

Representing One Letter Weighted Automata over the Tropical Semiring

Shaull Almagor ✉ 

Technion, Haifa, Israel

Ismaël Jecker ✉ 

Université Marie et Louis Pasteur, CNRS, institut FEMTO-ST, F-25000 Besançon, France

Filip Mazowiecki ✉ 

University of Warsaw, Poland

Łukasz Orlikowski ✉ 

University of Warsaw, Poland

David Purser ✉ 

University of Liverpool, UK

Henry Sinclair-Banks ✉ 

University of Warsaw, Poland

Max Planck Institute for Software Systems (MPI-SWS), Kaiserslautern, Germany

Abstract

We consider weighted automata over the tropical semiring $\mathbb{Z}_\infty(\min, +)$. Recently, it was shown that determinisation is decidable; in this paper we focus on the complexity when the alphabet is unary. In 2001, Lombardy showed this problem is decidable, a close inspection of his proof yields a coNP upper bound on the complexity. Earlier Gaubert showed that every weighted automaton in this setting can be effectively turned into an equivalent union of deterministic weighted automata. We prove Gaubert's result efficiently, presenting it as a generalisation of Chrobak's normal form for unary NFA. In particular, we prove that the equivalent union of deterministic weighted automata can be represented by a weighted automaton of quadratic size in the size of the original one, and this representation can be computed in polynomial time. Building on this, we show that determinisation, and even register minimisation (which generalises determinisation), is coNP -complete. We complete the paper with observations that the boundedness problem is also coNP -complete by reductions with determinisation. Lastly, we provide evidence that all of these problems are not FPT (by proving coW_1 -hardness) when parametrised by the number of deterministic automata in the union.

2012 ACM Subject Classification Theory of computation $\rightarrow$ Quantitative automata

Keywords and phrases weighted automata, determinisation, register minimisation

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2026.10

Related Version *Full Version containing omitted proofs:* <http://arxiv.org/abs/2606.26038>

Funding *Shaull Almagor:* Supported by the ISRAEL SCIENCE FOUNDATION (grant No. 989/22).
Ismaël Jecker: Supported by the Agence Nationale de la Recherche (ANR) grant ANR-25-CE48-2447 FAVOR.

Filip Mazowiecki: Supported by the Polish National Science Centre SONATA BIS-12 grant number 2022/46/E/ST6/00230.

Łukasz Orlikowski: Supported by the ERC grant INFSYS, agreement no. 950398.

Henry Sinclair-Banks: Supported by the ERC grant INFSYS, agreement no. 950398.

Acknowledgements We acknowledge the Weighted Automata Rock workshop funded by IDUB Thematic Research Programmes (POB III) of the University of Warsaw.



© Shaull Almagor, Ismaël Jecker, Filip Mazowiecki, Łukasz Orlikowski, David Purser, and Henry Sinclair-Banks;

licensed under Creative Commons License CC-BY 4.0

37th International Conference on Concurrency Theory (CONCUR 2026).

Editors: Ana Sokolova and Patrick Totzke; Article No. 10; pp. 10:1–10:20



Leibniz International Proceedings in Informatics

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

1 Introduction

Weighted automata over the tropical semiring $(\mathbb{Z} \cup \{\infty\}, \min, +)$ are a quantitative extension of finite automata by labelling transitions with integers. Instead of recognising input words, they assign them integer weights. To every run we assign the sum of the weights on its transition, and the output for a given input word aggregates the weights of accepting runs using the minimum function. Weighted automata over the tropical semiring play a central role in verification, see for example the surveys [26, 11]. For example, this model played a crucial role in Hashiguchi’s results proving decidability of the star height problem [18, 19].

The power of weighted automata over the tropical semiring makes the model convenient for expressing many properties, unfortunately at the cost that most problems are undecidable [3]. In contrast to Boolean finite automata, weighted automata cannot be always determined. Furthermore, decision problems for subclasses, like deterministic or unambiguous weighted automata, become significantly less complex and, in particular, decidable [26, 11]. This naturally raises the question whether an input automaton admits an equivalent deterministic weighted automaton; as a decision problem it is called the determinisation problem. A recent breakthrough established decidability of determinisation for tropical weighted automata [1], after decades from its formulation in the 1990s by Mohri [28, 29] (and already implicit in earlier work [9]).

In this paper, we focus on weighted automata over the tropical semiring when the alphabet is unary. In this setting decidability of determinisation was established by Lombardy [25]. While not stated explicitly, a **coNP** complexity bound can be inferred from [25], matching our upper bound. Indeed, Lombardy essentially reduces the problem to the setting when the input automaton is assumed to be unambiguous. Then, non-determinisability can be witnessed by word demonstrating a cycle in a certain automaton. More recently, determinisation for unambiguous automata over the unary alphabet was shown to be decidable in linear time [12, Theorem 1.4].

Our contributions. We revisit the unary alphabet setting with the aim of obtaining a precise structural and complexity-theoretic understanding of determinisation and related problems. Our approach is based on a refined analysis of runs using cycles of minimal average weight. Intuitively, any sufficiently long run of minimal weight must predominantly iterate such cycles. We show that this intuition can be turned into an effective normal form that captures all behaviours in a compact and algorithmically tractable way. Gaubert [15] shows that every unary weighted automaton can be transformed into an equivalent finite union of deterministic weighted automata. We improve this reduction showing that, with respect to the size of the original automaton, (i) there are linearly many automata in the union, (ii) each automaton in the union has quadratic size, and (iii) the transformation can be computed in polynomial time. This result is a weighted generalisation of Chrobak’s normal form for unary (Boolean) automata [10, 39], and yields a concise “lasso-shape” representation governed by cycles of minimal average weight.

We also consider the register minimisation problem, which is known to strictly generalise determinisation [4, 5]. Using our Chrobak-like decomposition, we obtain a **coNP** decision procedure both for register minimisation and determinisation problems. The proof is an analysis of the interactions between the deterministic components and identifying cycles that are asymptotically redundant. Note that over unrestricted alphabets, the register minimisation problem has recently been shown undecidable [2], in contrast to the aforementioned decidability of determinisation [1].

Another important problem is the boundedness problem, studied in Hashiguchi's work on star-height [18]. We show that determinisability coincides with boundedness. This link enables us to transfer techniques and results between the two settings. In particular, we obtain a coNP upper bound for boundedness.

We complete these results with lower bounds proving tight complexity bounds. We show that determinisation, register minimisation, and boundedness are all coNP -complete for unary weighted automata. Given our Chrobak-like result, it is natural to ask for FPT algorithms, when the input is explicitly a finite union of deterministic weighted automata. We prove that all three problems are coW_1 -hard when parameterised by the number of deterministic components in the decomposition, relying on Fernau, Hoffmann, and Wehar's work on the unary DFA intersection problem [13, 38]. This indicates that determinisation, register minimisation, and boundedness remain intractable even under strong structural restrictions.

Related work. Weighted automata are also studied over fields, in particular over the rationals $(\mathbb{Q}, +, \times)$ [35], encompassing probabilistic automata as a special case [16]. The determinisation and register minimisation problems have been shown to be decidable in the rational semiring [6], with the exact complexity still unresolved [7, 20] and polynomial time over the unary alphabet [24].

In general, one can also consider the ambiguity hierarchy, which is a fine-grained analysis of classes between deterministic and unrestricted non-deterministic automata. Most notable partial results on determinisability are for: unambiguous [29], finitely ambiguous [23], and polynomially ambiguous automata [22]. One can also generalise the determinisation problem by asking whether the input automaton admits an alternative representation that is unambiguous, finite-ambiguous, or polynomial-ambiguous [2, 33]. However, these problems trivialise in the case of unary alphabet over the tropical semiring. Recall that Lombardy showed that all unary weighted automata can be converted into unambiguous weighted automaton [25].

Taking the algebraic view of automata, a unary weighted automaton can be viewed as a matrix $A \in \mathbb{Z}^{d \times d}$ over the tropical semiring, where the *orbit* $\{A^n \mid n \in \mathbb{N}\}$ captures the behaviour of the automaton. Under this view, several works [36, 30, 17] have identified that the mean-value of “cycles” can be used to effectively reason about the orbit. Our work also contains several such observations, tailored to our setting. However, we stay in the view of state machines.

2 Preliminaries

For $k \in \mathbb{N} = \{0, 1, \dots\}$, we denote $[k] := \{1, \dots, k\}$. We denote by $\mathbb{N}_\infty$ and $\mathbb{Z}_\infty$ the sets $\mathbb{N} \cup \{\infty\}$ and $\mathbb{Z} \cup \{\infty\}$, respectively. We extend the addition and min operations to ∞ in the natural way: $a + \infty = \infty$ and $\min\{a, \infty\} = a$, for all $a \in \mathbb{Z}_\infty$. By $\arg \min\{f(x) : x \in A\}$ we mean the set of elements in A for which $f(x)$ is minimal for some function f and set A .

Weighted automata. A *Unary $(\min, +)$ Weighted Finite Automaton* (unary WFA for short) is a tuple $\mathcal{A} = \langle Q, Q_0, \Delta, F \rangle$, where Q is a finite set of *states*, $Q_0 \subseteq Q$ are the *initial states*, $\Delta \subseteq Q \times \mathbb{Z}_\infty \times Q$ is a *transition relation* such that for every $p, q \in Q$ there exists exactly¹

¹ This is without loss of generality: if there are two transitions with different weights, the higher weight can always be ignored in the $(\min, +)$ semantics. Missing transitions can be introduced with weight ∞ .

10:4 Representing One Letter Weighted Automata over the Tropical Semiring

one weight $c \in \mathbb{Z}_\infty$ such that $(p, c, q) \in \Delta$, and $F \subseteq Q$ is a set of *accepting states*. If $|Q_0| = 1$ and, for every $p \in Q$, there exists at most one transition (p, c, q) with $c \neq \infty$, then $\mathcal{A}$ is called *deterministic*. We denote by $\|\mathcal{A}\|$ the maximal absolute value of any finite weight in Δ .

