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

Erik Paul
epaul at informatik.uni-leipzig.de

January 25, 2019

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 \[24, 23, 16, 1, 5\] 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 \[10\], to decide the finite power property \[25, 26\] and to model certain timed discrete event systems \[7, 8\]. Additionally, they appear in the context of natural language processing \[17\].

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 \[27, 11\] 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 \[14\] 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 \[15\] for max-plus automata, but for finitely ambiguous max-plus automata it becomes decidable \[27, 11\]. The sequentiality problem asks whether for a given automaton, there exists an equivalent deterministic automaton. This was shown to be decidable by Mohri \[17\] 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 \[14, 13\]. 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 \[22\]. Our approach to the decidability of the equivalence problem uses ideas from \[11\]. We reduce the equivalence problem to the same decidable problem as \[11\], namely the decidability of the existence of an integer solution for a system of linear inequalities \[18\]. However, instead of the cycle decompositions which were used both in \[11\] and \[21\], we employ Parikh’s theorem \[19, Theorem 2\]. This idea was suggested by Mikołaj Bojańczyk in a discussion following the presentation of the proof from \[21\]. The proof presented here is a revised version of the one from \[21\].

The decidability of the unambiguity problem employs ideas from \[14\]. Here, we show how the dominance property can be generalized to max-plus tree automata. To show the decidability of the sequentiality problem for finitely ambiguous max-plus tree automata, we first combine results from \[4\] and \[17\] 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 \[20\], 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

For a set $X$, the power set of $X$ is denoted by $\mathcal{P}(X)$ and the cardinality of $X$ is denoted by $|X|$. For two sets $X$ and $Y$ and a mapping $f: X \rightarrow Y$, we call $X$ the domain of $f$, denoted by $\text{dom}(f)$, and $Y$ the range of $f$, denoted by $\text{range}(f)$. For a subset $X' \subseteq X$, the set $f(X') = \{ y \in Y | \exists x \in X' : f(x) = y \}$ is called the image or range of $X'$ under $f$. For an element $y \in Y$, the set $f^{-1}(y) = \{ x \in X | f(x) = y \}$ is called the preimage of $y$ under $f$. For a second mapping $g: X \rightarrow Y$, we write $f = g$ if for all $x \in X$ we have $f(x) = g(x)$.

An alphabet $\Sigma$ is a non-empty finite set. By $\Sigma^*$, we denote the set of all finite words over $\Sigma$. The empty word is denoted by $\varepsilon$, and the length of a word $w \in \Sigma^*$ by $|w|$. The number of occurrences of a letter $a \in \Sigma$ in a word $w$ is denoted by $|w|_a$. A subset $L \subseteq \Sigma^*$ is called a language over $\Sigma$.

Let $N_0 = \{0, 1, 2, \ldots\}$ and $N_+ = \{1, 2, 3, \ldots\}$. By $N_0^\infty$, we denote the set of all finite words over $N_0$. The set $N_0^\infty$ is partially ordered by the prefix relation $\preceq$ and totally ordered with respect to the lexicographic ordering $\leq$. A ranked alphabet is a pair $(\Gamma, \mathbf{r}_\Gamma)$, often abbreviated by $\Gamma$, where $\Gamma$ is a finite set and $\mathbf{r}_\Gamma : \Gamma \rightarrow N_0$ a mapping which assigns a rank to every symbol. For every $m \geq 0$ we define $\Gamma^{(m)} = \mathbf{r}_\Gamma^{-1}(m)$ as the set of all ranks of rank $m$. The rank of $\Gamma$ is defined as $\mathbf{r}(\Gamma) = \max\{\mathbf{r}(\alpha) | \alpha \in \Gamma \}$.

The set of (finite, labeled, and ordered) $\Gamma$-trees, denoted by $T_{\Gamma}$, is the set of all pairs $t = (\mathbf{pos}(t), \mathbf{label}(t))$, where $\mathbf{pos}(t) \subseteq N_0^\infty$ is a finite non-empty prefix-closed set of positions, $\mathbf{label}(t) : \mathbf{pos}(t) \rightarrow \Gamma$ is a mapping, and for every $w \in \mathbf{pos}(t)$ we have $w|_i \in \mathbf{pos}(t)$ iff $1 \leq i \leq \mathbf{r}(\mathbf{label}(w))$. We write $t(w)$ for $\mathbf{label}(w)$. We also refer to the elements of $\mathbf{pos}(t)$ as nodes, to $\varepsilon$ as the root of $t$, and to prefix-maximal nodes as leaves. The height of $t$ is defined by $\text{height}(t) = \max_{w \in \mathbf{pos}(t)} |w|$.

Now let $t, s \in T_{\Gamma}$ and $w \in \mathbf{pos}(t)$. The subtree of $t$ at $w$, denoted by $t|_w$, is a $\Gamma$-tree defined as follows. We let $\mathbf{pos}(t|_w) = \{ v \in N_0^\infty | v \preceq w \}$ and for $v \in \mathbf{pos}(t|_w)$, we let $\mathbf{label}(t|_w)(v) = t(w)$.

The substitution of $s$ into $w$ of $t$, denoted by $t(s \rightarrow w)$, is a $\Gamma$-tree defined as follows. We let $\mathbf{pos}(t(s \rightarrow w)) = \{ v \in \mathbf{pos}(t) | v \not\preceq w \}$ and $\{ v \in \mathbf{pos}(s) | v \preceq w \}$. For $v \in \mathbf{pos}(t(s \rightarrow w))$, we let $\mathbf{label}(t(s \rightarrow w))(v) = s(w)$ if $v = wu$ for some $u \in \mathbf{pos}(s)$, and otherwise $\mathbf{label}(t(s \rightarrow w))(v) = t(w)$.

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

For a ranked alphabet $\Gamma$, a tree over the alphabet $\Gamma_\circ = (\Gamma \cup \{\circ\}, \mathbf{r}_\Gamma \cup \{\circ \rightarrow 0\})$ is called a $\Gamma$-context. Let $t \in T_{\Gamma_\circ}$ be a $\Gamma$-context and let $w_1, \ldots, w_n \in \mathbf{pos}(t)$ be a lexicographically ordered enumeration of all leaves of $t$ labeled $\circ$. Then we call $t$ an $n$-$\Gamma$-context and define $\mathbf{label}(t)(i) = w_i$ for $i \in \{1, \ldots, n\}$. For an $n$-$\Gamma$-context $t$ and contexts $t_1, \ldots, t_n \in T_{\Gamma_\circ}$, we define $t(t_1, \ldots, t_n) = t(t_1 \rightarrow \mathbf{label}(t_1)(1) \ldots (t_n \rightarrow \mathbf{label}(t_n)(n))$ by substitution of $t_1, \ldots, t_n$ into the $\circ$-leaves of $t$. A 1-$\Gamma$-context is also called a $\Gamma$-word. For a $\Gamma$-word $s$, we define $s^1 = s$ and $s^{n+1} = s(s^n)$ for $n \geq 1$.

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

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

For a commutative semiring $(K, \oplus, \circ, 0, 1)$ and a number $n \geq 1$, the product semiring $(K^n, \oplus_n, \circ_n, 0_n, 1_n)$ is defined by componentwise operations and the constants $0_n = (0, \ldots, 0)$ and $1_n = (1, \ldots, 1)$. We will usually denote $\oplus_n$ and $\circ_n$ simply by $\oplus$ and $\circ$.

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

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). An FTA $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ is also written as a tuple $\mathcal{A}' = (Q, \Gamma, \delta, F)$ where $\delta = \{ d \in \Delta_{\mathcal{A}} | \mu(d) = 1 \}$ and $F = \{ q \in Q | \nu(q) = 1 \}$.

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

For \( r \in \Run_A(t) \), the weight of \( r \) is defined by

\[
\wt_A(t, r) = \bigcap_{w \in \pos(t)} \mu(t, r, w).
\]

The behavior of \( A \), denoted by \( [A] \), is the mapping defined for every \( t \in T_T \) by \( [A](t) = \Bigoplus_{r \in \Acc_A(t)} \big( \wt_A(t, r) \uplus \nu(\varepsilon(r)) \big) \), where the sum over the empty set is 0 by convention. The support of \( A \) is the set \( \supp(A) = \{ t \in T_T \mid [A](t) \neq 0 \} \). The support of an FTA \( A \) is also called the language accepted by \( A \) and denoted by \( \lang(A) \). A subset \( L \subseteq T_T \) is called recognizable if there exists an FTA \( A \) with \( L = \lang(A) \).

For a WTA \( A = (Q, \Gamma, \mu, \nu) \), a run of \( A \) on a \( \Gamma \)-context \( t \) is a run of the WTA \( A' = (Q, \Gamma, \mu', \nu) \) on \( t \), where \( \mu'(\varepsilon, q) = 1 \) for all \( q \in Q \) and \( \mu'(d) = \mu(d) \) for \( d \in \Delta_A \). We denote \( \Run_A(t) = \Run_{A'}(t) \) and for \( r \in \Run_A(t) \) write \( \wt_A(t, r) = \wt_{A'}(t, r) \). For an \( n \)-\( \Gamma \)-context \( t \in T_T \), states \( q_0, \ldots, q_n \), we denote by \( \Run_A^{\oplus}(q_0, \ldots, q_n, t, q_0) \) the set of all runs \( r \in \Run_A(t) \) such that \( r(\varepsilon) = q_0 \) and \( r(\varepsilon_i(t)) = q_i \) for every \( i \in \{1, \ldots, n\} \).

Similar to trees, we define restrictions, substitutions, and powers of runs as follows. Let \( t, s, v \in T_T \), \( r \in \Run_A(t), w \in \pos(t) \), and \( r_s \in \Run_A(s) \) with \( r_s(\varepsilon) = r(w) \). Then we define \( r|_w \in \Run_A(t|_w) \) by \( r|_w(v) = r(wv) \) for \( v \in \pos(t|_w) \). We define \( r|_w(v) \in \Run_A(t|_w) \) by \( r|_w(v) = r(wv) \) for \( v \in \pos(t) \). We let \( v = wu \) for some \( u \in \pos(s) \), and \( r|_w(v) = r(u) \) otherwise.

A WTA \( A \) is called deterministic if for every \( m \geq 0 \), \( \mu(p, a, q) \neq 0 \). We call \( A \) finitely ambiguous or \( M \)-ambiguous if \( [\Acc_A(t)] \leq M \) for some \( M \geq 1 \) and every \( t \in T_T \). A 1-ambiguous WTA is also called unambiguous.

We recall that for every recognizable language \( L \subseteq T_T \), there exists a deterministic FTA \( A \) with \( \lang(A) = L \).

