-leipzig.de

Abstract. Linear weighted extended top-down tree transducers with regular look-ahead and with weights from a semiring are formal models that are used in syntax-based statistical machine translation. The composition hierarchies of some restricted versions of such weighted tree transducers (also without regular look-ahead) are considered. In particular, combinations of the restrictions of ε -freeness (all rules consume input), nondeletion, and strictness (all rules produce output) are considered. The composition hierarchy is shown to be finite for all but one ε -free variant of these weighted transducers over any commutative semiring.

1 Introduction

Linear extended top-down tree transducers (l-xt) were introduced (under a different name) and investigated already in [1]. We present them in the framework of synchronous grammars [3] since in syntax-based statistical machine translation these transducers are applied, and since we utilize some results of [6, 14]. An l-xt M has a finite set of states and finitely many rules of the form $\langle \ell, q, r \rangle$, where q is a state and the left- and right-hand side ℓ and r are trees, which may also contain state-labeled leaves such that each state in r also occurs in ℓ . Linearity requires that each state occurs at most once both in ℓ and in r . In particular, in ε -rules the left-hand side ℓ and in non-strict rules the right-hand side r is just a state. The semantics of M is defined by means of synchronous rewriting using the derivation relation \Rightarrow . It is defined over sentential forms, which are triples (ξ, L, ζ) consisting of trees ξ and ζ with state-labeled leaves and a set L of links. A link is a pair (u, v) of positions pointing to occurrences of the same state in the trees ξ and ζ , respectively. A rule $\langle \ell, q, r \rangle$ can be applied to a sentential form (ξ, L, ζ) if there is a link $(u, v) \in L$ such that u and v point to an occurrence of the state q . In this case we write $(\xi, L, \zeta) \Rightarrow (\xi', L', \zeta')$, where

Z. Fülöp—Supported by NKFI grant K 108 448 and by grant 20391-3/2018/FEKUS-TRAT of the Ministry of Human Capacities, Hungary.

A. Maletti—Partially supported by the DFG research training group 1763 ‘QuantLA’.

© Springer Nature Switzerland AG 2019

M. Hospodár and G. Jirásková (Eds.): CIAA 2019, LNCS 11601, pp. 133–145, 2019.

https://doi.org/10.1007/978-3-030-23679-3_11

the sentential form (ξ', L', ζ') is obtained by replacing the linked occurrences of q in ξ and ζ by ℓ and r , respectively. In addition, L is updated to include links induced by occurrences of the same state in ℓ and r . The initial sentential form is $(q_0, \{(\varepsilon, \varepsilon)\}, q_0)$, in which q_0 is the initial state of M , and we apply derivation steps until no occurrences of linked states remain. Any remaining (unlinked) state occurrence in the input tree t can then be replaced by an arbitrary tree. An instance of t is obtained by replacing all state occurrences and $I(t)$ is the set of all instances of t . The tree transformation induced by M consists of all pairs (t', u) such that $(q_0, \{(\varepsilon, \varepsilon)\}, q_0) \Rightarrow^* (t, \emptyset, u)$ and $t' \in I(t)$. In order to increase their expressive power, l-xt can be equipped with regular look-ahead [4], which restricts the instances $I(t)$ such that an unlinked occurrence of a state q can only be replaced by an element of a given regular tree language $c(q)$. We abbreviate ‘l-xt with regular look-ahead’ by l-xt^{R} .

Weighted l-xt^{R} , abbreviated by l-wxt^{R} , are able to express quantitative properties of the tree transformations [7, 13, 14, 16]. In an l-wxt^{R} a weight from a semiring \mathbb{K} is associated to each rule, and these rule weights are multiplied in a derivation. Provided that several derivations exist, these derivation weights are summed up. In this manner, an l-wxt^{R} M assigns a weight $\|M\|_{\mathbb{K}}(t, u) \in \mathbb{K}$ to each pair (t, u) of trees. It turned out that both l-wxt and l-wxt^{R} over the probabilistic semiring can serve as formal models of tree transformations which are used in syntax-based statistical machine translation [10, 12].

We focus on the composition closure of l-wxt and l-wxt^{R} without ε -rules ($\not\text{l-wxt}$ and $\not\text{l-wxt}^{\text{R}}$, respectively) and some of their restricted subclasses because compositions of weighted tree transformations induced by them can be defined in terms of finite sums. Our motivation is that complex systems are often specified in terms of compositions of simpler tree transformations [17], which are easier to develop, train, and maintain [12]. More precisely, let \mathcal{C} be a class of weighted tree transformations (e.g. the class of all weighted tree transformations induced by $\not\text{l-wxt}^{\text{R}}$). The composition hierarchy of \mathcal{C} is $\mathcal{C} \subseteq \mathcal{C}^2 \subseteq \mathcal{C}^3 \subseteq \dots$, where \mathcal{C}^n denotes the n -fold composition of \mathcal{C} . It is either infinite (i.e., $\mathcal{C}^n \subsetneq \mathcal{C}^{n+1}$ for all n) or finite (i.e., $\mathcal{C}^n = \mathcal{C}^{n+1}$ for some n). In the latter case, the minimal such n is interesting since all compositions can be reduced to this length.

The additional standard restrictions we consider are strictness (‘s’), which requires that the right-hand side r is not a single state, and nondeletion (‘n’), which means that each state in the left-hand side ℓ occurs also in the right-hand side r , in both cases for each rule $\langle \ell, q, r \rangle$ of the $\not\text{l-wxt}^{\text{R}}$. Thus, for instance $\not\text{sl-wxt}^{\text{R}}$ abbreviates the expression ‘strict $\not\text{l-wxt}^{\text{R}}$ ’. The class of all weighted tree transformations induced by certain kind of $\not\text{l-wxt}^{\text{R}}$ is denoted by typesetter letters so for instance $\not\text{sl-WXT}^{\text{R}}(\mathbb{K})$ stands for the set of all weighted tree transformations computable by $\not\text{sl-wxt}^{\text{R}}$ over the semiring \mathbb{K} . We consider the composition hierarchies of the classes $\not\text{ns1-WXT}(\mathbb{K})$, which is also investigated in [15], and $\not\text{sl1-WXT}(\mathbb{K})$, $\not\text{sl1-WXT}^{\text{R}}(\mathbb{K})$, $\not\text{l1-WXT}^{\text{R}}(\mathbb{K})$, and $\not\text{l1-WXT}(\mathbb{K})$. As main result we show that the composition hierarchies of these classes collapse at levels 2, 2, 2, 3, and 4, respectively, for an arbitrary commutative semiring \mathbb{K} (cf. Theorem 16). We achieve our results by lifting the results [1, Theorem 6.2] and

[6, Theorems 26, 31, 33, 34], where it is shown that the corresponding hierarchies in the unweighted cases collapse at the same levels. To this end, we decompose an $\not\exists\text{-l-wxt}^R$ into a weighted relabeling that handles all weights and nondeterminism, and a BOOLEAN functional unambiguous $\not\exists\text{-l-wxt}^R$ (cf. Lemma 12). Moreover, we show that we can compose any such relabeling to the right of any l-wxt^R (cf. Lemma 13). These two constructions together will allow us to take all $\not\exists\text{-l-wxt}^R$ in a composition chain into a particularly simple normal form (cf. Theorem 14). After some additional technical tailoring, we can utilize the mentioned results [1, 6] and lift them to the corresponding weighted devices over any commutative semiring.

