Bottom-up and Top-down Tree Series Transformations

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

Heiko Vogler
Department of Computer Science
Technical University of Dresden
D-01062 Dresden, Germany
E-mail: `vogler@inf.tu-dresden.de`

October 23, 2000

Abstract: We generalize bottom-up tree transducers and top-down tree transducers to the concept of *bottom-up tree series transducer* and *top-down tree series transducer*, respectively, by allowing formal tree series as output rather than trees, where a formal tree series is a mapping from output trees to some semiring. We associate two semantics with a tree series transducer: a mapping which transforms trees into tree series (for short: tree to tree series transformation or t-ts transformation), and a mapping which transforms tree series into tree series (for short: tree series transformation or ts-ts transformation).

We show that the standard case of tree transducers is reobtained by choosing the boolean semiring under the t-ts semantics. Also, for each of the two types of tree series transducers and for both types of semantics, we prove a characterization which generalizes in a straightforward way the corresponding characterization result for the underlying tree transducer class. More precisely, we prove that polynomial bottom-up tree series transducers can be characterized by the composition of finite state relabeling tree series transducers and homomorphism tree series transducers. Moreover, we prove that the total deterministic top-down tree series transducers can be characterized by the composition of homomorphism tree series transducers and linear total deterministic top-down tree series transducers.

1 Introduction

The investigation of this paper was inspired by [Kui99] in which a first attempt was made to integrate formal power series over trees into a restricted form of top-down tree transducer. The resulting tree series transducers transform an input tree into a formal tree series which is a mapping from the set of output trees into some semiring. By this means it is possible to measure the computation of output trees. Here we introduce the concepts of bottom-up tree series transducer and of (unrestricted) top-down tree series transducer. Before we discuss our investigation, we will now briefly review the origins of tree series transducers which are a) tree transducers and b) automata with multiplicity (or cost functions).

Tree transducers have been introduced in [Rou68, Rou70, Tha70]. They can be viewed as generalizations of generalized sequential machines [Ber79] to trees where the trees are either read and processed from their leaves towards their root (bottom-up tree transducer) or from their roots towards their leaves (top-down tree transducer). A rule (or: transition) of a bottom-up tree transducer looks like

$$\sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(t)$$

and a rule of a top-down tree transducer has the form

$$q(\sigma(x_1, \dots, x_k)) \rightarrow \zeta$$

where $q, q_1, \dots, q_k$ are states of the tree transducers, σ is an input symbol of rank k, $x_1, \dots, x_k$ are variables ranging over trees (more precisely, over output trees in the bottom-up case, and over input trees

*Research of this author was supported by the Hungarian Scientific Foundation (OTKA) under Grant T 030084

in the top-down case), t is an output tree which may contain variables from the set $X_k = \{x_1, \ldots, x_k\}$, and ζ is an output tree which may contain constructs of the form $p(x_i)$ at its leaves where p is a state and $1 \le i \le k$. In the usual way, the rules of a tree transducer constitute a binary term rewriting relation (or: derivation relation) $\Rightarrow_M$ by means of which the tree transformation $\tau_M \subseteq T_\Sigma \times T_\Delta$ computed by M can be defined (where T_Σ and T_Δ are the sets of input trees and output trees, respectively).

Since the seventies, tree transducers have been studied intensively. The first substantial papers dealt with (de-)composition and hierarchy results [Bak79, Eng75, Eng77, Eng82]. In [FV92] a method of deciding the equivalence of the compositions of classes of tree transformations is overviewed. Survey articles and books are [GS84, GS97, CDG+97, FV98]. Recently, a characterization of tree transformation classes in terms of monadic second order logic has been proved [BE00, EM99].

Now let us briefly review the second origin of tree series transducers: automata with multiplicities. Let $M = (Q, \Sigma, \mu, q_0, Q_d)$ be a usual finite state string automaton where Q is the set of states, Σ the set of input symbols, $\mu : Q \times \Sigma \to \mathcal{P}(Q)$ the transition function, $q_0 \in Q$ the initial state, and $Q_d \subseteq Q$ the set of final states. In the usual way, the transition function is extended to the function $\tilde{\mu} : \mathcal{P}(Q) \times \Sigma^* \to \mathcal{P}(Q)$, and the language accepted by M is the language $L(M) = \{w \in \Sigma^* \mid \tilde{\mu}(\{q_0\}, w) \cap Q_d \neq \emptyset\}$.

In [Sch62], finite state string automata were generalized by associating with every transition an element $a \in A$ of a semiring $(A, +, \cdot, 0, 1)$ assumed to be the cost (or multiplicity) of this transition (also cf. [Eil74, SS78, BR88, KS86]). In this approach, the transition function μ turns into a $(Q \times Q)$-matrix $\mu \in (A\langle\langle\Sigma\rangle\rangle)^{Q\times Q}$ where $A\langle\langle\Sigma\rangle\rangle$ is the set of formal power series (in non-commuting variables), i.e., mappings of type $\Sigma \to A$. Intuitively, $\mu_{p,q}(\sigma) \in A$ is the cost of making a transition from state q to state p while reading σ (note the order of the indices p and q). The usual automaton (without cost functions) is reobtained by considering the boolean semiring. Then μ can be used to define the behaviour of a finite state automaton M with cost functions such that it is a formal power series $\tau_M \in A\langle\langle\Sigma^*\rangle\rangle$ obtained by iterating the transition matrix μ.

This approach was generalized in [Kui97a] to tree automata over semirings (for a theoretical investigation on formal tree series, i.e., mappings of the type $T_\Sigma \to A$, cf. [BR82]; in that paper, A is a vector space). Since the input symbols now have ranks, the transition matrix μ has to be replaced by a family $\mu = (\mu_k \mid k \ge 0)$ of transition matrices $\mu_k \in (A\langle\langle\Sigma(X_k)\rangle\rangle)^{Q\times Q^k}$ where $\Sigma(X_k) = \{\sigma(x_1, \ldots, x_k) \mid \sigma \in \Sigma^{(k)}\}$. Intuitively, if the tree automaton has already accepted the input trees $t_1, \ldots, t_k$ in the states $q_1, \ldots, q_k$, respectively, then $(\mu_k)_{q,q_1\ldots q_k}(\sigma(x_1, \ldots, x_k)) \in A$ is the cost for making a transition from $q_1, \ldots, q_k$ to q while reading the k-ary symbol σ. The semantics of a tree automaton M with cost functions is a formal tree series $\tau_M \in A\langle\langle T_\Sigma\rangle\rangle$. We mention that tree automata with cost functions have been used in order to select code in compilers on the basis of certain cost criteria [FSW94] (also cf. [Sei92]).

In [Kui99] such tree automata with cost functions have been equipped with output and thereby the concept of tree transducer with cost functions has been introduced. In order to make this concept understandable on the background of the development explained above, let us reconsider the tree automata with cost functions. Such an automaton can be considered as a generating device: the trees together with their costs are built up and maintained in the entries of a matrix of type $(A\langle\langle T_\Sigma\rangle\rangle)^{Q\times 1}$. An alternative point of view is to formalize a tree automaton with cost functions as a transducer which computes the partial identity. More precisely, it computes the function $\tau_M : T_\Sigma \to A\langle\langle T_\Sigma\rangle\rangle$ such that the formal tree series $\tau_M(s)$ has the value 0 for every tree t such that $t \neq s$. Then, the transitions are specified by a family $\mu = (\mu_k \mid k \ge 0)$ of mappings $\mu_k : \Sigma^{(k)} \to (A\langle\langle\Sigma(X_k)\rangle\rangle)^{Q\times Q^k}$ and the behaviour of the automaton can be specified by means of the initial homomorphism $h_\mu : T_\Sigma \to (A\langle\langle T_\Sigma\rangle\rangle)^{Q\times 1}$.

Now the tree transducers with cost functions as they are defined in [Kui99] (called: tree transducers there) can be understood as a modification of this alternative approach in the sense that the input tree is not reproduced, but another output tree is produced. Then the transitions are defined by a family $\mu = (\mu_k \mid k \ge 0)$ of mappings $\mu_k : \Sigma^{(k)} \to (A\langle\langle T_\Delta(X_k)\rangle\rangle)^{Q\times Q^k}$ for some output alphabet Δ. On the basis of these tree representations, in [Kui99] restricted types of top-down tree transducers with cost functions have been introduced and studied; the underlying transducer model is called *nondeterministically simple top-down tree transducer* in which, roughly speaking, to every subtree at most one state is applied (cf. [GS84]). Then a tree transducer M with cost functions computes the function $\tau_M : T_\Sigma \to A\langle\langle T_\Delta\rangle\rangle$. This finishes the short review about the origins of our investigation.

In the present paper, we take a more liberal point of view on tree representations. We consider tree representations of the form $\mu_k : \Sigma_k \to (A\langle\langle T_\Delta(X)\rangle\rangle)^{Q\times(Q(X_k))^*}$ such that only for finitely many indices $(q, w) \in Q \times (Q(X_k))^*$, $\mu_k(\sigma)_{q,w} \neq \tilde{0}$, where $\tilde{0}$ is the formal tree series which maps every tree to 0. The following table summarizes the explained development of the function μ.

type of concept	transition function	computed function
finite state automaton	$\mu : Q \times \Sigma \rightarrow \mathcal{P}(Q)$	$L(M) \subseteq \Sigma^*$
string automaton with cost functions	$\mu \in (A\langle\langle\Sigma\rangle\rangle)^{Q\times Q}$	$\tau_M \in A\langle\langle\Sigma^*\rangle\rangle$
tree automaton with cost functions	$\mu_k \in (A\langle\langle\Sigma(X_k)\rangle\rangle)^{Q\times Q^k}$	$\tau_M \in A\langle\langle T_\Sigma\rangle\rangle$
tree automaton with cost functions (alternative point of view)	$\mu_k(\sigma) \in (A\langle\langle\Sigma(X_k)\rangle\rangle)^{Q\times Q^k}$	$\tau_M : T_\Sigma \rightarrow A\langle\langle T_\Sigma\rangle\rangle$ (partial identity)
tree transducers with cost functions (in the sense of [Kui99])	$\mu_k(\sigma) \in (A\langle\langle T_\Delta(X_k)\rangle\rangle)^{Q\times Q^k}$	$\tau_M : T_\Sigma \rightarrow A\langle\langle T_\Delta\rangle\rangle$
tree series transducers (in the present paper)	$\mu_k(\sigma) \in (A\langle\langle T_\Delta(X)\rangle\rangle)^{Q\times(Q(X_k))^*}$	$\tau_M : T_\Sigma \rightarrow A\langle\langle T_\Delta\rangle\rangle$

Next we instantiate these tree representations in two different ways thereby obtaining the bottom-up and the top-down case. In the bottom-up case we additionally require that, for every index w, if $w \neq q_1(x_1)\ldots q_k(x_k)$ for some $q_1,\ldots,q_k \in Q$, then $\mu_k(\sigma)_{q,w} = \tilde{0}$. Moreover, if $\mu_k(\sigma)_{q,w} \neq \tilde{0}$, then $\mu_k(\sigma)_{q,w} \in A\langle\langle T_\Delta(X_k)\rangle\rangle$. Then we call μ a *bottom-up tree representation*. To show an example, let us consider the two rules

$$\delta(q_1(x_1), q_2(x_2), q_3(x_3)) \rightarrow q(\sigma(x_2, \sigma(x_3, x_2))) \ \mid\ p(\sigma(x_1, \sigma(\gamma(x_3), x_1)))$$

of some bottom-up tree transducer, where δ, σ, γ are input/output symbol of rank 3, 2, and 1, respectively, q_1, q_2, q are states, and $\mid$ delimits the right-hand sides of the two rules. Let us assume that the costs for these two rules are a and b, respectively, where $a, b \in A$. Then these rules are represented by

$$\mu_3(\delta)_{q,w}(\sigma(x_2, \sigma(x_3, x_2))) = a$$

$$\mu_3(\delta)_{p,w}(\sigma(x_1, \sigma(\gamma(x_3), x_1))) = b$$

where $w = q_1(x_1)q_2(x_2)q_3(x_3)$. We observe that the structure of w is very restricted: w is the left hand side of the rule from which the input symbol (and the commas) have been stripped of. On the other hand, the structure of the trees $\sigma(x_2, \sigma(x_3, x_2))$ and $\sigma(x_1, \sigma(\gamma(x_3), x_1))$ to which the mapping $\mu_3(\delta)_{q,w}$ is applied, is liberal in the sense that variables of the left hand side of the rule may occur an arbitrary number of times.

In the top-down case we require that, if $\mu_k(\sigma)_{q,w} \neq \tilde{0}$, then $\mu_k(\sigma)_{q,w} \in A\langle\langle \widehat{T_\Delta}(X_l)\rangle\rangle$, where $l = |w|$ and $\widehat{T_\Delta}(X_l)$ denotes the set of all trees t over $\Delta \cup X_l$ such that, for every $1 \leq i \leq k$, the variable x_i occurs exactly once in t and, reading the leaves of t from left to right, the variables occur in the order $x_1 < x_2 < \ldots < x_n$. Then μ is a *top-down tree representation*. As an example, let us consider the two rules

$$q(\delta(x_1, x_2, x_3)) \rightarrow \sigma(p(x_2), \sigma(q(x_1), p(x_1)) \ \mid\ \sigma(\gamma(q(x_2)), p(x_2))$$

with costs c and d, respectively. These rules are represented by

$$\mu_3(\delta)_{q,v}(\sigma(x_1, \sigma(x_2, x_3))) = c$$

$$\mu_3(\delta)_{q,v'}(\sigma(\gamma(x_1)), x_2)) = d$$

where $v = p(x_2)q(x_1)q(x_1)$ and $v' = q(x_2)p(x_2)$. We observe that now v and v' are more flexible: they are obtained by scanning the frontier of the right hand side of the rules and listing all objects of the form $p(x_i)$. On the other hand, the trees $\sigma(x_1, \sigma(x_2, x_3)))$ and $\sigma(\gamma(x_1)), x_2)$ to which the mappings $\mu_3(\delta)_{q,v}$ and $\mu_3(\delta)_{q,v'}$ are applied, respectively, have a much more restricted form than in the bottom-up case: they are linear in the variables and the variables occur ordered.

The bottom-up tree representation and the top-down tree representation define the bottom-up tree series transducer and the top-down tree series transducer, respectively, via the concept of the initial homomorphism $h_\mu : T_\Sigma \rightarrow (A\langle\langle T_\Sigma\rangle\rangle)^{Q\times 1}$. With every tree series transducer, we associate two semantics: the *tree to tree series semantics* (t-ts semantics) which is a mapping of type $T_\Sigma \rightarrow A\langle\langle T_\Delta\rangle\rangle$, and the *tree series to tree series semantics* (ts-ts semantics), which is a mapping of type $A\langle\langle T_\Sigma\rangle\rangle \rightarrow A\langle\langle T_\Delta\rangle\rangle$. As in the case of tree transducers (cf. [Eng75]) we define certain important subclasses, like linear, relabeling, and homomorphism tree series transducers.

The contents of this paper is as follows. In Section 2 we develop the notions and basic definitions of bottom-up and top-down tree transducers, semirings, formal tree series; we tried to make this paper self contained. In Section 3 we introduce the concepts of bottom-up tree series transducers and of top-down tree series transducers, and we discuss an example in detail. In Section 4 we show that, using the boolean semiring, there is a one-to-one correspondence between tree series transducers and tree transducers (of the same type). In Section 5 we prove two characterization results: 1) polynomial bottom-up tree series transducers can be characterized by the composition of finite state relabeling tree series transducers and homomorphism tree series transducers, and 2) deterministic top-down tree series transducers can be characterized by the composition of homomorphism tree series transducers and linear deterministic top-down tree series transducers.

The proofs of the correctness of constructions is usually done by an induction on the tree structure. In order to save space, we only show the induction step, and not the induction base.

2 Preliminaries

2.1 Sets and relations

Let H be a set. Then $\mathcal{P}(H)$ is the powerset of H. The empty set is denoted by $\emptyset$. Let F, G, and H be sets and let $\varrho \subseteq F \times G$ and $\tau \subseteq G \times H$ be two binary relations. The composition of ϱ and τ is the binary relation $\varrho \circ \tau = \{(a, c) \subseteq F \times H \mid \text{there is a } b \in G \text{ such that } (a, b) \in \varrho \text{ and } (b, c) \in \tau\}$. The notion of composition is extended to classes of relations. For two classes C_1 and C_2 of relations we define $C_1 \circ C_2 = \{\varrho \circ \tau \mid \varrho \in C_1 \text{ and } \tau \in C_2\}$.

2.2 Strings

If Σ is an alphabet, then Σ^* denotes the set of strings over Σ, while the empty string is denoted by ε. The length of a string $w \in \Sigma^*$ is defined in a standard way and denoted by $|w|$.

