Finite Sequentiality of Unambiguous Max-Plus Tree Automata

Erik Paul

January 12, 2020

Abstract

We show the decidability of the finite sequentiality problem for unambiguous max-plus tree automata. A max-plus tree automaton is called unambiguous if there is at most one accepting run on every tree. 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.

1 Introduction

A max-plus automaton is a finite automaton which assigns real numbers to words over a given alphabet. The transitions of a max-plus automaton each carry a weight from the real numbers. To every run of the automaton, a weight is associated by summing over the weights of the transitions which constitute the run. The weight of a word is given by the maximum over the weights of all runs on this word.

More generally, max-plus automata and their min-plus counterparts are weighted automata [38, 37, 26, 7, 13] over the max-plus or min-plus semiring. Min-plus automata were originally introduced by Imre Simon as a means to show the decidability of the *finite power property* [41, 42]. Since their introduction, max-plus and min-plus automata have enjoyed a continuing interest [25, 18, 22, 8, 12, 15] and they have been employed in many different contexts. To only name some examples, they can be used to determine the star height of a language [17], to prove the termination of some string rewriting systems [43], and to model certain discrete event systems [23]. Additionally, they appear in the context of natural language processing [29], where for reasons of numerical stability, probabilities are often computed in the min-plus semiring as negative log-likelihoods.

A very prominent open question about max-plus automata is the *sequentiality problem*, the problem of deciding whether for an arbitrary max-plus automaton there exists a deterministic equivalent. A max-plus automaton is called *deterministic* or *sequential* if at most one of its states is initial and for each pair of a state and an input symbol, there is at most one valid transition into a next state. Although the decidability of this problem is unknown for max-plus automata in general, it is known to be decidable for the subclasses of *unambiguous* [29], *finitely ambiguous* [22], and even *polynomially ambiguous* [21] automata. A max-plus automaton is called *unambiguous* if there exists at most one accepting run on every word. It is called *finitely ambiguous* if the number of runs on each word is bounded by a global constant. If on every word the number of accepting runs is bounded polynomially in the length of the word, the automaton is said to be *polynomially ambiguous*. Note that the ambiguity of a max-plus automaton is a decidable property, as it is easily reduced to deciding the ambiguity of a finite automaton. Deciding the sequentiality of a finite automaton is trivial, polynomial time algorithms for deciding the unambiguity, the finite ambiguity, and the polynomial ambiguity of a finite automaton can be found in [9, 44, 40]. Furthermore, the classes of functions definable by deterministic, unambiguous, finitely ambiguous, polynomially ambiguous, and arbitrary max-plus automata form a strictly ascending hierarchy [22, 19, 28].

A decidability problem which is closely related to the sequentiality problem is the *finite sequentiality problem.* The finite sequentiality problem asks whether a given max-plus automaton can be represented as a pointwise maximum of finitely many deterministic max-plus automata. In [18], it was left as an open question to determine the decidability of the finite sequentiality problem for finitely ambiguous max-plus automata. It was shown only recently that for the classes of unambiguous as well as finitely ambiguous automata, the finite sequentiality problem is decidable [4, 3]. The class of functions which allow a finitely sequential representation by max-plus automata lies strictly between the classes of functions definable by deterministic and by finitely ambiguous max-plus automata, and it is incomparable to the class of functions definable by unambiguous max-plus automata [22].

In this paper, we show that the finite sequentiality problem is decidable for unambiguous max-plus tree automata. Max-plus tree automata are a generalization of max-plus automata and operate on trees instead of words. In particular, max-plus tree automata are weighted tree automata [1, 6, 14, 16] over the max-plus semiring. Applications for max-plus tree automata include proving the termination of certain term rewriting systems [24], and they are also commonly employed in natural language processing [33] in the form of *probabilistic context-free grammars*. Our approach to show the decidability of the finite sequentiality problem employs ideas from [4]. In [4], the *fork property* is shown to be a decidable criterion to determine the existence of a finitely sequential equivalent. More precisely, unambiguous max-plus word automata are shown to possess a finitely sequential representation if and only if they do not satisfy the fork property. It is shown elementarily that an unambiguous automaton satisfying the fork property cannot possess a finitely sequential equivalent. The proof for the existence of a finitely sequential representation in case that the fork property is not satisfied, on the other hand, relies on the construction of finitely many unambiguous max-plus automata whose pointwise maximum is equivalent to the original automaton, and which all satisfy the *twins property*. It was shown by Mohri [29] that an unambiguous max-plus automaton which satisfies the twins property is determinizable. A finitely sequential representation is thus found by determinizing the unambiguous automata.

For tree automata, we generalize the fork property to the *tree fork property* by adding a condition which accounts for the nonlinear structure of trees. We then prove that an unambiguous max-plus tree automaton possesses a finitely sequential representation if and only if it does not satisfy the tree fork property. As in the word case, the most challenging part of the proof is to show the existence of a finitely sequential representation whenever the tree fork property is not satisfied. Like in the proof for word automata, we construct finitely many unambiguous max-plus tree automata which possess a deterministic equivalent. However, we need to take a different approach in order to obtain these automata. In [4], a modified *Schützenberger covering* [39, 35, 36] is first constructed from the unambiguous max-plus automaton, from which in turn an automaton is constructed which monitors the occurrence of certain states of the modified Schützenberger covering. This latter automaton is then decomposed into the finitely many unambiguous automata. This approach, however, is not applicable to trees, as the monitoring of states requires all relevant states to occur linearly. This happens trivially for word automata due to the inherent linear structure of words, but for tree automata examples can be found where relevant states occur nonlinearly. The approach we use here relies on constructing a max-plus automaton which tracks certain pairs of states of the original automaton. When applied to word automata, this immediately yields an automaton which can be decomposed into the desired unambiguous automata. Unfortunately, for tree automata this tracking of pairs of states again fails due to states occurring nonlinearly. Surprisingly however, our construction can be applied to the Schützenberger covering of the original tree automaton, as the states relevant for tracking all occur pairwise linearly in the Schützenberger covering. The most difficult part of our proof is to show that the Schützenberger covering indeed has the property we just indicated.

An extended abstract of this paper appeared as [32]. This paper differs from it in the following way. First, full proofs are included. Second, we combine some known results and ideas to obtain the decidability of the sequentiality problem for unambiguous max-plus tree automata. This result has never been stated explicitly, but follows rather easily from the main result of [11] and an idea from [29]. As this result fits the theme of our paper quite nicely, we decided to include it. Third, we show that it is decidable in PSPACE whether an unambiguous max-plus tree automaton satisfies the tree fork property. In [32], we only outlined the decidablity of the tree fork property. Fourth, we now give some examples to better illustrate the properties of the Schützenberger covering. Finally, we have added a section in which we investigate additional properties of the Schützenberger covering. The main motivation for this last section is to provide further tools for a possible generalization of our result, for example to finitely ambiguous max-plus tree automata.

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 $\mathrm{dom}(f)$, and Y the *range* of f, denoted by $\mathrm{range}(f)$. 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 $f(x) = g(x)$.

Let $\mathbb{N} = \{0, 1, 2, \ldots\}$. 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 $\leq_p$ and totally ordered with respect to the lexicographic ordering $\leq_l$. 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 $\{v \in \mathrm{pos}(t) \mid v \leq_p w\}$ is called a *branch* of t.

Now let $s, t \in T_\Gamma$ and $w \in \mathrm{pos}(t)$. The *subtree of* t *at* w, denoted by $t\restriction_w$, is a Γ-tree defined as follows. We let $\mathrm{pos}(t\restriction_w) = \{v \in \mathbb{N}^* \mid wv \in \mathrm{pos}(t)\}$ and for $v \in \mathrm{pos}(t\restriction_w)$, we let $\mathrm{label}_{t\restriction_w}(v) = t(wv)$.

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 $\mathrm{pos}(t\langle s \to w\rangle) = (\mathrm{pos}(t) \setminus \{v \in \mathrm{pos}(t) \mid w \leq_p v\}) \cup \{wv \mid v \in \mathrm{pos}(s)\}$. For $v \in \mathrm{pos}(t\langle s \to w\rangle)$, we let $\mathrm{label}_{t\langle s \to w\rangle}(v) = s(u)$ if $v = wu$ for some $u \in \mathrm{pos}(s)$, and otherwise $\mathrm{label}_{t\langle s \to w\rangle}(v) = t(v)$.

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 $\mathrm{pos}(t) = \{\varepsilon\} \cup \{iw \mid i \in \{1, \ldots, m\}, w \in \mathrm{pos}(t_i)\}$, $\mathrm{label}_t(\varepsilon) = a$, and $\mathrm{label}_t(iw) = t_i(w)$. For $a \in \Gamma^{(0)}$, the tree $a()$ is abbreviated by a.

For a ranked alphabet Γ, a tree over the alphabet $\Gamma_\diamond = (\Gamma \cup \{\diamond\}, \mathrm{rk}_\Gamma \cup \{\diamond \mapsto 0\})$ is called a Γ*-context*. Let $t \in T_{\Gamma_\diamond}$ be a Γ-context and let $w_1, \ldots, w_n \in \mathrm{pos}(t)$ 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 $t(t_1, \ldots, t_n) = t\langle t_1 \to \Diamond_1(t)\rangle \ldots \langle t_n \to \Diamond_n(t)\rangle$ 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 only consider the *max-plus semiring* $\mathbb{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \max, +, -\infty, 0)$ where the sum and the product operations are max and $+$, respectively, extended to $\mathbb{R} \cup \{-\infty\}$ in the usual way.

A *max-plus weighted bottom-up finite state tree automaton (short: max-plus-WTA) over* 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), $\mu\colon \bigcup_{m=0}^{\mathrm{rk}(\Gamma)} Q^m \times \Gamma^{(m)} \times Q \to \mathbb{R}_{\max}$ (the function of transition weights), and $\nu \colon Q \to \mathbb{R}_{\max}$ (the function of final weights). We define $\Delta_\mathcal{A} = \mathrm{dom}(\mu)$. A tuple $(\bar{p}, a, q) \in \Delta_\mathcal{A}$ is called a *transition* and $(\bar{p}, a, q)$ is called *valid* if $\mu(\bar{p}, a, q) \neq -\infty$. A state $q \in Q$ is called *final* if $\nu(q) \neq -\infty$.

For a tree $t \in T_\Gamma$, a mapping $r \colon \mathrm{pos}(t) \to Q$ is called a *quasi-run of* $\mathcal{A}$ *on* t. For a quasi-run r on t and a position $w \in \mathrm{pos}(t)$ 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 $w \in \mathrm{pos}(t)$ 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 $\mathrm{Run}_\mathcal{A}(t)$ and $\mathrm{Acc}_\mathcal{A}(t)$ 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 $\mathrm{Run}_\mathcal{A}(t, q)$ the set of all runs $r \in \mathrm{Run}_\mathcal{A}(t)$ such that $r(\varepsilon) = q$.

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

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

The *behavior of* $\mathcal{A}$, denoted by $[\![\mathcal{A}]\!]$, is the mapping defined for every $t \in T_\Gamma$ by

$$[\![\mathcal{A}]\!](t) = \max_{r \in \mathrm{Acc}_\mathcal{A}(t)} (\mathrm{wt}_\mathcal{A}(t, r) + \nu(r(\varepsilon))),$$

where the maximum over the empty set is $-\infty$ by convention.

For a max-plus-WTA $\mathcal{A} = (Q, \Gamma, \mu, \nu)$, a run of $\mathcal{A}$ on a Γ-context t is a run of the max-plus-WTA $\mathcal{A}' = (Q, \Gamma_\diamond, \mu', \nu)$ on t, where $\mu'(\diamond, q) = 0$ 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 $q_0, \dots, q_n$, we denote by $\mathrm{Run}^{\Diamond}_{\mathcal{A}}(q_1, \dots, q_n, t, q_0)$ 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 $i \in \{1, \dots, n\}$. For a Γ-word s, we write $p \xrightarrow{s|x} q$ if there exists a run $r \in \mathrm{Run}^{\Diamond}_{\mathcal{A}}(p, s, q)$ with $\mathrm{wt}^{\Diamond}_{\mathcal{A}}(s,r) = x$. In this case, r is said to *realize* $p \xrightarrow{s|x} q$. Note that $r \in \mathrm{Run}^{\Diamond}_{\mathcal{A}}(s)$ implies $x \neq -\infty$.

Similar to trees, we define restrictions, substitutions, and powers of runs as follows. Let $t, s \in T_\Gamma$, $r \in \mathrm{Run}_{\mathcal{A}}(t)$, $w \in \mathrm{pos}(t)$, and $r_s \in \mathrm{Run}_{\mathcal{A}}(s)$ with $r_s(\varepsilon) = r(w)$. Then we define $r\restriction_w \in \mathrm{Run}_{\mathcal{A}}(t\restriction_w)$ by $r\restriction_w(v) = r(wv)$ for every $v \in \mathrm{pos}(t\restriction_w)$. We define $r\langle r_s \to w\rangle \in \mathrm{Run}_{\mathcal{A}}(t\langle s \to w\rangle)$ by $r\langle r_s \to w\rangle(v) = r_s(u)$ if $v = wu$ for some $u \in \mathrm{pos}(s)$, and $r\langle r_s \to w\rangle(v) = r(v)$ otherwise. For a Γ-word s and a run $r \in \mathrm{Run}^{\Diamond}_{\mathcal{A}}(s)$ with $r(\varepsilon) = r(\Diamond_1(s))$, we let $v = \Diamond_1(s)$ and define $r^{0\langle v\rangle} = \{\varepsilon \mapsto r(\varepsilon)\}$ and $r^{n+1\langle v\rangle} = r\langle r^{n\langle v\rangle} \to v\rangle \in \mathrm{Run}^{\Diamond}_{\mathcal{A}}(s^{n+1})$ for $n \geq 0$.

For a max-plus-WTA $\mathcal{A}$, we define a relation $\leq$ on Q by $p \leq q$ iff there exists a Γ-word $s \in T_{\Gamma_\Diamond}$ such that $\mathrm{Run}^{\Diamond}_{\mathcal{A}}(q, s, p) \neq \emptyset$. We call $\mathcal{A}$ *trim* if for every $p \in Q$ there exists $t \in T_\Gamma$, $r \in \mathrm{Acc}(t)$, 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 $[\![\mathcal{A}]\!]$.

A max-plus-WTA $\mathcal{A}$ is called *deterministic* or *sequential* if for every $m \geq 0$, $a \in \Gamma^{(m)}$, and $\bar{p} \in Q^m$, there exists at most one $q \in Q$ with $\mu(\bar{p}, a, q) \neq -\infty$. We call $\mathcal{A}$ *unambiguous* if $|\mathrm{Acc}_{\mathcal{A}}(t)| \leq 1$ for every $t \in T_\Gamma$. We call the behavior $[\![\mathcal{A}]\!]$ of $\mathcal{A}$ *finitely sequential* if there exist finitely many deterministic max-plus-WTA $\mathcal{A}_1, \dots, \mathcal{A}_n$ over Γ such that $[\![\mathcal{A}]\!] = \max_{i=1}^n [\![\mathcal{A}_i]\!]$, where the maximum is taken pointwise.

3 Main Result

We will show that for an unambiguous max-plus-WTA $\mathcal{A}$, it is decidable whether its behavior $[\![\mathcal{A}]\!]$ is finitely sequential. Moreover, if it is finitely sequential, we will obtain that the deterministic max-plus-WTA $\mathcal{A}_1, \dots, \mathcal{A}_n$ can be effectively constructed. For this, we follow ideas from [4], where the decidability of the finite sequentiality problem was proved for unambiguous max-plus word automata.

The general outline of our proof is similar to that of [4] and presents itself as follows. We introduce the *tree fork property* and show that it is decidable whether an unambiguous max-plus-WTA $\mathcal{A}$ satisfies this property. Then we show that the behavior of an unambiguous max-plus-WTA is finitely sequential if and only if it does not satisfy the tree fork property. In conclusion, we obtain the decidability of the finite sequentiality problem for unambiguous max-plus-WTA.

Elementary proof methods can be used to show that $[\![\mathcal{A}]\!]$ is not finitely sequential if $\mathcal{A}$ satisfies the tree fork property. On the other hand, if $\mathcal{A}$ does not satisfy the tree fork property, we show how to construct finitely many unambiguous max-plus-WTA whose pointwise maximum is $[\![\mathcal{A}]\!]$, and which all satisfy the *twins property* [29]. Every unambiguous max-plus-WTA which satisfies the twins property possesses an effectively constructable deterministic equivalent [11]. Thus, we obtain finitely many deterministic max-plus-WTA whose pointwise maximum is $[\![\mathcal{A}]\!]$, which is hence finitely sequential.

We begin by showing that the *Lipschitz property* of deterministic max-plus word automata [22, 29] also holds for deterministic max-plus tree automata. On words, this Lipschitz property can be formulated follows. Let $\mathcal{A}$ be a deterministic max-plus word automaton and let L be the largest weight, in terms of absolute value, occurring in $\mathcal{A}$ (excluding $-\infty$). Then for two words $w_1 = uv_1$ and $w_2 = uv_2$ which have an accepting run in $\mathcal{A}$, the difference between $[\![\mathcal{A}]\!](w_1)$ and $[\![\mathcal{A}]\!](w_2)$ can be at most $|L|(|v_1| + |v_2| + 2)$. This is clear since the unique runs of $\mathcal{A}$ on w_1 and w_2 will be identical on the prefix u, and then with every remaining letter of each word the difference between both runs cannot grow more than $|L|$. For deterministic max-plus-WTA, we can show a similar statement as follows.

Lemma 1 (c.f. [22, End of Section 2.4][29, Section 3.2])**.** *Let $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ be a deterministic max-plus-WTA, let $X = (\mu(\Delta_{\mathcal{A}}) \cup \nu(Q)) \setminus \{-\infty\}$, and let $L = \max_{x \in X} |x|$. Furthermore, let $t_1, t_2 \in T_\Gamma$ be two trees with $[\![\mathcal{A}]\!](t_1) \neq -\infty$ and $[\![\mathcal{A}]\!](t_2) \neq -\infty$ and let $w_1 \in \mathrm{pos}(t_1)$ and $w_2 \in \mathrm{pos}(t_2)$ be two positions such that $t_1\restriction_{w_1} = t_2\restriction_{w_2}$. Then with $t = t_1\restriction_{w_1}$ we have*

$$|[\![\mathcal{A}]\!](t_1) - [\![\mathcal{A}]\!](t_2)| \leq L(|t_1| + |t_2| - 2|t| + 2).$$

Proof. Since $\mathcal{A}$ is deterministic, there exists exactly one run $r_1 \in \mathrm{Acc}_{\mathcal{A}}(t_1)$ and exactly one run $r_2 \in \mathrm{Acc}_{\mathcal{A}}(t_2)$. Likewise, there exists exactly one run $r \in \mathrm{Run}_{\mathcal{A}}(t)$. Due to $t_1\restriction_{w_1} = t_2\restriction_{w_2} = t$, we thus have

$r_1\restriction_{w_1} = r_2\restriction_{w_2} = r$. It follows that

$$\begin{aligned}
&|[\![\mathcal{A}]\!](t_1) - [\![\mathcal{A}]\!](t_2)| \\
&= \Big| \sum_{w \in \mathrm{pos}(t_1)} \mu(\mathbb{t}(t_1, r_1, w)) + \nu(r_1(\varepsilon)) - \Big(\sum_{w \in \mathrm{pos}(t_2)} \mu(\mathbb{t}(t_2, r_2, w)) + \nu(r_2(\varepsilon)) \Big) \Big| \\
&= \Big| \sum_{\substack{w \in \mathrm{pos}(t_1) \\ \neg(w_1 \leq_p w)}} \mu(\mathbb{t}(t_1, r_1, w)) + \nu(r_1(\varepsilon)) - \Big(\sum_{\substack{w \in \mathrm{pos}(t_2) \\ \neg(w_2 \leq_p w)}} \mu(\mathbb{t}(t_2, r_2, w)) + \nu(r_2(\varepsilon)) \Big) \Big| \\
&\leq L(|t_1| - |t| + 1) + L(|t_2| - |t| + 1) \\
&= L(|t_1| + |t_2| - 2|t| + 2).
\end{aligned}$$

□

Next, we recall the twins property. Let Γ be a ranked alphabet. We begin by introducing the concepts of *siblings* and *twins*. Intuitively, two states are called *siblings* if they can be "reached" by the same tree. Two siblings are called *twins* if for every Γ-word which can "loop" in both states, the maximal weight for the loop is the same in both states.

Definition 2. Let $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ be a max-plus-WTA. Two states $p, q \in Q$ are called *siblings* if there exists a tree $u \in T_\Gamma$ such that $\mathrm{Run}_\mathcal{A}(u, p) \neq \emptyset$ and $\mathrm{Run}_\mathcal{A}(u, q) \neq \emptyset$. We recall that $\mathrm{Run}_\mathcal{A}(u, p)$ and $\mathrm{Run}_\mathcal{A}(u, q)$ contain only valid runs.

Two siblings p, q are called *twins* if for every Γ-word s and weights

$$x = \max_{r \in \mathrm{Run}^\diamond_\mathcal{A}(p,s,p)} \mathrm{wt}^\diamond_\mathcal{A}(s, r) \qquad\qquad y = \max_{r \in \mathrm{Run}^\diamond_\mathcal{A}(q,s,q)} \mathrm{wt}^\diamond_\mathcal{A}(s, r),$$

we have $x = y$ whenever $x \neq -\infty$ and $y \neq -\infty$ holds. Here, the maximum over the empty set is $-\infty$ by convention.

