

The Equivalence, Unambiguity and Sequentiality Problems of Finitely Ambiguous Max-Plus Tree Automata are Decidable*

Erik Paul¹

¹ Institute of Computer Science, Leipzig University, 04109 Leipzig, Germany
{epaul}@informatik.uni-leipzig.de

Abstract

We show that the equivalence, unambiguity and sequentiality problems are decidable for finitely ambiguous max-plus tree automata.

1998 ACM Subject Classification F.4.3 Formal Languages

Keywords and phrases Tree Automata, Max-Plus Automata, Equivalence, Unambiguity, Sequentiality, Decidability

Digital Object Identifier 10.4230/LIPIcs.MFCS.2017.53

1 Introduction

A max-plus automaton is a finite automaton with transition weights in the real numbers. To each word, it assigns the maximum weight of all accepting paths on the word, where the weight of a path is the sum of the path's transition weights. Max-plus automata and their min-plus counterparts are weighted automata [19, 18, 13, 2, 4] over the max-plus or min-plus semiring. Under varying names, max-plus and min-plus automata have been studied and employed many times in the literature. They can be used to determine the star height of a language [7], to decide the finite power property [20, 21] and to model certain timed discrete event systems [5, 6]. Additionally, they appear in the context of natural language processing [14].

For practical applications, the decidable properties of an automaton model are usually of great interest. Typical problems considered include the emptiness, universality, inclusion, equivalence, sequentiality and unambiguity problems. We consider the last three of these problems for *finitely ambiguous* automata, which are automata in which the number of accepting paths for every word is bounded by a global constant. If there is at most one accepting path for every word, the automaton is called *unambiguous*. It is called *deterministic* or *sequential* if for each pair of a state and an input symbol, there is at most one valid transition into a next state. It is known [11] that finitely ambiguous max-plus automata are strictly more expressive than unambiguous max-plus automata, which in turn are strictly more expressive than deterministic max-plus automata.

Let us quickly recall the considered problems and the related results. The equivalence problem asks whether two automata are *equivalent*, which is the case if the weights assigned by them coincide on all words. In general, the equivalence problem is undecidable [12] for max-plus automata, but for finitely ambiguous max-plus automata it becomes decidable

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



© Erik Paul;
licensed under Creative Commons License CC-BY

42nd International Symposium on Mathematical Foundations of Computer Science (MFCS 2017).

Editors: Kim G. Larsen, Hans L. Bodlaender, and Jean-Francois Raskin; Article No. 53; pp. 53:1–53:13

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

[22, 9]. The sequentiality problem asks whether for a given automaton, there exists an equivalent deterministic automaton. This was shown to be decidable by Mohri [14] for unambiguous max-plus automata. Finally, the unambiguity problem asks whether for a given automaton, there exists an equivalent unambiguous automaton. This problem is known to be decidable for finitely ambiguous and even *polynomially ambiguous* max-plus automata [11, 10]. In conjunction with Mohri's results, it follows that the sequentiality problem is decidable for these classes of automata as well.

In this paper, we show that these three problems are decidable for finitely ambiguous max-plus tree automata, which are max-plus automata that operate on trees instead of words. In the form of *probabilistic context-free grammars*, max-plus tree automata are commonly employed in natural language processing [17]. Our approach to the decidability of the equivalence problem uses ideas from [9]. We use a similar induction argument and also reduce the equivalence problem to the same decidable problem, namely the decidability of the existence of an integer solution for a system of linear inequalities [15]. On words, the proof relies on the decomposition of words into subwords of bounded length, of which one is removed in the induction step. This argument cannot be applied to trees as easily. A tree can be decomposed into *contexts* of bounded height, but this requires contexts with multiple variables. Removing such a context does usually not yield a tree. Consequently, our induction is much more involved. We also point out and correct an important oversight in the main theorem of [9].

The decidability of the unambiguity problem employs ideas from [11]. Here, we show how the *dominance property* can be generalized to max-plus tree automata. To show the decidability of the sequentiality problem, we first combine results from [3] and [14] to show the decidability of this problem for unambiguous max-plus tree automata, and then combine this result with the decidability of the unambiguity problem.

Our solution of the equivalence problem can be applied to weighted logics. In [16], a fragment of a weighted logic is shown to have the same expressive power as finitely ambiguous weighted tree automata. Over the max-plus semiring, equivalence is decidable for formulas of this fragment due to our results.

2 Preliminaries

Let $\mathbb{N} = \{0, 1, 2, \dots\}$. By \mathbb{N}^* we denote the set of all finite words over \mathbb{N} . The empty word is denoted by ε , and the length of a word $w \in \mathbb{N}^*$ by $|w|$. The set \mathbb{N}^* is partially ordered by the prefix relation \leq_p and totally ordered with respect to the lexicographic ordering \leq_l . A *ranked alphabet* is a pair $(\Gamma, \text{rk}_\Gamma)$, often abbreviated by Γ , where Γ is a finite set and $\text{rk}_\Gamma: \Gamma \rightarrow \mathbb{N}$. For every $m \geq 0$ we define $\Gamma^{(m)} = \text{rk}_\Gamma^{-1}(m)$ as the set of all symbols of rank m . The rank $\text{rk}(\Gamma)$ of Γ is defined as $\max\{\text{rk}_\Gamma(a) \mid a \in \Gamma\}$.

The set of (*finite, labeled and ordered*) Γ -trees, denoted by T_Γ , is the set of all pairs $t = (\text{pos}(t), \text{label}_t)$, where $\text{pos}(t) \subset \mathbb{N}^*$ is a finite non-empty prefix-closed set, $\text{label}_t: \text{pos}(t) \rightarrow \Gamma$ is a mapping and for every $w \in \text{pos}(t)$ we have $w_i \in \text{pos}(t)$ iff $1 \leq i \leq \text{rk}_\Gamma(\text{label}_t(w))$. We write $t(w)$ for $\text{label}_t(w)$. We also refer to the elements of $\text{pos}(t)$ as *nodes*, to ε as the *root* of t and to prefix-maximal nodes as *leaves*.

Now let $s, t \in T_\Gamma$ and $w \in \text{pos}(t)$. The *subtree of t at w* , denoted by $t|_w$, is a Γ -tree defined as follows. We let $\text{pos}(t|_w) = \{v \in \mathbb{N}^* \mid wv \in \text{pos}(t)\}$ and for $v \in \text{pos}(t|_w)$, $\text{label}_{t|_w}(v) = t(wv)$. The *substitution of s into w of t* , denoted by $t\langle s \rightarrow w \rangle$, is a Γ -tree defined as follows. We let $\text{pos}(t\langle s \rightarrow w \rangle) = \{v \in \text{pos}(t) \mid w \not\leq_p v\} \cup \{wv \mid v \in \text{pos}(s)\}$. For $u \in \text{pos}(t\langle s \rightarrow w \rangle)$, we let $\text{label}_{t\langle s \rightarrow w \rangle}(u) = s(v)$ if $u = wv$, and otherwise $\text{label}_{t\langle s \rightarrow w \rangle}(u) = t(u)$.

For $a \in \Gamma^{(m)}$ and trees $t_1, \dots, t_m \in T_\Gamma$, we also write $a(t_1, \dots, t_m)$ to denote the tree t with $\text{pos}(t) = \{\varepsilon\} \cup \{iw \mid i \in \{1, \dots, m\}, w \in \text{pos}(t_i)\}$, $\text{label}_t(\varepsilon) = a$ and $\text{label}_t(iw) = t_i(w)$.

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

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

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

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

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

For $t \in T_\Gamma$, a mapping $r: \text{pos}(t) \rightarrow Q$ is called a *quasi-run of \mathcal{A} on t* . For a quasi-run r on t and $w \in \text{pos}(t)$ with $t(w) = a \in \Gamma^{(m)}$, the tuple $\mathfrak{t}(t, r, w) = (r(w_1), \dots, r(w_m), a, r(w))$ is called the *transition at w* . The quasi-run r is called a *(valid) run* if for every $w \in \text{pos}(t)$ the transition $\mathfrak{t}(t, r, w)$ is valid with respect to \mathcal{A} . We call a run r *accepting* if $r(\varepsilon)$ is final. By $\text{Run}_{\mathcal{A}}(t)$ and $\text{Acc}_{\mathcal{A}}(t)$ we denote the sets of all runs and all accepting runs of \mathcal{A} on t , respectively. For $r \in \text{Run}_{\mathcal{A}}(t)$ the *weight of r* is defined by $\text{wt}_{\mathcal{A}}(t, r) = \odot_{w \in \text{pos}(t)} \mu(\mathfrak{t}(t, r, w))$. The *tree series accepted by \mathcal{A}* , denoted by $\llbracket \mathcal{A} \rrbracket \in K\langle\langle T_\Gamma \rangle\rangle$, is the tree series defined for every $t \in T_\Gamma$ by $\llbracket \mathcal{A} \rrbracket(t) = \bigoplus_{r \in \text{Acc}_{\mathcal{A}}(t)} \text{wt}_{\mathcal{A}}(t, r) \odot \nu(r(\varepsilon))$ where the sum over the empty set is $\mathbb{0}$ by convention. The *support of \mathcal{A}* is the set $\text{supp}(\mathcal{A}) = \{t \in T_\Gamma \mid \llbracket \mathcal{A} \rrbracket(t) \neq \mathbb{0}\}$.

