Bideterministic Weighted Automata[*]

Peter Kostolányi[1]

Department of Computer Science, Comenius University in Bratislava,
Mlynská dolina, 842 48 Bratislava, Slovakia

Abstract

A finite automaton is called bideterministic if it is both deterministic and codeterministic – that is, if it is deterministic and 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 over positive semirings are proved to be minimal. On the contrary, it is observed that this property does not hold over commutative rings in general: non-minimal trim bideterministic weighted automata do exist over all semirings that are not zero-divisor free, and over many such semirings, these automata might not even admit equivalents that are both minimal and bideterministic. The problem of determining whether a given rational series is realised by a bideterministic automaton is shown to be decidable over fields and over tropical semirings. An example of a positive semiring over which this problem becomes undecidable is given as well.

Keywords: Weighted automaton, Bideterminism, Minimal automaton, Integral domain, Positive semiring, Decidability

1. Introduction

Unlike the classical nondeterministic finite automata without weights, weighted finite automata might not always be determinisable. Nevertheless, partly due to their relevance for applications such as natural language and speech processing [31] and partly due to their purely theoretical importance, deterministic weighed automata and the questions related to them – such as decidability of determinisability, existence of efficient determinisation algorithms, or characterisations of rational series realised by deterministic weighted automata – have received significant attention. Deterministic weighted automata were studied over specific classes of semirings, such as tropical semirings or fields [1, 8, 9, 23, 24, 26, 28, 31, 32], as well as over strong bimonoids [12], often under certain additional restrictions.

Determinism in weighted automata has been a rich source of intriguing questions, many of which remain unsettled. For instance, the decidability status of the determinisability problem for weighted automata is, despite some partial results [23, 24, 28], still open over tropical semirings [27, 28]. Decidability of the same problem for automata over fields – and in particular, over the rational numbers – remained unknown for a long time as well [28, 26], and the problem was only recently proved decidable by J. P. Bell and D. Smertnig [9]. Complexity of this problem stays open [9]. 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.

Within these lines, deterministic weighted automata with certain additional requirements on their weights have been studied. This includes, for instance, the research on crisp-deterministic weighted automata by M. Ćirić et al. [12]. Another less explored 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 undertakes the study of *bideterministic* finite automata in the weighted setting. A finite automaton is bideterministic if it is both deterministic and codeterministic – the latter meaning that the transpose of the automaton, obtained by reversing all transitions and exchanging the roles of initial and terminal states, is deterministic as well. This in particular 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 [34], as a particular case of reversible finite automata. The fundamental properties of bideterministic finite automata have later been explored mostly by H. Tamm and E. Ukkonen [44, 45], who have shown that a trim bideterministic automaton is always a minimal nondeterministic automaton for the language it recognises – in fact, it is the *only* minimal nondeterministic finite automaton recognising its language. Minimality is understood here in the strong sense, *i.e.*, with respect to the number of states. An alternative proof of this minimality property of trim bideterministic automata was recently presented by R. S. R. Myers, S. Milius, and H. Urbat [33]. Transition minimality of bideterministic finite automata has also been established [43].

In addition to the above-mentioned studies, bideterministic automata have been – explicitly or implicitly – considered in connection to the star height problem [29, 30, 19], from the perspective of language inference [5], in the theory of block codes [41], and within the study of presentations of inverse monoids [42, 21].

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 [44, 45] 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 to the former question is negative whenever S is not zero-divisor free – and thus also whenever it is a nontrivial commutative ring other than an integral domain. Moreover, we show that the answer to the latter question is negative for a large class of commutative semirings including a multitude of *finite* commutative *rings*, and fully understand the answers to this question over the rings $\mathbb{Z}/m\mathbb{Z}$ with $m \in \mathbb{N}$.

We also consider the *bideterminisability problem*, in which the task is to decide whether a given weighted automaton over a semiring S admits a bideterministic equivalent over S. We show that the problem is decidable in polynomial time when S is a field, as it is sufficient to simply minimise the input automaton and check whether the resulting automaton is bideterministic; on the other hand, we give examples showing that this simple procedure might not work over integral domains in general. Next, we establish decidability of the bideterminisability problem over tropical semirings (of nonnegative integers, integers, and rationals). These results suggest that the bideterminisability problem for weighted automata might be somewhat easier than the determinisability problem, whose decidability status over tropical semirings remains open [27, 28], and for which only algorithms lacking efficiency [9] or generality [26] are known in the case of fields. Finally, we give an example of a positive semiring over which the bideterminisability problem is undecidable.

2. Preliminaries

We denote by $\mathbb{N}$, $\mathbb{Z}$, and $\mathbb{Q}$, respectively, the sets of all *nonnegative* integers, integers, and rational numbers.

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* [18, 20] if $a + b = 0$ for some $a, b \in S$ implies $a = b = 0$ and *zero-divisor free* [20], or *entire* [18], if $a \cdot b = 0$ for some $a, b \in S$ implies that $a = 0$ or $b = 0$. A semiring is *positive* [17, 22] 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.