2 Preliminaries

We let $\mathbb{N} = \{0, 1, 2, \dots\}$ and $[n] = \{i \in \mathbb{N} \mid 1 \leq i \leq n\}$ for every $n \in \mathbb{N}$. For sets S and T , we let $2^S = \{S' \mid S' \subseteq S\}$, and we identify $S = \{s\}$ with the element s . Moreover, $T^S = \{f \mid f: S \rightarrow T\}$, and $|S|$ is the cardinality of S . For every $R \subseteq S \times T$, we let $\text{dom}(R) = \{s \mid \exists t \in T: (s, t) \in R\}$ and $\text{range}(R) = \{t \mid \exists s \in S: (s, t) \in R\}$. The *composition* of R with $R' \subseteq T \times U$ is $R; R' = \{(s, u) \mid \exists t \in T: (s, t) \in R, (t, u) \in R'\}$. Given $n \in \mathbb{N}$, we let S^n be the n -fold CARTESIAN product of S with itself and $S^* = \bigcup_{n \in \mathbb{N}} S^n$ is the set of all *words* over S . The *length* $|w|$ of $w \in S^n$ is n . The empty word $()$ is also denoted by ε . The *concatenation* of $v, w \in S^*$ is $v.w$ or vw .

A *ranked alphabet* (Σ, rk) consists of a nonempty, finite set Σ and $\text{rk}: \Sigma \rightarrow \mathbb{N}$, which we omit whenever it is obvious. We let $\Sigma^{(n)} = \{\sigma \in \Sigma \mid \text{rk}(\sigma) = n\}$ for every $n \in \mathbb{N}$. In the following, Σ, Δ , and Γ are arbitrary ranked alphabets. Let V be a set with $V \cap \Sigma = \emptyset$. The set $T_\Sigma(V)$ of Σ^m *trees indexed by* V is defined in the usual way, and we let $T_\Sigma = T_\Sigma(\emptyset)$. If $t \in T_\Sigma(V)$ is given as $t = \sigma(t_1, \dots, t_n)$, then we often omit the obvious quantification that $n \in \mathbb{N}$, $\sigma \in \Sigma^{(n)}$, and $t_1, \dots, t_n \in T_\Sigma(V)$. The map $\text{pos}: T_\Sigma(V) \rightarrow 2^{\mathbb{N}^*}$ assigning *positions* is inductively defined for all $t \in T_\Sigma(V)$ by $\text{pos}(t) = \{\varepsilon\}$ if $t \in V$ and $\text{pos}(t) = \{\varepsilon\} \cup \{i.w \mid i \in [n], w \in \text{pos}(t_i)\}$ if $t = \sigma(t_1, \dots, t_n)$. The *size* $|t|$ of a tree $t \in T_\Sigma(V)$ is $|\text{pos}(t)|$. The *label* of t at $w \in \text{pos}(t)$ is $t(w)$ and the *subtree* of t *rooted at* w is $t|_w$. For every set $D \subseteq \Sigma \cup V$ of labels, we let $\text{pos}_D(t) = \{w \in \text{pos}(t) \mid t(w) \in D\}$. The tree t is *linear* (resp. *nondeleting*) in $V' \subseteq V$ if $|\text{pos}_v(t)| \leq 1$ (resp., $|\text{pos}_v(t)| \geq 1$) for every $v \in V'$. We let $T_\Sigma^{\text{lin}}(V)$ be the set of all trees of $T_\Sigma(V)$ that are linear in V . Moreover, $\text{var}(t) = \{v \in V \mid \text{pos}_v(t) \neq \emptyset\}$. We use the countably infinite set $X = \{x_i \mid i \in \mathbb{N}\}$ and $X_n = \{x_i \mid i \in [n]\}$ for every $n \in \mathbb{N}$. For every $n \in \mathbb{N}$, we define the set $C_\Sigma(X_n) = \{t \in T_\Sigma^{\text{lin}}(X_n) \mid \text{var}(t) = X_n\}$ of n -*contexts* and the set $\widehat{C}_\Sigma(X_n) = \{t \in C_\Sigma(X_n) \mid \text{the order of variables in } t \text{ is } x_1, \dots, x_n\}$ of *straight* n -*contexts*. Let $X' \subseteq X$ and $\theta: X' \rightarrow T_\Sigma$. Each such mapping θ extends to a mapping $(\cdot)\theta: T_\Sigma(X) \rightarrow T_\Sigma(X)$ such that for all $t \in T_\Sigma(X)$ we have $t\theta = t$ if $t \in X \setminus X'$, $t\theta = \theta(t)$ if $t \in X'$, and $t\theta = \sigma(t_1\theta, \dots, t_n\theta)$ if $t = \sigma(t_1, \dots, t_n)$. Given $t \in T_\Sigma(X)$ and $t_1, \dots, t_n \in T_\Sigma$, we write $t[t_1, \dots, t_n]$ for $t\theta$, where $\theta: X_n \rightarrow T_\Sigma$ is given by $\theta(x_i) = t_i$ for every $i \in [n]$. Moreover, for every $t \in T_\Sigma$ and $k \in \mathbb{N}$, let

$$\text{decomp}(t) = \bigcup_{n \in \mathbb{N}} \{(c, t_1, \dots, t_n) \in \widehat{C}_\Sigma(X_n) \times (T_\Sigma)^n \mid t = c[t_1, \dots, t_n]\}$$

and $\text{subst}_k(t) = \{(u, \theta) \in T_\Sigma^{\text{lin}}(X_k) \times (T_\Sigma)^{\text{var}(u)} \mid t = u\theta\}$. Note that both $\text{decomp}(t)$ and $\text{subst}_k(t)$ are finite sets.

As weight structures we use commutative semirings [9, 11]. Formally, a *commutative semiring* is an algebraic structure $\mathbb{K} = (K, +, \cdot, 0, 1)$ such that $(K, +, 0)$ and $(K, \cdot, 1)$ are commutative monoids, \cdot distributes over finite sums, and for all $a \in K$ we have $a \cdot 0 = 0$. Given a mapping $f: S \rightarrow K$, it is **BOOLEAN** if $\text{range}(f) \subseteq \{0, 1\}$, and its *support* $\text{supp}(f)$ is $\text{supp}(f) = \{s \in S \mid f(s) \neq 0\}$. Any map $L: T_\Sigma \rightarrow K$ is a *weighted tree language*.

We will often utilize the **BOOLEAN** semiring $\mathbb{B} = (\{0, 1\}, \vee, \wedge, 0, 1)$, which is used to model the unweighted case, and the semiring $\mathbb{N} = (\mathbb{N}, +, \cdot, 0, 1)$. For the rest of this contribution, let $\mathbb{K} = (K, +, \cdot, 0, 1)$ be a commutative semiring.