The support of an FTA \mathcal{A} is also called the *language accepted by \mathcal{A}* and denoted by $\mathcal{L}(\mathcal{A})$. A subset $L \subseteq T_\Gamma$ is called *recognizable* if there exists an FTA \mathcal{A} with $L = \mathcal{L}(\mathcal{A})$.

A WTA \mathcal{A} is called *deterministic* if for every $m \geq 0$, $a \in \Gamma^{(m)}$ and $\vec{p} \in Q^m$ there exists at most one $q \in Q$ with $\mu(\vec{p}, a, q) \neq \mathbb{0}$. We call \mathcal{A} *finitely ambiguous* or *M -ambiguous* if $|\text{Acc}_{\mathcal{A}}(t)| \leq M$ for some $M \geq 1$ and every $t \in T_\Gamma$. A 1-ambiguous WTA is also called *unambiguous*. We recall that for every recognizable language $L \subseteq T_\Gamma$, there exists a deterministic FTA \mathcal{A} with $\mathcal{L}(\mathcal{A}) = L$.

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

3 The Equivalence Problem

For two max-plus-WTA \mathcal{A}_1 and \mathcal{A}_2 over an alphabet Γ , we say that \mathcal{A}_1 *dominates* \mathcal{A}_2 , denoted by $\mathcal{A}_1 \geq \mathcal{A}_2$, if for all trees $t \in T_\Gamma$ we have $\llbracket \mathcal{A}_1 \rrbracket(t) \geq \llbracket \mathcal{A}_2 \rrbracket(t)$. We say that \mathcal{A}_1 and \mathcal{A}_2 are *equivalent*, denoted by $\mathcal{A}_1 = \mathcal{A}_2$, if for all $t \in T_\Gamma$ we have $\llbracket \mathcal{A}_1 \rrbracket(t) = \llbracket \mathcal{A}_2 \rrbracket(t)$.

The equivalence problem for max-plus (tree) automata asks whether for two given max-plus (tree) automata \mathcal{A}_1 and \mathcal{A}_2 , it holds that $\mathcal{A}_1 = \mathcal{A}_2$. For words, this problem was shown to be undecidable in general [12], but it is decidable if both automata are finitely ambiguous [9]. In this section, we prove that the equivalence problem is decidable for finitely ambiguous max-plus-WTA. This section is based on ideas from [9].

► **Theorem 1.** *The equivalence problem for finitely ambiguous weighted tree automata over the max-plus semiring is decidable.*

In fact, we will show that if \mathcal{A}_1 is a finitely ambiguous max-plus-WTA and \mathcal{A}_2 any max-plus-WTA, then it is decidable whether \mathcal{A}_1 dominates \mathcal{A}_2 .

► **Theorem 2.** *Let \mathcal{A}_1 be a finitely ambiguous max-plus-WTA and \mathcal{A}_2 any max-plus-WTA. It is decidable whether or not $\mathcal{A}_1 \geq \mathcal{A}_2$.*

If both automata in Theorem 2 are finitely ambiguous, we can reverse their roles. Consequently, Theorem 1 is a corollary of Theorem 2. The remainder of this section is dedicated to the proof of Theorem 2.

As a first step, we show in the following lemma that every finitely ambiguous max-plus-WTA \mathcal{A} can be “normalized” such that all trees, which have an accepting run in \mathcal{A} , have the same number of accepting runs.

► **Lemma 3.** *Let $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ be an M -ambiguous max-plus-WTA. Then there exists a finitely ambiguous max-plus-WTA \mathcal{A}' with $\mathcal{A} = \mathcal{A}'$ and $|\text{Acc}_{\mathcal{A}'}(t)| \in \{0, M\}$ for all $t \in T_\Gamma$.*

For the rest of this section, fix an M -ambiguous max-plus-WTA \mathcal{A}_1 and a max-plus-WTA \mathcal{A}_2 . By Lemma 3, we can assume that for all $t \in T_\Gamma$ we have $|\text{Acc}_{\mathcal{A}_1}(t)| \in \{0, M\}$. Note that $\mathcal{A}_1 \geq \mathcal{A}_2$ can only hold if $\text{supp}(\mathcal{A}_2) \subseteq \text{supp}(\mathcal{A}_1)$, which is decidable since the supports of \mathcal{A}_1 and \mathcal{A}_2 are recognizable languages. Therefore, in the forthcoming considerations we will always assume that $\text{supp}(\mathcal{A}_2) \subseteq \text{supp}(\mathcal{A}_1)$ holds. We write $\mathcal{A}_i = (Q_i, \Gamma, \mu_i, \nu_i)$ for $i = 1, 2$.

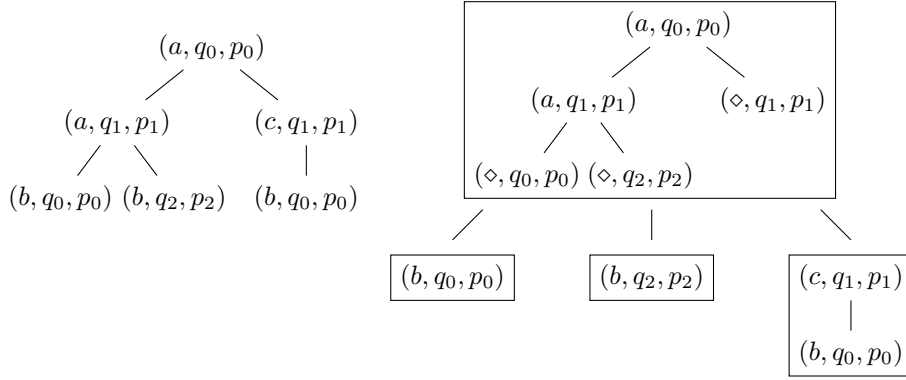
For any tree in $\text{supp}(\mathcal{A}_2)$, there are exactly M accepting runs of \mathcal{A}_1 on this tree. We want to apply pumping type arguments to all of these runs and a given accepting run of \mathcal{A}_2 simultaneously. For this, we encode the runs of \mathcal{A}_1 and the given run of \mathcal{A}_2 directly into the tree. Moreover, we want to decompose these trees, with all runs encoded, into smaller parts. Formally, such a decomposition will be a tree of trees. To mark where these smaller trees connect to each other, we use the new label \diamond .

► **Definition 4.** For a set X and an alphabet Σ , we define for $(a, x) \in \Sigma \times X$ the rank $\text{rk}_{\Sigma \times X}(a, x) = \text{rk}_\Sigma(a)$. We let $\Gamma_\diamond = (\Gamma \cup \{\diamond\}, \text{rk}_\Gamma \cup \{\diamond \mapsto 0\})$, where \diamond is a new symbol. Let $\Omega = Q_1^M \times Q_2$ and let π_Ω, π_Γ and π_{Γ_\diamond} be the projections of $\Gamma \times \Omega$ and $\Gamma_\diamond \times \Omega$ onto Ω, Γ and Γ_\diamond , respectively.

For a tree $t \in T_{\Gamma_\diamond \times \Omega}$, we define $\text{label}_t^{\Gamma_\diamond} = \pi_{\Gamma_\diamond} \circ \text{label}_t$ and $\text{label}_t^i = \pi_i \circ \pi_\Omega \circ \text{label}_t$ where π_i is the i -th projection on Ω . For $i \in \{1, \dots, M+1\}$, we define

$$\text{wt}_i(t) = \begin{cases} \sum_{\substack{w \in \text{pos}(t) \\ \text{label}_t^{\Gamma_\diamond}(w) \neq \diamond}} \mu_1(\mathfrak{t}((\text{pos}(t), \text{label}_t^{\Gamma_\diamond}), \text{label}_t^i, w)) & \text{if } 1 \leq i \leq M \\ \sum_{\substack{w \in \text{pos}(t) \\ \text{label}_t^{\Gamma_\diamond}(w) \neq \diamond}} \mu_2(\mathfrak{t}((\text{pos}(t), \text{label}_t^{\Gamma_\diamond}), \text{label}_t^i, w)) & \text{if } i = M+1. \end{cases}$$

A tree $t \in T_{\Gamma_\diamond \times \Omega}$ is called *accepting* if $\text{label}_t^1, \dots, \text{label}_t^M$ are pairwise distinct accepting runs of \mathcal{A}_1 on $(\text{pos}(t), \text{label}_t^{\Gamma_\diamond})$ and label_t^{M+1} is an accepting run of \mathcal{A}_2 on $(\text{pos}(t), \text{label}_t^{\Gamma_\diamond})$.



