A Nivat Theorem for Quantitative Automata on Unranked Trees

Manfred Droste and Doreen Götze(✉)

Institut für Informatik, Universität Leipzig, 04109 Leipzig, Germany
{droste,goetze}@informatik.uni-leipzig.de

Abstract. We derive a Nivat theorem for weighted unranked tree automata which states that their behaviors are exactly the functions which can be constructed from recognizable unranked tree languages and behaviors of very simple weighted unranked tree automata by using operations like relabelings and intersections. Thereby we prove the robustness of the weighted unranked tree automata model introduced recently. Moreover, we derive a similar theorem for weighted ranked tree automata. The characterizations work for valuation monoids as weight structures; they include all semirings, bounded lattices, and computations of averages of weights.

Keywords: Weighted tree automata · Nivat classes · Valuation monoids · Nivat theorem

1 Introduction

In 1967, Thatcher [32] investigated the theory of finite pseudoterms (nowadays known as unranked trees) and pseudoautomata (or unranked tree automata). In contrast to ranked trees (see [9,22,23] for surveys), for unranked trees the number of children of a node is not determined by the label of that node. Since then, due to the development of the modern document language XML and the fact that (fully structured) XML-documents can be formalized as unranked trees, the theory of unranked tree automata and unranked tree languages has developed intensively, cf. e.g. [2,7,28–30,32] and Chap. 8 of [9].

Classical unranked tree automata (amongst others) provide the opportunity to cope with qualitative questions, like reachability. More recently, also quantitative aspects gained much attention in automata theory. For instance, weighted real-time automata were investigated in [27], weighted modal systems in [4], and axiomatizations of weighted transition systems in [25]. For ranked trees, weighted automata were introduced in [1,6]; for surveys we refer to [13,21]. Weighted automata for unranked trees over semirings were investigated in [20,26]. A weighted unranked tree automata model over tree valuation monoids was introduced in [12]. Tree valuation monoids provide a very general weight structure

The second author was partially supported by the DFG Graduiertenkolleg 1763 (QuantLA).

L. Aceto et al. (Eds.): Larsen Festschrift, LNCS 10460, pp. 22–35, 2017.
DOI: 10.1007/978-3-319-63121-9_2

including all semirings, bounded (possibly non-distributive) lattices, and in addition, computations of averages or discounting of weights.

Nivat-type results provide a close relationship between weighted and unweighted automata models. In 1968, Nivat [31] (see also [5], Theorem 4.1) proved the fundamental theorem which characterizes rational transductions, and thereby established a connection between rational transductions and rational languages. Droste and Kuske [14] extended Nivat's theorem to weighted word automata over semirings. They showed that recognizable word series are exactly those which can be constructed from recognizable languages and very particular recognizable series using operations like morphisms and intersections. Recently, other extensions followed. Nivat theorems were given in [17,18] for weighted timed automata and weighted timed pushdown automata over timed valuation monoids and thereby implicitly also for weighted word automata over valuation monoids, in [3] for weighted picture automata over picture valuation monoids, in [10] for weighted graph automata over semirings, and in [33,34] for probabilistic automata on finite and infinite words and ranked trees, respectively.

The goal of this paper is such a Nivat result for weighted unranked tree automata over tree valuation monoids. Such automata 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 states at the children of the position. To define the behavior, we use extended runs which were already introduced in [20]. Additionally to the information of classical runs, extended runs also include runs of the weighted word automata called at positions of the input tree. Then the local weight of a position equals the weight of the transition taken for this position in the run of the position's parent. We use the valuation function of the tree valuation monoid 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. We obtain the weight of the tree as the sum of the weights of all its extended runs. In [12] it was shown that this model of weighted unranked tree automata is expressively equivalent to a suitable weighted MSO logic for unranked trees.

The main result of this paper gives a Nivat-type result for weighted unranked tree automata. We show that the behaviors of weighted unranked tree automata are exactly the functions which can be constructed from recognizable tree languages and behaviors of very simple weighted unranked tree automata by using operations like relabelings and intersections. Indeed, it even suffices to take functions mapping tree labels to tree valuation monoid elements instead of the very simple weighted unranked tree automata. It is clear that these functions define simple, recognizable tree series. Together with the results of [12], our present main result shows that the weighted unranked tree automata model of [12] is robust. In comparison to the proofs of the Nivat theorem for words (cf. [14]), for unranked trees technical difficulties arise from the technically more complex extended runs. Moreover, we also give a Nivat theorem for weighted ranked tree automata over tree valuation monoids.

2 Preliminaries

Let $\mathbb{N} = \{1, 2, \ldots\}$ be the set of all natural numbers and $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. For a set H, we denote by $|H|$ the cardinality of H and by H^* the set of all finite words over H. The empty word is denoted by ε. For sets $H_1, \ldots, H_n$ and $x \in H_1 \times \ldots \times H_n$, x_i equals the i-th component of x.

2.1 Trees and Tree Valuation Monoids

A *tree domain* is a finite, non-empty subset $\mathcal{B}$ of $\mathbb{N}^*$ such that for all $u \in \mathbb{N}^*$ and $i \in \mathbb{N}$, if $u.i \in \mathcal{B}$, then $u,\ u.1, \ldots,\ u.(i-1) \in \mathcal{B}$. An *unranked tree* over a set H (of labels) is a mapping $t \colon \mathcal{B} \to H$ such that $\mathrm{dom}(t) = \mathcal{B}$ is a tree domain. The set of all unranked trees over H is denoted by U_H. For every $h \in H$, we denote also by h the particular tree defined by $t : \{\varepsilon\} \to H$ and $t(\varepsilon) = h$. Let $t \in U_H$. The elements of $\mathrm{dom}(t)$ are called *positions* of t. Let $u \in \mathrm{dom}(t)$. We call $t(u)$ the *label* of t at u. The *rank* $\mathrm{rk}_t(u)$ of u is defined to be $\max\{i \in \mathbb{N} \mid u.i \in \mathrm{dom}(t)\}$. If $\mathrm{rk}_t(u) = 0$, then u is also called a *leaf of* t. We denote by $\mathrm{leaf}(t)$ the set of all leaves of t.