A *weighted tree automaton* is a tuple $A = (Q, \Sigma, Q_0, \text{wt})$, in which Q is a finite set of states, and $Q_0 \subseteq Q$ is a set of initial states, Σ is a ranked alphabet of input symbols, and $\text{wt}: (\bigcup_{n \in \mathbb{N}} Q^n \times \Sigma^{(n)} \times Q) \rightarrow K$ is a weight assignment to transitions. It is **BOOLEAN** if the weight assignment ‘wt’ is **BOOLEAN**, and it is (*bottom-up deterministic*) if $q = q'$ for all $(q_1, \dots, q_n, \sigma, q), (q_1, \dots, q_n, \sigma, q') \in \text{supp}(\text{wt})$. The semantics $\|A\|_{\mathbb{K}}: T_\Sigma \rightarrow K$ of A is $\|A\|_{\mathbb{K}}(t) = \sum_{q \in Q_0} \|A\|_{\mathbb{K}}^q(t)$ for every $t \in T_\Sigma$, where the map $\|A\|_{\mathbb{K}}^q: T_\Sigma \rightarrow K$ is inductively defined for every $q \in Q$ and tree $t = \sigma(t_1, \dots, t_n)$ by $\|A\|_{\mathbb{K}}^q(t) = \sum_{q_1, \dots, q_n \in Q} \text{wt}(q_1, \dots, q_n, \sigma, q) \cdot \prod_{i=1}^n \|A\|_{\mathbb{K}}^{q_i}(t_i)$. A weighted tree language $L: T_\Sigma \rightarrow K$ is (\mathbb{K} -)regular if there exists a weighted tree automaton A such that $L = \|A\|_{\mathbb{K}}$. The class of all such regular weighted tree languages is denoted by $\text{REG}_\Sigma(\mathbb{K})$. For a **BOOLEAN** deterministic weighted tree automaton A the weighted tree languages $\|A\|_{\mathbb{K}}$ and $\|A\|_{\mathbb{K}}^q$ for all $q \in Q$ are obviously **BOOLEAN**. The regular weighted tree languages are closed under the weighted relabelings [7, Theorem 5.3], which we introduce next.

A weighted tree transformation is a mapping $\tau: T_\Sigma(V) \times T_\Delta(V) \rightarrow K$. The domain ‘dom(τ)’ and range ‘range(τ)’ for a weighted tree transformation $\tau: T_\Sigma(V) \times T_\Delta(V) \rightarrow K$ are simply defined by $\text{dom}(\tau) = \text{dom}(\text{supp}(\tau))$ and $\text{range}(\tau) = \text{range}(\text{supp}(\tau))$. The transformation τ is (*functional* (resp., *finitary*)), if $\{u \mid (t, u) \in \text{supp}(\tau)\}$ contains at most one element (resp., is finite) for every $t \in T_\Sigma(V)$. For a functional τ , its support $\text{supp}(\tau)$ is a partial function.

A *weighted relabeling* is a mapping $\nu: \Sigma \times \Delta \rightarrow K$ such that $\nu(\sigma, \delta) = 0$ for all $\sigma \in \Sigma$ and $\delta \in \Delta$ with $\text{rk}(\sigma) \neq \text{rk}(\delta)$. Each weighted relabeling $\nu: \Sigma \times \Delta \rightarrow K$ extends to a finitary weighted tree transformation $\bar{\nu}: T_\Sigma(V) \times T_\Delta(V) \rightarrow K$, which is given as follows: for all variables $v, v' \in V$, trees $t = \sigma(t_1, \dots, t_n) \in T_\Sigma(V)$, and $u = \delta(u_1, \dots, u_k) \in T_\Delta(V)$, we define $\bar{\nu}(v, u) = \bar{\nu}(t, v') = 0$ and

$$\bar{\nu}(v, v') = \begin{cases} 1, & \text{if } v = v'; \\ 0, & \text{otherwise,} \end{cases} \quad \bar{\nu}(t, u) = \begin{cases} \nu(\sigma, \delta) \cdot \prod_{i=1}^n \bar{\nu}(t_i, u_i), & \text{if } n = k; \\ 0, & \text{otherwise.} \end{cases}$$

Note that the weighted tree transformation $\bar{\nu}$ is finitary. Since $\bar{\nu}$ and ν coincide on $\Sigma^{(0)} \times \Delta^{(0)}$, we will not distinguish carefully between them and use just ν

for both. In fact, for all $t \in T_\Sigma(V)$, $u \in T_\Delta(V)$, and each $(c', u_1, \dots, u_n) \in \text{decomp}(u)$

$$\nu(t, u) = \sum_{(c, t_1, \dots, t_n) \in \text{decomp}(t)} \nu(c, c') \cdot \prod_{i=1}^n \nu(t_i, u_i) . \quad (1)$$

There is actually at most one decomposition of t in (1) that yields a non-zero weight for the sum (since the shapes of c and c' and similarly t_i and u_i need to coincide for all $i \in [n]$). The analogous property holds provided that a decomposition of t is given. The class of all weighted tree transformations induced by weighted relabelings is denoted by $\text{WREL}(\mathbb{K})$.

Given a finitary weighted tree transformation $\tau: T_\Sigma \times T_\Delta \rightarrow K$ and a weighted tree language $L: T_\Delta \rightarrow K$, we can define the pre-image $\tau^{-1}(L): T_\Sigma \rightarrow K$ of L via τ for every $t \in T_\Sigma$ by $(\tau^{-1}(L))(t) = \sum_{u \in T_\Delta} \tau(t, u) \cdot L(u)$. Given another weighted tree transformation $\tau': T_\Delta \times T_\Gamma \rightarrow K$, we define the *composition* $\tau; \tau': T_\Sigma \times T_\Gamma \rightarrow K$ of τ followed by τ' for every $t \in T_\Sigma$ and $v \in T_\Gamma$ as $(\tau; \tau')(t, v) = \sum_{u \in T_\Delta} \tau(t, u) \cdot \tau'(u, v)$. Given classes $\mathcal{C}, \mathcal{C}'$ of tree transformations, we let $\mathcal{C}; \mathcal{C}' = \{\tau; \tau' \mid \tau \in \mathcal{C}, \tau' \in \mathcal{C}'\}$. We also write $\tau; L$ instead of $\tau^{-1}(L)$ for a weighted relabeling τ and a weighted tree language $L: T_\Delta \rightarrow K$.

Theorem 1 ([7, Theorem 5.1]). For every weighted relabeling $\tau \in \text{WREL}(\mathbb{K})$ of type $\tau: T_\Sigma \times T_\Delta \rightarrow K$ and regular weighted tree language $L \in \text{REG}_\Delta(\mathbb{K})$, the weighted tree language $\tau^{-1}(L)$ is again regular [i.e., $\tau^{-1}(L) \in \text{REG}_\Sigma(\mathbb{K})$].

3 Transformational Model

A *linear weighted extended top-down tree transducer with regular look-ahead* (for short: l-wxt^{R}) over \mathbb{K} [14] is a tuple $M = (Q, \Sigma, \Delta, q_0, R, \text{wt}, c)$, in which

- Q is a finite set of *states*, and $q_0 \in Q$ is an *initial state*,
- Σ and Δ are ranked alphabets of *input* and *output symbols*, respectively,
- $R \subseteq T_\Sigma^{\text{lin}}(Q) \times Q \times T_\Delta^{\text{lin}}(Q)$ is a finite set of *rules* such that $\text{var}(r) \subseteq \text{var}(\ell)$ and $\{\ell, r\} \not\subseteq Q$ for every rule $\langle \ell, q, r \rangle \in R$,
- $\text{wt}: R \rightarrow K$ is a *weight assignment* to the rules, and
- $c: Q \rightarrow \text{REG}_\Sigma(\mathbb{K})$ is a regular weighted *look-ahead* for each state.