■ **Figure 1** A tree in $T_{\Gamma \times \Omega}$ ($M = 1$) and a possible cycle decomposition with f defined as $f(\varepsilon) = 1$ and $f(1) = f(2) = f(3) = 0$.

Let $t \in T_{\Gamma \times \Omega}$, $n = |\{w \in \text{pos}(t) \mid \text{label}_t^{\Gamma \diamond}(w) = \diamond\}|$ be the number of \diamond -leaves in t and $\{w_1, \dots, w_n\}$ a lexicographically ordered enumeration of these leaves, i.e. $w_1 \leq_l \dots \leq_l w_n$. For $1 \leq i \leq n$, we define $W_i(t) = w_i$. The set

$$\Xi = \{t \in T_{\Gamma \times \Omega} \mid \forall w \in \text{pos}(t) : |w| \leq |\Omega| \text{ and } \text{label}_t^{\Gamma \diamond}(\varepsilon) \neq \diamond\}$$

forms a ranked alphabet where for t as above we define $\text{rk}_{\Xi}(t) = n$. We recall that $|w|$ denotes the length of w and $|\Omega|$ the cardinality of Ω .

Every tree over Ξ corresponds to a unique tree over $\Gamma \times \Omega$. This inclusion $\mathcal{J}: T_{\Xi} \hookrightarrow T_{\Gamma \times \Omega}$ is formally given as follows. For $\mathbf{t} \in T_{\Xi}$ with $\text{pos}(\mathbf{t}) = \{\varepsilon\}$ we let $\mathcal{J}(\mathbf{t}) = \mathbf{t}(\varepsilon)$. Otherwise let $m = \text{rk}_{\Xi}(\mathbf{t}(\varepsilon))$ and $\mathcal{J}(\mathbf{t}) = \mathbf{t}(\varepsilon) \langle \mathcal{J}(\mathbf{t}|_1) \rightarrow W_1(\mathbf{t}(\varepsilon)) \rangle \dots \langle \mathcal{J}(\mathbf{t}|_m) \rightarrow W_m(\mathbf{t}(\varepsilon)) \rangle$.

A tree $\mathbf{t} \in T_{\Xi}$ is called *matching* if for all $w \in \text{pos}(\mathbf{t})$ and $i \in \{1, \dots, \text{rk}_{\Xi}(\mathbf{t}(w))\}$ we have $\pi_{\Omega}(\mathbf{t}(w)(W_i(\mathbf{t}(w)))) = \pi_{\Omega}(\mathbf{t}(w_i)(\varepsilon))$, i.e. the “state labels” of the \diamond -letters match with the state labels of the root at the corresponding child.

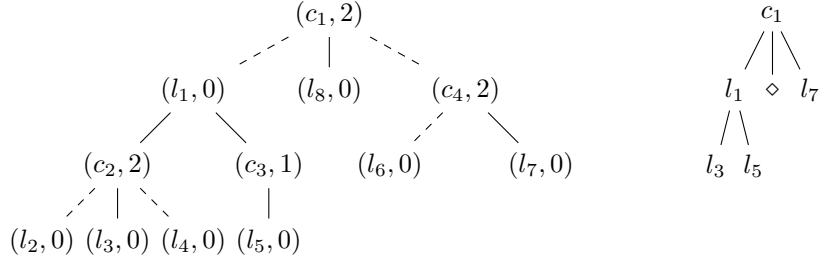
► **Definition 5.** Let $t \in T_{\Gamma \times \Omega}$. A *cycle decomposition* of t is a pair $\mathfrak{D} = (\mathbf{t}, f)$, where $\mathbf{t} \in T_{\Xi}$ and $f: \text{pos}(\mathbf{t}) \rightarrow \mathbb{N}$ is a mapping, satisfying the following.

1. We have $\mathcal{J}(\mathbf{t}) = t$ and \mathbf{t} is matching.
2. For all $w \in \text{pos}(\mathbf{t})$ we have $f(w) \leq \text{rk}_{\Xi}(\mathbf{t}(w))$.
3. If $w \in \text{pos}(\mathbf{t})$ with $f(w) > 0$, then with $v = W_{f(w)}(\mathbf{t}(w))$ we have $\pi_{\Omega}(\mathbf{t}(w)(\varepsilon)) = \pi_{\Omega}(\mathbf{t}(w)(v))$. In other words, the state labels of the root of $\mathbf{t}(w)$ coincide with the state labels of the $f(w)$ -th \diamond -leaf of $\mathbf{t}(w)$.
4. If $f(w) = 0$ for some $w \in \text{pos}(\mathbf{t})$, then for all $i > 0$ with $w_i \in \text{pos}(\mathbf{t})$, we have $f(w_i) > 0$.

We call $w \in \text{pos}(\mathbf{t})$ a *cycle* if $f(w) > 0$ and otherwise a *link*. The set of all cycles of \mathfrak{D} is denoted by $\text{Cyc}(\mathfrak{D})$. By $\text{Decomp}(t)$ we denote the set of all cycle decompositions of t . For $w \in \text{pos}(\mathbf{t})$ we call the set $\text{Anc}_{\mathfrak{D}}(w) = \{v \in \text{Cyc}(\mathfrak{D}) \mid v_i \leq_p w \text{ for some } i \geq 1 \text{ with } i \neq f(v)\}$ the *ancestors* of w , see also Figure 2. We have the following lemma.

► **Lemma 6.** For every $t \in T_{\Gamma \times \Omega}$, there exists a cycle decomposition $(\mathbf{t}, f) \in \text{Decomp}(t)$.

For $w \in \text{Cyc}(\mathfrak{D})$ we now define a new tree $\mathbf{t}_{\mathfrak{D}}(w)$ over the alphabet $\Xi \cup \{\diamond\}$, where \diamond has rank 0. Intuitively, $\mathbf{t}_{\mathfrak{D}}(w)$ is the subtree of \mathbf{t} at w , with all cycles apart from w itself removed and the subtree at $wf(w)$ replaced by \diamond , see also Figure 2.



■ **Figure 2** A cycle decomposition \mathfrak{D} with f included in the labels and the tree $\mathfrak{t}_{\mathfrak{D}}(\varepsilon)$. A c denotes a cycle and an l a link. The position of l_8 has no ancestors, l_5 has only the ancestor c_1 , and l_4 has ancestors c_1 and c_2 .

Formally, we construct the tree as follows. We write f into the labels of \mathfrak{t} , i.e. we define $\mathfrak{t}' = (\text{pos}(\mathfrak{t}), \text{label}_{\mathfrak{t}} \times f)$. By π_1 and π_2 we denote the projections of $\Xi \times \mathbb{N}$ to the respective entries. We let $\mathfrak{s} = \mathfrak{t}' \upharpoonright_w$. Now, as long as there is $v \in \text{pos}(\mathfrak{s})$ with $\pi_2(\mathfrak{s}(v)) > 0$ and $v \neq \varepsilon$, we redefine $\mathfrak{s} = \mathfrak{s} \langle \mathfrak{s} \upharpoonright_{v\pi_2(\mathfrak{s}(v))} \rightarrow v \rangle$. Finally, we let $\mathfrak{s}' = (\text{pos}(\mathfrak{s}), \pi_1 \circ \text{label}_{\mathfrak{s}})$ and $\mathfrak{t}_{\mathfrak{D}}(w) = \mathfrak{s}' \langle \diamond \rightarrow f(w) \rangle$. Note that $\mathfrak{t}_{\mathfrak{D}}(w)$ is matching.

For $i \in \{1, \dots, M+1\}$ and $w \in \text{Cyc}(\mathfrak{D})$ we define

$$\text{wt}_i^{\mathfrak{D}}(w) = \sum_{\substack{v \in \text{pos}(\mathfrak{t}_{\mathfrak{D}}(w)) \\ \text{label}_{\mathfrak{t}_{\mathfrak{D}}(w)}(v) \neq \diamond}} \text{wt}_i(\text{label}_{\mathfrak{t}_{\mathfrak{D}}(w)}(v))$$

$$b_i^{\mathfrak{D}} = \sum_{\substack{v \in \text{pos}(\mathfrak{t}) \\ v \notin \text{Cyc}(\mathfrak{D}) \\ \text{Anc}_{\mathfrak{D}}(v) = \emptyset}} \text{wt}_i(\mathfrak{t}(v)) + \begin{cases} \nu_1(\text{label}_{\mathfrak{t}(\varepsilon)}^i(\varepsilon)) & \text{if } 1 \leq i \leq M \\ \nu_2(\text{label}_{\mathfrak{t}(\varepsilon)}^{M+1}(\varepsilon)) & \text{if } i = M+1. \end{cases}$$

We have the following lemma.