► **Remark 1 (Initial and final weights and states).** Weighted automata are often defined with initial and final weights, i.e., Q_0 and F are replaced by initial and final vectors $\mathbf{init}, \mathbf{fin} \in \mathbb{Z}_\infty^Q$. The weight of a run then also includes the initial weight and final weight. Our choice is with only a small loss of generality, the expressivity of the two models differ only on the empty word, which has no effect on our results.

► **Remark 2.** Throughout this paper we use the $(\min, +)$ semiring. Since we work over $\mathbb{Z}$, our results also apply to $(\max, +)$ by symmetry.

Runs. A *run* of $\mathcal{A}$ is a sequence of transitions $\rho = (t_1, t_2, \dots, t_m)$ where $t_i = (p_i, c_i, q_i)$ such that $q_i = p_{i+1}$ for every $1 \leq i < m$, and $c_i < \infty$ for every $1 \leq i \leq m$; we write $\rho : p_1 \xrightarrow{m} q_m$ to denote this run. We say that ρ is *of length m from p_1 to q_m* ; we also use the notation $\text{len}(\rho) := m$ to denote the length of a run. For an infix $w[i, j]$, we denote the corresponding infix of ρ by $\rho[i, j] := (t_i, \dots, t_j)$. The *weight* of the run ρ is $\text{wt}(\rho) = \sum_{i=1}^m c_i$. A run is *accepting* if $p_1 \in Q_0$ and $q_m \in F$.

For $n \in \mathbb{N}$, we abuse the name of the unary WFA as the function it describes, and denote by $\mathcal{A}(n)$ the weight assigned by $\mathcal{A}$ to n , which is the minimal weight of an accepting run of $\mathcal{A}$ of length n . For convenience, we introduce some auxiliary notations. For $n \in \mathbb{N}$ and sets of states $Q_1, Q_2 \subseteq Q$, we define (with the convention $\min \emptyset = \infty$)

$$\text{mwt}_{\mathcal{A}}(Q_1 \xrightarrow{n} Q_2) := \min\{\text{wt}(\rho) \mid \exists q_1 \in Q_1, q_2 \in Q_2, \rho : q_1 \xrightarrow{n} q_2\}.$$

If Q_1 or Q_2 are singleton, then we denote them by a single state (e.g., $\text{mwt}_{\mathcal{A}}(P \xrightarrow{n} q)$ for some set $P \subseteq Q$ and state $q \in Q$). Accordingly, we can define $\mathcal{A}(n) = \text{mwt}_{\mathcal{A}}(Q_0 \xrightarrow{n} F)$. If there are no accepting runs on n , then $\mathcal{A}(n) = \infty$. The unary WFA then defines the function $\mathcal{A} : \mathbb{N} \rightarrow \mathbb{Z}_\infty$. We say the domain of $\mathcal{A}$, $\text{dom}(\mathcal{A})$, is the set $\{n \in \mathbb{N} \mid \mathcal{A}(n) < \infty\}$.

We write $p \xrightarrow{n} q$ when there exists some run ρ such that $\rho : p \xrightarrow{n} q$. We lift this notation to concatenations of runs, e.g., $\rho : p \xrightarrow{n} q \xrightarrow{m} r$ means that ρ is a run on $n + m$ from p to r that reaches q after the prefix n . We also incorporate this to mwt by writing $\text{mwt}(q \xrightarrow{n} p \xrightarrow{m} r)$ to refer to the minimal weight of a run $\rho : q \xrightarrow{n} p \xrightarrow{m} r$.

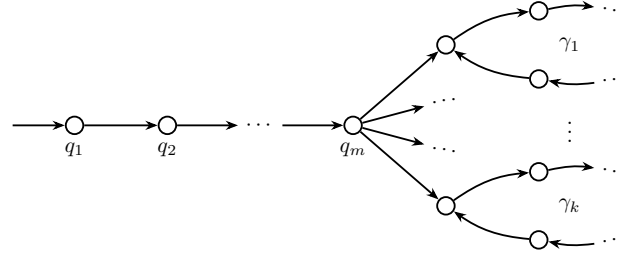
Average weight. Given a run ρ (typically a cycle in what follows), the *average weight* of ρ is $\text{wt}(\rho)/\text{len}(\rho)$, i.e., the average effect of each step in the ρ . When ρ is a cycle, we sometimes call the average weight the *slope*.

Trim assumption. A WFA is *trim* if every state is reachable from Q_0 by some run and can reach an accepting state with some run. Note that states that do not satisfy this can be found in linear time (by using a simple graph search algorithm), and can be removed from the WFA without changing the weight of any accepted word. We assume that all WFAs are trim throughout.

k -Width WFAs. Consider a WFA $\mathcal{A}$. The *width* of $\mathcal{A}$ is the maximal number of states that are simultaneously reachable in $\mathcal{A}$.

Cost Register Automata. A *Cost Register Automaton with linear register updates* of dimension k (k -CRA) is a tuple $\mathcal{N} = \langle Q, \Sigma, \delta, q_0, F, \text{upd}, \mathbf{fin} \rangle$ with the following components:

- $\langle Q, \Sigma, \delta, q_0, F \rangle$ is a deterministic finite automaton (DFA), with $\delta : Q \times \Sigma \rightarrow Q$ a deterministic transition function.



■ **Figure 1** The structure of an automaton (without any accepting states) in Chrobak normal form.

- For each state $q \in Q$ and letter $\sigma \in \Sigma$, the *update* $\text{upd}(q, \sigma)$ is a $k \times k$ matrix over $\mathbb{Z}_\infty$. We refer to its entries as $\text{upd}(q, \sigma)_{i,j}$ for $1 \leq i, j \leq k$.
- For each $q \in F$, the *output registers* are $\text{fin}(q) \subseteq [k]$.

A *valuation* of the k registers is a row vector $\vec{r} \in \mathbb{Z}_\infty^k$. Given such a valuation and a transition update $\text{upd}(q, \sigma)$, the *next valuation* is $\vec{r}' = \vec{r} \cdot \text{upd}(q, \sigma)$ (recall that all matrix products are in the $(\mathbb{Z}_\infty, \min, +)$ semiring). Denote $M = \text{upd}(q, \sigma)$ with entries $m_{i,j}$. Unfolding tropical matrix multiplication gives the following register update: $r'_i = \min\{r_1 + m_{1,i}, r_2 + m_{2,i}, \dots, r_k + m_{k,i}\}$. Consider a word $w = \sigma_1 \dots \sigma_n$ and the unique run $\rho = q_0 \xrightarrow{\sigma_1} q_1 \xrightarrow{\sigma_2} \dots \xrightarrow{\sigma_n} q_n$ of the DFA part of $\mathcal{N}$ on w . Then ρ induces a sequence of valuations $\vec{r}^0, \dots, \vec{r}^n$ given by the zero vector $\vec{r}^0 = \vec{0}$, and for every $0 \leq i < n$ we have $\vec{r}^{i+1} = \vec{r}^i \cdot \text{upd}(q_i, \sigma_{i+1})$. If $q_n \in F$, the value assigned by $\mathcal{N}$ to w is $\min\{\vec{r}^n(i) \mid i \in \text{fin}(q_n)\}$. If $q_n \notin F$ the value is ∞ .

3 Efficiently Computing Normal Forms for Unary WFA

A unary non-deterministic finite automaton is said to be in *Chrobak normal form* if it consists of a path $q_1 \rightarrow q_2 \rightarrow \dots \rightarrow q_m$, and k simple cycles $\gamma_1, \dots, \gamma_k$, with q_1 initial, and q_m non-deterministically branching into each of the k cycles, as depicted in Figure 1. The key result is that every n -state unary NFA can be converted, in polynomial time, into a unary NFA in Chrobak normal form with $m, k \leq O(n^2)$ and for each cycle having length at most n [10, 27, 39].

In this section, we consider two forms for unary WFAs: unions of deterministic unary WFAs and unary WFAs in Chrobak normal form. It is known that non-deterministic unary WFA can be represented as a union of deterministic unary WFAs, which are necessarily individually lasso-shaped automata [15, 25]. Of course, one can also unroll the cycles so that the initial simple path segments have the same length, and then merge these initial paths (by suitably adapting the weights on the transitions of the initial path and branching transitions), to yield a single unary WFA whose underlying unary DFA is in Chrobak normal form. However, these results are existential and do not entail that automata in this form can be small, nor found efficiently.

For the remainder of this section, we fix our attention on a trim unary WFA $\mathcal{A} = \langle Q, Q_0, \Delta, F \rangle$. In Section 3.1, we will first prove that $\mathcal{A}$ can be efficiently converted into a small union of quadratic size deterministic unary WFA (Theorem 3). After, in Section 3.2, we will strengthen our result to prove that one can efficiently compute an equivalent quadratic size Chrobak normal form for $\mathcal{A}$ in polynomial time (Corollary 15).

3.1 Small Unions of Small Deterministic Unary WFAs

► **Theorem 3.** *There exist deterministic unary WFAs $\mathcal{B}_1, \dots, \mathcal{B}_k$ such that*

- (1) *for every n , $\mathcal{A}(n) = \min_{1 \leq i \leq k} \{\mathcal{B}_i(n)\}$;*
- (2) *$k \leq |Q|$; and*
- (3) *for every $1 \leq i \leq k$:*
 - (i) *$\mathcal{B}_i$ consists of at most $2|Q|^2 + |Q|$ states and transitions,*
 - (ii) *$\|\mathcal{B}_i\| \leq 8|Q|^2\|\mathcal{A}\|$, and*
 - (iii) *$\mathcal{B}_i$ can be computed from $\mathcal{A}$ in polynomial time.*

In order to prove Theorem 3, we will identify a collection of “representative” cycles. Roughly speaking, for any $n \in \mathbb{N}$, any minimal-weight run of length n will mostly use cycles of minimal *average* weight to get to the desired length n . This is because any run of length n that repeatedly uses cycles with greater weight must end up having greater weight than a run that instead uses minimal average weight cycles. In some situations, cycles with non-minimal average weight must be taken; we will argue that such cycles will only be used a small number of times. We formalise this in Lemma 4. With this, our approach to prove Theorem 3 is to, more or less, categorise runs based on the lowest average weight short cycle used.

