On Finite and Polynomial Ambiguity of Weighted Tree Automata

Erik Paul[⋆]

Institute of Computer Science, Leipzig University, D-04109 Leipzig, Germany
`epaul at informatik.uni-leipzig.de`

Abstract. We consider finite and polynomial ambiguity of weighted tree automata. Concerning finite ambiguity, we show that a finitely ambiguous weighted tree automaton can be decomposed into a sum of unambiguous automata. For polynomial ambiguity, we show how to decompose a polynomially ambiguous weighted tree automaton into simpler polynomially ambiguous automata and then analyze the structure of these simpler automata. We also outline how these results can be used to capture the ambiguity of weighted tree automata with weighted logics.

Keywords: Weighted Tree Automata, Quantitative Tree Automata, Finite Ambiguity, Polynomial Ambiguity, Weighted Logics

1 Introduction

Weighted automata, a generalization of non-deterministic finite automata (NFA), have first been investigated by Schützenberger [22]. Since then, a large amount of further research has been conducted on them, cf. [21,18,3,9]. When considering complexity and decidability problems for these automata, the concept of ambiguity plays a large role. For instance, in [13] the equivalence problem for finitely ambiguous automata over the max-plus semiring is shown to be decidable, whereas for general non-deterministic automata over the max-plus semiring this problem is undecidable [17]. The ambiguity of an automaton is a measure for the maximum number of accepting runs on a given input. For example, if the number of accepting paths is bounded by a global constant for every word, we say that the automaton is finitely ambiguous. In the case that the number of accepting paths is bounded polynomially in the word length, we speak of polynomial ambiguity.

In this paper, we investigate these two types of ambiguity for weighted tree automata (WTA), a weighted automata model with trees as input. Our main results are the following:

- A finitely ambiguous WTA can be decomposed into a sum of several unambiguous WTA.

⋆ supported by Deutsche Forschungsgemeinschaft (DFG), Graduiertenkolleg 1763 (QuantLA)

- A polynomially ambiguous WTA can be decomposed into a sum of "simpler" polynomially ambiguous WTA. Here, for each of these simpler automata we can identify a set of transitions such that, intuitively speaking, in every run on any tree each of these transitions occurs at exactly one position of the tree. Furthermore, the possible number of runs on any tree is bounded if we specify the position of each of these transitions. The bound does not depend on the given tree.
- To each of the classes of unambiguous, finitely ambiguous and polynomially ambiguous WTA, we relate a class of sentences from a weighted MSO logic expressively equivalent to it.

Weighted tree automata have been considered by a number of researchers [2,4,19], see [12] for a survey. Likewise, the ambiguity of finite automata has been studied numerous times. For example, [24,23,1] present criteria for ambiguity and algorithms to determine the ambiguity of automata. For weighted automata on words (WA), it has also been shown that expressive power increases with growing degree of ambiguity. It is shown in [15] that the inclusions *deterministic WA* $\subsetneq$ *unambiguous WA* $\subsetneq$ *finitely ambiguous WA* are strict and in [14] it is shown that the inclusion *finitely ambiguous WA* $\subsetneq$ *polynomially ambiguous WA* is strict.

Our first two results give a deeper insight into the structure of WTA and generalize results by Seidl and Weber [24] and Klimann et al. [15] from words to trees. As trees do not have the linear structure of words, however, the corresponding proofs from the word case can not be adapted to the tree case in a trivial way. Both results are new even for WTA over the boolean semiring, i.e. for tree automata without weights.

The initial motivation for our investigations lies with logics and the third result. Weighted logics can be used to describe weighted automata over words and trees, as was shown by Droste, Gastin and Vogler [8,10]. Kreutzer and Riveros [16] later showed that weighted logics can even be used to characterize different degrees of ambiguity of weighted automata over words. With the help of the first two results, we can generalize Kreutzer's and Riveros's result to WTA. For polynomial ambiguity, we even obtain a stronger result, as we are able to capture the polynomial degree of a WTA not only in the boolean semiring, but in any commutative semiring.

2 Weighted Tree Automata

Let $\mathbb{N} = \{0, 1, 2, \ldots\}$. A *ranked alphabet* is a pair (Γ, rk_Γ), often abbreviated by Γ, where Γ is a finite set and $rk_\Gamma \colon \Gamma \to \mathbb{N}$. For every $m \geq 0$ we define $\Gamma^{(m)} = rk_\Gamma^{-1}(m)$ as the set of all symbols of rank m. The rank $rk(\Gamma)$ of Γ is defined as $\max\{rk_\Gamma(a) \mid a \in \Gamma\}$. The set of (*finite, labeled and ordered*) Γ-*trees*, denoted by T_Γ, is the smallest subset T of $(\Gamma \cup \{(,)\} \cup \{,\})^*$ such that if $a \in \Gamma^{(m)}$ with $m \geq 0$ and $s_1, \ldots, s_m \in T$, then $a(s_1, \ldots, s_m) \in T$. In case $m = 0$, we identify $a()$ with a.

