Bideterministic Weighted Automata*

Peter Kostolányi[0000−0002−5474−8781]

Department of Computer Science, Comenius University in Bratislava
Mlynská dolina, 842 48 Bratislava, Slovakia
kostolanyi@fmph.uniba.sk

Abstract. A deterministic finite automaton is called bideterministic if its transpose is deterministic as well. The study of such automata in a weighted setting is initiated. All trim bideterministic weighted automata over integral domains and positive semirings are proved to be minimal. On the contrary, it is observed that this property does not hold over finite commutative rings in general. Moreover, it is shown that the problem of determining whether a given rational series is realised by a bideterministic automaton is decidable over fields as well as over tropical semirings.

Keywords: Bideterministic weighted automaton · Minimal automaton · Integral domain · Positive semiring · Decidability

1 Introduction

It is well known that – in contrast to the classical case of automata without weights – weighted finite automata might not always be determinisable. Partly due to relevance of deterministic weighted automata for practical applications such as natural language and speech processing [24] and partly due to the purely theoretical importance of the determinisability problem, questions related to deterministic weighed automata – such as the decidability of determinisability, existence of efficient determinisation algorithms, or characterisations of series realised by deterministic weighted automata – have received significant attention. They were studied for weighted automata over specific classes of semirings, such as tropical semirings or fields [1, 6, 18–21, 24, 25], as well as over strong bimonoids [9], often under certain additional restrictions.

The questions mentioned above are known to be relatively hard. For instance, despite some partial results [18–21], the decidability status of the general determinisability problem for weighted automata is still open over tropical semirings or over the field of rationals [21]. It thus makes sense to take a look at stronger forms of determinism in weighted automata, which may be amenable to a somewhat easier analysis.

* The work was supported by the grant VEGA 1/0601/20.

One possibility is to study deterministic weighted automata with additional requirements on their weights. This includes for instance the research on crisp-deterministic weighted automata by M. Ćirić et al. [9]. Another possibility is to examine the weighted counterpart of some particularly simple subclass of deterministic finite automata without weights – that is, to impose further restrictions not only on weights of deterministic weighted automata, but on the concept of determinism itself. This is a direction that we follow in this article.

More tangibly, this article aims to initiate the study of *bideterministic* finite automata in the weighted setting. A finite automaton is bideterministic if it is deterministic and its transpose – *i.e.*, an automaton obtained by reversing all transitions and exchanging the roles of its initial and terminal states – is deterministic as well. Note that this implies that a bideterministic automaton always contains at most one initial and at most one terminal state. Bideterministic finite automata have been first touched upon from a theoretical perspective by J.-É. Pin [27], as a particular case of reversible finite automata. The fundamental properties of bideterministic finite automata have mostly been explored by H. Tamm and E. Ukkonen [35, 36] – in particular, they have shown that a trim bideterministic automaton is always a minimal nondeterministic automaton for the language it recognises, minimality being understood in the strong sense, *i.e.*, with respect to the number of states.[1] An alternative proof of this fact was recently presented by R. S. R. Myers, S. Milius, and H. Urbat [26].

Apart from these studies, bideterministic automata have been – explicitly or implicitly – considered in connection to the star height problem [22, 23, 14], from the perspective of language inference [3], in the theory of block codes [33], and in connection to presentations of inverse monoids [34, 16].

We define *bideterministic weighted automata* over a semiring by analogy to their unweighted counterparts, and study the conditions under which the fundamental property of H. Tamm and E. Ukkonen [35, 36] generalises to the weighted setting. Thus, given a semiring S, we ask the following questions: Are all trim bideterministic weighted automata over S minimal? Does every bideterministic automaton over S admit a bideterministic equivalent that is at the same time minimal? We answer both these questions in affirmative when S is an integral domain or a positive – *i.e.*, both zero-sum free and zero-divisor free – semiring. On the other hand, we show that the answer is negative for a large class of commutative semirings including a multitude of *finite* commutative *rings*.

Finally, we consider the problem of deciding whether a weighted automaton over a semiring S admits a bideterministic equivalent, and show that it is decidable when S is a field or a tropical semiring (of nonnegative integers, integers, or rationals). This suggests that the bideterminisability problem for weighted automata might be somewhat easier than the determinisability problem, whose decidability status over fields such as the rationals and over tropical semirings remains open [21].

[1] In fact, H. Tamm and E. Ukkonen [35, 36] have shown a stronger property: a trim bideterministic automaton is the *only* minimal nondeterministic finite automaton recognising its language.

2 Preliminaries

A *semiring* is a quintuple $(S, +, \cdot, 0, 1)$ such that $(S, +, 0)$ is a commutative monoid, $(S, \cdot, 1)$ is a monoid, multiplication distributes over addition both from left and from right, and $a \cdot 0 = 0 \cdot a = 0$ holds for all $a \in S$; it is said to be *commutative* when $\cdot$ is. A semiring S is *zero-sum free* [13, 15] if $a+b = 0$ for some $a, b \in S$ implies $a = b = 0$ and *zero-divisor free* [15], or *entire* [13], if $a \cdot b = 0$ for some $a, b \in S$ implies that $a = 0$ or $b = 0$. A semiring is *positive* [12, 17] if it is both zero-sum free and zero-divisor free. A *ring* is a semiring $(R, +, \cdot, 0, 1)$ such that R forms an abelian group with addition. An *integral domain* is a nontrivial zero-divisor free commutative ring. A *field* is an integral domain $(\mathbb{F}, +, \cdot, 0, 1)$ such that $\mathbb{F} \setminus \{0\}$ forms an abelian group with multiplication.

We now briefly recall some basic facts about noncommutative formal power series and weighted automata. More information can be found in [7, 10, 11, 29]. Alphabets are assumed to be finite and nonempty in what follows.

A *formal power series* over a semiring S and alphabet Σ is a mapping $r \colon \Sigma^* \to S$. The value of r upon $w \in \Sigma^*$ is usually denoted by (r, w) and called the *coefficient* of r at w; the coefficient of r at ε, the empty word, is referred to as the *constant coefficient*. The series r itself is written as

$$r = \sum_{w \in \Sigma^*} (r, w)\, w.$$

The set of all formal power series over S and Σ is denoted by $S\langle\langle \Sigma^* \rangle\rangle$.

