Chapter 9:
Weighted Tree Automata and Tree Transducers

Zoltán Fülöp[1,⋆] and Heiko Vogler[2]

[1] Department of Computer Science, University of Szeged,
Árpád tér 2., 6720 Szeged, Hungary
fulop@inf.u-szeged.hu

[2] Faculty of Computer Science, Technische Universität Dresden,
01062 Dresden, Germany
vogler@tcs.inf.tu-dresden.de

Dedicated to Symeon Bozapalidis

1 Introduction 314

2 Preliminaries 316

2.1 General Notation 316
2.2 Trees 316
2.3 Algebraic Concepts 317
2.4 Tree Series 320

3 Weighted Tree Automata 320

3.1 Bottom-up Tree Automata 320
3.2 Recognizable Tree Series 321
3.3 Closure Properties 326
3.4 Support of Recognizable Tree Series 328
3.5 Determinization of Weighted Tree Automata 330
3.6 Pumping Lemmata and Decidability 331
3.7 Finite Algebraic Characterizations of Recognizable Tree Series 335
3.8 Equational Tree Series 342
3.9 Rational Tree Series 347
3.10 MSO-Definable Tree Series 349
3.11 Other Models Related to Recognizable Tree Series 353
3.12 Further Results 360

⋆ Research of the first author was supported by the Hungarian Scientific Research Fund, grant T 46686, and by the Austrian–Hungarian Action Fund, grant 68öu2.

M. Droste, W. Kuich, H. Vogler (eds.), *Handbook of Weighted Automata*,
Monographs in Theoretical Computer Science. An EATCS Series,
DOI 10.1007/978-3-642-01492-5_9, © Springer-Verlag Berlin Heidelberg 2009

4 IO-Substitution and Tree Series Transformations 361

5 Weighted Tree Transducers 364

5.1 Tree Transducers 364
5.2 The Basic Model 365
5.3 Restricted Models 370
5.4 Composition and Decomposition 374
5.5 The Inclusion Diagram of Some Fundamental wtt Classes 386
5.6 Hierarchies 388
5.7 Further Models of Weighted Tree Transducers 391
5.8 Further Results 394

References 394

1 Introduction

Over the past four decades, the theory of finite state tree automata and tree transducers has been developed intensively (cf. [68, 69, 29, 64] for a survey). This classical theory deals with (formal) tree languages and with relations over trees; it contains, e.g., characterizations of the class of recognizable tree languages and composition results for certain classes of tree transformations. Tree automata and tree transducers have also proved useful as formal models for analyzing and transforming trees in applications like natural language processing [85, 114, 72, 83, 84], syntax-directed semantics [79, 51, 64, 34], picture generation [35], or the processing of semi-structured documents [136, 117, 126, 54, 111].

Now it is natural to generalize tree automata and tree transducers by changing from the qualitative point of view to a quantitative one. For instance, besides knowing that a pattern occurs in a tree, one might want to know also the number of such occurrences. Another example is that we would like to know the probability of the event that an output tree is the translation of an input tree. Then a tree language becomes a mapping, called a tree series, from the set of trees to a set S of quantities. Similarly, a tree transformation is turned into a mapping, called a tree series transformation, from trees into tree series over S. This extension results in the formal models of *weighted tree automaton* and *weighted tree transducer*, respectively. In order to be able to calculate with quantities, an algebraic structure is needed; and it has turned out that semirings are the most appropriate ones for this purpose. Then for an arbitrary run on an input tree, the weights of the involved transitions are combined by using the semiring multiplication and, if there exist several runs on a tree (which, in the case of transducers, lead to the same translation), then the semiring addition is applied to the weights of all these runs. The classical *unweighted case* is reobtained by considering the Boolean semiring $\mathbb{B}$ with disjunction and conjunction as addition and multiplication, respectively. In fact, for string automata the quantitative point of view has been investigated since the 1960s of the previous century and it led to the rich theory of formal power series [45, 125, 96, 8, 87].

Weighted tree automata have been studied quite intensively by now. The approaches in the studies differ in the class of semirings they employ, e.g., completely distributive lattices [78, 59], fields [7], commutative semirings [1], and continuous semirings [88, 58]. Every such class has its own benefits: Using lattices, a bridge to fuzzy sets and concepts is built; taking fields, the tools and results of linear algebra are available; using commutative semirings, more general results can be proved; continuous semirings allow for the solution of systems of linear equations which is a fundamental concept. Using the semiring of real numbers (with the usual addition and multiplication), probabilistic tree automata [100, 47] can be defined; such automata associate with every transition a weight in the interval $[0, 1]$, and the weights of all possible transitions in a state on a symbol sum up to 1. The investigation into weighted tree transducers was started in [91] and continued in [53, 65, 61, 66, 101, 104] and others. Results regarding composition, decomposition, and hierarchies were lifted from the unweighted to the weighted case. As for weighted tree automata, this lifting had to be done with much care because properties of $\mathbb{B}$ (like idempotency, finiteness, commutativity), which are used quite often in the unweighted case, are now gone. In fact, this makes the weighted case interesting. For a survey on some results on weighted tree automata and weighted tree transducers, we refer to [88, 58].

In this chapter of the *Handbook of Weighted Automata*, we have collected some important results for weighted tree automata and weighted tree transducers. We restrict ourselves to finite trees and we consider only ranked trees (in contrast to unranked trees such as those used to model fully structured XML-documents). In particular, we address closure properties of the class of recognizable tree series, results on the support of such tree series, the determinization of weighted tree automata, pumping lemmata and decidability results, and finite algebraic characterizations of recognizable tree series. We discuss the equivalence between recognizable tree series and equational, rational, and MSO-definable tree series, and we present a comparison of several other models of recognizability. The part on weighted tree automata ends with a list of further results which we will not discuss in detail. For weighted tree transducers, we show composition and decomposition results, an inclusion diagram of some fundamental classes of tree series transformations, and hierarchies obtained by composing weighted tree transducers. We briefly discuss other models of weighted tree transducers. Finally, we give a short list of further results on weighted tree transducers that are not addressed in our main sections.

We have tried to produce a self-contained chapter; thus, the reader who has some background in automata theory and formal languages can easily follow the development. For many theorems, we have included sketches of their proofs, and we have always indicated the original source where the reader can sometimes find more details.

The different topics which we address require different additional properties for the used semiring, e.g., commutativity, zero-divisor freeness, or that the semiring is a semifield. In order to avoid repetitions of the respective list of additional properties during the development of a topic, we adopt the following convention: We will place, if appropriate, at the beginning of a section or subsection a general statement about the additional properties which we assume to hold throughout that section or subsection, and we do *not* explicitly mention these assumptions in the individual statements.

2 Preliminaries

2.1 General Notation

Let $\mathbb{N}$ denote the set $\{0, 1, 2, \ldots\}$ of natural numbers. For a set A, we denote its set of subsets by $\mathcal{P}(A)$ and the set of strings over A by A^*. The empty string is denoted by ε and the length of a string w by $|w|$. We denote the cardinality of a finite set A by $|A|$.

Let H, I, and J be sets. An *$I \times J$ matrix over H* is a mapping $\mathcal{M} : I \times J \to H$; the set of all $I \times J$ matrices over H is denoted by $H^{I \times J}$. We write an entry $\mathcal{M}(i, j) \in H$ as $\mathcal{M}_{i,j}$. An *I-vector v over H* is defined analogously; the set of all I-vectors over H is denoted by H^I and an element $v(i) \in H$ of v is denoted by v_i.

For two functions $f : A \to B$ and $g : B \to C$, we denote their composition by $g \circ f$ where $(g \circ f)(a) = g(f(a))$ for every $a \in A$.

2.2 Trees

A *ranked alphabet* is a tuple (Σ, rk) where Σ is a finite set and $rk : \Sigma \to \mathbb{N}$ is a mapping called rank mapping. For every $k \geq 0$, we define $\Sigma^{(k)} = \{\sigma \in \Sigma \mid rk(\sigma) = k\}$. Sometimes, we write $\sigma^{(k)}$ to mean that $\sigma \in \Sigma^{(k)}$. Moreover, let H be a set disjoint with Σ. The set of *Σ-terms over H*, denoted by $T_\Sigma(H)$, is the smallest set T such that (i) $\Sigma^{(0)} \cup H \subseteq T$ and (ii) if $k \geq 1$, $\sigma \in \Sigma^{(k)}$, and $\xi_1, \ldots, \xi_k \in T$, then $\sigma(\xi_1, \ldots, \xi_k) \in T$. We denote $T_\Sigma(\emptyset)$ by T_Σ; obviously $T_\Sigma \neq \emptyset$ iff $\Sigma^{(0)} \neq \emptyset$. If H is finite, then we will also view $T_\Sigma(H)$ as $T_{\Sigma \cup H}$ where $(\Sigma \cup H)^{(0)} = \Sigma^{(0)} \cup H$ and $(\Sigma \cup H)^{(k)} = \Sigma^{(k)}$ for every $k \geq 1$. Since terms can be depicted in a very illustrative way as trees, i.e., particular graphs, it has become a custom to call Σ-terms also *Σ-trees*. In this chapter, we follow this custom. Every subset $L \subseteq T_\Sigma$ is called a *Σ-tree language*. Frequently, we will consider a tree $\xi \in T_\Sigma$ which has the form $\xi = \sigma(\xi_1, \ldots, \xi_k)$ for some $k \geq 0$, $\sigma \in \Sigma^{(k)}$, and $\xi_1, \ldots, \xi_k \in T_\Sigma$. Whenever we use this notation, for $k = 0$, the string $\sigma(\xi_1, \ldots, \xi_k)$ stands for σ, rather than $\sigma()$. In order to avoid repetition of the quantifications of k, σ, and $\xi_1, \ldots, \xi_k$, we henceforth only write that we consider a $\xi \in T_\Sigma$ of the form $\xi = \sigma(\xi_1, \ldots, \xi_k)$.

In the rest of this chapter, Σ and Δ will denote arbitrary ranked alphabets if not specified otherwise. Moreover, we assume that $\Sigma^{(0)} \neq \emptyset$ and $\Delta^{(0)} \neq \emptyset$.

We define the *height*, *size*, and *set of positions* of trees as the functions height : $T_\Sigma \to \mathbb{N}$, size : $T_\Sigma \to \mathbb{N}$, and pos : $T_\Sigma \to \mathcal{P}(\mathbb{N}^*)$, respectively, as follows: (i) for every $\alpha \in \Sigma^{(0)}$, we define $\mathrm{height}(\alpha) = 0$, $\mathrm{size}(\alpha) = 1$, and $\mathrm{pos}(\alpha) = \{\varepsilon\}$, and (ii) for every $\xi = \sigma(\xi_1, \ldots, \xi_k)$, where $k \geq 1$, we define $\mathrm{height}(\xi) = 1 + \mathsf{max}\{\mathrm{height}(\xi_i) \mid 1 \leq i \leq k\}$, $\mathrm{size}(\xi) = 1 + \sum_{1 \leq i \leq k} \mathrm{size}(\xi_i)$, and $\mathrm{pos}(\xi) = \{\varepsilon\} \cup \{iv \mid 1 \leq i \leq k, v \in \mathrm{pos}(\xi_i)\}$.

Now let $\xi, \zeta \in T_\Sigma$ and $w \in \mathrm{pos}(\xi)$. The *label of ξ at w*, denoted by $\xi(w)$, the *subtree of ξ at w*, denoted by $\xi|_w$, and the *replacement of the subtree of ξ at w by ζ*, denoted by $\xi[\zeta]_w$ are defined as follows: (i) for every $\alpha \in \Sigma^{(0)}$, we define $\alpha(\varepsilon) = \alpha$, $\alpha|_\varepsilon = \alpha$, and $\alpha[\zeta]_\varepsilon = \zeta$, and (ii) for every $\xi = \sigma(\xi_1, \ldots, \xi_k)$ with $k \geq 1$, we define $\xi(\varepsilon) = \sigma$, $\xi|_\varepsilon = \xi$, and $\xi[\zeta]_\varepsilon = \zeta$, and for every $1 \leq i \leq k$ and $v \in \mathrm{pos}(\xi_i)$, we define $\xi(iv) = \xi_i(v)$, $\xi|_{iv} = \xi_i|_v$, and $\xi[\zeta]_{iv} = \sigma(\xi_1, \ldots, \xi_{i-1}, \xi_i[\zeta]_v, \xi_{i+1}, \ldots, \xi_k)$. For a subset $Q \subseteq \Sigma$, we define $\mathrm{pos}_Q : T_\Sigma \to \mathcal{P}(\mathbb{N}^*)$ by $\mathrm{pos}_Q(\xi) = \{w \in \mathrm{pos}(\xi) \mid \xi(w) \in Q\}$.

We will often use the notion of variable. Let $Z = \{z_1, z_2, \ldots\}$ be a set of variables, disjoint with Σ, and $Z_k = \{z_1, \ldots, z_k\}$ for every $k \geq 0$.

Next, we define *tree substitution*. Let H be a set disjoint with Σ. For $\xi \in T_\Sigma(Z \cup H)$, a finite set $I \subseteq \mathbb{N}$, and a family $(\xi_i \mid i \in I)$ with $\xi_i \in T_\Sigma(H)$, the expression $\xi(\xi_i \mid i \in I)$ denotes the result of substituting in ξ every occurrence of z_i by ξ_i for every $i \in I$. In case $I = \{1, \ldots, n\}$, we write $\xi(\xi_1, \ldots, \xi_n)$. Moreover, if $I = \{1\}$ and $z = z_1$, then we write $\xi \cdot_z \xi_1$ instead of $\xi(\xi_1)$. The operation $\cdot_z$ is associative in the sense that for every $\xi' \in T_\Sigma(Z \cup H)$ we have $(\xi \cdot_z \xi') \cdot_z \xi_1 = \xi \cdot_z (\xi' \cdot_z \xi_1)$.

2.3 Algebraic Concepts

In this chapter, we will often denote an algebraic structure just by its carrier set, if its operations are clear from the context.

Let $(S, +, 0)$ be a commutative monoid. Then S is *naturally ordered* if the binary relation $\sqsubseteq$ on S is a partial order on S, where $\sqsubseteq$ is defined by $a \sqsubseteq b$ iff there is a $c \in S$ such that $a + c = b$. A monoid S is *locally finite* if, for every finite $S' \subseteq S$, the sub-monoid of S generated by S' is finite.

An *infinitary sum operation* $\sum$ associates with every countable index set I and family $(a_i \mid i \in I)$ of elements $a_i \in S$ an element $\sum_{i \in I} a_i$. If $\sum$ is commutative, associative, and extends $+$, then S is a *$\sum$-complete monoid* (cf., e.g., [76, 88, 40]); in particular, $\sum_{i \in \emptyset} a_i = 0$. A $\sum$-complete and naturally ordered monoid S is *$\sum$-continuous* if, for every I, family $(a_i \mid i \in I)$, and $b \in S$, the following implication holds: if $\sum_{i \in E} a_i \sqsubseteq b$ for every finite subset E of I, then $\sum_{i \in I} a_i \sqsubseteq b$. We call a monoid *complete* (resp., *continuous*) if there is an infinitary sum operation $\sum$ such that S is $\sum$-complete (resp., $\sum$-continuous).

A *semiring* $(S, +, \cdot, 0, 1)$ is an algebra which consists of a commutative monoid $(S, +, 0)$, called the additive monoid of S, and a monoid $(S, \cdot, 1)$, called the multiplicative monoid of S, such that multiplication distributes (from left and right) over addition, and moreover, $0 \neq 1$ and 0 is absorbing with respect to $\cdot$ (also from left and right). We call S *idempotent* if $a + a = a$; *zero-sum free* if $a + b = 0$ implies $a = b = 0$; *commutative* if its multiplicative monoid is commutative; *zero-divisor free* if $a \cdot b = 0$ implies $a = 0$ or $b = 0$ for every $a, b \in S$; *positive* if it is zero-sum free and zero-divisor free. Finally, S is *locally finite* if, for every finite $S' \subseteq S$, the subsemiring of S generated by S' is finite.

Let S be commutative and $(a_i \mid i \in I)$ be a finite family of elements $a_i \in S$. Then we denote the product of all the elements of the family by $\prod_{i \in I} a_i$; in particular, we have that $\prod_{i \in \emptyset} a_i = 1$.

Let Q be a finite set and $u, v \in S^Q$ two Q-vectors over S. Then we define the *inner product* of u and v as $u \cdot v = \sum_{q \in Q} u_q \cdot v_q$.

In the rest of this chapter, S will denote an arbitrary semiring $(S, +, \cdot, 0, 1)$ if not specified otherwise.

Among other semirings, we consider the following particular ones: the Boolean semiring $(\mathbb{B}, \vee, \wedge, 0, 1)$ where $\mathbb{B} = \{0, 1\}$, the semiring $\mathsf{Nat} = (\mathbb{N}, +, \cdot, 0, 1)$ of natural numbers, the arctic semiring $\mathsf{Arct} = (\mathbb{N} \cup \{-\infty\}, \mathsf{max}, +, -\infty, 0)$, the tropical semiring $\mathsf{Trop} = (\mathbb{N} \cup \{\infty\}, \mathsf{min}, +, \infty, 0)$, and the semiring of formal languages $\mathsf{Lang}_A = (\mathcal{P}(A^*), \cup, \cdot, \emptyset, \{\varepsilon\})$ over any set A (where $L \cdot L' = \{uv \mid u \in L, v \in L'\}$ for languages L, L').

Again, let $\sum$ be an infinitary sum operation. The semiring S is $\sum$-*complete* if its additive monoid is $\sum$-complete and the following distributive laws hold: $c \cdot (\sum_{i \in I} a_i) = \sum_{i \in I} (c \cdot a_i)$ and $(\sum_{i \in I} a_i) \cdot c = \sum_{i \in I} (a_i \cdot c)$ for every $c \in S$ and family $(a_i \mid i \in I)$ with $a_i \in S$. Moreover, S is *naturally ordered* if its additive monoid is naturally ordered, and S is $\sum$-*continuous* if it is $\sum$-complete and its additive monoid is $\sum$-continuous.

We call S a *ring (with unit)* if $(S, +, 0)$ is a group; the additive inverse of $a \in S$ is denoted by $-a$. A semiring which is not a ring is called *proper*. Thus, every positive semiring is proper. We call S a *semifield* if $(S \setminus \{0\}, \cdot, 1)$ is a commutative group, i.e., every element $a \in S \setminus \{0\}$ has an inverse, which we denote by a^{-1}. Moreover, S is a *field* if it is a ring and a semifield.

For a semiring S, an S-*semimodule* is a commutative monoid $(V, +, 0)$ equipped with a scalar multiplication $\circ : S \times V \to V$ satisfying the following laws:

$$\begin{aligned}
(a \cdot a') \circ v &= a \circ (a' \circ v), \\
a \circ (v + v') &= (a \circ v) + (a \circ v'), \\
(a + a') \circ v &= (a \circ v) + (a' \circ v), \\
1 \circ v &= v, \\
a \circ 0 &= 0 \circ v = 0
\end{aligned}$$

for every $a, a' \in S$ and $v, v' \in V$ (cf. [71], page 149). Note that the symbols $+$ and 0 are overloaded because they denote operations over both S and V. Also, at other places in this chapter, such overloading of symbols may occur. However, it will always be clear from the context which operation is meant. As usual, we drop $\circ$ from $a \circ v$ and just write av.

An S-semimodule $(V, +, 0)$ with scalar multiplication $\circ$ is *complete* if the monoid $(V, +, 0)$ is complete, say, with the infinitary sum $\sum$, and $a \circ \sum_{i \in I} v_i = \sum_{i \in I} (a \circ v_i)$ holds for every $a \in S$ and family $(v_i \mid i \in I)$ with $v_i \in V$.

Let V and V' be S-semimodules. A mapping $f : V^k \to V'$ is *multilinear* if $f(v_1, \ldots, v_{i-1}, au + bv, v_{i+1}, \ldots, v_k) = af(v_1, \ldots, v_{i-1}, u, v_{i+1}, \ldots, v_k) + bf(v_1, \ldots, v_{i-1}, v, v_{i+1}, \ldots, v_k)$ for every $1 \le i \le k$, $u, v, v_1, \ldots, v_k \in V$, and $a, b \in S$. A multilinear unary mapping $f : V \to V'$ is called *linear*.

If S is a field and $(V, +, 0)$ is a commutative group, then the S-semimodule $(V, +, 0)$ is an *S-vector space*. Later, we will consider the particular S-vector space $(S^Q, +, 0)$ where Q is a finite set. The zero element 0 is the vector with $0_q = 0$ for every $q \in Q$. Moreover, for every $u, v \in S^Q$, $q \in Q$, and $a \in S$, we have that $(u + v)_q = u_q + v_q$ and $(au)_q = a \cdot u_q$. We call a linear mapping $\gamma : V \to S$ (where S is viewed as S-vector space) a *linear form*.

Let V be an S-vector space. The vectors $v_1, \ldots, v_m \in V$ are *linearly independent* if, for every $a_1, \ldots, a_m \in S$, the equality $a_1 v_1 + \cdots + a_m v_m = 0$ implies that $a_1 = \cdots = a_m = 0$. A subset V' of V is linearly independent if the vectors in every finite subset of V' are linearly independent. Moreover, V' *generates* V if, for every $v \in V$, there are $m \ge 1$, $v_i \in V'$ and $a_i \in S$ for $1 \le i \le m$ such that $v = a_1 v_1 + \cdots + a_m v_m$. Finally, V' is a *basis of* V if it is linearly independent and generates V. If V admits a basis consisting of $\kappa \in \mathbb{N}$ elements, then it is called *κ-dimensional*; V is *finite-dimensional* if it is κ-dimensional for some $\kappa \in \mathbb{N}$. In a finite-dimensional vector space each basis has the same number of elements.

A *Σ-algebra* (V, θ) consists of a nonempty set V (*carrier set*) and an arity preserving interpretation θ of symbols from Σ as operations over V, i.e., $\theta(\sigma) : V^k \to V$ for every $k \ge 0$ and $\sigma \in \Sigma^{(k)}$. The *Σ-term algebra* (T_Σ, top), defined by $\mathrm{top}(\sigma)(\xi_1, \ldots, \xi_k) = \sigma(\xi_1, \ldots, \xi_k)$, is *initial* in the class of all Σ-algebras, i.e., for every Σ-algebra (V, θ), there is a unique Σ-algebra homomorphism from T_Σ to V, which we denote by h_V (if not specified otherwise). That means that for every $\sigma(\xi_1, \ldots, \xi_k) \in T_\Sigma$ we have $h_V(\sigma(\xi_1, \ldots, \xi_k)) = \theta(\sigma)(h_V(\xi_1), \ldots, h_V(\xi_k))$. Now let (V, θ) be a Σ-algebra and z a nullary symbol such that $z \notin \Sigma$. For every $v \in V$, we define the *v-extension of* (V, θ) to be the $\Sigma \cup \{z\}$-algebra (V, θ^v) where $\theta^v(z) = v$ and $\theta^v(\sigma) = \theta(\sigma)$ for every $k \ge 0$ and $\sigma \in \Sigma^{(k)}$. We denote the unique $\Sigma \cup \{z\}$-algebra homomorphism from $T_{\Sigma \cup \{z\}}$ to V by h_V^v. For more details, we refer to [73, 70, 139].

An *S-Σ-semimodule* $(V, +, 0, \theta)$ consists of an S-semimodule $(V, +, 0)$ and a Σ-algebra (V, θ) where $\theta(\sigma)$ is multilinear for every $\sigma \in \Sigma$. If S is a field and $(V, +, 0)$ is an S-vector space, then we call $(V, +, 0, \theta)$ an *S-Σ-vector space*. An S-Σ-semimodule $(V, +, 0, \theta)$ is *complete* if the S-semimodule $(V, +, 0)$ is complete and for every $\sigma \in \Sigma$, the operation $\theta(\sigma)$ preserves infinite sums in each of its arguments, i.e.,

$$\theta(\sigma)\Big(\dots, \sum_{i\in I} v_i, \dots\Big) = \sum_{i\in I} \theta(\sigma)(\dots, v_i, \dots).$$

Moreover, a complete S-Σ-semimodule $(V, +, 0, \theta)$ is *continuous* if $(V, +, 0)$ is continuous.

2.4 Tree Series

Let H be a set with $\Sigma \cap H = \emptyset$. A *tree series over* Σ*,* H*, and* S (or for short: tree series) is a mapping $r : T_\Sigma(H) \to S$. For every $\xi \in T_\Sigma(H)$, the element $r(\xi) \in S$ is called the *coefficient* of ξ and it is denoted by (r, ξ). Moreover, the tree series r is written as the formal sum $\sum_{\xi\in T_\Sigma(H)}(r, \xi).\xi$.

The *support of the tree series* r is defined as the set $\mathrm{supp}(r) = \{\xi \in T_\Sigma(H) \mid (r, \xi) \neq 0\}$. Moreover, r is *polynomial* (resp., a *monomial*) if $\mathrm{supp}(r)$ is finite (resp., a singleton). We will denote a polynomial r by $a_1.\xi_1 + \cdots + a_k.\xi_k$, where $\mathrm{supp}(r) = \{\xi_1, \dots, \xi_k\}$ and $a_i = (r, \xi_i)$ for $1 \le i \le k$. The set of all (resp., polynomial) tree series is denoted by $S\langle\langle T_\Sigma(H)\rangle\rangle$ (resp., $S\langle T_\Sigma(H)\rangle$).

Let $r \in S\langle\langle T_\Sigma(H)\rangle\rangle$ be a tree series. We call r *Boolean* if $(r, \xi) \in \{0, 1\}$ holds for every $\xi \in T_\Sigma(H)$. If there is an $a \in S$ such that for every $\xi \in T_\Sigma(H)$, we have $(r, \xi) = a$, then r is a *constant* and also denoted by $\widetilde{a}$. Note that the constants $\widetilde{0}$ and $\widetilde{1}$ are Boolean.

For a set A and $B \subseteq A$, the *characteristic function of* B *with respect to* S is the mapping $1_{(S,B)} : A \to S$ such that $1_{(S,B)}(a) = 1$ if $a \in B$, and $1_{(S,B)}(a) = 0$ otherwise for every $a \in A$. For a tree language $L \subseteq T_\Sigma$, we call the tree series $1_{(S,L)}$ the *characteristic tree series of* L *with respect to* S. Certainly, we have $\mathrm{supp}(1_{(S,L)}) = L$.

3 Weighted Tree Automata

3.1 Bottom-up Tree Automata

The theory of finite-state string automata and of recognizable string languages has been successfully generalized to trees. For instance, the class of recognizable tree languages is characterized by solutions of linear equations [116], rational expressions [135], monadic-second order logic [135, 33], congruences of finite index [26, 100], and finitely generated congruences [63, 86]. An excellent, detailed survey on recognizable tree languages and finite-state tree automata can be found in [68, 69] (also cf. [29]). Let us recall here the concept of a finite-state bottom-up tree automaton.

A *(finite-state) bottom-up tree automaton* is a tuple $\mathcal{A} = (Q, \Sigma, \delta, F)$, where Q is a finite nonempty set (*states*), δ is a Σ-indexed family $(\delta_\sigma \mid \sigma \in \Sigma)$

where $\delta_\sigma \subseteq Q^k \times Q$ for $\sigma \in \Sigma^{(k)}$ (*set of transitions at* σ), and $F \subseteq Q$ (*final states*). We call Σ the *input ranked alphabet*, and an element $\xi \in T_\Sigma$ an *input tree*. Here and in the rest of the chapter, we view Q^k as the set of strings over Q of length k. To define the semantics of $\mathcal{A}$, we consider the Σ-algebra $(\mathcal{P}(Q), \delta_\mathcal{A})$ with $\delta_\mathcal{A}(\sigma)(P_1, \ldots, P_k) = \{q \in Q \mid \exists (q_1 \ldots q_k, q) \in \delta_\sigma : q_i \in P_i$ for every $1 \leq i \leq k\}$ for every $P_1, \ldots, P_k \in \mathcal{P}(Q)$. *The tree language recognized by* $\mathcal{A}$ is $L_\mathcal{A} = \{\xi \in T_\Sigma \mid h_{\mathcal{P}(Q)}(\xi) \cap F \neq \emptyset\}$. The class of all *recognizable Σ-tree languages* is denoted by $\mathrm{Rec}(\Sigma)$.

It is well known that every bottom-up tree automaton can be transformed into an equivalent deterministic one. A bottom-up tree automaton is *(total) deterministic* if the relation δ_σ is a total function for every $\sigma \in \Sigma$. Then we view $h_{\mathcal{P}(Q)}$ as a mapping of type $T_\Sigma \to Q$ and write h_Q rather than $h_{\mathcal{P}(Q)}$.

Example 3.1. Consider the ranked alphabet $\Sigma = \{\sigma^{(2)}, \gamma^{(1)}, \alpha^{(0)}\}$ and the pattern $\sigma(z, \alpha)$. We say that the pattern $\sigma(z, \alpha)$ occurs in a tree $\xi \in T_\Sigma$ if there is a position $w \in \mathrm{pos}(\xi)$ and a tree $\xi' \in T_\Sigma$ such that $\xi = \xi[\sigma(\xi', \alpha)]_w$. We construct a bottom-up tree automaton $\mathcal{A}$ such that $L_\mathcal{A}$ is the set of all Σ-trees in which $\sigma(z, \alpha)$ occurs at least once. The automaton performs a nondeterministic guess-and-verify strategy; it selects nondeterministically an occurrence of α and verifies whether it is a right child of a σ. For this, let $\mathcal{A} = (Q, \Sigma, \delta, F)$ be defined by $Q = \{q, \overline{\alpha}, f\}$, $F = \{f\}$, and $\delta_\alpha = \{(\varepsilon, q), (\varepsilon, \overline{\alpha})\}$, $\delta_\gamma = \{(q, q), (f, f)\}$, and $\delta_\sigma = \{(qq, q), (qf, f), (fq, f), (q\overline{\alpha}, f)\}$. Then for every $\xi \in T_\Sigma$, we have that $f \in h_{\mathcal{P}(Q)}(\xi)$ iff the pattern $\sigma(z, \alpha)$ occurs in ξ.

3.2 Recognizable Tree Series

Bottom-up tree automata can be reformulated such that the reformulation easily leads to the concept of weighted tree automata. The idea behind this is to represent a tree language $L \subseteq T_\Sigma$ as a characteristic tree series $1_{(\mathbb{B}, L)} : T_\Sigma \to \mathbb{B}$ and then, in a second step, to replace the Boolean semiring $\mathbb{B}$ by S.

Consider now the system $\mathcal{A} = (Q, \Sigma, \mathbb{B}, \mu, \nu)$, called a *weighted tree automaton over* $\mathbb{B}$, where Q is as in Sect. 3.1, while μ is a family $(\mu_k : \Sigma^{(k)} \to \mathbb{B}^{Q^k \times Q} \mid k \geq 0)$ of mappings and $\nu \in \mathbb{B}^Q$ is a Q-vector over $\mathbb{B}$. We define the semantics of $\mathcal{A}$ as a mapping $r_\mathcal{A} : T_\Sigma \to \mathbb{B}$ in the following way. Let us introduce the Σ-algebra $(\mathbb{B}^Q, \mu_\mathcal{A})$, where

$$\mu_\mathcal{A}(\sigma)(v_1, \ldots, v_k)_q = \bigvee_{q_1, \ldots, q_k \in Q} (v_1)_{q_1} \wedge \cdots \wedge (v_k)_{q_k} \wedge \mu_k(\sigma)_{q_1 \ldots q_k, q},$$

for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, and $v_1, \ldots, v_k \in \mathbb{B}^Q$. Now let

$$r_\mathcal{A}(\xi) = h_\mu(\xi) \wedge \nu$$

for every $\xi \in T_\Sigma$, where h_μ is the unique Σ-homomorphism from the Σ-term algebra T_Σ to $(\mathbb{B}^Q, \mu_\mathcal{A})$. (Recall that, according to the notion of inner product of Q-vectors over $\mathbb{B}$ from Sect. 2.3, $h_\mu(\xi) \wedge \nu = \bigvee_{q \in Q} h_\mu(\xi)_q \wedge \nu_q$.)

It should be clear that bottom-up tree automata and weighted tree automata over $\mathbb{B}$ are semantically equivalent: for every bottom-up tree automaton $\mathcal{A} = (Q, \Sigma, \delta, F)$ one can construct the weighted tree automaton $\mathcal{B} = (Q, \Sigma, \mathbb{B}, \mu, \nu)$, where $\mu_k(\sigma) = 1_{(\mathbb{B},\delta_\sigma)}$ and $\nu = 1_{(\mathbb{B},F)}$. Then $r_\mathcal{B} = 1_{(\mathbb{B},L_\mathcal{A})}$. Clearly, the construction can be reversed.

Now we observe that weighted tree automata over $\mathbb{B}$ can be generalized to weighted tree automata over S in an obvious way: then S, $+$, and $\cdot$ take over the role of $\mathbb{B}$, $\vee$, and $\wedge$, respectively. The semantics of a weighted tree automaton $\mathcal{A}$ over S will be a tree series $r_\mathcal{A} : T_\Sigma \to S$. We expect that weighted tree automata can compute tree series like:

- height : $T_\Sigma \to \mathbb{N}$ over Arct,
- $\mathrm{size}_\delta : T_\Sigma \to \mathbb{N}$ and size : $T_\Sigma \to \mathbb{N}$ over Nat and also over Trop, where $\mathrm{size}_\delta(\xi) = |\mathrm{pos}_{\{\delta\}}(\xi)|$,
- $\#_{\sigma(z,\alpha)} : T_\Sigma \to \mathbb{N}$ over Nat, where $\#_{\sigma(z,\alpha)}(\xi)$ is the number of occurrences of the pattern $\sigma(z, \alpha)$ in ξ,
- $\mathrm{shortest}_\alpha : T_\Sigma \to \mathbb{N}$ over Trop, where $\mathrm{shortest}_\alpha(\xi)$ is the length of a shortest path in ξ from its root to one of its leaves with label α,
- yield : $T_\Sigma \to \mathcal{P}(\Sigma^*)$ over Lang_Σ, where yield(ξ) is the concatenation of the nullary symbols occurring in ξ from left to right,
- revpos : $T_\Sigma \to \mathcal{P}(\mathbb{N}^*)$ over $\mathsf{Lang}_\mathbb{N}$, where revpos(ξ) is the set of reversals of elements in pos(ξ),
- $\mathrm{revpos}_{\sigma(z,\alpha)} : T_\Sigma \to \mathcal{P}(\mathbb{N}^*)$ over $\mathsf{Lang}_\mathbb{N}$, where $\mathrm{revpos}_{\sigma(z,\alpha)}(\xi)$ is the set of reversals of positions of ξ at which the pattern $\sigma(z, \alpha)$ occurs; note that $\#_{\sigma(z,\alpha)}(\xi)$ is the cardinality of $\mathrm{revpos}_{\sigma(z,\alpha)}(\xi)$.

Now let us start with the formal definition of weighted tree automata. We follow the approach of [1], where this model was called an S-Σ-tree automaton.

Definition 3.2. *A* weighted tree automaton (over S) *(for short: wta) is a tuple $\mathcal{A} = (Q, \Sigma, S, \mu, \nu)$ where:*

- *Q is a finite nonempty set, the* set of states.
- *Σ is the* ranked input alphabet.
- *$\mu = (\mu_k \mid k \in \mathbb{N})$ is a* family of transition mappings[3] $\mu_k : \Sigma^{(k)} \to S^{Q^k \times Q}$.
- *$\nu \in S^Q$ is a Q-vector over S, the* root weight vector.

For every transition $(w, q) \in Q^k \times Q$, the element $\mu_k(\sigma)_{w,q} \in S$ is the weight of (w, q). We denote the set $\{\mu_k(\sigma)_{w,q} \mid k \geq 0, \sigma \in \Sigma^{(k)}, w \in Q^k, q \in Q\} \cup \{\nu_q \mid q \in Q\}$ of all weights which occur in $\mathcal{A}$, by wts$(\mathcal{A})$. Note that wts$(\mathcal{A}) \subseteq S$.

For a wta $\mathcal{A}$, we consider the Σ-algebra $(S^Q, \mu_\mathcal{A})$ where, for every $k \geq 0$ and $\sigma \in \Sigma^{(k)}$, the k-ary operation $\mu_\mathcal{A}(\sigma) : S^Q \times \cdots \times S^Q \to S^Q$ is defined by

$$\mu_\mathcal{A}(\sigma)(v_1, \ldots, v_k)_q = \sum_{q_1,\ldots,q_k \in Q} (v_1)_{q_1} \cdot \cdots \cdot (v_k)_{q_k} \cdot \mu_k(\sigma)_{q_1 \ldots q_k, q}$$

[3] In the literature, μ is also called a *tree representation.*

for every $q \in Q$ and $v_1, \ldots, v_k \in S^Q$. Let us denote here the unique Σ-algebra homomorphism from T_Σ to S^Q by h_μ. The tree series $r_\mathcal{A} \in S\langle\langle T_\Sigma \rangle\rangle$ *recognized* by $\mathcal{A}$ is defined by

$$(r_\mathcal{A}, \xi) = h_\mu(\xi) \cdot \nu$$

for every $\xi \in T_\Sigma$. (Again recall that, according to the notion of inner product of Q-vectors over S from Sect. 2.3, $h_\mu(\xi) \cdot \nu = \sum_{q \in Q} h_\mu(\xi)_q \cdot \nu_q$.) A tree series $r \in S\langle\langle T_\Sigma \rangle\rangle$ is *recognizable* if there is a wta $\mathcal{A}$ such that $r = r_\mathcal{A}$. The class of all tree series over Σ and S which are recognizable is denoted by $\mathrm{Rec}(\Sigma, S)$.

Due to the definitions of $\mu_\mathcal{A}$ and h_μ, we can observe that

$$h_\mu\big(\sigma(\xi_1, \ldots, \xi_k)\big)_q = \sum_{q_1,\ldots,q_k \in Q} h_\mu(\xi_1)_{q_1} \cdot \cdots \cdot h_\mu(\xi_k)_{q_k} \cdot \mu_k(\sigma)_{q_1 \ldots q_k, q}$$

for every $\sigma(\xi_1, \ldots, \xi_k) \in T_\Sigma$ and $q \in Q$.

Example 3.3. We construct a wta $\mathcal{A} = (Q, \Sigma, \mathsf{Arct}, \mu, \nu)$ which recognizes the tree series height. Let $Q = \{p_1, p_2\}$, $\Sigma = \{\sigma^{(2)}, \alpha^{(0)}\}$, and $\nu_{p_1} = 0$ and $\nu_{p_2} = -\infty$. Moreover, let

$$\begin{aligned} \mu_0(\alpha)_{\varepsilon,p_1} &= \mu_0(\alpha)_{\varepsilon,p_2} = 0, \\ \mu_2(\sigma)_{p_1p_2,p_1} &= \mu_2(\sigma)_{p_2p_1,p_1} = 1, \\ \mu_2(\sigma)_{p_2p_2,p_2} &= 0, \end{aligned}$$

and for every other transition (q_1q_2, q) we have $\mu_2(\sigma)_{q_1q_2,q} = -\infty$. We consider the tree $\xi = \sigma(\alpha, \alpha)$ and compute $h_\mu(\xi)_{p_1}$ and $h_\mu(\xi)_{p_2}$. Clearly, $h_\mu(\alpha)_{p_1} = \mu_0(\alpha)_{\varepsilon,p_1} = 0$ and $h_\mu(\alpha)_{p_2} = 0$. Then

$$h_\mu(\sigma(\alpha, \alpha))_{p_1} = \max_{q_1,q_2 \in Q}\{h_\mu(\alpha)_{q_1} + h_\mu(\alpha)_{q_2} + \mu_2(\sigma)_{q_1q_2,p_1}\} = 1$$

(note that $\mu_2(\sigma)_{p_1p_1,p_1} = \mu_2(\sigma)_{p_2p_2,p_1} = -\infty$ and $-\infty$ is unit for max) and similarly, $h_\mu(\sigma(\alpha, \alpha))_{p_2} = 0$. In general, we can prove by structural induction on ξ that $h_\mu(\xi)_{p_1} = \mathrm{height}(\xi)$ and $h_\mu(\xi)_{p_2} = 0$ for every $\xi \in T_\Sigma$. Thus, $r_\mathcal{A} = \mathrm{height}$, and hence $\mathrm{height} \in \mathrm{Rec}(\Sigma, \mathsf{Arct})$.

We have defined recognizable tree series in an initial algebra semantics style [70]. An alternative way is to define the semantics of a wta by means of its runs. A *run of* $\mathcal{A}$ *on* $\xi \in T_\Sigma$ is a mapping $\kappa : \mathrm{pos}(\xi) \to Q$; the *set of all runs of* $\mathcal{A}$ *on* ξ is denoted by $R_\mathcal{A}(\xi)$. For every $\kappa \in R_\mathcal{A}(\xi)$ and $w \in \mathrm{pos}(\xi)$, the *run induced by* κ *at position* w is the run $\kappa|_w \in R_\mathcal{A}(\xi|_w)$ and defined for every $w' \in \mathrm{pos}(\xi|_w)$ by $\kappa|_w(w') = \kappa(ww')$. For every $\xi = \sigma(\xi_1, \ldots, \xi_k) \in T_\Sigma$, the *weight* $\mathrm{wt}(\kappa)$ *of* κ is $\mathrm{wt}(\kappa) = \mathrm{wt}(\kappa|_1) \cdot \cdots \cdot \mathrm{wt}(\kappa|_k) \cdot \mu_k(\sigma)_{\kappa(1)\ldots\kappa(k),\kappa(\varepsilon)}$. The *run semantics of* $\mathcal{A}$ is the tree series $r^{\mathrm{run}}_\mathcal{A} \in S\langle\langle T_\Sigma \rangle\rangle$ such that

$$\big(r^{\mathrm{run}}_\mathcal{A}, \xi\big) = \sum_{\kappa \in R_\mathcal{A}(\xi)} \mathrm{wt}(\kappa) \cdot \nu_{\kappa(\varepsilon)}$$

for every $\xi \in T_\Sigma$. In fact, for every wta $\mathcal{A}$, the run semantics of $\mathcal{A}$ and the tree series recognized by $\mathcal{A}$ are the same. More precisely, the equation $h_\mu(\xi)_q = \sum_{\kappa \in R_{\mathcal{A}}(\xi), \kappa(\varepsilon)=q} \mathrm{wt}(\kappa)$ holds for every $\xi \in T_\Sigma$ and $q \in Q$, which easily implies $r_{\mathcal{A}} = r_{\mathcal{A}}^{\mathrm{run}}$.

Example 3.4. We construct a wta $\mathcal{A}'$ which recognizes the tree series $\#_{\sigma(z,\alpha)} : T_\Sigma \to \mathbb{N}$. This generalizes Example 3.1 in the sense that we not only consider whether the pattern $\sigma(z,\alpha)$ occurs in ξ, but also compute the number of those occurrences. For this, recall the bottom-up tree automaton $\mathcal{A} = (Q, \Sigma, \delta, F)$ of Example 3.1 and construct $\mathcal{A}' = (Q, \Sigma, \mathsf{Nat}, \mu, \nu)$ such that $\mu_k(\theta) = 1_{(\mathsf{Nat},\delta_\theta)}$ for every $k \geq 0$, $\theta \in \Sigma^{(k)}$; moreover, let $\nu = 1_{(\mathsf{Nat},F)}$.

Then for every $\xi \in T_\Sigma$ and run $\kappa \in R_{\mathcal{A}}(\xi)$ with $\kappa(\varepsilon) = f$, the weight $\mathrm{wt}(\kappa)$ is 1 iff at exactly one occurrence of σ the transition $(q\overline{\alpha}, f)$ was applied (and $\mathrm{wt}(\kappa) = 0$ otherwise). Since the application of this transition indicates an occurrence of the pattern $\sigma(z,\alpha)$, we have that $(r_{\mathcal{A}}^{\mathrm{run}}, \xi) = \#_{\sigma(z,\alpha)}(\xi)$. Thus, $\#_{\sigma(z,\alpha)} \in \mathrm{Rec}(\Sigma, \mathsf{Nat})$.

In fact, wta generalize in a natural way weighted finite automata as they are presented, e.g., in Part I of this handbook. Here, we follow the formal approach of [39], where a weighted finite automaton over a semiring S and an (unranked) alphabet Γ is a tuple $\mathcal{A} = (Q, \lambda, \mu, \gamma)$ and Q is a finite set of states, $\mu : \Gamma \to S^{Q \times Q}$ is the transition weight function, and $\lambda, \gamma \in S^Q$ are weight functions for entering and leaving a state. The behavior $\|\mathcal{A}\| : \Gamma^* \to S$ of $\mathcal{A}$ associates with every word $w = a_1 a_2 \ldots a_n \in \Gamma^*$ the value $(\|\mathcal{A}\|, w) = \sum_{p \in P(w)} \mathrm{wt}(p)$ where $P(w)$ is the set of all paths with label w, and $\mathrm{wt}(p)$ is the weight of p defined to be $\lambda_{q_0} \cdot \mu(a_1)_{q_0,q_1} \cdot \cdots \cdot \mu(a_n)_{q_{n-1},q_n} \cdot \gamma_{q_n}$ assuming that p has the form $q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_{n-1}} q_{n-1} \xrightarrow{a_n} q_n$. Let $\mathrm{Rec}_{\mathrm{w}}(\Gamma, S)$ denote the class of all behaviors of weighted finite automata, i.e., of all recognizable formal power series over Γ and S.

By rotating a word w counterclockwise by $90°$, we obtain the tree $\mathrm{tree}(w)$. Formally, let Γ_t be the ranked alphabet $\{a^{(1)} \mid a \in \Gamma\} \cup \{e^{0)}\}$, and $\mathrm{tree} : \Gamma^* \to T_{\Gamma_t}$ be defined by $\mathrm{tree}(\varepsilon) = e$ and $\mathrm{tree}(wa) = a(\mathrm{tree}(w))$ for every $a \in \Gamma$ and $w \in \Gamma^*$. Clearly, tree is a bijection; moreover, $L \subseteq \Gamma^*$ is a recognizable language iff the tree language $\mathrm{tree}(L)$ is recognizable. Then, given a weighted finite automaton $\mathcal{A} = (Q, \lambda, \mu, \gamma)$, we can construct the wta $\mathcal{A}_t = (Q, \Gamma_t, S, \theta, \gamma)$ over S and Γ_t where $\theta_0(e)_{\varepsilon,q} = \lambda_q$ and $\theta_1(a)_{q,p} = \mu(a)_{q,p}$ for every $a \in \Gamma$ and $q, p \in Q$. It is obvious that $\|\mathcal{A}\| = r_{\mathcal{A}_t}^{\mathrm{run}} \circ \mathrm{tree}$. Vice versa, given a wta $\mathcal{B}$ over some ranked alphabet Γ_t which results from an (unranked) alphabet Γ, the wta $\mathcal{B}$ can be viewed in an obvious way as a weighted finite automaton $\mathcal{B}'$ such that $r_{\mathcal{B}}^{\mathrm{run}} = \|\mathcal{B}'\| \circ \mathrm{tree}^{-1}$. By extending the mapping tree in the usual way to languages and classes of languages, we obtain that $\mathrm{tree}(\mathrm{Rec}_{\mathrm{w}}(\Gamma, S)) = \mathrm{Rec}(\Gamma_t, S)$ for every alphabet Γ.