► **Lemma 7.** *Let $t \in T_{\Gamma \times \Omega}$, $s = (\text{pos}(t), \text{label}_{\mathfrak{t}(\varepsilon)}^{\Gamma \diamond})$, $r_i = \text{label}_{\mathfrak{t}(\varepsilon)}^i$ and $\mathfrak{D} = (\mathfrak{t}, f) \in \text{Decomp}(t)$. Let $\{w_1, \dots, w_n\}$ be a lexicographically ordered enumeration of $\text{Cyc}(\mathfrak{D})$. Then*

$$\begin{aligned} \nu_1(r_i(\varepsilon)) + \text{wt}_{\mathcal{A}_1}(s, r_i) &= b_i^{\mathfrak{D}} + \text{wt}_i^{\mathfrak{D}}(w_1) + \dots + \text{wt}_i^{\mathfrak{D}}(w_n) \\ \nu_2(r_{M+1}(\varepsilon)) + \text{wt}_{\mathcal{A}_2}(s, r_{M+1}) &= b_{M+1}^{\mathfrak{D}} + \text{wt}_{M+1}^{\mathfrak{D}}(w_1) + \dots + \text{wt}_{M+1}^{\mathfrak{D}}(w_n) \end{aligned}$$

for $i \in \{1, \dots, M\}$.

Let $\{w_1, \dots, w_n\}$ be a lexicographically ordered enumeration of $\text{Cyc}(\mathfrak{D})$. We consider the system of linear inequalities

$$\begin{aligned} b_i^{\mathfrak{D}} + \text{wt}_i^{\mathfrak{D}}(w_1)X_1 + \dots + \text{wt}_i^{\mathfrak{D}}(w_n)X_n &< b_{M+1}^{\mathfrak{D}} + \text{wt}_{M+1}^{\mathfrak{D}}(w_1)X_1 + \dots + \text{wt}_{M+1}^{\mathfrak{D}}(w_n)X_n \\ 0 &< X_j \end{aligned}$$

where i ranges over $1, \dots, M$ and j over $1, \dots, n$. For a cycle decomposition $\mathfrak{D} \in \text{Decomp}(t)$, the system above is denoted by $\text{LIS}(\mathfrak{D})$.

► **Lemma 8.** *Let $t \in T_{\Gamma \times \Omega}$ be accepting and $\mathfrak{D} = (\mathfrak{t}, f) \in \text{Decomp}(t)$. For every choice of $X_1, \dots, X_n \in \mathbb{N}$, $X_i \geq 1$, there is an accepting tree $s \in T_{\Gamma \times \Omega}$ with*

$$\begin{aligned} \nu_1(\text{label}_s^i(\varepsilon)) + \text{wt}_i(s) &= b_i^{\mathfrak{D}} + \text{wt}_i^{\mathfrak{D}}(w_1)X_1 + \dots + \text{wt}_i^{\mathfrak{D}}(w_n)X_n \\ \nu_2(\text{label}_s^{M+1}(\varepsilon)) + \text{wt}_{M+1}(s) &= b_{M+1}^{\mathfrak{D}} + \text{wt}_{M+1}^{\mathfrak{D}}(w_1)X_1 + \dots + \text{wt}_{M+1}^{\mathfrak{D}}(w_n)X_n \end{aligned}$$

for every $i \in \{1, \dots, M\}$.

Proof. For $X_1 = \dots = X_n = 1$ we know by Lemma 7 that this is true for $s = t$. Otherwise let $w \in \text{Cyc}(\mathfrak{D})$. We can “insert” the tree $\mathfrak{t}_{\mathfrak{D}}(w)$ into \mathfrak{t} at w as follows. Let $\mathfrak{s} = \mathfrak{t}(\mathfrak{t}_{\mathfrak{D}}(w) \rightarrow wf(w)) \langle \mathfrak{t}|_{wf(w)} \rightarrow wf(w)f(w) \rangle$. As w is a cycle, we know that \mathfrak{s} is matching and with $s = \mathcal{J}(\mathfrak{s})$ we have

$$\begin{aligned} \nu_i(\text{label}_s^i(\varepsilon)) + \text{wt}_i(s) &= \nu_i(\text{label}_t^i(\varepsilon)) + \text{wt}_i(t) + \text{wt}_i^{\mathfrak{D}}(w) \\ &= b_i^{\mathfrak{D}} + \text{wt}_i^{\mathfrak{D}}(w_1) + \dots + \text{wt}_i^{\mathfrak{D}}(w_n) + \text{wt}_i^{\mathfrak{D}}(w) \end{aligned}$$

where the last equality follows from Lemma 7. For every $j \in \{1, \dots, n\}$ we apply this procedure $X_j - 1$ times to w_j to obtain s as needed. To see that s is indeed accepting, note that $\text{label}_t^1, \dots, \text{label}_t^M$ are pairwise distinct. Since s is obtained from t by inserting subtrees, $\text{label}_s^1, \dots, \text{label}_s^M$ must also be pairwise distinct. ◀

We are now ready to prove Theorem 2.

► **Lemma 9.** *Let $N = \sum_{k=0}^{|\Omega|} \text{rk}(\Gamma)^k$, $\Upsilon = \sum_{k=1}^{\text{rk}(\Xi)+1} (|\Xi| + 1)^k$, $\Omega = \sum_{k=1}^{\text{rk}(\Xi)+2} (|\Xi| + 1)^k$ and $\Theta = |\Xi|\Omega(2^\Omega + 2)$. Then the following statements are equivalent.*

- (i) $\mathcal{A}_1 \geq \mathcal{A}_2$.
- (ii) *For all accepting $t \in T_{\Gamma \times \Omega}$ with $|\text{pos}(t)| \leq N\Upsilon^2\Theta(2 + \text{rk}(\Xi))$ and all cycle decompositions $\mathfrak{D} \in \text{Decomp}(t)$, the system of linear inequalities $\text{LIS}(\mathfrak{D})$ does not possess an integer solution.*
- (iii) *For all accepting $t \in T_{\Gamma \times \Omega}$ and all cycle decompositions $\mathfrak{D} \in \text{Decomp}(t)$, the system of linear inequalities $\text{LIS}(\mathfrak{D})$ does not possess an integer solution.*

Property (ii) is clearly decidable. There are only finitely many trees to check, each tree has only finitely many cycle decompositions, and the satisfiability of the corresponding linear inequality systems with integers is decidable due to [15]. In particular, Theorem 2 holds.

Proof (sketch). (i) \Rightarrow (iii). We prove this by contradiction and assume that (iii) does not hold. Then there is an accepting $t \in T_{\Gamma \times \Omega}$ and a cycle decomposition $\mathfrak{D} \in \text{Decomp}(t)$ such that the system of inequalities $\text{LIS}(\mathfrak{D})$ has an integer solution. By Lemma 8 we can find an accepting tree $s \in T_{\Gamma \times \Omega}$ with

$$\begin{aligned} \nu_1(\text{label}_s^i(\varepsilon)) + \text{wt}_i(s) &= b_i^{\mathfrak{D}} + \text{wt}_i^{\mathfrak{D}}(w_1)X_1 + \dots + \text{wt}_i^{\mathfrak{D}}(w_n)X_n \\ \nu_2(\text{label}_s^{M+1}(\varepsilon)) + \text{wt}_{M+1}(s) &= b_{M+1}^{\mathfrak{D}} + \text{wt}_{M+1}^{\mathfrak{D}}(w_1)X_1 + \dots + \text{wt}_{M+1}^{\mathfrak{D}}(w_n)X_n \end{aligned}$$

for every $i \in \{1, \dots, M\}$. Thus by Lemma 7 with $s' = (\text{pos}(t), \text{label}_s^{\Gamma \diamond})$ and $r_i = \text{label}_s^i$ for $i \in \{1, \dots, M+1\}$ we have $\nu_1(r_i(\varepsilon)) + \text{wt}_{\mathcal{A}_1}(s', r_i) < \nu_2(r_{M+1}(\varepsilon)) + \text{wt}_{\mathcal{A}_2}(s', r_{M+1})$ for all $i \in \{1, \dots, M\}$. Since \mathcal{A}_1 is M -ambiguous and r_1, \dots, r_M are pairwise distinct, this means $\llbracket \mathcal{A}_1 \rrbracket(s') < \llbracket \mathcal{A}_2 \rrbracket(s')$, i.e. (i) does not hold.

(iii) \Rightarrow (i). We show this by contradiction and assume that (i) does not hold. Then there is some tree $s \in \text{supp}(\mathcal{A}_2)$ with $\llbracket \mathcal{A}_1 \rrbracket(s) < \llbracket \mathcal{A}_2 \rrbracket(s)$. Let $\text{Acc}_{\mathcal{A}_1}(s) = \{r_1, \dots, r_M\}$. Since $\llbracket \mathcal{A}_1 \rrbracket(s) < \llbracket \mathcal{A}_2 \rrbracket(s)$, there must be $r_{M+1} \in \text{Acc}_{\mathcal{A}_2}(s)$ with $\nu_1(r_i(\varepsilon)) + \text{wt}_{\mathcal{A}_1}(s, r_i) < \nu_2(r_{M+1}(\varepsilon)) + \text{wt}_{\mathcal{A}_2}(s, r_{M+1})$ for all $i \in \{1, \dots, M\}$. Consider the accepting tree $t = (\text{pos}(s), (\text{label}_s, r_1, \dots, r_{M+1})) \in T_{\Gamma \times \Omega}$ and let $\mathfrak{D} = (\mathfrak{t}, f) \in \text{Decomp}(t)$. Then according to Lemma 7, the system $\text{LIS}(\mathfrak{D})$ clearly has the integer solution $X_1 = \dots = X_n = 1$, i.e. (iii) does not hold.