Given series $r, s \in S\langle\langle \Sigma^* \rangle\rangle$, their *sum* $r + s$ and *product* $r \cdot s$ are defined by $(r + s, w) = (r, w) + (s, w)$ and

$$(r \cdot s, w) = \sum_{\substack{u, v \in \Sigma^* \\ uv = w}} (r, u)(s, v)$$

for all $w \in \Sigma^*$. Every $a \in S$ is identified with a series with constant coefficient a and all other coefficients zero, and every $w \in \Sigma^*$ with a series with coefficient 1 at w and zero coefficients at all $x \in \Sigma^* \setminus \{w\}$. Thus, for instance, $r = 2ab + 3abb$ is a series with $(r, ab) = 2$, $(r, abb) = 3$, and $(r, x) = 0$ for every $x \in \Sigma^* \setminus \{ab, abb\}$. One may observe that $(S\langle\langle \Sigma^* \rangle\rangle, +, \cdot, 0, 1)$ is a semiring again.

For I an index set, a family $(r_i \mid i \in I)$ of series from $S\langle\langle \Sigma^* \rangle\rangle$ is *locally finite* if $I(w) = \{i \in I \mid (r_i, w) \neq 0\}$ is finite for all $w \in \Sigma^*$. The *sum* over the family $(r_i \mid i \in I)$ can then be defined by

$$\sum_{i \in I} r_i = r,$$

where the coefficient (r, w) at each $w \in \Sigma^*$ is given by a *finite* sum

$$(r, w) = \sum_{i \in I(w)} (r_i, w).$$

The *support* of $r \in S\langle\langle \Sigma^* \rangle\rangle$ is the language $\operatorname{supp}(r) = \{w \in \Sigma^* \mid (r, w) \neq 0\}$. The *left quotient* of $r \in S\langle\langle \Sigma^* \rangle\rangle$ by a word $x \in \Sigma^*$ is a series $x^{-1}r$ such that $(x^{-1}r, w) = (r, xw)$ for all $w \in \Sigma^*$.

A *weighted (finite) automaton* over a semiring S and alphabet Σ is a quadruple $\mathcal{A} = (Q, \sigma, \iota, \tau)$, where Q is a finite set of states, $\sigma\colon Q \times \Sigma \times Q \to S$ a transition weighting function, $\iota\colon Q \to S$ an initial weighting function, and $\tau\colon Q \to S$ a terminal weighting function. We often assume without loss of generality that $Q = [n] = \{1, \ldots, n\}$ for some nonnegative integer n; we write $\mathcal{A} = (n, \sigma, \iota, \tau)$ instead of $\mathcal{A} = ([n], \sigma, \iota, \tau)$ in that case.

A *transition* of $\mathcal{A} = (Q, \sigma, \iota, \tau)$ is a triple $(p, c, q) \in Q \times \Sigma \times Q$ such that $\sigma(p, c, q) \neq 0$. A *run* of $\mathcal{A}$ is a word $\gamma = q_0 c_1 q_1 c_2 q_2 \ldots q_{n-1} c_n q_n \in (Q\Sigma)^* Q$, for some nonnegative integer n, such that $q_0, \ldots, q_n \in Q$, $c_1, \ldots, c_n \in \Sigma$, and (q_{k-1}, c_k, q_k) is a transition for $k = 1, \ldots, n$; we also say that γ is a run *from* q_0 *to* q_n. Moreover, we write $\lambda(\gamma) = c_1 c_2 \ldots c_n \in \Sigma^*$ for the *label* of γ and $\sigma(\gamma) = \sigma(q_0, c_1, q_1)\sigma(q_1, c_2, q_2) \ldots \sigma(q_{n-1}, c_n, q_n) \in S$ for the *value* of γ. The *monomial* $\|\gamma\| \in S\langle\langle \Sigma^* \rangle\rangle$ realised by the run γ is defined by

$$\|\gamma\| = (\iota(q_0)\sigma(\gamma)\tau(q_n))\, \lambda(\gamma).$$

If we denote by $\mathcal{R}(\mathcal{A})$ the set of all runs of the automaton $\mathcal{A}$, then the family of monomials $(\|\gamma\| \mid \gamma \in \mathcal{R}(\mathcal{A}))$ is obviously locally finite and the *behaviour* of $\mathcal{A}$ can be defined by the infinite sum

$$\|\mathcal{A}\| = \sum_{\gamma \in \mathcal{R}(\mathcal{A})} \|\gamma\|.$$

In particular, $\|\mathcal{A}\| = 0$ if $Q = \emptyset$. A series $r \in S\langle\langle \Sigma^* \rangle\rangle$ is *rational* over S if $r = \|\mathcal{A}\|$ for some weighted automaton $\mathcal{A}$ over S and Σ.

A state $q \in Q$ of a weighted automaton $\mathcal{A} = (Q, \sigma, \iota, \tau)$ over S and Σ is said to be *accessible* if there is a run in $\mathcal{A}$ from some $p \in Q$ satisfying $\iota(p) \neq 0$ to q.[2] Dually, a state $q \in Q$ is *coaccessible* if there is a run in $\mathcal{A}$ from q to some $p \in Q$ such that $\tau(p) \neq 0$. The automaton $\mathcal{A}$ is *trim* if all its states are both accessible and coaccessible [29].

Given a weighted automaton $\mathcal{A} = (Q, \sigma, \iota, \tau)$ and $q \in Q$, we denote by $\|\mathcal{A}\|_q$ the *future* of q, *i.e.*, the series realised by an automaton $\mathcal{A}_q = (Q, \sigma, \iota_q, \tau)$ where $\iota_q(q) = 1$ and $\iota_q(p) = 0$ for all $p \in Q \setminus \{q\}$.