To save parentheses, we will write c^q instead of $c(q)$ for every state $q \in Q$.

Next, we recall some common restrictions of the general model that have already been discussed in [7]. The l-wxt^{R} M is

- ε -free (resp., *strict*), if $\ell \notin Q$ (resp., $r \notin Q$) for every rule $\langle \ell, q, r \rangle \in R$,
- *nondeleting*, if $\text{var}(\ell) = \text{var}(r)$ for every rule $\langle \ell, q, r \rangle \in R$,
- **BOOLEAN**, if ‘wt’ and c^q are **BOOLEAN** for every state $q \in Q$,
- an *l-wxt* (i.e., without look-ahead) if $c^q(t) = 1$ for every state $q \in Q$ and tree $t \in T_\Sigma$, and
- an l-wt^{R} (i.e., non-extended) if $\text{pos}_\Sigma(\ell) = \{\varepsilon\}$ for every rule $\langle \ell, q, r \rangle \in R$.

Next we recall the semantics of an l-wxt^R $M = (Q, \Sigma, \Delta, q_0, R, \text{wt}, c)$, which is the weighted tree transformation $\|M\|_{\mathbb{K}}^q: T_{\Sigma} \times T_{\Delta} \rightarrow K$ defined inductively for every $t \in T_{\Sigma}$ and $u \in T_{\Delta}$ by

$$\|M\|_{\mathbb{K}}^q(t, u) = \sum_{\substack{(\ell, t_1, \dots, t_n) \in \text{decomp}(t) \\ (r, \theta) \in \text{subst}_n(u) \\ \rho = \langle \ell[q_1, \dots, q_n], q, r[q_1, \dots, q_n] \rangle \in R}} \text{wt}(\rho) \cdot \left(\prod_{x_i \in \text{var}(r)} \|M\|_{\mathbb{K}}^{q_i}(t_i, x_i\theta) \right) \cdot \left(\prod_{x_i \in \text{var}(r) \setminus \text{var}(r)} c^{q_i}(t_i) \right). \quad (2)$$

Using our remarks about $\text{decomp}(t)$ and $\text{subst}_n(u)$, all sets used in the index of the sum are finite, so the sum has only finitely many summands. Since we have $\langle \ell[q_1, \dots, q_n], q, r[q_1, \dots, q_n] \rangle \in R$, we know that $\ell \notin Q$ or $r \notin Q$. Consequently, $|t_i| < |t|$ or $|x_i\theta| < |u|$ for every $i \in [n]$ with $x_i \in \text{var}(r)$, which proves that the recursion is well-founded. Besides the rule weight $\text{wt}(\rho)$, we multiply the weights $\|M\|_{\mathbb{K}}^{q_i}(t_i, x_i\theta)$ of the recursive processing of those subtrees t_i that are further processed. The subtrees that are not further processed contribute their look-ahead weight $c^{q_i}(t_i)$. The semantics $\|M\|_{\mathbb{K}}: T_{\Sigma} \times T_{\Delta} \rightarrow K$ is then given for every $t \in T_{\Sigma}$ and $u \in T_{\Delta}$ by $\|M\|_{\mathbb{K}}(t, u) = \|M\|_{\mathbb{K}}^{q_0}(t, u)$. For nondeleting l-wxt^R the rightmost product in (2) yields 1, hence nondeleting l-wxt^R and nondeleting l-wxt are equally expressive. An l-wxt^R M is *functional* if $\|M\|_{\mathbb{K}}$ is functional. We note that each l-wxt^R M that is ε -free or functional induces a finitary $\|M\|_{\mathbb{K}}$. Next we relate l-wxt^R over the semiring \mathbb{N} and l-wxt^R over \mathbb{K} . For this, we recall that \mathbb{N} is the initial commutative semiring, so there exists a unique homomorphism [9, 11] from \mathbb{N} to \mathbb{K} , i.e., a mapping $h: \mathbb{N} \rightarrow K$ with $h(0) = 0$, $h(1) = 1$, $h(n + n') = h(n) + h(n')$, and $h(n \cdot n') = h(n) \cdot h(n')$ for all $n, n' \in \mathbb{N}$.

Lemma 2. Let $M = (Q, \Sigma, \Delta, q_0, R, \text{wt}, c)$ be an l-wxt^R over the semiring \mathbb{N} of nonnegative integers, and let $h: \mathbb{N} \rightarrow K$ be the unique semiring homomorphism from \mathbb{N} to \mathbb{K} . Then $h(\|M\|_{\mathbb{N}}(t, u)) = \|M'\|_{\mathbb{K}}(t, u)$ for all $(t, u) \in T_{\Sigma} \times T_{\Delta}$, where $M' = (Q, \Sigma, \Delta, q_0, R, \text{wt}', c')$ is the l-wxt^R over the semiring \mathbb{K} with $\text{wt}'(\rho) = h(\text{wt}(\rho))$ and $(c')^q(t) = h(c^q(t))$ for all $\rho \in R$, $q \in Q$, and $t \in T_{\Sigma}$.

We abbreviate the properties of l-wxt^R as follows: ‘ ε ’ = ε -free, ‘s’ = strict, ‘n’ = nondeleting, ‘B’ = BOOLEAN, ‘f’ = functional. We use these shorthands with the stems ‘l-wxt^R’, ‘l-wxt’, ‘l-wt^R’, and ‘l-wt’ to talk about an l-wxt^R, l-wxt, l-wt^R, or l-wt that additionally has the abbreviated properties attached as a prefix. For example, Bnl-wxt stands for “BOOLEAN nondeleting l-wxt”. We use the same abbreviations with the stem (i.e., the material behind the hyphen) in typesetter letters (and the semiring \mathbb{K} in parentheses) for the corresponding classes of induced weighted tree transformations. For instance, nl-WXT(\mathbb{K}) stands for the set of all weighted tree transformations computable by nl-wxt.

Utilizing the bimorphism characterizations of [7, Section 4] and the closure of the regular tree languages under linear homomorphisms [8, Theorem II.4.16], we easily obtain that both the domain as well as the range of each such transducer are regular, which we utilize without explicit mention.

Lemma 3. For every $\tau \in \text{1-WXT}^{\text{R}}(\mathbb{B})$ both $\text{dom}(\tau)$ and $\text{range}(\tau)$ are regular.

Finally, we recall the results for the composition hierarchies of the unweighted tree transformation classes, which we lift to the weighted case in Sect. 5.