We call a cycle *short* if it consists of at most $|Q|$ transitions. Note that all simple cycles are short. For every state $q \in Q$, we *distinguish* γ_q to be a short q -cycle that has minimal average weight among all short q -cycles. Recall that the average weight of a cycle is the ratio of its weight and length. Note that there may be multiple such cycles for a given state q , we can arbitrarily distinguish any one of these cycles.

► **Lemma 4.** *Let $n \in \mathbb{N}$ and $s, t \in Q$ such that $\text{mwt}(s \xrightarrow{n} t) < \infty$. There exists a minimal weight run ρ from s to t of length n with the following properties:*

- (1) *there is a decomposition $\rho = \tau\gamma_q^x\tau'$, for some distinguished cycle γ_q and $x \in \mathbb{N}$; and*
- (2) *$\text{len}(\tau) + \text{len}(\tau') < 2|Q|^2$.*

We remark that Lemma 4 does not specify that the run of length n starts from an initial state, or leads to an accepting state. Though, it immediately follows that if $\mathcal{A}(n) \neq \infty$, then there exists an accepting run of length n and weight $\mathcal{A}(n)$ with properties (1) and (2) as specified by Lemma 4.

To prove Lemma 4, we will perform a cycle replacement argument on an arbitrary minimal weight run of length n to obtain ρ . Some of the following ideas appear in Rackoff’s rearrangement and replacement arguments used to analyse the length of runs that witness unboundedness in vector addition systems [34, Lemma 4.5]. Similar arguments have also been used in the development of Chrobak normal form for finite automata, in particular in Lin’s proof [39] which makes a subtle correction to Chrobak’s [10]. However, we must take care to ensure the weights are correctly maintained. The main technical contribution is to show that this can still be completed efficiently.

Proof of Lemma 4. Initially, let π be an arbitrary run from s to t of length n with minimal weight $w \in \mathbb{Z}$. Let $Q' \subseteq Q$ be the subset of states that are visited by π ; note that $Q' \supseteq \{s, t\}$. First, for every state $q \in Q' \setminus \{s, t\}$, that is visited by π , we “mark” exactly one instance² of a transition in π that leads to q . We also mark the final transition (which leads to t) and note that we need not mark any transition to s . In total, there are at most $|Q| - 1$ many marked transitions.

² To be clear, some transitions may occur multiple times in π , if one is marked then only one instance of said transition is marked. We also note that, for a particular state $q \in Q' \setminus \{s, t\}$, it does not matter which incoming transition is marked.

▷ **Claim 5.** If $\text{len}(\pi) \geq |Q|^2$, then there is a short cycle without any marked transitions in π .

One by one, short cycles that do not contain marked transitions are removed from π and placed in C , a multiset that is initially empty. The order in which short cycles are removed does not matter but, thanks to Claim 5, we can always identify and remove a short cycle from π until the length of π is less than $|Q|^2$.

Now, let $q \in Q'$ be the state visited by π that has the lowest average weight distinguished cycle γ_q . Formally,

$$q := \arg \min_{p \in Q'} \left\{ \frac{\text{wt}(\gamma_p)}{\text{len}(\gamma_p)} \right\}.$$

We will now repeatedly replace cycles from C with several iterations of γ_q . We must do this in a way that preserves the length. For this we use the following combinatorial claim with $m = |Q|$, $\ell = \text{len}(\gamma_q)$, and S is the multiset of all lengths of all cycles in C .

▷ **Claim 6.** Let $m, \ell \in \mathbb{N}$ such that $\ell \leq m$ and let S be a multiset of naturals such that, for every $s \in S$, $1 \leq s \leq m$. If $\sum_{s \in S} s \geq m^2$, then there exist a multiset $S' \subseteq S$ and $e \in \mathbb{N}$ such that $\sum_{s \in S'} s = e\ell$.

If the sum of all lengths of cycles in C is at least $|Q|^2$, then we can identify a collection of short cycles $\sigma_1, \dots, \sigma_i$ such that $\text{len}(\sigma_1) + \dots + \text{len}(\sigma_i) = e \cdot \text{len}(\gamma_q)$, for some $e \in \mathbb{N}$. This allows us to remove $\sigma_1, \dots, \sigma_i$ to later be replaced with e iterations of γ_q . Soon, we will argue that we can reconstruct a run of length n of the form $\tau\gamma_q^x\tau'$. Recall that π was assumed to have minimal weight; we therefore know that $\text{wt}(\sigma_1) + \dots + \text{wt}(\sigma_i) = e \cdot \text{wt}(\gamma_q)$.

The above cycle replacement is repeated until the sum of the lengths of cycles in C that remain is less than $|Q|^2$. Finally, we will argue that a run from s to t of length n can be reconstructed. Recall that for every state in $Q' \setminus \{s\}$, there is a marked transition in π that was not removed during the cycle removal process. This means for every state $q \in Q' \setminus \{s\}$, q -cycles can be reinserted into π : specifically a q -cycle can be taken immediately after the marked transition leading to q . Furthermore, s -cycles can be inserted at the very beginning of the run. It is also important to note that since the final transition leading to t was marked, we know that π is (still) a run from s to t . Let π' be the run that is obtained by adding all short cycles left in C after replacement (specifically, π' does not contain the replacement x iterations of γ_q). Since the combined length of cycles left in C was at most $|Q|^2$ and $\text{len}(\pi) < |Q|^2$; we deduce that $\text{len}(\pi') < 2|Q|^2$.

What remains are the $x := \frac{n - \text{len}(\pi')}{\text{len}(\gamma_q)}$ replacement iterations of γ_q . In just the same way, these cycle iterations can be inserted after the marked transition for q . Specifically, let τ be the prefix of π' before and including the marked transition for q , and let τ' be the suffix of π' after the marked transition for q . Then $\rho = \tau\gamma_q^x\tau'$ is a minimal weight run from s to t of length n . Clearly $\text{len}(\tau) + \text{len}(\tau') = \text{len}(\pi') < 2|Q|^2$. ◀

Now that we have this standard “linear form” for minimal weight runs, we can construct a collection of deterministic unary WFAs $\mathcal{B}_1, \dots, \mathcal{B}_k$ such that, for every $n \in \mathbb{N}$, $\mathcal{A}(n) = \min_{1 \leq i \leq k} \{\mathcal{B}_i(n)\}$. In fact, we will construct one deterministic unary WFA $\mathcal{B}_q$, for every $q \in Q$ for which there exists a (short) cycle at q in $\mathcal{A}$. Recall that deterministic unary WFAs have a basic lasso-shaped structure: there is a simple path leading to a single simple cycle. As is the case with any unary deterministic automaton, $\mathcal{B}_q$ will have one run of length n , for every $n \in \mathbb{N}$. The cycle in $\mathcal{B}_q$ will have the same length and overall weight as γ_q but the accepting states and individual transition weights will likely differ. The length of the simple path of $\mathcal{B}_q$ will be bounded above by $2|Q|^2$.

For every $q \in Q$, we define Π_q to be the set of all runs in $\mathcal{A}$ that (i) start at any initial state, (ii) end at any final state, and (iii) are of the form $\tau\gamma_q^x\tau'$ such that $\text{len}(\tau) + \text{len}(\tau') < 2|Q|^2$ and $x \in \mathbb{N}$. It is important to note that Π_q contains all runs of length less than $2|Q|^2$, even those that do not visit q . This fact will be most convenient for proving that $\mathcal{A}$ can be converted into an equivalent quadratic size unary WFA in Chrobak normal form (in particular for Claim 16 on Page 10).

It is important to observe that runs across all Π_q suffice for finding minimal weight runs. Thanks to Lemma 4, it is true that for every $n \in \mathbb{N}$ such that $\mathcal{A}(n) \neq \infty$, there exists $q \in Q$ and $\pi \in \Pi_q$ such that $\text{wt}(\pi) = \mathcal{A}(n)$. Now, for every $q \in Q$, we will define a weight function $f_q : \mathbb{N} \rightarrow \mathbb{Z}_\infty$ based on the paths in Π_q . Precisely, $f_q(n) := \min \{\text{wt}(\pi) : \pi \in \Pi_q \text{ and } \text{len}(\pi) = n\}$.

▷ **Claim 7.** For every $n \in \mathbb{N}$, $\mathcal{A}(n) = \min_{q \in Q} \{f_q(n)\}$.

Our goal is to construct $\mathcal{B}_q$ so that, for every $n \in \mathbb{N}$, the run of length n from the initial state of $\mathcal{B}_q$ has weight $f_q(n)$. Later, we will fix which control states are accepting to capture the situations when $\mathcal{A}(n) \in \mathbb{Z}$ and $\mathcal{A}(n) = \infty$. Now, we will prove that $f_q(n)$ is periodic from $n \geq 2|Q|^2$. We will use this fact to create the cycle of $\mathcal{B}_q$.

▷ **Claim 8.** Fix $q \in Q$. For every $n \geq 2|Q|^2$, $f_q(n + \text{len}(\gamma_q)) = f_q(n) + \text{wt}(\gamma_q)$.

Proof. Without loss of generality, we shall assume that $\text{wt}(\gamma_q) \geq 0$; the scenario in which $\text{wt}(\gamma_q) < 0$ is analogous.

First, we shall consider the case in which $f_q(n) \in \mathbb{Z}$ (i.e. $f_q(n) \neq \infty$). Given that $f_q(n) \in \mathbb{Z}$, there exists a path $\tau\gamma_q^x\tau' \in \Pi_q$ of length n with weight $f_q(n)$. From this, one can immediately construct a run of length $n + \text{len}(\gamma_q)$: $\tau\gamma_q^{x+1}\tau' \in \Pi_q$. This run has weight $f_q(n) + \text{wt}(\gamma_q)$, thus $f_q(n + \text{len}(\gamma_q)) \leq f_q(n) + \text{wt}(\gamma_q)$.