As first type of restriction on wta, we define deterministic wta. A wta $\mathcal{A} = (Q, \Sigma, S, \mu, \nu)$ is:

- *bottom-up deterministic* (for short: bu-deterministic) if for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, and $w \in Q^k$ there is at most one $q \in Q$ such that $\mu_k(\sigma)_{w,q} \neq 0$,
- *total bu-deterministic* if for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, and $w \in Q^k$, there is exactly one state q such that $\mu_k(\sigma)_{w,q} \neq 0$,
- *top-down deterministic* (for short: td-deterministic) if for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, and $q \in Q$ there is at most one $w \in Q^k$ such that $\mu_k(\sigma)_{w,q} \neq 0$; moreover, $\nu_q \neq 0$ for at most one state q.

Note that, if the wta $\mathcal{A}$ is bu-deterministic, then for every input tree $\xi \in T_\Sigma$, there is at most one $q \in Q$ such that $h_\mu(\xi)_q \neq 0$. In this case, the operation $+$ of S is not used for the computation of $r_\mathcal{A}$. Also in the td-deterministic case $+$ is not used to compute $r_\mathcal{A}$.

We also note that, for every bu-deterministic wta $\mathcal{A}$, there exists a total bu-deterministic wta $\mathcal{A}'$ such that $r_\mathcal{A} = r_{\mathcal{A}'}$. This normal form can always be achieved in a standard way by using an additional dummy state for which the root weight vector ν yields 0.

Let $g \in \{\mathrm{bu}, \mathrm{td}\}$. Then a tree series $r \in S\langle\langle T_\Sigma \rangle\rangle$ is *g-deterministically recognizable* if there is a g-deterministic wta $\mathcal{A}$ such that $r = r_\mathcal{A}$. The corresponding classes of recognizable tree series are denoted by bud-Rec(Σ, S) and tdd-Rec(Σ, S). In Sect. 3.5, we will deal with the question under which conditions a wta can be determinized.

As second type of restriction on wta, we consider their root weights. A wta $\mathcal{A}$ has *Boolean root weights* if $\{\nu_q \mid q \in Q\} \subseteq \{0, 1\}$; in this case, we replace ν by the set $F = \{q \in Q \mid \nu_q = 1\}$. Then $(r_\mathcal{A}, \xi) = \sum_{q \in F} h_\mu(\xi)_q$. In fact, the wta of Examples 3.3 and 3.4 have Boolean root weights.

A tree series $r \in S\langle\langle T_\Sigma \rangle\rangle$ is *recognizable with Boolean root weights* if there is a wta $\mathcal{A}$ with Boolean root weights such that $r = r_\mathcal{A}$. The corresponding classes of recognizable tree series are denoted by indexing the original class with a capital B, e.g., bud-Rec$_\mathrm{B}(\Sigma, S)$ is the class of all tree series which are bu-deterministically recognizable with Boolean root weights.

Example 3.5. The remaining tree series shown in the list on page 322 are also recognizable as follows: size$_\delta$ and size are in $\mathrm{Rec}_\mathrm{B}(\Sigma, \mathsf{Nat}) \cap \text{bud-Rec}_\mathrm{B}(\Sigma, \mathsf{Trop})$; shortest$_\alpha \in \mathrm{Rec}_\mathrm{B}(\Sigma, \mathsf{Trop})$; yield $\in \text{bud-Rec}_\mathrm{B}(\Sigma, \mathsf{Lang}_\Sigma)$; revpos $\in$ bud-Rec$_\mathrm{B}$ $(\Sigma, \mathsf{Lang}_\mathbb{N})$; and revpos$_{\sigma(z,\alpha)} \in \mathrm{Rec}_\mathrm{B}(\Sigma, \mathsf{Lang}_\mathbb{N})$, cf. [13].

In general, wta and wta with Boolean root weights are equally powerful.

Theorem 3.6 ([14], Theorems 6.1.6 and 6.2.2). $\mathrm{Rec}(\Sigma, S) = \mathrm{Rec}_\mathrm{B}(\Sigma, S)$ *and* tdd-Rec$(\Sigma, S) =$ tdd-Rec$_\mathrm{B}(\Sigma, S)$.

Proof. For a given wta $\mathcal{A} = (Q, \Sigma, S, \mu, \nu)$, we construct the wta $\mathcal{A}' = (Q', \Sigma, S, \mu', \{q_f\})$ with Boolean root weights by defining $Q' = Q \cup \{q_f\}$ for a new state q_f. Moreover, for every $\sigma \in \Sigma^{(k)}$, $w \in Q^k$, and $q \in Q'$, we define $\mu'_k(\sigma)_{w,q} = \mu_k(\sigma)_{w,q}$ if $q \in Q$, and $\mu'_k(\sigma)_{w,q_f} = \sum_{q \in Q} \mu_k(\sigma)_{w,q} \cdot \nu_q$; and for every $w \in (Q')^k \setminus Q^k$ and $q \in Q'$, we define $\mu'_k(\sigma)_{w,q} = 0$. We note that this construction preserves td-determinism but not bu-determinism. It is obvious that $h_\mu(\xi) \cdot \nu = h_{\mu'}(\xi)_{q_f}$ and thus $r_\mathcal{A} = r_{\mathcal{A}'}$. □

In fact, the classes bud-Rec(Σ, S) and bud-Rec$_\mathrm{B}(\Sigma, S)$ may indeed differ. More precisely, if Σ contains at least one non-nullary symbol σ, and there is an element $a \in S \setminus \{0\}$ which has no multiplicative right inverse, then bud-Rec$(\Sigma, S) \setminus$ bud-Rec$_\mathrm{B}(\Sigma, S) \neq \emptyset$. A witness of this set is the polynomial tree series $r = a.\alpha + 1.\sigma(\alpha, \ldots, \alpha)$ (cf. [14], Lemma 6.1.3). However, for semifields bu-deterministic wta and bu-deterministic wta with Boolean root weights are equally powerful.

Theorem 3.7 ([14], Lemma 6.1.4). *Let S be a semifield. Then* bud-Rec$(\Sigma, S) =$ bud-Rec$_\mathrm{B}(\Sigma, S)$.

Proof. Let $\mathcal{A} = (Q, \Sigma, S, \mu, \nu)$ be a bu-deterministic wta. We construct the bu-deterministic wta $\mathcal{A}' = (Q, \Sigma, S, \mu', F)$ with $F = \{q \in Q \mid \nu_q \neq 0\}$ and $\mu'_k(\sigma)_{w,q} = \nu'(q_k)^{-1} \cdot \cdots \cdot \nu'(q_1)^{-1} \cdot \mu_k(\sigma)_{w,q} \cdot \nu'(q)$ for every $\sigma \in \Sigma^{(k)}$, $w = q_1 \ldots q_k \in Q^k$, and $q \in Q$; the auxiliary function $\nu' : Q \to S$ is defined by $\nu'(q) = \nu_q$ if $q \in F$ and $\nu'(q) = 1$ otherwise. Then $h_{\mu'}(\xi)_q = h_\mu(\xi)_q \cdot \nu'(q)$ for every $\xi \in T_\Sigma$ and $q \in Q$. Finally, we have $(r_\mathcal{A}, \xi) = h_\mu(\xi) \cdot \nu = \sum_{q \in F} h_\mu(\xi)_q \cdot \nu_q = \sum_{q \in F} h_\mu(\xi)_q \cdot \nu'(q) = \sum_{q \in F} h_{\mu'}(\xi)_q = (r_{\mathcal{A}'}, \xi)$. □

3.3 Closure Properties

As for tree languages, one can define operations on tree series in $S\langle\langle T_\Sigma\rangle\rangle$, e.g., the multiplication of a tree series with a semiring element, sum, Hadamard-product, top-concatenation, OI-substitution, α-concatenation (i.e., Cauchy-product), where α is a nullary symbol in Σ, α-Kleene star, and relabeling. Let us define these operations and show the corresponding closure properties of Rec(Σ, S).

Let $a \in S$ and $r \in S\langle\langle T_\Sigma\rangle\rangle$. Then the *scalar multiplication of a and r* is the tree series $ar \in S\langle\langle T_\Sigma\rangle\rangle$ defined by $(ar, \xi) = a \cdot (r, \xi)$ for every $\xi \in T_\Sigma$.

Let $r_1, r_2 \in S\langle\langle T_\Sigma\rangle\rangle$. The *sum of r_1 and r_2* and the *Hadamard product of r_1 and r_2* are the tree series $r_1 + r_2 \in S\langle\langle T_\Sigma\rangle\rangle$ and $r_1 \odot r_2 \in S\langle\langle T_\Sigma\rangle\rangle$, respectively, defined by $(r_1 + r_2, \xi) = (r_1, \xi) + (r_2, \xi)$ and $(r_1 \odot r_2, \xi) = (r_1, \xi) \cdot (r_2, \xi)$ for every $\xi \in T_\Sigma$. We can also sum up over an infinite family of tree series assuming that this family is locally finite. A family $(r_i \mid i \in I)$ of tree series is *locally finite* if for every $\xi \in T_\Sigma$, the set $I_\xi = \{i \in I \mid (r_i, \xi) \neq 0\}$ is finite. Then we define the sum $\sum_{i \in I} r_i \in S\langle\langle T_\Sigma\rangle\rangle$ by $(\sum_{i \in I} r_i, \xi) = \sum_{i \in I_\xi} (r_i, \xi)$ for every $\xi \in T_\Sigma$.

For every $\sigma \in \Sigma^{(k)}$, the *top-concatenation (with σ)* $\mathrm{top}_\sigma : S\langle\langle T_\Sigma\rangle\rangle^k \to S\langle\langle T_\Sigma\rangle\rangle$ is defined, for every $r_1, \ldots, r_k \in S\langle\langle T_\Sigma\rangle\rangle$ and $\xi \in T_\Sigma$ as follows: if $\xi = \sigma(\xi_1, \ldots, \xi_k)$, then $(\mathrm{top}_\sigma(r_1, \ldots, r_k), \xi) = (r_1, \xi_1) \cdot \cdots \cdot (r_k, \xi_k)$, otherwise $(\mathrm{top}_\sigma(r_1, \ldots, r_k), \xi) = 0$.

Next, we define the OI-substitution of tree series, which generalizes the OI-substitution of tree languages [55, 56]. Let $n \geq 0$, $\bar{\alpha} = (\alpha_1, \ldots, \alpha_n) \in (\Sigma^{(0)})^n$, and $\bar{r} = (r_1, \ldots, r_n) \in S\langle\langle T_\Sigma \rangle\rangle^n$. For every $\xi \in T_\Sigma$, the tree series $\xi \leftarrow_{\mathrm{OI},\bar{\alpha}} \bar{r}$ (abbreviated by s in this definition) is defined inductively on the structure of ξ: (i) if $\xi = \alpha_i$, then $s = r_i$, and (ii) if $\xi = \sigma(\xi_1, \ldots, \xi_k)$ and $k \geq 1$ or $k = 0$ and $\sigma \notin \{\alpha_1, \ldots, \alpha_n\}$, then $s = \mathrm{top}_\sigma(\xi_1 \leftarrow_{\mathrm{OI},\bar{\alpha}} \bar{r}, \ldots, \xi_k \leftarrow_{\mathrm{OI},\bar{\alpha}} \bar{r})$. We note that $(s, \xi') = 0$ unless ξ' can be obtained from ξ by substituting the α_i's with suitable trees. Then the *OI-substitution of* $\bar{r}$ *into* $r \in S\langle\langle T_\Sigma \rangle\rangle$ *at* $\bar{\alpha}$ is the tree series $r \leftarrow_{\mathrm{OI},\bar{\alpha}} \bar{r}$ in $S\langle\langle T_\Sigma \rangle\rangle$ which is defined to be $\sum_{\xi \in T_\Sigma} (r, \xi)(\xi \leftarrow_{\mathrm{OI},\bar{\alpha}} \bar{r})$. Note that the family $((r, \xi)(\xi \leftarrow_{\mathrm{OI},\bar{\alpha}} \bar{r}) \mid \xi \in T_\Sigma)$ of tree series is locally finite, and thus the summation is well defined.

Let $r_1, r_2 \in S\langle\langle T_\Sigma \rangle\rangle$ and $\alpha \in \Sigma^{(0)}$. The α*-concatenation of* r_1 *and* r_2 is the tree series $r_1 \leftarrow_{\mathrm{OI},(\alpha)} (r_2)$, abbreviated by $r_1 \circ_\alpha r_2$.

Let $r \in S\langle\langle T_\Sigma \rangle\rangle$ and $\alpha \in \Sigma^{(0)}$. The *nth* α*-iteration of* r is the tree series $r_\alpha^n \in S\langle\langle T_\Sigma \rangle\rangle$ defined inductively as follows: $r_\alpha^0 = \widetilde{0}$ and for every $n \geq 0$, $r_\alpha^{n+1} = r \circ_\alpha r_\alpha^n + 1.\alpha$. A tree series $r \in S\langle\langle T_\Sigma \rangle\rangle$ is α*-proper* if $(r, \alpha) = 0$. For every α-proper $r \in S\langle\langle T_\Sigma \rangle\rangle$, $\xi \in T_\Sigma$, and $n \geq \mathrm{height}(\xi) + 1$, we have that $(r_\alpha^{n+1}, \xi) = (r_\alpha^n, \xi)$ (cf. [41], Lemma 3.10). Then, for every $r \in S\langle\langle T_\Sigma \rangle\rangle$, the α*-Kleene star of* r is the tree series $r_\alpha^* \in S\langle\langle T_\Sigma \rangle\rangle$ defined as follows. If r is α-proper, then $(r_\alpha^*, \xi) = (r_\alpha^{\mathrm{height}(\xi)+1}, \xi)$ for every $\xi \in T_\Sigma$; otherwise, $r_\alpha^* = \widetilde{0}$.

Next, let $\tau : \Sigma \to \mathcal{P}(\Delta)$ be a *relabeling (from* Σ *to* Δ*)*, i.e., a mapping such that $\tau(\sigma) \subseteq \Delta^{(k)}$ for every $k \geq 0$ and $\sigma \in \Sigma^{(k)}$. This mapping is extended in a canonical way to a mapping $\tau : T_\Sigma \to \mathcal{P}(T_\Delta)$, and then to a mapping $\tau : S\langle\langle T_\Sigma \rangle\rangle \to S\langle\langle T_\Delta \rangle\rangle$ by defining $(\tau(r), \xi) = \sum_{\zeta \in T_\Sigma, \xi \in \tau(\zeta)} (r, \zeta)$ for every $r \in S\langle\langle T_\Sigma \rangle\rangle$ and $\xi \in T_\Delta$ (note that the summation is finite).

In fact, recognizability of tree series is preserved under the aforementioned operations. The proofs of these closure results are sometimes folklore and sometimes straightforward generalizations of the corresponding results for formal power series over strings or for recognizable tree languages (cf., e.g., [45, 125, 7, 96, 69, 21, 91]). In particular, in view of Theorem 3.6, we can use results of [41] in which wta is defined with Boolean root weights.

Theorem 3.8. *Let* S *be commutative. Then* $\mathrm{Rec}(\Sigma, S)$ *is closed under:*

- *scalar multiplication ([41], Lemma 6.3),*
- *sum ([41], Lemma 6.4),*
- *Hadamard-product ([12], Corollary 3.9, also cf. [7], Proposition 5.1 for a field* S*),*
- *top-concatenation ([41], Lemmas 6.1 and 6.2),*
- α*-concatenation ([41], Lemma 6.5),*
- α*-Kleene star ([41], an easy adaptation of Lemma 6.7), and*
- *OI-substitution.*

Moreover, if $r \in \mathrm{Rec}(\Sigma, S)$ *and* τ *is a relabeling from* Σ *to* Δ*, then* $\tau(r) \in \mathrm{Rec}(\Delta, S)$ *([43], Lemma 3.4, also cf. [87], Corollary 14 for continuous semirings).*

Proof. We prove the closure under OI-substitution. It has been proved for so-called well ω-additive semirings in [21], Lemma 24. Now we show that it also holds for commutative semirings. For this, let $n \geq 0$, $\bar{\alpha} = (\alpha_1, \ldots, \alpha_n) \in (\Sigma^{(0)})^n$, $\bar{r} = (r_1, \ldots, r_n) \in S\langle\langle T_\Sigma \rangle\rangle^n$, and $r \in S\langle\langle T_\Sigma \rangle\rangle$. Then

$$r \leftarrow_{\mathrm{OI},\bar{\alpha}} \bar{r} = \tau_1'\big(\big(r \circ_{\alpha_1} \tau_1(r_1)\big) \leftarrow_{\mathrm{OI},(\alpha_2,\ldots,\alpha_n)} (r_2, \ldots, r_n)\big)$$

where $\tau_1 : \Sigma \to \mathcal{P}(\Sigma_1)$ is the relabeling with $\Sigma_1 = \{\sigma_1 \mid \sigma \in \Sigma\}$ and $\tau_1(\sigma) = \{\sigma_1\}$; moreover, $\tau_1' : \Sigma \cup \Sigma_1 \to \mathcal{P}(\Sigma)$ is the relabeling with $\tau_1'(\sigma) = \tau_1'(\sigma_1) = \{\sigma\}$. This process can be repeated $n-1$ times where in the ith step we use τ_i, τ_i', and Σ_i instead of τ_1, τ_1', and Σ_1, respectively. Since recognizability is preserved under α-concatenation and also under relabeling, we obtain that $\mathrm{Rec}(\Sigma, S)$ is closed under OI-substitution for commutative semirings. □

Recognizability is also preserved under semiring homomorphisms. Formally, let S' be another semiring and $h : S \to S'$ a mapping; h extends to the mapping $h : S\langle\langle T_\Sigma \rangle\rangle \to S'\langle\langle T_\Sigma \rangle\rangle$ by defining $h(r) = h \circ r$ for every $r \in S\langle\langle T_\Sigma \rangle\rangle$. If h is a semiring homomorphism, then the preservation of recognizability is obtained by replacing every transition weight $\mu_k(\sigma)_{w,q}$ and every root weight ν_q of the wta over S by $h(\mu_k(\sigma)_{w,q})$ and $h(\nu_q)$, respectively.

Theorem 3.9 ([16], Lemma 3). *Recognizability is preserved under semiring homomorphisms, i.e., for every semiring homomorphism $h : S \to S'$ and $r \in S\langle\langle T_\Sigma \rangle\rangle$, if $r \in \mathrm{Rec}(\Sigma, S)$, then $h(r) \in \mathrm{Rec}(\Sigma, S')$.*

3.4 Support of Recognizable Tree Series

It is straightforward to embed the class $\mathrm{Rec}(\Sigma)$ of recognizable tree languages into the class $\text{bud-Rec}(\Sigma, S)$ by using the support function. First, however, we prove a useful technical lemma.

Lemma 3.10. *Let (Q, θ) be a finite Σ-algebra and $f : Q \to S$ a mapping. Then $f \circ h_Q \in \text{bud-Rec}(\Sigma, S)$.*

Proof. We construct the bu-deterministic wta $\mathcal{A} = (Q, \Sigma, S, \mu, f)$ by defining $\mu_k(\sigma) = 1_{(S,\theta(\sigma))}$; note that f is an element of S^Q. Clearly, $h_\mu(\xi)_q = 1$ if $h_Q(\xi) = q$, and 0, otherwise. Then $(r_{\mathcal{A}}, \xi) = h_\mu(\xi) \cdot f = h_\mu(\xi)_{h_Q(\xi)} \cdot f(h_Q(\xi))$ $= f(h_Q(\xi))$ for every $\xi \in T_\Sigma$. Hence, $f \circ h_Q \in \text{bud-Rec}(\Sigma, S)$. □

Lemma 3.11. *If $L \in \mathrm{Rec}(\Sigma)$, then $1_{(S,L)} \in \text{bud-Rec}(\Sigma, S)$. In particular, $\mathrm{Rec}(\Sigma) \subseteq \mathrm{supp}(\text{bud-Rec}(\Sigma, S))$.*

Proof. Let $L \in \mathrm{Rec}(\Sigma)$. Hence, there is a deterministic bottom-up tree automaton $\mathcal{A} = (Q, \Sigma, \delta, F)$ with $L = L_{\mathcal{A}}$. Then $1_{(S,L)} = 1_{(S,F)} \circ h_Q$, and by Lemma 3.10 it follows that $1_{(S,L)} \in \text{bud-Rec}(\Sigma, S)$. □

In the Boolean case, wta computes exactly the class of recognizable tree languages, i.e., $\mathrm{supp}(\mathrm{Rec}(\Sigma, \mathbb{B})) = \mathrm{Rec}(\Sigma)$. However, this equality also holds for a larger class of semirings (also cf. [125], Corollary II.5.3 for formal power series).

Theorem 3.12. *Let S be positive. Then* $\mathrm{supp}(\mathrm{Rec}(\Sigma, S)) = \mathrm{Rec}(\Sigma)$.

Proof. Consider the particular mapping $h : S \to \mathbb{B}$ defined by $h(a) = 1$ for $a \neq 0$ and $h(0) = 0$. Since S is positive, h is a semiring homomorphism from S to $\mathbb{B}$. Moreover, $\mathrm{supp}(h(r)) = \mathrm{supp}(r)$ for every $r \in S\langle\langle T_\Sigma \rangle\rangle$. By Theorem 3.9, we have $h(r) \in \mathrm{Rec}(\Sigma, \mathbb{B})$ for every $r \in \mathrm{Rec}(\Sigma, S)$, and thus $\mathrm{supp}(r) = \mathrm{supp}(h(r)) \in \mathrm{supp}(\mathrm{Rec}(\Sigma, \mathbb{B})) = \mathrm{Rec}(\Sigma)$. The other inclusion holds by Lemma 3.11. □

On the other hand, there is a ranked alphabet Σ such that $\mathrm{supp}(\mathrm{Rec}_{\mathrm{B}}(\Sigma, \mathbb{Z})) \setminus \mathrm{Rec}(\Sigma) \neq \emptyset$, where $\mathbb{Z}$ is the semiring of integers. This follows directly from the fact that there is a weighted finite automaton $\mathcal{A}$ such that $\mathrm{supp}(\|\mathcal{A}\|)$ is not recognizable (cf. [124], Example 6.2 and [8], Example III.3.1) and that weighted finite automata can be simulated by our wta in the way which we described on page 324.

We note that Theorem 3.12 holds for formal power series over commutative and (so-called) quasi-positive semirings, cf. [138], Corollary 5.2. Obviously, Theorem 3.12 shows that also the inverse of the implication of Lemma 3.11 holds for positive semirings. This is even true for the larger class of proper semirings.

Theorem 3.13. *Let S be a commutative and proper semiring. Then $L \in$* $\mathrm{Rec}(\Sigma)$ *iff* $1_{(S,L)} \in \mathrm{Rec}(\Sigma, S)$ *for every Σ-tree language L.*

Proof. Let $1_{(S,L)} \in \mathrm{Rec}(\Sigma, S)$. By [137], Theorem 2.1, the fact that S is not a ring implies that there is a semiring homomorphism $h : S \to \mathbb{B}$. By Theorem 3.9, we have $h(1_{(S,L)}) = 1_{(\mathbb{B},L)} \in \mathrm{Rec}(\Sigma, \mathbb{B})$. Thus $L = \mathrm{supp}(1_{(\mathbb{B},L)}) \in \mathrm{supp}(\mathrm{Rec}(\Sigma, \mathbb{B})) = \mathrm{Rec}(\Sigma)$. The inverse is proved in Lemma 3.11. □

In the following we show that also the inverse of Lemma 3.10 holds provided that the semiring is locally finite. Then a tree series $r \in \mathrm{Rec}(\Sigma, S)$ can be computed by a finite Σ-algebra. The construction in the next lemma is the one of [12], Sect. 4 (also cf. [5], Theorem 2.1; [80], Sect. 3.1; [16], Theorem 9; and [43], Lemma 6.1).

Lemma 3.14. *Let S be locally finite and $r \in \mathrm{Rec}(\Sigma, S)$. Then there is a finite Σ-algebra Q and a mapping $f : Q \to S$ such that $r = f \circ h_Q$.*

Proof. Let $\mathcal{A} = (P, \Sigma, S, \mu, \nu)$ be a wta such that $r_\mathcal{A} = r$. Let S' be the smallest subsemiring containing $\mathrm{wts}(\mathcal{A})$. Since S is locally finite, S' is finite. Now we consider the Σ-algebra $(Q, \mu'_\mathcal{A})$ where $Q = (S')^P$ and $\mu'_\mathcal{A}(\sigma)$ is the restriction of $\mu_\mathcal{A}(\sigma)$ to Q^k for every $k \geq 0$. Moreover, we define the mapping f by $f(v) = v \cdot \nu$. Clearly, $h_\mu(\xi) \in Q$ and $h_\mu(\xi) = h_Q(\xi)$ for every $\xi \in T_\Sigma$, and thus we have $(r, \xi) = h_\mu(\xi) \cdot \nu = h_Q(\xi) \cdot \nu = f(h_Q(\xi))$. □

Using Lemma 3.14, we can prove the following inverse image theorem.

Theorem 3.15 ([19], Theorem 4). *Let S be locally finite, $E \subseteq S$, and $r \in \mathrm{Rec}(\Sigma, S)$. Then $r^{-1}(E) \in \mathrm{Rec}(\Sigma)$.*

Proof. By Lemma 3.14, there is a finite Σ-algebra (Q, θ) and a mapping $f : Q \to S$ such that $r = f \circ h_Q$. Now we construct the deterministic bottom-up tree automaton $\mathcal{A} = (Q, \Sigma, \delta, F)$, where $\delta_\sigma = \theta(\sigma)$ and $F = \{q \in Q \mid f(q) \in E\}$. Then $(r, \xi) \in E$ iff $f(h_Q(\xi)) \in E$ iff $h_Q(\xi) \in F$ iff $\xi \in L_{\mathcal{A}}$. □

From the inverse image theorem, we can derive the following results for a recognizable tree series r over a locally finite S. We can show that the support of r is recognizable. Also, assuming that $(S, \leq)$ is a partially ordered set, all cut sets of r are recognizable; for every $a \in S$, the *a-cut* of r is the set $r_a = \{\xi \in T_\Sigma \mid (r, \xi) \geq a\}$ (cf. [129]). Moreover, we can prove that r is a *recognizable step function*, i.e., there are $n \geq 0$, recognizable tree languages $L_1, \ldots, L_n \subseteq T_\Sigma$, and $a_1, \ldots, a_n \in S$ such that $r = \sum_{i=1}^{n} a_i 1_{(S, L_i)}$. (Clearly, a recognizable step function is a recognizable tree series provided S is commutative, by Lemma 3.11 and Theorem 3.8.)

Corollary 3.16. *Let S be locally finite and $r \in \mathrm{Rec}(\Sigma, S)$. Then*

(A) $\mathrm{supp}(r) \in \mathrm{Rec}(\Sigma)$.
(B) If $(S, \leq)$ is a partially ordered set, then $r_a \in \mathrm{Rec}(\Sigma)$ for every $a \in S$ ([16], Theorem 9).
(C) r is a recognizable step function ([43], Lemma 6.1).

Proof. Since $\mathrm{supp}(r) = r^{-1}(S \setminus \{0\})$ and $r_a = r^{-1}(\{b \in S \mid b \geq a\})$, (A) and (B) follow from Theorem 3.15. To prove (C), we observe that $r = \sum_{a \in R} a 1_{(S, r^{-1}(a))}$ where $R = \{(r, \xi) \mid \xi \in T_\Sigma\}$ is the range of r. Since R is finite by Lemma 3.14 (because $R \subseteq f(Q)$) and $r^{-1}(a) \in \mathrm{Rec}(\Sigma)$ by Theorem 3.15, r is a recognizable step function. □

3.5 Determinization of Weighted Tree Automata

It is well known that the usual power set construction for finite-state string automata can be extended in a straightforward way to bottom-up tree automata [68, 69]. This means that bottom-up tree automata and deterministic bottom-up tree automata accept the same class of tree languages. On the other hand, there are recognizable tree languages which are not recognizable by deterministic top-down tree automata.

If the power set construction is extended to a wta (by identifying $\mathcal{P}(Q)$ with $\mathbb{B}^Q$ and then turning $\mathbb{B}$ into S), this might lead to a deterministic wta with infinitely many states because its state set is the set of all reachable Q-vectors over S. However, infinity can be avoided if the semiring S is locally finite. The following theorem generalizes [14], Theorem 6.3.3 and [12], Theorem 4.8.

Theorem 3.17. *Let S be locally finite, $r \in \mathrm{Rec}(\Sigma, S)$, and $g : S \to S$. Then $g(r) \in \text{bud-Rec}(\Sigma, S)$. In particular, $\mathrm{Rec}(\Sigma, S) = \text{bud-Rec}(\Sigma, S)$.*

Proof. By Lemma 3.14, we have that $r = f \circ h_Q$ for some finite Σ-algebra Q and mapping $f : Q \to S$. Then $g(r) = g \circ (f \circ h_Q) = (g \circ f) \circ h_Q$, and thus by Lemma 3.10, $g(r) \in \text{bud-Rec}(\Sigma, S)$. □

In fact, for $S = \mathbb{B}$, the construction of Lemma 3.14 is exactly the usual power set construction for bottom-up tree automata. Also we note that, in general, the condition that S is locally finite cannot be dropped from Theorem 3.17. For this, we consider the tree series r over the field $(\mathbb{Q}, +, \cdot, 0, 1)$ of rational numbers such that $(r, \gamma^n(\alpha)) = 1 + 2^n$, where $\Sigma = \{\gamma^{(1)}, \alpha^{(0)}\}$. Then $r \in \text{Rec}(\Sigma, \mathbb{Q})$ by constructing a wta with two states q_1 and q_2 such that $h_\mu(\xi)_{q_1} = 1$ and $h_\mu(\xi)_{q_2} = 2^n$. Now assume that there is a bu-deterministic wta $\mathcal{A}$ such that $r = r_{\mathcal{A}}$. Then we can derive a contradiction by using the observation that, for every $\xi \in T_\Sigma$, the coefficient $(r_{\mathcal{A}}, \xi)$ is an element of the carrier set of the smallest sub-monoid of $(\mathbb{Q}, \cdot, 1)$ containing $\text{wts}(\mathcal{A})$; thus, $r \notin \text{bud-Rec}(\Sigma, \mathbb{Q})$ (cf. [17], Lemma 6.3).

Finally we mention that, similarly to the unweighted case, the easy tree series $r = 1.\sigma(\alpha, \ldots, \alpha, \beta) + 1.\sigma(\beta, \alpha, \ldots, \alpha)$ separates the classes $\text{Rec}(\Sigma, S)$ and $\text{tdd-Rec}(\Sigma, S)$.

Theorem 3.18 ([14], Theorem 6.3.5). *Let S be commutative or zero-divisor free. Moreover, let Σ contain at least two nullary symbols and at least one symbol with rank ≥ 2. Then* $\text{Rec}(\Sigma, S) \setminus \text{tdd-Rec}(\Sigma, S) \neq \emptyset$.

3.6 Pumping Lemmata and Decidability

For the class of recognizable tree languages, there is a well-known pumping lemma (cf. [69], Proposition 5.2). Here, we will present pumping lemmata and decidability results for recognizable tree series.

As a technical concept, we need contexts. The set C_Σ of *contexts* is the set of Σ-trees over $\{z\}$ in which z occurs exactly once. In fact, C_Σ is the free monoid freely generated by the set C'_Σ with operation $\cdot_z$ and $z \in C_\Sigma$ as neutral element, cf. [7], Proposition 9.1; $C'_\Sigma \subseteq C_\Sigma$ is the set of those contexts in which the z occurs at a child of the root. For every $\zeta \in C_\Sigma$, we define $\zeta^0 = z$ and $\zeta^{n+1} = \zeta^n \cdot_z \zeta$.

As additional technical preparation, we extend the homomorphism induced by a wta to contexts. Formally, let $\mathcal{A} = (Q, \Sigma, S, \mu, \nu)$ be a wta and $v \in S^Q$. Now we consider the v-extension $(S^Q, \mu^v_{\mathcal{A}})$ of the Σ-algebra $(S^Q, \mu_{\mathcal{A}})$ (as defined in Sect. 2.3) and denote the unique $\Sigma \cup \{z\}$-algebra homomorphism from $T_{\Sigma \cup \{z\}}$ to S^Q by h^v_μ. Then $h_\mu(\zeta \cdot_z \xi) = h_\mu^{h_\mu(\xi)}(\zeta)$ for every $\zeta \in C_\Sigma$ and $\xi \in T_\Sigma$. If v has the particular form that there is a $q \in Q$ with $v_q = 1$ and $v_p = 0$ for every $p \neq q$, then we abbreviate h^v_μ by h^q_μ.

The pumping lemma that we present first deals with recognizable tree series over fields.

Theorem 3.19 ([7], Theorem 9.2). *Let S be a field and $r \in \text{Rec}(\Sigma, S)$. There is an $m \geq 1$ such that, for every $\xi \in \text{supp}(r)$ with* $\text{height}(\xi) \geq m$*,* *there are* $\zeta, \zeta' \in C_\Sigma$, *and* $\xi' \in T_\Sigma$ *such that* $\xi = \zeta' \cdot_z \zeta \cdot_z \xi'$ *and we have that* $\{\zeta' \cdot_z \zeta^n \cdot_z \xi' \mid n \geq 0\} \cap \mathrm{supp}(r)$ *is an infinite set.*

The second pumping lemma concerns bu-deterministically recognizable tree series over an arbitrary semiring.

Theorem 3.20 ([12], Theorem 5.6). *Let* $r \in \text{bud-Rec}(\Sigma, S)$. *There is an* $m \geq 1$ *such that, for every* $\xi \in \mathrm{supp}(r)$ *and position* $w = i_1 \ldots i_l \in \mathrm{pos}(\xi)$ *with* $i_1, \ldots, i_l \in \mathbb{N}$ *and* $l \geq m$, *there are indices* j, k *with* $0 \leq j < k \leq l$ *and* $a, a', b, b', c \in S$ *such that:*

- $l - j \leq m$ *and*
- $(r, \zeta' \cdot_z \zeta^n \cdot_z \xi') = a' \cdot a^n \cdot c \cdot b^n \cdot b'$ *for every* $n \geq 0$ *where*
  - $\zeta' = \xi[z]_u$ *with* $u = i_1 \ldots i_j$,
  - $\zeta = (\xi|_u)[z]_v$ *with* $v = i_{j+1} \ldots i_k$, *and*
  - $\xi' = \xi|_{uv}$.

If S *is zero-divisor free, then* $\zeta' \cdot_z \zeta^n \cdot_z \xi' \in \mathrm{supp}(r)$ *for every* $n \geq 0$.

We give a sketch of the proof. Assume that $\mathcal{A} = (Q, \Sigma, S, \mu, \nu)$ is a total bu-deterministic wta which recognizes r and let $m = |Q|$. Now let $\xi \in T_\Sigma$ be an input tree. Then there is a unique run κ on ξ for which all transitions have nonzero weight, i.e., for every $w \in \mathrm{pos}(\xi)$, the condition $\mu_k(\sigma)_{q_1 \ldots q_k, q} \neq 0$ holds with $\sigma^{(k)} = \xi(w)$, $q = \kappa(w)$, and $q_i = \kappa(wi)$ for $1 \leq i \leq k$. Let us denote the state $\kappa(\varepsilon)$ by $\widetilde{\mu}(\xi)$. Now assume there is a $w \in \mathrm{pos}(\xi)$ with $|w| \geq m$, which implies $\mathrm{height}(\xi) \geq m$. Then the standard pumping can be done because there is a repetition of states along w, i.e., there are contexts ζ and ζ' and a tree ξ' such that $\xi = \zeta' \cdot_z \zeta \cdot_z \xi'$ and $\widetilde{\mu}(\zeta \cdot_z \xi') = \widetilde{\mu}(\xi')$. Let u (and v) be the position of z in ζ' (and ζ, resp.). Then the element a of S is the product of the weights of all the transitions which are performed (in κ) at positions v' of ζ such that v' is lexicographically smaller than v and v' is not a prefix of v; b is the product of the weights of all the transitions (in κ) at the other positions of ζ except v; in both cases the order of the factors is determined by the left-to-right traversal over ζ. The elements a' and b' are defined similarly for u and ζ' instead of v and ζ, except that b' contains the root weight $\nu(\widetilde{\mu}(\xi))$ as an additional factor. Finally, c is the product of the weights of all the transitions which are performed at positions of ξ'. Since S may contain zero-divisors, even the unique run κ on ξ can have weight 0. However, if S is zero-divisor free and $\xi \in \mathrm{supp}(r)$, then $\mathrm{wt}(\kappa) \neq 0$, and thus also $\zeta' \cdot_z \zeta^n \cdot_z \xi' \in \mathrm{supp}(r)$.

Using Theorem 3.20, it can, e.g., be proved that the tree series height is not in $\text{bud-Rec}(\Sigma, \mathsf{Arct})$ (cf. [12], Example 5.9). Also, this pumping lemma can be used to prove decidability results which we discuss here for the question whether a tree series is constant. We assume that the semiring S is effectively given and also the considered tree series r is effectively given by a total bu-deterministic wta $\mathcal{A} = (Q, \Sigma, S, \mu, \nu)$. Let P be the set of all those $\widetilde{\mu}(\xi) \in Q$, where $\xi \in T_\Sigma$ with $\mathrm{height}(\xi) \leq |Q| - 1$ and there is $\zeta \in C_\Sigma$ such that $\mathrm{height}(\zeta) \leq 2 \cdot |Q| - 2$, and $\zeta \cdot_z \xi \in \mathrm{supp}(r)$. Intuitively, P is the set of states which are reachable by small trees ξ and there is a small context ζ such that $\zeta \cdot_z \xi$ is in the support of r. As the final preparation, we define the sets B_1 and B_2. The set B_1 is the set of weights of small contexts that can be pumped, i.e.,

$$B_1 = \big\{ h_\mu^{\widetilde{\mu}(\xi)}(\zeta)_{\widetilde{\mu}(\xi)} \;\big|\; \xi \in T_\Sigma,\ \zeta \in C_\Sigma,\ \text{height}(\zeta \cdot_z \xi) \leq |Q|, \\ \widetilde{\mu}(\xi) = \widetilde{\mu}(\zeta \cdot_z \xi) \in P \big\}.$$

The set B_2 is the set of weights of small trees, i.e.,

$$B_2 = \{(r, \xi) \mid \xi \in T_\Sigma \text{ and height}(\xi) \leq |Q| - 1\}.$$

The sets P, B_1, and B_2 are finite sets which can be constructed effectively. Then the following key lemma can be shown, making essential use of Theorem 3.20.

Lemma 3.21 ([12], Lemma 6.3). *Let S be commutative and $d \in S$. Moreover, let $r \in$ bud-Rec(Σ, S). Then $(r, \xi) = d$ for every $\xi \in \text{supp}(r)$ iff (i) $b \cdot d \in \{0, d\}$ for every $b \in B_1$ and (ii) $B_2 \subseteq \{0, d\}$.*

Theorem 3.22 ([12], Sect. 6). *Let S be commutative and $r \in$ bud-Rec(Σ, S). Then the following problems are decidable:*

(A) Constant-on-its-support problem, i.e., is there an $a \in S$ such that $(r, \xi) = a$ for every $\xi \in \text{supp}(r)$?
(B) Constant tree series problem, i.e., is there an $a \in S$ such that $(r, \xi) = a$ for every $\xi \in T_\Sigma$?
(C) Emptiness problem, i.e., is $r = \widetilde{0}$ (or equivalently, is $\text{supp}(r) = \emptyset$)?
(D) Boolean tree series problem, i.e., is $(r, \xi) \in \{0, 1\}$ for every $\xi \in T_\Sigma$?

Proof. First, we prove (A). By Lemma 3.21, we know that r is constant on its support iff there is a $d \in S$ such that conditions (i) and (ii) hold for this d. Now the decision procedure computes B_2. If $|B_2| > 2$ or $|B_2| = 2$ and $0 \notin B_2$, then r is not constant on its support. If $B_2 = \{0\}$, then (i) holds with $d = 0$, hence r is constant on its support, in fact, $r = \widetilde{0}$. If $B_2 = \{d\}$ or $B_2 = \{0, d\}$ for some $d \neq 0$, then check whether $b \cdot d \in \{0, d\}$ for every $b \in B_1$. If yes, then r is constant on its support with value d, if no, then it is not constant.

Proof of (B): This can be proved in a similar way to the first statement by first proving the modification of Lemma 3.21 in which $\xi \in \text{supp}(r)$ and $\{0, d\}$ are replaced by $\xi \in T_\Sigma$ and $\{d\}$, respectively. Proof of (C) and (D): These statements follow directly from Lemma 3.21 with $d = 0$ and $d = 1$, respectively. □

For the decision of the finiteness of $\text{supp}(r)$ we additionally require that S is zero-divisor free. Then, by Theorem 3.20, if $(r, \zeta' \cdot_z \zeta \cdot_z \xi') \neq 0$, then also $(r, \zeta' \cdot_z \zeta^n \cdot_z \xi') \neq 0$ for every $n \geq 0$. Hence, $\text{supp}(r)$ is finite iff $\text{height}(\xi) \leq$

$|Q|-1$ for every $\xi \in \mathrm{supp}(r)$. Thus, $\mathrm{supp}(r)$ is finite iff $\mathrm{supp}(r') = \emptyset$, where r' is the tree series defined by $(r',\xi) = (r,\xi)$ if $\mathrm{height}(\xi) \geq |Q|$, and $(r',\xi) = 0$ otherwise. It is not difficult to show that r' is in bud-Rec(Σ, S), effectively (see [12], Lemma 6.10). Thus, finiteness of r can be decided by Theorem 3.22(C): the emptiness problem.

Theorem 3.23 ([12], Theorem 6.11). *Let S be commutative and zero-divisor free, and $r \in$ bud-Rec(Σ, S) a recognizable tree series. Then the finiteness problem is decidable, i.e., it is decidable whether $\mathrm{supp}(r)$ is a finite set.*

In [103], the emptiness problem has been considered for arbitrary tree series in $r \in \mathrm{Rec}(\Sigma, S)$. This result, which is reported in the next theorem, is based on (i) a pumping lemma for deterministic wta over distributive Ω-algebras (cf. [103], Theorem 4), (ii) a decidable property which characterizes the emptiness of the tree series recognized by deterministic wta over zero-sum free distributive Ω-algebras (cf. [103], Proposition 4), and (iii) the simulation of a wta by a wta over a particular distributive Ω-algebra (cf. [103], Proposition 2).

Theorem 3.24 ([103], Corollary 3). *Let S be commutative and zero-sum free and $r \in \mathrm{Rec}(\Sigma, S)$. Then it is decidable whether $r = \widetilde{0}$.*

The emptiness problem is also decidable if S is a field. The proof, given in [127] on the base of [45], exploits some methods of linear algebra in an elegant way. This is possible because now the Σ-algebra $(S^Q, \mu_{\mathcal{A}})$ which is associated to a wta $\mathcal{A}$, is a finite-dimensional S-Σ-vector space; recall that this means that $(S^Q, +, 0)$ is an S-vector space and the mappings $\mu_{\mathcal{A}}(\sigma)$ are multilinear. As preparation, we recall a well-known statement from linear algebra.

Lemma 3.25. *Let V be a finite-dimensional S-vector space and let $V' \subseteq V$ be a subspace of V. Then $\dim(V') \leq \dim(V)$; moreover, if $\dim(V') = \dim(V)$, then $V' = V$.*

Theorem 3.26 ([127], Theorem 4.2; [18], Lemma 2). *Let S be a field and $r \in \mathrm{Rec}(\Sigma, S)$. Then it is decidable whether $r = \widetilde{0}$.*

Proof. Let $\mathcal{A} = (Q, \Sigma, S, \mu, \nu)$ be a wta with $r_{\mathcal{A}} = r$. Assume that $|Q| = n$. It suffices to prove that

$$r_{\mathcal{A}} = \widetilde{0} \quad \text{iff} \quad (r_{\mathcal{A}}, \xi) = 0 \text{ for every } \xi \in T_\Sigma \text{ with } \mathrm{height}(\xi) \leq n \qquad (1)$$

because the latter property is decidable.

Note that S^Q is an n-dimensional S-vector space. For every $m \geq 0$, we define the subspace $V_m = \langle \{h_\mu(\xi) \mid \xi \in T_\Sigma, \mathrm{height}(\xi) \leq m\} \rangle$ generated by the vectors $h_\mu(\xi)$ for trees ξ of height at most m. This forms the chain $V_0 \subseteq V_1 \subseteq \cdots \subseteq V_m \subseteq V_{m+1} \subseteq \cdots \subseteq S^Q$ of subspaces. By Lemma 3.25, we have $\dim(V_m) \leq \dim(V_{m+1}) \leq n$ for every $m \geq 0$. Moreover, it is easy to see that $V_{m+1} = \langle\{\mu_{\mathcal{A}}(\sigma)(v_1, \ldots, v_k) \mid k \geq 0, \sigma \in \Sigma^{(k)}, v_1, \ldots, v_k \in V_m\}\rangle$. This characterization of V_{m+1} in terms of V_m proves (by a straightforward induction on l) that if $V_m = V_{m+1}$ for some $m \geq 0$, then $V_m = V_{m+l}$ for every $l \geq 1$.

Moreover, there must be an m_0 such that $0 \leq m_0 \leq n$ and $\dim(V_{m_0}) = \dim(V_{m_0+1})$. By Lemma 3.25, we obtain that $V_{m_0} = V_{m_0+1,}$ and thus $V_{m_0} = V_{m_0+l}$ for every $l \geq 1$. Hence, $V_n = \bigcup_{m\geq 0} V_m = \langle\{h_\mu(\xi) \mid \xi \in T_\Sigma\}\rangle$.

Now we can verify equivalence (1) as follows: $r_{\mathcal{A}} = \widetilde{0}$ iff $(r_{\mathcal{A}}, \xi) = 0$ for every $\xi \in T_\Sigma$ iff $h_\mu(\xi) \cdot \nu = 0$ for every $\xi \in T_\Sigma$ iff $h_\mu(\xi) \cdot \nu = 0$ for every $\xi \in T_\Sigma$ with $\mathrm{height}(\xi) \leq n$ iff $(r_{\mathcal{A}}, \xi) = 0$ for every $\xi \in T_\Sigma$ with $\mathrm{height}(\xi) \leq n$, where we prove the "if" part of the last, but one equivalence in the following way. Let $\xi \in T_\Sigma$. Since $h_\mu(\xi) \in V_n$, it can be written as a linear combination $h_\mu(\xi) = \sum_{i=1}^{l} a_i h_\mu(\xi_i)$ for some $l \geq 1$, $a_1, \ldots, a_l \in S$, and trees $\xi_1, \ldots, \xi_l \in T_\Sigma$ of height at most n. Then by an easy calculation in S^Q, we obtain $h_\mu(\xi) \cdot \nu = \sum_{i=1}^{l} a_i \cdot (h_\mu(\xi_i) \cdot \nu)$. This proves equivalence (1), and thus the theorem. □

As a corollary, we obtain that the equivalence problem of recognizable tree series over a field is decidable.

Corollary 3.27 ([127], Theorem 4.2; [18], Lemma 2). *Let S be a field and $r_1, r_2 \in \mathrm{Rec}(\Sigma, S)$. Then it is decidable whether $r_1 = r_2$.*

Proof. Let $r_1, r_2 \in \mathrm{Rec}(\Sigma, S)$ be effectively given. Then by Theorem 3.8, which is effective, also $r = r_1 + (-1) \cdot r_2$ is in $\mathrm{Rec}(\Sigma, S)$. Clearly, $r_1 = r_2$ iff $r = \widetilde{0}$, which is decidable by Theorem 3.26. □

For the particular semirings $(\mathbb{R}_+, +, \cdot, 0, 1)$ (of non-negative reals) and Nat, the decidability of the equivalence of recognizable tree series has been proved in [20].

