Weighted Unranked Tree Automata over Tree Valuation Monoids and Their Characterization by Weighted Logics

Manfred Droste[1], Doreen Heusel[1], and Heiko Vogler[2](✉)

[1] Institut für Informatik, Universität Leipzig, D-04109 Leipzig, Germany
{droste,dheusel}@informatik.uni-leipzig.de
[2] Institut für Theoretische Informatik,
Technische Universität Dresden, D-01062 Dresden, Germany
Heiko.Vogler@tu-dresden.de

Abstract. We introduce a new behavior of weighted unranked tree automata. We prove a characterization of this behavior by two fragments of weighted MSO logic and thereby provide a solution of an open equivalence problem of Droste and Vogler. The characterization works for valuation monoids as weight structures; they include all semirings and, in addition, enable us to cope with average.

1 Introduction

In 1967, Thatcher investigated the theory of pseudoterms (nowadays known as unranked trees) and pseudoautomata (or unranked tree automata), see [30]. Since then, this theory has been further developed, cf. e.g. [2,3,22,26,27] and Chapter 8 of [8], due to the development of the modern document language XML and the fact that (fully structured) XML-documents can be formalized as unranked trees. An automaton model for unranked trees with ordered data values was investigated in [29], and important closure properties of symbolic unranked tree transducers were given in [32,19]. In [15,21], weighted automata on unranked trees over semirings were investigated in order to be able to deal with quantitative queries. For further background on weighted tree automata we refer to [11,18].

Weighted logics over semirings represent another approach for the investigation of quantitative aspects. For words, a weighted MSO logic which is expressively equivalent to weighted word automata was developed in [9]. Several analogous formalisms followed for infinite words [13], ranked trees [14], infinite trees [28], trace languages [25], picture languages [17], texts [23], and nested words [24].

In [15] a logic counterpart for weighted unranked tree automata over semirings was established. More precisely, each unranked tree series which is definable in syntactically restricted MSO logic is recognizable [15, Thm.6.5], and

D. Heusel—Partially supported by DFG Graduiertenkolleg 1763 (QuantLA).

A. Maletti (Ed.): CAI 2015, LNCS 9270, pp. 90–102, 2015.
DOI: 10.1007/978-3-319-23021-4_9

every recognizable unranked tree series is MSO-definable [15, Thm.5.9] and, if the semiring is commutative, even syntactically restricted MSO-definable. But surprisingly, there is a recognizable unranked tree series over a non-commutative semiring which is not definable in syntactically restricted MSO logic. In [15] it is stated as an open problem to determine a weighted automata model expressively equivalent to syntactically restricted MSO logic. One goal of our paper is to solve this problem.

For this, we present a new class of weighted unranked tree automata. Syntactically they do not differ from the ones of [15]. They still consist of a state set and a family of weighted word automata. The latter are used to calculate the local weight at a position of a tree by letting the weighted word automaton run on the states at the children of the position. However, we will define the semantics (or: behavior) of weighted unranked tree automata in a different way. We do not use runs anymore, but we choose the technically more involved extended runs, which were already introduced in [15]. Additionally to the information of classical runs, extended runs also include runs of the weighted word automata called at positions of the input tree. In addition we change the way how the weight of such an extended run is calculated. In [15], the local weight of a position was defined by the weight of the run chosen for the word emerged of its children's labels. Here the local weight of a position equals the weight of the transition taken for this position in the run of the position's parent.

In this paper we consider tree valuation monoids as weight structures which were defined in [10] (cf. [4,5,6,7,12]). Tree valuation monoids are additive monoids equipped with a valuation function that assigns a value of this monoid to any tree with labels from the additive monoid. We will use the valuation function to calculate the weights of an extended run in a global way, i.e. given a run we apply the valuation function to all local weights which appear along the extended run. Tree valuation monoids are very general: each semiring, and each bounded (possibly non-distributive) lattice [20] is a tree valuation monoid. In addition, these structures enable us to cope with non-binary valuation functions like average or discounting. Thus our weighted unranked tree automata subsume the weighted unranked tree automata over commutative semirings of [15] and the weighted ranked tree automata over tree valuation monoids [10].

The main results of this paper are the following. We define a weighted MSO logic for unranked trees over product tree valuation monoids analogously to [12] and characterize the behavior of our weighted unranked tree automata by two different fragments of the logic, see Theorem 5.1. Thereby we solve the open equivalence problem of [15] in Corollary 5.7, and generalize the respective results of [15] about weighted unranked tree automata over commutative semirings and the respective results of [10].

2 Unranked Trees and (Product) Tree Valuation Monoids

Let $\mathbb{N} = \{1, 2, \ldots\}$ be the set of all natural numbers and $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. For a set X, the set X^* comprises all finite words over X. If $X_1, \ldots, X_n$ are sets and $x \in X_1 \times \ldots \times X_n$, then x_i equals the i-th component of x.

We will base unranked trees on tree domains. A *tree domain* $\mathcal{B}$ is a finite, non-empty subset of $\mathbb{N}^*$ such that for all $u \in \mathbb{N}^*$ and $i \in \mathbb{N}$, $u.i \in \mathcal{B}$ implies u, $u.1, \ldots, u.(i-1) \in \mathcal{B}$. An *unranked tree* over a set X (of labels) is a mapping $t : \mathcal{B} \to X$ such that $\mathrm{dom}(t) = \mathcal{B}$ is a tree domain. The elements of $\mathrm{dom}(t)$ are called *positions* of t and $t(u)$ is called *label* of t at $u \in \mathrm{dom}(t)$. We call $u \in \mathrm{dom}(t)$ a *leaf* of t if there is no i such that $u.i \in \mathrm{dom}(t)$. The *set of all leaves* of t is denoted by $\mathrm{dom}_{\mathrm{leaf}}(t)$. With $\mathrm{rk}_t(u) = \max\{i \in \mathbb{N} \mid u.i \in \mathrm{dom}(t)\}$ we denote the *rank* of position u. The image of t is $\mathrm{im}(t) = \{t(u) \mid u \in \mathrm{dom}(t)\}$. We denote the set of all unranked trees over X by U_X. A *tree language* is a subset of U_X. We view each $d \in X$ as unranked tree in U_X, also denoted by d, whose tree domain only consists of the position ε which is labeled by d.

