Theory Comput Syst (2011) 48: 23–47
DOI 10.1007/s00224-009-9224-4

Weighted Logics for Unranked Tree Automata

Manfred Droste · Heiko Vogler

Published online: 27 June 2009

Abstract We define a weighted monadic second order logic for unranked trees and the concept of weighted unranked tree automata, and we investigate the expressive power of these two concepts. We show that weighted tree automata and a syntactically restricted weighted MSO-logic have the same expressive power in case the semiring is commutative or in case we deal only with ranked trees, but, surprisingly, not in general. This demonstrates a crucial difference between the theories of ranked trees and unranked trees in the weighted case.

Keywords Weighted logics · Formal power series · Unranked tree automata · Weighted tree automata

1 Introduction

The investigations of formal languages, automata, and logic on ranked and unranked trees started in the 60s of the previous century. For the ranked part, this is already a well-established research area, cf., e.g. [10, 22, 23] for survey books on these topics. To the unranked part, much attention has been payed recently [7, 8, 28] (also cf. Chap. 8 of [10]) which is mainly due to the development of the modern document language XML and the fact that (fully structured) XML-documents can be formalized as unranked trees.

One of the fundamental results in the theory of tree automata is the fact that a tree language is recognizable if and only if it is definable by a sentence of monadic second

M. Droste
Institute of Computer Science, Leipzig University, 04109 Leipzig, Germany
e-mail: droste@informatik.uni-leipzig.de

H. Vogler (✉)
Department of Computer Science, Technische Universität Dresden, 01062 Dresden, Germany
e-mail: vogler@tcs.inf.tu-dresden.de

order (MSO) logic (for the ranked case cf. [11, 42], for the unranked case cf. [28, 34, 35]). This characterization generalizes the corresponding theorem for the string case [9, 19].

In MSO-logic for unranked trees one can pose qualitative questions like whether, in a given bibliography database (formalized as an unranked tree), there is an entry which misses optional information of a certain kind, or whether there is a paper with three authors. As extension of this scenario, it is a natural problem for databases to pose *quantitative* queries to documents. For instance, one might ask *how many* entries miss optional information. Or, as another example: given the different efforts (measured as natural numbers) for completing a `book`-entry and an `article`-entry, respectively; then one might want to know the weighted sum which shows the whole effort to complete all `book`- and all `article`-entries in the database. Such quantities we call *weights*.

In this paper we present a weighted logic which is suitable for the formulation of such quantitative queries for unranked trees (weighted MSO-logic). This logic was heavily inspired by and goes back to the weighted MSO-logic presented in [12–14] for strings; the latter has been extended to infinite strings [15], finite and infinite strings with discounting [15], ranked trees [16], infinite trees [36], trace languages [32], picture languages [20], and texts and nested words [30, 31]. In all these approaches the weights are computed in some semiring which has shown to be the appropriate algebraic structure for coping with different weight scenarios in a uniform way [4, 17, 18, 25, 27, 37, 40].

As an automata-theoretic counterpart of our weighted MSO-logic, we will introduce *weighted tree automata over unranked trees* (for short: wta) where the weights are taken from a semiring S. These generalize bottom-up finite tree automata over unranked trees [7, 8, 28, 29, 34, 41] by adding weights. More precisely, in a wta M, with each pair consisting of a state $q \in Q$ and an input symbol σ, a weighted finite automaton $A_{q,\sigma}$ is associated which recognizes a formal power series over S and Q; then, for every $w \in Q^*$, the value $(\|A_{q,\sigma}\|, w) \in S$ is the weight of the state transition (w, q) at input symbol σ, where $\|A_{q,\sigma}\|$ denotes the behavior of $A_{q,\sigma}$. Clearly, bottom-up finite tree automata over unranked trees can be reobtained from our model by choosing the Boolean semiring for S. We note that also weighted tree automata over ranked trees [2, 3, 5, 21, 26] are special wta: only those unranked trees which obey the given ranks of symbols are considered.

For the comparison of MSO-logic and wta, we describe both, the behavior of wta and the semantics of sentences of our weighted MSO-logic by unranked tree series, i.e., by functions associating to each unranked tree a value in S.

The main results of this paper involve the syntactically restricted MSO-logic as it has been defined in [14] for words. It was shown for any semiring that weighted automata over words have precisely the same expressive power as the syntactically restricted MSO-logic [14]. This lifts up to ranked trees and, in case the semiring is commutative, also to unranked trees. Surprisingly, the equivalence does not extend to the case of unranked trees and non-commutative semirings. This demonstrates a crucial difference between the theories of ranked trees and unranked trees in the weighted case.

More precisely, we show the following results:

(1) Let r be an unranked tree series which is definable in (syntactically) restricted MSO-logic. Then r is recognizable (cf. Theorem 6.5).
(2) Let r be recognizable, then r is MSO-definable. Moreover, if S is commutative, then r is definable in syntactically restricted existential MSO-logic (cf. Theorem 6.9).
(3) There is a recognizable unranked tree series which is not definable in syntactically restricted MSO-logic (cf. Theorem 6.10).
(4) Let r be a ranked tree series. Then r is recognizable if and only if r is definable by ranked syntactically restricted existential MSO-logic (cf. Theorem 7.2).

We prove our results by direct automata-theoretic constructions along the lines of [13, 14] by generalizing from the case of ranked trees [16] to that of unranked trees. Previous alternative arguments using encoding of unranked trees as ranked trees turned out to be more complicated. We note that, in contrast to [13, 16] and in accordance with [14], our results employ a purely syntactically defined subclass of weighted MSO-logic. Moreover, the semirings occurring here need not be commutative. The latter fact implies that we need a linear ordering on the nodes of unranked trees in order to evaluate products which occur in the run semantics of a wta and in the interpretation of universal quantifications in a correct manner. This issue needs some care in handling. The construction of an MSO-sentence for the simulation of a wta is slightly more complicated than in the ranked case, because here a wta employs in its transitions weighted string automata over states, and the latter also have to be modelled.

We also expose an extended example where we define a bibliography database with bibtex entries and show how quantitative queries of the form mentioned above can be formulated in syntactically restricted MSO-logic.

We note that in [38], an MSO-logic for unranked trees with Presburger constraints on the children of nodes was presented; the satisfiability of this logic was shown to be undecidable. Recently, [39] presented a modal fixpoint logic with Presburger constraints which becomes decidable. It would be interesting to compare and possibly combine these approaches with the present one formulated for arbitrary semirings of numerical weights.

In order to avoid repeating over and over again the attribute “unranked”, we make the convention that the unranked case is the standard case for trees, tree automata, and tree series. Whenever we mean the ranked case, we will state this explicitly.

2 Preliminaries

2.1 Basic Notions and Trees

Let $\mathbb{N}$ and $\mathbb{N}_+$ be the sets $\{0, 1, 2, \ldots\}$ and $\{1, 2, \ldots\}$, respectively.

Let Σ be an alphabet, i.e., a finite nonempty set. The set of *Σ-trees*, denoted by U_Σ, is the smallest subset U of $(\Sigma \cup \{(,)\} \cup \{,\})^*$ such that if $\sigma \in \Sigma$ and $\xi_1, \ldots, \xi_k \in U$ with $k \geq 0$, then $\sigma(\xi_1, \ldots, \xi_k) \in U$. In case $k = 0$, we identify $\sigma()$ with σ; thus $\Sigma \subseteq U_\Sigma$. Any subset of U_Σ is called a *(Σ-)tree language*.

We define the set of *positions in a* Σ*-tree* by means of the mapping $\mathrm{pos} : U_\Sigma \to \mathcal{P}(\mathbb{N}_+^*)$ inductively on the argument $\xi \in U_\Sigma$ as follows: if $\xi = \sigma(\xi_1, \ldots, \xi_k)$ where $\sigma \in \Sigma$, $k \geq 0$ and $\xi_1, \ldots, \xi_k \in U_\Sigma$, then $\mathrm{pos}(\xi) = \{\varepsilon\} \cup \{iv \mid 1 \leq i \leq k, v \in \mathrm{pos}(\xi_i)\}$. Sometimes we will also write $i \cdot v$ for iv.

For every $\xi \in U_\Sigma$ and $w \in \mathrm{pos}(\xi)$, the *label of* ξ *at* w, denoted by $\xi(w) \in \Sigma$, and the *rank at* w, denoted by $\mathrm{rk}_\xi(w)$, are defined inductively as follows: if $\xi = \sigma(\xi_1, \ldots, \xi_k)$ for some $\sigma \in \Sigma$ with $k \geq 0$ and $\xi_1, \ldots, \xi_k \in U_\Sigma$, then $\xi(\varepsilon) = \sigma$ and $\mathrm{rk}_\xi(\varepsilon) = k$, and if $1 \leq i \leq k$ and $w = iv$, then $\xi(w) = \xi_i(v)$ and $\mathrm{rk}_\xi(w) = \mathrm{rk}_{\xi_i}(v)$.

Later on, when we define the behavior of a weighted tree automaton on an input tree ξ, we will need a linear ordering on the set $\mathrm{pos}(\xi)$. For this we choose here the depth-first left-to-right traversal over ξ which, at a position $w \in \mathrm{pos}(\xi)$, visits the subtrees one by one from left to right, and then it deals with w itself. We denote this linear ordering by $\sqsubseteq_\xi$.

In this paper, Σ *will always denote an arbitrary alphabet unless specified otherwise.*

2.2 Semirings, Formal Power Series, and Weighted Finite Automata

Here we recall the basic notions on semirings, formal power series, and weighted finite automata on strings. We refer the reader for more information to [18, 27, 37].

A semiring is a structure $(S, +, \cdot, 0, 1)$ (often abbreviated by S) where $(S, +, 0)$ is a commutative monoid, $(S, \cdot, 1)$ is a monoid, multiplication distributes over addition from both sides, and $0 \cdot s = s \cdot 0 = 0$ for every $s \in S$. For $A, B \subseteq S$, we say that A and B *commute elementwise*, if $a \cdot b = b \cdot a$ for all $a \in A$ and $b \in B$. A semiring is *commutative* if $\cdot$ is commutative.

In this paper, S *will always denote an arbitrary semiring unless specified otherwise.*

Let Z be an arbitrary set. A *(formal) power series* over S and Z is a mapping $r : Z \to S$. For $z \in Z$, the value $r(z)$ is here, as usual, denoted as (r, z). The set $Z \setminus r^{-1}(0)$ is called the *support of* r and denoted by $\mathrm{supp}(r)$. The set of all power series over S and Z is denoted by $S\langle\langle Z\rangle\rangle$. Let $L \subseteq Z$. Then the *characteristic series* $\mathbb{1}_L \in S\langle\langle Z\rangle\rangle$ *of* L is defined for every $z \in Z$ by $(\mathbb{1}_L, z) = 1$ if $z \in L$, and $(\mathbb{1}_L, z) = 0$ otherwise. Observe that if $S = \mathbb{B}$, the mapping $L \mapsto \mathbb{1}_L$ provides a bijection between languages and characteristic series.

We define the operations *sum* and *Hadamard-product* on $S\langle\langle Z\rangle\rangle$ as follows: for $r_1, r_2 \in S\langle\langle Z\rangle\rangle$ and $z \in Z$ we let $(r_1 + r_2, z) = (r_1, z) + (r_2, z)$ and $(r_1 \odot r_2, z) = (r_1, z) \cdot (r_2, z)$. Let $s \in S$ and $r \in S\langle\langle Z\rangle\rangle$. Then $r \cdot s \in S\langle\langle Z\rangle\rangle$ is defined by $(r \cdot s, z) = (r, z) \cdot s$ for every $z \in Z$.

Let Σ be an alphabet. Then we call an element $r \in S\langle\langle U_\Sigma\rangle\rangle$ a *tree series*.