(ii) \Leftrightarrow (iii). The direction (iii) \Rightarrow (ii) is clear. We prove (ii) \Rightarrow (iii) by induction on the size of the trees t . For “small” trees, it follows by assuming (ii) as true. For “large” trees t , we

show that if for a cycle decomposition $\mathfrak{D} = (t, f)$ the system $\text{LIS}(\mathfrak{D})$ has an integer solution, then we can find a smaller tree and a cycle decomposition \mathfrak{D}' of that tree for which $\text{LIS}(\mathfrak{D}')$ also has an integer solution. This constitutes a contradiction to our induction hypothesis.

The main issue is how to construct this smaller tree. For words, it is easy. If a word is sufficiently long, there are two cycles with the same label. We remove one of these cycles from the word, thereby making the word shorter, and in $\text{LIS}(\mathfrak{D})$ add the coefficient for this cycle to that of the other, identical cycle. It is clear that this is not possible for trees. By removing any cycle w , we also remove all other cycles v with $w \in \text{Anc}_{\mathfrak{D}}(v)$.

Our solution for this is as follows. Using the concept of ancestors, we construct a tree hierarchy on the cycles of \mathfrak{D} . The “child cycles” of a given cycle w are all cycles for which w is the prefix-largest ancestor. We call this tree \mathfrak{T} and consider two different cases. If \mathfrak{T} has sufficiently many leaves, there are two different leaves pointing to the “same cycle”. More precisely, we find $w_1 \neq w_2$ in $\text{Cyc}(\mathfrak{D})$ with $t_{\mathfrak{D}}(w_1) = t_{\mathfrak{D}}(w_2)$ and for all $v \in \text{Cyc}(\mathfrak{D})$ we have $w_1 \notin \text{Anc}_{\mathfrak{D}}(v)$ and $w_2 \notin \text{Anc}_{\mathfrak{D}}(v)$. The leaves of \mathfrak{T} correspond to cycles which are safe to remove as they are not ancestors of any other cycles. We can therefore remove w_2 and add this cycle’s coefficient to that of w_1 .

However, \mathfrak{T} might not have sufficiently many leaves for this argumentation. But assuming that the number of leaves stays below the bound Υ , sufficiently large trees \mathfrak{T} have arbitrarily long successions of nodes $w, w1, w1^2, \dots, w1^n$ each having only one child. Now consider the cycles $v_0, \dots, v_n \in \text{Cyc}(\mathfrak{D})$ which correspond to such a succession $w, w1, \dots, w1^n$. We can show that there is only a finite number of possible trees $t_{\mathfrak{D}}(v_i)$ for these cycles. If n is large enough, we can find $i_1 < i_2 < i_3 < i_4$ such that $t_{\mathfrak{D}}(v_{i_1}) = t_{\mathfrak{D}}(v_{i_2}) = t_{\mathfrak{D}}(v_{i_3}) = t_{\mathfrak{D}}(v_{i_4})$ and in addition $\{t_{\mathfrak{D}}(v_i) \mid i_1 \leq i \leq i_2\} = \{t_{\mathfrak{D}}(v_i) \mid i_3 \leq i \leq i_4\}$. We can then “remove” all cycles $v_{i_3}, \dots, v_{i_4-1}$ by inserting the subtree of t at v_{i_4} into the node v_{i_3} . The coefficients for the cycles removed in this way can then be added to the coefficients of the corresponding cycles in v_{i_1}, \dots, v_{i_2} . ◀

On the Proof for the Word Case

For words, Theorem 1 was shown by Hashiguchi et al. There are two different versions of the paper, namely [8, 9]. In both papers, it is first shown that for deterministic max-plus word automata $\mathcal{A}_1, \dots, \mathcal{A}_{M+1}$, it is decidable whether $\max_{i=1}^M \llbracket \mathcal{A}_i \rrbracket \geq \llbracket \mathcal{A}_{M+1} \rrbracket$. The approach for the generalization to finitely ambiguous automata is then different in both papers.

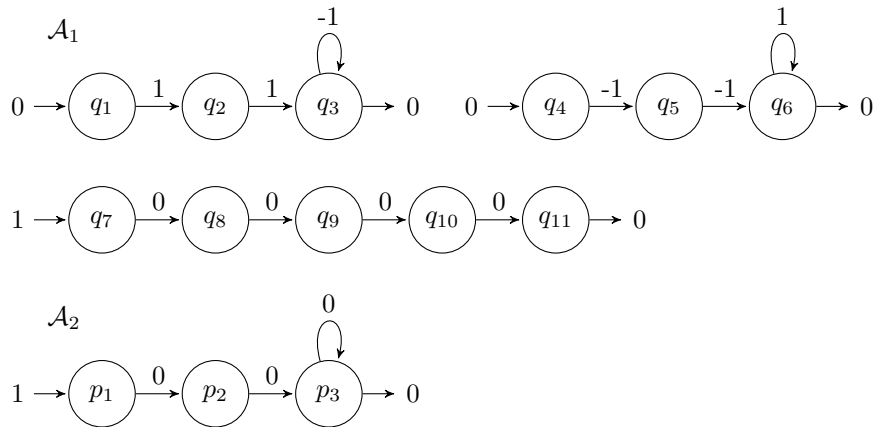
In [8], it is claimed that every finitely ambiguous max-plus word automaton can be written as a pointwise maximum of finitely many deterministic max-plus automata. This argumentation was withdrawn in [9] and the claim posed as an open problem. It does in fact not hold as shown in [1].

In [9], the argumentation is done directly on the runs of the finitely ambiguous max-plus automata. However, this causes problems when not all words have the same number of accepting runs. The two automata in Figure 3 over the one-letter alphabet $\{a\}$ constitute a counter example to Theorem 5.6 in [9], which is similar to our Lemma 9.

One easily checks that $\mathcal{A}_1 \geq \mathcal{A}_2$. There are two accepting runs of \mathcal{A}_1 on a^3 , namely $q_1 a q_2 a q_3 a q_3$ and $q_4 a q_5 a q_6 a q_6$, and one of \mathcal{A}_2 on a^3 , namely $p_1 a p_2 a p_3 a p_3$. The last a thus induces a cycle in the sense of [9]. From this, the linear inequality system

$$\begin{aligned} 2 + (-1) \cdot X &< 1 + 0 \cdot X \\ -2 + 1 \cdot X &< 1 + 0 \cdot X \\ 0 &\leq X \end{aligned}$$

is derived. It clearly has the solution $X = 2$. However, it is stated that from the satisfiability



■ **Figure 3** The automata \mathcal{A}_1 and \mathcal{A}_2 over the alphabet $\{a\}$ constitute a counter example to [9, Theorem 5.6]. The transition letters are omitted.

of this inequality system with an integer value it follows that $\mathcal{A}_1 \geq \mathcal{A}_2$ does not hold. The problem here is that the word a^4 , which is supposed to “realize” the solution of the inequality system, in fact possesses a third accepting run $q_7 a q_8 a q_9 a q_{10} a q_{11}$ which compensates the other two runs.

The proof of Theorem 5.6 in [9] can easily be fixed by normalizing the automaton \mathcal{A}_1 as we did in Lemma 3. If for some $M \geq 1$ we have $|\text{Acc}_{\mathcal{A}_1}(w)| \in \{0, M\}$ for every word w , all arguments of the proof work as intended.

4 The Unambiguity Problem

The unambiguity problem asks whether for a given max-plus-WTA \mathcal{A} there exists an unambiguous max-plus-WTA \mathcal{A}' such that $\llbracket \mathcal{A} \rrbracket = \llbracket \mathcal{A}' \rrbracket$. In this section, we show that the unambiguity problem is decidable for finitely ambiguous max-plus-WTA. We follow ideas from [11, Section 5], where the decidability of this problem was shown for finitely ambiguous max-plus word automata. The unambiguity problem is, in fact, even known to be decidable for polynomially ambiguous max-plus word automata [10]. We leave the question open as to whether the same holds true for polynomially ambiguous max-plus-WTA.

► **Theorem 10.** *For a finitely ambiguous max-plus-WTA \mathcal{A} it is decidable whether there exists an unambiguous max-plus-WTA \mathcal{A}' with $\llbracket \mathcal{A} \rrbracket = \llbracket \mathcal{A}' \rrbracket$. If \mathcal{A}' exists, it can be effectively constructed.*

The rest of this section is dedicated to the proof of Theorem 10.