A *tree valuation monoid* (*tv-monoid* for short) [11,15] is a quadruple $(D, +, \mathrm{Val}, \mathbb{0})$ such that $(D, +, \mathbb{0})$ is a commutative monoid and $\mathrm{Val} \colon U_D \to D$ is a function, called *(tree) valuation function*, which satisfies that $\mathrm{Val}(d) = d$ for every tree $d \in D$, and $\mathrm{Val}(t) = \mathbb{0}$ for every $t \in U_D$ with $\mathbb{0} \in \mathrm{im}(t)$.

Example 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 [8,16] for words and in [11] for trees.

2.2 Weighted Unranked Tree Automata

Here we recall the definition of the class of recognizable tree series which were introduced in connection with restricted weighted MSO logics, cf. [12]. Since weighted unranked tree automata use weighted word automata we first recall the definition of weighted word automata over tree valuation monoids.

Let $\mathbb{D}$ be a tv-momoid and Σ an *alphabet*, i.e. a non-empty, finite set. A *weighted word automaton* 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 $w = w_1 \ldots w_n$ with $w_1, \ldots, w_n \in \Sigma$ 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 $p_0, \ldots, p_n \in P$. 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 $w \in \Sigma^*$.

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 weighted word automata 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 $u \in \mathrm{dom}(t) \setminus (\mathrm{leaf}(t) \cup \{\varepsilon\})$;
- $l\colon \mathrm{leaf}(t) \to P_{\mathcal{A}}$ 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 (q, s, l) is *successful* if $s(u.1) \dots s(u.\,\mathrm{rk}_t(u))$ is successful for all $u \in \mathrm{dom}(t) \setminus \mathrm{leaf}(t)$ and if $l(u)$ is successful for all $u \in \mathrm{leaf}(t)$ (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.

We will define the local weight of a position u by the weight of the transition taken for u in the run of the parent of u. This gives a tree $\mu(t, (q, s, l)) \in U_{\mathbb{D}}$ of weights with the same domain as t; then we apply Val to obtain the weight of the run (q, s, l) on t. Formally, we define 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. Then $\mathrm{Val}(\mu(t, (q, s, l)))$ is the *weight of* (q, s, l) *on* t. The *behavior* of a WUTA $\mathcal{M}$ is the function $\|\mathcal{M}\|\colon U_\Sigma \to \mathbb{D}$ defined by

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

for all $t \in U_\Sigma$. 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 an *(unranked) 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 $\|\mathcal{M}\| = S$.

Remark: Every unranked tree automaton M (see [32] for a definition of unranked tree automata) over an alphabet Σ can be seen as a weighted unranked tree automaton over Σ and the boolean semiring $\mathbb{B} = (\{0, 1\}, \wedge, \vee, 0, 1)$. Let $M = (Q, A, \gamma)$ be an unranked tree automaton. In the following, we identify the weight functions γ and $\mu_{q,a}$ $(q \in Q, a \in \Sigma)$ with their support sets. The language of M is defined as follows. We say that M recognizes $t \in U_\Sigma$ if there is a function $r\colon \operatorname{dom}(t) \to Q$ (which we call *successful run*) with $r(\epsilon) \in \gamma$ and such that there is a successful run of $A_{r(u),t(u)}$ on $r(u.1) \ldots r(u.\operatorname{rk}(t(u)))$ with $\operatorname{wt}(r(u.1) \ldots r(u.\operatorname{rk}(t(u)))) = 1$ for all $u \in \operatorname{dom}(t)$. The tree language recognized by M is defined by $L(M) = \{t \in U_\Sigma \mid M \text{ recognizes } t\}$. A tree language $L \subseteq U_\Sigma$ is *recognizable* if there is a unranked tree automaton M with $L(M) = L$. Note that this behavior definition is expressively equivalent to the earlier extended run behavior over the boolean semiring, nevertheless, we later use this behavior definition to avoid the syntactically more complex extended runs.

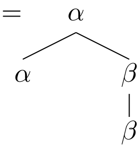
Example 2. Let $\mathbb{Q}_{\max}$ be the tv-monoid from Example 1 and Σ be an arbitrary, but fixed alphabet. In [12], a WUTA $\mathcal{M}$ which calculates the leaves-to-size ratio of a given input tree were given. The size of a tree is the number of all positions of the tree. For sake of completeness we include the definition of $\mathcal{M}$ and some considerations concerning $\mathcal{M}$'s behavior here. Let $\mathcal{M} = (\{c, n\}, \mathcal{A}, \gamma)$ over 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. The sub-automata $\mathcal{A}_{n,a}$ and $\mathcal{A}_{c,a}$ depicted in Fig. 1 for some $a \in \Sigma$.

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

$t_{\text{ex}} =$



.

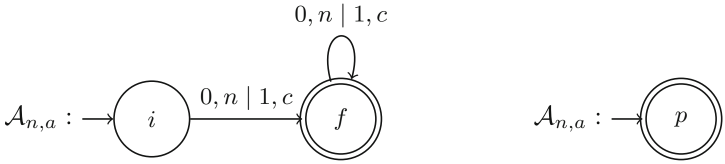
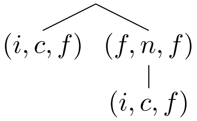
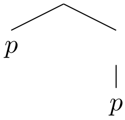


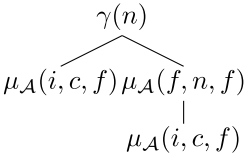
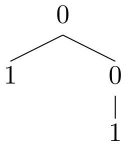
Fig. 1. Sub-automata of the example WUTA $\mathcal{M}$. Here, incoming arrows symbolize that a state is initial whereas a double border indicates that a state is final. An edge from one state p_1 to another state p_2 (p_1 and p_2 can be the same state) labeled with a, d stands for the transition (p_1, a, p_2) with weight d. Transitions with weight $-\infty$ are omitted.