2.3 Trees and tree transducers

Let Σ be a ranked alphabet. By $\Sigma^{(k)}$ we denote the set of all symbols of Σ which have rank k. Moreover, let H be a set disjoint with Σ. The set of *(finite, labeled and ordered) trees over* Σ *indexed by* H, denoted by $T_\Sigma(H)$, is the smallest subset T of $(\Sigma \cup H \cup \{(,)\} \cup \{,\})^*$, such that (i) $H \subseteq T$ and (ii) if $\sigma \in \Sigma^{(k)}$ with $k \geq 0$ and $s_1, \ldots, s_k \in T$, then $\sigma(s_1, \ldots, s_k) \in T$. In case $k = 0$, we identify $\sigma(\,)$ with σ. Moreover, $T_\Sigma(\emptyset)$ is denoted by T_Σ. It should be clear that $T_\Sigma = \emptyset$ if and only if $\Sigma^{(0)} = \emptyset$. Since we are not interested in this particular case, we assume that $\Sigma^{(0)} \neq \emptyset$ for every ranked alphabet Σ appearing as input or output ranked alphabet of some tree transducer in this paper.

We will need the set $X = \{x_1, x_2, \ldots\}$ of *variable symbols*. For every $k \geq 0$, we define $X_k = \{x_1, \ldots, x_k\}$, thus $X_0 = \emptyset$. We use the variables to occur in trees, so we will frequently consider the sets $T_\Sigma(X)$, $T_\Sigma(X_k)$, etc. of trees where Σ is a ranked alphabet.

The *tree substitution* is defined as follows. Let $t \in T_\Sigma(X_k)$ for some ranked alphabet Σ and $k \geq 0$. Moreover, let $t_1, \ldots, t_k$ be also trees over (maybe other) ranked alphabets. Then $t[x_1 \leftarrow t_1, \ldots, x_k \leftarrow t_k]$, or shortly just $t[t_1, \ldots, t_k]$, stands for the tree which is obtained from t by substituting, for every $1 \leq i \leq k$, the tree t_i for each occurrence of x_i.

A tree $t \in T_\Sigma(X_k)$ is called *linear* if every variable x_i, $1 \leq i \leq k$ occurs at most once in it.

We distinguish a subset $\widehat{T_\Sigma}(X_k)$ of $T_\Sigma(X_k)$ as follows. Let a tree $t \in T_\Sigma(X_k)$ be in $\widehat{T_\Sigma}(X_k)$ if for every $1 \leq i \leq k$, the variable x_i occurs exactly once in t and, reading the leaves of t from left to right, the variables occur in the order $x_1 < x_2 < \ldots < x_n$. Note that all elements of $\widehat{T_\Sigma}(X_k)$ are linear.

Let $t \in T_\Sigma(H)$. The *linearization of* t *with respect to* H, denoted by $lin_H(t)$, is defined as the unique pair (t', w) where $t' \in \widehat{T_\Sigma}(X_k)$ and $w = a_1 \ldots a_k \in H^*$ such that $t = t'[x_1 \leftarrow a_1, \ldots, x_k \leftarrow a_k]$.

If Q is a unary ranked alphabet, i.e., the rank of all symbols in Q is 1, then $Q(X_k)$ stands for the set $\{q(x_i) \mid q \in Q \text{ and } 1 \leq i \leq k\}$. A string $w \in (Q(X_k))^*$ is called linear if every variable x_i, $1 \leq i \leq k$ occurs at most once in it.

Next we shortly recall the concept of the bottom-up tree transducer and the top-down tree transducer. For more terminology and details about tree transducers the reader is advised to consult [Eng75].

A *bottom-up tree transducer* is a tuple $M = (Q, \Sigma, \Delta, Q_d, R)$ where Q is a unary ranked alphabet (of states), Σ and Δ are ranked alphabets (of input symbols and output symbols, resp.), $Q_d \subseteq Q$ is the set (of final states), and R is a finite set of rules of the form $\sigma(q_1(x_1), \ldots, q_k(x_k)) \to q(t)$, where $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $q_1, \ldots, q_k, q \in Q$, and $t \in T_\Delta(X_k)$. The *derivation relation (induced by M)* is the binary relation $\Rightarrow_M \subseteq T_{Q\cup\Sigma\cup\Delta} \times T_{Q\cup\Sigma\cup\Delta}$ defined by $\varphi \Rightarrow_M \psi$ iff there is a $\beta \in \widehat{T_\Sigma}(X_1)$, there is a rule $\sigma(q_1(x_1), \ldots, q_k(x_k)) \to q(t)$ in R, there are $t_1, \ldots, t_k \in T_\Delta$ such that $\varphi = \beta[x_1 \leftarrow \sigma(q_1(t_1), \ldots, q_k(t_k))]$ and $\psi = \beta[x_1 \leftarrow q(t[x_1 \leftarrow t_1, \ldots, x_k \leftarrow t_k])]$. The *tree transformation computed by M* is the set $\tau_M = \{(s,t) \mid s \Rightarrow_M^* q(t) \text{ for some } q \in Q_d\}$. The *class of all tree transformations computed by bottom-up tree transducers* is denoted by BOT_{tt}.

A *top-down tree transducer* is a tuple $M = (Q, \Sigma, \Delta, Q_d, R)$ where Q is a unary ranked alphabet (of states), Σ and Δ are ranked alphabets (of input symbols and output symbols, resp.), $Q_d \subseteq Q$ is the set (of initial states), and R is a finite set of rules of the form $q(\sigma(x_1, \ldots, x_k)) \to t$, where $q \in Q$, $k \geq 0$, $\sigma \in \Sigma^{(k)}$, and $t \in T_\Delta(Q(X_k))$. The *derivation relation (induced by M)* is the binary relation $\Rightarrow_M \subseteq T_{Q\cup\Sigma\cup\Delta} \times T_{Q\cup\Sigma\cup\Delta}$ defined by $\varphi \Rightarrow_M \psi$ iff there is a $\beta \in \widehat{T_\Delta}(X_1)$, there is a rule $q(\sigma(x_1, \ldots, x_k)) \to t$ in R, there are $s_1, \ldots, s_k \in T_\Sigma$ such that $\varphi = \beta[x_1 \leftarrow q(\sigma(s_1, \ldots, s_k))]$ and $\psi = \beta[x_1 \leftarrow t[x_1 \leftarrow s_1, \ldots, x_k \leftarrow s_k]]$. The *tree transformation computed by M* is the set $\tau_M = \{(s,t) \mid q(s) \Rightarrow_M^* t \text{ for some } q \in Q_d\}$. The *class of all tree transformations computed by top-down tree transducers* is denoted by TOP_{tt}.

2.4 Semirings

For a survey paper on the relevance of semirings and formal power series to formal languages and automata cf. [Kui97b].

A *semiring* is an algebraic structure $\underline{A} = (A, +, \cdot, 0, 1)$ with two operations sum $+$ and product $\cdot$ such that $(A, +, 0)$ is a commutative monoid, $(A, \cdot, 1)$ is a monoid, and for every $a, b, c \in A$ the following laws hold: $(a + b) \cdot c = a \cdot c + b \cdot c$, $a \cdot (b + c) = a \cdot b + a \cdot c$, and $a \cdot 0 = 0 \cdot a = 0$. Whenever the operations ($+$ and $\cdot$) and the neutral elements (0 and 1) are clear from the context, then we denote the semiring $(A, +, \cdot, 0, 1)$ just by A. We give some examples of semirings.

- The *boolean semiring* is the semiring $\mathbb{B} = (\{0, 1\}, \vee, \wedge, 0, 1)$ with disjunction and conjunction as sum and product, resp.

- Let Δ be an alphabet. The *semiring of formal languages (over Δ)* , denoted by $Lang(\Delta)$, is the semiring $(\mathcal{P}(\Delta^*), \cup, \cdot, \emptyset, \{\varepsilon\})$ is a semiring where $\cdot$ is the usual concatenation of string languages.

- The *semiring of natural numbers* is the semiring $Nat = (\mathbb{N}, +, \cdot, 0, 1)$ with the usual addition and multiplication operations.

Let $(A, +, \cdot, 0, 1)$ be a semiring. A is *commutative* if, for every $a, b \in A$, the equation $a \cdot b = b \cdot a$ holds.

A is *naturally ordered* if the binary relation $\sqsubseteq$ on A is a partial order on A, where $\sqsubseteq$ is defined by $a \sqsubseteq b$ iff there is a $c \in A$ such that $a + c = b$.

A is *complete* if it is possible to define sums for all families $(a_i \mid i \in I)$ of elements of A, where I is an arbitrary index set, such that the following three conditions are satisfied:

(i) $\sum_{i\in\emptyset} a_i = 0$, $\sum_{i\in\{j\}} a_i = a_j$, $\sum_{i\in\{j,k\}} a_i = a_j + a_k$ for $j \neq k$.

(ii) $\sum_{j\in J}(\sum_{i\in I_j} a_i) = \sum_{i\in I} a_i$, if $\bigcup_{j\in J} I_j = I$ and $I_j \cap I_k = \emptyset$ for $j \neq k$.

(iii) $\sum_{i\in I}(c \cdot a_i) = c \cdot (\sum_{i\in I} a_i)$, $\sum_{i\in I}(a_i \cdot c) = (\sum_{i\in I} a_i) \cdot c$.

A is *ω-continuous* if the following three conditions holds.

(i) A is naturally ordered.

(ii) A is complete.

(iii) If $\sum_{0\leq i\leq n} a_i \sqsubseteq c$ for all $n \in \mathbb{N}$, then $\sum_{i\in\mathbb{N}} a_i \sqsubseteq c$ for all sequences $(a_i \mid i \in \mathbb{N})$ in A and all $c \in A$.

2.5 Matrices

We recall some notions concerning matrices (cf. [Kui97b]).

Let I, J and S be arbitrary sets. An $I \times J$ *matrix over* S is a mapping $M : I \times J \to S$. The set of all $I \times J$ matrices over S is denoted by $S^{I \times J}$. An element $M(i,j) \in S$ is also denoted by $M_{i,j}$. If I (or J) is a singleton, then we identify it by $\{1\}$ and we write just 1 for I (or J). Moreover in that case an element of the form $M_{i,1}$ (or $M_{1,j}$) is written just M_i (or M_j).

Let A be some structure in which the operations (infinite) sum and multiplication are defined. Let $M : I \times J \to A$ be a matrix and $a \in A$. Then aM is the $I \times J$ matrix over A such that for every $(i,j) \in I \times J$, $(aM)_{i,j} = a \cdot M_{i,j}$ where $\cdot$ is the multiplication of A. Moreover, for a family $(M^{(k)} \mid k \in K)$ of $I \times J$ matrices over A we define the $I \times J$ matrix $\sum_{k \in K} M^{(k)}$ by

$$(\sum_{k \in K} M^{(k)})_{i,j} = \sum_{k \in K} (M^{(k)})_{i,j}.$$

2.6 Formal tree series

Let Δ be ranked alphabet and let $(A, +, \cdot, 0, 1)$ be a semiring. A *formal tree series (over* Δ *and* A*)* is a mapping $r : T_\Delta \to A$. For every $t \in T_\Delta$, the element $r(t) \in A$ is called *coefficient* of t and it is also denoted by (r,t); then the mapping itself is written as the formal sum $\hat{\sum}_{t \in T_\Delta} (r,t)t$. The *set of all formal tree series over* Δ *and* A is denoted by $A\langle\langle T_\Delta \rangle\rangle$.

Note that in this paper we make a distinction between the formal sum $\hat{\sum}$ which denotes a formal tree series, and the sum $\sum$ which is an operation over some domain.

Example 2.1 Let Δ be a ranked alphabet and let $mx = \max\{j \mid \Delta^{(j)} \neq \emptyset\}$. Moreover, let $N = \{i \mid 0 \leq i \leq mx\}$. Consider the mapping $occ : T_\Delta \to \mathcal{P}(N^*)$ which maps every tree to its set of occurrences. More precisely, for every $t \in \Delta^{(0)}$, $occ(t) = \{\varepsilon\}$, and for every $t = \sigma(t_1, \ldots, t_k)$ where $\sigma \in \Delta^{(k)}$, $k \geq 1$ and $s_1, \ldots, s_k \in T_\Delta$, $occ(t) = \{\varepsilon\} \cup \bigcup_{1 \leq i \leq k} \{i\} \cdot occ(t_i)$. For example, if $\Delta = \{\sigma^{(2)}, \alpha^{(0)}\}$ and $t = \sigma(\sigma(\alpha, \alpha), \alpha)$, then $occ(t) = \{\varepsilon, 1, 11, 12, 2\}$. Hence, occ is a formal tree series over the semiring $Lang(N)$. $\square$

In the rest of this paper A stands for a commutative and continuous semiring. Let $a \in A$ and let $r_1, r_2 \in A\langle\langle T_\Delta \rangle\rangle$. The formal tree series $a \cdot r_1$ is the formal tree series $\hat{\sum}_{t \in T_\Delta} (a \cdot (r_1, t))t$. Moreover, $r_1 + r_2$ denotes the formal tree series $\hat{\sum}_{t \in T_\Delta} ((r_1, t) + (r_2, t))t$ and, for every family $(r_i \mid i \in I)$ of formal tree series over Δ and A, $\sum_{i \in I} r_i = \hat{\sum}_{t \in T_\Delta} (\sum_{i \in I} (r_i, t), t)$.

Let $r : T_\Delta \to A$ be a formal tree series. If there is an $a \in A$ such that for every $t \in T_\Delta$ we have $(r,t) = a$, then r is called *the constant* a and is denoted by $\tilde{a}$. The *support of* r is defined as the set $supp(r) = \{t \in T_\Delta \mid (r,t) \neq 0\}$. Moreover, the formal tree series r is called a *polynomial* if $supp(r)$ is finite and is called a *singleton* if $supp(r)$ is a singleton. Hence a singleton is a polinomial. The *set of all polynomials over* Δ *and* A is denoted by $A\langle T_\Delta \rangle$.

A polynomial $r \in A\langle T_\Delta \rangle$ is written as $a_1 t_1 \hat{+} \ldots \hat{+} a_n t_n$, where $supp(r) = \{t_1, \ldots, t_n\}$ and, for every $1 \leq i \leq n$, $a_i = (r, t_i)$. Moreover, if $a_i = 1$, then $a_i t_i$ is written as t_i. Hence for $t \in T_\Delta$ either t denotes the tree in T_Δ or it denotes the formal tree series $r : T_\Delta \to A$ such that $(r,t) = 1$ and $(r,s) = 0$ for every $s \in T_\Delta - \{t\}$. However, it will always be clear from the context which meaning is intended.

Let Δ be a ranked alphabet. We define the binary relation $pick_\Delta = \{(r,t) \mid r \in \mathbb{B}\langle T_\Delta \rangle, t \in supp(r)\} \subseteq \mathbb{B}\langle\langle T_\Delta \rangle\rangle \times T_\Delta$. Note that $pick_\Delta$ is a set of pairs and not a set of coefficients in $\mathbb{B}$. The class of relations $pick_\Delta$ for an arbitrary ranked alphabet Δ is denoted by $PICK$.

Now we define the substitution of formal tree series. This is a straightforward generalization of the IO-substitution of languages [ES77, ES78, EV85].

Let $t \in T_\Delta(X_l)$ and $\vec{s} = (s_1, \ldots, s_l) \in (A\langle\langle T_\Delta(X) \rangle\rangle)^l$. Define $t \leftarrow \vec{s} \in A\langle\langle T_\Delta(X) \rangle\rangle$ as follows:

(i) If $t = x_j$, $1 \leq j \leq l$, then $t \leftarrow \vec{s} = s_j$.

(ii) If $t = \alpha \in \Delta^{(0)}$, then $t \leftarrow \vec{s} = \alpha$.

(iii) If $t \notin \Delta^{(0)} \cup X_l$, then $t \leftarrow \vec{s} = \hat{\sum}_{t_1, \ldots, t_l \in T_\Delta(X)} (s_1, t_1) \ldots (s_l, t_l)(t[t_1, \ldots, t_l])$.

Note that, for every $t \in T_\Delta$, $t \leftarrow (\,) = t$. Moreover, if $s_i = \tilde{0}$ for some $1 \leq i \leq l$, then also $t \leftarrow \vec{s} = \tilde{0}$ no matter whether x_i occurs in t or not.

Now let $r \in A\langle\langle T_\Delta(X_l)\rangle\rangle$ and $\vec{s} = (s_1, \ldots, s_l) \in (A\langle\langle T_\Delta(X)\rangle\rangle)^l$. Define $r \leftarrow \vec{s} \in A\langle\langle T_\Delta(X)\rangle\rangle$ by $r \leftarrow \tilde{s} = \sum_{t \in T_\Delta(X_l)}(r,t)(t \leftarrow \tilde{s})$. Certainly $\tilde{0} \leftarrow \vec{s} = \tilde{0}$.

If the semiring A is the boolean semiring $\mathbb{B}$, then the substitution of formal tree series is the same as the IO-substitution on formal languages as defined in [ES77, ES78].