Now we recall the notion of tree valuation monoids and product tree valuation monoids as defined in [10,12]. A *tree valuation monoid* (*tv-monoid* for short) is a quadruple $\mathbb{D} = (D, +, \mathrm{Val}, \mathbb{0})$ such that $(D, +, \mathbb{0})$ is a commutative monoid and $\mathrm{Val} : U_D \to D$ is a function, called *(tree) valuation function*, with $\mathrm{Val}(d) = d$ for every tree $d \in D$ and $\mathrm{Val}(t) = \mathbb{0}$ whenever $\mathbb{0} \in \mathrm{im}(t)$ for $t \in U_D$. A *product tree valuation monoid* (*ptv-monoid* for short) is a sextuple $\mathbb{D} = (D, +, \mathrm{Val}, \diamond, \mathbb{0}, \mathbb{1})$ which consists of a tv-monoid $(D, +, \mathrm{Val}, \mathbb{0})$, a constant $\mathbb{1} \in D$ with $\mathrm{Val}(t) = \mathbb{1}$ whenever $\mathrm{im}(t) = \{\mathbb{1}\}$ for $t \in U_D$, and an operation $\diamond \colon D^2 \to D$ with $\mathbb{0} \diamond d = d \diamond \mathbb{0} = \mathbb{0}$ and $\mathbb{1} \diamond d = d \diamond \mathbb{1} = d$ for all $d \in D$.

Example 2.1. $\mathbb{Q}_{\max} = (\mathbb{Q} \cup \{-\infty\}, \max, \mathrm{avg}, -\infty)$ with $\mathrm{avg}(t) = \frac{\sum_{u \in \mathrm{dom}(t)} t(u)}{|\,\mathrm{dom}(t)|}$ for all $t \in U_{\mathbb{Q} \cup \{-\infty\}}$ is a tv-monoid. The valuation function of this tv-monoid calculates the average of all weights of a tree. The idea for the average calculation was already suggested in [4,12] for words and in [10] for trees. From $\mathbb{Q}_{\max}$ we can obtain a ptv-monoid $\mathbb{Q}^p_{\max}$ by adding ∞ to the carrier set and setting $\diamond = \min$. We refer to [10] for further examples of (p)tv-monoids.

For the rest of this paper, let Σ be an alphabet, i.e., a finite, non-empty set, and $\mathbb{D}$ be a ptv-monoid.

3 Weighted Unranked Tree Automata

Here we introduce a new class of recognizable tree series. A tree series is recognizable if it can be recognized by a (classical) weighted unranked tree automaton over some tree valuation monoid using extended runs [15] for the definition of behavior. In the case of semirings, the semantics of a weighted unranked tree automaton based on runs and the semantics of this automaton based on extended runs are equivalent, cf. [15, Obs.6.8]. But for non-distributive structures, which are also considered here, this is not necessarily true. Besides, we define the weight of an extended run in a new way which is different from [15]. This will enable us to describe the behavior of weighted unranked tree automata by restricted weighted MSO formulas (see proof of Theorem 5.1).

A *weighted string automaton* (*WSA* for short) over Σ and $\mathbb{D}$ is a quadruple $\mathcal{A} = (P, I, \mu, F)$ where P is a non-empty, finite set of states, $I, F \subseteq P$ are the sets of initial and final states, respectively, and $\mu \colon P \times \Sigma \times P \to \mathbb{D}$.

A *run* of $\mathcal{A}$ on $w = w_1 \dots w_n$ with $w_1, \dots, w_n \in \Sigma$ and $n \geq 0$ is a sequence $\pi = (p_{i-1}, w_i, p_i)_{1 \leq i \leq n}$ if $n > 0$, and a state $\pi = p_0$ if $n = 0$ where $p_0, \dots, p_n \in P$. The run π is *successful* if $p_0 \in I$ and $p_n \in F$. In order to define the weight $\mathrm{wt}(\pi)$ of π using a tree valuation function Val, we define a tree t_π by letting $\mathrm{dom}(t_\pi) = \{1^i \mid 0 \leq i < n\}$ and $t_\pi(1^i) = \mu(p_{i-1}, w_i, p_i)$ $(0 \leq i < n)$ if $n > 0$, and $t_\pi(\varepsilon) = \mathbb{0}$ if $n = 0$. Then let $\mathrm{wt}(\pi) = \mathrm{Val}(t_\pi)$. The *behavior* of $\mathcal{A}$ is the function $\|\mathcal{A}\| : \Sigma^* \to \mathbb{D}$ with $\|\mathcal{A}\|(w) = \sum_{\pi \text{ successful run on } w} \mathrm{wt}(\pi)$ for $w \in \Sigma^*$. We call any mapping from Σ^* to $\mathbb{D}$ a *string series*. A string series S is called *recognizable* over $\mathbb{D}$ if there is a WSA $\mathcal{A}$ over Σ and $\mathbb{D}$ with $\|\mathcal{A}\| = S$.

A *weighted unranked tree automaton* (*WUTA* for short) over Σ and $\mathbb{D}$ is a triple $\mathcal{M} = (Q, \mathcal{A}, \gamma)$ where Q is a non-empty, finite set of states, $\mathcal{A} = (\mathcal{A}_{q,a} \mid q \in Q, a \in \Sigma)$ is a family of WSA over Q as alphabet and $\mathbb{D}$, and $\gamma \colon Q \to \mathbb{D}$ is a *root weight function*. Let $\mathcal{A}_{q,a} = (P_{q,a}, I_{q,a}, \mu_{q,a}, F_{q,a})$ for all $q \in Q$, $a \in \Sigma$. We assume the sets $P_{q,a}$ to be pairwise disjoint and let $P_\mathcal{A} = \bigcup_{q \in Q, a \in \Sigma} P_{q,a}$. Moreover, let $\mu_\mathcal{A}$ be the union of the transition functions $\mu_{q,a}$.

Intuitively, an extended run assigns a state $q \in Q$ to each position u of a given tree $t \in U_\Sigma$ and then consists of one run of $\mathcal{A}_{q,t(u)}$ on $q_1 \dots q_{\mathrm{rk}_t(u)}$ where q_i is the state assigned to the i-th child of u. Formally, an *extended run* of $\mathcal{M}$ on a tree t is a triple (q, s, l) such that

- $q \in Q$ is the *root state*;
- $s \colon \mathrm{dom}(t) \setminus \{\varepsilon\} \to P_\mathcal{A} \times Q \times P_\mathcal{A}$ is a function such that $s(1) \dots s(\mathrm{rk}_t(\varepsilon))$ is a run of $\mathcal{A}_{q,t(\varepsilon)}$ and $s(u.1) \dots s(u.\mathrm{rk}_t(u))$ is a run of $\mathcal{A}_{s(u)_2,t(u)}$ for every $u \in \mathrm{dom}(t) \setminus (\mathrm{dom}_{\mathrm{leaf}}(t) \cup \{\varepsilon\})$;
- $l \colon \mathrm{dom}_{\mathrm{leaf}}(t) \to P_\mathcal{A}$ is a function satisfying $l(\varepsilon) \in P_{q,t(\varepsilon)}$ if t only consists of the root, and if $u \neq \varepsilon$ is a leaf, then $l(u) \in P_{s(u)_2,t(u)}$.