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

and l= 

is an extended run on t_{ex}. Note that the domain of s excludes ε. therefore, above, the position ε is unlabeled for s. Similarly, the function l has leaves of t_{ex} as domain; therefore, above, the positions ε and 1 are unlabeled for l.

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

$\mu(t_{\text{ex}}, (n, s, l)) =$ 
$\gamma(n)$
$\mu_{\mathcal{A}}(i, c, f)$
$\mu_{\mathcal{A}}(f, n, f)$
$\mu_{\mathcal{A}}(i, c, f)$
$=$ 

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,

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

We will recall some properties of recognizable unranked tree series. Let $L \subseteq U_\Sigma$ and $S\colon U_\Sigma \to \mathbb{D}$. We define the *restriction of* S *on* L by the tree series $S \cap L\colon U_\Sigma \to \mathbb{D}$ by letting $(S \cap L)(t) = S(t)$ if $t \in L$ and $(S \cap L)(t) = \mathbb{0}$ if $t \notin L$.

Proposition 3 [12, **Lemma 3.4(2)**]. *Let* $\mathbb{D}$ *be a tv-monoid,* $L \subseteq U_\Sigma$ *and* $S\colon U_\Sigma \to \mathbb{D}$ *be recognizable. Then* $S \cap L$ *is also recognizable.*

Now we consider the closure under relabeling, similarly to [16,19]. Let Σ and Γ be two alphabets and $h\colon \Sigma \to 2^\Gamma$ be a mapping. Then h can be extended to a mapping $h'\colon U_\Sigma \to 2^{U_\Gamma}$ by letting $h'(t)$ be the set of all unranked trees t' over Γ such that $\operatorname{dom}(t') = \operatorname{dom}(t)$ and $t'(w) \in h(t(w))$ for each position $w \in \operatorname{dom}(t)$. For every $S\colon U_\Sigma \to \mathbb{D}$ the tree series $h''(S)\colon U_\Gamma \to \mathbb{D}$ 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.*

Proposition 4 [12, **Lemma 3.6**]. *Recognizable tree series are closed under relabeling.*

3 Nivat-Classes for Unranked Trees

Here we will define the set of all tree series which can be constructed from recognizable tree languages and behaviors of very simple weighted unranked tree automata by using operations like relabelings and intersections. Inspired by Weidner [35], we will call this set Nivat-class for unranked trees.

For the rest of this section let $\mathbb{D} = (D, +, \mathrm{Val}, \mathbb{0})$ be a tv-monoid, Γ be an alphabet and $\mathrm{g}\colon \Gamma \to D$ be a function. The function g later assigns labels to valuation monoid elements. The extension $\mathrm{g}'\colon U_\Gamma \to U_D$ of g is defined by $\mathrm{g}'(t)(u) = \mathrm{g}(t(u))$ for all $t \in \mathcal{O}(\Gamma)$ and $u \in \mathrm{dom}(t)$. In the following we denote g' also by g. Then $\mathrm{Val} \circ \mathrm{g}$ assigns to each tree $t \in U_\Gamma$ the weight $\mathrm{Val}(\mathrm{g}(t))$.

Definition 5. *Let Σ be an alphabet, $\mathbb{D}$ be a tv-monoid. The* Nivat-class $\mathcal{N}_{\mathbb{D}}(U_\Sigma)$ *for unranked trees consists of all $S\colon U_\Sigma \to \mathbb{D}$ for which there are:*

- *an alphabet Γ,*
- *a recognizable tree language $L \subseteq U_\Gamma$,*
- *a relabeling $h\colon \Gamma \to \Sigma$,*
- *a function* $\mathrm{g}\colon \Gamma \to D$

such that

$$S = h((\mathrm{Val} \circ \mathrm{g}) \cap L).$$

Example 6. Let $\mathbb{Q}_{\max}$ be the tv-monoid of Example 1 and Σ be an arbitrary, but fixed alphabet. We will show that the tree series defined by the WUTA in Example 2, which calculates the leaves-to-size ratio of trees, is in $\mathcal{N}_{\mathbb{Q}_{\max}}(U_\Sigma)$. For this, let

- $\Gamma = \Sigma \times \{\mathrm{leaf}, \mathrm{noleaf}\}$,
- $L = \{t \in U_\Gamma \mid \forall u \in \mathrm{dom}(t) \setminus \mathrm{leaf}(t)\colon t(u)_2 = \mathrm{noleaf}$
  $\wedge \forall u \in \mathrm{leaf}(t)\colon t(u)_2 = \mathrm{leaf}\}$,
- $h(a) = a_1$ for all $a \in \Gamma$,
- $g(a, \mathrm{leaf}) = 1$ and $g(a, \mathrm{noleaf}) = 0$ for all $a \in \Sigma$.

It is easy to check that L is indeed recognizable and that

$$h((\mathrm{Val} \circ \mathrm{g}) \cap L)(t) = \frac{\text{“number of leaves of } t\text{”}}{\text{“size of } t\text{”}}$$

for all $t \in U_\Sigma$.

Our main result will show that the Nivat-class for unranked trees and the set of all recognizable tree series are the same. For the proof of the inclusion of the

Nivat-class in the set of all recognizable tree series we first prove that $\mathrm{Val}(g)$ is recognizable by an especially simple weighted tree automaton $\mathcal{M}$. In particular, we can choose $\mathcal{M}$ with state set Γ and $\mathcal{A} = (A_{q,a} \mid q \in \Gamma, a \in \Gamma)$ where $\mathcal{A}_{q,a}$ has only one state.

