Finite Sequentiality of Finitely Ambiguous Max-Plus Tree Automata

Erik Paul
Institute of Computer Science, Leipzig University, Germany
epaul@informatik.uni-leipzig.de

Abstract

We show that the finite sequentiality problem is decidable for finitely ambiguous max-plus tree automata. A max-plus tree automaton is a weighted tree automaton over the max-plus semiring. A max-plus tree automaton is called finitely ambiguous if the number of accepting runs on every tree is bounded by a global constant. The finite sequentiality problem asks whether for a given max-plus tree automaton, there exist finitely many deterministic max-plus tree automata whose pointwise maximum is equivalent to the given automaton.

2012 ACM Subject Classification Theory of computation → Quantitative automata; Theory of computation → Tree languages

Keywords and phrases Weighted Tree Automata, Max-Plus Tree Automata, Finite Sequentiality, Decidability, Finite Ambiguity

Digital Object Identifier 10.4230/LIPIcs.ICALP.2020.137

Category Track B: Automata, Logic, Semantics, and Theory of Programming

Funding This work was partially supported by Deutsche Forschungsgemeinschaft (DFG), Graduiertenkolleg 1763 (QuantLA).

1 Introduction

A *max-plus automaton* is a finite automaton whose transitions are weighted by real numbers. A max-plus automaton assigns a weight to each of its runs by adding the weights of the transitions which constitute the run and it assigns a weight to every word by taking the maximum over the weights of all runs on the given word. Max-plus automata are weighted automata [40, 39, 27, 6, 13] over the max-plus semiring. In the form of min-plus automata, they were originally introduced by Imre Simon as a means to show the decidability of the *finite power property* [42, 43] and they enjoy a continuing interest [26, 19, 22, 7, 12, 16, 28]. They have found applications in many different contexts, for example to determine the star height of a language [18], to prove the termination of certain string rewriting systems [44], and to model discrete event systems [23]. They also appear in the context of natural language processing [29], where probabilities are often computed in the min-plus semiring as negative log-likelihoods for reasons of numerical stability.

Like finite automata, max-plus automata are by definition non-deterministic devices. However, while every finite automaton can be determinized [36], the same is in general not true for max-plus automata [22]. Actually, it is a long-standing open question whether given a max-plus automaton, the existence of an equivalent deterministic automaton can be decided. This problem is commonly known as the *sequentiality problem* and is one of the most prominent open questions about max-plus automata. For practical applications, the execution of a deterministic automaton is of course much more efficient than the execution of a non-deterministic one, so being able to decide whether a given automaton can be determinized is very much desirable. While open in general, the sequentiality problem has been shown to be decidable for some important subclasses of max-plus automata, namely for *unambiguous* [29], *finitely ambiguous* [22], and *polynomially ambiguous* [21] max-plus automata. Here, we call a max-plus automaton *unambiguous* if there exists at most one run on every word, *finitely ambiguous* if the number of runs on each word is bounded by a global constant, and *polynomially ambiguous* if the number of runs on each word is bounded polynomially in the length of the word. Note that the classes of deterministic, unambiguous, finitely ambiguous, polynomially ambiguous, and arbitrary max-plus automata form a strictly ascending hierarchy [22, 20, 28]. Also, deciding the degree of ambiguity of a max-plus automaton can easily be reduced to deciding the degree of ambiguity of a finite automaton. It is trivial to decide whether a finite automaton is deterministic. Polynomial time algorithms to decide whether a finite automaton is unambiguous, finitely ambiguous, or polynomially ambiguous can be found in [8, 45, 41].

While a given max-plus automaton may not be equivalent to a single deterministic max-plus automaton, this does not exclude the possibility that it is equivalent to the pointwise maximum of finitely many deterministic automata. The problem of deciding whether a max-plus automaton possesses such a *finitely sequential* representation is known as the *finite sequentiality problem*. The decidability of the finite sequentiality problem was posed as an open question in [19] and has been solved only recently for unambiguous [4] and finitely ambiguous [3] max-plus automata. Note that the class of max-plus automata which possess a finitely sequential representation lies strictly between the classes of deterministic and finitely ambiguous max-plus automata, and it is incomparable to the class of unambiguous max-plus automata [22].

In this paper, we show that the finite sequentiality problem is decidable for finitely ambiguous *max-plus tree automata*. Operating on trees instead of words, max-plus tree automata are a generalization of max-plus word automata and more generally, they are weighted tree automata [1, 5, 14, 17] over the max-plus semiring. Applications of max-plus tree automata include proving the termination of certain term rewriting systems [24] and they are commonly employed in natural language processing [35] in the form of *probabilistic context-free grammars*. Our approach to proving the decidability of the finite sequentiality problem for finitely ambiguous max-plus tree automata employs ideas from Bala's proof of the corresponding result for finitely ambiguous max-plus word automata [3]. However, due to lack of space, formal proofs had to be omitted in [3] and Bala's informal description of his methods does not suffice for reconstruction. Also, no other published version of [3] exists. We provide an honest attempt to compare our approach to his but note that our interpretation might not be accurate.

In his proof for max-plus word automata, Bala first introduces the *A-Fork property* and then proceeds to show that this property is a decidable criterion characterizing the finite sequentiality of a finitely ambiguous max-plus automaton. More precisely, he shows that a finitely ambiguous max-plus automaton possesses a finitely sequential representation if and only if the A-Fork property is not satisfied. To show the decidability of the A-Fork property, he shows its expressibility in a decidable fragment of Presburger arithmetic. To show that an automaton is not finitely sequential if the A-Fork property is satisfied, he uses pumping techniques similar to those employed in [4] for the finite sequentiality problem of unambiguous max-plus word automata. This part of his proof most likely employs Ramsey's Theorem [37] as it involves "colorings of finite hypercubes". His proof for the existence of a finitely sequential representation in case that the A-Fork property is not satisfied employs transducers and the notions of *critical pairs* and *close approximations*, none of which occur in our approach. We are thus unsure about the nature of this particular part of the proof, but it most likely uses a reduction to the decidability of the finite sequentiality problem for unambiguous automata.

Our approach is as follows. First, we introduce the *separation property*, a twofold modification of the A-Fork property. On the one hand, we endow our new property with a criterion accounting for the non-linear structure of trees. This new criterion is inspired by the criterion we added in [34] to the *fork property* [4], the property characterizing finite sequentiality of unambiguous max-plus word automata, in order to obtain the *tree fork property*, the property characterizing finite sequentiality of unambiguous max-plus tree automata. On the other hand, we strengthen the A-Fork property as with only the first modification, our new property would wrongly characterize some finitely sequential automata as not being finitely sequential. We then show that the separation property is decidable by employing Parikh's Theorem [31, 15] for a reduction to the decidability of the satisfiability of systems of linear inequalities over the rational numbers with integer solutions [30, 9]. This means in particular that we show the decidability of the finite sequentiality problem only for automata with weights in the rationals. Then we employ Ramsey's Theorem to show that no finitely sequential representation exists whenever the separation property is satisfied. Due to the criterion accounting for the non-linearity of trees, this is considerably more difficult than in [3] and it is in fact the most technical and the most challenging aspect of our result. Finally, we show that if the separation property is not satisfied for a given max-plus tree automaton, then we can construct finitely many unambiguous max-plus tree automata which all do not satisfy the tree fork property and whose pointwise maximum is equivalent to the automaton. By [34], these unambiguous automata then possess finitely sequential representations. Combining these, we obtain a finitely sequential representation of the original automaton.

2 Preliminaries

For a set X, we denote the power set of X by $\mathcal{P}(X)$ and the cardinality of X by $|X|$. For two sets X and Y and a mapping $f\colon X \to Y$, we call X the *domain* of f, denoted by $\mathrm{dom}(f)$. For a subset $X' \subseteq X$, we call the set $f(X') = \{y \in Y \mid \exists x \in X'\colon f(x) = y\}$ the *image* or *range of* X' *under* f. The *restriction of* f *to* X', denoted by $f{\restriction}_{X'}$, is the mapping $f{\restriction}_{X'}\colon X' \to Y$ defined by $f{\restriction}_{X'}(x) = f(x)$ for every $x \in X'$. For an element $y \in Y$, we call the set $f^{-1}(y) = \{x \in X \mid f(x) = y\}$ the *preimage of* y *under* f. For a second mapping $g\colon X \to Y$, we write $f = g$ if for all $x \in X$ we have $f(x) = g(x)$.

We let $\mathbb{N} = \{0, 1, 2, \dots\}$. By $\mathbb{N}^*$ we denote the set of all finite words over $\mathbb{N}$. The empty word is denoted by ε, and the length of a word $w \in \mathbb{N}^*$ by $|w|$. The set $\mathbb{N}^*$ is partially ordered by the prefix relation $\leq_{\mathrm{P}}$ and totally ordered with respect to the lexicographic ordering $\leq_{\mathrm{L}}$. Two words from $\mathbb{N}^*$ are called *prefix-dependent* if they are in prefix relation, and otherwise they are called *prefix-independent*.

A *ranked alphabet* is a pair $(\Gamma, \mathrm{rk}_\Gamma)$, often abbreviated by Γ, where Γ is a finite set and $\mathrm{rk}_\Gamma\colon \Gamma \to \mathbb{N}$ a mapping which assigns a *rank* to every symbol. For every $m \geq 0$ we define $\Gamma^{(m)} = \mathrm{rk}_\Gamma^{-1}(m)$ as the set of all symbols of rank m. The rank of Γ is defined as $\mathrm{rk}(\Gamma) = \max\{\mathrm{rk}_\Gamma(a) \mid a \in \Gamma\}$. The set of (*finite, labeled, and ordered*) Γ*-trees*, denoted by T_Γ, is the set of all pairs $t = (\mathrm{pos}(t), \mathrm{label}_t)$, where $\mathrm{pos}(t) \subset \mathbb{N}^*$ is a finite non-empty prefix-closed set of *positions*, $\mathrm{label}_t\colon \mathrm{pos}(t) \to \Gamma$ is a mapping, and for every $w \in \mathrm{pos}(t)$ we have $wi \in \mathrm{pos}(t)$ iff $1 \leq i \leq \mathrm{rk}_\Gamma(\mathrm{label}_t(w))$. We write $t(w)$ for $\mathrm{label}_t(w)$ and $|t|$ for $|\mathrm{pos}(t)|$. 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*. The *height* of t is defined as $\mathrm{height}(t) = \max_{w \in \mathrm{pos}(t)} |w|$. For a leaf $w \in \mathrm{pos}(t)$, the set $\{v \in \mathrm{pos}(t) \mid v \leq_{\mathrm{P}} w\}$ is called a *branch* of t.