Let $S^{m \times n}$ be the set of all $m \times n$ matrices over S. A *linear representation* of a weighted automaton $\mathcal{A} = (n, \sigma, \iota, \tau)$ over S and Σ is given by $\mathcal{P}_{\mathcal{A}} = (n, \mathbf{i}, \mu, \mathbf{f})$, where $\mathbf{i} = (\iota(1), \ldots, \iota(n))$, $\mu\colon (\Sigma^*, \cdot) \to (S^{n \times n}, \cdot)$ is a monoid homomorphism such that for all $c \in \Sigma$ and $i, j \in [n]$, the entry of $\mu(c)$ in the i-th row and j-th column is given by $\sigma(i, c, j)$, and $\mathbf{f} = (\tau(1), \ldots, \tau(n))^T$. The representation $\mathcal{P}_{\mathcal{A}}$ describes $\mathcal{A}$ unambiguously, and $(\|\mathcal{A}\|, w) = \mathbf{i}\mu(w)\mathbf{f}$ holds for all $w \in \Sigma^*$.

As a consequence of this connection to linear representations, methods of linear algebra can be employed in the study of weighted automata *over fields*. This leads to a particularly well-developed theory, including a polynomial-time minimisation algorithm, whose basic ideas go back to M.-P. Schützenberger [32] and which has been explicitly described by A. Cardon and M. Crochemore [8]. The reader may consult [7, 29, 30] for a detailed exposition.

[2] Note that the value of this run might be zero in case S is not zero-divisor free.

For our purposes, we only note that the gist of this minimisation algorithm lies in an observation that given a weighted automaton $\mathcal{A}$ over a field $\mathbb{F}$ and alphabet Σ with $\mathcal{P}_{\mathcal{A}} = (n, \mathbf{i}, \mu, \mathbf{f})$, one can find in polynomial time a finite language $L = \{x_1, \ldots, x_m\}$ of words over Σ that is prefix-closed, and the vectors $\mathbf{i}\mu(x_1), \ldots, \mathbf{i}\mu(x_m)$ form a basis of the vector subspace $\mathrm{Left}(\mathcal{A})$ of $\mathbb{F}^{1\times n}$ generated by the vectors $\mathbf{i}\mu(x)$ with $x \in \Sigma^*$. Such a language L is called a *left basic language* of $\mathcal{A}$. Similarly, one can find in polynomial time a *right basic language* of $\mathcal{A}$ – *i.e.*, a finite language $R = \{y_1, \ldots, y_k\}$ of words over Σ that is suffix-closed, and the vectors $\mu(y_1)\mathbf{f}, \ldots, \mu(y_k)\mathbf{f}$ form a basis of the vector subspace $\mathrm{Right}(\mathcal{A})$ of $\mathbb{F}^{n\times 1}$ generated by the vectors $\mu(y)\mathbf{f}$ with $y \in \Sigma^*$.

The actual minimisation algorithm then consists of two reduction steps. The original weighted automaton $\mathcal{A}$ with representation $\mathcal{P}_{\mathcal{A}} = (n, \mathbf{i}, \mu, \mathbf{f})$ is first transformed into an equivalent automaton $\mathcal{B}$ with $\mathcal{P}_{\mathcal{B}} = (k, \mathbf{i}', \mu', \mathbf{f}')$. Here, $k \leq n$ is the size of the right basic language $R = \{y_1, \ldots, y_k\}$ of $\mathcal{A}$ with $y_1 = \varepsilon$,

$$\mathbf{i}' = \mathbf{i}Y, \quad \mu'(c) = Y_\ell^{-1}\mu(c)Y \text{ for all } c \in \Sigma, \quad \text{and} \quad \mathbf{f}' = (1, 0, \ldots, 0)^T, \tag{1}$$

where $Y \in \mathbb{F}^{n\times k}$ is a matrix of full column rank with columns $\mu(y_1)\mathbf{f}, \ldots, \mu(y_k)\mathbf{f}$ and $Y_\ell^{-1} \in \mathbb{F}^{k\times n}$ is its left inverse matrix. The automaton $\mathcal{B}$ is then transformed into a *minimal* equivalent automaton $\mathcal{C}$ with $\mathcal{P}_{\mathcal{C}} = (m, \mathbf{i}'', \mu'', \mathbf{f}'')$. Here, $m \leq k$ is the size of the left basic language $L = \{x_1, \ldots, x_m\}$ of $\mathcal{B}$ with $x_1 = \varepsilon$,

$$\mathbf{i}'' = (1, 0, \ldots, 0), \quad \mu''(c) = X\mu'(c)X_r^{-1} \text{ for all } c \in \Sigma, \quad \text{and} \quad \mathbf{f}'' = X\mathbf{f}', \tag{2}$$

where $X \in \mathbb{F}^{m\times k}$ is a matrix of full row rank with rows $\mathbf{i}'\mu'(x_1), \ldots, \mathbf{i}'\mu'(x_m)$ and X_r^{-1} is its right inverse matrix. As the vector space $\mathrm{Left}(\mathcal{B})$ – which is the row space of X – is invariant under $\mu'(c)$ for all $c \in \Sigma$, it follows that

$$\mathbf{i}''X = \mathbf{i}', \quad \mu''(c)X = X\mu'(c) \text{ for all } c \in \Sigma, \quad \text{and} \quad \mathbf{f}'' = X\mathbf{f}', \tag{3}$$

showing that the automaton $\mathcal{C}$ is *conjugate* [4, 5] to $\mathcal{B}$ by the matrix X. Thus $\mathbf{i}''\mu''(x)X = \mathbf{i}'\mu'(x)$ for all $x \in \Sigma^*$, so that the vector $\mathbf{i}''\mu''(x)$ represents the coordinates of $\mathbf{i}'\mu'(x)$ with respect to the basis $(\mathbf{i}'\mu'(x_1), \ldots, \mathbf{i}'\mu'(x_m))$ of $\mathrm{Left}(\mathcal{B})$. In particular, note that $(\mathbf{i}''\mu''(x_1), \ldots, \mathbf{i}''\mu''(x_m))$ is the standard basis of $\mathbb{F}^m$.