For an alphabet Γ , a tree over the alphabet $\Gamma_\diamond = (\Gamma \cup \{\diamond\}, \text{rk}_\Gamma \cup \{\diamond \mapsto 0\})$ is called a Γ -context. 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 $d \in \Delta_{\mathcal{A}}$. We denote $\text{Run}_\diamond^{\mathcal{A}}(t) = \text{Run}_{\mathcal{A}'}(t)$ and for $r \in \text{Run}_\diamond^{\mathcal{A}}(t)$ write $\text{wt}_\diamond^{\mathcal{A}}(t, r) = \text{wt}_{\mathcal{A}'}(t, r)$.

For $s \in T_\Gamma$ with $|\{w \in \text{pos}(s) \mid s(w) = \diamond\}| = 1$ and $r \in \text{Run}_\diamond^{\mathcal{A}}(s)$ such that for $w_0 \in \text{pos}(s)$ with $s(w_0) = \diamond$ we have $r(\varepsilon) = r(w_0)$ the pair (s, r) is called an \mathcal{A} -circuit. We call (s, r) small if $|w| \leq |Q|$ for all $w \in \text{pos}(s)$.

Now let \mathcal{A} be a finitely ambiguous max-plus-WTA. We decompose \mathcal{A} into unambiguous max-plus-WTA as follows.

► **Lemma 11** ([16]). *Let \mathcal{A} be a finitely ambiguous max-plus-WTA over Γ , then there exist finitely many unambiguous max-plus-WTA $\mathcal{A}_1, \dots, \mathcal{A}_M$ over Γ with $\llbracket \mathcal{A} \rrbracket = \max_{i=1}^M \llbracket \mathcal{A}_i \rrbracket$ and $\text{supp}(\mathcal{A}_1) = \dots = \text{supp}(\mathcal{A}_M)$.*

Let $\mathcal{A}_1, \dots, \mathcal{A}_M$ be unambiguous max-plus-WTA with $\text{supp}(\mathcal{A}_1) = \dots = \text{supp}(\mathcal{A}_M)$ and $\llbracket \mathcal{A} \rrbracket = \max_{i=1}^M \llbracket \mathcal{A}_i \rrbracket$. We write $\mathcal{A}_i = (Q_i, \Gamma, \mu_i, \nu_i)$ for $i \in \{1, \dots, M\}$. The *product automaton* of $\mathcal{A}_1, \dots, \mathcal{A}_M$ is the trimmed automaton $\mathcal{B} = (Q, \Gamma, \mu, \nu)$ over the product semiring $(\mathbb{R}_{\max})^M$ defined as follows. We let $Q = Q_1 \times \dots \times Q_M$ and for $a \in \Gamma$ with $\text{rk}_\Gamma(a) = m$ and $\mathbf{p}_0, \dots, \mathbf{p}_m \in Q$ with $\mathbf{p}_i = (p_{i1}, \dots, p_{iM})$ we define with $x_j = \mu_j(p_{1j}, \dots, p_{mj}, a, p_{0j})$ and $y_j = \nu_j(p_{0j})$

$$\mu(\mathbf{p}_1, \dots, \mathbf{p}_m, a, \mathbf{p}_0) = \begin{cases} (x_1, \dots, x_M) & \text{if } (x_1, \dots, x_M) \in \mathbb{R}^M \\ (-\infty, \dots, -\infty) & \text{otherwise} \end{cases}$$

$$\nu(\mathbf{p}_0) = \begin{cases} (y_1, \dots, y_M) & \text{if } (y_1, \dots, y_M) \in \mathbb{R}^M \\ (-\infty, \dots, -\infty) & \text{otherwise.} \end{cases}$$

Then \mathcal{B} is unambiguous and for $t \in T_\Gamma$ we have $\llbracket \mathcal{B} \rrbracket(t) = (\llbracket \mathcal{A}_1 \rrbracket(t), \dots, \llbracket \mathcal{A}_M \rrbracket(t))$.

For $\mathbf{q}, \mathbf{p} \in Q$, we write $\mathbf{q} \preceq \mathbf{p}$ if there exists $t \in T_\Gamma$, $r \in \text{Acc}_\mathcal{B}(t)$ and $w_1, w_2 \in \text{pos}(t)$ with $w_1 \leq_p w_2$ such that $r(w_1) = \mathbf{q}$ and $r(w_2) = \mathbf{p}$. We write $\mathbf{q} \approx \mathbf{p}$ if $\mathbf{q} \preceq \mathbf{p}$ and $\mathbf{p} \preceq \mathbf{q}$. By $[\mathbf{q}]$ we denote the set of all $\mathbf{p} \in Q$ with $\mathbf{q} \approx \mathbf{p}$.

► **Definition 12.** Let $s \in T_{\Gamma_\diamond}$ be a Γ -context, $r \in \text{Run}_\mathcal{B}^\diamond(s)$ and write $\text{wt}_\mathcal{B}^\diamond(s, r) = (\theta_1, \dots, \theta_M)$. We define $\text{wt}_i(s, r) = \theta_i$ and $\text{wt}(s, r) = \max_{i=1}^M \text{wt}_i(s, r)$.

A coordinate $i \in \{1, \dots, M\}$ is called *victorious* if $\text{wt}_i(s, r) = \text{wt}(s, r)$. The set of all victorious coordinates of (s, r) is denoted by $\text{Vict}(s, r)$. For $\mathbf{q} \in Q$ we define

$$\text{Vict}([\mathbf{q}]) = \bigcap_{\substack{(s,r) \text{ small } \mathcal{B}\text{-circuit} \\ r(\varepsilon) \in [\mathbf{q}]}} \text{Vict}(s, r)$$

where the empty intersection is defined as $\{1, \dots, M\}$. For $P \subseteq Q$, we let $\text{Vict}(P) = \bigcap_{\mathbf{p} \in P} \text{Vict}([\mathbf{p}])$. We have the following lemma.

► **Lemma 13.** *There is an unambiguous max-plus-WTA \mathcal{A}' with $\llbracket \mathcal{A} \rrbracket = \llbracket \mathcal{A}' \rrbracket$ if and only if for all $t \in T_\Gamma$ and all $r \in \text{Acc}_\mathcal{B}(t)$ we have $\text{Vict}(r(\text{pos}(t))) \neq \emptyset$. The latter property is called the dominance property and is denoted by (\mathbf{P}) .*

(\mathbf{P}) is decidable as follows. We can consider Q as an (unranked) alphabet and construct an FTA which accepts exactly the accepting runs of \mathcal{B} , i.e. all pairs $(\text{pos}(t), r)$ for some $t \in T_\Gamma$ and $r \in \text{Acc}_\mathcal{B}(t)$. Also, for $P \subseteq Q$ we can construct an FTA which accepts all trees in T_Q in which every $p \in P$ occurs at least once as a label. By taking the intersection of these two automata and checking for emptiness, we can decide for every $P \subseteq Q$ whether there is any $t \in T_\Gamma$ and $r \in \text{Acc}_\mathcal{B}(t)$ with $P \subseteq r(\text{pos}(t))$. Checking whether all P for which this is true satisfy $\text{Vict}(P) \neq \emptyset$ is equivalent to checking (\mathbf{P}) .

► **Construction 14.** Let $N = \sum_{i=0}^{|Q|} \text{rk}(\Gamma)^i$, $R = \bigcup_{i=1}^M (\mu_i(\Delta_{\mathcal{A}_i}) \cup \nu_i(Q_i))$ and $C = \max R - \min(R \setminus \{-\infty\})$. For $\mathbf{x} = (x_1, \dots, x_M) \in \mathbb{R}_{\max}^M$ we let $\check{\mathbf{x}} = \min\{x_i \mid 1 \leq i \leq M, x_i \neq -\infty\}$ and $\underline{\mathbf{x}} = \mathbf{x} - (\check{\mathbf{x}}, \dots, \check{\mathbf{x}})$.

Assume that \mathcal{B} satisfies (\mathbf{P}) . We construct an unambiguous max-plus-WTA $\mathcal{A}' = (Q', \Gamma, \mu', \nu')$ with $\llbracket \mathcal{A} \rrbracket = \llbracket \mathcal{A}' \rrbracket$ and $Q' \subset \mathbb{R}_{\max}^M \times Q$ as follows.

Rule 1: For $(a, \mathbf{q}) \in \Delta_\mathcal{B} \cap (\Gamma \times Q)$ with $\mathbf{x} = \mu(a, \mathbf{q}) \in \mathbb{R}^M$, we let $(\underline{\mathbf{x}}, \mathbf{q}) \in Q'$ and $\mu'(a, (\underline{\mathbf{x}}, \mathbf{q})) = \check{\mathbf{x}}$.