We define the set of *positions in a tree* by means of the mapping $\mathrm{pos} \colon T_\Gamma \to \mathcal{P}(\mathbb{N}^*)$ inductively as follows: (i) if $t \in \Gamma^{(0)}$, then $\mathrm{pos}(t) = \{\varepsilon\}$, and (ii) if

$t = a(s_1, \ldots, s_m)$ where $a \in \Gamma^{(m)}$, $m \geq 1$ and $s_1, \ldots, s_m \in T_\Gamma$, then $\mathrm{pos}(t) = \{\varepsilon\} \cup \{iv \mid 1 \leq i \leq m,\ v \in \mathrm{pos}(s_i)\}$. Note that $\mathrm{pos}(t)$ is partially ordered by the prefix relation $\leq_p$ and totally ordered with respect to the lexicographic ordering $\leq_l$. We also refer to the elements of $\mathrm{pos}(t)$ as *nodes*, to ε as the *root* of t and to prefix-maximal nodes as *leaves*.

Now let $t, s \in T_\Gamma$, $w \in \mathrm{pos}(t)$ and $t = a(s_1, \ldots, s_m)$ for some $a \in \Gamma^{(m)}$ with $m \geq 0$ and $s_1, \ldots, s_m \in T_\Gamma$. The *label of t at w* and the *subtree of t at w*, denoted by $t(w)$ and $t|_w$, respectively, are defined inductively as follows: $t(\varepsilon) = a$ and $t|_\varepsilon = t$, and if $w = iv$ and $1 \leq i \leq m$, then $t(w) = s_i(v)$ and $t|_w = s_i|_v$.

A *commutative semiring* is a tuple $(K, \oplus, \odot, \mathbb{0}, \mathbb{1})$, abbreviated by K, with operations sum $\oplus$ and product $\odot$ and constants $\mathbb{0}$ and $\mathbb{1}$ such that $(K, \oplus, \mathbb{0})$ and $(K, \odot, \mathbb{1})$ are commutative monoids, multiplication distributes over addition, and $k \odot \mathbb{0} = \mathbb{0} \odot k = \mathbb{0}$ for every $k \in K$. In this paper, we only consider commutative semirings. Important examples of semirings are

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

A *(formal) tree series* is a mapping $S \colon T_\Gamma \to K$. The set of all tree series (over Γ and K) is denoted by $K\langle\langle T_\Gamma \rangle\rangle$. For two tree series $S, T \in K\langle\langle T_\Gamma \rangle\rangle$ and $k \in K$, the sum $S \oplus T$, the *Hadamard product* $S \odot T$, and the product $k \odot S$ are each defined pointwise.

Let $(K, \oplus, \odot, \mathbb{0}, \mathbb{1})$ be a commutative semiring. A *weighted bottom-up finite state tree automaton (short: WTA) over K and Γ* is a tuple $\mathcal{A} = (Q, \Gamma, \mu, \gamma)$ where Q is a finite set (of states), Γ is a ranked alphabet (of input symbols), $\mu \colon \bigcup_{m=0}^{rk(\Gamma)} Q^m \times \Gamma^{(m)} \times Q \to K$ (the weight function) and $\gamma \colon Q \to K$ (the function of final weights). We set $\Delta_\mathcal{A} = \bigcup_{m=0}^{rk(\Gamma)} Q^m \times \Gamma^{(m)} \times Q$. A tuple $(\vec{p}, a, q) \in \Delta_\mathcal{A}$ is called a *transition* and $(\vec{p}, a, q)$ is called *valid* if $\mu(\vec{p}, a, q) \neq \mathbb{0}$. The state q is referred to as the *parent state* of the transition and the states from $\vec{p}$ are referred to as the *child states* of the transition. A state $q \in Q$ is called *final* if $\gamma(q) \neq \mathbb{0}$.

A mapping $r \colon pos(t) \to Q$ is called a *quasi-run of $\mathcal{A}$ on t*. For $t \in T_\Gamma$, a quasi-run r and $w \in \mathrm{pos}(t)$ with $t(w) = a \in \Gamma^{(m)}$, the tuple

$$\mathbb{t}(r, w) = (r(w1), \ldots, r(wm), a, r(w))$$

is called the *transition with base point w* or *transition at w*. The quasi-run r is called a *(valid) run* if for every $w \in \mathrm{pos}(t)$ the transition $\mathbb{t}(r, w)$ is valid with respect to $\mathcal{A}$. We call a run r *accepting* if $r(\varepsilon)$ is final. If $r(\varepsilon) = q$ then a run r is also called a *q-run*. By $\mathrm{Run}_\mathcal{A}(t)$, $\mathrm{Run}_{\mathcal{A},q}(t)$, $\mathrm{Run}_{\mathcal{A},\mathbb{F}}(t)$ we denote the sets of all runs, all q-runs and all accepting runs of $\mathcal{A}$ on t, respectively.

For $r \in \mathrm{Run}_{\mathcal{A}}(t)$ the *weight of* r is defined by

$$\mathrm{wt}_{\mathcal{A}}(t,r) = \bigodot_{w \in \mathrm{pos}(t)} \mu(\mathbb{t}(r,w)).$$

The *tree series accepted by* $\mathcal{A}$, denoted by $[\![\mathcal{A}]\!] \in K\langle\!\langle T_\Gamma \rangle\!\rangle$, is the tree series defined for every $t \in T_\Gamma$ by $[\![\mathcal{A}]\!](t) = \bigoplus_{r \in \mathrm{Run}_{\mathcal{A},\mathbb{F}}(t)} \mathrm{wt}_{\mathcal{A}}(t,r) \odot \gamma(r(\varepsilon))$ where the sum over the empty set is $\mathbb{0}$ by convention.