An extended run is *successful* if $s(u.1) \dots s(u.\mathrm{rk}_t(u))$ is successful for all $u \in \mathrm{dom}(t) \setminus \mathrm{dom}_{\mathrm{leaf}}(t)$ and if $l(u)$ is successful for all $u \in \mathrm{dom}_{\mathrm{leaf}}(t)$ (i.e., $l(u)$ is an initial and final state of $\mathcal{A}_{s(u)_2,t(u)}$ if $u \neq \varepsilon$ respectively of $\mathcal{A}_{q,t(\varepsilon)}$ if $u = \varepsilon$). We let $\mathrm{succ}(\mathcal{M}, t)$ denote the set of all successful extended runs of $\mathcal{M}$ on t.

To define the weight of an extended run we proceed differently from [15] where the local weight of a position u was defined by the weight of the run chosen for the labels of the children of u. Here, we will define the local weight of u by the weight of the transition taken for u in the run of the parent of u. Each extended run (q, s, l) on t defines a tree $\mu(t, (q, s, l)) \in U_\mathbb{D}$ where $\mathrm{dom}(\mu(t, (q, s, l))) = \mathrm{dom}(t)$ and

$$\mu(t, (q, s, l))(u) = \begin{cases} \gamma(q) & \text{if } u = \varepsilon, \\ \mu_\mathcal{A}(s(u)) & \text{otherwise} \end{cases}$$

for all $u \in \mathrm{dom}(t)$. We call $\mu(t, (q, s, l))(u)$ the *local weight* of u and $\mathrm{Val}(\mu(t, (q, s, l)))$ the *weight of* (q, s, l) *on* t. The *behavior* of a WUTA $\mathcal{M}$ is the function $\|\mathcal{M}\| : U_\Sigma \to \mathbb{D}$ defined by

$$\|\mathcal{M}\|(t) = \sum_{(q,s,l) \in \mathrm{succ}(\mathcal{M},t)} \mathrm{Val}(\mu(t, (q, s, l)))$$

for all $t \in U_\Sigma$. Thus, if no successful extended run on t exists, we put $\|\mathcal{M}\|(t) = \mathbb{0}$.

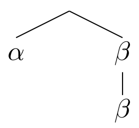
Any mapping from U_Σ to $\mathbb{D}$ is called a *tree series*. A tree series $S\colon U_\Sigma \to \mathbb{D}$ is called *recognizable* over $\mathbb{D}$ if there is a WUTA $\mathcal{M}$ over Σ and $\mathbb{D}$ with $\|\mathcal{M}\| = S$.

Example 3.1. Let $\mathbb{Q}_{\max}$ be the tv-monoid from Example 2.1. We will consider a WUTA $\mathcal{M}$ which calculates the leaves-to-size ratio of a given input tree, where the size of a tree is the number of all positions of the tree. Let $\mathcal{M} = (\{c, n\}, \mathcal{A}, \gamma)$ over an arbitrary, but fixed alphabet Σ with $\gamma(c) = 1$, $\gamma(n) = 0$, and

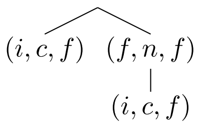
- $\mathcal{A}_{n,a} = (\{i, f\}, \{i\}, \mu_{n,a}, \{f\})$ where $\mu_{n,a}(i, n, f) = \mu_{n,a}(f, n, f) = 0$, $\mu_{n,a}(i, c, f) = \mu_{n,a}(f, c, f) = 1$ and $\mu_{n,a}(f, q, i) = \mu_{n,a}(i, q, i) = -\infty$
- $\mathcal{A}_{c,a} = (\{p\}, \{p\}, \mu_{c,a}, \{p\})$ where $\mu_{c,a}(p, q, p) = -\infty$

for all $q \in \{c, n\}$ and $a \in \Sigma$; for notational convenience, here we have dropped the condition on pairwise disjointness of the state sets.

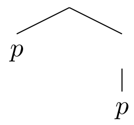
First, let us consider an example tree. For this, we choose $\Sigma = \{\alpha, \beta\}$ and tree $t_{ex} =$



. Then (n, s, l) with $s =$

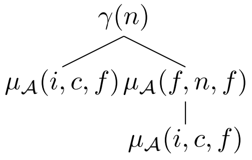


and $l =$

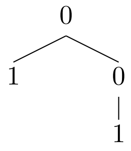


is an extended run on t_{ex}. Here an unlabeled position means that it is not in the domain of the represented function. Obviously (n, s, l) is successful, since the runs $s(1)s(2) = (i, c, f)(f, n, f)$ and $s(2.1) = (i, c, f)$ are successful in $\mathcal{A}_{n,\alpha}$ and $\mathcal{A}_{n,\beta}$, respectively, and the run p is successful in $\mathcal{A}_{c,\alpha}$ as well as in $\mathcal{A}_{c,\beta}$. The local weights of (n, s, l) are

$\mu(t_{ex}, (n, s, l)) =$


$\gamma(n)$
$\mu_{\mathcal{A}}(i, c, f)$
$\mu_{\mathcal{A}}(f, n, f)$
$\mu_{\mathcal{A}}(i, c, f)$

$=$



and thus the weight of (n, s, l) equals $\frac{1}{2}$.

Now let t be an arbitrary, but fixed tree. It is easy to see that for every successful extended run (q, s, l) on t, $l(u) = p$ for every leaf u of t. Assume that in addition (q, s, l) assigns the state n to each inner position of t. Let π_u be the unique run of $\mathcal{A}_{n,t(u)}$ for which t_{π_u} has no label equal to $-\infty$, thus, π_u leads directly from i to f and finally loops in f. If (q, s, l) consists for every inner position $u \neq \varepsilon$ of π_u, then (q, s, l) is the only successful extended run such that $\mu(t, (q, s, l))$ does not contain $-\infty$. Let π denote this unique extended run. For leaves u of t, $\mu(t, \pi)(u) = 1$ and for inner positions u', $\mu(t, \pi)(u') = 0$. Thus,

$$\|\mathcal{M}\|(t) = \operatorname{avg}(\mu(t, \pi)) = \frac{\sum_{u \in \operatorname{dom}(t)} \mu(t, \pi)(u)}{|\operatorname{dom}(t)|} = \frac{\text{“number of leaves of } t\text{”}}{\text{“size of } t\text{”}}\,.$$

Remark 3.2. The WUTA subsume the weighted ranked tree automata over tv-monoids of [10] as well as the weighted unranked tree automata over commutative semirings [15]. But there are tree series over non-commutative semirings which are recognizable by the weighted unranked tree automata of [15] but not by our WUTA. An example was given in the proof of [15, Thm.6.10].