Theorem 4 (see [1, Theorem 6.2] and [6, Theorems 26, 31, 33, 34]).

$$\not\in\text{ns1-WXT}(\mathbb{B})^3 = \not\in\text{ns1-WXT}(\mathbb{B})^2 \quad (3a)$$

$$\not\in\text{s1-WXT}(\mathbb{B})^3 = \not\in\text{s1-WXT}(\mathbb{B})^2 \quad (3b) \quad \not\in\text{s1-WXT}^{\text{R}}(\mathbb{B})^3 = \not\in\text{s1-WXT}^{\text{R}}(\mathbb{B})^2 \quad (3d)$$

$$\not\in\text{l-WXT}(\mathbb{B})^5 = \not\in\text{l-WXT}(\mathbb{B})^4 \quad (3c) \quad \not\in\text{l-WXT}^{\text{R}}(\mathbb{B})^4 = \not\in\text{l-WXT}^{\text{R}}(\mathbb{B})^3 \quad (3e)$$

Additionally, the classes of (3b) and (3d) as well as (3c) and (3e) coincide.

4 Faithful Representation

In this section, we deal with the question in which cases unweighted transducers faithfully represent certain BOOLEAN weighted transducers (note that the weighted tree transformation induced by a BOOLEAN l-wxt^{R} might not be BOOLEAN). Moreover, we consider for which such transducers M another such transducer M' exists, which induces a partial function $\|M'\|_{\mathbb{B}} \subseteq \|M\|_{\mathbb{B}}$ with the same domain as $\|M\|_{\mathbb{B}}$. We start with the definition of unambiguous l-wxt^{R} . To this end, we reinterpret BOOLEAN l-wxt^{R} over the semiring \mathbb{K} , which anyway only use the neutral elements 0 and 1, as BOOLEAN l-wxt^{R} over the semiring \mathbb{N} of nonnegative integers by identifying the neutral elements 0 and 1 in \mathbb{K} and \mathbb{N} . Thus, given a BOOLEAN l-wxt^{R} M (over \mathbb{K}), we write $\|M\|_{\mathbb{N}}$ for its semantics in the semiring of nonnegative integers. We also reinterpret M over the BOOLEAN semiring \mathbb{B} and write $\|M\|_{\mathbb{B}}$. Over the BOOLEAN semiring \mathbb{B} , we sometimes identify mappings $f: S \rightarrow \{0, 1\}$ with their support $\text{supp}(f)$ and vice versa, so $\|M\|_{\mathbb{B}}: T_{\Sigma} \times T_{\Delta} \rightarrow \{0, 1\}$ is identified with $\|M\|_{\mathbb{B}} \subseteq T_{\Sigma} \times T_{\Delta}$ for a BOOLEAN l-wxt^{R} M .

Definition 5. Let $M = (Q, \Sigma, \Delta, q_0, R, \text{wt}, c)$ be a BOOLEAN l-wxt^{R} over \mathbb{K} and $L \subseteq T_{\Sigma}$. We say that M is *unambiguous on L* if $\|M\|_{\mathbb{N}}(t, u) \in \{0, 1\}$ for every $(t, u) \in L \times T_{\Delta}$ (i.e., the mapping $\|M\|_{\mathbb{N}}$ restricted to $L \times T_{\Delta}$ is BOOLEAN). In the case of $L = T_{\Sigma}$, we also say that M is *unambiguous*.

Lemma 6. Let $M = (Q, \Sigma, \Delta, q_0, R, \text{wt}, c)$ be a BOOLEAN l-wxt^{R} over \mathbb{K} that is unambiguous on $L \subseteq T_{\Sigma}$. Then $\|M\|_{\mathbb{K}}(t, u) = \|M\|_{\mathbb{B}}(t, u)$ for all $(t, u) \in L \times T_{\Delta}$.

Next we consider how BOOLEAN weighted tree transformations behave under composition. We will identify another restriction, functionality, that is required for the faithful representation via unweighted composition.

Lemma 7. Let $\tau: T_{\Sigma} \times T_{\Delta} \rightarrow K$ and $\tau': T_{\Delta} \times T_{\Gamma} \rightarrow K$ be BOOLEAN weighted tree transformations. If τ is functional, then $\tau; \tau' = \text{supp}(\tau); \text{supp}(\tau')$.

With this knowledge we are now ready to state the main lemma of this section. Given BOOLEAN transducers that additionally obey certain functionality and unambiguity restrictions, the computation inside the BOOLEAN semiring faithfully represents the computation in the semiring \mathbb{K} . We recall that the neutral element of composition is the identity mapping $\text{id}_{T_\Sigma} = \{(t, t) \mid t \in T_\Sigma\}$ of the correct set T_Σ , whose range is clearly T_Σ .

Lemma 8. Let $n \geq 1$ be an integer and $M_i = (Q_i, \Sigma_i, \Sigma_{i+1}, q_i^0, R_i, \text{wt}_i, c_i)$ be a BOOLEAN l-wxt^R for every $i \in [n]$. If (i) M_i is unambiguous on

$$\text{range}(\|M_1\|_{\mathbb{K}}; \cdots; \|M_{i-1}\|_{\mathbb{K}})$$

and (ii) $\|M_1\|_{\mathbb{K}}; \cdots; \|M_{i-1}\|_{\mathbb{K}}$ is functional for every $i \in [n]$, then

$$\|M_1\|_{\mathbb{K}}; \cdots; \|M_n\|_{\mathbb{K}} = \|M_1\|_{\mathbb{B}}; \cdots; \|M_n\|_{\mathbb{B}} .$$

We identify a weighted tree transformation over \mathbb{B} with its support for the rest of this section. A *uniformizer* [2] of a tree transformation $\tau \subseteq T_\Sigma \times T_\Delta$ is a partial function $f: T_\Sigma \rightarrow T_\Delta$ such that $f \subseteq \tau$ and $\text{dom}(f) = \text{dom}(\tau)$. In other words, a uniformizer of τ is a maximal partial function contained in τ . We start with a simple proposition that shows how uniformizers behave under composition.

Lemma 9 ([2, Lemma 24]). Let $n \geq 1$ be an integer, $\Sigma_1, \dots, \Sigma_{n+1}$ be ranked alphabets, and $\tau_i \subseteq T_{\Sigma_i} \times T_{\Sigma_{i+1}}$ and $f_i: T_{\Sigma_i} \rightarrow T_{\Sigma_{i+1}}$ be tree transformations for all $i \in [n]$. If

1. $\text{range}(\tau_j) \subseteq \text{dom}(\tau_{j+1})$ for all $1 \leq j \leq n-1$ and
2. f_i is a uniformizer for τ_i for all $i \in [n]$,

then $f = f_1; \cdots; f_n$ is a uniformizer for $\tau = \tau_1; \cdots; \tau_n$ and $\text{dom}(\tau) = \text{dom}(\tau_1)$. If additionally τ is functional, then $f = \tau$.

Finally, we need two results from the theory of unweighted tree transducers. The first statement establishes the existence of uniformizers of tree transformations induced by $\not\in$ l-wxt^R over \mathbb{B} .

Lemma 10 (variant of [5, Lemma]). For every $w \subseteq \{\mathbf{n}, \mathbf{s}\}$, each tree transformation of $w\not\in$ l-WXT^R(\mathbb{B}) has a uniformizer in $w\not\in$ f1-WXT^R(\mathbb{B}).

The second statement builds also on [5, Lemma], which essentially says that functional top-down tree transducers can be determinized provided that they have regular look-ahead. We utilize the same idea to prove a corresponding lemma, where we use unambiguity instead of determinism. The lemma shows that we can remove ambiguity from a functional l-wxt^R over \mathbb{B} . We use the shorthand ‘u’ to abbreviate ‘unambiguous’.