An automaton $\mathcal{A}$ is called *trim* if (i) for every $q \in Q$ there exist $t \in T_\Gamma$, $r \in \mathrm{Run}_{\mathcal{A},\mathbb{F}}(t)$ and $w \in \mathrm{pos}(t)$ such that $q = r(w)$ and (ii) for every valid $d \in \Delta_{\mathcal{A}}$ there exist $t \in T_\Gamma$, $r \in \mathrm{Run}_{\mathcal{A},\mathbb{F}}(t)$ and $w \in \mathrm{pos}(t)$ such that $d = \mathbb{t}(r,w)$. The *trim part of* $\mathcal{A}$ is the automaton obtained by removing all states $q \in Q$ which do not satisfy (i) and setting $\mu(d) = \mathbb{0}$ for all valid $d \in \Delta_{\mathcal{A}}$ which do not satisfy (ii). This process obviously has no influence on $[\![\mathcal{A}]\!]$.

An automaton $\mathcal{A}$ is called *deterministic* if for every $m \geq 0$, $a \in \Gamma^{(m)}$ and $\vec{p} \in Q^m$ there exists at most one $q \in Q$ with $\mu(\vec{p}, a, q) \neq \mathbb{0}$. We call $\mathcal{A}$ *(k-)polynomially ambiguous* if $|\mathrm{Run}_{\mathcal{A},\mathbb{F}}(t)| \leq P(|\mathrm{pos}(t)|)$ for some polynomial P (of degree k) and every $t \in T_\Gamma$. If P can be chosen constant, i.e. $P \equiv m$, we call $\mathcal{A}$ *finitely ambiguous* or *m-ambiguous*. If we can put $P \equiv 1$, we call $\mathcal{A}$ *unambiguous*.

Example 1. We consider the alphabet $\Gamma = \{a, b\}$ where $rk_\Gamma(a) = 2$ and $rk_\Gamma(b) = 0$. Over the tropical semiring $(\mathbb{N} \cup \{\infty\}, \min, +, \infty, 0)$ we construct a WTA $\mathcal{A} = (Q, \Gamma, \mu, \gamma)$ with the following idea in mind. Given a tree $t \in T_\Gamma$, there should be exactly one run of $\mathcal{A}$ on t for every leaf b in t, given by mapping all nodes between this leaf and the root to a state q and all other nodes to a filler state p.

We let $Q = \{p, q\}$ and set $\gamma(q) = 0$, $\gamma(p) = \infty$,

$$\begin{aligned} 1 &= \mu(p,q,a,q) = \mu(q,p,a,q) \\ 0 &= \mu(p,p,a,p) = \mu(b,p) = \mu(b,q) \end{aligned}$$

and all other weights to ∞. It is easy to see that this automaton assigns to every tree the minimum amount of a's we have to visit to reach any leaf b starting from the root. As there is a bijection between the runs of $\mathcal{A}$ on a tree t and the leaves of t, $\mathcal{A}$ is polynomially ambiguous, but not finitely ambiguous.

3 Finite Ambiguity

We come to our first main result, namely that a finitely ambiguous WTA can be written as a sum of unambiguous WTA.

Theorem 2. *Let $\mathcal{A} = (Q, \Gamma, \mu, \gamma)$ be a finitely ambiguous weighted bottom-up finite state tree automaton. Then there exist finitely many unambiguous weighted bottom-up finite state tree automata $\mathcal{A}_1, \ldots, \mathcal{A}_n$ satisfying*

$$[\![\mathcal{A}]\!] = [\![\mathcal{A}_1]\!] \oplus \ldots \oplus [\![\mathcal{A}_n]\!].$$

While the basic idea for the proof is taken from [15, Section 4], we have to follow a different line of argumentation due to the non-linear structure of trees. In the first step, we add a deterministic coordinate to our automaton . On the transitions of this new automaton we then define an equivalence relation. Here, two transitions will be equivalent in the following sense. If a run r on a tree t has transition d at some position w, then for every transition d' equivalent to d we can modify r on the subtree at w such that we obtain a new run with transition d' at w. It follows from this that every transition whose equivalence class contains at least two transitions can not occur more than m times in any single run, if $\mathcal{A}$ is m-ambiguous. This contrasts to the word case, where such transitions could occur at most once per run instead of at most m times. For two different runs on the same tree, sorting all transitions occurring in each run first by equivalence class and then lexicographically, shows a difference for at least one equivalence class. This property is the key to the decomposition.

For Γ and m fixed, the number n is exponential in the number of states.

4 Polynomial Ambiguity

We now come to the tree series definable by polynomially ambiguous WTA. Given a polynomially ambiguous WTA $\mathcal{A}$ we define the function $\mathbb{r}_{\mathcal{A}} \colon \mathbb{N} \to \mathbb{N}$ that counts the maximum number of possible runs for all trees with a limited number of nodes, i.e. $\mathbb{r}_{\mathcal{A}}(n) = \max\{|\mathrm{Run}_{\mathcal{A},\mathbb{F}}(t)| \mid t \in T_\Gamma, |\mathrm{pos}(t)| \leq n\}$. We then define the *degree of polynomial ambiguity of* $\mathcal{A}$ by

$$\begin{aligned}\mathrm{degree}(\mathcal{A}) &= \min\{k \in \mathbb{N} \mid \mathcal{A} \text{ is } k\text{-polynomially ambiguous}\}\\ &= \min\{k \in \mathbb{N} \mid \mathbb{r}_{\mathcal{A}} \in \mathcal{O}(n^k)\}.\end{aligned}$$

This is well defined if $\mathcal{A}$ is polynomially ambiguous.