Furthermore, it is easy to show that unranked tree automata over Σ [30,2,22,27] are equivalent to WUTA over Σ and the boolean semiring $\mathbb{B}$. Thus, for each WUTA over $\mathbb{B}$ there is an equivalent deterministic WUTA [30, Thm.1]. A WUTA over $\mathbb{B}$ is *deterministic* if for every $a \in \Sigma$ and $q_1, q_2 \in Q$, if $q_1 \neq q_2$, then there is no $w \in Q^*$ such that $\|\mathcal{A}_{q_1,a}\|(w) = \|\mathcal{A}_{q_2,a}\|(w) = \mathbb{1}$.

Next we will derive some properties of recognizable tree series. Let S_1, S_2 be two tree series and $d \in \mathbb{D}$. The *scalar product* $d \diamond S_1$, the *sum* $S_1 + S_2$ and the *(Hadamard) product* $S_1 \diamond S_2$ are defined pointwise by $(d \diamond S_1)(t) = d \diamond S_1(t)$, $(S_1 + S_2)(t) = S_1(t) + S_2(t)$ and $(S_1 \diamond S_2)(t) = S_1(t) \diamond S_2(t)$ for all $t \in U_\Sigma$. For a tree language $L \subseteq U_\Sigma$, the *characteristic function of* L, called $\mathbb{1}_L$, equals $\mathbb{1}$ for all $t \in L$ and $\mathbb{0}$ for all $t \in U_\Sigma \setminus L$. A tree series S is a *recognizable step function* if there are recognizable tree languages $L_1, \ldots, L_k$ forming a partition of U_Σ and values $d_1, \ldots, d_k \in \mathbb{D}$ such that $S = \sum_{i=1}^{k} d_i \diamond \mathbb{1}_{L_i}$.

Lemma 3.3. *([10], Lemma 5.9) The class of recognizable step functions over Σ and $\mathbb{D}$ is closed under the operations $+$ and the Hadamard product $\diamond$.*

The next theorem can be proved by applying standard automata constructions (assuming, in (2), the unranked tree automaton for L to be deterministic).

Theorem 3.4. *Let $\mathbb{D}$ be a ptv-monoid.*
1. *The class of recognizable tree series is closed under sum.*
2. *Let L be a recognizable tree language and S a recognizable tree series. Then $\mathbb{1}_L \diamond S$ (which equals $S \diamond \mathbb{1}_L$) is also recognizable.*

A ptv-monoid $\mathbb{D}$ is *regular* if for all $d \in \mathbb{D}$ and all alphabets Σ a WUTA $\mathcal{M}_d$ exists with $\|\mathcal{M}_d\|(t) = d$ for each $t \in U_\Sigma$. Using Theorem 3.4 one can easily show the following lemma.

Lemma 3.5. *Let $\mathbb{D}$ be a regular ptv-monoid. Each recognizable step function S over $\mathbb{D}$ is a recognizable tree series.*

Now we consider the closure under relabeling, similarly to [14,12]. Let Σ and Γ be two alphabets and $h : \Sigma \to 2^\Gamma$ be a mapping. Then h can be extended to a mapping $h' : U_\Sigma \to 2^{U_\Gamma}$ by letting $h'(t)$ be the set of all trees t' over Γ such that $\mathrm{dom}(t') = \mathrm{dom}(t)$ and $t'(u) \in h(t(u))$ for each position $u \in \mathrm{dom}(t)$. For every tree series S over $\mathbb{D}$ and Σ the tree series $h''(S)$ over $\mathbb{D}$ and Γ is defined by

$$h''(S)(t') = \sum_{t \in U_\Sigma \,\wedge\, t' \in h'(t)} S(t)$$

for all $t' \in U_\Gamma$. Clearly, the index set of the summation is finite. We denote h' and h'' also by h which we call a *relabeling*. The proof for the following lemma works by an automaton construction already applied in a similar way in [16,12].

Lemma 3.6. *Recognizable tree series are closed under relabeling.*

We will show that under suitable conditions the Hadamard product $\diamond$ preserves the recognizability of arbitrary tree series. For this, we recall some properties of ptv-monoids already defined in [10]. We call $\mathbb{D}$ *left-multiplicative* if $d \diamond \mathrm{Val}(t) = \mathrm{Val}(t')$ for all $d \in \mathbb{D}$, $t, t' \in U_{\mathbb{D}}$ with $\mathrm{dom}(t) = \mathrm{dom}(t')$, $t'(\varepsilon) = d \diamond t(\varepsilon)$, and $t'(u) = t(u)$ for every $u \in \mathrm{dom}(t) \setminus \{\varepsilon\}$. Furthermore, $\mathbb{D}$ is *left-Val-distributive* if $d \diamond \mathrm{Val}(t) = \mathrm{Val}(t')$ for all $d \in \mathbb{D}$, $t, t' \in U_D$ with $\mathrm{dom}(t) = \mathrm{dom}(t')$ and $t'(u) = d \diamond t(u)$ for every $u \in \mathrm{dom}(t)$. Two subsets $D_1, D_2 \subseteq \mathbb{D}$ *commute* if $d_1 \diamond d_2 = d_2 \diamond d_1$ for all $d_1 \in D_1$, $d_2 \in D_2$. We call $\mathbb{D}$ *conditionally commutative* if $\mathrm{Val}(t_1) \diamond \mathrm{Val}(t_2) = \mathrm{Val}(t)$ for all $t_1, t_2, t \in U_{\mathbb{D}}$ with $\mathrm{dom}(t_1) = \mathrm{dom}(t_2) = \mathrm{dom}(t)$, $\mathrm{im}(t_1)$ and $\mathrm{im}(t_2)$ commute and $t(u) = t_1(u) \diamond t_2(u)$ for all $u \in \mathrm{dom}(t)$. A ptv-monoid $\mathbb{D}$ is a *conditionally commutative tree valuation semiring (cctv-semiring)* if $(D, +, \diamond, \mathbb{0}, \mathbb{1})$ is a semiring and if $\mathbb{D}$ is conditionally commutative and, moreover, left-multiplicative or left-Val-distributive. For examples, we refer to [10].

Let $\mathcal{W}_{\mathcal{M}}$ comprises all the weights of automaton $\mathcal{M}$, i.e., all transition weights of any automaton $\mathcal{A}_{q,a}$ of $\mathcal{M}$ and all root weights of $\mathcal{M}$.