We note that in [Kui99] an OI-substitution of formal tree series is defined. The reason why we work with IO-substitution will be discussed at the end of Section 4.

2.7 Tree series transformations

A *tree to tree series transformation* (for short: t-ts transformation) is a mapping $\tau : T_\Sigma \to A\langle\langle T_\Delta\rangle\rangle$. A *tree series to tree series transformation* (for short: ts-ts transformation) is a mapping $\rho : A\langle\langle T_\Sigma\rangle\rangle \to A\langle\langle T_\Delta\rangle\rangle$.

Every t-ts transformation can be extended in a natural way to a ts-ts transformation. Let $\tau : T_\Sigma \to A\langle\langle T_\Delta\rangle\rangle$ be a t-ts transformation. The *extension of* τ is the mapping $\widetilde{\tau} : A\langle\langle T_\Sigma\rangle\rangle \to A\langle\langle T_\Delta\rangle\rangle$ such that for every $r \in A\langle\langle T_\Sigma\rangle\rangle$, $\widetilde{\tau}(r) = \sum_{s \in T_\Sigma}((r,s)\tau(s))$.

We also extend a mapping $\tau : T_\Sigma \to A\langle\langle T_\Delta\rangle\rangle^{I \times J}$ to $\widetilde{\tau} : A\langle\langle T_\Sigma\rangle\rangle \to A\langle\langle T_\Delta\rangle\rangle^{I \times J}$. In this case for an $r \in A\langle\langle T_\Sigma\rangle\rangle$ the extension $\widetilde{\tau}(r)$ is defined by the same equation but now it will be an $I \times J$ matrix over $A\langle\langle T_\Delta\rangle\rangle$.

Let $\tau_1 : T_\Sigma \to A\langle\langle T_\Delta\rangle\rangle$ and $\tau_2 : T_\Delta \to A\langle\langle T_\Gamma\rangle\rangle$ be two t-ts transformations. Then define the *composition of* τ_1 *and* τ_2, denoted by $\tau_1 \mathbin{\tilde{\circ}} \tau_2$, as the t-ts transformation $\tau_1 \mathbin{\tilde{\circ}} \tau_2 : T_\Sigma \to A\langle\langle T_\Gamma\rangle\rangle$ where $(\tau_1 \mathbin{\tilde{\circ}} \tau_2)(s) = \widetilde{\tau_2}(\tau_1(s))$ for every $s \in T_\Sigma$. We note that $\tilde{\circ}$ is not a composition of relations in the sense we defined it in Subsection 2.1. The *composition of two classes* C_1 *and* C_2 of t-ts transformations, denoted by $C_1 \mathbin{\tilde{\circ}} C_2$ is the class $\{\tau_1 \mathbin{\tilde{\circ}} \tau_2 \mid \tau_i \in C_i, 1 \le i \le 2\}$ of t-ts transformations.

There is a close relationship between the composition of t-ts transformations and ts-ts transformations which will be made clear in the next lemma.

Lemma 2.2 Let $\tau_1 : T_\Sigma \to A\langle\langle T_\Delta\rangle\rangle$, $\tau_2 : T_\Delta \to A\langle\langle T_\Gamma\rangle\rangle$, and $\tau : T_\Sigma \to A\langle\langle T_\Gamma\rangle\rangle$ be three t-ts transformations. Then, if $\tau_1 \mathbin{\tilde{\circ}} \tau_2 = \tau$, then $\widetilde{\tau_1} \circ \widetilde{\tau_2} = \widetilde{\tau}$.

Proof. Let $r \in A\langle\langle T_\Sigma\rangle\rangle$. Then

$$\begin{array}{ll} & \widetilde{\tau}(r) \\ = & \sum_{s \in T_\Sigma}(r,s)\tau(s) \\ = & \sum_{s \in T_\Sigma}(r,s)(\tau_1 \mathbin{\tilde{\circ}} \tau_2)(s) \\ = & \sum_{s \in T_\Sigma}(r,s)\widetilde{\tau_2}(\tau_1(s)) \\ = & \sum_{s \in T_\Sigma}(r,s)(\sum_{t \in T_\Delta}(\tau_1(s),t)\tau_2(t)) \\ = & \sum_{s \in T_\Sigma}\sum_{t \in T_\Delta}((r,s)\tau_1(s),t)\tau_2(t) \\ = & \sum_{t \in T_\Delta}(\sum_{s \in T_\Sigma}(r,s)\tau_1(s),t)\tau_2(t) \\ = & \sum_{t \in T_\Delta}(\widetilde{\tau_1}(r),t)\tau_2(t) \\ = & \widetilde{\tau_2}(\widetilde{\tau_1}(r)) \\ = & (\widetilde{\tau_1} \circ \widetilde{\tau_2})(r). \end{array}$$

□

For every ranked alphabet Σ there is a particular tree series transformation $\iota_\Sigma : T_\Sigma \to A\langle\langle T_\Sigma\rangle\rangle$ defined by $\iota_\Sigma(s) = 1 \cdot s$ for every $s \in T_\Sigma$. We call ι_Σ the identity tree series transformation over Σ and A.

3 Tree series transducers

In this section we define the notion of a bottom-up tree series transducer and top-down tree series transducer which are generalizations of the notions of bottom-up tree transducers and top-down tree transducers, resp., as they occur in the literature (cf. e.g. [Eng75]). The generalization consists of allowing the transducers to compute a formal tree series as output rather than only trees. Also, a top-down tree series transducer generalizes the concept of the tree transducer as it is defined in [Kui99], because the latter can only cover the nondeterministically simple case of top-down tree transducers [GS84] (cf. Fig. 1 where going north-east along an edge means to generalize from the boolean semiring to an arbitrary semiring, and going north-west along an edge means to generalize from the nondeterministically simple case to the arbitrary nondeterministic case).

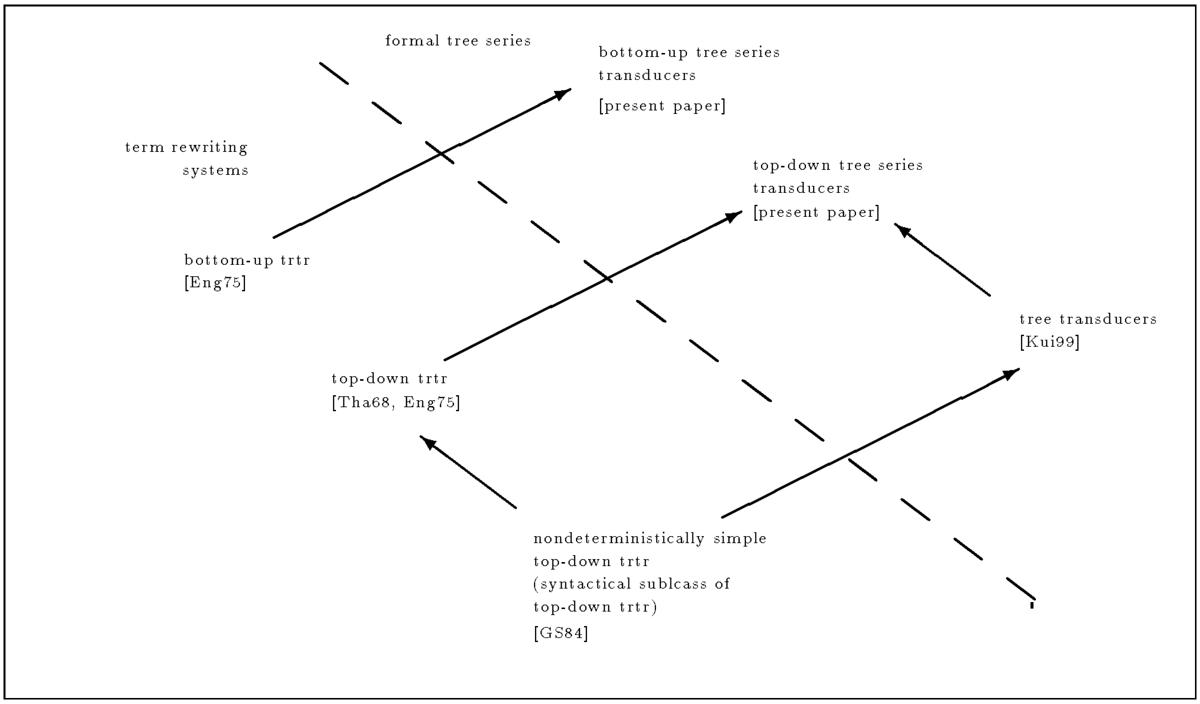

Figure 1: Generalization diagram.

In the following let Σ and Δ be two ranked alphabets, and Q be a unary ranked alphabet. Moreover, let $(A, +, \cdot, 0, 1)$ be a commutative continuous semiring.

Definition 3.1 A *tree representation (over Q, Σ, Δ, and A)* is a family $\mu = (\mu_k \mid k \geq 0)$ of mappings

$$\mu_k : \Sigma^{(k)} \to (A\langle\langle T_\Delta(X)\rangle\rangle)^{Q\times(Q(X_k))^*}$$

such that only for finitely many indices $(q, w) \in Q \times (Q(X_k))^*$, $\mu_k(\sigma)_{q,w} \neq \tilde{0}$.

A tree representation is *bottom-up* if the following additional conditions hold. For every $\sigma \in \Sigma^{(k)}$, $q \in Q$, and $w \in (Q(X_k))^*$, if $w \neq q_1(x_1)\dots q_k(x_k)$ for some $q_1, \dots, q_k \in Q$, then $\mu_k(\sigma)_{q,w} = \tilde{0}$. Moreover, if $\mu_k(\sigma)_{q,w} \neq \tilde{0}$, then $\mu_k(\sigma)_{q,w} \in A\langle\langle T_\Delta(X_k)\rangle\rangle$,

A tree representation is *top-down* if the following additional condition holds. For every $\sigma \in \Sigma^{(k)}$, $q \in Q$, and $w \in (Q(X_k))^*$, if $\mu_k(\sigma)_{q,w} \neq \tilde{0}$, then $\mu_k(\sigma)_{q,w} \in A\langle\langle \widehat{T_\Delta}(X_l)\rangle\rangle$, where $l = |w|$. □

Definition 3.2 Let μ be either a bottom-up tree representation or a top-down tree representation. For every $k \geq 0$ and $\sigma \in \Sigma^{(k)}$, μ induces the mapping

$$\overline{\mu_k(\sigma)} : (A\langle\langle T_\Delta(X)\rangle\rangle)^{Q\times 1} \times \dots \times (A\langle\langle T_\Delta(X)\rangle\rangle)^{Q\times 1} \to (A\langle\langle T_\Delta(X)\rangle\rangle)^{Q\times 1}$$

with k arguments. The mapping $\overline{\mu_k(\sigma)}$ is defined for every $P_1, \dots, P_k \in (A\langle\langle T_\Delta(X)\rangle\rangle)^{Q\times 1}$ and $q \in Q$ as:

$$\overline{\mu_k(\sigma)}(P_1, \dots, P_k)_q = \sum_{w=q_1(x_{i_1})\dots q_l(x_{i_l})\in(Q(X_k))^*} \mu_k(\sigma)_{q,w} \leftarrow ((P_{i_1})_{q_1}, \dots, (P_{i_l})_{q_l}).$$

□

Note that, for $k = 0$, $(Q(X_k))^* = \{\varepsilon\}$. Hence, according to our convention about matrix notation, in case $\sigma \in \Sigma^{(0)}$ we write $\mu_0(\sigma)_{q,\varepsilon}$ as $\mu_0(\sigma)_q$.

Since $((A\langle\langle T_\Delta(X)\rangle\rangle)^{Q\times 1}, (\overline{\mu_k(\sigma)} \mid k \geq 0, \sigma \in \Sigma^{(k)}))$ and also $((A\langle\langle T_\Delta\rangle\rangle)^{Q\times 1}, (\overline{\mu_k(\sigma)} \mid k \geq 0, \sigma \in \Sigma^{(k)}))$ are Σ-algebras, there is a unique homomorphism

$$h_\mu : T_\Sigma \to (A\langle\langle T_\Delta\rangle\rangle)^{Q\times 1}$$

defined for every $\sigma \in \Sigma^{(k)}$, $t_1, \dots, t_k \in T_\Sigma$ by

$$h_\mu(\sigma(t_1, \dots, t_k)) = \overline{\mu_k(\sigma)}(h_\mu(t_1), \dots, h_\mu(t_k)).$$

The mapping h_μ extends to $\widetilde{h_\mu} : A\langle\langle T_\Sigma\rangle\rangle \to (A\langle\langle T_\Delta\rangle\rangle)^{Q\times 1}$, for the definition see Subsection 2.7.

Definition 3.3 A *bottom-up tree series transducer (over A)* is a tuple $M = (Q, \Sigma, \Delta, A, Q_d, \mu)$ where Q is a unary ranked alphabet (of states), Σ and Δ are ranked alphabets (of input symbols and output symbols, resp.), A is a semiring, $Q_d \subseteq Q$ (set of final states), and μ is a bottom-up tree representation over Q, Σ, Δ, and A. □

Definition 3.4 A *top-down tree series transducer (over A)* is a tuple $M = (Q, \Sigma, \Delta, A, Q_d, \mu)$ where Q is a unary ranked alphabet (of states), Σ and Δ are ranked alphabets (of input symbols and output symbols, resp.), A is a semiring, $Q_d \subseteq Q$ (set of initial states), and μ is a top-down tree representation over Q, Σ, Δ, and A. □

We associate with every bottom-up (and top-down) tree series transducer M two transformations, the t-ts transformation computed by M and the ts-ts transformation computed by M. These are defined as follows.

Definition 3.5 Let $M = (Q, \Sigma, \Delta, A, Q_d, \mu)$ be a bottom-up tree series transducer or a top-down tree series transducer.

1. The *t-ts transformation computed by M* is the mapping $\tau_M : T_\Sigma \to A\langle\langle T_\Delta \rangle\rangle$ which is defined by $\tau_M(s) = \sum_{q \in Q_d} h_\mu(s)_q$ for every $s \in T_\Sigma$.
2. The *ts-ts transformation computed by M* is the extension $\widetilde{\tau_M} : A\langle\langle T_\Sigma \rangle\rangle \to A\langle\langle T_\Delta \rangle\rangle$ of τ_M, cf. Subsection 2.7. □

Note that $\widetilde{\tau_M}(s) = \sum_{q \in Q_d} \widetilde{h_\mu}(s)_q$ for every $s \in T_\Sigma$. Sometimes we use the notion of *t-ts semantics (and ts-ts semantics) of M* to speak about the t-ts transformation (and ts-ts transformation, respectively) computed by M.

Now we give an example for a bottom-up tree series transducer. In the theory of term rewriting systems one is interested among others in sufficient conditions for the property of termination (cf. e.g. [Jan97, BN98]). One such condition is expressed in the following implication. Let R be a term rewriting system, then R is terminating if, for every derivation $t_1 \Rightarrow_R t_2 \Rightarrow_R t_3 \ldots$, there is no pair i, j with $i < j$ such that t_i can be embedded homeomorphically into t_j. For two trees s and t, *t can be embedded homeomorphically into s* (for short: $s \geq_{emb} t$) if t is obtained from s by dropping an arbitrary number of occurrences of symbols in s (cf. Definition 5.4.2 of [BN98]).

Example 3.6 For the particular ranked alphabet $\Sigma = \{\sigma^{(2)}, \gamma^{(1)}, \alpha^{(0)}\}$ and $s, t \in T_\Sigma$, we have $s \geq_{emb} t$ if t is obtained from s by dropping an arbitrary number of occurrences of σ's or γ's.

The relation $\geq_{emb}$, as tree transformation, can be computed by the bottom-up tree transducer $M_{drop} = (Q, \Sigma, \Sigma, Q_d, R_{drop})$, i.e., $\tau_{M_{drop}} = \geq_{emb}$, where $Q = Q_d = \{*\}$ and R_{drop} consists of the following rules:

$$\begin{array}{rcl}
\sigma(*(x_1), *(x_2)) & \to & *(\sigma(x_1, x_2)) \\
\sigma(*(x_1), *(x_2)) & \to & *(x_1) \\
\sigma(*(x_1), *(x_2)) & \to & *(x_2) \\
\gamma(*(x_1)) & \to & *(\gamma(x_1)) \\
\gamma(*(x_1)) & \to & *(x_1) \\
\alpha & \to & *(\alpha).
\end{array}$$

Clearly, for a given tree $s \in T_\Sigma$ there are (finitely) many trees $t \in T_\Sigma$ such that $s \geq_{emb} t$. Also it is clear that, for a given tree $t \in T_\Sigma$, there are (infinitely) many trees $s \in T_\Sigma$ such that $s \geq_{emb} t$.

