Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science 366 (2006) 228–247
Theoretical Computer Science
www.elsevier.com/locate/tcs

Weighted tree automata and weighted logics

Manfred Droste[a,*], Heiko Vogler[b]

[a] *Institute of Computer Science, Leipzig University, D-04109 Leipzig, Germany*
[b] *Department of Computer Science, Dresden University of Technology, D-01062 Dresden, Germany*

Abstract

We define a weighted monadic second order logic for trees where the weights are taken from a commutative semiring. We prove that a restricted version of this logic characterizes the class of formal tree series which are accepted by weighted bottom-up finite state tree automata. The restriction on the logic can be dropped if additionally the semiring is locally finite. This generalizes corresponding classical results of Thatcher, Wright, and Doner for tree languages and it extends recent results of Droste and Gastin [Weighted automata and weighted logics, in: Automata, Languages and Programming—32nd International Colloquium, ICALP 2005, Lisbon, Portugal, 2005, Proceedings, Lecture Notes in Computer Science, Vol. 3580, Springer, Berlin, 2005, pp. 513–525, full version in Theoretical Computer Science, to appear.] from formal power series on words to formal tree series.

Keywords: Weighted automata; Tree automata; MSO-logics; Formal power series

1. Introduction

In automata theory, Büchi's and Elgot's fundamental theorems [12,20] on the coincidence of regular languages with languages definable in monadic second order logic (for short: MSO-logic) led to various extensions. Thatcher and Wright [39] and Doner [16] established a corresponding equivalence for trees. Engelfriet [22,23] and Courcelle [14] proved a characterization of a large class of context-free graph languages as images of MSO-definable functions on sets of trees. Bloem and Engelfriet [4] proved that the tree transductions computed by a particular class of attributed tree transducers coincide with the MSO-definable tree transductions. On a different strand, following Schützenberger [35], weighted automata form a quantitative extension of classical automata, cf. [34,3,31] for surveys; this recently led to practical applications in digital image compression [15,28], in model-checking of probabilistic systems [38,1] and in natural language processing [13].

Very recently, Droste and Gastin [18] defined a weighted logic and obtained a characterization of the behavior of weighted automata on words by sentences of this weighted logic, thereby extending Büchi's and Elgot's classical results. Intuitively, this logic allows us to "count" how often a formula is true, thus it includes quantitative aspects. In this paper, we will consider a weighted logic for trees and we will achieve an extension of the result of Thatcher and Wright [39] and Doner [16] to the case of weighted tree automata.

* Corresponding author.
E-mail addresses: droste@informatik.uni-leipzig.de (M. Droste), vogler@tcs.inf.tu-dresden.de (H. Vogler).

0304-3975/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2006.08.025

Weighted tree automata have been considered by a number of researchers [10,2,37,29,24,33,6,19] and have been used for efficient code selection [25,7]. In such tree automata, transitions are assigned a weight which might model, e.g., resources used for its execution, length of time needed, or its reliability. To cope with such situations uniformly, the weights are taken as elements of a semiring. In our weighted logic we take the elements of the given semiring as atomic formulas. In order to define the semantics of the negation of a formula, we restrict in our syntax negation to atomic formulas. In comparison to the classical logics for trees, this is no essential restriction but forces us to include universal quantification into the syntax. However, already in weighted logics for words [18], in general there are series definable in this general weighted logic which are not recognizable by weighted automata. Fortunately, we can restrict universal quantification appropriately. Our main result states that for any commutative semiring the behaviors of weighted tree automata are precisely the series arising as semantics of sentences of our restricted weighted MSO-logic (cf. Theorem 5.1). If the semiring is locally finite, all sentences of our weighted MSO-logic have recognizable semantics (cf. Theorem 6.5).

The development of our weighted logic for trees is quite similar to the weighted logic for words [18]. Assuming familiarity with the unweighted logics for trees, background results on weighted tree automata, and the arguments for the word case in [18], several of our proofs can be obtained straightforwardly as an amalgamation and adjustment of the existing proofs from these three strands. However, a main difference arises for the treatment of the universal quantifier on first order variables. Roughly speaking, this difference is due to the need for synchronizing of variables in different subtrees; this problem does not arise in the case of words.

Moreover, crucial differences and open problems occur with respect to constructibility and decidability questions. Given any computable field (e.g., the rational numbers), we can show that for any restricted weighted MSO-sentence we can effectively construct an equivalent weighted tree automaton (cf. Corollary 5.8). Hence, given two such sentences, we can decide whether their semantics are equal (cf. Corollary 5.9). But, our procedures are not primitive recursive. This is due to the present lack of a comparable algebraic theory for weighted tree automata like it was developed for weighted automata on words, cf. [3,31]. Consequently, it is open at present how to check (for suitable semirings like arbitrary fields) whether a given weighted MSO-sentence for trees is restricted; for words this is possible [18]. This calls for further research on such questions for weighted tree automata.

2. Tree languages and tree automata

Let $\mathbb{N} = \{0, 1, 2, \dots\}$. A *ranked alphabet* is a tuple (Σ, rk_Σ) where Σ is a finite set and $rk_\Sigma : \Sigma \to \mathbb{N}$. For every $m \geqslant 0$, the set of all symbols of Σ with rank m is denoted by $\Sigma^{(m)}$. In the sequel we use the phrase "let Σ be a ranked alphabet", if either the function rk is an arbitrary one or it is known from the context. We denote the value $\max\{rk_\Sigma(\sigma) \mid \sigma \in \Sigma\}$ by $\max_\Sigma$. In examples, a particular ranked alphabet Σ is specified by showing the set Σ and attaching the respective rank to every symbol as superscript as, e.g., $\Sigma = \{\sigma^{(2)}, \alpha^{(0)}\}$. The set of (*finite*, *labeled and ordered*) *Σ-trees*, denoted by T_Σ, is the smallest subset T of $(\Sigma \cup \{(,)\} \cup \{,\})^*$ such that if $\sigma \in \Sigma^{(m)}$ with $m \geqslant 0$ and $s_1, \dots, s_m \in T$, then $\sigma(s_1, \dots, s_m) \in T$. In case $m = 0$, we identify $\sigma()$ with σ. Clearly $T_\Sigma = \emptyset$ iff $\Sigma^{(0)} = \emptyset$. Since we are not interested in the case that $T_\Sigma = \emptyset$, we assume that $\Sigma^{(0)} \neq \emptyset$ for every ranked alphabet Σ appearing in this paper.

We define the set of *positions in a tree* by means of the mapping $\mathrm{pos}: T_\Sigma \to \mathcal{P}(\mathbb{N}^*)$ inductively as follows: (i) if $t \in \Sigma^{(0)}$, then $\mathrm{pos}(t) = \{\varepsilon\}$, and (ii) if $t = \sigma(s_1, \dots, s_m)$ where $\sigma \in \Sigma^{(m)}$, $m \geqslant 1$ and $s_1, \dots, s_m \in T_\Sigma$, then $\mathrm{pos}(t) = \{\varepsilon\} \cup \{i.v \mid 1 \leqslant i \leqslant m, v \in \mathrm{pos}(s_i)\}$.

For every $t \in T_\Sigma$ and $w \in \mathrm{pos}(t)$, the *label of t at w* and *the subtree of t at w*, denoted by $t(w) \in \Sigma$ and $t|_w \in T_\Sigma$, respectively, are defined inductively as follows: if $t = \sigma(s_1, \dots, s_m)$ for some $\sigma \in \Sigma^{(m)}$ with $m \geqslant 0$ and $s_1, \dots, s_m \in T_\Sigma$, then $t(\varepsilon) = \sigma$ and $t|_\varepsilon = t$, and if $w = i.v$ and $1 \leqslant i \leqslant m$, then $t(w) = s_i(v)$ and $t|_w = s_i|_v$.

Next we recall, for the convenience of the reader, basic back-ground on bottom-up finite state tree automata, also see [26,27]. A *bottom-up finite state tree automaton* (for short: bu-fta) is a tuple $M = (Q, \Sigma, \delta, F)$ where Q is a finite set (of states), Σ is a ranked alphabet (of input symbols), $\delta = (\delta_\sigma \mid \sigma \in \Sigma)$ is a family of transition functions of the form $\delta_\sigma : Q^m \to \mathcal{P}(Q)$ where $m \geqslant 0$ and $\sigma \in \Sigma^{(m)}$, and $F \subseteq Q$ (final states). A bottom-up finite state tree automaton is *deterministic* if for every $m \geqslant 0$, $\sigma \in \Sigma^{(m)}$, and $q_1, \dots, q_m \in Q$, the set $\delta_\sigma(q_1, \dots, q_m)$ is a singleton. In this case we identify the set $\delta_\sigma(q_1, \dots, q_m)$ with its element.

Now we define the run semantics of a bu-fta M. Let $t \in T_\Sigma$. A *run of M on t* is a mapping $r : \mathrm{pos}(t) \to Q$. A run r is *valid* if whenever $w \in \mathrm{pos}(t)$ and $t(w) = \sigma \in \Sigma^{(m)}$ for some $m \geqslant 0$, then $r(w) \in \delta_\sigma(r(w.1), \dots, r(w.m))$. In other

words, a run r is valid if at every position w the state $r(w)$ is the result of applying the transition performed at w to the states assigned to the predecessors of w.

Let $R_M(t)$ and $R_M^v(t)$ denote the sets of *all runs of M on t* and *all valid runs of M on t*, respectively. We note that if M is deterministic, then there is a unique valid run of M on t, denoted by r_t.

The *tree language accepted by M* is the set $\mathcal{L}(M) = \{t \in T_\Sigma \mid \exists r \in R_M^v(t) : r(\varepsilon) \in F\}$. (We note that the run semantics of M is equivalent to the initial algebra semantics which is usually associated with M, cf. e.g. [39, p. 58].) A tree language $\mathcal{L}$ is *recognizable* (resp. *deterministically recognizable*) if there is a (resp. deterministic) bottom-up finite state tree automaton M such that $\mathcal{L} = \mathcal{L}(M)$. By applying the usual powerset construction, one obtains that every recognizable tree language is also deterministically recognizable (cf. [39, Theorem 1]). It is well known that the class of recognizable tree languages is closed under the boolean operations (i.e., union, intersection, and complement; cf. [39, Theorem 2]).

Next we briefly recall the MSO-logic on trees and the generalization of Büchi's result, namely that MSO-definable tree languages are exactly the recognizable tree languages [39,16]. Let Σ be a ranked alphabet. The *set* $\mathrm{MSO}(\Sigma)$ *of all formulas of MSO-logic over* Σ is defined as the smallest set F such that

(1) F contains all *atomic formulas* $\mathrm{label}_\sigma(x)$, $\mathrm{edge}_i(x, y)$, and $x \in X$ and

(2) if $\varphi, \psi \in F$, then also $\varphi \vee \psi$, $\varphi \wedge \psi$, $\neg\varphi$, $\exists x.\varphi$, $\exists X.\varphi$, $\forall x.\varphi$, $\forall X.\varphi \in F$,

where $\sigma \in \Sigma$, x, y are first order variables, $1 \leqslant i \leqslant \max_\Sigma$, and X is a second order variable. The set of free variables of φ is denoted by $\mathrm{Free}(\varphi)$.

Let $\mathcal{V}$ be a finite set of first order and second order variables. The ranked alphabet $\Sigma_\mathcal{V} = (\Sigma \times \{0,1\}^\mathcal{V}, rk)$ is defined by $rk((\sigma, f)) = rk_\Sigma(\sigma)$ for every $f \in \{0,1\}^\mathcal{V}$. For a symbol $(\sigma, f) \in \Sigma_\mathcal{V}$ we denote σ by $(\sigma, f)_1$ and f by $(\sigma, f)_2$. A $\Sigma_\mathcal{V}$-tree s is *valid* if for every first order variable $x \in \mathcal{V}$, there is exactly one $w \in \mathrm{pos}(s)$ such that $(s(w)_2)(x) = 1$. The subset of $T_{\Sigma_\mathcal{V}}$ containing all valid trees is denoted by $T^v_{\Sigma_\mathcal{V}}$. We put $\Sigma_\varphi = \Sigma_{\mathrm{Free}(\varphi)}$.

Every valid $\Sigma_\mathcal{V}$-tree s corresponds to a pair (t, ρ) where $t \in T_\Sigma$ and ρ is a $(\mathcal{V}, t)$-*assignment*; such an assignment is a function which maps first order variables in $\mathcal{V}$ to elements of $\mathrm{pos}(t)$ and second order variables in $\mathcal{V}$ to subsets of $\mathrm{pos}(t)$. More precisely, we say that *s and (t, ρ) correspond to each other* if $\mathrm{pos}(t) = \mathrm{pos}(s)$, t is obtained from s by replacing $s(w)$ by $s(w)_1$ for every $w \in \mathrm{pos}(t)$, and for every first order variable x, second order variable X, and $w \in \mathrm{pos}(s)$, we have that $(s(w)_2)(x) = 1$ iff $\rho(x) = w$, and $(s(w)_2)(X) = 1$ iff $w \in \rho(X)$. In the sequel we will identify a valid $\Sigma_\mathcal{V}$-tree with the corresponding pair (t, ρ).

Let s be an arbitrary $\Sigma_\mathcal{V}$-tree, x be a first order variable, and $w \in \mathrm{pos}(s)$. Then $s[x \to w]$ is the $\Sigma_{\mathcal{V}\cup\{x\}}$-labeled tree obtained from s by putting $(s[x \to w](v)_2)(x) = 1$ iff $v = w$. Similarly, if X is a second order variable and $I \subseteq \mathrm{pos}(s)$, then $s[X \to I]$ is the $\Sigma_{\mathcal{V}\cup\{X\}}$-tree obtained from s by putting $(s[X \to I](v)_2)(X) = 1$ iff $v \in I$. If here $s = (t, \rho)$, we also write $s[x \to w] = (t, \rho[x \to w])$ and $s[X \to I] = (t, \rho[X \to I])$.