Let Δ be an alphabet. As usual, a *weighted finite string automaton (for short: wsa) over* S *and* Δ is a quadruple $A = (P, \lambda, \mu, \nu)$ such that P is a finite set (of *states*), $\mu : P \times \Delta \times P \to S$ is a mapping (called *transition weight function*), and $\lambda, \nu : P \to S$ are functions (called *initial weight function* and *final weight function*, respectively). A *run (through A)* is a sequence $r = (p_0, a_1, p_1)(p_1, a_2, p_2) \ldots (p_{n-1}, a_n, p_n)$ where $p_i \in P$ and $a_i \in \Delta$ with $0 \leq i \leq n$, and we say that r is a *run from* p_0 *to* p_n *with label*

$w = a_1 \ldots a_n$; the set of all such runs is denoted by $P_{p_0,p_n}(w)$. The *weight of r* is the product $\mathrm{wt}(r) = \prod_{i=1}^{n} \mu(p_{i-1}, a_i, p_i)$. Note that $r = \varepsilon$ if $n = 0$, and $\mathrm{wt}(\varepsilon) = 1$ because, as usual, in S products over empty index sets are defined to be 1. The *behavior of A* (or: *power series recognized by A*) is the power series $\|A\| \in S\langle\langle \Delta^* \rangle\rangle$ such that for every $w \in \Delta^*$ we have

$$(\|A\|, w) = \sum_{p,p' \in P} \lambda(p) \cdot \mu(p, w, p') \cdot \nu(p')$$

where $\mu(p, w, p') = \sum_{r \in P_{p,p'}(w)} \mathrm{wt}(r)$. A power series $r \in S\langle\langle \Delta^* \rangle\rangle$ is *recognizable over S and* Δ if there is a wsa A over S and Δ such that $r = \|A\|$.

3 Weighted Tree Automata

In this section we extend the concept of (nondeterministic) bottom-up finite tree automata on trees [7] (also cf. [28, 29, 34]) by weights taken from some semiring $(S, +, \cdot, 0, 1)$. The classical concept of tree automata is obtained by letting $S = \mathbb{B}$.

A *weighted tree automaton (for short: wta) over S* is a quadruple $M = (Q, \Sigma, A, \gamma)$ where Q is a finite set (of *states*), Σ is an alphabet (of *input symbols*), $A = (A_{q,\sigma} \mid q \in Q, \sigma \in \Sigma)$ is a family of wsa over S and Q, and $\gamma : Q \to S$ is a mapping (*root weight function*). A wta M is *deterministic* if for every $\sigma \in \Sigma$ and $q_1 \ldots q_k \in Q^*$ there is at most one $q \in Q$ such that $(\|A_{q,\sigma}\|, q_1 \ldots q_k) \neq 0$.

Now we define the run semantics of a wta M. Given a tree $\xi \in U_\Sigma$, any function $\kappa : \mathrm{pos}(\xi) \to Q$ is called a *run of M on* ξ, and we define the *weight of* κ by

$$\mathrm{wt}_M(\kappa) = \prod_{w \in \mathrm{pos}(\xi)} \Big(\|A_{\kappa(w),\xi(w)}\|,\ \kappa(w1) \ldots \kappa(w\ \mathrm{rk}_\xi(w))\Big);$$

in the product we follow the linear ordering $\sqsubseteq_\xi$. Note that, if S is commutative, then one can choose any linear ordering and obtain the same result for $\mathrm{wt}_M(\kappa)$. Clearly, if M is deterministic, then for every $\xi \in U_\Sigma$ there is at most one run κ on ξ such that $\mathrm{wt}_M(\kappa) \neq 0$. We let $R_M(\xi)$ be the set of all runs of M on ξ.

The *tree series accepted by M over S* is the tree series $r_M \in S\langle\langle U_\Sigma \rangle\rangle$ defined by

$$(r_M, \xi) = \sum_{\kappa \in R_M(\xi)} \mathrm{wt}_M(\kappa) \cdot \gamma(\kappa(\varepsilon))$$

for every $\xi \in U_\Sigma$. A tree series $r \in S\langle\langle U_\Sigma \rangle\rangle$ is *recognizable over S* if there is a wta M over S such that $r = r_M$. We will denote the class of all recognizable tree series over S and Σ by $\mathrm{Rec}(S, \Sigma)$.

Example 3.1 Let $\Sigma = \{\alpha, \beta\}$ and *Trop* be the ("tropical") semiring $(\mathbb{N} \cup \{\infty\}, \min, +, \infty, 0)$ where the sum and the product operations are min and +, resp., extended to $\mathbb{N} \cup \{\infty\}$ in the obvious way. Now we consider the tree series $\#_{\alpha\alpha} \in \mathit{Trop}\langle\langle U_\Sigma \rangle\rangle$ such that $(\#_{\alpha\alpha}, \xi)$ is the number of positions $w \in \mathrm{pos}(\xi)$ for which the first and second descendant of w are labeled by α, i.e., $\xi(w \cdot 1) = \xi(w \cdot 2) = \alpha$.

As preparation we define $Q=\{q_\alpha, q_\beta\}$ and the two power series $\widetilde{\infty}$ and r over *Trop* and Q by letting $(\widetilde{\infty}, w)=\infty$ and

$$(r,w)=\begin{cases}1 & \text{if } w\in\{q_\alpha\}^2 Q^*\\ 0 & \text{otherwise}\end{cases}$$

for every $w\in Q^*$. Clearly, r and $\widetilde{\infty}$ are recognizable over *Trop* and Q.

Now we construct the wta $M=(Q,\Sigma,A,\gamma)$ over *Trop* with $\gamma(q_\alpha)=\gamma(q_\beta)=0$. Moreover, for every $q_a\in Q$ and $b\in\Sigma$ we choose a wsa $A_{q_a,b}$ over *Trop* and Q such that $\|A_{q_a,b}\|=r$ if $a=b$, and $\|A_{q_a,b}\|=\widetilde{\infty}$ otherwise. It is clear that for every $\xi\in U_\Sigma$ there is exactly one run $\kappa_\xi:\mathrm{pos}(\xi)\to Q$ such that $\mathrm{wt}_M(\kappa_\xi)\neq\infty$: for every position w of ξ we have that $\kappa_\xi(w)=q_{\xi(w)}$. Then $\mathrm{wt}_M(\kappa_\xi)=\#_{\alpha\alpha}(\xi)$, showing that $r_M=\#_{\alpha\alpha}$. E.g., for the tree $\xi=\alpha(\alpha(\alpha,\beta),\alpha(\alpha,\alpha,\beta),\beta)$ we have

position w in ξ	ε	1	11	12	2	21	22	23	3
$(\|A_{q_{\xi(w)},\xi(w)}\|,\ q_{\xi(w1)},\dots,q_{\xi(w\,\mathrm{rk}_\xi(w))})$	1	0	0	0	1	0	0	0	0

Hence $(r_M,\xi)=\min_{\kappa\in R_M(\xi)}\{\mathrm{wt}_M(\kappa)+\gamma(\kappa(\varepsilon))\}=\mathrm{wt}_M(\kappa_\xi)=2$.

The unweighted case can be obtained as follows. A tree language $L\subseteq U_\Sigma$ is *recognizable* if there is a wta M over the Boolean semiring $\mathbb{B}=(\{0,1\},\vee,\wedge,0,1)$ with disjunction $\vee$ and conjunction $\wedge$, such that $L=\mathrm{supp}(r_M)$. Using the bijection between tree languages and characteristic series, it is easy to see that this notion coincides with the usual definition of recognizability for tree languages. In fact, nondeterministic (and deterministic) bottom-up tree automata on trees in the sense of [7, 34] are precisely the wta (and deterministic wta, respectively) over $\mathbb{B}$. Subsequently, we will often use the fact that the class of recognizable tree languages is closed under intersection and complement, cf. Theorem H of [7] (also cf. Theorem 8.3.8 of [10]). In Theorem B of [7] (also cf. Theorem 8.2.8 of [10]) it is proved by using the well-known power set construction that every recognizable tree language is recognizable by a deterministic wta over $\mathbb{B}$. Using this property, it is easy to see that for every recognizable tree language L, the series $\mathbb{1}_L\in S\langle\langle U_\Sigma\rangle\rangle$ is recognizable.

Next we note basic properties of the classes of recognizable step functions and recognizable tree series. A tree series $r\in S\langle\langle U_\Sigma\rangle\rangle$ is a *recognizable step function* if there are an $n\geq 1$, coefficients $s_1,\dots,s_n\in S$, and recognizable tree languages $L_1,\dots,L_n\subseteq U_\Sigma$ such that $r=\sum_{i=1}^n \mathbb{1}_{L_i}\cdot s_i$. Equivalently, r assumes only finitely many values, and for each $s\in S$ the language $r^{-1}(s)$ is recognizable. In fact, we can choose $L_1,\dots,L_n$ as above such that they form a partition of U_Σ. The proofs of the following properties are completely analogous to the ranked case [16].

Proposition 3.2 *Let $r_1,r_2\in S\langle\langle U_\Sigma\rangle\rangle$ and $s\in S$.*

1. *The constant tree series $\widetilde{s}\in S\langle\langle U_\Sigma\rangle\rangle$ which maps every tree to s, is recognizable.*
2. *If r_1 and r_2 are recognizable, then r_1+r_2 and $r_1\cdot s$ are recognizable.*
3. *Let $S_1,S_2\subseteq S$ be two subsemirings such that S_1 and S_2 commute elementwise. Let r_1 and r_2 be recognizable over S_1 and S_2, respectively. Then $r_1\odot r_2$ is recognizable over S.*

4. *If r_1 and r_2 are recognizable step functions, then $r_1 + r_2$, $r_1 \odot r_2$, and $r_1 \cdot s$ are recognizable step functions.*
5. *Every recognizable step function is a recognizable tree series.*
6. *Let S' be another semiring, $\varphi : S \to S'$ a semiring morphism, and $r \in \mathrm{Rec}(S, \Sigma)$. Then the tree series $\varphi(S) = \varphi \circ r : U_\Sigma \to S'$ with $(\varphi(r), \xi) = \varphi((r, \xi))$ for every $\xi \in U_\Sigma$ is recognizable.*

We note that, as in the ranked case, the class of recognizable tree series is closed under relabelings. For this, let Σ and Δ be two alphabets and $\tau : \Sigma \to \mathcal{P}(\Delta)$ be a mapping. This mapping is extended to a mapping $\tau' : U_\Sigma \to \mathcal{P}(U_\Delta)$ by defining inductively $\tau'(\sigma(\xi_1, \ldots, \xi_k)) = \{\beta(\zeta_1, \ldots, \zeta_k) \mid \beta \in \tau(\sigma), \zeta_1 \in \tau'(\xi_1), \ldots, \zeta_k \in \tau'(\xi_k)\}$ for every $\sigma \in \Sigma$, $k \geq 0$, and $\xi_1, \ldots, \xi_k \in U_\Sigma$. Note that the set $(\tau')^{-1}(\zeta)$ is finite for every $\zeta \in U_\Delta$. Next we extend τ' to a mapping $\tau'' : S\langle\langle U_\Sigma\rangle\rangle \to S\langle\langle U_\Delta\rangle\rangle$, called *relabeling*, by defining $(\tau''(r), \zeta) = \sum_{\xi \in U_\Sigma, \zeta \in \tau'(\xi)} (r, \xi)$ for every $r \in S\langle\langle U_\Sigma\rangle\rangle$ and $\zeta \in U_\Delta$. In the sequel we will drop the primes from τ' and τ''. The proof of the next lemma is again completely analogous to the ranked case (cf. Lemma 3.4 of [16]).