Now we might be interested in the following question: given two trees $s, t \in T_\Sigma$, how many possibilities are there to embed t into s? For instance, consider the trees $s = \sigma(\alpha, \gamma(\gamma(\alpha)))$ and $t = \sigma(\alpha, \gamma(\alpha))$. Hence, $s \geq_{emb} t$. One way of obtaining t from s is to drop the γ at occurrence 2 of s, another way to obtain t is to drop the γ at occurrence 21. As another example, consider $t' = s$. Then $s \geq_{emb} t'$ and there is one way of obtaining t' from s, viz. dropping no occurrence of a symbol. Finally, consider $t'' = \sigma(\gamma(\alpha), \alpha)$. Clearly, $s \not\geq_{emb} t''$ and hence, the number of different ways in which t'' can be obtained from s is zero.

Thus, we define the t-ts transformation $\#emb : T_\Sigma \to Nat\langle\langle T_\Sigma \rangle\rangle$ such that, for every $s, t \in T_\Sigma$,

$$(\#emb(s), t) = \begin{cases} n & \text{if } s \geq_{emb} t \text{ and there are } n \text{ different ways to embed } t \text{ into } s \\ 0 & \text{otherwise} \end{cases}$$

Note that, $(\#emb(s), t) \neq 0$ iff $s \geq_{emb} t$. Thus, e.g.,

$$\#emb(\sigma(\alpha, \gamma(\gamma(\alpha)))) = \sigma(\alpha, \gamma(\gamma(\alpha))) \hat{+} 2\sigma(\alpha, \gamma(\alpha)) \hat{+} \sigma(\alpha, \alpha) \hat{+} \gamma(\gamma(\alpha)) \hat{+} 2\gamma(\alpha) \hat{+} 2\alpha.$$

Now, based on M_{drop}, we construct a bottom-up tree series transducer $M^{\#}_{drop} = (Q, \Sigma, \Sigma, Nat, Q_d, \mu)$ such that $\tau_{M^{\#}_{drop}} = \#emb$, where μ is defined as follows.

$$\begin{aligned}
(\mu)_2(\sigma)_{*,*(x_1)*(x_2)} &= \sigma(x_1, x_2) \hat{+} x_1 \hat{+} x_2 \\
(\mu)_1(\gamma)_{*,*(x_1)} &= \gamma(x_1) \hat{+} x_1 \\
(\mu)_0(\alpha)_{*,\varepsilon} &= \alpha
\end{aligned}$$

Now let us compute $\tau_{M^{\#}_{drop}}(s)$ for our example tree $s = \sigma(\alpha, \gamma(\gamma(\alpha)))$.

$\tau_{M^{\#}_{\gamma}}(s) = \sum_{q \in Q_d} h_\mu(s)_q = \sum_{q \in \{*\}} h_\mu(s)_q = h_\mu(s)_*$. Since h_μ is defined by induction on the structure of its argument, we first compute $h_\mu(\alpha)_*$, second $h_\mu(\gamma(\alpha))_*$, third $h_\mu(\gamma(\gamma(\alpha)))_*$, and finally $h_\mu(\sigma(\alpha, \gamma(\gamma(\alpha))))_*$.

$$\begin{aligned}
& h_\mu(\alpha)_* \\
= \;& \overline{\mu_0(\alpha)}()_* \\
= \;& \textstyle\sum_{w=\varepsilon} \mu_0(\alpha)_{*,w} \leftarrow () \\
= \;& \mu_0(\alpha)_{*,\varepsilon} \leftarrow () \\
= \;& \alpha
\end{aligned}$$

$$\begin{aligned}
& h_\mu(\gamma(\alpha))_* \\
= \;& \overline{\mu_1(\gamma)}(h_\mu(\alpha))_* \\
= \;& \textstyle\sum_{w=*(x_1)} \mu_1(\gamma)_{*,w} \leftarrow (h_\mu(\alpha)_*) \\
= \;& \mu_0(\gamma)_{*,*(x_1)} \leftarrow (h_\mu(\alpha)_*) \\
= \;& (\gamma(x_1) \hat{+} x_1) \leftarrow (\alpha) \\
= \;& \textstyle\sum_{t \in T_\Sigma(X_1)} (\gamma(x_1) \hat{+} x_1, t)(t \leftarrow (\alpha)) \\
= \;& (\gamma(x_1) \leftarrow (\alpha)) \;+\; (x_1 \leftarrow (\alpha)) \\
= \;& (\hat{\textstyle\sum}_{s \in T_\Sigma(X)} (\alpha, s)(\gamma(x_1)[s])) \;+\; (\hat{\textstyle\sum}_{s \in T_\Sigma(X)} (\alpha, s)(x_1[s])) \\
= \;& \gamma(\alpha) \hat{+} \alpha
\end{aligned}$$

$$\begin{aligned}
& h_\mu(\gamma(\gamma(\alpha)))_* \\
= \;& \ldots \\
= \;& (\gamma(x_1) \hat{+} x_1) \leftarrow (\gamma(\alpha) \hat{+} \alpha) \\
= \;& \textstyle\sum_{t \in T_\Sigma(X_1)} (\gamma(x_1) \hat{+} x_1, t)(t \leftarrow (\gamma(\alpha) \hat{+} \alpha)) \\
= \;& (\gamma(x_1) \leftarrow (\gamma(\alpha) \hat{+} \alpha)) \;+\; (x_1 \leftarrow (\gamma(\alpha) \hat{+} \alpha)) \\
= \;& (\hat{\textstyle\sum}_{s \in T_\Sigma(X)} (\gamma(\alpha) \hat{+} \alpha, s)(\gamma(x_1)[s])) \;+\; (\hat{\textstyle\sum}_{s \in T_\Sigma(X)} (\gamma(\alpha) \hat{+} \alpha, s)(x_1[s])) \\
= \;& (\gamma(\gamma(\alpha)) \hat{+} \gamma\alpha) + (\gamma(\alpha) \hat{+} \alpha) \\
= \;& \gamma(\gamma(\alpha)) \hat{+} 2\gamma(\alpha) \hat{+} \alpha
\end{aligned}$$

$$\begin{aligned}
& h_\mu(\sigma(\alpha, \gamma(\gamma(\alpha))))_* \\
= \;& \overline{\mu_2(\sigma)}(h_\mu(\alpha), h_\mu(\gamma(\gamma(\alpha))))_* \\
= \;& \textstyle\sum_{w=*(x_1)*(x_2)} \mu_2(\sigma)_{*,w} \leftarrow (h_\mu(\alpha)_*, h_\mu(\gamma(\gamma(\alpha)))_*) \\
= \;& \mu_2(\sigma)_{*,w} \leftarrow (h_\mu(\alpha)_*, h_\mu(\gamma(\gamma(\alpha)))_*) \\
= \;& (\sigma(x_1, x_2) \hat{+} x_1 \hat{+} x_2) \leftarrow (\alpha, \gamma(\gamma(\alpha)) \hat{+} 2\gamma(\alpha) \hat{+} \alpha) \\
= \;& \textstyle\sum_{t \in T_\Sigma(X_2)} (\sigma(x_1, x_2) \hat{+} x_1 \hat{+} x_2, t)(t \leftarrow (\alpha, \gamma(\gamma(\alpha)) \hat{+} 2\gamma(\alpha) \hat{+} \alpha))
\end{aligned}$$

$$
\begin{aligned}
&= (\sigma(x_1,x_2) \leftarrow (\alpha, \gamma(\gamma(\alpha)) \hat{+} 2\gamma(\alpha) \hat{+} \alpha)) \\
&\quad + (x_1 \leftarrow (\alpha, \gamma(\gamma(\alpha)) \hat{+} 2\gamma(\alpha) \hat{+} \alpha)) \\
&\quad + (x_2 \leftarrow (\alpha, \gamma(\gamma(\alpha)) \hat{+} 2\gamma(\alpha) \hat{+} \alpha)) \\
&= \hat{\sum}_{s_1,s_2 \in T_\Sigma(X)} (\alpha, s_1)(\gamma(\gamma(\alpha)) \hat{+} 2\gamma(\alpha) \hat{+} \alpha, s_2)(\sigma(x_1,x_2)[s_1,s_2]) \\
&\quad + \hat{\sum}_{s_1,s_2 \in T_\Sigma(X)} (\alpha, s_1)(\gamma(\gamma(\alpha)) \hat{+} 2\gamma(\alpha) \hat{+} \alpha, s_2)(x_1[s_1,s_2]) \\
&\quad + \hat{\sum}_{s_1,s_2 \in T_\Sigma(X)} (\alpha, s_1)(\gamma(\gamma(\alpha)) \hat{+} 2\gamma(\alpha) \hat{+} \alpha, s_2)(x_2[s_1,s_2]) \\
&= (\sigma(\alpha, \gamma(\gamma(\alpha)) \hat{+} 2\sigma(\alpha, \gamma(\alpha)) \hat{+} \sigma(\alpha, \alpha)) \\
&\quad + \alpha \\
&\quad + (\gamma(\gamma(\alpha)) \hat{+} 2\gamma(\alpha) \hat{+} \alpha) \\
&= \sigma(\alpha, \gamma(\gamma(\alpha))) \hat{+} 2\sigma(\alpha, \gamma(\alpha)) \hat{+} \sigma(\alpha, \alpha) \hat{+} \gamma(\gamma(\alpha)) \hat{+} 2\gamma(\alpha) \hat{+} 2\alpha
\end{aligned}
$$

Thus, we have verified that $\tau_{M^{\#}_{drop}}(s) = \#emb(s)$ for the tree $s = \sigma(\alpha, \gamma(\gamma(\alpha)))$. □

Next we define some restricted versions of bottom-up tree series transducers. These are the same restrictions which are imposed on tree transducers, see e.g. [Eng75, Bak79, GS84].

Definition 3.7 Let $M = (Q, \Sigma, \Delta, A, Q_d, \mu)$ be a bottom-up tree series transducer.

1. M is *deterministic* if for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $q_1, \ldots, q_k \in Q$, there is at most one $q \in Q$, such that $\mu_k(\sigma)_{q,w} \neq \tilde{0}$ where $w = q_1(x_1) \ldots q_k(x_k)$, and if $\mu_k(\sigma)_{q,w} \neq \tilde{0}$ for some q and w, then $\mu_k(\sigma)_{q,w}$ is a singleton.
2. M is *total* if $Q_d = Q$ and for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $q_1, \ldots, q_k \in Q$, there is at least one $q \in Q$ such that $\mu_k(\sigma)_{q,w} \neq \tilde{0}$.
3. M is a *bottom-up homomorphism tree series transducer* if it is total deterministic and the set Q of states is a singleton.
4. M is *linear* if for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $q, q_1, \ldots, q_k \in Q$, every $t \in supp(\mu_k(\sigma)_{q,q_1(x_1)\ldots q_k(x_k)})$ is linear.
5. M is a *bottom-up finite state relabeling tree series transducer* if for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$ and $q, q_1, \ldots, q_k \in Q$, every $t \in supp(\mu_k(\sigma)_{q,q_1(x_1)\ldots q_k(x_k)})$ has the form $\delta(x_1, \ldots, x_k)$ for some $\delta \in \Delta^{(k)}$.
6. M is *polynomial* if for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $q, q_1, \ldots, q_k \in Q$, $\mu_k(\sigma)_{q,q_1(x_1)\ldots q_k(x_k)} \in A\langle T_\Delta(X)\rangle$.

□

Notice that bottom-up finite state relabeling tree series transducers are polynomial.

The *class of t-ts transformations induced by bottom-up tree series transducers* is denoted by BOT_{t-ts}. The classes of t-ts transformations which correspond to the syntactic subclasses of Definition 3.7 1.-6. are denoted by *d-*BOT_{t-ts}, *t-*BOT_{t-ts}, *h-*BOT_{t-ts}, *l-*BOT_{t-ts}, *r-*BOT_{t-ts}, and *p-*BOT_{t-ts}, resp. If more than one syntactic restriction holds, then BOT_{t-ts} is prefixed by the string of corresponding letters, e.g. *ltd-*BOT_{t-ts} denotes the class of t-ts transformations induced by linear, total, deterministic bottom-up tree series transducers. If all the considered bottom-up tree series transducers are over the same semiring A, then the corresponding class of t-ts transformations is indexed by A as e.g. in *ltd-*$BOT_{t-ts}(A)$. The *class of ts-ts transformations induced by bottom-up tree series transducers* is denoted by BOT_{ts-ts}. The denotation of the classes of ts-ts transformations computed by subclasses of bottom-up tree series transducers is obtained from the denotation of the corresponding classes of t-ts transformations by replacing $t - ts$ by $ts - ts$.

Note that the bottom-up tree series transducer $M^{\#}_{drop}$ considered in Example 3.6 is in fact a bottom-up homomorphism tree series transducer. In what follows, like in case of $M^{\#}_{drop}$, we will denote the only state of a homomorphism tree series transducer by $*$.

Now we introduce the corresponding restrictions for top-down tree series transducers.

Definition 3.8 Let $M = (Q, \Sigma, \Delta, A, Q_d, \mu)$ be a top-down tree series transducer.

1. M is *deterministic* if the following conditions hold. Q_d is a singleton. For every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $q \in Q$, there is at most one $w \in (Q(X_k))^*$ such that $\mu_k(\sigma)_{q,w} \neq \tilde{0}$, and if $\mu_k(\sigma)_{q,w} \neq \tilde{0}$ for some q and w, then $\mu_k(\sigma)_{q,w}$ is a singleton.
2. M is *total* if for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $q \in Q$, there is at least one $w \in (Q(X_k))^*$ such that $\mu_k(\sigma)_{q,w} \neq \tilde{0}$, and $Q_d \neq \emptyset$.

3. M is a *top-down homomorphism tree series transducer* if it is total deterministic and the set Q of states is a singleton.

4. M is *linear* if for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $q \in Q$, $w \in (Q(X_k))^*$, if $\mu_k(\sigma)_{q,w} \neq \tilde{0}$, then w is linear.

5. M is a *top-down finite state relabeling tree series transducer* if for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $q \in Q$, $w \in (Q(X_k))^*$, if $\mu_k(\sigma)_{q,w} \neq \tilde{0}$, then $w = q_1(x_1)\ldots q_k(x_k)$ and every $t \in supp(\mu_k(\sigma)_{q,w})$ has the form $\delta(x_1,\ldots,x_k)$ for some $\delta \in \Delta^{(k)}$.

6. M is *polynomial* if for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $q \in Q$, $w \in (Q(X_k))^*$, if $\mu_k(\sigma)_{q,w} \in A\langle T_\Delta(X)\rangle$. □

Again like in the bottom-up case, top-down finite state relabeling tree series transducers are polynomial.

The *class of t-ts transformations induced by top-down tree series transducers* is denoted by TOP_{t-ts}. The classes of t-ts transformations which correspond to the syntactic subclasses of Definition 3.8 1.-6. are denoted by *d-*TOP_{t-ts}, *t-*TOP_{t-ts}, *h-*TOP_{t-ts}, *l-*TOP_{t-ts}, *r-*TOP_{t-ts}, and *p-*TOP_{t-ts}, resp. For combinations of syntactic restrictions and for the restriction to one semiring A we use a notation similar to that of the bottom-up case. The *class of ts-ts transformations induced by top-down tree series transducers* is denoted by TOP_{ts-ts} and also here we use the same notation system as in the bottom-up case.

We note that a tree transducer in the sense of [Kui97a] is a top-down tree series transducer in which the elements μ_k of the tree representation μ have the particular form

$$\mu_k : \Sigma_k \longrightarrow (A\langle\langle T_\Delta(X_k)\rangle\rangle)^{Q\times Q_k}$$

where $Q_k = Q(\{x_1\}) \times \ldots \times Q(\{x_k\})$.

Finally, we show that, like in case of conventional tree transducers, the difference between the bottom-up and the top-down approaches disappears for certain basic classes of tree series transducers (cf. Lemma 3.2 of [Eng75] and the remark after Definition 3.14 of [Eng75]).

Lemma 3.9 1) *h-*TOP_{t-ts} = *h-*BOT_{t-ts} and *h-*TOP_{ts-ts} = *h-*BOT_{ts-ts}.
2) *r-*TOP_{t-ts} = *r-*BOT_{t-ts} and *r-*TOP_{ts-ts} = *r-*BOT_{ts-ts}

Proof. First we show that 1) holds. We prove only the first statement because the second one follows immediately from the definition of the ts-ts semantics..