Lemma 7. *Let Γ be an alphabet, $\mathbb{D} = (D, +, \mathrm{Val}, \mathbb{0})$ be a tv-monoid, and $\mathrm{g}\colon \Gamma \to D$. Then $\mathrm{Val} \circ \mathrm{g}$ is recognizable.*

Proof. We will build a WUTA $\mathcal{M}$ which recognizes $\mathrm{Val} \circ \mathrm{g}$. Basically, $\mathcal{M}$ will have exactly one successful extended run per input tree $t \in U_\Gamma$. The local weight of this extended run at a position u shall be $\mathrm{g}(t(u))$. For this we set the state set of $\mathcal{M}$ to Γ and let the root weight of state $a \in \Gamma$ be $\mathrm{g}(a)$. Moreover, each subautomaton $\mathcal{A}_{a,a}$ $(a \in \Gamma)$ of $\mathcal{M}$ has only one state q which is initial and final (so that there is only one successful run), and each transition of $\mathcal{A}_{a,a}$ labeled with $b \in \Gamma$ has weight $\mathrm{g}(b)$ (this generates the local weight which we would like to produce). All other subautomata do not produce any successful run. The latter secures that one only gets a successful extended run if for a-labeled positions $\mathcal{A}_{a,a}$ is called. Formally, let $\mathcal{M} = (\Gamma, \mathcal{A}, \gamma)$ with $\gamma(a) = g(a)$ for all $a \in \Gamma$ and $\mathcal{A} = (\mathcal{A}_{q,a} \mid q \in \Gamma, a \in \Gamma)$ with $\mathcal{A}_{a,a} = (\{q\}, \{q\}, \mu_{a,a}, \{q\})$ with $\mu_{a,a}(b) = g(b)$ for all $b \in \Gamma$ and $\mathcal{A}_{c,a} = (\{q\}, \{q\}, \underline{\mathbb{0}}, \emptyset)$ for all $a, c \in \Gamma$ with $a \neq c$ where $\underline{\mathbb{0}}$ denotes a function which maps all triples to $\mathbb{0}$.

Using the intuition behind $\mathcal{M}$ given above, one can easily check that $\|\mathcal{M}\|(t) = \mathrm{Val}(g(t))$ for all $t \in U_\Gamma$. □

Now we prove our main theorem.

Theorem 8. *Let Σ be an alphabet, $\mathbb{D}$ be a tv-monoid, and $S\colon U_\Sigma \to \mathbb{D}$ a tree series. Then S is recognizable iff $S \in \mathcal{N}_{\mathbb{D}}(U_\Sigma)$.*

Proof. We start with the proof of the "if"-implication. For this, let Γ be an alphabet, $L \subseteq U_\Gamma$ a recognizable tree language, $h\colon \Gamma \to \Sigma$ a relabeling, and $\mathrm{g}\colon \Gamma \to D$ a function such that $S = h((\mathrm{Val} \circ \mathrm{g}) \cap L)$. By Lemma 7, $\mathrm{Val} \circ \mathrm{g}$ is recognizable, and thus, by Proposition 3, also $(\mathrm{Val} \circ \mathrm{g}) \cap L$ is recognizable. Hence by Proposition 4, $S = h((\mathrm{Val} \circ \mathrm{g}) \cap L)$ is recognizable.

For the converse, let S be recognizable and $\mathcal{M} = (Q, \mathcal{A}, \gamma)$ be a WUTA with $\|\mathcal{M}\| = S$. Moreover 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 $P_\mathcal{A} = \bigcup_{q \in Q, a \in \Sigma} P_{q,a}$. Let $\mu_\mathcal{A}$ be the union of the transition functions $\mu_{q,a}$.

We will simulate the behavior of $\mathcal{M}$ by appropriately chosen Γ, L, h, and g. The main idea for the choice of Γ, L, h, and g is that L will be the set of successful extended runs of $\mathcal{M}$, g will determine the local weights of the extended runs in L, the valuation function Val will calculate the weights of the extended runs, and h will be a projection of the extended runs on their related trees (this results in the "sum over all trees" since h is a relabeling). Since L shall be a set of trees over an alphabet Γ, we have to encode (successful) extended runs by trees. As indicated in Example 2, each component q, s, and l, respectively, of an extended run (q, s, l) on a tree t can be viewed as a tree with possibly unlabeled positions over the tree domain $\mathrm{dom}(t)$. Then q has labels in $\Gamma_1 = Q$, s in $\Gamma_2 = P_{\mathcal{A}} \times Q \times P_{\mathcal{A}}$, and l in $P_{\mathcal{A}}$ whereby the root is the only labeled position in q and the only unlabeled position in s, respectively, and in l only leaves are labeled. We combine these three trees to one tree via building tuples from the labels of q, s, and l. The resulting tree then has labels in $\Gamma_1 \cup \Gamma_2 \cup \Gamma_3$ where $\Gamma_3 = ((P_{\mathcal{A}} \times Q \times P_{\mathcal{A}}) \times P_{\mathcal{A}})$. Thus we set $\Gamma = (\Gamma_1 \cup \Gamma_2 \cup \Gamma_3) \times \Sigma$. We built the Cartesian product with Σ, so that later h can extract the related tree. Let $t \in U_\Sigma$, (q, s, l) an extended run on t, and τ a tree over Γ. We say that τ encodes the pair $((q, s, l), t)$ if $\mathrm{dom}(t) = \mathrm{dom}(\tau)$ and $\tau(\varepsilon) = (q, t(\varepsilon))$, $\tau(u) = (s(u), t(u))$ for all $u \in \mathrm{dom}(t) \setminus (\{\varepsilon\} \cup \mathrm{leaf}(t))$, and $\tau(u) = ((s(u), l(u)), t(u))$ for all $u \in \mathrm{leaf}(t) \setminus \{\varepsilon\}$. From now on we identify a pair $((q, s, l), t)$ and its encoding $\tau \in U_\Gamma$.