We will show that the runs of a polynomially ambiguous WTA have a very characteristic structure. Consequently, this structure naturally induces a sort of standard form for polynomially ambiguous WTA. For automata in this standard form it is then much easier to grasp the fundamental principle of polynomial ambiguity for tree automata. A first basic tool we will need for all of this is a form of reachability between states. The second is the degree of a state. For notational purposes we also need a more elaborate concept for runs.

4.1 General Definitions and Observations

For now let $\mathcal{A} = (Q, \Gamma, \mu, \gamma)$ be a polynomially ambiguous WTA. The sets $\mathrm{Run}_{\mathcal{A}}(t; \vec{w}, \vec{q})$ and $\mathrm{Run}_{\mathcal{A}}(t; \vec{w}, \vec{d})$ shall denote the sets of all runs of $\mathcal{A}$ on a tree t such that at the positions $w_1, \ldots, w_n$ we have the states $q_1, \ldots, q_n$ or transitions $d_1, \ldots, d_n$, respectively.

Definition 3. *Let $t \in T_\Gamma$, $\vec{w} = (w_1, \ldots, w_n) \in \mathrm{pos}(t)^n$, $\vec{q} = (q_1, \ldots, q_n) \in Q^n$ and $\vec{d} = (d_1, \ldots, d_n) \in \Delta_{\mathcal{A}}^n$. Then we let*

$$\mathrm{Run}_{\mathcal{A}}(t; \vec{w}, \vec{q}) = \{r \in \mathrm{Run}_{\mathcal{A}}(t) \mid r(w_i) = q_i \text{ for all } i = 1, \ldots, n\}$$

$$\mathrm{Run}_{\mathcal{A}}(t; \vec{w}, \vec{d}) = \{r \in \mathrm{Run}_{\mathcal{A}}(t) \mid \mathbb{t}(r, w_i) = d_i \textit{ for all } i = 1, \ldots, n\}.$$

The sets $\mathrm{Run}_{\mathcal{A},\mathbb{F}}(t; \vec{w}, \vec{q})$, $\mathrm{Run}_{\mathcal{A},q}(t; \vec{w}, \vec{q})$, $\mathrm{Run}_{\mathcal{A},\mathbb{F}}(t; \vec{w}, \vec{d})$ and $\mathrm{Run}_{\mathcal{A},q}(t; \vec{w}, \vec{d})$ for $q \in Q$ are defined in a similar manner to these and $\mathrm{Run}_{\mathcal{A},\mathbb{F}}(t)$ and $\mathrm{Run}_{\mathcal{A},q}(t)$.

We define the concept of reachability through a relation $\preccurlyeq$. Intuitively, $q_1 \preccurlyeq q_2$ means that there is a "path" from q_1 down to q_2.

Definition 4. *We define two relations $\preccurlyeq$ and $\approx$ on Q by letting*

$$q_1 \preccurlyeq q_2 \Leftrightarrow \exists t \in T_\Gamma \ \exists w \in \mathrm{pos}(t) : \mathrm{Run}_{\mathcal{A},q_1}(t; w, q_2) \neq \emptyset$$
$$q_1 \approx q_2 \Leftrightarrow q_1 \preccurlyeq q_2 \wedge q_2 \preccurlyeq q_1.$$

The relation $\preccurlyeq$ is reflexive and transitive. Hence, the relation $\approx$ is an equivalence relation inducing equivalence classes $[q]_\approx \in Q_{/\approx}$. One may think of the classes as strongly connected components of states. We set $\mathfrak{C}(q) = [q]_\approx$ and $\mathfrak{Q} = Q_{/\approx}$ and refer to $\mathfrak{C}(q)$ as the *component of* q and to $\mathfrak{Q}$ as the *components of* Q. Then again, $\preccurlyeq$ induces a partial order $\preccurlyeq$ on $\mathfrak{Q}$, defined by $\mathfrak{C}(q_1) \preccurlyeq \mathfrak{C}(q_2) \Leftrightarrow q_1 \preccurlyeq q_2$.

We also need the notion of a *bridge*, similar to the one used in [24]. A bridge is basically a transition which, from a top-down perspective, leaves a component of Q.

Definition 5. *A valid transition $\mathfrak{b} = (p_1, \ldots, p_m, a, q) \in \Delta_{\mathcal{A}}$ is called a* bridge out of $\mathfrak{C}(q)$ *if $\mathfrak{C}(p_i) \neq \mathfrak{C}(q)$ for all $i \in \{1, \ldots, m\}$. Notice that all valid transitions of the form (a, q) with $a \in \Gamma^{(0)}$ and $q \in Q$ are bridges.*

We now define the degree of a state as the degree of the automaton resulting, intuitively, from making this state the only new final state of $\mathcal{A} = (Q, \Gamma, \mu, \gamma)$.

Definition 6. *For every $p \in Q$ we define the WTA $\mathcal{F}_p = (Q, \Gamma, \mu, \gamma_p)$ with $\gamma_p(p) = \mathbb{1}$ and $\gamma_p(q) = \mathbb{0}$ for $q \in Q$, $q \neq p$.*

The intuition is that for $t \in T_\Gamma$ the accepting runs of the automaton $\mathcal{F}_p$ on t are exactly the p-runs of $\mathcal{A}$ on t, i.e. the ones that "begin" with p at the root.