Let $M = (\{*\},\Sigma,\Delta,A,\{*\},\mu)$ be a top-down homomorphism tree series transducer. We construct the bottom-up homomorphism tree series transducer $M' = (\{*\},\Sigma,\Delta,A,\{*\},\mu')$ in the following way. For every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, and $t \in T_\Sigma(X_k)$, let $lin_X(t) = (t', x_{i_1}\ldots x_{i_l})$. Now let us define $(\mu'_k(\sigma)_{*,*(x_1)\ldots*(x_k)}, t) = (\mu_k(\sigma)_{*,*(x_{i_1})\ldots*(x_{i_l})}, t')$. It is easy to see that $\tau_M = \tau_{M'}$, which proves the inclusion *h-*$TOP_{t-ts} \subseteq$ *h-*BOT_{t-ts}. To show the converse of the inclusion, let us take a bottom-up homomorphism tree series transducer $M = (\{*\},\Sigma,\Delta,A,\{*\},\mu)$. Now we construct the top-down homomorphism tree series transducer $M' = (\{*\},\Sigma,\Delta,A,\{*\},\mu')$ as follows. For every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, sequence $*(x_{i_1})\ldots*(x_{i_l})$ where $1 \leq i_1,\ldots,i_l \leq k$, and tree $t' \in \widehat{T_\Delta}(X_l)$, let $t = t'[x_1 \leftarrow x_{i_1},\ldots,x_l \leftarrow x_{i_l}]$. Now let us define $(\mu'_k(\sigma)_{*,*(x_{i_1})\ldots*(x_{i_l})}, t') = (\mu_k(\sigma)_{*,*(x_1)\ldots*(x_k)}, t)$. Again it is easy to see that $\tau_M = \tau_{M'}$, which proves *h-*$BOT_{t-ts} \subseteq$ *h-*TOP_{t-ts}.

The proof of 2) is left to the reader, cf. the corresponding statement for tree transducers in [Eng75]. □

From now on we call a bottom-up homomorphism tree series transducers as well as a top-down homomorphism tree series transducer a *homomorphism tree series transducers*. Similarly we drop the notions "bottom-up" and "top-down" from bottom-up and top-down finite state relabeling tree series transducers. Let us denote the classes *h-*TOP_{t-ts} and *r-*TOP_{t-ts} (and *h-*TOP_{ts-ts} and *r-*TOP_{ts-ts}) by HOM_{t-ts} and $QREL_{t-ts}$ (HOM_{ts-ts} and $QREL_{ts-ts}$), respectively.

It is an exercise to show that for every ranked alphabet Σ and semiring A ι_Σ is in any of the classes of tree to tree series transformations defined in this section. We will use this fact in Section 5.

4 Tree series transducers over $\mathbb{B}$

In this section we show that, under the t-ts semantics, every bottom-up tree transducer can be simulated by a polynomial bottom-up tree series transducer over $\mathbb{B}$ and vice versa, every polynomial bottom-up tree series transducer over $\mathbb{B}$ can be simulated by a bottom-up tree transducer. The same correspondence holds for the top-down case.

4.1 The bottom-up case

Definition 4.1 Let $M = (Q, \Sigma, \Delta, Q_d, R)$ be a bottom-up tree transducer and $M' = (Q, \Sigma, \Delta, \mathbb{B}, Q_d, \mu)$ be a polynomial bottom-up tree series transducer over $\mathbb{B}$.

M and M' are *related* if the following equivalence is true: for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $t \in T_\Delta(X_k)$, and $q, q_1, \ldots, q_k \in Q$,

$$(\mu_k(\sigma)_{q,q_1(x_1)\ldots q_k(x_k)}, t) = 1 \text{ iff there is a rule } \sigma(q_1(x_1), \ldots, q_k(x_k)) \to q(t) \text{ in } R.$$

$\square$

Lemma 4.2 Let M be a bottom-up tree transducer with Σ and Δ as input and as output ranked alphabet, respectively, and M' be a polynomial bottom-up tree series transducer such that M and M' are related. Then, $\tau_M = \tau_{M'} \circ pick_\Delta$.

Proof. Let M and M' be specified as in Definition 4.1. First, we can prove by induction on s that, for every $s \in T_\Sigma$, $t \in T_\Delta$, and $q \in Q$ the following equivalence holds: $(h_\mu(s)_q, t) = 1$ iff $s \Rightarrow_M^* q(t)$.

$\underline{s = \sigma(s_1, \ldots, s_k) \text{ with } k \geq 1:}$

$$
\begin{array}{ll}
 & s \Rightarrow_M^* q(t) \\
\text{iff} & \sigma(s_1, \ldots, s_k) \Rightarrow_M^* q(t) \\
\text{iff} & \text{there are } q_1, \ldots, q_k \in Q \text{ and } t_1, \ldots, t_k \in T_\Delta \text{ such that} \\
 & \sigma(s_1, \ldots, s_k) \Rightarrow_M^* \sigma(q_1(t_1), \ldots, q_k(t_k)) \Rightarrow_M q(t) \\
\text{iff} & \text{there are } q_1, \ldots, q_k \in Q,\ t_1, \ldots, t_k \in T_\Delta,\ t' \in T_\Delta(X_k), \\
 & \text{and there is a rule } \sigma(q_1(x_1), \ldots, q_k(x_k)) \to q(t') \text{ in } R \text{ such that} \\
 & s_i \Rightarrow_M^* q_i(t_i) \text{ and } t = t'[t_1, \ldots, t_k] \\
\text{iff} & \text{there are } q_1, \ldots, q_k \in Q,\ t_1, \ldots, t_k \in T_\Delta,\ t' \in T_\Delta(X_k) \text{ such that} \\
 & (\mu_k(\sigma)_{q,q_1(x_1)\ldots q_k(x_k)}, t') = 1 \text{ and } (h_\mu(s_i)_{q_i}, t_i) = 1 \text{ and } t = t'[t_1, \ldots, t_k] \\
\text{iff} & ((\sum_{w=q_1(x_1)\ldots q_k(x_k)} \sum_{t' \in T_\Delta(X_k)} \\
 & (\mu_k(\sigma)_{q,w}, t')(\sum_{t_1,\ldots,t_k \in T_\Delta} (h_\mu(s_1)_{q_1}, t_1) \ldots (h_\mu(s_k)_{q_k}, t_k) t'[t_1, \ldots, t_k]))_q, t) = 1 \\
\text{iff} & ((\sum_{w=q_1(x_1)\ldots q_k(x_k)} \sum_{t' \in T_\Delta(X_k)} \\
 & (\mu_k(\sigma)_{q,w}, t')(t' \leftarrow (h_\mu(s_1)_{q_1}, \ldots, h_\mu(s_k)_{q_k})))_q, t) = 1 \\
\text{iff} & ((\sum_{w=q_1(x_1)\ldots q_k(x_k)} \mu_k(\sigma)_{q,w} \leftarrow (h_\mu(s_1)_{q_1}, \ldots, h_\mu(s_k)_{q_k}))_q, t) = 1 \\
\text{iff} & (\mu_k(\sigma)(h_\mu(s_1), \ldots, h_\mu(s_k))_q, t) = 1 \\
\text{iff} & (h_\mu(s)_q, t) = 1.
\end{array}
$$

Second, we can prove that $\tau_M = \tau_{M'} \circ pick_\Delta$. Let $s \in T_\Sigma$ and $t \in T_\Delta$.

$$
\begin{array}{ll}
 & (s, t) \in \tau_M \\
\text{iff} & \exists q \in Q_d : s \Rightarrow_M^* q(t) \\
\text{iff} & \exists q \in Q_d : (h_\mu(s)_q, t) = 1 \\
\text{iff} & \exists q \in Q_d : (h_\mu(s)_q, t) \in pick_\Delta \\
\text{iff} & (\sum_{q \in Q_d} h_\mu(s)_q, t) \in pick_\Delta \\
\text{iff} & (\tau_{M'}(s), t) \in pick_\Delta \\
\text{iff} & (s, t) \in \tau_{M'} \circ pick_\Delta \qquad \square
\end{array}
$$

Lemma 4.3 For every bottom-up tree transducer M there is a polynomial bottom-up tree series transducer M' such that M and M' are related.

Proof. Let $M = (Q, \Sigma, \Delta, Q_d, R)$ be a bottom-up tree transducer. Construct the bottom-up tree series transducer $M' = (Q, \Sigma, \Delta, \mathbb{B}, Q_d, \mu)$ with the bottom-up tree representation μ over Q, Σ, Δ, and $\mathbb{B}$ as follows: for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $t \in T_\Delta(X_k)$, and $q, q_1, \ldots, q_k \in Q$, $(\mu_k(\sigma)_{q,q_1(x_1)\ldots q_k(x_k)}, t) = 1$ if there is a rule $\sigma(q_1(x_1), \ldots, q_k(x_k)) \to q(t)$ in R, and $(\mu_k(\sigma)_{q,q_1(x_1)\ldots q_k(x_k)}, t) = 0$ if there is no such rule. Obviously, M' is a polynomial, and M and M' are related. $\square$

Example 4.4 (Cf. Example 2.1 of [Eng75].) Consider the bottom-up tree transducer $M = (Q, \Sigma, \Delta, Q_d, R)$ where $Q = \{q\}$, $\Sigma = \{\bar{\gamma}^{(1)}, \gamma^{(1)}, \alpha^{(0)}\}$, $\Delta = \{\sigma^{(2)}, \gamma_1^{(1)}, \gamma_2^{(1)}, \alpha^{(0)}\}$, $Q_d = \{q\}$, and R consists of the following rules:

$$
\begin{array}{rcl}
\bar{\gamma}(q(x_1)) & \rightarrow & q(\sigma(x_1, x_1)) \\
\gamma(q(x_1)) & \rightarrow & q(\gamma_1(x_1)) \\
\gamma(q(x_1)) & \rightarrow & q(\gamma_2(x_1)) \\
\alpha & \rightarrow & q(\alpha)
\end{array}
$$

This is an example of Property (B1) in [Eng75]: a bottom-up tree transducer is able to process a subtree of an input tree in a nondeterministic way and copy the results of this process.

Using Lemma 4.3 we construct the bottom-up tree series transducer $M' = (Q, \Sigma, \Delta, \mathbb{B}, Q_d, \mu)$ as follows:

$$
\begin{array}{rcl}
\mu_1(\bar{\gamma})_{q,q(x_1)} & = & \sigma(x_1, x_1) \\
\mu_1(\gamma)_{q,q(x_1)} & = & \gamma_1(x_1) \hat{+} \gamma_2(x_1) \\
\mu_0(\alpha)_{q,\varepsilon} & = & \alpha
\end{array}
$$

Then, according to Lemmas 4.3 and 4.2, M' is the bottom-up tree series transducer such that $\tau_M = \tau_{M'} \circ pick_\Delta$. Note that M' is not deterministic, hence is not a homomorphism tree series transducer.

Let us compute now in detail $h_\mu(\alpha)$, $h_\mu(\gamma\alpha)$, and $h_\mu(\bar{\gamma}\gamma\alpha)$, and eventually $\tau_{M'}(r)$ for the polynomial formal tree series $r = \bar{\gamma}\gamma\alpha$.

$$h_\mu(\alpha)_q = \overline{\mu_0}(\alpha)()_q = \mu_0(\alpha)_{q,\varepsilon} = \alpha$$

$$
\begin{array}{l}
h_\mu(\gamma\alpha)_q = \\
= \overline{\mu_1(\gamma)}(h_\mu(\alpha))_q \\
= \sum_{w = p_1(x_{i_1}) \dots p_l(x_{i_l}) \in Q(\{x_1\})^*} \mu_1(\gamma)_{q,w} \leftarrow (h_\mu(\alpha)_{p_1}, \dots, h_\mu(\alpha)_{p_l}) \\
= \mu_1(\gamma)_{q,q(x_1)} \leftarrow (h_\mu(\alpha)_q) \\
= (\gamma_1(x_1) \hat{+} \gamma_2(x_1)) \leftarrow (\alpha) \\
= \gamma_1(\alpha) \hat{+} \gamma_2(\alpha).
\end{array}
$$

$$
\begin{array}{l}
h_\mu(\bar{\gamma}\gamma\alpha)_q \\
= \overline{\mu_1(\bar{\gamma})}(h_\mu(\gamma\alpha))_q \\
= \mu_1(\bar{\gamma})_{q,q(x_1)} \leftarrow (h_\mu(\gamma\alpha)_q) \\
= \sigma(x_1, x_1) \leftarrow (\gamma_1(\alpha) \hat{+} \gamma_2(\alpha)) \\
= \hat{\sum}_{t_1 \in T_\Delta(X)} (\gamma_1(\alpha) \hat{+} \gamma_2(\alpha), t_1) \sigma(x_1, x_1)[t_1] \\
= \sigma(\gamma_1\alpha, \gamma_1\alpha) \hat{+} \sigma(\gamma_2\alpha, \gamma_2\alpha).
\end{array}
$$

Finally,

$$
\begin{array}{l}
\tau_{M'}(\bar{\gamma}\gamma\alpha) \\
= \sum_{q \in Q_d} h_\mu(\bar{\gamma}\gamma\alpha)_q \\
= h_\mu(\bar{\gamma}\gamma\alpha)_q \\
= \sigma(\gamma_1\alpha, \gamma_1\alpha) \hat{+} \sigma(\gamma_2\alpha, \gamma_2\alpha).
\end{array}
$$

□

Lemma 4.5 For every polynomial bottom-up tree series transducer M over $\mathbb{B}$ there is a bottom-up tree transducer M' such that M' and M are related.

Proof. Let $M = (Q, \Sigma, \Delta, \mathbb{B}, Q_d, \mu)$ be a polynomial bottom-up tree series transducer. Construct the bottom-up tree transducer $M' = (Q, \Sigma, \Delta, Q_d, R)$ as follows. For every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $t \in T_\Delta(X_k)$, and $q, q_1, \dots, q_k \in Q$, the rule $\sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(t)$ is in R if $(\mu_k(\sigma)_{q,q_1(x_1)\dots q_k(x_k)}, t) = 1$. Note that, since M is polynomial, R is a finite set. Obviously, M and M' are related. □

From Lemmas 4.2, 4.3, and 4.5 immediately a characterization of bottom-up tree transducers in terms of polynomial bottom-up tree series transducers follows.

Theorem 4.6 *p-$BOT_{t-ts}(\mathbb{B}) \circ PICK = BOT_{tt}$.*

4.2 The top-down case

Next we show that top-down tree transducers (as they occur in the literature) are characterized by polynomial top-down tree series transducers over the boolean semiring $\mathbb{B}$.

Definition 4.7 Let $M = (Q, \Sigma, \Delta, Q_d, R)$ be a top-down tree transducer and let $M' = (Q, \Sigma, \Delta, \mathbb{B}, Q_d, \mu)$ be a polynomial top-down tree series transducer over $\mathbb{B}$.

M and M' are *related* if the following equivalence is true: for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $q \in Q$, $w \in (Q(X_k))^*$, $t' \in \widehat{T_\Delta}(X_l)$ where $l = |w|$,

$$(\mu_k(\sigma)_{q,w}, t') = 1 \text{ iff there is a rule } q(\sigma(x_1, \ldots, x_k)) \to t \text{ in } R \text{ such that } lin_{Q(X)}(t) = (t', w).$$

□

Lemma 4.8 Let M be a top-down tree transducer with Σ and Δ as input and as output ranked alphabets, respectively, and M' be a polynomial top-down tree series transducer such that M and M' are related. Then, $\tau_M = \tau_{M'} \circ pick_\Delta$ holds.

Proof. Let M and M' be specified as in Definition 4.7. We can prove that, for every $s \in T_\Sigma$, $t \in T_\Delta$, and $q \in Q$ the following equivalence holds: $(h_\mu(s)_q, t) = 1$ iff $q(s) \Rightarrow_M^* t$.