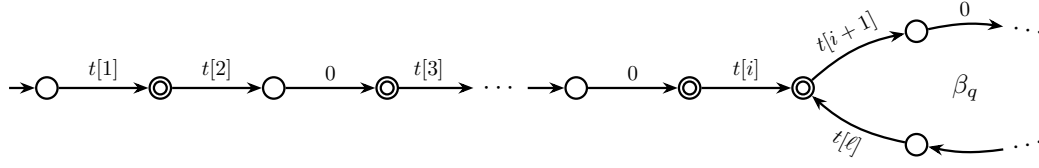
Now, for sake of contradiction, assume that $f_q(n + \text{len}(\gamma_q)) < f_q(n) + \text{wt}(\gamma_q)$. Suppose that $\tau\gamma_q^y\tau'$ is the run in Π_q of length $n + \text{len}(\gamma_q)$ with minimal weight, $f_q(n + \text{len}(\gamma_q))$. Since $n + \text{len}(\gamma_q) \geq 2|Q|^2 + \text{len}(\gamma_q)$ and $\text{len}(\tau) + \text{len}(\tau') < 2|Q|^2$, we know that $y \geq 1$. This means that $\tau\gamma_q^{y-1}\tau' \in \Pi_q$ is a run of length n and weight $f_q(n + \text{len}(\gamma_q)) - \text{wt}(\gamma_q)$. Thus $f_q(n) \leq f_q(n + \text{len}(\gamma_q)) - \text{wt}(\gamma_q)$. This creates a contradiction as $f_q(n + \text{len}(\gamma_q)) - \text{wt}(\gamma_q) < f_q(n)$. Thus, in the case that $f_q(n) \in \mathbb{Z}$, $f_q(n + \text{len}(\gamma_q)) = f_q(n) + \text{wt}(\gamma_q)$.

Now, to conclude the proof, we shall consider the case in which $f_q(n) = \infty$. In this case, there is no path of length n in Π_q . Suppose, for sake of contradiction, that $f_q(n + \text{len}(\gamma_q)) \neq \infty$. This means that there is a path $\tau\gamma_q^y\tau' \in \Pi_q$ of length $n + \text{len}(\gamma_q)$. Since $n + \text{len}(\gamma_q) \geq 2|Q|^2 + \text{len}(\gamma_q)$ and $\text{len}(\tau) + \text{len}(\tau') < 2|Q|^2$, we know that $y \geq 1$. This means that $\tau\gamma_q^{y-1}\tau' \in \Pi_q$ is a run of length n and weight $f_q(n + \text{len}(\gamma_q)) - \text{wt}(\gamma_q) < \infty$. This creates a contradiction, and so we conclude that if $f_q(n) = \infty$, then $f_q(n + \text{len}(\gamma_q)) = f_q(n) + \text{wt}(\gamma_q) = \infty$. ◁

Now we are ready to construct $\mathcal{B}_q$. There will be $2|Q|^2 + \text{len}(\gamma_q)$ many states: $2|Q|^2 - 1$ will form a simple path leading to a cycle of $\text{len}(\gamma_q) + 1$ many states. We use β_q to refer to the cycle of $\mathcal{B}_q$.

First, we shall consider which states of $\mathcal{B}_q$ are accepting. This part is straightforward: for every $n < 2|Q|^2 + \text{len}(\gamma_q)$ for which $f_q(n) \in \mathbb{Z}$, we mark the n -th state of $\mathcal{B}_q$ as accepting. Otherwise, if $f_q(n) \neq \infty$, then the n -th state is not an accepting state.

Now, let $n_1 < n_2 < \dots < n_\ell$ be the lengths of words accepted by $\mathcal{A}$ such that (i) for every $i \in [\ell]$, $f_q(n_i) \in \mathbb{Z}$, (ii) $n_{\ell-1} < 2|Q|^2 + \text{len}(\gamma_q) \leq n_\ell$, and (iii) for every $n \notin \{n_1, \dots, n_\ell\}$ such that $n < n_\ell$, $f_q(n) = \infty$. In other words, $n_1, n_2, \dots, n_{\ell-1}$ are the lengths of words with accepting runs in Π_q that are at most $2|Q|^2 + \text{len}(\gamma_q)$, and n_ℓ is the length of the shortest word with a run in Π_q whose length is at least $2|Q|^2 + \text{len}(\gamma_q)$. Now, consider the sequence of weight differences of consecutive accepted lengths: $t = (f_q(n_1), f_q(n_2) - f_q(n_1), \dots, f_q(n_\ell) - f_q(n_{\ell-1}))$. This sequence satisfies the telescoping property that, for every $i \in [\ell]$, $\sum_{j=1}^i t[j] = f_q(n_1) + (f_q(n_2) - f_q(n_1)) + \dots + (f_q(n_i) - f_q(n_{i-1})) = f_q(n_i)$.



■ **Figure 2** The deterministic unary WFA $\mathcal{B}_q$.

We will use the sequence t for the transition weights. Overall, we wish to construct $\mathcal{B}_q$ so that when the n -th state is reached, the current weight of the run is $f_q(n)$, assuming that $f_q(n) \neq \infty$. Concretely, before n_i -th state is reached, the weight of the run must be $f_q(n_i)$. To achieve this, on the first transition (the transition leading out from the initial state), the transition weight will be $t[1] = f_q(n_1)$. Then, for every $i \in [\ell]$, the transition from the n_i -th state has the weight $t[i + 1]$. On transitions from non-accepting states, we just set the transition weight to 0. See Figure 2 for an example of $\mathcal{B}_q$ in which $n_1 = 1, n_2 = 3, \dots, n_i = 2|Q|^2, \dots, n_\ell = 2|Q|^2 + \text{len}(\gamma_q)$.

▷ **Claim 9.** $\|\mathcal{B}_q\| \leq 8|Q|^2\|\mathcal{A}\|$.

▷ **Claim 10.** $\text{wt}(\beta_q) = \text{wt}(\gamma_q)$.

▷ **Claim 11.** For every $n \in \mathbb{N}$, $\mathcal{B}_q(n) = f_q(n)$.

We can now put the pieces together to prove Theorem 3. All that remains is to prove that $\mathcal{B}_q$ can be computed from $\mathcal{A}$ in polynomial time.

Proof of Theorem 3. For every $q \in Q$, we construct $\mathcal{B}_q$ as is described after Claim 8. Recall Claims 7 and 11: for every $n \in \mathbb{N}$, $\mathcal{A}(n) = \min_{q \in Q} \{f_q(n)\}$ and $\mathcal{B}_q = f_q(n)$, for every $q \in Q$. Together, we obtain condition 1 of Theorem 3.

Condition 2 follows from the fact that there is at most one deterministic unary WFA $\mathcal{B}_q$ for every state $q \in Q$ of $\mathcal{A}$.

Condition 3i immediately follows from the construction of $\mathcal{B}_q$: there are at most $2|Q| + \text{len}(\gamma_q) \leq 2|Q|^2 + |Q|$ states and transitions (see Figure 2 on Page 9). Condition 3ii is covered by Claim 9. Finally, for condition 3iii, it is important to note that we only need to know the length of the distinguished q -cycles; for this, we use Claim 13. We also need to compute the transition weights for $\mathcal{B}_q$; for this, we use Claim 14. These two claims both rely on the basic fact (Claim 12) that the weight of a minimal run of length n can be computed in polynomial time from $\mathcal{A}$ and n . We also note that Claim 12 can be used to decide which of the states in $\mathcal{B}_q$ are accepting in polynomial time.

▷ **Claim 12.** Let $\mathcal{A}'$ be an arbitrary unary WFA with states Q' , let $S, T \subseteq Q'$, and let $n \in \mathbb{N}$. In polynomial time from $\mathcal{A}'$ and n , we can compute the weight of a minimal run that starts at some state $s \in S$, ends at some state $t \in T$, and has length n .

▷ **Claim 13** (Also follows from [21]). Let $q \in Q$. We can compute, in polynomial time from $\mathcal{A}$, the length of the q -cycle with the minimal average weight among all short q -cycles.

▷ **Claim 14.** $f_q(n)$ can be computed in polynomial time with respect to $\mathcal{A}$ and n . ◀

3.2 Extending Chrobak's Normal Form to Unary WFAs

We will now strengthen Theorem 3 by proving that there exists a quadratic size unary WFA in Chrobak normal form that is equivalent to $\mathcal{A}$ (which can be computed in polynomial time). To show this we consider the the deterministic unary WFAs $\mathcal{B}_1, \dots, \mathcal{B}_k$ output by Theorem 3,

10:10 Representing One Letter Weighted Automata over the Tropical Semiring

and observe that the stem of each $\mathcal{B}_i$ is identical, meaning their stem can be merged, branching at the end of the stem to the cycle of the relevant $\mathcal{B}_i$. We state and prove this formally in the following corollary, for which we recall that we fixed our attention to a trim unary WFA $\mathcal{A} = \langle Q, Q_0, \Delta, F \rangle$.

► **Corollary 15.** *There exists a non-deterministic unary WFA $\mathcal{B}$ such that*

- (1) *for every n , $\mathcal{B}(n) = \mathcal{A}(n)$;*
- (2) *the underlying unary NFA of $\mathcal{B}$ is in Chrobak normal form;*
- (3) *$\mathcal{B}$ consists of at most $3|Q|^2$ states and transitions, in particular:*
 - (i) *the initial path of $\mathcal{B}$ has less than $2|Q|^2$ states and*
 - (ii) *there are at most $|Q|$ many cycles in $\mathcal{B}$, each of length at most $|Q|$;*
- (4) *$\|\mathcal{B}\| \leq 8|Q|^2\|\mathcal{A}\|$; and*
- (5) *$\mathcal{B}$ can be computed in polynomial time.*

Proof. Suppose $\mathcal{B}_1, \dots, \mathcal{B}_k$ are the deterministic unary WFAs output by Theorem 3. Observe that in the construction, the initial paths (the first $2|Q|^2 - 1$ states and transitions) of every $\mathcal{B}_i$ have the same weights and the same accepting states. This is formalised in the following claim and proof.