Definition 7. *For a state $p \in Q$ we define* $\mathrm{degree}_{\mathcal{A}}(p) = \mathrm{degree}(\mathcal{F}_p)$ *and we define* $\mathrm{degree}_{\mathcal{A}}(\mathfrak{C}(p)) = \mathrm{degree}_{\mathcal{A}}(p)$. *We will simply write* $\mathrm{degree}(p)$ *and* $\mathrm{degree}(\mathfrak{C}(p))$ *if the automaton $\mathcal{A}$ considered is clear from the context.*

This is well defined, as for $p \approx q$ one can show that $\mathrm{degree}_{\mathcal{A}}(p) = \mathrm{degree}_{\mathcal{A}}(q)$.

It is now easy to show that every valid transition with a parent state q of degree greater than 0 is either (i) a bridge or (ii) exactly one child state belongs to the component of q and all other child states have degree 0. Applying this to a given run r on a tree $t \in T_\Gamma$, we see that states of degree greater than 0 follow branches in the tree. More formally, for $w \in \mathrm{pos}(t)$ with $\mathrm{degree}(t(w)) > 0$ we have $\{v \in \mathrm{pos}(t) \mid w \leq_p v \wedge r(v) \approx r(w)\} = \{v \in \mathrm{pos}(t) \mid w \leq_p v \leq_p w'\}$ for some $w' \in \mathrm{pos}(t)$.

However, for a given component $\mathfrak{c} \in \mathfrak{Q}$ it may still be possible to find a tree t and a run r on t where for two prefix independent positions w and w' we have

$r(w) \in \mathfrak{c}$ and $r(w') \in \mathfrak{c}$, or where $r(w) \in \mathfrak{c}$ holds for no position w. For a WTA in *polynomial standard form*, both of these possibilities will be ruled out: for every component $\mathfrak{c}$ it holds that $\{v \in \mathrm{pos}(t) \mid r(v) \in \mathfrak{c}\} = \{v \in \mathrm{pos}(t) \mid w_1 \leq_p v \leq_p w_2\}$ for some $w_1, w_2 \in \mathrm{pos}(t)$, and this set is non-empty.

4.2 Decomposition into a Sum of Standardized Automata

Definition 8. *We call a (polynomially ambiguous) WTA $\mathcal{A} = (Q, \Gamma, \mu, \gamma)$* standardized *or say it is in* polynomial standard form *if*

(i) *$\mathcal{A}$ is polynomially ambiguous, trim and possesses only one final state $q_f \in Q$ and*

(ii) *for every $p \in Q$ with $\mathrm{degree}_{\mathcal{A}}(p) > 0$ there is exactly one bridge out of $\mathfrak{C}(p)$ and this bridge occurs exactly once in every accepting run r. Formally*

$$\{d \in \Delta_{\mathcal{A}} \mid d \textit{ is a bridge out of } \mathfrak{C}(p)\} = \{\mathfrak{b}(p)\}$$

for some $\mathfrak{b}(p) \in \Delta_{\mathcal{A}}$ and

$$\forall t \in T_\Gamma \ \forall r \in \mathrm{Run}_{\mathcal{A},\mathbb{F}}(t) : |\{w \in \mathrm{pos}(t) \mid \mathfrak{t}(r, w) = \mathfrak{b}(p)\}| = 1.$$

The fundamental concept of standardized WTA is close to the notion of *chain NFAs* as introduced in [24].

Theorem 9. *Let $\mathcal{A} = (Q, \Gamma, \mu, \gamma)$ be a polynomially ambiguous WTA. Then there exist $n \in \mathbb{N}$ and WTA $\mathcal{A}_1, \ldots, \mathcal{A}_n$ in polynomial standard form such that* $\mathrm{degree}(\mathcal{A}_i) \leq \mathrm{degree}(\mathcal{A})$ *for all $i \in \{1, \ldots, n\}$ and*

$$[\![\mathcal{A}]\!] = \bigoplus_{i=1}^{n} [\![\mathcal{A}_i]\!].$$

For a fixed alphabet the number n of automata needed for this is double exponential in the number of states.

Example 10. The WTA from Example 1 is in polynomial standard form. There are two components, $\{p\}$ and $\{q\}$, and the transitions (b, p) and (b, q) are the only bridges. We have $\mathrm{degree}(p) = 0$ and $\mathrm{degree}(q) = 1$.

Proof. (sketch) The theorem is proved in two steps. In the first, we add an entry containing words of bounded length over $\{1, \ldots, rk(\Gamma)\}$ to the states of $\mathcal{A}$. For any such word u and any bridge $(p_1, \ldots, p_m, a, p)$ in $\mathcal{A}$, we will then have a transition $((p_1, u1), \ldots, (p_m, um), a, (p, u))$ in the new automaton $\mathcal{A}'$. For the other transitions, we do the same with the difference that the child states will contain the same word as the parent state.

In the second step, we make copies of $\mathcal{A}'$ and "remove" bridges in the copies appropriately, i.e. we leave at most one bridge out of each component and then trim the automata. The modified copies then fulfill Theorem 9.

4.3 Analysis of the Polynomial Standard Form

Now assume a WTA $\mathcal{A} = (Q, \Gamma, \mu, \gamma)$ in polynomial standard form. We can show that there exist degree($\mathcal{A}$) many bridges in $\mathcal{A}$ such that, given any tree, the number of runs on that tree is bounded by a constant if we fix the position of these bridges. The constant does not depend on the given tree. This property gives a rather intuitive understanding of what polynomial ambiguity means: if our automaton has degree n, then fixing the positions of n predetermined transitions will determine every run up to a bounded number of possibilities.

We consider the set Λ of all bridges that leave components of non-trivial degree, defined as follows.