A max-plus-WTA is said to satisfy the *twins property* if all of its siblings are twins. For unambiguous max-plus-WTA, the twins property is a criterion for deciding the sequentiality problem. An unambiguous max-plus-WTA possesses a deterministic equivalent if and only if it satisfies the twins property. For words, this result is due to [29, Theorem 12], for trees, we cite the following theorem which states that the twins property is a sufficient condition for determinizability.

Theorem 3 ([11, Lemma 5.10])**.** *Let $\mathcal{A}$ be a trim unambiguous max-plus-WTA. If $\mathcal{A}$ satisfies the twins property, there exists a deterministic max-plus-WTA $\mathcal{A}'$ with $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$ which can be effectively constructed.*

The converse, namely that the twins property is also a necessary condition for determinizability, follows from the Lipschitz property of deterministic max-plus automata. For max-plus word automata, consider the following. If an unambiguous max-plus word automaton $\mathcal{A}$ does not satisfy the twins property, we can find states p and q which are siblings and not twins. We assume that our witnesses for this are u and s as above. Then we consider words of the form $w_1 = us^N v_p$ and $w_2 = us^N v_q$, where v_p and v_q are two fixed words which lead from p and q, respectively, to some final state. For every fixed $L \in \mathbb{R}$, we can choose N sufficiently large to ensure that $|[\![\mathcal{A}]\!](w_1) - [\![\mathcal{A}]\!](w_2)| > |L|(|v_p| + |v_q| + 2)$. Due to the Lipschitz property of deterministic max-plus automata, it is thus not possible to determinize $\mathcal{A}$ if it does not satisfy the twins property. For trees, we can proceed in the same way.

Lemma 4. *Let $\mathcal{A}$ be a trim unambiguous max-plus-WTA. If there exists a deterministic max-plus-WTA $\mathcal{A}'$ with $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$, then $\mathcal{A}$ satisfies the twins property.*

Proof. We follow the idea for the proof of [29, Theorem 9]. Let $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ be a trim unambiguous max-plus-WTA and let $p, q \in Q$ be siblings, i.e., there exists a tree $u \in T_\Gamma$ and runs $r^p \in \mathrm{Run}_\mathcal{A}(u, p)$ and $r^q \in \mathrm{Run}_\mathcal{A}(u, q)$. Let $s \in T_{\Gamma_\diamond}$ be a Γ-word such that $p \xrightarrow{s|x} p$ and $q \xrightarrow{s|y} q$ for weights $x, y \in \mathbb{R}$. Since $\mathcal{A}$ is trim, there exist Γ-words $\hat{u}_p, \hat{u}_q \in T_{\Gamma_\diamond}$ such that $p \xrightarrow{\hat{u}_p|z_p} p'$ and $q \xrightarrow{\hat{u}_q|z_q} q'$ for two final states $p', q' \in Q$ and weights $z_p, z_q \in \mathbb{R}$. We let $\kappa_p = \mathrm{wt}_\mathcal{A}(u, r^p) + z_p + \nu(p')$ and $\kappa_q = \mathrm{wt}_\mathcal{A}(u, r^q) + z_q + \nu(q')$ and for $n \geq 1$ define the trees $t_p^{(n)} = \hat{u}_p(s^n(u))$ and $t_q^{(n)} = \hat{u}_q(s^n(u))$. Due to the unambiguity of $\mathcal{A}$, we see that for every $n \geq 1$ we have

$$\begin{aligned}
[\![\mathcal{A}]\!](t_p^{(n)}) &= \kappa_p + nx \\
[\![\mathcal{A}]\!](t_q^{(n)}) &= \kappa_q + ny.
\end{aligned}$$

Assume that there exists a deterministic max-plus-WTA $\mathcal{A}'$ with $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$. Then by Lemma 1, there exists $L \in \mathbb{R}$ such that for all $n \geq 1$ we have

$$|[\![\mathcal{A}]\!](t_p^{(n)}) - [\![\mathcal{A}]\!](t_q^{(n)})| \leq |L|(|\hat{u}_p| + |\hat{u}_q| + 2).$$

From the equations above we thus obtain that for every $n \geq 1$ we have

$$|\kappa_p - \kappa_q + n(x - y)| \leq |L|(|\hat{u}_p| + |\hat{u}_q| + 2).$$

This can only hold if $x = y$. It follows that $\mathcal{A}$ satisfies the twins property. □

The twins property is decidable for both max-plus word automata [2, 5, 29, 30, 20] and max-plus tree automata [10, Section 3]. Thus, by combining Lemma 4 with the results from [11] (see Theorem 3) and [10], we obtain the decidability of the sequentiality problem for unambiguous max-plus tree automata.

Theorem 5. *For an unambiguous max-plus-WTA $\mathcal{A}$ it is decidable whether there exists a deterministic max-plus-WTA $\mathcal{A}'$ with $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$. If such an automaton $\mathcal{A}'$ exists, it can be effectively constructed.*

Deciding whether a max-plus word automaton satisfies the twins property is PSPACE-complete [20]. For max-plus tree automata, the problem is thus PSPACE-hard, but no upper complexity bound is stated in [10]. Note that in general, it is undecidable whether two given siblings are twins [20], but for so-called *cycle-unambiguous* max-plus automata, it was shown to be decidable on both words [2, Section 4] and trees [11, Section 5.4]. A max-plus tree automaton $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ is called *cycle-unambiguous* if for every Γ-word $s \in T_{\Gamma_\diamond}$ and every state $q \in Q$, there is at most one run which loops s in q, i.e., the set $\mathrm{Run}^\diamond_{\mathcal{A}}(q, s, q)$ is either a singleton or empty. It is easy to see that every trim unambiguous max-plus tree automaton is cycle-unambiguous. Thus, for every two states of an unambiguous max-plus tree automaton, it is decidable whether they are twins. As we will employ the reasoning from [11] in more detail, we provide a short direct proof.

Lemma 6 ([11, Section 5.4])**.** *Let $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ be a cycle-unambiguous max-plus-WTA. Two states $p, q \in Q$ are siblings if and only if there exists a tree $u \in T_\Gamma$ of height at most $|Q|^2$ such that $\mathrm{Run}_{\mathcal{A}}(u, p) \neq \emptyset$ and $\mathrm{Run}_{\mathcal{A}}(u, q) \neq \emptyset$. Two siblings $p, q \in Q$ are not twins if and only if there exists a Γ-word s of height at most $4|Q|^2$ such that $p \xrightarrow{s|x} p$ and $q \xrightarrow{s|y} q$ with $x \neq y$.*

Proof. Let $p, q \in Q$ be two states. First, to check whether p and q are siblings, we see as follows that it suffices to check whether they can both be reached by a tree u of height at most $|Q|^2$. Assume we have a tree $u \in T_\Gamma$ and two runs $r^p \in \mathrm{Run}_{\mathcal{A}}(t, p)$ and $r^q \in \mathrm{Run}_{\mathcal{A}}(t, q)$. If $\mathrm{height}(u) > |Q|^2$, then by pigeon hole principle, we can find a simultaneous loop in r^p and r^q; that is, we can find two positions $w_1 <_p w_2$ in u with $r^p(w_1) = r^p(w_2)$ and also $r^q(w_1) = r^q(w_2)$. By removing everything between w_1 and w_2 from u, we obtain the smaller tree $u\langle u\restriction_{w_2} \to w_1\rangle$ which still reaches p and q.

If p and q are siblings, we see as follows that we only need to check Γ-words s of height at most $4|Q|^2$ to decide whether p and q are twins. Assume p and q are not twins and our witness for this is the Γ-word s with $\mathrm{height}(s) > 4|Q|^2$. Let $r_p \in \mathrm{Run}^\diamond_{\mathcal{A}}(p, s, p)$ be the run on s which loops in p with weight $x = \mathrm{wt}^\diamond_{\mathcal{A}}(s, r_p)$ and let $r_q \in \mathrm{Run}^\diamond_{\mathcal{A}}(q, s, q)$ be the run on s which loops in q with weight $y = \mathrm{wt}^\diamond_{\mathcal{A}}(s, r_q)$. Furthermore, let $w \in \mathrm{pos}(s)$ with $|w| = \mathrm{height}(s)$ and let $w' \in \mathrm{pos}(s)$ be the longest common prefix of w and $\diamondsuit_1(s)$. Then either $|w'| > 2|Q|^2$ or $|w| - |w'| > 2|Q|^2$, or both. In the first case, there exist two disjoint simultaneous loops in r_p and r_q above $\diamondsuit_1(s)$. More precisely, by pigeon hole principle we can find positions $w_1 <_p w_2 \leq_p w_3 <_p w_4$ with $w_4 \leq_p w' \leq_p \diamondsuit_1(s)$ in s for which $(r^p(w_1), r^q(w_1)) = (r^p(w_2), r^q(w_2))$ and $(r^p(w_3), r^q(w_3)) = (r^p(w_4), r^q(w_4))$. In the second case, there exist two disjoint simultaneous loops in r_p and r_q which are prefix-independent from $\diamondsuit_1(s)$. That is, there exist positions $w_1 <_p w_2 \leq_p w_3 <_p w_4$ with $w' <_p w_1$ and $w_4 \leq_p w$ in s for which $(r^p(w_1), r^q(w_1)) = (r^p(w_2), r^q(w_2))$ and $(r^p(w_3), r^q(w_3)) = (r^p(w_4), r^q(w_4))$.

Let x_{12} and x_{34} be the weights of the loops in the run r^p, and let y_{12} and y_{34} be the weights of the loops in the run r^q. We obtain a smaller Γ-word s' and runs r'_p and r'_q of distinct weights which loop in p and q, respectively, by removing either one of the two loops or both loops as follows. If $x - x_{12} \neq y - y_{12}$, we remove the w_1-w_2 loop. Otherwise, if $x - x_{34} \neq y - y_{34}$, we remove the w_3-w_4 loop. If we have both $x - x_{12} = y - y_{12}$ and $x - x_{34} = y - y_{34}$, we obtain that $2x - x_{12} - x_{34} = 2y - y_{12} - y_{34}$. From $x \neq y$, it follows that $x - x_{12} - x_{34} \neq y - y_{12} - y_{34}$, so we remove both loops. From the cycle-unambiguity of $\mathcal{A}$, we see that these two runs are the only runs on the smaller Γ-word, so we have found a smaller witness. □

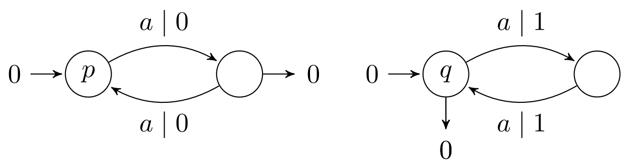


Figure 1: A max-plus word automaton $\mathcal{A}$ over the alphabet $\{a\}$ which is unambiguous, whose behavior is finitely sequential, but which does not satisfy the twins property as p and q are siblings but not twins. The behavior $[\![\mathcal{A}]\!]$ of $\mathcal{A}$ assigns 0 to all words of odd length and $|w|$ to all words w of even length.

There exist unambiguous max-plus automata which cannot be determinized, but whose behavior is finitely sequential [22, Section 3.1], see also Figure 1. Thus, for the finite sequentiality problem we inevitably have to deal with unambiguous automata in which not all siblings are twins. In the following, we will call two such states *rivals*. For cycle-unambiguous automata, thus in particular for unambiguous automata, the following definition is equivalent to being siblings and not twins.

Definition 7. Let $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ be a max-plus-WTA. Two states $p, q \in Q$ are called *rivals* if there exists a tree $u \in T_\Gamma$ such that $\mathrm{Run}_\mathcal{A}(u, p) \neq \emptyset$ and $\mathrm{Run}_\mathcal{A}(u, q) \neq \emptyset$ and a Γ-word s such that $p \xrightarrow{s|x} p$ and $q \xrightarrow{s|y} q$ with $x \neq y$. In this case, u and s are also said to be *witnesses* for the fact that p and q are rivals.

If $\mathcal{A}$ is cycle-unambiguous, p and q are rivals if and only if they are siblings and not twins as we do not have to consider a maximum over runs. Also note that by our definition of $\mathrm{Run}^\Diamond_\mathcal{A}(s)$, we have $x \neq -\infty$ and $y \neq -\infty$ above.

We now turn to the tree fork property which, as we will show, is satisfied by an unambiguous max-plus-WTA if and only if its behavior is not finitely sequential. The property consists of two separate conditions. The first condition intuitively states that there exist two rivals p and q and a Γ-word t which can loop in p, and which can also lead from p to q. The second condition states that there exist two rivals which can occur at prefix-independent positions.

Definition 8. Let $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ be a max-plus-WTA. 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 t with $p \xrightarrow{t|z_p} p$ and $p \xrightarrow{t|z_q} q$ for some weights $z_p, z_q \in \mathbb{R}$. In this case, t is also called 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 $r(\Diamond_2(t)) = q$.

The tree fork property can be regarded as an extension of the *fork property* which was introduced in [4] and which for max-plus word automata plays the same role as the tree fork property does for max-plus tree automata. Condition (i) is essentially a tree version of the fork property. Casually put, if we take only condition (i) and replace "Γ-word" by "word", we obtain the fork property. The automaton depicted in Figure 2 is unambiguous and satisfies the fork property. Condition (ii) is new and possesses no counterpart in the fork property.

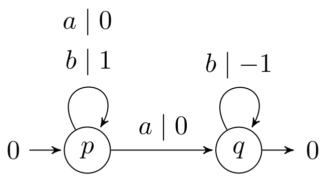


Figure 2: An unambiguous max-plus word automaton $\mathcal{A}$ over the alphabet $\{a, b\}$ which satisfies the fork property. With $u = a$ and $s = b$, we see that p and q are rivals, and a is a p-q-fork. All b's after the last a in a word are treated differently from the b's before the last a. A deterministic automaton cannot "guess" which a is the last in the word, and since there may be arbitrarily many a's in a word, even finitely many deterministic automata cannot compensate this inability to guess.

We have the following theorem which relates the tree fork property to the finite sequentiality problem.

Theorem 9. *Let $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ be a trim unambiguous max-plus-WTA over Γ. Then there exist deterministic max-plus-WTA $\mathcal{A}_1, \ldots, \mathcal{A}_n$ over Γ with $[\![\mathcal{A}]\!] = \max_{i=1}^{n} [\![\mathcal{A}_i]\!]$ if and only if $\mathcal{A}$ does not satisfy the tree fork property. If such automata $\mathcal{A}_1, \ldots, \mathcal{A}_n$ exist, they can be effectively constructed. Furthermore, there is a PSPACE-algorithm to decide whether $\mathcal{A}$ satisfies the tree fork property. In particular, the finite sequentiality problem is decidable for unambiguous max-plus-WTA.*

Proof. Here, we only show that it is decidable whether $\mathcal{A}$ satisfies the tree fork property. The existence of a PSPACE-algorithm for deciding the tree fork property is deferred to Lemma 10. The rest of the proof is deferred to Sections 3.1 and 3.2, where we show that the behavior of $\mathcal{A}$ is finitely sequential if and only if $\mathcal{A}$ does not satisfy the tree fork property.

To decide whether $\mathcal{A}$ satisfies condition (i), we first show that if there exists a p-q-fork t for two rivals p and q, then there exists a p-q-fork t' of height at most $2|Q|^2$. The argumentation for this is similar to the proof of Lemma 6 that the property of being siblings is decidable. Assume that t is a p-q-fork with $\text{height}(t) > 2|Q|^2$ and that r_p and r_q are runs that realize $p \xrightarrow{t|z_p} p$ and $p \xrightarrow{t|z_q} q$ for some weights $z_p, z_q \in \mathbb{R}$. We let $w \in \text{pos}(t)$ be a position with $|w| = \text{height}(t)$ and let w' be the longest common prefix of w and $\Diamond_1(t)$. Then either $|w'| > |Q|^2$ or $|w| - |w'| > |Q|^2$, or both. In the first case, there exist by pigeon hole principle two positions $w_1 <_p w_2$ in t with $w_2 \leq_p w' \leq_p \Diamond_1(t)$ and $(r_p(w_1), r_q(w_1)) = (r_p(w_2), r_q(w_2))$. In the second case, there exist two positions $w_1 <_p w_2$ in t with $w' <_p w_1$ and $(r_p(w_1), r_q(w_1)) = (r_p(w_2), r_q(w_2))$. By removing the part of t between w_1 and w_2, we obtain that $t' = t\langle t\restriction_{w_2} \to w_1 \rangle$ is a p-q-fork as well. Iterating this process, we obtain a p-q-fork of height at most $2|Q|^2$.

Next, we identify all pairs of rivals, which is possible since by Lemma 6, we can decide for every pair of states whether they are siblings and not twins. Then, for every pair of rivals p, q and all Γ-words t of height at most $2|Q|^2$, we check whether t is a p-q-fork. If this yields no p-q-fork, $\mathcal{A}$ does not satisfy condition (i).

In order to decide whether $\mathcal{A}$ satisfies condition (ii), we first compute the relation $\leq$ on Q. This is possible since Q is a finite set and $\leq$ is the smallest transitive and reflexive relation satisfying $\mu(q_1, \ldots, q_m, a, q_0) \neq -\infty \to q_0 \leq q_i$ for all transitions $(q_1, \ldots, q_m, a, q_0) \in \Delta_{\mathcal{A}}$ and $i \in \{1, \ldots, m\}$. Then, by the trimness of $\mathcal{A}$, condition (ii) is satisfied if and only if there exist two rivals p and q, a transition $(q_1, \ldots, q_m, a, q_0) \in \Delta_{\mathcal{A}}$ with $\mu(q_1, \ldots, q_m, a, q_0) \neq -\infty$, and indices $i, j \in \{1, \ldots, m\}$ with $i \neq j$, $q_i \leq p$, and $q_j \leq q$. □

Let $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ be a cycle-unambiguous max-plus-WTA. In the following lemma, we present a nondeterministic PSPACE-algorithm which admits a successful run if and only if $\mathcal{A}$ satisfies the tree fork property. By Savitch's determinization theorem, deciding the tree fork property is thus in PSPACE. We do not make any statement about the hardness of the problem. We define the size $|\mathcal{A}|$ of a $\mathcal{A}$ as the size of its representation, i.e.,

$$|\mathcal{A}| = |Q| + \sum_{\substack{(q_1,\ldots,q_m,a,q_0)\in\Delta_{\mathcal{A}} \\ \mu(q_1,\ldots,q_m,a,q_0)\neq-\infty}} (m+2).$$

Lemma 10. *The problem of deciding whether a cycle-unambiguous max-plus-WTA satisfies the tree fork property is in PSPACE.*

Proof. Let $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ be a cycle-unambiguous max-plus-WTA and let $\Delta = \{d \in \Delta_{\mathcal{A}} \mid \mu(d) \neq -\infty\}$ be the set of all valid transitions of $\mathcal{A}$.

For the algorithm, we nondeterministically guess two states p, q and deterministically verify whether they are rivals and satisfy either condition (i) or condition (ii) of the tree fork property. If they are rivals and satisfy at least one of the conditions, the algorithm terminates successfully, otherwise it does not. Thus, the algorithm admits a successful run if and only if $\mathcal{A}$ satisfies the tree fork property.

Enumerating all pairs of siblings. We enumerate all siblings of $\mathcal{A}$ using the following reachability algorithm. We initialize the set of all pairs $S \subseteq Q \times Q$ which are siblings with $S = \emptyset$. Then we iterate the following operation.

Let $S' = S$. For every two transitions $(p_1, \ldots, p_m, a, p_0), (q_1, \ldots, q_m, a, q_0) \in \Delta$, add (p_0, q_0) to S' if $\{(p_1, q_1), \ldots, (p_m, q_m)\} \subseteq S$. If $S' = S$, store S as the set of all siblings and terminate. If $S \subsetneq S'$, let $S = S'$ and continue the iteration. This iteration terminates after at most $|Q \times Q| = |Q|^2$ steps and therefore runs in polynomial time. In particular, this part of the algorithm is in PSPACE and the output S can be stored in polynomial space.

Test for siblings. If $(p, q) \in S$, then p and q are siblings and the algorithm continues, otherwise it is not successful.

Deciding condition (i). We initialize the set of all pairs of states reachable from (p,p) by $R = \{(p,p)\}$. Then we iterate the following operation.

Let $R' = R$. For every two transitions $(p_1,\ldots,p_m,a,p_0),(q_1,\ldots,q_m,a,q_0) \in \Delta$, add (p_0,q_0) to R' if $\{(p_1,q_1),\ldots,(p_m,q_m)\} \subseteq S$ and $R \cap \{(p_1,q_1),\ldots,(p_m,q_m)\} \neq \emptyset$. If $(p,q) \in R'$, there exists a p-q-fork and the algorithm continues to "Test for rivals". If $(p,q) \notin R'$ and $R \subsetneq R'$, the algorithm sets $R = R'$ and continues the iteration in search for a p-q-fork. If $(p,q) \notin R'$ and $R = R'$, the search for a p-q-fork failed and the algorithm continues to "Deciding condition (ii)". This iteration also terminates after at most $|Q|^2$ steps and is thus in PSPACE.

Deciding condition (ii). We initialize set of states reachable from p by $R_p = \{p\}$. Then we iterate the following operation.