Now we can define Γ, L, h, and g:

- $\Gamma = (\Gamma_1 \cup \Gamma_2 \cup \Gamma_3) \times \Sigma$
- $L = \{((q, s, l), t) \in U_\Gamma \mid (q, s, l) \in \mathrm{succ}(\mathcal{M}, t)\}$
- $h(q', a) = a$
- $\mathrm{g}(q', a) = \begin{cases} \gamma(a) & \text{if } q' \in \Gamma_1 \\ \mu(p_1, q, p_2) & \text{if } q' = (p_1, q, p_2) \in \Gamma_2 \text{ or } q' = ((p_1, q, p_2), p) \in \Gamma_3 \end{cases}$

for $a \in \Sigma$ and $(q', a) \in \Gamma$.

We show that L is actually recognizable. For this, we construct an unranked tree automaton $M_{\text{runs}} = (Q_{\text{runs}}, A, \gamma_{\text{runs}})$ which has only τ as successful run on an input tree $\tau = ((q, s, l), t) \in U_\Gamma$ iff (q, s, l) is a successful extended run of $\mathcal{M}$ on t. Thus, the state set Q_{runs} of M_{runs} will be Γ and only subautomata $A_{\alpha,\alpha}$ $(\alpha \in \Gamma)$ actually accept a non-empty language. The subautomaton $\mathcal{A}_{\alpha,\alpha}$ of M_{runs} will be a version of the subautomaton $\mathcal{A}_{q,a}$ (where q is the Q-component and a is the Σ-component, respectively, of α) of $\mathcal{M}$ without weights. Hence

- $\tau(1) \ldots \tau(\mathrm{rk}_\tau(\varepsilon)) \in L(\delta(\tau(\varepsilon), \tau(\varepsilon)))$ iff $s(1) \ldots s(\mathrm{rk}_t(u))$ is a successful run in $\mathcal{A}_{q,t(\varepsilon)}$,
- $\tau(u.1) \ldots \tau(u.\,\mathrm{rk}_\tau(u)) \in L(\delta(\tau(u), \tau(u)))$ iff $s(u.1) \ldots s(u.\,\mathrm{rk}_t(u))$ is a successful run in $\mathcal{A}_{s(u)_2,t(u)}$ for all $u \in \mathrm{dom}(t) \setminus (\{\varepsilon\} \cup \mathrm{leaf}(t))$,
- $\varepsilon \in L(\delta(\tau(u), \tau(u)))$ iff $l(u)$ is a successful run in $\mathcal{A}_{s(u)_2,t(u)}$ for all leaves u

for all trees $\tau = ((q, s, l), t) \in U_\Gamma$. To guarantee that trees accepted by M_{runs} have their root label in $\Gamma_1 \times \Sigma$, we let $\Gamma_1 \times \Sigma$ be the set of final states γ_{runs} of M_{runs}. Moreover, subautomata associated to a state in $\Gamma_2 \times \Sigma$ will not accept the empty word, hence, M_{runs} does not allow runs where states in $\Gamma_2 \times \Sigma$ occur at leaf positions. Leaf positions shall be labeled with states in $\Gamma_3 \times \Sigma$. We achieve this by letting $\mathcal{A}_{\alpha,\alpha}$ for $\alpha \in \Gamma_3 \times \Sigma$ be a word automaton which at most accepts the empty word.

Formally, let $M_{\text{runs}} = (\Gamma, A, Q \times \Sigma)$ with

- $A_{\alpha,\alpha} = (P_{q,a}, I_{q,a}, \mu_{q,a}, F_{q,a})$ for $\alpha = (q, a) \in (\Gamma_1 \times \Sigma)$ where $P_{q,a}, I_{q,a}, F_{q,a}$ are as in $\mathcal{A}_{q,a}$ and

$$\begin{aligned} \mu_{q,a} = &\{(p_1, (p_1, q', p_2), p_2) \mid p_1, p_2 \in P_{q,a}, q' \in Q\} \\ &\cup \{(p_1, ((p_1, q', p_2), p), p_2) \mid p_1, p_2, p \in P_{q,a}, q' \in Q\} \end{aligned}$$

- $A_{\alpha,\alpha} = (P_{q,a} \cup \overline{I_{q,a}}, \overline{I_{q,a}}, \mu'_{q,a}, F_{q,a})$ for $\alpha = ((p'_1, q, p'_2), a) \in (\Gamma_2 \times \Sigma)$ where $P_{q,a}, I_{q,a}, F_{q,a}$ are as in $\mathcal{A}_{q,a}$, $\overline{I_{q,a}}$ is a disjoint copy of $I_{q,a}$, and

$$\begin{aligned} \mu'_{q,a} = T_{q,a} &\cup \{(\overline{i_1}, (i, q', p_2), p_2) \mid i \in I_{q,a}, p_2 \in P_{q,a}, q' \in Q\} \\ &\cup \{(\bar{i}, ((i, q', p_2), p), p_2) \mid i \in I_{q,,a}, p_2, p \in P_{q,a}, q' \in Q\} \end{aligned}$$

  with

$$\begin{aligned} \mu_{q,a} = \{&(p_1, (p_1, q', p_2), p_2) \mid p_1, p_2 \in P_{q,a}, q' \in Q\} \\ &\cup \{(p_1, ((p_1, q', p_2), p), p_2) \mid p_1, p_2, p \in P_{q,a}, q' \in Q\} \end{aligned}$$

- $A_{\alpha,\alpha} = (\{p_\alpha\}, \{p_\alpha\}, \emptyset, \{p_\alpha\})$ for $\alpha = (((p'_1, q, p'_2), p'), a) \in (\Gamma_3 \times \Sigma)$ with $p' \in I_{q,a} \cap F_{q,a}$ and $A_{\alpha,\alpha} = (\{p_\alpha\}, \{p_\alpha\}, \emptyset, \emptyset)$ for $\alpha = (((p'_1, q, p'_2), p'), a)$ with $p' \notin I_{q,a} \cap F_{q,a}$
- $A_{\alpha,\beta} = (\{p\}, \{p\}, \emptyset, \emptyset)$

for $\alpha, \beta \in \Gamma$ and $\alpha \neq \beta$. One can easily prove that $L(M_{\text{runs}}) = L$.