Lemma 3.3 *Let $r \in S\langle\langle U_\Sigma\rangle\rangle$ and $\tau : \Sigma \to \mathcal{P}(\Delta)$ be a relabeling. If r is recognizable, then $\tau(r)$ is recognizable.*

4 Weighted MSO-Logic

In this section, we will introduce our weighted MSO-logic for trees. The set $\mathrm{MSO}(S, \Sigma)$ of all formulas of *weighted MSO-logic over S and Σ on trees* is defined to be the smallest set F such that:

1. F contains all the *atomic formulas* s, $\mathrm{label}_\sigma(x)$, $\mathrm{desc}(x, y)$, $(x \leq y)$, $(x \sqsubseteq y)$, and $(x \in X)$ and the negations $\neg\mathrm{label}_\sigma(x)$, $\neg\mathrm{desc}(x, y)$, $\neg(x \leq y)$, $\neg(x \sqsubseteq y)$, and $\neg(x \in X)$, and
2. if φ and ψ are in F, then also $\varphi \vee \psi$, $\varphi \wedge \psi$, $\exists x.\varphi$, $\forall x.\varphi$, $\exists X.\varphi$, $\forall X.\varphi$ are in F,

where $s \in S$, $\sigma \in \Sigma$, x, y are first order variables, and X is a second order variable. We denote by $\mathrm{MSO}^-(S, \Sigma)$ the fragment of all $\mathrm{MSO}(S, \Sigma)$-formulas not containing formulas of the form s with $s \in S$ as subformulas.

In order to define the semantics of formulas with free variables, we extend the alphabet Σ in the usual way. If $\mathcal{V}$ is a finite set of first and second order variables, we denote the alphabet $\Sigma \times \{0, 1\}^\mathcal{V}$ by $\Sigma_\mathcal{V}$. A $\Sigma_\mathcal{V}$-tree ξ is *valid* if for every first order variable $x \in \mathcal{V}$, ξ contains precisely one position assigning 1 to x. The subset of $U_{\Sigma_\mathcal{V}}$ containing all valid trees is denoted by $U^v_{\Sigma_\mathcal{V}}$; clearly, the tree language $U^v_{\Sigma_\mathcal{V}}$ is recognizable. We put $\Sigma_\varphi = \Sigma_{\mathrm{Free}(\varphi)}$.

In the sequel we will identify a valid $\Sigma_\mathcal{V}$-tree ξ with the corresponding pair (ζ, ρ) where $\zeta \in U_\Sigma$ and ρ is a $(\mathcal{V}, \zeta)$-*assignment*; such an assignment maps first order variables in $\mathcal{V}$ to elements of $\mathrm{pos}(\zeta)$ and second order variables in $\mathcal{V}$ to subsets of $\mathrm{pos}(\zeta)$.

Let ξ be an arbitrary $\Sigma_\mathcal{V}$-tree, x be a first order variable, and $w \in \mathrm{pos}(\xi)$. Then $\xi[x \to w]$ is the $\Sigma_{\mathcal{V} \cup \{x\}}$-labeled tree obtained from ξ which assigns 1 to x at position w, and 0 elsewhere. Similarly, if X is a second order variable and $I \subseteq \mathrm{pos}(\xi)$, then $\xi[X \to I]$ is the $\Sigma_{\mathcal{V}\cup\{X\}}$-tree obtained from ξ which assigns 1 to X precisely at the positions in I. If here $\xi = (\zeta, \rho)$, we also write $\xi[x \to w] = (\zeta, \rho[x \to w])$ and $\xi[X \to I] = (\zeta, \rho[X \to I])$. The following is analogous to the corresponding definition for the case of strings in [12, 13].

Definition 4.1 Let $\varphi \in \mathrm{MSO}(S, \Sigma)$ and $\mathcal{V}$ be a finite set of variables containing $\mathrm{Free}(\varphi)$. The *semantics* of φ is the formal tree series $[\![\varphi]\!]_{\mathcal{V}} \in S\langle\langle U_{\Sigma_{\mathcal{V}}}\rangle\rangle$ defined as follows. If $\xi \in U_{\Sigma_{\mathcal{V}}}$ is not valid, then we put $([\![\varphi]\!]_{\mathcal{V}}, \xi) = 0$. Otherwise, we define $([\![\varphi]\!]_{\mathcal{V}}, \xi) \in S$ inductively as follows where (ζ, ρ) corresponds to ξ.

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

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

$$([\![\mathrm{desc}(x, y)]\!]_{\mathcal{V}}, \xi) = \begin{cases} 1 & \text{if there is an } i \text{ such that } \rho(y) = \rho(x) \cdot i \\ 0 & \text{otherwise} \end{cases}$$

$$([\![x \le y]\!]_{\mathcal{V}}, \xi) = \begin{cases} 1 & \text{if } \rho(x) = \rho(y) = \varepsilon \text{ or if there are } w \in \mathrm{pos}(\xi) \text{ and } i, j \ge 1 \\ & \text{such that } \rho(x) = w \cdot i, \rho(y) = w \cdot j, \text{ and } i \le j \\ 0 & \text{otherwise} \end{cases}$$

$$([\![x \sqsubseteq y]\!]_{\mathcal{V}}, \xi) = \begin{cases} 1 & \text{if } \rho(x) \sqsubseteq_\xi \rho(y) \\ 0 & \text{otherwise} \end{cases}$$

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

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

if φ is of the form $\mathrm{label}_\sigma(x)$, $\mathrm{desc}(x, y)$, $(x \le y)$, $(x \sqsubseteq y)$, or $(x \in X)$

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

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

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

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

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

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

where in the product over $\mathrm{pos}(\xi)$ we follow the depth-first left-to-right traversal $\sqsubseteq_\xi$ over $\mathrm{pos}(\xi)$; moreover, for the product over subsets I of $\mathrm{pos}(\xi)$, we employ the lexicographic linear order on the set $\{0,1\}^{\mathrm{pos}(\xi)}$ of pos(ξ)-sequences of 0's and 1's where the sequences are ordered according to $\sqsubseteq_\xi$.

We write $[\![\varphi]\!]$ rather than $[\![\varphi]\!]_{\mathrm{Free}(\varphi)}$. Let $Z \subseteq \mathrm{MSO}(S,\Sigma)$. A tree series $r \in S\langle\langle U_\Sigma\rangle\rangle$ is called *Z-definable* if there is a sentence $\varphi \in Z$ such that $r = [\![\varphi]\!]$.

We note that, whereas in classical logic both disjunction and conjunction distribute over each other, in general semirings only multiplication distributes over addition, and hence in our weighted logic only conjunction distributes over disjunction. This phenomenon is well known already in many-valued logics, e.g., restrictions of the Łukasiewicz logic.

As in the string case (Proposition 3.3 of [13]) and the ranked tree case (Lemma 4.7 of [16]) we note that the semantics $[\![\varphi]\!]_{\mathcal{V}}$ for every $\mathcal{V}$ containing Free(φ) are consistent, which follows by a standard induction on the structure of φ.

Lemma 4.2 *Let* $\varphi \in \mathrm{MSO}(S,\Sigma)$ *and* $\mathcal{V}$ *a finite set of variables containing* Free(φ). *Then, for every* $(\zeta,\rho) \in U^v_{\Sigma_{\mathcal{V}}}$, *we have that* $([\![\varphi]\!]_{\mathcal{V}}, (\zeta,\rho)) = ([\![\varphi]\!], (\zeta, \rho|_{\mathrm{Free}(\varphi)}))$. *In particular,* $[\![\varphi]\!]$ *is recognizable iff* $[\![\varphi]\!]_{\mathcal{V}}$ *is recognizable, and* $[\![\varphi]\!]$ *is a recognizable step function iff* $[\![\varphi]\!]_{\mathcal{V}}$ *is a recognizable step function.*

Now we recall how the classical equivalence result between MSO-definable and recognizable tree languages [28, 34, 35] can be formulated in the present context. Let $\varphi \in \mathrm{MSO}^-(\mathbb{B},\Sigma)$. The *tree language defined by* φ, denoted by $L(\varphi)$, is the set supp($[\![\varphi]\!]$). We call a tree language $L \subseteq U^v_{\Sigma_{\mathcal{V}}}$ *definable* if there is an $\mathrm{MSO}^-(\mathbb{B},\Sigma)$-formula φ with $\mathrm{Free}(\varphi) \subseteq \mathcal{V}$ such that $L = L(\varphi)$. Using the bijection between tree languages and characteristic series, it is easy to see that this notion coincides with the usual definition of definability for tree languages. Then a tree language $L \subseteq U_\Sigma$ is recognizable iff L is definable by a sentence over Σ.

5 Unambiguous Formulas

Here we introduce unambiguous formulas for trees; for this we follow the lines of [14]. As motivation, consider $\varphi, \psi \in \mathrm{MSO}(\mathbb{N},\Sigma)$ over the semiring $\mathbb{N}$ of natural numbers and suppose that $[\![\varphi]\!]$ and $[\![\psi]\!]$ both assume only values 0 or 1. Then $[\![\varphi \vee \psi]\!]$ may assume value 2, and $[\![\exists x.\varphi]\!]$ may assume arbitrarily high numbers as value. Next we introduce a subclass of formulas for which this phenomenon cannot occur.

Definition 5.1 ([13], Definition 5.1) The class of *unambiguous* formulas in $\mathrm{MSO}(S,\Sigma)$ is defined inductively as follows:

1. Every atomic formula of the form $\mathrm{label}_\sigma(x)$, $\mathrm{desc}(x,y)$, $(x \le y)$, $(x \sqsubseteq y)$, or $(x \in X)$, and their negations are unambiguous.
2. If φ, ψ are unambiguous, then $\varphi \wedge \psi$, $\forall x.\varphi$, and $\forall X.\varphi$ are unambiguous.
3. If φ, ψ are unambiguous and $\mathrm{supp}([\![\varphi]\!]) \cap \mathrm{supp}([\![\psi]\!]) = \emptyset$, then $\varphi \vee \psi$ is unambiguous.

4. Let φ be unambiguous and $\mathcal{V} = \text{Free}(\varphi)$. If for every $\xi \in U_{\Sigma_\mathcal{V}}$ there is at most one position $w \in \text{pos}(\xi)$ such that $(\llbracket\varphi\rrbracket_{\mathcal{V}\cup\{x\}}, \xi[x \to w]) \neq 0$, then $\exists x.\varphi$ is unambiguous.
5. Let φ be unambiguous and $\mathcal{V} = \text{Free}(\varphi)$. If for every $\xi \in U_{\Sigma_\mathcal{V}}$ there is at most one set $I \subseteq \text{pos}(\xi)$ such that $(\llbracket\varphi\rrbracket_{\mathcal{V}\cup\{X\}}, \xi[X \to I]) \neq 0$, then $\exists X.\varphi$ is unambiguous.

Proposition 5.2 ([13], Proposition 5.2) *Let $\varphi \in \text{MSO}(S, \Sigma)$ be unambiguous. Then, viewing φ as an $\text{MSO}^-(\mathbb{B}, \Sigma)$-formula defining the tree language $L(\varphi)$, it holds that $\llbracket\varphi\rrbracket = \mathbb{1}_{L(\varphi)}$. In particular, $\llbracket\varphi\rrbracket$ is a recognizable step function.*

Proof We proceed by structural induction similar to [13]. We note that, if $\varphi = (x \sqsubseteq y)$, then $L(\varphi)$ is a recognizable tree language by our choice of the linear order $\sqsubseteq_\xi$. □

Next we note that there is a purely syntactic definition of formulas φ^+ and φ^- in $\text{MSO}^-(S, \Sigma)$ for any $\varphi \in \text{MSO}^-(S, \Sigma)$ with the following properties:

- the formulas φ^+ and φ^- are unambiguous,
- $L(\varphi^+) = L(\varphi)$ and $L(\varphi^-) = L(\neg\varphi)$, and
- $\llbracket\varphi^+\rrbracket = \mathbb{1}_{L(\varphi)}$ and $\llbracket\varphi^-\rrbracket = \mathbb{1}_{L(\neg\varphi)}$.