Let $R'_p = R_p$. For every transition $(p_1,\ldots,p_m,a,p_0) \in \Delta$, add p_0 to R'_p if $R_p \cap \{p_1,\ldots,p_m\} \neq \emptyset$. If $R'_p = R_p$, store R_p as the set of all states reachable from p. If $R_p \subsetneq R'_p$, let $R_p = R'_p$ and continue the iteration.

In the same fashion, we compute the set R_q of all states reachable from q. This part of the algorithm runs in polynomial time and the sets R_p and R_q can thus be stored in polynomial space.

Next, we verify for every transition $(p_1,\ldots,p_m,a,p_0) \in \Delta$ whether there exist indices i,j with $1 \leq i < j \leq m$ such that $p_i \in R_p$ and $p_j \in R_q$. If such a transition is found, condition (ii) is satisfied and the algorithm continues to "Test for rivals". Otherwise, p and q satisfy neither condition (i) nor condition (ii) of the tree fork property and the algorithm is not successful.

Test for rivals. Finally, we verify that there exists a Γ-word s such that $p \xrightarrow{s|x} p$ and $q \xrightarrow{s|y} q$ with $x \neq y$. By Lemma 6, it suffices to consider Γ-words of height at most $4|Q|^2$. As even with this size restriction, we can not necessarily store such a Γ-word s in polynomial space, we guess s dynamically and verify that it satisfies $p \xrightarrow{s|x} p$ and $q \xrightarrow{s|y} q$ with $x \neq y$.

More precisely, we guess the positions of s and their labels in lexicographic order. Whenever we have guessed all subtrees below a node $w \in \mathrm{pos}(s)$, we compute two tuples of weights for this node, one each for p and q. The tuple for p is defined as follows. If $s{\restriction_w}$ contains a leaf $\diamond$, the tuple contains for each state $p_0 \in Q$ an entry with the weight of the unique run from $\mathrm{Run}^{\diamond}_{\mathcal{A}}(p, s{\restriction_w}, p_0)$. If s does not contain a leaf $\diamond$, the tuple contains for each state $p_0 \in Q$ an entry with the weight of the unique run from $\mathrm{Run}^{\diamond}_{\mathcal{A}}(s{\restriction_w}, p_0)$. The tuple for q is defined similarly. After this computation, the subtrees below w and all data stored about them is discarded.

This procedure allows us to compute the weights of the unique runs from $\mathrm{Run}^{\diamond}_{\mathcal{A}}(p,s,p)$ and $\mathrm{Run}^{\diamond}_{\mathcal{A}}(q,s,q)$ without fully storing the runs in memory. Since s is bounded in height by $4|Q|^2$ and the rank of our symbols is bounded by $\mathrm{rk}(\Gamma)$, at every point in time we have to store information for at most $\mathrm{rk}(\Gamma) \cdot 4|Q|^2 + 1$ positions of s, where the "+1" stems from the root. For each of these positions, we store the position itself, which is of length at most $4|Q|^2$, the label of this position, and two tuples of weights, each of length $|Q|$. Thus, guessing s and computing the weights x and y above can be realized in polynomial space.

In the following, we present a more detailed version of the algorithm we just described. We fix an enumeration $Q = \{q_1,\ldots,q_n\}$ of Q. First, we initialize a single bit b with 0 in which we store whether we have already guessed a context symbol $\diamond$ for s. Then we set $w = \varepsilon$ as the next position to process and execute the following algorithm.

Part 1. Guess label for w

If $|w| < 4|Q|^2$ and $b = 0$, guess a letter $a \in \Gamma \cup \{\diamond\}$.
If $|w| < 4|Q|^2$ and $b = 1$, guess a letter $a \in \Gamma$.
If $|w| = 4|Q|^2$ and $b = 0$, guess a letter $a \in \Gamma^{(0)} \cup \{\diamond\}$.
If $|w| = 4|Q|^2$ and $b = 1$, guess a letter $a \in \Gamma^{(0)}$.
Store the pair (w,a).
If $a = \diamond$, set b to 1.
If $a \in \Gamma^{(0)} \cup \{\diamond\}$, continue to "Part 2".
If $\mathrm{rk}_\Gamma(a) > 0$, set $w = w1$ as the next position to process and continue to "Part 1".

Part 2. Combine weights for w

Let a be the label we guessed for w, i.e., the letter a for which we have stored the pair (w,a) in our memory. Let $m = \mathrm{rk}_{\Gamma_\diamond}(a)$ be the rank of a. By assumption, we have already processed all subtrees below w and thus, for each $i \in \{1,\ldots,m\}$ we have tuples $\bar{x}^{(i)}, \bar{y}^{(i)} \in \mathbb{R}^n_{\max}$ for wi. If $a = \diamond$, then for each $j \in \{1,\ldots,n\}$, let $x_j = 0$ if $q_j = p$ and $x_j = -\infty$ otherwise, and let $y_j = 0$ if $q_j = q$ and $y_j = -\infty$

otherwise. If $a \neq \diamond$, compute for every $j \in \{1, \ldots, n\}$ the weights

$$x_i = \max_{1 \leq j_1, \ldots, j_m \leq |Q|} (\mu(q_{j_1}, \ldots, q_{j_m}, a, q_i) + x^{(1)}_{j_1} + \ldots + x^{(m)}_{j_m})$$
$$y_i = \max_{1 \leq j_1, \ldots, j_m \leq |Q|} (\mu(q_{j_1}, \ldots, q_{j_m}, a, q_i) + y^{(1)}_{j_1} + \ldots + y^{(m)}_{j_m}).$$

Store the tuples $\bar{x} = (x_1, \ldots, x_n)$ and $\bar{y} = (y_1, \ldots, y_n)$ for w and discard the tuples $\bar{x}^{(i)}, \bar{y}^{(i)}$ for all positions $w1, \ldots, wm$ and discard all tuples of the form (wi, a'). Then choose the next position to process as follows. If $w = \varepsilon$, continue to "Part 3". Otherwise, we can write $w = vi$ with $i \in \mathbb{N}$. Let a' be the label we guessed for v. If $i = \mathrm{rk}_\Gamma(a')$, redefine $w = v$ as the next position to process and continue to "Part 2". If $i < \mathrm{rk}_\Gamma(a')$, redefine $w = v(i+1)$ as the next position to process and continue to "Part 1".
Part 3. By assumption, we have computed tuples of weights $\bar{x}, \bar{y} \in \mathbb{R}^n_{\max}$ for ε. Let i, j be the indices such that $p = q_i$ and $q = q_j$. If $b = 1$ and $x_i \neq y_j$, the algorithm terminates successfully. Otherwise, the algorithm is not successful. ☐

The following two sections are dedicated to completing the proof of Theorem 9.

3.1 Necessity

In this section, we show that if an unambiguous max-plus-WTA $\mathcal{A}$ satisfies either condition (i) or condition (ii) of the tree fork property, then $[\![\mathcal{A}]\!]$ is not finitely sequential. For condition (i), we adapt the corresponding proof from the word case [4, Theorem 2]. The proof relies on the Lipschitz property of deterministic max-plus automata and its approach is similar to the proof of Lemma 4 that the twins property is a necessary condition for determinizability.

Theorem 11. *Let $\mathcal{A}$ be a trim unambiguous max-plus-WTA over Γ. If $\mathcal{A}$ satisfies condition (i) of the tree fork property, then there do not exist deterministic max-plus-WTA $\mathcal{A}_1, \ldots, \mathcal{A}_n$ over Γ with $[\![\mathcal{A}]\!] = \max_{i=1}^n [\![\mathcal{A}_i]\!]$.*

Proof. For contradiction, assume that $\mathcal{A}$ satisfies condition (i) of the tree fork property and that there exist deterministic max-plus-WTA $\mathcal{A}_1, \ldots, \mathcal{A}_n$ over Γ with $[\![\mathcal{A}]\!] = \max_{i=1}^n [\![\mathcal{A}_i]\!]$. We write $\mathcal{A}_i = (Q_i, \Gamma, \mu_i, \nu_i)$ and let $N = \max_{i=1}^n |Q_i|$. Let p, q, t, z_p, z_q be as in condition (i) of the tree fork property and for the rivals p and q, let u, s, x, y be as in the definition of rivals. We let $r^p \in \mathrm{Run}_\mathcal{A}(u, p)$ and define $z_u = \mathrm{wt}_\mathcal{A}(u, r^p)$. Furthermore, by trimness there exists a Γ-word $\hat{u}$ with $q \xrightarrow{\hat{u}|z_{\hat{u}}} q_f$ for some weight $z_{\hat{u}} \in \mathbb{R}$ and some state $q_f \in Q$ with $\nu(q_f) \neq -\infty$.

We define the constant $L \in \mathbb{R}$ to be the largest weight, in terms of absolute value, which occurs in the automata $\mathcal{A}_1, \ldots, \mathcal{A}_n$ as follows. We let $X = \bigcup_{i=1}^n \mu_i(\Delta_{\mathcal{A}_i}) \cup \nu_i(Q_i)$ and define $L = \max_{x \in X \setminus \{-\infty\}} |x|$. Furthermore, we define natural numbers $N_0, \ldots, N_n$ inductively as follows. We let $N_n = 0$ and if $N_{l+1}, \ldots, N_n$ are defined, then we define N_l such that for all $k \in \{l+1, \ldots, n\}$ we have

$$N_l|x - y| > L\Big((k - l)|t| + (\sum_{i=l+1}^{k} N_i|s|) + 2|\hat{u}| + 2\Big) + (k - l)|z_p| + (\sum_{i=l+1}^{k-1} N_i|x|) + N_k|y|.$$

We define trees $t'_0, \ldots, t'_n$ inductively by $t'_0 = s^{N_0}(t(u))$ and $t'_{k+1} = s^{N_{k+1}}(t(t'_k))$; for clarity, in the word case we would have $t'_k = uts^{N_0}ts^{N_1} \cdots ts^{N_k}$. Then for $k \in \{1, \ldots, n\}$, we let $t_k = \hat{u}(t'_k)$. Due to the unambiguity of $\mathcal{A}$, we see that for every $k \in \{1, \ldots, n\}$ we have

$$[\![\mathcal{A}]\!](t_k) = z_u + kz_p + (\sum_{i=0}^{k-1} N_i x) + z_q + N_k y + z_{\hat{u}} + \nu(q_f).$$

Thus, for $k > l$, we have

$$\begin{aligned} |[\![\mathcal{A}]\!](t_k) - [\![\mathcal{A}]\!](t_l)| &= |N_l(x - y) + (k - l)z_p + (\sum_{i=l+1}^{k-1} N_i x) + N_k y| \\ &\geq N_l|x - y| - (k - l)|z_p| - (\sum_{i=l+1}^{k-1} N_i|x|) - N_k|y| \\ &> L\Big((k - l)|t| + (\sum_{i=l+1}^{k} N_i|s|) + 2|\hat{u}| + 2\Big). \end{aligned}$$

Note that the first inequality is an application of the reverse triangle inequality. The second inequality follows from the definition of N_l. Now let $j \in \{1, \dots, n\}$, then by choice of L and because $\mathcal{A}_j$ is deterministic, we have by Lemma 1 that

$$|[\![\mathcal{A}_j]\!](t_k) - [\![\mathcal{A}_j]\!](t_l)| \leq L\Big((k-l)|t| + (\sum_{i=l+1}^{k} N_i|s|) + 2|\hat{u}| + 2\Big).$$

In conclusion, we have $n+1$ trees t_i, and n automata $\mathcal{A}_i$, so by pigeonhole principle and the assumption that $[\![\mathcal{A}]\!] = \max_{i=1}^{n}[\![\mathcal{A}_i]\!]$, there must be $j \in \{1, \dots, n\}$ and $k, l \in \{0, \dots, n\}$ with $k > l$ such that $[\![\mathcal{A}]\!](t_k) = [\![\mathcal{A}_j]\!](t_k)$ and $[\![\mathcal{A}]\!](t_l) = [\![\mathcal{A}_j]\!](t_l)$. However, we have $|[\![\mathcal{A}]\!](t_k) - [\![\mathcal{A}]\!](t_l)| > |[\![\mathcal{A}_j]\!](t_k) - [\![\mathcal{A}_j]\!](t_l)|$, which is a contradiction. □

Next, we address condition (ii) of the tree fork property. On words, states cannot occur in prefix-independent positions. Thus, this condition is new for the tree case. Intuitively, the reason that the behavior of an unambiguous max-plus-WTA $\mathcal{A}$ cannot be finitely sequential if it satisfies condition (ii) is as follows. Assume we have a 2-Γ-context t and two rivals p and q as in condition (ii) and let u and s be as in the definition of rivals. Then we can construct trees of the form $t(s^n(u), s^n(u))$ such that, by increasing n, the difference between the weights on the two subtrees $s^n(u)$ is arbitrarily large. However, every deterministic automaton necessarily assigns the same weight to both subtrees.

Theorem 12. *Let $\mathcal{A}$ be a trim unambiguous max-plus-WTA over Γ. If $\mathcal{A}$ satisfies condition (ii) of the tree fork property, then there do not exist deterministic max-plus-WTA $\mathcal{A}_1, \dots, \mathcal{A}_n$ over Γ with $[\![\mathcal{A}]\!] = \max_{i=1}^{n}[\![\mathcal{A}_i]\!]$.*

Proof. For contradiction, we assume that $\mathcal{A}$ satisfies condition (ii) of the tree fork property and that there exist deterministic max-plus-WTA $\mathcal{A}_1, \dots, \mathcal{A}_n$ over Γ with $[\![\mathcal{A}]\!] = \max_{i=1}^{n}[\![\mathcal{A}_i]\!]$. First, we construct a tree of the above mentioned form $t(s^n(u), s^n(u))$ and choose n large enough to ensure that in each of the deterministic automata, some sub-Γ-word s^m of s^n loops in some state. Then we show that every choice of a weight for such a loop leads to a contradiction.

Let p, q, t, r be as in condition (ii) of the tree fork property, $v_1 = \Diamond_1(t)$, and $v_2 = \Diamond_2(t)$. For the rivals p and q, let u and s be as in the definition of rivals and $v = \Diamond_1(s)$. We let $r_u^p \in \mathrm{Run}_{\mathcal{A}}(u, p)$, $r_u^q \in \mathrm{Run}_{\mathcal{A}}(u, q)$, $r_s^p \in \mathrm{Run}_{\mathcal{A}}^{\Diamond}(p, s, p)$, and $r_s^q \in \mathrm{Run}_{\mathcal{A}}^{\Diamond}(q, s, q)$. Furthermore, we write $\mathcal{A}_i = (Q_i, \Gamma, \mu_i, \nu_i)$ and let $N = \max_{i=1}^{n}|Q_i|$.

By the following argument, we may assume that $\nu(r(\varepsilon)) \neq -\infty$. By trimness, there exists a Γ-word s'' and a run $r'' \in \mathrm{Run}_{\mathcal{A}}^{\Diamond}(s'')$ with $r''(\Diamond_1(s'')) = r(\varepsilon)$ and $\nu(r''(\varepsilon)) \neq -\infty$. Thus, if $\nu(r(\varepsilon)) = -\infty$, we can consider the 2-$\Gamma$-context $s''(t)$ with the run $r''\langle r \to \Diamond_1(s'')\rangle$ instead of t and r.

We now consider the tree $t' = t(s^N(u), s^N(u))$ together with the run

$$r' = r\langle (r_s^p)^{N\langle v\rangle}\langle r_u^p \to v^N\rangle \to v_1\rangle\langle (r_s^q)^{N\langle v\rangle}\langle r_u^q \to v^N\rangle \to v_2\rangle.$$

Since $r' \in \mathrm{Run}_{\mathcal{A}}(t')$ and $\nu(r'(\varepsilon)) \neq -\infty$, we have $[\![\mathcal{A}]\!](t') \neq -\infty$, so for some $j \in \{1, \dots, n\}$ we have $[\![\mathcal{A}_j]\!](t') = [\![\mathcal{A}]\!](t')$. By pigeonhole principle, since $N \geq |Q_j|$, we have $r'(v_1 v^{n_1}) = r'(v_1 v^{n_2})$ for some $n_1, n_2 \in \{0, \dots, N\}$ with $n_1 < n_2$. Since $\mathcal{A}_j$ is deterministic, we also obtain $r'(v_2 v^{n_1}) = r'(v_2 v^{n_2}) = r'(v_1 v^{n_1})$. Let $m = n_2 - n_1$ and let $x, y, z \in \mathbb{R}$ be the weights such that $p \xrightarrow{s|x} p$ and $q \xrightarrow{s|y} q$ in $\mathcal{A}$ and $r'(v_1 v^{n_1}) \xrightarrow{s^m|z} r'(v_1 v^{n_1})$ in $\mathcal{A}_j$. In particular, $x \neq y$. We may assume that $x < y$. We consider two cases.

First, if $z \geq \frac{m}{2}(x+y)$, then for the tree $t^+ = t(s^{N+m}(u), s^N(u))$ we obtain

$$\max_{i=1}^{n}[\![\mathcal{A}_i]\!](t^+) \geq [\![\mathcal{A}_j]\!](t^+) = [\![\mathcal{A}_j]\!](t') + z \geq [\![\mathcal{A}_j]\!](t') + \frac{m}{2}(x+y) > [\![\mathcal{A}_j]\!](t') + mx = [\![\mathcal{A}]\!](t^+).$$

Note that this follows because $\mathcal{A}$ and $\mathcal{A}_j$ are both unambiguous, i.e., if we construct an accepting run on a given tree, we know that the weight of this run must be the weight assigned to the tree by the automaton.

For the other case, namely that $z \leq \frac{m}{2}(x+y)$, we see that for the tree $t^- = t(s^N(u), s^{N-m}(u))$ we obtain

$$\max_{i=1}^{n}[\![\mathcal{A}_i]\!](t^-) \geq [\![\mathcal{A}_j]\!](t^-) = [\![\mathcal{A}_j]\!](t') - z \geq [\![\mathcal{A}_j]\!](t') - \frac{m}{2}(x+y) > [\![\mathcal{A}_j]\!](t') - my = [\![\mathcal{A}]\!](t^-).$$

In both cases, we see that $[\![\mathcal{A}]\!] = \max_{i=1}^{n}[\![\mathcal{A}_i]\!]$ does not hold, which is a contradiction. □

Together, Theorems 11 and 12 show that if a trim unambiguous max-plus-WTA satisfies the tree fork property, then its behavior is not finitely sequential.

3.2 Sufficiency

In this section, we show that the behavior of an unambiguous max-plus-WTA $\mathcal{A}$ which does not satisfy the tree fork property is finitely sequential. For simplicity, we begin with a description of our method of proof on max-plus word automata and compare it to the proof method of Bala and Koniński [4].

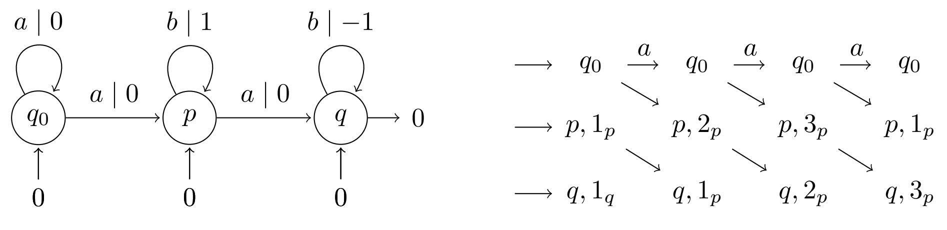
Both proofs work by distributing the runs of $\mathcal{A}$ across a finite set of unambiguous max-plus word automata such that all of these automata satisfy the twins property. This distribution essentially has the aim of separating the rivals of $\mathcal{A}$. By Theorem 3, these unambiguous automata can then be determinized. The major difference between our approach and that of [4] lies in the way we obtain these unambiguous automata. To understand our approach, let p and q be two rivals of $\mathcal{A}$. Furthermore, let $u = u_1 \cdots u_n$ be a word for which there exist valid runs $r^p = p_0 \xrightarrow{u_1} p_1 \xrightarrow{u_2} \cdots \xrightarrow{u_{n-1}} p_{n-1} \xrightarrow{u_n} p$ and $r^q = q_0 \xrightarrow{u_1} q_1 \xrightarrow{u_2} \cdots \xrightarrow{u_{n-1}} q_{n-1} \xrightarrow{u_n} q$ of $\mathcal{A}$ on u. We also define $p_n = p$ and $q_n = q$.

We now show that the first occurrence of either p or q in the runs r^p and r^q serves as a "distinguisher" between the two runs. We let i be the smallest index with the property that $p_i \in \{p, q\}$. Similarly, we let j be the smallest index with the property that $q_j \in \{p, q\}$. We obtain valid runs $p_i \xrightarrow{u_{i+1} \cdots u_n} p$ and $q_j \xrightarrow{u_{j+1} \cdots u_n} q$.

Now assume it would hold that $i = j$ and $p_i = q_j$, i.e., the first occurrences are at the same position in the word, and also the states at this position are the same in both runs. Then with $t = u_{i+1} \cdots u_n$, we see that we have valid runs $p_i \xrightarrow{t} p$ and $p_i \xrightarrow{t} q$, where $p_i \in \{p, q\}$. Thus, $\mathcal{A}$ would satisfy the fork property. Since our assumption is that $\mathcal{A}$ does not satisfy the fork property, we have either $i \neq j$ or $p_i \neq q_j$.