Now let $s, t \in T_\Gamma$ and $w \in \text{pos}(t)$. The *subtree of t at w*, denoted by $t|_w$, is a Γ-tree defined as follows. We let $\text{pos}(t|_w) = \{v \in \mathbb{N}^* \mid wv \in \text{pos}(t)\}$ and for $v \in \text{pos}(t|_w)$, we let $\text{label}_{t|_w}(v) = t(wv)$. The *substitution of s into w of t*, denoted by $t\langle s \to w\rangle$, is a Γ-tree defined as follows. We let $\text{pos}(t\langle s \to w\rangle) = (\text{pos}(t) \setminus \{v \in \text{pos}(t) \mid w \leq_{\text{P}} v\}) \cup \{wv \mid v \in \text{pos}(s)\}$. For $v \in \text{pos}(t\langle s \to w\rangle)$, we let $\text{label}_{t\langle s\to w\rangle}(v) = s(u)$ if $v = wu$ for some $u \in \text{pos}(s)$, and otherwise $\text{label}_{t\langle s\to w\rangle}(v) = t(v)$. For $a \in \Gamma^{(m)}$ and trees $t_1, \ldots, t_m \in T_\Gamma$, we also write $a(t_1, \ldots, t_m)$ to denote the tree t with $\text{pos}(t) = \{\varepsilon\} \cup \{iw \mid i \in \{1, \ldots, m\}, w \in \text{pos}(t_i)\}$, $\text{label}_t(\varepsilon) = a$, and $\text{label}_t(iw) = t_i(w)$. For $a \in \Gamma^{(0)}$, the tree $a()$ is abbreviated by a.

For a ranked alphabet Γ, a tree over the alphabet $\Gamma_\diamond = (\Gamma \cup \{\diamond\}, \text{rk}_\Gamma \cup \{\diamond \mapsto 0\})$ is called a *Γ-context*. Let $t \in T_{\Gamma_\diamond}$ be a Γ-context and let $w_1, \ldots, w_n \in \text{pos}(t)$ be a lexicographically ordered enumeration of all leaves of t labeled $\diamond$. Then we call t an *n-Γ-context* and define $\Diamond_i(t) = w_i$ for $i \in \{1, \ldots, n\}$. For an n-Γ-context t and contexts $t_1, \ldots, t_n \in T_{\Gamma_\diamond}$, we define $t(t_1, \ldots, t_n) = t\langle t_1 \to \Diamond_1(t)\rangle \cdots \langle t_n \to \Diamond_n(t)\rangle$ by substitution of $t_1, \ldots, t_n$ into the $\diamond$-leaves of t. A 1-Γ-context is also called a *Γ-word*. For a Γ-word s, we define $s^0 = \diamond$ and $s^{n+1} = s(s^n)$ for $n \geq 0$.

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 $\kappa \odot \mathbb{0} = \mathbb{0} \odot \kappa = \mathbb{0}$ for every $\kappa \in K$. In this paper, we mainly consider the following two semirings.

- The *Boolean semiring* $\mathbb{B} = (\{0, 1\}, \vee, \wedge, 0, 1)$ with disjunction $\vee$ and conjunction $\wedge$.
- The *max-plus semiring* $\mathbb{Q}_{\max} = (\mathbb{Q} \cup \{-\infty\}, \max, +, -\infty, 0)$ where the sum and the product operations are max and $+$, respectively, extended to $\mathbb{Q} \cup \{-\infty\}$ in the usual way.

For a commutative semiring $(K, \oplus, \odot, \mathbb{0}, \mathbb{1})$ and an integer $n \geq 1$, the *product semiring* $(K^n, \oplus_n, \odot_n, \mathbb{0}_n, \mathbb{1}_n)$ is defined by componentwise operations and the constants $\mathbb{0}_n = (\mathbb{0}, \ldots, \mathbb{0})$ and $\mathbb{1}_n = (\mathbb{1}, \ldots, \mathbb{1})$. We will usually denote $\oplus_n$ and $\odot_n$ simply by $\oplus$ and $\odot$.

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 $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ where Q is a finite set (of states), Γ is a ranked alphabet (of input symbols), $\mu\colon \bigcup_{m=0}^{\text{rk}(\Gamma)} Q^m \times \Gamma^{(m)} \times Q \to K$ (the function of transition weights), and $\nu\colon Q \to K$ (the function of final weights). We define $\Delta_\mathcal{A} = \text{dom}(\mu)$. A tuple $d \in \Delta_\mathcal{A}$ is called a *transition* and d is called *valid* if $\mu(d) \neq \mathbb{0}$. A state $q \in Q$ is called *final* if $\nu(q) \neq \mathbb{0}$.

We call a WTA over the max-plus semiring a *max-plus-WTA* and a WTA over the Boolean semiring a *finite tree automaton* (FTA). We also write a WTA $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ over $\mathbb{B}$ as a tuple $\mathcal{A}' = (Q, \Gamma, \delta, F)$ where $\delta = \{d \in \Delta_\mathcal{A} \mid \mu(d) = 1\}$ and $F = \{q \in Q \mid \nu(q) = 1\}$.

For a tree $t \in T_\Gamma$, a mapping $r\colon \text{pos}(t) \to Q$ is called a *quasi-run of $\mathcal{A}$ on t*. For a quasi-run r on t and a position $w \in \text{pos}(t)$ with $t(w) = a \in \Gamma^{(m)}$, the tuple $\mathbb{t}(t, r, w) = (r(w1), \ldots, r(wm), a, r(w))$ is called the *transition at w*. The quasi-run r is called a *(valid) run* if for every $w \in \text{pos}(t)$ the transition $\mathbb{t}(t, r, w)$ is valid with respect to $\mathcal{A}$. We call a run r *accepting* if $r(\varepsilon)$ is final. By $\text{Run}_\mathcal{A}(t)$ and $\text{Acc}_\mathcal{A}(t)$ we denote the sets of all runs and all accepting runs of $\mathcal{A}$ on t, respectively. For a state $q \in Q$, we denote by $\text{Run}_\mathcal{A}(t, q)$ the set of all runs $r \in \text{Run}_\mathcal{A}(t)$ such that $r(\varepsilon) = q$.

For a run $r \in \text{Run}_\mathcal{A}(t)$, the *weight of r* is defined by $\text{wt}_\mathcal{A}(t, r) = \bigodot_{w \in \text{pos}(t)} \mu(\mathbb{t}(t, r, w))$. The *behavior of $\mathcal{A}$*, denoted by $[\![\mathcal{A}]\!]$, is the mapping defined for every $t \in T_\Gamma$ by $[\![\mathcal{A}]\!](t) = \bigoplus_{r \in \text{Acc}_\mathcal{A}(t)} (\text{wt}_\mathcal{A}(t, r) \odot \nu(r(\varepsilon)))$, where the sum over the empty set is $\mathbb{0}$ by convention. The *support* of a WTA $\mathcal{A}$ is the set $\text{supp}(\mathcal{A}) = \{t \in T_\Gamma \mid [\![\mathcal{A}]\!](t) \neq \mathbb{0}\}$. The support of an FTA $\mathcal{A}$ is also called the *language accepted by* $\mathcal{A}$ and denoted by $\mathcal{L}(\mathcal{A})$. A subset $L \subseteq T_\Gamma$ is called *recognizable* if there exists an FTA $\mathcal{A}$ with $L = \mathcal{L}(\mathcal{A})$.

For a WTA $\mathcal{A} = (Q, \Gamma, \mu, \nu)$, a run of $\mathcal{A}$ on a Γ-context t is a run of the WTA $\mathcal{A}' = (Q, \Gamma_\diamond, \mu', \nu)$ on t, where $\mu'(\diamond, q) = \mathbb{1}$ for all $q \in Q$ and $\mu'(d) = \mu(d)$ for all $d \in \Delta_\mathcal{A}$. We denote $\mathrm{Run}^\diamond_\mathcal{A}(t) = \mathrm{Run}_{\mathcal{A}'}(t)$ and for $r \in \mathrm{Run}^\diamond_\mathcal{A}(t)$ write $\mathrm{wt}^\diamond_\mathcal{A}(t, r) = \mathrm{wt}_{\mathcal{A}'}(t, r)$. For an n-Γ-context $t \in T_{\Gamma_\diamond}$ and states $q_0, \ldots, q_n$, we denote by $\mathrm{Run}^\diamond_\mathcal{A}(q_1, \ldots, q_n, t, q_0)$ the set of all runs $r \in \mathrm{Run}^\diamond_\mathcal{A}(t)$ such that $r(\varepsilon) = q_0$ and $r(\Diamond_i(t)) = q_i$ for every $i \in \{1, \ldots, n\}$.

We consider the set $\Gamma \times Q$ as an alphabet by defining $\mathrm{rk}_{\Gamma \times Q}(a, q) = \mathrm{rk}_\Gamma(a)$ for every pair $(a, q) \in \Gamma \times Q$ and identify every tree $t' \in T_{\Gamma \times Q}$ with the pair (t, r) given by $t = (\mathrm{pos}(t'), \pi_\Gamma \circ \mathrm{label}_{t'}) \in T_\Gamma$ and $r = \pi_Q \circ \mathrm{label}_{t'}$, where $\pi_\Gamma \colon \Gamma \times Q \to \Gamma$ and $\pi_Q \colon \Gamma \times Q \to Q$ are the projections. For a Γ-word $s \in T_{\Gamma_\diamond}$, a state $q \in Q$, and a run $r_s \in \mathrm{Run}^\diamond_\mathcal{A}(q, s, q)$, we define $(s, r_s)^0 = (\diamond, q)$ and $(s, r_s)^{n+1} = (s, r_s)\langle (s, r_s)^n \to \Diamond_1(s) \rangle$ for $n \geq 0$.