3.7 Finite Algebraic Characterizations of Recognizable Tree Series

The Myhill–Nerode theorem for recognizable string languages has been extended to recognizable tree languages [100, 130, 86, 69]. That is, a Σ-tree language is recognizable if and only if its syntactic Σ-algebra is finite. In this section, we discuss three similar characterizations of recognizable tree series where the characterizations are based on fields (cf. Theorem 3.31), semifields (cf. Theorem 3.35), and commutative and zero-divisor free semirings (cf. Theorem 3.36), respectively.

In this section, we will again use the notations C_Σ, h_μ^v, and h_μ^q introduced for contexts in the beginning of Sect. 3.6.

For the development of these characterizations, we will use both the right and the left quotient of a tree series. For every $r \in S\langle\langle T_\Sigma\rangle\rangle$ and $\zeta \in C_\Sigma$, the *right quotient of r with respect to ζ* is the tree series $r\zeta^{-1} \in S\langle\langle T_\Sigma\rangle\rangle$, where

$(r\zeta^{-1}, \xi) = (r, \zeta \cdot_z \xi)$ for every $\xi \in T_\Sigma$. For the definition of the left quotient, we need mappings of the type $C_\Sigma \to S$. Since they are very similar to tree series, we can adapt the notions and operations from tree series to this setting. We call a mapping of this type a *context series* and denote the class of all context series by $S\langle\langle C_\Sigma \rangle\rangle$. Then for every $r \in S\langle\langle T_\Sigma \rangle\rangle$ and $\xi \in T_\Sigma$, the *left quotient of* r *with respect to* ξ is the context series $\xi^{-1}r \in S\langle\langle C_\Sigma \rangle\rangle$ defined by $(\xi^{-1}r, \zeta) = (r, \zeta \cdot_z \xi)$ for every $\zeta \in C_\Sigma$.

Characterizations for Fields

In this subsection, we assume that S *is a field.*

Since S is a field, both $(S\langle\langle T_\Sigma \rangle\rangle, +, \widetilde{0})$ and $(S\langle\langle C_\Sigma \rangle\rangle, +, \widetilde{0})$ are S-vector spaces. For every $r \in S\langle\langle T_\Sigma \rangle\rangle$, we denote by RQ_r the subspace of $S\langle\langle T_\Sigma \rangle\rangle$ generated by all the right quotients $r\zeta^{-1}$ for $\zeta \in C_\Sigma$, and by LQ_r the subspace of $S\langle\langle C_\Sigma \rangle\rangle$ generated by all the left quotients $\xi^{-1}r$ for $\xi \in T_\Sigma$. Then we can prove the following relation.

Lemma 3.28 ([24], Theorem 3.1). *Let* $r \in S\langle\langle T_\Sigma \rangle\rangle$. *Then* RQ_r *is finite-dimensional iff* LQ_r *is finite-dimensional, and in this case* $\dim(RQ_r) = \dim(LQ_r)$.

Proof. Assume LQ_r is n-dimensional and let $\xi_1^{-1}r, \ldots, \xi_n^{-1}r$ be a basis of LQ_r. Consider the mapping $\psi : RQ_r \to S^n$, where $\psi(r\zeta^{-1}) = ((r, \zeta \cdot_z \xi_1), \ldots, (r, \zeta \cdot_z \xi_n))$ for every context $\zeta \in C_\Sigma$, and then ψ is linearly extended to RQ_r. We can prove that ψ is injective, which implies $\dim(RQ_r) \leq n$. To prove the injectivity of ψ, we take an arbitrary element $s = a_1(r\zeta_1^{-1}) + \cdots + a_m(r\zeta_m^{-1})$ of RQ_r and show that $\psi(s) = 0^n$ implies $s = \widetilde{0}$. In fact, $\psi(s) = 0^n$ means $\sum_{i=1}^m a_i \cdot (r, \zeta_i \cdot_z \xi_j) = 0$ for $j = 1, \ldots, n$. Now for every $\xi \in T_\Sigma$, we have $(s, \xi) = \sum_{i=1}^m a_i \cdot (r\zeta_i^{-1}, \xi) = \sum_{i=1}^m a_i \cdot (\xi^{-1}r, \zeta_i)$. By letting $\xi^{-1}r = \sum_{j=1}^n b_j(\xi_j^{-1}r)$ and reordering the members of the sum appropriately, we obtain $(s, \xi) = 0$. Analogously, by assuming that RQ_r is finite-dimensional, we can prove that $\dim(LQ_r) \leq \dim(RQ_r)$. $\square$

For every recognizable tree series r, the S-vector space LQ_r is finite-dimensional. In [24], Theorem 2.1 even the equivalence was proved, i.e., $r \in \mathrm{Rec}(\Sigma, S)$ iff LQ_r is finite-dimensional. However, since we will have a slightly different proof of this equivalence (cf. Theorem 3.31), we now cite only the mentioned implication.

Lemma 3.29 ([24], Theorem 2.1). *Let* $r \in S\langle\langle T_\Sigma \rangle\rangle$. *If* $r \in \mathrm{Rec}(\Sigma, S)$, *then* LQ_r *is finite-dimensional.*

Proof. Let $\mathcal{A} = (Q, \Sigma, S, \mu, \nu)$ be a wta such that $r_\mathcal{A} = r$. We define the mapping $\varphi : S^Q \to S\langle\langle C_\Sigma \rangle\rangle$ by $(\varphi(v), \zeta) = h_\mu^v(\zeta) \cdot \nu$ for every $v \in S^Q$ and $\zeta \in C_\Sigma$. The fact that all the mappings $\mu_\mathcal{A}^v$ are multilinear implies that φ

is a linear mapping between the two S-vector spaces S^Q and $S\langle\langle C_\Sigma\rangle\rangle$. Then $(\varphi(h_\mu(\xi)), \zeta) = h_\mu^{h_\mu(\xi)}(\zeta) \cdot \nu = h_\mu(\zeta \cdot_z \xi) \cdot \nu = (r, \zeta \cdot_z \xi) = (\xi^{-1}r, \zeta)$, and thus $\varphi(h_\mu(\xi)) = \xi^{-1}r$. Hence LQ_r is a subspace of the range $\mathrm{ran}(\varphi)$ of φ, and thus $\dim(LQ_r) \leq \dim(\mathrm{ran}(\varphi))$. Since, in general, for a linear mapping $\varphi : V \to V'$ between two S-vector spaces V and V' the dimension of $\mathrm{ran}(\varphi)$ cannot be larger than $\dim(V)$, we obtain that in our case $\dim(\mathrm{ran}(\varphi)) \leq \dim(S^Q) = |Q|$. Thus, LQ_r is finite-dimensional. □

Next, we recall from [23, 18] the facts that the S-vector space LQ_r can be enriched to an S-Σ-vector space and that LQ_r and the so-called syntactic S-Σ-vector space of r are isomorphic.

The enrichment of the S-vector space $(LQ_r, +, \widetilde{0})$ with a Σ-algebraic structure is done in the following way. Since every S-vector space has a basis (assuming Zorn's lemma), also LQ_r has a basis, say B. Now for every $\sigma \in \Sigma^{(k)}$, we define the mapping $\theta_r(\sigma) : B^k \to LQ_r$ by $\theta_r(\sigma)(\eta_1^{-1}r, \dots, \eta_k^{-1}r) = \sigma(\eta_1, \dots, \eta_k)^{-1}r$ for all base vectors $\eta_1^{-1}r, \dots, \eta_k^{-1}r \in B$. Then we extend $\theta_r(\sigma)$ to a k-ary multilinear mapping on LQ_r which we also denote by $\theta_r(\sigma)$. Thus, we obtain the S-Σ-vector space $(LQ_r, +, \widetilde{0}, \theta_r)$.

For the definition of the syntactic S-Σ-vector space of r, we consider the S-Σ-vector space $(S\langle T_\Sigma\rangle, +, \widetilde{0}, \mathrm{top})$ of polynomials with $\mathrm{top}(\sigma) = \mathrm{top}_\sigma$ for every $\sigma \in \Sigma$. This is the initial algebra in the class of all S-Σ-vector spaces. Now let $r \in S\langle\langle T_\Sigma\rangle\rangle$ be a tree series. For a polynomial $s = a_1.\xi_1 + \cdots + a_k.\xi_k$ in $S\langle T_\Sigma\rangle$, we define the *left quotient of r with respect to s* by letting $s^{-1}r = a_1(\xi_1^{-1}r) + \cdots + a_k(\xi_k^{-1}r)$. Then we define the equivalence relation $\sim_r$ over $S\langle T_\Sigma\rangle$ such that for every $s_1, s_2 \in S\langle T_\Sigma\rangle$ we have $s_1 \sim_r s_2$ if and only if $s_1^{-1}r = s_2^{-1}r$. It is not difficult to prove that $\sim_r$ is a congruence relation over the S-Σ-vector space $S\langle T_\Sigma\rangle$. We call $\sim_r$ the *syntactic congruence* of r, and we call the quotient space $(S\langle T_\Sigma\rangle/\sim_r, +_{\sim_r}, [\widetilde{0}]_{\sim_r}, \mathrm{top}_{\sim_r})$ the *syntactic S-Σ-vector space of r*.

Next, we relate the two S-Σ-vector spaces $S\langle T_\Sigma\rangle/\sim_r$ and LQ_r. For the initial homomorphism $\Phi_r : S\langle T_\Sigma\rangle \to LQ_r$, it is easy to prove that $\Phi_r(s) = s^{-1}r$ for every $s \in S\langle T_\Sigma\rangle$ and that Φ_r is surjective. Since the kernel of Φ_r is $\sim_r$, we immediately obtain the following result by applying the homomorphism theorem of universal algebra ([73], Theorem 11.1).

Lemma 3.30 ([23], Proposition 3). *For every $r \in S\langle\langle T_\Sigma\rangle\rangle$, the S-Σ-vector spaces $S\langle T_\Sigma\rangle/\sim_r$ and LQ_r are isomorphic.*

Now we can prove the first Myhill–Nerode-like theorem for recognizable tree series. For this, let $r \in S\langle\langle T_\Sigma\rangle\rangle$ and $\sim$ be a congruence on the S-Σ-vector space $S\langle T_\Sigma\rangle$. We say that $\sim$ *respects* r if there is a linear form $\gamma : S\langle T_\Sigma\rangle/\sim \to S$ such that $(r, \xi) = \gamma([\xi]_\sim)$ for every $\xi \in T_\Sigma$.

Theorem 3.31 ([24], Theorems 2.1 and 3.1; [23], Propositions 2 and 3). *Let $r \in S\langle\langle T_\Sigma\rangle\rangle$. Then the following five statements are equivalent:*

(A) $r \in \mathrm{Rec}(\Sigma, S)$.

(B) There is a congruence $\sim$ on $S\langle T_\Sigma\rangle$ such that $S\langle T_\Sigma\rangle/\!\sim$ is finite-dimensional and $\sim$ respects r.
(C) The S-vector space $S\langle T_\Sigma\rangle/\!\sim_r$ is finite-dimensional.
(D) The S-vector space RQ_r is finite-dimensional.
(E) The S-vector space LQ_r is finite-dimensional.

Proof. Statement (A) implies statement (E) by Lemma 3.29. Statements (C), (D), and (E) are equivalent due to Lemmata 3.28 and 3.30. Next, we prove that statement (C) implies statement (B). For this, let $\sim = \sim_r$ and define the linear form $\gamma : S\langle T_\Sigma\rangle/\!\sim \to S$ such that $\gamma([s]_\sim) = (s^{-1}r, z)$. Since $(s^{-1}r, z) = a_1 \cdot (r, \xi_1) + \cdots + a_k \cdot (r, \xi_k)$ for every polynomial $s = a_1.\xi_1 + \cdots + a_k.\xi_k$, γ is a linear form. Moreover, we have $\gamma([\xi]_\sim) = (r, \xi)$.

Finally, we prove that statement (B) implies statement (A), where we abbreviate an equivalence class $[\xi]_\sim$ by $[\xi]$. Let $Q = \{[\xi_1], \ldots, [\xi_n]\}$, where $\xi_1, \ldots, \xi_n \in T_\Sigma$, be a basis of $S\langle T_\Sigma\rangle/\!\sim$ and construct the wta $\mathcal{A} = (Q, \Sigma, S, \mu, \nu)$ in the following way. For every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, and $1 \leq i, i_1, \ldots, i_k \leq n$, let $\mu_k(\sigma)_{[\xi_{i_1}]\ldots[\xi_{i_k}],[\xi_i]} = [\sigma(\xi_{i_1}, \ldots, \xi_{i_k})]_{[\xi_i]}$, where the latter denotes the coefficient of $[\xi_i]$ in the representation of $[\sigma(\xi_{i_1}, \ldots, \xi_{i_k})]$ as a linear combination of the base vectors. Moreover, for every $1 \leq i \leq n$, let $\nu_{[\xi_i]} = \gamma([\xi_i])$. Note that, in general, $\mathcal{A}$ is nondeterministic. We can prove easily that $h_\mu(\xi) = h_\sim(\xi)$ for every $\xi \in T_\Sigma$, where $h_\sim : S\langle T_\Sigma\rangle \to S\langle T_\Sigma\rangle/\!\sim$ is the canonical S-Σ-vector space homomorphism and we identify the isomorphic vector spaces S^Q and $S\langle T_\Sigma\rangle/\sim$. Now let $\xi \in T_\Sigma$ and assume that $[\xi] = a_1.[\xi_1] + \cdots + a_n.[\xi_n]$ for some $a_1, \ldots, a_n \in S$. Then $(r_\mathcal{A}, \xi) = h_\mu(\xi) \cdot \nu = \sum_{i=1}^n [\xi]_{[\xi_i]} \cdot \gamma([\xi_i]) = \gamma(\sum_{i=1}^n a_i \cdot [\xi_i]) = \gamma([\xi]) = (r, \xi)$, which proves that $\mathcal{A}$ recognizes r. ☐

Characterizations for Semifields

Now we show a second Myhill–Nerode-like theorem, which characterizes recognizable tree series in terms of congruences of finite index over the term algebra T_Σ. However, this characterization holds only for bu-deterministically recognizable tree series, while we can relax from fields to semifields (cf. Theorem 3.35). Since some of the auxiliary results which we need and which are interesting on their own hold even for arbitrary commutative and zero-divisor free semirings (and every semifield is zero-divisor free), we do not immediately require that S is a semifield, but we make the following assumption.

> *In this subsection, we assume that S is commutative and zero-divisor free.*

Now we define the Myhill–Nerode relation $\equiv_r \subseteq T_\Sigma \times T_\Sigma$ by $\xi_1 \equiv_r \xi_2$ iff there are $a, b \in S \setminus \{0\}$ such that $a(\xi_1^{-1}r) = b(\xi_2^{-1}r)$. In fact, the factors a and b are needed, because if they were dropped then, e.g., $\equiv_{\text{size}}$ would have infinite index for $S = \mathsf{Trop}_{\text{sf}}$, where $\mathsf{Trop}_{\text{sf}}$ is the tropical semifield of reals. However, size $\in$ bud-Rec$(\Sigma, \mathsf{Trop}_{\text{sf}})$ (cf. [11], Example 3), and thus recognizability would not imply a finite index of the Myhill–Nerode relation. Note that for different semirings with the same carrier set S, a tree series $r \in S\langle\langle T_\Sigma \rangle\rangle$ may yield different $\equiv_r$ relations with respect to those semirings.

It is straightforward to prove that $\equiv_r$ is an equivalence relation (the zero-divisor freeness guarantees transitivity) and is invariant under contexts, i.e., $\equiv_r$ is a congruence with respect to the Σ-term algebra (cf. [109], Lemma 2).

Next, let us prove that $\equiv_r$ has finite index if $r \in \text{bud-Rec}(\Sigma, S)$. As technical preparation, for a total bu-deterministic wta $\mathcal{A} = (Q, \Sigma, S, \mu, \nu)$, we define the *underlying deterministic bottom-up tree automaton* $B(\mathcal{A}) = (Q, \Sigma, \delta, F)$ such that $F = \{q \in Q \mid \nu_q \neq 0\}$, $\delta = (\delta_\sigma | \sigma \in \Sigma)$ and $\delta_\sigma(q_1, \ldots, q_k) = q$, where q is the unique state such that $\mu_k(\sigma)_{q_1 \ldots q_k, q} \neq 0$. Let $\equiv_{B(\mathcal{A})}$ be the kernel of the homomorphism $h_Q : T_\Sigma \to Q$. By standard arguments, it follows that $\equiv_{B(\mathcal{A})}$ is a congruence on T_Σ which has finite index at most $|Q|$.

We note that, for a total bu-deterministic wta $\mathcal{A}$ and every $\zeta \in C_\Sigma$, $\xi \in T_\Sigma$, and $q \in Q$, we have that $h_\mu(\zeta \cdot_z \xi)_q = h_\mu^p(\zeta)_q \cdot h_\mu(\xi)_p$ where $p = h_Q(\xi)$. This property will be used in the proof of the next lemma.

Lemma 3.32 ([109], Theorem 4). *For every total bu-deterministic wta $\mathcal{A}$, the index of $\equiv_{r_\mathcal{A}}$ is at most the number of states of $\mathcal{A}$. In particular, $\equiv_r$ has finite index for every $r \in$ bud-Rec(Σ, S).*

Proof. Let $\mathcal{A} = (Q, \Sigma, S, \mu, \nu)$ be a total bu-deterministic wta. Since the index of $\equiv_{B(\mathcal{A})}$ is at most $|Q|$, it suffices to show that $\equiv_{B(\mathcal{A})} \subseteq \equiv_{r_\mathcal{A}}$. To prove this inclusion, let $\xi_1, \xi_2 \in T_\Sigma$ such that $\xi_1 \equiv_{B(\mathcal{A})} \xi_2$. Hence, $h_Q(\xi_1) = h_Q(\xi_2)$, and thus also $h_Q(\zeta \cdot_z \xi_1) = h_Q(\zeta \cdot_z \xi_2)$ for every $\zeta \in C_\Sigma$. Let p abbreviate $h_Q(\xi_1)$. Now consider $\zeta \in C_\Sigma$ and let q abbreviate $h_Q(\zeta \cdot_z \xi_1)$. Then $h_\mu(\xi_2)_p \cdot (r, \zeta \cdot_z \xi_1)$ $= h_\mu(\xi_2)_p \cdot h_\mu(\zeta \cdot_z \xi_1)_q \cdot \nu_q = h_\mu(\xi_2)_p \cdot h_\mu^p(\zeta)_q \cdot h_\mu(\xi_1)_p \cdot \nu_q = h_\mu(\xi_1)_p \cdot h_\mu(\zeta \cdot_z \xi_2)_q \cdot \nu_q$ $= h_\mu(\xi_1)_p \cdot (r, \zeta \cdot_z \xi_2)$. Hence, $\xi_1 \equiv_{r_\mathcal{A}} \xi_2$.

For the proof of the second claim, recall that for every $r \in$ bud-Rec(Σ, S) there is a total bu-deterministic wta $\mathcal{A}$ such that $r = r_\mathcal{A}$. Then the second claim follows from the first one. □

Next, we prove that $\equiv_r$ has a particular property, called (MN). In the definition of (MN), we will have to discard those trees $\xi \in T_\Sigma$ that cannot be completed to a tree in supp(r). Formally, we define $L_r = \{\xi \in T_\Sigma \mid \xi^{-1} r = \widetilde{0}\}$. Now let $\equiv$ be an equivalence relation on T_Σ. Then we say that $\equiv$ *satisfies (MN) for r* if there is a *representation mapping* φ for $\equiv$, i.e., a mapping $\varphi : T_\Sigma/\!\equiv \to T_\Sigma$ such that $\varphi([\xi]_\equiv) \in [\xi]_\equiv$ for every $\xi \in T_\Sigma$, and there is a mapping $a_\varphi : T_\Sigma \to S \setminus \{0\}$ such that:

(MN1) For every $\xi \in T_\Sigma$, we have that $(r, \xi) = a_\varphi(\xi) \cdot (r, \varphi([\xi]_\equiv))$.
(MN2) For every $\xi = \sigma(\xi_1, \ldots, \xi_k) \in T_\Sigma \setminus L_r$ and $\xi' = \sigma(\xi'_1, \ldots, \xi'_k) \in T_\Sigma$ with $\xi_i \equiv \xi'_i$ for every $1 \leq i \leq k$, we have that

$$a_\varphi(\xi'_k) \cdot \dots \cdot a_\varphi(\xi'_1) \cdot a_\varphi(\xi) = a_\varphi(\xi_k) \cdot \dots \cdot a_\varphi(\xi_1) \cdot a_\varphi(\xi').$$

We note that, if S is a semifield, Condition (MN2) amounts to say that for every $\xi = \sigma(\xi_1, \ldots, \xi_k) \in T_\Sigma \setminus L_r$ there is a $b \in S \setminus \{0\}$ such that $a_\varphi(\xi) =$

$b \cdot \prod_{i=1}^{k} a_\varphi(\xi_i)$. In the sequel, we will abbreviate $[\xi]_\equiv$ by $[\xi]$ and $\varphi([\xi]_\equiv)$ by $\overline{\xi}$, because $\equiv$ and φ will be clear from the context.

Lemma 3.33. *Let $r \in S\langle\langle T_\Sigma\rangle\rangle$, where S is a semifield. Then $\equiv_r$ satisfies (MN) for r.*

Proof. Take any representation mapping $\varphi : T_\Sigma/{\equiv_r} \to T_\Sigma$. Since $\xi \equiv_r \overline{\xi}$ for every $\xi \in T_\Sigma$, there are $a, b \in S \setminus \{0\}$ such that $a(\xi^{-1}r) = b(\overline{\xi}^{-1}r)$. Let us fix some arbitrary such a and b, and call them a_ξ and b_ξ henceforth. Then for every context $\zeta \in C_\Sigma$, we have $a_\xi \cdot (r, \zeta \cdot_z \xi) = b_\xi \cdot (r, \zeta \cdot_z \overline{\xi})$. Now we define the mapping $a_\varphi : T_\Sigma \to S \setminus \{0\}$ by $a_\varphi(\xi) = a_\xi^{-1} \cdot b_\xi$ and get

$$(r, \zeta \cdot_z \xi) = a_\varphi(\xi) \cdot \big(r, \zeta \cdot_z \overline{\xi}\big). \tag{2}$$

In particular, with $\zeta = z$, we obtain $(r, \xi) = a_\varphi(\xi) \cdot (r, \overline{\xi})$, which proves that (MN1) holds.

Now let $\xi_1, \ldots, \xi_k \in T_\Sigma$ such that $\xi = \sigma(\xi_1, \ldots, \xi_k) \notin L_r$. Thus, there is a context $\zeta_0 \in C_\Sigma$ such that $(r, \zeta_0 \cdot_z \xi) \neq 0$, hence by (2) also $(r, \zeta_0 \cdot_z \overline{\xi}) \neq 0$.

Let us compute as follows:

$$\begin{aligned}
(r, \zeta_0 \cdot_z \xi) &= \big(r, \big(\zeta_0 \cdot_z \sigma(z, \xi_2, \ldots, \xi_k)\big) \cdot_z \xi_1\big) \\
&= a_\varphi(\xi_1) \cdot \big(r, \big(\zeta_0 \cdot_z \sigma(z, \xi_2, \ldots, \xi_k)\big) \cdot_z \overline{\xi_1}\big) \\
&= a_\varphi(\xi_1) \cdot \big(r, \zeta_0 \cdot_z \sigma\big(\overline{\xi_1}, \xi_2, \ldots, \xi_k\big)\big),
\end{aligned}$$

where at the second equation we applied (2) with $\xi = \xi_1$. Clearly, this process can be applied to all the ξ_i's, thus we obtain

$$(r, \zeta_0 \cdot_z \xi) = \prod_{i=1}^{k} a_\varphi(\xi_i) \cdot \big(r, \zeta_0 \cdot_z \sigma\big(\overline{\xi_1}, \ldots, \overline{\xi_k}\big)\big). \tag{3}$$

Using (2) with $\zeta = \zeta_0$, the fact that the inverse of $(r, \zeta_0 \cdot_z \overline{\xi})$ exists, and (3), we obtain that $\prod_{i=1}^{k} a_\varphi(\xi_i)^{-1} \cdot a_\varphi(\xi) = (r, \zeta_0 \cdot_z \sigma(\overline{\xi_1}, \ldots, \overline{\xi_k})) \cdot (r, \zeta_0 \cdot_z \overline{\xi})^{-1}$.

Now we consider $\xi_1', \ldots, \xi_k' \in T_\Sigma$ such that $\xi_i' \in [\xi_i]$ for every $1 \le i \le k$, and we denote $\sigma(\xi_1', \ldots, \xi_k')$ by ξ'. Thus, $\xi' \equiv_r \xi$, because $\equiv_r$ is a congruence. Since $(r, \zeta_0 \cdot_z \xi) \neq 0$, also $(r, \zeta_0 \cdot_z \xi') \neq 0$, and hence $\xi' \notin L_r$. Thus, we can prove (3) in the same way as above for ξ' instead of ξ, and obtain $\prod_{i=1}^{k} a_\varphi(\xi_i')^{-1} \cdot a_\varphi(\xi') = (r, \zeta_0 \cdot_z \sigma(\overline{\xi_1'}, \ldots, \overline{\xi_k'})) \cdot (r, \zeta_0 \cdot_z \overline{\xi'})^{-1}$. Since $\overline{\xi_i} = \overline{\xi_i'}$ and $\overline{\xi} = \overline{\xi'}$, we obtain eventually $\prod_{i=1}^{k} a_\varphi(\xi_i)^{-1} \cdot a_\varphi(\xi) = \prod_{i=1}^{k} a_\varphi(\xi_i')^{-1} \cdot a_\varphi(\xi')$ which is the same as (MN2) after multiplying with $\prod_{i=1}^{k} a_\varphi(\xi_i)$ and $\prod_{i=1}^{k} a_\varphi(\xi_i')$. □

We will need the following auxiliary result.

Lemma 3.34. *Let $r \in S\langle\langle T_\Sigma\rangle\rangle$ and let $\equiv$ be a congruence on T_Σ which satisfies (MN) for r. Then $\equiv$ saturates L_r, i.e., L_r is the union of some equivalence classes.*

Proof. Let φ and a_φ be the mappings such that (MN) is satisfied for r. Moreover, let $\xi, \xi' \in T_\Sigma$ such that $\xi \equiv \xi'$ and $\xi \in L_r$. Since $\equiv$ is a congruence, we have that $[\zeta \cdot_z \xi] = [\zeta \cdot_z \xi']$ for every context $\zeta \in C_\Sigma$, and thus $\overline{\zeta \cdot_z \xi} = \overline{\zeta \cdot_z \xi'}$. By (MN1), we have that $(r, \zeta \cdot_z \xi) = a_\varphi(\zeta \cdot_z \xi) \cdot (r, \overline{\zeta \cdot_z \xi})$ for every context $\zeta \in C_\Sigma$. Since $a_\varphi(\zeta \cdot_z \xi) \neq 0$ and S is zero-divisor free, it follows that $(r, \overline{\zeta \cdot_z \xi}) = 0$, and thus $(r, \overline{\zeta \cdot_z \xi'}) = 0$. Hence, $(r, \zeta \cdot_z \xi') = a_\varphi(\zeta \cdot_z \xi') \cdot (r, \overline{\zeta \cdot_z \xi'}) = 0$ again by (MN1). Since this implication holds for every context ζ, we obtain that $\xi' \in L_r$. □

Now we can prove a Myhill–Nerode-like theorem for bu-deterministically recognizable tree series over semifields.

Theorem 3.35 ([14], Theorem 7.3.1). *Let S be a semifield and $r \in S\langle\langle T_\Sigma \rangle\rangle$. Then the following three statements are equivalent:*

(A) $r \in \text{bud-Rec}(\Sigma, S)$.
(B) There is a congruence $\equiv$ on T_Σ which has finite index and satisfies (MN) for r.
(C) $\equiv_r$ has finite index.

Proof. Statement (A) implies statement (C) by Lemma 3.32. By Lemma 3.33, we have that statement (C) implies statement (B).

For the proof that statement (B) implies statement (A), let $\varphi : T_\Sigma/\equiv \to T_\Sigma$ and $a_\varphi : T_\Sigma \to S \setminus \{0\}$ such that (MN1) and (MN2) hold. We construct the bu-deterministic wta $\mathcal{A} = (Q, \Sigma, S, \mu, \nu)$ where $Q = T_\Sigma/\equiv$, $\nu_{[\xi]} = (r, \varphi([\xi]))$ for every $\xi \in T_\Sigma$, and for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, and $[\xi_1], \ldots, [\xi_k], [\xi] \in Q$:

$$\mu_k(\sigma)_{[\xi_1]\ldots[\xi_k],[\xi]} = \begin{cases} \prod_{i=1}^k a_\varphi(\xi_i)^{-1} \cdot a_\varphi(\xi) & \text{if } [\xi] = [\sigma(\xi_1, \ldots, \xi_k)] \text{ and } \xi \notin L_r, \\ 0 & \text{otherwise.} \end{cases}$$

We note that $\mu_k(\sigma)$ is well defined. To see this, let $[\xi_1'] = [\xi_1], \ldots, [\xi_k'] = [\xi_k]$ and $[\xi'] = [\xi]$. Then $[\sigma(\xi_1', \ldots, \xi_k')] = [\sigma(\xi_1, \ldots, \xi_k)]$, hence $[\xi] = [\sigma(\xi_1, \ldots, \xi_k)]$ iff $[\xi'] = [\sigma(\xi_1', \ldots, \xi_k')]$. Moreover, $\xi \notin L_r$ iff $\xi' \notin L_r$ by Lemma 3.34. Finally, the property (MN2) of $\equiv$ assures that $\mu_k(\sigma)_{[\xi_1']\ldots[\xi_k'],[\xi']}$ has the same value.

Next, it is straightforward to prove by induction on $\xi \in T_\Sigma$ that for every $[\xi'] \in Q$ we have

$$h_\mu(\xi)_{[\xi']} = \begin{cases} a_\varphi(\xi) & \text{if } [\xi] = [\xi'] \text{ and } \xi \notin L_r, \\ 0 & \text{otherwise.} \end{cases}$$

In the proof, we have to use the fact that for every tree $\xi = \sigma(\xi_1, \ldots, \xi_k)$, if $\xi \notin L_r$, then $\xi_i \notin L_r$.

Finally, for every $\xi \in T_\Sigma$, we obtain $(r_\mathcal{A}, \xi) = h_\mu(\xi) \cdot \nu = h_\mu(\xi)_{[\xi]} \cdot \nu_{[\xi]}$ because $[\xi] \neq [\xi']$ implies $h_\mu(\xi)_{[\xi']} = 0$. If in addition $\xi \notin L_r$, then $h_\mu(\xi)_{[\xi]} \cdot \nu_{[\xi]} = a_\varphi(\xi) \cdot (r, \varphi([\xi])) = (r, \xi)$ by (MN1). If $\xi \in L_r$, and thus, in particular, $(r, \xi) = 0$, then $h_\mu(\xi)_{[\xi]} \cdot \nu_{[\xi]} = 0 \cdot (r, \varphi([\xi])) = 0 = (r, \xi)$. □

To show the use of Theorem 3.35, let us consider the tree series size. By an obvious automaton construction, we have that size $\in$ bud-Rec$(\Sigma, \mathsf{Trop}_{\mathrm{sf}})$, and hence $\equiv_{\mathrm{size}}$ over $\mathsf{Trop}_{\mathrm{sf}}$ is of finite index, in fact, the index is 1. On the other hand, we can prove that size $\notin$ bud-Rec$(\Sigma, \mathbb{Q})$ if Σ contains at least a binary symbol σ and a nullary symbol α and $(\mathbb{Q}, +, \cdot, 0, 1)$ is the field of rational numbers (cf. [14], Example 7.3.2). For this, we prove that $\xi_1 \equiv_{\mathrm{size}} \xi_2$ iff $(\mathrm{size}, \xi_1) = (\mathrm{size}, \xi_2)$ which shows that $\equiv_{\mathrm{size}}$ has infinite index over $\mathbb{Q}$. Assume that $(\mathrm{size}, \xi_1) = (\mathrm{size}, \xi_2)$. Then for every context $\zeta \in C_\Sigma$, we have $(\mathrm{size}, \zeta \cdot_z \xi_1) = (\mathrm{size}, \zeta \cdot_z \xi_2)$ and hence $\xi_1 \equiv_{\mathrm{size}} \xi_2$. Now assume that $\xi_1 \equiv_{\mathrm{size}} \xi_2$. Hence, there is an $a \in \mathbb{Q} \setminus \{0\}$ such that for every $\zeta \in C_\Sigma$ we have $(\mathrm{size}, \zeta \cdot_z \xi_1) = a \cdot (\mathrm{size}, \zeta \cdot_z \xi_2)$. Instantiating this equation twice, with $\zeta = z$ and $\zeta = \sigma(\alpha, z)$, we obtain: $(\mathrm{size}, \xi_1) = a \cdot (\mathrm{size}, \xi_2)$ and $2 + (\mathrm{size}, \xi_1) = a \cdot (2 + (\mathrm{size}, \xi_2))$, respectively; this implies that $a = 1$, and hence $(\mathrm{size}, \xi_1) = (\mathrm{size}, \xi_2)$.

Characterizations for Commutative, Zero-Divisor Free Semirings

Now let us recall the third Myhill–Nerode-like characterization which is due to [109]. It shows, for the class of commutative and zero-divisor free semirings, a characterization of bud-Rec(Σ, S) in terms of a slightly different property. We say that a congruence $\equiv$ on T_Σ *respects a tree series* $r \in S\langle\langle T_\Sigma \rangle\rangle$ if there exists a mapping $f : T_\Sigma/{\equiv} \to S$ and a mapping $c : T_\Sigma \to S \setminus \{0\}$ such that:

- $(r, \xi) = c(\xi) \cdot f([\xi]_\equiv)$ for every $\xi \in T_\Sigma$.
- For every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, there is a mapping $b_\sigma : (T_\Sigma/{\equiv})^k \to S$ such that for every $\xi_1, \ldots, \xi_k \in T_\Sigma$ we have that $c(\sigma(\xi_1, \ldots, \xi_k)) = c(\xi_1) \cdot \cdots \cdot c(\xi_k) \cdot b_\sigma([\xi_1]_\equiv, \ldots, [\xi_k]_\equiv)$.

Theorem 3.36 ([109], Theorem 19). *Let S be a commutative and zero-divisor free semiring. Moreover, let $r \in S\langle\langle T_\Sigma \rangle\rangle$. Then the following two statements are equivalent:*

(A) $r \in$ bud-Rec(Σ, S).
(B) There is a congruence $\equiv$ on T_Σ which has finite index and respects r.

We note that, for every semifield S, the second statements of Theorems 3.36 and 3.35 are equivalent.

3.8 Equational Tree Series

By definition, an equational subset of a Σ-algebra $\mathcal{A}$ is a component of the least solution of some system of linear equations [116, 32, 69]. It was shown in [116] that every equational subset of $\mathcal{A}$ is the homomorphic image of a recognizable tree language and vice versa. In particular, the class of equational subsets of T_Σ is the class of recognizable Σ-tree languages. Here, we show how these results are generalized to recognizable tree series [7, 88, 21, 58]. Since the solutions of systems of linear equations are obtained by fixpoints, we refer the reader to [57] for an introduction to that part of the theory of fixpoints that has applications to weighted automata.

A *system of linear equations* (for short: system) is a finite family E of equations $z_i = s_i$ where $1 \le i \le n$ for some $n \ge 1$, and $Z_n = \{z_1, \ldots, z_n\}$ is a set of variables, and $s_i \in S\langle T_\Sigma(Z_n)\rangle$. The system E is *proper* if for every $1 \le i \le n$, the tree series s_i is z_j-proper, i.e., $(s_i, z_j) = 0$, for every $1 \le j \le n$.

We solve such systems in S-Σ-semimodules. For this, we generalize the concept of OI-substitution as introduced in Sect. 3.3 as follows. Let $(V, +, 0, \theta)$ be an S-Σ-semimodule and $\bar{v} = (v_1, \ldots, v_n) \in V^n$. Moreover, let $s \in S\langle T_\Sigma(Z_n)\rangle$. The *OI-substitution of* $\bar{v}$ *into* s is the element $s \leftarrow_{\mathrm{OI}} \bar{v}$ in V which is defined to be $\sum_{\zeta \in \mathrm{supp}(s)} (s, \zeta)(\zeta \leftarrow_{\mathrm{OI}} \bar{v})$; the element $\zeta \leftarrow_{\mathrm{OI}} \bar{v} \in V$ is defined in exactly the same way as $\zeta \leftarrow_{OI,\bar{z}} \bar{v}$ (with $\bar{z} = (z_1, \ldots, z_n)$ and $\bar{v} \in S\langle\langle T_\Sigma\rangle\rangle^n$) except that in case (ii) the symbol σ is not interpreted by top_σ but by $\theta(\sigma)$. Note that in the expression $(s, \zeta)(\zeta \leftarrow_{\mathrm{OI}} \bar{v})$ the subexpressions (s, ζ) and $\zeta \leftarrow_{\mathrm{OI}} \bar{v}$ are combined by means of the scalar multiplication of the S-semimodule $(V, +, 0)$.

Now let E be the system $z_i = s_i$ with $1 \le i \le n$ and $(V, +, 0, \theta)$ an S-Σ-semimodule. A *solution of* E in V is a vector $\bar{v} = (v_1, \ldots, v_n) \in V^n$ such that $v_i = (s_i \leftarrow_{\mathrm{OI}} \bar{v})$ for every i. In other words, $\bar{v}$ is a fixpoint of the mapping $\Phi_E : V^n \to V^n$ defined by $\Phi_E(\bar{u}) = (s_1 \leftarrow_{\mathrm{OI}} \bar{u}, \ldots, s_n \leftarrow_{\mathrm{OI}} \bar{u})$ for every $\bar{u} \in V^n$. Let additionally the monoid $(V, +, 0)$ be naturally ordered by $\sqsubseteq$; this relation is extended componentwise to V^n. An element $v \in V$ is *equational* (*p-equational*) if it is a component of the least solution of a system (resp., proper system), if it exists. The *class of all equational elements (p-equational elements)* in the S-Σ-semimodule V is denoted by $\mathrm{Eq}(V)$ (resp., $\mathrm{Eq_p}(V)$).

Solutions over the S-Σ-Semimodule of Tree Series

Before dealing with equational elements in general, we first solve proper systems in the particular S-Σ-semimodule $(S\langle\langle T_\Sigma\rangle\rangle, +, \widetilde{0}, \mathrm{top})$ where $\mathrm{top}(\sigma) = \mathrm{top}_\sigma$ for every $\sigma \in \Sigma$. (Note that for the concept of solution we do not need a partial order.)

Lemma 3.37 ([7], Proposition 6.1). *Every proper system has a unique solution in* $(S\langle\langle T_\Sigma\rangle\rangle, +, \widetilde{0}, \mathrm{top})$.

Proof. Let E be a proper system $z_i = s_i$ with $1 \le i \le n$. Moreover, let $\bar{r} = (r_1, \ldots, r_n)$ be a vector of tree series in $S\langle\langle T_\Sigma\rangle\rangle$, and assume that $\bar{r}$ is a solution of E. Hence, $r_i = (s_i \leftarrow_{\mathrm{OI}} \bar{r}) = \sum_{\zeta \in \mathrm{supp}(s_i)} (s_i, \zeta)(\zeta \leftarrow_{\mathrm{OI}} \bar{r})$. We prove that this solution is the only solution.

Since E is proper, every ζ has the form $\zeta = \delta(\zeta_1, \ldots, \zeta_k)$ for some $k \ge 0$, $\delta \in \Sigma^{(k)}$, and $\zeta_1, \ldots, \zeta_k \in T_\Sigma(Z_n)$, and either $\zeta \in T_\Sigma$ or $\zeta \in T_\Sigma(Z_n) \setminus T_\Sigma$. Hence, we can continue with

$$r_i = \sum_{\zeta \in T_\Sigma} (s_i, \zeta).\zeta + \sum_{\delta(\zeta_1, \ldots, \zeta_k) \in T_\Sigma(Z_n) \setminus T_\Sigma} \big(s_i, \delta(\zeta_1, \ldots, \zeta_k)\big)\, \mathrm{top}_\delta(\zeta'_1, \ldots, \zeta'_k)$$

where $\zeta_j' = (\zeta_j \leftarrow_{\mathrm{OI}} \bar{r})$. Then for every $\alpha \in \Sigma^{(0)}$, we have that

$$(r_i, \alpha) = (s_i, \alpha), \tag{4}$$

and for every $\xi = \sigma(\xi_1, \ldots, \xi_k) \in T_\Sigma$ with $k \geq 1$, we have that

$$(r_i, \xi) = (s_i, \xi) + \sum_{\sigma(\zeta_1,\ldots,\zeta_k) \in T_\Sigma(Z_n) \setminus T_\Sigma} \big(s_i, \sigma(\zeta_1, \ldots, \zeta_k)\big) \cdot \prod_{i=1}^{k} (\zeta_i', \xi_i). \tag{5}$$

In the summation, we can restrict to those $\sigma(\zeta_1, \ldots, \zeta_k) \in T_\Sigma(Z_n) \setminus T_\Sigma$ such that $\xi_j \in \mathrm{supp}(\zeta_j \leftarrow_{\mathrm{OI}} \bar{r})$ for every $1 \leq j \leq k$. Hence, $(\zeta_j \leftarrow_{\mathrm{OI}} \bar{r}, \xi_j)$ is the product of coefficients of the form $(r_j, \widehat{\xi})$, where $1 \leq j \leq n$ and $\widehat{\xi}$ is a subtree of ξ_j (equal to ξ_j if $\zeta_j \in Z_n$), and hence a strict subtree of ξ. All in all, the value of (r_i, ξ) is uniquely determined by s_i and by the values of the r_j's on strict subtrees of ξ. Hence, $\bar{r}$ is uniquely determined.

On the other hand, (4) and (5) can be used as defining equations. Thus, $\bar{r}$ exists. □

We note that the solution $\bar{r}$ of E in Lemma 3.37 can be explicitly given by $(r_i, \xi) = (\Phi_E^{m+1}((\widetilde{0}, \ldots, \widetilde{0}))_i, \xi)$ where $m = \mathrm{height}(\xi)$ and $\Phi_E^{m+1}((\widetilde{0}, \ldots, \widetilde{0}))_i$ denotes the ith component of $\Phi_E^{m+1}((\widetilde{0}, \ldots, \widetilde{0}))$.

Example 3.38. Consider $\Sigma = \{\sigma^{(2)}, \gamma^{(1)}, \alpha^{(0)}\}$ and the tree series $\#_{\sigma(z,\alpha)} : T_\Sigma \to \mathbb{N}$ which maps every tree ξ to the number of occurrences of the pattern $\sigma(z, \alpha)$ in ξ (cf. page 322). We consider the proper system E

$$\begin{aligned} z_1 &= \gamma(z_1) + \sigma(z_1, z_2) + \sigma(z_2, z_1) + \sigma(z_2, \alpha), \\ z_2 &= \alpha + \gamma(z_2) + \sigma(z_2, z_2). \end{aligned}$$

It is easy to see that (r_1, r_2) is a solution of E in the S-Σ-semimodule $S\langle\langle T_\Sigma \rangle\rangle$ with $r_1 = \#_{\sigma(z,\alpha)}$ and $r_2 = 1_{(\mathsf{Nat}, T_\Sigma)}$.

There is a close relationship between wta with Boolean root weights and particularly simple proper systems; it is based on the idea of identifying states with variables. A proper system E is *simple* if its equations have the form $z_i = s_i$ with $\mathrm{supp}(s_i) \subseteq \{\sigma(z_{i_1}, \ldots, z_{i_k}) \mid k \geq 0, \sigma \in \Sigma^{(k)}, z_{i_1}, \ldots, z_{i_k} \in Z_n\}$. Then let $\mathcal{A} = (Q, \Sigma, S, \mu, F)$ be a wta with Boolean root weights such that F is a singleton, and E a simple system $z_i = s_i$ with $1 \leq i \leq n$. We call $\mathcal{A}$ and E *related* if $Q = Z_n$ and

$$s_i = \sum_{\substack{k \geq 0,\ \sigma \in \Sigma^{(k)} \\ q_1, \ldots, q_k \in Q}} \mu_k(\sigma)_{q_1 \ldots q_k, z_i} . \sigma(q_1, \ldots, q_k).$$

If $\mathcal{A}$ and E are related, then $\bar{r} = (r_{z_1}, \ldots, r_{z_n})$ is a solution of E, where $r_{z_i} \in S\langle\langle T_\Sigma \rangle\rangle$ is defined by $(r_{z_i}, \xi) = h_\mu(\xi)_{z_i}$ for every $\xi \in T_\Sigma$. This can be proved by a straightforward induction on ξ.

Obviously, if $\mathcal{A}$ and E are related, then E has to be simple. But we can extend this relationship to arbitrary proper systems, because every proper system E can be simulated by a simple system $\widetilde{E}$. For instance, the system E of Example 3.38 is not simple, because the term $\sigma(z_2, \alpha)$ on the right-hand side of the z_1-equation does not have the appropriate form. However, if we define a new system $\widetilde{E}$ which is the same as E except that the disturbing α is replaced by z_3 and the equation $z_3 = \alpha$ is added, then $\widetilde{E}$ is simple; moreover, $\widetilde{E}$ is related to the wta of Example 3.4 (renaming $f, q, \overline{\alpha}$ into z_1, z_2, z_3, resp.).

In general, for two proper systems E_1 and E_2, we say that E_1 is *simulated* by E_2 if every component of a solution of E_1 is also a component of a solution of E_2. The construction of the simple system $\widetilde{E}$ is easy and it proceeds in two steps: in the first step, a system E' is constructed in which the height of every tree $\zeta \in \mathrm{supp}(s_i)$ where $z_i = s_i$ is an equation of E', is not greater than 1; this can be achieved by introducing appropriate auxiliary equations which break down too high trees. In the second step, equations of the form $z_i = s_i$ of E' where $\mathrm{supp}(s_i)$ contains a tree of height 1 with nullary symbols (e.g., $z_i = \sigma(\alpha, z_j)$) are split up into appropriate equations (like $z_i = \sigma(y_1, z_j)$ and $y_1 = \alpha$ with a new variable y_1).

Lemma 3.39 ([7], Lemma 6.3; [58], Corollary 3.6). *For every proper system E, there is a simple system $\widetilde{E}$ which simulates E.*

Now we can show that the recognizable tree series in $S\langle\langle T_\Sigma\rangle\rangle$ are exactly the p-equational elements in $(S\langle\langle T_\Sigma\rangle\rangle, +, \widetilde{0}, \theta)$ where we assume that S is naturally ordered, i.e., $\sqsubseteq$ is a partial order; this partial order extends to the set $S\langle\langle T_\Sigma\rangle\rangle$ by defining $r \sqsubseteq s$ iff $(r, \xi) \sqsubseteq (s, \xi)$ for every $\xi \in T_\Sigma$, and to $(S\langle\langle T_\Sigma\rangle\rangle)^n$ by componentwise comparison. Hence, we can speak about equational elements in $(S\langle\langle T_\Sigma\rangle\rangle, +, \widetilde{0}, \theta)$.

Theorem 3.40 ([58], Corollary 3.6). *Let S be naturally ordered. Then the following two statements hold:*

(A) $\mathrm{Rec}(\Sigma, S) = \mathrm{Eq_p}(S\langle\langle T_\Sigma\rangle\rangle)$.
(B) If S is commutative and continuous, then $\mathrm{Rec}(\Sigma, S) = \mathrm{Eq}(S\langle\langle T_\Sigma\rangle\rangle)$.