This fundamental property is also used in the corresponding proof of [4], but our way of exploiting it differs from [4]. In their proof for word automata, Bala and Koniński use this property implicitly to show that certain states of a modified Schützenberger covering of $\mathcal{A}$ occur at most once in every run [4, Lemma 6]. They can therefore construct a new max-plus automaton which for each run keeps a record of all occurrences of these states. The above mentioned unambiguous automata are then obtained by separating runs with differing records into different automata. For tree automata, the number of these occurrences is unfortunately not bounded, for reasons which we will also indicate below.

For now, we continue outlining our new approach, which is to construct an automaton which adds a distinguishing marker to every run when first encountering one of the rivals p or q. This marker consists of a number, which is used to distinguish occurrences at different positions, and the state from $\{p, q\}$ which was visited first. Whenever reading a letter which causes some valid run to visit p or q for the first time, the automaton selects the smallest marker which was not used by any valid run on the prefix read so far, and annotates it to the run. For example, assume that neither p nor q occur in any valid run the word u, but that our run r on ua leads to p. Then r obtains the marker 1_p. Now assume there is a valid run on uaa which leads to p and which visited neither p nor q before that. Then this run obtains the marker 2_p, since 1_p is already assigned to r. Next, assume that after reading $uaaa$ another marker for p has to be assigned, and that r cannot be extended to a valid run on uaa. Then we assign the marker 1_p, as now no valid run on uaa exists to which the marker 1_p is assigned. See Figure 3 for an example of this annotation process on the word aaa for the automaton depicted there.



a | 0
b | 1
b | −1
a | 0
a | 0
q₀
p
q
0
0
0
0
0
q₀ a q₀ a q₀ a q₀
p, 1_p   p, 2_p   p, 3_p   p, 1_p
q, 1_q   q, 1_p   q, 2_p   q, 3_p

Figure 3: On the left, an unambiguous max-plus word automaton over the alphabet $\{a, b\}$ which does not satisfy the twins property but whose behavior is finitely sequential. On the right, an illustration of the runs of the automaton on the words ε, a, aa, and aaa together with appropriate markers. Arrows indicate a transition. The states p and q are rivals with witnesses $u = \varepsilon$ and $s = b$.

With this procedure, runs like r^p and r^q above receive different markers since either one run obtains a marker later than the other, and therefore a different marker, or at least the states they visit first are different, which also leads to different markers. To separate the rivals of $\mathcal{A}$, we can thus make a copy of $\mathcal{A}$ for every marker, and only allow runs which carry the respective automaton's marker. Whenever a different marker would be assigned, the execution of the run is blocked.

Note here that the number of markers we need for this annotation process is bounded. Since the automaton $\mathcal{A}$ is unambiguous, the number of *valid* runs on every given word is bounded by the number of states in $\mathcal{A}$. If this were not the case, there would exist two distinct valid runs on the same word which lead to the same state, from which a counterexample to the unambiguity of $\mathcal{A}$ could be constructed. In particular, the number of markers assigned at any given "time" is bounded by the number of states of $\mathcal{A}$.

All of this can easily be generalized to the situation where there is more than one pair of rivals. Then, runs simply obtain a marker for each pair of rivals of the automaton, and the copies of $\mathcal{A}$ allow a distinguished marker for each of these pairs.

Unfortunately, these ideas do not translate to trees as easily. For example, consider the runs in Figure 4. Intuitively, both runs should obtain the marker 1_p. However, since p and q are rivals, this marker does not serve the purpose of distinguishing runs as it does in the word case. The first p occurs in different subtrees of both runs, thus the annotation of distinct markers is not possible. Also, it is easy to construct an automaton where a rival p can occur at arbitrarily many pairwise prefix-independent positions, thus a simple lexicographic distinction is not possible. This is also the reason why the approach from [4] does not work for tree automata.

$$\begin{aligned}
\nu(q) &= 0 \\
\mu(p, a, q) &= 0 \\[1em]
\mu(p, b, p) &= 1 \\
\mu(q, b, q) &= -1 \\[1em]
\mu(p, q_0, c, p) = \mu(q_0, p, c, q) &= 0 \\
\mu(d, p) = \mu(d, q_0) &= 0
\end{aligned}$$

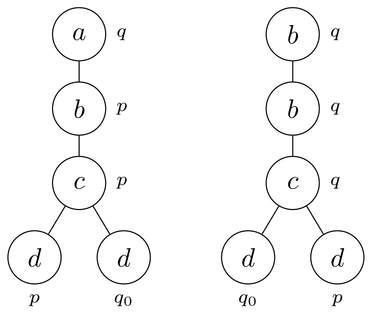


Figure 4: Two accepting runs of the max-plus tree automaton $\mathcal{A} = (\{q_0, p, q\}, \Gamma, \mu, \nu)$ over the ranked alphabet $\Gamma = \{a, b, c, d\}$ where $c \in \Gamma^{(2)}$, $a, b \in \Gamma^{(1)}$, and $d \in \Gamma^{(0)}$. All unspecified weights are assumed to be $-\infty$. The states p and q are rivals.

Our solution is to distribute not the runs of the automaton $\mathcal{A}$, but the runs of its *Schützenberger covering*. The Schützenberger covering of a max-plus automaton $\mathcal{A}$ is a max-plus automaton which possesses the same behavior as $\mathcal{A}$. It has already been employed in a number of decidability results for max-plus automata [22, 4, 3, 31]. Its construction is inspired by a paper of Schützenberger [39] and was made explicit by Sakarovitch in [35], see also [36].

To better explain the idea behind its construction, we first point out a certain aspect of the classical powerset construction for finite automata [34]. Assume that $\mathcal{D}$ is the result of applying the powerset construction to an NFA $\mathcal{B}$. Then we might say that for a word $w = w_1 w_2$, the state which $\mathcal{D}$ is in after reading the prefix w_1 is the set of all states which $\mathcal{B}$ *could* be in after reading w_1. Similarly, the Schützenberger covering of a max-plus automaton $\mathcal{A}$ annotates to every state of a run of $\mathcal{A}$ on a word w the set of all states which "$\mathcal{A}$ could be in" at this point, i.e., which can be reached by some valid run on the considered prefix of w. Like the powerset construction, these ideas easily carry over to trees.

The reason we consider the Schützenberger covering of $\mathcal{A}$ is that each pair $\mathbf{p}, \mathbf{q}$ of its rivals satisfies the following property. For every tree t, either (1) $\mathbf{p}$ and $\mathbf{q}$ do not occur together in any run on t or (2) $\mathbf{p}$ and $\mathbf{q}$ occur only linearly, i.e., there is a distinguished branch of t such that for every run on t, all occurrences of $\mathbf{p}$ and $\mathbf{q}$ lie on this branch. In particular, the situation of Figure 4 is not possible. All pairs which satisfy the first condition can simply be separated into different automata, all pairs which satisfy the second condition can be handled like in the word case. The proof of this is non-trivial and needs some preparation. We begin with the formal definition of the Schützenberger covering.

For the rest of this section, let $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ be a trim unambiguous max-plus-WTA which does not satisfy the tree fork property.

Definition 13 (Schützenberger covering, [35]). The Schützenberger covering $\mathcal{S} = (Q_{\mathcal{S}}, \Gamma, \mu_{\mathcal{S}}, \nu_{\mathcal{S}})$ of $\mathcal{A}$ is the trim part of the max-plus-WTA $(Q \times \mathcal{P}(Q), \Gamma, \mu', \nu')$ defined for $a \in \Gamma$ with $\mathrm{rk}_\Gamma(a) = m$ and

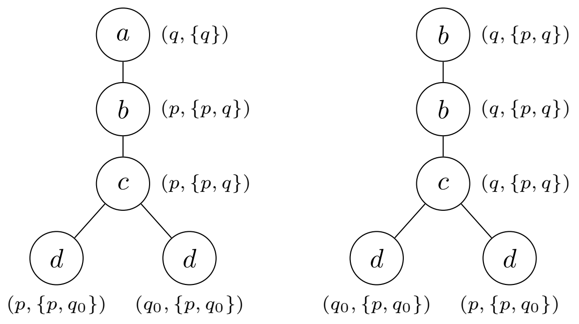


Figure 5: Two accepting runs of the Schützenberger covering of the automaton from Figure 4. The states $(p,\{p,q\})$ and $(q,\{p,q\})$ are rivals. The state $(p,\{p,q_0\})$ is not the rival of any state.

$(p_0,P_0),\dots,(p_m,P_m)\in Q\times\mathcal{P}(Q)$ by

$$\mu'((p_1,P_1),\dots,(p_m,P_m),a,(p_0,P_0)) =$$
$$\begin{cases}\mu(p_1,\dots,p_m,a,p_0) & \text{if } P_0=\{q_0\in Q\mid \exists(q_1,\dots,q_m)\in P_1\times\ldots\times P_m \text{ with } \mu(q_1,\dots,q_m,a,q_0)\neq-\infty\}\\ -\infty & \text{otherwise}\end{cases}$$
$$\nu'(p_0,P_0)=\nu(p_0).$$

We let $\pi_1\colon Q\times\mathcal{P}(Q)\to Q$, $(p,P)\mapsto p$ and $\pi_2\colon Q\times\mathcal{P}(Q)\to\mathcal{P}(Q)$, $(p,P)\mapsto P$ be the projections.

It is elementary to show that for a run of $\mathcal{S}$ on a tree t, the second entry of the state at a position w consists of all states of $\mathcal{A}$ which can be reached by a valid run of $\mathcal{A}$ on $t\restriction_w$. In particular, every two runs on the same tree coincide on their second entries. Furthermore, projecting all states of a run of $\mathcal{S}$ to their first coordinate yields a run of $\mathcal{A}$, and the weights of these runs coincide. It follows that $\mathcal{S}$ is unambiguous and satisfies $[\![\mathcal{S}]\!]=[\![\mathcal{A}]\!]$. Also, $\mathcal{S}$ is trim by definition.

We can also make the following observation about the rivals of $\mathcal{S}$. Let $\mathbf{p}$ and $\mathbf{q}$ be rivals of $\mathcal{S}$ and let u and s be as in the definition of rivals. Since all runs of $\mathcal{S}$ on u coincide on the second entry of the state at the root, $\mathbf{p}$ and $\mathbf{q}$ also coincide on their second entry. Moreover, as projecting the runs of $\mathcal{S}$ on u and s to their first entries yields runs of $\mathcal{A}$ on u and s, respectively, we additionally see that the first entries of $\mathbf{p}$ and $\mathbf{q}$ are rivals in $\mathcal{A}$. Thus, if two states $\mathbf{p},\mathbf{q}\in Q_{\mathcal{S}}$ are rivals in $\mathcal{S}$, then $\mathbf{p}=(p,P)$ and $\mathbf{q}=(q,P)$ for some set $P\subseteq Q$ and two states $p,q\in Q$ which are rivals in $\mathcal{A}$.

In the Schützenberger covering of the automaton from Figure 4, only the states $(p,\{p,q\})$ and $(q,\{p,q\})$ are rivals. See also Figure 5 for the runs of the Schützenberger covering on the trees from Figure 4. In the following lemma, we formally show that the properties we just described indeed hold for $\mathcal{S}$.

Lemma 14. *Let $t\in T_\Gamma$ be a tree. Then the following statements hold.*

(i) For every run $r\in\mathrm{Run}_{\mathcal{S}}(t)$ and position $w\in\mathrm{pos}(t)$ we have $\pi_2\circ r(w)=\{p\in Q\mid \exists r'\in\mathrm{Run}_{\mathcal{A}}(t\restriction_w,p)\}$.

(ii) For every two runs $r_1,r_2\in\mathrm{Run}_{\mathcal{S}}(t)$, it holds that $\pi_2\circ r_1=\pi_2\circ r_2$.

(iii) The projection π_1 induces a bijection $\pi_1\colon\mathrm{Run}_{\mathcal{S}}(t)\to\mathrm{Run}_{\mathcal{A}}(t)$ by $r\mapsto\pi_1\circ r$.

(iv) For every run $r\in\mathrm{Run}_{\mathcal{S}}(t)$ and every position $w\in\mathrm{pos}(t)$, we have $\pi_1\circ r(w)\in\pi_2\circ r(w)$.

(v) $\mathcal{S}$ is trim, unambiguous, and satisfies $[\![\mathcal{S}]\!]=[\![\mathcal{A}]\!]$.

(vi) For every Γ-word s and two states $\mathbf{p},\mathbf{q}\in Q_{\mathcal{S}}$ with $\mathbf{p}\xrightarrow{s|x}\mathbf{q}$, we have $\pi_1(\mathbf{p})\xrightarrow{s|x}\pi_1(\mathbf{q})$.

(vii) If two states $\mathbf{p},\mathbf{q}\in Q_{\mathcal{S}}$ are rivals in $\mathcal{S}$, then $\mathbf{p}=(p,P)$ and $\mathbf{q}=(q,P)$ for some set $P\subseteq Q$ and two states $p,q\in Q$ which are rivals in $\mathcal{A}$.

Proof. (i) Let $t\in T_\Gamma$ and $r\in\mathrm{Run}_{\mathcal{B}}(t)$ and for contradiction, let $w\in\mathrm{pos}(t)$ be a prefix-maximal position for which (i) does not hold. We deduce that (i) holds for w. We let $a=t(w)$, $m=\mathrm{rk}_\Gamma(a)$, and write $r(w)=(p,P)$ and $r(wi)=(p_i,P_i)$ for $i\in\{1,\dots,m\}$.

First, let $q \in P$, then there are states $(q_1, \ldots, q_m) \in P_1 \times \ldots \times P_m$ with $\mu(q_1, \ldots, q_m, a, q) \neq -\infty$. By assumption, for every $i \in \{1, \ldots, m\}$ we find a run $r_i \in \mathrm{Run}_{\mathcal{A}}(t\lceil_{wi}, q_i)$. Then the quasi-run $r' \colon \mathrm{pos}(t\lceil_w) \to Q$ defined by $r'(\varepsilon) = q$ and $r'(iv) = r_i(v)$ is a run of $\mathcal{A}$ on $t\lceil_w$ with $r'(\varepsilon) = q$.

On the other hand, let $r' \in \mathrm{Run}_{\mathcal{A}}(t\lceil_w)$ and let $q = r'(\varepsilon)$. Then for every $i \in \{1, \ldots, m\}$ we have that $r'\lceil_i \in \mathrm{Run}_{\mathcal{A}}(t\lceil_{wi})$, so by assumption, $r'(i) \in P_i$. Moreover, $\mu(r'(1), \ldots, r'(m), a, q) \neq -\infty$, so $q \in P$. Thus, (i) holds for w, which is a contradiction, so w does not exist.

(ii) follows from (i).

(iii) Let $t \in T_\Gamma$. By definition of $\mu_{\mathcal{S}}$, it is clear that for $r \in \mathrm{Run}_{\mathcal{S}}(t)$ we have $\pi_1 \circ r \in \mathrm{Run}_{\mathcal{A}}(t)$. The injectivity of $\pi_1 \colon \mathrm{Run}_{\mathcal{S}}(t) \to \mathrm{Run}_{\mathcal{A}}(t)$ follows from (ii) since for every two runs $r_1, r_2 \in \mathrm{Run}_{\mathcal{S}}(t)$ we have $\pi_2 \circ r_1 = \pi_2 \circ r_2$. For surjectivity, we let $r' \in \mathrm{Run}_{\mathcal{A}}(t)$ and define a run $r \in \mathrm{Run}_{\mathcal{S}}(t)$ inductively as follows. For a leaf $w \in \mathrm{pos}(t)$, we let $r(w) = (r'(w), \{p_0 \in Q \mid \mu(t(w), p_0) \neq -\infty\})$. For $w \in \mathrm{pos}(t)$ with $\mathrm{rk}_\Gamma(t(w)) = m$ such that r is defined on $w1, \ldots, wm$ with $\pi_2 \circ r(wi) = P_i$, we let $r(w) = (r'(w), \{p_0 \in Q \mid \exists (p_1, \ldots, p_m) \in P_1 \times \ldots \times P_m \text{ with } \mu(p_1, \ldots, p_m, a, p_0) \neq -\infty\})$. Then $r \in \mathrm{Run}_{\mathcal{S}}(t)$ and $\pi_1 \circ r = r'$.

(iv) follows from (i) and (iii).

(v) $\mathcal{S}$ is trim by definition. Let $t \in T_\Gamma$. By definition of $\mu_{\mathcal{S}}$, for every run $r \in \mathrm{Run}_{\mathcal{S}}(t)$ we have $\mathrm{wt}_{\mathcal{S}}(t, r) = \mathrm{wt}_{\mathcal{A}}(t, \pi_1 \circ r)$. By definition of $\nu_{\mathcal{S}}$, we also have $\nu_{\mathcal{S}}(r(\varepsilon)) = \nu(\pi_1 \circ r(\varepsilon))$. By (iii), we thus have $|\mathrm{Acc}_{\mathcal{S}}(t)| = |\mathrm{Acc}_{\mathcal{A}}(t)| \leq 1$, which means $\mathcal{S}$ is unambiguous, and $[\![\mathcal{S}]\!](t) = [\![\mathcal{A}]\!](t)$.

(vi) Let s be a Γ-word and $\mathbf{p}, \mathbf{q} \in Q_{\mathcal{S}}$ be two states with $\mathbf{p} \xrightarrow{s|x} \mathbf{q}$, then there exists a run $r \in \mathrm{Run}^{\diamond}_{\mathcal{S}}(\mathbf{p}, s, \mathbf{q})$ with $\mathrm{wt}^{\diamond}_{\mathcal{S}}(s, r) = x$. By definition of $\mu_{\mathcal{S}}$, we have $\pi_1 \circ r \in \mathrm{Run}^{\diamond}_{\mathcal{A}}(s)$ and $\mathrm{wt}^{\diamond}_{\mathcal{S}}(s, r) = \mathrm{wt}^{\diamond}_{\mathcal{A}}(s, \pi_1 \circ r)$, so $\pi_1(\mathbf{p}) \xrightarrow{s|x} \pi_1(\mathbf{q})$.

(vii) Let $\mathbf{p}, \mathbf{q} \in Q_{\mathcal{S}}$ be rivals in $\mathcal{S}$ and write $\mathbf{p} = (p, P_p)$, $\mathbf{q} = (q, P_q)$. Let $u \in T_\Gamma$ and $s \in T_{\Gamma_\diamond}$ be as in the definition of rivals and let $r^{\mathbf{p}} \in \mathrm{Run}_{\mathcal{S}}(u, \mathbf{p})$ and $r^{\mathbf{q}} \in \mathrm{Run}_{\mathcal{S}}(u, \mathbf{q})$. By (ii), we have $P_p = \pi_2 \circ r^{\mathbf{p}}(\varepsilon) = \pi_2 \circ r^{\mathbf{q}}(\varepsilon) = P_q$. By (iii), we have $\pi_1 \circ r^{\mathbf{p}} \in \mathrm{Run}_{\mathcal{A}}(u, p)$ and $\pi_1 \circ r^{\mathbf{q}} \in \mathrm{Run}_{\mathcal{A}}(u, q)$, so p and q are siblings. Finally, from $(p, P_p) \xrightarrow{s|x} (p, P_p)$ and $(q, P_q) \xrightarrow{s|y} (q, P_q)$, we obtain by (vi) that $p \xrightarrow{s|x} p$ and $q \xrightarrow{s|y} q$. Since $x \neq y$, p and q are rivals in $\mathcal{A}$. □

In the theorems to follow, we will use fact (vii) of Lemma 14 without explicit further notice.

In order to prove some deeper results about the rivals of $\mathcal{S}$, we need two preparatory lemmata. As a first simplification, we show that we may assume that two rivals p and q of $\mathcal{A}$ are always comparable with respect to the relation $\leq$. To see this, note that by condition (ii) of the tree fork property, p and q may not occur in prefix-independent positions in a run. If in addition, p and q can also not appear in prefix-dependent positions in a run, they never appear together in the same run of $\mathcal{A}$. Thus, we can create two copies of $\mathcal{A}$, one in which we remove p and one in which we remove q, and the pointwise maximum of these two automata will be equivalent to the behavior of $\mathcal{A}$.

Lemma 15. *We may assume that for all rivals $p, q \in Q$ we have either $p \leq q$ or $q \leq p$, or both.*

Proof. Let $p, q \in Q$ be rivals for which neither $p \leq q$ nor $q \leq p$. Then we can show that p and q never occur together in the same run as follows. Assume we have a tree $t \in T_\Gamma$, a run $r \in \mathrm{Run}_{\mathcal{A}}(t)$, and positions $w_1, w_2 \in \mathrm{pos}(t)$ with $r(w_1) = p$ and $r(w_2) = q$. Then w_1 and w_2 may not be prefix-independent since p and q are rivals, and by assumption $\mathcal{A}$ does not satisfy condition (ii) of the tree fork property. However, if w_1 and w_2 are prefix-dependent, we have a witness for either $p \leq q$ or $q \leq p$. This is a contradiction, and thus r as chosen does not exist.

We let $Q_1 = Q \setminus \{p\}$, $Q_2 = Q \setminus \{q\}$, and let $\mathcal{A}_i = (Q_i, \Gamma, \mu_i, \nu_i)$ for $i = 1, 2$, where μ_i and ν_i are the appropriate restrictions of μ and ν to the state sets Q_i. As p and q do not occur together in any run of $\mathcal{A}$, every run of $\mathcal{A}$ is also a run of at least one of the automata $\mathcal{A}_1, \mathcal{A}_2$. Thus, we have $[\![\mathcal{A}]\!] = \max_{i=1}^{2} [\![\mathcal{A}_i]\!]$ and both $\mathcal{A}_1$ and $\mathcal{A}_2$ are trim and unambiguous and do not satisfy the tree fork property.