Definition 11. *Fix $p \in Q$ with $\mathrm{degree}_{\mathcal{A}}(p) > 0$. As there is exactly one bridge $\mathfrak{b} \in \Delta_{\mathcal{A}}$ out of $\mathfrak{C}(p)$ we define $\mathfrak{b}(\mathfrak{C}(p)) = \mathfrak{b}$ and $\mathfrak{b}(p) = \mathfrak{b}$ as this bridge. We set $\Lambda = \{\mathfrak{b}(q) \mid q \in Q,\ \mathrm{degree}_{\mathcal{A}}(q) > 0\}$.*

The degree inherent to an automaton in standard form can now be captured in the following way.

Theorem 12. *Let $p \in Q$ with $l = \mathrm{degree}_{\mathcal{A}}(p) \geq 0$.*

(I) *There exists a set $\mathfrak{N}(p) = \{\mathfrak{b}_1, \dots, \mathfrak{b}_l\} \subseteq \Lambda$ and a constant $C > 0$ such that for all $t \in T_\Gamma$ and $w_1, \dots, w_l \in \mathrm{pos}(t)$ we have*

$$|\mathrm{Run}_{\mathcal{A},p}(t; w_1, \dots, w_l, \mathfrak{b}_1, \dots, \mathfrak{b}_l)| \leq C.$$

(II) *Furthermore there exists a sequence of trees $(t_n)_{n \in \mathbb{N}}$ in T_Γ and a constant $C' > 0$ such that for all $n \in \mathbb{N}$:*
- $|\mathrm{pos}(t_n)| \leq C' \cdot n$ *and*
- $|\mathrm{Run}_{\mathcal{A},p}(t_n)| \geq n^l$.

That is, we can show that if the WTA $\mathcal{F}_p$ (cf. Definition 6) is of degree l, then for all trees the runs of $\mathcal{F}_p$ on those trees are determined up to a constant C by fixing the location of l bridges. Furthermore, the degree of $\mathcal{F}_p$ is not only an upper bound on the amount of runs for a given tree, but also a lower bound. By considering this theorem for the only final state of $\mathcal{A}$, we easily see that it is true for the whole automaton $\mathcal{A}$ as well.

Example 13. In the WTA from Example 1 we have $\mathfrak{N}(p) = \emptyset$ and $\mathfrak{N}(q) = \{(b, q)\}$. By choosing which leaf to "mark" with q, we uniquely determine a run. Therefore, (I) clearly holds for both p and q in this automaton.

For (II) in the case of q we consider the trees $t_0 = b()$ and $t_{n+1} = a(t_n, b)$ for $n \geq 0$. We have $|\mathrm{pos}(t_n)| = 2n + 1$ and one run for every leaf in a tree, i.e. $|\mathrm{Run}_{\mathcal{A},q}(t_n)| = n + 1$.

As a corollary of Theorem 12 we also get that the ambiguity of a WTA $\mathcal{A}$ is either bounded below and above by a fixed polynomial or has a lower exponential bound. While this is a well known result for word automata [24], we could not find a similar result for tree automata in the literature.

Corollary 14. *Let $\mathcal{A} = (Q, \Gamma, \mu, \gamma)$ be a weighted bottom-up finite state tree automaton. Either $\mathcal{A}$ is polynomially ambiguous and $\mathbb{r}_{\mathcal{A}} \in \Theta(n^k)$ for $k = \mathrm{degree}(\mathcal{A})$ or there exists a sequence of trees $(t_n)_{n\in\mathbb{N}}$ in T_Γ and a constant $C > 0$ such that for all $n \in \mathbb{N}$* ***(i)*** *$|\mathrm{pos}(t_n)| \leq C \cdot n$ and* ***(ii)*** *$|\mathrm{Run}_{\mathcal{A},\mathbb{F}}(t_n)| \geq 2^n$.*

5 Application: Weighted Logics

As stated in the introduction, our investigations were part of the attempt to characterize weighted tree automata with weighted logics. Therefore, we briefly outline how our weighted logic works, which results we obtained with it and what the significance of our investigations is to these results. For further details see [8,10,20].

The standard MSO-logic for trees is given by the following grammar.

$$\varphi ::= \mathrm{label}_a(x) \mid \mathrm{edge}_i(x,y) \mid x \in X \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x.\varphi \mid \exists X.\varphi$$

where $a \in \Gamma$, x, y are first order variables, $1 \leq i \leq rk(\Gamma)$, and X is a second order variable. The set of free variables of φ is denoted by $\mathrm{Free}(\varphi)$. Let $t \in T_\Gamma$ be a tree and $\mathcal{V}$ be a set of first order and second order variables with $\mathrm{Free}(\varphi) \subseteq \mathcal{V}$. A mapping which assigns to every first order variable $x \in \mathcal{V}$ a position $w \in \mathrm{pos}(t)$ and to every second order variable $X \in \mathcal{V}$ a set of positions $I \subseteq \mathrm{pos}(t)$ is called a $(\mathcal{V}, t)$-*assignment.* For a first order variable $x \in \mathcal{V}$ and a position $w \in \mathrm{pos}(t)$ we write $\rho[x \to w]$ to denote the $(\mathcal{V} \cup \{x\}, t)$-assignment given by $\rho[x \to w](x) = w$ and $\rho[x \to w](y) = \rho(y)$ for all variables $y \neq x$. The assignment $\rho[X \to I]$, where X is a second-order variable and $I \subseteq \mathrm{pos}(t)$, is defined analogously. We write $(t, \rho) \models \varphi$ if (t, ρ) satisfies φ using standard MSO-semantics. We then have the generalization of Büchi's and Elgot's fundamental theorems [5,11] to trees, namely that MSO-definable tree languages are exactly the recognizable tree languages [25,7].