Theorem 3.7. *Let $\mathbb{D}$ be a cctv-semiring.*

1. *Let S_1 be a recognizable step function and S_2 a recognizable tree series. Then $S_1 \diamond S_2$ is also recognizable.*
2. *Let $\mathcal{M}_i = (Q_i, \mathcal{A}_i, \gamma_i)$ be a WUTA ($i \in \{1, 2\}$) such that $\mathcal{W}_{\mathcal{M}_1}$ and $\mathcal{W}_{\mathcal{M}_2}$ commute. Then $\|\mathcal{M}_1\| \diamond \|\mathcal{M}_2\|$ is recognizable.*

4 Weighted MSO Logic for Unranked Trees

We introduce a weighted MSO logic and its semantics for unranked trees over tv-monoids. As in [10], we follow [9] incorporating an idea of [1]. Let $\mathcal{V}_1$ and $\mathcal{V}_2$ be countable, infinite sets of first order and second order variables, respectively. The syntax of the weighted MSO logic over $\mathbb{D}$ is defined by the EBNF:

$$\begin{aligned}\beta &::= \mathrm{label}_a(x) \mid \mathrm{desc}(x, y) \mid x \leq y \mid x \sqsubseteq y \mid x \in X \mid \neg\beta \mid \beta \wedge \beta \mid \forall x.\beta \mid \forall X.\beta \\ \varphi &::= d \mid \beta \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x.\varphi \mid \forall x.\varphi \mid \exists X.\varphi\end{aligned}$$

where $d \in \mathbb{D}$, $a \in \Sigma$, $x, y \in \mathcal{V}_1$, and $X \in \mathcal{V}_2$. We call the formulas β *boolean formulas* and the formulas φ *weighted MSO formulas* (or *wMSO formulas*).

To define the semantics of the wMSO formulas, we follow the common approach for MSO logics using assignments and extended alphabets to deal with free variables, cf. [31]. The set $\mathrm{free}(\varphi)$ of free variables occurring in φ is defined as usual. A *sentence* is a formula without free variables. Let φ be a wMSO formula, $\mathcal{V}$ a finite set of variables with $\mathrm{free}(\varphi) \subseteq \mathcal{V}$, and $t \in U_\Sigma$. A $(\mathcal{V}, t)$*-assignment* is a mapping $\sigma : \mathcal{V} \to \mathrm{dom}(t) \cup 2^{\mathrm{dom}(t)}$ with $\sigma(x) \in \mathrm{dom}(t)$ for $x \in \mathcal{V}_1$ and $\sigma(X) \subseteq \mathrm{dom}(t)$ for $X \in \mathcal{V}_2$. As usual, we encode each $(\mathcal{V}, t)$-assignment by a tree over the extended alphabet $\Sigma_{\mathcal{V}} = \Sigma \times \{0, 1\}^{\mathcal{V}}$; we call a tree over $\Sigma_{\mathcal{V}}$ *valid* if it arises in this way. For details we refer to [9,15,10]. From now on we identify a pair (t, σ) and its encoding $s \in U_{\Sigma_{\mathcal{V}}}$. For $x \in \mathcal{V}_1$, the update $s[x \to u] \in U_{\Sigma_{\mathcal{V} \cup \{x\}}}$ for $u \in \mathrm{dom}(t)$ is defined by $s[x \to u] = (t, \sigma[x \to u]) = (t, \sigma')$ where

Table 1. The semantics of wMSO formulas

$$[\![\text{label}_a(x)]\!]_{\mathcal{V}}(s) = \begin{cases} \mathbb{1} & \text{if } t(\sigma(x)) = a, \\ \mathbb{0} & \text{otherwise} \end{cases} \qquad [\![\text{desc}(x,y)]\!]_{\mathcal{V}}(s) = \begin{cases} \mathbb{1} & \text{if } \exists i \in \mathbb{N} : \sigma(y) = \sigma(x).i, \\ \mathbb{0} & \text{otherwise} \end{cases}$$

$$[\![x \leq y]\!]_{\mathcal{V}}(s) = \begin{cases} \mathbb{1} & \text{if } \sigma(x) = \sigma(y) = \varepsilon \vee \exists u \in \text{dom}(s) : \exists i,j \in \mathbb{N}, i \leq j : \\ & \sigma(x) = u.i, \sigma(y) = u.j, \\ \mathbb{0} & \text{otherwise} \end{cases}$$

$$[\![x \sqsubseteq y]\!]_{\mathcal{V}}(s) = \begin{cases} \mathbb{1} & \text{if } \sigma(x) \sqsubseteq_s \sigma(y), \\ \mathbb{0} & \text{otherwise} \end{cases} \qquad [\![x \in X]\!]_{\mathcal{V}}(s) = \begin{cases} \mathbb{1} & \text{if } \sigma(x) \in \sigma(X), \\ \mathbb{0} & \text{otherwise} \end{cases}$$

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

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

$$[\![\exists x.\varphi]\!]_{\mathcal{V}}(s) = \sum_{u \in \text{dom}(s)} [\![\varphi]\!]_{\mathcal{V}\cup\{x\}}(s[x \to u]) \quad [\![\exists X.\varphi]\!]_{\mathcal{V}}(s) = \sum_{I \subseteq \text{dom}(s)} [\![\varphi]\!]_{\mathcal{V}\cup\{X\}}(s[X \to I])$$

$$[\![\forall X.\beta]\!]_{\mathcal{V}}(s) = \begin{cases} \mathbb{1} & \text{if } [\![\beta]\!]_{\mathcal{V}\cup\{X\}}(s[X \to I]) = \mathbb{1} \text{ for all } I \subseteq \text{dom}(s), \\ \mathbb{0} & \text{otherwise} \end{cases}$$

$$[\![\forall x.\varphi]\!]_{\mathcal{V}}(s) = \text{Val}(s_D) \text{ for } s_D \in U_D \text{ given by } \text{dom}(s_D) = \text{dom}(s) \text{ and}$$
$$s_D(u) = [\![\varphi]\!]_{\mathcal{V}\cup\{x\}}(s[x \to u]) \text{ for all } u \in \text{dom}(s)$$

$\sigma'|_{\mathcal{V}\setminus\{x\}} = \sigma|_{\mathcal{V}\setminus\{x\}}$ and $\sigma'(x) = u$. The update $s[X \to I] \in U_{\Sigma_{\mathcal{V}\cup\{X\}}}$ for $X \in \mathcal{V}_2$ and $I \subseteq \text{dom}(t)$ is defined similarly.

The *semantics of a wMSO formula* φ over a ptv-monoid $\mathbb{D}$ and an alphabet Σ is the tree series $[\![\varphi]\!]_{\mathcal{V}} : U_{\Sigma_{\mathcal{V}}} \to D$ which equals $\mathbb{0}$ for non-valid trees and which is defined inductively for each valid tree $s = (t, \sigma)$ as shown in Table 1. Here $\sqsubseteq_s$ is a linear ordering on the positions of s. For the rest of this paper this linear ordering will be the depth-first left-to-right traversal. Then the formula $x \leq y$ can be expressed with the help of $x \sqsubseteq y$. Subsequently, we write $[\![\varphi]\!]$ for $[\![\varphi]\!]_{\text{free}(\varphi)}$. Any boolean wMSO formula β can be viewed as a classical MSO formula which defines the recognizable tree language $L_{\mathcal{V}}(\beta)$ and we can easily show that $[\![\beta]\!]_{\mathcal{V}} = \mathbb{1}_{L_{\mathcal{V}}(\beta)}$. Furthermore, we can prove by induction that $[\![\varphi]\!]_{\mathcal{V}}(t,\sigma) = [\![\varphi]\!](t, \sigma|_{\text{free}(\varphi)})$ for every wMSO formula φ, $(t,\sigma) \in U_{\Sigma_{\mathcal{V}}}$, and set of variables $\mathcal{V}$ with $\text{free}(\varphi) \subseteq \mathcal{V}$.