We call a WTA $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ over K and Γ *trim* if for every $p \in Q$, there exist $t \in T_\Gamma$, $r \in \mathrm{Acc}_\mathcal{A}(t)$, and $w \in \mathrm{pos}(t)$ with $r(w) = p$. The *trim part of* $\mathcal{A}$ is the automaton obtained from $\mathcal{A}$ by removing all states $p \in Q$ for which no such t, r, and w exist. This process obviously has no influence on $[\![\mathcal{A}]\!]$. We call $\mathcal{A}$ *complete* if for every $m \geq 0$, $a \in \Gamma^{(m)}$, and $(q_1, \ldots, q_m) \in Q^m$, there exists at least one $q \in Q$ with $\mu(q_1, \ldots, q_m, a, q) \neq \mathbb{0}$. We call $\mathcal{A}$ *deterministic* or *sequential* if for every $m \geq 0$, $a \in \Gamma^{(m)}$, and $(q_1, \ldots, q_m) \in Q^m$, there exists at most one $q \in Q$ with $\mu(q_1, \ldots, q_m, a, q) \neq \mathbb{0}$. If there exists an integer $M \geq 1$ such that $|\mathrm{Acc}_\mathcal{A}(t)| \leq M$ for every $t \in T_\Gamma$, we call $\mathcal{A}$ *M-ambiguous*. We call $\mathcal{A}$ *finitely ambiguous* if it is M-ambiguous for some $M \geq 1$ and *unambiguous* if it is 1-ambiguous. We call the behavior $[\![\mathcal{A}]\!]$ of $\mathcal{A}$ *finitely sequential* if there exist finitely many deterministic WTA $\mathcal{A}_1, \ldots, \mathcal{A}_n$ over K and Γ such that $[\![\mathcal{A}]\!] = \bigoplus_{i=1}^n [\![\mathcal{A}_i]\!]$, where the sum is taken pointwise.

3 The Criterion for Finite Sequentiality

We will show that for a finitely ambiguous max-plus-WTA $\mathcal{A}$, it is decidable whether its behavior $[\![\mathcal{A}]\!]$ is finitely sequential. For this, we first formulate the *separation property*, a generalization of Bala's *A-Fork property* [3]. Then we show that it is decidable whether the separation property is satisfied and that the behavior of a max-plus-WTA is finitely sequential if and only if the separation property is not satisfied. In the following, let Γ be a ranked alphabet. We begin by recalling the tree fork property and the related concepts of *rivals*, *reachers*, *distinguishers*, and *forks*. Intuitively, two states of a finitely ambiguous max-plus-WTA $\mathcal{A}$ are called *rivals* if they can be reached by the same tree u and they can loop in the same Γ-word s but the weights of these loops differ. The tree u is then called a *reacher* of p and q and the Γ-word s a *distinguisher* for p and q. For two rivals p and q, a Γ-word f is called a *p-q-fork* if f can both loop in p and also go from p to q, in a bottom-up sense. We say that $\mathcal{A}$ satisfies the tree fork property if there exist two rivals p and q such that either there exists a p-q-fork or p and q can occur at prefix-independent positions in some run of $\mathcal{A}$. Formally, these definitions are as follows.

▶ **Definition 1.** *Let $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ be a finitely ambiguous max-plus-WTA. Two states $p, q \in Q$ are called* rivals *if there exists a tree $u \in T_\Gamma$ with $\mathrm{Run}_\mathcal{A}(u, p) \neq \emptyset$ and $\mathrm{Run}_\mathcal{A}(u, q) \neq \emptyset$ and a Γ-word s with runs $r_p \in \mathrm{Run}^\diamond_\mathcal{A}(p, s, p)$ and $r_q \in \mathrm{Run}^\diamond_\mathcal{A}(q, s, q)$ such that $\mathrm{wt}^\diamond_\mathcal{A}(s, r_p) \neq \mathrm{wt}^\diamond_\mathcal{A}(s, r_q)$. In this case, we call u a p-q-*reacher *and s a p-q-*distinguisher*. We say that $\mathcal{A}$ satisfies the *tree fork property *if at least one of the following two conditions is satisfied.*

(i) *There exist rivals $p, q \in Q$ and a Γ-word f with $\mathrm{Run}_\mathcal{A}(p, f, p) \neq \emptyset$ and $\mathrm{Run}_\mathcal{A}(p, f, q) \neq \emptyset$. In this case, we call f a p-q-*fork*.

(ii) *There exist rivals $p, q \in Q$, a 2-Γ-context $t \in T_{\Gamma_\diamond}$, and a run $r \in \mathrm{Run}^\diamond_\mathcal{A}(t)$ with $r(\Diamond_1(t)) = p$ and $r(\Diamond_2(t)) = q$. In this case, we call t a p-q-*split*.

We have the following theorem relating the tree fork property to finite sequentiality of unambiguous max-plus-WTA.

▶ **Theorem 2** ([34]). *The behavior of a trim unambiguous max-plus-WTA $\mathcal{A}$ is finitely sequential if and only if $\mathcal{A}$ does not satisfy the tree fork property. If $[\![\mathcal{A}]\!]$ is finitely sequential, a finitely sequential representation of $\mathcal{A}$ can be effectively constructed.*

For finitely ambiguous max-plus-WTA, however, the tree fork property does not capture finite sequentiality. To see why, consider an unambiguous max-plus-WTA $\mathcal{A}$ satisfying the tree fork property [34, 4, 22] and let L be the largest weight used in $\mathcal{A}$. Then construct a one-state max-plus-WTA $\mathcal{B}$ whose every transition weight and every final weight is L. Clearly, $\mathcal{B}$ is deterministic and we have $[\![\mathcal{B}]\!] \geq [\![\mathcal{A}]\!]$. By taking the disjoint union $\mathcal{A} \cup \mathcal{B}$ of $\mathcal{A}$ and $\mathcal{B}$, we obtain a 2-ambiguous max-plus-WTA which satisfies the tree fork property but whose behavior coincides with that of the deterministic automaton $\mathcal{B}$. In this particular example, the states relevant for the tree fork property to be satisfied are not relevant at all for the behavior of the automaton.

For the rest of this paper, let $\mathcal{A}$ be a trim finitely ambiguous max-plus-WTA over the ranked alphabet Γ. In order to reduce the finite sequentiality problem of finitely ambiguous max-plus-WTA to that of unambiguous max-plus-WTA, we decompose $\mathcal{A}$ into a maximum of finitely many unambiguous max-plus-WTA $\mathcal{A}_1, \ldots, \mathcal{A}_N$ and then analyze the interplay of these latter automata. We can do so as in fact, every finitely ambiguous WTA can be decomposed into finitely many unambiguous WTA [22, 32]. This is a common approach when dealing with finite ambiguity [22, 25, 33] and is also used by Bala in the corresponding proof for words [3]. In the simplest case, if $\mathcal{A}_1, \ldots, \mathcal{A}_N$ all do not satisfy the tree fork property, we find a finitely sequential representation of $\mathcal{A}$ by constructing such a representation for each $\mathcal{A}_n$ and then combining all of these. However, if some $\mathcal{A}_n$ does satisfy the tree fork property, we have to analyze whether this automaton contributes enough to the behavior of $\mathcal{A}$ for there not to exist a finitely sequential representation of $\mathcal{A}$. For an easier analysis of the automata $\mathcal{A}_1, \ldots, \mathcal{A}_N$, we normalize their final weights and consider their product as follows.

▶ **Lemma 3** ([41, 22, 32, 10]). *We can effectively find an integer $M \in \mathbb{N}$ and construct a trim WTA $\mathcal{U} = (Q, \Gamma, \mu, \nu)$ over $\mathbb{Q}_{\max}^M$ and Γ such that*

- *$\mathcal{U}$ is unambiguous,*
- *$\mu(\Delta_\mathcal{U}) \subseteq \mathbb{Q}^M \cup \{(-\infty, \ldots, -\infty)\}$ and $\nu(Q) \subseteq \{(0, \ldots, 0), (-\infty, \ldots, -\infty)\}$, and*
- *for every $t \in T_\Gamma$ we have $[\![\mathcal{A}]\!](t) = \max_{i=1}^M \pi_i([\![\mathcal{U}]\!](t))$,*

where $\pi_i \colon \mathbb{Q}_{\max}^M \to \mathbb{Q}_{\max}$ is the projection to the i-th coordinate for every $i \in \{1, \ldots, M\}$.

Let $\mathcal{U}$ be the automaton we obtain for $\mathcal{A}$ from Lemma 3. For a tree $t \in T_\Gamma$, a Γ-word $s \in T_{\Gamma_\diamond}$, runs $r_t \in \mathrm{Run}_\mathcal{U}(t)$, $r_s \in \mathrm{Run}_\mathcal{U}^\diamond(s)$, states $p, q \in Q$, and a coordinate $i \in \{1, \ldots, M\}$, we let $\mathrm{wt}_i(t, r_t) = \pi_i(\mathrm{wt}_\mathcal{U}(t, r_t))$, $\mathrm{wt}_i^\diamond(s, r_s) = \pi_i(\mathrm{wt}_\mathcal{U}^\diamond(s, r_s))$, and $\mathrm{wt}_i^\diamond(p, s, q) = \mathrm{wt}_i^\diamond(s, r_p^q)$ for the unique run $r_p^q \in \mathrm{Run}_\mathcal{U}^\diamond(p, s, q)$. We define the concepts of rivals, reachers, distinguishers, and forks for $\mathcal{U}$ as follows.