Lemma 11. $w\text{f1-WXT}^{\text{R}}(\mathbb{B}) = w\text{ful-WXT}^{\text{R}}(\mathbb{B})$ for all $w \subseteq \{\not\in, \mathbf{n}, \mathbf{s}\}$.

5 Main Results

We start with a construction that shows that a weighted relabeling can be separated from an ε -free l-wxt^R that handles all the weights and the nondeterminism leaving a BOOLEAN functional ε -free l-wxt^R that is also unambiguous.

Lemma 12. For all $w \subseteq \{\mathbf{n}, \mathbf{s}\}$

$$\begin{aligned} w \not\subseteq \mathbf{1}\text{-WXT}^{\mathbf{R}}(\mathbb{K}) &\subseteq \text{WREL}(\mathbb{K}) ; w \not\subseteq \mathbf{Bful}\text{-WXT}^{\mathbf{R}}(\mathbb{K}) \\ w \not\subseteq \mathbf{1}\text{-WXT}(\mathbb{K}) &\subseteq \text{WREL}(\mathbb{K}) ; w \not\subseteq \mathbf{Bful}\text{-WXT}(\mathbb{K}) . \end{aligned}$$

Proof. Let $M = (Q, \Sigma, \Delta, q_0, R, \text{wt}, c)$ be an arbitrary ε -free l-wxt^R. Since we also need access to the transitions of a weighted tree automaton computing the regular look-ahead, let $A = (Q', \Sigma, Q'_0, \underline{\text{wt}})$ be a weighted tree automaton such that $Q \subseteq Q'$ and $\|A\|_{\mathbb{K}}^q = c^q$ for every $q \in Q$ (e.g., we can take the disjoint union of the weighted tree automata computing c^q for all $q \in Q$). Let $P = R \cup \text{supp}(\underline{\text{wt}})$ be the set of all rules and transitions used in M and A . We first construct the ranked alphabet (Σ', rk') consisting of the symbols $\Sigma' = \Sigma \cup (\Sigma \times P)$ such that $\text{rk}'(\sigma) = \text{rk}(\sigma)$ and $\text{rk}'(\langle \sigma, \rho \rangle) = \text{rk}(\sigma)$ for every $\sigma \in \Sigma$ and $\rho \in P$. Next, we construct the weighted relabeling $\nu: \Sigma \times \Sigma' \rightarrow K$ as follows: $\nu(\sigma, \sigma') = \delta_{\sigma\sigma'}$ and

$$\nu(\sigma, \langle \sigma', \rho \rangle) = \begin{cases} \delta_{\sigma\sigma'} \cdot \text{wt}(\rho), & \text{if } \rho \in R; \\ \delta_{\sigma\sigma'} \cdot \underline{\text{wt}}(\rho), & \text{if } \rho \in \text{supp}(\underline{\text{wt}}) \end{cases}$$

for all $\sigma, \sigma' \in \Sigma$ and $\rho \in P$, where $\delta_{\sigma\sigma'}$ is the usual KRONECKER delta. In other words, the relabeling either (i) keeps the symbol or (ii) keeps the input symbol, but annotates a rule or transition and charges the corresponding weight. Intuitively, the relabeling annotates the rules and transitions to be executed, but the relabeling does not ensure that the annotation can actually be executed at the annotated position. This check and the execution are performed by the BOOLEAN ε -free l-wxt^R $M' = (Q, \Sigma', \Delta, q_0, R', \text{wt}', \underline{c})$, to which the BOOLEAN weighted tree automaton $A' = (Q', \Sigma', Q'_0, \underline{\text{wt}}')$ is associated via $\underline{c}^q = \|A'\|_{\mathbb{K}}^q$ for every state $q \in Q$. We set

- $\rho' = \langle \langle \ell(\varepsilon), \rho \rangle (\ell|_1, \dots, \ell|_{\text{rk}(\sigma)}), q, r \rangle \in R'$ and $\text{wt}'(\rho') = 1$ for every rule $\rho = \langle \ell, q, r \rangle \in R$,
- $\underline{\text{wt}}'(q'_1, \dots, q'_n, \langle \sigma, \rho \rangle, q') = 1$ for every $\rho = (q'_1, \dots, q'_n, \sigma, q') \in \text{supp}(\underline{\text{wt}})$, and
- no additional rules are in R' and $\underline{\text{wt}}'(\rho) = 0$ for all other transitions ρ .

Note that $\ell(\varepsilon) \in \Sigma$ in the first item because M is ε -free. Hence M' is ε -free. Moreover, A' is deterministic and both M' and A' are BOOLEAN because $\underline{c}^q = \|A'\|_{\mathbb{K}}^q$ is also BOOLEAN. So the constructed l-wxt^R M' is BOOLEAN and ε -free. Moreover, it inherits the properties ‘nondeleting’ and ‘strict’ from M . If M has trivial look-ahead c , then we set \underline{c} to the trivial look-ahead for $T_{\Sigma'}$ for the statements on l-wxt. Finally, it is straightforward to establish that M' is functional and unambiguous as it can at most execute the annotated rules and transitions (and can perform this in at most one fashion).

It remains to prove that $\|M\|_{\mathbb{K}} = \nu$; $\|M'\|_{\mathbb{K}}$, for which we first prove that $\|M\|_{\mathbb{K}}^q = \nu$; $\|M'\|_{\mathbb{K}}^q$ for every $q \in Q$ and $\|A\|_{\mathbb{K}}^{q'} = \nu$; $\|A'\|_{\mathbb{K}}^{q'}$ for every $q' \in Q'$. These proofs can be found in the appendix.

We prove that we can compose any such relabeling to the right of any $l\text{-wxt}^R$.

Lemma 13. For all $w \subseteq \{\mathbf{n}, \mathbf{s}\}$

$$\begin{aligned} w\not\!l\text{-WXT}^R(\mathbb{K}) ; \text{WREL}(\mathbb{K}) &\subseteq w\not\!l\text{-WXT}^R(\mathbb{K}) \\ w\not\!l\text{-WXT}(\mathbb{K}) ; \text{WREL}(\mathbb{K}) &\subseteq w\not\!l\text{-WXT}(\mathbb{K}) . \end{aligned}$$

Proof. Let $M = (Q, \Sigma, \Delta, q_0, R, \text{wt}, c)$ be an ε -free $l\text{-wxt}^R$ and $\nu: \Delta \times \Delta' \rightarrow K$ be a weighted relabeling. We construct the $l\text{-wxt}^R$ $M' = (Q, \Sigma, \Delta', q_0, R', \text{wt}', c)$ such that $\langle \ell, q, r' \rangle \in R'$ and $\text{wt}'(\langle \ell, q, r' \rangle) = \sum_{\langle \ell, q, r'' \rangle \in R} \text{wt}(\langle \ell, q, r'' \rangle) \cdot \nu(r'', r')$ for every translation rule $\langle \ell, q, r \rangle \in R$ and $r' \in T_{\Delta'}(Q)$ with $\nu(r, r') \neq 0$. No additional rules are in R' . Since the left-hand sides and the shape of the right-hand sides remains the same, it is clear that M' inherits the properties ‘ ε -free’, ‘nondeleting’, and ‘strict’ directly from M . Since the look-ahead coincides these results also hold for $l\text{-wxt}$.