For a WTA \( A \), we define a relation \( \preceq \) on \( Q \) by \( p \preceq q \) if there exists a \( \Gamma \)-word \( s \in T_T \) such that \( \Run_A^{\oplus}(p, s, p) \neq 0 \). We write \( p \approx q \) if \( p \preceq q \) and \( q \preceq p \). By \( [p] \) we denote the set of all \( q \in Q \) with \( p \approx q \).

A WTA \( A \) is called trim if for every \( p \in Q \), there exist \( t \in T_T, r \in Acc_A(t) \), and \( w \in pos(t) \) such that \( p = r(w) \). The trim part of \( A \) is the automaton obtained by removing all states \( p \in Q \) for which no such \( t, r, \) and \( w \) exist. This process obviously has no influence on \( [A] \).

### 3 The Equivalence Problem

For two max-plus-WTA \( A_1 \) and \( A_2 \) over an alphabet \( \Gamma \), we say that \( A_1 \) dominates \( A_2 \), denoted by \( [A_1] \geq [A_2] \), if for all trees \( t \in T_T \) we have \( [A_1](t) \geq [A_2](t) \). We say that \( A_1 \) and \( A_2 \) are equivalent if \( [A_1] = [A_2] \).

The equivalence problem for max-plus (tree) automata asks whether for two given max-plus (tree) automata \( A_1 \) and \( A_2 \), it holds that \( [A_1] = [A_2] \). For words, this problem was shown to be undecidable in general [15], but it is decidable if both automata are finitely ambiguous [11]. In this section, we prove that the equivalence problem is decidable for finitely ambiguous max-plus-WTA. Like in [11], we reduce the equivalence problem to the decidability of the existence of an integer solution for a system of linear inequalities [18]. This problem is decidable only for systems over the rationals, which is why for the equivalence problem, we consider only max-plus-WTA over the max-plus semiring \( \mathbb{Q}_{\max} = (\mathbb{Q} \cup \{-\infty\}, \max, +, -\infty, 0) \) restricted to the rationals. The proof presented here is a revised version of the one from [21]. It is largely based on ideas from [11], but employs Parikh’s theorem [19, Theorem 2] instead of the cycle decompositions which were used both in [11] and [21]. This idea was suggested by Mikolał Bojańczyk in a discussion following the presentation of the proof from [21]. We formulate the main result of this section as follows.

**Theorem 1.** The equivalence problem for finitely ambiguous max-plus tree automata with transition and final weights from \( \mathbb{Q} \cup \{-\infty\} \) is decidable.

In fact, we will show that if \( A_1 \) is a finitely ambiguous max-plus-WTA and \( A_2 \) any max-plus-WTA, then it is decidable whether \( A_1 \) dominates \( A_2 \).
Theorem 2. Let $A_1$ be a finitely ambiguous max-plus-WTA and $A_2$ any max-plus-WTA, both with transition and final weights from $Q \cup \{-\infty\}$. It is decidable whether or not $\|A_1\| \geq \|A_2\|$.

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 part of the proof, we will employ the following concepts.

Let $\Sigma = \{a_1, \ldots, a_n\}$ be an alphabet. The Parikh vector $p(w) \in \mathbb{N}_+^n$ of a word $w \in \Sigma^*$ is given by $p(w) = (|w|_{a_1}, |w|_{a_2}, \ldots, |w|_{a_n})$. For a language $L \subseteq \Sigma^*$, the Parikh image of $L$ is the set $p(L) = \{p(w) \mid w \in L\}$.

A subset $J \subseteq \mathbb{N}_+^n$ is called \textit{linear} if there exist $k \geq 0$ and $\alpha, \beta_1, \ldots, \beta_k \in \mathbb{N}_0^k$ such that

$$J = \{\alpha + \sum_{i=1}^k \beta_i n_i \mid n_i \in \mathbb{N}_0\}.$$ 

The set $J$ is called \textit{semilinear} if it is the union of finitely many linear subsets of $\mathbb{N}_+^n$.

A context-free grammar (short: CFG) $G = (\Sigma, \Gamma, P, S)$ is a tuple $(N, \Sigma, P, S)$ where $(1) N$ is a finite set of nonterminal symbols, $(2) \Sigma$ is a finite set of terminal symbols with $N \cap \Sigma = \emptyset$, $(3) P \subseteq N \times (N \cup \Sigma)^*$ is a finite set of productions or rules, and $(4) S \in N$ is the initial symbol. We usually denote a rule $(A, w) \in P$ by $A \rightarrow w$.

Let $G = (N, \Sigma, P, S)$ be a context-free grammar. For $u, v \in (\Sigma \cup \{\epsilon\})^*$ we write $u \Rightarrow_G v$ if there exists $u', u'' \in (\Sigma \cup \{\epsilon\})^*$ and a production $A \rightarrow w \in P$ so that $u = u'Au''$ and $v = u'wu''$. The language generated by $G$ is the language

$$L(G) = \{w \in \Sigma^* \mid \exists n \geq 0 \exists u_1, \ldots, u_n \in (\Sigma \cup \{\epsilon\})^*: S \Rightarrow_G u_1 \Rightarrow_G \ldots \Rightarrow_G u_n \Rightarrow_G w\}.$$ 

A language $L \subseteq \Sigma^*$ is called \textit{context-free} if there exists a context-free grammar $G$ with $L = L(G)$.

As a first step, we show in the following lemma that every finitely ambiguous max-plus-WTA $A$ can be normalized such that all trees, on which there exists at least one accepting run of $A$, have the same number of accepting runs. The idea here is that we can simply add dummy runs with low weight for every tree which does not already have a sufficient number of runs.

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

Proof. First, we show that for every $n \in \{1, \ldots, M\}$, the set $L_n = \{t \in T_R \mid |\text{Acc}_A(t)| \geq n\}$ is recognizable. For this, consider the FTA $A_n = (Q^n \times P\{(1, \ldots, n)^2\}, \Gamma, \delta_n, F_n)$, where $P\{(1, \ldots, n)^2\}$ denotes the power set of $\{1, \ldots, n\}^2$, defined as follows. For $a \in \Gamma$ with $\rho_\Gamma(a) = m$, $\overline{p_0}, \ldots, \overline{p_m} \in Q^n$ with $\overline{p_0} = (p_{10}, \ldots, p_{i0})$ and $R_0 \subseteq \{(1, \ldots, n)^2\}$, we let $((p_{i1}, R_{i1}), \ldots, (p_{im}, R_{im}), a, (\overline{p}_0, R_0)) \in \delta_n$ iff for all $j \in \{1, \ldots, n\}$ we have $\mu((p_{i1}, \ldots, p_{im}), a, \overline{p}_0) \neq -\infty$ and $R_0 = \{\delta_n \mid j \in \{1, \ldots, n\} \mid p_{i0} \neq \rho_\Gamma(a) \}$.

It is easy to see that there is an accepting run of $A_n$ on $t \in T_R$ if and only if there are at least $n$ pairwise distinct accepting runs of $A$ on $t$. Therefore, $L(A_n) = L_n$. Since recognizable languages are closed under complement and intersection, for $n \in \{1, \ldots, M - 1\}$ the languages $L'_n = L_n \setminus L_{n+1} = \{t \in T_R \mid |\text{Acc}_A(t)| = n\}$ are also recognizable and we can find deterministic FTA $A'_n = (Q'_n, \Gamma, \delta'_n, F'_n)$ with $L(A'_n) = L'_n$.

Now let $\kappa$ be the smallest weight used in $A$, i.e., with $R = \mu(\Delta_A) \cup \nu(Q)$ we let $\kappa = \min(R \setminus \{-\infty\})$. For $n \in \{1, \ldots, M - 1\}$, we define the max-plus-WTA $A''_n = (Q''_n, \Gamma, \mu''_n, \nu''_n)$ by

$$\mu''_n(d) = \begin{cases} \kappa & \text{if } d \in \delta''_n \\ -\infty & \text{otherwise} \end{cases} \quad \text{and} \quad \nu''_n(q) = \begin{cases} \kappa & \text{if } q \in F''_n \\ -\infty & \text{otherwise}. \end{cases}$$ 

Finally, we construct $A'$ as follows. For each $n \in \{1, \ldots, M - 1\}$, we take $M - n$ copies of $A''_n$ and unite them with $A$, where we assume that all sets of states are pairwise disjoint. By choice of $\kappa$, this does not influence the behavior of $A$. By choice of the languages $L_{n''}$, every tree which had at least one accepting run in $A$ now has exactly $M$ accepting runs in $A'$ and all other trees still have no accepting run in $A'$. \(\square\)
Next, we show that every max-plus-WTA $A$ can be normalized such that all final weights are equal either to $-\infty$ or to 0. The idea is that the final weight can be included in the transition weight of the transition at the root, see also [3].

**Lemma 4** ([3]). Let $A = (Q, \Gamma, \mu, \nu)$ be a max-plus-WTA. Then there exists a max-plus-WTA $A'$ = $(Q', \Gamma, \mu', \nu')$ with $[A] = [A']$, $\nu'(Q') \subseteq \{-\infty, 0\}$, and $|Acc_A(t)| = |Acc_{A'}(t)|$ for every $t \in T_R$.

**Proof.** We define a max-plus-WTA $A'$ = $(Q', \Gamma, \mu', \nu')$ as follows. We let $Q' = Q \cup \{\ast\}$, where $\ast \notin Q$ is a new state, and define $\nu'(\ast) = 0$ and $\nu'(q) = -\infty$ for all $q \in Q$. For every $d = (p_1, \ldots, p_m, a, p_0) \in \Delta_A$, we let $\mu'(d) = \mu(d)$ and $\mu'(p_1, \ldots, p_m, a, \ast) = \mu(d) + \nu(p_0)$. On all remaining transitions we define $\mu'$ as $-\infty$.

It is easy to see that for every tree $t \in T_R$, we have a bijection $f: Acc_A(t) \rightarrow Acc_{A'}(t)$ given by $(f(r))(\epsilon) = \ast$ and $(f(r))(w) = r(w)$ for $w \in \text{pos}(t) \setminus \{\epsilon\}$, and for this bijection it holds that $wt_A(t, r) + \nu(r(\epsilon)) = wt_{A'}(t, f(r))$. □

For the rest of this section, we fix an $M$-ambiguous max-plus-WTA $A_1$ and a max-plus-WTA $A_2$, both with transition and final weights from $Q \cup \{-\infty\}$. We write $A_i = (Q_i, \Gamma, \mu_i, \nu_i)$ for $i = 1, 2$. By Lemma 3, we can assume that for all $t \in T_R$ we have $|Acc_{A_1}(t)| \subseteq \{0, M\}$. By Lemma 4, we may furthermore assume that $\nu_1(Q_1) \subseteq \{-\infty, 0\}$ and $\nu_2(Q_2) \subseteq \{-\infty, 0\}$. Note that $[A_1] \geq [A_2]$ can only hold if $\text{supp}(A_2) \subseteq \text{supp}(A_1)$, which is decidable since the supports of $A_1$ and $A_2$ are recognizable languages [23, 12]. This also follows from the proof of Lemma 3. Therefore, in the forthcoming considerations we will assume that $\text{supp}(A_2) \subseteq \text{supp}(A_1)$ holds.