▶ **Definition 4.** *Let $i \in \{1, \ldots, M\}$, $p, q \in Q$, $t \in T_\Gamma$, and $r \in \mathrm{Acc}_\mathcal{U}(t)$.*

- *We call p and q i-*rivals *if there exists a tree $u \in T_\Gamma$ such that $\mathrm{Run}_\mathcal{U}(u, p) \neq \emptyset$ and $\mathrm{Run}_\mathcal{U}(u, q) \neq \emptyset$ and a Γ-word s such that $\mathrm{Run}_\mathcal{U}^\diamond(p, s, p) \neq \emptyset$, $\mathrm{Run}_\mathcal{U}^\diamond(q, s, q) \neq \emptyset$, and $\mathrm{wt}_i^\diamond(p, s, p) \neq \mathrm{wt}_i^\diamond(q, s, q)$. In this case, we also call u a p-q-*reacher *and s an i-p-q-*distinguisher.
- *We call a Γ-word f an i-p-q-*fork *if p and q are i-rivals, $\mathrm{Run}_\mathcal{U}^\diamond(p, f, p) \neq \emptyset$, and $\mathrm{Run}_\mathcal{U}^\diamond(p, f, q) \neq \emptyset$.*

- *We say that* (t,r) *is* i-p-q-fork-broken *if there exist positions* $w_p, w_q \in \mathrm{pos}(t)$ *such that* $w_q <_P w_p$*,* $r(w_p) = p$*,* $r(w_q) = q$*, and* $(t\langle\diamond \to w_p\rangle)\restriction_{w_q}$ *is an* i-p-q*-fork.*
- *We say that* (t,r) *is* i-p-q-split-broken *if* p *and* q *are* i*-rivals and there exist two prefix-independent positions* $w_p, w_q \in \mathrm{pos}(t)$ *with* $r(w_p) = p$ *and* $r(w_q) = q$*.*

When appropriate, we may drop some of the hyphenated modifiers from the terms above; for example, we will refer to (t,r) *as* i-fork-broken *if there exist states* $p, q \in Q$ *such that* (t,r) *is* i-p-q*-fork-broken and as* i-split-broken *if there exist states* $p, q \in Q$ *such that* (t,r) *is* i-p-q*-split-broken. We call* (t,r) i-broken *if it is* i*-fork-broken or* i*-split-broken.*

Our concept of brokenness, is inspired by Bala's notion of "broken paths" [3]. Of course, as his proof is concerned with words, the concept of split-brokenness does not exist. His notion of brokenness corresponds to our notion of fork-brokenness. Employing the notion of brokenness, Bala characterizes finite sequentiality of finitely ambiguous max-plus word automata using the *A-Fork property*. Translated to tree automata, the A-Fork property is defined as follows. We say that $\mathcal{U}$ satisfies the A-Fork property if for every constant $C > 0$, there exists a tree $t \in T_\Gamma$ and an accepting run $r \in \mathrm{Acc}_{\mathcal{U}}(t)$ such that for some weight-maximal coordinate i, i.e., with $\mathrm{wt}_i(t,r) = \max_{j=1}^{M} \mathrm{wt}_j(t,r)$, we have that (t,r) is i-broken and for every coordinate j such that (t,r) is not j-broken, we have $\mathrm{wt}_j(t,r) < \mathrm{wt}_i(t,r) - C$. In other words, the A-Fork property is satisfied if broken coordinates are able to dominate non-broken coordinates by an arbitrarily large margin. Bala shows that a finitely ambiguous max-plus word automaton is finitely sequential if and only if the corresponding automaton $\mathcal{U}$ does not satisfy the A-Fork property. For tree automata, however, this criterion does not capture finite sequentiality. More precisely, if we know that there do not exist a tree t and a run r on t such that (t,r) is split-broken, then the A-Fork property does capture finite sequentiality also for tree automata. However, if $\mathcal{U}$ satisfies the A-Fork property due to split-broken coordinates dominating non-broken coordinates, the behavior of $\mathcal{A}$ may still be finitely sequential. This is evidenced by the following example.

▶ **Example 5.** Consider the scenario for $\mathcal{U}$ as defined in Figure 1. The support of $\mathcal{U}$ consists of all trees of the form $c(b^k(a_i^m(d_i)), b^l(a_j^n(d_j)))$ with $i, j \in \{1,2\}$, $k, l > 0$, and $m, n \geq 0$. A valid run on such a tree necessarily assigns states from $\{p_1, p_2, p\}$ to the left branch of the tree and states from $\{q_1, q_2, q\}$ to the right branch of the tree. Moreover, if a branch begins with a letter d_i, this branch is assigned states from $\{p_i, q_i, p, q\}$. In particular, we see that $\mathcal{U}$ is unambiguous. The states p and q are 2-rivals as we see from the p-q-reacher $u = b(a_1(d_1))$ and the 2-p-q-distinguisher $s = b(\diamond)$. By considering the trees $t_n = c(b(a_1^n(d_1)), b(a_2^n(d_2)))$, we see that runs exist where p and q occur prefix-independently and the weight of coordinate 2 is arbitrarily larger than the weights of coordinates 1 and 3 since we have $[\![\mathcal{U}]\!](t_n) = (-n, 0, -n)$. However, in t_n the subtrees below p and q are distinct, thus a deterministic automaton can distinguish between them.

In fact, if $\mathcal{U}$ is given this way, we can construct a finitely sequential representation of $[\![\mathcal{A}]\!]$ as follows. All trees of the form $c(b^k(a_1^m(d_1)), b^l(a_1^n(d_1)))$ are assigned the weight $(-m-n+k+l-2, k-l, k+l-2)$, so coordinate 3 is always dominant. Similarly, coordinate 1 is dominant for trees of the form $c(b^k(a_2^m(d_2)), b^l(a_2^n(d_2)))$. These trees can be handled by the deterministic max-plus-WTA obtained from $\mathcal{U}$ by removing the states q_1 and q_2, letting $\mu(p, p, c, \top) = (0,0,0)$, and replacing μ with $\pi_1 \circ \mu$ and $\pi_3 \circ \mu$, respectively. For trees of the form $c(b^k(a_1^m(d_1)), b^l(a_2^n(d_2)))$, we remove the states p_2 and q_1 from $\mathcal{U}$ and then construct three deterministic max-plus-WTA by replacing μ by $\pi_1 \circ \mu, \pi_2 \circ \mu$, and $\pi_3 \circ \mu$, respectively. For the trees $c(b^k(a_2^m(d_2)), b^l(a_1^n(d_1)))$ we can proceed similarly. The pointwise maximum of the automata constructed this way is then equivalent to $[\![\mathcal{A}]\!]$. This example shows in particular that if $\mathcal{U}$ satisfies the A-Fork property, $[\![\mathcal{A}]\!]$ can still be finitely sequential.

$$\nu(\top) = (0,0,0) \qquad \mu(p,b,p) = (1,1,1) \qquad \mu(p_1,a_1,p_1) = \mu(q_1,a_1,q_1) = (-1,0,0)$$
$$\mu(p,q,c,\top) = (0,0,0) \qquad \mu(q,b,q) = (1,-1,1) \qquad \mu(p_2,a_2,p_2) = \mu(q_2,a_2,q_2) = (0,0,-1)$$
$$\mu(p_1,b,p) = \mu(p_2,b,p) = \mu(q_1,b,q) = \mu(q_2,b,q) = (0,0,0)$$
$$\mu(d_1,p_1) = \mu(d_1,q_1) = \mu(d_2,p_2) = \mu(d_2,q_2) = (0,0,0)$$

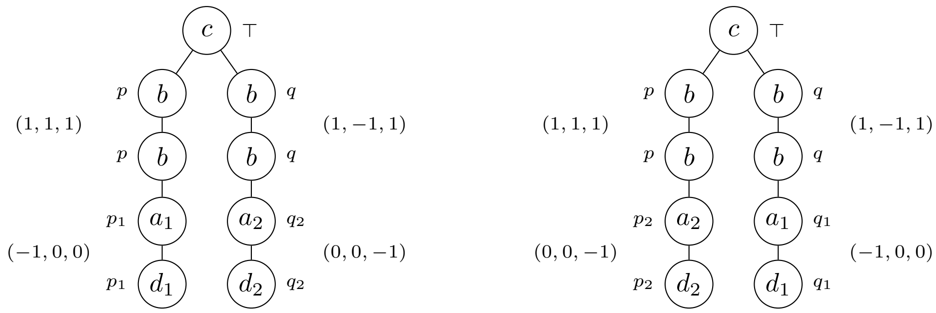

Figure 1 A scenario for the automaton $\mathcal{U}$: The automaton $(\{p_1, p_2, q_1, q_2, p, q, \top\}, \Gamma, \mu, \nu)$ over the ranked alphabet $\Gamma = \{a_1, a_2, b, c, d_1, d_2\}$ where $c \in \Gamma^{(2)}$, $a_1, a_2, b \in \Gamma^{(1)}$, and $d_1, d_2 \in \Gamma^{(0)}$. All unspecified weights are assumed to be $-\infty$. The states p and q are 2-rivals.

Our fundamental idea to adapt the A-Fork property to tree automata is to formulate our version not for $\mathcal{U}$ but for a *covering* of $\mathcal{U}$. Oversimplifying, a *covering* of an automaton is a new automaton obtained by enhancing the states of the original automaton with additional capacities to store information. A prominent example of a covering construction is the *Schützenberger covering* of an automaton. The Schützenberger covering in particular has already been employed in a number of decidability results for max-plus automata [22, 4, 3, 33, 34]. For more background on coverings, see [38].