Finally, let us mention that any weighted automaton $\mathcal{A}$ over $\mathbb{F}$ and Σ with $\mathcal{P}_{\mathcal{A}} = (n, \mathbf{i}, \mu, \mathbf{f})$ gives rise to a linear mapping $\Lambda[\mathcal{A}]\colon \mathrm{Left}(\mathcal{A}) \to \mathbb{F}\langle\langle \Sigma^* \rangle\rangle$, uniquely defined by

$$\Lambda[\mathcal{A}]\colon \mathbf{i}\mu(x) \mapsto \sum_{w\in\Sigma^*} (\mathbf{i}\mu(x)\mu(w)\mathbf{f})\ w = x^{-1}\|\mathcal{A}\| \tag{4}$$

for all $x \in \Sigma^*$. This mapping is always injective when $\mathcal{A}$ is a minimal automaton realising its behaviour [30].

3 Bideterministic Weighted Automata over a Semiring

In the same way as for finite automata without weights [35, 36], we say that a weighted automaton $\mathcal{A}$ is *bideterministic* if both $\mathcal{A}$ and its transpose are deterministic; in particular, $\mathcal{A}$ necessarily contains at most one state with nonzero initial weight and at most one state with nonzero terminal weight. This is made more precise by the following definition.

Definition 1. *Let S be a semiring and Σ an alphabet. A weighted automaton $\mathcal{A} = (Q, \sigma, \iota, \tau)$ over S and Σ is* bideterministic *if all of the following conditions are satisfied:*

(*i*) *There is at most one state $p \in Q$ such that $\iota(p) \neq 0$.*
(*ii*) *If $\sigma(p, c, q) \neq 0$ and $\sigma(p, c, q') \neq 0$ for $p, q, q' \in Q$ and $c \in \Sigma$, then $q = q'$.*
(*iii*) *There is at most one state $q \in Q$ such that $\tau(q) \neq 0$.*
(*iv*) *If $\sigma(p, c, q) \neq 0$ and $\sigma(p', c, q) \neq 0$ for $p, p', q \in Q$ and $c \in \Sigma$, then $p = p'$.*

The conditions (*i*) and (*ii*) assure that the automaton $\mathcal{A}$ is *deterministic*, while the conditions (*iii*) and (*iv*) assure the same property for its transpose.

It has been shown by H. Tamm and E. Ukkonen [35, 36] that a trim bideterministic automaton without weights is always a minimal nondeterministic automaton for the language it recognises. As a consequence, every language recognised by some bideterministic automaton also admits a minimal automaton that is bideterministic. Moreover, by uniqueness of minimal deterministic finite automata and existence of efficient minimisation algorithms, it follows that it is decidable whether a language is recognised by a bideterministic automaton.

In what follows, we ask whether these properties generalise to bideterministic weighted automata over some semiring S. That is, given a semiring S, we are interested in the following three questions.[3]

Question 1. Is every trim bideterministic weighted automaton over S necessarily minimal?

Question 2. Does every bideterministic automaton over S admit an equivalent minimal weighted automaton over S that is bideterministic?

Question 3. Is it decidable whether a weighted automaton over S admits a bideterministic equivalent?

An affirmative answer to Question 1 clearly implies an affirmative answer to Question 2 as well. We study the first two questions in Section 4 and the last question in Section 5.

4 The Minimality Property of Bideterministic Automata

We now study the conditions on a semiring S under which the trim bideterministic weighted automata over S are always minimal, and answer the Question 1, as well as the related Question 2, for three representative classes of semirings. In particular, we show that every trim bideterministic weighted automaton over a *field* – or, more generally, over an *integral domain* – is minimal. The same property is observed for bideterministic weighted automata over *positive semirings*, including for instance the *tropical semirings* and *semirings of formal languages*. On the other hand, we prove that both questions have negative answers over a large class of commutative semirings other than integral domains, which also includes numerous *finite commutative rings*.

[3] Minimality of an automaton is understood with respect to the number of states in what follows.

4.1 Fields and Integral Domains

The minimality property of trim bideterministic weighted automata *over fields* follows by the fact that the Cardon-Crochemore minimisation algorithm for these automata, described in Section 2, preserves both bideterminism and the number of useful states of a bideterministic automaton, as we now observe.

Theorem 2. *Let $\mathcal{A}$ be a bideterministic weighted automaton over a field $\mathbb{F}$. Then the Cardon-Crochemore minimisation algorithm applied to $\mathcal{A}$ outputs a bideterministic weighted automaton $\mathcal{C}$. Moreover, if $\mathcal{A}$ trim, then $\mathcal{C}$ has the same number of states as $\mathcal{A}$.*

Proof. Let $\mathcal{P} = (n, \mathbf{i}, \mu, \mathbf{f})$ be a linear representation of some bideterministic weighted automaton $\mathcal{D}$. Then there is at most one nonzero entry in each row and column of $\mu(c)$ for each $c \in \Sigma$, and at most one nonzero entry in $\mathbf{i}$ and $\mathbf{f}$.

Moreover, the words $x_1, \ldots, x_m$ of the left basic language of $\mathcal{D}$ correspond bijectively to accessible states of $\mathcal{D}$ and the vector $\mathbf{i}\mu(x_i)$ contains, for $i = 1, \ldots, m$, exactly one nonzero entry at the position determined by the state corresponding to x_i. Similarly, the words $y_1, \ldots, y_k$ of the right basic language of $\mathcal{D}$ correspond to coaccessible states and the vector $\mu(y_i)\mathbf{f}$ contains, for $i = 1, \ldots, k$, exactly one nonzero entry. Thus, using these vectors to form the matrices X and Y as in Section 2, we see that one obtains monomial matrices after removing the zero columns from X and the zero rows from Y. As a result, a right inverse X_r^{-1} of X can be obtained by taking the reciprocals of all nonzero entries of X and transposing the resulting matrix, and similarly for a left inverse Y_ℓ^{-1} of Y.

The matrices $X\mu(c)X_r^{-1}$ and $Y_\ell^{-1}\mu(c)Y$ for $c \in \Sigma^*$ clearly contain at most one nonzero entry in each row and column, and the vectors $\mathbf{i}Y$ and $X\mathbf{f}$ contain at most one nonzero entry as well. This means that the reduction step (1) applied to a bideterministic automaton $\mathcal{A}$ yields a bideterministic automaton $\mathcal{B}$, and that the reduction step (2) applied to the bideterministic automaton $\mathcal{B}$ yields a bideterministic minimal automaton $\mathcal{C}$ as an output of the algorithm.