We call a tuple $\bar{v} \in Q_1^{M+1}$ an outcome vector if there exists a tree $t \in T_R$, runs $r_1, \ldots, r_M \in Acc_{A_1}(t)$, and a run $r_{M+1} \in Acc_{A_2}(t)$ with $Acc_{A_i}(t) = \{r_1, \ldots, r_M\}$ and $\bar{v} = (wt_{A_1}(t, r_1), \ldots, wt_{A_1}(t, r_M), wt_{A_2}(t, r_{M+1}))$. We denote the set of all outcome vectors by $\mathbb{O}$. We can make the following observation.

**Proposition 5.** $A_1$ does not dominate $A_2$ iff there exists a vector $(v_1, \ldots, v_{M+1}) \in \mathbb{O}$ such that for all $i \in \{1, \ldots, M\}$ we have $v_i < v_{M+1}$.

We give an overview of the rest of the proof. We first construct a weighted tree automaton $A$ over the product semiring $(Q_{\max})^{M+1}$ such that the weights realized by the runs of $A$ are exactly the vectors from $\mathbb{O}$. We then define Parikh vectors of runs by counting the transitions occurring in a run, just like the Parikh vector of a word counts the number of occurrences of the letters in a word. By arranging the weight vectors of the transitions of $A$ as columns into a matrix $\Omega$, we see that the weight of a run of $A$ is simply the result of multiplying the matrix $\Omega$ with the Parikh vector of the run.

We proceed to show that the set of Parikh vectors of the accepting runs of $A$ can also be expressed as the Parikh image of a context-free language over the alphabet of transitions from $\Delta_A$. By Parikh’s theorem, the Parikh image of a context-free language is semilinear, and thus so is the set of Parikh vectors of the accepting runs of $A$.

It follows that the set $\mathbb{O}$ can be represented as the image of a semilinear set, namely the set of Parikh vectors of the accepting runs of $A$, under a matrix with rational entries, namely the matrix $\Omega$. We then use Proposition 5 to reduce the dominance problem to the satisfiability problem of systems of linear inequalities over the rationals with an integer solution. The latter problem is decidable [2, Theorem 3.4]. We begin by constructing $A$.

**Lemma 6.** There exists a weighted tree automaton $A = (Q, \Gamma, \mu, \nu)$ over the product semiring $(Q_{\max})^{M+1}$ such that $\mathbb{O} = \{wt_A(t, r) \mid t \in T_R, r \in Acc_A(t)\}$. The automaton $A$ can be effectively constructed from $A_1$ and $A_2$.

**Proof.** We let $Q = Q_1^M \times Q_2 \times \mathcal{P}(\{1, \ldots, M\})^2$. The first $M+1$ entries of the states from $Q$ are used to simulate the $M$ runs of $A_1$ and one run of $A_2$, and the last entry is used to keep a record of which runs from $A_1$ are distinct in order to ensure that accepting runs of $A$ simulate all accepting runs of $A_1$ in the respective entries. For $a \in \Gamma$ with $r_k(a) = m$ and $p_0, \ldots, p_m \in Q$ with $p_i = (p_{1i}, \ldots, p_{Mi}, p_{MI+1}, R_i)$, we define weights as follows. For $j \in \{1, \ldots, M\}$ we let $x_{j} = \mu(p_{1j}, \ldots, p_{mj}, a, p_{0j})$ and $y_{j} = \nu(p_{0j})$, and we let $x_{M+1} = \mu_2(p_{1M+1}, \ldots, p_{MM+1}, a, p_{0M+1})$ and $y_{M+1} = \nu_2(p_{0M+1})$. Furthermore, we let $R = \{(k, l) \in \{1, \ldots, M\}^2 \mid k \neq l\} \cup \bigcup_{i=1}^M R_i$. Then we define $\mu$ and $\nu$ by

\[
\begin{align*}
\mu(p_1, \ldots, p_m, a, p_0) &= \begin{cases} 
(x_1, \ldots, x_{M+1}) & \text{if } (x_1, \ldots, x_{M+1}) \in Q_1^{M+1} \text{ and } R_0 = R \\
(-\infty, \ldots, -\infty) & \text{otherwise}
\end{cases} \\
\nu(p_0) &= \begin{cases} 
(y_1, \ldots, y_{M+1}) & \text{if } (y_1, \ldots, y_{M+1}) \in Q_2^{M+1} \text{ and } R_0 = R = \{(k, l) \in \{1, \ldots, M\}^2 \mid k \neq l\} \\
(-\infty, \ldots, -\infty) & \text{otherwise}
\end{cases}
\end{align*}
\]
It is easy to see that for an accepting run of $A$ on a tree $t$, projecting on each of the first $M + 1$ entries yields $M$ distinct accepting runs of $A_1$ and one accepting run of $A_2$ on $t$, and that transition weights are preserved by this projection.

Furthermore, for $M$ distinct accepting runs $r_1, \ldots, r_M$ of $A_1$ and one accepting run $r_{M+1}$ of $A_2$ on a tree $t$, we can construct a mapping $R: \text{pos}(t) \to P\{1, \ldots, M\}^2$ such that $(r_1, \ldots, r_{M+1}, R)$ is an accepting run of $A$ on $t$ with $\text{wt}_{A}(t, (r_1, \ldots, r_{M+1}, R)) = (\text{wt}_{A_1}(t, r_1), \ldots, \text{wt}_{A_1}(t, r_M), \text{wt}_{A_2}(t, r_{M+1}))$.

Let $A = (Q, \Gamma, \mu, \nu)$ be the automaton from Lemma 6 and let $d_1, \ldots, d_p$ be an enumeration of $\Delta_A$. We define a matrix $\Omega \in \mathbb{Q}^{(M+1) \times D}$ by $\Omega = (\mu(d_1), \ldots, \mu(d_D))$ where every vector $\mu(d_i)$ is considered to be a column vector. Furthermore, for a run $r$ of $A$ on a tree $t$, we define the transition Parikh vector of $r$ by

$$p(t, r) = \{(w \in \text{pos}(t) \mid t(t, r, w) = d_1), \ldots, (w \in \text{pos}(t) \mid t(t, r, w) = d_D)\}.$$ 

In the following Lemma, we show that multiplying $\Omega$ with every possible transition Parikh vector of $A$ yields precisely $\mathbb{Q}$.

**Lemma 7.** We have $\mathbb{Q} = \{\Omega \cdot p(t, r) \mid t \in T_R, r \in \text{Acc}_A(t)\}$.

**Proof.** Let $\bar{v} \in \mathbb{Q}$, then by assumption on $A$, there exists a tree $t \in T_R$ and a run $r \in \text{Acc}_A(t)$ with $\bar{v} = \text{wt}_A(t, r)$. By definition of $\text{wt}_A$ and the commutativity of “+”, it follows that $\text{wt}_A(t, r) = \Omega \cdot p(t, r)$.

On the other hand, let $t \in T_R$ and $r \in \text{Acc}_A(t)$. Then with the same arguments and our assumption on $A$, we have $\Omega \cdot p(t, r) = \text{wt}_A(t, r) = \mathbb{Q}$.

Next, we construct a context-free language whose Parikh image coincides with the set of possible transition Parikh vectors of $A$.

**Lemma 8.** There exists a context-free language $L$ over the alphabet $\Delta_A$ such that $p(L) = \{p(t, r) \mid t \in T_R, r \in \text{Acc}_A(t)\}$. A context-free grammar $G$ generating $L$ can be found effectively from $A$.

**Proof.** We define a context-free grammar $G = (Q \cup \{S\}, \Delta_A, P, S)$, where $S$ is a new symbol, by

$$P = \{p \to (p_1, \ldots, p_m, a, p)p_1 \ldots p_m \mid \mu(p_1, \ldots, p_m, a, p) \in \mathbb{Q}^{M+1}\}$$

and

$$\cup \{S \to p \mid \nu(p) \in \mathbb{Q}^{M+1}\}.$$ 

Then $L = \mathcal{L}(G)$ is context-free and we see as follows that $p(L) = \{p(t, r) \mid t \in T_R, r \in \text{Acc}_A(t)\}$.

“$\subseteq$” Let $w \in L$. We construct a tree $t \in T_R$ and a run $r \in \text{Acc}_A(t)$ such that $p(w) = p(t, r)$. Since $w \in L$, we find words $u_1, \ldots, u_n \in (Q \cup \Delta_A)^*$ such that $u_n = w$ and $S \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \ldots \Rightarrow u_n$. We construct by induction for every $i \in \{1, \ldots, n\}$ a $\Gamma$-context $t_i \in T_{\Delta_A}$ and a run $r_i \in \text{Run}_A(t_i)$ such that $\nu(r_i(\varepsilon)) \in \mathbb{Q}^{M+1}$ and for every $p \in Q$ and $d \in \Delta_A$ we have

$$|u_i|_p = \{|v \in \text{pos}(t) \mid t_i(v) = \diamond \text{ and } r_i(v) = p\}$$

and

$$|u_i|_d = \{|v \in \text{pos}(t) \mid t(t, r, w) = d\}.$$ 

for $i = 1$, we know the definition of $G$ that $u_1 = p$ with $\nu(p) \in \mathbb{Q}^{M+1}$, so we let $t_1 = \diamond$ and $r_1(\varepsilon) = p$. Now assume we have constructed $t_i$ and $r_i$ with the properties above. We have $u_i \Rightarrow_G u_{i+1}$, so by definition of $G$, there exists a transition $d = (p_1, \ldots, p_m, a, p) \in \Delta_A$ with $\mu(d) \in \mathbb{Q}^{M+1}$ and words $u', u'' \in (Q \cup \Delta_A)^*$ such that $u_i = u'p_d u''$ and $u_{i+1} = u'dp_1 \ldots p_m u''$. Thus $|u_i|_p \geq 1$, so by induction we find $v \in \text{pos}(t_i)$ with $t_i(v) = \diamond$ and $r_i(v) = p$. We let $t_{i+1} = t_i(a(\varepsilon, \ldots, a) \to v)$ and define $r_{i+1}$ by $r_{i+1}(v') = r_i(v')$ for $v' \in \text{pos}(t_i)$ and $r_{i+1}(v_j) = p_j$ for $j \in \{1, \ldots, m\}$. It is easy to check that $t_{i+1}$ and $r_{i+1}$ satisfy all of the above properties.