Proof. For the proof of statement (A), let $\mathcal{A}$ be a wta. By the construction in the proof of Theorem 3.6, we can assume that $\mathcal{A}$ is a wta (Q, Σ, S, μ, F) with Boolean root weights and $F = \{q_f\}$; note that $(r_{\mathcal{A}}, \xi) = h_\mu(\xi)_{q_f}$ for every $\xi \in T_\Sigma$. Now construct the proper system E with equations $z_q = s_q$ for $q \in Q$ that is related to $\mathcal{A}$. Then $\bar{r} = (r_q \mid q \in Q)$ is a solution of E in $S\langle\langle T_\Sigma\rangle\rangle$, where $(r_q, \xi) = h_\mu(\xi)_q$. Since E is proper, it follows from Lemma 3.37 that this solution is the unique solution in the S-Σ-semimodule $S\langle\langle T_\Sigma\rangle\rangle$, which then is also its least solution. Thus, $r_{\mathcal{A}}$ is equational.

Conversely, let $r \in \mathrm{Eq_p}(S\langle\langle T_\Sigma\rangle\rangle)$. Then there is a proper system E of equations $z_i = s_i$ with $1 \le i \le n$ such that r is the, say, first component of the least solution $\bar{r} = (r_1, \ldots, r_n)$ of E. By Lemma 3.39, we can assume that E is simple. Clearly, we can construct a wta $\mathcal{A} = (Z_n, \Sigma, S, \mu, \{z_1\})$ that is related to E. Note that $(r_{\mathcal{A}}, \xi) = h_\mu(\xi)_{z_1}$ for every $\xi \in T_\Sigma$. Then $r_{\mathcal{A}} = r_1$ and $r \in \mathrm{Rec}(\Sigma, S)$.

For statement (B), it remains to show that every system E can be simulated by a proper system E' (here simulation means that every component of the least solution of E, if it exists, is a component of the solution of E'). This has been proved in [88], Theorem 3.2 (also cf. [58], Theorem 3.2) for commutative and continuous semirings. □

Solutions over Arbitrary S-Σ-Semimodules

Now let us turn to equational elements in an arbitrary S-Σ-semimodule $(V, +, 0, \theta)$. In order to guarantee the existence of least solutions, we require that $(V, +, 0)$ is a continuous monoid for the natural order $\sqsubseteq$. Thus, by [88], Theorem 2.3, $(V, \sqsubseteq)$ is a complete partially ordered set (for short: cpo). Since $\sqsubseteq$ can be extended to V^n (by componentwise comparison) and Φ_E is continuous on the cpo $(V^n, \sqsubseteq)$ by [21], Theorem 8 (also cf. [58], Proposition 2.6), i.e., preserves least upper bounds of ω-chains, it follows from Tarski's fixpoint theorem [139] that the least fixpoint of Φ_E exists and is the least upper bound of the ω-chain $\Phi_E^n(\perp)$ where $\perp = (0, \ldots, 0)$ is the least element of V^n.

Least solutions are preserved under homomorphisms. To see this, let $(V', \oplus, 0, \theta')$ be another S-Σ-semimodule. A mapping $h : V \to V'$ is an S-Σ-semimodule homomorphism if h is both a monoid homomorphism from $(V, +, 0)$ to $(V', \oplus, 0)$ and a Σ-algebra homomorphism from (V, θ) to (V', θ'), and moreover it satisfies the law $h(av) = ah(v)$ for every $a \in S$ and $v \in V$.

Lemma 3.41 ([21], Theorem 16). *Let $(V, +, 0, \theta)$ and $(V', \oplus, 0, \theta')$ be two continuous S-Σ-semimodules. Moreover, let $h : V \to V'$ be an S-Σ-semimodule homomorphism. If E is a system $z_i = s_i$ with $1 \le i \le n$ and $\bar{v} \in V^n$ is the least solution of E in V, then $h(\bar{v})$ is the least solution of E in V' (where h is extended to V^n componentwise).*

The following Mezei–Wright-like theorem is based on the idea that one can compute equational elements of V by first calculating "symbolically" in $S\langle\langle T_\Sigma\rangle\rangle$ and then evaluating the resulting tree series by a homomorphism from $S\langle\langle T_\Sigma\rangle\rangle$ to V. Let S be continuous. Then the S-Σ-semimodule $(S\langle\langle T_\Sigma\rangle\rangle, +, \widetilde{0}, \mathrm{top})$ is initial in the class of all continuous S-Σ-semimodules $(V, +, 0, \theta)$ (cf. [21], Theorem 4). Hence, there is exactly one S-Σ-semimodule homomorphism from $S\langle\langle T_\Sigma\rangle\rangle$ to V; this homomorphism we denote by h_V.

Theorem 3.42 (Mezei–Wright-like Theorem, cf. [21]). *Let S be commutative and continuous. Moreover, let $(V, +, 0, \theta)$ be a continuous S-Σ-semimodule. Then $\mathrm{Eq}(V) = h_V(\mathrm{Rec}(\Sigma, S))$.*

Proof. By Theorem 3.40(B), we only have to prove that $\mathrm{Eq}(V) = h_V(\mathrm{Eq}(S\langle\langle T_\Sigma\rangle\rangle))$. By Lemma 3.41, the least solution of a system E in V is the image under h_V of its least solution in $S\langle\langle T_\Sigma\rangle\rangle$. Hence, the same holds for the components of these solutions. □

There is another way of defining equational tree series which is based on so-called [IO]-substitution [25]; we note that in Sect. 4 we will define another variant called IO-substitution. For $s \in S\langle T_\Sigma(Z_n)\rangle$ and $\bar{r} = (r_1, \dots, r_n) \in S\langle\langle T_\Sigma\rangle\rangle^n$, the *[IO]-substitution of* $\bar{r}$ *into* s is the tree series $s \leftarrow_{[\mathrm{IO}]} \bar{r}$ defined by $s \leftarrow_{[\mathrm{IO}]} \bar{r} = \sum_{\zeta \in \mathrm{supp}(s)} (s, \zeta)(\zeta \leftarrow_{[\mathrm{IO}]} \bar{r})$ and $\zeta \leftarrow_{[\mathrm{IO}]} \bar{r} = \sum_{\zeta_1, \dots, \zeta_n \in T_\Sigma} (r_{i_1}, \zeta_{i_1}) \cdot \dots \cdot (r_{i_l}, \zeta_{i_l}).\zeta(\zeta_1, \dots, \zeta_n)$ for every $\zeta \in T_\Sigma(Z_n)$ where $z_{i_1}, \dots, z_{i_l}$, $1 \le i_1 < \dots < i_l \le n$, are all the variables which occur in ζ. An *[IO]-solution* of a system and the *class* $[\mathrm{IO}]\text{-}\mathrm{Eq}(S\langle\langle T_\Sigma\rangle\rangle)$ *of [IO]-equational tree series* are defined in the same way as above except that OI-substitution is replaced by [IO]-substitution.

Then [IO]-equational tree series can be characterized as the image of recognizable tree series under nondeleting tree homomorphisms. For this, we consider a family $h_{\Sigma,\Delta} = (h_\sigma \mid \sigma \in \Sigma)$ such that $h_\sigma \in T_\Delta(Z_k)$ if σ has rank k. The *tree homomorphism induced by* $h_{\Sigma,\Delta}$ is the mapping $h : T_\Sigma \to T_\Delta$ defined inductively by $h(\sigma(\xi_1, \dots, \xi_k)) = h_\sigma(h(\xi_1), \dots, h(\xi_k))$. A tree homomorphism is *nondeleting* if z_i occurs in h_σ for every $\sigma \in \Sigma^{(k)}$ and $1 \le i \le k$. Let $H^{\mathrm{nd}}_{\Sigma,\Delta}$ denote the class of all nondeleting tree homomorphisms induced by some family $\mathrm{h}_{\Sigma,\Delta}$.

We generalize a tree homomorphism $h : T_\Sigma \to T_\Delta$ to a mapping $h : S\langle\langle T_\Sigma\rangle\rangle \to S\langle\langle T_\Delta\rangle\rangle$ by defining $(h(r), \xi) = \sum_{\xi' \in T_\Sigma, h(\xi') = \xi} (r, \xi')$. In order to get the sum (which might have an infinite index set) well defined, we assume that S is complete.

Theorem 3.43 ([25], Theorem 16). *Let* S *be commutative and complete. Then* $[\mathrm{IO}]\text{-}\mathrm{Eq}(S\langle\langle T_\Delta\rangle\rangle) = H^{\mathrm{nd}}_{\Sigma,\Delta}(\mathrm{Rec}(\Sigma, S))$.

3.9 Rational Tree Series

Kleene's fundamental theorem on the equivalence between recognizable and rational string languages [81] has been extended to trees [135] and to tree series over commutative, complete, and continuous semirings [21, 88, 58], and over commutative semirings [41, 119, 120]. In [1], a characterization of weighted regular tree grammars in terms of rational tree series is sketched. We refer to [124] for a survey on rational formal power series over strings.

In order to keep the technical overhead of this chapter small, we will only show here some characteristic details of the equivalence proof along the approach of [41].

Throughout Sect. 3.9, we assume that S *is commutative.*

The set of *rational tree series expressions over* Σ *and* S, denoted by $\mathrm{RatExp}(\Sigma, S)$, is the smallest set R which satisfies conditions (1)–(5). For every $\eta \in \mathrm{RatExp}(\Sigma, S)$, we define $[\![\eta]\!] \in S\langle\langle T_\Sigma\rangle\rangle$ simultaneously.

1. For every $k \ge 0$, $\sigma \in \Sigma^{(k)}$, and $\eta_1, \dots, \eta_k \in R$, the expression $\sigma(\eta_1, \dots, \eta_k) \in R$ and $[\![\sigma(\eta_1, \dots, \eta_k)]\!] = \mathrm{top}_\sigma([\![\eta_1]\!], \dots, [\![\eta_k]\!])$.

2. For every $\eta \in R$ and $a \in S$, the expression $(a\eta) \in R$ and $[\![(a\eta)]\!] = a[\![\eta]\!]$.
3. For every $\eta_1, \eta_2 \in R$, the expression $(\eta_1 + \eta_2) \in R$ and $[\![(\eta_1 + \eta_2)]\!] = [\![\eta_1]\!] + [\![\eta_2]\!]$.
4. For every $\eta_1, \eta_2 \in R$ and $\alpha \in \Sigma^{(0)}$, the expression $(\eta_1 \circ_\alpha \eta_2) \in R$ and $[\![(\eta_1 \circ_\alpha \eta_2)]\!] = [\![\eta_1]\!] \circ_\alpha [\![\eta_2]\!]$.
5. For every $\eta \in R$ and $\alpha \in \Sigma^{(0)}$, the expression $(\eta_\alpha^*) \in R$ and $[\![(\eta_\alpha^*)]\!] = [\![\eta]\!]_\alpha^*$.

A tree series $r \in S\langle\!\langle T_\Sigma \rangle\!\rangle$ is a *rational tree series over Σ and S* if there is an $\eta \in \mathrm{RatExp}(\Sigma, S)$ such that $r = [\![\eta]\!]$. The *class of all rational tree series over Σ and S* is denoted by $\mathrm{Rat}(\Sigma, S)$. We say that a class $\mathcal{C} \subseteq S\langle\!\langle T_\Sigma \rangle\!\rangle$ is *closed under the rational operations* if it is closed under top-concatenation top_σ for every $\sigma \in \Sigma$, multiplication with coefficients in S, sum, α-concatenation and α-Kleene star for every $\alpha \in \Sigma^{(0)}$.

Obviously, every polynomial is a rational tree series (note that $\widetilde{0} = [\![0\alpha]\!]$ for any $\alpha \in \Sigma^{(0)}$; recall that we required in general that $\Sigma^{(0)} \neq \emptyset$). Thus, $\mathrm{Rat}(\Sigma, S)$ is the smallest subclass of $S\langle\!\langle T_\Sigma \rangle\!\rangle$ that contains $S\langle T_\Sigma \rangle$, and is closed under the rational operations.

Example 3.44. Consider $\Sigma = \{\sigma^{(2)}, \gamma^{(1)}, \alpha^{(0)}\}$ and again the tree series $\#_{\sigma(z,\alpha)}$ from page 322. We fix a $z \notin \Sigma$ and define the ranked alphabet $\Delta = \Sigma \cup \{z^{(0)}\}$. Then we define the rational expressions $\eta, \eta_1, \eta_2 \in \mathrm{RatExp}(\Delta, \mathsf{Nat})$ by

$$\begin{aligned}
\eta &= \eta_1 \circ_z \sigma(z, \alpha) \circ_z \eta_2, \\
\eta_1 &= \big(\gamma(z) + \sigma(\eta_2, z) + \sigma(z, \eta_2)\big)_z^*, \\
\eta_2 &= \big(\gamma(z) + \sigma(z, z)\big)_z^* \circ_z \alpha.
\end{aligned}$$

It is obvious that $[\![\eta_1]\!], [\![\eta_2]\!] \in \mathsf{Nat}\langle\!\langle T_\Delta \rangle\!\rangle$ with $[\![\eta_1]\!] = 1_{(\mathsf{Nat}, C_\Sigma)}$ and $[\![\eta_2]\!] = 1_{(\mathsf{Nat}, T_\Sigma)}$. Then $[\![\eta]\!] = \#_{\sigma(z,\alpha)}$.

In Example 3.44, we did not distinguish between the two tree series $[\![\eta]\!]$ and $[\![\eta]\!]|_{T_\Sigma}$. In general, we will not distinguish between a tree series $r \in S\langle\!\langle T_\Sigma \rangle\!\rangle$ and a tree series $r' \in S\langle\!\langle T_\Sigma(Q) \rangle\!\rangle$ for which $r = r'|_{T_\Sigma}$ and $\mathrm{supp}(r') \subseteq T_\Sigma$.

First, let us show why every recognizable tree series is rational. More precisely, we consider the wta $\mathcal{A} = (Q, \Sigma, S, \mu, F)$ with Boolean root weights and we will show that $r_\mathcal{A} \in \mathrm{Rat}(\Sigma \cup Q, S)$ where the states are assumed to be nullary symbols. (In fact, similar to the case of tree languages, we will need the states as substitution symbols in q-concatenations and q-Kleene stars of tree series for $q \in Q$.) The basic idea of the proof is the same as in Kleene's proof [81]: starting with the empty set, the set of permitted intermediate states is enlarged until it reaches the set of all states; at every level of this process, a tree series is defined as rational expression over the tree series of the previous level.

Formally, for every $P \subseteq Q$ and $q \in Q$, we define the tree series $r_\mathcal{A}(P, q) \in S\langle\!\langle T_\Sigma(Q) \rangle\!\rangle$ such that for every $\xi \in T_\Sigma(Q)$,

$$(r_\mathcal{A}(P, q), \xi) = \begin{cases} \sum_{\kappa \in R_\mathcal{A}^P(\xi, q)} \mathrm{wt}(\kappa) & \text{if } \xi \in T_\Sigma(Q) \setminus Q, \\ 0 & \text{if } \xi \in Q \end{cases}$$

where $R^P_{\mathcal{A}}(\xi, q)$ is the set of all those runs $\kappa \in R_{\mathcal{A}}(\xi)$ such that (i) $\kappa(\varepsilon) = q$, (ii) $\kappa(w) \in P$ for every $w \in \mathrm{pos}(\xi) \setminus (\mathrm{pos}_Q(\xi) \cup \{\varepsilon\})$, and (iii) $\kappa(w) = \xi(w)$ for every $w \in \mathrm{pos}_Q(\xi)$. The next lemma shows what happens when one state is added to the set P of inner states.

Lemma 3.45 ([41], Lemma 5.1). *Let $M = (Q, \Sigma, S, \mu, F)$ be a wta with Boolean root weights. Let $P \subseteq Q$ and $q \in Q$, and let $p \in Q \setminus P$. Then $r_{\mathcal{A}}(P \cup \{p\}, q) = r_{\mathcal{A}}(P, q) \circ_p r_{\mathcal{A}}(P, p)^*_p$.*

In the sequel, we will denote $\bigcup_{Q \text{ finite set}} \mathrm{Rat}(\Sigma \cup Q, S)$ by $\mathrm{Rat}(\Sigma \cup Q_\infty, S)$; similarly $\mathrm{Rec}(\Sigma \cup Q_\infty, S)$ is defined.

Theorem 3.46 ([41], Theorem 5.2). $\mathrm{Rec}(\Sigma, S) \subseteq \mathrm{Rat}(\Sigma \cup Q_\infty, S)$.

Proof. Let $r \in \mathrm{Rec}(\Sigma, S)$. By Theorem 3.6, we can assume that there is a wta $\mathcal{A} = (Q, \Sigma, S, \mu, F)$ with Boolean root weights such that $r_{\mathcal{A}} = r$. Let $Q = \{q_1, \ldots, q_n\}$. We prove that $r_{\mathcal{A}} \in \mathrm{Rat}(\Sigma \cup Q, S)$.

Since $\mathrm{supp}(r_{\mathcal{A}}(Q, q))$ may contain trees in which states occur, whereas this is not true for $\mathrm{supp}(r_{\mathcal{A}})$, we filter out from the tree series $r_{\mathcal{A}}(Q, q)$ trees in $T_\Sigma(Q) \setminus T_\Sigma$. Obviously, $r_{\mathcal{A}} = \sum_{q \in F} (\ldots (r_{\mathcal{A}}(Q, q) \circ_{q_1} \widetilde{0}) \circ_{q_2} \widetilde{0} \ldots) \circ_{q_n} \widetilde{0}$. Thus, it remains to show that $r_{\mathcal{A}}(Q, q) \in \mathrm{Rat}(\Sigma \cup Q, S)$.

For this, we prove the following statement by induction on the number of elements in P: for every $P \subseteq Q$ and $q \in Q$, the tree series $r_{\mathcal{A}}(P, q)$ is in $\mathrm{Rat}(\Sigma \cup Q, S)$. For the induction base, i.e., $P = \emptyset$, we can easily observe that

$$r_{\mathcal{A}}(\emptyset, q) = \sum_{\substack{k \geq 0,\ \sigma \in \Sigma^{(k)} \\ p_1, \ldots, p_k \in Q}} \mu_k(\sigma)_{p_1 \ldots p_k, q} . \sigma(p_1, \ldots, p_k),$$

which is a polynomial, and hence $r_{\mathcal{A}}(\emptyset, q)$ is rational. For the induction step, we assume that $r_{\mathcal{A}}(P, q)$ is rational for every $q \in Q$. Now let $p \in Q \setminus P$. Then it follows from Lemma 3.45 that also $r_{\mathcal{A}}(P \cup \{p\}, q)$ is rational because it is built up from rational tree series by rational operations. $\square$

Second, the inclusion $\mathrm{Rat}(\Sigma, S) \subseteq \mathrm{Rec}(\Sigma, S)$ follows from the fact that $\mathrm{Rec}(\Sigma, S)$ contains every polynomial in $S\langle\langle T_\Sigma \rangle\rangle$ and that $\mathrm{Rec}(\Sigma, S)$ is closed under the rational operations (cf. Theorem 3.8). Thus, we obtain the following Kleene theorem for recognizable tree series and commutative semirings.

Theorem 3.47 ([41], Theorem 7.1). $\mathrm{Rec}(\Sigma \cup Q_\infty, S) = \mathrm{Rat}(\Sigma \cup Q_\infty, S)$.

3.10 MSO-Definable Tree Series

Büchi's and Elgot's fundamental theorem [28, 46] shows the equivalence of recognizability and definability by means of formulas of monadic second order logic (MSO-logic) for the class of string languages. This result was extended to various other structures, including trees [135, 33] and unranked trees [117, 98, 99]. Then weighted MSO-logic was introduced in [37, 38], see also [39], and the equivalence between recognizability and definability of power series was proved. Most recently, this equivalence on the quantitative level has been extended to finite and infinite strings with discounting [42], trees [43], unranked trees [44], infinite trees [122], trace languages [115], picture languages [60], and texts and nested words [112, 113]. Here, we will report on Büchi–Elgot's theorem for recognizable tree series [43, 44] and we follow the approach of [37, 38], see also [39].

The main idea of [37, 38] for defining weighted MSO-logic is to consider formulas of MSO-logic in their negation normal form (i.e., all negation operators are moved down to the atoms) and then to allow elements of S to occur additionally as atomic formulas. Formally, the *set of all formulas of weighted MSO-logic over Σ and S on trees*, denoted by $\mathrm{MSO}(\Sigma, S)$, is defined to be the smallest set G such that:

(i) G contains all the *atomic formulas* a, $\mathrm{label}_\sigma(x)$, $\mathrm{edge}_i(x,y)$, $(x \sqsubseteq y)$, and $(x \in X)$, and the negations $\neg\mathrm{label}_\sigma(x)$, $\neg\mathrm{edge}_i(x,y)$, $\neg(x \sqsubseteq y)$, and $\neg(x \in X)$, and
(ii) if $\varphi, \psi \in G$, then also $\varphi \vee \psi$, $\varphi \wedge \psi$, $\exists x.\varphi$, $\forall x.\varphi$, $\exists X.\varphi$, $\forall X.\varphi \in G$,

where $a \in S$, x, y are first order variables, $\sigma \in \Sigma$, $1 \le i \le \max\{rk(\sigma) \mid \sigma \in \Sigma\}$ and X is a second order variable.

Next, we define the semantics of a formula $\varphi \in \mathrm{MSO}(\Sigma, S)$. We denote the set of free variables of φ by $\mathrm{Free}(\varphi)$. Let $\mathcal{V}$ be a finite set of variables containing $\mathrm{Free}(\varphi)$ and $\xi \in T_\Sigma$. A $(\mathcal{V}, \xi)$-assignment is a function ρ that maps the first order variables in $\mathcal{V}$ to elements of $\mathrm{pos}(\xi)$ and the second order variables in $\mathcal{V}$ to subsets of $\mathrm{pos}(\xi)$. We call a $(\mathrm{Free}(\varphi), \xi)$-assignment also simply an assignment for (φ, ξ), or a (φ, ξ)-assignment. We let the $\mathcal{V}$-semantics $[\![\varphi]\!]_\mathcal{V}$ of φ be the function which maps each pair $\zeta = (\xi, \rho)$ with $\xi \in T_\Sigma$ and $(\mathcal{V}, \xi)$-assignment ρ to the value $([\![\varphi]\!]_\mathcal{V}, \zeta) \in S$. We define this value inductively (over the structure of φ) as follows, where $\le_\xi$ denotes the linear order on $\mathrm{pos}(\xi)$ induced by the postorder tree walk on ξ:

$$
\begin{aligned}
([\![a]\!]_\mathcal{V}, \zeta) &= a,\\
([\![\mathrm{label}_\sigma(x)]\!]_\mathcal{V}, \zeta) &= 1 \text{ if } \xi(\rho(x)) = \sigma, \text{ and } 0 \text{ otherwise},\\
([\![\mathrm{edge}_i(x,y)]\!]_\mathcal{V}, \zeta) &= 1 \text{ if } \rho(y) = \rho(x)\, i, \text{ and } 0 \text{ otherwise},\\
([\![x \sqsubseteq y]\!]_\mathcal{V}, \zeta) &= 1 \text{ if } \rho(x) \le_\xi \rho(y), \text{ and } 0 \text{ otherwise},\\
([\![x \in X]\!]_\mathcal{V}, \zeta) &= 1 \text{ if } \rho(x) \in \rho(X), \text{ and } 0 \text{ otherwise},\\
([\![\neg\varphi]\!]_\mathcal{V}, \zeta) &= 1 \text{ if } ([\![\varphi]\!]_\mathcal{V}, \zeta) = 0, \text{ and } 0 \text{ if } ([\![\varphi]\!]_\mathcal{V}, \zeta) = 1\\
&\quad \big(\text{where } \varphi \text{ is of the form } \mathrm{label}_\sigma(x),\ \mathrm{edge}_i(x,y),\ x \sqsubseteq y, \text{ or } x \in X\big),\\
([\![\varphi \vee \psi]\!]_\mathcal{V}, \zeta) &= ([\![\varphi]\!]_\mathcal{V}, \zeta) + ([\![\psi]\!]_\mathcal{V}, \zeta),\\
([\![\varphi \wedge \psi]\!]_\mathcal{V}, \zeta) &= ([\![\varphi]\!]_\mathcal{V}, \zeta) \cdot ([\![\psi]\!]_\mathcal{V}, \zeta),\\
([\![\exists x.\varphi]\!]_\mathcal{V}, \zeta) &= \sum_{w \in \mathrm{pos}(\zeta)} ([\![\varphi]\!]_{\mathcal{V} \cup \{x\}}, \zeta[x \to w]),
\end{aligned}
$$

$$(\llbracket \forall x.\varphi \rrbracket_{\mathcal{V}}, \zeta) = \prod_{w \in \mathrm{pos}(\zeta)} (\llbracket \varphi \rrbracket_{\mathcal{V} \cup \{x\}}, \zeta[x \to w]),$$

$$(\llbracket \exists X.\varphi \rrbracket_{\mathcal{V}}, \zeta) = \sum_{I \subseteq \mathrm{pos}(\zeta)} (\llbracket \varphi \rrbracket_{\mathcal{V} \cup \{X\}}, \zeta[X \to I]),$$

$$(\llbracket \forall X.\varphi \rrbracket_{\mathcal{V}}, \zeta) = \prod_{I \subseteq \mathrm{pos}(\zeta)} (\llbracket \varphi \rrbracket_{\mathcal{V} \cup \{X\}}, \zeta[X \to I]).$$

The factors in the product over $\mathrm{pos}(\zeta)$ are ordered according to $\leq_\zeta$; moreover, for the product over subsets I of $\mathrm{pos}(\zeta)$, we employ the lexicographical linear order on the set $\{0,1\}^{\mathrm{pos}(\zeta)}$, where the elements of $\mathrm{pos}(\zeta)$ are ordered by $\leq_\zeta$.

We write $\llbracket \varphi \rrbracket$ for $\llbracket \varphi \rrbracket_{\mathrm{Free}(\varphi)}$. Now let $Y \subseteq \mathrm{MSO}(\Sigma, S)$. A tree series $r \in S\langle\langle T_\Sigma \rangle\rangle$ is Y*-definable* if there is a sentence $\varphi \in Y$ such that $S = \llbracket \varphi \rrbracket$. By [135, 33], $\mathrm{Rec}(\Sigma)$ is the class of all $\mathrm{MSO}(\Sigma, \mathbb{B})$-definable tree languages.

Example 3.48. We consider the ranked alphabet $\Sigma = \{\sigma^{(2)}, \gamma^{(1)}, \alpha^{(0)}\}$ and the formula $\varphi = \exists x.\mathrm{label}_\sigma(x) \wedge (\exists y.\mathrm{edge}_2(x,y) \wedge \mathrm{label}_\alpha(y))$ in $\mathrm{MSO}(\Sigma, S)$ for an arbitrary S. Clearly, for $S = \mathbb{B}$, the tree series $\llbracket \varphi \rrbracket$ maps $\xi \in T_\Sigma$ to 1 iff the pattern $\sigma(z, \alpha)$ occurs at least once in ξ. For $S = \mathsf{Nat}$, $\llbracket \varphi \rrbracket = \#_{\sigma(z,\alpha)}$, i.e., it computes the number of occurrences of $\sigma(z, \alpha)$ in a given tree ξ.

In order to obtain an equivalence between recognizability and definability, we have to restrict either the $\mathrm{MSO}(\Sigma, S)$-logic or the class of involved semirings because, e.g., the tree series $\llbracket \forall x.\forall y.2 \rrbracket$ over the semiring of natural numbers is not recognizable. For the restriction of the logic, we first recall the concept of unambiguous formulas (cf. [38], Definition 5.1). We denote by $\mathrm{MSO}^-(\Sigma, S)$ the fragment of $\mathrm{MSO}(\Sigma, S)$ obtained by not permitting atomic subformulas of the form a, where $a \in S$. Such formulas can be viewed as classical (unweighted) MSO-formulas over trees. Then roughly speaking, an unambiguous formula is a formula of $\mathrm{MSO}^-(\Sigma, S)$ in which the disjunction $\varphi \vee \psi$ is allowed only if, for every $\xi \in T_\Sigma$, there is no assignment ρ for $(\varphi \vee \psi, \xi)$ such that (ξ, ρ) satisfies both φ and ψ, and finally, for every first order and second order existential quantification $\exists x.\varphi$ and $\exists X.\varphi$, respectively, and every $\xi \in T_\Sigma$, there is at most one variable assignment to x and X, respectively, which fulfills φ. In fact, for every unambiguous formula φ, $\xi \in T_\Sigma$, and (φ, ξ)-assignment ρ, we have that $(\llbracket \varphi \rrbracket, (\xi, \rho)) = 1 \in S$ if (ξ, ρ) satisfies φ, and $(\llbracket \varphi \rrbracket, (\xi, \rho)) = 0$ otherwise (cf. [38], Proposition 5.2).

Note that the definition of unambiguity is based on the semantics of the formula. However, due to the algorithm in [39], Definition 4.3, for every $\varphi \in \mathrm{MSO}^-(\Sigma, S)$, there is a purely syntactic definition of formulas φ^+ and φ^- in $\mathrm{MSO}^-(\Sigma, S)$ such that:

- The formulas φ^+ and φ^- are unambiguous and $\mathrm{Free}(\varphi^+) = \mathrm{Free}(\varphi^-) = \mathrm{Free}(\varphi)$.
- For every $\xi \in T_\Sigma$ and (φ, ξ)-assignment ρ, (ξ, ρ) satisfies φ iff (ξ, ρ) satisfies φ^+ iff (ξ, ρ) does not satisfy φ^-.

We note that the atomic formula $(x \sqsubseteq y)$ is needed for this disambiguation. Moreover, for any $\varphi, \psi \in \mathrm{MSO}^{-}(\Sigma, S)$, we define the formulas $\varphi \overset{+}{\rightarrow} \psi$ and $\varphi \overset{+}{\leftrightarrow} \psi$ in $\mathrm{MSO}^{-}(\Sigma, S)$ as follows: $\varphi \overset{+}{\rightarrow} \psi = \varphi^{-} \vee (\varphi^{+} \wedge \psi^{+})$ and $\varphi \overset{+}{\leftrightarrow} \psi = (\varphi^{+} \wedge \psi^{+}) \vee (\varphi^{-} \wedge \psi^{-})$. Using this, we can define a formula to be *syntactically unambiguous* if it is of the form φ^{+}, φ^{-}, $\varphi \overset{+}{\rightarrow} \psi$, or $\varphi \overset{+}{\leftrightarrow} \psi$ for $\varphi, \psi \in \mathrm{MSO}^{-}(\Sigma, S)$. Clearly, each syntactically unambiguous formula is unambiguous.

The collection of *almost unambiguous* formulas is the smallest subset of $\mathrm{MSO}(\Sigma, S)$ which contains a for every $a \in S$ and all syntactically unambiguous formulas, and which is closed under disjunction and conjunction. In fact, for every almost unambiguous sentence φ, the tree series $[\![\varphi]\!]$ is a recognizable step function and vice versa, each recognizable step function is definable by some almost unambiguous sentence (cf. [44], Proposition 5.5).

Now we can define the fragment of $\mathrm{MSO}(\Sigma, S)$-logic which characterizes $\mathrm{Rec}(\Sigma, S)$. A formula $\varphi \in \mathrm{MSO}(\Sigma, S)$ is called *syntactically restricted* [39], if it satisfies the following conditions:

1. Whenever φ contains a conjunction $\psi \wedge \psi'$ as subformula but not in the scope of a universal first order quantifier, then each value of S occurring in ψ commutes with each value of S occurring in ψ'.
2. Whenever φ contains $\forall X.\psi$ as a subformula, then ψ is a syntactically unambiguous formula.
3. Whenever φ contains $\forall x.\psi$ as a subformula, then ψ is almost unambiguous.

We let $\mathrm{srMSO}(\Sigma, S)$ denote the set of all syntactically restricted formulas of $\mathrm{MSO}(\Sigma, S)$.

Theorem 3.49. *Let $r \in S\langle\!\langle T_\Sigma \rangle\!\rangle$. Then the following statements hold:*

(A) $r \in \mathrm{Rec}(\Sigma, S)$ if and only if r is $\mathrm{srMSO}(\Sigma, S)$-definable ([44], Theorem 7.2).

(B) Let S be commutative and locally finite. Then $r \in \mathrm{Rec}(\Sigma, S)$ if and only if r is $\mathrm{MSO}(\Sigma, S)$-definable ([43], Theorem 6.5).

We only indicate a sketch of the proof. For (A), given a wta $\mathcal{A}$ with $r = r_{\mathcal{A}}$, we can use the structure of $\mathcal{A}$ to explicitly write down an $\mathrm{srMSO}(\Sigma, S)$-sentence φ with $[\![\varphi]\!] = r_{\mathcal{A}}$. Conversely, given an $\mathrm{srMSO}(\Sigma, S)$-sentence ψ, we proceed by induction over the structure of ψ to construct a wta $\mathcal{A}$ such that $r_{\mathcal{A}} = [\![\psi]\!]$. For this, we encode pairs (ξ, ρ), where $\xi \in T_\Sigma$ and ρ is a $(\mathcal{V}, \xi)$-assignment, as trees over an extended alphabet $\Sigma_{\mathcal{V}} = \Sigma \times \{0,1\}^{\mathcal{V}}$ (as it is done also in the unweighted case). This enables us to view the function $[\![\varphi]\!]_{\mathcal{V}}$, where φ is an arbitrary $\mathrm{srMSO}(\Sigma, S)$-formula, as a formal tree series over S and $\Sigma_{\mathcal{V}}$ and we show that it is recognizable. When dealing with conjunctions and universal quantifications, we need the assumptions (1) and (2)–(3), respectively, on φ given above. For (B), there are alternative arguments exploiting that S is locally finite.

3.11 Other Models Related to Recognizable Tree Series

In the literature, several other concepts of recognizability of tree series were investigated: recognizability (a) by S-Σ-tree automata [21] where S is a commutative semiring, (b) by finite, polynomial tree automata over S [88, 58] where S is a commutative and continuous semiring, (c) by multilinear representations over fields [7], (d) by S-Σ-representations [19] where S is a field, (e) by polynomially-weighted tree automata [128], and (f) by wta over distributive multioperator monoids [90, 103, 62, 132] (which were already mentioned in Sect. 3.6). Here, we recall the models (a)–(d) and sketch how they are related to wta (for more details cf. [67]; for an informal comparison of model (a) with wta cf. [12]). Finally, we briefly discuss the concepts (e) and (f).

S-Σ-Tree Automata

In this subsection, we assume that S is commutative.

In [21], the recognizability of tree series by S-Σ-tree automata was defined. For the finite nonempty set $Q = \{q_1, \ldots, q_\kappa\}$, we consider the S-semimodule S^Q. Now let $k \geq 0$ and consider a mapping $\mu : Q^k \to S^Q$. A *multilinear extension of* μ is a mapping $\overline{\mu} : (S^Q)^k \to S^Q$ such that $\overline{\mu}$ is multilinear and for every $p_1, \ldots, p_k \in Q$ we have $\overline{\mu}(1_{p_1}, \ldots, 1_{p_k}) = \mu(p_1, \ldots, p_k)$ where 1_{p_i} is the p_i-unit vector in S^Q. It can easily be seen that such a multilinear extension of μ exists and is unique. In fact, it has the form

$$\overline{\mu}(v_1, \ldots, v_k)_q = \sum_{p_1, \ldots, p_k \in Q} (v_1)_{p_1} \cdot \cdots \cdot (v_k)_{p_k} \cdot \mu(p_1, \ldots, p_k)_q .$$

Thus, we speak about *the* multilinear extension of μ. Since scalar factors can be pulled out from arguments of $\overline{\mu}$ in different order, the definition of multilinear extension only makes sense if S is commutative.

The S-Σ-tree automaton [21] is the same as the wta of Definition 3.2 except that for every $\sigma^{(k)}$, $\mu_k(\sigma)$ is a function from Q^k to S^Q and $\mu_{\mathcal{A}}(\sigma)$ is the multilinear extension of $\mu_k(\sigma)$. Obviously, this leads to the same Σ-algebra $(S^Q, \mu_{\mathcal{A}})$. Thus, the S-Σ-tree automaton is just a reformulation of the wta.

Finite, Polynomial Tree Automata

In this subsection, we assume that S is commutative and continuous.

In [88] and [58], the following tree automaton model was defined. A *finite tree automaton (over S and Σ)* (for short: fta) is a tuple $\mathcal{A} = (Q, \mathcal{M}, I, F)$ where Q is a finite nonempty set (of *states*), $\mathcal{M} = (\mathcal{M}_k \mid k \geq 1)$ is a family of *transition matrices* $\mathcal{M}_k$ such that $\mathcal{M}_k \in S\langle\langle T_\Sigma(Z_k)\rangle\rangle^{Q \times Q^k}$ and for almost every k it holds that every entry of $\mathcal{M}_k$ is $\widetilde{0} \in S\langle\langle T_\Sigma(Z_k)\rangle\rangle$ (recall that $Z_k =$

$\{z_1, \ldots, z_k\})$, $I \in S\langle\langle T_\Sigma(Z_1)\rangle\rangle^Q$ is the *initial state vector*, and $F \in S\langle\langle T_\Sigma\rangle\rangle^Q$ is the *final state vector*.

Intuitively, an fta $\mathcal{A}$ produces a tree series in a top-down fashion. It starts with the tree series I_q for every $q \in Q$ and then repeatedly "unfolds" transition matrices; finally, it replaces the remaining occurrences of variables in Z by elements of F_p for appropriate $p \in Q$.

The semantics of an fta $\mathcal{A}$ is defined by means of the fixpoint of a mapping $\Phi_{\mathcal{A}}$, i.e., in a bottom-up fashion. As preparation, we define the substitution of matrices over tree series. Let $\mathcal{M}_k \in S\langle\langle T_\Sigma(Z_k)\rangle\rangle^{Q\times Q^k}$ be a transition matrix and $v_1, \ldots, v_k \in S\langle\langle T_\Sigma\rangle\rangle^Q$. Then we define the matrix $\mathcal{M}_k(v_1, \ldots, v_k) \in S\langle\langle T_\Sigma\rangle\rangle^Q$ of tree series for every $q \in Q$ by

$$\mathcal{M}_k(v_1, \ldots, v_k)_q = \sum_{q_1,\ldots,q_k \in Q} (\mathcal{M}_k)_{q,q_1\ldots q_k} \leftarrow_{\mathrm{OI}} \big((v_1)_{q_1}, \ldots, (v_k)_{q_k}\big).$$

Since S is continuous with $\sqsubseteq$ as partial order, $(S, \sqsubseteq)$ is a cpo. By extending $(S, \sqsubseteq)$ to $(S\langle\langle T_\Sigma\rangle\rangle, \sqsubseteq)$, and in its turn, extending $(S\langle\langle T_\Sigma\rangle\rangle, \sqsubseteq)$ to $(S\langle\langle T_\Sigma\rangle\rangle^Q, \sqsubseteq)$ componentwise, also $(S\langle\langle T_\Sigma\rangle\rangle^Q, \sqsubseteq)$ is a cpo. Moreover, the mapping $\Phi_{\mathcal{A}} : S\langle\langle T_\Sigma\rangle\rangle^Q \to S\langle\langle T_\Sigma\rangle\rangle^Q$ defined for every $v \in S\langle\langle T_\Sigma\rangle\rangle^Q$ by $\Phi_{\mathcal{A}}(v) = \sum_{k\geq 1} \mathcal{M}_k(v, \ldots, v) + F$ is continuous (cf. [58], page 228). Thus, by Tarski's fixpoint theorem, $\Phi_{\mathcal{A}}$ has a least fixpoint $\mathrm{fix}\,\Phi_{\mathcal{A}}$ and this is the least upper bound of the *approximation sequence* $(\Phi^n_{\mathcal{A}}(\bot) \mid n \geq 0)$ *of* M with $\bot_q = \widetilde{0}$ for every $q \in Q$, i.e., $\mathrm{fix}\,\Phi_{\mathcal{A}} = \mathsf{sup}\{\Phi^n_{\mathcal{A}}(\bot) \mid n \geq 0\}$. Then the *tree series recognized by* M is

$$r_{\mathcal{A}} = \sum_{q\in Q} \big(I_q \leftarrow_{\mathrm{OI}} \big((\mathrm{fix}\,\Phi_{\mathcal{A}})_q\big)\big).$$

In order to relate this notion of recognizability with the one induced by our wta, we have to restrict the fta. We call an fta $\mathcal{A} = (Q, \mathcal{M}, I, F)$ *polynomial* if for every $k \geq 1$ all the entries of $\mathcal{M}_k$ are in $S\langle T_\Sigma(Z_k)\rangle$; moreover, for every $q \in Q$, there is an $a \in S$ such that $I_q = a.z_1$; and finally, for every $q \in Q$, the entry F_q is in $S\langle T_\Sigma\rangle$. A tree series $r \in S\langle\langle T_\Sigma\rangle\rangle$ is *recognizable by a polynomial fta over* S if there is a polynomial fta $\mathcal{A} = (Q, \mathcal{M}, I, F)$ over S and Σ such that $r = r_{\mathcal{A}}$. Note that $r_{\mathcal{A}} = \sum_{q\in Q}(I_q, z_1)(\mathrm{fix}\,\Phi_{\mathcal{A}})_q$. The class of all tree series recognizable by some polynomial fta over S is denoted by Rec-FPTA(Σ, S).

Now we will compare the concept of recognizability by polynomial fta with that of wta. Since polynomial fta are close to systems of linear equations (as discussed in Sect. 3.8), we will use the equivalence of recognizable and equational (cf. Theorem 3.40). For every polynomial fta $\mathcal{A} = (Q, \mathcal{M}, I, F)$ over Σ and S, one easily associates the system E of equations $z_q = s_q$ for every $q \in Q$, with $s_q = F_q + \sum_{k\geq 1, q_1,\ldots,q_k \in Q} (\mathcal{M}_k)_{q,q_1\ldots q_k} \leftarrow_{\mathrm{OI}} (z_{q_1}, \ldots, z_{q_k})$. Obviously, $\Phi_E = \Phi_{\mathcal{A}}$. Thus, $(\mathrm{fix}\,\Phi_{\mathcal{A}})_q$ is a recognizable tree series for every $q \in Q$ (by Theorem 3.40). Since recognizability is preserved under scalar multiplication and summation (by Theorem 3.8), also $r_{\mathcal{A}}$ is recognizable. For the other direction, let $\mathcal{A} = (Q, \Sigma, S, \mu, \nu)$ be a wta. One can construct the polynomial fta $\mathcal{B} = (Q, \mathcal{M}, I, F)$ by defining $(\mathcal{M}_k)_{q,q_1\ldots q_k} = \sum_{\sigma\in\Sigma^{(k)}} \mu_k(\sigma)_{q_1\ldots q_k,q}.\sigma(z_1, \ldots, z_k)$,

$F_q = \sum_{\alpha \in \Sigma^{(0)}} \mu_0(\alpha)_{\varepsilon,q}.\alpha$, and $I_q = \nu_q.z_1$ for every $k \geq 1$ and $q, q_1, \ldots, q_k \in Q$. Then for every $\xi \in T_\Sigma$, $q \in Q$, $n \geq 0$, if $n \geq \mathrm{height}(\xi)$, then $((\Phi_{\mathcal{B}}^{n+1}(\perp))_q, \xi) = h_\mu(\xi)_q$; this can be proved by induction on n. Then, using the fact that $((\Phi_{\mathcal{B}}^{n+1}(\perp))_q, \xi) = (\mathrm{fix}\,(\Phi_{\mathcal{B}})_q, \xi)$ for every $n \geq \mathrm{height}(\xi)$, it follows that $r_{\mathcal{A}} = r_{\mathcal{B}}$.

Theorem 3.50 ([58], Corollary 3.6). $\mathrm{Rec}(\Sigma, S) = \text{Rec-FPTA}(\Sigma, S)$.

Multilinear Representations

In this subsection, we assume that S is a field.

In [7], the recognizability of tree series was defined in terms of multilinear mappings over finite-dimensional S-vector spaces in the following way. A *multilinear representation* of T_Σ is a tuple (V, μ, γ) where $(V, +, 0, \mu)$ is a non-trivial S-Σ-vector space and $\gamma : V \to S$ is a linear form. Then (V, μ, γ) defines the tree series $r \in S\langle\langle T_\Sigma \rangle\rangle$, where $(r, \xi) = \gamma(h_V(\xi))$ for every $\xi \in T_\Sigma$. A tree series $r \in S\langle\langle T_\Sigma \rangle\rangle$ is *recognizable by multilinear mappings over an S-vector space* if there is a multilinear representation (V, μ, γ) which defines r such that V is finite-dimensional. We denote the class of all tree series over Σ and S recognizable by multilinear mappings over an S-vector space by $\text{Rec-ML}(\Sigma, S)$.

Example 3.51 ([7], Example 4.1). We consider the tree series size_δ as defined on page 322. Then size_δ is recognizable by multilinear mappings over the $\mathbb{Q}$-vector space $(\mathbb{Q}^2, +, (0,0))$, where $(\mathbb{Q}, +, \cdot, 0, 1)$ is the field of rational numbers. Let $\{e_1, e_2\}$ be the basis of $\mathbb{Q}^2$ with $e_1 = (1,0)$ and $e_2 = (0,1)$. We define the multilinear representation $(\mathbb{Q}^2, \mu, \gamma)$ as follows. For every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, and $i_1, \ldots, i_k \in \{1, 2\}$ let

$$\mu(\sigma)(e_{i_1}, \ldots, e_{i_k}) = \begin{cases} e_1 + e_2 & \text{if } i_1 = \cdots = i_k = 1 \text{ and } \sigma = \delta, \\ e_1 & \text{if } i_1 = \cdots = i_k = 1 \text{ and } \sigma \neq \delta, \\ e_2 & \text{if } i_j = 2 \text{ for exactly one } 1 \leq j \leq k, \\ 0_2 & \text{otherwise.} \end{cases}$$

Since $\mu(\sigma)$ is multilinear, it suffices to define it on the base vectors. Finally, let $\gamma(e_1) = 0$ and $\gamma(e_2) = 1$ (since γ is a linear form, this definition extends to arbitrary vectors). Then we can prove by a straightforward induction that $h_{\mathbb{Q}^2}(\xi) = e_1 + (\mathrm{size}_\delta, \xi)e_2$ for every $\xi \in T_\Sigma$. Thus, we obtain $\gamma(h_{\mathbb{Q}^2}(\xi)) = (\mathrm{size}_\delta, \xi)$.

In fact, the concepts of recognizability by multilinear mappings over a finite-dimensional S-vector space V and recognizability by a wta over the field S coincide. This is based on the idea of viewing the state set Q of the wta as a basis of V; then V and S^Q are isomorphic S-vector spaces. Then roughly speaking, a multilinear representation (V, μ', γ) of T_Σ and a wta $\mathcal{A} = (Q, \Sigma, S, \mu, \nu)$ are related if μ is the restriction of μ' to base vectors (or, equivalently: μ' is the multilinear extension of μ) and $\nu(q) = \gamma(1_q)$ for every $q \in Q$ where 1_q is the q-unit vector. Obviously, $h_V = h_\mu$, and thus (V, μ', γ) defines the tree series $r_{\mathcal{A}}$. Since the relatedness implicitly contains a construction for both directions, we obtain the following characterization.

Theorem 3.52 ([12]; [67], Theorem 4.6). $\mathrm{Rec}(\Sigma, S) = \mathrm{Rec\text{-}ML}(\Sigma, S)$.