For this we can proceed completely analogously to Definition 4.3 of [14] using the atomic formula $x \sqsubseteq y$ for first order quantifications. Extending [16], this includes the case of formulas containing set quantifiers for which we extend the depth-first left-to-right ordering on $\text{pos}(\xi)$ to the lexicographic linear order on the set $\{0, 1\}^{\text{pos}(\xi)}$ of subsets of $\text{pos}(\xi)$.

Moreover, for any $\varphi, \psi \in \text{MSO}^-(S, \Sigma)$, we define the formulas $\varphi \overset{+}{\to} \psi$ and $\varphi \overset{+}{\leftrightarrow} \psi$ in $\text{MSO}^-(S, \Sigma)$ as follows: $\varphi \overset{+}{\to} \psi = \varphi^- \vee (\varphi^+ \wedge \psi^+)$ and $\varphi \overset{+}{\leftrightarrow} \psi = (\varphi^+ \wedge \psi^+) \vee (\varphi^- \wedge \psi^-)$. Using this, we define a formula to be *syntactically unambiguous* if it is of the form $\varphi^+, \varphi^-, \varphi \overset{+}{\to} \psi$ or $\varphi \overset{+}{\leftrightarrow} \psi$ for $\varphi, \psi \in \text{MSO}^-(S, \Sigma)$. Clearly, each syntactically unambiguous formula is unambiguous.

Proposition 5.3 *For each classical* MSO-*sentence φ, we can effectively construct a syntactically unambiguous* $\text{MSO}(S, \Sigma)$-*sentence φ' defining the same language, i.e., $\llbracket\varphi'\rrbracket = \mathbb{1}_{L(\varphi)}$.*

Proof Using also conjunctions and universal quantifications, transform φ into an equivalent MSO-sentence ψ in which negation is only applied to atomic formulas. Then put $\varphi' = \psi^+$. □

In the next example and also later (in the proof of Theorem 6.9) we will use the following macro. For $\varphi \in \text{MSO}^-(S, \Sigma)$ and $\psi \in \text{MSO}(S, \Sigma)$ let $\varphi \to \psi = \varphi^- \vee (\varphi^+ \wedge \psi)$. Then for each $\xi \in U_\Sigma$ we have that

$$(\llbracket\varphi \to \psi\rrbracket, \xi) = \begin{cases} (\llbracket\psi\rrbracket, \xi) & \text{if } \xi \in L(\varphi) \\ 1 & \text{otherwise.} \end{cases}$$

Example 5.4 We will show that the series $\#_{\alpha\alpha}$ of Example 3.1 is MSO(*Trop*, Σ)-definable. For this we construct the MSO(*Trop*, Σ)-formula

$$\text{count} = \forall x.\forall y_1.\forall y_2.\big(\big(\text{firstChild}(x, y_1) \wedge \text{next}(y_1, y_2) \wedge \text{label}_\alpha(y_1) \wedge \text{label}_\alpha(y_2)\big) \to 1\big)$$

where we use the macros:

$$\text{firstChild}(x, y) = \text{desc}(x, y) \wedge \forall z.(\text{desc}(x, z) \xrightarrow{+} y \le z)$$

$$\text{next}(y_1, y_2) = (y_1 \le y_2) \wedge \neg(y_2 \le y_1) \wedge \forall z.\big(y_1 \le z \xrightarrow{+} (z \le y_1 \vee y_2 \le z)\big).$$

Clearly, the implication in *count* can only yield the semiring-1, i.e., the natural number 0, or the semiring-constant 1, which is the natural number 1. Since universal quantification is interpreted in *Trop* as the summation of natural numbers, it is easy to see that $[\![\text{count}]\!] = \#_{\alpha\alpha}$.

Finally we give a characterization of recognizable step functions. For this we define the collection of *almost unambiguous* formulas in MSO(*S*, Σ), denoted by auMSO(*S*, Σ), to be the smallest subset of MSO(*S*, Σ) containing all constants s ($s \in S$) and all syntactically unambiguous formulas and which is closed under disjunction and conjunction. We call two formulas $\varphi, \psi \in$ MSO(*S*, Σ) *equivalent* if $[\![\varphi]\!] = [\![\psi]\!]$. Since in S multiplication distributes over addition, one can check that each almost unambiguous formula ψ is equivalent to a formula ψ' of the form $\psi' = \bigvee_{j=1}^{n}(\psi_j^{+} \wedge s_j)$ for some $n \in \mathbb{N}$, $s_j \in S$, and $\psi_j \in \text{MSO}^-(S, \Sigma)$ ($j = 1, \dots, n$).

Proposition 5.5 *For each* $\psi \in$ auMSO(*S*, Σ), *the series* $[\![\psi]\!]$ *is a recognizable step function. Conversely, each recognizable step function* $r \in S\langle\langle U_\Sigma\rangle\rangle$ *is* auMSO(*S*, Σ)-*definable.*

Proof The first part is a consequence of the description of ψ noted before, Lemma 4.2 and Proposition 3.2. For the converse, let $r = \sum_{i=1}^{n} \mathbb{1}_{L_i} \cdot s_i$. Since each language L_i ($i = 1, \dots, n$) is recognizable, it is definable by an $\text{MSO}^-(\mathbb{B}, \Sigma)$-sentence φ_i. Now we consider φ_i as an $\text{MSO}^-(S, \Sigma)$-sentence. Then $\psi = \bigvee_{i=1}^{n} \varphi_i^{+} \wedge s_i$ is almost unambiguous and defines r. □

6 Syntactically Restricted Weighted MSO-Logic

In this section we present our syntactically defined weighted MSO-logic and show our first main result.

For an arbitrary formula $\varphi \in$ MSO(*S*, Σ), let val(φ) denote the set containing all values of S occurring in φ. Now we define the syntactically restricted MSO(*S*, Σ)-formulas as in [14].

Definition 6.1 A formula $\varphi \in$ MSO(*S*, Σ) is called *syntactically restricted*, 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 $\mathrm{val}(\psi)$ and $\mathrm{val}(\psi')$ commute elementwise.
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 $\mathrm{srMSO}(S, \Sigma)$ denote the set of all syntactically restricted formulas of $\mathrm{MSO}(S, \Sigma)$.

Here condition (1) requires us to be able to check for $s, s' \in S$ whether $s \cdot s' = s' \cdot s$. We assume this basic ability to be given in syntax checks of formulas from $\mathrm{MSO}(S, \Sigma)$. Note that for $\psi, \psi' \in \mathrm{MSO}(S, \Sigma)$, $\mathrm{val}(\psi)$ and $\mathrm{val}(\psi')$ trivially commute elementwise, if S is commutative (which was the general assumption of [16]) or if ψ or ψ' is in $\mathrm{MSO}^-(S, \Sigma)$, thus in particular, if ψ or ψ' is unambiguous. Hence for each $\mathrm{MSO}(S, \Sigma)$-formula φ it can be easily checked effectively whether φ is syntactically restricted or not.

A formula $\varphi \in \mathrm{MSO}(S, \Sigma)$ is *existential*, if it is of the form $\varphi = \exists X_1. \ldots \exists X_n.\psi$ where ψ does not contain any set quantifier. The set of all syntactically restricted and existential formulas of $\mathrm{MSO}(S, \Sigma)$ is denoted $\mathrm{srEMSO}(S, \Sigma)$.

The first three main results of our paper are summarized in the following theorem; it will be proved in Sects. 6.1 and 6.2.

Theorem 6.2 *Let S be any semiring and Σ an alphabet. Let $r \in S\langle\langle U_\Sigma\rangle\rangle$ be a tree series. The following implications hold.*

1. *If r is* $\mathrm{srMSO}(S, \Sigma)$*-definable, then r is recognizable.*
2. *If r is recognizable, then r is* MSO*-definable.*
3. *If S is commutative and r is recognizable, then r is* $\mathrm{srEMSO}(S, \Sigma)$*-definable.*

We note that our proofs will be effective. That is, given a syntactically restricted sentence φ of $\mathrm{MSO}(S, \Sigma)$, we can construct a wta M with $r_M = [\![\varphi]\!]$ (provided the operations of S are given effectively). For the converse, given M, we will explicitly describe a sentence $\varphi \in \mathrm{srEMSO}(S, \Sigma)$ with $[\![\varphi]\!] = r_M$.

Slightly extending [16], we call an $\mathrm{MSO}(S, \Sigma)$-formula φ *restricted*, if

1. Whenever φ contains a conjunction $\psi \wedge \psi'$ as subformula but not in the scope of a universal first order quantifier, then $\mathrm{val}(\psi)$ and $\mathrm{val}(\psi')$ commute elementwise.
2. Whenever φ contains $\forall X.\psi$ as a subformula, then ψ is an unambiguous formula.
3. Whenever φ contains $\forall x.\psi$ as a subformula, then $[\![\psi]\!]$ is a recognizable step function.

Note that in particular conditions (2) and (3) are not purely syntactic, but use the semantics of formulas. By Proposition 5.5 clearly each syntactically restricted formula $\varphi \in \mathrm{MSO}(S, \Sigma)$ is restricted. Vice versa, by Propositions 5.5 and 5.3 each restricted formula is equivalent to a syntactically restricted formula.

6.1 Definable Series are Recognizable

As in the string case [13, 14] and that of ranked trees [16], we prove this implication by induction on the structure of the formula. For any formula $\varphi \in \mathrm{MSO}(S,\Sigma)$, we let $S_\varphi = S_{\mathrm{val}(\varphi)}$, the subsemiring of S generated by all constants occurring in φ.

Lemma 6.3 *Let $\varphi, \psi \in \mathrm{MSO}(S,\Sigma)$.*

1. *Let φ be atomic or the negation of an atomic formula. Then $[\![\varphi]\!]$ is a recognizable step function.*
2. *If $[\![\varphi]\!]$ and $[\![\psi]\!]$ are recognizable tree series, then $[\![\varphi \vee \psi]\!]$ is recognizable.*
3. *Assume that $\mathrm{val}(\varphi)$ and $\mathrm{val}(\psi)$ commute elementwise and that $[\![\varphi]\!] \in \mathrm{Rec}(S_\varphi, \Sigma_\varphi)$ and $[\![\psi]\!] \in \mathrm{Rec}(S_\psi, \Sigma_\psi)$. Then $[\![\varphi \wedge \psi]\!]$ is recognizable.*
4. *If $[\![\varphi]\!]$ is recognizable, then $[\![\exists x.\varphi]\!]$ and $[\![\exists X.\varphi]\!]$ are recognizable.*
5. *If φ is unambiguous, then $[\![\forall X.\varphi]\!]$ is a recognizable step function.*

Proof 1. If $\varphi = s$ where $s \in S$, then $[\![\varphi]\!] = \mathbb{1}_{U_\Sigma} \cdot s$ is a recognizable step function. For the other cases apply Proposition 5.2.

2. and 3. Let $\mathcal{V} = \mathrm{Free}(\varphi) \cup \mathrm{Free}(\psi)$. By Definition 4.1, $[\![\varphi \vee \psi]\!] = [\![\varphi]\!]_\mathcal{V} + [\![\psi]\!]_\mathcal{V}$ and $[\![\varphi \wedge \psi]\!] = [\![\varphi]\!]_\mathcal{V} \odot [\![\psi]\!]_\mathcal{V}$. Now we can apply Proposition 3.2 and Lemma 4.2.

4. Analogous to the corresponding result for ranked trees (compare [16]), using Lemma 3.3.

5. Since φ is unambiguous, so is $\forall X.\varphi$, and we can apply Proposition 5.2. □

For the proof that recognizability is preserved under universal first order quantification, we use the proof technique of [13, 14].