This procedure can be iterated to separate all rivals which are not in $\leq$-relation. The termination of this procedure is guaranteed by the fact that the set of states becomes strictly smaller with every iteration. Eventually, we find trim unambiguous max-plus-WTA $\mathcal{A}_1, \ldots, \mathcal{A}_n$, all of which do not satisfy the tree fork property, such that $[\![\mathcal{A}]\!] = \max_{i=1}^{n} [\![\mathcal{A}_i]\!]$ and all rivals in an automaton $\mathcal{A}_i$ are pairwise in $\leq$-relation. □

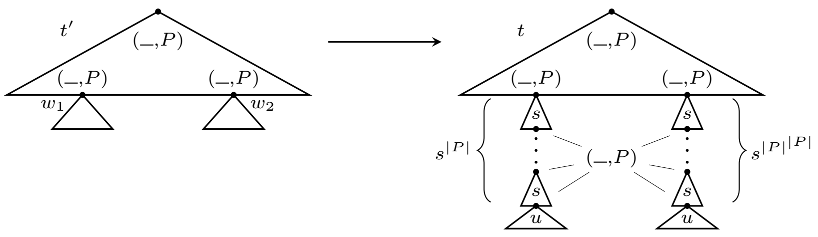


Figure 6: An illustration for the proof of Lemma 17.

Next, we note an elementary statement about self-maps $f\colon X \to X$. Namely, if X is a finite set and $f\colon X \to X$ a mapping, then for every $a \in X$ there exists some element $b \in X$ and an integer $n \geq 1$ such that after n iterations of f, both a and b are mapped to b. To see this, consider the elements $a, f(a), f^2(a), \ldots, f^{|X|}(a)$. By pigeon hole principle, there are numbers $0 \leq m_1 < m_2 \leq |X|$ with $f^{m_1}(a) = f^{m_2}(a)$. Then if we choose $n \geq m_1$ as a multiple of $m_2 - m_1$ and $b = f^n(a)$, we see that $f^n(a) = b = f^n(b)$.

Lemma 16. *Let X be a finite set and $f\colon X \to X$ a mapping. Then for every $a \in X$, there exists an element $b \in X$ and an integer $n \geq 1$ with $f^n(a) = b = f^n(b)$. Here, f^n is the n-th iterate of f, i.e., $f^0 = \mathrm{id}_X$ and $f^{m+1} = f \circ f^m$.*

We now identify the first important property which all rivals of $\mathcal{S}$ satisfy. Namely, if $P \subseteq Q$ is the second entry of *some* rival, then it cannot occur in the form of a "triangle" in any valid run of $\mathcal{S}$. More precisely, if we have a run r and positions w, wv_1, and wv_2 such that the second entry of $r(w)$, $r(wv_1)$, and $r(wv_2)$ is P, then wv_1 and wv_2 are prefix-dependent.

Lemma 17. *Let $(p,P),(q,P) \in Q_{\mathcal{S}}$ be rivals in $\mathcal{S}$. Furthermore, let $t' \in T_\Gamma$ be a tree, $r' \in \mathrm{Run}_{\mathcal{S}}(t')$ a run of $\mathcal{S}$ on t', and $w_1, w_2 \in \mathrm{pos}(t')$ be positions in t'. If $\pi_2 \circ r'(\varepsilon) = \pi_2 \circ r'(w_1) = \pi_2 \circ r'(w_2) = P$, then w_1 and w_2 are prefix-dependent.*

Proof. We proceed by contradiction and assume that t', r', w_1, w_2 as in the statement of the lemma exist such that w_1 and w_2 are prefix-independent. We show that then, $\mathcal{A}$ satisfies condition (i) of the tree fork property. For the rivals (p,P) and (q,P), let u and s be as in the definition of rivals and let $v = \lozenge_1(s)$. As the proof is rather technical, we first provide a proof sketch and then follow up with a more precise presentation of the argumentation. See also Figure 6 for some visual aid.

By assumption, u can reach (p,P) and s can loop in (p,P), thus the trees $s^{|P|}(u)$ and $s^{|P|^{|P|}}(u)$ can reach (p,P). Due to the construction of $\mathcal{S}$, this means both of these trees can also reach the states of r' at w_1 and w_2. In particular, there exists a run of $\mathcal{S}$ on the tree $t = t'\langle s^{|P|}(u) \to w_1\rangle\langle s^{|P|^{|P|}}(u) \to w_2\rangle$ and for this run, the second entry of every state at the beginning or end of an s-loop is P. In addition, t leads to a state with second entry P, so there in fact exist $|P|$ runs of $\mathcal{S}$ on t, one for each state in P. We let $r_1, \ldots, r_{|P|}$ be the projections of these runs to their first entry and obtain $|P|$ runs of $\mathcal{A}$ on t where for each run the state at the root and all states at the beginning or end of an s-loop are from P.

By pigeonhole principle, there is some subloop s^n below w_2 which loops in all runs at the same time, i.e., where for some n_1 we have $r_i(w_2v^{n_1}) = r_i(w_2v^{n_1+n})$ for all runs r_i. For each r_i, we let $q_i = r_i(w_2v^{n_1}) \in P$ be the state which r_i loops in and let x_i be the weight of this loop.

If $x_i \neq x_j$ for some i and j, the states q_i and q_j are rivals in $\mathcal{A}$ with witnesses u and s^n. By Lemma 15, we may therefore assume $q_i \leq q_j$. Again by pigeon hole principle, the run r_i loops below w_1 in s^m for some $m \geq 1$ with some state $p_i \in P$, say with weight y_i. Due to $x_i \neq x_j$, we have $mx_i \neq ny_i$ or $mx_j \neq ny_i$. Since u can reach every state from P, the state p_i is thus a rival of q_i or q_j with witnesses u and s^{nm}. From the existence of r_i and the assumption that $q_i \leq q_j$, we see that p_i can occur prefix-independently both from q_i and from q_j. This is a contradiction to the assumption that $\mathcal{A}$ does not satisfy the tree fork property. It must therefore hold that $x_1 = \ldots = x_{|P|}$.

We let x and y be the weights such that $\mathcal{A}$ loops s in p with weight x and in q with weight y. Then from $x \neq y$ it follows that $nx \neq x_1$ or $ny \neq x_1$, so the states q_i are either all rivals of p or all rivals of q with witnesses u and s^n. We assume all q_i to be rivals of p and apply Lemma 16 to the mapping $f\colon P \to \{q_1, \ldots, q_{|P|}\}, r_i(\varepsilon) \mapsto q_i$ with $a = p$ to obtain $q_j \in P$ and $m \geq 1$ such that $f^m(p) = q_j = f^m(q_j)$. Then with $\tilde{s} = t\langle \diamond \to w_2v^{n_1}\rangle$, we see that the Γ-word $\tilde{s}^m$ is a q_j-p-fork, i.e., $\mathcal{A}$ satisfies condition (i) of the tree fork property.

We now turn to the more technical presentation of the proof. We define the tree $t = t'\langle s^{|P|}(u) \to w_1\rangle\langle s^{|P|^{|P|}}(u) \to w_2\rangle$ and construct a run $r \in \mathrm{Run}_{\mathcal{S}}(t)$ of $\mathcal{S}$ on t as follows. By assumption, there exists a run $r^{\mathbf{p}} \in \mathrm{Run}_{\mathcal{S}}(u, (p, P))$ and a run $r_s \in \mathrm{Run}^{\diamond}_{\mathcal{S}}((p, P), s, (p, P))$. We let $r'_1 = r_s^{|P|\langle v\rangle}\langle r^{\mathbf{p}} \to v^{|P|}\rangle$ and $r'_2 = r_s^{|P|^{|P|}\langle v\rangle}\langle r^{\mathbf{p}} \to v^{|P|^{|P|}}\rangle$. Then $r'_1 \in \mathrm{Run}_{\mathcal{S}}(s^{|P|}(u), (p, P))$ and $r'_2 \in \mathrm{Run}_{\mathcal{S}}(s^{|P|^{|P|}}(u), (p, P))$.

By Lemma 14(iv), we have $\pi_1 \circ r'(w_1), \pi_1 \circ r'(w_2) \in P$, so by Lemma 14(i) we can find $r''_1 \in \mathrm{Run}_{\mathcal{A}}(s^{|P|}(u))$ with $r''_1(\varepsilon) = \pi_1 \circ r'(w_1)$ and $r''_2 \in \mathrm{Run}_{\mathcal{A}}(s^{|P|^{|P|}}(u))$ with $r''_2(\varepsilon) = \pi_1 \circ r'(w_2)$. Then $r = r'\langle \pi_1^{-1}(r''_1) \to w_1\rangle\langle \pi_1^{-1}(r''_2) \to w_2\rangle \in \mathrm{Run}_{\mathcal{S}}(t)$ is a run of $\mathcal{S}$ on t and we have $\pi_2 \circ r(w_1 v^i) = P$ for $0 \le i \le |P|$ and $\pi_2 \circ r(w_2 v^i) = P$ for $0 \le i \le |P|^{|P|}$.

By Lemma 14(i) and because $\pi_2 \circ r(\varepsilon) = P$, we can now find $|P|$ runs $r_1, \dots, r_{|P|} \in \mathrm{Run}_{\mathcal{A}}(t)$ on t such that $\{r_1(\varepsilon), \dots, r_{|P|}(\varepsilon)\} = P$. We have $r_j(w_2 v^i) \in P$ for every $j \in \{1, \dots, |P|\}$ and every $i \in \{0, \dots, |P|^{|P|}\}$. For each $i \in \{0, \dots, |P|^{|P|}\}$, we define the tuple $\bar{q}_i = (r_1(w_2 v^i), \dots, r_{|P|}(w_2 v^i))$. Since $\bar{q}_i \in P^{|P|}$ for every i, we can find $n_1 < n_2$ with $\bar{q}_{n_1} = \bar{q}_{n_2}$ by pigeonhole principle. Let $n = n_2 - n_1$ and write $\bar{q}_{n_1} = (q_1, \dots, q_{|P|})$.

We now show that $q_1, \dots, q_{|P|}$ are either all rivals of p, or they are all rivals of q. For this, note first that $q_j \xrightarrow{s^n|x_j} q_j$ for all $j \in \{1, \dots, |P|\}$ with weights $x_1, \dots, x_{|P|} \in \mathbb{R}$. Also, by the existence of the run $r^{\mathbf{p}}$ on u and Lemma 14(i), all states in P are siblings.

We show first that $x_1 = \ldots = x_{|P|}$. We assume that by contradiction, $x_i \neq x_j$ for some $i \neq j$. Then q_i and q_j are rivals in $\mathcal{A}$ with witnesses u and s^n. By Lemma 15, we can therefore assume that $q_i \le q_j$ or $q_j \le q_i$. We assume $q_i \le q_j$ and let s^i_j be a Γ-word such that there exists a run $r^i_j \in \mathrm{Run}^{\diamond}_{\mathcal{S}}(q_j, s^i_j, q_i)$. Furthermore, by pigeonhole principle, we can find $m_1, m_2 \in \{0, \dots, |P|\}$ with $r_i(w_1 v^{m_1}) = r_i(w_2 v^{m_2})$ and $m_1 < m_2$. We let $p_i = r_i(w_1 v^{m_1})$ and $m = m_2 - m_1$ and show that p_i is a rival of either q_i or q_j. We have $p_i \xrightarrow{s^m|y_i} p_i$ for some weight $y_i \in \mathbb{R}$. Since $p_i \in P$, we know that p_i, q_i, and q_j are all siblings. Also, we have $p_i \xrightarrow{s^{nm}|ny_i} p_i$, $q_i \xrightarrow{s^{nm}|mx_i} q_i$, and $q_j \xrightarrow{s^{nm}|mx_j} q_j$. Since $x_i \neq x_j$, we have $ny_i \neq mx_i$ or $ny_i \neq mx_j$, or both. Thus, p_i is a rival of either q_i or of q_j.

Under these assumptions, we see that $\mathcal{A}$ satisfies condition (ii) of the tree fork property as follows. Either the 2-Γ-context $t_1 = t\langle \diamond \to w_1 v^{m_1}\rangle\langle \diamond \to w_2 v^{n_1}\rangle$ together with the run $r_i\restriction_{\mathrm{pos}(t_1)}$ or the 2-Γ-context $t_2 = t_1(\diamond, s^i_j)$ together with the run $r_i\restriction_{\mathrm{pos}(t_1)}\langle r^i_j \to \Diamond_2(t_1)\rangle$ is a witness for condition (ii) to be satisfied. Since our assumption for this section is that $\mathcal{A}$ does not satisfy the tree fork property, this is a contradiction. In conclusion, $x_1 = \ldots = x_{|P|}$.

To see that $q_1, \dots, q_{|P|}$ are either all rivals of p, or they are all rivals of q, consider the following. Using the same arguments as above, we find for every $i \in \{1, \dots, |P|\}$ a run $r^{q_i} \in \mathrm{Run}_{\mathcal{A}}(u, q_i)$. Furthermore, we have $p \xrightarrow{s^n|nx} p$, $q \xrightarrow{s^n|ny} q$, and $q_i \xrightarrow{s^n|x_1} q_i$ for every $i \in \{1, \dots, |P|\}$. Since $x \neq y$, we have either $nx \neq x_1$ or $ny \neq x_1$. Without loss of generality, we assume $nx \neq x_1$, thus all q_i are rivals of p.

We now show that $\mathcal{A}$ satisfies condition (i) of the tree fork property. We define a mapping $f\colon P \to \{q_1, \dots, q_{|P|}\}$ by $r_i(\varepsilon) \mapsto q_i$ for $i \in \{1, \dots, |P|\}$; recall that $\{q_1, \dots, q_{|P|}\} \subseteq P$, $\{r_1(\varepsilon), \dots, r_{|P|}(\varepsilon)\} = P$, and $r_i(\varepsilon) \neq r_j(\varepsilon)$ for $i \neq j$. By Lemma 16, there exists $m \ge 1$ and $i \in \{1, \dots, |P|\}$ with $f^m(p) = q_i = f^m(q_i)$. From this, we obtain that with $\tilde{s} = t\langle \diamond \to w_2 v^{n_1}\rangle$ we have $q_i \xrightarrow{\tilde{s}^m|z} q_i$ and $q_i \xrightarrow{\tilde{s}^m|z'} p$ for weights $z, z' \in \mathbb{R}$. As p and q_i are rivals, this means that $\mathcal{A}$ satisfies condition (i) of the tree fork property. ☐

In the previous lemma, we showed that if P is the second entry of some rival from $\mathcal{S}$, then states with second entry P do not occur in the form of a triangle. In the next lemma, we show that even prefix-independent occurrences are restricted to a certain degree. Namely, if we have two rivals (p, P) and (q, P) with $p \le q$, then all occurrences of P as a second entry are prefix-dependent on (p, P).

Lemma 18. *Let $(p, P), (q, P) \in Q_{\mathcal{S}}$ be rivals in $\mathcal{S}$ with $p \le q$. Furthermore, let $t' \in T_\Gamma$ be a tree, $r' \in \mathrm{Run}_{\mathcal{S}}(t')$ a run of $\mathcal{S}$ on t', and $w_1 \in \mathrm{pos}(t')$ a position in t' with $r'(w_1) = (p, P)$. Then all positions $w_2 \in \mathrm{pos}(t')$ with $\pi_2 \circ r'(w_2) = P$ are prefix-dependent on w_1.*

Proof. We proceed by contradiction and take $(p, P), (q, P), t', r', w_1$ as in the statement of the lemma and assume that there exists a position $w_2 \in \mathrm{pos}(t')$ which is prefix-independent from w_1 and for which $\pi_2 \circ r'(w_2) = P$. We show that under these assumptions, $\mathcal{A}$ satisfies condition (ii) of the tree fork property. For the rivals (p, P) and (q, P), let u and s be as in the definition of rivals and let $v = \Diamond_1(s)$. As in the proof of the previous lemma, we first provide a short proof sketch, see also Figure 7 for some visual aid.

As we have seen in the proof of Lemma 17, the tree $s^{|P|}(u)$ can reach (p, P), so due to the construction of $\mathcal{S}$, it can also reach the state of r' at w_2. Thus, there exists a run of $\mathcal{S}$ on the tree $t = t'\langle s^{|P|}(u) \to w_2\rangle$

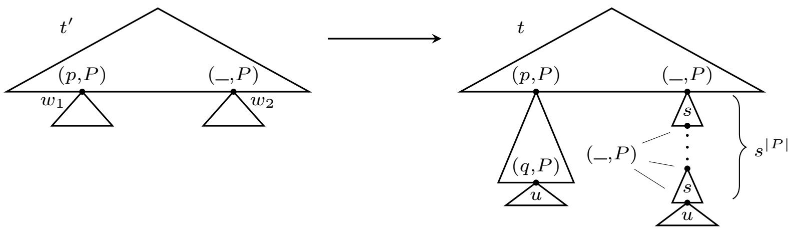


Figure 7: An illustration for the proof of Lemma 18.

for which the state at w_1 is (p, P) and for which the second entry of every state at the beginning or end of an s-loop is P. We let r be the projection of this run to the first entries of the states.

By pigeonhole principle, we find some subloop s^n below w_2 in r which loops in a state $p' \in P$. Let z be the weight of this loop and let x and y be the weights such that $\mathcal{A}$ loops s in p with weight x and in q with weight y. Due to $x \neq y$, we have $nx \neq z$ or $ny \neq z$. Since u can reach every state from P, the state p' is a rival of p or q with witnesses u and s^n. From the fact that $r(w_1) = p$ and the assumption that $p \leq q$, we see that p' can occur prefix-independently both from p and from q. This is a contradiction to the assumption that $\mathcal{A}$ does not satisfy the tree fork property.

In more detail, the proof is as follows. We define the tree $t = t'\langle s^{|P|}(u) \to w_2\rangle$ and construct a run $r \in \mathrm{Run}_{\mathcal{A}}(t)$ of $\mathcal{A}$ on t as follows. By assumption, there exists a run $r^{\mathbf{P}} \in \mathrm{Run}_{\mathcal{S}}(u, (p, P))$ and a run $r_s \in \mathrm{Run}^{\diamond}_{\mathcal{S}}((p, P), s, (p, P))$. We let $r'_2 = r_s^{|P|\langle v\rangle}\langle r^{\mathbf{P}} \to v^{|P|}\rangle$. Then $r'_2 \in \mathrm{Run}_{\mathcal{S}}(s^{|P|}(u), (p, P))$.

By Lemma 14(iv), we have $\pi_1 \circ r'(w_2) \in P$, so by Lemma 14(i) we can find $r''_2 \in \mathrm{Run}_{\mathcal{A}}(s^{|P|}(u))$ with $r''_2(\varepsilon) = \pi_1 \circ r'(w_2)$. Then $r = \pi_1(r')\langle r''_2 \to w_2\rangle \in \mathrm{Run}_{\mathcal{A}}(t)$ is a run of $\mathcal{A}$ on t and we have $r(w_2v^i) \in P$ for $0 \leq i \leq |P|$.

By pigeonhole principle, we can find $n_1, n_2 \in \{0, \ldots, |P|\}$ with $r(w_2v^{n_1}) = r(w_2v^{n_2})$ and $n_1 < n_2$. We let $p' = r(w_1v^{n_1})$ and $n = n_2 - n_1$ and show that p' is a rival of either p or q. We know that $p' \xrightarrow{s^n|z} p'$ for some weight $z \in \mathbb{R}$. Since $p' \in P$, we can also find a run $r^{p'} \in \mathrm{Run}_{\mathcal{A}}(u, p')$ which means that p' is a sibling of both p and q. We now have $p' \xrightarrow{s^n|z} p'$, $p \xrightarrow{s^n|nx} p$, and $q \xrightarrow{s^n|ny} q$. Since $x \neq y$, we have $nx \neq z$ or $ny \neq z$, or both. Thus, p' is a rival of either p or of q.

We see that $\mathcal{A}$ satisfies condition (ii) of the tree fork property as follows. Since we assumed $p \leq q$, there exists a Γ-word s^p_q and a run $r^p_q \in \mathrm{Run}^{\diamond}_{\mathcal{A}}(q, s^p_q, p)$. Therefore, either the 2-Γ-context $t_1 = t\langle \diamond \to w_1\rangle\langle \diamond \to w_2v^{n_1}\rangle$ together with the run $r\restriction_{\mathrm{pos}(t_1)}$ or the 2-Γ-context $t_2 = t_1(s^p_q, \diamond)$ together with the run $r\restriction_{\mathrm{pos}(t_1)}\langle r^p_q \to \Diamond_1(t_1)\rangle$ is a witness for condition (ii) to be satisfied. Since our assumption for this section is that $\mathcal{A}$ does not satisfy the tree fork property, this is a contradiction. $\square$

We can now prove that every run of $\mathcal{S}$ satisfies at least one of the following two conditions. If (p, P) and (q, P) are rivals in $\mathcal{S}$ with $p \leq q$, then for every run r of $\mathcal{S}$ on a tree t either (i) (p, P) does not occur in r or (ii) all states with second entry P occur along a distinguished branch of t. This property enables us to apply the idea from the word case of using markers to indicate the first visit of a rival in a run. If u is a witness for (p, P) and (q, P) to be siblings, there is in particular a run on u which leads to (p, P). This run then satisfies condition (ii) and since by Lemma 14(ii) the second entries of runs on the same tree coincide, *all* states with second entry P occur along a distinguished branch of u in *every* run of $\mathcal{S}$ on u. This is true in particular for the two rivals (p, P) and (q, P).