The question arises whether the concept of recognizability by multilinear mappings generalizes recognizability by bottom-up tree automata, i.e., whether $\mathrm{Rec}(\Sigma) \subseteq \mathrm{supp}(\mathrm{Rec\text{-}ML}(\Sigma, S))$ for every field S. The answer is yes, and this follows from Lemma 3.11, the fact that $\text{bud-Rec}(\Sigma, S) \subseteq \mathrm{Rec}(\Sigma, S)$, and Theorem 3.52. If we combine the involved two constructions, then we realize that every deterministic bottom-up tree automaton can be straightforwardly transformed into a multilinear representation. We note that the application of this transformation to a *non*deterministic bottom-up tree automaton would, in general, lead to a semantically different multilinear representation. This can be easily seen for the field $\mathbb{Z}_2 = \{0, 1\}$ where two accepting runs cancel each other because $1 + 1 = 0$.

S-Σ-Representations

Also, in this subsection, we assume that S is a field.

Next, we compare recognizability by wta over fields with the concept of representability [19]; this comparison is due to [23]. For this, let $n \geq 1$ and consider the *dual monoid* $(C_\Sigma, \diamond_z, z)$ of $(C_\Sigma, \cdot_z, z)$, where $\zeta \diamond_z \zeta' = \zeta' \cdot_z \zeta$ for every $\zeta, \zeta' \in C_\Sigma$. Then an *$S$-$\Sigma$-representation of dimension n* is a triple $R = (\varphi, \psi, \lambda)$ where $\varphi : C_\Sigma \to S^{n\times n}$ is a morphism from the monoid $(C_\Sigma, \diamond_z, z)$ to the monoid $(S^{n\times n}, \cdot, \mathcal{I}_n)$ of $n \times n$-matrices over S with the usual matrix multiplication and the unit matrix $\mathcal{I}_n$; moreover, $\psi : \Sigma^{(0)} \to S^{1\times n}$, and $\lambda \in S^{n\times 1}$ such that the following consistency condition is true: $\psi(\alpha) \cdot \varphi(\zeta) = \psi(\alpha') \cdot \varphi(\zeta')$ for every $\alpha, \alpha' \in \Sigma^{(0)}$ and $\zeta, \zeta' \in C_\Sigma$ such that $\zeta \cdot_z \alpha = \zeta' \cdot_z \alpha'$. The *tree series r_R represented by R* is defined by $(r_R, \xi) = \psi(\alpha) \cdot \varphi(\zeta) \cdot \lambda$ for every $\xi \in T_\Sigma$, $\zeta \in C_\Sigma$, and $\alpha \in \Sigma^{(0)}$ such that $\xi = \zeta \cdot_z \alpha$. A tree series r is *S-Σ-representable* if there is an S-Σ-representation R of some dimension n such that $r = r_R$. We denote the class of all S-Σ-representable tree series by $\mathrm{Rep}(\Sigma, S)$.

Example 3.53. Let us again consider the tree series $\#_{\sigma(z,\alpha)}$ as defined in the list of tree series on page 322, but now we use the field $\mathbb{Q}$ of rational numbers as the underlying semiring rather than Nat, i.e., $\#_{\sigma(z,\alpha)} \in \mathbb{Q}\langle\langle T_\Sigma \rangle\rangle$.

Now we consider the $\mathbb{Q}$-Σ-representation $R = (\varphi, \psi, \lambda)$ of dimension 3 where

$$\varphi(\zeta) = \begin{pmatrix} 1 & 0 & \#_{\sigma(z,\alpha)}(\zeta) \\ 0 & a(\zeta) & b(\zeta) \\ 0 & 0 & 1 \end{pmatrix}, \qquad \psi(\alpha) = \begin{pmatrix} 1 & 1 & 0 \end{pmatrix}, \qquad \lambda = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

and $a(\zeta) = 1$ if $\zeta = z$ and 0 otherwise, and $b(\zeta) = 1$ if z occurs at the second child of a position of ζ and 0 otherwise. Here, we have generalized $\#_{\sigma(z,\alpha)}$ in the obvious way such that it also works on contexts, e.g., $\#_{\sigma(z,\alpha)}(\sigma(z,\alpha)) = 1$. Since for every $\xi \in T_\Sigma$ and $\zeta \in C_\Sigma$ such that $\xi = \zeta \cdot_z \alpha$, we have that $(r_R, \xi) = \#_{\sigma(z,\alpha)}(\zeta) + b(\zeta)$, we obtain that $r_R = \#_{\sigma(z,\alpha)}$.

Theorem 3.54 ([23], Theorems 1 and 2). $\mathrm{Rec}(\Sigma, S) = \mathrm{Rep}(\Sigma, S)$.

Proof. Let $r \in \mathrm{Rec}(\Sigma, S)$. By Lemma 3.29, the S-vector space LQ_r (as defined on page 336) is finite-dimensional. Let $\{\eta_1^{-1} r, \ldots, \eta_n^{-1} r\}$ be a basis of LQ_r for some $\eta_1, \ldots, \eta_n \in T_\Sigma$. Then we define the S-Σ-representation $R = (\varphi, \psi, \lambda)$ of dimension n, where $\varphi : C_\Sigma \to S^{n\times n}$ is defined for every $\zeta \in C_\Sigma$ and $1 \le p, q \le n$ by $\varphi(\zeta)_{p,q} = ((\zeta \cdot_z \eta_p)^{-1} r)_q$ (note that the vector $(\zeta \cdot_z \eta_p)^{-1} r$ of LQ_r can be written as a linear combination of the base vectors; then $((\zeta \cdot_z \eta_p)^{-1} r)_q$ is the coefficient of $\eta_q^{-1} r$ in this representation). Moreover, for every $\alpha \in \Sigma^{(0)}$ and $1 \le p \le n$ we define $\psi(\alpha)_{1,p} = (\alpha^{-1} r)_p$ and $\lambda_{p,1} = (r, \eta_p)$.

For the proof of the consistency condition, we consider an arbitrary $\xi \in T_\Sigma$ such that $\xi = \zeta \cdot_z \alpha$ for some $\zeta \in C_\Sigma$ and $\alpha \in \Sigma^{(0)}$. Then using the concept of left quotient $\zeta^{-1} s$ also for $\zeta \in C_\Sigma$ and $s \in S\langle\langle C_\Sigma \rangle\rangle$, it is easy to compute that $\xi^{-1} r = \sum_{q=1}^{n} (\psi(\alpha) \cdot \varphi(\zeta))_q (\eta_q^{-1} r)$ (this follows from the facts (1) $(\zeta \cdot_z \xi)^{-1} r = \zeta^{-1}(\xi^{-1} r)$, and (2) the mapping $s \mapsto \zeta^{-1} s$ is linear). Since $\eta_1^{-1} r, \ldots, \eta_n^{-1} r$ is a basis of LQ_r, the coefficients of $\eta_q^{-1} r$ are uniquely determined; thus, if $\xi = \zeta' \cdot_z \alpha'$ is another decomposition of ξ, then we obtain that $\psi(\alpha) \cdot \varphi(\zeta) = \psi(\alpha') \cdot \varphi(\zeta')$. This proves the consistency condition. Using the above equation for $\xi^{-1} r$, it is straightforward to prove that $r = r_R$. Thus, r is S-Σ-representable.

Next, let $R = (\varphi, \psi, \lambda)$ be an S-Σ-representation of dimension n. We extend the mapping $R_{\varphi\psi} : T_\Sigma \to S^n$ defined by $R_{\varphi\psi}(\xi) = \psi(\alpha) \cdot \varphi(\zeta)$ for every $\xi \in T_\Sigma$ with $\xi = \zeta \cdot_z \alpha$, linearly to the mapping $R_{\varphi\psi} : S\langle T_\Sigma \rangle \to S^n$; thus $R_{\varphi\psi}$ is an S-Σ-semimodule homomorphism. Moreover, we extend the mapping $\varphi : C_\Sigma \to S^{n\times n}$ linearly to the mapping $\varphi : S\langle C_\Sigma \rangle \to S^{n\times n}$.

In the sequel, we will use that $R_{\varphi\psi}(r' \circ_z r) = R_{\varphi\psi}(r) \cdot \varphi(r')$ for every $r \in S\langle T_\Sigma \rangle$ and $r' \in S\langle C_\Sigma \rangle$, which can be seen as follows. First, for every $\zeta \in C_\Sigma$ and $\xi \in T_\Sigma$, we can compute $R_{\varphi\psi}(\zeta \cdot_z \xi) = R_{\varphi\psi}(\zeta \cdot_z \zeta' \cdot_z \alpha)$, where $\xi = \zeta' \cdot_z \alpha$ is an arbitrary decomposition. This is equal to $\psi(\alpha) \cdot \varphi(\zeta \cdot_z \zeta') = \psi(\alpha) \cdot \varphi(\zeta') \cdot \varphi(\zeta)$ because φ is a monoid homomorphism. Finally, this is equal to $R_{\varphi\psi}(\zeta' \cdot_z \alpha) \cdot \varphi(\zeta) = R_{\varphi\psi}(\xi) \cdot \varphi(\zeta)$. Secondly, we can generalize the equation proved in the first step by using the fact that $R_{\varphi\psi}$ and φ are linear.

Now we define the multilinear representation (V, μ, γ) where $V = \mathrm{im}(R_{\varphi\psi})$ is the image of $S\langle T_\Sigma \rangle$ under $R_{\varphi\psi}$. For every $k \ge 0$, $\sigma \in \Sigma^{(k)}$, we define the mapping $\mu(\sigma)$ by $\mu(\sigma)(v_1, \ldots, v_k) = R_{\varphi\psi}(\mathrm{top}_\sigma(s_1, \ldots, s_k))$ for every $v_1, \ldots, v_k \in \mathrm{im}(R_{\varphi\psi})$, where $s_i \in S\langle T_\Sigma \rangle$ is a preimage of v_i, i.e., $R_{\varphi\psi}(s_i) = v_i$; in fact, one can prove the independence of this definition from the chosen preimages by using the equation $R_{\varphi\psi}(r' \circ_z r) = R_{\varphi\psi}(r) \cdot \varphi(r')$. Note that μ is in fact multilinear because $R_{\varphi\psi}$ is a homomorphism. Moreover, we define $\gamma : \mathrm{im}(R_{\varphi\psi}) \to S$ by $\gamma(v) = v \cdot \lambda$.

We can see that $R_{\varphi\psi} : S\langle T_\Sigma\rangle \to \mathrm{im}(R_{\varphi\psi})$ is a surjective S-Σ-semimodule homomorphism. Since $(S\langle T_\Sigma\rangle, +, \widetilde{0}, \mathrm{top})$ is initial, we obtain that $R_{\varphi\psi} = h_V$. Then $(r_R, \xi) = R_{\varphi\psi}(\xi) \cdot \lambda = \gamma(h_V(\xi))$ for every $\xi \in T_\Sigma$, and hence r_R can be recognized by multilinear mappings over an S-vector space. By Theorem 3.52, it follows that $r \in \mathrm{Rec}(\Sigma, S)$. ☐

We note that there is also a direct way to construct from a given wta over a field S an S-Σ-representation. For instance, the $\mathbb{Q}$-Σ-representation R which we have considered in Example 3.53, can be obtained from the wta $\mathcal{A} = (Q, \Sigma, \mathbb{Q}, \mu, \nu)$ of Example 3.4 (with Nat already replaced by $\mathbb{Q}$) by identifying the states q, $\overline{\alpha}$, and f with the indices 1, 2, and 3, respectively, and defining $\varphi(\zeta)_{p,q} = h^p_\mu(\zeta)_q$, $\psi(\alpha)_q = h_\mu(\alpha)_q$, and $\lambda_q = \nu(q)$ for every $\zeta \in C_\Sigma$ and $p, q \in Q$.

Theorem 3.54 provides a characterization of recognizable S-Σ-tree series in terms of S-Σ-representations of finite dimension. This characterization can now be used to prove that if Σ contains at least one binary symbol and $\mathbb{N} \subseteq S$, then height $\notin \mathrm{Rec}(\Sigma, S)$ (at the same time, height $\in \mathrm{Rec}(\Sigma, \mathsf{Arct})$, cf. Example 3.3). The proof of this non-membership is done by contradiction, and it exploits the well-known Cayley–Hamilton theorem which states a property of the *characteristic polynomial* $\chi_\mathcal{M}$ *of a matrix* $\mathcal{M}$. For every $\mathcal{M} \in S^{n\times n}$, this polynomial is defined by $\chi_\mathcal{M}(x) = \det(\mathcal{M} - x\mathcal{I}_n)$, where $\det(\mathcal{N})$ is the determinant of a matrix $\mathcal{N}$. Clearly, the degree of $\chi_M(x)$ is n, and the coefficient of x^n is $(-1)^n$.

Theorem 3.55 ([97], Chap. XIV, Theorem 3.1). *Let* $n \geq 1$ *and* $\mathcal{M} \in S^{n\times n}$. *Then* $\chi_\mathcal{M}(\mathcal{M}) = 0$.

Now assuming that height $\in \mathrm{Rec}(\Sigma, S)$, we know that there is an S-Σ-representation $R = (\varphi, \psi, \lambda)$ such that height $= r_R$. In the next lemma, we apply Theorem 3.55 to matrices of the form $\varphi(\zeta)$ where $\zeta \in C_\Sigma$. We abbreviate $\zeta_1 \cdot_z \zeta_2^n$ (and $\zeta_1 \cdot_z \zeta_2^n \cdot_z \alpha$) by $\zeta_1\zeta_2^n$ ($\zeta_1\zeta_2^n\alpha$, respectively) for all contexts $\zeta_1, \zeta_2 \in C_\Sigma$ (for ζ^n see the beginning of Sect. 3.6).

Lemma 3.56 ([7], Proposition 9.3; [131]). *Let* $R = (\varphi, \psi, \lambda)$ *be an* S-Σ-*representation of dimension* n *and* $\zeta_2 \in C_\Sigma$. *Then there are* $a_1, \ldots, a_n \in S$ *such that for every* $\zeta_1 \in C_\Sigma$, $\alpha \in \Sigma^{(0)}$ *and* $i \geq 0$ *we have that*

$$(-1)^n \cdot \bigl(r_R, \zeta_1\zeta_2^{i+n}\alpha\bigr) + a_1 \cdot \bigl(r_R, \zeta_1\zeta_2^{i+n-1}\alpha\bigr) + \cdots + a_n \cdot \bigl(r_R, \zeta_1\zeta_2^{i}\alpha\bigr) = 0.$$

Proof. Let $\chi_{\varphi(\zeta_2)}(x) = (-1)^n x^n + a_1 x^{n-1} + \cdots + a_n$ for some $a_1, \ldots, a_n \in S$. By Theorem 3.55, we have $\chi_{\varphi(\zeta_2)}(\varphi(\zeta_2)) = 0$.

Now let $\zeta_1 \in C_\Sigma$, $\alpha \in \Sigma^{(0)}$, and $i \geq 0$. Then, by multiplying both sides of the equation $\chi_{\varphi(\zeta_2)}(\varphi(\zeta_2)) = 0$ with the vector $\psi(\alpha) \cdot \varphi(\zeta_2)^i$ from the left, and with the matrix $\varphi(\zeta_1)$ from the right, applying algebraic laws, and the fact that φ is a monoid morphism, we obtain the equation

$$\sum_{l=0}^{n} a_{n-l} \cdot \big(\psi(\alpha) \cdot \varphi\big(\zeta_1 \zeta_2^{i+l}\big)\big) = 0$$

where $a_0 = (-1)^n$. By multiplying the above equation with λ from the right and taking into account that $\psi(\alpha) \cdot \varphi(\zeta_1 \zeta_2^{i+l}) \cdot \lambda = (r_R, \zeta_1 \zeta_2^{i+l} \alpha)$, we obtain the statement of this lemma. $\square$

Before continuing with the proof that height $\notin \mathrm{Rec}(\Sigma, S)$, let us give an example of a characteristic polynomial and the application of Lemma 3.56.

Example 3.57. Recall the $\mathbb{Q}$-Σ-representation $R = (\varphi, \psi, \lambda)$ of dimension 3 from Example 3.53 and the matrix

$$\varphi(\zeta) = \begin{pmatrix} 1 & 0 & \#_{\sigma(z,\alpha)}(\zeta) \\ 0 & a(\zeta) & b(\zeta) \\ 0 & 0 & 1 \end{pmatrix}$$

for any $\zeta \in C_\Sigma$. Since $\chi_{\varphi(\zeta)}$ does not depend on $\varphi(\zeta)_{q,f}$ and $\varphi(\zeta)_{\overline{\alpha},f}$, we obtain that for every $\zeta \in C_\Sigma$:

- If $\zeta \neq z$, then $a(\zeta) = 0$ and $\chi_{\varphi(\zeta)}(x) = -x^3 + 2x^2 - x$.
- If $\zeta = z$, then $a(\zeta) = 1$ and $\chi_{\varphi(\zeta)}(x) = -x^3 + 3x^2 - 3x + 1$.

Thus, in particular, for every $\zeta \neq z$, $\zeta' \in C_\Sigma$, and $i \geq 0$, we obtain the following recurrence equation by applying Lemma 3.56:

$$\#_{\sigma(z,\alpha)}\big(\zeta' \zeta^{i+3} \alpha\big) = 2 \cdot \#_{\sigma(z,\alpha)}\big(\zeta' \zeta^{i+2} \alpha\big) - \#_{\sigma(z,\alpha)}(\zeta' \zeta^{i+1} \alpha).$$

Theorem 3.58 ([7], Example 9.2; [131]). *If Σ contains at least one binary symbol and $\mathbb{N} \subseteq S$, then* height $\notin \mathrm{Rec}(\Sigma, S)$.

Proof. Assume that height $\in \mathrm{Rec}(\Sigma, S)$. Then by Theorem 3.54, there is an S-Σ-representation $R = (\varphi, \psi, \lambda)$ of some dimension n with height $= r_R$.

Consider the tree $\zeta_2 = \sigma(z, \alpha)$ for some $\sigma \in \Sigma^{(2)}$. By Lemma 3.56, there are $a_1, \ldots, a_n \in S$ such that for every $\zeta_1 \in C_\Sigma$ and $\alpha \in \Sigma^{(0)}$ and $i \geq 0$ we have that

$$(-1)^n \cdot \big(\mathrm{height}, \zeta(i+n)\big) + a_1 \cdot \big(\mathrm{height}, \zeta(i+n-1)\big) + \cdots + a_n \cdot \big(\mathrm{height}, \zeta(i)\big) = 0 \tag{6}$$

where $\zeta(l)$ abbreviates $\zeta_1 \zeta_2^l \alpha$. Now let $\xi \in T_\Sigma$ be an arbitrary tree such that $(\mathrm{height}, \xi) = n$. Moreover, let $\zeta_1 = \sigma(z, \xi)$. Then

$$\big(\mathrm{height}, \zeta(j)\big) = 1 + \mathsf{max}(j, n) \tag{7}$$

for every $j \geq 0$. Using (6) with $i = 0$, and (7) with $j = n, \ldots, 0$ we obtain that $(-1)^n \cdot (1 + n) + a_1 \cdot (1 + n) + \cdots + a_n \cdot (1 + n) = 0$. On the other hand, using (6) with $i = 1$, and (7) with $j = n + 1, \ldots, 1$, we obtain that $(-1)^n \cdot (1 + (n + 1)) + a_1 \cdot (1 + n) + \cdots + a_n \cdot (1 + n) = 0$. We obtain $(-1)^n = 0$, which is a contradiction. $\square$

Polynomially-Weighted Tree Automata

In this subsection, we assume that S is commutative.

Let us now briefly discuss the concept (e) of recognizability (cf. the beginning of this subsection). This is the recognizability by polynomially-weighted tree automata, which were defined in [128]. Such a tree automaton uses a polynomial over S to compute the weight of a transition at a node by applying it to the weights of the subtrees of that node. In fact, polynomially-weighted tree automata are strictly more powerful than wta, cf. [67], Theorem 7.2 and Theorem 7.5. Moreover, it is decidable whether a polynomially-weighted tree automaton $\mathcal{A} = (Q, \Sigma, S, \mu, \nu)$ is bounded, where S can be any of the arctic semiring, the tropical semiring, and the semiring of finite subsets of $\mathbb{N}$, i.e., whether there is an $a \in S$ such that for every $\xi \in T_\Sigma$ and $q \in Q$ we have that $h_\mu(\xi)_q \sqsubseteq a$, cf. [128]. This result is generalized in [15] to finitely factorizing, monotonic, and naturally ordered semirings.

Wta over Distributive Multioperator Monoids

Finally, we consider the concept (f) of recognizability. In [90], the semiring was generalized to the distributive multioperator monoid (for short: DM-monoid) and wta over DM-monoids were introduced. A DM-monoid $(S, +, 0, \Omega)$ consists of a monoid $(S, +, 0)$ which is equipped with an Ω-algebraic structure, where the operations of Ω distribute over $+$. Such wta generalize polynomially-weighted tree automata in the sense that the weight of a transition on $\sigma \in \Sigma^{(k)}$ is a finite sum of Ω-polynomials; an Ω-polynomial is inductively built up from variables $z_1, \ldots, z_k$, elements of S, and operations of Ω. In other words, the weight is taken from $S\langle T_\Omega(Z_k)\rangle$. In [103, 62, 132], simple wta over DM-monoids have been investigated; there every transition is weighted by a single k-ary operation taken from Ω. In fact, simple (and hence, also arbitrary) wta over DM-monoids are strictly more powerful than polynomially-weighted tree automata, cf. [67], Theorems 8.6 and 8.9.

3.12 Further Results

There are further results on recognizable tree series which we did not address in the previous sections. Here, we list (some of) them in a very rough form.

In [62], Kleene's theorem has been proved for simple wta over distributive multioperator monoids and, as a consequence, for wta over arbitrary (i.e., not necessarily commutative) semirings. The latter result generalizes Theorem 3.47. Moreover, in [92, 95], Kleene's theorem has been proved for sorted algebras. In [10], a general Kleene-type theorem has been proved which is applicable to all grove theories that are Conway theories.

In [89], the concept of full abstract family of tree series, for short: AFT, has been defined. This is a family of tree series which contains $\widetilde{0}$ and is closed under sum, top-concatenation, and least solution of equational systems, and additionally under linear nondeleting recognizable tree transductions. In fact, $\mathrm{Rec}(\Sigma, S)$ is an AFT if S is a commutative and continuous semiring, cf. [89], Theorem 3.5.

In [36], it has been shown how aa-deterministically recognizable tree series over a semifield can be learned by a minimal adequate teacher. The aa stands for *all accepting* and means that the root weight vector of the used wta maps every state to 1. A minimal adequate teacher [2] answers coefficient queries and equivalence queries faithfully. In [108], this result has been extended to arbitrary tree series in bud-$\mathrm{Rec}(\Sigma, S)$ where S is a commutative semifield. We refer to [75] for an investigation about learning nondeterministic recognizable tree series over a field.

In [77], forward and backward bisimulation minimization algorithms for weighted tree automata have been investigated. For the origin of forward and backward bisimulation on weighted string automata, we refer to [27].

In [59], fuzzy tree automata are investigated. Such automata can be considered as weighted tree automata over a completely distributive lattice. In that paper, Kleene's theorem and the equivalence between equational and rational fuzzy sets over an arbitrary algebra is proved by using the theory of fixpoints and μ-clones of monotonic functions over a complete lattice.

In [9], recognizable tree series are studied in a coalgebraic way and several representation theorems are proved.

There also exist weighted pushdown tree automata (for the string case cf. [121]). In [94], algebraic tree series have been characterized by (weighted) pushdown tree automata; this generalizes the result of [74] from the unweighted to the weighted case. In [22], algebraic (or: context-free) tree series have been characterized as the closure of polynomials under second-order substitution of tree series and iteration; moreover, they have been compared with recursive program schemes [30, 31]. In [93] Corollary 3.6, it is proved that the class of algebraic tree series is closed under linear and nondeleting algebraic tree transductions.

4 IO-Substitution and Tree Series Transformations

As a preparation for Sect. 5 on weighted tree transducers, we recall the concept and the most important properties of the IO-substitution of tree series as well as the concept of tree series transformation and their composition.

Besides Z, we use a further set $X = \{x_1, x_2, \ldots\}$ of variables and we define $X_k = \{x_1, \ldots, x_k\}$ for every $k \geq 0$. For a finite set Q and a set $U \subseteq X$, we define $Q(U) = \{q(x) \mid q \in Q, x \in U\}$. We write just $Q(U)^*$ for $(Q(U))^*$. For every $w \in Q(X)^*$ and $x \in X$, we denote by $|w|_x$ the number of occurrences of x in w; thus $|w| = \sum_{x \in X} |w|_x$; e.g., $|q(x_1)p(x_2)p(x_1)|_{x_1} = 2$ and $|q(x_1)p(x_2)p(x_1)| = 3$. Hence, $|w|$ coincides with the usual definition of the length of a string w provided we consider $Q(U)$ as an alphabet. We say that

$w \in Q(U)^*$ is *linear in* U (resp., *nondeleting in* U), if every $x \in U$ occurs at most once (resp., at least once) in w. Moreover, we use the notation $|\xi|_z$ for $\xi \in T_\Delta(Z)$ and $z \in Z$, and the notions of linearity and nondeletion in $U \subseteq Z$ also for $\xi \in T_\Delta(U)$, accordingly.

For $r \in S\langle\langle T_\Delta(Z)\rangle\rangle$, we define $\mathrm{var}(r) = \bigcup_{\xi \in \mathrm{supp}(r)} \mathrm{var}(\xi)$, with $\mathrm{var}(\xi) = \{z \in Z \mid |\xi|_z > 0\}$. A tree series $r \in S\langle\langle T_\Delta(U)\rangle\rangle$, with $U \subseteq Z$, is *linear* (resp., *nondeleting*) in U, if every $\xi \in \mathrm{supp}(r)$ is linear (resp., nondeleting) in U.

Next, we extend tree substitution (cf. the end of Sect. 2.2) to IO-substitution of polynomial tree series. Certain statements that we need concerning this IO-substitution, e.g., property P5 below, only hold for commutative S. Since we do not want to monitor all the time whether we need commutativity or not and for the sake of succinctness, we make the following convention.

In the rest of Sect. 4, we assume that S is commutative.

Let $r \in S\langle T_\Delta(Z)\rangle$, $I \subseteq \mathbb{N}$ finite, and $(s_i \mid i \in I)$ a family of tree series $s_i \in S\langle T_\Delta(H)\rangle$ for some set H. The *IO-substitution* of tree series $(s_i \mid i \in I)$ into r, denoted by $r \leftarrow_{\mathrm{IO}} (s_i \mid i \in I)$, is defined by

$$r \leftarrow_{\mathrm{IO}} (s_i \mid i \in I) = \sum_{\substack{\xi \in T_\Delta(Z),\\ (\forall i \in I): \zeta_i \in T_\Delta(H)}} \left((r,\xi) \cdot \prod_{i \in I} (s_i,\zeta_i)\right).\xi(\zeta_i \mid i \in I).$$

Note that the above sum is (locally) finite because there are finitely many choices of ξ and ζ_i, $i \in I$, such that all coefficients (r,ξ) and (s_i,ζ_i), $i \in I$, are not 0. In case $I = \{1,\ldots,n\}$, we write $r \leftarrow_{\mathrm{IO}} (s_1,\ldots,s_n)$.

In the following, we summarize some properties of IO-substitution. The notations r, I, and $(s_i \mid i \in I)$ stand for the same as above.

P0 *Empty substitution*: If $I = \emptyset$, then we have $r \leftarrow_{\mathrm{IO}} (s_i \mid i \in I) = r$.

P1 *Missing variables* ([53], Observation 2.6a): If a variable z_j with $j \in I$ does not occur in a tree $\xi \in T_\Delta(Z)$, then the tree obtained by the substitution $\xi(\zeta_i \mid i \in I)$ does not depend on ζ_j. Still, the value (s_j,ζ_j) contributes to the coefficient of $\xi(\zeta_i \mid i \in I)$ in the tree series $r \leftarrow_{\mathrm{IO}} (s_i \mid i \in I)$ for every choice of ζ_j, hence that coefficient is the result of an infinite summation in S.

P2 *Dropping an index* ([104], Observation 3): For every $j \in I$ such that $z_j \notin \mathrm{var}(r)$ and $s_j = 1.\zeta$ for some $\zeta \in T_\Delta(H)$, we have $r \leftarrow_{\mathrm{IO}} (s_i \mid i \in I) = r \leftarrow_{\mathrm{IO}} (s_i \mid i \in I \setminus \{j\})$.

P3 *Zero propagation* ([53], Observation 2.6b): If $r = \widetilde{0}$ or $s_i = \widetilde{0}$ for some $i \in I$, then $r \leftarrow_{\mathrm{IO}} (s_i \mid i \in I) = \widetilde{0}$.

P4 *Preserving polynomials* ([53], Proposition 2.7): The tree series $r \leftarrow_{\mathrm{IO}} (s_i \mid i \in I)$ is polynomial.

P5 *Linearity in coefficients* ([53], Proposition 2.8): For every $a \in S$ and family $(a_i \mid i \in I)$ of elements of S, we have $ar \leftarrow_{\mathrm{IO}} (a_i s_i \mid i \in I) = \big(a \cdot \prod_{i \in I} a_i\big)\big(r \leftarrow_{\mathrm{IO}} (s_i \mid i \in I)\big)$.

P6 *Linearity in variables*: Let $r \in S\langle T_\Delta(Z_n)\rangle$ for some $n \geq 0$ and $r_1, \dots, r_n \in S\langle T_\Delta(Z)\rangle$ such that $\mathrm{var}(r_i) \cap \mathrm{var}(r_j) = \emptyset$ for $1 \leq i \neq j \leq n$. If r is linear (and nondeleting) in Z_n and every r_i is linear (and nondeleting) in $\mathrm{var}(r_i)$, then the tree series $r \leftarrow_{\mathrm{IO}} (r_1, \dots, r_n)$ is linear (and nondeleting) in $\bigcup_{i=1}^{n} \mathrm{var}(r_i)$.

P7 *Distributivity* ([53], Proposition 2.9): Let $J \subseteq \mathbb{N}$ and $J_i \subseteq \mathbb{N}$ for every $i \in I$ be further finite sets, and furthermore $(r_j \mid j \in J)$ and $(s_{j_i} \mid j_i \in J_i)$ families of tree series in $S\langle T_\Delta(Z)\rangle$ and $S\langle T_\Delta(H)\rangle$, respectively. Then

$$\Big(\sum_{j \in J} r_j\Big) \leftarrow_{\mathrm{IO}} \Big(\sum_{j_i \in J_i} s_{j_i} \mid i \in I\Big) = \sum_{\substack{j \in J, \\ \forall i \in I: j_i \in J_i}} r_j \leftarrow_{\mathrm{IO}} (s_{j_i} \mid i \in I).$$

P8 *Weak associativity* ([104], Corollary 7): Let $J \subseteq \mathbb{N}$ be finite and assume that $\mathrm{var}(r) \subseteq \{z_j \mid j \in J\}$. Moreover, let $(r_j \mid j \in J)$ be a family with $r_j \in S\langle T_\Delta(Z)\rangle$ and $(I_j \mid j \in J)$ be a partition of I such that $\mathrm{var}(r_j) \subseteq \{z_{i_j} \mid i_j \in I_j\}$ for every $j \in J$. Then

$$\begin{aligned}&\big(r \leftarrow_{\mathrm{IO}} (r_j \mid j \in J)\big) \leftarrow_{\mathrm{IO}} (s_i \mid i \in I)\\ &\quad = r \leftarrow_{\mathrm{IO}} \big(r_j \leftarrow_{\mathrm{IO}} (s_i \mid i \in I_j) \mid j \in J\big).\end{aligned}$$

A mapping $\tau : T_\Sigma \to \mathcal{P}_{\mathrm{fin}}(T_\Delta)$ is called a *tree transformation*, where $\mathcal{P}_{\mathrm{fin}}(T_\Delta)$ denotes the set of finite subsets of T_Δ. The *composition of the tree transformations* τ *and* $\tau' : T_\Delta \to \mathcal{P}_{\mathrm{fin}}(T_\Gamma)$ is the tree transformation $\tau;\tau' : T_\Sigma \to \mathcal{P}_{\mathrm{fin}}(T_\Gamma)$ defined by $(\tau;\tau')(\xi) = \bigcup_{\eta \in \tau(\xi)} \tau'(\eta)$. A mapping $\tau : T_\Sigma \to S\langle T_\Delta\rangle$ is a *(tree to) tree series transformation (over* S*)*. Then τ extends to a mapping of type $S\langle T_\Sigma\rangle \to S\langle T_\Delta\rangle$ by letting $\tau(r) = \sum_{\xi \in T_\Sigma} (r,\xi)\tau(\xi)$ for every $r \in S\langle T_\Sigma\rangle$. Also, for a finite set Q, a mapping $\tau : T_\Sigma \to S\langle T_\Delta\rangle^Q$ extends to a mapping of type $S\langle T_\Sigma\rangle \to S\langle T_\Delta\rangle^Q$ by letting $\tau(r)_q = \sum_{\xi \in T_\Sigma} (r,\xi)\tau(\xi)_q$ for every $r \in S\langle T_\Sigma\rangle$ and $q \in Q$. The *composition of the tree series transformations* $\tau : T_\Sigma \to S\langle T_\Delta\rangle$ *and* $\tau' : T_\Delta \to S\langle T_\Gamma\rangle$ is the tree series transformation $\tau;\tau' : T_\Sigma \to S\langle T_\Gamma\rangle$ defined by $(\tau;\tau')(\xi) = \tau'(\tau(\xi))$ for every $\xi \in T_\Sigma$. Then we extend composition to classes of tree series transformations in the following way. Let $\mathrm{C}(S)$ and $\mathrm{D}(S)$ be classes of polynomial tree series transformations over S. The composition of $\mathrm{C}(S)$ and $\mathrm{D}(S)$ is the class $\mathrm{C}(S);\mathrm{D}(S) = \{\tau;\tau' \mid$ there are ranked alphabets $\Sigma, \Delta,$ and Γ such that $\tau : T_\Sigma \to S\langle T_\Delta\rangle, \tau' : T_\Delta \to S\langle T_\Gamma\rangle, \tau \in \mathrm{C}(S),$ and $\tau' \in \mathrm{D}(S)\}$.

A tree series transformation $\tau : T_\Sigma \to S\langle T_\Sigma\rangle$ is called a *weighted identity* if, for every $\xi \in T_\Sigma$, we have $\tau(\xi) = a.\xi$ for some $a \in S$. The particular weighted identity $\iota : T_\Sigma \to S\langle T_\Sigma\rangle$, defined by $\iota(\xi) = 1.\xi$ for every $\xi \in T_\Sigma$, is called *identity (tree series transformation)*. It should be clear that $\iota;\tau = \tau$ for every $\tau : T_\Sigma \to S\langle T_\Delta\rangle$ and $\tau;\iota = \tau$ for every $\tau : T_\Gamma \to S\langle T_\Sigma\rangle$.

5 Weighted Tree Transducers

5.1 Tree Transducers

Tree transducers generalize finite-state tree automata in the way that, besides processing an input tree, they produce output trees. The classical generalizations are the *bottom-up tree transducers* [134, 48], the *top-down tree transducers* [123, 133, 48], and the *top-down tree transducers with regular look-ahead* [49]; their names come from the direction in which they process the input tree, where we assume that trees grow downward. Here, we recall their definition, however, we start with the definition of *generalized finite-state tree transducers* (for short: gfst) [48] which generalize all of the three classical tree transducers.

A gfst is a system $\mathcal{M} = (Q, \Sigma, \Delta, R, F)$, where Q is a finite set (of *states*) with $Q \cap (\Sigma \cup \Delta) = \emptyset$, R is a finite set of *rules*, and $F \subseteq Q$ is a set of *distinguished states*. We call Σ and Δ the *input* and the *output ranked alphabet*, respectively. Each rule in R has the form $(q, k, l, \sigma \to \zeta, \varphi)$, where $q \in Q$, $k \geq 0$, $l \geq 0$, $\sigma \in \Sigma^{(k)}$, $\zeta \in T_\Delta(Z_l)$, and $\varphi : Z_l \to Q(X_k)$ such that if $k = 0$, then $l = 0$. Such a rule can be visualized as

$$q\big(\sigma(x_1, \ldots, x_k)\big) \to \zeta, \langle q_1(x_{i_1}) \ldots q_l(x_{i_l})\rangle, \tag{8}$$

where $\varphi(z_j) = q_j(x_{i_j})$ for every $1 \leq j \leq l$, and it can be interpreted as follows. A q-translation of a Σ-tree rooted by the k-ary symbol σ is a Δ-tree which is computed in the way that a q_j-translation of the i_jth descendant of σ is substituted for every $1 \leq j \leq l$ and for every occurrence of z_j in ζ (if any). Of course, there may be several q-translations of such a σ-rooted tree. For a gfst $\mathcal{M}$, we consider the Σ-algebra $(\mathcal{P}_{\text{fin}}(T_\Delta)^Q, \mu_\mathcal{M})$ where, for every $k \geq 0$ and $\sigma \in \Sigma^{(k)}$, the k-ary operation $\mu_\mathcal{M}(\sigma) : \mathcal{P}_{\text{fin}}(T_\Delta)^Q \times \cdots \times \mathcal{P}_{\text{fin}}(T_\Delta)^Q \to \mathcal{P}_{\text{fin}}(T_\Delta)^Q$ is defined by

$$\mu_\mathcal{M}(\sigma)(v_1, \ldots, v_k)_q = \bigcup_{\substack{\text{for every rule}\\ \text{of the form (8)}}} \{\zeta(\zeta_1, \ldots, \zeta_l) \mid \zeta_j \in (v_{i_j})_{q_j},\ 1 \leq j \leq l\}.$$

Let us denote the unique Σ-algebra homomorphism from T_Σ to $\mathcal{P}_{\text{fin}}(T_\Delta)^Q$ by $h_\mathcal{M}$. Then the *tree transformation* $\tau_\mathcal{M} : T_\Sigma \to \mathcal{P}_{\text{fin}}(T_\Delta)$ *computed by* $\mathcal{M}$ is defined as

$$\tau_\mathcal{M}(\xi) = \bigcup_{q \in F} h_\mathcal{M}(\xi)_q$$

for every $\xi \in T_\Sigma$. Let GFST denote the class of all tree transformations computed by gfst.

Top-down tree transducers, top-down tree transducers with regular look-ahead, and bottom-up tree transducers can be derived from gfst as follows, cf. [48, 49]. The gfst $\mathcal{M}$ is a top-down tree transducer (resp., with regular look-ahead) if, for every rule (8), the tree ζ is linear and nondeleting (resp., linear) in Z_l. For the sake of simplicity, we consider only top-down tree transducers. Then the rule (8) is written as $q(\sigma(x_1,\dots,x_k)) \to \overline{\zeta}$, where $\overline{\zeta} = \zeta(q_1(x_{i_1}),\dots,q_l(x_{i_l}))$. Moreover, the gfst $\mathcal{M}$ is a bottom-up tree transducer if, for every rule (8), we have $k = l$ and $i_1 = 1,\dots,i_k = k$. Then the rule (8) is written as $\sigma(q_1(x_1),\dots,q_k(x_k)) \to q(\zeta(x_1,\dots,x_k))$. This allows to consider both top-down and bottom-up tree transducers as special term rewrite systems [82, 3] and to define their semantics, i.e., the computed tree transformation, in terms of term rewriting in the standard way, cf. [123, 48, 49]. See [48] for the exact definition of the term rewrite semantics of top-down and bottom-up tree transducers, and cf. [48], Lemmas 5.5 and 5.6 for the equivalence of the initial algebra semantics (as defined above) and the term rewrite semantics. We denote the classes of all tree transformations computed by top-down tree transducers and bottom-up tree transducers by TOP and BOT, respectively.

5.2 The Basic Model

Tree transducers can be generalized to weighted tree transducers over a semiring in a similar way as bottom-up tree automata were generalized to wta (cf. Sect. 3.2). The idea behind this is to represent a tree transformation $\tau : T_\Sigma \to \mathcal{P}_{\mathrm{fin}}(T_\Delta)$ as a tree series transformation $\tau : T_\Sigma \to \mathbb{B}\langle T_\Delta\rangle$ over the Boolean semiring $\mathbb{B}$ and then to generalize from $\mathbb{B}$ to an arbitrary semiring S.

For the first step, consider the system $\mathcal{M} = (Q,\Sigma,\Delta,\mathbb{B},\mu,F)$, called a *weighted tree transducer over* $\mathbb{B}$, where Q, Σ, Δ, and F are as for a gfst, while $\mu = (\mu_k \mid k \geq 0)$ is a family of mappings

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

such that $\mu_k(\sigma)_{w,q} \neq \widetilde{0}$ only for finitely many $(w,q) \in Q(X_k)^* \times Q$. Moreover, $\mu_k(\sigma)_{w,q} \in \mathbb{B}\langle T_\Delta(Z_l)\rangle$ with $l = |w|$, for every $(w,q) \in Q(X_k)^* \times Q$. Then we consider the Σ-algebra $(\mathbb{B}\langle T_\Delta\rangle^Q, \mu_{\mathcal{M}})$ where for every $k \geq 0$ and $\sigma \in \Sigma^{(k)}$, the k-ary operation $\mu_{\mathcal{M}}(\sigma) : \mathbb{B}\langle T_\Delta\rangle^Q \times \dots \times \mathbb{B}\langle T_\Delta\rangle^Q \to \mathbb{B}\langle T_\Delta\rangle^Q$ is defined as follows. For every $q \in Q$ and $v_1,\dots,v_k \in \mathbb{B}\langle T_\Delta\rangle^Q$, we have

$$\mu_{\mathcal{M}}(\sigma)(v_1,\dots,v_k)_q = \bigvee_{\substack{w \in Q(X_k)^*,\\ w = q_1(x_{i_1})\dots q_l(x_{i_l})}} \mu_k(\sigma)_{w,q} \leftarrow_{\mathrm{IO}} \big((v_{i_1})_{q_1},\dots,(v_{i_l})_{q_l}\big).$$

Let us denote the unique Σ-algebra homomorphism from T_Σ to $\mathbb{B}\langle T_\Delta\rangle^Q$ by h_μ. Now the tree series transformation $\tau_{\mathcal{M}} : T_\Sigma \to \mathbb{B}\langle T_\Delta\rangle$ *computed by* $\mathcal{M}$ is defined by

$$\tau_{\mathcal{M}}(\xi) = \bigvee_{q \in F} h_\mu(\xi)_q$$

for every $\xi \in T_\Sigma$.

It should be clear that gfst and weighted tree transducers over $\mathbb{B}$ are semantically equivalent in the sense that, for every gfst $\mathcal{M}$, we can construct a weighted tree transducer $\mathcal{N}$ over $\mathbb{B}$ such that $\tau_{\mathcal{M}}(\xi) = \mathrm{supp}(\tau_{\mathcal{N}}(\xi))$ for every $\xi \in T_\Sigma$, and vice versa. In fact, if $\mathcal{M} = (Q, \Sigma, \Delta, R, F)$ and $\mathcal{N} = (Q, \Sigma, \Delta, \mathbb{B}, \mu, F)$, then the connection between them is the following: for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $w \in Q(X_k)^*$, and $q \in Q$, we have $\mu_k(\sigma)_{w,q} = \bigvee(1.\zeta \mid q(\sigma(x_1, \ldots, x_k)) \to \zeta, \langle w \rangle$ is in $R)$.

For the second step, we observe that, as in the case of wta, weighted tree transducers over $\mathbb{B}$ can easily be generalized to weighted tree transducers over an arbitrary semiring S. For the same reason as in Sect. 4, we make the following conventions.

In the rest of Sect. 5, we assume that S is commutative. Moreover, $\leftarrow_{\mathrm{IO}}$ will be abbreviated by $\leftarrow$.

Definition 5.1. *A* weighted tree transducer (over S)[4] *(for short: wtt) is a tuple $\mathcal{M} = (Q, \Sigma, \Delta, S, \mu, F)$, where:*

- *Q is a finite nonempty set, the* set of states, *with $Q \cap (\Sigma \cup \Delta) = \emptyset$.*
- *Σ and Δ are the* input and the output ranked alphabets, *respectively.*
- *$\mu = (\mu_k \mid k \in \mathbb{N})$ is a* family of rule mappings

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

such that $\mu_k(\sigma)_{w,q} \neq \widetilde{0}$ only for finitely many $(w, q) \in Q(X_k)^* \times Q$, and $\mu_k(\sigma)_{w,q} \in S\langle T_\Delta(Z_l)\rangle$ with $l = |w|$, for every $(w, q) \in Q(X_k)^* \times Q$.
- *$F \subseteq Q$ is the* set of designated states.

For such a wtt $\mathcal{M}$ we consider the Σ-algebra $(S\langle T_\Delta\rangle^Q, \mu_{\mathcal{M}})$ where, for every $k \geq 0$ and $\sigma \in \Sigma^{(k)}$, the k-ary operation $\mu_{\mathcal{M}}(\sigma) : S\langle T_\Delta\rangle^Q \times \cdots \times S\langle T_\Delta\rangle^Q \to S\langle T_\Delta\rangle^Q$ is defined as follows. For every $q \in Q$ and $v_1, \ldots, v_k \in S\langle T_\Delta\rangle^Q$, we have

$$\mu_{\mathcal{M}}(\sigma)(v_1, \ldots, v_k)_q = \sum_{\substack{w \in Q(X_k)^*,\\ w = q_1(x_{i_1}) \ldots q_l(x_{i_l})}} \mu_k(\sigma)_{w,q} \leftarrow \big((v_{i_1})_{q_1}, \ldots, (v_{i_l})_{q_l}\big).$$

Let us denote the unique Σ-algebra homomorphism from T_Σ to $S\langle T_\Delta\rangle^Q$ by h_μ. Then the tree series transformation $\tau_{\mathcal{M}} : T_\Sigma \to S\langle T_\Delta\rangle$ *computed by* $\mathcal{M}$ is defined as

$$\tau_{\mathcal{M}}(\xi) = \sum_{q \in F} h_\mu(\xi)_q$$

for every $\xi \in T_\Sigma$. We denote by $\mathrm{WTT}(S)$ the class of all tree series transformations over S that are computable by a wtt.

An equivalent definition of $\tau_{\mathcal{M}}$ can be given as follows. For every $q \in Q$, we define the tree series transformation $\tau_{\mathcal{M},q} : T_\Sigma \to S\langle T_\Delta\rangle$ by induction: for every $\xi = \sigma(\xi_1, \ldots, \xi_k) \in T_\Sigma$, let

[4] In the literature, a weighted tree transducer is also known as a (polynomial) *tree series transducer*. Moreover, μ is also called a *tree representation*.

$$\tau_{\mathcal{M},q}(\xi) = \sum_{\substack{w \in Q(X_k)^*, \\ w = q_1(x_{i_1}) \dots q_l(x_{i_l})}} \mu_k(\sigma)_{w,q} \leftarrow \big(\tau_{\mathcal{M},q_1}(\xi_{i_1}), \dots, \tau_{\mathcal{M},q_l}(\xi_{i_l})\big).$$

Then we can easily show that $\tau_{\mathcal{M},q}(\xi) = h_\mu(\xi)_q$, hence $\tau_{\mathcal{M}}(\xi) = \sum_{q \in F} \tau_{\mathcal{M},q}(\xi)$.