Here, we construct from $\mathcal{U}$ an unambiguous automaton $\boldsymbol{\mathcal{U}}$ with the same behavior as $\mathcal{U}$ and whose states are tuples from $Q \times \mathcal{P}(Q) \times \mathcal{P}(Q^4 \times \mathcal{P}(Q^2))$. Every run $\mathbf{r}$ of $\boldsymbol{\mathcal{U}}$ on a tree $t \in T_\Gamma$ will correspond uniquely to a run of $\mathcal{U}$ on t, given by projecting to the first entry. For a position w, the second entry of $\mathbf{r}(w)$ will be the set of all states $q \in Q$ which can be reached by $t{\restriction_w}$, i.e., for which $\mathrm{Run}_{\mathcal{U}}(t{\restriction_w}, q)$ is non-empty. The third entry of $\mathbf{r}(w)$ will contain a tuple (p, q, p', q', Y) if and only if $t{\restriction_w}$ can reach p and q with two runs r_p and r_q, these runs visited p' and q' simultaneously at some position v in the past, and Y consists of all pairs of states which these runs visited simultaneously up to v. More precisely, the third entry of $\mathbf{r}(w)$ will consist of all tuples (p, q, p', q', Y) such that (1) there exist runs $r_p \in \mathrm{Run}_{\mathcal{U}}(t{\restriction_w}, p)$ and $r_q \in \mathrm{Run}_{\mathcal{U}}(t{\restriction_w}, q)$, (2) for some position below w, i.e., some position $v \in \mathrm{pos}(t{\restriction_w})$, we have $r_p(v) = p'$ and $r_q(v) = q'$, and (3) Y is the set of all pairs of states $(r_p(vu), r_q(vu))$ with $u \in \mathrm{pos}(t{\restriction_{wv}})$. Our intention of considering the covering $\boldsymbol{\mathcal{U}}$ is to increase the knowledge we have about each pair of rivals. For two rivals of $\mathcal{U}$, all we know is what the definition of rivals specifies. For two rivals of $\boldsymbol{\mathcal{U}}$ on the other hand, we can show that they are necessarily of the form (p, P, V) and (q, P, V) where p and q are rivals of $\mathcal{U}$. This allows us to infer statements about the rivals of $\boldsymbol{\mathcal{U}}$ which are not necessarily true for the rivals of $\mathcal{U}$. The precise construction of $\boldsymbol{\mathcal{U}}$ is as follows.

▶ **Construction 6.** We define $\boldsymbol{\mathcal{U}} = (\mathbf{Q}, \Gamma, \boldsymbol{\mu}, \boldsymbol{\nu})$ as the trim part of the automaton $\boldsymbol{\mathcal{U}}' = (\mathbf{Q}', \Gamma, \boldsymbol{\mu}', \boldsymbol{\nu}')$ defined as follows. We let $\mathbf{Q}' = Q \times \mathcal{P}(Q) \times \mathcal{P}(Q^4 \times \mathcal{P}(Q^2))$ and for subsets $P_1, \ldots, P_m \subseteq Q$ and a letter $a \in \Gamma$ with $\mathrm{rk}_\Gamma(a) = m$, we let $\mathrm{succ}(P_1, \ldots, P_m, a) = \{q_0 \mid \exists (q_1, \ldots, q_m) \in P_1 \times \ldots \times P_m \text{ with } \mu(q_1, \ldots, q_m, a, q_0) \in \mathbb{Q}^M\}$. For $i \in \{1, \ldots, \mathrm{rk}_\Gamma(a)\}$ and states $(p, q, p', q', Y) \in Q^4 \times \mathcal{P}(Q^2)$, we let $\mathrm{succ}(P_1, \ldots, P_m, (p, q, p', q', Y), i, a) =$

$\text{succ}(P_1, \ldots, P_{i-1}, \{p\}, P_{i+1}, \ldots, P_m, a) \times \text{succ}(P_1, \ldots, P_{i-1}, \{q\}, P_{i+1}, \ldots, P_m, a) \times \{p'\} \times \{q'\} \times \{Y\}$. For $V \subseteq Q^4 \times \mathcal{P}(Q^2)$ and $p, q \in Q$, we let $\text{visited}(p, q, V) = \{(p', q') \mid (p, q, p', q', Y) \in V \text{ for some } Y \subseteq Q^2\}$. Then for $a \in \Gamma$ with $\text{rk}_\Gamma(a) = m$ and $(p_0, P_0, V_0), \ldots, (p_m, P_m, V_m) \in \mathbf{Q}'$, we define $\boldsymbol{\nu}'(p_0, P_0, V_0) = \nu(p_0)$ and $\boldsymbol{\mu}'((p_1, P_1, V_1), \ldots, (p_m, P_m, V_m), a, (p_0, P_0, V_0)) =$

$$\begin{cases} \mu(p_1, \ldots, p_m, a, p_0) & \text{if } P_0 = \text{succ}(P_1, \ldots, P_m, a) \text{ and with} \\ & V = \bigcup_{i=1}^m \bigcup_{(p,q,p',q',Y) \in V_i} \text{succ}(P_1, \ldots, P_m, (p, q, p', q', Y), i, a) \\ & \text{we have } V_0 = V \cup \{(p, q, p, q, Y) \mid p, q \in P_0 \text{ and} \\ & \qquad\qquad Y = \text{visited}(p, q, V) \cup \{(p, q)\}\} \\ (-\infty, \ldots, -\infty) & \text{otherwise.} \end{cases}$$

Then $\boldsymbol{\mathcal{U}}$ satisfies the properties described above. We let $\pi_1 \colon \mathbf{Q} \to Q$, $\pi_2 \colon \mathbf{Q} \to \mathcal{P}(Q)$, and $\pi_3 \colon \mathbf{Q} \to \mathcal{P}(Q^4 \times \mathcal{P}(Q^2))$ be the projections, and let $\mathbf{wt}_i$ and $\mathbf{wt}_i^\diamond$ be defined for $\boldsymbol{\mathcal{U}}$ in the same way we defined wt_i and $\text{wt}_i^\diamond$ for $\mathcal{U}$. Furthermore, we note that the concepts of rivals, reachers, distinguishers, and forks as defined for $\mathcal{U}$ in Definition 4 apply to $\boldsymbol{\mathcal{U}}$ in a similar fashion.

Finally, we introduce our version of the A-Fork property. To allow for easier proofs, we use a different formulation and consequently a different name. But in fact, $\boldsymbol{\mathcal{U}}$ satisfies the separation property if and only if it satisfies the A-Fork property in the way we translated it to trees earlier.

▶ **Definition 7.** *Let $C \in \mathbb{N}$. We call a set $I \subseteq \{1, \ldots, M\}$ C-separable if there exists a tree $t \in T_\Gamma$ and a run $\mathbf{r} \in \text{Acc}_{\boldsymbol{\mathcal{U}}}(t)$ such that*

(i) *if $i \in I$, then $(t, \mathbf{r})$ is i-broken and*

(ii) *if $j \in \{1, \ldots, M\} \setminus I$, then $\mathbf{wt}_j(t, \mathbf{r}) \leq \mathbf{wt}_i(t, \mathbf{r}) - C$ for all $i \in I$.*

In this case, we also say that $(t, \mathbf{r})$ is I-C-separated. We call I separable if it is C-separable for every $C \in \mathbb{N}$ and define $\mathcal{I}$ as the set of all separable subsets $I \subseteq \{1, \ldots, M\}$. If $\mathcal{I}$ is non-empty, we say that $\boldsymbol{\mathcal{U}}$ satisfies the separation property or, for short, that $\boldsymbol{\mathcal{U}}$ is broken.

Our main result is to prove the following theorem relating the separation property to the finite sequentiality problem of finitely ambiguous max-plus-WTA.

▶ **Theorem 8.** *The behavior $[\![\mathcal{A}]\!]$ of $\mathcal{A}$ is finitely sequential if and only if $\boldsymbol{\mathcal{U}}$ is not broken. Moreover, it is decidable whether $\boldsymbol{\mathcal{U}}$ is broken. In particular, it is decidable whether $[\![\mathcal{A}]\!]$ is finitely sequential.*

We separate the proof of Theorem 8 into three parts. Due to lack of space, we have to restrict ourselves to brief descriptions of our methods. In Section 3.1, we outline that it is decidable whether $\boldsymbol{\mathcal{U}}$ is broken. This part of the proof does not follow any idea from [3] as in his proof, Bala reduces the decidability of the A-Fork property to the decidability of a decidable fragment of Presburger arithmetic. In Section 3.2, we show that if $\boldsymbol{\mathcal{U}}$ is broken, then $[\![\mathcal{A}]\!]$ is not finitely sequential. This part of the proof employs ideas from [4, 3, 34], but extends these non-trivially. This particular proof is the most challenging and technical aspect of our result. Finally, in Section 3.3, we outline how to construct finitely many deterministic max-plus-WTA whose pointwise maximum is equivalent to $[\![\mathcal{A}]\!]$ in case that $\boldsymbol{\mathcal{U}}$ is not broken. Although this part is inspired by an idea in [3], we are not sure whether we employ this idea in the same way.

For all of our proofs, it is crucial that for every two states of $\boldsymbol{\mathcal{U}}$, we can decide whether they are rivals [2, Section 4], [11, Section 5.4]. For two rivals of an unambiguous automaton, it is in fact quite easy to give an upper bound on the size of their smallest distinguisher and smallest reacher. Thus, deciding whether two states are rivals reduces to checking for finitely many trees whether they can reach both states and checking for finitely many Γ-words whether they are a distinguisher for these two states. For Section 3.3, we require an even more precise statement, namely that if s is a distinguisher for two rivals $\mathbf{p}$ and $\mathbf{q}$, then we can obtain a **p**-**q**-distinguisher of height at most $4|\mathbf{Q}|^2$ by removing loops from the unique runs looping in $\mathbf{p}$ and $\mathbf{q}$. For this, we employ the notion of a *truncation*. Simply put, for a Γ-word s and a run $\mathbf{r}$ on s, a truncation of $(s, \mathbf{r})$ is any pair $(s', \mathbf{r}')$ of a Γ-word s' and a run $\mathbf{r}'$ on s' which can be obtained by repeatedly cutting loops from $(s, \mathbf{r})$.

▶ **Definition 9.** *Let $s, s' \in T_{\Gamma_\diamond}$ be Γ-words, $\mathbf{r} \in \mathrm{Run}^{\diamond}_{\mathcal{U}}(s)$, and $\mathbf{r}' \in \mathrm{Run}^{\diamond}_{\mathcal{U}}(s')$. We say that $(s', \mathbf{r}')$ is a* truncation *of $(s, \mathbf{r})$, denoted by $(s, \mathbf{r}) \mathrel{>\!\!\!\!\!\times} (s', \mathbf{r}')$, if there exists a mapping $g\colon \mathrm{pos}(s') \to \mathrm{pos}(s)$ such that $g(\varepsilon) = \varepsilon$, $g(\Diamond_1(s')) = \Diamond_1(s)$, for all $w \in \mathrm{pos}(s')$ we have $\mathbb{t}(s', \mathbf{r}', w) = \mathbb{t}(s, \mathbf{r}, g(w))$, and for all $w_1, w_2 \in \mathrm{pos}(s')$ we have $g(w_1) \le_L g(w_2)$ if and only if $w_1 \le_L w_2$ and $g(w_1) \le_P g(w_2)$ if and only if $w_1 \le_P w_2$.*

We can use truncations to bound the size of distinguishers as follows.