Lemma 6.4 *Let $\varphi \in \mathrm{MSO}(S,\Sigma)$ such that $[\![\varphi]\!]$ is a recognizable step function. Then $[\![\forall x.\varphi]\!]$ is recognizable.*

Proof Let $\mathcal{W} = \mathrm{Free}(\varphi) \cup \{x\}$ and $\mathcal{V} = \mathrm{Free}(\forall x.\varphi) = \mathcal{W} \setminus \{x\}$. By Proposition 4.2, $[\![\varphi]\!]_\mathcal{W} = \sum_{j=1}^n \mathbb{1}_{L_j} \cdot s_j$ for some $n \geq 0$, $s_j \in S$, and recognizable tree languages $L_1, \ldots, L_n \subseteq U_{\Sigma_\mathcal{W}}$. We can assume that the sets $L_1, \ldots, L_n$ form a partition of $U_{\Sigma_\mathcal{W}}$.

Let $\widetilde{\Sigma} = \Sigma \times \{1, \ldots, n\}$. A tree $\xi \in U_{\widetilde{\Sigma}_\mathcal{V}}$ corresponds to the tuple (ζ, ν) where $\zeta \in U_{\Sigma_\mathcal{V}}$ is obtained from ξ by dropping the second component from the label of every node, and $\nu : \mathrm{pos}(\xi) \to \{1, \ldots, n\}$ is defined by $\nu(w) = j$ if $\xi(w) = (\sigma, j, f)$ for some $\sigma \in \Sigma$ and $f \in \{0,1\}^\mathcal{V}$. Vice versa, every such tuple (ζ, ν) corresponds to a tree $\xi \in U_{\widetilde{\Sigma}_\mathcal{V}}$. Hence we can assume that elements of $U_{\widetilde{\Sigma}_\mathcal{V}}$ have the form (ζ, ν). Then let

$$\widetilde{L} = \{(\zeta, \nu) \in U_{\widetilde{\Sigma}_\mathcal{V}} \mid \forall w \in \mathrm{pos}(\zeta), 1 \leq j \leq n: \text{ if } \nu(w) = j, \text{ then } \zeta[x \to w] \in L_j\}.$$

Note that for every $\zeta \in U_{\Sigma_\mathcal{V}}$ there is a unique ν such that $(\zeta, \nu) \in \widetilde{L}$, because the L_j's form a partition of $U_{\Sigma_\mathcal{W}}$. Next we claim that $\widetilde{L}$ is a recognizable tree language.

Let $j \in \{1, \ldots, n\}$. Since $L_j \subseteq U_{\Sigma_\mathcal{W}}$ is a recognizable tree language, L_j is definable by a sentence ψ_j over $\Sigma_\mathcal{W}$. By a standard procedure, we can find a formula ψ'_j over Σ with $\mathrm{Free}(\psi'_j) \subseteq \mathcal{W}$ which defines L_j.

Now we can follow the argument in the proof of Lemma 5.4 of [14] to obtain a sentence defining the language $\widetilde{L}$, proving our claim.

By Theorem 3.1 of [29] (also cf. [33]) there is a wta $\widetilde{M} = (\widetilde{Q}, \widetilde{\Sigma}_{\mathcal{V}}, \widetilde{A}, \widetilde{F})$ over $\mathbb{B}$ such that $\mathcal{L}(\widetilde{M}) = \widetilde{L}$. We can assume that $\widetilde{M}$ is deterministic. Now we define the wta $M = (\widetilde{Q}, \widetilde{\Sigma}_{\mathcal{V}}, A, \gamma)$ over S by constructing, for every $q \in \widetilde{Q}$ and $(\sigma, l, f) \in \widetilde{\Sigma}_{\mathcal{V}}$, a wsa $A_{q,(\sigma,l,f)}$ over S and $\widetilde{Q}$ such that for every $m \geq 0$ and $q_1, \ldots, q_m \in \widetilde{Q}$,

$$(\|A_{q,(\sigma,l,f)}\|,\ q_1 \ldots q_m) = \begin{cases} s_l & \text{if } (\|\widetilde{A}_{q,(\sigma,l,f)}\|,\ q_1, \ldots, q_m) = 1 \\ 0 & \text{otherwise.} \end{cases}$$

It is obvious how to construct such a $A_{q,(\sigma,l,f)}$. Moreover, for every $q \in \widetilde{Q}$, we define $\gamma(q) = 1$ if $q \in \widetilde{F}$, and 0 otherwise. Clearly, M is also deterministic. Thus, for every $(\zeta, \nu) \in U_{\widetilde{\Sigma}_{\mathcal{V}}}$, we have that

$$(r_M, (\zeta, \nu)) = \begin{cases} \prod_{w \in \mathrm{pos}(\zeta)} s_{\nu(w)} & \text{if } (\zeta, \nu) \in \widetilde{L} \\ 0 & \text{otherwise.} \end{cases}$$

Now we define the deterministic relabeling $\tau : S\langle\langle U_{\widetilde{\Sigma}_{\mathcal{V}}}\rangle\rangle \to S\langle\langle U_{\Sigma_{\mathcal{V}}}\rangle\rangle$ by $\tau((\sigma, \nu, f)) = (\sigma, f)$ for every $(\sigma, \nu, f) \in \widetilde{\Sigma}_{\mathcal{V}}$. Then for $\zeta \in U_{\Sigma_{\mathcal{V}}}$ let $\nu : \mathrm{pos}(\zeta) \to \{1, \ldots, n\}$ is the unique mapping such that $(\zeta, \nu) \in \widetilde{L}$. Observing the form of $[\![\varphi]\!]$, we have:

$$\begin{aligned}(\tau(r_M), \zeta) &= \sum_{(\zeta,\theta) \in \tau^{-1}(\zeta)} (r_M, (\zeta, \theta)) = (r_M, (\zeta, \nu)) = \prod_{w \in \mathrm{pos}(\zeta)} s_{\nu(w)} \\ &= \prod_{w \in \mathrm{pos}(\zeta)} ([\![\varphi]\!], \zeta[x \to w]) = ([\![\forall x.\varphi]\!], \zeta).\end{aligned}$$

Hence $\tau(r_M) = [\![\forall x.\varphi]\!]$ and thus, by Lemma 3.3, $[\![\forall x.\varphi]\!]$ is recognizable. □

Theorem 6.5 *Let* $\varphi \in \mathrm{MSO}(S, \Sigma)$ *be restricted. Then* $[\![\varphi]\!] \in \mathrm{Rec}(S, \Sigma)$.

Proof We claim that $[\![\varphi]\!] \in \mathrm{Rec}(S_\varphi, \Sigma_\varphi)$ for any formula $\varphi \in \mathrm{MSO}(S, \Sigma)$, which implies the result. Our claim follows by induction over the structure of φ from Lemmas 6.3 and 6.4, applied to suitable subsemirings of S. □

Corollary 6.6 *Let S be a computable semiring. There is an effective procedure which produces, for a given* $\mathrm{srMSO}(S, \Sigma)$*-formula* φ*, a wta M such that* $[\![\varphi]\!] = r_M$.

Proof We can construct M by following the proof of Proposition 5.5 and Theorem 6.5. □

6.2 Recognizable Series are Definable

In this section we will construct for each wta M an MSO-sentence θ_M such that $r_M = [\![\theta_M]\!]$. Moreover, if S is commutative, then we can even take θ_M to be a syntactically restricted existential MSO-sentence.

When proving that for every wta $M = (Q, \Sigma, A, \gamma)$ the tree series r_M is definable, we will have to specify the behavior of the wsa $A_{q,\sigma}$ on an input tree ξ by means of a formula. In order to do this succinctly, we make the following assumptions for the rest of this section. Each wsa $A_{q,\sigma}$ is described by the tuple $(P_{q,\sigma}, \lambda_{q,\sigma}, \mu_{q,\sigma}, \nu_{q,\sigma})$. We assume that the state sets of two different wsa are disjoint. Moreover, we denote the union of all the state sets $P_{q,\sigma}$, initial weight mappings $\lambda_{q,\sigma}$, transition functions $\mu_{q,\sigma}$, and final weight mappings $\nu_{q,\sigma}$ by P_A, λ_A, μ_A, and ν_A, respectively. Moreover, Q and P_A are disjoint. Given a triple α of elements, we denote by α_i its i-th component $(1 \leq i \leq 3)$.

Definition 6.7 Let $M = (Q, \Sigma, A, \gamma)$ be a wta and $\xi \in U_\Sigma$.

- The set of all *Q-transitions* (*of* M) is the set $B_M = \bigcup_{q \in Q, \sigma \in \Sigma} P_{q,\sigma} \times Q \times P_{q,\sigma}$.
- An *extended run on* ξ is a triple (q, s, t) where $q \in Q$, $s : \mathrm{pos}(\xi) \setminus \{\varepsilon\} \to B_M$, and $t : \mathrm{pos}_{\mathrm{leaf}}(\xi) \to P_A$ where $\mathrm{pos}_{\mathrm{leaf}}(\xi)$ are the positions of ξ which are leaves.
- An extended run (q, s, t) on ξ is *valid* if for every $w \in \mathrm{pos}(\xi)$ the following conditions hold:
  1. for every i with $1 \leq i \leq \mathrm{rk}_\xi(w)$, we have that $s(wi)_1, s(wi)_3 \in P_{q,\xi(w)}$ if $w = \varepsilon$, and $s(wi)_1, s(wi)_3 \in P_{s(w)_2,\xi(w)}$ otherwise,
  2. for every i with $1 \leq i \leq \mathrm{rk}_\xi(w) - 1$ we have that $s(wi)_3 = s(w(i+1))_1$, and
  3. if $w \in \mathrm{pos}_{\mathrm{leaf}}(\xi)$ and $w = \varepsilon$ (i.e., $\xi \in \Sigma$), then $t(w) \in P_{q,\xi(\varepsilon)}$ and if $w \in \mathrm{pos}_{\mathrm{leaf}}(\xi)$ and $w \neq \varepsilon$, then $t(w) \in P_{s(w)_2,\xi(w)}$.
- The *set of all valid extended runs on* ξ is denoted by $R^e_M(\xi)$.
- For every valid extended run (q, s, t) on ξ we define its weight $\mathrm{wt}(q, s, t) \in S$ by

$$\mathrm{wt}(q, s, t) = \prod_{w \in \mathrm{pos}(\xi)} \mathrm{wt}(q, s, t)_w$$

  where in the product we follow the depth-first left-to-right traversal over $\mathrm{pos}(\xi)$; and for every $w \in \mathrm{pos}(\xi) \setminus \mathrm{pos}_{\mathrm{leaf}}(\xi)$ we let

$$\mathrm{wt}(q, s, t)_w = \lambda_A(s(w1)_1) \cdot \mu_A(s(w1)) \cdot \cdots \cdot \mu_A(s(w\,\mathrm{rk}_\xi(w))) \cdot \nu_A(s(w\,\mathrm{rk}_\xi(w))_3);$$

  if $w \in \mathrm{pos}_{\mathrm{leaf}}(\xi)$, we let $\mathrm{wt}(q, s, t)_w = \lambda_A(t(w)) \cdot \nu_A(t(w))$.

Clearly, for every $\xi \in U_\Sigma$, we can express the weight of a run $\kappa \in R_M(\xi)$ also in terms of the weights of extended runs. For this we define the mapping $\mathrm{proj}_\xi : R^e_M(\xi) \to R_M(\xi)$ for every $(q, s, t) \in R^e_M(\xi)$ and $w \in \mathrm{pos}(\xi)$ by

$$\mathrm{proj}_\xi((q, s, t))(w) = \begin{cases} q & \text{if } w = \varepsilon \\ s(w)_2 & \text{otherwise.} \end{cases}$$

Observation 6.8 *Let $M = (Q, \Sigma, A, \gamma)$ be a wta. Then for every run $\kappa \in R_M(\xi)$ we have that*

$$\mathrm{wt}_M(\kappa) = \sum_{(q,s,t) \in \mathrm{proj}_\xi^{-1}(\kappa)} \mathrm{wt}(q, s, t).$$

Proof This follows directly from the definition of $\mathrm{wt}_M(\kappa)$, applying the distributivity law (which preserves the order of the factors) and putting all the individual runs of the wsa at positions of ξ together to a single extended run on ξ. □

Theorem 6.9 *Let $r \in S\langle\langle U_\Sigma\rangle\rangle$ be recognizable. Then r is* MSO*-definable. Moreover, if S is commutative, then r is* srEMSO(S,Σ)*-definable.*

Proof Let r be recognized by some wta $M = (Q, \Sigma, A, \gamma)$ over S. As preparation for the proof of both statements, we will first describe valid extended runs on trees in U_Σ by means of an srMSO(S,Σ)-formula valid_M.