Example 5.2. As an example, we consider the wtt $\mathcal{M} = (Q, \Sigma, \Delta, \mathsf{Trop}, \mu, F)$ with $Q = \{q, q_0, q_\alpha, q_\beta\}$, $\Sigma = \{\delta^{(2)}, \gamma^{(1)}, \alpha^{(0)}, \beta^{(0)}\}$, $\Delta = \{\sigma^{(2)}, \gamma_1^{(1)}, \gamma_2^{(1)}, \alpha^{(0)}, \beta^{(0)}\}$, and $F = \{q_0\}$. Moreover, we specify the rule mappings such that, for every $\xi \in T_\Sigma$ and $\theta \in \{\alpha, \beta\}$, the following statements hold (where we have dropped the parentheses in subtrees of the form $\gamma(\zeta)$):

$$\tau_{\mathcal{M},q_\theta}(\xi) = \begin{cases} k.\theta & \text{if } \xi = \gamma^k\theta \text{ for } k \geq 0, \\ \widetilde{\infty} & \text{otherwise,} \end{cases}$$

$$\tau_{\mathcal{M},q}(\xi) = \begin{cases} \mathsf{min}_{\zeta \in L(n,\theta)} |\zeta|_{\gamma_2}.\zeta & \text{if } \xi = \gamma^n\theta \text{ for } n \geq 0, \\ \widetilde{\infty} & \text{otherwise,} \end{cases}$$

where $L(n, \theta) = \{\gamma_{i_1} \dots \gamma_{i_n}\theta \mid i_1, \dots, i_n \in \{1, 2\}\}$, and

$$\tau_{\mathcal{M},q_0}(\xi) = \begin{cases} \mathsf{min}_{\zeta \in L(n,\alpha)}(|\zeta|_{\gamma_2} + k) \,.\, \sigma(\zeta, \zeta) \\ \qquad \text{if } \xi = \delta(\gamma^n\theta, \gamma^k\alpha) \text{ for } n, k \geq 0, \\ \mathsf{min}_{\zeta_1, \zeta_2 \in L(n,\beta)}(|\sigma(\zeta_1, \zeta_2)|_{\gamma_2} + k) \,.\, \sigma(\zeta_1, \zeta_2) \\ \qquad \text{if } \xi = \delta(\gamma^n\theta, \gamma^k\beta) \text{ for } n, k \geq 0, \\ \widetilde{\infty} \quad \text{otherwise.} \end{cases}$$

For this purpose, we define μ such that, for every $\theta \in \{\alpha, \beta\}$,

$$\begin{aligned} \mu_1(\gamma)_{q_\theta(x_1),q_\theta} &= 1.z_1, \\ \mu_0(\theta)_{\varepsilon,q_\theta} &= 0.\theta, \\ \mu_1(\gamma)_{q(x_1),q} &= 0.\gamma_1(z_1) + 1.\gamma_2(z_1), \\ \mu_0(\theta)_{\varepsilon,q} &= 0.\theta, \\ \mu_2(\delta)_{q(x_1)q_\alpha(x_2),q_0} &= 0.\sigma(z_1, z_1), \\ \mu_2(\delta)_{q(x_1)q(x_1)q_\beta(x_2),q_0} &= 0.\sigma(z_1, z_2), \end{aligned}$$

and all the other not mentioned tuples $(w, p) \in Q(X_k)^* \times Q$ lead to $\widetilde{\infty}$.

Now we consider the input tree $\xi = \delta(\gamma^n\theta, \gamma^k\beta)$. Assuming that $\tau_{\mathcal{M},q}(\gamma^n\theta)$, $\tau_{\mathcal{M},q_\alpha}(\gamma^k\beta)$, and $\tau_{\mathcal{M},q_\beta}(\gamma^k\beta)$ have the desired form (as reported above), the tree series transformation $\tau_{\mathcal{M},q_0}$ can be evaluated on ξ as follows (where we abbreviate $\tau_{\mathcal{M},p}$ by τ_p for every $p \in Q$):

$$\begin{aligned} &\tau_{q_0}\big(\delta\big(\gamma^n\theta, \gamma^k\beta\big)\big) \\ &\quad = \mathsf{min}\,\big\{\mu_2(\delta)_{q(x_1)q_\alpha(x_2),q_0} \leftarrow \big(\tau_q\big(\gamma^n\theta\big), \tau_{q_\alpha}\big(\gamma^k\beta\big)\big), \\ &\qquad\qquad \mu_2(\delta)_{q(x_1)q(x_1)q_\beta(x_2),q_0} \leftarrow \big(\tau_q\big(\gamma^n\theta\big), \tau_q\big(\gamma^n\theta\big), \tau_{q_\beta}\big(\gamma^k\beta\big)\big)\big\} \end{aligned}$$

$$
\begin{aligned}
&= \min\big\{0.\sigma(z_1,z_1) \leftarrow \big(\tau_q\big(\gamma^n\theta\big),\widetilde{\infty}\big),\\
&\qquad\qquad 0.\sigma(z_1,z_2) \leftarrow \big(\tau_q\big(\gamma^n\theta\big),\tau_q\big(\gamma^n\theta\big),\tau_{q_\beta}\big(\gamma^k\beta\big)\big)\big\}\\
&\overset{\text{(P3)}}{=} \min\big\{\widetilde{\infty}, 0.\sigma(z_1,z_2) \leftarrow \big(\tau_q\big(\gamma^n\theta\big),\tau_q\big(\gamma^n\theta\big),\tau_{q_\beta}\big(\gamma^k\beta\big)\big)\big\}\\
&= 0.\sigma(z_1,z_2) \leftarrow \big(\tau_q\big(\gamma^n\theta\big),\tau_q\big(\gamma^n\theta\big),\tau_{q_\beta}\big(\gamma^k\beta\big)\big)\\
&= \min_{\substack{\xi'\in T_\Delta(Z_3)\\ \zeta_1,\zeta_2,\zeta_3\in T_\Delta}} a.\,\xi'(\zeta_1,\zeta_2,\zeta_3)\\
&\quad \big(\text{where}\quad a = \big(0.\sigma(z_1,z_2),\xi'\big) + \big(\tau_q\big(\gamma^n\theta\big),\zeta_1\big) + \big(\tau_q\big(\gamma^n\theta\big),\zeta_2\big)\\
&\qquad\qquad + \big(\tau_{q_\beta}\big(\gamma^k\beta\big),\zeta_3\big)\big)\\
&= \min_{\zeta_1,\zeta_2\in T_\Delta} \big(\big(\tau_q\big(\gamma^n\theta\big),\zeta_1\big) + \big(\tau_q\big(\gamma^n\theta\big),\zeta_2\big) + k\big).\sigma(\zeta_1,\zeta_2)\\
&\quad \big(\text{choosing}\quad \xi' = \sigma(z_1,z_2)\quad\text{and}\quad \zeta_3=\beta\big)\\
&= \min_{\zeta_1,\zeta_2\in L(n,\theta)} \big(|\sigma(\zeta_1,\zeta_2)|_{\gamma_2} + k\big).\sigma(\zeta_1,\zeta_2).
\end{aligned}
$$

We shortly discuss the copy and the deletion capabilities of weighted tree transducers where ξ stands for an input tree and ξ' is a subtree of ξ.

(T) A wtt can process several copies of ξ'; the weight of each processing of ξ' is included into that of ξ (even if some of the copies are processed in the same way). For instance, in Example 5.2 for $\xi = \delta(\xi', \gamma^k\beta)$ and $\xi' = \gamma^n\theta$, we have that

$$\tau_{\mathcal{M},q_0}(\xi) = 0.\sigma(z_1,z_2) \leftarrow \big(\tau_{\mathcal{M},q}(\xi'),\tau_{\mathcal{M},q}(\xi'),\dots\big)$$

due to the equation $\mu_2(\delta)_{q(x_1)q(x_1)q_\beta(x_2),q_0} = 0.\sigma(z_1,z_2)$. Then by the definition of IO-substitution, two trees ζ_1,ζ_2 are chosen from the support of $\tau_{\mathcal{M},q}(\xi')$, and z_i is replaced by ζ_i in $\sigma(z_1,z_2)$; moreover, both values $(\tau_{\mathcal{M},q}(\xi'), \zeta_1)$ and $(\tau_{\mathcal{M},q}(\xi'),\zeta_2)$ are included in $\tau_{\mathcal{M},q_0}(\xi)$:

$$\tau_{\mathcal{M},q_0}(\xi) = \min_{\zeta_1,\zeta_2\in T_\Delta} \big(\big(\tau_{\mathcal{M},q}(\xi'),\zeta_1\big) + \big(\tau_{\mathcal{M},q}(\xi'),\zeta_2\big) + \cdots\big).\sigma(\zeta_1,\zeta_2).$$

We note that, for top-down tree transducers, this corresponds to the property "(T) copying an input subtree followed by processing these copies nondeterministically (and independently)" [48].

(B1) A wtt can process ξ' and then copy the result of the processing; the weight of the processing of ξ' is included into that of ξ only once, no matter how many times the result is copied. This happens, e.g., in Example 5.2 for $\xi = \delta(\xi',\gamma^k\alpha)$ and $\xi' = \gamma^n\theta$: due to the equation $\mu_2(\delta)_{q(x_1)q_\alpha(x_2),q_0} = 0.\sigma(z_1,z_1)$, we have that

$$\tau_{\mathcal{M},q_0}(\xi) = 0.\sigma(z_1,z_1) \leftarrow \big(\tau_{\mathcal{M},q}(\xi'),\dots\big)$$

and, by the definition of IO-substitution, a tree ζ in the support of $\tau_{\mathcal{M},q}(\xi')$ is chosen and then copied to both occurrences of z_1 in $\sigma(z_1,z_1)$, whereas its weight $(\tau_{\mathcal{M},q}(\xi'),\zeta)$ is included in $\tau_{\mathcal{M},q}(\xi)$ only once:

$$\tau_{\mathcal{M},q_0}(\xi) = \min_{\zeta \in T_\Delta} \big((\tau_{\mathcal{M},q}(\xi'), \zeta) + \cdots\big).\sigma(\zeta, \zeta).$$

For bottom-up tree transducers, this corresponds to the property "(B1) nondeterministically processing an input subtree followed by copying the result of this process" [48].

(B2) A wtt can process ξ' (and thereby check a property of ξ') and then it can delete the result of this processing. In this case, the weight of the processing of ξ' is included into that of ξ. This situation occurs, e.g., in Example 5.2 for $\xi_1 = \delta(\xi', \gamma^k\alpha)$ and $\xi_2 = \delta(\xi', \gamma^k\beta)$ where $\xi' = \gamma^n\theta$. Then

$$\tau_{\mathcal{M},q_0}(\xi_1) = 0.\sigma(z_1, z_1) \leftarrow \big(\ldots,\ \tau_{\mathcal{M},q_\alpha}(\gamma^k\alpha)\big),$$
$$\tau_{\mathcal{M},q_0}(\xi_2) = 0.\sigma(z_1, z_2) \leftarrow \big(\ldots,\ \ldots,\ \tau_{\mathcal{M},q_\beta}(\gamma^k\beta)\big)$$

due to the equations

$$\mu_2(\delta)_{q(x_1)q_\alpha(x_2),q_0} = 0.\sigma(z_1, z_1),$$
$$\mu_2(\delta)_{q(x_1)q(x_1)q_\beta(x_2),q_0} = 0.\sigma(z_1, z_2),$$

resp., and by the definition of IO-substitution the value k is included in $\tau_{\mathcal{M},q_0}(\xi_1)$ and $\tau_{\mathcal{M},q_0}(\xi_2)$:

$$\tau_{\mathcal{M},q_0}(\xi_2) = \min_{\zeta \in T_\Delta} (\cdots + k).\sigma(\zeta, \zeta),$$
$$\tau_{\mathcal{M},q_0}(\xi_1) = \min_{\zeta_1, \zeta_2 \in T_\Delta} (\cdots + \cdots + k).\sigma(\zeta_1, \zeta_2).$$

For bottom-up tree transducers, this phenomenon is known as "(B2) checking a property of an input subtree followed by deletion" [48].

In the literature (e.g., in [91, 53, 65, 102, 104]), wtt are often defined such that a rule mapping has the type

$$\mu_k : \Sigma^{(k)} \to S\langle\!\langle T_\Delta(Z)\rangle\!\rangle^{Q(X_k)^* \times Q},$$

i.e., the tree series $\mu_k(\sigma)_{w,q}$ is not necessarily polynomial, and there the wtt of our Definition 5.1 is called polynomial. We would like to ask the reader to keep this is mind, when we later on refer to statements proved in that literature for the more general type of wtt. Clearly, we will only refer to statements which hold for the polynomial restriction of that general type.

In some works, a wtt $\mathcal{M}$ has, instead of designated states, a so-called *root output*. This root output is specified by a mapping $\nu : Q \to S\langle C_\Delta\rangle$ [102, 104] or more generally by $\nu : Q \to S\langle T_\Delta(Z_1)\rangle$ [91] and the tree series transformation computed by $\mathcal{M}$ is defined as $\tau_{\mathcal{M}}(\xi) = \sum_{q \in Q} \nu(q) \leftarrow h_\mu(\xi)_q$ for every $\xi \in T_\Sigma$. Certainly, the wtt with root output generalizes our wtt because the set F of designated states can be simulated by the particular root output defined by $\nu(q) = 1.z_1$ if $q \in F$ and $\nu(q) = \widetilde{0}$, otherwise for $q \in Q$. On the other hand, our wtt and the wtt with root output are semantically equivalent, because for every wtt $\mathcal{M}$ with root output one can construct a wtt $\mathcal{M}'$ with $\tau_{\mathcal{M}} = \tau_{\mathcal{M}'}$, see [104], Lemma 10.

5.3 Restricted Models

We define three types of restrictions of wtt. The first of them is made concerning the possible directions in which weighted tree transducers process the input trees. We say that a wtt $\mathcal{M} = (Q, \Sigma, \Delta, S, \mu, F)$ is a:

- *top-down wtt with regular look-ahead* (resp., *top-down wtt*) if, for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $w \in Q(X_k)^*$, and $q \in Q$, the tree series $\mu_k(\sigma)_{w,q}$ is linear (resp., linear and nondeleting) in $Z_{|w|}$,
- *bottom-up wtt* if, for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $w \in Q(X_k)^*$, and $q \in Q$, such that $\mu_k(\sigma)_{w,q} \neq \widetilde{0}$ we have that $w = q_1(x_1) \ldots q_k(x_k)$ for some $q_1, \ldots, q_k \in Q$.

The corresponding class of tree series transformations over S is denoted by $\mathrm{TOP}^{\mathrm{R}}(S)$ (resp., $\mathrm{TOP}(S)$, $\mathrm{BOT}(S)$). If $\mathcal{M}$ is a top-down wtt with regular look-ahead or a top-down wtt (resp., bottom-up wtt), then F is called the set of *initial* (resp., *final* states).

Example 5.3. If we drop from the wtt $\mathcal{M}$ of Example 5.2 the state q_α (and all the equations that involve q_α), then we obtain a top-down wtt $\mathcal{M}'$ with regular look-ahead such that for every $\xi \in T_\Sigma$ and $\theta \in \{\alpha, \beta\}$

$$\tau_{\mathcal{M}',q_0}(\xi) = \begin{cases} \mathsf{min}_{\zeta_1,\zeta_2 \in L(n,\theta)}(|\sigma(\zeta_1,\zeta_2)|_{\gamma_2} + k) \,.\, \sigma(\zeta_1,\zeta_2) \\ \qquad \text{if } \xi = \delta(\gamma^n\theta, \gamma^k\beta) \text{ for } n,k \geq 0, \\ \widetilde{\infty} \quad \text{otherwise.} \end{cases}$$

In fact, the state q_β can be called a *look-ahead state.*

If we replace in the rule mapping μ_2 of $\mathcal{M}'$ the equation

$$\mu_2(\delta)_{q(x_1)q(x_1)q_\beta(x_2),q_0} = 0.\sigma(z_1,z_2)$$

by

$$\mu_2(\delta)_{q(x_1)q(x_1),q_0} = 0.\sigma(z_1,z_2),$$

and we drop all the equations that involve q_β, then we obtain a top-down wtt $\mathcal{M}''$ such that for every $\xi \in T_\Sigma$

$$\tau_{\mathcal{M}'',q_0}(\xi) = \begin{cases} \mathsf{min}_{\zeta_1,\zeta_2 \in L(n,\theta)} \, |\sigma(\zeta_1,\zeta_2)|_{\gamma_2} \,.\, \sigma(\zeta_1,\zeta_2) \\ \qquad \text{if } \xi = \delta(\gamma^n\theta, \zeta) \text{ for } n \geq 0 \text{ and } \zeta \in T_\Sigma, \\ \widetilde{\infty} \quad \text{otherwise.} \end{cases}$$

If we drop from the wtt $\mathcal{M}$ of Example 5.2 the state q_β, then we obtain a bottom-up wtt $\mathcal{M}'''$ such that for every $\xi \in T_\Sigma$

$$\tau_{\mathcal{M}''',q_0}(\xi) = \begin{cases} \mathsf{min}_{\zeta \in L(n,\theta)}(|\zeta|_{\gamma_2} + k) \,.\, \sigma(\zeta,\zeta) \\ \qquad \text{if } \xi = \delta(\gamma^n\theta, \gamma^k\alpha) \text{ for } n,k \geq 0, \\ \widetilde{\infty} \quad \text{otherwise.} \end{cases}$$

(a)	(b)
$\gamma(\sigma(\sigma(\alpha,\alpha),\alpha))$	$\gamma(\sigma(\sigma(\alpha,\alpha),\alpha))$
$\Rightarrow \gamma(\sigma(\sigma(q(\alpha),\alpha),\alpha))$	$\Rightarrow \gamma(\sigma(\sigma(q(\alpha),\alpha),\alpha))$
$\Rightarrow \gamma(\sigma(\sigma(q(\alpha),\overline{\alpha}(\alpha)),\alpha))$	$\Rightarrow \gamma(\sigma(\sigma(q(\alpha),q(\alpha)),\alpha))$
$\Rightarrow \gamma(\sigma(f(\alpha),\alpha))$	$\Rightarrow \gamma(\sigma(q(\sigma(\alpha,\alpha)),\alpha))$
$\Rightarrow \gamma(\sigma(f(\alpha),q(\alpha)))$	$\Rightarrow \gamma(\sigma(q(\sigma(\alpha,\alpha)),\overline{\alpha}(\alpha)))$
$\Rightarrow \gamma(f(\sigma(\alpha,\alpha)))$	$\Rightarrow \gamma(f(\sigma(\alpha,\alpha)))$
$\Rightarrow f(\gamma(\sigma(\alpha,\alpha)))$	$\Rightarrow f(\gamma(\sigma(\alpha,\alpha)))$

Fig. 1. Two leftmost derivations of $f(\gamma(\sigma(\alpha,\alpha)))$ from $\gamma(\sigma(\sigma(\alpha,\alpha),\alpha))$

Example 5.4. Let us consider again $\Sigma = \{\sigma^{(2)}, \gamma^{(1)}, \alpha^{(0)}\}$ and the pattern $\sigma(z,\alpha)$. We construct a bottom-up tree transducer $\mathcal{N}$ whose domain is the set of all Σ-trees in which $\sigma(z,\alpha)$ occurs at least once, i.e., the tree language recognized by the bottom-up tree automaton $\mathcal{A}$ of Example 3.1. Given an input tree $\xi \in T_\Sigma$, the bottom-up tree transducer $\mathcal{N}$ nondeterministically selects an occurrence of $\sigma(z,\alpha)$ in ξ, if any, then either leaves it unchanged or deletes σ and α from it. If $\sigma(z,\alpha)$ does not occur in ξ, then $\mathcal{N}$ will not compute any output tree for ξ. For this, let $\mathcal{N} = (Q, \Sigma, \Sigma, R, F)$, where $Q = \{q, \overline{\alpha}, f\}$, $F = \{f\}$, and R is the set of the following rules:

$$
\begin{aligned}
\alpha &\to q(\alpha), & \alpha &\to \overline{\alpha}(\alpha),\\
\gamma\big(q(x_1)\big) &\to q\big(\gamma(x_1)\big), & \gamma\big(f(x_1)\big) &\to f\big(\gamma(x_1)\big),\\
\sigma\big(q(x_1), q(x_2)\big) &\to q\big(\sigma(x_1,x_2)\big), & \sigma\big(q(x_1), \overline{\alpha}(x_2)\big) &\to f\big(\sigma(x_1,x_2)\big) \mid f(x_1),\\
\sigma\big(f(x_1), q(x_2)\big) &\to f\big(\sigma(x_1,x_2)\big), & \sigma\big(q(x_1), f(x_2)\big) &\to f\big(\sigma(x_1,x_2)\big).
\end{aligned}
$$

Let $\Rightarrow_{\mathcal{N}}$ be the term rewrite relation induced by R. We consider the input tree $\xi = \gamma(\sigma(\sigma(\alpha,\alpha),\alpha))$ and show a derivation starting from ξ in Fig. 1(a) (where we have dropped $\mathcal{N}$ from $\Rightarrow_{\mathcal{N}}$). Hence, we have $\gamma(\sigma(\alpha,\alpha)) \in \tau_{\mathcal{N}}(\gamma(\sigma(\sigma(\alpha,\alpha), \alpha)))$. In fact, this derivation is a *leftmost derivation* [66] meaning that in every step we applied a rule to the leftmost redex. Note that there is another leftmost derivation starting also from ξ with the same result; see Fig. 1(b).

Now we view $\mathcal{N}$ as a bottom-up wtt over the semiring Nat. More exactly, let $\mathcal{M} = (Q, \Sigma, \Sigma, \mathsf{Nat}, \mu, F)$ be the bottom-up wtt, where μ is defined as follows. For every $k \in \{0,1,2\}$, $\delta \in \Sigma^{(k)}$, and $q_1, \ldots, q_k, p \in Q$, we have

$$
\mu_k(\delta)_{q_1(x_1)\ldots q_k(x_k),p} = \sum_{\delta(q_1(x_1),\ldots,q_k(x_k)) \to p(\zeta(x_1,\ldots,x_k)) \in R} 1.\zeta .
$$

For instance, $\mu_2(\sigma)_{q(x_1)\overline{\alpha}(x_2),f} = 1.\sigma(z_1,z_2) + 1.z_1$. One can show that $(\tau_{\mathcal{M}}(\xi), \eta)$ is the number of leftmost derivations of $f(\eta)$ from ξ using the rewrite relation $\Rightarrow_{\mathcal{N}}$, for every $\xi \in T_\Sigma$ and $\eta \in T_\Delta$, cf. [66], Example 5.6.

The second type of restriction concerns the state behavior: it can be total and/or deterministic. Here, we will only impose these restrictions to top-down wtt and bottom-up wtt. Let $\mathcal{M} = (Q, \Sigma, \Delta, S, \mu, F)$ be a top-down wtt or a bottom-up wtt.

- The top-down wtt $\mathcal{M}$ is *total* (resp., *deterministic*) if, for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, and $q \in Q$, there is at least one (resp., at most one) $w \in Q(X_k)^*$ and $\zeta \in T_\Delta(Z)$, such that $\zeta \in \mathrm{supp}(\mu_k(\sigma)_{w,q})$. Moreover, in a deterministic top-down wtt, we additionally require that the set F is a singleton.
- The bottom-up wtt $\mathcal{M}$ is *total* (resp., *deterministic*) if, for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, and $w \in Q(X_k)^*$, there is at least one (resp., at most one) $q \in Q$ and $\zeta \in T_\Delta(Z)$, such that $\zeta \in \mathrm{supp}(\mu_k(\sigma)_{w,q})$.

We note that if $\mathcal{M}$ is a deterministic top-down or a deterministic bottom-up wtt, then the operation $+$ of S is not used for the computation of $\tau_\mathcal{M}$. Moreover, in the case that $\mathcal{M}$ is total we do not require $F = Q$.

The restrictions total and deterministic are abbreviated by t and d, respectively. Any combination x over $\{\mathrm{t}, \mathrm{d}\}$ can be applied to a top-down (resp., bottom-up) wtt, hence we obtain an x top-down (resp., bottom-up) wtt. The class of tree series transformations computed by all so-obtained x top-down wtt or x bottom-up wtt is denoted by prefixing with x the notation of the class of tree series transformations computed by that kind of wtt. For example, the class of tree series transformations over S computed by total and deterministic top-down wtt is denoted by td-TOP(S). Moreover, a total and deterministic top-down (resp., bottom-up) wtt $\mathcal{M}$ is called a *homomorphism top-down wtt* (resp., *homomorphism bottom-up wtt*), provided $Q = F = \{q\}$. The class of tree series transformations over S which are computable by homomorphism top-down wtt (resp., homomorphism bottom-up wtt) is denoted by h-TOP(S) (resp., h-BOT(S)).

Finally, the third type of restriction concerns the form of the polynomials that may occur in the rule mappings. A wtt $\mathcal{M} = (Q, \Sigma, \Delta, S, \mu, F)$ is:

- *Boolean*, if for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $w \in Q(X_k)^*$, and $q \in Q$, the tree series $\mu_k(\sigma)_{w,q}$ is Boolean,
- *linear* (resp., *nondeleting*), if for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $w \in Q(X_k)^*$, and $q \in Q$ such that $\mu_k(\sigma)_{w,q} \neq \widetilde{0}$, both $\mu_k(\sigma)_{w,q}$ is linear (resp., nondeleting) in $Z_{|w|}$ and w is linear (resp., nondeleting) in X_k.

Moreover, if $\mathcal{M}$ is a bottom-up wtt, then $\mathcal{M}$ is:

- *wta*, if $\Sigma = \Delta$ and for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $w \in Q(X_k)^*$, and $q \in Q$, we have that $\mu_k(\sigma)_{w,q} = a.\sigma(z_1, \ldots, z_k)$ for some $a \in S$.

The first three restrictions are abbreviated by b, l, and n, respectively. Any combination of $\{\mathrm{b}, \mathrm{l}, \mathrm{n}\}$ can be applied to any of the wtt classes defined above. For example, the class of tree series transformations over S computed by linear top-down wtt with regular look-ahead is denoted by l-TOP$^\mathrm{R}$(S), while the class of tree series transformations over S computed by deterministic, linear, and nondeleting bottom-up wtt is denoted by dln-BOT(S).

Each wta bottom-up wtt computes a weighted identity. The class of all weighted identities over S computed by wta bottom-up wtt is denoted by $\mathrm{WTA}(S)$.

By two easy constructions, we can show that linear and nondeleting top-down wtt have the same computation power as linear and nondeleting bottom-up wtt and hereby generalize the corresponding result concerning tree transducers, cf. [48], Theorem 2.9.

Theorem 5.5 ([53], Theorem 5.24). $\text{ln-TOP}(S) = \text{ln-BOT}(S)$ *and* $\text{bln-TOP}(S) = \text{bln-BOT}(S)$.

Although bottom-up homomorphism tree transducers compute the same class of tree transformations as top-down homomorphism tree transducers (cf. [48]), this equivalence does not hold for the corresponding classes of tree series transformations, cf. [53], Proposition 3.14. However, $\text{bh-TOP}(S) = \text{bh-BOT}(S)$, therefore, we do not use different notations but denote this class by $\text{b-HOM}(S)$, cf. [53], Corollary 4.15. We note that the equation was proved in [53] only for idempotent S. However, it is easy to see that idempotency is not necessary because, as mentioned, $+$ is not used for the computation of the tree series transformation computed by any homomorphism wtt.

We also note that the identity tree series transformation ι can be computed by any of the above kind of wtt.

Finally, we show that wtt over positive semirings have the same computational power as gfst. This relation has been discussed in [53], Sect. 4.1 for Boolean wtt and gfst; and it has been proved in detail in [53], Sects. 4.2 and 4.3 for the case of bottom-up wtt and top-down wtt, respectively, over idempotent semirings.

The heart of such a comparison is the concept of the relatedness of wtt and gfst. We define a wtt $\mathcal{M} = (Q, \Sigma, \Delta, S, \mu, F)$ and a gfst $\mathcal{N} = (Q, \Sigma, \Delta, R, F)$ as being *related* if the following holds:

$$q\big(\sigma(x_1, \ldots, x_k)\big) \to \zeta, \langle w \rangle \text{ is in } R \quad \text{iff} \quad \zeta \in \operatorname{supp}\big(\mu_k(\sigma)_{w,q}\big)$$

for every $q \in Q$, $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $w \in Q(X_k)^*$, and $\zeta \in T_\Delta(Z_{|w|})$.

The same concept of relatedness can be defined for top-down tree transducers and top-down wtt, and for bottom-up tree transducers and bottom-up wtt. Assuming that S is positive, it is easy to prove by induction on ξ that $h_{\mathcal{N}}(\xi)_q = \operatorname{supp}(h_\mu(\xi)_q)$ for every $\xi \in T_\Sigma$ and $q \in Q$. The key equation in this proof is that for $\xi = \sigma(\xi_1, \ldots, \xi_k)$ and $q \in Q$,

$$\bigcup_{\substack{\text{for every rule}\\ \text{of the form (8)}}} \{\zeta(\zeta_1, \ldots, \zeta_l) \mid \zeta_j \in h_{\mathcal{N}}(\xi_{i_j})_{q_j},\ 1 \leq j \leq l\}$$
$$= \operatorname{supp}\Bigg(\sum_{w = q_1(x_{i_1}) \ldots q_l(x_{i_l}) \in Q(X_k)^*} \mu_k(\sigma)_{w,q} \leftarrow \big(h_\mu(\xi_{i_1})_{q_1}, \ldots, h_\mu(\xi_{i_l})_{q_l}\big) \Bigg).$$

To prove this, we need the following facts. If S is positive, then for all tree series $r_1, r_2 \in S\langle T_\Delta\rangle$, we have $\mathrm{supp}(r_1 + r_2) = \mathrm{supp}(r_1) \cup \mathrm{supp}(r_2)$, and for $r \in S\langle T_\Delta(Z)\rangle$, finite $I \subseteq \mathbb{N}$, and family $(s_i \mid i \in I)$ with $s_i \in S\langle T_\Delta\rangle$ we have $\mathrm{supp}(r \leftarrow (s_i \mid i \in I)) = \{\zeta(\zeta_i \mid i \in I) \mid \zeta \in \mathrm{supp}(r), \zeta_i \in \mathrm{supp}(s_i)\}$. Then the equation follows from the induction hypothesis, the relatedness of $\mathcal{M}$ and $\mathcal{N}$, and the fact that S is positive.

In order to relate the semantics of a wtt $\mathcal{M}$ with that of a gfst $\mathcal{N}$ we will use a generalization of the mapping supp. For this, define the mapping

$$\mathrm{supp}_{S,\Sigma,\Delta} : (T_\Sigma \to S\langle T_\Delta\rangle) \to \big(T_\Sigma \to \mathcal{P}_{\mathrm{fin}}(T_\Delta)\big)$$

by $(\mathrm{supp}_{S,\Sigma,\Delta}(\tau))(\xi) = \mathrm{supp}(\tau(\xi))$ for every $\tau : T_\Sigma \to S\langle T_\Delta\rangle$ and $\xi \in T_\Sigma$. Let supp_S denote the class of all mappings $\mathrm{supp}_{S,\Sigma,\Delta}$.

Now it easily follows that $\tau_\mathcal{N} = \mathrm{supp}_{S,\Sigma,\Delta}(\tau_\mathcal{M})$ if $\mathcal{M}$ and $\mathcal{N}$ are related and S is positive. Since, for every given wtt $\mathcal{M}$ we can construct a gfst $\mathcal{N}$ which is related to $\mathcal{M}$, and vice versa, we obtain the following result.

Theorem 5.6 ([53], Theorem 4.13 and Theorem 4.6). *Let S be positive. Then:*

(A) $\mathrm{supp}_S(\mathrm{WTT}(S)) = \mathrm{GFST}$.
(B) $\mathrm{supp}_S(\mathrm{TOP}(S)) = \mathrm{TOP}$.
(C) $\mathrm{supp}_S(\mathrm{BOT}(S)) = \mathrm{BOT}$.

The next theorem shows that the range of a wta bottom-up wtt is a recognizable tree series and vice versa. For an arbitrary weighted identity $\tau : T_\Sigma \to S\langle T_\Sigma\rangle$, we define the tree series $\mathrm{range}(\tau) \in S\langle T_\Sigma\rangle$ by $(\mathrm{range}(\tau), \xi) = (\tau(\xi), \xi)$ for every $\xi \in T_\Sigma$. We let $\mathrm{WTA}(\Sigma, S)$ denote the class of all those weighted identities in $\mathrm{WTA}(S)$ which have the type $T_\Sigma \to S\langle T_\Sigma\rangle$.

Theorem 5.7. $\mathrm{Rec}(\Sigma, S) = \mathrm{range}(\mathrm{WTA}(\Sigma, S))$.

Proof. A wta $\mathcal{A} = (Q, \Sigma, S, \mu, F)$ with Boolean root weights and a wta bottom-up wtt $\mathcal{M} = (Q, \Sigma, \Sigma, S, \nu, F)$ are related if for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $q_1, \ldots, q_k, q \in Q$, and $a \in S$: $\mu_k(\sigma)_{q_1\ldots q_k,q} = a$ iff $\nu_k(\sigma)_{q_1(x_1)\ldots q_k(x_k),q} = a.\sigma(z_1, \ldots, z_k)$. Then it is straightforward to prove by induction that $h_\mu(\xi)_q.\xi = h_\nu(\xi)_q$ for every $\xi \in T_\Sigma$ and $q \in Q$; this implies that $(r_\mathcal{A}, \xi).\xi = \tau_\mathcal{M}(\xi)$, and finally that $r_\mathcal{A} = \mathrm{range}(\tau_\mathcal{M})$. Then the statement of the theorem is implied by Theorem 3.6. □

5.4 Composition and Decomposition

In this subsection, we will investigate composition and decomposition results for $\mathrm{WTT}(S)$. A composition result has the form $\mathrm{C}(S);\mathrm{D}(S) \subseteq \mathrm{E}(S)$, where $\mathrm{C}(S)$, $\mathrm{D}(S)$, and $\mathrm{E}(S)$ are subclasses of $\mathrm{WTT}(S)$. Clearly, the smaller the class $\mathrm{E}(S)$ is (for fixed $\mathrm{C}(S)$ and $\mathrm{D}(S)$) the stronger the composition result is. If $\mathrm{C}(S) = \mathrm{E}(S)$, then we say that $\mathrm{E}(S)$ is closed under right-composition with D(S), and if D(S) = E(S), then we say that E(S) is closed under left-composition with C(S). If C(S) = D(S) = E(S), then C(S) is closed under composition. The first composition results were obtained for gfst in [48] and for top-down and bottom-up tree transducers in [48], [49], and [4].

Decomposition results have the form E(S) $\subseteq$ C(S);D(S) where again C(S), D(S), and E(S) are subclasses of WTT(S). Of course, a decomposition E(S) $\subseteq$ C(S);D(S) makes sense only if C(S) and D(S) are subclasses of E(S). Such decomposition results were first obtained for gfst, top-down, and bottom-up tree transducers in [48] and [49]. In the rest of this section, we will generalize some of the composition and decomposition results obtained in the above cited papers.

5.4.1 Results Concerning wtt

First, we consider composition results of the form C(S);D(S) $\subseteq$ E(S), where C(S) $\subseteq$ TOP(S) and D(S) $\subseteq$ BOT(S). Thus, for every top-down wtt $\mathcal{M}_1$ of type c and bottom-up wtt $\mathcal{M}_2$ of type d, we have to construct a wtt $\mathcal{M}$ of type e such that $\tau_{\mathcal{M}_1};\tau_{\mathcal{M}_2} = \tau_{\mathcal{M}}$. There is a general approach to achieve this goal: $\mathcal{M}$ is the *syntactic composition of* $\mathcal{M}_1$ *and* $\mathcal{M}_2$, denoted by $\mathcal{M}_1 \circ \mathcal{M}_2$ (cf. [4], pages 195 and 199). The wtt $\mathcal{M}_1 \circ \mathcal{M}_2$ is obtained, roughly speaking, by letting $\mathcal{M}_2$ work on the pieces of output produced by $\mathcal{M}_1$. However, since these pieces of output may contain variables from Z, which $\mathcal{M}_2$ cannot process, we first extend $\mathcal{M}_2$ appropriately, cf. [104], Definition 14.

To define this extension, let $\mathcal{M} = (Q, \Sigma, \Delta, S, \mu, F)$ be a bottom-up wtt. Let $l \geq 0$ and $\overline{q} \in Q^J$ for some $J \subseteq Z_l$. We define the $(\Sigma \cup Z_l)$-algebra $(S\langle T_\Delta(Z_l)\rangle^Q, \mu^{\overline{q}}_{\mathcal{M}})$ where every $z \in Z_l$ is a nullary symbol and we define $\mu^{\overline{q}}_{\mathcal{M}}(z)()_q = 1.z$ if $z \in J$ and $q = \overline{q}_z$, and $\mu^{\overline{q}}_{\mathcal{M}}(z)()_q = \widetilde{0}$ otherwise; and for every $k \geq 0$ and $\sigma \in \Sigma^{(k)}$, the k-ary operation $\mu^{\overline{q}}_{\mathcal{M}}(\sigma) : S\langle T_\Delta(Z_l)\rangle^Q \times \cdots \times S\langle T_\Delta(Z_l)\rangle^Q \to S\langle T_\Delta(Z_l)\rangle^Q$ is defined in the same way as $\mu_{\mathcal{M}}(\sigma)$. Let us denote the unique $(\Sigma \cup Z_l)$-algebra homomorphism from $T_{\Sigma\cup Z_l}$ to $S\langle T_\Delta(Z_l)\rangle^Q$ by $h^{\overline{q}}_\mu$, and let us denote its extension to a mapping of type $S\langle T_{\Sigma\cup Z_l}\rangle \to S\langle T_\Delta(Z_l)\rangle^Q$ (cf. page 363) also by $h^{\overline{q}}_\mu$.

Now we define the concept of syntactic composition and, in fact, we compose an *arbitrary* wtt with a bottom-up wtt; this will be useful for the results concerning bottom-up wtt. The *syntactic composition* (called simple composition in [104]) of a wtt $\mathcal{M}_1 = (Q_1, \Sigma, \Delta, S, \mu^1, F_1)$ and a bottom-up wtt $\mathcal{M}_2 = (Q_2, \Delta, \Gamma, S, \mu^2, F_2)$, see [104], Definition 22, is the wtt $\mathcal{M}_1 \circ \mathcal{M}_2 = (Q_1 \times Q_2, \Sigma, \Gamma, S, \mu, F_1 \times F_2)$, such that for every $k, l \geq 0$, $\sigma \in \Sigma^{(k)}$, $p, p_1, \ldots, p_l \in Q_1$, $q, q_1, \ldots, q_l \in Q_2$, and $1 \leq i_1, \ldots, i_l \leq k$, we have

$$\mu_k(\sigma)_{(p_1,q_1)(x_{i_1})\ldots(p_l,q_l)(x_{i_l}),(p,q)} = h^{\overline{q}}_{\mu^2}\big(\mu^1_k(\sigma)_{p_1(x_{i_1})\ldots p_l(x_{i_l}),p}\big)_q$$

where $\overline{q} \in Q_2^{Z_l}$ with $\overline{q}_{z_i} = q_i$ for every $z_i \in Z_l$. We note that this composition generalizes the syntactic composition of bottom-up tree transducers introduced in [4], page 199.

Remark 5.8. If both $\mathcal{M}_1$ and $\mathcal{M}_2$ are (deterministic) bottom-up wtt, then $\mathcal{M}_1 \circ \mathcal{M}_2$ is a (deterministic) bottom-up wtt. Moreover, if both $\mathcal{M}_1$ and $\mathcal{M}_2$ are homomorphism bottom-up wtt and $\mathcal{M}_2$ is Boolean, then $\mathcal{M}_1 \circ \mathcal{M}_2$ is a homomorphism bottom-up wtt. If, in addition, $\mathcal{M}_1$ is Boolean too, then $\mathcal{M}_1 \circ \mathcal{M}_2$ is a Boolean homomorphism bottom-up wtt.

If $\mathcal{M}_1$ is a (linear) top-down wtt and $\mathcal{M}_2$ is a linear bottom-up wtt, then $\mathcal{M}_1 \circ \mathcal{M}_2$ is a (linear) top-down wtt with regular look-ahead. If $\mathcal{M}_1$ is a (linear, nondeleting, or linear and nondeleting) top-down wtt and $\mathcal{M}_2$ is a linear and nondeleting bottom-up wtt, then $\mathcal{M}_1 \circ \mathcal{M}_2$ is a (linear, nondeleting, or linear and nondeleting) top-down wtt. The proof needs property P6 of IO-substitution.

Now we will show that, under certain conditions, $\mathcal{M}_1 \circ \mathcal{M}_2$ computes $\tau_{\mathcal{M}_1}; \tau_{\mathcal{M}_2}$. For this, we formulate an important property of (certain restricted) bottom-up wtt, namely, that h_μ distributes over substitutions $\xi(\xi_i \mid z_i \in \mathrm{var}(\xi))$ for $\xi \in T_\Sigma(Z)$ and $\xi_i \in T_\Sigma$.

Lemma 5.9 ([104], Proposition 18). *Let $\mathcal{M} = (Q, \Sigma, \Delta, S, \mu, F)$ be a bottom-up wtt, $q \in Q$, $l \geq 0$, $\xi \in T_\Sigma(Z_l)$, and $\xi_i \in T_\Sigma$ for every $z_i \in \mathrm{var}(\xi)$. If (a) $\mathcal{M}$ is Boolean and deterministic or (b) ξ is linear in Z_l, then*

$$h_\mu\big(\xi\big(\xi_i \;\big|\; z_i \in \mathrm{var}(\xi)\big)\big)_q = \sum_{\overline{q} \in Q^{\mathrm{var}(\xi)}} h_\mu^{\overline{q}}(\xi)_q \leftarrow \big(h_\mu(\xi_i)_{\overline{q}_{z_i}} \;\big|\; z_i \in \mathrm{var}(\xi)\big).$$

Proof. The proof is performed by induction on ξ. In the proof, properties P7 and P8 of IO-substitution are used. Moreover, for item (a), a version of P8 is used which also assures associativity of tree series substitution, cf. [104], Lemma 8. □

Next we show sufficient conditions which guarantee that $\mathcal{M}_1 \circ \mathcal{M}_2$ computes $\tau_{\mathcal{M}_1}; \tau_{\mathcal{M}_2}$. Note that the following lemma generalizes [4], Theorem 6.

Lemma 5.10 ([104], Lemma 23). *Let $\mathcal{M}_1$ be a wtt and $\mathcal{M}_2$ a bottom-up wtt. If (a) $\mathcal{M}_1$ is a bottom-up wtt and $\mathcal{M}_2$ is total, deterministic, and Boolean or (b) $\mathcal{M}_1$ is a top-down wtt, then for every $\xi \in T_\Sigma$, $p \in Q_1$, and $q \in Q_2$, we have $h_{\mu^2}(h_{\mu^1}(\xi)_p)_q = h_\mu(\xi)_{(p,q)}$ and $\tau_{\mathcal{M}_1 \circ \mathcal{M}_2} = \tau_{\mathcal{M}_1}; \tau_{\mathcal{M}_2}$.*

Proof. The first equation is proved by induction on ξ. The proof needs items (a) and (b) of Lemma 5.9 in the cases (a) and (b), respectively. Moreover, it needs properties P2, P5, and P7 of IO-substitution of the tree series. Then the second equation follows straightforwardly. □

We are ready to prove the following composition results.

Lemma 5.11 ([102], Lemma 2). *For every combination* x *over* $\{\mathrm{l}, \mathrm{n}\}$*:*

(A) $\mathrm{x\text{-}TOP}(S); \mathrm{x\text{-}BOT}(S) \subseteq \mathrm{x\text{-}WTT}(S)$.
(B) $\mathrm{x\text{-}TOP}(S); \mathrm{xl\text{-}BOT}(S) \subseteq \mathrm{x\text{-}TOP}^{\mathrm{R}}(S)$.

(C) x-TOP(S); ln-BOT$(S) \subseteq$ x-TOP(S).

Proof. For the proof of (A), let $\mathcal{M}_1$ be a top-down wtt and $\mathcal{M}_2$ a bottom-up wtt. By Lemma 5.10(b), we have $\tau_{\mathcal{M}_1}; \tau_{\mathcal{M}_2} = \tau_{\mathcal{M}_1 \circ \mathcal{M}_2}$. The preservation of properties x follows by Remark 5.8. Statements (B) and (C) are proved in the same way as (A) by using once again Remark 5.8. □

Now we turn to a decomposition result of WTT(S) that generalizes [48], Lemma 5.8.

Lemma 5.12 ([102], Lemma 1). *For every combination* x *over* $\{\mathrm{b}, \mathrm{l}\}$*:*

(A) x-WTT$(S) \subseteq$ xbn-HOM(S); x-BOT(S).
(B) x-TOP$^{\mathrm{R}}(S) \subseteq$ xbn-HOM(S); xl-BOT(S).

Proof. First, we prove (A). Let $\mathcal{M} = (Q, \Sigma, \Delta, S, \mu, F)$ be a wtt. We construct a homomorphism top-down wtt $\mathcal{M}_1$ and a bottom-up wtt $\mathcal{M}_2$ such that $\tau_{\mathcal{M}} = \tau_{\mathcal{M}_1}; \tau_{\mathcal{M}_2}$. Note that $\mathcal{M}_2$, being a bottom-up wtt, is allowed to make exactly one computation on every subtree. The idea behind the decomposition is that $\mathcal{M}_1$ copies the input subtrees so that $\mathcal{M}_2$ can simulate different computations of $\mathcal{M}$ on a subtree using different copies of that subtree. More exactly, let $mx = \mathsf{max}(\{1\} \cup \{|w|_{x_i} \mid 1 \leq i \leq k, \sigma \in \Sigma^{(k)}, (w,q) \in Q(X_k)^* \times Q, \mu_k(\sigma)_{w,q} \neq \widetilde{0}\})$, i.e., the maximal number of copies of a subtree taken by a rule, and consider the ranked alphabet $\Gamma = \{\sigma^{(k \cdot mx)} \mid k \geq 0, \sigma \in \Sigma^{(k)}\}$. Note that $mx = 1$ if $\mathcal{M}$ is linear. Now we construct $\mathcal{M}_1 = (\{\star\}, \Sigma, \Gamma, S, \mu^1, \{\star\})$ with

$$\mu^1_k(\sigma)\underbrace{\star(x_1)\ldots\star(x_1)}_{mx\ \text{times}}\ldots\underbrace{\star(x_k)\ldots\star(x_k)}_{mx\ \text{times}},\star = 1.\sigma(z_1,\ldots,z_{k\cdot mx})$$

for every $k \geq 0$ and $\sigma \in \Sigma^{(k)}$. Clearly, $\mathcal{M}_1$ is a Boolean and nondeleting homomorphism top-down wtt; moreover, $\mathcal{M}_1$ is linear and computes the identity if $\mathcal{M}$ is linear.