Let φ be a formula in $\mathrm{MSO}(\Sigma)$ and $s = (t, \rho)$ be a valid $\Sigma_\mathcal{V}$-tree such that $\mathrm{Free}(\varphi) \subseteq \mathcal{V}$. Then the relation "$(t, \rho)$ satisfies φ", denoted by $(t, \rho) \models \varphi$, is defined as usual. We put

$$\mathcal{L}_\mathcal{V}(\varphi) = \{(t, \rho) \in T^v_{\Sigma_\mathcal{V}} \mid (t, \rho) \models \varphi\}$$

and we will simply write $\mathcal{L}(\varphi)$ instead of $\mathcal{L}_{\mathrm{Free}(\varphi)}(\varphi)$. Now we recall the equivalence between recognizable tree languages and MSO-definable tree language; cf. [39, Theorems 14 and 17], [16, Theorems 3.7 and 3.9], or [27, Proposition 12.2]: The tree language $\mathcal{L}_\mathcal{V}(\varphi)$ is recognizable over $\Sigma_\mathcal{V}$. Conversely, for every recognizable tree language $\mathcal{L}$ over Σ, there is an MSO-sentence φ such that $\mathcal{L} = \mathcal{L}(\varphi)$. It follows from this, but can also easily be shown directly, that the set $T^v_{\Sigma_\mathcal{V}}$ is recognizable.

Let us denote by $\mathrm{MSO}^-(\Sigma)$ the set of all $\mathrm{MSO}(\Sigma)$-formulas in which negation is applied only to atomic formulas. By applying de Morgan's laws and by pulling negation over a first or second order quantifier (as, e.g., by $\neg(\forall x.\varphi) \to \exists x.\neg\varphi$), we obtain the well-known fact that for every $\varphi \in \mathrm{MSO}(\Sigma)$ there is a $\psi \in \mathrm{MSO}^-(\Sigma)$ such that $\mathcal{L}(\varphi) = \mathcal{L}(\psi)$.

3. Tree series and weighted tree automata

A *semiring* is an algebraic structure $(K, +, \cdot, 0, 1)$ with operations sum $+$ and product $\cdot$ and constants 0 and 1 such that $(K, +, 0)$ is a commutative monoid and $(K, \cdot, 1)$ is a monoid, multiplication distributes over addition, and $a \cdot 0 = 0 \cdot a = 0$ for every $a \in K$. Whenever the operations and constants of a semiring are clear from the context,

we abbreviate $(K, +, \cdot, 0, 1)$ by K. The semiring K is *commutative* if $\cdot$ is commutative. Important examples of semirings are

- the *boolean semiring* $\mathbb{B} = (\{0, 1\}, \vee, \wedge, 0, 1)$ with disjunction $\vee$ and conjunction $\wedge$,
- the *semiring of natural numbers* $(\mathbb{N}, +, \cdot, 0, 1)$, abbreviated by $\mathbb{N}$, with the usual addition and multiplication,
- the *tropical semiring Trop* $= (\mathbb{N} \cup \{\infty\}, \min, +, \infty, 0)$ where the sum and the product operations are min and $+$, resp., extended to $\mathbb{N} \cup \{\infty\}$ in the usual way.

A (*formal*) *tree series* is a mapping $S : T_\Sigma \to K$ where usually (S, t) is written rather than $S(t)$. The *support of* S is the set $\mathrm{supp}(S) = \{t \in T_\Sigma \mid (S, t) \neq 0\}$, and the *image of* S is the set $\mathrm{im}(S) = \{(S, t) \mid t \in T_\Sigma\}$. The set of all tree series (over Σ and K) is denoted by $K\langle\langle T_\Sigma \rangle\rangle$. For two tree series $S, T \in K\langle\langle T_\Sigma \rangle\rangle$ and $k \in K$, the *sum* $S + T$, the *Hadamard product* $S \odot T$, and the product $k \cdot S$ are each defined pointwise for every $t \in T_\Sigma$ as follows: $(S+T, t) = (S, t)+(T, t)$, $(S \odot T, t) = (S, t) \cdot (T, t)$, and $(k \cdot S, t) = k \cdot (S, t)$.

For every $L \subseteq T_\Sigma$, the *characteristic tree series* $\mathbb{1}_L : T_\Sigma \to K$ is defined for every $t \in T_\Sigma$ by $(\mathbb{1}_L, t) = 1$ if $t \in L$, and $(\mathbb{1}_L, t) = 0$ otherwise.

A tree series $S : T_\Sigma \to K$ is a *recognizable step function* if there are an $n \geqslant 1$ and recognizable tree languages $L_1, \ldots, L_n \subseteq T_\Sigma$ such that $S = \sum_{i=1}^n k_i \cdot \mathbb{1}_{L_i}$.

Lemma 3.1. *Let $S : T_\Sigma \to K$ be a recognizable step function. Then there is a partitioning $U_1, \ldots, U_m$ of T_Σ where $U_1, \ldots, U_m$ are recognizable tree languages, and there are $l_1, \ldots, l_m \in K$ such that $S = \sum_{i=1}^m l_i \cdot \mathbb{1}_{U_i}$.*

Proof. Let $S = \sum_{i=1}^n k_i \cdot \mathbb{1}_{L_i}$ for some $n \geqslant 1$ and recognizable tree languages $L_1, \ldots, L_n \subseteq T_\Sigma$. Let F be the set of all mappings of type $\{1, \ldots, n\} \to \{1, c\}$. For every mapping $f \in F$, define the tree language $U_f = \bigcap_{i=1}^n L_i^{f(i)}$ where $L_i^1 = L_i$ and $L_i^c = T_\Sigma \setminus L_i$. Note that the U_f's form a partitioning of T_Σ. For every such f define $l_f = \sum_{i \in f^{-1}(1)} k_i$. Then clearly, $S = \sum_{f \in F} l_f \cdot \mathbb{1}_{U_f}$. $\square$

This shows that S is a recognizable step function iff $\mathrm{im}(S)$ is finite and for every $k \in K$, the set $S^{-1}(k) = \{t \in T_\Sigma \mid (S, t) = k\}$ is a recognizable tree language.

Now we introduce weighted bottom-up finite state tree automata and their behavior. For further investigations, see [2,29,11,24,33,6,19].

Let $(K, +, \cdot, 0, 1)$ be a commutative semiring. Moreover, let Q be a finite set (of states) and Σ a ranked alphabet (of input symbols). A (*bottom-up*) *tree representation* (*over Q, Σ, and K*) is a family $\mu = (\mu_m \mid m \in \mathbb{N})$ of mappings $\mu_m : \Sigma^{(m)} \to K^{Q^m \times Q}$. A *weighted bottom-up finite state tree automaton* (*over K*) (for short: *wta*) is a quadruple $M = (Q, \Sigma, \mu, \gamma)$ where μ is a tree representation over Q, Σ, and K, and $\gamma : Q \to K$ is the weight function for leaving a state (final weights). (We note that we view $K^{Q^m \times Q}$ as the set of $(Q^m \times Q)$-matrices with entries taken from K; thus, following the usual matrix notations, for every $m \geqslant 0$, $\sigma \in \Sigma^{(m)}$, and $q_1, \ldots, q_m, q \in Q$ we write $\mu_m(\sigma)_{q_1 \ldots q_m, q}$ to denote the entry in the matrix $\mu_m(\sigma)$ at row $(q_1, \ldots, q_m)$ and at column q.)

In this paper we will associate the run semantics with a wta as defined, e.g., in [19]. (We note that the run semantics is equivalent to the initial algebra semantics, cf. [6, Lemma 4.1.13].)

Let $t \in T_\Sigma$ and $r : \mathrm{pos}(t) \to Q$ be a run of M on t; also let $w \in \mathrm{pos}(t)$. The *weight* $\mathrm{wt}_M(t, r, w)$ *of r on t at w* is defined as follows: if $t(w) = \sigma \in \Sigma^{(m)}$ for some $m \geqslant 0$, then

$$\mathrm{wt}_M(t, r, w) = \mu_m(\sigma)_{r(w.1) \ldots r(w.m), r(w)}.$$

The *weight of r* on t is defined by

$$\mathrm{wt}_M(t, r) = \prod_{w \in \mathrm{pos}(t)} \mathrm{wt}_M(t, r, w)$$

(recall that K is a commutative semiring). We note that here we associate with *every* run a weight and need not define the concept of valid runs; so to speak, a run on t is "valid" or "makes sense", if all values $\mathrm{wt}_M(t, r, w)$ $(w \in \mathrm{pos}(t))$ are non-zero. Note that $\mathrm{wt}_M(t, r) = 0$ can also arise due to zero-divisors in K.

The *tree series accepted by M*, denoted by $S_M \in K\langle\langle T_\Sigma \rangle\rangle$, is the tree series defined, for every $t \in T_\Sigma$, by

$$(S_M, t) = \sum_{r \in R_M(t)} \mathrm{wt}_M(t, r) \cdot \gamma(r(\varepsilon)).$$

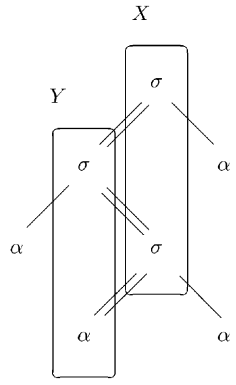

Fig. 1. Example of an input tree with its zig-zag path and the sets X and Y of nodes.

A tree series $S \in K\langle\langle T_\Sigma \rangle\rangle$ is called *recognizable* if there is a wta M over K such that $S = S_M$. The *class of all recognizable tree series* is denoted by $K^{\mathrm{rec}}\langle\langle T_\Sigma \rangle\rangle$. Clearly, $\mathrm{supp}(\mathbb{B}^{\mathrm{rec}}\langle\langle T_\Sigma \rangle\rangle)$ is the class of recognizable tree languages.

Example 3.2 (*length of the zig-zag path*). Let $\Sigma = \{\sigma^{(2)}, \alpha^{(0)}\}$ be a ranked alphabet. For every $t \in T_\Sigma$, the zig-zag path of t is the unique path of t which starts at its root and alternatingly moves down to the first or second successor of the current node, until a leaf is reached. For instance, the zig-zag path of $t = \sigma(\sigma(\alpha, \sigma(\alpha, \alpha)), \alpha)$ contains four nodes which are labeled by σ, σ, σ, and α, respectively (cf. Fig. 1 ignoring the sets X and Y at the time being). Thus, we can define the tree series $zigzag \in \mathbb{N}\langle\langle T_\Sigma \rangle\rangle$ over the natural number semiring such that for every $t \in T_\Sigma$, the value $(zigzag, t)$ is the number of nodes (or positions) on the zig-zag path of t.

Next we describe a wta $M = (Q, \Sigma, \mu, \gamma)$ over $\mathbb{N}$ such that $S_M = zigzag$. Let t be an arbitrary input tree. The idea of processing t is as follows. M can read every input subtree in the blind state d thereby producing weight 1. At every node v, M can start to check, whether v lies on the zig-zag path of t or not (check for membership on the zig-zag path); this is done by using two states 1 and 2; the state 2 has final weight $\gamma(2) = 1$, all the other states have final weight 0. Formally, we define the wta $M = (Q, \Sigma, \mu, \gamma)$ over the semiring of natural numbers as follows:

- $Q = \{1, 2, d\}$,
- $\gamma(2) = 1$ and $\gamma(1) = \gamma(d) = 0$,
- blind acceptance:

  $$\mu_0(\alpha)_{\varepsilon,d} = \mu_2(\sigma)_{dd,d} = 1,$$

  switching from blind acceptance to checking:

  $$\mu_2(\sigma)_{dd,1} = \mu_2(\sigma)_{dd,2} = 1$$

  check for membership on the zig-zag path:

  $$\mu_0(\alpha)_{\varepsilon,2} = \mu_0(\alpha)_{\varepsilon,1} = \mu_2(\sigma)_{1d,2} = \mu_2(\sigma)_{d2,1} = 1,$$

  for every other combination $q_1, q_2, q_3 \in Q$ we define

  $$\mu_2(\sigma)_{q_1q_2,q_3} = 0.$$

Clearly, we only have to take into consideration those runs on an input tree t which end up at the root in state 2 (because for every other run r we have $wt_M(t, r) \cdot \gamma(r(\varepsilon)) = 0$). Intuitively, it is clear that there are as many such runs on t as the zig-zag path has positions; every such run has weight 1; this sums up to $(zigzag, t)$. □

Next we turn to basic properties of recognizable tree series.

Lemma 3.3. *Let K be a semiring.*
(1) *Let $\mathcal{L} \subseteq T_\Sigma$ be a recognizable tree language. Then $\mathbb{1}_\mathcal{L} \in K\langle\langle T_\Sigma \rangle\rangle$ is recognizable.*

(2) *For every* $k \in K$, *the constant tree series* $\widetilde{k} \in K\langle\langle T_\Sigma\rangle\rangle$ *which maps every tree to* k, *is recognizable*.
(3) *Let* K *be commutative and* $S, T \in K\langle\langle T_\Sigma\rangle\rangle$ *be recognizable. Then* $S + T$ *and* $S \odot T$ *are recognizable*.

Proof. (1) Let $\mathcal{L} \subseteq T_\Sigma$ be accepted by the deterministic bottom-up finite state tree automaton $M = (Q, \Sigma, \delta, F)$. Then construct the wta $M_K = (Q, \Sigma, \mu, \gamma)$ over K as follows: for every $m \geqslant 0$, $\sigma \in \Sigma^{(m)}$, and $q, q_1, \ldots, q_m \in Q$, let $\mu_m(\sigma)_{q_1 \ldots q_m, q} = 1$ if $q = \delta_m(q_1, \ldots, q_m)$, and $\mu_m(\sigma)_{q_1 \ldots q_m, q} = 0$ otherwise, and for every $q \in Q$, let $\gamma(q) = 1$ if $q \in F$ and $\gamma(q) = 0$ otherwise. Clearly, $S_{M_K} = \mathbb{1}_{\mathcal{L}(M)} = \mathbb{1}_{\mathcal{L}}$.

(2) Consider the wta $M = (Q, \Sigma, \mu, \gamma)$ where $Q = \{q\}$, for every $m \geqslant 0$ and $\sigma \in \Sigma^{(m)}$, define $\mu_m(\sigma)_{q^m, q} = 1$ where $\gamma(q) = k$. Clearly, $S_M = \widetilde{k}$.