$\underline{s = \sigma(s_1, \ldots, s_k) \text{ with } k \geq 1}$:

$$\begin{array}{ll}
 & q(s) \Rightarrow_M^* t \\
\text{iff} & \text{there are } q_1, \ldots, q_l \in Q,\ i_1, \ldots, i_l \in \{1, \ldots, k\},\ t_1, \ldots, t_l \in T_\Delta, \text{ and } t' \in \widehat{T_\Delta}(X_l) \text{ such that} \\
 & q(\sigma(s_1, \ldots, s_k)) \Rightarrow_M t'[q_1(s_{i_1}), \ldots, q_l(s_{i_l})] \Rightarrow_M^* t'[t_1, \ldots, t_l] = t \\
\text{iff} & \text{there are } q_1, \ldots, q_l \in Q,\ i_1, \ldots, i_l \in \{1, \ldots, k\},\ t_1, \ldots, t_l \in T_\Delta, \text{ and } t' \in \widehat{T_\Delta}(X_l) \text{ such that} \\
 & q(\sigma(x_1, \ldots, x_k)) \to t'[q_1(x_{i_1}), \ldots, q_l(x_{i_l})] \text{ is in } R \text{ and } q_j(s_{i_j}) \Rightarrow_M^* t_j \text{ and } t'[t_1, \ldots, t_l] = t \\
\text{iff}^* & \text{there are } q_1, \ldots, q_l \in Q,\ i_1, \ldots, i_l \in \{1, \ldots, k\},\ t_1, \ldots, t_l \in T_\Delta, \text{ and } t' \in \widehat{T_\Delta}(X_l) \text{ such that} \\
 & (\mu_k(\sigma)_{q, q_1(x_{i_1}) \ldots q_l(x_{i_l})}, t') = 1 \text{ and } (h_\mu(s_{i_j})_{q_j}, t_j) = 1 \text{ and } t'[t_1, \ldots, t_l] = t \\
\text{iff} & (\sum_{w = q_1(x_{i_1}) \ldots q_l(x_{i_l})} \sum_{t' \in T_\Delta(X_l)} (\mu_k(\sigma)_{q,w}, t') \\
 & \hat{\sum}_{t_1, \ldots, t_l \in T_\Delta(X)} (h_\mu(s_{i_1})_{q_1}, t_1) \ldots (h_\mu(s_{i_l})_{q_l}, t_l) t'[t_1, \ldots, t_l], t) = 1 \\
\text{iff} & (\sum_{w = q_1(x_{i_1}) \ldots q_l(x_{i_l})} \sum_{t' \in T_\Delta(X_l)} (\mu_k(\sigma)_{q,w}, t')(t' \leftarrow (h_\mu(s_{i_1})_{q_1}, \ldots, h_\mu(s_{i_l})_{q_l}), t) = 1 \\
\text{iff} & (\sum_{w = q_1(x_{i_1}) \ldots q_l(x_{i_l})} \mu_k(\sigma)_{q,w} \leftarrow (h_\mu(s_{i_1})_{q_1}, \ldots, h_\mu(s_{i_l})_{q_l}), t) = 1 \\
\text{iff} & (\overline{\mu_k(\sigma)}(h_\mu(s_1), \ldots, h_\mu(s_k))_q, t) = 1 \\
\text{iff} & (h_\mu(s)_q, t) = 1.
\end{array}$$

At the step which is marked by * we have used the fact that the top-down tree representation $(\mu_k(\sigma)_{q, q_1(x_{i_1}) \ldots q_l(x_{i_l})}, t') = 0$ if $t' \notin \widehat{T_\Delta}(X_l)$. The equation $\tau_M = \tau_{M'} \circ pick_\Delta$ can be verified in the same way as in the proof of Lemma 4.2. □

Lemma 4.9 For every top-down tree transducer M there is a polynomial top-down tree series transducer M' such that M and M' are related.

Proof. Let $M = (Q, \Sigma, \Delta, Q_d, R)$ be a top-down tree transducer. Construct the top-down tree series transducer $M' = (Q, \Sigma, \Delta, \mathbb{B}, Q_d, \mu)$ with the top-down tree representation μ over Q, Σ, Δ, and $\mathbb{B}$ as follows: for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $q \in Q$, $w \in (Q(X_k))^*$, $t' \in \widehat{T_\Delta}(X_l)$ where $l = |w|$, define $(\mu_k(\sigma)_{q,w}, t') = 1$ if there is a rule $q(\sigma(x_1, \ldots, x_k)) \to t$ in R such that $lin_{Q(X)}(t) = (t', w)$, otherwise $(\mu_k(\sigma)_{q,w}, t') = 0$. Obviously, M' is and polynomial. Moreover, M and M' are related. □

Example 4.10 (cf. Example 2.2 of [Eng75].) Consider the top-down tree transducer $M = (Q, \Sigma, \Delta, Q_d, R)$ where $Q = \{q\}$, $\Sigma = \{\bar{\gamma}^{(1)}, \gamma^{(1)}, \alpha^{(0)}\}$, $\Delta = \{\sigma^{(2)}, \gamma_1^{(1)}, \gamma_2^{(1)}, \alpha^{(0)}\}$, $Q_d = \{q\}$, and R consists of the following rules:

$$\begin{array}{rcl}
q(\bar{\gamma}(x_1)) & \to & \sigma(q(x_1), q(x_1)) \\
q(\gamma(x_1)) & \to & \gamma_1(q(x_1)) \\
q(\gamma(x_1)) & \to & \gamma_2(q(x_1)) \\
q(\alpha) & \to & \alpha
\end{array}$$

This exhibits an example of the property (T) of [Eng75]: a top-down tree transducer is able to copy a subtree of its input tree and process the copies differently.

By Lemma 4.9 we construct the top-down tree representation μ over Q, Σ, Δ, and $\mathbb{B}$ as follows:

$$\begin{aligned}\mu_1(\bar{\gamma})_{q,q(x_1)q(x_1)} &= \sigma(x_1,x_2)\\ \mu_1(\gamma)_{q,q(x_1)} &= \gamma_1(x_1)\hat{+}\gamma_2(x_1)\\ \mu_0(\alpha)_{q,\varepsilon} &= \alpha\end{aligned}$$

Then, according to Lemmas 4.8 and 4.9, $M' = (Q, \Sigma, \Delta, \mathbb{B}, Q_d, \mu)$ is the top-down tree series transducer such that $\tau_M = \tau_{M'} \circ pick_\Delta$.

Let us again compute in detail $h_\mu(\alpha)$, $h_\mu(\gamma\alpha)$, and $h_\mu(\bar{\gamma}\gamma\alpha)$, and eventually $\tau_{M'}(r)$ for the polynomial formal tree series $r = \bar{\gamma}\gamma\alpha$.

$$h_\mu(\alpha)_q = \overline{\mu_0(\alpha)}()_q = \mu_0(\alpha)_{q,\varepsilon} = \alpha$$

$$\begin{aligned}&h_\mu(\gamma\alpha)_q =\\ &= \overline{\mu_1(\gamma)}(h_\mu(\alpha))_q\\ &= \textstyle\sum_{w=p_1(x_{i_1})\dots p_l(x_{i_l})\in Q(\{x_1\})^*} \mu_1(\gamma)_{q,w} \leftarrow (h_\mu(\alpha)_{p_1},\dots,h_\mu(\alpha)_{p_l})\\ &= \mu_1(\gamma)_{q,q(x_1)} \leftarrow (h_\mu(\alpha)_q)\\ &= (\gamma_1(x_1)\hat{+}\gamma_2(x_1)) \leftarrow (\alpha)\\ &= \textstyle\sum_{t\in T_\Delta(X_1)}(\gamma_1(x_1)\hat{+}\gamma_2(x_1),t)\, t \leftarrow (\alpha)\\ &= (\gamma_1(x_1) \leftarrow (\alpha)) \hat{+} (\gamma_2(x_1) \leftarrow (\alpha))\\ &= \gamma_1(\alpha)\hat{+}\gamma_2(\alpha).\end{aligned}$$

$$\begin{aligned}&h_\mu(\bar{\gamma}\gamma\alpha)_q\\ &= \overline{\mu_1(\bar{\gamma})}(h_\mu(\gamma\alpha))_q\\ &= \textstyle\sum_{w=p_1(x_{i_1})\dots p_l(x_{i_l})\in Q(\{x_1\})^*} \mu_1(\bar{\gamma})_{q,w} \leftarrow (h_\mu(\gamma\alpha)_{p_1},\dots,h_\mu(\gamma\alpha)_{p_l})\\ &= \mu_1(\bar{\gamma})_{q,q(x_1)q(x_1)} \leftarrow (h_\mu(\gamma\alpha)_q, h_\mu(\gamma\alpha)_q)\\ &= \sigma(x_1,x_2) \leftarrow (\gamma_1(\alpha)\hat{+}\gamma_2(\alpha), \gamma_1(\alpha)\hat{+}\gamma_2(\alpha))\\ &= \hat{\textstyle\sum}_{t_1,t_2\in T_\Delta(X)}(\gamma_1(\alpha)\hat{+}\gamma_2(\alpha),t_1)(\gamma_1(\alpha)\hat{+}\gamma_2(\alpha),t_2)\sigma(x_1,x_2)[t_1,t_2]\\ &= \sigma(\gamma_1\alpha,\gamma_1\alpha)\hat{+}\sigma(\gamma_1\alpha,\gamma_2\alpha)\hat{+}\sigma(\gamma_2\alpha,\gamma_1\alpha)\hat{+}\sigma(\gamma_2\alpha,\gamma_2\alpha).\end{aligned}$$

Finally,

$$\begin{aligned}&\tau_{M'}(\bar{\gamma}\gamma\alpha)\\ &= \textstyle\sum_{q\in Q_d} h_\mu(\bar{\gamma}\gamma\alpha)_q\\ &= h_\mu(\bar{\gamma}\gamma\alpha)_q\\ &= \sigma(\gamma_1\alpha,\gamma_1\alpha)\hat{+}\sigma(\gamma_1\alpha,\gamma_2\alpha)\hat{+}\sigma(\gamma_2\alpha,\gamma_1\alpha)\hat{+}\sigma(\gamma_2\alpha,\gamma_2\alpha).\end{aligned}$$ □

Lemma 4.11 For every polynomial top-down tree series transducer M over $\mathbb{B}$ there is a top-down tree transducer M' such that M and M' are related.

Proof. Let $M = (Q, \Sigma, \Delta, \mathbb{B}, Q_d, \mu)$ be a polynomial top-down tree series transducer. Construct the top-down tree transducer $M' = (Q, \Sigma, \Delta, Q_d, R)$ as follows. Let R be the smallest set of rules satisfying the condition that for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $q \in Q$, $w = q_1(x_{i_1})\dots q_l(x_{i_l}) \in (Q(X_k))^*$, and $t' \in \widehat{T_\Delta}(X_l)$ with $(\mu_k(\sigma)_{q,w}, t') = 1$, the rule $q(\sigma(x_1,\dots,x_k)) \to t$ is in R, where $t = t'[x_1 \leftarrow q_1(x_{i_1}),\dots,x_l \leftarrow q_l(x_{i_l})]$. (Note that $lin_{Q(X)}(t) = (t', w)$.) Again, since M is polynomial, R is a finite set. Obviously, M and M' are related. □

Thus, it follows from Lemmas 4.8, 4.9, and 4.11 that also for the top-down case we obtain a characterization of top-down tree transducers in terms of polynomial top-down tree series transducers.

Theorem 4.12 $p\text{-}TOP_{t-ts}(\mathbb{B}) \circ PICK = TOP_{tt}$.

We would like to finish this section by comparing the embedding of top-down tree transducers into top-down tree series transducers as done in our Lemma 4.9 with the embedding of *nondeterministically simple root-to-frontier tree transducers* into tree transducers as done on page 139 of [Kui99]. This comparison is reasonable, because nondeterministically simple root-to-frontier transducers are particular top-down tree transducers, and tree transducers of [Kui99] are particular top-down tree series transducers.

Consider the top-down tree transducer in Example 4.10. In fact, this is a nondeterministically simple root-to-frontier transducer in the sense of [Kui99]. In the construction of [Kui99], the tree representation μ (using our notation) has the following entry:

$$(\mu_1(\bar{\gamma})_{q,q(x_1)}, t) = 1$$

where $t = \sigma(x_1, x_1)$. Compare this with the corresponding entry of the top-down tree representation in our Example 4.10:

$$(\mu_1(\bar{\gamma})_{q,q(x_1)q(x_1)}, t) = 1$$

where $t = \sigma(x_1, x_2)$.

We observe that in Kuich's approach the copying of the input subtree is done in the term t, whereas in our approach the copying is done in the index w of the top-down tree representation. Both solutions are possible. Also, it is clear that, if the solution of Kuich is chosen, then the tree series in Definition 3.2 have to be substituted in an OI-like way (as it was done in [Kui99] on page 136) and not in an IO-like way as we have defined it on page 6. On the other hand, bottom-up tree series transductions cannot be described by using OI-substitution. Thus, in order to deal with one substitution only, we choose the IO-substitution.

5 Two characterization results

In this section we characterize the classes $p\text{-}BOT_{t-ts}$ and $td\text{-}TOP_{t-ts}$. Using Lemma 2.2, the corresponding characterizations of the classes $p\text{-}BOT_{ts-ts}$ and $td\text{-}TOP_{ts-ts}$ will follow immediately. So let us concentrate now on the classes of t-ts transformations.

5.1 The characterization for the bottom-up case

First we characterize the class $p\text{-}BOT_{t-ts}$ in terms of the composition of finite state relabeling tree series transducers and homomorphism tree series transducers. The proof is based on one decomposition lemma (cf. Theorem 3.15 of [Eng75]) and one composition lemma (cf. Lemma 4.1 of [Eng75]).

Lemma 5.1 1) $p\text{-}BOT_{t-ts} \subseteq QREL_{t-ts} \,\tilde{\circ}\, p\text{-}HOM_{t-ts}$
2) $dp\text{-}BOT_{t-ts} \subseteq d\text{-}QREL_{t-ts} \,\tilde{\circ}\, p\text{-}HOM_{t-ts}$

Proof. First we prove 1). For this, let $M = (Q, \Sigma, \Delta, A, Q_d, \mu)$ be a polynomial bottom-up tree series transducer. We will construct a finite state relabeling tree series transducer M_1 and a polynomial homomorphism tree series transducer M_2, such that $\tau_M = \tau_{M_1} \,\tilde{\circ}\, \tau_{M_2}$.

Define the ranked alphabet $\Omega = \bigcup_{\sigma\in\Sigma} \Omega_\sigma$, where, for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$ we let $\Omega_\sigma = \{[\sigma, q, w, t]^{(k)} \mid q \in Q, w = q_1(x_1)\dots q_k(x_k) \in (Q(X_k))^*, t \in supp(\mu_k(\sigma)_{q,w})\}$. Note that, since M is polynomial, Ω is a finite set.

Now let $M_1 = (Q, \Sigma, \Omega, A, Q_d, \mu_1)$ be such that for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $q \in Q$, $w = q_1(x_1)\dots q_k(x_k) \in (Q(X_k))^*$ and $t' \in T_\Omega(X_k)$

$$((\mu_1)_k(\sigma)_{q,w}, t') = \begin{cases} (\mu_k(\sigma)_{q,w}, t) & \text{if } t' = [\sigma, q, w, t](x_1, \dots, x_k) \\ 0 & \text{otherwise.} \end{cases}$$

Let $M_2 = (\{*\}, \Omega, \Delta, A, \{*\}, \mu_2)$ be such that for every $k \geq 0$, $[\sigma, q, w, t] \in \Omega^{(k)}$, and $t' \in T_\Delta(X_k)$ define

$$((\mu_2)_k([\sigma, q, w, t])_{*,*(x_1)\dots *(x_k)}, t') = \begin{cases} 1 & \text{if } t' = t \\ 0 & \text{otherwise.} \end{cases}$$

Now we prove that $\tau_M = \tau_{M_1} \,\tilde{\circ}\, \tau_{M_2}$.

$$
\begin{array}{ll}
 & \tau_M(s) \\
= & \sum_{q\in Q_d} \widetilde{h_\mu(s)_q} \\
=^{+} & \sum_{q\in Q_d} \widetilde{h_{\mu_2}}(h_{\mu_1}(s)_q)_* \\
= & \widetilde{h_{\mu_2}}(\sum_{q\in Q_d} h_{\mu_1}(s)_q)_* \\
= & \widetilde{h_{\mu_2}}(\tau_{M_1}(s))_* \\
= & \widetilde{\tau_{M_2}}(\tau_{M_1}(s)) \\
= & (\tau_{M_1} \tilde{\circ} \tau_{M_2})(s)
\end{array}
$$

The equation marked by $+$ has not been proved yet. In order to verify it, we extend μ_2 to μ_2' by letting $\mu_2' = ((\mu_2)_0', (\mu_2)_k \,|\, k \geq 1)$, where $(\mu_2)_0' : (\Omega^{(0)} \cup X_{mx}) \to A\langle\langle T_{\Delta\cup\{*\}(X_{mx})}\rangle\rangle^{\{*\}\times\{\varepsilon\}}$. Here $mx = max\{k \,|\, \Omega^{(k)} \neq \emptyset\}$. The mapping $(\mu_2)_0'$ is defined such that $(\mu_2)_0'(\alpha) = (\mu_2)_0(\alpha)$ for every $\alpha \in \Omega^{(0)}$, and $(\mu_2)_0'(x_i)_* = *(x_i)$ for every $x_i \in X_{mx}$.

In this extension the elements of X_{mx} and of $\{*\}(X_{mx})$ are considered as 0-ary symbols, therefore μ_2' is a homomorphism tree representation over $\{*\}$, $\Omega \cup X_{mx}$, $\Delta \cup \{*\}(X_{mx})$, and A.

Now we can verify the equation $+$ by showing that for every $q \in Q$ and $s \in T_\Sigma$ the equality $h_\mu(s)_q = \widetilde{h_{\mu_2}}(h_{\mu_1}(s)_q)_*$ holds. This proof will be perfomed by induction on s.