Example 4.1. Let $\mathbb{Q}^p_{\max}$ be the ptv-monoid from Example 2.1. The boolean formula $\text{leaf}(x) = \forall y.\neg\,\text{desc}(x,y)$ maps every $t \in U_\Sigma$ and assignment σ to ∞ if $\sigma(x)$ is a leaf and to $-\infty$ if $\sigma(x)$ is not a leaf. Analogously to [10], we can show that the formula $\varphi = \forall x.((\text{leaf}(x) \wedge 1) \vee (\neg\text{leaf}(x) \wedge 0))$ defines the leaves-to-size ratio for trees which was previously computed by the WUTA of Example 3.1.

Next we introduce some fragments of the weighted MSO logic which will be essential for our main result. A wMSO formula is an *almost boolean formula* if it consists only of conjunctions and disjunctions of boolean formulas and elements of D. We call a wMSO formula $\forall$-*restricted* if all its subformulas $\forall x.\varphi$ satisfy that

φ is almost boolean. Let $\text{const}(\varphi)$ be the set of all $d \in D$ occurring in a formula φ. Similarly to [12,10] we call φ *strongly* $\wedge$*-restricted* if whenever φ contains a subformula $\varphi_1 \wedge \varphi_2$, then both φ_1 and φ_2 are almost boolean or φ_1 or φ_2 is boolean; and *commutatively* $\wedge$*-restricted* if whenever φ contains a subformula $\varphi_1 \wedge \varphi_2$, then φ_1 is almost boolean or $\text{const}(\varphi_1)$ and $\text{const}(\varphi_2)$ commute. Note that each strongly $\wedge$-restricted wMSO formula is commutatively $\wedge$-restricted. For examples of weighted logic formulas and a discussion on the above restrictions, we refer the reader to [9,14,15,12].

5 Weighted Tree Automata and Weighted MSO Logic

Here we characterize the class of behaviors of WUTA by the fragments of the weighted MSO logic.

Theorem 5.1. *Let $S : U_\Sigma \to \mathbb{D}$ be a tree series.*
1. *If $\mathbb{D}$ is regular, then S is recognizable iff $S = [\![\varphi]\!]$ for some $\forall$-restricted and strongly $\wedge$-restricted wMSO sentence φ.*
2. *If $\mathbb{D}$ is a cctv-semiring, then S is recognizable iff $S = [\![\varphi]\!]$ for some $\forall$-restricted and commutatively $\wedge$-restricted wMSO sentence φ.*

For ranked trees, examples were given in [10] showing that it is not possible to drop the constraints on $\mathbb{D}$ in statements (1) or (2). These examples could be easily extended to the unranked tree setting. It remains to prove Theorem 5.1. For this, the following proposition will be very useful; it can be proved as the corresponding result in [14] by using Theorem 3.4(2).

Proposition 5.2. *Let φ be a wMSO formula and $\mathcal{V}$ a finite set of variables with $\text{free}(\varphi) \subseteq \mathcal{V}$. Then $[\![\varphi]\!]$ is recognizable iff $[\![\varphi]\!]_\mathcal{V}$ is recognizable, and $[\![\varphi]\!]$ is a recognizable step function iff $[\![\varphi]\!]_\mathcal{V}$ is a recognizable step function.*

Analogously to [10] one can show:

Lemma 5.3. *If φ is an almost boolean formula, then $[\![\varphi]\!]$ is a recognizable step function. Conversely, if $S : U_\Sigma \to \mathbb{D}$ is a recognizable step function, then $S = [\![\varphi]\!]$ for some almost boolean sentence φ.*

Now we can show that our logic operators preserve the recognizability of the semantics of wMSO formulas by adapting the proofs for the corresponding Propositions 5.15-5.17 of [10].

Proposition 5.4. *Let φ and ψ be wMSO formulas over Σ and $\mathbb{D}$. If $[\![\varphi]\!]$ and $[\![\psi]\!]$ are recognizable, then $[\![\varphi \vee \psi]\!]$, $[\![\exists x.\varphi]\!]$, and $[\![\exists X.\varphi]\!]$ are recognizable. Furthermore, $[\![\varphi \wedge \psi]\!]$ and $[\![\psi \wedge \varphi]\!]$ are recognizable if $[\![\varphi]\!]$ is recognizable and ψ is boolean.*

Proposition 5.5. *Let φ be an almost boolean formula over $\mathbb{D}$ and Σ. Then $[\![\forall x.\varphi]\!]$ is recognizable.*

Proof. Let $\mathcal{W} = \text{free}(\varphi) \cup \{x\}$ and $\mathcal{V} = \text{free}(\forall x.\varphi) = \mathcal{W} \setminus \{x\}$. Since φ is almost boolean and by Lemma 5.3, $[\![\varphi]\!]_{\mathcal{W}} = \sum_{i=1}^{n} d_i \diamond \mathbb{1}_{L_i}$ for some partition $L_1, \ldots, L_n$ of all valid trees over $\Sigma_{\mathcal{W}}$ (for invalid trees s, $[\![\varphi]\!]_{\mathcal{W}}(s) = \mathbb{0}$). Let $\tilde{\Sigma} = \Sigma \times \{1, \ldots, n\}$. We extend every valid tree $(t, \sigma) \in U_{\Sigma_{\mathcal{W}}}$ to a tree (t, ν, σ) over $\tilde{\Sigma}_{\mathcal{V}}$ by the unique mapping $\nu : \text{dom}(t) \to \{1, \ldots, n\}$ that encodes to which L_i the update of (t, σ) and x belongs. Hence, $\nu(u) = i$ iff $(t, \sigma[x \to u]) \in L_i$ for all $u \in \text{dom}(t)$. Let $\tilde{L} \subseteq U_{\tilde{\Sigma}_{\mathcal{V}}}$ be the tree language of all such trees (t, ν, σ). In [15] it was already shown that $\tilde{L}$ is recognizable. Let $\mathcal{M} = (Q, \mathcal{B}, F)$ over $\tilde{\Sigma}_{\mathcal{V}}$ be a deterministic unranked tree automaton that recognizes $\tilde{L}$. We may assume that every subautomaton $\mathcal{B}_{q,\tilde{a}} = (Q_{q,\tilde{a}}, I_{q,\tilde{a}}, T_{q,\tilde{a}}, F_{q,\tilde{a}})$ (for $q \in Q$, $\tilde{a} \in \tilde{\Sigma}_{\mathcal{V}}$) of $\mathcal{M}$ is deterministic. Thus for every tree $\tilde{t} \in U_{\tilde{\Sigma}_{\mathcal{V}}}$ there is exactly one extended run π of $\mathcal{M}$ on $\tilde{t}$, and in addition there is exactly one run π_u of $\mathcal{B}_{\pi(u),\tilde{t}(u)}$ on $\pi(u.1) \ldots \pi(u.\text{rk}_{\tilde{t}}(u))$ for each $u \in \text{dom}(\tilde{t})$.