(3) Let $S, T \in K\langle\langle T_\Sigma\rangle\rangle$ be accepted by wta M_1 and M_2, respectively. We can assume that the sets of states of M_1 and M_2 are disjoint. Then we can construct in the well-known way (by "taking the union of M_1 and M_2") a wta M such that $S_M = S + T$ (cf. e.g., [19, Lemma 6.4]). The claim for $S \odot T$ has been proved in [5, Corollary 3.9] (also cf. [2, Proposition 5.1]). □

As an immediate consequence of Lemma 3.3, every recognizable step function is a recognizable series. Finally, we recall that the class of recognizable tree series is closed under relabelings. For this, let Σ and Δ be two ranked alphabets and $\tau : \Sigma \to \mathcal{P}(\Delta)$ be a mapping such that $\tau(\sigma) \subseteq \Delta^{(m)}$ for every $m \geqslant 0$ and $\sigma \in \Sigma^{(m)}$. This mapping is extended to a mapping $\tau' : T_\Sigma \to \mathcal{P}(T_\Delta)$ by defining inductively $\tau'(\sigma(s_1, \ldots, s_m)) = \{\delta(t_1, \ldots, t_m) \mid \delta \in \tau(\sigma), t_1 \in \tau'(s_1), \ldots, t_m \in \tau'(s_m)\}$ for every $m \geqslant 0$, $\sigma \in \Sigma^{(m)}$, and $s_1, \ldots, s_m \in T_\Sigma$. Note that every such mapping τ' is a relabeling (in the sense of [21, Definition 3.1]); that is, it can be computed by a one-state, linear, and nondeleting bottom-up or top-down tree transducer. Also note that the set $\{s \mid t \in \tau'(s)\}$ is finite for every $t \in T_\Delta$. We will denote τ' also by τ. Next we extend τ to a mapping $\tau : K\langle\langle T_\Sigma\rangle\rangle \to K\langle\langle T_\Delta\rangle\rangle$ by defining $(\tau(S), t) = \sum_{s \in T_\Sigma, t \in \tau(s)} (S, s)$ for every $S \in K\langle\langle T_\Sigma\rangle\rangle$ and $t \in T_\Delta$. In the sequel we call mappings like τ *relabelings*.

We note that relabelings can be computed by particular linear nondeleting tree transducers in the sense of [30]. There the closure of the class of recognizable tree series under such transducers is proved, cf. [30, Corollary 14]; however, throughout that paper it is assumed that semirings are commutative and continuous. Since we want to prove our main results for commutative semirings which are not necessarily continuous, we give a direct construction for this closure property.

Lemma 3.4. *Let* $S \in K\langle\langle T_\Sigma\rangle\rangle$ *and* $\tau : K\langle\langle T_\Sigma\rangle\rangle \to K\langle\langle T_\Delta\rangle\rangle$ *be a relabeling. If* S *is recognizable, then* $\tau(S)$ *is recognizable*.

Proof. Let $M = (Q, \Sigma, \mu, \gamma)$ be a wta over K and $\tau : \Sigma \to \mathcal{P}(\Delta)$ such that $\tau(\sigma) \subseteq \Delta^{(m)}$ for every $\sigma \in \Sigma^{(m)}$. Construct the wta $M' = (Q, \Delta, \mu', \gamma)$ over K where $\mu'_m(\delta)_{q_1 \ldots q_m, q} = \sum_{\sigma \in \Sigma, \delta \in \tau(\sigma)} \mu_m(\sigma)_{q_1 \ldots q_m, q}$ for every $m \geqslant 0$, $\delta \in \Delta^{(m)}$, and $q_1, \ldots, q_m, q \in Q$.

First we note the following auxiliary statement: for every $t \in T_\Delta$ and $r' \in R_{M'}(t)$ the equation

$$\mathrm{wt}_{M'}(t, r') = \sum_{s \in T_\Sigma, t \in \tau(s)} \mathrm{wt}_M(s, r') \tag{1}$$

holds. Note that for every $s \in T_\Sigma$ with $t \in \tau(s)$ we have $\mathrm{pos}(s) = \mathrm{pos}(t)$ and thus also $R_M(s) = R_{M'}(t)$. The proof of Eq. (1) is straightforward and thus left to the reader.

Then we can compute as follows: $(S_{M'}, t) = \sum_{r' \in R_{M'}(t)} \mathrm{wt}_{M'}(t, r') \cdot \gamma(r'(\varepsilon)) \overset{(1)}{=} \sum_{r' \in R_{M'}(t)} \sum_{s \in T_\Sigma, t \in \tau(s)} \mathrm{wt}_M(s, r') \cdot \gamma(r'(\varepsilon)) = \sum_{s \in T_\Sigma, t \in \tau(s)} \sum_{r' \in R_{M'}(t)} \mathrm{wt}_M(s, r') \cdot \gamma(r'(\varepsilon)) = \sum_{s \in T_\Sigma, t \in \tau(s)} (S_M, s) = (\tau(S_M), t)$. □

4. Weighted MSO-logic on trees

Here we will define the weighted MSO-logic on trees. This is a generalization of the weighted MSO-logic (on strings) as given in [18]. On the other hand, it generalizes (unweighted) MSO-logic on trees in the same way as weighted MSO-logic generalizes (unweighted) MSO-logic on strings. We will also provide basic properties of our weighted MSO-logic.

In this section we will always assume that the underlying semiring $(K, +, \cdot, 0, 1)$ is commutative. Moreover, we assume that Σ is a ranked alphabet.

Definition 4.1. The *set* $\mathrm{MSO}(K, \Sigma)$ *of all formulas of weighted MSO-logic over* K *and* Σ *on trees* (for short: weighted MSO-logic) is defined to be the smallest set F such that

(1) F contains all the *atomic formulas* k, $\mathrm{label}_\sigma(x)$, $\mathrm{edge}_i(x, y)$, and $x \in X$ and the negations $\neg\mathrm{label}_\sigma(x)$, $\neg\mathrm{edge}_i(x, y)$, and $\neg(x \in X)$, and

(2) if $\varphi, \psi \in F$, then also $\varphi \vee \psi$, $\varphi \wedge \psi$, $\exists x.\varphi$, $\exists X.\varphi$, $\forall x.\varphi$, $\forall X.\varphi \in F$,

where $k \in K$, $\sigma \in \Sigma$, x, y are first order variables, $1 \leqslant i \leqslant \max(rk_\Sigma(\Sigma))$, and X is a second order variable.

Clearly, if we drop formulas of the form k from $\mathrm{MSO}(K, \Sigma)$, then we obtain $\mathrm{MSO}^-(\Sigma)$. That is, $\mathrm{MSO}^-(\Sigma) \subseteq \mathrm{MSO}(K, \Sigma)$.

Definition 4.2. Let $\varphi \in \mathrm{MSO}(K, \Sigma)$ and $\mathcal{V}$ be a finite set of variables containing $\mathrm{Free}(\varphi)$. The *semantics* of φ is the formal tree series $[\![\varphi]\!]_\mathcal{V} \in K\langle\langle T_{\Sigma_\mathcal{V}}\rangle\rangle$ defined as follows: if $s \in T_{\Sigma_\mathcal{V}}$ is not valid, then we put $([\![\varphi]\!]_\mathcal{V}, s) = 0$. Otherwise, we define $([\![\varphi]\!]_\mathcal{V}, s) \in K$ inductively as follows where (t, ρ) corresponds to s.

$$([\![k]\!]_\mathcal{V}, s) = k,$$

$$([\![\mathrm{label}_\sigma(x)]\!]_\mathcal{V}, s) = \begin{cases} 1 & \text{if } t(\rho(x)) = \sigma, \\ 0 & \text{otherwise,} \end{cases}$$

$$([\![\mathrm{edge}_i(x, y)]\!]_\mathcal{V}, s) = \begin{cases} 1 & \text{if } \rho(y) = \rho(x).i, \\ 0 & \text{otherwise,} \end{cases}$$

$$([\![x \in X]\!]_\mathcal{V}, s) = \begin{cases} 1 & \text{if } \rho(x) \in \rho(X), \\ 0 & \text{otherwise ,} \end{cases}$$

$$([\![\neg\varphi]\!]_\mathcal{V}, s) = \begin{cases} 1 & \text{if } ([\![\varphi]\!]_\mathcal{V}, s) = 0, \\ 0 & \text{if } ([\![\varphi]\!]_\mathcal{V}, s) = 1 \end{cases}$$

$$\text{if } \varphi \text{ is of the form } \mathrm{label}_\sigma(x),\ \mathrm{edge}_i(x, y),\ \text{or } x \in X,$$

$$([\![\varphi \vee \psi]\!]_\mathcal{V}, s) = ([\![\varphi]\!]_\mathcal{V}, s) + ([\![\psi]\!]_\mathcal{V}, s),$$

$$([\![\varphi \wedge \psi]\!]_\mathcal{V}, s) = ([\![\varphi]\!]_\mathcal{V}, s) \cdot ([\![\psi]\!]_\mathcal{V}, s),$$

$$([\![\exists x.\varphi]\!]_\mathcal{V}, s) = \sum_{w \in \mathrm{pos}(s)} ([\![\varphi]\!]_{\mathcal{V}\cup\{x\}}, s[x \to w]),$$

$$([\![\exists X.\varphi]\!]_\mathcal{V}, s) = \sum_{I \subseteq \mathrm{pos}(s)} ([\![\varphi]\!]_{\mathcal{V}\cup\{X\}}, s[X \to I]),$$

$$([\![\forall x.\varphi]\!]_\mathcal{V}, s) = \prod_{w \in \mathrm{pos}(s)} ([\![\varphi]\!]_{\mathcal{V}\cup\{x\}}, s[x \to w]),$$

$$([\![\forall X.\varphi]\!]_\mathcal{V}, s) = \prod_{I \subseteq \mathrm{pos}(s)} ([\![\varphi]\!]_{\mathcal{V}\cup\{X\}}, s[X \to I]).$$

We write $[\![\varphi]\!]$ rather than $[\![\varphi]\!]_{\mathrm{Free}(\varphi)}$. Note that if φ is a sentence, then $[\![\varphi]\!] \in K\langle\langle T_\Sigma\rangle\rangle$. Also note that the unweighted case is obtained by considering the Boolean semiring $\mathbb{B}$: in fact, if we view an (unweighted) formula φ in $\mathrm{MSO}^-(\Sigma)$ as a formula in $\mathrm{MSO}(\mathbb{B}, \Sigma)$, and calculate $[\![\varphi]\!] \in \mathbb{B}\langle\langle T_\Sigma\rangle\rangle$, then the disjunction $\vee$ and the conjuction $\wedge$ occurring in φ are interpreted (as usual) by infimum and supremum, respectively; so, in particular, both distributivity laws hold. In general, for arbitrary commutative semirings, only conjunction distributes over disjunction.

In the weighted case we will also work with formulas from $\mathrm{MSO}^-(\Sigma)$ and, in particular, with the quantifier-free fragment of $\mathrm{MSO}^-(\Sigma)$, denoted by qf-$\mathrm{MSO}^-(\Sigma)$; this is the set of all formulas in $\mathrm{MSO}^-(\Sigma)$ which do not contain any quantifier. For such formulas, it will be necessary to disambiguate disjunctions. For this purpose, we define the syntactic transformations $(.)^+$ and $(.)^-$ on qf-$\mathrm{MSO}^-(\Sigma)$ by simultaneous induction as follows: for every $\varphi, \psi \in$ qf-$\mathrm{MSO}^-(\Sigma)$,

(1) if φ is atomic or the negation of an atomic formula, then we define $\varphi^+ = \varphi$ and $\varphi^- = \neg\varphi$ with the convention that $\neg\neg\psi$ is ψ,

(2) $(\varphi \vee \psi)^+ = \varphi^+ \vee (\varphi^- \wedge \psi^+)$ and $(\varphi \vee \psi)^- = \varphi^- \wedge \psi^-$, and
(3) $(\varphi \wedge \psi)^- = \varphi^- \vee (\varphi^+ \wedge \psi^-)$ and $(\varphi \wedge \psi)^+ = \varphi^+ \wedge \psi^+$.

Clearly, $\mathcal{L}_{\mathcal{V}}(\varphi^+) = \mathcal{L}_{\mathcal{V}}(\varphi)$ and $\mathcal{L}_{\mathcal{V}}(\varphi^-) = T^v_{\Sigma_{\mathcal{V}}} \setminus \mathcal{L}_{\mathcal{V}}(\varphi)$.

While specifying a particular tree series by means of a weighted MSO-formula it is often helpful to use the implication symbol as a macro (as in the unweighted case). However, in the weighted case, the macro has to be defined more carefully, because negation is only available on atomic formulas. Also we want to disambiguate disjunctions. Formally, let $\varphi \in \text{qf-MSO}^-(\Sigma)$ and $\psi \in \text{MSO}(K, \Sigma)$. Then we define $\varphi \to \psi$ as a shorthand for the formula $\varphi^- \vee (\varphi^+ \wedge \psi)$. If $\varphi, \psi \in \text{qf-MSO}^-(\Sigma)$, let $\varphi \leftrightarrow \psi$ abbreviate $(\varphi \to \psi) \wedge (\psi \to \varphi)$.

We can now show that the formal power series semantics of such disambiguous formulas only take 0 and 1 as values.

Observation 4.3. *Let $\varphi \in \text{qf-MSO}^-(\Sigma)$ and $\mathcal{V}$ be a finite set of variables containing* $\text{Free}(\varphi)$. *Then $[\![\varphi^+]\!]_{\mathcal{V}} = \mathbb{1}_{\mathcal{L}_{\mathcal{V}}(\varphi)}$ and $[\![\varphi^-]\!]_{\mathcal{V}} = \mathbb{1}_{\mathcal{L}_{\mathcal{V}}(\neg\varphi)}$. In particular, $[\![\varphi^+]\!]_{\mathcal{V}}$ and $[\![\varphi^-]\!]_{\mathcal{V}}$ are recognizable.*

Proof. Straightforward induction on the structure of φ. □

Next we show that universal first order quantification preserves characteristic tree series.