Now let $t \in U_\Sigma$. Then

$$\begin{aligned} h((\mathrm{Val} \circ \mathrm{g}) \cap L)(t) &= \sum_{\tau \in U_\Gamma \wedge t \in h(\tau)} ((\mathrm{Val} \circ \mathrm{g}) \cap L)(\tau) \\ &= \sum_{\tau = ((q,s,l),t) \wedge (q,s,l) \in \mathrm{succ}(\mathcal{M},t)} \mathrm{Val}(g(\tau)) \\ &= \sum_{\tau = ((q,s,l),t) \wedge (q,s,l) \in \mathrm{succ}(\mathcal{M},t)} \mathrm{Val}(\mu(t, (s, q, l))) \\ &= \sum_{(q,s,l) \in \mathrm{succ}(\mathcal{M},t)} \mathrm{Val}(\mu(t, (s, q, l))) \\ &= \|\mathcal{M}\|(t)\,. \end{aligned}$$

□

4 The Ranked Tree Case

In this section we will show a version of Theorem 8 for ranked trees. For this, we briefly recall the definitions of ranked alphabets, ranked trees and weighted ranked tree automata over tv-monoids as well as some considerations on the relationship between weighted ranked tree automata and weighted unranked tree automata.

A *ranked alphabet* is a pair $(\Sigma, \mathrm{rk}_\Sigma)$, where Σ is an alphabet and $\mathrm{rk}_\Sigma \colon \Sigma \to \mathbb{N}_0$ is a mapping which assigns to each symbol of Σ its rank. We denote by $\Sigma^{(k)}$ the set of all symbols which have rank k and by $a^{(k)}$ that a is in $\Sigma^{(k)}$. Usually we drop rk_Σ and denote a ranked alphabet simply by Σ. In this paper we assume that $\Sigma^{(0)} \neq \emptyset$. We define $\max_\Sigma = \max\{\mathrm{rk}_\Sigma(\sigma) \mid \sigma \in \Sigma\}$. A *ranked tree* over a ranked alphabet Σ is an unranked tree over the set Σ such that for all $u \in \mathrm{dom}(t)$, $\mathrm{rk}_t(u) = k$ whenever $t(u) \in \Sigma^{(k)}$. We denote the set of all ranked trees over Σ by T_Σ.

Let Σ be a ranked alphabet and $(D, +, \mathrm{Val}, \mathbb{0})$ a tv-monoid. A *weighted ranked tree automaton* (*WRTA* for short) over Σ and D is a triple $\mathcal{M} = (Q, \mu, F)$ where Q is a non-empty finite set of states, $\mu = (\mu_m)_{0 \leq m \leq \max_\Sigma}$ is a family of transition mappings $\mu_m \colon Q^m \times \Sigma^{(m)} \times Q \to D$, and $F \subseteq Q$ is a set of final states.

A *run r of $\mathcal{M}$ on a tree $t \in T_\Sigma$* is a mapping $r \colon \mathrm{dom}(t) \to Q$. Since the domain of a run is a tree domain, each run r on t defines a tree $\mu(t, r) \in T_D$ where $\mathrm{dom}(\mu(t, r)) = \mathrm{dom}(t)$ and $\mu(t, r)(u) = \mu_m(r(u.1) \ldots r(u.m), t(u), r(u))$ with $t(u) \in \Sigma^{(m)}$ for all $u \in \mathrm{dom}(t)$. We call r on t *successful* if $r(\varepsilon) \in F$. The *behavior* of a WRTA $\mathcal{M}$ is the function $\|\mathcal{M}\| \colon T_\Sigma \to \mathbb{D}$ defined by

$$\|\mathcal{M}\|(t) = \sum \big(\mathrm{Val}(\mu(t, r)) \mid r \text{ is successful run of } \mathcal{M} \text{ on } t\big)$$

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

A *ranked tree series* is a mapping $S \colon T_\Sigma \to \mathbb{D}$. A tree series S is called *recognizable* if $S = \|\mathcal{M}\|$ for some WRTA $\mathcal{M}$. As is well-known, a ranked tree automaton (cf. [9]) can be seen as a weighted ranked tree automaton over the boolean semiring, and conversely.

In passing, we note the following result. It shows that on ranked trees WRTA and WUTA have the same expressive power.

Proposition 9 [24, **Lemma 3.10 and Lemma 3.11**]. *Let Σ be a ranked alphabet.*

1. *For every WRTA $\mathcal{N}$ over Σ there exists a WUTA $\mathcal{M}$ such that $\|\mathcal{M}\|(t) = \|\mathcal{N}\|(t)$ for all $t \in T_\Sigma$ and $\|\mathcal{M}\|(t) = \mathbb{0}$ for all $t \in U_\Sigma \setminus T_\Sigma$.*
2. *For every WUTA $\mathcal{M}$ over Σ there exists a WRTA $\mathcal{N}$ such that $\|\mathcal{N}\|(t) = \|\mathcal{M}\|(t)$ for all $t \in T_\Sigma$.*

Now we define a Nivat-class for ranked trees. For this, we define for all $L \subseteq T_\Sigma$ and $S \colon T_\Sigma \to \mathbb{D}$ the *restriction of S on L* analogously to the respective definition of the restriction of a tree series and an unranked tree language. Moreover, let Σ and Γ be two ranked alphabets and $h \colon \Sigma \to 2^\Gamma$ be a mapping with $\mathrm{rk}_\Gamma(b) = \mathrm{rk}_\Sigma(a)$ for all $b \in h(a)$. We extend h to a mapping h' from ranked trees over Σ to the power set of ranked trees over Γ, and afterwards to a mapping h'' from ranked tree series over Σ and $\mathbb{D}$ to ranked tree series over Γ and $\mathbb{D}$ analogously as we did in the unranked tree case. Again, we denote h' and h'' also by h which we call a *relabeling*. Let Γ be a ranked alphabet, $\mathrm{g} \colon \Gamma \to D$ be a function, and the extension $\mathrm{g}' \colon U_\Gamma \to U_D$ of g be defined by $\mathrm{g}'(t)(u) = \mathrm{g}(t(u))$ for all $t \in T_\Gamma$ and $u \in \mathrm{dom}(t)$. In the following we denote g' also by g.