When $\mathcal{A}$ is in addition trim, then what has been said implies that the words of the right basic language of $\mathcal{A}$ correspond bijectively to states of $\mathcal{A}$, so that the automaton $\mathcal{B}$ obtained via (1) has the same number of states as $\mathcal{A}$. This automaton is obviously trim as well, and the words of the left basic language of $\mathcal{B}$ correspond bijectively to states of $\mathcal{B}$. Hence, the automaton $\mathcal{C}$ obtained via (2) also has the same number of states as $\mathcal{A}$. □

As every integral domain can be embedded into its field of fractions, the property established above holds for automata over integral domains as well.

Corollary 3. *Every trim bideterministic weighted automaton over an integral domain is minimal.*

4.2 Other Commutative Rings

We now show that the property established above for automata over integral domains *cannot* be generalised to automata over commutative rings, by exhibiting a suitable class of commutative *semirings* S such that bideterministic weighted automata over S do not even always admit a minimal bideterministic equivalent.

Theorem 4. *Let S be a commutative semiring with elements $s, t \in S$ such that $st = 0$ and $s^2 \neq 0 \neq t^2$. Then there is a trim bideterministic weighted automaton $\mathcal{A}$ over S such that none of the minimal automata for $\|\mathcal{A}\|$ is bideterministic.*

Proof. Consider a trim bideterministic weighted automaton $\mathcal{A}$ over S depicted in Fig. 1. Clearly, $\|\mathcal{A}\| = s^2 \cdot aba + t^2 \cdot bb$. The automaton $\mathcal{A}$ is not minimal, as the same series is realised by a smaller automaton $\mathcal{B}$ in Fig. 2:

$$\|\mathcal{B}\| = s^2 \cdot aba + t^2 \cdot bb = \|\mathcal{A}\|.$$

The answer to Question 1 of Section 3 is thus negative over S.

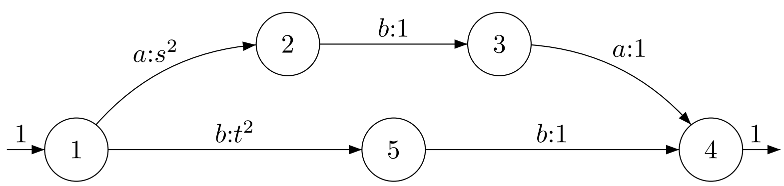

Fig. 1: The trim bideterministic weighted automaton $\mathcal{A}$ over S.

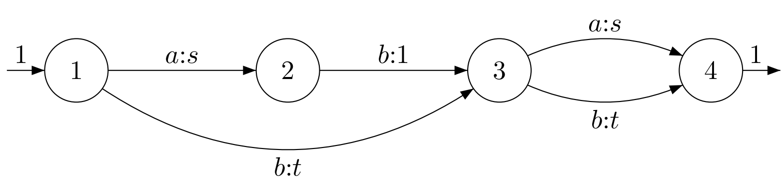

Fig. 2: The four-state weighted automaton $\mathcal{B}$ over S equivalent to $\mathcal{A}$.

We show that $\|\mathcal{A}\|$ is actually not realised by any bideterministic weighted automaton over S with less than five states. This implies that $\mathcal{A}$ is a counterexample to Question 2 of Section 3, and eventually completes the proof.

Indeed, consider a bideterministic weighted automaton $\mathcal{C} = (Q, \sigma, \iota, \tau)$ such that $\|\mathcal{C}\| = \|\mathcal{A}\|$. At least one state with nonzero initial weight is needed to realise $\|\mathcal{A}\|$ by $\mathcal{C}$, as $\|\mathcal{A}\| \neq 0$. Let us call this state 1.

As $(\|\mathcal{A}\|, aba) = s^2 \neq 0$, there is a transition on a in $\mathcal{C}$ leading from 1. This cannot be a loop at 1, as otherwise ba would have a nonzero coefficient in $\|\mathcal{C}\|$, contradicting $\|\mathcal{C}\| = \|\mathcal{A}\|$. It thus leads to some new state, say, 2.

There has to be a transition on b leading from 2 and in the same way as above, we observe that it can lead neither to 1, nor to 2, as otherwise a or aa would have a nonzero coefficient in $\|\mathcal{C}\|$. It thus leads to some new state 3.

Exactly the same reasoning gives us existence of another state 4, to which a transition on a leads from 3, and which has a nonzero terminal weight $\tau(4)$.

Existence of one more state has to be established in order to finish the proof. To this end, observe that $(\|\mathcal{A}\|, bb) = t^2 \neq 0$, so that $\mathcal{C}$ has a transition from 1 on b, which cannot be a loop at 1, as otherwise b would have a nonzero coefficient in $\|\mathcal{C}\|$. This transition cannot lead to 2 either, as there already is a transition on b from 2 to 3, so that bb would have coefficient 0 in $\|\mathcal{C}\|$. Likewise, it cannot lead to 3, as there already is a transition on b from 2 to 3 and $\mathcal{C}$ is supposed to be bideterministic. Finally, it also cannot lead to 4, as otherwise there would have to be a loop labelled by b at 4 and b would have a nonzero coefficient in $\|\mathcal{C}\|$. The transition on b from 1 thus indeed leads to some new state 5. □

Note that the class of commutative semirings from Theorem 4 also includes many *finite* commutative *rings*. In particular, the ring $\mathbb{Z}_m$ of integers modulo m falls into this class whenever m has at least two distinct prime factors. The characterisation of commutative rings, over which all trim bideterministic weighted automata are minimal, remains open. It would have been nice to know at least what the situation is over finite rings $\mathbb{Z}_{p^n}$ for p prime and $n \geq 2$.

4.3 Positive Semirings

We now observe that the minimality property does hold for trim bideterministic weighted automata *over positive semirings*. Recall that a semiring is *positive* if it is both zero-sum free and zero-divisor free. This class includes for instance the *tropical semirings*, *semirings of formal languages*, and the *Boolean semiring*.