On top of the MSO-logic we construct a weighted logic, called wMSO-logic, with the following grammar.

$$\theta ::= \varphi \mid k \mid \theta \oplus \theta \mid \theta \odot \theta \mid \Sigma x.\theta \mid \Sigma X.\theta \mid \Pi x.\theta$$

where $\varphi \in \mathrm{MSO}(\Gamma)$, $k \in K$, x is a first order variable and X is a second order variable. The operators Σx and ΣX are referred to as first order sum quantification and second order sum quantification, respectively, and Πx is referred to as (first order) product quantification. Moreover, the operators Σx, ΣX and Πx also bind the variables x and X, respectively. A wMSO-formula without free variables is also called a *sentence.*

For a formula $\theta \in \mathrm{wMSO}(\Gamma)$, a tree $t \in T_\Gamma$, a set $\mathcal{V}$ of first and second order variables with $\mathrm{Free}(\theta) \subseteq \mathcal{V}$ and a $(\mathcal{V}, t)$-assignment ρ we define the value $[\![\theta]\!](t, \rho)$ inductively in the following way.

$$[\![\theta]\!](t,\rho) = \begin{cases} \mathbb{1} & \text{if } (t,\rho) \models \theta \\ \mathbb{0} & \text{otherwise} \end{cases} \quad \text{for } \theta \in \mathrm{MSO}(\Gamma)$$

$$\begin{aligned}
[\![k]\!](t,\rho) &= k \\
[\![\theta_1 \oplus \theta_2]\!](t,\rho) &= [\![\theta_1]\!](t,\rho) \oplus [\![\theta_2]\!](t,\rho) \\
[\![\theta_1 \odot \theta_2]\!](t,\rho) &= [\![\theta_1]\!](t,\rho) \odot [\![\theta_2]\!](t,\rho) \\
[\![\Sigma x.\tau]\!](t,\rho) &= \bigoplus_{w \in \mathrm{pos}(t)} [\![\tau]\!](t,\rho[x \to w]) \\
[\![\Pi x.\tau]\!](t,\rho) &= \bigodot_{w \in \mathrm{pos}(t)} [\![\tau]\!](t,\rho[x \to w]) \\
[\![\Sigma X.\tau]\!](t,\rho) &= \bigoplus_{I \subseteq \mathrm{pos}(t)} [\![\tau]\!](t,\rho[X \to I])
\end{aligned}$$

where $k \in K$ and $\theta_1, \theta_2, \tau \in \mathrm{wMSO}(\Gamma)$.

Example 15. We consider the semiring $(\mathbb{N}, +, \cdot, 0, 1)$ and the alphabet $\Gamma = \{a, b\}$ where $rk_\Gamma(a) = 2$ and $rk_\Gamma(b) = 0$. The following formula outputs for every $t \in T_\Gamma$ the amount of a's taking two b's as child nodes.

$$\Sigma x. \Big(\mathrm{label}_a(x) \wedge \exists y. \big(\mathrm{edge}_1(x,y) \wedge \mathrm{label}_b(y)\big) \wedge \exists y. \big(\mathrm{edge}_2(x,y) \wedge \mathrm{label}_b(y)\big)\Big)$$

In order to characterize different degrees of ambiguity, we use restrictions of above logic. The formulas given by the grammar

$$\theta_b ::= \varphi \mid k \mid \theta_b \oplus \theta_b \mid \theta_b \odot \theta_b$$

with $\varphi \in \mathrm{MSO}(\Gamma)$ and $k \in K$ are called *almost boolean* and define so-called *recognizable step functions* [8,10]. We call a formula *unambiguous* if it is almost boolean, a product quantifier followed by an almost boolean formula or a finite product of such formulas. A formula containing no sum quantifiers, and in which for every subformula $\Pi.\theta$ the formula θ is almost boolean, is called *finitely ambiguous*. This class of formulas is actually the closure of unambiguous formulas under $\oplus$ and $\odot$. Finally, a formula is called *polynomially ambiguous* if it does not contain second order sum quantification and for every subformula $\Pi.\theta$ the formula θ is almost boolean. We have the following theorem.

Theorem 16. *The following classes of automata and sets of sentences are expressively equivalent:*

(a) *unambiguous WTA and unambiguous sentences*
(b) *finitely ambiguous WTA and finitely ambiguous sentences*
(c) *polynomially ambiguous WTA of polynomial degree k and polynomially ambiguous formulas with first order sum quantifier depth k.*

Example 17. The WTA from Example 1, calculating the minimum amount of a's between the root and any leaf, is described by the formula

$$\Sigma x. \Pi y. \left(\mathrm{label}_b(x) \odot ((1 \odot (\mathrm{label}_a(y) \wedge y \leq_p x)) \oplus \neg(\mathrm{label}_a(y) \wedge y \leq_p x))\right).$$

The prefix relation is MSO-definable [6, Section 3.3].