Subsequently we will use second order variables X_q $(q \in Q)$, Y_t $(t \in B_M)$, and Z_p $(p \in P_A)$. We choose arbitrary but fixed enumerations $Q = \{q_1, \ldots, q_n\}$, $B_M = \{t_1, \ldots, t_m\}$, and $P_A = \{p_1, \ldots, p_l\}$. In finite conjunctions (and disjunctions) over the index sets Q, B_M, and P_A we follow the ordering induced by these enumerations.

Recall the macros $\mathrm{next}(x, y)$ and $\mathrm{firstChild}(x, y)$ from Example 5.4. Moreover, let

- $\mathrm{root}(x) = \forall y.\neg(\mathrm{desc}(y, x))$ and $\neg\mathrm{root}(x) = \exists y.\mathrm{desc}(y, x)$,
- $\mathrm{leaf}(x) = \forall y.\neg\mathrm{desc}(x, y)$ and $\neg\mathrm{leaf}(x) = \exists y.\mathrm{firstChild}(x, y)$, and
- $\mathrm{lastChild}(x, y) = \mathrm{desc}(x, y) \wedge \forall z.(\mathrm{desc}(x, z) \stackrel{+}{\rightarrow} z \le y)$.

The *extended run formula* is the formula $\mathrm{run}_M \in \mathrm{MSO}^-(\Sigma)$ which checks whether the given structure corresponds to an extended run. Formally,

$$\mathrm{run}_M = \forall x.\big(\mathrm{run}_{M,1}(x) \wedge \mathrm{run}_{M,2}(x) \wedge \mathrm{run}_{M,3}(x)\big)$$

where

$$\mathrm{run}_{M,1}(x) = \mathrm{root}(x) \stackrel{+}{\rightarrow} \left(\bigvee_{q\in Q} \left((x \in X_q) \wedge \bigwedge_{\substack{q' \in Q,\\ q' \neq q}} \neg(x \in X_{q'}) \right) \wedge \bigwedge_{t \in B_M} \neg(x \in Y_t) \right)$$

$$\mathrm{run}_{M,2}(x) = \neg\mathrm{root}(x) \stackrel{+}{\rightarrow} \left(\bigvee_{t\in B_M} \left((x \in Y_t) \wedge \bigwedge_{\substack{t' \in B_M,\\ t' \neq t}} \neg(x \in Y_{t'}) \right) \wedge \bigwedge_{q \in Q} \neg(x \in X_q) \right)$$

$$\mathrm{run}_{M,3}(x) = \left(\mathrm{leaf}(x) \stackrel{+}{\rightarrow} \bigvee_{p\in P_A} \left((x \in Z_p) \wedge \bigwedge_{\substack{p' \in P_A,\\ p' \neq p}} \neg(x \in Z_{p'}) \right) \right)$$
$$\wedge \left(\left(\bigvee_{p \in P_A} (x \in Z_p) \right) \stackrel{+}{\rightarrow} \mathrm{leaf}(x) \right).$$

Next we define the formula valid_M which has all the valid extended runs as models. We set

$$\mathrm{valid}_M = \left(\mathrm{run}_M \wedge \mathrm{valid}_{M,1} \wedge \mathrm{valid}_{M,2} \wedge \mathrm{valid}_{M,3} \right)^+$$

where $\text{valid}_{M,i}$ describes Property (i) of the definition of valid extended runs. Formally,

$$\text{valid}_{M,1} = \forall x. \bigwedge_{\sigma\in\Sigma} \bigwedge_{q\in Q} \bigwedge_{(p,q,p')\in B_M} \Big(\text{label}_\sigma(x) \wedge ((x \in X_q) \vee (x \in Y_{(p,q,p')}))\Big) \xrightarrow{+} \text{in}_Y(x,q,\sigma)$$

where $\text{in}_Y(x,q,\sigma) = \forall y.\ \text{desc}(x,y) \xrightarrow{+} \bigvee_{\substack{p_1,p_2\in P_{q,\sigma},\\ q'\in Q}} (y \in Y_{(p_1,q',p_2)})$

$$\text{valid}_{M,2} = \forall x.\forall y.\ \text{next}(x,y) \xrightarrow{+} \Bigg(\bigwedge_{(p,q,p')\in B_M} (x \in Y_{(p,q,p')}) \xrightarrow{+} \bigvee_{\substack{(p'',q',\overline{p})\in B_M,\\ p'=p''}} (y \in Y_{(p'',q',\overline{p})})\Bigg)$$

$$\text{valid}_{M,3} = \forall x. \bigwedge_{\sigma\in\Sigma} \bigwedge_{q\in Q} \bigwedge_{(p,q,p')\in B_M} \Big(\text{leaf}(x) \wedge \text{label}_\sigma(x) \wedge ((x \in X_q) \vee (x \in Y_{(p,q,p')})) \xrightarrow{+} \bigvee_{p\in P_{q,\sigma}} (x \in Z_p)\Big).$$

Now we prove the first statement of the theorem. Intuitively, we describe the weight of a valid extended run by means of the formula φ_M. According to the definition, the outermost universal quantification over x performs a depth-first left-to-right traversal over the input tree ξ, thereby touching the positions of ξ in the linear ordering $\sqsubseteq_\xi$. When a leaf $w \in \text{pos}(\xi)$ with label σ and state $p \in P_A$ is next, the interpretation of the formula accumulates the product $\lambda_A(p) \cdot \nu_A(p)$. When a non-leaf w is next, it accumulates the initial weight $\lambda_A(p)$ if the valid extended run on ξ at $w1$ is the state (p,q,p') for some q, p'; moreover, using another universal quantification, the descendants of w are traversed one by one from left to right, and at each descendant the value $\mu_A(p,q,p')$ is appended to the current product; at the last child of w, additionally the value $\nu_A(p')$ is taken into the product as its last factor. Finally, at the root of ξ, the root weight is accumulated. Recall the definition of the macro $\varphi \to \psi$ given before Example 5.4

$$\begin{aligned}\varphi_M = \text{valid}_M \wedge \forall x.\Bigg[&\Bigg(\text{leaf}(x) \to \bigwedge_{p\in P_A} \big((x \in Z_p) \to \lambda_A(p)\cdot\nu_A(p)\big)\Bigg)\\ &\wedge \Bigg(\neg\text{leaf}(x) \to \forall y. \bigwedge_{(p,q,p')\in B_M} \Big(\big[(\text{firstChild}(x,y) \wedge (y \in Y_{(p,q,p')})) \to \lambda_A(p)\big]\\ &\quad \wedge \big[(\text{desc}(x,y) \wedge (y \in Y_{(p,q,p')})) \to \mu_A(p,q,p')\big]\\ &\quad \wedge \big[(\text{lastChild}(x,y) \wedge (y \in Y_{(p,q,p')})) \to \nu_A(p')\big]\Big)\Bigg)\\ &\wedge \Bigg(\text{root}(x) \to \bigvee_{q\in Q} ((x \in X_q) \wedge \gamma(q))\Bigg)\Bigg].\end{aligned}$$

Due to the nested universal quantification, the formula φ_M is not syntactically restricted. Finally, let

$$\theta_M = \exists X_{q_1}. \ldots \exists X_{q_n}. \exists Y_{t_1}. \ldots \exists Y_{t_m}. \exists Z_{p_1}. \ldots \exists Z_{p_l}. \varphi_M.$$

Then, by Observation 6.8 and the analysis of the weights of valid extended runs, we obtain $r_M = [\![\theta_M]\!]$.

Now let us prove the second statement of the theorem and assume that S is commutative. Then we replace in θ_M the formula φ_M by the formula φ_M^* and obtain the formula θ_M^*.

$$\begin{aligned}\varphi_M^* = \text{valid}_M \wedge \forall y.\Bigg[&\Big(\text{leaf}(y) \to \bigwedge_{p\in P_A} ((y \in Z_p) \to \lambda_A(p) \cdot \nu_A(p))\Big)\\ \wedge \Bigg(&\bigwedge_{(p,q,p')\in B_M} \Big(\big[(\exists x.\text{firstChild}(x,y) \wedge (y \in Y_{(p,q,p')})) \to \lambda_A(p)\big]\\ &\wedge \big[(y \in Y_{(p,q,p')}) \to \mu_A(p,q,p')\big]\\ &\wedge \big[(\exists x.\text{lastChild}(x,y) \wedge (y \in Y_{(p,q,p')})) \to \nu_A(p')\big]\Big)\Bigg)\\ \wedge \Bigg(&\text{root}(y) \to \bigvee_{q\in Q} ((y \in X_q) \wedge \gamma_A(q))\Bigg)\Bigg].\end{aligned}$$

Now it can be easily checked that θ_M^* is syntactically restricted. Since S is commutative, we obtain that $[\![\theta_M^*]\!] = r_M$. □

Finally we show that the conditions in Theorem 6.9(2) on the commutation of elements cannot be dropped.

Theorem 6.10 *There is a semiring S and a recognizable tree series $r \in S\langle\langle U_\Sigma\rangle\rangle$ which is not* srMSO(Σ, S)*-definable.*

Proof We consider the semiring S of finite formal languages $S = (\text{Fin}(\{a,b\}^*), \cup, \circ, \emptyset, \{\varepsilon\})$ where $\text{Fin}(\{a,b\}^*)$ comprises all finite subsets of $\{a,b\}^*$ and $\circ$ is the usual (non-commutative) concatenation of formal languages; we will identify a singleton $\{w\} \in \mathcal{P}(\{a,b\}^*)$ with its element w. Let $\Sigma = \{\sigma\}$ and *flat* $\in S\langle\langle U_\Sigma\rangle\rangle$ be the tree series such that for every $\xi \in U_\Sigma$ we have that

$$(\mathit{flat}, \xi) = \begin{cases} a^n b^n & \text{if } \xi = \sigma(\underbrace{\sigma, \ldots, \sigma}_{n}) \text{ for some } n \geq 1\\ \emptyset & \text{otherwise.}\end{cases}$$

Now consider the wta $M = (Q, \Sigma, A, \gamma)$ over S where $Q = \{q_0, q_1\}$, $\gamma(q_0) = \emptyset$, and $\gamma(q_1) = \{\varepsilon\}$. Moreover, for every $q \in Q$ we need a wsa $A_{q,\sigma}$ such that for every

$w \in Q^*$ we have

$$(\|A_{q_0,\sigma}\|, w) = \begin{cases} a & \text{if } w = \varepsilon \\ \emptyset & \text{otherwise} \end{cases}$$

and

$$(\|A_{q_1,\sigma}\|, w) = \begin{cases} b^n & \text{if } w = q_0^n \text{ and } n \geq 1 \\ \emptyset & \text{otherwise.} \end{cases}$$

It is clear that such two wsa exist, and in fact, $r_M = \mathit{flat}$.