Since $u_n = w \in \Delta_A^*$, the $\Gamma$-context $t_n$ is actually a $\Gamma$-tree, the run $r_n \in \text{Run}_A(t_n)$ is an accepting run of $A$ on $t_n$, and we have $p(w) = p(u_n) = p(t_n, r_n)$. Thus, we have $p(L) \subseteq \{p(t, r) \mid t \in T_R, r \in \text{Acc}_A(t)\}$.

“$\supseteq$” Now let for every $t \in T_R$ and $r \in \text{Acc}_A(t)$. We construct a word $w \in L$ with $p(w) = p(t, r)$. For this, we construct by induction for every $i \in \text{pos}(t)$ words $u_1, \ldots, u_n$ such that $v \Rightarrow_G u_1 \Rightarrow_G u_2 \Rightarrow_G \ldots \Rightarrow_G u_n = u_n \in \Delta_A$, and $p(u_n) = p(t_{\{1, \ldots, i\}}, r_{\{1, \ldots, i\}})$. We proceed by a reverse induction on the length of $v$. For $|v| = \text{height}(t)$, we let $n = 1$ and $u_1 = t(t, r, v)$, then we have $v \Rightarrow_G u_1 = u_1 \in \Delta_A$, and $p(u_1) = p(t_{\{1\}}, r_{\{1\}})$. For $|v| < \text{height}(t)$, we assume that $t(t, r, v) = d = (p_1, \ldots, p_m, a, p)$ and that for every $i \in \{1, \ldots, m\}$ we have words $u_1^{(i)}, \ldots, u_n^{(i)}$ with $u_i^{(i)} \Rightarrow_G u_1^{(i)} \Rightarrow_G \ldots \Rightarrow_G u_n^{(i)} \in \Delta_A$, and $p(u_n^{(i)}) = p(t_{\{1, \ldots, i\}}, r_{\{1, \ldots, i\}})$.
Since \( r \in \text{Acc}(t) \), we have \( \mu(d) \in \mathbb{Q}^{M+1} \), so by the definition of \( G \), we have \( p \Rightarrow_G dp_1 \ldots p_m \). Thus, we see that

\[
\begin{align*}
p & \Rightarrow_G dp_1 \ldots p_m \\
& \Rightarrow_G du^{(1)}_1 p_2 \ldots p_m \Rightarrow_G \ldots \Rightarrow_G du^{(1)}_{n_1} p_2 \ldots p_m \\
& \Rightarrow_G du^{(1)}_{n_1} u^{(2)}_1 p_3 \ldots p_m \Rightarrow_G \ldots \Rightarrow_G du^{(1)}_{n_1} u^{(2)}_{n_2} p_3 \ldots p_m \\
& \vdots \\
& \Rightarrow_G du^{(1)}_{n_1} \ldots u^{(m-1)}_{n_{m-1}} u^{(m)}_1 \Rightarrow_G \ldots \Rightarrow_G du^{(1)}_{n_1} \ldots u^{(m)}_m.
\end{align*}
\]

From this, we obtain words \( u_1, \ldots, u_n \in (\mathbb{Q} \cup \Delta_A)^* \) with \( p \Rightarrow_G u_1 \Rightarrow_G \ldots \Rightarrow_G u_n \) such that \( u_n = du^{(1)}_{n_1} \ldots u^{(m)}_{n_m} \in \Delta_A \), and therefore \( p(u_n) = p(d) + \sum_{i=1}^{m} p(u^{(i)}_i) = p(d) + \sum_{i=1}^{m} p(t_{i|v}, r_{i|v}) = p(t_{i|v}, r_{i|v}) \).

For \( v = \varepsilon \), we thus obtain words \( u_1, \ldots, u_n \) such that \( r(\varepsilon) \Rightarrow_G u_1 \Rightarrow_G \ldots \Rightarrow_G u_n, u_n \in \Delta_A^* \), and \( p(u_n) \equiv p(t, r) \). Due to \( r \in \text{Acc}_A(t) \), we have \( r(\varepsilon) \in \mathbb{Q}^{(M+1)} \), which means that \( S \Rightarrow_G r(\varepsilon) \). Therefore \( u_n \in L \), which shows that \( p(L) \supseteq \{ p(t, r) \mid t \in T, r \in \text{Acc}_A(t) \} \).

Finally, we recall Parikh’s theorem, after which we are ready to conclude the proof of Theorem 2.

**Theorem 9** ([19, Theorem 2],[6]). *For every context-free language \( L \), the set \( p(L) \) is semilinear. Furthermore, indices \( k, k_1, \ldots, k_k \) and vectors \( \alpha^{(i)}, \beta^{(i)}_j \in \mathbb{N}_0^k \) \((i \in \{1, \ldots, k\}, j \in \{1, \ldots, k_i\})\) with

\[
p(L) = \bigcup_{i=1}^{k} \{ \alpha^{(i)} + \sum_{j=1}^{k_i} n_j \cdot \beta^{(i)}_j \mid n_1, \ldots, n_{k_i} \in \mathbb{N}_0 \}
\]

can be effectively found from every context-free grammar generating \( L \).*

**Proof of Theorem 2.** Let \( L \) be as in Lemma 8. By Lemma 7 and Lemma 8, we then have \( \emptyset = \{ \Omega \cdot p(t, r) \mid t \in T, r \in \text{Acc}_A(t) \} = \{ \Omega \cdot \bar{v} \mid \bar{v} \in p(L) \} \).

For \( L \), let \( k, k_1, \ldots, k_k, \alpha^{(i)}, \beta^{(i)}_j \in \mathbb{N}_0^k \) \((i \in \{1, \ldots, k\}, j \in \{1, \ldots, k_i\})\) be as in Theorem 9. Then

\[
\emptyset = \bigcup_{i=1}^{k} \{ \Omega \cdot \alpha^{(i)} + \sum_{j=1}^{k_i} n_j \cdot \beta^{(i)}_j \mid n_1, \ldots, n_{k_i} \in \mathbb{N}_0 \}.
\]

Let \( \bar{\omega}_1, \ldots, \bar{\omega}_{M+1} \) be the rows of \( \Omega \). Then by Proposition 5, \( A_1 \) does not dominate \( A_2 \) iff there exist \( i \in \{1, \ldots, k\} \) and \( n_1, \ldots, n_{k_i} \in \mathbb{N}_0 \) with

\[
\bar{\omega}_l \cdot \alpha^{(i)} + \sum_{j=1}^{k_i} (\bar{\omega}_l \cdot \beta^{(i)}_j) \cdot n_j < \bar{\omega}_{M+1} \cdot \alpha^{(i)} + \sum_{j=1}^{k_i} (\bar{\omega}_{M+1} \cdot \beta^{(i)}_j) \cdot n_j
\]

for every \( l \in \{1, \ldots, M\} \). In other words, for every \( i \in \{1, \ldots, k\} \) we have a system of linear inequalities

\[
\bar{\omega}_l \cdot \alpha^{(i)} + \sum_{j=1}^{k_i} (\bar{\omega}_l \cdot \beta^{(i)}_j) \cdot X_j < \bar{\omega}_{M+1} \cdot \alpha^{(i)} + \sum_{j=1}^{k_i} (\bar{\omega}_{M+1} \cdot \beta^{(i)}_j) \cdot X_j \quad (l = 1, \ldots, M)
\]

\[
0 \leq X_j \quad (j = 1, \ldots, k_i),
\]

and \( A_1 \) does not dominate \( A_2 \) iff one of these systems possesses an integer solution. The first \( M \) inequalities of each system form a system of the form \( A'X < \bar{b} \) for a matrix \( A' \). The satisfiability of this system with a non-negative integer solution is equivalent to that of the system \( A'X \leq -\bar{b} \) since every non-negative integer solution \( \bar{x} \) of the first can be inflated by a sufficiently large integer \( C \) to a solution \( C'X \) of the latter system. Thus, we effectively need to check the satisfiability of systems of the form \( A'X \leq \bar{b} \) for a matrix \( A' \) and a vector \( \bar{b} \), both with entries from \( \mathbb{Q} \), with an integer solution. By [2, Theorem 3.4], the satisfiability of such systems with an integer solution is decidable, so we can decide whether \( A_1 \) dominates \( A_2 \) or not. \( \square \)
4 The Unambiguity Problem

The unambiguity problem asks whether for a given max-plus-WTA $A$, there exists an unambiguous max-plus-WTA $A'$ such that $[A] = [A']$. In this section, we show that the unambiguity problem is decidable for finitely ambiguous max-plus-WTA. We follow ideas from [14, Section 5], where the decidability of this problem was shown for finitely ambiguous max-plus word automata. The unambiguity problem is in fact known to be decidable even for polynomially ambiguous max-plus word automata [13]. 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 $A$, it is decidable whether there exists an unambiguous max-plus-WTA $A'$ with $[A] = [A']$. If $A'$ exists, it can be effectively constructed.

The rest of this section is dedicated to the proof of Theorem 10. In the following, we will employ the concept of an $A$-circuit of a WTA $A$. For a WTA $A = (Q, \Gamma, \mu, \nu)$, a $\Gamma$-word $s \in T_\Gamma$, and a run $r \in \text{Run}_A(s)$ with $r(\varepsilon) = r(\Diamond_1(s))$, the pair $(s, r)$ is called an $A$-circuit. We call $(s, r)$ small if $|s| \leq 2|Q|$.

Now let $A$ be a finitely ambiguous max-plus-WTA. We decompose $A$ into unambiguous max-plus-WTA as follows.

**Lemma 11.** Let $A$ be a finitely ambiguous max-plus-WTA over $\Gamma$, then there exist finitely many unambiguous max-plus-WTA $A_1, \ldots, A_M$ over $\Gamma$ with $[A] = \max_{i=1}^M [A_i]$ and $\text{supp}(A_1) = \ldots = \text{supp}(A_M)$.

**Proof.** By [20, Theorem 1] we can find finitely many unambiguous max-plus-WTA $A_1, \ldots, A_M$ over $\Gamma$ with $[A] = \max_{i=1}^M [A_i]$. We write $A_i = (Q_i, \Gamma_i, \mu_i, \nu_i)$. Let $L = \bigcup_{i=1}^M \text{supp}(A_i)$ and let $\kappa$ be the smallest weight used in the automata $A_1, \ldots, A_M$, i.e., for $R = \bigcup_{i=1}^M (\mu_i(\Delta_{A_i}) \cup \nu_i(Q_i))$ we let $\kappa = \min(R \setminus \{-\infty\})$.