Observation 4.4. *Let $\varphi \in \text{MSO}^-(\Sigma)$ and $\mathcal{V} = \text{Free}(\varphi) \cup \{x\}$. If $[\![\varphi]\!]_{\mathcal{V}} = \mathbb{1}_{\mathcal{L}_{\mathcal{V}}(\varphi)}$, then $[\![\forall x.\varphi]\!] = \mathbb{1}_{\mathcal{L}(\forall x.\varphi)}$.*

Proof. Let $s = (t, \rho) \in T_{\Sigma_{\text{Free}(\forall x.\varphi)}}$. Clearly, $([\![\forall x.\varphi]\!], s) \in \{0, 1\}$. In fact,

$$\begin{aligned}([\![\forall x.\varphi]\!], s) = 1 \text{ iff } & \forall\, w \in \text{pos}(s) : ([\![\varphi]\!]_{\text{Free}(\varphi)\cup\{x\}}, s[x \to w]) = 1 \\ \text{iff } & \forall\, w \in \text{pos}(s) : s[x \to w] \in \mathcal{L}_{\text{Free}(\varphi)\cup\{x\}}(\varphi) \\ \text{iff } & s \in \mathcal{L}(\forall x.\varphi).\end{aligned}$$ □

Examples for sentences in this weighted logic for words have been given in [18]. Here we give two further examples for our setting of trees.

Example 4.5 (*number of leaves*). Let $\Sigma = \{\sigma^{(2)}, \alpha^{(0)}, \beta^{(0)}\}$ be a ranked alphabet and K be an arbitrary semiring. Consider the formula $\varphi = \exists x.\text{label}_\alpha(x)$ in $\text{MSO}(K, \Sigma)$. For every tree t, then

$$\begin{aligned}\big([\![\exists x.\text{label}_\alpha(x)]\!], t\big) &= \sum_{w \in \text{pos}(t)} ([\![\text{label}_\alpha(x)]\!]_{\{x\}}, t[x \to w]) \\ &= |\{w \in \text{pos}(t) \mid t(w) = \alpha\}|,\end{aligned}$$

the number (in K) of α-leaves of t. For instance, if $K = \mathbb{N}$, the semiring of natural numbers, this is the actual number; if $K = \mathbb{Z}/3\mathbb{Z}$, this is the actual number modulo 3; if $K = \mathbb{B}$, this is 1 if there is an α-leaf in t, and 0 otherwise.

Example 4.6 (*length of the zig-zag path*). Consider the formal tree series *zigzag* of Example 3.2 which associates to every tree the number of the positions on its zig-zag path. Here we will show that *zigzag* is $\text{MSO}(\mathbb{N}, \Sigma)$-definable. Consider the following formula in $\text{MSO}(\mathbb{N}, \Sigma)$ (cf. Fig. 1 for the input tree $\sigma(\sigma(\alpha, \sigma(\alpha, \alpha)), \alpha)$, the zig-zag path indicated as double line, and the sets X and Y of nodes):

$$\varphi = \exists X.\exists Y.(\mathit{root} \in X \setminus Y) \wedge (\mathit{edge}_1(X) = Y) \wedge (\mathit{edge}_2(Y) = X) \wedge \mathit{count}(X, Y),$$

where the four macros are defined as follows:

- $(\mathit{root} \in X \setminus Y) = \exists z.\mathit{root}(z) \wedge (z \in X) \wedge \neg(z \in Y)$,
- $\mathit{root}(z) = \forall z'.\neg\text{edge}_1(z', z) \wedge \neg\text{edge}_2(z', z)$,
- $(\mathit{edge}_1(X) = Y) = \forall x.\forall y.\text{edge}_1(x, y) \to \big(\neg(y \in X) \wedge (x \in X \leftrightarrow y \in Y)\big)$,
- $(\mathit{edge}_2(Y) = X) = \forall x.\forall y.\ \text{edge}_2(y, x) \to \big(\neg(x \in Y) \wedge (y \in Y \leftrightarrow x \in X)\big)$,
- $\mathit{count}(X, Y) = \exists z.(z \in X) \vee (z \in Y)$.

Since the root of a tree is unique, we obtain that $[\![(root \in X \setminus Y)]\!] = \mathbb{1}_L$ for $L = \{(t, \rho) \in T_{\Sigma_{\{X,Y\}}} \mid \varepsilon \in \rho(X) \setminus \rho(Y)\}$. Since the predecessor of a node is uniquely determined, by Observation 4.4 the series $[\![(edge_1(X) = Y)]\!]$ and $[\![(edge_2(Y) = X]\!]$ are recognizable step functions with values in $\{0, 1\}$. Also, for every $s = (t, \rho) \in T_{\Sigma_{\{X,Y\}}}$, the following two statements are equivalent:

(1) $([\![root \in X \setminus Y) \wedge (edge_1(X) = Y) \wedge (edge_2(Y) = X)]\!], s) = 1$,

(2)
- $\rho(X) \cup \rho(Y)$ is exactly the set of nodes of the zig-zag path through the root of s,
- the root of s is in $\rho(X) \setminus \rho(Y)$,
- for every $w \in pos(s) \cap \rho(Y)$, the predecessor of w is in $\rho(X) \setminus \rho(Y)$, and
- for every $w \in (pos(s) \cap \rho(X)) \setminus \{\varepsilon\}$, the predecessor of w is in $\rho(Y) \setminus \rho(X)$.

Then $[\![\varphi]\!] \in \mathbb{N}\langle\langle T_\Sigma \rangle\rangle$ is the tree series such that, for every tree $t \in T_\Sigma$, the number $([\![\varphi]\!], t)$ is the length of the zig-zag path through t. For instance, $([\![\varphi]\!], \sigma(\sigma(\alpha, \sigma(\alpha, \alpha)), \alpha)) = 4$. Here, $\mathbb{N}$ can also be replaced by an arbitrary commutative semiring, and similar remarks as at the end of Example 4.5 apply.

Recall that we have defined the semantics $[\![\varphi]\!]_{\mathcal{V}}$ for any finite set of variables $\mathcal{V}$ containing $\mathrm{Free}(\varphi)$. Now we show that these semantics' are consistent with each other.

Lemma 4.7. *Let $\varphi \in \mathrm{MSO}(K, \Sigma)$ and $\mathcal{V}$ a finite set of variables containing* $\mathrm{Free}(\varphi)$. *Then, for every $(t, \rho) \in T^v_{\Sigma_{\mathcal{V}}}$,*

$$([\![\varphi]\!]_{\mathcal{V}}, (t, \rho)) = ([\![\varphi]\!], (t, \rho|_{\mathrm{Free}(\varphi)})).$$

Moreover, $[\![\varphi]\!]$ is recognizable iff $[\![\varphi]\!]_{\mathcal{V}}$ is recognizable.

Proof. The first statement can be proved analogously to [18, Proposition 3.3]. For the final claim consider the mapping $\pi : \Sigma_{\mathcal{V}} \to \Sigma_\varphi$ such that $\pi(s) = (t, \rho|_{\mathrm{Free}(\varphi)})$ for every $s = (t, \rho) \in T_{\Sigma_{\mathcal{V}}}$. We first assume that $[\![\varphi]\!]$ is recognizable. We define the relabeling in the sense of Section 3 which is induced by $\pi^{-1} : \Sigma_\varphi \to \mathcal{P}(\Sigma_{\mathcal{V}})$. Then $[\![\varphi]\!]_{\mathcal{V}} = \pi^{-1}([\![\varphi]\!]) \odot \mathbb{1}_{T^v_{\Sigma_{\mathcal{V}}}}$. Thus, by Lemma 3.4, also $\pi^{-1}([\![\varphi]\!])$ is recognizable. Since $T^v_{\Sigma_{\mathcal{V}}}$ is a recognizable tree language, it follows from Lemma 3.3 that $[\![\varphi]\!]_{\mathcal{V}}$ is recognizable.

Conversely, let $[\![\varphi]\!]_{\mathcal{V}}$ be recognizable. Let $F \subseteq T^v_{\Sigma_{\mathcal{V}}}$ be the set of those trees $s = (t, \rho)$ such that $\rho(x) = \varepsilon$ (resp. $\rho(X) = \{\varepsilon\}$) for every variable x (resp. X) in $\mathcal{V} \setminus \mathrm{Free}(\varphi)$. Then F is a recognizable tree language and for every $(t, \rho') \in T^v_{\Sigma_\varphi}$ there is a unique $(t, \rho) \in F$ such that $\pi((t, \rho)) = (t, \rho')$. Then $[\![\varphi]\!] = \pi([\![\varphi]\!]_{\mathcal{V}} \odot \mathbb{1}_F)$ and by Lemmata 3.3 and 3.4 we have that $[\![\varphi]\!]$ is recognizable. □

Now let $Z \subseteq \mathrm{MSO}(K, \Sigma)$. We will call a tree series $S : T_\Sigma \to K$ *Z-definable* if there is a sentence $\varphi \in Z$ such that $S = [\![\varphi]\!]$. It has been shown in [18] that there are $\mathrm{MSO}(\Sigma, K)$-definable series which are not recognizable. Consider e.g. the formula $\varphi = \forall x.\forall y.2$ over the semiring $\mathbb{N}$ of natural numbers and a monadic ranked alphabet $\Sigma = \{\delta^{(1)}, \alpha^{(0)}\}$ (cf. [18, Example 3.4]). Then, for every $t = \delta^n(\alpha)$, $([\![\varphi]\!], t) = 2^{(n+1)^2}$. However, these values cannot be reached by any wta. Thus, we look for a careful restriction of $\mathrm{MSO}(\Sigma, K)$ by means of which the class $K^{\mathrm{rec}}\langle\langle T_\Sigma \rangle\rangle$ is characterised. We follow the lines of [18].

Definition 4.8. A formula $\varphi \in \mathrm{MSO}(K, \Sigma)$ is *restricted*, if it does not contain a universal set quantification of the form $\forall X.\psi$, and whenever φ contains a universal first order quantification of the form $\forall x.\psi$, then $[\![\psi]\!]$ is a recognizable step function.

The *set of all restricted formulas of* $\mathrm{MSO}(K, \Sigma)$ is denoted by $\mathrm{RMSO}(K, \Sigma)$. The *set of all restricted existential formulas of* $\mathrm{MSO}(K, \Sigma)$, denoted by $\mathrm{REMSO}(K, \Sigma)$, is the set of all restricted formulas of $\mathrm{MSO}(K, \Sigma)$ of the form $\varphi = \exists X_1, \ldots, X_n.\psi$ with ψ containing no set quantification. Observe that in contrast, in $\mathrm{RMSO}(K, \Sigma)$-formulas existential set quantifiers might occur also in the interior of the formula, possibly after a universal first order quantifier. The set of all tree series $S \in K\langle\langle T_\Sigma \rangle\rangle$ which are definable by some sentence in $\mathrm{RMSO}(K, \Sigma)$ (resp. in $\mathrm{REMSO}(K, \Sigma)$) is denoted by $K^{\mathrm{rmso}}\langle\langle T_\Sigma \rangle\rangle$ (resp. $K^{\mathrm{remso}}\langle\langle T_\Sigma \rangle\rangle$). An example of a formula in $\mathrm{REMSO}(K, \Sigma)$ is φ of Example 4.6.

Only a very special case of the following lemma will be needed later on. We include a general statement due to the independent importance of implications.

Lemma 4.9. *Let* $\varphi \in \text{qf-MSO}^-(\Sigma)$, *let* $\psi \in \text{MSO}(K, \Sigma)$, *and let* $\mathcal{V}$ *be a finite set of variables containing* $\text{Free}(\varphi) \cup \text{Free}(\psi)$. *Then* $[\![\varphi \to \psi]\!]_{\mathcal{V}} = [\![\varphi^-]\!]_{\mathcal{V}} + [\![\varphi^+]\!]_{\mathcal{V}} \odot [\![\psi]\!]_{\mathcal{V}}$. *Moreover, if* $[\![\psi]\!]$ *is a recognizable step function* (*resp.*, *recognizable*), *then so is* $[\![\varphi \to \psi]\!]$.

Proof. The equation claimed is immediate from the definition of the macro $\varphi \to \psi$. Since the intersection of two recognizable tree languages is again recognizable, it easily follows that the class of recognizable step functions is closed under the Hadamard product. This, Lemma 4.7, and Observation 4.3 imply the claim for $[\![\varphi \to \psi]\!]$ if $[\![\psi]\!]$ is a recognizable step function. If $[\![\psi]\!]$ is just recognizable, apply Lemma 4.7, Observation 4.3 and Lemma 3.3. □

5. The main result

The main result of this paper shows the coincidence of recognizability, RMSO-, and REMSO-definability of tree series.

Theorem 5.1. *Let* Σ *be a ranked alphabet and* K *any commutative semiring. Then* $K^{\text{rec}}\langle\langle T_\Sigma \rangle\rangle = K^{\text{rmso}}\langle\langle T_\Sigma \rangle\rangle = K^{\text{remso}}\langle\langle T_\Sigma \rangle\rangle$.

This result generalizes the corresponding result of [18] for series on words to series on trees; we obtain the word result from Theorem 5.1 by considering monadic ranked alphabets Σ. Theorem 5.1 will be proved in Sections 5.1 and 5.2. Throughout, we assume that K is a commutative semiring.

5.1. Definable tree series are recognizable

As in the unweighted case, we prove this implication by induction on the structure of the formula. We start with atomic formulas.

Lemma 5.2. *Let* $\varphi \in \text{MSO}(K, \Sigma)$ *be atomic or the negation of an atomic formula. Then* $[\![\varphi]\!]$ *is recognizable.*

Proof. If $\varphi = k$ where $k \in K$, we apply Lemma 3.3. Now let φ be of the form $\text{label}_\sigma(x)$ or $(\text{edge}_i(x, y))$ or $(x \in X)$ or a negation of these formulas. Then $\varphi \in \text{qf-MSO}^-(\Sigma)$, so $\mathcal{L}(\varphi)$ is a recognizable tree language. Hence $[\![\varphi]\!] = \mathbb{1}_{\mathcal{L}(\varphi)}$ is recognizable. □

Now we turn to disjunction and conjunction.

Lemma 5.3. *Let* $\varphi, \psi \in \text{MSO}(K, \Sigma)$ *such that* $[\![\varphi]\!]$ *and* $[\![\psi]\!]$ *are recognizable tree series. Then* $[\![\varphi \vee \psi]\!]$ *and* $[\![\varphi \wedge \psi]\!]$ *are recognizable.*

Proof. Let $\mathcal{V} = \text{Free}(\varphi) \cup \text{Free}(\psi)$. By Definition 4.2, $[\![\varphi \vee \psi]\!] = [\![\varphi]\!]_{\mathcal{V}} + [\![\psi]\!]_{\mathcal{V}}$ and $[\![\varphi \wedge \psi]\!] = [\![\varphi]\!]_{\mathcal{V}} \odot [\![\psi]\!]_{\mathcal{V}}$ which are recognizable due to Lemmas 3.3 and 4.7. □

Next we prove that tree series defined by formulas of the form $\exists x.\varphi$ or $\exists X.\varphi$ are recognizable.