We prove that $\mathit{flat} \notin \text{srMSO}(\Sigma, S)$ by contradiction. Assume that φ is a sentence in $\text{srMSO}(\Sigma, S)$ such that $[\![\varphi]\!] = \mathit{flat}$. We translate the scenario on trees into one on words. For this we transform the tree $\xi = \sigma(\underbrace{\sigma, \ldots, \sigma}_{n})$ into the word $\xi' = \underbrace{\sigma \ldots \sigma}_{n} \sigma$, and the sentence φ into a sentence φ' of the syntactically restricted weighted MSO on words (as defined in [14]) such that $([\![\varphi]\!], \xi) = ([\![\varphi']\!], \xi')$. The sentence φ' is obtained from φ by replacing

- $x \sqsubseteq y$ by: $x \leq y$,
- $x \leq y$ by: $((x = y) \wedge \max(y)) \vee (\exists z.(x \leq y) \wedge (y \leq z) \wedge \neg(z \leq y))^+$,
- $\text{desc}(x, y)$ by: $\max(x) \wedge \neg(x \leq y)$

where $\max(y)$ is the macro $\forall z.z \leq y$. Since φ is syntactically restricted, so is φ'.

By Theorem 4.7 of [14], the power series $[\![\varphi']\!]$ is recognizable by a wsa A over S and Σ. That is, for the input word σ^{n+1} the wsa A computes the semiring value $a^n b^n$, for every $n \geq 1$. But this will lead to a contradiction using a pumping argument as follows.

We choose $n = |P| \cdot l + 1$ where P is the set of states of A and l is the maximal length of a word which occurs as weight in the initial-final weight function or the transitions of A. Due to S and the fact that $([\![\varphi']\!], \sigma^{n+1}) = a^n b^n$, there is at least one path of A with label σ^{n+1} and weight $a^n b^n$. This path is longer than $|P|$, thus it uses a state at least twice. Now consider an input word σ^{k+n+1} where k is the length of the identified cycle. Since $(\|A\|, \sigma^{k+n+1}) = a^{k+n} b^{k+n}$, the cycle must have a weight $\neq \varepsilon$, and an easy analysis of its weight yields a contradiction. □

7 The Ranked Case

In this section we derive from the results of the previous sections the fact that every ranked tree series r over an arbitrary semiring is recognizable if and only if it is definable by a ranked $\text{srEMSO}(S, \Sigma)$-sentence. For this we first introduce recognizable ranked trees series and ranked $\text{srEMSO}(S, \Sigma)$-logic.

A *ranked alphabet* is a tuple (Σ, rk) such that Σ is an alphabet and the mapping $\text{rk} : \Sigma \to \mathbb{N}$ associates with every symbol σ a natural number, called the *rank of* σ. Then $\Sigma^{(k)} = \{\sigma \in \Sigma \mid \text{rk}(\sigma) = k\}$ $(k \in \mathbb{N})$. A *ranked* Σ*-tree over* (Σ, rk) is a Σ-tree ξ such that $\text{rk}(\xi(w)) = \text{rk}_\xi(w)$ for every $w \in \text{pos}(\xi)$. We denote the set of ranked

Σ-trees by T_Σ; clearly, $T_\Sigma \subseteq U_\Sigma$. Thus, in a Σ-tree ξ (in contrast to ranked trees) there can be different positions $w, w' \in \text{pos}(\xi)$ which are labeled by the same symbol (i.e., $\xi(w) = \xi(w')$), but the ranks of w and w' are different (i.e., $\text{rk}_\xi(w) \neq \text{rk}_\xi(w')$).

A *weighted ranked tree automaton over* S (for short: ranked wta) is a tuple $N = (Q, \Sigma, \mu, \gamma)$ where Q is a finite nonempty set (of *states*), Σ is the ranked alphabet (of *input symbols*), $\mu = (\mu_k \mid k \in \mathbb{N})$ is a *family of transition mappings* $\mu_k : \Sigma^{(k)} \to S^{Q^k \times Q}$, and $\gamma : Q \to S$ (the *root weight function*).

A *run of* N *on* $\xi \in T_\Sigma$ is a mapping $\kappa : \text{pos}(\xi) \to Q$. Then the *weight* $\text{wt}_N(\kappa) \in S$ *of* κ is defined by

$$\text{wt}_N(\kappa) = \prod_{w \in \text{pos}(\xi)} \mu_k(\sigma)_{\kappa(w1)\ldots\kappa(wk),\kappa(w)}$$

where, in the product, we follow the linear ordering $\sqsubseteq_\xi$. The *set of all runs of* N *on* ξ is denoted by $R_N(\xi)$. The *run semantics of* N is the tree series $r_N \in S\langle\langle T_\Sigma \rangle\rangle$ such that for every $\xi \in T_\Sigma$

$$(r_N, \xi) = \sum_{\kappa \in R_N(\xi)} \text{wt}(\kappa) \cdot \gamma(\kappa(\varepsilon)).$$

The next lemma shows an obvious relationship between ranked wta and wta restricted to ranked trees. For a ranked wta (Q, Σ, μ, γ) we call Σ its *input alphabet*; similarly for wta.

Lemma 7.1

1. *For every ranked wta* N *with ranked input alphabet* Σ *there is a wta* M *such that* $(r_M)|_{T_\Sigma} = r_N$ *and* $(r_M)|_{(U_\Sigma \setminus T_\Sigma)} = \widetilde{0}$.
2. *For every wta* M *with input alphabet* Σ *there is a ranked wta* N *such that* $(r_M)|_{T_\Sigma} = r_N$.

Proof For the first statement let $N = (Q, \Sigma, \mu, \gamma)$ be a ranked wta. For every $q \in Q$, $k \geq 0$, and $\sigma \in \Sigma^{(k)}$ we define the power series $r_{q,\sigma} \in S\langle\langle Q^* \rangle\rangle$ by letting $(r_{q,\sigma}, \vec{q}) = \mu_k(\sigma)_{\vec{q},q}$ if $\vec{q} \in Q^k$, and 0 otherwise. Clearly, there is a wsa $A_{q,\sigma}$ such that $r_{q,\sigma} = \|A_{q,\sigma}\|$. Then the wta $M = (Q, \Sigma, A, \gamma)$ satisfies statement 1.

The construction for the second statement is, roughly speaking, the reverse of that one for the first statement. □

The *ranked weighted MSO-logic* is the same as the weighted MSO-logic, but we drop the atomic formulas of the form $\text{desc}(x, y)$ and $(x \leq y)$, and we add the atomic formulas $\text{edge}_i(x, y)$ where $1 \leq i \leq \max\Sigma$ and $\max\Sigma = \max\{k \mid \Sigma^{(k)} \neq \emptyset\}$. Note that we keep the atomic formula $(x \sqsubseteq y)$ because they are needed in the disambiguation φ^+ of a formula φ.

Theorem 7.2 *Let* Σ *be a ranked alphabet*, S *be an arbitrary semiring*, *and* $r \in S\langle\langle T_\Sigma \rangle\rangle$. *Then* r *is recognizable if and only if* r *is definable by a syntactically restricted ranked weighted MSO-sentence*.

This result follows immediately from Lemmas 7.3 and 7.4 given below. First we show:

Lemma 7.3 *Let* Σ *be a ranked alphabet and* φ *be a syntactically restricted ranked weighted MSO-sentence. Then* $[\![\varphi]\!] \in S\langle\langle T_\Sigma\rangle\rangle$ *is recognizable.*

Proof Given φ, we can construct the weighted MSO-sentence ψ by replacing in φ every atomic formula $\mathrm{edge}_i(x, y)$ by the macro:

$$\left(\exists y_1.\dots.\exists y_i.\mathrm{firstChild}(x, y_1) \wedge \bigwedge_{1\le j\le i-1} \mathrm{next}(y_j, y_{j+1}) \wedge (y_i = y)\right)^+.$$

Then ψ is syntactically restricted and clearly, $[\![\varphi]\!] = [\![\psi]\!]|_{T_\Sigma}$. By Theorem 6.5 there is a wta M such that $r_M = [\![\psi]\!]$. By Lemma 7.1(2) there is a ranked wta N such that $r_N = (r_M)|_{T_\Sigma}$. Hence $[\![\varphi]\!] = r_N$. □

Lemma 7.4 *Let* Σ *be a ranked alphabet and* $r \in S\langle\langle T_\Sigma\rangle\rangle$ *be recognizable. Then* r *is definable by a syntactically restricted ranked weighted MSO-sentence.*

Proof Let $N = (Q, \Sigma, \mu, \gamma)$ be a ranked wta. The *set of all transitions at* Σ*-symbols* is the set $B_N = \{(\vec{q}, \sigma, q) \mid m \ge 0, \vec{q} \in Q^m, \sigma \in \Sigma^{(m)}, q \in Q\}$. Choose an unambiguous run-formula ψ with free variables Y_t where $t \in B_N$ (e.g., as in Definition 5.10 of [16]) such that for every $\xi \in T_\Sigma$, there is a bijection between the set $R_N(\xi)$ and the set of those assignments (ξ, ρ) which satisfy ψ. Then we define the weighted MSO-sentence

$$\varphi = \psi \wedge \forall y.\Bigg(\bigwedge_{\substack{(\vec{q},\sigma,q)\in B_N\\ \vec{q}\in Q^m, m\ge 0}} \Big[(y \in Y_{(\vec{q},\sigma,q)}) \to \mu_m(\sigma)_{\vec{q},q}\Big]$$
$$\wedge \Big[\mathit{root}(y) \to \bigvee_{(\vec{q},\sigma,q)\in B_N} (y \in Y_{(\vec{q},\sigma,q)}) \wedge \gamma(q)\Big]\Bigg).$$

Then φ is syntactically restricted and $r_N = [\![\varphi]\!]$. □

8 An XML Example

In this section we show how formulas of weighted MSO-logic can be used to specify quantitative queries in XML-oriented databases. In fact, every formula used here will be syntactically restricted. For this purpose, let us assume that we want to maintain our private database in which we have stored all the bibtex entries which are relevant for our work. The syntax of such a database `bibliography` can be easily expressed by the following document type definition (DTD) [1, 6, 24] (here we only show a fragment):

```
<!DOCTYPE biblography [
   <!ELEMENT bibliography (entry)*>
   <!ELEMENT entry        ((key)+,article|book)>
   <!ELEMENT key          (#PCDATA)>
   <!ELEMENT article      (mandatory,(empty|optional)>
   <!ELEMENT mandatory    (author*,title,journal,year)>
   <!ELEMENT author       (#PCDATA)>
   <!ELEMENT title        (#PCDATA)>
   ...
   <!ELEMENT empty        EMPTY>
   <!ELEMENT optional     (volume,pages)>
   ...
]>
```

In the DTD the comma, vertical bar, and star represent the sequence, alternative, resp. iteration of syntactic constructs; at `(#PCDATA)` an arbitrary text is allowed. This DTD specifies a set of XML-documents, in particular, a set of databases with bibtex entries; note that an entry can have any number of keys.

An example of such a database is partially shown in Fig. 1 as an unranked tree. It has three entries; the leftmost entry contains the mandatory pieces of information only (*incomplete entry*), the rightmost one contains the mandatory and the optional pieces of information (*complete entry*). Now let us assume that we would like to complete our database in the sense of adding the optional pieces of information to every incomplete article entry. However, before doing so, we would like to estimate the effort for this maintenance, i.e., count the number of incomplete bibtex entries. For this purpose we can use the following formula of weighted MSO-logic:

$$\texttt{how-many-incomplete?} = \exists x.\, \mathrm{label}_{\texttt{article}}(x) \wedge \texttt{incomplete}(x)$$

$$\texttt{incomplete}(x) = \exists y.\, \big(\mathrm{desc}(x,y)\ \wedge\ \mathrm{label}_{\texttt{empty}}(y)\big).$$

If we interpret our formula `how-many-incomplete?` on the tree of Fig. 1 where the middle entry is assumed to be incomplete, then we obtain the value 2 (assuming that we use the usual semiring of natural numbers).