▷ **Claim 16.** For every $p, q \in Q$ and every $n < 2|Q|^2$, $f_p(n) = f_q(n)$.

Proof. This follows from the fact that, for every $q \in Q$, Π_q contains all runs of length less than $2|Q|^2$ that start in any initial state, and that end in any final state. Thus, for every $p, q \in Q$, $\{\pi \in \Pi_p : \text{len}(\pi) < 2|Q|^2\} = \{\pi \in \Pi_q : \text{len}(\pi) < 2|Q|^2\}$. Hence $f_p(n) = f_q(n)$, for every $n < 2|Q|^2$. ◁

To add some detail, recall that the transition updates in the initial path of $\mathcal{B}_q$ are either $t[i] = f_q(n_i) - f_q(n_{i-1})$ or 0 (depending only on whether the transition was outgoing from an accepting state). Given that the accepting states in the initial path of $\mathcal{B}_q$ only depend on whether $\mathcal{A}$ has an accepting run of length n (for $n < 2|Q|^2$), thanks to Claim 16, it follows that the n -th transition of each $\mathcal{B}_q$ has the same weight.

Accordingly, it is possible to merge the first $2|Q|^2 - 1$ states and transitions of $\mathcal{B}_1, \dots, \mathcal{B}_k$ to obtain $\mathcal{B}$. After the $(2|Q|^2 - 1)$ -st transition, there will then be a non-deterministic branch to each of the cycles from $\mathcal{B}_1, \dots, \mathcal{B}_k$. It is immediate that $\mathcal{B}$ satisfies the structural conditions 2, 3i, and 3ii of Corollary 15. The remaining conditions come directly from Theorem 3. The first condition follows from condition 1 of Theorem 3, the fourth follows from condition 3ii of Theorem 3, and the fifth follows from condition 3iii of Theorem 3. ◀

4 Register Minimisation of CRAs

Tropical cost register automata (CRA) are deterministic automata augmented with registers taking values in $\mathbb{Z}$, denoted by k -CRA when there are k registers. Each transition specifies how the register values are updated linearly (within the tropical semiring) with respect to the other transitions. Intuitively, register updates take the following form $r_i \leftarrow \min\{r_1 + m_{1,i}, r_2 + m_{2,i}, \dots, r_k + m_{k,i}\}$, where r_i 's are registers and $m_{i,j}$ are constants (recall the formal definition in Section 2). Note that since CRAs are deterministic, then over a unary alphabet they are lasso-shaped. The complexity of their behaviour stems from the register updates.

A natural question is whether their representations can be simplified by reducing the number of registers. This leads to the notion of *CRA Dimension Reduction*: given a k -CRA $\mathcal{A}$ and an integer $k' < k$, does there exist an equivalent k' -CRA $\mathcal{B}$? In this section we will show that the problem is in **coNP** (and later, that it is **coNP**-complete in Theorem 27). Note that in the general alphabet case, this problem is undecidable [2].

To study this problem, we exploit the fundamental result that CRA are expressively equivalent to WFA [4]. In one direction, the idea is that a WFA can non-deterministically track each register of a CRA using its states. For the converse, a CRA can simulate a non-deterministic WFA by keeping a register for each state of the WFA, and updating the registers according to the transitions of the WFA.

Crucially, this equivalence can be refined: the number of registers corresponds exactly to the width of the WFA. The following is proved in [2].

► **Theorem 17** ([2]). *The class of functions representable by a k -CRA is exactly that representable by width k WFAs (and the translation between the two is effective).*

By this theorem, CRA Dimension Reduction is equivalent to the *WFA Width Reduction* problem which asks, given a k -width WFA $\mathcal{A}$ and $k' < k$, does there exist an equivalent k' -width WFA $\mathcal{B}$? Both problems generalise the determinisation problem which can be stated as asking whether there is an equivalent 1-width WFA or a 1-CRA.

Since the two problems are equivalent, it suffices to work entirely in the WFA setting, which we do throughout to prove the following:

► **Theorem 18.** *The CRA Dimension Reduction and the unary WFA Width Reduction problems are in coNP .*

Proof overview. To prove Theorem 18, we develop a procedure that computes the minimal width of a unary WFA $\mathcal{A}$. Our procedure goes through the three following steps.

1. First, we decompose $\mathcal{A}$ in polynomial time into a family of deterministic unary WFA $(\mathcal{B}_i)_{i=1}^{k''}$.
2. Next, we group in polynomial time all components $\mathcal{B}_i$ that share the same asymptotic behaviours, which yields a smaller family of unary WFAs $(\mathcal{C}_i)_{i=1}^{k'}$.
3. Finally, we identify and discard *redundant* components $\mathcal{C}_i$, namely those whose domain is eventually covered by other components with a smaller average cycle weight. We show that we can establish whether a component is redundant in coNP (Proposition 20).

We then prove that the number of non-redundant components obtained in the last step is exactly the minimal width of any unary WFA equivalent to $\mathcal{A}$ (Propositions 24 and 25).

Complexity. Note that the first two steps can be carried out in polynomial time, and that redundancy of a given component $\mathcal{C}_i$ can be decided in coNP , that is, if $\mathcal{C}_i$ is not redundant, then it has a polynomial size non-redundancy witness.

To solve the unary WFA Width Reduction problem, it suffices to decide whether there exists a k' -width WFA equivalent to $\mathcal{A}$, without requiring minimality of k' . Our procedure yields a coNP algorithm: a certificate of non-existence of a k' -width unary WFA consists of $k' + 1$ non-redundancy witnesses for distinct components.

Let us now address the unary WFA Width Minimisation problem: given a k -width unary WFA $\mathcal{A}$ and an integer $k' < k$, is k' the *minimal* integer such that there exists a k' -width unary WFA equivalent to $\mathcal{A}$? Formally, let $\text{opt}(\mathcal{A})$ be the minimal value such that there exists an equivalent $\text{opt}(\mathcal{A})$ -width unary WFA. Our procedure yields a D^{P} algorithm, where D^{P} is the class of languages that are the intersection of a language in coNP and a language in NP (see [32, Section 2]). Indeed, we can decide in coNP whether $k' \geq \text{opt}(\mathcal{A})$, and dually in NP whether $k' - 1 < \text{opt}(\mathcal{A})$. The intersection of these two conditions characterises $k' = \text{opt}(\mathcal{A})$, which concludes the argument.

4.1 The Procedure

For the rest of the section we fix a unary WFA $\mathcal{A}$, and we now detail each step of the procedure, before proving its correctness.

Step 1: Decomposition. We begin by applying Theorem 3 to decompose $\mathcal{A}$ in polynomial time into a family of deterministic unary WFAs $(\mathcal{B}_i)_{i=1}^{k''}$ such that

$$\mathcal{A}(n) = \min_{1 \leq i \leq k''} \{\mathcal{B}_i(n)\} \text{ for all } n \in \mathbb{N}. \quad (1)$$

Step 2: Grouping components. We now group together the components $\mathcal{B}_i$ that share the same asymptotic behaviour, that is, the same average cycle weight. Formally, let $\alpha_1 < \alpha_2 < \dots < \alpha_{k'}$ be the distinct average weights of the cycles of the automata $(\mathcal{B}_i)_{i=1}^{k''}$. For each $1 \leq i \leq k'$, we define $\mathcal{C}_i$ as the union of all components $\mathcal{B}_j$ whose cycle has average weight α_i . Note that the average cycle weights of the components $\mathcal{B}_j$ can be computed and compared in polynomial time, therefore, as we do not determinise $\mathcal{C}_i$, this construction can be done in polynomial time. Since each $\mathcal{B}_j$ appears in one $\mathcal{C}_i$, Equation (1) yields

$$\mathcal{A}(n) = \min_{1 \leq i \leq k'} \{\mathcal{C}_i(n)\} \text{ for all } n \in \mathbb{N}. \quad (2)$$

Moreover, since all the cycles in $\mathcal{C}_i$ have average weight α_i , we get the following:

► **Lemma 19.** *There exists $N \in \mathbb{N}$ such that, for all $1 \leq i \leq k'$ and all $n \in \mathbb{N}$, every run ρ of $\mathcal{C}_i$ of length n satisfies $|\text{wt}(\rho) - n \cdot \alpha_i| \leq N$. Moreover N is polynomial in the description of the $\mathcal{C}_i$.*

Step 3: Discarding redundant components. We now formalise which elements of the decomposition $(\mathcal{C}_i)_{i=1}^{k'}$ are necessary. A component $\mathcal{C}_i$ is called *redundant* if it is eventually dominated domain-wise by the previous components:

$$\mathcal{C}_i \text{ is redundant if } \text{dom}(\mathcal{C}_i) \setminus \bigcup_{j=1}^{i-1} \text{dom}(\mathcal{C}_j) \text{ is finite.}$$

We show that while the words witnessing the non-redundancy might be exponentially large, we can decide their non-existence in **coNP**:

► **Proposition 20.** *We can establish in **coNP** whether a given component is redundant.*

Proof. A component $\mathcal{C}_i$ is redundant if and only if its domain differs only finitely from the union of the domains of the preceding components. Let $\mathcal{A}$ be the automaton recognising $\text{dom}(\mathcal{C}_i)$, and let $\mathcal{B}$ be the disjoint union of the automata recognising the domains of all $\mathcal{C}_j$ with $j < i$. Then redundancy reduces to checking whether $L(\mathcal{A}) \setminus L(\mathcal{B})$ is finite. This is known to be decidable in **coNP** [8, Lemma 7.15]. ◀

4.2 Minimal Width Equals the Number of Non-redundant Components

We now show that the number of non-redundant components is exactly the minimal width of any unary WFA equivalent to $\mathcal{A}$. Let $i_1 < \dots < i_k$ be the indices of the non-redundant components $\mathcal{C}_i$. We begin with an auxiliary lemma showing that $\mathcal{A}$ is eventually determined by the union of its non-redundant components.