It remains to prove that $\|M\|_{\mathbb{K}} ; \nu = \|M'\|_{\mathbb{K}}$, which we prove again with the help of $\|M\|_{\mathbb{K}}^q ; \nu = \|M'\|_{\mathbb{K}}^q$ for every $q \in Q$. The proof details are in the appendix.

We now have the two ingredients that allow us to normalize composition chains of ε -free $l\text{-wxt}^R$.

Theorem 14. For all $n \geq 1$ and $w \subseteq \{\mathbf{n}, \mathbf{s}\}$

$$\begin{aligned} w\not\!l\text{-WXT}^R(\mathbb{K})^n &\subseteq \text{WREL}(\mathbb{K}) ; w\not\!l\text{-WXT}^R(\mathbb{K})^n \\ w\not\!l\text{-WXT}(\mathbb{K})^n &\subseteq \text{WREL}(\mathbb{K}) ; w\not\!l\text{-WXT}(\mathbb{K})^n . \end{aligned}$$

Proof. We prove the statements by induction on n . For $n = 1$, they are proved in Lemma 12. Suppose that the property holds for $n \geq 1$. Then using Lemmas 12 and 13 and the induction hypothesis in sequence we obtain

$$\begin{aligned} w\not\!l\text{-WXT}^R(\mathbb{K})^{n+1} &\subseteq w\not\!l\text{-WXT}^R(\mathbb{K})^n ; \text{WREL}(\mathbb{K}) ; w\not\!l\text{-WXT}^R(\mathbb{K}) \\ &\subseteq w\not\!l\text{-WXT}^R(\mathbb{K})^n ; w\not\!l\text{-WXT}^R(\mathbb{K}) \subseteq \text{WREL}(\mathbb{K}) ; w\not\!l\text{-WXT}^R(\mathbb{K})^{n+1} \end{aligned}$$

and the same reasoning proves the statement also for $l\text{-wxt}$.

Finally, we compose a weighted relabeling to the left of an ε -free $l\text{-wxt}^R$ to eliminate the weighted relabeling again.

Lemma 15. $\text{WREL}(\mathbb{K}) ; w\not\!l\text{-WXT}^R(\mathbb{K}) \subseteq w\not\!l\text{-WXT}^R(\mathbb{K})$ for all $w \subseteq \{\mathbf{n}, \mathbf{s}\}$.

Proof. Let $\nu: \Sigma \times \Sigma' \rightarrow K$ be a weighted relabeling, $M = (Q, \Sigma', \Delta, q_0, R, \text{wt}, c)$ be an arbitrary ε -free $l\text{-wxt}^R$. The ε -free $l\text{-wxt}^R$ $M' = (Q, \Sigma, \Delta, q_0, R', \text{wt}', c)$ is given by $\langle \ell', q, r \rangle \in R'$ and $\text{wt}'(\langle \ell', q, r \rangle) = \sum_{\langle \ell'', q, r \rangle \in R} \nu(\ell', \ell'') \cdot \text{wt}(\langle \ell'', q, r \rangle)$ for every rule $\rho = \langle \ell, q, r \rangle \in R$ and $\ell' \in T_{\Sigma}(Q)$ with $\nu(\ell', \ell) \neq 0$. No additional rules

are in R' . In addition, we let $c^q = \nu$; c^q for every $q \in Q$, which is regular by Theorem 1. Obviously, the constructed $\text{l-wxt}^R M'$ is ε -free, and it inherits the properties ‘nondeleting’ and ‘strict’ from M .

To prove that ν ; $\|M\|_{\mathbb{K}} = \|M'\|_{\mathbb{K}}$, we first prove that ν ; $\|M\|_{\mathbb{K}}^q = \|M'\|_{\mathbb{K}}^q$ for every $q \in Q$. Both proofs can be found in the appendix.

Now we are ready to state and prove our main results.

Theorem 16.

$$\not\in\text{ns1-WXT}(\mathbb{K})^3 = \not\in\text{ns1-WXT}(\mathbb{K})^2 \quad (4a)$$

$$\not\in\text{s1-WXT}(\mathbb{K})^3 = \not\in\text{s1-WXT}(\mathbb{K})^2 \quad (4b) \quad \not\in\text{s1-WXT}^R(\mathbb{K})^3 = \not\in\text{s1-WXT}^R(\mathbb{K})^2 \quad (4d)$$

$$\not\in\text{l1-WXT}(\mathbb{K})^5 = \not\in\text{l1-WXT}(\mathbb{K})^4 \quad (4c) \quad \not\in\text{l1-WXT}^R(\mathbb{K})^4 = \not\in\text{l1-WXT}^R(\mathbb{K})^3 \quad (4e)$$

Additionally, the classes of (4b) and (4d) as well as (4c) and (4e) coincide.

Proof. All the right-to-left inclusions are trivial. The left-to-right inclusions are shown as follows. To prove (4a), let $\tau \in \not\in\text{ns1-WXT}(\mathbb{K})^3$ be of type $\tau: T_{\Sigma} \times T_{\Delta} \rightarrow K$. According to Theorem 14 there exist a weighted relabeling $\nu: \Sigma \times \Sigma' \rightarrow K$ and $\tau': T_{\Sigma'} \times T_{\Delta} \rightarrow K$ such that $\tau = \nu$; τ' and $\tau' \in \not\in\text{nsBful-WXT}(\mathbb{K})^3$. Since the composition of functional weighted tree transformations is naturally again functional, we can apply Lemma 8 to obtain that $\tau' \in \not\in\text{nsful-WXT}(\mathbb{B})^3$. Using (3a), we obtain $\tau' \in \not\in\text{ns1-WXT}(\mathbb{B})^2$. Let $\tau' = \tau'_1$; τ'_2 with $\tau'_1, \tau'_2 \in \not\in\text{ns1-WXT}(\mathbb{B})$. Next we restrict the range of τ'_1 to the regular tree language $\text{dom}(\tau'_2)$. In this way, we obtain the tree transformation $\tau''_1 \in \not\in\text{ns1-WXT}(\mathbb{B})$. Clearly, τ''_1 ; $\tau'_2 = \tau'_1$; τ'_2 . Using Lemma 10 we additionally obtain uniformizers $f''_1, f''_2 \in \not\in\text{nsf1-WXT}(\mathbb{B})$ for τ''_1 and τ'_2 , respectively. Since τ' is functional, we fulfill all the requirements of Lemma 9 and we can conclude that f''_1 ; $f''_2 = \tau''_1$; $\tau'_2 = \tau'_1$; $\tau'_2 = \tau'$. Hence $\tau' \in \not\in\text{nsf1-WXT}(\mathbb{B})^2$. With the help of Lemma 11 we obtain $\tau' \in \not\in\text{nsful-WXT}(\mathbb{B})^2$, which immediately also yields $\tau' \in \not\in\text{nsBful-WXT}(\mathbb{K})^2$ by Lemma 8. Finally, we utilize Lemma 15 to show that $\tau = \nu$; $\tau' \in \not\in\text{ns1-WXT}(\mathbb{K})^2$ as desired.

This approach works in the same manner for (4d) and (4e). However, instead of (3a), we use the results (3d) and (3e), respectively.