To show another use of weighted MSO-logic, let us now assume that the document type definition as shown above is extended as follows:

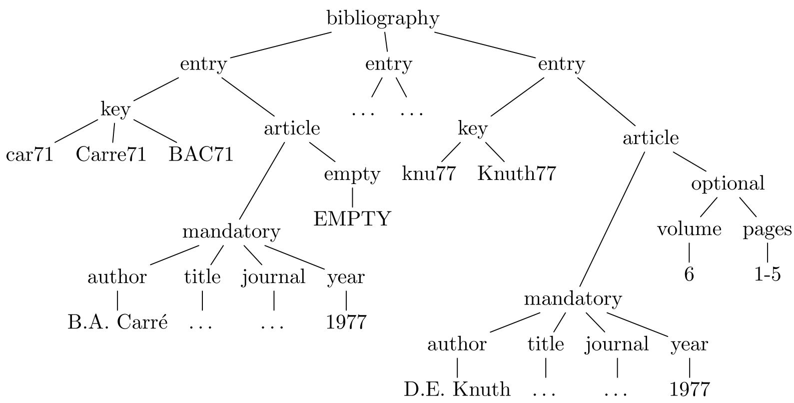

Fig. 1 An example of an XML-database

```
<!ELEMENT book          (mandatory-b,(empty-b|optional-b)>
<!ELEMENT mandatory-b   (author*,title,publisher,year)>
<!ELEMENT empty-b       EMPTY>
<!ELEMENT optional-b    (volume,edition,summary)>
<!ELEMENT summary       (#PCDATA)>
```

In the part `optional-b` a short summary of the book should occur. Now let us again try to estimate the effort for completing bibtex entries. Clearly, to write a summary of a book takes much more effort than just to add the optional information pieces of an article. So let us describe the corresponding efforts by k and m (time units), respectively, for some natural numbers k and m. Then the estimation of the total effort is computed by the following syntactically restricted sentence in $\mathrm{MSO}(\mathbb{N}, \Sigma)$:

$$\texttt{how-much-effort?} = \exists x.\big(\mathrm{label}_{\texttt{book}}(x) \wedge \texttt{incomplete}(x) \wedge k\big) \\ \vee \big(\mathrm{label}_{\texttt{article}}(x) \wedge \texttt{incomplete}(x) \wedge m\big).$$

If, e.g., our database contains 20 incomplete book entries and 500 incomplete article entries, then the interpretation of `how-much-effort?` on the corresponding tree would yield the effort $20 \cdot k + 500 \cdot m$ (time units).

As a final example, we would like to count the number of `article`-entries with k authors where k is any natural number. This can be achieved by interpreting the following sentence of $\mathrm{srMSO}(\mathbb{N}, \Sigma)$:

$$\texttt{number-k-authored?} = \exists x.\,\mathrm{label}_{\texttt{mandatory}}(x) \wedge \texttt{has-k-authors}(x)$$

$$\texttt{has-k-authors}(x)$$

$$= \exists y_1.\ldots\exists y_k.\,\mathrm{firstChild}(x, y_1) \wedge \bigwedge\nolimits_{1 \le i \le k-1} (\mathrm{label}_{\texttt{author}}(y_i) \wedge \mathrm{next}(y_i, y_{i+1})) \\ \wedge\, \mathrm{label}_{\texttt{author}}(y_k) \wedge \big(\forall z.\mathrm{next}(y_k, z) \xrightarrow{+} \mathrm{label}_{\texttt{title}}(z)\big).$$

9 Discussion and Open Problems

In [14], the authors considered also several classes of syntactically defined sentences properly containing the class of syntactically restricted sentences, and they showed that their expressive power is still captured by weighted finite automata provided that the semiring satisfies suitable local finiteness conditions. We note that these results for words also transfer almost verbatim to the present setting of unranked trees, with analogous proofs.

From our results the following open problems arise:

1. Find an extension of the syntactically restricted $\mathrm{MSO}(\Sigma, S)$-logic which is still syntactically definable and expressively equivalent to the class of wta.

2. Find a subclass of wta which is expressively equivalent to the syntactically restricted $\mathrm{MSO}(S, \Sigma)$-logic.

3. Are there crucially different results if we replace our depth-first left-to-right order on trees by another linear order?

Acknowledgements We thank the anonymous referees whose careful suggestions have resulted in improvements of this paper.

References

1. Abiteboul, S., Bunemann, P., Suciu, D.: Data on the Web. Morgan Kaufmann, San Francisco (2000)
2. Alexandrakis, A., Bozapalidis, S.: Weighted grammars and Kleene's theorem. Inf. Process. Lett. **24**(1), 1–4 (1987)
3. Berstel, J., Reutenauer, C.: Recognizable formal power series on trees. Theor. Comput. Sci. **18**(2), 115–148 (1982)
4. Berstel, J., Reutenauer, C.: Rational Series and Their Languages. EATCS-Monographs, vol. 12. Springer, Berlin (1988)
5. Borchardt, B.: The Theory of Recognizable Tree Series. Ph.D. thesis, TU Dresden, Germany, Verlag für Wissenschaft und Forschung (2004)
6. Bray, T., Paoli, J., Sperberg-McQueen, C.M.: Extensible Markup Language (XML 1.0 W3C Recommendation). http://www.w3.org/TR/REC-xml/ (1998)
7. Brüggemann-Klein, A., Murata, M., Wood, D.: Regular tree and regular hedge languages over unranked alphabets: Version 1, April 2001. Technical Report HKUST-TCSC-2001-0, The Hongkong University of Science and Technology (2001)
8. Brüggemann-Klein, A., Wood, D.: Regular tree languages over non-ranked alphabets. Available at http://citeseer.ist.psu.edu/br98regular.html (1998)
9. Büchi, J.R.: Weak second-order arithmetic and finite automata. Zeitschr. Math. Logik Grundl. Math. **6**, 66–92 (1960)
10. Comon, H., Dauchet, M., Gilleron, R., Jacquemard, F., Lugiez, D., Tison, S., Tommasi, M.: Tree automata techniques and applications. Available on: http://www.grappa.univ-lille3.fr/tata (1997)
11. Doner, J.: Tree acceptors and some of their applications. J. Comput. Syst. Sci. **4**, 406–451 (1970)
12. Droste, M., Gastin, P.: Weighted automata and weighted logics. In: Automata, Languages and Programming—32nd International Colloquium, ICALP 2005, Lisbon, Portugal. Lecture Notes in Comput. Sci., vol. 3580, pp. 513–525. Springer, Berlin (2005)
13. Droste, M., Gastin, P.: Weighted automata and weighted logics. Theor. Comput. Sci. **380**(1–2), 69–86 (2007)
14. Droste, M., Gastin, P.: Weighted automata and weighted logics. In: Droste, M., Kuich, W., Vogler, H. (eds.) Handbook of Weighted Automata. Springer, Berlin (2009). Chap. 5. To appear
15. Droste, M., Rahonis, G.: Weighted automata and weighted logics with discounting. In: Holub, J., Zdárek, J. (eds.) Proc. of: Implementation and Application of Automata, 12th CIAA, Prague. Lecture Notes in Comp. Sci., vol. 4783, pp. 73–84. Springer, Berlin (2007)
16. Droste, M., Vogler, H.: Weighted tree automata and weighted logics. Theor. Comput. Sci. **366**, 228–247 (2006)
17. Droste, M., Kuich, W., Vogler, H. (eds.): Handbook of Weighted Automata. Springer, Berlin (2009). To appear
18. Eilenberg, S.: Automata, Languages, and Machines—Volume A. Pure and Applied Mathematics, vol. 59. Academic Press, New York (1974)
19. Elgot, C.C.: Decision problems of finite automata design and related arithmetics. Trans. Am. Math. Soc. **98**, 21–52 (1961)
20. Fichtner, I.: Weighted picture automata and weighted logics. Theory Comput. Syst. (2009). To appear
21. Fülöp, Z., Vogler, H.: Weighted tree automata and tree transducers. In: Droste, M., Kuich, W., Vogler, H. (eds.) Handbook of Weighted Automata. Springer, Berlin (2009). Chap. 9. To appear
22. Gécseg, F., Steinby, M.: Tree Automata. Springer, Berlin (1984). Akadémiai Kiadó, Budapest
23. Gécseg, F., Steinby, M.: Tree languages. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, vol. 3, pp. 1–68. Springer, Berlin (1997). Chap. 1
24. Klarlund, N., Schwentick, Th., Suciu, D.: XML: models, schemas, types, logics, and queries. In: Logics for Emerging Applications on Databases, pp. 1–41. Springer, Berlin (2003)
25. Kuich, W.: Semirings and formal power series: their relevance to formal languages and automata. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, vol. 1, pp. 609–677. Springer, Berlin (1997). Chap. 9

26. Kuich, W.: Formal power series over trees. In: Bozapalidis, S. (ed.) 3rd International Conference on Developments in Language Theory, DLT 1997, Thessaloniki, Greece, Proceedings, pp. 61–101. Aristotle University of Thessaloniki (1998)
27. Kuich, W., Salomaa, A.: Semirings, Automata, Languages. Monogr. Theoret. Comput. Sci. EATCS Ser., vol. 5. Springer, Berlin (1986)
28. Libkin, L.: Logics for unranked trees: an overview. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) Automata, Languages and Programming: 32nd International Colloquium, ICALP 2005, Lisbon, Portugal. Lecture Notes in Comput. Sci., vol. 3580, pp. 35–50. Springer, Berlin (2005)
29. Libkin, L.: Logics for unranked trees: an overview. Log. Methods Comput. Sci. **2**(3:2), 1–31 (2006)
30. Mathissen, C.: Definable transductions and weighted logics for texts. In: Proc. of the 11th Int. Conf. on Developments in Language Theory (DLT), Turku. Lecture Notes in Comput. Sci., vol. 4588, pp. 324–336. Springer, Berlin (2007)
31. Mathissen, C.: Weighted logics for nested words and algebraic formal power series. In: Proc. of the 35th Int. Colloquium on Automata, Languages and Programming (ICALP), Reykjavik. Lecture Notes in Comput. Sci., vol. 5126, pp. 221–232. Springer, Berlin (2008)
32. Meinecke, I.: Weighted logics for traces. In: Proc. of: Computer Science—Theory and Applications, 1st CSR, St. Petersburg. Lecture Notes in Computer Science, vol. 3967, pp. 235–246. Springer, Berlin (2006)
33. Neven, F.: Design and Analysis of Query Languages for Structured Documents. PhD thesis, University of Limburg (1999)
34. Neven, F.: Automata, logic, and XML. In: Bradfield, J. (ed.) Computer Science Logic: 16th International Workshop, CSL 2002, Edinburgh, Scotland, UK. Lecture Notes in Comput. Sci., vol. 2471, pp. 2–26. Springer, Berlin (2002)
35. Neven, F., Schwentick, Th.: Query automata over finite trees. Theor. Comput. Sci. **275**, 633–674 (2002)
36. Rahonis, G.: Weighted Muller tree automata and weighted logics. J. Autom. Lang. Program. **12**, 455–483 (2007)
37. Salomaa, A., Soittola, M.: Automata-Theoretic Aspects of Formal Power Series. Texts and Monographs in Computer Science. Springer, Berlin (1978)
38. Seidl, H., Schwentick, T., Muscholl, A.: Numerical document queries. In: IEEE Symposium on Principles of Database Systems, pp. 155–166. ACM Press, New York (2003)
39. Seidl, H., Schwentick, T., Muscholl, A., Habermehl, P.: Counting in trees for free. In: Proc. 31st International Colloq. on Automata, Languages, and Programming, ICALP. Lecture Notes in Comput. Sci., vol. 3142, pp. 1136–1149. Springer, Berlin (2004)
40. Schützenberger, M.P.: On the definition of a family of automata. Inf. Control **4**, 245–270 (1961)
41. Thatcher, J.W.: Characterizing derivation trees of context-free grammars through a generalization of finite automata theory. J. Comput. Syst. Sci. **4**, 317–322 (1967)
42. Thatcher, J.W., Wright, J.B.: Generalized finite automata theory with an application to a decision problem of second-order logic. Math. Syst. Theory **2**(1), 57–81 (1968)