A result similar to Theorem 16 has been shown by Kreutzer and Riveros [16]to hold true for weighted automata over words. Most of their proofs can easily be adapted to work for tree automata, but not all. To be precise, we need the results of Sections 3 and 4 for the translation of automata to logics in (b) and (c). For polynomial ambiguity, we even obtain a stronger result, as we are able to capture polynomial degree not only in the boolean semiring, but in any commutative semiring. For this, we show by induction on the polynomial degree that for a WTA in polynomial standard form, first order sum quantifiers can be used to sum over all possible positions for the bridges identified in Theorem 12 (I). Having specified the positions of all these bridges, we are then essentially in the case of finite ambiguity, and can apply (b). The number of first order sum quantifiers needed to describe the WTA with a wMSO-formula hence equals its polynomial degree.

6 Conclusion

As shown, our results about the structure of weighted tree automata have proven to be useful in the context of weighted logics for trees. Two questions now arise. First, which other problems could be tackled with the newly gained knowledge? Decidability problems for WTA are an obvious candidate here. Second, can we get similar results for other automata models? For example, one might intuitively assume picture automata and graph acceptors to behave in a similar manner, but this is in no way obvious and calls for further investigation.

Acknowledgements. I would like to thank Professor Manfred Droste, Peter Leupold and Vitaly Perevoshchikov for helpful discussions and suggestions.

References

1. Allauzen, C., Mohri, M., Rastogi, A.: General algorithms for testing the ambiguity of finite automata. In: Ito, M., Toyama, M. (eds.) Developments in Language Theory, Lect. Notes Comput. Sc., vol. 5257, pp. 108–120. Springer Berlin Heidelberg (2008)
2. Berstel, J., Reutenauer, C.: Recognizable formal power series on trees. Theoretical Computer Science 18(2), 115 – 148 (1982)
3. Berstel, J., Reutenauer, C.: Rational Series and Their Languages, Monogr. Theoret. Comput. Sci. EATCS Ser., vol. 12. Springer Berlin Heidelberg (1988)
4. Bozapalidis, S.: Effective construction of the syntactic algebra of a recognizable series on trees. Acta Informatica 28(4), 351–363 (1991)
5. Büchi, J.: Weak second-order arithmetic and finite automata. Z. Math. Logik und Grundl. Math. 6, 66–92 (1960)
6. Comon, H., Dauchet, M., Gilleron, R., Löding, C., Jacquemard, F., Lugiez, D., Tison, S., Tommasi, M.: Tree automata techniques and applications. Available on: `http://www.grappa.univ-lille3.fr/tata` (2007)
7. Doner, J.: Tree acceptors and some of their applications. J. Comput. Syst. Sci. 4(5), 406–451 (1970)
8. Droste, M., Gastin, P.: Weighted automata and weighted logics. Theor. Comput. Sci. 380(1-2), 69–86 (2007)

9. Droste, M., Kuich, W., Vogler, H. (eds.): Handbook of Weighted Automata. Monogr. Theoret. Comput. Sci. EATCS Ser., Springer Berlin Heidelberg (2009)
10. Droste, M., Vogler, H.: Weighted tree automata and weighted logics. Theor. Comput. Sci. 366(3), 228–247 (2006)
11. Elgot, C.: Decision problems of finite automata design and related arithmetics. T. Am. Math. Soc. 98, 21–52 (1961)
12. Fülöp, Z., Vogler, H.: Weighted tree automata and tree transducers. In: Droste, M., Kuich, W., Vogler, H. (eds.) Handbook of Weighted Automata, chap. 9, pp. 313–403. Monogr. Theoret. Comput. Sci. EATCS Ser., Springer (2009)
13. Hashiguchi, K., Ishiguro, K.: Decidability of the equivalence problem for finitely ambiguous finance automata. Surikaisekikenkyusho Kokyuroku 960, 23–36 (1996)
14. Kirsten, D.: A Burnside approach to the termination of Mohri's algorithm for polynomially ambiguous min-plus-automata. RAIRO-Inf. Theor. Appl. 42(3), 553–581 (2008)
15. Klimann, I., Lombardy, S., Mairesse, J., Prieur, C.: Deciding unambiguity and sequentiality from a finitely ambiguous max-plus automaton. Theor. Comput. Sci. 327(3), 349–373 (2004)
16. Kreutzer, S., Riveros, C.: Quantitative monadic second-order logic. In: 28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS. pp. 113–122. IEEE Computer Society (2013)
17. Krob, D.: The equality problem for rational series with multiplicities in the tropical semiring is undecidable. Int. J. of Algebr. Comput. 04(03), 405–425 (1994)
18. Kuich, W., Salomaa, A.: Semirings, Automata, Languages, Monogr. Theoret. Comput. Sci. EATCS Ser., vol. 5. Springer Berlin Heidelberg (1986)
19. Maletti, A.: Relating tree series transducers and weighted tree automata. Internat. J. Found. Comput. Sci. 16(4), 723–741 (2005)
20. Rahonis, G.: Weighted Muller tree automata and weighted logics. J. Autom. Lang. Comb. 12(4), 455–483 (2007)
21. Salomaa, A., Soittola, M.: Automata-Theoretic Aspects of Formal Power Series. Texts Monogr. Comput. Sci., Springer New York (1978)
22. Schützenberger, M.: On the definition of a family of automata. Information and Control 4(2–3), 245 – 270 (1961)
23. Seidl, H.: On the finite degree of ambiguity of finite tree automata. Acta Inform. 26(6), 527–542 (1989)
24. Seidl, H., Weber, A.: On the degree of ambiguity of finite automata. Theor. Comput. Sci. 88(2), 325–349 (1991)
25. Thatcher, J., Wright, J.: Generalized finite automata theory with an application to a decision problem of second-order logic. Math. Syst. Theory 2(1), 57–81 (1968)