Then let $d \notin Q$ be a new state and $Q' = Q \cup \{d\}$. For every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, and $w \in Q(X_k)^*$ such that $|w|_{x_i} \leq mx$ for $1 \leq i \leq k$, construct the string $w' \in Q'(X_{k\cdot mx})^*$ in two steps as follows. First, construct $\widetilde{w} \in Q(X_{k\cdot mx})^*$ by replacing, for every $1 \leq i \leq k$, the jth occurrence of x_i in w by $x_{(i-1)\cdot mx+j}$. Note that $\widetilde{w}$ is linear in $X_{k\cdot mx}$. Then for every $1 \leq j \leq k \cdot mx$ such that $|\widetilde{w}|_{x_j} = 0$, append $d(x_j)$ to $\widetilde{w}$. Certainly, the string w' obtained in this way is linear and nondeleting in $X_{k\cdot mx}$. Then construct the wtt $\mathcal{M}' = (Q', \Gamma, \Delta, S, \mu', F)$, where μ' is defined as follows. For every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, let $\mu'_{k\cdot mx}(\sigma)_{d(x_1)\ldots d(x_{k\cdot mx}),d} = 1.\alpha$, where $\alpha \in \Delta^{(0)}$ is arbitrary; and, for every $(w,q) \in Q(X_k)^* \times Q$ such that $\mu_k(\sigma)_{w,q} \neq \widetilde{0}$, let $\mu'_{k\cdot mx}(\sigma)_{w',q} = \mu_k(\sigma)_{w,q}$. Every other entry of $\mu'_{k\cdot mx}(\sigma)$ is $\widetilde{0}$. Note that $\mathcal{M}'$ need not be a bottom-up wtt because there may be $\mu'_l(\sigma)_{w,q} \neq \widetilde{0}$ with $\sigma \in \Gamma^{(l)}$ such that the order of the variables in w is not $x_1, \ldots, x_l$. However, by an appropriate reordering of the symbols $q(x_i)$ in w and the corresponding substitution variables z_j in

$\mu'_l(\sigma)_{w,q}$, we can turn $\mathcal{M}'$ into a bottom-up wtt $\mathcal{M}_2 = (Q', \Gamma, \Delta, S, \mu^2, F)$ such that $\tau_{\mathcal{M}'} = \tau_{\mathcal{M}_2}$. The proof of $\tau_{\mathcal{M}} = \tau_{\mathcal{M}_1}; \tau_{\mathcal{M}_2}$ can be done as follows. We define $h : T_\Sigma \to T_\Gamma$ for every Σ-tree $\xi = \sigma(\xi_1, \ldots, \xi_k)$ by

$$h(\xi) = \sigma\big(\underbrace{h(\xi_1), \ldots, h(\xi_1)}_{mx \text{ times}}, \ldots, \underbrace{h(\xi_k), \ldots, h(\xi_k)}_{mx \text{ times}}\big).$$

Clearly, $\tau_{\mathcal{M}_1}(\xi) = 1.h(\xi)$. Thus, it is sufficient to prove that $h_\mu(\xi)_q = h_{\mu^2}(h(\xi))_q$ for every $\xi \in T_\Sigma$ and $q \in Q$. Moreover, it is obvious that $\mathcal{M}_2$ is Boolean (linear, resp.), whenever $\mathcal{M}$ is so. This finishes the proof of (A). Finally, if $\mathcal{M}$ is a top-down wtt with regular look-ahead, then $\mathcal{M}'$ is a linear top-down wtt with regular look-ahead, and consequently, $\mathcal{M}_2$ is a linear bottom-up wtt, which proves (B). $\square$

Putting Lemmata 5.11 and 5.12 together, we obtain the following characterizations of WTT(S) and TOP^R(S), which generalize the corresponding characterizations of gfst and top-down tree transducers with regular look-ahead that were obtained in [48], Theorems 5.10 and 5.15.

Theorem 5.13 ([102], Theorem 3). *For every combination* x *over* {l}*:*

(A) x-WTT(S) = xb-HOM(S); x-BOT(S).
(B) x-TOP$^{\mathrm{R}}$(S) = xb-HOM(S); xl-BOT(S).

By definition, xl-WTT(S) = xl-TOP$^{\mathrm{R}}$(S) for every combination x over {b, n}. Moreover, if the wtt $\mathcal{M}$ in the proof of Lemma 5.12(A) is linear, then the homomorphism top-down wtt $\mathcal{M}_1$ computes the identity. Hence, we obtain l-WTT(S) $\subseteq$ l-BOT(S) and bl-WTT(S) $\subseteq$ bl-BOT(S). These arguments verify the following characterization result, which generalizes [48], Theorem 5.13.

Theorem 5.14 ([102], Theorem 4). l-BOT(S) = l-TOP$^{\mathrm{R}}$(S) *and* bl-BOT(S) = bl-TOP$^{\mathrm{R}}$(S).

5.4.2 Results Concerning Bottom-up wtt

Here, we investigate composition results of the form C(S);D(S) $\subseteq$ E(S), where C(S), D(S), and E(S) are subclasses of BOT(S). First, we prove that BOT(S) is closed under right-composition with db-BOT(S). This generalizes the corresponding result for bottom-up tree transformations, cf. [48], Theorem 4.6 and [4], Theorem 6, and the result BOT(S); b-HOM(S) $\subseteq$ BOT(S) obtained in [53], Corollary 5.5.

Theorem 5.15 ([104], Theorem 24). *For every combination x over* {d, h, l, n} *we have* x-BOT(S); xdb-BOT(S) $\subseteq$ x-BOT(S).

Proof. The proof immediately follows from Remark 5.8 and Lemma 5.10(a) because, by adding a dummy state, each bottom-up wtt can be turned into a total one computing the same tree series transformation. $\square$

Next, we show a characterization of bottom-up wtt which generalizes Nivat's characterization of finite-state sequential machines [118], also cf. [6]. Note that Nivat's result was generalized for bottom-up tree transducers in [48]. To this end, we define finite-state relabeling bottom-up wtt in the way that we impose a further restriction on bottom-up wtt. A bottom-up wtt $\mathcal{M} = (Q, \Sigma, \Delta, S, \mu, F)$ is a *finite-state relabeling bottom-up wtt* if for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $w = q_1(x_1) \dots q_k(x_k) \in Q(X_k)^*$, and $q \in Q$, we have $\mathrm{supp}(\mu_k(\sigma)_{w,q}) \subseteq \{\delta(z_1, \dots, z_k) \mid \delta \in \Delta^{(k)}\}$. We note that a wta bottom-up wtt is a particular finite-state relabeling bottom-up wtt. The class of all tree series transformations over S which are computable by finite-state relabeling bottom-up wtt, is denoted by $\mathrm{QREL}(S)$. Note that $\mathrm{QREL}(S) \subseteq \text{ln-BOT}(S) = \text{ln-TOP}(S)$, cf. Theorem 5.5.

Theorem 5.16 ([53], Theorem 5.7). *For every combination x over $\{\mathrm{l}, \mathrm{n}\}$:*

(A) $\text{x-BOT}(S) = \mathrm{QREL}(S); \text{xb-HOM}(S)$.
(B) $\text{x-BOT}(S) = \text{xl-BOT}(S); \text{xb-HOM}(S)$.

Moreover:

(C) $\mathrm{BOT}(S) = \text{l-TOP}(S); \text{b-HOM}(S)$.

Proof. We first prove (A). By Theorem 5.15, the right-hand side of the equation is a subset of its left-hand side. In order to prove the other inclusion, let $\mathcal{M} = (Q, \Sigma, \Delta, S, \mu, F)$ be a bottom-up wtt. We will construct a finite-state relabeling bottom-up wtt $\mathcal{M}_1$ and a Boolean homomorphism bottom-up wtt $\mathcal{M}_2$, such that, up to renaming of states, $\mathcal{M}$ is the syntactic composition of $\mathcal{M}_1$ and $\mathcal{M}_2$. For this, define the ranked alphabet $\Omega = \bigcup_{\sigma \in \Sigma} \Omega_\sigma$, where, for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, we let $\Omega_\sigma = \{[\sigma, w, q, \zeta]^{(k)} \mid w \in Q(X_k)^*, q \in Q, \zeta \in \mathrm{supp}(\mu_k(\sigma)_{w,q})\}$. Obviously, Ω is a finite set. Now let $\mathcal{M}_1 = (Q, \Sigma, \Omega, S, \mu^1, F)$ be such that for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $w \in Q(X_k)^*$, $q \in Q$, and $\zeta' \in T_\Omega(Z_k)$,

$$\big(\mu^1_k(\sigma)_{w,q}, \zeta'\big) = \begin{cases} (\mu_k(\sigma)_{w,q}, \zeta) & \text{if } \zeta' = [\sigma, w, q, \zeta](z_1, \dots, z_k), \\ 0 & \text{otherwise.} \end{cases}$$

Let $\mathcal{M}_2 = (\{\star\}, \Omega, \Delta, S, \mu^2, \{\star\})$ be such that for every $k \geq 0$, $[\sigma, w, q, \zeta] \in \Omega^{(k)}$, and $\zeta' \in T_\Delta(Z_k)$, we have

$$\big(\mu^2_k([\sigma, w, q, \zeta])_{\star(x_1) \dots \star(x_k), \star}, \zeta'\big) = \begin{cases} 1 & \text{if } \zeta' = \zeta, \\ 0 & \text{otherwise.} \end{cases}$$

If $\mathcal{M}$ is nondeleting (resp., linear), then so is $\mathcal{M}_2$. Moreover, by identifying Q and $Q \times \{\star\}$, the bottom-up wtt $\mathcal{M}$ becomes the syntactic composition of $\mathcal{M}_1$ and $\mathcal{M}_2$. Thus, by Lemma 5.10(a), we have $\tau_\mathcal{M} = \tau_{\mathcal{M}_1}; \tau_{\mathcal{M}_2}$, which proves that the left-hand side of the equation is a subset of its right-hand side.

Now the inclusion $\subseteq$ for both (B) and (C) should be clear by (A) and the note we made above this theorem. Finally, by Theorem 5.15, the inclusion $\supseteq$ follows for (B), and by (B) and Theorem 5.14 the inclusion $\supseteq$ follows for (C). $\square$

Let us compare the equations of Theorem 5.13(B) and of Theorem 5.16(B) for x being the empty combination. We see that the classes $\mathrm{TOP}^{\mathrm{R}}(S)$ and $\mathrm{BOT}(S)$ can be characterized by the composition of the two classes l-BOT(S) and b-HOM(S) and that the difference between them lies in the order of their composition.

For further composition results on bottom-up wtt, we need another concept of syntactic composition. Let us explain why. For this, we consider two bottom-up wtt $\mathcal{M}_1 = (Q_1, \Sigma, \Delta, S, \mu^1, F_1)$ and $\mathcal{M}_2 = (Q_2, \Delta, \Gamma, S, \mu^2, F_2)$ and their composition $\mathcal{M}_1 \circ \mathcal{M}_2 = (Q_1 \times Q_2, \Sigma, \Gamma, S, \mu, F_1 \times F_2)$. We obtain the entry $\mu_k(\sigma)_{(p_1,q_1)(x_1)\dots(p_k,q_k)(x_k),(p,q)}$ by applying the homomorphism $h^{\overline{q}}_{\mu^2}$ to the entry $\mu^1_k(\sigma)_{p_1(x_1)\dots p_k(x_k),p}$, where $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $p, p_1, \dots, p_k \in Q_1$, $q \in Q_2$, and $\overline{q} \in Q_2^{Z_k}$ with $q_i = \overline{q}_{z_i}$ for $1 \leq i \leq k$, and then selecting the q-component from the resulting Q-vector. As Lemma 5.10(a) states, the syntactic composition yields equality on the level of the semantics also, provided that $\mathcal{M}_2$ is total, deterministic, and Boolean. However, the following problem arises in the case when $\mathcal{M}_2$ does not have these properties. Let us suppose that $\mathcal{M}_1$ translates a tree $\xi \in T_\Sigma$ into an output tree $\zeta \in T_\Delta$ with weight $a \in S$ and that, during the translation, it deletes the translation $\zeta' \in T_\Delta$ with weight $a' \in S$ of a subtree ξ' of ξ. Still, due to the definition of IO-substitution of tree series, the weight a' of ζ' contributes to the weight a of ζ, whereas ζ' does not contribute to ζ. Furthermore, let us suppose that $\mathcal{M}_2$ transforms ζ into $\widetilde{\zeta} \in T_\Gamma$ with weight $b \in S$ and ζ' into $\widetilde{\zeta}' \in T_\Gamma$ with weight $b' \in S$. Since the input of $\mathcal{M}_2$ is ζ, it does not process the deleted tree ζ', and thus the weight b' does not contribute to b. However, when $\mathcal{M}_1 \circ \mathcal{M}_2$ processes the input tree ξ, it transforms its subtree ξ' into ζ' with weight a' using the family of rule mappings μ^1, and immediately also transforms ζ' into $\widetilde{\zeta}'$ with weight b' using the family of rule mappings μ^2. Then although $\mathcal{M}_1 \circ \mathcal{M}_2$ deletes the translation $\widetilde{\zeta}'$ of ζ', both a' and b' still contribute to the weight of the overall translation $\widetilde{\zeta}$, which contrasts the situation encountered when $\mathcal{M}_1$ and $\mathcal{M}_2$ run separately. In the case that $\mathcal{M}_2$ is Boolean, the weight b' can only be 0 or 1, so that one just has to avoid the case that $b' = 0$. This can be achieved by requiring that $\mathcal{M}_2$ is total and deterministic; see Lemma 5.10(a). However, we do not want to restrict $\mathcal{M}_2$ and, therefore, following [104], we propose another construction. Namely, we manipulate $\mathcal{M}_2$ such that it has a state $\diamond$, called a *blind state*, which is not a final state and which transforms each input tree into an output tree $\alpha \in \Delta^{(0)}$ with weight 1. Then we compose $\mathcal{M}_1$ and $\mathcal{M}_2$ by processing in state $\diamond$ the subtrees that $\mathcal{M}_1$ deletes. We note that the concept of blind state was introduced in [48], Theorem 2.8 (called e for *erasing* there) in order to construct a linear bottom-up tree transducer from a linear top-down tree transducer; it occurred already in the proof of Lemma 5.12.

A state $\diamond \in Q$ of a bottom-up wtt $\mathcal{M} = (Q, \Sigma, \Delta, S, \mu, F)$ is a *blind state* if $\diamond \notin F$, there is an $\alpha \in \Delta^{(0)}$ such that $\mu_k(\sigma)_{\diamond(x_1)\dots\diamond(x_k),\diamond} = 1.\alpha$, and $\mu_k(\sigma)_{q_1(x_1)\dots q_k(x_k),\diamond} \neq \widetilde{0}$ implies that $q_i = \diamond$ for every $1 \leq i \leq k$ (for every

$k \geq 0$, $\sigma \in \Sigma^{(k)}$, and $q_1, \ldots, q_k \in Q$). It is easy to prove that $h_\mu(\xi)_\diamond = 1.\alpha$ for every $\xi \in T_\Sigma$. Moreover, for every bottom-up wtt $\mathcal{M} = (Q, \Sigma, \Delta, S, \mu, F)$, we can construct a bottom-up wtt $\mathcal{M}' = (Q', \Sigma, \Delta, S, \mu', F)$ with blind state such that $\tau_\mathcal{M} = \tau_{\mathcal{M}'}$ in the following way. Let $Q' = Q \cup \{\diamond\}$, for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, and $q, q_1, \ldots, q_k \in Q$, let $\mu'_k(\sigma)_{\diamond(x_1)\ldots\diamond(x_k),\diamond} = 1.\alpha$, $\mu'_k(\sigma)_{q_1(x_1)\ldots q_k(x_k),q} = \mu_k(\sigma)_{q_1(x_1)\ldots q_k(x_k),q}$, and let all remaining entries be $\widetilde{0}$.

Now we formalize this new concept of syntactic composition, called bottom-up syntactic composition (cf. [104], Definition 17, where it was called just composition). Let $\mathcal{M}_1 = (Q_1, \Sigma, \Delta, S, \mu^1, F_1)$ and $\mathcal{M}_2 = (Q_2, \Delta, \Gamma, S, \mu^2, F_2)$ be bottom-up wtt such that $\diamond$ is a blind state of $\mathcal{M}_2$. The *bottom-up syntactic composition of* $\mathcal{M}_1$ *and* $\mathcal{M}_2$, denoted by $\mathcal{M}_1 \circ_{\mathrm{bu}} \mathcal{M}_2$ is the bottom-up wtt $(Q_1 \times Q_2, \Sigma, \Gamma, S, \mu, F)$, where $F = F_1 \times F_2$, and for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $p, p_1, \ldots, p_k \in Q_1$, $q \in Q_2 \setminus \{\diamond\}$, and $\overline{q}, \overline{\diamond} \in Q_2^{Z_k}$ with $\overline{q}_{z_i} = q_i$ and $\overline{\diamond}_{z_i} = \diamond$ for $1 \leq i \leq k$, we have

$$\mu_k(\sigma)_{(p_1,q_1)(x_1)\ldots(p_k,q_k)(x_k),(p,q)} = h_{\mu^2}^{\overline{q}}\Bigg(\sum_{\substack{\zeta \in T_\Delta(Z_k), \\ (\forall 1 \leq i \leq k): z_i \notin \mathrm{var}(\zeta) \Leftrightarrow q_i = \diamond}} \big(\mu_k^1(\sigma)_{p_1(x_1)\ldots p_k(x_k),p}, \zeta\big).\zeta \Bigg)_q$$

and

$$\mu_k(\sigma)_{(p_1,\diamond)(x_1)\ldots(p_k,\diamond)(x_k),(p,\diamond)} = h_{\mu^2}^{\overline{\diamond}}\big(\mu_k^1(\sigma)_{p_1(x_1)\ldots p_k(x_k),p}\big)_\diamond.$$

All the remaining entries in μ are $\widetilde{0}$.

It should be clear that $\mathcal{M}_1 \circ_{\mathrm{bu}} \mathcal{M}_2$ does not always compute $\tau_{\mathcal{M}_1}; \tau_{\mathcal{M}_2}$ because the class of tree transformations computed by bottom-up tree transducers, i.e., by bottom-up wtt over $\mathbb{B}$ is not closed under composition; see [48], Theorem 2.5. However, the desired equality on the level of semantics holds for a linear $\mathcal{M}_1$, and we obtain the following generalization of [4], Theorem 6.

Lemma 5.17 ([104], Lemma 19). *Let* $\mathcal{M}_1$ *be a linear bottom-up wtt and* $\mathcal{M}_2$ *a bottom-up wtt with blind state. Then for every* $\xi \in T_\Sigma$, $p \in Q_1$, *and* $q \in Q_2$, *we have* $h_{\mu^2}(h_{\mu^1}(\xi)_p)_q = h_\mu(\xi)_{(p,q)}$ *and* $\tau_{\mathcal{M}_1 \circ_{\mathrm{bu}} \mathcal{M}_2} = \tau_{\mathcal{M}_1}; \tau_{\mathcal{M}_2}$.

Proof. Let $\diamond$ be the blind state of $\mathcal{M}_2$. The first equation can be proved by induction and case analysis, where the two cases are $q = \diamond$ and $q \neq \diamond$. In the first case, the proof uses the fact that $h_{\mu^2}(\zeta)_\diamond = 1.\alpha$ for every $\zeta \in \Delta$, and properties P5 and P7 of IO-substitution of tree series, while in the second case, it additionally uses Lemma 5.9 and property P2. Then the second equation follows straightforwardly from the first one. □

Now we can show that $\mathrm{BOT}(S)$ is closed under left-composition with l-$\mathrm{BOT}(S)$. This generalizes the corresponding result for bottom-up tree transformations, cf. [48], Theorem 4.5 and [4], Theorem 6.

Theorem 5.18 ([104], Theorem 20). *For every combination* x *over* $\{\mathrm{l}, \mathrm{n}\}$, *we have* $x\mathrm{l}\text{-}\mathrm{BOT}(S); x\text{-}\mathrm{BOT}(S) \subseteq x\text{-}\mathrm{BOT}(S)$.

Proof. Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be bottom-up wtt. We may assume without loss of generality that $\mathcal{M}_2$ has a blind state. Moreover, if $\mathcal{M}_1$ and $\mathcal{M}_2$ are linear (resp., nondeleting), then also $\mathcal{M}_1 \circ_{\mathrm{bu}} \mathcal{M}_2$ is linear (resp., nondeleting). Then the statement follows from Lemma 5.17. ☐

5.4.3 Results Concerning Top-down wtt

First, we show that $\mathrm{TOP}(S)$ is closed under right-composition with ln-$\mathrm{TOP}(S)$. This result generalizes [4], Theorem 1, in particular [4], Corollary 2(1).

Theorem 5.19 ([104], Theorem 26). *For every combination* x *over* $\{\mathrm{d}, \mathrm{l}, \mathrm{n}\}$, *we have* x-$\mathrm{TOP}(S)$; xln-$\mathrm{TOP}(S) \subseteq$ x-$\mathrm{TOP}(S)$.

Proof. Let $\mathcal{M}_1$ be a top-down wtt and $\mathcal{M}_2$ a linear and nondeleting top-down wtt. By Theorem 5.5, there is a linear and nondeleting bottom-up wtt $\mathcal{M}_2'$ such that $\tau_{\mathcal{M}_2} = \tau_{\mathcal{M}_2'}$. By Lemma 5.10(b), we have $\tau_{\mathcal{M}_1} ; \tau_{\mathcal{M}_2'} = \tau_{\mathcal{M}_1 \circ \mathcal{M}_2'}$. Since $\mathcal{M}_2'$ is linear and nondeleting, $\mathcal{M}_1 \circ \mathcal{M}_2'$ is a top-down wtt, cf. Remark 5.8. If $\mathcal{M}_1$ and $\mathcal{M}_2$ have property x, then $\mathcal{M}_1 \circ \mathcal{M}_2'$ has property x. ☐

One would expect from [4], Theorem 1, in particular [4], Corollary 3(2), that in the case $\mathrm{x} = \mathrm{d}$ the linearity of $\mathcal{M}_2$ can be dropped. However, as shown in [53], Example 5.11, this is not the case.

The next result generalizes another case of [4], Theorem 1, viz. the case that $\mathcal{M}_1$ is total and deterministic, cf. [4], Corollary 2(4). However, linearity of $\mathcal{M}_2$ is still needed (for the same reason), and moreover, $\mathcal{M}_1$ must be Boolean as shown in [53], Example 5.12. It will be proved in Theorem 5.25 that $\mathcal{M}_2$ need not be linear if it is deterministic.

Theorem 5.20 ([104], Theorem 30). tdb-$\mathrm{TOP}(S)$; l-$\mathrm{TOP}(S) \subseteq \mathrm{TOP}(S)$.

Proof. Let $\mathcal{N}_1 = (Q_1, \Sigma, \Delta, S, \mu^1, \{q_1\})$ be a total, deterministic, and Boolean top-down wtt and $\mathcal{M} = (Q, \Delta, \Gamma, S, \mu, F)$ a linear top-down wtt. Let us apply the proof of Lemma 5.12(B) to $\mathcal{M}$. Since $\mathcal{M}$ is linear, the homomorphism top-down wtt $\mathcal{M}_1$ of that proof computes the identity, and so we obtain a linear bottom-up wtt $\mathcal{M}_2 = (Q_2, \Delta, \Gamma, S, \mu^2, F)$ such that $\tau_{\mathcal{M}} = \tau_{\mathcal{M}_2}$. Note that $Q_2 = Q \cup \{d\}$ and d is a blind state. Now let $\mathcal{M}_3 = (Q_1 \times Q_2, \Sigma, \Gamma, S, \mu, \{q_1\} \times F)$ be the syntactic composition of $\mathcal{N}_1$ and $\mathcal{M}_2$, i.e., $\mathcal{M}_3 = \mathcal{N}_1 \circ \mathcal{M}_2$. By Lemma 5.10(b), we have $\tau_{\mathcal{M}_3} = \tau_{\mathcal{N}_1} ; \tau_{\mathcal{M}_2}$. Moreover, by Remark 5.8, $\mathcal{M}_3$ is a top-down wtt with regular look-ahead. Since $\mathcal{N}_1$ is total, deterministic, and Boolean, and since $\mathcal{M}_2$ is linear and d is a blind state of $\mathcal{M}_2$, the wtt $\mathcal{M}_3$ has the following properties:

(i) There is an $\alpha \in \Gamma^{(0)}$, such that $h_\mu(\xi)_{(p,d)} = 1.\alpha$ for every $\xi \in T_\Sigma$ and $p \in Q_1$.
(ii) For every $(p, q) \in Q_1 \times Q_2$, $k \geq 0$, $w = (p_1, q_1)(x_{i_1}) \dots (p_n, q_n)(x_{i_n}) \in (Q_1 \times Q_2)(X_k)^*$, $1 \leq j \leq n$, $\sigma \in \Sigma^{(k)}$, and $\zeta \in \mathrm{supp}(\mu_k(\sigma)_{w,(p,q)})$, we have $z_j \notin \mathrm{var}(\zeta)$ iff $q_j = d$.

This means that the look-ahead is trivial, and thus $\mathcal{M}_3$ can be transformed into an equivalent top-down wtt $\mathcal{M}_3'$ in the following way. For a tree $\zeta \in T_\Gamma(Z_n)$, which is linear in Z_n, let us denote by $\mathrm{norm}_n(\zeta)$ the linear and nondeleting tree in $T_\Gamma(Z_k)$ defined by $\mathrm{norm}_n(\zeta) = \zeta(\varphi(z_i) \mid 1 \leq i \leq n)$, where $\mathrm{var}(\zeta) = \{z_{i_1}, \ldots, z_{i_k}\}$ with $i_1 < \cdots < i_k$ and φ is any mapping $Z_n \to Z_k$ such that $\varphi(z_{i_j}) = z_j$ for all $1 \leq j \leq k$. Moreover, for a string $w \in (Q_1 \times Q_2)(X_k)^*$, let us denote by $\mathrm{del}(w)$ the string which is obtained from w by deleting all symbols of the form $(p, d)(x_i)$ from w. Now $\mathcal{M}_3'$ is obtained from $\mathcal{M}_3$ by changing μ to μ' in the following way: for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $w' \in (Q_1 \times Q_2)(X_k)^*$, and $(p, q) \in Q_1 \times Q_2$,

$$\mu_k'(\sigma)_{w',(p,q)} = \sum_{\substack{w \in (Q_1 \times Q_2)(X_k)^* \\ \mathrm{del}(w) = w'}} \Bigg(\sum_{\zeta \in T_\Gamma(Z_{|w|})} \big(\mu_k(\sigma)_{w,(p,q)}, \zeta\big) \mathrm{norm}_{|w|}(\zeta) \Bigg).$$

Then we can prove that for every $\xi \in T_\Sigma$ and $(p, q) \in Q_1 \times Q_2$ such that $q \neq d$, we have $h_{\mu'}(\xi)_{(p,q)} = h_\mu(\xi)_{(p,q)}$. The proof is performed by induction on ξ, using properties (i) and (ii) of $\mathcal{M}_3$ and P2 of IO-substitution of tree series. Thus, $\tau_{\mathcal{M}_3} = \tau_{\mathcal{M}_3'}$ follows, which finishes the proof. □

Now we generalize [48], Theorem 3.6 and show that $\mathrm{TOP}(S)$ can be decomposed into bn-$\mathrm{HOM}(S)$ and l-$\mathrm{TOP}(S)$.

Lemma 5.21 ([53], Lemma 5.9). *For every combination* x *over* $\{\mathrm{t}, \mathrm{d}\}$*, we have* x-$\mathrm{TOP}(S) \subseteq$ bn-$\mathrm{HOM}(S)$; xl-$\mathrm{TOP}(S)$.

Proof. Let $\mathcal{M} = (Q, \Sigma, \Delta, S, \mu, F)$ be a top-down wtt. We construct a Boolean and nondeleting homomorphism top-down wtt $\mathcal{M}_1$ and a linear top-down wtt $\mathcal{M}_2$ such that $\tau_\mathcal{M} = \tau_{\mathcal{M}_1}; \tau_{\mathcal{M}_2}$. The proof is very similar to that of Lemma 5.12. In fact, $\mathcal{M}_1$ is constructed in the same way as in that proof and we construct $\mathcal{M}_2$ similarly to $\mathcal{M}'$. The main difference is that we do not need the extra state d because we do not need to force $\mathcal{M}_2$ to be a bottom-up wtt. Thus, the family of rule mappings underlying $\mathcal{M}_2$ slightly differs from that of $\mathcal{M}'$. Let $\mathcal{M}_2 = (Q, \Gamma, \Delta, S, \mu^2, F)$, and for every $(w, q) \in Q(X_k)^* \times Q$ such that $\mu_k(\sigma)_{w,q} \neq \widetilde{0}$, let $\mu^2_{k \cdot mx}(\sigma)_{\widetilde{w},q} = \mu_k(\sigma)_{w,q}$, where the string $\widetilde{w}$ is being defined as in the proof of Lemma 5.12. Every other entry of $\mu^2_{k \cdot mx}(\sigma)$ is $\widetilde{0}$. It should be clear that $\mathcal{M}_2$ is a linear top-down wtt. The proof of $\tau_\mathcal{M} = \tau_{\mathcal{M}_1}; \tau_{\mathcal{M}_2}$ can be done in the same way as in the proof of Lemma 5.12. Moreover, it is obvious that $\mathcal{M}_2$ inherits the properties x from $\mathcal{M}$. □

Next, we turn to further composition results for top-down wtt. For this, we define the top-down syntactic composition of two top-down wtt $\mathcal{M}_1$ and $\mathcal{M}_2$, denoted by $\mathcal{M}_1 \circ_{\mathrm{td}} \mathcal{M}_2$, thereby generalizing the corresponding concept defined for top-down tree transducers in [4], page 195. The wtt $\mathcal{M}_1 \circ_{\mathrm{td}} \mathcal{M}_2$ is obtained, as for wtt, by letting $\mathcal{M}_2$ work on the pieces of output produced by $\mathcal{M}_1$. In order to avoid too complex formulas, we consider only the case that $\mathcal{M}_1$ is deterministic, cf. [53], Definitions 5.13 and 5.14; [61], Definitions 5.2 and 5.3.

The pieces of output produced by $\mathcal{M}_1$ may contain variables from Z, which $\mathcal{M}_2$ cannot process. Therefore, just as in the case of the syntactic composition of bottom-up wtt, we first extend a top-down wtt $\mathcal{M} = (Q, \Sigma, \Delta, S, \mu, F)$ such that it can process input trees containing variables from Z_n for some $n \geq 0$. We define the $(\Sigma \cup Z_n)$-algebra $(S\langle T_\Delta(Q(Z_n))\rangle^Q, \mu_\mathcal{M}^{(n)})$ where every $z_j \in Z_n$ is a nullary symbol and $\mu_\mathcal{M}^{(n)}(z_j)()_q = 1.q(z_j)$ for every $q \in Q$. For every $k \geq 0$ and $\sigma \in \Sigma^{(k)}$, the k-ary operation $\mu_\mathcal{M}^{(n)}(\sigma) : S\langle T_\Delta(Q(Z_n))\rangle^Q \times \cdots \times S\langle T_\Delta(Q(Z_n))\rangle^Q \to S\langle T_\Delta(Q(Z_n))\rangle^Q$ is defined in the same way as $\mu_\mathcal{M}(\sigma)$. Let us denote the unique $(\Sigma \cup Z_n)$-algebra homomorphism from $T_{\Sigma \cup Z_n}$ to $S\langle T_\Delta(Q(Z_n))\rangle^Q$ by $h_\mu^{(n)}$.

For the definition of the family of rule mappings μ of $\mathcal{M}_1 \circ_{\text{td}} \mathcal{M}_2$, we need two more technical concepts: (a) substitution of particular strings and (b) linearization of a tree. For (a), let Q_1 and Q_2 be finite nonempty sets (e.g., state sets of top-down wtt), $k \geq 0$, $w \in Q_1(X_k)^*$ with $|w| = l$, and let $u \in Q_2(Z_l)^*$. Then $u\langle w\rangle$ is the string in $(Q_1 \times Q_2)(X_k)^*$ which is obtained from u by replacing, for every $q \in Q_2$ and $z_i \in Z_l$, the expression $q(z_i)$ by $(p, q)(x_j)$ where $p(x_j)$ is the ith symbol of w.

Let us now turn to (b). For a ranked alphabet Γ, the set of all trees in $T_\Gamma(Z_m)$, $m \geq 0$, which are both linear and nondeleting in Z_m and in which the order of the variables is $z_1, \ldots, z_m$, is denoted by $C_\Gamma^{(m)}$. Let $\xi \in T_\Gamma(H)$, where H is a set. The *linearization of* ξ *with respect to* H, denoted by $\text{lin}_H(\xi)$, is defined as the unique pair (ξ', u) where $\xi' \in C_\Gamma^{(m)}$ and $u = a_1 \ldots a_m \in H^*$ such that $\xi = \xi'(a_1, \ldots, a_m)$.

Now the *top-down syntactic composition* of a deterministic top-down wtt $\mathcal{M}_1 = (Q_1, \Sigma, \Delta, S, \mu^1, \{q_1\})$ and a top-down wtt $\mathcal{M}_2 = (Q_2, \Delta, \Gamma, S, \mu^2, F_2)$ is the top-down wtt $\mathcal{M}_1 \circ_{\text{td}} \mathcal{M}_2 = (Q_1 \times Q_2, \Sigma, \Gamma, S, \mu, \{q_1\} \times F_2)$, where the family of rule mappings μ is defined as shown in Fig. 2.

Using similar arguments as for the syntactic composition of a wtt with a bottom-up wtt, we will show that, under certain conditions, $\mathcal{M}_1 \circ_{\text{td}} \mathcal{M}_2$ computes $\tau_{\mathcal{M}_1}; \tau_{\mathcal{M}_2}$. For this, we first formulate a property of top-down wtt, namely, that h_μ distributes over substitutions $\xi(\xi_1, \ldots, \xi_l)$ for $\xi \in T_\Delta(Z_l)$ and $\xi_1, \ldots, \xi_l \in T_\Delta$. Note that this property corresponds to the one of (restricted) bottom-up wtt formulated in Lemma 5.9.

Lemma 5.22 ([61], Statement in the proof of Lemma 5.5). *Let* $\mathcal{M} = (Q, \Delta, \Gamma, S, \mu, F)$ *be a top-down wtt and* $l \geq 0$. *For every* $q \in Q$, $\xi \in T_\Delta(Z_l)$, *and* $\xi_1, \ldots, \xi_l \in T_\Delta$,

$$h_\mu\big(\xi(\xi_1, \ldots, \xi_l)\big)_q = \sum_{\kappa=1}^{r} a_\kappa \zeta_\kappa \leftarrow \big(h_\mu(\xi_{i_{\kappa,1}})_{q_{\kappa,1}}, \ldots, h_\mu(\xi_{i_{\kappa,m_\kappa}})_{q_{\kappa,m_\kappa}}\big),$$

where $h_\mu^{(l)}(\xi)_q = a_1.\hat{\zeta}_1 + \cdots + a_r.\hat{\zeta}_r$ *for* $a_1, \ldots, a_r \in S \setminus \{0\}$ *and* $\hat{\zeta}_1, \ldots, \hat{\zeta}_r \in T_\Gamma(Q(Z_l))$, *and* $\text{lin}_{Q(Z_l)}(\hat{\zeta}_\kappa) = (\zeta_\kappa, q_{\kappa,1}(z_{i_{\kappa,1}}) \ldots q_{\kappa,m_\kappa}(z_{i_{\kappa,m_\kappa}}))$, $\zeta_\kappa \in C_\Gamma^{(m_\kappa)}$ *for every* $1 \leq \kappa \leq r$.

For every $\sigma \in \Sigma^{(k)}$ with $k \geq 0$, for every $(w,p) \in Q_1(X_k)^* \times Q_1$ with $l = \|w\|$, if $\mu^1_k(\sigma)_{w,p} = a.\zeta$ for $a \in S \setminus \{0\}$ and $\zeta \in T_\Delta(Z_l)$, then { for every $q \in Q_2$, if $h^{(l)}_{\mu^2}(\zeta)_q = a_1.\hat{\zeta}_1 + \cdots + a_r.\hat{\zeta}_r$ for $a_1,\ldots,a_r \in S \setminus \{0\}$ and $\hat{\zeta}_1,\ldots,\hat{\zeta}_r \in T_\Gamma(Q_2(Z_l))$ (by P4 also $h^{(l)}_{\mu^2}(\zeta)$ is polynomial), then define, for every $1 \leq j \leq r$, $\mu_k(\sigma)_{v_j,(p,q)} = \sum_{1 \leq i \leq r, v_i = v_j} (a \cdot a_i).\zeta_i$, where, for every $1 \leq \kappa \leq r$, $\mathrm{lin}_{Q_2(Z_l)}(\hat{\zeta}_\kappa\ ) = (\zeta_\kappa, u_\kappa)$, $\zeta_\kappa \in C^{(m_\kappa)}_\Gamma$, $u_\kappa \in Q_2(Z_l)^*$, $\|u_\kappa\| = m_\kappa$ and $v_\kappa = u_\kappa\langle w\rangle$ }. Moreover, for every $\sigma \in \Sigma^{(k)}$ with $k \geq 0$, $p \in Q_1$, $q \in Q_2$ and $v \in (Q_1 \times Q_2)(X_k)^*$ not defined by the above conditions, let $\mu_k(\sigma)_{v,(p,q)} = \tilde{0}$.

Fig. 2. Definition of μ

Proof. The proof is performed by induction on ξ and it needs properties P5, P7, and a version of P8 which also assures associativity of tree series substitution, cf. [61], Corollary 2.6. □

The following sufficient conditions guarantee that $\mathcal{M}_1 \circ_{\mathrm{td}} \mathcal{M}_2$ computes $\tau_{\mathcal{M}_1};\tau_{\mathcal{M}_2}$, cf. Lemma 5.10.

Lemma 5.23 ([61], Lemma 5.5; [53], Lemma 5.17). *Let $\mathcal{M}_1$ be a total, deterministic, and Boolean top-down wtt and $\mathcal{M}_2$ a top-down wtt. If (a) $\mathcal{M}_1$ is a homomorphism top-down wtt or (b) $\mathcal{M}_2$ is deterministic, then for every $\xi \in T_\Sigma$, $p \in Q_1$, and $q \in Q_2$, we have $h_{\mu^2}(h_{\mu^1}(\xi)_p)_q = h_\mu(\xi)_{(p,q)}$ and $\tau_{\mathcal{M}_1 \circ_{\mathrm{td}} \mathcal{M}_2} = \tau_{\mathcal{M}_1};\tau_{\mathcal{M}_2}$.*

Proof. The first equation is proved by induction on ξ. The proof needs Lemma 5.22 and properties P5, P7, and P8 of IO-substitution of tree series. Then the second equation follows straightforwardly. □

The following theorem generalizes [48], Theorem 3.7.

Theorem 5.24 ([53], Theorem 5.18). *For every x over* $\{\mathrm{t},\mathrm{d}\}$, *we have* x-TOP$(S)$ = b-HOM(S); xl-TOP(S).

Proof. It follows from Lemma 5.21 and Lemma 5.23(a). □

By comparing the above equation and Theorem 5.16(C) for x being the empty combination, we observe that TOP(S) and BOT(S) can be characterized by the composition of the two classes b-HOM(S) and l-TOP(S). However, the orders of the subclasses in the two compositions are different.

Next, we generalize [133], Lemma 6.9 and [4], Corollary 3(3) (note that in that corollary the second *PDT* should be *DT*).

Theorem 5.25 ([53], Theorem 5.18). *For every x over* $\{\mathrm{t}, \mathrm{l}, \mathrm{n}\}$*, we have* xtdb-TOP$(S)$; xd-TOP$(S) \subseteq$ xd-TOP(S).

Proof. It immediately follows from Lemma 5.23(b) and the fact that the top-down syntactic composition preserves the properties d, t, l, and n. □

5.5 The Inclusion Diagram of Some Fundamental wtt Classes

By the *inclusion diagram* of certain classes, we mean their Hasse diagram with respect to the partial order $\subseteq$, cf. [64], Sect. 2.2. In this subsection, we will be interested in the inclusion diagram of the classes WTT(S), TOP$^{\mathrm{R}}(S)$, TOP(S), and BOT(S) and their linear, and linear and nondeleting subclasses, where S is a proper semiring (altogether 12 classes of tree series transformations). We note that in [105] the same inclusion diagram was obtained for a positive semiring S. Moreover, by [107], the results of [105] can easily be generalized to the more general case that S is a proper semiring.

We already know that l-WTT(S) = l-TOP$^{\mathrm{R}}(S)$ and ln-WTT(S) = ln-TOP$^{\mathrm{R}}(S)$ = ln-TOP(S) by definition. Moreover, ln-TOP(S) = ln-BOT(S) and l-BOT(S) = l-TOP$^{\mathrm{R}}(S)$ by Theorems 5.5 and 5.14, respectively.

In Fig. 3, we visualize all the equalities and the inclusions among the involved 12 classes. In the rest of this subsection, we show that all inclusions are proper and that the unrelated classes are incomparable provided S is a proper semiring. For this, it is sufficient to verify the following four inequalities:

$$\mathrm{TOP}(S) \setminus \mathrm{BOT}(S) \neq \emptyset, \tag{9}$$
$$\mathrm{BOT}(S) \setminus \mathrm{TOP}^{\mathrm{R}}(S) \neq \emptyset, \tag{10}$$
$$\text{l-BOT}(S) \setminus \mathrm{TOP}(S) \neq \emptyset, \tag{11}$$
$$\text{l-TOP}(S) \setminus \text{ln-TOP}(S) \neq \emptyset. \tag{12}$$

First, we show that the above inequalities hold for $S = \mathbb{B}$ (and hence in this particular case the diagram in Fig. 3 is an inclusion diagram). Since tree transducers and wtt over $\mathbb{B}$ can be identified, cf. Sect. 5.2 and Theorem 5.6 for $S = \mathbb{B}$, we refer to the corresponding results in the theory of tree transducers. In fact, (9) follows from [48], Theorem 2.3, while (10) follows from [49], Corollary 2.4(1), and (11) from [48], Example 2.6. Finally, (12) is trivial: no nondeleting top-down tree transducer can translate, e.g., $\sigma(\alpha, \beta)$ to α.

Now we will lift these inequalities to every semiring S that is proper. For this, however, we need some preparation, cf. the end of Sect. 3.3. Let S' be another semiring and consider a mapping $f : S \to S'$. For every tree series transformation $\tau : T_\Sigma \to S\langle T_\Delta \rangle$, we define $f(\tau) : T_\Sigma \to S'\langle T_\Delta \rangle$ such that for every $\xi \in T_\Sigma$ we have $f(\tau)(\xi) = f \circ \tau(\xi)$. Moreover, for every wtt $\mathcal{M} = (Q, \Sigma, \Delta, S, \mu, F)$, we define the wtt $f(\mathcal{M}) = (Q, \Sigma, \Delta, S', \mu', F)$ over S' such that $\mu'_k(\sigma)_{w,q} = f(\mu_k(\sigma)_{w,q})$ for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, and $(w, q) \in Q(X_k)^* \times Q$. Then we can prove the following two statements, cf. Theorem 3.9.

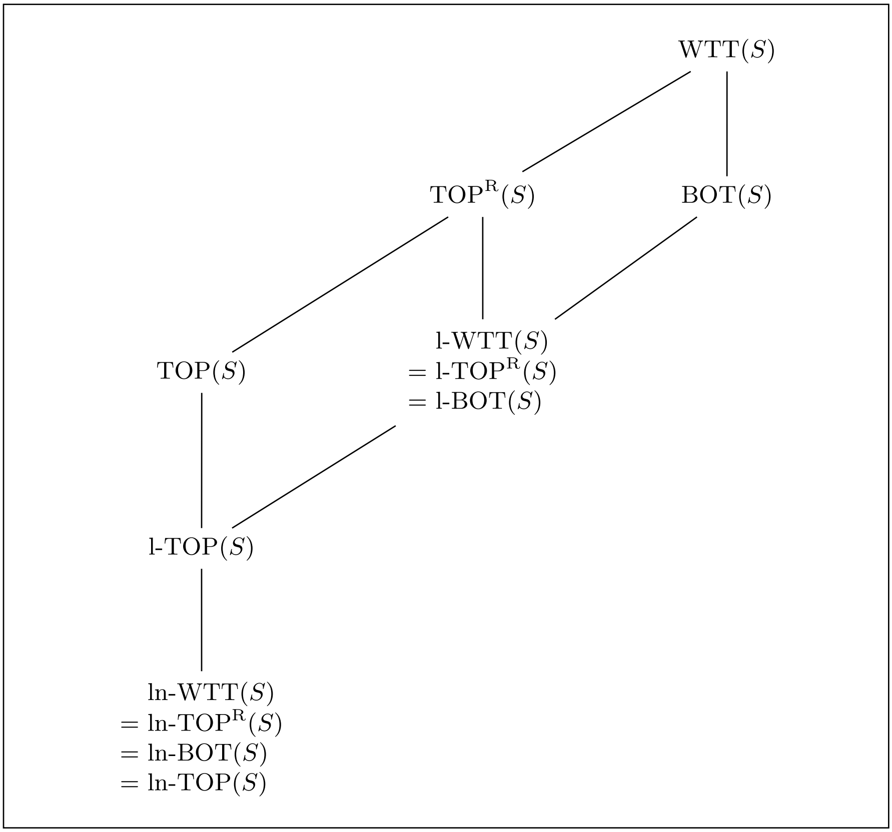

Fig. 3. The inclusion diagram of the classes $\mathrm{WTT}(S)$, $\mathrm{TOP}^{\mathrm{R}}(S)$, $\mathrm{TOP}(S)$, and $\mathrm{BOT}(S)$ and their linear, and linear and nondeleting subclasses, provided S is a proper semiring

Lemma 5.26 ([105], Lemma 2). *Let $\mathcal{M}$ be a wtt of any of the types in Fig. 3 and $f : S \to S'$ be a mapping such that $f(0) = 0$.*

(A) $f(\mathcal{M})$ is of the same type as $\mathcal{M}$.
(B) If f is a semiring homomorphism, then $\tau_{f(\mathcal{M})} = f(\tau_{\mathcal{M}})$.

Proof. Statement (A) can be proved by a direct inspection of each case. For statement (B), it can easily be shown by induction that for every $\xi \in T_\Sigma$, $\eta \in T_\Delta$, and $q \in Q$, we have $(h_{\mu'}(\xi)_q, \eta) = f((h_\mu(\xi)_q, \eta))$. Then $\tau_{f(\mathcal{M})} = f(\tau_{\mathcal{M}})$ follows easily. $\square$

Now let $f : S \to S'$ be a semiring homomorphism and $\mathrm{C}(S)$ any of the tree series transformation classes in Fig. 3. By Lemma 5.26, $f(\mathrm{C}(S)) \subseteq \mathrm{C}(S')$, where, of course, $f(\mathrm{C}(S))$ denotes $\{f(\tau) \mid \tau \in \mathrm{C}(S)\}$. Now we can prove the following lemma.

Lemma 5.27 ([105], Lemma 2). *If $f : S \to S'$ is a surjective semiring homomorphism and $\mathrm{C}(S)$ is any of the classes in Fig. 3, then $f(\mathrm{C}(S)) = \mathrm{C}(S')$.*

Proof. Let $\mathcal{M} = (Q, \Sigma, \Delta, S', \mu, F)$ be a wtt of one of the types in Fig. 3. Define a mapping $g : S' \to S$ such that $g(0) = 0$ and for every $a \in S'$, we have $f(g(a)) = a$, this is possible because f is surjective. Then by Lemma 5.26(A), the wtt $g(\mathcal{M})$ is over S and of the same type as $\mathcal{M}$. Moreover, $f(g(\mathcal{M}))$ and $\mathcal{M}$ are syntactically the same, i.e., $f(g(\mathcal{M})) = \mathcal{M}$. Thus, by Lemma 5.26(B), we have $f(\tau_{g(\mathcal{M})}) = \tau_{f(g(\mathcal{M}))} = \tau_{\mathcal{M}}$. □

An easy computation verifies that for tree series transformations $\tau : T_\Sigma \to S\langle T_\Delta\rangle$ and $\tau' : T_\Delta \to S\langle T_\Gamma\rangle$ and semiring homomorphism $f : S \to S'$, we have $f(\tau;\tau') = f(\tau); f(\tau')$ (cf. [105], Lemma 3), and hence, $f(\mathrm{C}(S);\mathrm{D}(S)) = f(\mathrm{C}(S)); f(\mathrm{D}(S))$. Moreover, for $n \geq 1$, we denote the n-fold composition $\mathrm{C}(S);\ldots;\mathrm{C}(S)$ by $\mathrm{C}(S)^n$. Thus, $f(\mathrm{C}(S)^n) = f(\mathrm{C}(S))^n$ for every $n \geq 1$.