▶ **Lemma 10** ([11, Lemma 5.10],[34]). *Let $\mathbf{p}, \mathbf{q} \in \mathbf{Q}$ be i-rivals for some $i \in \{1, \ldots, M\}$, let $u \in T_\Gamma$ be a **p**-**q**-reacher, let $f \in T_{\Gamma_\diamond}$ be a **p**-**q**-fork, let $s \in T_{\Gamma_\diamond}$ be an i-**p**-**q**-distinguisher, and let $\mathbf{r_p} \in \mathrm{Run}^{\diamond}_{\mathcal{U}}(\mathbf{p}, s, \mathbf{p})$ and $\mathbf{r_q} \in \mathrm{Run}^{\diamond}_{\mathcal{U}}(\mathbf{q}, s, \mathbf{q})$. Then there exists a **p**-**q**-reacher u' with* $\mathrm{height}(u') \le |\mathbf{Q}|^2$*, a **p**-**q**-fork f' with* $\mathrm{height}(f') \le 2|\mathbf{Q}|^2$*, and an i-**p**-**q**-distinguisher s' with* $\mathrm{height}(s') \le 4|\mathbf{Q}|^2$ *such that for the runs $\mathbf{r}'_\mathbf{p} \in \mathrm{Run}^{\diamond}_{\mathcal{U}}(\mathbf{p}, s', \mathbf{p})$ and $\mathbf{r}'_\mathbf{q} \in \mathrm{Run}^{\diamond}_{\mathcal{U}}(\mathbf{q}, s', \mathbf{q})$, $(s', \mathbf{r}'_\mathbf{p})$ is a truncation of $(s, \mathbf{r_p})$ and $(s', \mathbf{r}'_\mathbf{q})$ is a truncation of $(s, \mathbf{r_q})$. In particular, for every two states $\mathbf{p}, \mathbf{q} \in \mathbf{Q}$, it is decidable whether $\mathbf{p}$ and $\mathbf{q}$ are rivals.*

3.1 Decidability

In order to show that it is decidable whether $\mathcal{U}$ is broken, we construct a covering $\bar{\mathcal{U}}$ of $\mathcal{U}$ with the capability to detect the broken coordinates of a run of $\mathcal{U}$. The covering $\bar{\mathcal{U}}$ possesses the same behavior as $\mathcal{U}$ and each run of $\bar{\mathcal{U}}$ corresponds to exactly one run of $\mathcal{U}$. We construct $\bar{\mathcal{U}}$ by adding to each state of $\mathbf{Q}$ one entry containing all states reachable on the current subtree, one entry containing all states visited on the current run, one entry containing all pairs $(\mathbf{p}, \mathbf{q})$ of states such that $\mathbf{q}$ is reachable by a run which visited $\mathbf{p}$ at a position where our current run also visited $\mathbf{p}$, and one entry containing a record of all broken coordinates. This allows us to infer the brokenness of a run directly from the state at its root. More precisely, we construct $\bar{\mathcal{U}} = (\bar{Q}, \Gamma, \bar{\mu}, \bar{\nu})$ with states from $\bar{Q} = \mathbf{Q} \times \mathcal{P}(\mathbf{Q}) \times \mathcal{P}(\mathbf{Q}) \times \mathcal{P}(\mathbf{Q}^2) \times \{0,1\}^M$ such that if $t \in T_\Gamma$, $\bar{r} \in \mathrm{Run}_{\bar{\mathcal{U}}}(t)$, $w \in \mathrm{pos}(t)$, and $\bar{r}(w) = (\mathbf{q}, P, V, W, \bar{a})$, then the following statements hold. We have (1) the projection $\pi_\mathbf{Q}\colon \bar{Q} \to \mathbf{Q}$ to the first coordinate induces a bijection $\pi_\mathbf{Q}\colon \mathrm{Acc}_{\bar{\mathcal{U}}}(t) \to \mathrm{Acc}_{\mathcal{U}}(t)$ preserving weights of runs, (2) $P = \{\mathbf{p} \in \mathbf{Q} \mid \mathrm{Run}_{\mathcal{U}}(t{\restriction_w}, \mathbf{p}) \neq \emptyset\}$, (3) $V = \{\pi_\mathbf{Q} \circ \bar{r}(wv) \mid v \in \mathrm{pos}(t{\restriction_w})\}$, (4) $W = \{(\mathbf{p}_1, \mathbf{p}_2) \in \mathbf{Q}^2 \mid$ for some $v \in \mathrm{pos}(t{\restriction_w})$ we have $\pi_\mathbf{Q} \circ \bar{r}(wv) = \mathbf{p}_1$ and $\mathrm{Run}^{\diamond}_{\mathcal{U}}(\mathbf{p}_1, t\langle \diamond \to wv\rangle{\restriction_w}, \mathbf{p}_2) \neq \emptyset\}$, and (5) $\bar{a}[i] = 1$ if and only if $(t, \pi_\mathbf{Q} \circ \bar{r}){\restriction_w}$ is i-broken.

We let $d_1, \ldots, d_D$ be an enumeration of $\Delta_{\bar{\mathcal{U}}}$ and for a tree $t \in T_\Gamma$ and a run $\bar{r} \in \mathrm{Run}_{\bar{\mathcal{U}}}(t)$, define the *transition Parikh vector* of $(t, \bar{r})$ by $\mathbb{p}(t, \bar{r}) = (|\{w \in \mathrm{pos}(t) \mid \mathbb{t}(t, \bar{r}, w) = d_1\}|, \ldots, |\{w \in \mathrm{pos}(t) \mid \mathbb{t}(t, \bar{r}, w) = d_D\}|)$, i.e., we count the number of occurrences of each transition in $(t, \bar{r})$. We can employ Parikh's Theorem [31, 15] to show that the set $\mathbb{p}(\bar{\mathcal{U}}) = \{\mathbb{p}(t, r) \mid t \in T_\Gamma, r \in \mathrm{Acc}_{\bar{\mathcal{U}}}(t)\}$ is *semilinear*, i.e., that there exist integers $k, k_1, \ldots, k_k$, vectors $\alpha^{(l)} \in \mathbb{N}^D$, and matrices $\beta^{(l)} \in \mathbb{N}^{D \times k_l}$ ($l \in \{1, \ldots, k\}$) with $\mathbb{p}(\bar{\mathcal{U}}) = \bigcup_{l=1}^{k} \{\alpha^{(l)} + \beta^{(l)} \bar{X} \mid \bar{X} = (X_1, \ldots, X_{k_l}) \in \mathbb{N}^{k_l}\}$. We note that Parikh's Theorem is effective, so we can effectively compute all of these integers, vectors, and matrices.

Next, we let $\Omega = (\bar{\mu}(d_1), \ldots, \bar{\mu}(d_D)) \in \mathbb{Q}^{M \times D}$ be a matrix containing the transition weight vectors of $d_1, \ldots, d_D$, let $\omega_1, \ldots, \omega_M$ be the rows of Ω, and let $\tilde{C} = 1 + \max\{|\omega_i \alpha^{(l)} - \omega_j \alpha^{(l)}| \mid i, j \in \{1, \ldots, M\}, l \in \{1, \ldots, k\}\}$. For $I \subseteq \{1, \ldots, M\}$, we let $D_I = \{l \in \{1, \ldots, D\} \mid d_l =$

$(\bar{q}_1, \ldots, \bar{q}_m, \bar{a}, (\mathbf{q}, P, V, W, \bar{a})) \in \Delta_{\bar{\mathcal{U}}}$ such that $\bar{a}[i] = 1 \leftrightarrow i \in I\}$ be the set of all indices of transitions whose root state indicates brokenness of exactly the coordinates in I. Furthermore, for every $l \in \{1, \ldots, D\}$ we let $\alpha^{(l)}[1], \ldots, \alpha^{(l)}[D]$ be the entries of $\alpha^{(l)}$ and $\beta^{(l)}[1], \ldots, \beta^{(l)}[D]$ be the rows of $\beta^{(l)}$. It is then elementary to show that every set $I \subseteq \{1, \ldots, M\}$ is separable iff it is $\tilde{C}$-separable iff for some $l \in \{1, \ldots, M\}$, the system of linear inequalities below possesses an integer solution. The satisfiability of systems of linear inequalities over the rationals with integer solutions is decidable [30][9, Theorem 3.4]. There are only finitely many such systems to consider, so it is decidable whether I is separable. To decide whether $\boldsymbol{\mathcal{U}}$ is broken, it suffices to check whether there exists a separable subset $I \subseteq \{1, \ldots, M\}$.

$$(\omega_i \beta^{(l)} - \omega_j \beta^{(l)})\bar{X} \geq \omega_j \alpha^{(l)} - \omega_i \alpha^{(l)} + \tilde{C} \qquad \bar{X} \geq 0 \qquad (i \in I, j \in I^c)$$

$$\sum_{d \in D_I} \beta^{(l)}[d]\bar{X} \geq 1 - \sum_{d \in D_I} \alpha^{(l)}[d] \qquad - \sum_{I \subsetneq I' \subseteq M} \sum_{d \in D_{I'}} \beta^{(l)}[d]\bar{X} \geq \sum_{I \subsetneq I' \subseteq M} \sum_{d \in D_{I'}} \alpha^{(l)}[d]$$

3.2 Necessity

To show that $[\![\mathcal{A}]\!]$ is not finitely sequential if $\boldsymbol{\mathcal{U}}$ is broken, we assume that $[\![\mathcal{A}]\!]$ can be represented as a finite maximum of deterministic max-plus-WTA $\mathcal{A}_1, \ldots, \mathcal{A}_N$ and employ Ramsey's Theorem [37] to obtain a contradiction. Although not stated explicitly, Bala's proof for words [3] most likely also involves some form of Ramsey's Theorem as his proof of $\mathcal{U}$ being broken implying $[\![\mathcal{A}]\!]$ to not be finitely sequential "deals with colorings of finite hypercubes". In our proof for tree automata, we are able to handle fork-brokenness without employing Ramsey's Theorem. This suggest that applying our approach to word automata yields a proof which is simpler than the corresponding one used in [3]. The reason for this is that our separation property considers sets of coordinates instead of the single coordinates which the A-Fork property considers. For the separable sets $I \in \mathcal{I}$ which are minimal with respect to set inclusion, we are able to prove a statement for I-C-separated pairs $(t, \mathbf{r})$ which greatly facilitates dealing with fork-brokenness and enables us to deal with split-brokenness in the first place. Namely, if $(t, \mathbf{r})$ is I-C-separated for a sufficiently large C and no subset of I is separable, then for every $\mathbf{p} \in \mathbf{r}(\mathrm{pos}(t))$, every Γ-word s with $\mathrm{height}(s) \leq 4|\mathbf{Q}|^2$ and $\mathrm{Run}^{\diamond}_{\boldsymbol{\mathcal{U}}}(\mathbf{p}, s, \mathbf{p}) \neq \emptyset$, and every two coordinates $i, j \in I$ we have $\mathbf{wt}^{\diamond}_i(\mathbf{p}, s, \mathbf{p}) = \mathbf{wt}^{\diamond}_j(\mathbf{p}, s, \mathbf{p})$.