Definition 10. *The* Nivat-class $\mathcal{N}_{\mathbb{D}}(T_\Sigma)$ *for ranked trees consists of all $S \colon T_\Sigma \to \mathbb{D}$ for which there are:*

- *an alphabet Γ,*
- *a recognizable ranked tree language $L \subseteq T_\Gamma$,*
- *a relabeling $h \colon \Gamma \to \Sigma$,*
- *a function $\mathrm{g} \colon \Gamma \to D$*

such that

$$S = h((\mathrm{Val} \circ \mathrm{g}) \cap L)\,.$$

Example 11. Let Σ be a ranked alphabet. We will show that also the ranked tree series which calculates the leaves-to-size ratio of trees is in $\mathcal{N}_{\mathbb{Q}_{\max}}(T_\Sigma)$. We let $\Gamma = \Sigma$, $L = T_\Gamma$, $h(a) = a$ for all $a \in \Gamma$, and $g(a^{(0)}) = 1$ and $g(a^{(k)}) = 0$ for all $a \in \Sigma$ and $k > 0$. Obviously L is recognizable and $h((\mathrm{Val} \circ \mathrm{g}) \cap L)$ calculates the leaves-to-size ratio of ranked trees. A WRTA that recognizes $h((\mathrm{Val} \circ \mathrm{g}) \cap L)$ was given in [11].

Next we prove a Nivat theorem for ranked trees.

Theorem 12. *Let Σ be a ranked alphabet, $\mathbb{D}$ a tv-monoid, and $S\colon T_\Sigma \to \mathbb{D}$ a tree series. Then S is recognizable iff $S \in \mathcal{N}_{\mathbb{D}}(T_\Sigma)$.*

Proof. Let Γ be an alphabet, $L \subseteq T_\Gamma$ a recognizable ranked tree language, $h\colon \Gamma \to \Sigma$ a relabeling, and $\mathrm{g}\colon \Gamma \to D$ a function such that $S = h((\mathrm{Val} \circ \mathrm{g}) \cap L)$. As is easy to see, $\mathrm{Val} \circ \mathrm{g}$ is accepted by a one state automaton $\mathcal{M}$. Indeed, $\mathcal{M} = (\{q\}, \mu, \{q\})$ with $\mu_m(q \dots q, a, q) = g(a)$ for all $a \in \Gamma^{(m)}$ and $m \in \mathbb{N}$. By the versions of Proposition 3 and Proposition 4 for ranked trees (cf. [11]), $(\mathrm{Val} \circ \mathrm{g}) \cap L$ is recognizable, and thus, $S = h((\mathrm{Val} \circ \mathrm{g}) \cap L)$ is recognizable.

For the converse implication, let $S = \|\mathcal{M}\|$ for some WRTA $\mathcal{M} = (Q, \mu, F)$. Then let

- $\Gamma = \bigcup_{0 \le m \le \max_\Sigma} Q^m \times \Sigma^{(m)} \times Q$ with $\mathrm{rk}(q_1 \dots q_m, a, q) = \mathrm{rk}(a)$ for all $q_1, \dots, q_m, q \in Q$, $a \in \Sigma^{(m)}$ be the set of all transitions of $\mathcal{M}$,
- $L = \{t \in T_\Gamma \mid \forall u \in \mathrm{dom}(t) \text{ with } t(u) \in \Gamma^{(m)}\colon t(u.i)_3 = (t(u)_1)_i$ for all $1 \le i \le m$ and $t(\varepsilon)_3 \in F\}$ describing the set of all successful runs of $\mathcal{M}$,
- $h((q_1, \dots, q_m), a, q) = a$,
- $g((q_1, \dots, q_m), a, q) = \mu_m(q_1 \dots q_m, a, q)$

for all $q_1, \dots, q_m, q \in Q$, $a \in \Sigma^{(m)}$. One can check that L is recognizable and $h((\mathrm{Val} \circ \mathrm{g}) \cap L) = S$. □

Conclusion

We proved two Nivat theorems for weighted unranked tree automata and for weighted ranked tree automata over tree valuation monoids.

In [17,33,34], the Nivat theorem was used to show the expressive equivalence of a suitable MSO logic and the respective automata model. We think that, similarly, Theorem 8 could be used to derive an alternative proof to the one in [12] showing that the weighted MSO logic defined there and weighted unranked tree automata over tree valuation monoids are expressively equivalent.