► **Lemma 21.** *There exist $N \in \mathbb{N}$ and a partition*

$$\{n \in \text{dom}(\mathcal{A}) \mid n \geq N\} = D_1 \cup \dots \cup D_k$$

into k infinite regular languages such that $\mathcal{A}(n) = \mathcal{C}_{i_j}(n)$ for all $1 \leq j \leq k$ and all $n \in D_j$.

This follows from Lemma 19, which shows that each component $\mathcal{C}_i$ is asymptotically linear with slope α_i , together with the fact that the components are ordered by increasing values of α_i .

We now construct an automaton of width k equivalent to $\mathcal{A}$. The key step is to show that each non-redundant component can be determinised. For this, we rely on a classical characterisation of determinisability in terms of B -gap witnesses.

► **Definition 22.** *For a bound $B \in \mathbb{N}$, a B -gap witness is a pair of words $x, y \in \mathbb{N}$ and a state $q \in Q$ such that upon reading x , we have $\text{mwt}(Q_0 \xrightarrow{x} q) > \text{mwt}(Q_0 \xrightarrow{x} Q) + B$, but $\text{mwt}(Q_0 \xrightarrow{xy} F) = \text{mwt}(Q_0 \xrightarrow{x} q \xrightarrow{y} F)$.*

The following characterisation is folklore; we use the formulation of [1, Theorem 2.1].

► **Theorem 23** (folklore, [1, Theorem 2.1]). *$\mathcal{A}$ is determinisable if and only if there is a $B \in \mathbb{N}$ such that there is no B -gap witness.*

The determinisation construction that corresponds to the “if” direction of Theorem 23 yields a deterministic equivalent of size polynomial in B (the maximal gap) and single-exponential in the number of states of $\mathcal{A}$. We can now derive the upper bound.

► **Proposition 24.** *There exists a k -width unary WFA equivalent to $\mathcal{A}$ of size single-exponential in $\mathcal{A}$.*

We now establish the matching lower bound, showing that no automaton of smaller width can be equivalent to $\mathcal{A}$. The argument exploits the existence of the k infinite regular domains $D_1, \dots, D_k$ where each non-redundant component dominates according to its slope. We show that these regions enforce k pairwise distinct behaviours in any equivalent automaton, which requires width at least k .

► **Proposition 25.** *Every unary WFA $\mathcal{A}'$ equivalent to $\mathcal{A}$ has width at least k .*

5 Determinisation, Boundedness, and Hardness

In this section, we consider the determinisation problem directly and show how it relates with boundedness problems. The determinisation problem is a special case of the register minimisation problem for the case of a single register (see Theorem 17). Accordingly, our decision procedures and **coNP** complexity upper bound carries over. We will now prove that the problems are both **coNP**-hard. Furthermore, recall from Section 3 that a unary WFA can be represented as the union of k deterministic automata. We provide strong evidence that determinisation and boundedness are not fixed parameter tractable when k is the parameter.

First, we show that determinisation is effectively interreducible with value boundedness. The *value boundedness* problem is a specific variant of the boundedness problem which requires a sequence of words with increasing value (a single non-accepting word, with weight ∞ , does not suffice). This can be seen as the core of the weighted problem as the detection of non-accepting words is purely a reachability criterion.

► **Definition 26.** A weighted automaton $\mathcal{A}$ is bounded if there exists $B \in \mathbb{N}$ such that $|\mathcal{A}(n)| < B$ and not bounded otherwise. Moreover, $\mathcal{A}$ is value bounded if there exists $B \in \mathbb{N}$ such that if $\mathcal{A}(n) < \infty$, then $|\mathcal{A}(n)| < B$, and not value bounded otherwise³.

The boundedness problem asks, given a weighted automaton $\mathcal{A}$, whether $\mathcal{A}$ is bounded, and the value boundedness problem asks whether $\mathcal{A}$ is value bounded.

We show that the value boundedness problem is interreducible with the determinisation problem for unary weighted automata. Our approach is to show that (i) the connection is almost immediate if the minimal average cycle has weight zero, and (ii) there are reductions that adjust the systems to have minimal average cycle of zero weight suited to each direction of the equivalence. As a consequence we will show the following theorem.

► **Theorem 27.** The determinisation, boundedness, value boundedness, register reduction, and width reduction problems for unary weighted automata are *coNP-complete*.

5.1 Automata with Zero-Weight Minimal Average Cycles

In this section, we will establish the tight connection between value boundedness and determinisation in the case where the minimal average cycle has weight zero. We recall Definition 22 and Theorem 23 that tell us that $\mathcal{A}$ is determinisable if and only if there is no B -gap witness for some B . When the minimal average slope has weight zero, bounding B -gap witnesses, and thus determinisability, it turns out to correspond exactly to the value boundedness problem.

Recall that we assume $\mathcal{A}$ is trim, so that any run visiting a reachable cycle can be extended into an accepting word. Given a weighted automaton $\mathcal{A}$, a reachable cycle ρ is of minimal average weight if $\text{wt}(\rho)/\text{len}(\rho) \leq \text{wt}(\rho')/\text{len}(\rho')$, for all reachable cycles ρ' . Let $\alpha = \text{wt}(\rho)/\text{len}(\rho)$ be the minimal average slope achieved by a cycle in $\mathcal{A}$.⁴

▷ **Claim 28 ([21]).** A cycle achieving the minimal average slope α is simple, and therefore has of length less than $|Q|$. Moreover, α can be computed in polynomial time.

▷ **Claim 29.** Let $\mathcal{A}'$ be an automaton with minimal average slope of zero. There exists $B_0 \in \mathbb{N}$ such that, for all $x \in \mathbb{N}$, $\mathcal{A}'(x) \geq -B_0$ and $\ell, k \leq 2|Q|$ such that $\mathcal{A}'(\ell + kn) \leq B_0$ for all $n \in \mathbb{N}$.

▷ **Claim 30.** Let $\mathcal{A}'$ be an automaton with minimal average slope of zero. If $\mathcal{A}'$ is value bounded by M , then there is a B such that there is no B -gap witness, so $\mathcal{A}'$ is determinisable.

Proof. We note that $\mathcal{A}'$ does not have negative weight cycles, thus, if the weight of the run exceeds $M_{\mathcal{A}'} = M + |Q| \cdot \|\mathcal{A}'\|$ then it can never be extended to a minimum weight run (which has value less than M). Suppose we have a B -gap witness, $\text{mwt}(Q_0 \xrightarrow{x} q) > \text{mwt}(Q_0 \xrightarrow{x} Q) + B$, but $\text{mwt}(Q_0 \xrightarrow{xy} F) = \text{mwt}(Q_0 \xrightarrow{x} q \xrightarrow{y} F) \leq M_{\mathcal{A}'}$. But we have $Q_0 \xrightarrow{x} q$ used in a minimal weight run, and thus $\text{mwt}(Q_0 \xrightarrow{x} q) \leq M_{\mathcal{A}'}$. By Claim 29, we have $\text{mwt}(Q_0 \xrightarrow{x} Q) \geq -B_0$ for a constant $B_0 \in \mathbb{N}$. In summary, we have

$$M_{\mathcal{A}'} \geq \text{mwt}(Q_0 \xrightarrow{x} q) > \text{mwt}(Q_0 \xrightarrow{x} Q) + B \geq -B_0 + B,$$

and so all B -gap witnesses have $B \leq M_{\mathcal{A}'} + B_0$. ◁

³ We avoid the term *unbounded* to prevent ambiguity.

⁴ Section 3 considered minimal average (short) q -cycles. Here the property is global to the automaton.

▷ **Claim 31.** Let $\mathcal{A}'$ be an automaton with minimal average slope of zero. If $\mathcal{A}'$ is not value bounded then there is a B -gap witness for all B , and so $\mathcal{A}'$ is not determinisable.

Proof. Suppose $\mathcal{A}'$ is not value bounded; we will construct arbitrary large B -gap witnesses. We have sequences $m_1, m_2, \dots \in \mathbb{N}$, $L_1, L_2, \dots \in \mathbb{N}$ such that $\mathcal{A}'(m_i) = L_i$ and $L_i \xrightarrow{i \rightarrow \infty} \infty$. Consider a run such that $\text{mwt}(Q_0 \xrightarrow{m_i} F) = L_i$. Consider the largest m such that $\ell + km < m_i$. Let q be the state reached after $\ell + km$ steps on the run attaining weight L_i . Since $m_i - \ell - km < |Q|$ we have $\text{mwt}(Q_0 \xrightarrow{\ell + km} q) \geq L_i - |Q| \cdot \|\mathcal{A}'\|$. On the other hand we have, by Claim 29, we have $\text{mwt}(Q_0 \xrightarrow{\ell + km} Q) < B_0$ for a constant $B_0 \in \mathbb{N}$. Hence, we have a $(L_i - |Q| \cdot \|\mathcal{A}'\| - B_0)$ -gap witness formed by $x = \ell + km$ and $y = m_i - \ell - km$. Since $|Q|, \|\mathcal{A}'\|, B_0$ are bounded and $L_i \rightarrow \infty$, we have unbounded gap witnesses. ◁

5.2 Union of k Deterministic Automata

We observed in Section 3 that unary weighted automata can be represented as the union of deterministic automata, and can efficiently be transformed into this representation (Theorem 3). It is natural to consider how the number of such components affects the complexity.

In this section, we consider the boundedness, value boundedness and determinisability problems when the input is a union of k deterministic automata. Recall that a parameterised problem is fixed parameter tractable (FPT) if there exists an algorithm that solves any instance (I, k) of size n in time $f(k) \cdot n^{O(1)}$, where f is a computable function. In the sequel, we call the parametrised versions of boundedness, value boundedness, and determinisability k -det boundedness, k -det value boundedness, and k -det determinisability, respectively. We will prove that all three problems are not fixed parameter tractable.