The language $L$ is recognizable, therefore for $i \in \{1, \ldots, M\}$, the language $L_i = L \setminus \text{supp}(A_i)$ is also recognizable and there exists a deterministic FTA $A''_i = (Q_i', \Gamma_i', \delta_i', F_i')$ with $L(A_i) = L_i$. We define the max-plus-WTA $A''_i = (Q_i', \Gamma_i', \mu''_i, \nu''_i)$ by

$$\mu''_i(d) = \begin{cases} \kappa & \text{if } d \in \delta_i' \\ -\infty & \text{otherwise} \end{cases} \quad \text{and} \quad \nu''_i(q) = \begin{cases} \kappa & \text{if } q \in F_i' \\ -\infty & \text{otherwise} \end{cases}$$

We assume without loss of generality that $Q_i \cap Q_i' = \emptyset$ and define $A'' = (Q_i \cup Q_i', \Gamma, \mu''_1, \nu''_1)$ with

$$\mu''_i(d) = \begin{cases} \mu_i(d) & \text{if } d \in \Delta_{A_i} \\ \mu''_i(d) & \text{if } d \in \Delta_{A''_i} \\ -\infty & \text{otherwise} \end{cases}$$

as the union of $A_i$ and $A''_i$. Then $A''$ is unambiguous since $A_i$ is unambiguous, $A''_i$ is deterministic, and $\text{supp}(A_i) \cap \text{supp}(A''_i) = \emptyset$. Furthermore, for $t \in \text{supp}(A_i)$ we have $[A''_i](t) = [A_i](t)$.

For every $t \in \text{supp}(A''_i)$, there exists some $j \in \{1, \ldots, M\}$ with $t \in \text{supp}(A_j)$ and due to the choice of $\kappa$ we have $[A_j](t) \geq [A''_i](t)$. In conclusion, for all $i \in \{1, \ldots, M\}$ we have that $A''_i$ is unambiguous, $\text{supp}(A''_i) = L$, and $\max_{i=1}^M [A''_i] = [A] = [A']$. \hfill \Box

Let $A_1, \ldots, A_M$ be unambiguous max-plus-WTA with $\text{supp}(A_1) = \ldots = \text{supp}(A_M)$ and $[A] = \max_{i=1}^M [A_i]$. The product automaton $B$ of $A_1, \ldots, A_M$ is a weighted tree automaton over the product semiring $(\mathbb{R}_\max)^M$ which, intuitively, executes all of the automata $A_1, \ldots, A_M$ in parallel. We write $A_i = (Q_i, \Gamma_i, \mu_i, \nu_i)$ for $i \in \{1, \ldots, M\}$ and define $B = (Q, \Gamma, \mu, \nu)$ as follows. We let $Q = Q_1 \times \ldots \times Q_M$ and for $a \in \Gamma$ with $\text{rk}_\Gamma(a) = m$ and $p_0, \ldots, p_m \in Q$ with $p_i = (p_{i1}, \ldots, p_{im})$ we define, with $x_j = \mu_j(p_{i1}, \ldots, p_{im}, a, p_{i0})$ and $y_j = \nu_j(p_{i0})$,

$$\mu(p_0, \ldots, p_m, a, p_0) = \begin{cases} (x_1, \ldots, x_M) & \text{if } (x_1, \ldots, x_M) \in \mathbb{R}_\max^M \\ (-\infty, \ldots, -\infty) & \text{otherwise} \end{cases}$$

$$\nu(p_0) = \begin{cases} (y_1, \ldots, y_M) & \text{if } (y_1, \ldots, y_M) \in \mathbb{R}_\max^M \\ (-\infty, \ldots, -\infty) & \text{otherwise} \end{cases}$$

Then $B$ is unambiguous and for $t \in T_\Gamma$ we have $[B](t) = ([A_1](t), \ldots, [A_M](t))$. We may assume that $B$ is trim.
Definition 12 (Victorious coordinate). Let $s \in T_r$ be a $\Gamma$-context, $r \in \text{Run}^\gamma_\mathcal{B}(s)$, and write $\text{wt}(s, r) = (\kappa_1, \ldots, \kappa_M).$ We define $\text{wt}(s, r) = \kappa_i$ and $\text{wt}(s, r) = \max_{i=1}^M \text{wt}_i(s, r)$.

A coordinate $i \in \{1, \ldots, M\}$ 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 $\text{Vct}(s, r)$. For $q \in Q$ we define

$$\text{Vct}([q]) = \bigcap_{(s, r) \text{ small } \mathcal{B}\text{-circuit}} \text{Vct}(s, r)$$

where the empty intersection is defined as $\{1, \ldots, M\}$. For $P \subseteq Q$, we let $\text{Vct}(P) = \bigcap_{p \in P} \text{Vct}([p])$.

We have the following lemma which relates victorious coordinates to the decidability of the unambiguity problem.

Lemma 13. There is an unambiguous max-plus-WTA $\mathcal{A}' = (Q', \Gamma, \mu', \nu')$ with $[\mathcal{A}] = [\mathcal{A}']$ if and only if for all $t \in T_r$ and all $r \in \text{Acc}_\mathcal{B}(t)$ we have $\text{Vct}(r(\text{pos}(t))) \neq \emptyset$. The latter property is called the dominance property and is denoted by $(P)$. The dominance property is decidable, and therefore so is the unambiguity problem.

Proof. Here, we only show that $(P)$ is decidable. We defer the proof that $(P)$ is necessary to Lemma 14. The proof for the sufficiency of $(P)$ takes some more preparation and is split into several lemmata.

$(P)$ 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 $t \in T_r$ 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 exists $t \in T_r$ and $r \in \text{Acc}_\mathcal{B}(t)$ for $P \subseteq r(\text{pos}(t))$. Checking whether all $P$ for which this is true satisfy $\text{Vct}(P) \neq \emptyset$ is equivalent to checking $(P)$. Note that $\text{Vct}(P)$ can be effectively computed since there are only finitely many small $\mathcal{B}$-circuits.

First, we prove that $(P)$ is a necessary condition, i.e., that from the existence of an unambiguous automaton $\mathcal{A}'$ with $[\mathcal{A}] = [\mathcal{A}']$ it follows that $\mathcal{B}$ satisfies $(P)$.

Lemma 14. If there exists an unambiguous max-plus-WTA $\mathcal{A}' = (Q', \Gamma, \mu', \nu')$ with $[\mathcal{A}] = [\mathcal{A}']$ then $\mathcal{B}$ satisfies $(P)$.

Proof. We proceed by contradiction and assume that $\mathcal{A}'$ as above exists and that $(P)$ is not satisfied. Then there exists a tree $t \in T_r$ and a run $r \in \text{Acc}_\mathcal{B}(t)$ with $\text{Vct}(r(\text{pos}(t))) = \emptyset$. We let $\mathcal{C}$ be the set of all small circuits which are relevant to show this, i.e., $\mathcal{C} = \{(s, r_s) : \text{small } \mathcal{B}\text{-circuit } | [r_s(\varepsilon)] \cap r(\text{pos}(t)) \neq \emptyset\}$. Let $(s, r_s) \in \mathcal{C}$ and $q = r_s(\varepsilon)$.

We may assume that $q \in r(\text{pos}(t))$ due to the following argument. If $q \in r(\text{pos}(t))$ does not hold, there exists some $p \in r(\text{pos}(t))$ with $p \approx q$. Then there exist $\Gamma$-words $s_p^q, s_p^q \in T_r$ and runs $r_p^q = \text{Run}_\mathcal{B}(s_p^q, s_p^q, p)$ and $r_p^q \in \text{Run}_\mathcal{B}(s_p^q, s_p^q, q)$. Thus, with $s' = s_p^q(s_p^q)$ and $r' = r_p^q(s_p^q)$, we obtain a circuit $(s', r_p^q) = (s, r_s)$ by $r'(q) = s_p^q$ and $r'(q) = q$. We can insert $(s', r_p^q)$ into $t$ and $r$ to obtain a tree $t'$ and a run $r' \in \text{Acc}_\mathcal{B}(t')$ with $q \in r'(\text{pos}(t'))$.

Now let $w_q \in p(\text{pos}(t))$ with $r(w_q) = q$ and $w = \bigcap (s)$. Then $(s_i, r_i, q_i, s_i, q_i(w_i))$ is a circuit which we can insert at $w_q$. We do this for all small circuits in $C$ simultaneously and obtain a tree $t' \in T_r$ together with a run $r' \in \text{Acc}_\mathcal{B}(t')$. We assume without loss of generality that after this, the circuit $(s_i, r_i, q_i(w_i))$ is still at position $w_q$. Since supp $\mathcal{B}$ = 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 $r_{i_2}'(w_q s_{i_2}^{-1} t') = r_{i_1}'(w_q s_{i_1}^{-1} t')$. From this, we obtain an $\mathcal{A}'$-circuit $(s_{i_2}^{-1} r_{i_1}, \ddot{r})$ which corresponds to a $\mathcal{B}$-circuit $(s_{i_2}^{-1} r_{i_1}, \ddot{r})$. We can now insert $s_{i_2}^{-1} s_{i_1}$ at $w_q$ repeatedly to create copies of these circuits. Clearly, this works for all small circuits in $C$.

Let $c_1, \ldots, c_n$ be an enumeration of $\mathcal{C}$. By $(s_1, r_1)$ and $(s_i, r_i)$, we denote the circuits in $\mathcal{A}'$ and $\mathcal{B}$, respectively, we obtain from $c_1$ in the way we obtained $(s_{i_1}^{-1} r_{i_1})$ and $(s_{i_1}^{-1} r_{i_1}(w_i))$ from $(s, r_s)$. Clearly, $\text{Vct}(c_i) = \text{Vct}(s_i, r_i)$ for all $i$. For $v = (v_1, \ldots, v_n) \in \mathbb{N}_0^n$, we denote the tree obtained by adding $v_i$ copies of $s_i$ to $t'$ for each $i \in \{1, \ldots, n\}$.

By assumption, there exists $I \in \{1, \ldots, n\}$ such that $\bigcap_{i=1}^I \text{Vct}(c_i) \neq \emptyset$ and $\bigcap_{i=1}^{I+1} \text{Vct}(c_i) = \emptyset$. In the following, we show that $\bigcap_{i=1}^{I+1} \text{Vct}(c_i) \neq \emptyset$, which yields the desired contradiction. For $k, l \in \mathbb{N}_0$, let $t_{k,l} = t_{(k,\ldots,k,l,0,\ldots,0)}$, where $l$ is at index $I + 1$. Since $\mathcal{A}'$ is unambiguous, we see that with $x = [\mathcal{A}'](t_{0,0})$, $\kappa = \text{wt}_{\mathcal{A}'}(s_1, r_1) + \ldots + \text{wt}_{\mathcal{A}'}(s_I, r_I)$, and $\lambda = \text{wt}_{\mathcal{A}'}(s_{I+1}, r_{I+1})$ we have

$$[\mathcal{A}'](t_{k,l}) = x + \kappa k + l \lambda$$

for all $k, l \in \mathbb{N}_0$.  