$\underline{s = \sigma(s_1, \ldots, s_k)}$:

$$
\begin{array}{ll}
 & \underline{h_\mu(\sigma(s_1,\ldots,s_k))_q} \\
= & \underline{\mu_k(\sigma)}(h_\mu(s_1),\ldots,h_\mu(s_k))_q \\
= & \sum_{w=q_1(x_1)\ldots q_k(x_k)} \mu_k(\sigma)_{q,w} \leftarrow (h_\mu(s_1)_{q_1},\ldots,h_\mu(s_k)_{q_k}) \\
\stackrel{\text{I.H.}}{=} & \sum_{w=q_1(x_1)\ldots q_k(x_k)} \mu_k(\sigma)_{q,w} \leftarrow (\widetilde{h_{\mu_2}}(h_{\mu_1}(s_1)_{q_1})_*,\ldots,(\widetilde{h_{\mu_2}}(h_{\mu_1}(s_k)_{q_k})_*) \\
=^{++} & \sum_{w=q_1(x_1)\ldots q_k(x_k)} \widetilde{h_{\mu_2'}}((\mu_1)_k(\sigma)_{q,w})_* \leftarrow (\widetilde{h_{\mu_2}}(h_{\mu_1}(s_1)_{q_1})_*,\ldots,\widetilde{h_{\mu_2}}(h_{\mu_1}(s_k)_{q_k})_*) \\
=^{+++} & \widetilde{h_{\mu_2}}(\sum_{w=q_1(x_1)\ldots q_k(x_k)}(\mu_1)_k(\sigma)_{q,w} \leftarrow (h_{\mu_1}(s_1)_{q_1},\ldots,h_{\mu_1}(s_k)_{q_k}))_* \\
= & \widetilde{h_{\mu_2}}(h_{\mu_1}(\sigma(s_1,\ldots,s_k))_q)_*
\end{array}
$$

Here equation $^{++}$ holds by definition of μ_1 and μ_2'. This can be checked easily. Moreover, equation $^{+++}$ also holds because the result of the computation of a homomorphism tree series transducer on an input tree $t[t_1,\ldots,t_k]$ is of the form $t'[t_1',\ldots,t_k']$, where $t', t_1', \ldots, t_k'$ are the result of the computations on the input trees $t, t_1, \ldots, t_k$, respectively. In fact, the extension desribed above is necessary to make the tree series transducer M_2 able to handle the input t, which may contain variables.

It is clear that, if M is deterministic, then also M_1 is deterministic, hence the inclusion 2) is also proved. □

In the following lemma, we compose a (deterministic) polynomial bottom-up tree series transducer with a polynomial bottom-up homomorphism tree series transducer. The construction involved is closely related to the direct product construction [Bak79] for bottom-up tree transducers.

Lemma 5.2
$$
\begin{array}{lrcl}
1) & p\text{-}BOT_{t-ts} \,\tilde{\circ}\, p\text{-}HOM_{t-ts} & = & p\text{-}BOT_{t-ts} \\
2) & dp\text{-}BOT_{t-ts} \,\tilde{\circ}\, p\text{-}HOM_{t-ts} & = & dp\text{-}BOT_{t-ts}.
\end{array}
$$

Proof. First we prove 1). The inclusion $p\text{-}BOT_{t-ts} \subseteq p\text{-}BOT_{t-ts} \,\tilde{\circ}\, p\text{-}HOM_{t-ts}$ follows from the obvious fact that for every $\tau \in p\text{-}BOT_{t-ts}$, we have $\tau = \tau \,\tilde{\circ}\, \iota$ where ι is a suitable identity tree to tree series transformation. Moreover, $\iota \in p\text{-}HOM_{t-ts}$, see our note at the end of Section 3.

In order to prove the converse inclusion, let $M_1 = (Q_1, \Sigma, \Delta, A, Q_{d_1}, \mu_1)$ be a polynomial bottom-up and $M_2 = (\{*\}, \Delta, \Gamma, A, \{*\}, \mu_2)$ be a homomorphism tree series transducer.

Construct the bottom-up tree series transducer $M = (Q_1, \Sigma, \Gamma, A, Q_{d_1}, \mu)$ where, for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, and $q, q_1, \ldots, q_k \in Q_1$,

$$\mu_k(\sigma)_{q,q_1(x_1)\ldots q_k(x_k)} = \widetilde{h_{\mu_{2,k}}}((\mu_1)_k(\sigma)_{q,q_1(x_1)\ldots q_k(x_k)})_*$$

where $\mu_{2,k}$ is the following extension of μ_2: $\mu_{2,k} = ((\mu_{2,k})_0, (\mu_2)_l \,|\, l \geq 1)$ and $(\mu_{2,k})_0 : \Delta^{(0)} \cup X_{mx} \to A\langle\langle T_{\Gamma\cup\{*\}(X_{mx})}\rangle\rangle^{\{\varepsilon\}}$ such that,

- for every $\alpha \in \Delta^{(0)}$, $(\mu_{2,k})_0(\alpha) = (\mu_2)_0(\alpha)$ and
- for every $1 \leq i \leq k$, $(\mu_{2,k})_0(x_i)_{*,\varepsilon} = *(x_i)$,
- for every $j \notin \{1, \ldots, k\}$, $(\mu_{2,k})_0(x_j)_{*,\varepsilon} = \tilde{0}$.

It should be clear that M is also a polynomial tree series transducer.

Now we prove the equality $\tau_M = \tau_{M_1} \tilde{\circ} \tau_{M_2}$ which shows the correctness of the construction. Let $s \in T_\Sigma$.

$$
\begin{array}{ll}
 & (\tau_{M_1} \tilde{\circ} \tau_{M_2})(s) \\
= & \widetilde{\tau_{M_2}}(\tau_{M_1}(s)) \\
= & \widetilde{\tau_{M_2}}(\sum_{q \in Q_{d_1}} h_{\mu_1}(s)_q) \\
= & \widetilde{h_{\mu_2}}(\sum_{q \in Q_{d_1}} h_{\mu_1}(s)_q)_* \\
= & \sum_{q \in Q_{d_1}} \widetilde{h_{\mu_2}}(h_{\mu_1}(s)_q)_* \\
=^+ & \sum_{q \in Q_{d_1}} h_\mu(s)_q \\
= & \tau_M(s)
\end{array}
$$

At the equation $+$ we have used the fact that, for every $q \in Q_1$ and $s \in T_\Sigma$, the equality $\widetilde{h_{\mu_2}}(h_{\mu_1}(s)_q)_* = h_\mu(s)_q$ holds. This fact we will prove now by induction on s.

$\underline{\sigma(s_1, \ldots, s_k):}$

$$
\begin{array}{ll}
 & h_\mu(\sigma(s_1, \ldots, s_k))_q \\
= & \overline{\mu_k(\sigma)}(h_\mu(s_1), \ldots, h_\mu(s_k))_q \\
= & \sum_{w=q_1(x_1)\ldots q_k(x_k)} \mu_k(\sigma)_{q,w} \leftarrow (h_\mu(s_1)_{q_1}, \ldots, h_\mu(s_k)_{q_k}) \\
\overset{I.H.}{=} & \sum_{w=q_1(x_1)\ldots q_k(x_k)} \mu_k(\sigma)_{q,w} \leftarrow \underbrace{(\widetilde{h_{\mu_2}}(h_{\mu_1}(s_1)_{q_1})_*, \ldots, \widetilde{h_{\mu_2}}(h_{\mu_1}(s_k)_{q_k})_*)}_{\varphi} \\
=^{++} & \sum_{w=q_1(x_1)\ldots q_k(x_k)} \widetilde{h_{\mu_{2,k}}}((\mu_1)_k(\sigma)_{q,q_1(x)\ldots q_k(x_k)})_* \leftarrow \varphi \\
= & \sum_{v=q_1(x_1)\ldots q_k(x_k)} \widetilde{h_{\mu_{2,k}}}((\mu_1)_k(\sigma)_{q,q_1(x)\ldots q_k(x_k)})_* \leftarrow \varphi \\
=^{+++} & \sum_{v=q_1(x_1)\ldots q_k(x_k)} \widetilde{h_{\mu_2}}((\mu_1)_k(\sigma)_{q,q_1(x)\ldots q_k(x_k)} \leftarrow (h_{\mu_1}(s_1)_{q_1}, \ldots, h_{\mu_1}(s_k)_{q_k}))_* \\
= & \widetilde{h_{\mu_2}}(\sum_{v=q_1(x_1)\ldots q_k(x_k)} (\mu_1)_k(\sigma)_{q,q_1(x)\ldots q_k(x_k)} \leftarrow (h_{\mu_1}(s_1)_{q_1}, \ldots, h_{\mu_1}(s_k)_{q_k}))_* \\
= & \widetilde{h_{\mu_2}}(\overline{(\mu_1)_k(\sigma)}(h_{\mu_1}(s_1), \ldots, h_{\mu_1}(s_k))_q)_* \\
= & \widetilde{h_{\mu_2}}(h_{\mu_1}(\sigma(s_1, \ldots, s_k))_q)_*
\end{array}
$$

The equation $^{++}$ holds by definition of μ. Moreover, equation $^{+++}$ holds because of the way in which the bottom-up homomorphism tree series transducer M_2 computes and because of the definition of $\mu_{2,k}$ (cf. also Lemma 1.1 of [Eng75]). This finishes the proof of 1).

If M_1 is deterministic in the above construction, then M is also deterministic, hence 2) is also proved. $\square$

Now the characterization of the class $p\text{-}BOT_{t-ts}$ follows immediately from Lemmas 5.1 and 5.2 and the fact that every finite state relabeling tree series transducer is polynomial.

Theorem 5.3 1) $p\text{-}BOT_{t-ts} = QREL_{t-ts} \tilde{\circ}\, p\text{-}HOM_{t-ts}$
2) $dp\text{-}BOT_{t-ts} = d\text{-}QREL_{t-ts} \tilde{\circ}\, p\text{-}HOM_{t-ts}$.

It follows immediately from Lemma 2.2 that the classes $p\text{-}BOT_{ts-ts}$ and $dp\text{-}BOT_{ts-ts}$ can be characterized in the same way.

Corollary 5.4 1) $p\text{-}BOT_{ts-ts} = QREL_{ts-ts} \circ p\text{-}HOM_{ts-ts}$
2) $dp\text{-}BOT_{ts-ts} = d\text{-}QREL_{ts-ts} \circ p\text{-}HOM_{ts-ts}$.

5.2 The characterization for the top-down case

Second we characterize the class $(t)d\text{-}TOP_{t-ts}$ (cf. Theorem 5.7). We prepare this characterization by proving two lemmas (cf. Lemma 3.6 of [Eng75]).

Lemma 5.5 1) $TOP_{t-ts} \subseteq HOM_{t-ts} \tilde{\circ}\, l\text{-}TOP_{t-ts}$
2) $d\text{-}TOP_{t-ts} \subseteq HOM_{t-ts} \tilde{\circ}\, ld\text{-}TOP_{t-ts}$
3) $td\text{-}TOP_{t-ts} \subseteq HOM_{t-ts} \tilde{\circ}\, ltd\text{-}TOP_{t-ts}$.

Proof. First we prove 1). Therefore, let $M = (Q, \Sigma, \Delta, A, Q_d, \mu)$ be a top-down tree series transducer. Construct the homomorphism tree series transducer $H = (\{*\}, \Sigma, \Sigma', A, \{*\}, \mu_H)$ and the linear top-down tree series transducer $M' = (Q, \Sigma', \Delta, A, Q_d, \mu')$ such that $\tau_M = \tau_H \,\tilde{\circ}\, \tau_{M'}$.

Let $n = max\{\#occ(x_i, w) \mid k \geq 0, \sigma \in \Sigma^{(k)}, q \in Q, w \in (Q(X_k))^*, \mu_k(\sigma)_{q,w} \neq \tilde{0}, 1 \leq i \leq k\}$ where $\#occ(x_i, w)$ denotes the number of occurrences of x_i in w.

Define $\Sigma' = \{\sigma'^{(k \cdot n)} \mid \sigma \in \Sigma^{(k)}\}$ and, for every $k \geq 0$ and and $\sigma \in \Sigma^{(k)}$, define

$$(\mu_H)_k(\sigma)_{*,\bar{w}} = \sigma'(x_1, \ldots, x_n, \ldots, x_{(k-1)\cdot n+1}, \ldots, x_{k \cdot n})$$

where $\bar{w} = \underbrace{*(x_1) \ldots *(x_1)}_{n} \ldots \underbrace{*(x_k) \ldots *(x_k)}_{n}$ and for every $w \neq \bar{w}$ define

$$(\mu_H)_k(\sigma)_{*,w} = \tilde{0}.$$

Thus, H is a homomorphism tree series transducer.

Now we turn to the construction of M'. Let $w = q_1(x_{i_1}) \ldots q_l(x_{i_l}) \in (Q(X_k))^*$. Define $w' = q_1(x_{j_1}) \ldots q_l(x_{j_l})$ where, for every $1 \leq d \leq l$, $x_{j_d} = x_{(u-1)\cdot n+\kappa}$ if $x_{i_d} = x_u$ and $q_d(x_{i_d})$ contains the κth occurrence of x_u in w. (Cf. the proof of Lemma 3.6 in [Eng75].) Then define for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $q \in Q$, $w \in (Q(X_k))^*$:

$$\mu'_{k \cdot n}(\sigma')_{q,w} = \begin{cases} \mu_k(\sigma)_{q,v'} & \text{if there is a } v \in (Q(X_k))^* \text{ such that } v' = w \\ \tilde{0} & \text{otherwise} \end{cases}$$

It is easy to see that M' is linear. Moreover, if M is (total) deterministic, then it is easy to check that M' is also (total) deterministic.

Before we prove that $\tau_M = \tau_H \tilde{\circ} \tau_{M'}$ we define the mapping $copy : T_\Sigma \longrightarrow T_{\Sigma'}$ and show its connection with τ_H. The mapping $copy$ is defined by induction as follows: $copy(\sigma(s_1, \ldots, s_k)) = \sigma'(s'_1, \ldots, s'_{k \cdot n})$ where for every $1 \leq j \leq k \cdot n$, let $s'_j = copy(s_i)$ where i is determined by the conditions $j = (i-1)\cdot n+\kappa$ and $1 \leq \kappa < n$. We extend the mapping $copy$ in a natural way to the t-ts transformation $\widetilde{copy} : T_\Sigma \longrightarrow A\langle T_{\Sigma'} \rangle$ by defining, for every $s \in T_\Sigma$, $\widetilde{copy}(s) = 1 \cdot copy(s)$. Then, for every $s \in T_\Sigma$, $\widetilde{copy}(s) = \tau_H(s)$.

Now we prove that $\tau_M = \tau_H \,\tilde{\circ}\, \tau_{M'}$. Let $s \in T_\Sigma$.

$$\begin{array}{cl} & \tau_M(s) \\ = & \sum_{q \in Q_d} h_\mu(s)_q \\ =^+ & \sum_{q \in Q_d} h_{\mu'}(copy(s))_q \\ = & \tau_{M'}(copy(s)) \\ = & \widetilde{\tau_{M'}}(\widetilde{copy}(s)) \\ = & \widetilde{\tau_{M'}}(\tau_H(s)) \\ = & (\tau_H \tilde{\circ} \tau_{M'})(s) \end{array}$$

We can verify equation + by proving the following statement by induction on s: for every $s \in T_\Sigma$, $q \in Q$, $h_\mu(s)_q = h_{\mu'}(copy(s))_q$.

$\underline{s = \sigma(s_1, \ldots, s_k)}$:

$$\begin{array}{cl} & h_\mu(\sigma(s_1, \ldots, s_k))_q \\ = & \mu_k(\sigma)(h_\mu(s_1), \ldots, h_\mu(s_k))_q \\ = & \sum_{w = q_1(x_{i_1}) \ldots q_l(x_{i_l})} \mu_k(\sigma)_{q,w} \leftarrow (h_\mu(s_{i_1})_{q_1}, \ldots, h_\mu(s_{i_l})_{q_l}) \\ \stackrel{\text{I.H.}}{=} & \sum_{w = q_1(x_{i_1}) \ldots q_l(x_{i_l})} \mu_k(\sigma)_{q,w} \leftarrow (h_{\mu'}(copy(s_{i_1}))_{q_1}, \ldots, h_{\mu'}(copy(s_{i_l}))_{q_l}) \\ =^{++} & \sum_{w' = q_1(x_{j_1}) \ldots q_l(x_{j_l})} \mu'_k(\sigma')_{q,w'} \leftarrow (h_{\mu'}(copy(s_{i_1}))_{q_1}, \ldots, h_{\mu'}(copy(s_{i_l}))_{q_l}) \\ = & \sum_{u = p_1(x_{\varrho_1}) \ldots p_m(x_{\varrho_m})} \mu'_k(\sigma')_{q,u} \leftarrow (h_{\mu'}(s'_{\varrho_1})_{p_1}, \ldots, h_{\mu'}(s'_{\varrho_m})_{p_m}) \\ = & \mu'_k(\sigma')(h_{\mu'}(s'_1), \ldots, h_{\mu'}(s'_{k \cdot n}))_q \\ = & h'_\mu(\sigma'(s'_1, \ldots, s'_{k \cdot n}))_q \\ = & h'_\mu(copy(\sigma(s_1, \ldots, s_k)))_q. \end{array}$$

At equation ++ we used the property that $\mu'_k(\sigma')_{q,u} = \tilde{0}$ if $u \neq w'$ for some $w \in (Q(X_k))^*$ with $\mu_k(\sigma)_{q,w} \neq \tilde{0}$.