Lemma 5.4. *Let* $\varphi \in \text{MSO}(K, \Sigma)$ *such that* $[\![\varphi]\!]$ *is recognizable. Then* $[\![\exists x.\varphi]\!]$ *and* $[\![\exists X.\varphi]\!]$ *are recognizable.*

Proof. $[\![\exists X.\varphi]\!]$: let $\mathcal{V} = \text{Free}(\exists X.\varphi)$; note that $X \notin \mathcal{V}$. Consider the relabeling $\pi : T_{\Sigma_{\mathcal{V}\cup\{X\}}} \to T_{\Sigma_{\mathcal{V}}}$ defined by erasing the X-row, i.e., $\pi(t, \rho) = (t, \rho|_{\mathcal{V}})$. Let $s \in T_{\Sigma_{\mathcal{V}}}$. We note that s is valid iff $s[X \to I]$ is valid for every $I \subseteq \text{pos}(s)$. Hence, by definition of $[\![\exists X.\varphi]\!]$ and of relabeling, we have for every $s \in T_{\Sigma_{\mathcal{V}}}$:

$$\begin{aligned}([\![\exists X.\varphi]\!], s) &= \sum_{I \subseteq \text{pos}(s)} ([\![\varphi]\!]_{\mathcal{V}\cup\{X\}}, s[X \to I]) \\ &= \sum_{t \in \pi^{-1}(s)} ([\![\varphi]\!]_{\mathcal{V}\cup\{X\}}, t) \\ &= (\pi([\![\varphi]\!]_{\mathcal{V}\cup\{X\}}), s).\end{aligned}$$

Thus, $[\![\exists X.\varphi]\!] = \pi([\![\varphi]\!]_{\mathcal{V}\cup\{X\}})$. Since $\mathrm{Free}(\varphi) \subseteq \mathcal{V} \cup \{X\}$ and $[\![\varphi]\!]$ is recognizable, by Lemma 4.7 also $[\![\varphi]\!]_{\mathcal{V}\cup\{X\}}$ is recognizable. Hence by Lemma 3.4, also $[\![\exists X.\varphi]\!]$ is recognizable.

$[\![\exists x.\varphi]\!]$: Let $\mathcal{V} = \mathrm{Free}(\exists x.\varphi)$; note that $x \notin \mathcal{V}$. Consider the relabeling $\pi : T_{\Sigma_{\mathcal{V}\cup\{x\}}} \to T_{\Sigma_{\mathcal{V}}}$ which is defined by erasing the x-row, as above. Let $s \in T_{\Sigma_{\mathcal{V}}}$. We note that s is valid iff $s[x \to w]$ is valid for every $w \in \mathrm{pos}(s)$. Then we have for every $s \in T_{\Sigma_{\mathcal{V}}}$:

$$\begin{aligned}([\![\exists x.\varphi]\!], s) &= \sum_{w\in\mathrm{pos}(s)} ([\![\varphi]\!]_{\mathcal{V}\cup\{x\}}, s[x \to w]) \\ &=^{(1)} \sum_{\substack{t\in\pi^{-1}(s)\\ t \text{ is valid}}} ([\![\varphi]\!]_{\mathcal{V}\cup\{x\}}, t) \\ &=^{(2)} \sum_{t\in\pi^{-1}(s)} ([\![\varphi]\!]_{\mathcal{V}\cup\{x\}}, t) \\ &= (\pi([\![\varphi]\!]_{\mathcal{V}\cup\{x\}}), s).\end{aligned}$$

At (1) observe that, if s is valid, there is a bijection between the index sets $\mathrm{pos}(s)$ and $T^v_{\Sigma_{\mathcal{V}\cup\{x\}}} \cap \pi^{-1}(s)$; if s is not valid, then the set $T^v_{\Sigma_{\mathcal{V}\cup\{x\}}} \cap \pi^{-1}(s)$ is empty. For (2), note that $([\![\varphi]\!]_{\mathcal{V}\cup\{x\}}, t) = 0$ for every non-valid t. Now we proceed as in the previous case and obtain that $[\![\exists x.\varphi]\!]$ is recognizable. □

In the next lemma we prove that tree series definable by formulas of the form $\forall x.\varphi$ are recognizable if $[\![\varphi]\!]$ is a recognizable step function. The idea of the construction of the wta is the same as in [18, Lemma 4.4]. However, here we also have to "synchronize" variables in different subtrees (a more detailed explanation is given inside the proof); this problem does not arise in the case of words.

Lemma 5.5. *Let $\varphi \in \mathrm{MSO}(K, \Sigma)$ such that $[\![\varphi]\!]$ is a recognizable step function. Then $[\![\forall x.\varphi]\!]$ is recognizable.*

Proof. Let $\mathcal{W} = \mathrm{Free}(\varphi)$ and $\mathcal{V} = \mathrm{Free}(\forall x.\varphi) = \mathcal{W} \setminus \{x\}$. By assumption, $[\![\varphi]\!] = \sum_{j=1}^n k_j \cdot \mathbb{1}_{L_j}$ for some $n \geqslant 1$, $k_j \in K$, and recognizable tree languages $L_1, \ldots, L_n \subseteq T_{\Sigma_{\mathcal{W}}}$. By Lemma 3.1 we can assume that the sets $L_1, \ldots, L_n$ form a partition of $T_{\Sigma_{\mathcal{W}}}$.

First we assume that $x \in \mathcal{W}$. Let $\widetilde{\Sigma} = \Sigma \times \{1, \ldots, n\}$ be the ranked alphabet with rank function $rk((\sigma, j)) = rk_\Sigma(\sigma)$ for every $(\sigma, j) \in \widetilde{\Sigma}$. A tree $s \in T_{\widetilde{\Sigma}_{\mathcal{V}}}$ corresponds to the tuple (s', v) where $s' \in T_{\Sigma_{\mathcal{V}}}$ is obtained from s by dropping the second component from the label of every node, and $v : \mathrm{pos}(s) \to \{1, \ldots, n\}$ is defined by $v(w) = j$ if $s(w) = (\sigma, j, f)$ for some $\sigma \in \Sigma$ and $f \in \{0, 1\}^{\mathcal{V}}$. Vice versa, every such tuple (s', v) corresponds to a tree $s \in T_{\widetilde{\Sigma}_{\mathcal{V}}}$. Hence we can assume that elements of $T_{\widetilde{\Sigma}_{\mathcal{V}}}$ have the form (s', v). Then let

$$\widetilde{L} = \{(s', v) \in T_{\widetilde{\Sigma}_{\mathcal{V}}} \mid \forall w \in \mathrm{pos}(s'), 1 \leqslant j \leqslant n : \text{ if } v(w) = j, \text{ then } s'[x \to w] \in L_j\}.$$

Note that for every $s' \in T_{\Sigma_{\mathcal{V}}}$ there is a unique v such that $(s', v) \in \widetilde{L}$, because the L_j's form a partition of $T_{\Sigma_{\mathcal{W}}}$. Next we prove that $\widetilde{L}$ is a recognizable tree language. In fact, $\widetilde{L} = \bigcap_{1\leqslant j\leqslant n} \widetilde{L_j}$ where

$$\widetilde{L_j} = \{(s', v) \in T_{\widetilde{\Sigma}_{\mathcal{V}}} \mid \forall w \in \mathrm{pos}(s') : \text{ if } v(w) = j, \text{ then } s'[x \to w] \in L_j\}.$$

We prove that $\widetilde{L_j}$ is recognizable. Let $M_j = (Q, \Sigma_{\mathcal{W}}, \delta, F)$ be a deterministic bottom-up finite state tree automaton such that $\mathcal{L}(M_j) = L_j$. We will construct the deterministic bottom-up finite state tree automaton $\widetilde{M_j} = (\widetilde{Q}, \widetilde{\Sigma}_{\mathcal{V}}, \widetilde{\delta}, \widetilde{F})$ which recognizes $\widetilde{L_j}$. The main idea in this construction is taken from the corresponding result of [18] and it is roughly described as follows. On an input tree $s = (s', v)$, the automaton $\widetilde{M_j}$ simulates the work of M_j on s' and, whenever a position w of s' is encountered for which $v(w) = j$ holds, then, additionally, $\widetilde{M_j}$ splits off a copy of M_j; this copy behaves as if at w the x would occur. However, here we have to refine this idea a bit, because we have to guarantee that the placement of x is done at most once in s'. For this, we maintain a bit b in every state of $\widetilde{M_j}$ which indicates whether the x was placed ($b = 1$) or not ($b = 0$). Then, while traversing with $\widetilde{M_j}$ over the input tree s, we only form the possible transitions on a k-ary symbol (σ, l, f) from those states of the successors for which the sum of their bits is not greater than 1.

Formally, we construct $\widetilde{M}_j = (\widetilde{Q}, \widetilde{\Sigma}_{\mathcal{V}}, \widetilde{\delta}, \widetilde{F})$ by setting $\widetilde{Q} = \mathcal{P}(Q \times \{0,1\})$ and $\widetilde{F} = \{U \subseteq Q \times \{0,1\} \mid U \cap (Q \times \{1\}) \subseteq F \times \{1\}\}$, and $\widetilde{\delta} = \{\widetilde{\delta}_{(\sigma,l,f)}\}_{(\sigma,l,f)\in\widetilde{\Sigma}_{\mathcal{V}}}$ is defined for every $(\sigma, l, f) \in \widetilde{\Sigma}_{\mathcal{V}}^{(m)}$, $m \geqslant 0$, and $P_1, \ldots, P_m \in \widetilde{Q}$ by

$$\widetilde{\delta}_{(\sigma,l,f)}(P_1, \ldots, P_m) = P,$$

where

$$P = \begin{cases} P' & \text{if } l \neq j, \\ P' \cup P'' & \text{if } l = j. \end{cases}$$

and

$$P' = \left\{ \left(\delta_{(\sigma, f[x\to 0])}(p_1, \ldots, p_m), \sum_{i=1}^{m} b_i \right) \middle| (p_i, b_i) \in P_i \text{ for every } 1 \leqslant i \leqslant m \text{ and } \sum_{i=1}^{m} b_i \leqslant 1 \right\}$$

and

$$P'' = \left\{ \left(\delta_{(\sigma, f[x\to 1])}(p_1, \ldots, p_m), 1 \right) \mid (p_i, 0) \in P_i \text{ for every } 1 \leqslant i \leqslant m \right\}.$$

Roughly speaking, P' formalizes the propagation of the bit value 0, if all successors of the current symbol (labeled by (σ, l, f)) have bit value 0, or of the bit value 1, if exactly one of the successors has bit value 1. If $l = j$, then additionally to this propagation, $\widetilde{M}_j$ can change from the bit vector $(0, \ldots, 0)$ to the bit value 1, thereby placing the x to this position and splitting off a new copy of M_j. Now we will show that $\widetilde{M}_j$ recognizes $\widetilde{L}_j$.

Since M_j and $\widetilde{M}_j$ are deterministic, we have that for every $(s', v) \in T_{\widetilde{\Sigma}_{\mathcal{V}}}$ the sets $R^v_{M_j}(s'[x \to w])$ and $R^v_{\widetilde{M}_j}((s', v))$ of runs are singletons. Recall that these runs are denoted by $r_{s'[x\to w]}$ and $r_{(s',v)}$, respectively. Then for every $(s', v) \in T_{\widetilde{\Sigma}_{\mathcal{V}}}$ we have that

$$r_{(s',v)}(\varepsilon) = \left\{ \left(r_{s'[x\to\emptyset]}(\varepsilon), 0 \right) \right\} \cup \{ (r_{s'[x\to w]}(\varepsilon), 1) \mid w \in \mathrm{pos}(s'), v(w) = j \}, \tag{2}$$

where $s'[x \to \emptyset] \in T_{\Sigma_{\mathcal{V}}}$ denotes the tree obtained from $s' \in T_{\Sigma_{\mathcal{V}}}$ by replacing every label (σ, f) by (σ, f_x) and $f_x : \mathcal{V} \cup \{x\} \to \{0,1\}$ extends f by setting $f_x(x) = 0$. Eq. (2) can be proved by structural induction on $s = (s', v)$ as follows.