We wish to transform $\mathcal{M}$ into a WUTA $\mathcal{M}'$ over $\tilde{\Sigma}_{\mathcal{V}}$ such that for every tree $\tilde{t}$ the unique runs π and π_u ($u \in \text{dom}(\tilde{t})$) form an extended run $\tilde{\pi} = (q, s, l)$ with

$$\mu(\tilde{t}, \tilde{\pi})(u) = d_i \Leftrightarrow \|\mathcal{B}_{s(u)_2,\tilde{t}(u)}\|(s(u.1)_2 \ldots s(u.\text{rk}_{\tilde{t}}(u))_2) = \mathbb{1} \text{ and } \tilde{t}(u)_2 = i$$

for all $u \in \text{dom}(\tilde{t})$. Then $\mu(\tilde{t}, \tilde{\pi})(u) = [\![\varphi]\!]_{\mathcal{W}}(t, \sigma[x \to u])$ and $\text{Val}(\mu(\tilde{t}, \tilde{\pi})) = [\![\forall x \varphi]\!]_{\mathcal{V}}(t, \sigma)$ where $(t, \nu, \sigma) = \tilde{t}$. All other extended runs on $\tilde{t}$ shall get the weight $\mathbb{0}$. For this, we extend the states of $\mathcal{M}$ by values from $\{1, \ldots, n\}$. The value in the state assigned to a position u encodes $\tilde{t}(u)_2$. We let $\mathcal{A}_{(q,j),(a,i,f)}$ be a WUTA with an empty set of final states whenever $j \neq i$ to ensure that for a successful extended run a state with value i is assigned to a position with label (a, i, f). The automaton $\mathcal{A}_{(q,i),(a,i,f)}$ will be a modified version of $\mathcal{B}_{q,(a,i,f)}$; it is defined over the alphabet $Q \times \{1, \ldots, n\}$ such that there is a transition $(p_1, (q, i'), p_2)$ with weight $d_{i'}$ for every $i' \in \{1, \ldots, n\}$ iff $T_{q,(a,i,f)}(p_1, q, p_2) = \mathbb{1}$. Formally, $\mathcal{M}' = (Q', \mathcal{A}, \gamma)$ such that $Q' = Q \times \{1, \ldots, n\}$, $\gamma(q, i) = d_i$ if $F(q) = \mathbb{1}$ and $\gamma(q, i) = \mathbb{0}$ if $F(q) = \mathbb{0}$, and $\mathcal{A} = (\mathcal{A}_{q,a} \mid q \in Q', a \in \tilde{\Sigma}_{\mathcal{V}})$ where for $\tilde{a} = (a, i, f)$ we have $\mathcal{A}_{(q,i),\tilde{a}} = (Q_{q,\tilde{a}}, I_{q,\tilde{a}}, \mu_{(q,i),\tilde{a}}, F_{q,\tilde{a}})$ with

$$\mu_{(q,i),\tilde{a}}(p_1, (q', i'), p_2) = \begin{cases} d_{i'} & \text{if } T_{q,\tilde{a}}(p_1, q', p_2) = \mathbb{1}\,, \\ \mathbb{0} & \text{otherwise} \end{cases}$$

for $p_1, p_2 \in Q_{q,a}$, and $(q', i') \in Q'$; and $\mathcal{A}_{(q,j),(a,i,f)}$ has an empty set of final states if $i \neq j$.

Obviously, $\|\mathcal{M}'\|(\tilde{t}) = \text{Val}(\mu(\tilde{t}, \tilde{\pi})) = [\![\forall x.\varphi]\!]((t, \sigma))$ for all trees $\tilde{t} = (t, \nu, \sigma) \in U_{\tilde{\Sigma}_{\mathcal{V}}}$ where $\tilde{\pi}$ is the extended run arisen from π and the π_us. Let the relabeling $h : \tilde{\Sigma}_{\mathcal{V}} \to \Sigma_{\mathcal{V}}$ be defined by $h((a, i, f)) = (a, f)$. One can show that $h(\|\mathcal{M}\|)(s) = [\![\forall x.\varphi]\!](s)$ for all $s \in U_{\Sigma_{\mathcal{V}}}$. Hence, $[\![\forall x.\varphi]\!]$ is recognizable by Lemma 3.6. □

Now we will prove our main result, Theorem 5.1.

Proof of Theorem 5.1. By Lemma 5.3 and Lemma 3.5, the semantics of almost boolean formulas over a regular ptv-monoid $\mathbb{D}$ is recognizable. For (1) the recognizability of the tree series $[\![\varphi]\!]$ for a formula φ is guaranteed by Propositions 5.4

and 5.5. For (2) we can proceed as in [10] and show by induction on the structure of φ that there is a WUTA recognizing $[\![\varphi]\!]$ whose weights are in the subsemiring generated by $\langle \mathrm{const}(\varphi) \cup \{\mathbb{0}, \mathbb{1}\}, +, \diamond\rangle$.

For the converse, let $\mathcal{M}$ be a WUTA recognizing S. In [15, Thm.6.9] the behavior of $\mathcal{M}$ was described with a formula using two universal quantifiers which occur nested. Due to the current definition of the behavior, $\|\mathcal{M}\|$ can be expressed by a $\forall$-restricted and strongly $\wedge$-restricted wMSO sentence φ. $\square$

Remark 5.6. We can show that Theorem 5.1 generalizes the respective main theorem of [10] for ranked trees. For this, we use Remark 3.2, the transformation from the weighted MSO logic over ranked trees to the weighted MSO logic over unranked trees [15, Lemma7.3], and a reverse transformation for wMSO sentences without subformula of the form $x \sqsubseteq y$.