References

1. Alexandrakis, A., Bozapalidis, S.: Weighted grammars and Kleene's theorem. Inf. Process. Lett. **24**(1), 1–4 (1987)
2. Brüggemann-Klein, A., Wood, D.: Regular Tree Languages Over Non-Ranked Alphabets (1998)

3. Babari, P., Droste, M.: A Nivat theorem for weighted picture automata and weighted MSO logic. In: Dediu, A.-H., Formenti, E., Martín-Vide, C., Truthe, B. (eds.) LATA 2015. LNCS, vol. 8977, pp. 703–715. Springer, Cham (2015). doi:10.1007/978-3-319-15579-1_55
4. Bauer, S.S., Fahrenberg, U., Juhl, L., Larsen, K.G., Legay, A., Thrane, C.R.: Weighted modal transition systems. Form. Methods Syst. Des. **42**(2), 193–220 (2013)
5. Berstel, J.: Transductions and Context-Free Languages. Teubner Studienbücher: Informatik. Teubner, Leipzig (1979)
6. Berstel, J., Reutenauer, C.: Recognizable formal power series on trees. Theoret. Comput. Sci. **18**(2), 115–148 (1982)
7. 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)
8. 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). doi:10.1007/978-3-540-87531-4_28
9. 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. Release 12 October 2007
10. Droste, M., Dück, S.: Weighted automata and logics on graphs. In: Italiano, G.F., Pighizzini, G., Sannella, D.T. (eds.) MFCS 2015. LNCS, vol. 9234, pp. 192–204. Springer, Heidelberg (2015). doi:10.1007/978-3-662-48057-1_15
11. 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). doi:10.1007/978-3-642-24897-9_2
12. Droste, M., Heusel, D., Vogler, H.: Weighted unranked tree automata over tree valuation monoids and their characterization by weighted logics. In: Maletti, A. (ed.) CAI 2015. LNCS, vol. 9270, pp. 90–102. Springer, Cham (2015). doi:10.1007/978-3-319-23021-4_9
13. Droste, M., Kuich, W., Vogler, H. (eds.): Handbook of Weighted Automata. EATCS Monographs on Theoretical Computer Science. Springer, Heidelberg (2009)
14. Droste, M., Kuske, D.: Weighted automata. In: Pin, J.-E. (ed.) Automata: From Mathematics to Applications. European Mathematical Society (to appear)
15. Droste, M., Meinecke, I.: Describing average- and longtime-behavior by weighted MSO logics. In: Hliněný, P., Kučera, A. (eds.) MFCS 2010. LNCS, vol. 6281, pp. 537–548. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15155-2_47
16. Droste, M., Meinecke, I.: Weighted automata and weighted MSO logics for average- and longtime-behaviors. Inf. Comput. **220–221**, 44–59 (2012)
17. Droste, M., Perevoshchikov, V.: A Nivat theorem for weighted timed automata and weighted relative distance logic. In: Esparza, J., Fraigniaud, P., Husfeldt, T., Koutsoupias, E. (eds.) ICALP 2014. LNCS, vol. 8573, pp. 171–182. Springer, Heidelberg (2014). doi:10.1007/978-3-662-43951-7_15
18. Droste, M., Perevoshchikov, V.: Logics for weighted timed pushdown automata. In: Beklemishev, L.D., Blass, A., Dershowitz, N., Finkbeiner, B., Schulte, W. (eds.) Fields of Logic and Computation II. LNCS, vol. 9300, pp. 153–173. Springer, Cham (2015). doi:10.1007/978-3-319-23534-9_9
19. Droste, M., Vogler, H.: Weighted tree automata and weighted logics. Theoret. Comput. Sci. **366**, 228–247 (2006)

20. Droste, M., Vogler, H.: Weighted logics for unranked tree automata. Theory Comput. Syst. **48**, 23–47 (2011)
21. Fülöp, Z., Vogler, H.: Weighted tree automata and tree transducers. In: Droste et al. [13], chap. 9, pp. 313–403
22. Gécseg, F., Steinby, M.: Tree Automata. Akadémiai Kiadó (1984). http://www.arxiv.org
23. Gécseg, F., Steinby, M.: Tree languages. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, vol. 3, pp. 1–68. Springer, Heidelberg (1997)
24. Götze, D.: Weighted unranked tree automata over tree valuation monoids (submitted)
25. Hansen, M., Larsen, K.G., Mardare, R., Pedersen, M.R., Xue, B.: A complete approximation theory for weighted transition systems. In: Fränzle, M., Kapur, D., Zhan, N. (eds.) SETTA 2016. LNCS, vol. 9984, pp. 213–228. Springer, Cham (2016). doi:10.1007/978-3-319-47677-3_14
26. Högberg, J., Maletti, A., Vogler, H.: Bisimulation minimisation of weighted automata on unranked trees. Fundamenta Informaticae **92**, 103–130 (2009)
27. Larsen, K., Behrmann, G., Brinksma, E., Fehnker, A., Hune, T., Pettersson, P., Romijn, J.: As cheap as possible: effcient cost-optimal reachability for priced timed automata. In: Berry, G., Comon, H., Finkel, A. (eds.) CAV 2001. LNCS, vol. 2102, pp. 493–505. Springer, Heidelberg (2001). doi:10.1007/3-540-44585-4_47
28. 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). doi:10.1007/11523468_4
29. Murata, M.: Forest-regular languages and tree-regular languages. Unpublished manuscript (1995)
30. Neven, F.: Automata, logic, and XML. In: Bradfield, J. (ed.) CSL 2002. LNCS, vol. 2471, pp. 2–26. Springer, Heidelberg (2002). doi:10.1007/3-540-45793-3_2
31. Nivat, M.: Transduction des langages de Chomsky. Ph.D. thesis, University of Paris (1968)
32. Thatcher, J.W.: Characterizing derivation trees of context-free grammars through a generalization of finite automata theory. J. Comput. Syst. Sci. **1**, 317–322 (1967)
33. Weidner, T.: Probabilistic automata and probabilistic logic. In: Rovan, B., Sassone, V., Widmayer, P. (eds.) MFCS 2012. LNCS, vol. 7464, pp. 813–824. Springer, Heidelberg (2012). doi:10.1007/978-3-642-32589-2_70
34. Weidner, T.: Probabilistic regular expressions and MSO logic on finite trees. In: Proceedings of FSTTCS 2015. LIPIcs, vol. 45, pp. 503–516. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2015)
35. Weidner, T.: Probabilistic Logic, Probabilistic Regular Expressions, and Constraint Temporal Logic. Ph.D. thesis, Universität Leipzig (2016)