Theorem 5. *Every trim bideterministic weighted automaton over a positive semiring is minimal.*

Proof. Let $\mathcal{A}$ be a trim bideterministic weighted automaton over a positive semiring S. By positivity of S, the language $\mathrm{supp}(\|\mathcal{A}\|)$ is recognised by a trim bideterministic finite automaton $\mathcal{A}'$ obtained from $\mathcal{A}$ by "forgetting about weights". This is a minimal nondeterministic automaton for $\mathrm{supp}(\|\mathcal{A}\|)$ by the minimality property of trim bideterministic automata without weights [35, 36].

Now, if $\mathcal{A}$ was not minimal, there would be a smaller weighted automaton $\mathcal{B}$ over S such that $\|\mathcal{B}\| = \|\mathcal{A}\|$. By "forgetting about its weights", we would obtain a nondeterministic finite automaton $\mathcal{B}'$ recognising $\mathrm{supp}(\|\mathcal{B}\|) = \mathrm{supp}(\|\mathcal{A}\|)$. However, $\mathcal{B}'$ is smaller than $\mathcal{A}'$, contradicting the minimality of $\mathcal{A}'$. □

5 Decidability of Bideterminisability

Let us now consider the problem of deciding whether a given weighted automaton admits a bideterministic equivalent. While the decidability status of the *determinisability* problem is open both over fields such as the rationals and over tropical semirings [21], we prove that the *bideterminisability* problem is decidable both over effective fields and over tropical semirings (of nonnegative integers, integers, and rationals).

5.1 Fields

We prove decidability of the bideterminisability problem for automata over fields by strengthening Theorem 2 – we show that the Cardon-Crochemore minimisation algorithm outputs a bideterministic automaton not only when applied to a bideterministic automaton, but also when applied to any bideterminisable automaton. To decide bideterminisability, it thus suffices to run this algorithm and find out whether its output is bideterministic.

Lemma 6. *Let $\mathcal{A}$ be a weighted automaton over a field $\mathbb{F}$ such that some of the minimal automata equivalent to $\mathcal{A}$ is deterministic. Then the Cardon-Crochemore algorithm applied to $\mathcal{A}$ outputs a deterministic automaton.*

Proof. Let $\mathcal{C}$ with $\mathcal{P}_{\mathcal{C}} = (m, \mathbf{i}, \mu, \mathbf{f})$ be the output of the Cardon-Crochemore algorithm upon $\mathcal{A}$ and $L = \{x_1, \ldots, x_m\}$ with $x_1 = \varepsilon$ the left basic language used in reduction step (2). Then $\mathbf{i}\mu(x)$ represents, for all $x \in \Sigma^*$, the coordinates of the series $x^{-1}\|\mathcal{A}\|$ with respect to the basis $(x_1^{-1}\|\mathcal{A}\|, \ldots, x_m^{-1}\|\mathcal{A}\|)$ of the vector space $\mathcal{Q}(\|\mathcal{A}\|)$ generated by left quotients of $\|\mathcal{A}\|$ by words.

To see this, recall that $(\mathbf{i}\mu(x_1), \ldots, \mathbf{i}\mu(x_m))$ is the standard basis of $\mathbb{F}^m$ and that the linear mapping $\Lambda[\mathcal{C}]$ given as in (4) is injective by minimality of $\mathcal{C}$. As the image of $\Lambda[\mathcal{C}]$ spans $\mathcal{Q}(\|\mathcal{C}\|) = \mathcal{Q}(\|\mathcal{A}\|)$, we see that

$$\left(x_1^{-1}\|\mathcal{A}\|, \ldots, x_m^{-1}\|\mathcal{A}\|\right) = (\Lambda[\mathcal{C}](\mathbf{i}\mu(x_1)), \ldots, \Lambda[\mathcal{C}](\mathbf{i}\mu(x_m)))$$

is indeed a basis of $\mathcal{Q}(\|\mathcal{A}\|)$. Moreover, given an arbitrary word $x \in \Sigma^*$ with $\mathbf{i}\mu(x) = (a_1, \ldots, a_m) \in \mathbb{F}^m$, we obtain

$$\begin{aligned} x^{-1}\|\mathcal{A}\| &= \Lambda[\mathcal{C}](\mathbf{i}\mu(x)) = \Lambda[\mathcal{C}](a_1\mathbf{i}\mu(x_1) + \ldots + a_m\mathbf{i}\mu(x_m)) = \\ &= a_1\Lambda[\mathcal{C}](\mathbf{i}\mu(x_1)) + \ldots + a_m\Lambda[\mathcal{C}](\mathbf{i}\mu(x_m)) = \\ &= a_1x_1^{-1}\|\mathcal{A}\| + \ldots + a_mx_m^{-1}\|\mathcal{A}\|, \end{aligned}$$

from which the said property follows.

Now, assume for contradiction that $\mathcal{C}$ is not deterministic. By minimality of $\mathcal{C}$, there is some $x \in \Sigma^*$ such that $\mathbf{i}\mu(x)$ contains at least two nonzero entries. However, by our assumptions, there also is an m-state *deterministic* automaton $\mathcal{D}$ such that $\|\mathcal{D}\| = \|\mathcal{A}\|$. Linear independence of $x_1^{-1}\|\mathcal{A}\|, \ldots, x_m^{-1}\|\mathcal{A}\|$ implies that the m states of $\mathcal{D}$ can be labelled as $q_1, \ldots, q_m$ so that $x_i^{-1}\|\mathcal{A}\|$ is a scalar multiple of $\|\mathcal{D}\|_{q_i}$ for $i = 1, \ldots, m$. By determinism of $\mathcal{D}$, every $x^{-1}\|\mathcal{A}\|$ with $x \in \Sigma^*$ is a scalar multiple of some $\|\mathcal{D}\|_{q_i}$ with $i \in [m]$, and hence also of some $x_i^{-1}\|\mathcal{A}\|$. It thus follows that there is some $x \in \Sigma^*$ such that $x^{-1}\|\mathcal{A}\|$ has two different coordinates with respect to $(x_1^{-1}\|\mathcal{A}\|, \ldots, x_m^{-1}\|\mathcal{A}\|)$: a contradiction. □