For the proof, we choose a sufficiently large C, a set $I \in \mathcal{I}$ which is minimal with respect to set inclusion, and an I-C-separated pair $(t, \mathbf{r})$. Then we construct from $(t, \mathbf{r})$ new trees which require more than N deterministic max-plus-WTA in order to be assigned the correct weight. If $(t, \mathbf{r})$ is fork-broken, we construct trees and runs of $\boldsymbol{\mathcal{U}}$ with increasingly more alterations of forks and distinguishers. This approach is similar to the method used in [4] and [34] to deal with fork-brokenness. The challenge we face in adapting the proof from [34] to our situation is that we have to ensure that in the runs we construct, the coordinates from I dominate the other coordinates. In our constructions we may therefore only make "small" modifications to $(t, \mathbf{r})$. Our solution relies on the minimality of I and involves constructing more than the $N + 1$ trees sufficient for the proofs in [4, 34]. The case where $(t, \mathbf{r})$ is split-broken is much more complicated and is in fact the only reason we have to use the covering automaton $\boldsymbol{\mathcal{U}}$ instead of $\mathcal{U}$. As split-brokenness does not apply to words, this was not an issue in [3]. Our approach vastly extends the idea used to deal with split-brokenness in [34]. In [34], we relied on replacing the subtrees below the prefix-independent rivals of a split-broken run by other suitable trees. However, Example 5 shows that this is not possible in our scenario as these subtrees may be indispensable to ensure that broken coordinates dominate all non-broken coordinates. Here, we instead modify the subtrees present and construct trees and runs of $\mathcal{U}$. For this, we employ the properties of $\boldsymbol{\mathcal{U}}$, pumping techniques, and Ramsey's Theorem.

3.3 Sufficiency

To show that $[\![\mathcal{A}]\!]$ is finitely sequential if $\boldsymbol{\mathcal{U}}$ is not broken, we show that in this case we can construct M unambiguous max-plus-WTA which all do not satisfy the tree fork property and whose pointwise maximum is equivalent to $[\![\mathcal{A}]\!]$. By Theorem 2, we obtain a finitely sequential representation of $\mathcal{A}$ by constructing one for each of the unambiguous max-plus-WTA. We essentially construct the unambiguous automata by removing problematic runs from $\boldsymbol{\mathcal{U}}$ and then projecting to the coordinates $1, \ldots, M$. Our fundamental idea is the following. First, let ξ be the smallest difference between the weights of two coordinates of a loop in a Γ-word of height at most $4|\mathbf{Q}|^2$, i.e., for $X = \{|\mathbf{wt}_i^{\diamond}(\mathbf{p}, s, \mathbf{p}) - \mathbf{wt}_j^{\diamond}(\mathbf{q}, s, \mathbf{q})| \mid \mathbf{p}, \mathbf{q} \in \mathbf{Q}, i, j \in \{1, \ldots, M\}, s$ is a Γ-word with $\text{height}(s) \leq 4|\mathbf{Q}|^2, \text{Run}_{\boldsymbol{\mathcal{U}}}^{\diamond}(\mathbf{p}, s, \mathbf{p}) \neq \emptyset, \text{Run}_{\boldsymbol{\mathcal{U}}}^{\diamond}(\mathbf{q}, s, \mathbf{q}) \neq \emptyset\}$, let $\xi = \min X \setminus \{0\}$. Then assume that $(t, \mathbf{r})$ is i-$\mathbf{p}$-$\mathbf{q}$-broken and that the maximum of $\text{wt}_{\boldsymbol{\mathcal{U}}}(t, \mathbf{r})$ is in coordinate i. Furthermore, assume that in $\mathbf{r}$, some i-$\mathbf{p}$-$\mathbf{q}$-distinguisher s of height at most $4|\mathbf{Q}|^2$ loops N times in $\mathbf{p}$, where $N \in \mathbb{N}$ is some integer, and that $\mathbf{wt}_i(\mathbf{p}, s, \mathbf{p}) < \mathbf{wt}_i(\mathbf{q}, s, \mathbf{q})$. By removing the loops of s in $\mathbf{p}$ from $(t, \mathbf{r})$ and inserting them back as loops in $\mathbf{q}$, we increase the weight of coordinate i by $N\xi$, but leave the weights of all non-broken coordinates unchanged. Thus, coordinate i then dominates all non-broken coordinates by a margin of at least $N\xi$. As i cannot dominate all non-broken coordinates by an arbitrarily large margin, N cannot be arbitrarily large. In turn, this means that if N is sufficiently large, then $\text{wt}_{\boldsymbol{\mathcal{U}}}(t, \mathbf{r})$ cannot take its maximum weight in coordinate i. This implies that the weight of coordinate i can be discarded if some distinguisher loops in both of its rivals too many times.

We employ this idea in the following way. We define the set $R = \{(i, \mathbf{p}, \mathbf{q}, s) \in \{1, \ldots, M\} \times \mathbf{Q}^2 \times T_{\Gamma_\diamond} \mid i \in \{1, \ldots, M\}$, s is an i-$\mathbf{p}$-$\mathbf{q}$-distinguisher, $\text{height}(s) \leq 4|\mathbf{Q}|^2\}$ and define the constant $N = \lceil M\tilde{C}\xi^{-1} \rceil$, where $\tilde{C}$ is as in Section 3.1. We note that R is computable, as by Lemma 10, we can decide for every two states $\mathbf{p}, \mathbf{q} \in \mathbf{Q}$ whether they are rivals or not. Let $t \in T_\Gamma$, $\mathbf{r} \in \text{Run}_{\boldsymbol{\mathcal{U}}}(t)$, $(i, \mathbf{p}, \mathbf{q}, s) \in R$, $\mathbf{r}_{\mathbf{p}} \in \text{Run}_{\boldsymbol{\mathcal{U}}}^{\diamond}(\mathbf{p}, s, \mathbf{p})$, and $\mathbf{r}_{\mathbf{q}} \in \text{Run}_{\boldsymbol{\mathcal{U}}}^{\diamond}(\mathbf{q}, s, \mathbf{q})$. We call $(t, \mathbf{r})$

- $(i, \mathbf{p}, \mathbf{q}, s)$*-fork-broken* if there exist positions $u_{\mathbf{p}}, v_{\mathbf{p}}, w_{\mathbf{p}}, w_{\mathbf{q}}, u_{\mathbf{q}}, v_{\mathbf{q}} \in \text{pos}(t)$ with $v_{\mathbf{q}} <_{\text{P}} u_{\mathbf{q}} \leq_{\text{P}} w_{\mathbf{q}} <_{\text{P}} w_{\mathbf{p}} \leq_{\text{P}} v_{\mathbf{p}} <_{\text{P}} u_{\mathbf{p}}$ such that $(t, \mathbf{r})\langle \diamond \to u_{\mathbf{p}} \rangle\!\restriction_{v_{\mathbf{p}}} \gg (s, \mathbf{r}_{\mathbf{p}})^{N+1}$, $(t, \mathbf{r})\langle \diamond \to u_{\mathbf{q}} \rangle\!\restriction_{v_{\mathbf{q}}} \gg (s, \mathbf{r}_{\mathbf{q}})^{N+1}$, $\mathbf{r}(w_{\mathbf{p}}) = \mathbf{p}$, $\mathbf{r}(w_{\mathbf{q}}) = \mathbf{q}$, and $t\langle \diamond \to w_{\mathbf{p}} \rangle\!\restriction_{w_{\mathbf{q}}}$ is a $\mathbf{p}$-$\mathbf{q}$-fork.
- $(i, \mathbf{p}, \mathbf{q}, s)$*-split-broken* if there exist positions $u_{\mathbf{p}}, v_{\mathbf{p}}, u_{\mathbf{q}}, v_{\mathbf{q}} \in \text{pos}(t)$ such that $v_{\mathbf{p}} <_{\text{P}} u_{\mathbf{p}}$, $v_{\mathbf{q}} <_{\text{P}} u_{\mathbf{q}}$, $v_{\mathbf{p}}$ and $v_{\mathbf{q}}$ are prefix-independent, $(t, \mathbf{r})\langle \diamond \to u_{\mathbf{p}} \rangle\!\restriction_{v_{\mathbf{p}}} \gg (s, \mathbf{r}_{\mathbf{p}})^{N+1}$, and $(t, \mathbf{r})\langle \diamond \to u_{\mathbf{q}} \rangle\!\restriction_{v_{\mathbf{q}}} \gg (s, \mathbf{r}_{\mathbf{q}})^{N+1}$.