Theorem 19. *Let $(p, P), (q, P) \in Q_{\mathcal{S}}$ be rivals in $\mathcal{S}$ with $p \leq q$. Then for every tree $t \in T_\Gamma$ and every run $r \in \mathrm{Run}_{\mathcal{S}}(t)$ of $\mathcal{S}$ on t, at least one of the following two conditions holds.*

(i) The state (p, P) does not occur in r, i.e., $r(w) \neq (p, P)$ for all $w \in \mathrm{pos}(t)$.

(ii) All states with second entry P occur linearly in r, i.e., for all $w_1, w_2 \in \mathrm{pos}(t)$ with $\pi_2 \circ r(w_1) = \pi_2 \circ r(w_2) = P$ we have $w_1 \leq_p w_2$ or $w_2 \leq_p w_1$.

Proof. Let $(p, P), (q, P), t, r$ be as in the statement of the theorem. Assume that (i) does not hold, i.e., there is a position $w \in \mathrm{pos}(t)$ with $r(w) = (p, P)$. Let $w_1, w_2 \in \mathrm{pos}(t)$ be two positions with $\pi_2 \circ r(w_1) = \pi_2 \circ r(w_2) = P$. By Lemma 18, we see that then w_1 and w_2 are prefix-dependent on w. From the definition of the prefix relation, we see that if either $w_1 \leq_p w$ or $w_2 \leq_p w$, then all three positions are in prefix relation. We thus consider the case that $w \leq_p w_1$ and $w \leq_p w_2$. In this case, we see from Lemma 17 that w_1 and w_2 are prefix-dependent as follows. We write $w_1 = wv_1$ and $w_2 = wv_2$ and define $t' = t\lceil_w$ and $r' = r\lceil_w$. Then we have $r' \in \mathrm{Run}_{\mathcal{S}}(t')$, $r'(\varepsilon) = (p, P)$, and $\pi_2 \circ r'(v_1) = \pi_2 \circ r'(v_2) = P$. Thus, by Lemma 17 the positions v_1 and v_2 are prefix-dependent. $\square$

In the following example, we illustrate some more complex interactions which may exist between rivals, in particular between the rivals of a Schützenberger covering.

Example 20. We extend the max-plus-WTA from Figure 4 to an automaton $\mathcal{A} = (\{q_0, p, p', p'', q\}, \Gamma, \mu, \nu)$ over the alphabet $\Gamma = \{a, b, c, d, e, f\}$ where $f \in \Gamma^{(3)}$, $c \in \Gamma^{(2)}$, $a, b, e \in \Gamma^{(1)}$, and $d \in \Gamma^{(0)}$. As this example is somewhat complex, we first give some intuition of what we are trying to show with the example and how we achieve this.

Let $P = \{p, p', p'', q\}$ and let $\mathcal{S}$ be the Schützenberger covering of $\mathcal{A}$. We construct $\mathcal{A}$ such that it satisfies the following conditions.

(i) $\mathcal{A}$ is unambiguous and does not satisfy the tree fork property. We achieve unambiguity simply by making $\mathcal{A}$ top-down deterministic.

(ii) The problem showcased in Figure 4 still occurs, i.e., a nonlinearity in the first occurrence of rivals.

(iii) The state q is a rival of all of p, p', and p''.

(iv) We have $p'' \leq q \leq p \leq q \leq p'$. In particular, we cannot trivially separate these states to different automata.

(v) In $\mathcal{S}$, the state (q, P) is a rival of all of $(p, P), (p', P)$, and (p'', P).

(vi) In $\mathcal{S}$, we have $(p'', P) \leq (q, P) \leq (p, P) \leq (q, P)$, i.e., these three states cannot be trivially separated, and we have $(p'', P) \leq (p', P)$.

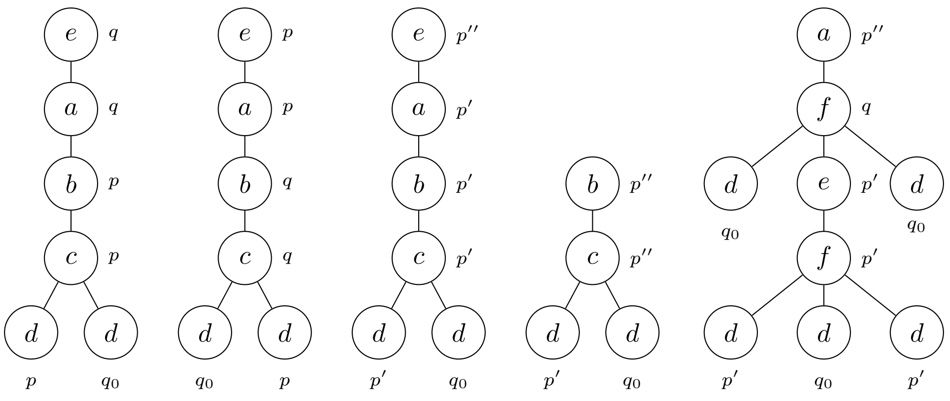
(vii) In $\mathcal{S}$, the state (p', P) may occur at arbitrarily many pairwise prefix-independent positions in the same run.

The sole purpose of the letter c is to ensure condition (ii). The purpose of b is to ensure conditions (iii) and (v), the purpose of a is to ensure the first part of condition (vi), the purpose of e is to ensure the second part of condition (vi), and the purpose of f is to ensure condition (vii).

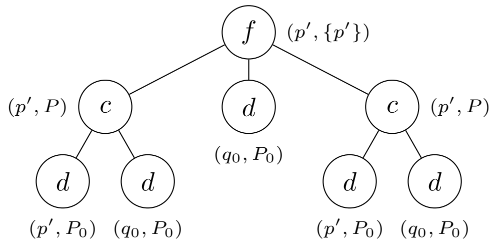
It is surprising that an automaton with the properties above exists since (1) Theorem 19 tells us that whenever (p'', P) occurs in a run, then all states with second entry P occur at pairwise prefix-dependent positions, (2) both (p'', P) and (p', P) may occur together in the same run, and (3) the state (p', P) may occur at two prefix-independent positions in the same run. We define μ and ν as follows.

$$
\begin{aligned}
\mu(d, q_0) = \mu(d, p) = \mu(d, p') &= 0 \\
\mu(p, q_0, c, p) = \mu(q_0, p, c, q) = \mu(p', q_0, c, p') = \mu(p', q_0, c, p'') &= 0 \\
\mu(p, b, p) = \mu(p', b, p') = \mu(p'', b, p'') &= 1 \\
\mu(q, b, q) &= -1 \\
\mu(p, a, q) = \mu(q, a, p) = \mu(p', a, p') = \mu(q, a, p'') &= 0 \\
\mu(p, e, p) = \mu(q, e, q) = \mu(p', e, p') = \mu(p', e, p'') &= 0 \\
\mu(q_0, p', q_0, f, q) = \mu(p', q_0, p', f, p') &= 0 \\
\nu(p'') &= 0
\end{aligned}
$$

All unspecified weights are $-\infty$. The following trees together with the runs given on them showcase the above transitions in a more graphical way.



With witnesses $u = c(d, d)$ and $s = b(\diamond)$, we see that conditions (iii) and (v) above are satisfied. Due to $(q, P) \xrightarrow{a(\diamond)|0} (p, P) \xrightarrow{a(\diamond)|0} (q, P) \xrightarrow{a(\diamond)|0} (p'', P)$ and $(p', P) \xrightarrow{e(\diamond)|0} (p'', P)$, we see that condition (vi) is also satisfied. Let $P_0 = \{q_0, p, p'\}$, then the following tree together with the run of $\mathcal{S}$ on it illustrates that (p', P) may occur nonlinearly, i.e., condition (vii) is satisfied as well.



We note that the states p and p' are also rivals in $\mathcal{A}$ with witnesses $u = d$ and $s = a(b(a(\diamond)))$. Furthermore, $\mathcal{S}$ contains many more rivals than the ones mentioned above, among others the rivals $(p', \{p', q\})$ and $(q, \{p', q\})$ with witnesses $u = f(d, d, d)$ and $s = b(\diamond)$ and the rivals $(p, \{p, p', p''\})$ and $(p', \{p, p', p''\})$ with witnesses $u = a(f(d, d, d))$ and $s = a(b(a(\diamond)))$.

We are now ready to construct the automaton which tracks the first occurrences of rivals, and whose runs we will later distribute across multiple automata in order to separate all rivals.

Construction 21. Let $R_1, \ldots, R_n \subseteq Q_{\mathcal{S}}$ be an enumeration of all (unordered) pairs of rivals of $\mathcal{S}$, i.e., for all $i \in \{1, \ldots, n\}$ we have $R_i = \{(p_i, P_i), (q_i, P_i)\}$ such that (p_i, P_i) and (q_i, P_i) are rivals in $\mathcal{S}$ and for every two rivals $(p, P), (q, P) \in Q_{\mathcal{S}}$, we have $R_i = \{(p, P), (q, P)\}$ for some $i \in \{1, \ldots, n\}$. Since by Lemma 15, we may assume that all rivals in $\mathcal{A}$ are in $\leq$-relation, we assume in the following that p_i and q_i are named such that $p_i \leq q_i$ for all $i \in \{1, \ldots, n\}$.

For each pair of rivals R_i, we define a set of markers by $I_i = \{0, |Q| + 1\} \cup (\{1, \ldots, |Q|\} \times R_i)$. The set of all combined records of markers is defined by $I = I_1 \times \ldots \times I_n$. For $\bar{a} \in I$, we denote by $\bar{a}[i]$ the i-th entry of $\bar{a}$.

Intuitively, the states of our new automaton will consist of a state from $\mathcal{S}$ together with a record of markers from I. However, in order to properly update markers, we need to know in each step the records of *all* other runs as well. Thus, our states will be from $Q_{\mathcal{S}} \times I \times \mathcal{P}(Q_{\mathcal{S}} \times I)$.

In order to define the transition function of our new automaton, we first define how markers are updated. In some sense, this is similar to the *context successor* defined in [4]. Assume we transition into the state $\mathbf{q} \in Q_{\mathcal{S}}$, we have m subtrees below our current position in the tree, the runs we consider on these subtrees have obtained markers $\bar{a}_1, \ldots, \bar{a}_m \in I$, and the sets of states we *could* be in on these trees, together with their markers, are given by $A_1, \ldots, A_m \subseteq Q_{\mathcal{S}} \times I$.

Every pair $(\mathbf{p}, \bar{a}) \in A_k$ corresponds to exactly one run of $\mathcal{S}$ on the k-th subtree together with its markers. Since $\mathcal{S}$ is unambiguous, we can therefore assume that $|A_k| \leq |Q|$. Also, since $\bar{a}_k$ is the marker of a run on the k-th subtree, we may assume that $(Q_{\mathcal{S}} \times \{\bar{a}_k\}) \cap A_k \neq \emptyset$.

For $k \in \{1, \ldots, m\}$ and $i \in \{1, \ldots, n\}$, we define the sets of unassigned counters $B_k[i] \subseteq \{1, \ldots, |Q|\}$ by

$$B_k[i] = \{1, \ldots, |Q|\} \setminus \{j \mid \exists (\mathbf{p}, \bar{a}) \in A_k \text{ with } \bar{a}[i] \in \{j\} \times R_i\}.$$

Then if for all $k \in \{1, \ldots, m\}$ we have $|A_k| \leq |Q|$ and $(Q_{\mathcal{S}} \times \{\bar{a}_k\}) \cap A_k \neq \emptyset$, we define the record of markers $\bar{b}$ for our current position by (explanations below)

$$\bar{b}[i] = \begin{cases} 0 & \text{if } m = 0 \text{ and } \mathbf{q} \notin R_i \\ (1, \mathbf{q}) & \text{if } m = 0 \text{ and } \mathbf{q} \in R_i \\ \bar{a}_k[i] & \text{if } k \in \{1, \ldots, m\} \text{ satisfies: } \bar{a}_l[i] = 0 \text{ for all } l \neq k \text{ and either } \bar{a}_k[i] \neq 0 \text{ or } \mathbf{q} \notin R_i \\ (\min B_k[i], \mathbf{q}) & \text{if } \mathbf{q} \in R_i \text{ and } k \in \{1, \ldots, m\} \text{ satisfies:} \\ & \bar{a}_k[i] = 0 \text{ and for all } l \neq k \text{ and all } (\mathbf{p}, \bar{a}) \in A_l\colon \bar{a}[i] = 0 \\ |Q| + 1 & \text{otherwise} \end{cases}$$

for $i \in \{1, \ldots, n\}$. If $|A_k| > |Q|$ or $Q_{\mathcal{S}} \times \{\bar{a}_k\} \cap A_k = \emptyset$ for some k, we let $\bar{b}[1] = \ldots = \bar{b}[n] = |Q| + 1$.

Note that $\min B_k[i]$ in above case distinction always exists since $|A_k| \leq |Q|$, $(Q_{\mathcal{S}} \times \{\bar{a}_k\}) \cap A_k \neq \emptyset$, and in the case in question we have $\bar{a}_k[i] = 0$. We define $\mathcal{I}(\mathbf{q}, \bar{a}_1, \ldots, \bar{a}_m, A_1, \ldots, A_m) = \bar{b}$.

Case 1 of the definition above means our current position is a leaf and $\mathbf{q}$ is not from R_i, so we assign the dummy marker 0. Case 2 means our current position is a leaf and $\mathbf{q}$ is from R_i, so we assign the marker $(1, \mathbf{q})$. Case 3 means that either (1) there is exactly one subtree below our current position which already obtained a marker different from 0 and we keep this marker for our current position, or (2) the markers of all subtrees are 0 and $\mathbf{q}$ is also not from R_i, so we continue with the dummy marker 0.

Case 4 means the markers of all subtrees below our current position are 0, the state $\mathbf{q}$ is from R_i, and there is at most one subtree on which runs exist that obtained a marker for R_i. Then, we take the smallest number which is not already used in a marker for R_i in any run on this subtree, and use this number together with $\mathbf{q}$ as the marker for our current position.

Case 5, the "otherwise-case", applies in two situations. This case means that either (1) two distinct subtrees below our current position have already obtained a marker, or that (2) all markers below our current position are 0 and $\mathbf{q}$ is from R_i, but we cannot apply case 4 as there are two distinct subtrees on which runs exist which obtained markers for R_i. In other words, markers were assigned nonlinearly, and our run satisfies only condition (i) of Theorem 19. In this case, we assign the dummy marker $|Q| + 1$.

The extra case covers the situation where in case 4, the set $B_k[i]$ would be empty. This case is necessary to ensure our definition is formally complete, but in our applications of the operator $\mathcal{I}$ it will not actually occur.

We define our "run-marking" max-plus-WTA $\mathcal{B} = (\tilde{Q}, \Gamma, \tilde{\mu}, \tilde{\nu})$ as follows. We let $\tilde{Q}' = Q_{\mathcal{S}} \times I \times \mathcal{P}(Q_{\mathcal{S}} \times I)$ and let $\mathcal{B}$ be the trim part of the automaton $\mathcal{B}' = (\tilde{Q}', \Gamma, \tilde{\mu}', \tilde{\nu}')$ defined for $a \in \Gamma$ with $\mathrm{rk}_\Gamma(a) = m$ and $(\mathbf{p}_0, \bar{a}_0, A_0), \ldots, (\mathbf{p}_m, \bar{a}_m, A_m) \in Q_{\mathcal{S}} \times I \times \mathcal{P}(Q_{\mathcal{S}} \times I)$ by

$$\tilde{\mu}'((\mathbf{p}_1, \bar{a}_1, A_1), \ldots, (\mathbf{p}_m, \bar{a}_m, A_m), a, (\mathbf{p}_0, \bar{a}_0, A_0)) =$$

$$\begin{cases} \mu_{\mathcal{S}}(\mathbf{p}_1, \ldots, \mathbf{p}_m, a, \mathbf{p}_0) & \text{if } \bar{a}_0 = \mathcal{I}(\mathbf{p}_0, \bar{a}_1, \ldots, \bar{a}_m, A_1, \ldots, A_m) \text{ and} \\ & A_0 = \{(\mathbf{q}_0, \bar{b}_0) \in Q_{\mathcal{S}} \times I \mid \exists((\mathbf{q}_1, \bar{b}_1), \ldots, (\mathbf{q}_m, \bar{b}_m)) \in A_1 \times \ldots \times A_m \\ & \text{with } \mu_{\mathcal{S}}(\mathbf{q}_1, \ldots, \mathbf{q}_m, a, \mathbf{q}_0) \neq -\infty \text{ and } \bar{b}_0 = \mathcal{I}(\mathbf{q}_0, \bar{b}_1, \ldots, \bar{b}_m, A_1, \ldots, A_m)\} \\ -\infty & \text{otherwise} \end{cases}$$

$$\tilde{\nu}'(\mathbf{p}_0, \bar{a}_0, A_0) = \nu_{\mathcal{S}}(\mathbf{p}_0).$$

For the rest of this section, we show that the automaton $\mathcal{B}$ "does what we want": We show that $\mathcal{B}$ is unambiguous, that it has the same behavior as $\mathcal{A}$, and that we can indeed separate its rivals by distributing runs with a different marker across different automata which then satisfy the twins property.

Let $\tilde{\pi}_1 \colon Q_{\mathcal{S}} \times I \times \mathcal{P}(Q_{\mathcal{S}} \times I) \to Q_{\mathcal{S}}$, $(\mathbf{p}, \bar{a}, A) \mapsto \mathbf{p}$, $\tilde{\pi}_2 \colon Q_{\mathcal{S}} \times I \times \mathcal{P}(Q_{\mathcal{S}} \times I) \to I$, $(\mathbf{p}, \bar{a}, A) \mapsto \bar{a}$, and $\tilde{\pi}_3 \colon Q_{\mathcal{S}} \times I \times \mathcal{P}(Q_{\mathcal{S}} \times I) \to \mathcal{P}(Q_{\mathcal{S}} \times I)$, $(\mathbf{p}, \bar{a}, A) \mapsto A$ be the projections. We prove the following basic observations about $\mathcal{B}$.

Lemma 22. *Let $t \in T_\Gamma$ be a tree. Then the following statements hold.*

(i) For every run $r \in \mathrm{Run}_{\mathcal{B}}(t)$ we have $(\tilde{\pi}_1 \circ r(w), \tilde{\pi}_2 \circ r(w)) \in \tilde{\pi}_3 \circ r(w)$. In particular, the only applications of the operator $\mathcal{I}$ are for sets A_k and tuples $\bar{a}_k$ with $(Q_{\mathcal{S}} \times \{\bar{a}_k\}) \cap A_k \neq \emptyset$.

(ii) For every two runs $r_1, r_2 \in \mathrm{Run}_{\mathcal{B}}(t)$ and every position $w \in \mathrm{pos}(t)$ we have $\tilde{\pi}_3 \circ r_1(w) = \tilde{\pi}_3 \circ r_2(w)$.

(iii) For every run $r \in \mathrm{Run}_{\mathcal{B}}(t)$ and position $w \in \mathrm{pos}(t)$ we have $\tilde{\pi}_3 \circ r(w) = \{(\mathbf{q}, \bar{b}) \in Q_{\mathcal{S}} \times I \mid \exists r' \in \mathrm{Run}_{\mathcal{B}}(t\restriction_w)$ with $r'(\varepsilon) = (\mathbf{q}, \bar{b}, \tilde{\pi}_3 \circ r(w))\}$.

(iv) The projection $\tilde{\pi}_1$ induces a bijection $\tilde{\pi}_1 \colon \mathrm{Run}_{\mathcal{B}}(t) \to \mathrm{Run}_{\mathcal{S}}(t)$ by $r \mapsto \tilde{\pi}_1 \circ r$.

(v) $\mathcal{B}$ is trim, unambiguous, and satisfies $[\![\mathcal{B}]\!] = [\![\mathcal{A}]\!]$.

(vi) For every run $r \in \mathrm{Run}_{\mathcal{B}}(t)$ and position $w \in \mathrm{pos}(t)$ we have $|\tilde{\pi}_3 \circ r(w)| \leq |Q|$. In particular, the only applications of the operator $\mathcal{I}$ are for sets A_k with $|A_k| \leq |Q|$.

(vii) For every Γ-word s and two states $\tilde{p}, \tilde{q} \in \tilde{Q}$ with $\tilde{p} \xrightarrow{s|x} \tilde{q}$, we have $\tilde{\pi}_1(\tilde{p}) \xrightarrow{s|x} \tilde{\pi}_1(\tilde{q})$.