Theorem 7. *Let $\mathcal{A}$ be a weighted automaton over a field. If $\mathcal{A}$ has a bideterministic equivalent, then the Cardon-Crochemore algorithm applied to $\mathcal{A}$ outputs a bideterministic automaton.*

Proof. Let $\mathcal{A}$ admit a bideterministic equivalent $\mathcal{B}$, and assume that it is trim. Then $\mathcal{B}$ is minimal by Corollary 3, so Lemma 6 implies that the algorithm applied to $\mathcal{A}$ yields a deterministic automaton $\mathcal{D}$. If $\mathcal{D}$ was not bideterministic, then there would be $u, v \in \Sigma^*$ such that $u^{-1}\|\mathcal{D}\|$ is not a scalar multiple of $v^{-1}\|\mathcal{D}\|$ and $\mathrm{supp}(u^{-1}\|\mathcal{D}\|) \cap \mathrm{supp}(v^{-1}\|\mathcal{D}\|) \neq \emptyset$. On the other hand, bideterminism of $\mathcal{B}$ implies[4] $\mathrm{supp}(u^{-1}\|\mathcal{B}\|) \cap \mathrm{supp}(v^{-1}\|\mathcal{B}\|) = \emptyset$ when $u^{-1}\|\mathcal{B}\|$ is not a scalar multiple of $v^{-1}\|\mathcal{B}\|$. This contradicts the assumption that $\|\mathcal{B}\| = \|\mathcal{D}\| = \|\mathcal{A}\|$. □

Corollary 8. *Bideterminisability of weighted automata over effective fields is decidable in polynomial time.*

5.2 Tropical Semirings

We now establish decidability of the bideterminisability problem for weighted automata over the tropical (min-plus) semirings $\mathbb{N}_{\min} = (\mathbb{N} \cup \{\infty\}, \min, +, \infty, 0)$, $\mathbb{Z}_{\min} = (\mathbb{Z} \cup \{\infty\}, \min, +, \infty, 0)$, and $\mathbb{Q}_{\min} = (\mathbb{Q} \cup \{\infty\}, \min, +, \infty, 0)$.

Theorem 9. *Bideterminisability of weighted automata over the semirings $\mathbb{N}_{\min}$, $\mathbb{Z}_{\min}$, and $\mathbb{Q}_{\min}$ is decidable.*

Proof. By positivity of tropical semirings, the minimal deterministic finite automaton $\mathcal{B}$ for $\mathrm{supp}(\|\mathcal{A}\|)$ is bideterministic whenever a tropical automaton $\mathcal{A}$ is bideterminisable. Given $\mathcal{A}$, we may thus remove the weights and minimise the automaton to get $\mathcal{B}$. If $\mathcal{B}$ is not bideterministic, $\mathcal{A}$ is not bideterminisable. If $\mathcal{B}$ is empty, $\mathcal{A}$ is bideterminisable. If $\mathcal{B}$ is bideterministic and nonempty, $\mathcal{A}$ is bideterminisable if and only if it is equivalent to some $\mathcal{B}'$ obtained from $\mathcal{B}$ by assigning weights to its transitions, its initial state, and its terminal state.

We show that existence of such $\mathcal{B}'$ is decidable given $\mathcal{A}$ and $\mathcal{B}$. Denote the unknown weights by $x_1, \ldots, x_N$, and let $\mathbf{x} = (x_1, \ldots, x_N)$. Here, x_1 corresponds to the unknown initial weight, $x_2, \ldots, x_{N-1}$ to the unknown transition weights, and x_N to the unknown terminal weight. Moreover, for each $w \in \mathrm{supp}(\|\mathcal{A}\|)$, let $\Psi(w) = (1, \eta_2, \ldots, \eta_{N-1}, 1)$, where η_i denotes, for $i = 2, \ldots, N-1$, the number of times the unique successful run of $\mathcal{B}$ upon w goes through the transition corresponding to the unknown weight x_i.

In order for $\mathcal{B}'$ to exist, the unknown weights have to satisfy the equations $\Psi(w) \cdot \mathbf{x}^T = (\|\mathcal{A}\|, w)$ for all $w \in \mathrm{supp}(\|\mathcal{A}\|)$. If this system has a solution, then its solution set coincides with the one of a *finite* system of equations

$$\Psi(w_i) \cdot \mathbf{x}^T = (\|\mathcal{A}\|, w_i) \qquad \text{for } i = 1, \ldots, M, \tag{5}$$

where $w_1, \ldots, w_M \in \mathrm{supp}(\|\mathcal{A}\|)$ are such that $(\Psi(w_1), \ldots, \Psi(w_M))$ is a basis of the vector space over $\mathbb{Q}$ generated by $\Psi(w)$ for $w \in \mathrm{supp}(\|\mathcal{A}\|)$. This basis can be effectively obtained, e.g., from the representation of $\{\Psi(w) \mid w \in \mathrm{supp}(\|\mathcal{A}\|)\}$ as a semilinear set. Hence, $w_1, \ldots, w_M$ can be found as well.

[4] This is a slight extension of a well-known property of bideterministic automata without weights – see, e.g., L. Polák [28, Section 5].

We may thus solve the system (5) over $\mathbb{N}$, $\mathbb{Z}$, or $\mathbb{Q}$ depending on the semiring considered. While Gaussian elimination is sufficient to solve the system over $\mathbb{Q}$, the solution over $\mathbb{Z}$ and $\mathbb{N}$ requires more sophisticated methods, namely an algorithm for solving systems of linear Diophantine equations in the former case [31], and integer linear programming in the latter case [31].

If there is no solution, $\mathcal{A}$ is not bideterminisable. Otherwise, any solution $\mathbf{x}$ gives us a bideterministic tropical automaton $\mathcal{B}_\mathbf{x}$ obtained from $\mathcal{B}$ by assigning the weights according to $\mathbf{x}$. By what has been said, either all such automata $\mathcal{B}_\mathbf{x}$ are equivalent to $\mathcal{A}$, or none of them is. Equivalence of a deterministic tropical automaton with a nondeterministic one is decidable [2], so we may take any of the automata $\mathcal{B}_\mathbf{x}$ and decide whether $\|\mathcal{B}_\mathbf{x}\| = \|\mathcal{A}\|$. If so, we may set $\mathcal{B}' = \mathcal{B}_\mathbf{x}$ and $\mathcal{A}$ is bideterminisable. Otherwise, $\mathcal{A}$ is not bideterminisable. □

Note that the decision algorithm described makes use of deciding equivalence of a nondeterministic tropical automaton with a deterministic one, which is **PSPACE**-complete [2]. Nevertheless, we leave the complexity of the bideterminisability problem open.