To prove (4b) we proceed in the same manner as in case (4a) and obtain that $\tau = \nu$; τ' , where ν is a weighted relabeling and $\tau' \in \not\in\text{sf1-WXT}(\mathbb{B})^3$. Then we use

$$\not\in\text{s1-WXT}(\mathbb{B})^3 = \not\in\text{ns1-WXT}(\mathbb{B}) ; \not\in\text{s1-WXT}^R(\mathbb{B})$$

from [6, Theorem 20], and again as in the proof of case (4a) we obtain the statement $\tau' \in \not\in\text{nsBful-WXT}(\mathbb{K}) ; \not\in\text{sBful-WXT}^R(\mathbb{K})$. It is well-known that regular look-ahead can be simulated by a nondeleting, strict, and linear weighted top-down tree transducer. Thus, we obtain that τ' is an element of

$$\begin{aligned} & \not\in\text{nsBful-WXT}(\mathbb{K}) ; \not\in\text{sBful-WXT}^R(\mathbb{K}) \subseteq \not\in\text{ns1-WXT}(\mathbb{K}) ; \text{ns1-WT}(\mathbb{K}) ; \not\in\text{s1-WXT}(\mathbb{K}) \\ & \subseteq \not\in\text{ns1-WXT}(\mathbb{K}) ; \not\in\text{s1-WXT}(\mathbb{K}) \subseteq \not\in\text{s1-WXT}(\mathbb{K})^2, \end{aligned}$$

where the second inclusion is due to [13, Theorem 8]. Then by Lemma 15 we obtain that $\tau \in \not\in \mathbf{s1-WXT}(\mathbb{K})^2$.

Finally, for (4c) we also proceed as usual: $\tau = \nu$; τ' , where ν is a weighted relabeling and $\tau' \in \not\in \mathbf{ful-WXT}(\mathbb{B})^5$. We continue with

$$\not\in \mathbf{1-WXT}(\mathbb{B})^5 = \not\in \mathbf{1-WXT}(\mathbb{B})^4 \subseteq \not\in \mathbf{1-WXT}^R(\mathbb{B})^4 \subseteq \mathbf{1-WT}^R(\mathbb{B}); \not\in \mathbf{s1-WXT}^R(\mathbb{B})^2,$$

where the first equality is by (3c) and the second inclusion is due to [6, Theorem 24]. Continuing on as before, we arrive at

$$\tau' \in \mathbf{Bful-WT}^R(\mathbb{K}); \not\in \mathbf{sBful-WXT}^R(\mathbb{K})^2.$$

Removing the regular look-ahead with the constructions already mentioned, we obtain

$$\tau' \in \mathbf{ns1-WT}(\mathbb{K}); \mathbf{1-WT}(\mathbb{K}); \not\in \mathbf{s1-WXT}(\mathbb{K})^2 \subseteq \not\in \mathbf{1-WXT}(\mathbb{K})^4$$

and then by Lemma 15 we conclude that $\tau \in \not\in \mathbf{1-WXT}(\mathbb{K})^4$. The final equalities between the classes of (4b) and (4d) as well as (4c) and (4e) follow directly from the presented arguments.

References

1. Arnold, A., Dauchet, M.: Morphismes et bimorphismes d'arbres. *Theor. Comput. Sci.* **20**(1), 33–93 (1982). [https://doi.org/10.1016/0304-3975\(82\)90098-6](https://doi.org/10.1016/0304-3975(82)90098-6)
2. Benedikt, M., Engelfriet, J., Maneth, S.: Determinacy and rewriting of functional top-down and MSO tree transformations. *J. Comput. Syst. Sci.* **85**, 57–73 (2017). <https://doi.org/10.1016/j.jcss.2016.11.001>
3. Chiang, D.: An introduction to synchronous grammars. In: Calzolari, N., Cardie, C., Isabelle, P. (eds.) *Proceedings of 44th Annual Meeting ACL. Association for Computational Linguistics* (2006). Part of a tutorial given with Kevin Knight. <https://www3.nd.edu/~dchiang/papers/synchtut.pdf>
4. Engelfriet, J.: Top-down tree transducers with regular look-ahead. *Math. Syst. Theory* **10**(1), 289–303 (1977). <https://doi.org/10.1007/BF01683280>
5. Engelfriet, J.: On tree transducers for partial functions. *Inf. Process. Lett.* **7**(4), 170–172 (1978). [https://doi.org/10.1016/0020-0190\(78\)90060-1](https://doi.org/10.1016/0020-0190(78)90060-1)
6. Engelfriet, J., Fülöp, Z., Maletti, A.: Composition closure of linear extended top-down tree transducers. *Theory Comput. Syst.* **60**(2), 129–171 (2017). <https://doi.org/10.1007/s00224-015-9660-2>
7. Fülöp, Z., Maletti, A., Vogler, H.: Weighted extended tree transducers. *Fund. Inform.* **111**(2), 163–202 (2011). <https://doi.org/10.3233/FI-2011-559>
8. Gécseg, F., Steinby, M.: *Tree Automata*, 2nd edn. Akadémiai Kiadó, Budapest (1984). <https://arxiv.org/abs/1509.06233>
9. Golan, J.S.: *Semirings and Their Applications*. Kluwer Academic, Dordrecht (1999). <https://doi.org/10.1007/978-94-015-9333-5>
10. Graehl, J., Knight, K., May, J.: Training tree transducers. *Comput. Linguist.* **34**(3), 391–427 (2008). <https://doi.org/10.1162/coli.2008.07-051-R2-03-57>
11. Hebisch, U., Weinert, H.J.: *Semirings - Algebraic Theory and Applications in Computer Science. Series in Algebra*. World Scientific, Singapore (1998). <https://doi.org/10.1142/3903>

12. Knight, K., Graehl, J.: An overview of probabilistic tree transducers for natural language processing. In: Gelbukh, A. (ed.) CICALing 2005. LNCS, vol. 3406, pp. 1–24. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30586-6_1
13. Lagoutte, A., Maletti, A.: *Survey*: weighted extended top-down tree transducers part III—composition. In: Kuich, W., Rahonis, G. (eds.) Algebraic Foundations in Computer Science. LNCS, vol. 7020, pp. 272–308. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24897-9_13
14. Maletti, A.: The power of weighted regularity-preserving multi bottom-up tree transducers. *Int. J. Found. Comput. Sci.* **26**(7), 987–1005 (2015). <https://doi.org/10.1142/S0129054115400109>
15. Maletti, A.: Compositions of tree-to-tree statistical machine translation models. In: Brlek, S., Reutenauer, C. (eds.) DLT 2016. LNCS, vol. 9840, pp. 293–305. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53132-7_24
16. Maletti, A., Graehl, J., Hopkins, M., Knight, K.: The power of extended top-down tree transducers. *SIAM J. Comput.* **39**(2), 410–430 (2009). <https://doi.org/10.1137/070699160>
17. May, J., Knight, K., Vogler, H.: Efficient inference through cascades of weighted tree transducers. In: Hajič, J., Carberry, S., Clark, S., Nivre, J. (eds.) Proceedings of 48th Annual Meeting ACL, pp. 1058–1066. Association for Computational Linguistics (2010). <http://www.aclweb.org/anthology/P10-1108>