Proof. (i) Let $t \in T_\Gamma$ and $r \in \mathrm{Run}_{\mathcal{B}}(t)$ and for contradiction, let $w \in \mathrm{pos}(t)$ be a prefix-maximal position for which (i) does not hold. We let $m = \mathrm{rk}_\Gamma(t(w))$ and write $r(w) = (\mathbf{p}, \bar{a}, A)$ and $r(wj) = (\mathbf{p}_j, \bar{a}_j, A_j)$ for $j \in \{1, \ldots, m\}$. Since r is a run of $\mathcal{B}$ on t, we have $\mu_{\mathcal{S}}(\mathbf{p}_1, \ldots, \mathbf{p}_m, a, \mathbf{p}) \neq -\infty$ and $\bar{a} = \mathcal{I}(\mathbf{p}, \bar{a}_1, \ldots, \bar{a}_m, A_1, \ldots, A_m)$. By assumption, we have $(\mathbf{p}_j, \bar{a}_j) \in A_j$ for all $j \in \{1, \ldots, m\}$, so $(\mathbf{p}, \bar{a}) \in A$ follows from the definition of $\tilde{\mu}$. This is a contradiction, thus w does not exist.

(ii) Let $t \in T_\Gamma$ and $r_1, r_2 \in \mathrm{Run}_{\mathcal{B}}(t)$ and let $w \in \mathrm{pos}(t)$ be a prefix-maximal position for which (ii) does not hold. From the definition of $\tilde{\mu}$, it is immediately clear that $\tilde{\pi}_3 \circ r_1(w) = \tilde{\pi}_3 \circ r_2(w)$, so w does not exist.

(iii) Let $t \in T_\Gamma$ and $r \in \mathrm{Run}_\mathcal{B}(t)$ and let $w \in \mathrm{pos}(t)$ be a prefix-maximal position for which (iii) does not hold. We will deduce that (iii) holds for w. We let $m = \mathrm{rk}_\Gamma(t(w))$ and write $r(w) = (\mathbf{p}, \bar{a}, A)$ and $r(wj) = (\mathbf{p}_j, \bar{a}_j, A_j)$ for $j \in \{1, \dots, m\}$.

First, let $(\mathbf{q}, \bar{b}) \in A$, then there are states $((\mathbf{q}_1, \bar{b}_1), \dots, (\mathbf{q}_m, \bar{b}_m)) \in A_1 \times \dots \times A_m$ with $\mu(\mathbf{q}_1, \dots, \mathbf{q}_m, a, \mathbf{q}) \neq -\infty$ and $\bar{b} = \mathcal{I}(\mathbf{q}, \bar{b}_1, \dots, \bar{b}_m, A_1, \dots, A_m)$. By assumption on w, for every j we find $r_j \in \mathrm{Run}_\mathcal{B}(t\restriction_{wj})$ with $r_j(\varepsilon) = (\mathbf{q}_j, \bar{b}_j, A_j)$. Then by definition of $\tilde{\mu}$, we see that the quasi-run $r' \colon \mathrm{pos}(t\restriction_w) \to \tilde{Q}$ defined by $r'(\varepsilon) = (\mathbf{q}, \bar{b}, A)$ and $r'(jv) = r_j(v)$ is a run of $\mathcal{B}$ on $t\restriction_w$ with $r'(\varepsilon) = (\mathbf{q}, \bar{b}, A)$.

On the other hand, let $r' \in \mathrm{Run}_\mathcal{B}(t\restriction_w)$, with $r'(\varepsilon) = (\mathbf{q}, \bar{b}, A)$ for some $(\mathbf{q}, \bar{b}) \in Q_\mathcal{S} \times I$. Then from (i) we obtain $(\mathbf{q}, \bar{b}) \in A$. Thus, (iii) holds for w.

(iv) Let $t \in T_\Gamma$. By definition of $\tilde{\mu}$, it is clear that for $r \in \mathrm{Run}_\mathcal{B}(t)$ we have $\tilde{\pi}_1 \circ r \in \mathrm{Run}_\mathcal{S}(t)$. For the injectivity of $\tilde{\pi}_1 \colon \mathrm{Run}_\mathcal{B}(t) \to \mathrm{Run}_\mathcal{S}(t)$, let $r_1, r_2 \in \mathrm{Run}_\mathcal{B}(t)$ with $\tilde{\pi}_1 \circ r_1 = \tilde{\pi}_1 \circ r_2$. Let $w \in \mathrm{pos}(t)$ be a prefix-maximal position from the set $\{v \in \mathrm{pos}(t) \mid r_1(v) \neq r_2(v)\}$. Then $\tilde{\pi}_1 \circ r_1(w) = \tilde{\pi}_1 \circ r_2(w)$ and for all $j \in \{1, \dots, \mathrm{rk}_\Gamma(t(w))\}$ we have $r_1(wj) = r_2(wj)$. From the definition of $\tilde{\mu}$, it is immediately clear that $r_1(w) = r_2(w)$ follows, i.e., w as chosen does not exist.

For surjectivity, we let $r' \in \mathrm{Run}_\mathcal{S}(t)$ and define a run $r \in \mathrm{Run}_\mathcal{B}(t)$ inductively as follows. For a leaf $w \in \mathrm{pos}(t)$, we let $\mathbf{p} = r'(w)$, $\bar{a} = \mathcal{I}(\mathbf{p})$, $A = \{(\mathbf{q}_0, \mathcal{I}(\mathbf{q}_0) \mid \mu_\mathcal{S}(t(w), \mathbf{q}_0) \neq -\infty\}$, and $r(w) = (\mathbf{p}, \bar{a}, A)$.

Now let $w \in \mathrm{pos}(t)$ with $\mathrm{rk}_\Gamma(t(w)) = m$ such that r is defined on $w1, \dots, wm$. We write $\mathbf{p} = r'(w)$ and $r(wj) = (\mathbf{p}_j, \bar{a}_j, A_j)$ for $j \in \{1, \dots, m\}$. We let $\bar{a}_0 = \mathcal{I}(\mathbf{p}_0, \bar{a}_1, \dots, \bar{a}_m, A_1, \dots, A_m)$ and $A = \{(\mathbf{q}_0, \bar{b}_0) \in Q_\mathcal{S} \times I \mid \exists((\mathbf{q}_1, \bar{b}_1), \dots, (\mathbf{q}_m, \bar{b}_m)) \in A_1 \times \dots \times A_m$ with $\mu_\mathcal{S}(\mathbf{q}_1, \dots, \mathbf{q}_m, a, \mathbf{q}_0) \neq -\infty$ and $\bar{b}_0 = \mathcal{I}(\mathbf{q}_0, \bar{b}_1, \dots, \bar{b}_m, A_1, \dots, A_m)\}$, and $r(w) = (\mathbf{p}, \bar{a}, A)$. Thus, we obtain a run $r \in \mathrm{Run}_\mathcal{B}(t)$ with $\tilde{\pi}_1 \circ r(w) = r'$.

(v) $\mathcal{B}$ is trim by definition. Let $t \in T_\Gamma$. By definition of $\tilde{\mu}$, for every run $r \in \mathrm{Run}_\mathcal{S}(t)$ we have $\mathrm{wt}_\mathcal{B}(t, r) = \mathrm{wt}_\mathcal{S}(t, \tilde{\pi}_1 \circ r)$. By definition of $\tilde{\nu}$, we also have $\tilde{\nu}(r(\varepsilon)) = \nu(\tilde{\pi}_1 \circ r(\varepsilon))$. By (iv), we have $|\mathrm{Acc}_\mathcal{B}(t)| = |\mathrm{Acc}_\mathcal{S}(t)| \leq 1$, which means $\mathcal{B}$ is unambiguous, and we have $[\![\mathcal{B}]\!](t) = [\![\mathcal{S}]\!](t) = [\![\mathcal{A}]\!](t)$.

(vi) The automaton $\mathcal{A}$ is assumed to be trim and unambiguous, so we have $|\mathrm{Run}_\mathcal{A}(t)| \leq |Q|$ for every $t \in T_\Gamma$. Furthermore, the projections π_1 and $\tilde{\pi}_1$ are bijections by Lemma 14(iii) and (iv) above. Let $t \in T_\Gamma$, $r \in \mathrm{Run}_\mathcal{B}(t)$, and $w \in \mathrm{pos}(t)$. From (iii), we see that $|\tilde{\pi}_3 \circ r(w)| \leq |\mathrm{Run}_\mathcal{B}(t\restriction_w)| = |\mathrm{Run}_\mathcal{S}(t\restriction_w)| = |\mathrm{Run}_\mathcal{A}(t\restriction_w)| \leq |Q|$.

(vii) Let s be a Γ-word and $\tilde{p}, \tilde{q} \in \tilde{Q}$ be two states with $\tilde{p} \xrightarrow{s|x} \tilde{q}$, then there is a run $r \in \mathrm{Run}^\diamond_\mathcal{B}(\tilde{p}, s, \tilde{q})$ with $\mathrm{wt}^\diamond_\mathcal{B}(s, r) = x$. By definition of $\tilde{\mu}$, we have $\tilde{\pi}_1 \circ r \in \mathrm{Run}^\diamond_\mathcal{S}(s)$ and $\mathrm{wt}^\diamond_\mathcal{B}(s, r) = \mathrm{wt}^\diamond_\mathcal{S}(s, \tilde{\pi}_1(r))$, so we have $\tilde{\pi}_1(\tilde{p}) \xrightarrow{s|x} \tilde{\pi}_1(\tilde{q})$. ☐

Next, we prove two basic statements about how $\mathcal{B}$ sets markers. Assume we have some run in which a state $(\mathbf{p}, \bar{a}, A)$ occurs. First, we show that if $\bar{a}[i] \neq 0$ for some i, then in the past, we must have visited one of the rivals in R_i. Second, we show that if A contains a state $(\mathbf{q}, \bar{b})$ with $\bar{b}[i] \neq 0$ for some i, then we must have visited some state with second entry P_i in the past.

Lemma 23. *Let $t \in T_\Gamma$ be a tree, $r \in \mathrm{Run}_\mathcal{B}(t)$ be a run of $\mathcal{B}$ on t, let $w \in \mathrm{pos}(t)$ be a position in t, assume that $r(w) = (\mathbf{p}, \bar{a}, A)$, and let $i \in \{1, \dots, n\}$. Then the following statements hold.*

(i) If $\bar{a}[i] \neq 0$, then there is a position $v \in \mathrm{pos}(t)$ with $w \leq_p v$ and $\tilde{\pi}_1 \circ r(v) \in R_i$.

(ii) If there exists $(\mathbf{q}, \bar{b}) \in A$ with $\bar{b}[i] \neq 0$, then there is a position $v \in \mathrm{pos}(t)$ with $w \leq_p v$ such that $\pi_2 \circ \tilde{\pi}_1 \circ r(v) = P_i$.

Proof. (i) Assume $\bar{a}[i] \neq 0$. We choose v prefix-maximal from the set $\{w' \in \mathrm{pos}(t) \mid w \leq_p w'$ and $r(w') = (\mathbf{q}, \bar{b}, B)$ with $\bar{b}[i] \neq 0\}$. This set is not empty since it contains w. We write $r(v) = (\mathbf{q}, \bar{b}, B)$. If $\mathbf{q} \notin R_i$ would hold, we see from the definition of $\tilde{\mu}$, the definition of the operator $\mathcal{I}$, and the fact that we chose v prefix-maximal from above set, that either case 1 or case 3 of the definition of $\mathcal{I}$ would apply in the definition of $\bar{b}[i]$. Thus, $\bar{b}[i] = 0$ would hold, which is not the case. Therefore, $\mathbf{q} \in R_i$ holds.

(ii) Assume there is $(\mathbf{q}, \bar{b}) \in A$ with $\bar{b}[i] \neq 0$. By Lemma 22(iii), there is a run $r' \in \mathrm{Run}_\mathcal{B}(t\restriction_w)$ with $r'(\varepsilon) = (\mathbf{q}, \bar{b}, A)$. Then by (i), there exists $v \in \mathrm{pos}(t\restriction_w)$ with $\tilde{\pi}_1 \circ r'(v) \in R_i$. Furthermore, we have $r\restriction_w \in \mathrm{Run}_\mathcal{B}(t\restriction_w)$. Combining Lemma 22(iv) and Lemma 14(ii), we have $P_i = \pi_2 \circ \tilde{\pi}_1 \circ r'(v) = \pi_2 \circ \tilde{\pi}_1 \circ r\restriction_w(v)$. Thus, we see that $\pi_2 \circ \tilde{\pi}_1 \circ r(wv) = P_i$. ☐

Next, we essentially prove that markers for R_i are properly set in runs where states with P_i as a second entry occur only linearly. That is, we show that in these runs, a marker for R_i is only set when a rival from R_i is actually visited, and that it cannot be altered afterwards.

Lemma 24. *Let $t \in T_\Gamma$, $i \in \{1, \dots, n\}$, and $r \in \mathrm{Run}_{\mathcal{B}}(t)$ such that for all positions $v_1, v_2 \in \mathrm{pos}(t)$ with $\pi_2 \circ \tilde{\pi}_1 \circ r(v_1) = \pi_2 \circ \tilde{\pi}_1 \circ r(v_2) = P_i$ we have $v_1 \leq_p v_2$ or $v_2 \leq_p v_1$. If $w \in \mathrm{pos}(t)$ is the prefix-largest position of t with $\tilde{\pi}_1 \circ r(w) \in R_i$ then the following properties are satisfied*

(i) The marker $\tilde{\pi}_2 \circ r(w)$ is defined using case 2 or case 4 of the definition of the operator $\mathcal{I}$.

(ii) For all positions $v \in \mathrm{pos}(t)$ with $v \leq_p w$ we have $\tilde{\pi}_2 \circ r(v)[i] = \tilde{\pi}_2 \circ r(w)[i] \in \{1, \dots, |Q|\} \times \{\tilde{\pi}_1 \circ r(w)\}$.

(iii) For all positions $v \in \mathrm{pos}(t) \setminus \{w\}$ such that either v and w are prefix-independent or $w \leq_p v$, we have $\tilde{\pi}_2 \circ r(v)[i] = 0$.

Proof. Let $m = \mathrm{rk}_\Gamma(t(w))$. If $m = 0$, $\tilde{\pi}_2 \circ r(w)[i]$ is obviously defined using case 2. Otherwise, since w is the prefix-largest among all positions w' with $\tilde{\pi}_1 \circ r(w') \in R_i$, we have by Lemma 23(i) that $\tilde{\pi}_2 \circ r(v)[i] = 0$ for all $v \in \mathrm{pos}(t) \setminus \{w\}$ with $w \leq_p v$. In particular, we have $\tilde{\pi}_2 \circ r(wj)[i] = 0$ for all $j \in \{1, \dots, m\}$. Thus, by Lemma 23(ii) and our assumptions on r and w, we see that case 4 of the definition of the operator $\mathcal{I}$ applies in the definition of $\tilde{\pi}_2 \circ r(w)[i]$. Thus, $\tilde{\pi}_2 \circ r(w)[i] \in \{1, \dots, |Q|\} \times \{\tilde{\pi}_1 \circ r(w)\}$.

We show (ii). For contradiction, let $v \in \mathrm{pos}(t)$ be the prefix-largest position with $v \leq_p w$ and $\tilde{\pi}_2 \circ r(v)[i] \neq \tilde{\pi}_2 \circ r(w)[i]$. Let $m = \mathrm{rk}_\Gamma(t(v))$ and $j \in \{1, \dots, m\}$ such that $w = vjv'$ for some v'. Then $\tilde{\pi}_2 \circ r(vj)[i] = \tilde{\pi}_2 \circ r(w)[i] \neq 0$, and by Lemma 23(i) and our assumption on r, we have $\tilde{\pi}_2 \circ r(vk)[i] = 0$ for all $k \neq j$. Thus, case 3 of the definition of $\mathcal{I}$ applies in the definition of $\tilde{\pi}_2 \circ r(v)[i]$, so $\tilde{\pi}_2 \circ r(v)[i] = \tilde{\pi}_2 \circ r(vj)[i] = \tilde{\pi}_2 \circ r(w)[i]$. This means v as chosen does not exist.

Finally let $v \in \mathrm{pos}(t) \setminus \{w\}$ be such that either v and w are prefix-independent or $w \leq_p v$. Then from Lemma 23(i) and our assumption on r we immediately obtain $\tilde{\pi}_2 \circ r(v)[i] = 0$. □

In the next lemma, we show that if two states are rivals in $\mathcal{B}$, then their records of markers differ. The reasoning for this is exactly the same as in our intuitive description at the beginning of this section.

Lemma 25. *If $(\mathbf{p}, \bar{a}, A)$ and $(\mathbf{q}, \bar{b}, B)$ are rivals in $\mathcal{B}$, then $\mathbf{p}$ and $\mathbf{q}$ are rivals in $\mathcal{S}$ and for $i \in \{1, \dots, n\}$ with $R_i = \{\mathbf{p}, \mathbf{q}\}$, we have $\bar{a}[i] \neq \bar{b}[i]$ and $\bar{a}[i], \bar{b}[i] \in \{1, \dots, |Q|\} \times \{\mathbf{p}, \mathbf{q}\}$.*

Proof. Let $\tilde{p} = (\mathbf{p}, \bar{a}, A)$ and $\tilde{q} = (\mathbf{q}, \bar{b}, B)$ be rivals in $\mathcal{B}$. Let u and s be as in the definition of rivals and let $r^{\tilde{p}} \in \mathrm{Run}^{\diamond}_{\mathcal{B}}(u, \tilde{p})$ and $r^{\tilde{q}} \in \mathrm{Run}^{\diamond}_{\mathcal{B}}(u, \tilde{q})$. Then by Lemma 22(iv), we have $\tilde{\pi}_1(r^{\tilde{p}}) \in \mathrm{Run}_{\mathcal{S}}(u, \mathbf{p})$ and $\tilde{\pi}_1(r^{\tilde{q}}) \in \mathrm{Run}_{\mathcal{S}}(u, \mathbf{q})$. By Lemma 22(vii), we also have $\mathbf{p} \xrightarrow{s|x} \mathbf{p}$ and $\mathbf{q} \xrightarrow{s|y} \mathbf{q}$ with $x \neq y$, thus $\mathbf{p}$ and $\mathbf{q}$ are rivals in $\mathcal{S}$. Let $i \in \{1, \dots, n\}$ with $R_i = \{\mathbf{p}, \mathbf{q}\}$. We may assume that $\mathbf{p} = (p_i, P_i)$ and $\mathbf{q} = (q_i, P_i)$.

We show that $\bar{a}[i], \bar{b}[i] \notin \{0, |Q| + 1\}$. We let $r^{\mathbf{p}} = \tilde{\pi}_1 \circ r^{\tilde{p}}$ and $r^{\mathbf{q}} = \tilde{\pi}_1 \circ r^{\tilde{q}}$. We have $r^{\mathbf{p}}(\varepsilon) = (p_i, P_i)$ and we assumed $p_i \leq q_i$, so by Theorem 19 we obtain that for every two positions $v_1, v_2 \in \mathrm{pos}(u)$ with $\pi_2 \circ r^{\mathbf{p}}(v_1) = \pi_2 \circ r^{\mathbf{p}}(v_2) = P_i$, we have $v_1 \leq_p v_2$ or $v_2 \leq_p v_1$. This also holds for $r^{\mathbf{q}}$ since by by Lemma 14(ii) we have $\pi_2 \circ r^{\mathbf{p}} = \pi_2 \circ r^{\mathbf{q}}$.

Let $w_p \in \mathrm{pos}(u)$ be the prefix-largest position of u with $r^{\mathbf{p}}(w_p) \in R_i$ and $w_q \in \mathrm{pos}(u)$ be the prefix-largest position with $r^{\mathbf{q}}(w_q) \in R_i$. That w_p and w_q exist is clear from the fact that $r^{\mathbf{p}}(\varepsilon) \in R_i$ and $r^{\mathbf{q}}(\varepsilon) \in R_i$. By Lemma 24(ii), we have $\bar{a}[i] = \tilde{\pi}_2 \circ r^{\tilde{p}}(w_p)[i] \in \{1, \dots, |Q|\} \times \{\tilde{\pi}_1 \circ r^{\tilde{p}}(w_p)\}$ and $\bar{b}[i] = \tilde{\pi}_2 \circ r^{\tilde{q}}(w_q)[i] \in \{1, \dots, |Q|\} \times \{\tilde{\pi}_1 \circ r^{\tilde{q}}(w_q)\}$.

We show that $\bar{a}[i] \neq \bar{b}[i]$ and consider two cases. First, if $w_p = w_q$ we assume for contradiction that $\bar{a}[i] = \bar{b}[i]$. Then we see that $r^{\mathbf{p}}(w_p) = r^{\mathbf{q}}(w_q) \in R_i$, and we also have $r^{\mathbf{p}}(\varepsilon) = \mathbf{p}$ and $r^{\mathbf{q}}(\varepsilon) = \mathbf{q}$. It follows that with $s = u\langle \diamond \to w_p \rangle$, we have $\pi_1 \circ r^{\mathbf{p}}(w_p) \xrightarrow{s|z_1} p_i$ and $\pi_1 \circ r^{\mathbf{p}}(w_p) \xrightarrow{s|z_2} q_i$ for weights $z_1, z_2 \in \mathbb{R}$. Thus, $\mathcal{A}$ satisfies condition (i) of the tree fork property, which is a contradiction. Therefore, $\bar{a}[i] = \bar{b}[i]$ cannot hold when $w_p = w_q$.