Rule 2: Assume for $(\mathbf{z}_1, \mathbf{p}_1), \dots, (\mathbf{z}_m, \mathbf{p}_m) \in Q'$ that we have $d = (\mathbf{p}_1, \dots, \mathbf{p}_m, a, \mathbf{p}_0) \in \Delta_{\mathcal{B}}$ for some $a \in \Gamma$, $\mathbf{p}_0 \in Q$ and $\mathbf{x} = \mu(d) \in \mathbb{R}^M$. We let $\mathbf{t} = \sum_{i=1}^m \mathbf{z}_i + \mathbf{x}$ and define $\mathbf{y} \in \mathbb{R}_{\max}^M$ through

$$y_i = \begin{cases} -\infty & \text{if } t_i < \max\{t_j \mid 1 \leq j \leq M\} - (2N + 1)C \\ t_i & \text{otherwise.} \end{cases}$$

We let $(\underline{\mathbf{y}}, \mathbf{p}_0) \in Q'$ and $\mu'((\mathbf{z}_1, \mathbf{p}_1), \dots, (\mathbf{z}_m, \mathbf{p}_m), a, (\underline{\mathbf{y}}, \mathbf{p}_0)) = \check{\mathbf{y}}$.

Finally, assume $(\mathbf{z}, \mathbf{p}) \in Q'$ and $\mathbf{x} = \nu(\mathbf{p}) \in \mathbb{R}^M$. Then we let $\nu'(\mathbf{z}, \mathbf{p}) = \max_{i=1}^M z_i + x_i$.

► **Lemma 15.** \mathcal{A}' is an unambiguous max-plus-WTA with $\llbracket \mathcal{A} \rrbracket = \llbracket \mathcal{A}' \rrbracket$.

Proof (sketch). \mathcal{A}' is unambiguous as there is a bijection between the accepting runs of \mathcal{B} and \mathcal{A}' . The idea behind \mathcal{A}' is as follows. From a bottom-up perspective, \mathcal{A}' remembers in each coordinate of \mathbf{z} the weight which \mathcal{B} would have assigned to the run in this coordinate “so far”. Since this can become unbounded, we normalize the smallest coordinate to 0 in each transition, make this coordinate’s weight the transition weight, and remember only the difference to this weight in the remaining coordinates. Still, these differences can become unbounded. Therefore, once the difference exceeds the bound $(2N + 1)C$, the coordinates with small weights are discarded by being set to $-\infty$.

We can show that the coordinate k which in \mathcal{B} eventually yields the largest weight will not be discarded. First, we can show that a victorious coordinate of a run will never be smaller than the largest weight (over all coordinates) minus NC . Second, we can show that if l is victorious, then the weight of coordinate k will never be smaller than the weight of l minus $NC + C$. Our assumption is that (\mathbf{P}) holds, so there exists some victorious coordinate in every accepting run. Therefore, the weight of k will never be smaller than the largest weight minus $(2N + 1)C$ and is never discarded. ◀

We now prove that (\mathbf{P}) is a necessary condition, i.e. that from the existence of an unambiguous automaton \mathcal{A}' with $\llbracket \mathcal{A} \rrbracket = \llbracket \mathcal{A}' \rrbracket$ it follows that \mathcal{B} satisfies (\mathbf{P}) .

► **Lemma 16.** If there exists an unambiguous max-plus-WTA $\mathcal{A}' = (Q', \Gamma, \mu', \nu')$ with $\llbracket \mathcal{A} \rrbracket = \llbracket \mathcal{A}' \rrbracket$ then \mathcal{B} satisfies (\mathbf{P}) .

Proof. Let $t \in T_{\Gamma}$ and $r \in \text{Run}_{\mathcal{B}}(t)$. Let $\mathcal{C} = \{(s, r_s) \text{ small } \mathcal{B}\text{-circuit} \mid [r_s(\varepsilon)] \cap r(\text{pos}(t)) \neq \emptyset\}$. Let $\mathbf{p} \in r(\text{pos}(t))$, $(s, r_s) \in \mathcal{C}$ and $\mathbf{q} = r_s(\varepsilon) \in [\mathbf{p}]$.

We can assume that $\mathbf{q} \in r(\text{pos}(t))$ due to the following argument. Since to $\mathbf{p} \preceq \mathbf{q} \preceq \mathbf{p}$, we can find $t_{\mathbf{q}}^{\mathbf{p}}, t_{\mathbf{p}}^{\mathbf{q}} \in T_{\Gamma}$, $r_{\mathbf{q}}^{\mathbf{p}} \in \text{Run}_{\mathcal{B}}(t_{\mathbf{q}}^{\mathbf{p}})$ and $r_{\mathbf{p}}^{\mathbf{q}} \in \text{Run}_{\mathcal{B}}(t_{\mathbf{p}}^{\mathbf{q}})$ such that $r_{\mathbf{q}}^{\mathbf{p}}(\varepsilon) = \mathbf{p}$, $r_{\mathbf{p}}^{\mathbf{q}}(\varepsilon) = \mathbf{q}$ and for some $w_{\mathbf{q}} \in \text{pos}(t_{\mathbf{q}}^{\mathbf{p}})$ and $w_{\mathbf{p}} \in \text{pos}(t_{\mathbf{p}}^{\mathbf{q}})$ we have $r_{\mathbf{q}}^{\mathbf{p}}(w_{\mathbf{q}}) = \mathbf{q}$ and $r_{\mathbf{p}}^{\mathbf{q}}(w_{\mathbf{p}}) = \mathbf{p}$. Thus with $s' = t_{\mathbf{q}}^{\mathbf{p}} \langle t_{\mathbf{p}}^{\mathbf{q}} \langle \diamond \rightarrow w_{\mathbf{p}} \rangle \rightarrow w_{\mathbf{q}} \rangle$ we obtain a circuit $(s', r_{s'})$ with $r_{s'}(\varepsilon) = \mathbf{p}$ and $r_{s'}(w_{\mathbf{q}}) = \mathbf{q}$. We can insert $(s', r_{s'})$ into t and r to obtain a tree t' and a run $r' \in \text{Acc}_{\mathcal{B}}(t')$ with $\mathbf{q} \in r'(\text{pos}(t'))$.

Now let $w_{\mathbf{q}} \in \text{pos}(t)$ with $r(w_{\mathbf{q}}) = \mathbf{q}$ and $w \in \text{pos}(s)$ with $s(w) = \diamond$. We let $s^1 = s$ and for $n \geq 1$ define $s^{n+1} = s \langle s^n \rightarrow w \rangle$. Then from r_s we obtain a circuit $(s^{|Q'|}, r_s^{|Q'|})$ which we can insert at $w_{\mathbf{q}}$ to obtain a tree $t' \in T_{\Gamma}$ and a run $r' \in \text{Acc}_{\mathcal{B}}(t')$. We do this for all small circuits in \mathcal{C} simultaneously. We assume without loss of generality that after this the circuit $(s^{|Q'|}, r_s^{|Q'|})$ is still at position $w_{\mathbf{q}}$. Since $\text{supp } \mathcal{B} = \text{supp } \mathcal{A}'$, we find a run $r'' \in \text{Acc}_{\mathcal{A}'}(t')$. By pigeon hole principle, we find $0 \leq i_1 < i_2 \leq |Q'|$ with $r''(w_{\mathbf{q}} w^{i_1}) = r''(w_{\mathbf{q}} w^{i_2})$. From this, we obtain an \mathcal{A}' -circuit $(s^{i_2-i_1}, \hat{r})$ which corresponds to a \mathcal{B} -circuit $(s^{i_2-i_1}, r_s^{i_2-i_1})$. We can now insert $s^{i_2-i_1}$ at $w_{\mathbf{q}}$ repeatedly to create copies of these circuits. Clearly, this works for all small circuits in \mathcal{C} .

Let c_1, \dots, c_n be an enumeration of \mathcal{C} . We write $c_i = (s_i, r_i)$. By (\hat{s}_i, \hat{r}_i) and $(\check{s}_i, \check{r}_i)$, we denote the circuits in \mathcal{A}' and \mathcal{B} , respectively, we obtain from c_i in the way we obtained $(s^{i_2-i_1}, \hat{r})$ and $(s^{i_2-i_1}, r^{i_2-i_1})$ from (s, r_s) . For $\mathbf{v} = (v_1, \dots, v_n) \in \mathbb{N}^n$, we denote by $t_{\mathbf{v}}$ the tree obtained by adding v_i copies of \hat{s}_i to t for each $i \in \{1, \dots, n\}$. Since \mathcal{B} and \mathcal{A}' are both unambiguous, we can make the following observations.