Finally, let us note that it can be shown that the decidability result just established *does not* generalise to all effective positive semirings.

References

1. Allauzen, C., Mohri, M.: Efficient algorithms for testing the twins property. Journal of Automata, Languages and Combinatorics **8**(2), 117–144 (2003)
2. Almagor, S., Boker, U., Kupferman, O.: What's decidable about weighted automata? Information and Computation **282** (2022), article 104651
3. Angluin, D.: Inference of reversible languages. Journal of the ACM **29**(3), 741–765 (1982)
4. Béal, M.P., Lombardy, S., Sakarovitch, J.: On the equivalence of $\mathbb{Z}$-automata. In: Automata, Languages and Programming, ICALP 2005. pp. 397–409. Springer (2005)
5. Béal, M.P., Lombardy, S., Sakarovitch, J.: Conjugacy and equivalence of weighted automata and functional transducers. In: Computer Science – Theory and Applications, CSR 2006. pp. 58–69. Springer (2006)
6. Bell, J., Smertnig, D.: Noncommutative rational Pólya series. Selecta Mathematica **27**(3), article 34 (2021)
7. Berstel, J., Reutenauer, C.: Noncommutative Rational Series with Applications. Cambridge University Press (2011)
8. Cardon, A., Crochemore, M.: Détermination de la représentation standard d'une série reconnaissable. Informatique Théorique et Applications **14**(4), 371–379 (1980)
9. Ćirić, M., Droste, M., Ignjatović, J., Vogler, H.: Determinization of weighted finite automata over strong bimonoids. Information Sciences **180**, 3497–3520 (2010)
10. Droste, M., Kuich, W., Vogler, H. (eds.): Handbook of Weighted Automata. Springer (2009)
11. Droste, M., Kuske, D.: Weighted automata. In: Pin, J.É. (ed.) Handbook of Automata Theory, Vol. 1, chap. 4, pp. 113–150. European Mathematical Society (2021)
12. Eilenberg, S.: Automata, Languages, and Machines, Vol. A. Academic Press (1974)

13. Golan, J.S.: Semirings and their Applications. Kluwer Academic Publishers (1999)
14. Gruber, H.: Digraph complexity measures and applications in formal language theory. Discrete Mathematics & Theoretical Computer Science **14**(2), 189–204 (2012)
15. Hebisch, U., Weinert, H.J.: Semirings. World Scientific (1998)
16. Janin, D.: Free inverse monoids up to rewriting. Tech. rep., LaBRI – Laboratoire Bordelais de Recherche en Informatique (2015), available at `https://hal.archives-ouvertes.fr/hal-01182934`
17. Kirsten, D.: An algebraic characterization of semirings for which the support of every recognizable series is recognizable. Theoretical Computer Science **534**, 45–52 (2014)
18. Kirsten, D., Lombardy, S.: Deciding unambiguity and sequentiality of polynomially ambiguous min-plus automata. In: Symposium on Theoretical Aspects of Computer Science, STACS 2009. pp. 589–600 (2009)
19. Kirsten, D., Mäurer, I.: On the determinization of weighted automata. Journal of Automata, Languages and Combinatorics **10**(2–3), 287–312 (2005)
20. Kostolányi, P.: Determinisability of unary weighted automata over the rational numbers. Theoretical Computer Science **898**, 110–131 (2022)
21. Lombardy, S., Sakarovitch, J.: Sequential? Theoretical Computer Science **356**, 224–244 (2006)
22. McNaughton, R.: The loop complexity of pure-group events. Information and Control **11**(1–2), 167–176 (1967)
23. McNaughton, R.: The loop complexity of regular events. Information Sciences **1**(3), 305–328 (1969)
24. Mohri, M.: Finite-state transducers in language and speech processing. Computational Linguistics **23**(2), 269–311 (1997)
25. Mohri, M.: Weighted automata algorithms. In: Droste, M., Kuich, W., Vogler, H. (eds.) Handbook of Weighted Automata, chap. 6, pp. 213–256. Springer (2009)
26. Myers, R.S.R., Milius, S., Urbat, H.: Nondeterministic syntactic complexity. In: Foundations of Software Science and Computation Structures, FOSSACS 2021. pp. 448–468 (2021)
27. Pin, J.É.: On reversible automata. In: Latin American Symposium on Theoretical Informatics, LATIN 1992. pp. 401–416 (1992)
28. Polák, L.: Minimalizations of NFA using the universal automaton. International Journal of Foundations of Computer Science **16**(5), 999–1010 (2005)
29. Sakarovitch, J.: Elements of Automata Theory. Cambridge University Press (2009)
30. Sakarovitch, J.: Rational and recognisable power series. In: Droste, M., Kuich, W., Vogler, H. (eds.) Handbook of Weighted Automata, chap. 4, pp. 105–174. Springer (2009)
31. Schrijver, A.: Theory of Linear and Integer Programming. John Wiley & Sons (1986)
32. Schützenberger, M.P.: On the definition of a family of automata. Information and Control **4**(2–3), 245–270 (1961)
33. Shankar, P., Dasgupta, A., Deshmukh, K., Rajan, B.S.: On viewing block codes as finite automata. Theoretical Computer Science **290**(3), 1775–1797 (2003)
34. Stephen, J.B.: Presentations of inverse monoids. Journal of Pure and Applied Algebra **63**(1), 81–112 (1990)
35. Tamm, H., Ukkonen, E.: Bideterministic automata and minimal representations of regular languages. In: Implementation and Application of Automata, CIAA 2003. pp. 61–71 (2003)
36. Tamm, H., Ukkonen, E.: Bideterministic automata and minimal representations of regular languages. Theoretical Computer Science **328**(1–2), 135–149 (2004)