Concretely, we show that all these problems are coW_1 -hard when parameterised by the number of deterministic automata in the union. Based on standard assumptions in parameterised complexity, this provides strong evidence that the problems are not FPT [14].

► **Lemma 32.** *k -det boundedness, k -det value boundedness and k -det determinisability are coW_1 -hard.*

Proof. In [13] it was shown that problem of nonemptiness of k deterministic unary automata is W_1 -hard. Let us fix an instance of this problem, namely k deterministic automata $(\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_k)$. Firstly, notice that $\bigcap_{i \in [k]} L(\mathcal{A}_i) \neq \emptyset$ if and only if $\bigcup L(\overline{\mathcal{A}_i}) \neq \{a\}^*$, where $\overline{\mathcal{A}_i}$ is the automaton recognising the complement of $L(\mathcal{A}_i)$ and can be computed in time polynomial in the size of $\mathcal{A}_i$. Notice also that $\overline{\mathcal{A}_i}$ is still a deterministic automaton. Hence, the universality problem parameterised by the number of deterministic automata in the union is coW_1 -hard.

k -det boundedness. Let us fix an instance of the universality problem (parametrised by the number of automata in the union), namely k deterministic unary automata $(\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_k)$. Let $\mathcal{A}$ be the automaton, which is the union of $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_k$. For each $i \in [k]$, we construct a deterministic unary WFA $\mathcal{B}_i$. Concretely, the states of $\mathcal{B}_i$ will be the states of $\mathcal{A}_i$. Also, accepting and initial states of $\mathcal{A}_i$ will be accepting and initial states of $\mathcal{B}_i$, respectively. Finally, we set the weight of every transition in $\mathcal{B}_i$ to zero. Let $\mathcal{B}$ be the automaton, which is the union of $\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_k$. Notice that if $\mathcal{A}$ is universal, then for every $n \in \mathbb{N}$, we have $\mathcal{B}(n) = 0$. On the other hand, if $\mathcal{A}$ does not accept a^n , then we have $\mathcal{B}(n) = \infty$. Since, for every $i \in [k]$, $\mathcal{B}_i$ can be constructed in time polynomial in size of $\mathcal{A}_i$ we conclude that k -det boundedness is coW_1 -hard.

k -det value boundedness. Once again, let us fix as instance of the universality problem, namely k deterministic unary automata $(\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_k)$. Let $\mathcal{A}$ be the automaton, which is the union of $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_k$ and let $U = \{n \mid a^n \notin L(\mathcal{A})\}$.

Notice, that if U is finite, but nonempty, then $U \subseteq [0, n]$ where n is the number of states of $\mathcal{A}$. This is because if there exists integer $\ell > n$ such that $\ell \in U$ then, for every $i \in [k]$, automaton $\mathcal{A}_i$ either cannot read a^ℓ or after reading a^ℓ is in a state q which lies on a simple cycle. Thus, also $\ell + n! \in U$, implying that U is infinite, as for every $i \in [k]$, if automaton $\mathcal{A}_i$ after reading a^ℓ is in a state q , then after reading $a^{\ell+n!}$ is also in the state q as it can only repeat the simple cycle in which q lies and the length of this cycle divides $n!$. Observe that in polynomial time in n , one can check whether there is $\ell \in [0, n] \cap U$ and, if there exists such ℓ , output a value unbounded single state, one transition (with weight one) unary WFA.

Otherwise, we can assume that U , if nonempty, is infinite and, for each $i \in [k]$, we construct a deterministic unary WFA $\mathcal{B}_i$. Concretely, the states of $\mathcal{B}_i$ will be the states of $\mathcal{A}_i$. Also, accepting and initial states of $\mathcal{A}_i$ will be accepting and initial states of $\mathcal{B}_i$, respectively. Finally, we set the weight of every transition in $\mathcal{B}_i$ to zero. We also construct an unary WFA $\mathcal{B}_{k+1}$ with a single state q , which is initial and accepting, and a single transition $(q, 1, q)$. Clearly, for every $i \in [k]$, $\mathcal{B}_i$ is a deterministic unary WFA and can be constructed in time polynomial in size of $\mathcal{A}_i$. Moreover, also $\mathcal{B}_{k+1}$ is a deterministic unary WFA and can be constructed in constant time. Hence, in order to show that k -det value boundedness is coW_1 -hard, it suffices to show that automaton $\mathcal{A}$ is universal if and only if $\mathcal{B}$, the union of $\mathcal{B}_1, \dots, \mathcal{B}_k$, is bounded.

Observe that if $n \in U$ then, for every $i \in [k]$, we have that $\mathcal{B}_i(n) = \infty$. Moreover, we have $\mathcal{B}_{k+1}(n) = n$. Hence, if $\mathcal{A}$ is not universal, meaning that U is nonempty, and hence infinite we can conclude that $\mathcal{B}$ is not value bounded. On the other hand, if $\mathcal{B}$ is value bounded then U needs to be finite, and hence empty, which means that $\mathcal{A}$ is universal. This concludes the proof that k -det value boundedness is coW_1 -hard.

k -det determinisability. To observe that k -det determinisability is also coW_1 -hard, we reduce from k -det value boundedness to k -det determinisability. Concretely, having k deterministic unary WFAs $(\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_k)$, for every $i \in [k]$, we construct a unary WFA $\mathcal{B}_i$ from $\mathcal{A}_i$ by replacing each transition $t = (q, c, q')$ by two transitions (q, c, s_t) and $(s_t, 0, q')$ using new non-accepting state s_t . Observe that, w.l.o.g. we can assume that, for every transition, $c \geq 0$, as the previous reduction produced a unary WFA only with such weights. Moreover, we construct $\mathcal{B}_{k+1}$ which is a zero-weight two-state component, alternating between an accepting and a non-accepting state, and with the initial state being the non-accepting state. Clearly, for every $i \in [k]$, $\mathcal{B}_i$ is a deterministic unary WFA and can be constructed in time polynomial in size of $\mathcal{A}_i$. Moreover, $\mathcal{B}_{k+1}$ is deterministic and can be constructed in constant time. Hence, in order to show that k -det determinisability is coW_1 -hard it is enough to show that automaton $\mathcal{A}$, that is a union of $\mathcal{A}_1, \dots, \mathcal{A}_k$, is value bounded if and only if unary WFA $\mathcal{B}$, that is a union of $\mathcal{B}_1, \dots, \mathcal{B}_k$, is determinisable. Firstly, notice that because of $\mathcal{B}_{k+1}$ we have, for every $n \in \mathbb{N}$, $B(2n + 1) = 0$. Moreover, for every $i \in [k]$ and $n \in \mathbb{N}$, we have $\mathcal{B}_i(2n) = \mathcal{A}_i(n)$. Hence, $\mathcal{A}$ is value bounded if and only if $\mathcal{B}$ is value bounded. Due to $\mathcal{B}_{k+1}$, the minimal average slope of $\mathcal{B}$ is zero. Thus, by Claim 30 and Claim 31, we have $\mathcal{B}$ is determinisable if and only if $\mathcal{B}$ is value bounded. $\blacktriangleleft$

5.3 Determinisability and Value Boundedness

In the preceding subsection, we showed that universality of k DFAs, and value boundedness of k unary WFAs reduce to the determinisation problem. We can observe that the reduction from value boundedness to determinisation applies with only minor modification, without assuming the input is a union of deterministic and that the minimal average slope is non-negative.

► **Lemma 33.** *Boundedness and value boundedness reduce to determinisability in polynomial time.*

We can also show that the connection is tight for the value boundedness problem.

► **Lemma 34.** *Determinisability reduces to value boundedness in polynomial time.*

Proof. Given a weighted automaton $\mathcal{A}$, let $\alpha = r/s$, $s \leq |Q|$, $\gcd(r, s) = 1$ be the minimum average slope in $\mathcal{A}$, we define $\mathcal{A}'(n) = s \cdot \mathcal{A}(n) - r \cdot n$. This is obtained by multiplying the weight of every transition by s , and subtracting r .

Since $\mathcal{A}'$ has minimal average slope of zero, we have $\mathcal{A}'$ determinisable iff $\mathcal{A}'$ is value bounded by Claim 30 and Claim 31. The proof is completed by the following claim that $\mathcal{A}$ and $\mathcal{A}'$ are equideterminisable, and so it suffices to determine whether $\mathcal{A}'$ is value bounded.

▷ Claim 35. $\mathcal{A}$ is determinisable if and only if $\mathcal{A}'$ is determinisable. ◀

► **Remark 36** (Remark on Lemma 34). We note that we do not show that determinisability of a unary WFA $\mathcal{A}$ reduces to boundedness (only to value boundedness), unless $\mathcal{A}$ already happens to have a universal domain. Words not in the domain are no issue for determinisability, so it seems we would need to repair these holes by assigning them a bounded value, but we would need to do this without affecting the weight of other words: the natural approach seems to require the addition of an automaton for exactly $\overline{\text{dom}(\mathcal{A})}$ (with weights zero for example); this automaton can have exponential size.

5.4 Proving Theorem 27

Through a chain of reductions, we have observed that NFA universality reduces to boundedness (Lemma 32), reduces to value boundedness (Lemma 33), reduces to determinisation (Lemma 33), which is a special case of CRA dimension reduction/WFA width reduction. The former, NFA universality is coNP-hard [13] while Theorem 18 shows the latter are in coNP.

► **Theorem 27.** *The determinisation, boundedness, value boundedness, register reduction, and width reduction problems for unary weighted automata are coNP-complete.*

6 Future Directions

This paper considers, given unary WFA or their equivalent Cost Register Automata with linear register updates, how to represent them as simply as possible (as a union of deterministic automata, in Chrobak normal form, with lower width, or as CRA using fewer registers).