Let $s = (\sigma, l, f)(s_1, \ldots, s_m) \in T_{\widetilde{\Sigma}_{\mathcal{V}}}$ with $s = (s', v)$ and $s_i = (s'_i, v_i)$ for every $1 \leqslant i \leqslant m$. Moreover, let $s' = (\sigma, f)(s'_1, \ldots, s'_m)$. We assume that Eq. (2) holds for $s_1, \ldots, s_m$ (I.H.). Then we first prove the following equation:

$$\begin{aligned} &\left\{ \left(\delta_{(\sigma, f[x\to 0])}(p_1, \ldots, p_m), \sum_{i=1}^{m} b_i \right) \middle| (p_i, b_i) \in r_{s_i}(\varepsilon), \sum_{i=1}^{m} b_i \leqslant 1 \right\} \\ &= \{ (r_{s'[x\to\emptyset]}(\varepsilon), 0) \} \cup \{ (r_{s'[x\to uw]}(\varepsilon), 1) \mid 1 \leqslant u \leqslant m, w \in \mathrm{pos}(s'_u), v_u(w) = j \}. \end{aligned} \tag{3}$$

Proof of Eq. (3).

$$\begin{aligned} &\left\{ \left(\delta_{(\sigma, f[x\to 0])}(p_1, \ldots, p_m), \sum_{i=1}^{m} b_i \right) \middle| \text{ for every } 1 \leqslant i \leqslant m\text{: } (p_i, b_i) \in r_{s_i}(\varepsilon) \text{ and } \sum_{i=1}^{m} b_i \leqslant 1 \right\}, \\ &\overset{\text{I.H.2}}{=} \left\{ \left(\delta_{(\sigma, f[x\to 0])}(p_1, \ldots, p_m), \sum_{i=1}^{m} b_i \right) \middle| \text{ for every } 1 \leqslant i \leqslant m : \right. \\ &\qquad \left. [(p_i, b_i) = (r_{s'_i[x\to\emptyset]}(\varepsilon), 0) \text{ or } (p_i, b_i) = (r_{s'_i[x\to w]}(\varepsilon), 1), w \in \mathrm{pos}(s'_i), v_i(w) = j] \text{ and } \sum_{i=1}^{m} b_i \leqslant 1 \right\}, \\ &= \left\{ \left(\delta_{(\sigma, f[x\to 0])}(r_{s'_1[x\to\emptyset]}(\varepsilon), \ldots, r_{s'_m[x\to\emptyset]}(\varepsilon)), 0 \right) \right\} \\ &\qquad \cup \left\{ (\delta_{(\sigma, f[x\to 0])}(r_{s'_1[x\to\emptyset]}(\varepsilon), \ldots, r_{s'_u[x\to w]}(\varepsilon), \ldots, r_{s'_m[x\to\emptyset]}(\varepsilon)), 1) | 1 \leqslant u \leqslant m, w \in \mathrm{pos}(s'_u), v_u(w) = j \right\}, \\ &= \left\{ \left(r_{s'[x\to\emptyset]}(\varepsilon), 0 \right) \right\} \cup \left\{ \left(r_{s'[x\to uw]}(\varepsilon), 1 \right) \mid 1 \leqslant u \leqslant m, w \in \mathrm{pos}(s'_u), v_u(w) = j \right\}. \end{aligned}$$

This finishes the proof of Eq. (3). Now let $l \neq j$. Then

$$\begin{aligned}
r_s(\varepsilon) &= \widetilde{\delta}_{(\sigma,l,f)}(r_s(1), \dots, r_s(m)) = \widetilde{\delta}_{(\sigma,l,f)}(r_{s_1}(\varepsilon), \dots, r_{s_m}(\varepsilon)) \\
&= \left\{ \left(\delta_{(\sigma, f[x\to 0])}(p_1, \dots, p_m), \sum_{i=1}^{m} b_i \right) \middle| (p_i, b_i) \in r_{s_i}(\varepsilon), \sum_{i=1}^{m} b_i \leqslant 1 \right\} \\
&= \left\{ \left(r_{s'[x\to\emptyset]}(\varepsilon), 0\right) \right\} \cup \left\{ \left(r_{s'[x\to uw]}(\varepsilon), 1\right) \mid 1 \leqslant u \leqslant m,\ w \in \mathrm{pos}(s'_u),\ v_u(w) = j \right\} \text{ (by Eq. (3))} \\
&= \left\{ \left(r_{s'[x\to\emptyset]}(\varepsilon), 0\right) \right\} \cup \left\{ \left(r_{s'[x\to w]}(\varepsilon), 1\right) \mid w \in \mathrm{pos}(s'),\ v(w) = j \right\}.
\end{aligned}$$

Now let $l = j$. Then

$$\begin{aligned}
r_s(\varepsilon) &= \widetilde{\delta}_{(\sigma,l,f)}(r_{s_1}(\varepsilon), \dots, r_{s_m}(\varepsilon)) \\
&= \left\{ \left(\delta_{(\sigma, f[x\to 0])}(p_1, \dots, p_m), \sum_{i=1}^{m} b_i \right) \middle| (p_i, b_i) \in r_{s_i}(\varepsilon), \sum_{i=1}^{m} b_i \leqslant 1 \right\} \\
&\quad \cup \{(\delta_{(\sigma, f[x\to 1])}(r_{s'_1[x\to\emptyset]}(\varepsilon), \dots, r_{s'_m[x\to\emptyset]}(\varepsilon)), 1)\} \\
&= \left\{ \left(r_{s'[x\to\emptyset]}(\varepsilon), 0\right) \right\} \cup \left\{ \left(r_{s'[x\to uw]}(\varepsilon), 1\right) \mid 1 \leqslant u \leqslant m,\ w \in \mathrm{pos}(s'_u),\ v_u(w) = j \right\} \cup \{(r_{s'[x\to\varepsilon]}(\varepsilon), 1)\} \\
&\quad \text{(by Eq.(3)and definition of run)} \\
&= \left\{ \left(r_{s'[x\to\emptyset]}(\varepsilon), 0\right) \right\} \cup \left\{ \left(r_{s'[x\to w]}(\varepsilon), 1\right) \mid w \in \mathrm{pos}(s'),\ v(w) = j \right\}.
\end{aligned}$$

This finishes the proof of Eq. (2). Also, by the assumption that $\mathcal{L}(M_j) = L_j$, for every $w \in \mathrm{pos}(s')$, we have

$$s'[x \to w] \in L_j \quad \text{iff } r_{s'[x\to w]}(\varepsilon) \in F.$$

It follows that $(s', v) \in \widetilde{L_j}$ iff (for every $w \in \mathrm{pos}(s')$: $v(w) = j$ implies $r_{s'[x\to w]}(\varepsilon) \in F$) iff* $r_{(s',v)}(\varepsilon) \in \widetilde{F}$ iff $(s', v) \in \mathcal{L}(\widetilde{M_j})$ where we use Eq. (2) at the equivalence, which is indicated by the $*$. Hence $\widetilde{L_j}$ is recognizable by $\widetilde{M_j}$.

Thus, there is a bottom-up finite state tree automaton $\widetilde{M} = (\widetilde{Q}, \widetilde{\Sigma}_{\mathcal{V}}, \widetilde{\delta}, \widetilde{F})$ such that $\mathcal{L}(\widetilde{M}) = \widetilde{L}$. We can assume that $\widetilde{M}$ is deterministic. Finally, we construct the wta $M = (\widetilde{Q}, \widetilde{\Sigma}_{\mathcal{V}}, \mu, \gamma)$ by defining, for every $m \geqslant 0$, $(\sigma, l, f) \in \widetilde{\Sigma}_{\mathcal{V}}$, and $q_1, \dots, q_m, q \in \widetilde{Q}$,

$$\mu_m((\sigma, l, f))_{q_1 \dots q_m, q} = \begin{cases} k_l & \text{if } \widetilde{\delta}_{(\sigma,l,f)}(q_1, \dots, q_m) = q, \\ 0 & \text{if otherwise.} \end{cases}$$

Moreover, for every $q \in \widetilde{Q}$, we define $\gamma(q) = 1$ if $q \in \widetilde{F}$ and 0 otherwise. Clearly, also M is deterministic. Thus, it should be clear that for every $(s', v) \in T_{\widetilde{\Sigma}_{\mathcal{V}}}$ (note that K is commutative)

$$(S_M, (s', v)) = \begin{cases} \prod_{1 \leqslant j \leqslant n} k_j^{|v^{-1}(j)|} & \text{if } (s', v) \in \widetilde{L}, \\ 0 & \text{if otherwise.} \end{cases}$$

Now we define the relabeling $\tau : K\langle\langle T_{\widetilde{\Sigma}_{\mathcal{V}}}\rangle\rangle \to K\langle\langle T_{\Sigma_{\mathcal{V}}}\rangle\rangle$ by $\tau((\sigma, v, f)) = (\sigma, f)$ for every $(\sigma, v, f) \in \widetilde{\Sigma}_{\mathcal{V}}$. Then for every $s' \in T_{\Sigma_{\mathcal{V}}}$:

$$\begin{aligned}
(\tau(S_M), s') &= \sum_{(s',\theta) \in \tau^{-1}(s')} (S_M, (s', \theta)) \\
&= (S_M, (s', v)) \quad \text{where } v : \mathrm{pos}(s') \to \{1, \dots, n\} \text{ is the unique mapping such that } (s', v) \in \widetilde{L} \\
&= \prod_{1 \leqslant j \leqslant n} k_j^{|v^{-1}(j)|} \\
&= \prod_{w \in \mathrm{pos}(s')} ([\![\varphi]\!], s'[x \to w]) \quad \text{due to the fact that } [\![\varphi]\!] \text{ is a recognizable step function} \\
&= ([\![\forall x.\varphi]\!], s').
\end{aligned}$$

Hence $\tau(S_M) = [\![\forall x.\varphi]\!]$ and thus, by Lemma 3.4 $[\![\forall x.\varphi]\!]$ is recognizable.

Now assume that $x \notin \mathcal{W}$, and thus $\mathcal{V} = \mathcal{W}$. Then we consider the formula $\varphi' = \varphi \vee \mathrm{edge}_1(x, x)$. By Lemmas 5.2 and 5.3 we have that $[\![\varphi']\!]$ is recognizable, in particular, $[\![\varphi']\!]$ is a recognizable step function, and by an easy calculation,

we obtain that $[\![\varphi']\!]_{\mathcal{V}\cup\{x\}} = [\![\varphi]\!]_{\mathcal{V}\cup\{x\}}$. Thus, $[\![\forall x.\varphi']\!]_{\mathcal{V}} = [\![\forall x.\varphi]\!]_{\mathcal{V}}$ which is recognizable by what we have shown above. □

Then the next theorem follows from Lemmata 5.2, 5.3, 5.4, and 5.5.

Theorem 5.6. $K^{\mathrm{rmso}}\langle\langle T_\Sigma\rangle\rangle \subseteq K^{\mathrm{rec}}\langle\langle T_\Sigma\rangle\rangle$.

Next we turn to effectiveness and decidability questions and show results for a particular class of semirings, viz. computable fields. We will need the following preparation.

Proposition 5.7 (*Seidl [36, Theorem 4.2], also cf. Bozapalidis [10]*). *Let K be a computable field. It is decidable for two effectively given weighted bottom-up finite state tree automata M and M', whether $S_M = S'_M$ or not.*

Using this, we can show:

Corollary 5.8. *Let K be a computable field. There is an effective procedure which produces, for a given restricted* MSO(K, Σ)*-formula φ, a weighted bottom-up finite state tree automaton M such that $[\![\varphi]\!] = S_M$.*

Proof. By induction on the structure of φ. For this, we have to show that all our preceding proofs are constructive. This is mostly well known or easy, and it remains to consider the proof of Lemma 5.5. So let $[\![\varphi]\!]$ be a recognizable step function. We may assume that $[\![\varphi]\!] = S_M$ for some given wta M, and we have to construct a wta which accepts the tree series $[\![\forall x.\varphi]\!]$.

We list all tuples $(k_1, \dots, k_n, M_1, \dots, M_n)$ such that $n \geqslant 1$, $k_1, \dots, k_n \in K$ and $M_1, \dots, M_n$ are deterministic bottom-up finite state tree automata. For each such tuple, we construct a wta M' such that $S_{M'} = \sum_{1\leqslant j\leqslant n} k_j \cdot \mathbb{1}_{L(M_j)}$, and we use Proposition 5.7 to check whether $S_M = S_{M'}$. Since S_M *is* a recognizable step function, working through our list we will eventually obtain a tuple for which this equality holds. Now we can use the values k_j and automata M_j of this tuple in our proof of Lemma 5.5 to construct the required weighted automaton for $[\![\forall x.\varphi]\!]$. □

The result of Corollary 5.8 also holds for other semirings K than fields, see Section 6. However, whether it holds for all commutative semirings remains open at present. As an immediate consequence of Corollary 5.8 and Proposition 5.7 we obtain:

Corollary 5.9. *Let K be a computable field, and let φ and φ' be two* RMSO(K, Σ)*-sentences. Then it is decidable whether $[\![\varphi]\!] = [\![\varphi']\!]$.*

5.2. Recognizable tree series are definable

Here, let K again denote an arbitrary commutative semiring. In this section we show that recognizable tree series are REMSO-definable. That is, given a wta M, we will explicitly present a formula ξ of weighted REMSO-logic such that $S_M = [\![\xi]\!]$. As in the proof of the weighted case for strings [18], ξ can be chosen to have the form $\exists X_1 \dots \exists X_n.\varphi$ where every second order variable X_i corresponds to a transition of M and identifies those nodes of a tree t at which M applies this transition. The subformula φ represents a particular run of M, i.e., it (1) guarantees that the sets of nodes which are identified by $X_1, \dots, X_n$, form a partitioning of the set of all nodes of t, and that the transitions of neighbored nodes match (cf. the subformula $\psi_{M,v}$ given below), (2) computes the weight of the run, and (3) computes the weight for leaving M at the root of t. We prepare this theorem by defining some formulas and notions which are relevant for the description of runs of a wta by means of weighted logic.

Definition 5.10. Let $M = (Q, \Sigma, \mu, \gamma)$ be a wta.

- Let $X_1, \dots, X_n$ be set variables. The *partition formula associated with* $X_1, \dots, X_n$ is the formula partition $\in$ MSO$^-(\Sigma)$ defined as follows:

$$\text{partition} = \forall x.\left(\bigvee_{1\leqslant i\leqslant n}\left((x \in X_i) \wedge \bigwedge_{\substack{1\leqslant j\leqslant n\\ j\neq i}} \neg(x \in X_j)\right)\right).$$

- The *set of all transitions at Σ-symbols* is the set $B_M = \{(\vec{q}, \sigma, q) \mid m \geqslant 0, \vec{q} \in Q^m, \sigma \in \Sigma^{(m)}, q \in Q\}$. An *enumeration of the set* B_M is a bijection $\nu : B_M \to \{1, \ldots, n\}$ where $n = |B_M|$; in the sequel we will abbreviate $X_{\nu((\vec{q},\sigma,q))}$ by $X_{\vec{q},\sigma,q}$.
- Let ν be an enumeration of B_M. The *run-formula associated with M and ν* is the formula $\psi_{M,\nu} \in \mathrm{MSO}^-(\Sigma)$ with free variables $X_1, \ldots, X_n$ defined as follows:

$$\psi_{M,\nu} = \text{partition} \wedge \psi_{M,\nu,1} \wedge \psi_{M,\nu,2},$$

where

$$\psi_{M,\nu,1} = \bigwedge_{(\vec{q},\sigma,q)\in B_M} \forall x.\Big((x \in X_{\vec{q},\sigma,q}) \to \mathrm{label}_\sigma(x)\Big),$$

$$\psi_{M,\nu,2} = \bigwedge_{\substack{(\vec{q},\sigma,q)\in B_M,\\ \vec{q}=(q_1,\ldots,q_m)}} \forall x.\left((x \in X_{\vec{q},\sigma,q}) \to \exists y_1 \ldots \exists y_m.\left(\mathrm{edge}(x, y_1 \ldots y_m) \wedge \bigvee_{\substack{(\vec{p}_1,\sigma_1,q_1)\in B_M,\\ \cdots\\ (\vec{p}_m,\sigma_m,q_m)\in B_M}} (y_1 \in X_{\vec{p}_1,\sigma_1,q_1}) \wedge \cdots \wedge (y_m \in X_{\vec{p}_m,\sigma_m,q_m})\right)^+\right),$$

$$\mathrm{edge}(x, y_1 \ldots y_m) = \mathrm{edge}_1(x, y_1) \wedge \ldots \wedge \mathrm{edge}_m(x, y_m).$$

Let us make some observations on the formulas in Definition 5.10. These formulas do not contain any set quantification and whenever there is a subformula of the form $\forall x.\psi$, then $[\![\psi]\!]$ is a recognizable step function. Thus, the formula $\psi_{M,\nu}$ is in $\mathrm{RMSO}(K, \Sigma)$. Moreover, by Observation 4.4 we have that $[\![\psi_{M,\nu}]\!]\mathcal{V} = \mathbb{1}_{\mathcal{L}_\mathcal{V}(\psi_{M,\nu})}$.