9
Due to the definition of victorious coordinates, we can find a number \( N \in \mathbb{N}_0 \) such that for all \( l \geq N \), the tuple \([\mathcal{B}](t_{0,l})\) has its maximum in some victorious coordinate from \( \text{Vict}(c_{l+1}) \); this is because with every repetition of a circuit, non-victorious coordinates fall behind victorious coordinates in terms of weight by a small fixed margin. Then for every \( l' \geq 0 \), we have

\[
[\mathcal{A}](t_{0,N+l'}) = [\mathcal{A}](t_{0,N}) + l' \cdot \text{wt}(s_{l+1}, r^B_{l+1}).
\]

Since \([\mathcal{A}'] = [\mathcal{A}]\), it follows that

\[
\text{wt}(s_{l+1}, r^B_{l+1}) = [\mathcal{A}](t_{0,N+1}) - [\mathcal{A}](t_{0,N}) = x + (N + 1)\lambda - (x + N\lambda) = \lambda.
\]

Similarly, due to the assumption that \( \bigcap_{l=1}^N \text{Vict}(c_l) \neq \emptyset \), we can find for every \( l \in \mathbb{N}_0 \) a number \( M_l \in \mathbb{N}_0 \) such that for all \( k \geq M_l \), the tuple \([\mathcal{B}](t_{k,l})\) has its maximum in some victorious coordinate \( j_l \in \bigcap_{l=1}^{k+1} \text{Vict}(c_l) \). We let \( M = \max_{l=0}^M M_l \). By pigeon hole principle, there exist \( l_1, l_2 \in \{0, \ldots, M\} \) with \( l_1 < l_2 \) and \( j_l = j_{l_2} \). Then we see that, again due to \([\mathcal{A}'] = [\mathcal{A}]\), we have

\[
(l_2 - l_1)\text{wt}_{j_{l_1}}(s_{l+1}, r^B_{l+1}) = [\mathcal{A}](t_{M,l_2}) - [\mathcal{A}](t_{M,l_1}) = (l_2 - l_1)\lambda.
\]

It follows that

\[
\text{wt}(s_{l+1}, r^B_{l+1}) = \lambda = \text{wt}_{j_{l_1}}(s_{l+1}, r^B_{l+1}),
\]

which means that \( j_{l_1} \in \text{Vict}(c_{l+1}) \). Since \( j_{l_1} \in \bigcap_{l=1}^{l_1+1} \text{Vict}(c_l) \) also holds, we have \( \bigcap_{l=1}^{l+1} \text{Vict}(c_l) \neq \emptyset \), which is a contradiction to the choice of \( I \). In conclusion, \( t \) and \( r \) as chosen do not exist and therefore \( \mathcal{B} \) satisfies \((P)\).

We now come to the sufficiency of \((P)\). In the following, we assume that \( \mathcal{B} \) satisfies \((P)\) and construct an unambiguous max-plus-WTA \( \mathcal{A}' \) with \([\mathcal{A}'] = [\mathcal{A}]\).

The idea behind \( \mathcal{A}' \) is as follows. The states of \( \mathcal{A}' \) will be taken from \( \mathbb{R}^{\max}_M \times Q \). From a bottom-up perspective, \( \mathcal{A}' \) remembers in each \( \mathbb{R}^{\max}_\text{max} \)-coordinate 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 a certain bound \((2N + 1)C\), the coordinates with small weights are discarded by being set to \(-\infty\), and only the large weights are remembered. Here, \( N \) is the maximum possible number of nodes of a tree over \( \Gamma \) of height at most \( 2|Q| \). The constant \( C \) is the largest difference between all weights occurring in the automata \( A_1, \ldots, A_M \).

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

Formally, we define \( \mathcal{A}' \) as follows.

**Construction 15.** We let

\[
N = \sum_{i=0}^{2|Q|} \text{rk}(\Gamma)^i = \max\{\text{pos}(t) \mid t \in \mathcal{T}_\Gamma, \text{height}(t) \leq 2|Q|\}
\]

\[
R = \bigcup_{i=1}^M (\mu_i(\Delta_{\mathcal{A}}) \cup \nu_i(\mathcal{Q}))
\]

\[
C = \max R - \min(R \setminus \{-\infty\}).
\]

For \( x = (x_1, \ldots, x_M) \in \mathbb{R}^{\max}_M \), we denote the smallest weight of \( x \) by

\[
\bar{x} = \min\{x_i \mid 1 \leq i \leq M, x_i \neq -\infty\},
\]

and define the normalization of \( x \) by

\[
\bar{x} = x - (\bar{x}, \ldots, \bar{x}).
\]

We construct an unambiguous max-plus-WTA \( \mathcal{A}' = (Q', \Gamma, \mu', \nu') \) with \([\mathcal{A}'] = [\mathcal{A}]\) and \( Q' \subset \mathbb{R}^{\max}_M \times Q \) as follows.
Lemma 17. The projection later on. The following is clear from the construction. Rule 3 Assume we have \( l \) know by Rule 2 that \( z \) and \( y \).

We let \( \langle \mathbf{y}, p_0 \rangle \in Q' \) and \( \nu'((z_1, p_1), \ldots, (z_m, p_m), a, \langle \mathbf{y}, p_0 \rangle) = \mathbf{y} \).

Rule 3 Assume \( (z, p) \in Q' \) and \( x = \nu(p) \in \mathbb{R}^M \). Then we let \( \nu'(z, p) = \max_{i=1}^{M} (z_i + x_i) \).

Note that from the above definition, it is not obvious that \( Q' \) is finite, which is what we will show later on. The following is clear from the construction.

Proposition 16. The projection \( \pi: Q' \to Q, (z, p) \mapsto p \) induces a bijection between the accepting runs of \( \mathcal{B} \) and \( \mathcal{A}' \). In particular, \( \mathcal{A}' \) is unambiguous. Using a simple induction we can show the following relationship between the runs of \( \mathcal{B} \) and \( \mathcal{A}' \).

Lemma 17. Let \( t \in T_\mathcal{G}, r \in \text{Run}_\mathcal{G}(t) \), and \( r' = \pi^{-1}(r) \in \text{Run}_{\mathcal{A}'}(t) \). We write \( r'(z) = (z, p) \), then for every \( l \in \{1, \ldots, M\} \) we have

(i) if \( z_l \neq -\infty \) then \( w_{t_l}(t, r) = w_{\mathcal{A}'}(t, r') + z_l \)

(ii) if \( z_l = -\infty \) then for some \( w \in \text{pos}(t) \) we have \( w_{t_l}(t_1, r_1) < w_{\mathcal{A}'}(t_1, r_1) - (2N + 1)C \).

Proof. (i) We proceed by induction on the height of \( t \). If height(\( t \)) = 0, the statement follows from Rule 1. Otherwise, we let \( a = t(\varepsilon), m = \text{rk}(a) \), and \( r'(i) = (z_i, p_i) \) for \( i \in \{1, \ldots, m\} \). Since \( z_i \neq -\infty \), we know by Rule 2 that \( z_i \neq -\infty \) holds for all \( i \in \{1, \ldots, m\} \). Thus, by induction we have \( w_{t_l}(t_1, r_1) = w_{\mathcal{A}'}(t_1, r_1) + z_l \) for all \( i \in \{1, \ldots, m\} \). It follows by Rule 2 that with \( x = \mu(p_1, \ldots, p_m, a, p) \) and \( y = \mu'((z_1, p_1), \ldots, (z_m, p_m), a, (z, p)) \) we have

\[
\begin{align*}
wt_l(t, r) &= x_l + \sum_{i=1}^{m} wt_l(t_i, r_i) \\
&= x_l + \sum_{i=1}^{m} (wt_{\mathcal{A}'}(t_i, r'_i) + z_l) \\
&= z_l + y + \sum_{i=1}^{m} wt_{\mathcal{A}'}(t_i, r'_i) \\
&= wt_{\mathcal{A}'}(t, r') + z_l.
\end{align*}
\]

(ii) Assume \( z = -\infty \) and let \( w \) be a prefix-maximal position with the property that for \( (z', p') = r'(w) \) we have \( z'_l = -\infty \). By Rule 1, \( w \) cannot be a leaf. We let \( a = t(w), m = \text{rk}(a), r'(w) = (z_0, p_0) \), and \( r'(w) \) for \( i \in \{1, \ldots, m\} \). By choice of \( w \), we have \( z_i \neq -\infty \) for all \( i \in \{1, \ldots, m\} \), so by (i) we have \( wt_l(t'_w, r'_w) = wt_{\mathcal{A}'}(t'_w, r'_w) + z_l \) for all \( i \in \{1, \ldots, m\} \). Let \( x = \mu(p_1, \ldots, p_m, a, p) \) and \( y = \mu'((z_1, p_1), \ldots, (z_m, p_m), a, (z_0, p_0)) \). Then since \( z_0 \neq -\infty \), there exists by Rule 2 some \( j \in \{1, \ldots, M\} \) such that \( z_0_j \neq -\infty \) and \( x_l + \sum_{i=1}^{m} z_l < z_0 + y - (2N + 1)C \). Thereby, we have

\[
\begin{align*}
wt_l(t'_w, r'_w) &= x_l + \sum_{i=1}^{m} wt_l(t'_w, r'_w) \\
&= x_l + \sum_{i=1}^{m} (wt_{\mathcal{A}'}(t'_w, r'_w) + z_l) \\
&< z_0 + y + \sum_{i=1}^{m} wt_{\mathcal{A}'}(t'_w, r'_w) - (2N + 1)C \\
&= wt_{\mathcal{A}'}(t'_w, r'_w) - (2N + 1)C \\
&\leq wt(t'_w, r'_w) - (2N + 1)C.
\end{align*}
\]
The dominance property (P) is defined only through small circuits. Thus, in order to use (P), we describe in the following how to decompose pairs \((t, r)\) of a \(\Gamma\)-tree or a \(\Gamma\)-word \(t\) and a run \(r\) of \(B\) on \(t\) into small circuits. Intuitively, we cut circuits from the bottom of the tree using the pigeon hole principle.