Lemma 5.28. *If S is a proper semiring, then for every two classes $\mathrm{C}(S)$ and $\mathrm{D}(S)$ in Fig. 3 and $m, n \geq 1$, the inequality $\mathrm{C}(\mathbb{B})^m \setminus \mathrm{D}(\mathbb{B})^n \neq \emptyset$ implies $\mathrm{C}(S)^m \setminus \mathrm{D}(S)^n \neq \emptyset$.*

Proof. We use a proof by contraposition. Assume $\mathrm{C}(S)^m \subseteq \mathrm{D}(S)^n$. By [137], Theorem 2.1, there is a (surjective) semiring homomorphism $f : S \to \mathbb{B}$. For this f, we have $f(\mathrm{C}(S)^m) \subseteq f(\mathrm{D}(S)^n)$. Furthermore, $f(\mathrm{C}(S)^m) = f(\mathrm{C}(S))^m$, which equals $\mathrm{C}(\mathbb{B})^m$ by Lemma 5.27. In the same way, we get $f(\mathrm{D}(S)^n) = \mathrm{D}(\mathbb{B})^n$. This implies $\mathrm{C}(\mathbb{B})^m \subseteq \mathrm{D}(\mathbb{B})^n$. □

Now we can state the main result of this subsection.

Theorem 5.29 ([105], Theorem 3). *If S is a proper semiring, then the diagram in Fig. 3 is the inclusion diagram of the depicted classes of tree series transformations.*

Proof. We saw that the inequalities (9)–(12) hold for $S = \mathbb{B}$. Then by Lemma 5.28, they also hold for every proper semiring S. □

5.6 Hierarchies

A *hierarchy* is a family $(K_n \mid n \geq 1)$, where K_n is a class such that $K_n \subseteq K_{n+1}$ for every $n \geq 1$. Recall that $\mathrm{C}(S)^n$ denotes the n-fold composition of a tree series transformation class $\mathrm{C}(S)$. Then for every class $\mathrm{C}(S)$ which we consider in this chapter, $(\mathrm{C}(S)^n \mid n \geq 1)$ is a hierarchy because $\mathrm{C}(S)$ contains the identity ι. In this subsection, we present the inclusion diagram consisting of the hierarchies $(\mathrm{TOP}(S)^n \mid n \geq 1)$ and $(\mathrm{BOT}(S)^n \mid n \geq 1)$, where S is a proper semiring. Hereby, we generalize the inclusion results concerning the n-fold compositions of the classes of top-down tree transformations and of bottom-up tree transformations, called hierarchy results, cf. [4], Theorem 13; [50], Theorem 3.14; and [68], Sect. 8. of Chap. IV. We note that such a generalization was made in [61] for top-down and bottom-up wtt over commutative, idempotent, and positive semirings, then it was shown in [105] that the idempotency is not necessary. By [107], these results hold even for proper semirings.

First, we show the generalization of [4], Theorem 13.

Lemma 5.30 ([61], Theorems 5.1 and 5.7). *For every $n \geq 1$, we have:*

(A) $\mathrm{TOP}(S)^n \subseteq \mathrm{BOT}(S)^{n+1}$.
(B) $\mathrm{BOT}(S)^n \subseteq \mathrm{TOP}(S)^{n+1}$.

Proof. Let us write T, B, and H for $\mathrm{TOP}(S)$, $\mathrm{BOT}(S)$, and $\mathrm{HOM}(S)$ for the sake of readability. Then using Theorems 5.24 and 5.16(C), we can compute as follows.

$$\begin{aligned}
\mathrm{T}^n &\subseteq \text{l-T}; \mathrm{T}^n; \text{b-H} = \text{l-T}; (\text{b-H}; \text{l-T})^n; \text{b-H} \\
&= (\text{l-T}; \text{b-H})^{n+1} = \mathrm{B}^{n+1}, \\
\mathrm{B}^n &\subseteq \text{b-H}; \mathrm{B}^n; \text{l-T} = \text{b-H}; (\text{l-T}; \text{b-H})^n; \text{l-T} \\
&= (\text{b-H}; \text{l-T})^{n+1} = \mathrm{T}^{n+1}.
\end{aligned}$$ □

In Fig. 4, we visualize the inclusions among the involved classes $\mathrm{TOP}(S)^n$ and $\mathrm{BOT}(S)^n$, $n \geq 1$. In the rest of this subsection, we show that all inclusions are proper and that the unrelated classes are incomparable provided S is a proper semiring. For this, it suffices to verify the following two inequalities for every $n \geq 1$:

$$\mathrm{TOP}(S)^n \setminus \mathrm{BOT}(S)^n \neq \emptyset, \tag{13}$$
$$\mathrm{BOT}(S)^n \setminus \mathrm{TOP}(S)^n \neq \emptyset. \tag{14}$$

Lemma 5.31 ([52]). *If $\mathrm{TOP}(S)^n \subset \mathrm{TOP}(S)^{n+1}$ for every $n \geq 1$, then both (13) and (14) hold.*

Proof. We again use the abbreviations T, B, and H introduced above. We first prove (13). For this, assume the opposite, i.e., that $\mathrm{T}^n \subseteq \mathrm{B}^n$. Then we obtain

$$\begin{aligned}
\mathrm{T}^{n+2} &= \text{b-H}; \text{l-T}; \mathrm{T}^n; \text{b-H}; \text{l-T} \subseteq \text{b-H}; \text{l-B}; \mathrm{B}^n; \text{b-H}; \text{l-T} \\
&\subseteq \text{b-H}; \mathrm{B}^n; \text{l-T} \qquad\qquad = \mathrm{T}^{n+1},
\end{aligned}$$

which contradicts the assumption of the lemma. In the first three steps of the computation, we used Theorem 5.24, Theorem 5.14, and Theorems 5.15 and 5.18, while the equality in the last step comes from the second computation in the proof of Lemma 5.30.

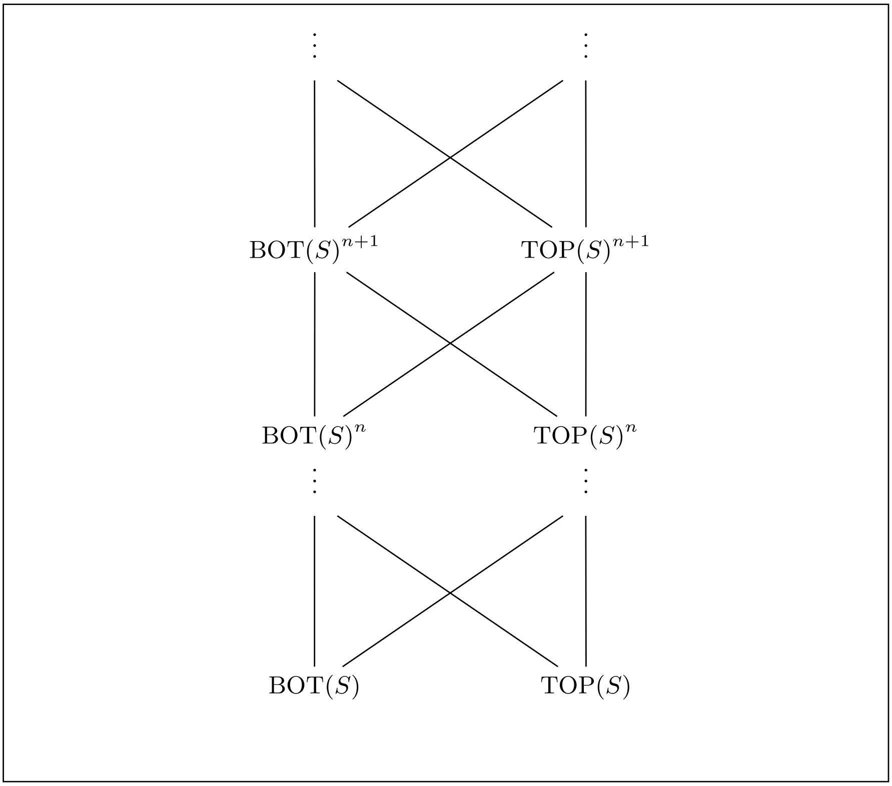

Fig. 4. The inclusion diagram of the classes $\mathrm{TOP}(S)^n$ and $\mathrm{BOT}(S)^n$, where $n \geq 1$ and S is a proper semiring

To prove (14), assume that $\mathrm{B}^n \subseteq \mathrm{T}^n$. Then we get

$$\begin{aligned}\mathrm{T}^{n+1} &= \text{b-H}; \mathrm{B}^n; \text{l-T} && \subseteq \text{b-H}; \mathrm{T}^n; \text{l-T}\\ &= \text{b-H}; (\text{b-H}; \text{l-T})^n; \text{l-T} && \subseteq \text{b-H}; (\text{b-H}; \text{l-B})^n; \text{l-B}\\ &= (\text{b-H}; \text{l-B})^n && \subseteq \text{l-B}; (\text{b-H}; \text{l-B})^n; \text{b-H}\\ &= (\text{l-B}; \text{b-H})^{n+1} && = \mathrm{B}^{n+1},\end{aligned}$$

which contradicts (13). In the first, third, and fourth steps, we used again the equality from Lemma 5.30, Theorem 5.24, and Theorem 5.14, respectively. In the fifth step, we used that both b-H and l-B are closed under composition; see Remark 5.8 and Lemma 5.10, and Theorem 5.18, respectively. Finally, the last step follows from Theorem 5.16(B). □

Now we can prove the main result of this subsection.

Theorem 5.32 ([105], Theorem 2). *If S is a proper semiring, then the diagram in Fig. 4 is the inclusion diagram of the depicted classes of tree series transformations.*

Proof. Since top-down tree transducers and top-down wtt over $\mathbb{B}$ can be identified, cf. Sect. 5.2, we obtain that $\mathrm{TOP}(\mathbb{B})^n \subset \mathrm{TOP}(\mathbb{B})^{n+1}$ for every $n \geq 1$, by [50], Theorem 3.14. Thus, by Lemmata 5.28 and 5.31, the inequalities (13) and (14) hold for every proper semiring S, which proves the theorem. $\square$

5.7 Further Models of Weighted Tree Transducers

In this subsection, we will discuss other models of weighted tree transducers that occur in the literature. Actually, we would like to describe them as modifications of our wtt concept. However, an obstacle for this is that most of them have rule mappings of the type

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

i.e., $\mu_k(\sigma)_{w,q}$ is not necessarily a polynomial. To remedy this problem, we first extend our wtt to so-called *infinite wtt* (for short: inf-wtt) that are defined exactly as wtt but the rule mapping μ_k has the above type. Moreover, we require that S is complete in order to have IO-substitution well defined. The tree transformation computed by an inf-wtt is defined in the same way as for wtt except that we use the Σ-algebra $(S\langle\langle T_\Delta\rangle\rangle^Q, \mu_\mathcal{M})$ and IO-substitution of arbitrary tree series. An inf-wtt is *polynomial* (for short: p) if the rule mapping μ_k maps into $S\langle T_\Delta(Z)\rangle^{Q(X_k)^* \times Q}$, i.e., a polynomial inf-wtt is the same as our wtt. The class of tree series transformations computed by certain restrictions of inf-wtt is denoted in the same way as the classes for the corresponding wtt except that we add ‘inf’ as index, like: l-$\mathrm{TOP}(S)_{\mathrm{inf}}$.

Bottom-up Inf-wtt with OI-Substitution

In [91], so-called *tree transducers* were defined. Such a tree transducer is a bottom-up inf-wtt $\mathcal{M} = (Q, \Sigma, \Delta, S, \mu, \nu)$ over a commutative and continuous semiring S with a root output $\nu : Q \to S\langle\langle T_\Delta(Z_1)\rangle\rangle$. As with our inf-wtt, a Σ-algebra is associated with $\mathcal{M}$, however, the operation $\mu_\mathcal{M}(\sigma)$ is defined in terms of the OI-substitution $\leftarrow_{\mathrm{OI}}$ for tree series (introduced in Sect. 3.3). Then the tree series transformation computed by $\mathcal{M}$ is defined as $\tau_\mathcal{M}(\xi) = \sum_{q\in Q} \nu(q) \leftarrow_{\mathrm{OI}} h_\mu(\xi)_q$ for every $\xi \in T_\Sigma$. In [91], it is shown that the polynomial versions of such tree transducers over $\mathbb{B}$ and the so-called *nondeterministically simple top-down tree transducers* are semantically equivalent. The latter are special top-down tree transducers with rules of the form $q(\sigma(x_1, \ldots, x_k)) \to \zeta(q_1(x_1), \ldots, q_k(x_k))$, where $k \geq 0$ and $\sigma \in \Sigma^{(k)}$, cf. [68], Exercise 4 in Chap. IV. A *recognizable tree transducer* is another restricted version of the tree transducer of [91]; in this model the tree series $\mu_k(\sigma)_{q_1(x_1)\ldots q_k(x_k),q}$ is in $\mathrm{Rec}(\Delta \cup Z_k, S)$ for every $k \geq 0$, $\sigma \in \Sigma^{(k)}$, and $q, q_1, \ldots, q_k \in Q$, and $\nu(q)$ has the form $a_q.z_1$ for every $q \in Q$. In [91], Corollary 17, it is shown that, for every linear, nondeleting, and recognizable tree transducer $\mathcal{M}$ and recognizable tree series $r \in S\langle\langle T_\Sigma\rangle\rangle$, the tree series $\tau_\mathcal{M}(r)$ is also recognizable. Finally, we mention that in [53], top-down inf-wtt and tree transducers of [91] are related.

Inf-wtt with OI-Substitution

In [102], the idea of [91] was generalized and inf-wtt (rather than bottom-up inf-wtt) with root output and OI-substitution over a complete semiring were investigated. We denote the class of tree series transformations computed by such inf-wtt by $\mathrm{WTT}_{\mathrm{OI}}(S)_{\mathrm{inf}}$; the restricted classes are denoted by applying the same system as in Sect. 5.3. In [102], Lemmas 5 and 6 it was proved that $\mathrm{x}\text{-}\mathrm{TOP}^{\mathrm{R}}_{\mathrm{OI}}(S)_{\mathrm{inf}} = \mathrm{x}\text{-}\mathrm{TOP}_{\mathrm{OI}}(S)_{\mathrm{inf}}$ for every combination $\mathrm{x} \in \{\mathrm{t}, \mathrm{d}, \mathrm{h}, \mathrm{p}, \mathrm{b}, \mathrm{l}, \mathrm{n}\}$ (where p stands for polynomial), and $\mathrm{xp}\text{-}\mathrm{TOP}^{\mathrm{R}}_{\mathrm{OI}}(S)_{\mathrm{inf}} = \mathrm{xp}\text{-}\mathrm{WTT}_{\mathrm{OI}}(S)_{\mathrm{inf}}$ for every combination $\mathrm{x} \in \{\mathrm{t}, \mathrm{d}, \mathrm{h}, \mathrm{b}, \mathrm{l}, \mathrm{n}\}$. Moreover, top-down wtt and polynomial top-down inf-wtt with OI-substitution have the same computation power, i.e., $\mathrm{x}\text{-}\mathrm{TOP}(S) = \mathrm{xp}\text{-}\mathrm{TOP}_{\mathrm{OI}}(S)_{\mathrm{inf}}$, for every combination $\mathrm{x} \in \{\mathrm{t}, \mathrm{d}, \mathrm{h}, \mathrm{b}, \mathrm{l}, \mathrm{n}\}$, cf. [102], Theorem 7.

Top-down Inf-wtt and Bottom-up Inf-wtt with IO$^{\mathrm{o}}$-Substitution

It was observed already in [53] that top-down and bottom-up inf-wtt do not generalize all fundamental properties of top-down and bottom-up tree transducers. For example, it was proved in [53], Proposition 3.14 that the computation power of homomorphism top-down wtt and of homomorphism bottom-up wtt over the semiring Nat are incomparable, while they are equal in the unweighted case, cf. [48], Lemma 3.2. Therefore, in [65], an alternative semantics, based on the so-called IO$^{\mathrm{o}}$-substitution of tree series, of top-down and bottom-up inf-wtt was suggested for consideration. Roughly speaking, the IO$^{\mathrm{o}}$-substitution differs from the IO-substitution defined on page 362 in that we take into account the number of occurrences of z_i in ξ for every $i \in I$ when computing the coefficient of a tree $\xi(\zeta_i \mid i \in I)$. Formally, the *IO$^{\mathrm{o}}$-substitution* of tree series $(s_i \mid i \in I)$ into r is defined by

$$r \leftarrow_{\mathrm{IO}^{\mathrm{o}}} (s_i)_{i\in I} = \sum_{\substack{\xi\in T_\Delta(Z),\\ (\forall i\in I):\zeta_i\in\mathrm{supp}(s_i)}} \Big((r,\xi) \cdot \prod_{i\in I} (s_i,\zeta_i)^{|\xi|_{z_i}} \Big).\xi(\zeta_i \mid i \in I).$$

We note that [IO]-substitution as defined on page 347 takes into account whether z_i occurs in ξ or not but does not use the number of occurrences. Then top-down inf-wtt and bottom-up inf-wtt with IO$^{\mathrm{o}}$-substitution are defined in the same way as in this chapter except that we use IO$^{\mathrm{o}}$-substitution in the definition of $\mu_{\mathcal{M}}(\sigma)$ instead of IO-substitution. The classes of tree series transformations computed by top-down inf-wtt or bottom-up inf-wtt with IO$^{\mathrm{o}}$-substitution are denoted by indexing the corresponding notation with o, like: $\mathrm{l}\text{-}\mathrm{TOP}_{\mathrm{o}}(S)_{\mathrm{inf}}$. It turned out that the IO$^{\mathrm{o}}$-substitution does not provide anything new for top-down inf-wtt because, for every combination x over $\{\mathrm{t}, \mathrm{d}, \mathrm{h}, \mathrm{p}, \mathrm{b}, \mathrm{l}, \mathrm{n}\}$, we have $\mathrm{x}\text{-}\mathrm{TOP}(S)_{\mathrm{inf}} = \mathrm{x}\text{-}\mathrm{TOP}_{\mathrm{o}}(S)_{\mathrm{inf}}$, cf. [65], Theorem 5.2. However, for bottom-up inf-wtt it does because for every partially ordered semiring S with $1 \preceq 1 + 1$ and $\mathrm{x}, \mathrm{y} \in \{\mathrm{td}, \mathrm{d}, \mathrm{h}\}$, the classes x-BOT$(S)_{\text{inf}}$ and y-BOT$_{\text{o}}(S)_{\text{inf}}$ are incomparable with respect to inclusion, cf. [110], Theorem 5.10. Also, for every combination x over $\{\text{t}, \text{d}, \text{h}, \text{p}, \text{b}, \text{l}, \text{n}\}$, we have xln-BOT$(S)_{\text{inf}}$ = xln-BOT$_{\text{o}}(S)_{\text{inf}}$ and xpb-BOT$(S)_{\text{inf}}$ = xpb-BOT$_{\text{o}}(S)_{\text{inf}}$ provided S is idempotent, cf. [65], Theorems 5.5 and 5.8. Moreover, the following results of [65] generalize the corresponding ones concerning tree transducers. By Theorem 5.12, we have xh-TOP$(S)_{\text{inf}}$ = xh-BOT$_{\text{o}}(S)_{\text{inf}}$ for every zero-divisor free semiring S. Moreover, xln-TOP$(S)_{\text{inf}}$ = xln-BOT$_{\text{o}}(S)_{\text{inf}}$, cf. Theorem 5.5 and Proposition 5.30. Finally, for every combination x over $\{\text{p}, \text{b}\}$, we have xl-TOP$(S)_{\text{inf}} \subseteq$ xl-BOT$_{\text{o}}(S)_{\text{inf}}$, cf. Theorem 5.26.

Top-down and Bottom-up Inf-wtt with Term Rewrite Semantics

In this chapter, the semantics of top-down and bottom-up inf-wtt was defined in an algebraic framework, more precisely, as an initial algebra semantics. In [66] an alternative approach was suggested by introducing weighted tree transducers of which the semantics is defined in an operational style. A weighted tree transducer of [66] is a tree transducer in which each (term rewriting) rule is associated with a weight taken from S. Along a successful leftmost derivation, the weights of the involved rules are multiplied and, for every pair of input tree and output tree, the weights of its successful leftmost derivations are summed up. In [66], it is shown in a constructive way that the two approaches, i.e., weighted tree transducers with initial algebra semantics and weighted tree transducers with term rewrite semantics, are semantically equivalent for both, the top-down and the bottom-up case, cf. Theorems 6.9 and 5.10.

Deterministic Bottom-up Inf-wtt over Multiplicative Monoids

In [101], the concept of a *deterministic bottom-up weighted tree transducer* (for short: deterministic bu-w-tt) was defined in a similar way to our deterministic bottom-up inf-wtt, except for the following. Since in case of a deterministic bottom-up inf-wtt the "additive part" of S is needless, cf. [53], Proposition 3.12, the author defines deterministic bottom-up inf-wtt over a multiplicative monoid A with absorbing element 0. Every deterministic bottom-up inf-wtt over S is a deterministic bu-w-tt over the monoid $(S, \cdot, 1)$. However, deterministic bu-w-tt are more general than our deterministic bottom-up inf-wtt because there exists a monoid $(A, \cdot, 1)$ with absorbing element 0 for which there does not exist a semiring $(A, +, \cdot, 0, 1)$ (cf. [101], Observation 2.2). Deterministic bu-w-tt are defined with both IO-substitution and IO$^{\text{o}}$-substitution semantics and the restricted versions of both models are considered for every combination x over $\{\text{t}, \text{h}, \text{l}, \text{n}\}$. In this way, there are 24 classes of tree series transformations computed by restricted deterministic bu-w-tt. Also, the underlying monoid is restricted, namely a nonperiodic monoid; a periodic, commutative and nonregular monoid; a periodic, commutative, and regular monoid; a commutative and idempotent monoid; and a periodic and commutative group is considered as the underlying monoid. For each kind of underlying monoid, the inclusion diagram of the 24 classes is presented, cf. [101], Theorems 4.8, 4.17, 4.20, 4.23, and 4.25, respectively.

Chapter 5 of [106] is a revised and extended version of [101]. The author considers deterministic bottom-up inf-wtt and deterministic top-down inf-wtt (over S). Thus, the bottom-up model is more restricted than the deterministic bu-w-tt of [101]. On the other hand, it is more general because deterministic bottom-up inf-wtt have final output tree series. Similarly, deterministic top-down inf-wtt have initial output tree series. For these models, results similar to those in [101] are obtained.

5.8 Further Results

It follows from Theorem 5.5 and Lemma 5.11(C) that the class ln-TOP(S) is closed under composition. In [89], Theorem 2.4 and [93], Theorem 3.7 this has been generalized to top-down inf-wtt in which the family of rule mappings has the property that $\mu_k(\sigma)_{w,q}$ is a recognizable tree series and algebraic tree series, respectively.

In [103] it was shown that bottom-up inf-wtt can be simulated by weighted tree automata over distributive multioperator monoids [90, 103, 62]. This model has already been discussed as concept (f) on pages 353 and 360.

Acknowledgments. We are indebted to Frank Drewes, Manfred Droste, Joost Engelfriet, and Andreas Maletti for their effort to help us in forming the final version of this chapter. We also thank Ferenc Gécseg, Johanna Högberg, Magnus Steinby, and Janis Voigtländer for their valuable comments.

References

1. A. Alexandrakis and S. Bozapalidis. Weighted grammars and Kleene's theorem. *Information Processing Letters*, 24(1):1–4, 1987.
2. D. Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75:87–106, 1987.
3. F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, Cambridge, 1998.
4. B.S. Baker. Composition of top-down and bottom-up tree transductions. *Information and Control*, 41(2):186–213, 1979.
5. R. Bělohlávek. Determinism and fuzzy automata. *Information Sciences*, 143:205–209, 2002.
6. J. Berstel. *Transductions and Context-Free Languages.* Teubner, Stuttgart, 1979.
7. J. Berstel and C. Reutenauer. Recognizable formal power series on trees. *Theoretical Computer Science*, 18(2):115–148, 1982.

8. J. Berstel and C. Reutenauer. *Rational Series and Their Languages*, volume 2 of *Monographs in Theoretical Computer Science. An EATCS Series*, Springer, Berlin, 1988.
9. R.E. Block and G. Griffing. Recognizable formal series on trees and cofree coalgebraic systems. *Journal of Algebra*, 215:543–573, 1999.
10. S.L. Bloom and Z. Ésik. An extension theorem with an application to formal tree series. *Journal of Automata, Languages and Combinatorics*, 8:145–185, 2003.
11. B. Borchardt. The Myhill–Nerode theorem for recognizable tree series. In Z. Ésik and Z. Fülöp, editors, *Proceedings of the 7th International Conference on Developments in Language Theory (DLT), Szeged, Hungary*, volume 2710 of *Lecture Notes in Computer Science*, pages 146–158. Springer, Berlin, 2003.
12. B. Borchardt. A pumping lemma and decidability problems for recognizable tree series. *Acta Cybernetica*, 16(4):509–544, 2004.
13. B. Borchardt. Code selection by tree series transducers. In M. Domaratzki, A. Okhotin, K. Salomaa, and S. Yu, editors, *Proceedings of the 9th International Conference on Implementation and Application of Automata (CIAA), Kingston, Canada*, volume 3317 of *Lecture Notes in Computer Science*, pages 57–67. Springer, Berlin, 2005.
14. B. Borchardt. *The Theory of Recognizable Tree Series*. Verlag für Wissenschaft und Forschung, Berlin, 2005 (PhD thesis, TU Dresden, Germany, 2004).
15. B. Borchardt, Z. Fülöp, Z. Gazdag, and A. Maletti. Bounds for tree automata with polynomial costs. *Journal of Automata, Languages and Combinatorics*, 10:107–157, 2005.
16. B. Borchardt, A. Maletti, B. Šešelja, A. Tepavčevic, and H. Vogler. Cut sets as recognizable tree languages. *Fuzzy Sets and Systems*, 157:1560–1571, 2006.
17. B. Borchardt and H. Vogler. Determinization of finite state weighted tree automata. *Journal of Automata, Languages and Combinatorics*, 8(3):417–463, 2003.
18. S. Bozapalidis. Effective construction of the syntactic algebra of a recognizable series on trees. *Acta Informatica*, 28:351–363, 1991.
19. S. Bozapalidis. Representable tree series. *Fundamenta Informaticae*, 21:367–389, 1994.
20. S. Bozapalidis. Positive tree representations and applications to tree automata. *Information and Control*, 139(2):130–153, 1997.
21. S. Bozapalidis. Equational elements in additive algebras. *Theory of Computing Systems*, 32(1):1–33, 1999.
22. S. Bozapalidis. Context-free series on trees. *Information and Computation*, 169:186–229, 2001.
23. S. Bozapalidis and A. Alexandrakis. Représentations matricielles des séries d'arbre reconnaissables. *Theoretical Informatics and Applications, RAIRO*, 23(4):449–459, 1989.

24. S. Bozapalidis and O. Louscou-Bozapalidou. The rank of a formal tree power series. *Theoretical Computer Science*, 27:211–215, 1983.
25. S. Bozapalidis and G. Rahonis. On the closure of recognizable tree series under tree homomorphisms. *Journal of Automata, Languages and Combinatorics*, 10:185–202, 2005.
26. W.S. Brainerd. The minimalization of tree automata. *Information and Control*, 13:484–491, 1968.
27. P. Buchholz. Bisimulation relations for weighted automata. *Theoretical Computer Science*, 393(1–3):109–123, 2008.
28. J.R. Büchi. Weak second-order arithmetic and finite automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 6:66–92, 1960.
29. H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available on: http://www.grappa.univ-lille3.fr/tata, 1997.
30. B. Courcelle. Equivalences and transformations of regular systems—Applications to recursive program schemes and grammars. *Theoretical Computer Science*, 42:1–122, 1986.
31. B. Courcelle. Recursive applicative program schemes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, volume B*, pages 459–492. Elsevier, Amsterdam, 1990.
32. B. Courcelle. Basic notions of universal algebra for language theory and graph grammars. *Theoretical Computer Science*, 163:1–54, 1996.
33. J. Doner. Tree acceptors and some of their applications. *Journal of Computer and System Sciences*, 4:406–451, 1970.
34. F. Drewes. Computation by tree transductions. PhD thesis, University of Bremen, 1996.
35. F. Drewes. *Grammatical Picture Generation—A Tree-Based Approach*, *Texts in Theoretical Computer Science. An EATCS Series*. Springer, Berlin, 2006.
36. F. Drewes and H. Vogler. Learning deterministically recognizable tree series. *Journal of Automata, Languages and Combinatorics*, 12:333–354, 2007.
37. M. Droste and P. Gastin. Weighted automata and weighted logics. In L. Caires, G.F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, editors, *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP), Lisbon, Portugal*, volume 3580 of *Lecture Notes in Computer Science*, pages 513–525. Springer, Berlin, 2005.
38. M. Droste and P. Gastin. Weighted automata and weighted logics. *Theoretical Computer Science*, 380:69–86, 2007.
39. M. Droste and P. Gastin. Weighted automata and weighted logics. In this *Handbook*. Chapter 5. Springer, Berlin, 2009.
40. M. Droste and W. Kuich. Semirings and formal power series. In this *Handbook*. Chapter 1. Springer, Berlin, 2009.

41. M. Droste, Chr. Pech and H. Vogler. A Kleene theorem for weighted tree automata. *Theory of Computing Systems*, 38:1–38, 2005.
42. M. Droste and G. Rahonis. Weighted automata and weighted logics with discounting. In J. Holub and J. Zdárek, editors, *Proceedings of the 12th International Conference on Implementation and Application of Automata (CIAA), Prague*, volume 4783 of *Lecture Notes in Computer Science*, pages 73–84. Springer, Berlin, 2007.
43. M. Droste and H. Vogler. Weighted tree automata and weighted logics. *Theoretical Computer Science*, 366:228–247, 2006.
44. M. Droste and H. Vogler. Weighted logics for unranked tree automata. *Theory of Computing Systems*, 2009, in press.
45. S. Eilenberg. *Automata, Languages, and Machines—Volume A*, volume 59 of *Pure and Applied Mathematics*. Academic Press, San Diego, 1974.
46. C.C. Elgot. Decision problems of finite automata design and related arithmetics. *Transactions of the American Mathematical Society*, 98:21–52, 1961.
47. C.A. Ellis. Probabilistic tree automata. *Information and Control*, 19(5):401–416, 1971.
48. J. Engelfriet. Bottom-up and top-down tree transformations—A comparison. *Mathematical Systems Theory*, 9(3):198–231, 1975.
49. J. Engelfriet. Top–down tree transducers with regular look-ahead. *Mathematical Systems Theory*, 10:289–303, 1977.
50. J. Engelfriet. Three hierarchies of transducers. *Mathematical Systems Theory*, 15(2):95–125, 1982.
51. J. Engelfriet. Tree transducers and syntax-directed semantics. In *Proceedings of the 7th Colloquium on Trees in Algebra and Programming (CAAP), Lille, France, 1982*.
52. J. Engelfriet. On incomparability of tree series transformation classes. Personal communication, 2008.
53. J. Engelfriet, Z. Fülöp, and H. Vogler, Bottom-up and top-down tree series transformations. *Journal of Automata, Languages and Combinatorics*, 7:11–70, 2002.
54. J. Engelfriet and S. Maneth. A comparison of pebble tree transducers with macro tree transducers. *Acta Informatica*, 39(9):613–698, 2003.
55. J. Engelfriet and E.M. Schmidt. IO and OI.I. *Journal of Computer and System Sciences*, 15(3):328–353, 1977.
56. J. Engelfriet and E.M. Schmidt. IO and OI.II. *Journal of Computer and System Sciences*, 16(1):67–99, 1978.
57. Z. Ésik. Fixed point theory. In this *Handbook*. Chapter 2. Springer, Berlin, 2009.
58. Z. Ésik and W. Kuich. Formal tree series. *Journal of Automata, Languages and Combinatorics*, 8(2):219–285, 2003.
59. Z. Ésik and G. Liu. Fuzzy tree automata. *Fuzzy Sets and Systems*, 158:1450–1460, 2007.

60. I. Fichtner. Weighted picture automata and weighted logics. *Theory of Computing Systems*, 2009, in press.
61. Z. Fülöp, Z. Gazdag and H. Vogler, Hierarchies of tree series transformations. *Theoretical Computer Science*, 314:387–429, 2004.
62. Z. Fülöp, A. Maletti, and H. Vogler. A Kleene theorem for weighted tree automata over distributive multioperator monoids. *Theory of Computing Systems*, 44:455–499, 2009.
63. Z. Fülöp and S. Vágvölgyi. Congruential tree languages are the same as recognizable tree languages. *Bulletin of the European Association for Theoretical Computer Science*, 30:175–185, 1989.
64. Z. Fülöp and H. Vogler. *Syntax-Directed Semantics—Formal Models Based on Tree Transducers*, *Monographs in Theoretical Computer Science. An EATCS Series.* Springer, Berlin, 1998.
65. Z. Fülöp and H. Vogler. Tree series transformations that respect copying. *Theory of Computing Systems*, 36(3):247–293, 2003.
66. Z. Fülöp and H. Vogler. Weighted tree transducers. *Journal of Automata, Languages and Combinatorics*, 9:31–54, 2004.
67. Z. Fülöp and H. Vogler. Comparison of several classes of weighted tree automata. Technical report, TUD-FI06-08-Dez.2006, TU Dresden, 2006.
68. F. Gécseg and M. Steinby. *Tree Automata.* Akadémiai Kiadó, Budapest, 1984.
69. F. Gécseg and M. Steinby. Tree languages. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages, volume 3*, pages 1–68. Springer, Berlin, 1997.
70. J.A. Goguen, J.W. Thatcher, E.G. Wagner, and J.B. Wright. Initial algebra semantics and continuous algebras. *Journal of the ACM*, 24:68–95, 1977.
71. J.S. Golan. *Semirings and Their Applications.* Kluwer Academic, Dordrecht, 1999.
72. J. Graehl and K. Knight. Training tree transducers. In *Human Language Technology Conference/North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT–NAACL), Boston, Massachusetts, USA*, pages 105–112. Association for Computational Linguistics, Stroudsburg, 2004.
73. G. Grätzer. *Universal Algebra.* van Nostrand, Princeton, 1968.
74. I. Guessarian. Pushdown tree automata. *Theory of Computing Systems*, 16:237–263, 1983.
75. A. Habrard and J. Oncina. Learning multiplicity tree automata. In Y. Sakakibara, S. Kobayashi, K. Sato, T. Nishino, and E. Tomita, editors, *Proceedings of the 8th International Colloquium on Grammatical Inference: Algorithms and Applications (ICGI)*, volume 4201 of *Lecture Notes in Computer Science*, pages 268–280. Springer, Berlin, 2006.
76. U. Hebisch and H.J. Weinert. *Semirings—Algebraic Theory and Applications in Computer Science.* World Scientific, Singapore, 1998.

77. J. Högberg, A. Maletti, and J. May. Bisimulation minimisation of weighted tree automata. In T. Harju, J. Karhumäki, and A. Lepistö, editors, *Proceedings of the 11th International Conference on Developments in Language Theory (DLT)*, volume 4588 of *Lecture Notes in Computer Science*, pages 229–241. Springer, Berlin, 2007.
78. Y. Inagaki and T. Fukumura. On the description of fuzzy meaning of context-free languages. In L.A. Zadeh, editor, *Fuzzy Sets and Their Applications to Cognitive and Decision Processes*, pages 301–328. Academic Press, New York, 1975.
79. E.T. Irons. A syntax directed compiler for ALGOL 60. *Communications of the ACM*, 4(1):51–55, 1961.
80. D. Kirsten and I. Mäurer. On the determinization of weighted automata. *Journal of Automata, Languages and Combinatorics*, 10:287–312, 2005.
81. S.E. Kleene. Representation of events in nerve nets and finite automata. In C.E. Shannon and J. McCarthy, editors, *Automata Studies*, pages 3–42. Princeton University Press, Princeton, 1956.
82. J.W. Klop. Term rewrite systems. In S. Abramsky, D.M. Gabbay, T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science, volume 2*, pages 1–116. Oxford Science Publications, Oxford, 1992.
83. K. Knight and J. Graehl. An overview of probabilistic tree transducers for natural language processing. In A. Gelbukh, editor, *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing), Mexico City, Mexico*, volume 3406 of *Lecture Notes in Computer Science*, pages 1–24. Springer, Berlin, 2005.
84. K. Knight and J. May. Applications of weighted automata in natural language processing. In this *Handbook*. Chapter 14. Springer, Berlin, 2009.
85. H.-P. Kolb, J. Michaelis, U. Mönnich, and F. Morawietz. An operational and denotational approach to non-context-freeness. *Theoretical Computer Science*, 293:261–289, 2003.
86. D. Kozen. On the Myhill–Nerode theorem for trees. *Bulletin of the European Association for Theoretical Computer Science*, 47:170–173, 1992.
87. W. Kuich. Semirings and formal power series: Their relevance to formal languages and automata. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages, volume 1*, Chapter 9, pages 609–677. Springer, Berlin, 1997.
88. W. Kuich. Formal power series over trees. In S. Bozapalidis, editor, *Proceedings of the 3rd International Conference on Developments in Language Theory (DLT), Thessaloniki, Greece*, pages 61–101. Aristotle University of Thessaloniki, Thessaloniki, 1998.
89. W. Kuich. Full abstract families of tree series I. In J. Karhumäki, H. Maurer, G. Paun, and G. Rozenberg, editors, *Jewels Are Forever*, pages 145–156. Springer, Berlin, 1999.
90. W. Kuich. Linear systems of equations and automata on distributive multioperator monoids. In D. Dorninger, G. Eigenthaler, and M. Goldstern, editors, *Contributions to General Algebra 12—Proceedings of the*

58th Workshop on General Algebra "58. Arbeitstagung Allgemeine Algebra", Vienna University of Technology, June 3–6, 1999, pages 1–10. Verlag Johannes Heyn, Klagenfurt, 1999.

91. W. Kuich. Tree transducers and formal tree series. *Acta Cybernetica*, 14:135–149, 1999.
92. W. Kuich. Formal series over algebras. In M. Nielsen and B. Rovan, editors, *Mathematical Foundations of Computer Science (MFCS), Bratislava, Slovakia*, volume 1893 of *Lecture Notes in Computer Science*, pages 488–496. Springer, Berlin, 2000.
93. W. Kuich. Full abstract families of tree series II. In R. Freund and A. Kelemenova, editors, *Proceedings of the International Workshop Grammar Systems 2000*, pages 347–358. Schlesische Universität Troppau 2000.
94. W. Kuich. Pushdown tree automata, algebraic tree systems, and algebraic tree series. *Information and Computation*, 165:69–99, 2001.
95. W. Kuich. Formal series over sorted algebras. *Discrete Mathematics*, 254:231–258, 2002.
96. W. Kuich and A. Salomaa. *Semirings, Automata, Languages*, volume 5 of *Monographs in Theoretical Computer Science. An EATCS Series* Springer, Berlin, 1986.
97. S. Lang. *Algebra*, 3rd edition. Addison–Wesley, Reading, 1993.
98. L. Libkin. Logics for unranked trees: An overview. In L. Caires, G.F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, editors, *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP), Lisbon, Portugal*, volume 3580 of *Lecture Notes in Computer Science*, pages 35–50. Springer, Berlin, 2005.
99. L. Libkin. Logics for unranked trees: An overview. *Logical Methods in Computer Science*, 2(3:2):1–31, 2006.
100. M. Magidor and G. Moran. Probabilistic tree automata and context free languages. *Israel Journal of Mathematics*, 8:340–348, 1970.
101. A. Maletti. Hasse diagrams for classes of deterministic bottom-up tree-to-tree-series transformations. *Theoretical Computer Science*, 339:200–240, 2005.
102. A. Maletti. The power of tree series transducers of type I and II. In C. De Felice and A. Restivo, editors, *Proceedings of the 9th International Conference on Developments in Language Theory (DLT), Palermo, Italy*, volume 3572 of *Lecture Notes in Computer Science*, pages 338–349. Springer, Berlin, 2005.
103. A. Maletti. Relating tree series transducers and weighted tree automata. *International Journal of Foundations of Computer Science*, 16(4):723–741, 2005.
104. A. Maletti. Compositions of tree series transformations. *Theoretical Computer Science*, 366:248–271, 2006.
105. A. Maletti. Hierarchies of tree series transformations revisited. In O.H. Ibarra and Z. Dang, editors, *Proceedings of the 10th International*

Conference on Developments in Language Theory (DLT), volume 4036 of *Lecture Notes in Computer Science*, pages 215–225. Springer, Berlin, 2006.

106. A. Maletti. *The Power of Tree Series Transducers.* Der Andere Verlag, Tönning, 2006 (PhD thesis, TU Dresden, Germany, 2006).
107. A. Maletti. Generalizing the inclusion and the hierarchy results from positive semirings to semirings which are not rings. Personal communication, 2007.
108. A. Maletti. Learning deterministically recognizable tree series—Revisited. In S. Bozapalidis and G. Rahonis, editors, *Proceedings of the 2nd International Conference on Algebraic Informatics*, volume 4728 of *Lecture Notes in Computer Science*, pages 218–235. Springer, Berlin, 2007.
109. A. Maletti. Myhill-Nerode theorem for recognizable tree series—Revisited. In E. Laber, C. Bornstein, L. Nogueira, and L. Faria, editors, *Proceedings of the 8th International Conference on Latin American Theoretical Informatics (LATIN)*, volume 4957 of *Lecture Notes in Computer Science*, pages 106–120. Springer, Berlin, 2008.
110. A. Maletti and H. Vogler. Incomparability results for classes of polynomial tree series transformations. *Journal of Automata, Languages and Combinatorics*, 10(4):535–568, 2005.
111. S. Maneth, A. Berlea, T. Perst, and H. Seidl. XML type checking with macro tree transducers. In *Proceedings of the Twenty-Fourth ACM SIGMOD–SIGACT–SIGART Symposium on Principles of Database Systems (PODS), Baltimore, Maryland*, pages 283–294. ACM, New York, 2005.
112. Chr. Mathissen. Definable transductions and weighted logics for texts. In T. Harju, J. Karhumäki, and A. Lepistö, editors, *Proceedings of the 11th International Conference on Developments in Language Theory (DLT), Turku*, volume 4588 of *Lecture Notes in Computer Science*, pages 324–336. Springer, Berlin, 2007.
113. Chr. Mathissen. Weighted logics for nested words and algebraic formal power series. In L. Aceto, I. Damgård, L.A. Goldberg, M.M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz, editors, *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP), Reykjavik*, volume 5126 of *Lecture Notes in Computer Science*, pages 221–232. Springer, Berlin, 2008.
114. J. May and K. Knight. Tiburon: A weighted tree automata toolkit. In O.H. Ibarra and H.-C. Yen, editors, *Proceedings of the 11th International Conference on Implementation and Application of Automata (CIAA)*, volume 4094 of *Lecture Notes in Computer Science*, pages 102–113. Springer, Berlin, 2006.
115. I. Meinecke. Weighted logics for traces. In D. Grigoriev, J. Harrison, and E.A. Hirsch, editors, *Proceedings of Computer Science—Theory and*

Applications, 1st CSR, St. Petersburg, volume 3967 of *Lecture Notes in Computer Science*, pages 235–246. Springer, Berlin, 2006.
116. J. Mezei and J.B. Wright. Algebraic automata and context-free sets. *Information and Control*, 11:3–29, 1967.
117. F. Neven. Automata, logic, and XML. In J. Bradfield, editor, *Computer Science Logic: 16th International Workshop (CSL), Edinburgh, Scotland, UK*, volume 2471 of *Lecture Notes in Computer Science*, pages 2–26. Springer, Berlin, 2002.
118. M. Nivat. Transduction des langages de Chomsky. *Université de Grenoble. Annales de l'Institut Fourier*, 18:339–456, 1968.
119. Chr. Pech. Kleene-type results for weighted tree automata. PhD thesis, TU Dresden, 2003.
120. Chr. Pech. Kleene's theorem for weighted tree-automata. In A. Lingas and B.J. Nilsson, editors, *Proceedings of the 14th International Symposium on Fundamentals of Computation Theory (FCT), Malmö, Sweden*, volume 2751 of *Lecture Notes in Computer Science*, pages 387–399. Springer, Berlin, 2003.
121. I. Petre and A. Salomaa. Algebraic systems and pushdown automata. In this *Handbook*. Chapter 7. Springer, Berlin, 2009.
122. G. Rahonis. Weighted Muller tree automata and weighted logics. *Journal of Automata, Languages, and Programming*, 12:455–483, 2007.
123. W.C. Rounds. Mappings and grammars on trees. *Theory of Computing Systems*, 4(3):257–287, 1970.
124. J. Sakarovitch. Rational and recognisable power series. In this *Handbook*. Chapter 4. Springer, Berlin, 2009.
125. A. Salomaa and M. Soittola. *Automata-Theoretic Aspects of Formal Power Series*, *Texts and Monographs in Computer Science*. Springer, Berlin, 1978.
126. T. Schwentick. Automata for XML—A survey. *Journal of Computer and System Sciences*, 73(3):289–315, 2007.
127. H. Seidl. Deciding equivalence of finite tree automata. *SIAM Journal on Computing*, 19(3):424–437, 1990.
128. H. Seidl. Finite tree automata with cost functions. *Theoretical Computer Science*, 126(1):113–142, 1994.
129. B. Šešelja and A. Tepavčević. Completion of ordered structures by cuts of fuzzy sets, an overview. *Fuzzy Sets and Systems*, 136(1):1–19, 2003.
130. M. Steinby. A theory of tree language varieties. In M. Nivat and A. Podelski, editors, *Tree Automata and Languages*, *Studies in Computer Science and Artificial Intelligence*, North-Holland, Amsterdam, 1992.
131. T. Stüber. Using the theorem of Cayley–Hamilton for proving tree series not to be recognizable. Personal communication, 2007
132. T. Stüber, H. Vogler, and Z. Fülöp. Decomposition of weighted multi-operator tree automata. *International Journal Foundations of Computer Science*, 20(2):221–245, 2009.

133. J.W. Thatcher. Generalized^2 sequential machine maps. *Journal of Computer and System Sciences*, 4(4):339–367, 1970.
134. J.W. Thatcher. Tree automata: An informal survey. In A.V. Aho, editor, *Currents in the Theory of Computing*, pages 143–172. Prentice Hall, Englewood Cliffs, 1973.
135. J.W. Thatcher and J.B. Wright. Generalized finite automata theory with an application to a decision problem of second-order logic. *Theory of Computing Systems*, 2(1):57–81, 1968.
136. V. Vianu. A Web odyssey: From Codd to XML. In *Proceedings of the 20th ACM SIGACT–SIGMOD–SIGART Symposium on Principles of Database Systems (PODS), Santa Barbara, California, USA*, pages 1–15. ACM, New York, 2001.
137. H. Wang. On characters of semirings. *Houston Journal of Mathematics*, 23(3):391–405, 1997.
138. H. Wang. On rational series and rational languages. *Theoretical Computer Science*, 205(1–2):329–336, 1998.
139. W. Wechler. *Universal Algebra for Computer Scientists*, 1st edition, volume 25 of *Monographs in Theoretical Computer Science. An EATCS Series*. Springer, Berlin, 1992.