Theorem 5.11. $K^{\mathrm{rec}}\langle\langle T_\Sigma\rangle\rangle \subseteq K^{\mathrm{remso}}\langle\langle T_\Sigma\rangle\rangle$.

Proof. Let $M = (Q, \Sigma, \mu, \gamma)$ be a wta, $\nu : B_M \to \{1, \ldots, n\}$ be an enumeration of the set B_M, and $\mathcal{V} = \{X_1, \ldots, X_n\}$. We put $\psi = \psi_{M,\nu}$. Let $t \in T_\Sigma$. First we show that there is a bijection between the set $R_M(t)$ of all runs of M on t and the set

$$\mathrm{Ass}_{t,M} = \{\rho \mid \rho \text{ is a } (\mathcal{V}, t)\text{-assignment}, (t, \rho) \vDash \psi\}$$

of all $(\mathcal{V}, t)$-assignments which satisfy ψ. For this, let $r : pos(t) \to Q$ be a run of M on t. We define the $(\mathcal{V}, t)$-assignment $\rho_r : \mathcal{V} \to \mathcal{P}(\mathrm{pos}(t))$ as follows: For every $1 \leqslant i \leqslant n$, if $i = \nu(((q_1, \ldots, q_m), \sigma, q))$, then define $\rho_r(X_i) = \{w \in \mathrm{pos}(t) \mid r(w.1) = q_1, \ldots, r(w.m) = q_m, t(w) = \sigma, r(w) = q\}$. It is obvious that $(t, \rho_r) \vDash \psi$. Conversely, let ρ be a $(\mathcal{V}, t)$-assignment such that $(t, \rho) \vDash \psi$. Then define the run $r_\rho : \mathrm{pos}(t) \to Q$ as follows: for every $w \in \mathrm{pos}(t)$, due to the fact that in particular $(t, \rho) \vDash \text{partition}$, there are uniquely determined $m \geqslant 0$, $\vec{q} \in Q^m$, $\sigma \in \Sigma^{(m)}$, and $q \in Q$ such that $w \in \rho(X_{\vec{q},\sigma,q})$, and we put $r_\rho(w) = q$. Since $(t, \rho) \vDash \psi$, it follows that r_ρ is a run on t.

Now consider the following formula of weighted MSO-logic:

$$\varphi = \psi \wedge \bigwedge_{(\vec{q},\sigma,q)\in B_M} \forall x.\Big((x \in X_{\vec{q},\sigma,q}) \to \mu_m(\sigma)_{\vec{q},q}\Big) \wedge \exists z.\left(root(z) \wedge \bigvee_{(\vec{q},\sigma,q)\in B_M} (z \in X_{\vec{q},\sigma,q}) \wedge \gamma(q)\right),$$

where $root(z) = \forall x.\big(\neg\mathrm{edge}_1(x, z) \wedge \cdots \wedge \neg\mathrm{edge}_{\max_\Sigma}(x, z)\big)$. Clearly, for every $t \in T_\Sigma$ there is exactly one $w \in \mathrm{pos}(t)$ such that $(t, [z \to w]) \vDash root(z)$, namely $w = \varepsilon$. By Lemma 4.9, φ is restricted.

Let $t \in T_\Sigma$ and $r : \mathrm{pos}(t) \to Q$ be a run of M on t. By the discussion above there is a uniquely determined $(\mathcal{V}, t)$-assignment ρ_r such that $(t, \rho_r) \vDash \psi$. Then we can calculate as follows:

$$\begin{aligned}
[\![\varphi]\!]_{\mathcal{V}}(t, \rho_r) &= \left(\prod_{(\vec{q},\sigma,q)\in B_M} \mu_m(\sigma)_{\vec{q},q}^{|\rho_r(X_{\vec{q},\sigma,q})|} \right) \cdot \gamma(r(\varepsilon)) \\
&\quad \text{(since } K \text{ is commutative, we can now permute the occurrences of values } \mu_m(\sigma)_{\vec{q},q}) \\
&= \left(\prod_{w\in \mathrm{pos}(t)} \mu_m(t(w))_{r(w.1)\ldots r(w.m),r(w)} \right) \cdot \gamma(r(\varepsilon)) \\
&\quad \text{(where } m = rk_\Sigma(t(w))) \\
&= \left(\prod_{w\in \mathrm{pos}(t)} \mathrm{wt}_M(t, r, w) \right) \cdot \gamma(r(\varepsilon)) \\
&= \mathrm{wt}_M(t, r) \cdot \gamma(r(\varepsilon)).
\end{aligned}$$

Now let $\xi = \exists X_1 \ldots \exists X_n.\varphi$. Then $\xi \in \mathrm{REMSO}(K, \Sigma)$. We show that $S_M = [\![\xi]\!]$. Let $t \in T_\Sigma$ and put $\mathcal{V} = \{X_1, \ldots, X_n\}$. Then

$$\begin{aligned}
[\![\xi]\!](t) &= \sum_{\rho \text{ is a } (\mathcal{V},t)\text{-assignm.}} [\![\varphi]\!](t, \rho) \\
&= \sum_{\substack{\rho \text{ is a } (\mathcal{V},t)\text{-assignm.} \\ (t,\rho)\vDash\psi}} [\![\varphi]\!](t, \rho) =^* \sum_{r\in R_M(t)} [\![\varphi]\!](t, \rho_r) \\
&= \sum_{r\in R_M(t)} \mathrm{wt}_M(t, r) \cdot \gamma(r(\varepsilon)) = (S_M, t).
\end{aligned}$$

where the equation marked by $*$ is due to the bijection between runs and assignments shown above. □

Theorems 5.6 and 5.11 imply the main result, Theorem 5.1.

Let us extend the syntax of our weighted logic by adding an atomic formula $root(z)$ which holds for (t, ρ) iff $\rho(z) = \varepsilon$, the root of t. That is

$$\big([\![root(z)]\!], (t, \rho)\big) = \begin{cases} 1 & \text{if } \rho(z) = \varepsilon, \\ 0 & \text{otherwise.} \end{cases}$$

Then an easy analysis of the proof of Theorem 5.11 shows that we can restrict ourselves to just one application of the universal first order quantifier.

Corollary 5.12. *Let $S \in K^{\mathrm{rec}}\langle\langle T_\Sigma \rangle\rangle$. Then $S = [\![\xi]\!]$ for a formula ξ of the form $\xi = \exists X_1 \ldots \exists X_n.\forall x.\theta$ where θ contains the formula $root(z)$ and only first order existential quantifications, and $[\![\theta]\!]$ is a recognizable step function.*

6. Locally finite semirings

As noted before, easy examples show that there are $\mathrm{MSO}(K, \Sigma)$-definable series which are not recognizable (compare [18]). In this section, we wish to show that for a large class of semirings K, all $\mathrm{MSO}(K, \Sigma)$-definable tree series are recognizable. Again let Σ denote an arbitrary ranked alphabet.

If K is a semiring and $A \subseteq K$, we denote by $\langle A \rangle$ the subsemiring of K generated by A. Clearly, we may construct $\langle A \rangle$ by first taking the closure A' of $A \cup \{0, 1\}$ under multiplication, then $\langle A \rangle$ is the closure of A' under addition. A semiring K is called *locally finite*, if each finitely generated subsemiring of K is finite. A monoid is called locally finite, if each finitely generated submonoid is finite. Clearly, a semiring $(K, +, \cdot, 0, 1)$ is locally finite iff both monoids $(K, +, 0)$ and $(K, \cdot, 1)$ are locally finite (apply the remark on $\langle A \rangle$ above, or cf. [17, Section 4] or [9, Lemma 2.5]).

For example, any Boolean algebra $(B, \vee, \wedge, 0, 1)$ is locally finite. The max–min semiring $\mathbb{R}_{\max,\min} = (\mathbb{R}_+ \cup \{\infty\}, \max, \min, 0, \infty)$ of positive reals, used for maximum capacity problems of networks, is locally finite. More generally, any distributive lattice $(L, \vee, \wedge, 0, 1)$ with smallest element 0 and largest element 1 is locally finite.

The following result slightly generalizes [8, Theorem 9] where wta with final states rather than final weights were considered.

Lemma 6.1. *Let K be a locally finite commutative semiring, $k \in K$, and $S \in K^{\mathrm{rec}}\langle\langle T_\Sigma\rangle\rangle$. Then $S^{-1}(k)$ is a recognizable tree language. In particular, S is a recognizable step function.*

Proof. Let $M = (Q, \Sigma, \mu, \gamma)$ be a wta over K which accepts S. We will construct a deterministic bottom-up finite state tree automaton M_k such that $\mathcal{L}(M_k) = S_M^{-1}(k)$. Let $M_k = (Q', \Sigma, \delta, F)$ such that

- $Q' = (K')^Q$ where $K' = \langle\{\mu_m(\sigma)_{q_1\dots q_m,q} \mid m \geqslant 0, \sigma \in \Sigma^{(m)}, q, q_1, \dots, q_m \in Q\} \cup \{\gamma(q) \mid q \in Q\}\rangle \subseteq K$; i.e., Q' is the set of Q-vectors over K';
- for every $m \geqslant 0$, $\sigma \in \Sigma^{(m)}$, $P_1, \dots, P_m \in Q'$, and $q \in Q$ we define

$$\delta_\sigma(P_1, \dots, P_m)_q = \sum_{q_1,\dots,q_m \in Q} (P_1)_{q_1} \cdot \ldots \cdot (P_m)_{q_m} \cdot \mu_l(\sigma)_{q_1\dots q_m,q},$$

and

- $F = \{P \in Q' \mid k = \sum_{q\in Q} P_q \cdot \gamma(q)\}$.

We recall that, since M_k is deterministic, for every $t \in T_\Sigma$ there is a unique valid run of M_k on t, denoted by r_t, and $t \in \mathcal{L}(M_k)$ iff $r_t(\varepsilon) \in F$. Now we claim for every position $w \in \mathrm{pos}(t)$, that

$$r_t(w)_q = \sum_{\substack{r\in R_M(t|_w)\\ r(\varepsilon)=q}} \prod_{u\in\mathrm{pos}(t|_w)} \mathrm{wt}_M(t|_w, r, u) \quad \text{for every } q \in Q.$$

This claim can be proved straightforwardly by induction on the "height" of w in t; this is defined as the maximal length of a path from w to the leaves in t. Next we compute (S_M, t) where at $(*)$ we will use the above equation for $w = \varepsilon$.

$$\begin{aligned}(S_M, t) &= \sum_{q\in Q} \sum_{\substack{r\in R_M(t)\\ r(\varepsilon)=q}} \left(\prod_{w\in\mathrm{pos}(t)} \mathrm{wt}_M(t, r, w)\right) \cdot \gamma(q)\\ &=^{(*)} \sum_{q\in Q} (r_t(\varepsilon))_q \cdot \gamma(q).\end{aligned}$$

Using this equality we can argue as follows: $t \in S_M^{-1}(k)$ iff $(S_M, t) = k$ iff $r_t(\varepsilon) \in F$ iff $t \in \mathcal{L}(M_k)$. This proves that $\mathcal{L}(M_k) = S_M^{-1}(k)$.

Finally, it is clear that $S = \sum_{k\in K'} k \cdot \mathbb{1}_{S^{-1}(k)}$ and hence S is a recognizable step function. □

We note the following basic result (cf. [3, Lemma III.1.2] for the string case).

Lemma 6.2. *Let Σ be a ranked alphabet, K and L two commutative semirings, $\varphi : K \to L$ a semiring morphism, and $S \in K^{\mathrm{rec}}\langle\langle T_\Sigma\rangle\rangle$. Then the series $\varphi(S) = \varphi \circ S : T_\Sigma \to L$ with $(\varphi(S), t) = \varphi((S, t))$ for every $t \in T_\Sigma$ is recognizable.*

Proof. Let M be a wta over K which recognizes S. Applying φ to all weights occurring in M yields a wta which recognizes $\varphi(S)$ (compare [8, Lemma 3]). □

Although most of the following result was shown in [32], we indicate an alternative proof, since this permits straightforward effective constructions which we will need later on.