Theorems 6.5 and 6.10 of [15] show that weighted unranked tree automata over non-commutative semirings are more expressive than the restricted weighted MSO logic. Our slightly changed definition of the behavior of WUTA enables us to prove an equivalence result as follows. Let $\mathbb{K} = (K, +, \cdot, \mathbb{0}, \mathbb{1})$ be a semiring. We associate $\mathbb{K}$ with the cctv-semiring $(K, +, \mathrm{Val}, \cdot, \mathbb{0}, \mathbb{1})$ with $\mathrm{Val}(t) = \prod_{u \in \mathrm{dom}(t)} t(u)$ where we multiply according to a depth-first left-to-right traversal, i.e. for a position u we first collect the weights of its subtrees one by one from left to right and then we multiply with the weight of u itself. Now we obtain:

Corollary 5.7. *Let Σ be an alphabet, $(K, +, \cdot, \mathbb{0}, \mathbb{1})$ a semiring, and S a tree series over Σ and $\mathbb{K}$. Then S is recognizable over $\mathbb{K}$ iff $S = [\![\varphi]\!]$ for a $\forall$-restricted and commutatively $\wedge$-restricted wMSO sentence φ.*

Hence, for commutative semirings, by Remark 3.2 and Corollary 5.7 we obtain the main equivalence results Theorem 6.5 and Theorem 6.9 of [15] as a consequence.

References

1. Bollig, B., Gastin, P.: Weighted versus probabilistic logics. In: Diekert, V., Nowotka, D. (eds.) DLT 2009. LNCS, vol. 5583, pp. 18–38. Springer, Heidelberg (2009)
2. Brüggemann-Klein, A., Murata, M., Wood, D.: Regular tree and regular hedge languages over unranked alphabets: version 1. Technical Report HKUST-TCSC-2001-0, The Honkong University of Sience and Technologie (2001)
3. Brüggemann-Klein, A., Wood, D.: Regular tree languages over non-ranked alphabets (1998). http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.50.5397
4. Chatterjee, K., Doyen, L., Henzinger, T.A.: Quantitative languages. In: Kaminski, M., Martini, S. (eds.) CSL 2008. LNCS, vol. 5213, pp. 385–400. Springer, Heidelberg (2008)
5. Chatterjee, K., Doyen, L., Henzinger, T.A.: Alternating weighted automata. In: Kutyłowski, M., Charatonik, W., Gębala, M. (eds.) FCT 2009. LNCS, vol. 5699, pp. 3–13. Springer, Heidelberg (2009)
6. Chatterjee, K., Doyen, L., Henzinger, T.A.: Probabilistic weighted automata. In: Bravetti, M., Zavattaro, G. (eds.) CONCUR 2009. LNCS, vol. 5710, pp. 244–258. Springer, Heidelberg (2009)

7. Chatterjee, K., Doyen, L., Henzinger, T.A.: Expressiveness and closure properties for quantitative languages. In: Proceedings of LICS 2009, pp. 199–208. IEEE Computer Society (2009)
8. Comon, H., Dauchet, M., Gilleron, R., Löding, C., Jacquemard, F., Lugiez, D., Tison, S., Tommasi, M.: Tree automata techniques and applications (2007). http://www.grappa.univ-lille3.fr/tata
9. Droste, M., Gastin, P.: Weighted automata and weighted logics. Theoretical Computer Science **380**, 69–86 (2007)
10. Droste, M., Götze, D., Märcker, S., Meinecke, I.: Weighted tree automata over valuation monoids and their characterization by weighted logics. In: Kuich, W., Rahonis, G. (eds.) Algebraic Foundations in Computer Science. LNCS, vol. 7020, pp. 30–55. Springer, Heidelberg (2011)
11. Droste, M., Kuich, W., Vogler, H. (eds.): Handbook of Weighted Automata. EATCS Monographs on Theoretical Computer Science, Springer (2009)
12. Droste, M., Meinecke, I.: Weighted automata and weighted MSO logics for average- and longtime-behaviors. Information and Computation **220–221**, 44–59 (2012)
13. Droste, M., Rahonis, G.: Weighted automata and weighted logics with discounting. Theory of Computing Systems **410**(37), 3481–3494 (2009)
14. Droste, M., Vogler, H.: Weighted tree automata and weighted logics. Theoretical Computer Science **366**, 228–247 (2006)
15. Droste, M., Vogler, H.: Weighted logics for unranked tree automata. Theory of Computing Systems **48**, 23–47 (2011)
16. Droste, M., Vogler, H.: Kleene and Büchi theorems for weighted automata and multi-valued logics over arbitrary bounded lattices. Theoretical Computer Science **418**, 14–36 (2012)
17. Fichtner, I.: Weighted picture automata and weighted logics. Theory of Computing Systems **48**(1), 48–78 (2011)
18. Fülöp, Z., Vogler, H.: Weighted tree automata and tree transducers, chap. 9. In: Droste et al. [11] (2009)
19. Fülöp, Z., Vogler, H.: Forward and backward application of symbolic tree transducers. Acta Informatica **51**, 297–325 (2014)
20. Grätzer, G.: General Lattice Theory, 2nd edn. Birkhäuser Verlag (January 2003)
21. Högberg, J., Maletti, A., Vogler, H.: Bisimulation minimisation of weighted automata on unranked trees. Fundamenta Informaticae **92**, 103–130 (2009)
22. Libkin, L.: Logics for unranked trees: an overview. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 35–50. Springer, Heidelberg (2005)
23. Mathissen, C.: Definable transductions and weighted logics for texts. Theoretical Computer Science **411**, 631–659 (2010)
24. Mathissen, C.: Weighted logics for nested words and algebraic formal power series. Logical Methods in Computer Science **6** (2010)
25. Meinecke, I.: Weighted logics for traces. In: Grigoriev, D., Harrison, J., Hirsch, E.A. (eds.) CSR 2006. LNCS, vol. 3967, pp. 235–246. Springer, Heidelberg (2006)
26. Murata, M.: Forest-regular languages and tree-regular languages (1995). (unpublished manuscript)
27. Neven, F.: Automata, logic, and XML. In: Bradfield, J.C. (ed.) CSL 2002 and EACSL 2002. LNCS, vol. 2471, p. 2. Springer, Heidelberg (2002)
28. Rahonis, G.: Weighted Muller tree automata and weighted logics. Journal of Automata, Languages and Combinatorics **12**, 455–483 (2007)
29. Tan, T.: Extending two-variable logic on data trees with order on data values and its automata. ACM Transactions on Computational Logic **15**, 8:1–8:39 (2014)

30. Thatcher, J.W.: Characterizing derivation trees of context-free grammars through a generalization of finite automata theory. Journal of Computer and System Sciences **1**, 317–322 (1967)
31. Thomas, W.: Languages, automata, and logic. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, vol. A, pp. 389–455. Springer (1997)
32. Veanes, M., Bjørner, N.: Symbolic tree transducers. In: Clarke, E., Virbitskaite, I., Voronkov, A. (eds.) PSI 2011. LNCS, vol. 7162, pp. 377–393. Springer, Heidelberg (2012)