Now if M is deterministic (and total) in the above proof, then M' will also be deterministic (and total). This verifies 2) and 3). □

Lemma 5.6 1) $td\text{-}TOP_{t-ts} \mathbin{\tilde{\circ}} d\text{-}TOP_{t-ts} \;=\; d\text{-}TOP_{t-ts}$
2) $td\text{-}TOP_{t-ts} \mathbin{\tilde{\circ}} td\text{-}TOP_{t-ts} \;=\; td\text{-}TOP_{t-ts}$.

Proof. First we prove 1). The inclusion $d\text{-}TOP_{t-ts} \subseteq td\text{-}TOP_{t-ts} \mathbin{\tilde{\circ}} d\text{-}TOP_{t-ts}$ follows from that for every $\tau \in d\text{-}TOP_{t-ts}$ we have $\tau = \iota \mathbin{\tilde{\circ}} \tau$, where ι is a suitable identity tree to tre series transformation.

The converse inclusion can be seen as follows. Let $M_1 = (Q_1, \Sigma, \Delta, q_1, A, \mu_1)$ and $M_2 = (Q_2, \Delta, \Gamma, q_2, A, \mu_2)$ be deterministic top-down tree series transducers and assume also that M_1 is total. We construct a third deterministic top-down tree series transducer $M = (Q, \Sigma, \Gamma, q_0, A, \mu)$ such that $\tau_M = \tau_{M_1} \mathbin{\tilde{\circ}} \tau_{M_2}$. Our construction is the generalization of the usual direct product construction decribed among others in [Bak79, Eng75, FV98].

First we extend μ_2 to μ_2' by letting $\mu_2' = ((\mu_2)_0', (\mu_2)_k \mid k \geq 1)$, where $(\mu_2)_0' : (\Delta^{(0)} \cup X_{mx}) \to A\langle\langle T_{\Gamma \cup Q_2(X_{mx})}\rangle\rangle^{Q_2 \times \{\varepsilon\}}$. Here $mx = max\{k \mid \Sigma^{(k)} \neq \emptyset\}$. The mapping $(\mu_2)_0'$ is defined such that $(\mu_2)_0'(\alpha) = (\mu_2)_0(\alpha)$ for every $\alpha \in \Delta^{(0)}$, and $(\mu_2)_0'(x_i)_p = p(x_i)$ for every $x_i \in X_{mx}$ and $p \in Q_2$.

In this extension the elements of X_{mx} and of $Q_2(X_{mx})$ are considered as 0-ary symbols, therefore μ_2' is a top-down tree representation over Q_2, $\Delta \cup X_{mx}$, $\Gamma \cup Q_2(X_{mx})$, and A.

Now we can construct M as follows. Let $Q = Q_2 \times Q_1$ and $q_0 = \langle q_2, q_1\rangle$. Moreover, the mapping μ is defined as follows. For every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $\langle p, q\rangle \in Q$, $v \in (Q(X_k))^*$, $t' \in \widehat{T_\Gamma}(X_m)$, where $m = |v|$, and $a, \hat{a} \in A$,

$(\mu_k(\sigma)_{\langle p,q\rangle, v}, t') = a\hat{a}$ iff

there exist $w \in (Q_1(X_k))^*$, $t \in \widehat{T_\Delta}(X_l)$, and $\hat{t} \in T_\Gamma(Q_2(X_l))$ such that the following conditions hold:

- $l = |w|$,
- $((\mu_1)_k(\sigma)_{q,w}, t) = a$,
- $(h_{\mu_2'}(t)_p, \hat{t}\,) = \hat{a}$,
- $lin_{Q_2(X_l)}(\hat{t}\,) = (t', u)$, and
- $v = u\langle w\rangle$.

The notion $u\langle w\rangle$ is defined in the following way. Let $w \in (Q_1(X_k))^*$ with $|w| = l$ and let $u \in (Q_2(X_l))^*$. Then $u\langle w\rangle$ is a word in $(Q(X_k))^*$ obtained by replacing, for every $p \in Q_2$ and $x_i \in X_l$, the expression $p(x_i)$ in u by $\langle p, q\rangle(x_j)$ where $q(x_j)$ is the ith letter in w.

Now we prove that $\tau_M = \tau_{M_1} \mathbin{\tilde{\circ}} \tau_{M_2}$.

$$
\begin{array}{ll}
 & \tau_M(s) \\
= & h_\mu(s)_{\langle q_2, q_1\rangle} \\
=^+ & \widetilde{h_{\mu_2}}(h_{\mu_1}(s)_{q_1})_{q_2} \\
= & \widetilde{\tau_{M_2}}(\tau_{M_1}(s)) \\
= & (\tau_{M_1} \mathbin{\tilde{\circ}} \tau_{M_2})(s)
\end{array}
$$

Hence it is sufficient to show that for every $s \in T_\Sigma$, $p \in Q_1$ and $q \in Q_2$ the equation $h_\mu(s)_{\langle p,q\rangle} = \widetilde{h_{\mu_2}}(h_{\mu_1}(s)_q)_p$. We prove by induction on s.

<u>$s = \sigma(s_1, \ldots, s_k)$:</u>

$$
\begin{array}{ll}
 & h_\mu(\sigma(s_1, \ldots, s_k))_{\langle p,q\rangle} \\
= & \mu_k(\sigma)(h_\mu(s_1), \ldots, h_\mu(s_k))_{\langle p,q\rangle} \\
= & \mu_k(\sigma)_{\langle p,q\rangle, v} \leftarrow (h_\mu(s_{i1})_{\langle p_1, q_1\rangle}, \ldots, h_\mu(s_{im})_{\langle p_m, q_m\rangle}) \\
 & \text{for some } v = \langle p_1, q_1\rangle(x_{i1}) \ldots \langle p_m, q_m\rangle(x_{im}) \in (Q_2 \times Q_1(X_k))^* \\
\overset{\text{I.H.}}{=} & \mu_k(\sigma)_{\langle p,q\rangle, v} \leftarrow (\widetilde{h_{\mu_2}}(h_{\mu_1}(s_{i1})_{q_1})_{p_1}, \ldots, \widetilde{h_{\mu_2}}(h_{\mu_1}(s_{im})_{q_m})_{p_m}) \\
 & \text{for some } v = \langle p_1, q_1\rangle(x_{i1}) \ldots \langle p_m, q_m\rangle(x_{im}) \in (Q_2 \times Q_1(X_k))^* \\
=^+ & \widetilde{h_{\mu_2'}}((\mu_1)_k(\sigma)_{q,w})_p \leftarrow (\widetilde{h_{\mu_2}}(h_{\mu_1}(s_{i1})_{q_1})_{p_1}, \ldots, \widetilde{h_{\mu_2}}(h_{\mu_1}(s_{im})_{q_m})_{p_m}) \\
 & \text{for some } w \in (Q_1(X_k))^* \text{ with } w = q_1'(x_{j1}) \ldots q_l'(x_{jl}) \text{ and } u \in (Q_2(X_l))^* \\
 & \text{such that the following conditions hold:} \\
 & \widetilde{h_{\mu_2'}}((\mu_1)_k(\sigma)_{q,w})_p = \hat{a}\hat{t},\ lin_{Q_2(X_l)}(\hat{t}) = (t', u) \text{ and } v = u\langle w\rangle. \\
=^{++} & \widetilde{h_{\mu_2}}((\mu_1)_k(\sigma)_{q,w} \leftarrow (h_{\mu_1}(s_{j1})_{q_1'}, \ldots, h_{\mu_1}(s_{jl})_{q_l'})_p \\
= & \widetilde{h_{\mu_2}}(h_{\mu_1}(\sigma(s_1, \ldots, s_k))_q)_p
\end{array}
$$

The equation + holds because of the construction of M while ++ holds because it describes how a top-down tree series transducers works on an input of the form $t[t_1, \ldots, t_k]$. Moreover, at ++ we also used that M_1 is total. In fact, it is possible that some of the variables $x_{j1}, \ldots, x_{jl}$ do not occur among the ones $x_{i1}, \ldots, x_{im}$ (because M_2 may delete them). Hence some of the formal tree series $h_{\mu_1}(s_{j1})_{q'_1}, \ldots, h_{\mu_1}(s_{jl})_{q'_l}$ may not be among $h_{\mu_1}(s_{i1})_{q_1}, \ldots, h_{\mu_1}(s_{im})_{q_m}$. However, if M_1 is total, then each of $h_{\mu_1}(s_{j1})_{q'_1}, \ldots, h_{\mu_1}(s_{jl})_{q'_l}$ exists. This finishes the proof of 1).

Certainly, if M_1 is deterministic, then M will also be deterministic. Hence 2) is also shown. □

From Lemmas 5.5 and 5.6 we obtain the following characterization immediately.

Theorem 5.7 $HOM_{t-ts} \,\tilde{\circ}\, ld\text{-}TOP_{t-ts} = d\text{-}TOP_{t-ts}$

As a corollary from Lemma 5.6, Theorem 5.7 and Lemma 2.2 we obtain characterizations for the corresponding classes of ts-ts transformations.

Corollary 5.8
1) $td\text{-}TOP_{ts-ts} \circ d\text{-}TOP_{ts-ts} = d\text{-}TOP_{ts-ts}$
2) $td\text{-}TOP_{ts-ts} \circ td\text{-}TOP_{ts-ts} = td\text{-}TOP_{ts-ts}$
3) $HOM_{ts-ts} \circ ld\text{-}TOP_{ts-ts} = d\text{-}TOP_{ts-ts}$

6 Further research topics

Clearly, the present paper is only the starting point for further investigations on tree series transducers.

For instance, the homomorphism h_μ can be considered as a homomorphism tree transducer with external functions [FHVV93] in which the algebra does not have trees as carrier sets but formal tree series. Which results can be achieved by generalizing from homomorphism tree transducers to top-down, attributed or macro tree transducers? Clearly, for this we would need an appropriate definition of tree representation.

The extension $\tilde{\tau}$ of a tree series transformation τ in, say BOT_{t-ts} is a transformation of formal tree series. Which results can be proved if the arguments of such transformations are themselves computed by some tree transducer device?

In a natural way, one can define bottom-up (and top-down) finite tree series automata, e.g., a bottom-up tree series transducer M is a *bottom-up finite-state tree series automaton* if $\Sigma = \Delta$ and, for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $q, q_1, \ldots, q_k \in Q$, if $\mu_k(\sigma)_{q,q_1(x_1)\ldots q_k(x_k)} \neq \tilde{0}$, then $supp(\mu_k(\sigma)_{q,q_1(x_1)\ldots q_k(x_k)}) = \{\sigma(x_1, \ldots, x_k)\}$. Let BOT_{tsa} denote the class of t-ts transformations computed by bottom-up finite-state tree series automata. Now it is a natural question whether – as it holds in the tree transducer case – deterministic polynomial bottom-up finite-state tree series automata are as powerful as their nondeterministic version, i.e., whether $p\text{-}BOT_{tsa} = dp\text{-}BOT_{tsa}$. Unfortunately, the naive adaptation of the powerset construction does not work, because then also coefficients are summed up which do not count in the nondeterministic case. However, we think that $p\text{-}BOT_{tsa} \subseteq REL \circ dp\text{-}BOT_{tsa}$ where REL denotes a class of relabelings that add to every node the set P of states such that $p \in P$ iff the nondeterministic automaton can reach this node in state q, and q is successful (for the notion of *sucessful* cf. [BE97]).

Acknowledgement

The second author would like to thank Manfred Droste and Dietrich Kuske for fruitful discussions on this topic.

References

[Bak79] B. S. Baker. Composition of top-down and bottom-up tree transductions. *Inform. and Control*, 41:186–213, 1979.

[BE97] R. Bloem and J. Engelfriet. Characterization of properties and relations defined in monadic second order logic on the nodes of trees. Technical Report Technical Report 97-03, Leiden University, August 1997.

[BE00] R. Bloem and J. Engelfriet. A comparison of tree transductions defined by monadic second order logic and by attribute grammars. *Journal of Computer and System Sciences*, 61:1–50, 2000.

[Ber79] J. Berstel. *Transductions and Context-Free Languages*. Teubner Stuttgart, 1979.

[BN98] F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.

[BR82] J. Berstel and C. Reutenauer. Recognizable formal power series in trees. *Theoretical Computer Science*, (18):115–148, 1982.

[BR88] J. Berstel and Ch. Reutenauer. *Rational Series and Their Languages*, volume 12 of *EATCS-Monographs*. Springer Verlag, 1988.

[CDG+97] H. Comon, M. Dauchet, R. Gilleron, F. Jacquema, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available on: http://www.grappa.univ-lille3.fr/tata, 1997.

[Eil74] S. Eilenberg. *Automata, Languages, and Machines, Vol.A*. Academic Press, 1974.

[EM99] J. Engelfriet and S. Maneth. Macro tree transducers, attribute grammars, and MSO definable tree translations. *Inform. and Comput.*, 154:34–91, 1999.

[Eng75] J. Engelfriet. Bottom–up and top–down tree transformations — a comparison. *Math. Systems Theory*, 9(3):198–231, 1975.

[Eng77] J. Engelfriet. Top–down tree transducers with regular look–ahead. *Math. Systems Theory*, 10:289–303, 1977.

[Eng82] J. Engelfriet. Three Hierarchies of Transducers. *Math. Systems Theory*, 15:95–125, 1982.

[ES77] J. Engelfriet and E. M. Schmidt. IO and OI.I. *J. Comput. System Sci.*, 15(3):328–353, 1977.

[ES78] J. Engelfriet and E.M. Schmidt. IO and OI.II. *J. Comput. System Sci.*, 16(1):67–99, 1978.

[EV85] J. Engelfriet and H. Vogler. Macro tree transducers. *J. Comput. System Sci.*, 31:71–146, 1985.

[FHVV93] Z. Fülöp, F. Herrmann, S. Vágvölgyi, and H. Vogler. Tree transducers with external functions. *Theoret. Comput. Sci.*, 108:185–236, 1993.

[FSW94] C. Ferdinand, H. Seidl, and R. Wilhelm. Tree automata for code selection. *Acta Informatica*, 31:741–760, 1994.

[FV92] Z. Fülöp and S. Vágvölgyi. Decidability of the inclusion in monoids generated by tree transformation classes. In M. Nivat and A. Podelski, editors, *Tree Automata and Languages*, pages 381–408. Elsevier Science Publishers B.V., 1992.

[FV98] Z. Fülöp and H. Vogler. *Syntax-directed semantics — Formal Models Based on Tree Transducers*. Monographs in Theoretical Computer Science, An EATCS Series. Springer-Verlag, 1998.

[GS84] F. Gécseg and M. Steinby. *Tree Automata*. Akadémiai Kiadó, Budapest, 1984.

[GS97] F. Gécseg and M. Steinby. Tree languages. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, chapter 1, pages 1–68. Springer-Verlag, 1997.

[Jan97] M. Jantzen. Basics on term rewriting. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, chapter 5, pages 269–337. Springer-Verlag, 1997.

[KS86] W. Kuich and A. Salomaa. *Semirings, Automata, Languages*. EATCS Monographs on Theoretical Computer Science, Springer Verlag, 1986.

[Kui97a] W. Kuich. Formal power series over trees. In *Proc. of the 3rd International Conference Developments in Language Theory*, pages 60–101. Aristotle University of Thessaloniki, 1997.

[Kui97b] W. Kuich. Semirings and formal power series. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages, Vol. 1*, chapter 9, pages 609–677. Springer, 1997.

[Kui99] W. Kuich. Tree transducers and formal tree series. *Acta Cybernetica*, 14, 135-149 1999.

[Rou68] W.C. Rounds. *Trees, transducers and transformations.* PhD thesis, Stanford University, 1968.

[Rou70] W.C. Rounds. Mappings and grammars on trees. *Math. Systems Theory*, 4:257–287, 1970.

[Sch62] M. P. Schützenberger. Certain elementary families of automata. In *Proceedings of Symposium on Mathematical Theory of Automata*, pages 139–153, 1962. Polytechnic Institute of Broocklyn.

[Sei92] H. Seidl. Finite tree automata with cost functions. In *17th CAAP*, volume 581 of *Lect. Notes Comput. Sci.*, pages 279–299, Universität des Saarlandes, 1992. Springer–Verlag.

[SS78] A. Salomaa and M. Soittola. *Automata-Theoretic Aspects of Formal Power Series.* Texts and Monographs in Computer Science, Springer Verlag, 1978.

[Tha70] J.W. Thatcher. Generalized^2 sequential machine maps. *J. Comput. System Sci.*, 4:339–367, 1970.