Lemma 6.3. *Let Σ be a ranked alphabet.*

(1) (*Cf.* [32]). *Let $S \in \mathbb{Z}^{\mathrm{rec}}\langle\langle T_\Sigma\rangle\rangle$ and $a, b \in \mathbb{Z}$ with $b \neq 0$. Then $S^{-1}(a + b\mathbb{Z})$ is a recognizable tree language.*
(2) *Let $S \in \mathbb{N}^{\mathrm{rec}}\langle\langle T_\Sigma\rangle\rangle$ and $a \in \mathbb{N}$. Then the languages $S^{-1}(\{n \in \mathbb{N} \mid n \geqslant a\})$, $S^{-1}(\{n \in \mathbb{N} \mid n \leqslant a\})$, and $S^{-1}(a)$ are recognizable tree languages.*

Proof. Analogous to the proof of [3, Corollaries III.2.4 and III.2.5], using Lemmas 6.1 and 6.2.[1] □

Proposition 6.4. *Let K be a locally finite commutative semiring, $\tau : \Sigma \to \Delta$ a relabeling and $S : T_\Sigma \to K$ a recognizable series. Then the series $\Pi_\tau(S) : T_\Delta \to K$ defined by $(\Pi_\tau(S), t) := \prod_{s\in\tau^{-1}(t)} (S, s)$ $(t \in T_\Delta)$ is recognizable.*

Proof. By Lemma 6.1, S has the form $S = \sum_{j=1}^n k_j \cdot \mathbb{1}_{L_j}$ with $n \in \mathbb{N}$, $k_j \in K$ and recognizable tree languages $L_j \subseteq T_\Sigma$ $(j = 1, \dots, n)$ which form a partition of T_Σ. For any $t \in T_\Delta$, let $m_j(t) := |\tau^{-1}(t) \cap L_j|$. Then $(\Pi_\tau(S), t) = \prod_{j=1}^n k_j^{m_j(t)}$. For each $j \in \{1, \dots, n\}$, the submonoid of $(K, \cdot, 1)$ generated by $\{k_j\}$ is finite. Choose a minimal $a_j \in \mathbb{N}$ such that $k_j^{a_j} = k_j^{a_j+x}$ for some $x > 0$ and let b_j be the smallest such $x > 0$. Then $\langle k_j\rangle = \{1, k_j, k_j^2, \dots, k_j^{a_j+b_j-1}\}$, and for each $t \in T_\Delta$, we have $k_j^{m_j(t)} = k_j^{d_j(t)}$ for some uniquely determined $d_j(t) \in \mathbb{N}$ with $0 \leqslant d_j(t) \leqslant a_j + b_j - 1$. Note that if $0 \leqslant d < a_j$, then $k_j^{m_j(t)} = k_j^d$ iff $m_j(t) = d$, and if $a_j \leqslant d < a_j + b_j$, then $k_j^{m_j(t)} = k_j^d$ iff $m_j(t) \in d + b_j\mathbb{N}$. For each $0 \leqslant d < a_j + b_j$ and $1 \leqslant j \leqslant n$ let $M_d^j := \{t \in T_\Delta \mid d_j(t) = d\}$. Then $(\Pi_\tau(S), t) = \prod_{j=1}^n k_j^{d_j(t)} = \sum_{d_1,\dots,d_n : 0\leqslant d_j<a_j+b_j\,(j=1,\dots,n)} k_1^{d_1} \cdot \ldots \cdot k_n^{d_n} \cdot (\mathbb{1}_{M_{d_1}^1 \cap \cdots \cap M_{d_n}^n}, t)$.

For every $1 \leqslant j \leqslant n$, let $\mathbb{1}'_{L_j} : T_\Sigma \to \mathbb{N}$ be the characteristic function of L_j with values $0, 1 \in \mathbb{N}$. By Lemma 3.4 the series $S_j := \tau(\mathbb{1}'_{L_j}) : T_\Delta \to \mathbb{N}$ is recognizable, and $(S_j, t) = \sum_{s\in\tau^{-1}(t)} (\mathbb{1}'_{L_j}, s) = m_j(t)$ $(t \in T_\Delta)$. Hence $M_d^j = \{t \in T_\Delta \mid m_j(t) = d\} = S_j^{-1}(d)$ if $0 \leqslant d < a_j$, and $M_d^j = \{t \in T_\Delta \mid m_j(t) \in d + b_j\mathbb{N}\} = S_j^{-1}(d + b_j\mathbb{N})$ if $a_j \leqslant d < a_j + b_j$. In each case, M_d^j is recognizable by Lemma 6.3. Hence $\Pi_\tau(S)$ is recognizable. □

As a consequence, we obtain for locally finite and commutative semirings that the semantics of all sentences of our MSO-logic are recognizable series.

Theorem 6.5. *Let K be a locally finite commutative semiring. Then $K^{\mathrm{rec}}\langle\langle T_\Sigma\rangle\rangle = K^{\mathrm{mso}}\langle\langle T_\Sigma\rangle\rangle$.*

Proof. The inclusion $K^{\mathrm{rec}}\langle\langle T_\Sigma\rangle\rangle \subseteq K^{\mathrm{mso}}\langle\langle T_\Sigma\rangle\rangle$ is immediate by Theorem 5.11. For the converse, we prove by structural induction for any MSO(K, Σ)-formula φ that $[\![\varphi]\!]$ is recognizable. We may apply Lemmas 5.2–5.4, and for universal first order quantification we use Lemmas 6.1 and 5.5. The induction step for universal second-order quantification follows from Proposition 6.4, using a relabeling $\tau : \Sigma_{\mathcal{V}\cup\{X\}} \to \Sigma_{\mathcal{V}}$ which deletes X where $\mathcal{V} = \mathrm{Free}(\varphi)$ similarly as in the proof of Lemma 5.4. □

Finally, we turn to constructibility and decision problems for computable locally finite commutative semirings K. By Lemma 6.1, every MSO(K, Σ)-formula φ not containing universal second order quantifiers is restricted. Hence the set RMSO(K, Σ) is recursive.

Corollary 6.6. *Let K be a computable locally finite commutative semiring. Given an* MSO(K, Σ)*-formula φ, we can effectively compute a wta M such that $[\![\varphi]\!] = S_M$.*

Proof. We follow the proof of Theorem 6.5. The constructions underlying the lemmata used are all effective. We consider the case of universal first order quantification. If M is a wta, we can compute the finite subsemiring K' of the proof of Lemma 6.1 and hence also the automata M_k for each $k \in K'$. Now follow the proof of Lemma 5.5. Finally, we consider universal second order quantification. In the proof of Proposition 6.4, we obtain wta for the series S_j. Following the proof of Lemma 6.3 we obtain bottom-up finite state tree automata for the languages M_d^j and hence a wta for the series $\Pi_\tau(S)$. Thus, the proof of Theorem 6.5 is constructive. □

Corollary 6.7. *Let K be a computable locally finite commutative semiring. It is decidable whether two given* MSO(K, Σ)*-formulas φ and ψ satisfy $[\![\varphi]\!] = [\![\psi]\!]$.*

---

[1] Lemma 6.3, part 2 is also stated in [32]. However, its proof uses the Inverse Image Theorem of [32] which requires the additional hypothesis that the image of S is finite. This assumption is avoided here.

Proof. By Corollary 6.6, construct two wta M and M' such that $[\![\varphi]\!] = S_M$ and $[\![\psi]\!] = S_{M'}$. It remains to decide whether $S_M = S_{M'}$. For this, compute a finite subsemiring K' of K which contains all possible values of S_M and $S_{M'}$. Now, following the proof of Lemma 6.1, check whether the tree languages $S_M^{-1}(k)$ and $S_{M'}^{-1}(k)$ are equal, for each $k \in K'$. □

7. Conclusions and open problems

The proof of Corollary 5.8 shows that if K is a field, the set $\mathrm{RMSO}(K, \Sigma)$ is recursively enumerable. Two open questions arise. First, the proof of Corollary 5.8 used a "brute force" argument to show the effectiveness of Lemma 5.5. For actual constructions, it would be desirable to have a more explicit method. Second, an open question is whether for an arbitrary $\mathrm{MSO}(K, \Sigma)$-formula φ we can decide whether φ is restricted. Proceeding by structural induction, for this one would need a method for deciding whether an effectively given tree series is a recognizable step function. For weighted automata on words, this was achieved in [18] using results from the theory of formal power series, employing, in particular, Burnside-type results on monoids of matrices over the field K. For tree series, such a theory seems to be lacking. Thus, we do not know whether the set $\mathrm{RMSO}(K, \Sigma)$ is recursive.

The present paper shows that the weighted logic introduced in [18] is robust, because it admits an extension to trees. This motivates the question whether there is an equivalent weighted version of the characterization of the large class of context-free graph languages (mentioned in the introduction) by images of MSO-definable functions on sets of trees [22,23,14]. Also one might investigate the weighted extension of the difficult result that particular attributed tree transducers are characterized by MSO-definable graph transduction [4].

Acknowledgments

We are grateful to the referees for suggesting improvements to the first version of the paper. Also we are grateful to Andreas Maletti for pointing out a more succinct wta in Example 3.2.

References

[1] C. Baier, B. Haverkort, H. Herrmanns, J.-P. Katoen, Model-checking algorithms for continuous time markov chains, IEEE Trans. on Software Eng. 29 (7) (2003) 1–18.
[2] J. Berstel, C. Reutenauer, Recognizable formal power series on trees, Theoret. Comput. Sci. 18 (2) (1982) 115–148.
[3] J. Berstel, Ch. Reutenauer, Rational Series and Their Languages, EATCS-Monographs, Vol. 12, Springer, Berlin, 1988.
[4] R. Bloem, J. Engelfriet, A comparison of tree transductions defined by monadic second order logic and by attribute grammars, J. Comput. System Sci. 61 (2000) 1–50.
[5] B. Borchardt, A pumping lemma and decidability problems for recognizable tree series, Acta Cybernet. 16 (4) (2004) 509–544.
[6] B. Borchardt, The theory of recognizable tree series. Ph.D. Thesis, TU Dresden, 2004.
[7] B. Borchardt, Code selection by tree series transducers, in: Ninth Internat. Conf. on Implementation and Application of Automata (CIAA'04), Kingston, Canada, Proceedings, Lecture Notes in Computer Science, Vol. 3317, Springer, Berlin, 2005, pp. 57–67.
[8] B. Borchardt, A. Maletti, B. Šešelja, A. Tepavčevic, H. Vogler, Cut sets as recognizable tree languages, Fuzzy Sets and Systems 157 (2006) 1560–1571.
[9] B. Borchardt, H. Vogler, Determinization of finite state weighted tree automata, J. Automata Languages and Combinatorics 8 (3) (2003) 417–463.
[10] S. Bozapalidis, Effective construction of the syntactic algebra of a recognizable series on trees, Acta Inform. 28 (1991) 351–363.
[11] S. Bozapalidis, Equational elements in additive algebras, Theory Comput. Systems 32 (1) (1999) 1–33.
[12] J.R. Büchi, Weak second–order arithmetic and finite automata, Z. Math. Logik Math. 6 (1960) 66–92.
[13] A.L. Buchsbaum, R. Giancarlo, J.R. Westbrook, On the determinization of weighted finite automata, SIAM J. Comput. 30 (5) (2000) 1502–1531.
[14] B. Courcelle, The monadic second-order logic of graphs V: on closing the gap between definability and recognizability, Theoret. Comput. Sci. 80 (1991) 153–202.
[15] K. Culik, J. Kari, Image compression using weighted finite automata, Comput. Graphics 17 (1993) 305–313.
[16] J. Doner, Tree acceptors and some of their applications, J. Comput. System Sci. 4 (1970) 406–451.
[17] M. Droste, P. Gastin, On aperiodic and star-free formal power series in partially commuting variables, in: Formal Power Series and Algebraic Combinatorics (Moscow 2000), Springer, Berlin, 2000, pp. 158–169, full version in Theory of Computing Systems, to appear.
[18] M. Droste, P. Gastin, Weighted automata and weighted logics, in: Automata, Languages and Programming—32nd International Colloquium, ICALP 2005, Lisbon, Portugal, 2005, Proceedings, Lecture Notes in Computer Science, Vol. 3580, Springer, Berlin, 2005, pp. 513–525, full version in Theoretical Computer Science, to appear.
[19] M. Droste, C. Pech, H. Vogler, A Kleene theorem for weighted tree automata, Theory Comput. Systems 38 (2005) 1–38.

[20] C.C. Elgot, Decision problems of finite automata design and related arithmetics, Trans. Amer. Math. Soc. 98 (1961) 21–52.
[21] J. Engelfriet, Bottom-up and top-down tree transformations—a comparison, Math. Systems Theory 9 (3) (1975) 198–231.
[22] J. Engelfriet, A characterization of context-free NCE graph languages by monadic second-order logic on trees, in: G. Rozenberg, H. Ehrig, H.-J. Kreowski (Eds.), Graph Grammars and their Application to Computer Science Fourth Internat. Workshop, Bremen, Germany, 1990. Proceedings, Lecture Notes in Computer Science, Vol. 532, 1991, pp. 311–327.
[23] J. Engelfriet, V. van Oostrom, Logical description of context-free graph languages, J. Comput. System Sci. 55 (1997) 489–503.
[24] Z. Ésik, W. Kuich, Formal tree series, J. Automata Languages, Combinatorics 8 (2) (2003) 219–285.
[25] C. Ferdinand, H. Seidl, R. Wilhelm, Tree automata for code selection, Acta Inform. 31 (8) (1994) 741–760.
[26] F. Gécseg, M. Steinby, Tree Automata, Akadémiai Kiadó, Budapest, 1984.
[27] F. Gécseg, M. Steinby, Tree languages, in: G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, Vol. 3, Springer, Berlin, 1997, pp. 1–68, Chapter 1.
[28] U. Hafner, Low bit-rate image and video coding with weighted finite automata, Ph.D. Thesis, Universität Würzburg, Germany, 1999.
[29] W. Kuich, Formal power series over trees, in: S. Bozapalidis (Ed.), Third Internat. Conf. on Developments in Language Theory, DLT 1997, Thessaloniki, Greece, Proceedings, Aristotle University of Thessaloniki, 1998, pp. 61–101.
[30] W. Kuich, Tree transducers and formal tree series, Acta Cybernet. 14 (1999) 135–149.
[31] W. Kuich, A. Salomaa, Semirings, Automata, Languages, EATCS Monographs on Theoretical Computer Science, Springer, Berlin, 1986.
[32] O. Louscou-Bozapalidis, Some remarks on recognizable tree series, Internat. J. Computer Math. 70 (1999) 649–655.
[33] Chr. Pech, Kleene-type results for weighted tree automata. Ph.D. Thesis, TU Dresden, 2003.
[34] A. Salomaa, M. Soittola, Automata-Theoretic Aspects of Formal Power Series. Texts and Monographs in Computer Science, Springer, Berlin, 1978.
[35] M.P. Schützenberger, On the definition of a family of automata, Inform. and Control 4 (1961) 245–270.
[36] H. Seidl, Deciding equivalence of finite tree automata, SIAM J. Comput. 19 (3) (1990) 424–437.
[37] H. Seidl, Finite tree automata with cost functions, Theoret. Comput. Sci. 126 (1) (1994) 113–142.
[38] E. Stark, On behaviour equivalence for probabilistic i/o automata and its relationship to probabilistic bisimulation, J. Automata Languages Combinatorics 8 (2003) 361–395.
[39] J.W. Thatcher, J.B. Wright, Generalized finite automata theory with application to a decision problem of second-order logic, Math. Systems Theory 2 (1) (1968) 57–81.