**Construction 18.** Let \(t \in T_{\Gamma} \) be a \(\Gamma\)-tree or a \(\Gamma\)-word and \(r \in \text{Run}_B(t)\). A circuit decomposition of \(t\) and \(r\) is a stub \((t_0, r_0)\), where \(t_0 \in T_{\Gamma}\) with \(\text{height}(t_0) \leq 2|Q|\) and \(r_0 \in \text{Run}_B(t_0)\), together with a finite sequence of small \(B\)-circuits \((s_1, r_1), \ldots, (s_n, r_n)\) defined as follows. If \(\text{height}(t) \leq 2|Q|\), then we let \(t_0 = t\) and \(r_0 = r\) and conclude the decomposition. Otherwise, we cut a small circuit from \(t\) and \(r\).

If \(t\) is a \(\Gamma\)-tree, we proceed as follows. We choose \(uw \in \text{pos}(t)\) with \(|uw| = \text{height}(t)\) and \(|v| = |Q|\). By pigeon hole principle, we find \(u \leq_p w_1 <_p w_2 \leq_p uw\) with \(r(w_1) = r(w_2)\). We let \(s = (t(\emptyset \rightarrow w_2))|_{w_1}\), then for \(w \in \text{pos}(s)\) we see that

\[
\text{height}(t) \geq |w_1|w = |w_1| + |w| \geq |u| + |w| = \text{height}(t) - |Q| + |w|
\]

from which \(|w| \leq |Q|\) and therefore \(\text{height}(s) \leq |Q|\) follows. Thus from \(r\) we obtain a small circuit \((s, r'')\) through \(r''(w) = r(w_1w)\) for \(w \in \text{pos}(s)\). With \(t' = t|_{w_2} \rightarrow w_1\) we obtain from \(r\) a run \(r' \in \text{Run}_B(t')\) through \(r' = r|_{w_2} \rightarrow w_1\). We continue the decomposition with \(t'\) and \(r'\). This procedure ends after finitely many steps.

If \(t\) is a \(\Gamma\)-word, we proceed in the following way in order to ensure that the process above never creates a \(2\-\Gamma\)-context when cutting a circuit. If there exists a position \(v' \in \text{pos}(t)\) which is prefix-independent from \(\tilde{\jmath}(t)\) and for which \(\text{height}(t|_{v'}) \geq |Q|\), we let \(wv \in \text{pos}(t)\) with \(v' \leq_p wv, \ |wv| = |v'| + \text{height}(t|_{v'})\), and \(|v| = |Q|\). By pigeon hole principle, we find \(u \leq_p w_1 <_p w_2 \leq_p uvw\) with \(r(w_1) = r(w_2)\). We let \(s = (t(\emptyset \rightarrow w_2))|_{w_1}\), then for \(w \in \text{pos}(s)\) we see that

\[
|v'| + \text{height}(t|_{v'}) \geq |w_1|w = |w_1| + |w| \geq |u| + |w| = |v'| + \text{height}(t|_{v'}) - |Q| + |w|
\]

from which \(|w| \leq |Q|\) and therefore \(\text{height}(s) \leq |Q|\) follows. Thus from \(r\) we obtain a small circuit \((s, r'')\) through \(r''(w) = r(w_1w)\) for \(w \in \text{pos}(s)\). With \(t' = t|_{w_2} \rightarrow w_1\) we obtain from \(r\) a run \(r' \in \text{Run}_B(t')\) through \(r' = r|_{w_2} \rightarrow w_1\). Note that both \(s\) and \(t'\) are \(\Gamma\)-words since \(v'\) is prefix-independent from \(\tilde{\jmath}(t)\). We continue the decomposition with \(t'\) and \(r'\).

If \(t\) is a \(\Gamma\)-word but \(\text{height}(t|_{v'}) < |Q|\) for all \(v' \in \text{pos}(t)\) which are prefix-independent from \(\tilde{\jmath}(t)\), we proceed as follows. First, we show that \(|\tilde{\jmath}(t)| \geq |Q|\). We know that \(\text{height}(t) > 2|Q|\), thus there exists a position \(w \in \text{pos}(t)\) with \(|w| \geq 2|Q|\). If \(w \leq_p \tilde{\jmath}(t)\), it immediately follows that \(|\tilde{\jmath}(t)| \geq |Q|\). Otherwise, since \(\tilde{\jmath}(t)\) is a leaf, \(w\) and \(\tilde{\jmath}(t)\) are prefix-independent and we can write \(w = v_i v_i\) and \(\tilde{\jmath}(t) = v_j v_j\) for some \(i, j \in \mathbb{N}_+\) with \(i \neq j\). As \(w\) is prefix-independent from \(\tilde{\jmath}(t)\), we see that \(|v_i| \leq \text{height}(t|_{v_i}) < |Q|\) and therefore \(|v_i| \geq 2|Q| - |Q| = |Q|\). In particular, we have \(|\tilde{\jmath}(t)| \geq |v_i| \geq |Q|\).

Since \(|\tilde{\jmath}(t)| \geq |Q|\), we can write \(\tilde{\jmath}(t) = uw\) with \(|v| = |Q|\). By pigeon hole principle, we find \(u \leq_p w_1 <_p w_2 \leq_p uw\) with \(r(w_1) = r(w_2)\). We let \(s = (t(\emptyset \rightarrow w_2))|_{w_1}\) and show that \(\text{height}(s) \leq |Q|\). Let \(w \in \text{pos}(s)\). If \(uw \leq_p \tilde{\jmath}(t)\), we have \(|w| \leq |v| = |Q|\). Otherwise, if \(uw\) is prefix-independent from \(\tilde{\jmath}(t)\), we can write \(uw = uw' v_i\) and \(\tilde{\jmath}(t) = uw' v_j\) for some \(i, j \in \mathbb{N}_+\) with \(i \neq j\). Then \(uw' v_i\) is prefix-independent from \(\tilde{\jmath}(t)\) which means we have \(|v_i| \leq \text{height}(t|_{uw' v_i}) < |Q|\). Due to \(|v'| \leq |Q|\), we see that \(|v'| \leq |Q|\) and therefore \(|w| = |v'| v_i | < |Q| + 1 + |Q|\). Therefore, \(\text{height}(s) \leq 2|Q|\). Thus from \(r\) we obtain a small circuit \((s, r'')\) through \(r''(w) = r(w_1w)\) for \(w \in \text{pos}(s)\). With \(t' = t|_{w_2} \rightarrow w_1\) we obtain from \(r\) a run \(r' \in \text{Run}_B(t')\) through \(r' = r|_{w_2} \rightarrow w_1\). Note that both \(s\) and \(t'\) are \(\Gamma\)-words since \(w_2 \leq_p \tilde{\jmath}(t)\). We continue the decomposition with \(t'\) and \(r'\).

In the following coordinates, we show that the weights of victorious coordinates never become much smaller than the maximum weight over all coordinates.

**Lemma 19.** Let \(t \in T_{\Gamma} \) be a \(\Gamma\)-tree or a \(\Gamma\)-word and \(r \in \text{Run}_B(t)\). If \(k \in \text{Vic}(r(\text{pos}(t)))\) then \(\text{wt}_k(t, r) \geq \text{wt}_k(t, r) - NC\).

**Proof.** Take a circuit decomposition of \(t\) and \(r\) as in Construction 18 with stub \((t_0, r_0)\) and small circuits \((s_1, r_1), \ldots, (s_n, r_n)\). Since \(|\text{pos}(t_0)| \leq N\), we have for all \(j \in \{1, \ldots, M\}\) that

\[
\text{wt}_k(t, r) = \text{wt}_k(t_0, r_0) + \sum_{i=1}^{n} \text{wt}_k(s_i, r_i)
\]

\[
\geq \text{wt}_j(t_0, r_0) - NC + \sum_{i=1}^{n} \text{wt}_j(s_i, r_i)
\]

\[
12
\]
\[= \text{wt}_j(t, r) - NC.\]

This is true in particular for \(j\) with \(\text{wt}(t, r) = \text{wt}_j(t, r)\). \(\Box\)

We are now able to show that \(Q'\) is finite. We proceed by contradiction and show that if \(Q'\) would be infinite, we would be able to find arbitrarily long successions \((z_n, p) \preceq \ldots \preceq (z_1, p)\) in \(Q'\) with \(z_1, \ldots, z_n\) pairwise distinct. Then we show that such successions can in fact not be arbitrarily long, as from every \(z_i\) to the next, the difference in weights of at least one non-victorious coordinate to the victorious coordinate grows by at least \(\delta\), where \(\delta\) is a fixed constant. Thus, after some \(z_i\), these differences exceed \((2N + 1)C\) for all non-victorious coordinates, and all subsequent \(z_i\) are the same.

**Lemma 20.** \(Q'\) is a finite set.

Proof. We show first that if \(Q'\) is infinite, then for at least one \(p \in Q\) we can find arbitrarily long successions \((z_n, p) \preceq \ldots \preceq (z_1, p)\) with \(z_1, \ldots, z_n\) pairwise distinct. Let \(P_0 \subseteq Q'\) be the set of all states added to \(Q'\) by Rule 1. For \(i > 0\), let \(P_{i+1} \subseteq Q'\) be the set of all states added to \(Q'\) by Rule 2 using only states \((z_1, p_1), \ldots, (z_m, p_m) \in P_i\). Then for all \(i \geq 0\) we have \(P_i \subseteq P_{i+1}, P_{i+1} \setminus P_i \neq \emptyset\) since \(Q\) is infinite, and \(P_i\) is finite.

Let \(i > 0\) and \((z, p) \in P_{i+1} \setminus P_i\). Then there are \((z_1, p_1), \ldots, (z_m, p_m) \in P_i\) with at least one \((z_j, p_j) \in \tilde{P}_i \setminus P_{i-1}\) such that \((z, p)\) is added to \(Q'\) by Rule 2 using \((z_1, p_1), \ldots, (z_m, p_m) \in P_i\) and a valid transition \((p_1, \ldots, p_m, a, p) \in \Delta^p\). In particular, we have \((z, p) \preceq (z_j, p_j)\).

Now let \(H > 0\), \(n > H|Q|\) and \(p \in P_n \setminus P_{n-1}\). Then according to the argumentation we just did, we can find \((z_0, p_0) \preceq \ldots \preceq (z_0, p_0)\) with \((z_0, p_0) \in P_0\) and \((z_i, p_i) \in P_i \setminus P_{i-1}\) for \(i > 0\). In particular, \((z_0, p_0), \ldots, (z_k, p_k)\) are pairwise distinct. By pigeon hole principle, at least one \(p \in Q\) occurs \(H\) or more times among \(P_0, \ldots, P_n\). Hence, we find \(i_1 < \ldots < i_H\) with \(p_{i_1} = \ldots = p_{i_H} = p\) and have \((z_{i_H}, p) \preceq \ldots \preceq (z_1, p)\) with \(z_1, \ldots, z_{i_H}\) pairwise distinct.