Following the idea above, we can show that if $(t, \mathbf{r})$ is $(i, \mathbf{p}, \mathbf{q}, s)$-broken, then $\mathbf{wt}_i(t, \mathbf{r}) < [\![\mathcal{A}]\!](t)$. Furthermore, we can show that for every $i \in \{1, \ldots, M\}$, we can construct a complete and deterministic FTA $\mathcal{B}_i$ over the alphabet $\Gamma \times \mathbf{Q}$ which accepts a tree $(t, \mathbf{r}) \in T_{\Gamma \times \mathbf{Q}}$ if and only if there does not exist $(i, \mathbf{p}, \mathbf{q}, s) \in R$ such that $(t, \mathbf{r})$ is $(i, \mathbf{p}, \mathbf{q}, s)$-broken. Via a product-like construction, we can employ $\mathcal{B}_i$ to construct a WTA $\mathcal{C}_i$ which "filters" the runs of $\boldsymbol{\mathcal{U}}$ so that the runs of $\mathcal{C}_i$ correspond exactly to those runs of $\boldsymbol{\mathcal{U}}$ which are not $(i, \mathbf{p}, \mathbf{q}, s)$-broken for any $(i, \mathbf{p}, \mathbf{q}, s) \in R$. Then $\mathcal{C}_i$ is not i-broken as otherwise, we can construct an accepting run r of $\mathcal{C}_i$ on a tree t whose projection to $\mathbf{Q}$ yields an i-broken pair $(t, \mathbf{r})$ which loops an i-$\mathbf{p}$-$\mathbf{q}$-distinguisher s for $N + 1$ times in both $\mathbf{p}$ and $\mathbf{q}$. As by Lemma 10, s can be truncated to a distinguisher of height at most $4|\mathbf{Q}|^2$, $(t, \mathbf{r})$ is thus $(i, \mathbf{p}, \mathbf{q}, s)$-broken. We obtain the contradiction that r is not accepting. Consequently, projecting the weights of each automaton $\mathcal{C}_i$ to the i-th coordinate yields M unambiguous max-plus-WTA whose pointwise-maximum coincides with $[\![\mathcal{A}]\!]$ and which all do not satisfy the tree fork property.

References

1 Athanasios Alexandrakis and Symeon Bozapalidis. Weighted grammars and Kleene's theorem. *Information Processing Letters*, 24(1):1–4, 1987.
2 Cyril Allauzen and Mehryar Mohri. Efficient algorithms for testing the twins property. *Journal of Automata, Languages and Combinatorics*, 8(2):117–144, 2003.
3 Sebastian Bala. Which finitely ambiguous automata recognize finitely sequential functions? (extended abstract). In Krishnendu Chatterjee and Jirí Sgall, editors, *38th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 8087 of *Lecture Notes in Computer Science*, pages 86–97. Springer, 2013.
4 Sebastian Bala and Artur Koniński. Unambiguous automata denoting finitely sequential functions. In Adrian-Horia Dediu, Carlos Martín-Vide, and Bianca Truthe, editors, *7th International Conference on Language and Automata Theory and Applications (LATA)*, volume 7810 of *Lecture Notes in Computer Science*, pages 104–115. Springer, 2013.
5 Jean Berstel and Christophe Reutenauer. Recognizable formal power series on trees. *Theoretical Computer Science*, 18:115–148, 1982.
6 Jean Berstel and Christophe Reutenauer. *Rational Series and Their Languages*, volume 12 of *EATCS Monographs in Theoretical Computer Science*. Springer, 1988.
7 Johanna Björklund, Frank Drewes, and Niklas Zechner. An efficient best-trees algorithm for weighted tree automata over the tropical semiring. In Adrian-Horia Dediu, Enrico Formenti, Carlos Martín-Vide, and Bianca Truthe, editors, *9th International Conference on Language and Automata Theory and Applications (LATA)*, volume 8977 of *Lecture Notes in Computer Science*, pages 97–108. Springer, 2015.
8 Meera Blattner and Tom Head. Automata that recognize intersections of free submonoids. *Information and Control*, 35(3):173–176, 1977.
9 Alexander Bockmayr, Volker Weispfenning, and Michael Maher. Solving numerical constraints. In Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, volume 1, chapter 12, pages 751–842. Elsevier and MIT Press, 2001.
10 Björn Borchardt. A pumping lemma and decidability problems for recognizable tree series. *Acta Cybernetica*, 16(4):509–544, 2004.
11 Matthias Büchse, Jonathan May, and Heiko Vogler. Determinization of weighted tree automata using factorizations. *Journal of Automata, Languages and Combinatorics*, 15(3/4):229–254, 2010.
12 Laure Daviaud, Pierre Guillon, and Glenn Merlet. Comparison of max-plus automata and joint spectral radius of tropical matrices. In Kim G. Larsen, Hans L. Bodlaender, and Jean-François Raskin, editors, *42nd International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 83 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 19:1–19:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017.
13 Manfred Droste, Werner Kuich, and Heiko Vogler, editors. *Handbook of Weighted Automata*. EATCS Monographs in Theoretical Computer Science. Springer, 2009.
14 Zoltán Ésik and Werner Kuich. Formal tree series. *Journal of Automata, Languages and Combinatorics*, 8(2):219–285, 2003.
15 Javier Esparza, Pierre Ganty, Stefan Kiefer, and Michael Luttenberger. Parikh's theorem: A simple and direct automaton construction. *Information Processing Letters*, 111(12):614–619, 2011.
16 Emmanuel Filiot, Ismaël Jecker, Nathan Lhote, Guillermo A. Pérez, and Jean-François Raskin. On delay and regret determinization of max-plus automata. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12. IEEE Computer Society, 2017.
17 Zoltán Fülöp and Heiko Vogler. Weighted tree automata and tree transducers. In Manfred Droste, Werner Kuich, and Heiko Vogler, editors, *Handbook of Weighted Automata*, EATCS Monographs in Theoretical Computer Science, chapter 9, pages 313–403. Springer, 2009.
18 Kōsaburō Hashiguchi. Algorithms for determining relative star height and star height. *Information and Computation*, 78(2):124–169, 1988.

19 Kōsaburō Hashiguchi, Kenichi Ishiguro, and Shūji Jimbo. Decidability of the equivalence problem for finitely ambiguous finance automata. *International Journal of Algebra and Computation*, 12(3):445–461, 2002.

20 Daniel Kirsten. A Burnside approach to the termination of Mohri's algorithm for polynomially ambiguous min-plus-automata. *Informatique Théorique et Applications*, 42(3):553–581, 2008.

21 Daniel Kirsten and Sylvain Lombardy. Deciding unambiguity and sequentiality of polynomially ambiguous min-plus automata. In Susanne Albers and Jean-Yves Marion, editors, *26th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 3 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 589–600. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2009.

22 Ines Klimann, Sylvain Lombardy, Jean Mairesse, and Christophe Prieur. Deciding unambiguity and sequentiality from a finitely ambiguous max-plus automaton. *Theoretical Computer Science*, 327(3):349–373, 2004.

23 Jan Komenda, Sébastien Lahaye, Jean-Louis Boimond, and Ton van den Boom. Max-plus algebra in the history of discrete event systems. *Annual Reviews in Control*, 45:240–249, 2018.

24 Adam Koprowski and Johannes Waldmann. Max/plus tree automata for termination of term rewriting. *Acta Cybernetica*, 19(2):357–392, 2009.

25 Stephan Kreutzer and Cristian Riveros. Quantitative monadic second-order logic. In *28th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 113–122. IEEE Computer Society, 2013.

26 Daniel Krob. The equality problem for rational series with multiplicities in the tropical semiring is undecidable. *International Journal of Algebra and Computation*, 4(3):405–426, 1994.

27 Werner Kuich and Arto Salomaa. *Semirings, Automata, Languages*, volume 5 of *EATCS Monographs in Theoretical Computer Science*. Springer, 1986.

28 Filip Mazowiecki and Cristian Riveros. Pumping lemmas for weighted automata. In Rolf Niedermeier and Brigitte Vallée, editors, *35th Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 96 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 50:1–50:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018.

29 Mehryar Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–311, 1997.

30 George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, 1988.

31 Rohit Jivanlal Parikh. On context-free languages. *Journal of the ACM*, 13(4):570–581, 1966.

32 Erik Paul. On finite and polynomial ambiguity of weighted tree automata. In Srečko Brlek and Christophe Reutenauer, editors, *20th International Conference on Developments in Language Theory (DLT)*, volume 9840 of *Lecture Notes in Computer Science*, pages 368–379. Springer, 2016.

33 Erik Paul. The equivalence, unambiguity and sequentiality problems of finitely ambiguous max-plus tree automata are decidable. In Kim G. Larsen, Hans L. Bodlaender, and Jean-François Raskin, editors, *42nd International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 83 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 53:1–53:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017.

34 Erik Paul. Finite sequentiality of unambiguous max-plus tree automata. In Rolf Niedermeier and Christophe Paul, editors, *36th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 126 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 55:1–55:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019.

35 Slav Petrov. Latent variable grammars for natural language parsing. In *Coarse-to-Fine Natural Language Processing*, Theory and Applications of Natural Language Processing, chapter 2, pages 7–46. Springer, 2012.

36 Michael O. Rabin and Dana S. Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3(2):114–125, 1959.

37 Frank P. Ramsey. On a problem of formal logic. *Proceedings of the London Mathematical Society*, series 2, 30:264–286, 1930.

38 Jacques Sakarovitch. *Elements of Automata Theory.* Cambridge University Press, 2009.

39 Arto Salomaa and Matti Soittola. *Automata-Theoretic Aspects of Formal Power Series.* Texts and Monographs in Computer Science. Springer, 1978.

40 Marcel-Paul Schützenberger. On the definition of a family of automata. *Information and Control*, 4(2-3):245–270, 1961.

41 Helmut Seidl. On the finite degree of ambiguity of finite tree automata. *Acta Informatica*, 26(6):527–542, 1989.

42 Imre Simon. Limited subsets of a free monoid. In *19th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 143–150. IEEE Computer Society, 1978.

43 Imre Simon. Recognizable sets with multiplicities in the tropical semiring. In Michal P. Chytil, Ladislav Janiga, and Václav Koubek, editors, *13th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 324 of *Lecture Notes in Computer Science*, pages 107–120. Springer, 1988.

44 Johannes Waldmann. Weighted automata for proving termination of string rewriting. *Journal of Automata, Languages and Combinatorics*, 12(4):545–570, 2007.

45 Andreas Weber and Helmut Seidl. On the degree of ambiguity of finite automata. *Theoretical Computer Science*, 88(2):325–349, 1991.