Now assume without loss of generality that $w_p \leq_p w_q$ with $w_p \neq w_q$ and write $w_q = w_p j v$ and $r^{\tilde{q}}(w_p j) = (\mathbf{q}', \bar{b}', A_j)$. By Lemma 22(i) and Lemma 22(ii), we then have $(\mathbf{q}', \bar{b}') \in A_j = \tilde{\pi}_3 \circ r^{\tilde{p}}(w_p j)$. By Lemma 24(i), we know that $\tilde{\pi}_2 \circ r^{\tilde{p}}(w_p)[i]$ is defined using case 4 of the definition of $\mathcal{I}$, so $\bar{a}[i] \neq \bar{b}[i]$ must hold. □

We turn to our final construction where we distribute the runs of $\mathcal{B}$ across multiple automata. For every record of markers $\bar{c} \in I$, we construct one automaton $\mathcal{B}_{\bar{c}}$ which for each pair of rivals R_i admits only runs using the markers 0 and $\bar{c}[i]$. All runs in which rivals occur nonlinearly are covered by admitting the marker $|Q| + 1$. All other runs are covered by admitting an appropriate marker from $\{1, \dots, |Q|\} \times R_i$.

Construction 26. For every tuple $\bar{c} \in I$, we define a max-plus-WTA $\mathcal{B}_{\bar{c}} = (\tilde{Q}_{\bar{c}}, \Gamma, \tilde{\mu}, \tilde{\nu})$ by removing states from $\mathcal{B}$ through

$$\tilde{Q}_{\bar{c}} = \{(\mathbf{p}, \bar{a}, A) \in \tilde{Q} \mid \text{for all } i \in \{1, \ldots, n\} \text{ it holds: if } \bar{c}[i] = |Q| + 1 \text{ then } \mathbf{p} \neq (p_i, P_i),$$
$$\text{and if } \bar{c}[i] \neq |Q| + 1 \text{ then } \bar{a}[i] \in \{0, \bar{c}[i]\}\}.$$

Finally, we formally prove that the automata $\mathcal{B}_{\bar{c}}$ are unambiguous, that their pointwise maximum is equivalent to the behavior of $\mathcal{A}$, and that they all satisfy the twins property, which means that they can be determinized. We note that the construction of the automata $\mathcal{B}_{\bar{c}}$ is optimized for provability, so we omit analyzing their size and the complexity of their construction.

Theorem 27. *We have* $[\![\mathcal{A}]\!] = \max_{\bar{c} \in I} [\![\mathcal{B}_{\bar{c}}]\!]$ *and for every* $\bar{c} \in I$*, the automaton* $\mathcal{B}_{\bar{c}}$ *is unambiguous and satisfies the twins property.*

Proof. The unambiguity of $\mathcal{B}_{\bar{c}}$ follows from the unambiguity of $\mathcal{B}$. To see that $\mathcal{B}_{\bar{c}}$ satisfies the twins property, let $(\mathbf{p}, \bar{a}, A), (\mathbf{q}, \bar{b}, B) \in \tilde{Q}_{\bar{c}}$ be rivals in $\mathcal{B}_{\bar{c}}$. Then $(\mathbf{p}, \bar{a}, A)$ and $(\mathbf{q}, \bar{b}, B)$ are also rivals in $\mathcal{B}$, so by Lemma 25 for some $i \in \{1, \ldots, n\}$ we have $\bar{a}[i] \neq \bar{b}[i]$ and $\bar{a}[i], \bar{b}[i] \notin \{0, |Q| + 1\}$. By definition of $\mathcal{B}_{\bar{c}}$, this means $(\mathbf{p}, \bar{a}, A), (\mathbf{q}, \bar{b}, B) \in \tilde{Q}_{\bar{c}}$ is impossible, so there are no rivals in $\mathcal{B}_{\bar{c}}$ and $\mathcal{B}_{\bar{c}}$ satisfies the twins property.

To show that $[\![\mathcal{A}]\!] = \max_{\bar{c} \in I} [\![\mathcal{B}_{\bar{c}}]\!]$, we show that for every tree $t \in T_\Gamma$ we have $\mathrm{Run}_{\mathcal{B}}(t) = \bigcup_{\bar{c} \in I} \mathrm{Run}_{\mathcal{B}_{\bar{c}}}(t)$. From this, it follows that $\max_{\bar{c} \in I} [\![\mathcal{B}_{\bar{c}}]\!] = [\![\mathcal{B}]\!] = [\![\mathcal{A}]\!]$. The inclusion "$\supseteq$" is clear.

Let $t \in T_\Gamma$, $r \in \mathrm{Run}_{\mathcal{B}}(t)$, and let $O = \{i \in \{1, \ldots, n\} \mid \text{there is a position } w \in \mathrm{pos}(t) \text{ with } \tilde{\pi}_1 \circ r(w) = (p_i, P_i)\}$. Let $i \in O$ and assume we have two positions $v_1, v_2 \in \mathrm{pos}(t)$ such that $\pi_2 \circ \tilde{\pi}_1 \circ r(v_1) = \pi_2 \circ \tilde{\pi}_1 \circ r(v_2) = P_i$. Then, since $\tilde{\pi}_1(r) \in \mathrm{Run}_{\mathcal{S}}(t)$ by Lemma 22(iv), we obtain by Theorem 19 that $v_1 \leq_p v_2$ or $v_2 \leq_p v_1$. We can therefore let $w_i \in \mathrm{pos}(t)$ be the prefix-largest position in t with $\tilde{\pi}_1 \circ r(w_i) \in R_i$. Then from Lemma 24(ii) and Lemma 24(iii), we obtain that for all positions $v \in \mathrm{pos}(t)$ with $v \leq_p w_i$ we have $\tilde{\pi}_2 \circ r(v)[i] = \tilde{\pi}_2 \circ r(w_i)[i] \in \{1, \ldots, |Q|\} \times \{\tilde{\pi}_1 \circ r(w_i)\}$, and for all other positions $v \in \mathrm{pos}(t)$ we have $\tilde{\pi}_2 \circ r(v)[i] = 0$.

We define a tuple $\bar{c} \in I$ as follows. If $i \in O$, we let $\bar{c}[i] = \tilde{\pi}_2 \circ r(w_i)[i]$, where w_i is defined as above. If $i \notin O$, we let $\bar{c}[i] = |Q| + 1$. Then we have $r \in \mathrm{Run}_{\mathcal{B}_{\bar{c}}}(t)$. Thus, $\mathrm{Run}_{\mathcal{B}}(t) = \bigcup_{\bar{c} \in I} \mathrm{Run}_{\mathcal{B}_{\bar{c}}}(t)$. ☐

We now obtain a finitely sequential representation of $\mathcal{A}$ by applying Theorem 3 to the automata $\mathcal{B}_{\bar{c}}$. In particular, we see that the behavior of a trim unambiguous max-plus-WTA is finitely sequential if it does not satisfy the tree fork property. This concludes the proof of Theorem 9.

4 Further Insights

In this section, we show some additional properties of the rivals of the Schützenberger covering $\mathcal{S}$. These properties are not necessary for the proof of Theorem 9, but they do give a better idea of the limits of interaction there may exist between the rivals of $\mathcal{S}$. If two rivals (p, P) and (q, P) of $\mathcal{S}$ cannot occur together in the same run, they can be trivially separated like in Lemma 15. Therefore, in the following we only consider the case that $(p, P) \leq (q, P)$. Under this assumption, the first property we show is that Theorem 19 is true even if we replace (p, P) by (q, P) in (i), i.e., we have the following theorem.

Theorem 28. *Let* $(p, P), (q, P) \in Q_{\mathcal{S}}$ *be rivals in* $\mathcal{S}$ *with* $p \leq q$*. Then for every tree* $t \in T_\Gamma$ *and every run* $r \in \mathrm{Run}_{\mathcal{S}}(t)$ *of* $\mathcal{S}$ *on* t*, at least one of the following two conditions holds.*

(i) The state (q, P) *does not occur in* r*, i.e.,* $r(w) \neq (q, P)$ *for all* $w \in \mathrm{pos}(t)$*.*

(ii) All states with second entry P *occur linearly in* r*, i.e., for all* $w_1, w_2 \in \mathrm{pos}(t)$ *with* $\pi_2 \circ r(w_1) = \pi_2 \circ r(w_2) = P$ *we have* $w_1 \leq_p w_2$ *or* $w_2 \leq_p w_1$*.*

Theorem 28 follows from Theorem 19 by applying Lemma 29 below. In short, Lemma 29 tells us that from $(p, P) \leq (q, P)$, it follows that there is a rival (q', P) of (q, P) with $(q, P) \leq (q', P)$. Thus, Theorem 28 follows by applying Theorem 19 to the rivals (q, P) and (q', P). Together, Theorems 19 and 28 tell us that whenever we have two rivals (p, P) and (q, P) with $(p, P) \leq (q, P)$ in $\mathcal{S}$, then whenver one of $(p, P), (q, P)$ occurs in a run on a tree t, then for every run on that tree all states with second entry P occur along a distinguished branch of t. We prove Lemma 29.

Lemma 29. *Let* $(p, P), (q, P) \in Q_{\mathcal{S}}$ *be rivals in* $\mathcal{S}$ *with witnesses* u *and* s *such that* $(p, P) \leq (q, P)$*. Then there exists a state* $q' \in P$ *with* $(q, P) \leq (q', P)$ *such that* (q', P) *and* (q, P) *are rivals in* $\mathcal{S}$ *with witnesses* u *and* s^n *for some* $n \geq 1$*.*

Proof. Let (p,P), (q,P), u, and s be as in the statement of the lemma. Furthermore, let s' be a Γ-word such that $\mathrm{Run}^{\diamond}_{\mathcal{S}}((q,P),s',(p,P)) \neq \emptyset$, which exists due to $(p,P) \leq (q,P)$

By construction of $\mathcal{S}$ and our assumption on s', there exists for every $p' \in P$ at least one $q' \in P$ with $\mathrm{Run}^{\diamond}_{\mathcal{S}}((q',P),s',(p',P)) \neq \emptyset$. On the other hand, we obtain from the unambiguity of $\mathcal{S}$ that for every $p' \in P$, there can be at most one $q' \in P$ with $\mathrm{Run}^{\diamond}_{\mathcal{S}}((q',P),s',(p',P)) \neq \emptyset$. It follows that s' induces a mapping $g\colon P \to P$ which maps p' to q' if $\mathrm{Run}^{\diamond}_{\mathcal{S}}((q',P),s',(p',P)) \neq \emptyset$ and g satisfies $p \mapsto q$. With an identical argumentation, we obtain that s induces a mapping $h\colon P \to P$ which maps p' to q' if $\mathrm{Run}^{\diamond}_{\mathcal{S}}((q',P),s,(p',P)) \neq \emptyset$ and h satisfies $p \mapsto p$ and $q \mapsto q$.

Let R' be the smallest set satisfying $q \in R'$, $g(R') \subseteq R'$, and $h(R') \subseteq R'$, i.e., such that R' contains q and is closed under g and h. Then let $R = \{p\} \cup R'$ and let $p_1, \dots, p_m$ be an enumeration of R. By pigeon hole principle, there exist integers $0 \leq n_1 < n_2$ such that $(h^{n_1}(g(p_1)), \dots, h^{n_1}(g(p_m))) = (h^{n_2}(g(p_1)), \dots, h^{n_2}(g(p_m)))$. By definition of R and the fact that $g(p) = q$, we have $h^{n_1}(g(p_1)), \dots, h^{n_1}(g(p_m)) \in R$. Thus, $f = h^{n_1} \circ g$ is a mapping $f\colon R \to R$ and we can apply Lemma 16 to f with $a = p$. We obtain $q' \in R$ and $n \geq 1$ with $f^n(p) = q' = f^n(q')$. From the definitions of h and g, it follows that $(s'(s^{n_1}))^n$ is a (q',P)-(p,P)-fork, and therefore also a q'-p-fork. Since $\mathcal{A}$ does not satisfy the tree fork property, q' is therefore not a rival of p. Moreover, we have $f^{n-1} \circ h^{n_1}(q) = f^n(p) = q'$, so we see that $\mathrm{Run}^{\diamond}_{\mathcal{S}}((q',P),s^{n_1}((s'(s^{n_1}))^{n-1}),(q,P)) \neq \emptyset$ and therefore $(q,P) \leq (q',P)$.

Due to the fact that $h^{n_2-n_1}(q') = q'$, we see from the definition of h that $(q',P) \xrightarrow{s^{n_2-n_1}|z} (q',P)$ for some $z \in \mathbb{R}$. Let $k = n_2 - n_1$. We know that $(p,P) \xrightarrow{s|x} (p,P)$ and $(q,P) \xrightarrow{s|y} (q,P)$ for some $x,y \in \mathbb{R}$ with $x \neq y$. It follows that $(p,P) \xrightarrow{s^k|kx} (p,P)$ and $(q,P) \xrightarrow{s^k|ky} (q,P)$, so (q',P) is either a rival of (p,P) or of (q,P) with witnesses u and s^k. As (q',P) and (p,P) being rivals implies by Lemma 14(vii) that q' and p are rivals in $\mathcal{A}$ and we found that the latter is not the case, it must hold that (q',P) and (q,P) are rivals. □

Now assume that (p,P) and $(q,P) \in Q_{\mathcal{S}}$ with $(p,P) \leq (q,P)$ are rivals in $\mathcal{S}$ with witnesses u and s. Furthermore, assume (p_1,P) and (p_2,P) may occur at prefix-independent positions, i.e., there is a tree $t \in T_\Gamma$, prefix-independent positions $w_1, w_2 \in \mathrm{pos}(t)$, and a run $r \in \mathrm{Run}_{\mathcal{S}}(t)$ with $r(w_1) = (p_1,P)$ and $r(w_2) = (p_2,P)$. In the following, we want to analyze how (p_1,P) may occur together with (p,P) or (q,P) in a run.

First, Theorems 19 and 28 tell us that (p_1,P) may not occur prefix-independently from (p,P) and (q,P). Second, if $(p_1,P) \leq (p,P)$ or $(p_1,P) \leq (q,P)$, then by our assumption the state (p_2,P) may occur prefix-independently from (p,P) or (q,P). This is impossible again by Theorems 19 and 28, so $(p_1,P) \leq (p,P)$ and $(p_1,P) \leq (q,P)$ both cannot hold. Third, we know that $\mathrm{Run}_{\mathcal{S}}(s^{|P|}(u),(p_1,P)) \neq \emptyset$, from which follows that there is some $\hat{p} \in P$ and an integer $k \geq 1$ with $(p_1,P) \leq (\hat{p},P)$ and $\mathrm{Run}^{\diamond}_{\mathcal{S}}((\hat{p},P),s^k,(\hat{p},P)) \neq \emptyset$. Thus, $(\hat{p},P)$ is a rival of either (p,P) or (q,P) and by our assumption, it may occur prefix-independently from (p_2,P). Then if $(q,P) \leq (p_1,P)$ held, we would have $(p,P) \leq (q,P) \leq (p_1,P) \leq (\hat{p},P)$. By applying Theorem 28 either to (p,P) and $(\hat{p},P)$ or to (q,P) and $(\hat{p},P)$, we see that (p_2,P) may not occur prefix-independently from $(\hat{p},P)$, which does not match our assumption. It follows that $(q,P) \leq (p_1,P)$ is also impossible. Finally, the only remaining possibility is $(p,P) \leq (p_1,P)$. This is in fact possible, as shown in Example 20, where we have $(p'',P) \leq (q,P)$, $(p'',P) \leq (p',P)$, and (p',P) may occur at prefix-independent positions.

In conclusion, we obtain that if (p,P) and (q,P) are rivals in $\mathcal{S}$ with $(p,P) \leq (q,P)$, then all states with second entry P which can occur at prefix-independent positions can be trivially separated from (q,P) as they may never occur in the same run. We decided not to employ this fact in Section 3.2 as doing so does not lead to a shorter proof or a significantly simpler construction.

References

[1] A. Alexandrakis and S. Bozapalidis. Weighted grammars and Kleene's theorem. *Information Processing Letters*, 24(1):1–4, 1987.

[2] C. Allauzen and M. Mohri. Efficient algorithms for testing the twins property. *Journal of Automata, Languages and Combinatorics*, 8(2):117–144, 2003.

[3] S. Bala. Which finitely ambiguous automata recognize finitely sequential functions? In K. Chatterjee and J. 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] S. Bala and A. Koniński. Unambiguous automata denoting finitely sequential functions. In A. Dediu, C. Martín-Vide, and B. 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] M. Béal, O. Carton, C. Prieur, and J. Sakarovitch. Squaring transducers: an efficient procedure for deciding functionality and sequentiality. *Theoretical Computer Science*, 292(1):45–63, 2003.

[6] J. Berstel and C. Reutenauer. Recognizable formal power series on trees. *Theoretical Computer Science*, 18:115–148, 1982.

[7] J. Berstel and C. Reutenauer. *Rational Series and Their Languages*, volume 12 of *EATCS Monographs in Theoretical Computer Science*. Springer, 1988.

[8] J. Björklund, F. Drewes, and N. Zechner. An efficient best-trees algorithm for weighted tree automata over the tropical semiring. In A. Dediu, E. Formenti, C. Martín-Vide, and B. 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.

[9] M. Blattner and T. Head. Automata that recognize intersections of free submonoids. *Information and Control*, 35(3):173–176, 1977.

[10] M. Büchse and A. Fischer. Deciding the twins property for weighted tree automata over extremal semifields. In F. Drewes and M. Kuhlmann, editors, *2nd Workshop on Applications of Tree Automata Techniques in Natural Language Processing (ATANLP)*, pages 11–20. The Association for Computer Linguistics, 2012.

[11] M. Büchse, J. May, and H. Vogler. Determinization of weighted tree automata using factorizations. *Journal of Automata, Languages and Combinatorics*, 15(3/4):229–254, 2010.

[12] L. Daviaud, P. Guillon, and G. Merlet. Comparison of max-plus automata and joint spectral radius of tropical matrices. In Larsen et al. [27], pages 19:1–19:14.

[13] M. Droste, W. Kuich, and H. Vogler, editors. *Handbook of Weighted Automata*. EATCS Monographs in Theoretical Computer Science. Springer, 2009.

[14] Z. Ésik and W. Kuich. Formal tree series. *Journal of Automata, Languages and Combinatorics*, 8(2):219–285, 2003.

[15] E. Filiot, I. Jecker, N. Lhote, G. A. Pérez, and J. 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.

[16] Z. Fülöp and H. Vogler. Weighted tree automata and tree transducers. In Droste et al. [13], chapter 9, pages 313–403.

[17] K. Hashiguchi. Algorithms for determining relative star height and star height. *Information and Computation*, 78(2):124–169, 1988.

[18] K. Hashiguchi, K. Ishiguro, and S. Jimbo. Decidability of the equivalence problem for finitely ambiguous finance automata. *International Journal of Algebra and Computation*, 12(3):445–461, 2002.

[19] D. 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.

[20] D. Kirsten. Decidability, undecidability, and PSPACE-completeness of the twins property in the tropical semiring. *Theoretical Computer Science*, 420:56–63, 2012.

[21] D. Kirsten and S. Lombardy. Deciding unambiguity and sequentiality of polynomially ambiguous min-plus automata. In S. Albers and J. 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] I. Klimann, S. Lombardy, J. Mairesse, and C. Prieur. Deciding unambiguity and sequentiality from a finitely ambiguous max-plus automaton. *Theoretical Computer Science*, 327(3):349–373, 2004.

[23] J. Komenda, S. Lahaye, J. Boimond, and T. van den Boom. Max-plus algebra in the history of discrete event systems. *Annual Reviews in Control*, 45:240–249, 2018.

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

[25] D. 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.

[26] W. Kuich and A. Salomaa. *Semirings, Automata, Languages*, volume 5 of *EATCS Monographs in Theoretical Computer Science*. Springer, 1986.

[27] K. G. Larsen, H. L. Bodlaender, and J. Raskin, editors. *42nd International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 83 of *Leibniz International Proceedings in Informatics (LIPIcs)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017.

[28] F. Mazowiecki and C. Riveros. Pumping lemmas for weighted automata. In R. Niedermeier and B. 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] M. Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–311, 1997.

[30] M. Mohri. Weighted automata algorithms. In Droste et al. [13], chapter 6, pages 213–254.

[31] E. Paul. The equivalence, unambiguity and sequentiality problems of finitely ambiguous max-plus tree automata are decidable. In Larsen et al. [27], pages 53:1–53:13.

[32] E. Paul. Finite sequentiality of unambiguous max-plus tree automata. In R. Niedermeier and C. 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.

[33] S. 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.

[34] M. O. Rabin and D. S. Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3(2):114–125, 1959.

[35] J. Sakarovitch. A construction on finite automata that has remained hidden. *Theoretical Computer Science*, 204(1-2):205–231, 1998.

[36] J. Sakarovitch. *Elements of Automata Theory.* Cambridge University Press, 2009.

[37] A. Salomaa and M. Soittola. *Automata-Theoretic Aspects of Formal Power Series.* Texts and Monographs in Computer Science. Springer, 1978.

[38] M.-P. Schützenberger. On the definition of a family of automata. *Information and Control*, 4(2-3):245–270, 1961.

[39] M.-P. Schützenberger. Sur les relations rationnelles entre monoïdes libres. *Theoretical Computer Science*, 3(2):243–259, 1976.

[40] H. Seidl. On the finite degree of ambiguity of finite tree automata. *Acta Informatica*, 26(6):527–542, 1989.

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

[42] I. Simon. Recognizable sets with multiplicities in the tropical semiring. In M. P. Chytil, L. Janiga, and V. 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.

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

[44] A. Weber and H. Seidl. On the degree of ambiguity of finite automata. *Theoretical Computer Science*, 88(2):325–349, 1991.