In general, CRA can be defined over any update function that, in particular, need not be linear [4]. The case of polynomial updates over the tropical semiring is particularly interesting, here the constant exponent essentially provides scalar multiplication alongside minimum and addition. Such models, which are considered in [37], generalise both tropical and rational unary weighted automata, the latter often studied as Linear Recurrence Sequences. Despite their apparent simplicity, LRS have interesting algorithmic problems, particularly the Skolem and Positivity problems which remain wide open [31]. A natural direction is to consider boundedness, register minimisation, and normal forms for these tropical CRA with polynomial updates.

References

- 1 Shaull Almagor, Guy Arbel, and Sarai Sheinvald. Determinization of min-plus weighted automata is decidable. In *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 247–257. SIAM, 2026. doi:10.1137/1.9781611978971.11.
- 2 Shaull Almagor, Guy Arbel, and Sarai Sheinvald. Unambiguability and register minimisation of min-plus models. In *ICALP 2026*, 2026.
- 3 Shaull Almagor, Udi Boker, and Orna Kupferman. What’s decidable about weighted automata? *Inf. Comput.*, 282:104651, 2022. doi:10.1016/J.IC.2020.104651.
- 4 Rajeev Alur, Loris D’Antoni, Jyotirmoy Deshmukh, Mukund Raghothaman, and Yifei Yuan. Regular functions and cost register automata. In *2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 13–22. IEEE, 2013. doi:10.1109/LICS.2013.65.
- 5 Rajeev Alur and Mukund Raghothaman. Decision problems for additive regular functions. In Fedor V. Fomin, Rusins Freivalds, Marta Z. Kwiatkowska, and David Peleg, editors, *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part II*, Lecture Notes in Computer Science, pages 37–48. Springer, 2013. doi:10.1007/978-3-642-39212-2_7.
- 6 Jason P. Bell and Daniel Smertnig. Computing the linear hull: Deciding deterministic? and unambiguous? for weighted automata over fields. In *38th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2023, Boston, MA, USA, June 26-29, 2023*, pages 1–13. IEEE, 2023. doi:10.1109/LICS56636.2023.10175691.
- 7 Yahia Idriss Benalioua, Nathan Lhote, and Pierre-Alain Reynier. Minimizing cost register automata over a field. In Rastislav Kráľovic and Antonín Kucera, editors, *49th International Symposium on Mathematical Foundations of Computer Science, MFCS 2024, Bratislava, Slovakia, August 26-30, 2024*, LIPIcs, pages 23:1–23:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.MFCS.2024.23.
- 8 Dmitry Chistikov, Stefan Kiefer, Andrzej S. Murawski, and David Purser. The big-o problem. *Log. Methods Comput. Sci.*, 18(1), 2022. doi:10.46298/LMCS-18(1:40)2022.
- 9 Christian Choffrut. Une caractérisation des fonctions séquentielles et des fonctions sous-séquentielles en tant que relations rationnelles. *Theoretical Computer Science*, 5(3):325–337, 1977. doi:10.1016/0304-3975(77)90049-4.
- 10 Marek Chrobak. Finite automata and unary languages. *Theor. Comput. Sci.*, 47(3):149–158, 1986. doi:10.1016/0304-3975(86)90142-8.
- 11 Laure Daviaud. Register complexity and determinisation of max-plus automata. *ACM SIGLOG News*, 7(2):4–14, 2020. doi:10.1145/3397619.3397621.
- 12 Karolina Drabik, Anita Dür, Fabian Frei, Filip Mazowiecki, and Karol Wegrzycki. Fined-grained complexity of ambiguity problems on automata and directed graphs. *CoRR*, abs/2501.14725, 2025. doi:10.48550/arXiv.2501.14725.
- 13 Henning Fernau, Stefan Hoffmann, and Michael Wehar. Finite Automata Intersection Non-Emptiness: Parameterized Complexity Revisited, 2021. arXiv:2108.05244.
- 14 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006. doi:10.1007/3-540-29953-X.
- 15 Stéphane Gaubert. Rational series over dioids and discrete event systems. In *11th International Conference on Analysis and Optimization of Systems Discrete Event Systems: Sophia-Antipolis, June 15-16-17, 1994*, pages 247–256. Springer, 2005.
- 16 Hugo Gimbert and Youssouf Oualhadj. Probabilistic automata on finite words: Decidable and undecidable problems. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide, and Paul G. Spirakis, editors, *Automata, Languages and Programming, 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part II*, Lecture Notes in Computer Science, pages 527–538. Springer, 2010. doi:10.1007/978-3-642-14162-1_44.
- 17 Mark Hartmann and Cristina Argüelles. Transience bounds for long walks. *Mathematics of operations research*, 24(2):414–439, 1999. doi:10.1287/MOOR.24.2.414.

- 18 Kosaburo Hashiguchi. Limitedness theorem on finite automata with distance functions. *J. Comput. Syst. Sci.*, 24(2):233–244, 1982. doi:10.1016/0022-0000(82)90051-4.
- 19 Kosaburo Hashiguchi. Regular languages of star height one. *Inf. Control.*, 53(3):199–210, 1982. doi:10.1016/S0019-9958(82)91028-2.
- 20 Ismaël Jecker, Filip Mazowiecki, and David Purser. Determinisation and unambiguisation of polynomially-ambiguous rational weighted automata. In Pawel Sobocinski, Ugo Dal Lago, and Javier Esparza, editors, *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2024, Tallinn, Estonia, July 8-11, 2024*, pages 46:1–46:13. ACM, 2024. doi:10.1145/3661814.3662073.
- 21 Richard M Karp. A characterization of the minimum cycle mean in a digraph. *Discrete mathematics*, 23(3):309–311, 1978. doi:10.1016/0012-365X(78)90011-0.
- 22 Daniel Kirsten and Sylvain Lombardy. Deciding unambiguity and sequentiality of polynomially ambiguous min-plus automata. In *26th International Symposium on Theoretical Aspects of Computer Science STACS 2009*, pages 589–600. IBFI Schloss Dagstuhl, 2009. doi:10.4230/LIPIcs.STACS.2009.1850.
- 23 Ines Klimann, Sylvain Lombardy, Jean Mairesse, and Christophe Prieur. Deciding unambiguity and sequentiality from a finitely ambiguous max-plus automaton. *Theoretical Computer Science*, 327(3):349–373, 2004. doi:10.1016/J.TCS.2004.02.049.
- 24 Peter Kostolányi. Determinisability of unary weighted automata over the rational numbers. *Theor. Comput. Sci.*, 898:110–131, 2022. doi:10.1016/J.TCS.2021.11.002.
- 25 Sylvain Lombardy. Sequentialization and unambiguity of $(\max,+)$ rational series over one letter. In *Workshop on max-plus algebras and Their Applications to Discrete-event Systems, Theoretical Computer Science, and Optimization*, page 6pp. IFAC, Elsevier Sciences, 2001.
- 26 Sylvain Lombardy and Jacques Sakarovitch. Sequential? *Theor. Comput. Sci.*, 356(1-2):224–244, 2006. doi:10.1016/J.TCS.2006.01.028.
- 27 Andrew Martinez. Efficient computation of regular expressions from unary nfes. In Jürgen Dassow, Maia Hoeberechts, Helmut Jürgensen, and Detlef Wotschke, editors, *Fourth International Workshop on Descriptive Complexity of Formal Systems - DCFS 2002, London, Canada, August 21 - 24, 2002. Pre-proceedings*, pages 174–187. Department of Computer Science, The University of Western Ontario, Canada, 2002.
- 28 Mehryar Mohri. Compact representations by finite-state transducers. In *32nd Annual Meeting of the Association for Computational Linguistics*, pages 204–209, 1994. doi:10.3115/981732.981760.
- 29 Mehryar Mohri. Finite-state transducers in language and speech processing. *Computational linguistics*, 23(2):269–311, 1997.
- 30 Karl Nachtigall. Powers of matrices over an extremal algebra with applications to periodic graphs. *Mathematical Methods of Operations Research*, 46(1):87–102, 1997. doi:10.1007/BF01199464.
- 31 Joël Ouaknine and James Worrell. On linear recurrence sequences and loop termination. *ACM SIGLOG News*, 2(2):4–13, 2015. doi:10.1145/2766189.2766191.
- 32 Christos H. Papadimitriou and Mihalis Yannakakis. The complexity of facets (and some facets of complexity). *J. Comput. Syst. Sci.*, 28(2):244–259, 1984. doi:10.1016/0022-0000(84)90068-0.
- 33 Antoni Puch and Daniel Smertnig. Factoring through monomial representations: arithmetic characterizations and ambiguity of weighted automata. *CoRR*, abs/2410.03444, 2024. doi:10.48550/arXiv.2410.03444.
- 34 Charles Rackoff. The covering and boundedness problems for vector addition systems. *Theor. Comput. Sci.*, 6:223–231, 1978. doi:10.1016/0304-3975(78)90036-1.
- 35 Marcel Paul Schützenberger. On the definition of a family of automata. *Inf. Control.*, 4(2-3):245–270, 1961. doi:10.1016/S0019-9958(61)80020-X.
- 36 Sergei Sergeev and Hans Schneider. Csr expansions of matrix powers in max algebra. *Transactions of the American Mathematical society*, 364(11):5969–5994, 2012.

- 37 Gerco van Heerdt, Justin Hsu, Joël Ouaknine, and Alexandra Silva. Convex language semantics for nondeterministic probabilistic automata. *Theor. Comput. Sci.*, 1040:115191, 2025. doi:10.1016/J.TCS.2025.115191.
- 38 Michael Wehar. *On the Complexity of Intersection Non-Emptiness Problems*. PhD thesis, University at Buffalo, State University of New York, USA, 2016. URL: http://www.michaelwehar.com/documents/mwehar_dissertation.pdf.
- 39 Anthony Widjaja To (Lin). Unary finite automata vs. arithmetic progressions. *Inf. Process. Lett.*, 109(17):1010–1014, 2009. doi:10.1016/J.IPL.2009.06.005.