Now we show that there can be no arbitrarily long successions \((z_n, p) \preceq \ldots \preceq (z_1, p)\) with \(z_1, \ldots, z_n\) pairwise distinct in \(Q'\). This shows in particular that \(Q'\) must be finite. We define the constant

\[
\delta = \min_{(s, r) \text{ small B-circuit}} \frac{\text{wt}(s, r)}{\max\{\text{wt}_i(s, r) \mid \text{wt}_i(s, r) < \text{wt}(s, r)\}}
\]

where the minimum over the empty set is defined as \(\infty\). Assume we have \((x, p) \preceq (y, p)\) with \(x \neq y\). Let \(t \in T_r\) be a tree and \(r' \in \text{Run}_\lambda(t)\) a run on \(t\) with \(r'(e) = (x, p)\) and \(r''(w) = (y, p)\) for some \(w \in \text{pos}(t)\) with \(|w| > 0\). Then for \(s = t(\phi \rightarrow w)\), the run \(r''\) yields a run \(r \in \text{Run}_\lambda^p(s)\) by projecting onto \(Q\).

Take a circuit decomposition of \(s\) and \(r\) as in Construction 18 with stub \((s_0, r_0)\) and small circuits \((s_1, r_1), \ldots, (s_n, r_n)\). Note that now, \((s_0, r_0)\) is also a small circuit. Since \(B\) satisfies \((P)\), there exists \(k \in \text{Vic}(r(\lambda(t)))\). Due to Lemma 17(ii) and Lemma 19, we have \(x_k, y_k \in \mathbb{R}\).

For all \(i \in \{0, \ldots, n\}\) and \(j \in \{1, \ldots, M\}\), we have either \(w_t(s_i, r_i) = \text{wt}_k(s_i, r_i)\) or \(w_t(s_i, r_i) \leq \text{wt}_k(s_i, r_i) - \delta\). Hence, for all \(j \in \{1, \ldots, M\}\) we have either \(x_k - y_k = y_k - y_j\) or \(x_k - y_j \geq y_k - y_j + \delta\).

Since \(x \neq y\), we have \(x_k - y_j \geq y_k - y_j + \delta\) for at least one \(j \in \{1, \ldots, M\}\).

Now let \((z_n, p) \preceq \ldots \preceq (z_1, p)\) be a succession as above with \(z_1, \ldots, z_n\) pairwise distinct. Then in every step from \(z_n\) to \(z_{n+1}\), at least one non-victorious coordinate the difference the victorious coordinates grows by at least \(\delta\). If this differences exceeds \((2N + 1)C\), the coordinate is set to \(-\infty\). Thus, at some point all non-victorious coordinates are \(-\infty\). It follows that \(n\) cannot be arbitrarily large.

Next, we show that if a coordinate yields the maximum weight in \(B\), then during the whole computation of the weight of the run, the distance to the maximum weight does not exceed the bound \((2N + 1)C\).

**Lemma 21.** Let \(t \in T_r\) and \(r \in \text{Acc}_\lambda(t)\). If for \(l \in \{1, \ldots, M\}\) we have \([A](t) = [A_l](t)\), then for all \(w \in \text{pos}(t)\) we have \(\text{wt}_l(t|w, r|_w) \geq \text{wt}(t|w, r|_w) - (2N + 1)C\).

Proof. We let \(t' = t(\phi \rightarrow w)\) and let \(r'\) be the run on \(t'\) we obtain from \(r\) through \(r'(w) = r(w)\). We write \(r(\phi) = (q_1, \ldots, q_M)\) and let \(k \in \text{Vic}(r(\lambda(t)))\). By assumption, we have

\[
\nu_l(q_l) + \text{wt}(t', r') + \text{wt}(t|w, r|_w) \geq \nu_k(q_k) + \text{wt}_k(t', r') + \text{wt}_k(t|w, r|_w).
\]

Due to Lemma 19, we have

\[
\text{wt}_l(t', r') \leq \text{wt}(t', r') + NC.
\]
Thus, applying Lemma 19 also to $\text{wt}_k(t|_w, r|_w)$ we can conclude
\[
\text{wt}_k(t|_w, r|_w) \geq \nu_k(q_l) - \nu_k(q_l) + \text{wt}_k(t', r') - \text{wt}_k(t', r') - NC + \text{wt}_k(t|_w, r|_w) \\
\geq -CN + \text{wt}(t|_w, r|_w) - NC \\
= \text{wt}(t|_w, r|_w) - (2N + 1)C.
\]

We are now ready to show that the behaviors of $A'$ and $A$ coincide.

**Lemma 22.** We have $[A] = [A']$.

**Proof.** It is clear that $\text{supp} A = \text{supp} B = \text{supp} A'$. Let $t \in \text{supp} A$ and let $r \in \text{Acc}_B(t)$ and $r' \in \text{Acc}_A(t)$ be the unique accepting runs on $t$. We write $r(\epsilon) = (z, q)$ and let $l \in \{1, \ldots, M\}$ with $[A_l](t) = \max_{i=1}^M [A_i](t)$. Combining Lemma 17(ii) and Lemma 21, we see that we have $z_l \neq -\infty$. Thus, by Lemma 17(i) we have for all $i \in \{1, \ldots, M\}$ that
\[
\nu_i(q_l) + z_i \leq \nu_i(q_l) + \text{wt}_i(t, r) - \text{wt}_A(t, r') \\
= [A_l](t) - \text{wt}_A(t, r') \\
\leq [A_i](t) - \text{wt}_A(t, r') \\
= \nu_i(q_l) + \nu_i(q_l) - \text{wt}_A(t, r') \\
= \nu_i(q_l) + z_i.
\]

Therefore, again by Lemma 17(i) we have
\[
[A](t) = \max_{i=1}^M [A_i](t) \\
= [A](t) \\
= \nu_i(q_l) + \text{wt}_i(t, r) \\
= \nu_i(q_l) + \text{wt}_A(t, r') + z_l \\
= \text{wt}_A(t, r') + \max_{i=1}^M (\nu_i(q_l) + z_i) \\
= \text{wt}_A(t, r') + \nu'(z, q) \\
= [A'](t).
\]

In conclusion, $A'$ is an unambiguous max-plus-WTA with $[A] = [A']$.

**5 The Sequentiality Problem**

The sequentiality problem asks whether for a given max-plus-WTA $A$, there exists a deterministic max-plus-WTA $A'$ such that $[A] = [A']$. 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 [14].

Let $A = (Q, \Gamma, \mu, \nu)$ be an unambiguous max-plus-WTA. We say that $A$ satisfies the *twins property* [17, 4] if the following holds. Whenever for $p, q \in Q$ there exists a tree $u \in T_\Gamma$ such that $\text{Run}_A(u, p) \neq \emptyset$ and $\text{Run}_A(u, q) \neq \emptyset$ and a $\Gamma$-word $s \in T_\Gamma$ and runs $r_p \in \text{Run}_A(p, s, p)$ and $r_q \in \text{Run}_A(q, s, q)$, then $\text{wt}_A^\omega(s, r_p) = \text{wt}_A^\omega(s, r_q)$.

**Lemma 23.** Let $A = (Q, \Gamma, \mu, \nu)$ be a trim unambiguous max-plus-WTA. There exists a deterministic max-plus-WTA $A'$ with $[A] = [A']$ if and only if $A$ satisfies the twins property. If it exists, it can be effectively constructed.

**Proof.** If $A$ satisfies the twins property, we know due to [4, Lemma 5.10] that a deterministic max-plus-WTA $A'$ with $[A'] = [A]$ can be effectively constructed.

We show that the twins property is also a necessary condition. The proof follows the idea for the proof of [17, Theorem 9]. Let $p, q \in Q$ such that there exists a tree $u \in T_\Gamma$ with runs $r^p \in \text{Run}_A(u, p)$
and \( r^q \in \text{Run}_A(u, q) \) and a \( \Gamma \)-word \( s \in T_{\Gamma} \) with runs \( r_p \in \text{Run}_A^\ominus(p, s, p) \) and \( r_q \in \text{Run}_A^\ominus(q, s, q) \). Since \( A \) is trim, there exist \( \Gamma \)-words \( \hat{u}_p \) and \( \hat{u}_q \) with runs \( \hat{r}_p \in \text{Run}_A^\ominus(p, \hat{u}_p, p') \) and \( \hat{r}_q \in \text{Run}_A^\ominus(q, \hat{u}_q, q') \) such that \( p' \) and \( q' \) are final.

We define \( t_p^{(n)} = \hat{u}_p(s^n(u)) \) and \( t_q^{(n)} = \hat{u}_q(s^n(u)) \) for \( n \geq 1 \). Then since \( A \) is unambiguous, we see that with the constants \( \lambda_p = \text{wt}_A(s, r_p) \) and \( \lambda_q = \text{wt}_A(s, r_q) \) and the constants \( \kappa_p = \text{wt}_A(u, r_p) + \text{wt}_A(\hat{u}_p, \hat{r}_p) + \nu(p') \) and \( \kappa_q = \text{wt}_A(u, r_q) + \text{wt}_A(\hat{u}_q, \hat{r}_q) + \nu(q') \) we have

\[
[A](t_p^{(n)}) = \kappa_p + n\lambda_p \\
[A](t_q^{(n)}) = \kappa_q + n\lambda_q
\]

for all \( n \geq 1 \).

Now let \( A' \) be a deterministic max-plus-WTA with \( \|A'\| = [A] \) and let \( \kappa_1 \) be the largest weight in terms of absolute value which occurs in \( A' \), excluding \( -\infty \). Since \( A' \) is deterministic, we see that for the constant \( \kappa_2 = [\kappa_1]|\text{pos}(\hat{u}_p)| + |\text{pos}(\hat{u}_q)| + 2 \), we have

\[
|[[A]](t_p^{(n)}) - [[A]](t_q^{(n)})| \leq \kappa_2
\]

for all \( n \geq 1 \). This can only hold if \( \lambda_p = \lambda_q \). Thus, \( A \) satisfies the twins property. \( \square \)

**Lemma 24** ([4, Theorem 5.17]). For an unambiguous max-plus-WTA \( A \) it is decidable whether \( A \) satisfies the twins property.

**Theorem 25.** For a finitely ambiguous max-plus-WTA \( A \) it is decidable whether there exists a deterministic max-plus-WTA \( A' \) with \( \|A\| = [A'] \). If \( A' \) exists, it can be effectively constructed.

**Proof.** Let \( A \) 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, \( A \) can also not be determinizable. Otherwise we can effectively construct an unambiguous max-plus-WTA \( A' \) with \( \|A\| = [A'] \). Due to Lemma 24, we can decide whether \( A' \) satisfies the twins property, which according to Lemma 23 is equivalent to deciding whether \( A \) is determinizable. \( \square \)

**References**