For $i \in \{1, \dots, n\}$ we let $\rho_i = \text{wt}_{\mathcal{A}'}(\hat{s}_i, \hat{r}_i)$. Then for some constant ρ_0 we have $\llbracket \mathcal{A}' \rrbracket(t_{\mathbf{v}}) = \rho_0 + v_1 \rho_1 + \dots + v_n \rho_n$. Due to the definition of victorious coordinates, for every $\mathbf{v}' = (v_2, \dots, v_n) \in \mathbb{N}^{n-1}$ there is $N_{\mathbf{v}'}^{(1)} \in \mathbb{N}$ such that for all $v_1 > N_{\mathbf{v}'}^{(1)}$ the tuple $\llbracket \mathcal{B} \rrbracket(t_{(v_1, \dots, v_n)})$ has its maximum in entry j_1 for some $j_1 \in \text{Vict}(\hat{s}_1, \check{r}_1)$. Then with $\rho_i^{(1)} = \text{wt}_{j_1}(\hat{s}_i, \check{r}_i)$ for $i \in \{1, \dots, n\}$ and some constant $\rho_0^{(1)}$ we have for all $v_1 > N_{\mathbf{v}'}^{(1)}$ that $\llbracket \mathcal{B} \rrbracket(t_{\mathbf{v}}) = \rho_0^{(1)} + v_1 \rho_1^{(1)} + \dots + v_n \rho_n^{(1)}$. By varying \mathbf{v} , we see that from $\llbracket \mathcal{A}' \rrbracket(t_{\mathbf{v}}) = \llbracket \mathcal{B} \rrbracket(t_{\mathbf{v}})$ it follows that $\rho_i = \rho_i^{(1)}$ for all $i \in \{1, \dots, n\}$. We can do the same for the other circuits $(\hat{s}_2, \check{r}_2), \dots, (\hat{s}_n, \check{r}_n)$ and see that if $j_i \in \text{Vict}(\hat{s}_i, \check{r}_i)$ for every $i \in \{1, \dots, n\}$ then $\text{wt}_{j_1}(\hat{s}_i, \check{r}_i) = \dots = \text{wt}_{j_n}(\hat{s}_i, \check{r}_i)$ for every $i \in \{1, \dots, n\}$. In particular, $j_1 \in \text{Vict}(\hat{s}_i, \check{r}_i)$ for all $i \in \{1, \dots, n\}$. This means $j_1 \in \text{Vict}(r(\text{pos}(t)))$ and \mathcal{B} satisfies **(P)**. ◀

5 The Sequentiality Problem

The sequentiality problem asks whether for a given max-plus-WTA \mathcal{A} there exists a deterministic max-plus-WTA \mathcal{A}' such that $\llbracket \mathcal{A} \rrbracket = \llbracket \mathcal{A}' \rrbracket$. The term “sequentiality” stems from the fact that in the weighted setting, deterministic automata are also often called *sequential*. In this section, we show that the sequentiality problem is decidable for finitely ambiguous max-plus-WTA. For words, this is known due to [11].

Let $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ be a max-plus-WTA. We say that \mathcal{A} satisfies the *twins property* [14, 3] if the following holds. Whenever for $q, q' \in Q$ there exist $t \in T_{\Gamma}$, $r, r' \in \text{Run}_{\mathcal{A}}(t)$ with $r(\varepsilon) = q$, $r'(\varepsilon) = q'$ and \mathcal{A} -circuits $(s, r_1), (s, r_2)$ with $r_1(\varepsilon) = q$ and $r_2(\varepsilon) = q'$ then $\text{wt}_{\mathcal{A}}^{\diamond}(s, r_1) = \text{wt}_{\mathcal{A}}^{\diamond}(s, r_2)$.

► **Lemma 17.** *Let \mathcal{A} be a trim unambiguous max-plus-WTA. There exists a deterministic max-plus-WTA \mathcal{A}' with $\llbracket \mathcal{A} \rrbracket = \llbracket \mathcal{A}' \rrbracket$ if and only if \mathcal{A} satisfies the twins property. If it exists, it can be effectively constructed.*

Proof (sketch). If \mathcal{A} satisfies the twins property, we know due to [3, Lemma 5.10] that a deterministic max-plus-WTA \mathcal{A}' with $\llbracket \mathcal{A}' \rrbracket = \llbracket \mathcal{A} \rrbracket$ can be effectively constructed.

To show that the twins property is also a necessary condition, we can apply an idea similar to the proof of [14, Theorem 9]. ◀

► **Lemma 18** ([3, Theorem 5.17]). *For an unambiguous max-plus-WTA \mathcal{A} it is decidable whether \mathcal{A} satisfies the twins property.*

► **Theorem 19.** *For a finitely ambiguous max-plus-WTA \mathcal{A} it is decidable whether there exists a deterministic max-plus-WTA \mathcal{A}' with $\llbracket \mathcal{A} \rrbracket = \llbracket \mathcal{A}' \rrbracket$. If \mathcal{A}' exists, it can be effectively constructed.*

Proof. Let \mathcal{A} be a finitely ambiguous max-plus-WTA. Due to Theorem 10 we can decide whether there exists an equivalent unambiguous max-plus-WTA. If this is not the case, \mathcal{A} can also not be determinizable. Otherwise we can effectively construct an unambiguous max-plus-WTA \mathcal{A}' with $\llbracket \mathcal{A} \rrbracket = \llbracket \mathcal{A}' \rrbracket$. Due to Lemma 18 we can decide whether \mathcal{A}' satisfies the twins property, which according to Lemma 17 is equivalent to deciding whether \mathcal{A} is determinizable. ◀

References

- 1 Sebastian Bala and Artur Koniński. Unambiguous automata denoting finitely sequential functions. In Adrian-Horia Dediu, Carlos Martín-Vide, and Bianca Truthe, editors, *Proc. LATA*, volume 7810 of *LNCS*, pages 104–115. Springer, 2013.
- 2 Jean Berstel and Christophe Reutenauer. *Rational Series and Their Languages*. Springer, 1988.
- 3 Matthias Büchse, Jonathan May, and Heiko Vogler. Determinization of weighted tree automata using factorizations. *Journal of Automata, Languages and Combinatorics*, 15(3/4):229–254, 2010.
- 4 Manfred Droste, Werner Kuich, and Heiko Vogler. *Handbook of Weighted Automata*. Springer, 2009.
- 5 Stéphane Gaubert. Performance evaluation of $(\max,+)$ automata. *IEEE T. Automat. Contr.*, 40(12):2014–2025, 1995.
- 6 Stéphane Gaubert and Jean Mairesse. Modeling and analysis of timed Petri nets using heaps of pieces. *IEEE T. Automat. Contr.*, 44(4):683–697, 1999.
- 7 Kosaburo Hashiguchi. Algorithms for determining relative star height and star height. *Inf. Comput.*, 78(2):124–169, 1988.
- 8 Kosaburo Hashiguchi and Kenichi Ishiguro. Decidability of the equivalence problem for finitely ambiguous finite automata. *Sūri Kaiseki Kenkyūsho Kōkyūroku*, 960:23–36, 1996.
- 9 Kosaburo Hashiguchi, Kenichi Ishiguro, and Shuji Jimbo. Decidability of the equivalence problem for finitely ambiguous finite automata. *IJAC*, 12(3):445–461, 2002.
- 10 Daniel Kirsten and Sylvain Lombardy. Deciding unambiguity and sequentiality of polynomially ambiguous min-plus automata. In Susanne Albers and Jean-Yves Marion Marion, editors, *Proc. STACS*, volume 3 of *LIPICs*, pages 589–600. LZI, 2009.
- 11 Ines Klimann, Sylvain Lombardy, Jean Mairesse, and Christophe Prieur. Deciding unambiguity and sequentiality from a finitely ambiguous max-plus automaton. *Theor. Comput. Sci.*, 327(3):349–373, 2004.
- 12 Daniel Krob. The equality problem for rational series with multiplicities in the tropical semiring is undecidable. *IJAC*, 4(3):405–426, 1994.
- 13 Werner Kuich and Arto Salomaa. *Semirings, Automata, Languages*. Springer, 1986.
- 14 Mehryar Mohri. Finite-state transducers in language and speech processing. *Comput. Linguist.*, 23(2):269–311, 1997.
- 15 George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, 1988.
- 16 Erik Paul. On finite and polynomial ambiguity of weighted tree automata. In Srećko Brlek and Christophe Reutenauer, editors, *Proc. DLT*, volume 9840 of *LNCS*, pages 368–379. Springer, 2016.
- 17 Slav Petrov. Latent variable grammars for natural language parsing. In *Coarse-to-Fine Natural Language Processing*, chapter 2, pages 7–46. Springer, 2012.
- 18 Arto Salomaa and Matti Soittola. *Automata-Theoretic Aspects of Formal Power Series*. Springer, 1978.
- 19 Marcel-Paul Schützenberger. On the definition of a family of automata. *Inform. Control*, 4(2–3):245–270, 1961.
- 20 Imre Simon. Limited subsets of a free monoid. In *Proc. FOCS*, pages 143–150. IEEE Computer Society, 1978.
- 21 Imre Simon. Recognizable sets with multiplicities in the tropical semiring. In Michal Chytil, Ladislav Janiga, and Václav Koubek, editors, *Proc. MFCS*, volume 324 of *LNCS*, pages 107–120. Springer, 1988.
- 22 Andreas Weber. Finite-valued distance automata. *Theor. Comput. Sci.*, 134(1):225–251, 1994.