The *ring of all polynomials* over a commutative ring R in indeterminates $x_1, \ldots, x_n$ for $n \in \mathbb{N} \setminus \{0\}$ is denoted by $R[x_1, \ldots, x_n]$. For $\mathbb{F}$ a field, $\mathbb{F}(x_1, \ldots, x_n)$ denotes the field of all *rational fractions* over $\mathbb{F}$ in $x_1, \ldots, x_n$, *i.e.*, the fraction field of $\mathbb{F}[x_1, \ldots, x_n]$. The subring of a ring R generated by a set $X \subseteq R$ is the smallest subring of R containing X. Given commutative rings R, R' such that $R \subseteq R'$ and $X \subseteq R'$, we denote by $R[X]$ the *subring of R' generated by X over R*, *i.e.*, the subring of R' generated by $R \cup X$.

An *ideal* of a semiring S is a nonempty set $I \subseteq S$ such that $a + b$, as, and sa are in I for all $a, b \in I$ and $s \in S$. Given $X \subseteq S$, we denote by (X) the *ideal of S generated by X*, *i.e.*, the smallest ideal of S containing X. If $X = \{x_1, \ldots, x_n\}$ is a finite set, we write $(x_1, \ldots, x_n)$ instead of $(\{x_1, \ldots, x_n\})$. The *quotient ring* of a *ring* R by an ideal I of R is denoted by R/I.

Alphabets are assumed to be finite and nonempty. Deterministic finite automata (without weights) are understood to have partial transition functions in this article, which is a natural assumption in the context of bideterministic automata. More precisely, we identify deterministic finite automata with nondeterministic finite automata having at most one initial state such that there is at most one transition upon each letter leading from each state. In particular, a deterministic finite automaton defined like this can be empty. Minimisation of such automata works essentially in the same way as for automata with complete transition functions – the only difference is that there is no dead state in the minimised automaton.

We now briefly recall some basic facts about noncommutative formal power series and weighted automata. More information can be found in [10, 15, 16, 36].

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

$$\mathrm{supp}(r) = \{w \in \Sigma^* \mid (r, w) \neq 0\}.$$

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 $n \in \mathbb{N}$, 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 and that γ *passes through* the states $q_0, \ldots, q_n$. 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; we also say that γ is *a run upon* $\lambda(\gamma)$. 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 [36].

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 $\mathcal{A}_q = (Q, \sigma, \iota_q, \tau)$ with $\iota_q(q) = 1$ and $\iota_q(p) = 0$ for all $p \in Q \setminus \{q\}$. Similarly, we denote by ${}_q\|\mathcal{A}\|$ the *past* of q, *i.e.*, the series realised by an automaton ${}_q\mathcal{A} = (Q, \sigma, \iota, \tau_q)$ with $\tau_q(q) = 1$ and $\tau_q(p) = 0$ 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 [40] and which has been explicitly described by A. Cardon and M. Crochemore [11]. The reader may consult [10, 36, 37] for a detailed exposition.

We only note here 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 such that L 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 $\mathbf{i}\mu(x)$ for $x \in \Sigma^*$. 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 $\mu(y)\mathbf{f}$ for $y \in \Sigma^*$.

The actual minimisation algorithm then consists of two reduction steps. The input 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 $x_1 = \varepsilon$,

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

---

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

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 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 resulting automaton $\mathcal{C}$ is *conjugate* [6, 7] 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 the space 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 representation $\mathcal{P}_{\mathcal{A}} = (n, \mathbf{i}, \mu, \mathbf{f})$ gives rise to a linear mapping $\Lambda[\mathcal{A}] \colon \text{Left}(\mathcal{A}) \to \mathbb{F}\langle\!\langle \Sigma^* \rangle\!\rangle$, 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 [37].

3. Bideterministic Weighted Automata over a Semiring

In the same way as for finite automata without weights [44, 45], 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 3.1. 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 remaining two conditions assure that $\mathcal{A}$ is *codeterministic*.

It has been shown by H. Tamm and E. Ukkonen [44, 45] that a trim bideterministic automaton without weights is always a minimal nondeterministic automaton for the language it recognises. Every language recognised by some bideterministic automaton thus also admits a minimal automaton that is bideterministic. Moreover, by uniqueness of minimal deterministic finite automata and existence of efficient minimisation algorithms, it is decidable whether a rational 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.

---

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

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 observe that the first question has a negative answer over every semiring that is not zero-divisor free – and thus, in particular, also over every nontrivial *commutative ring* other than an integral domain. For the second question, we obtain a negative answer for a large class of commutative semirings, which also includes numerous finite commutative rings. Moreover, we completely characterise the rings $\mathbb{Z}/m\mathbb{Z}$ with $m \in \mathbb{N}$, over which this question admits an affirmative answer.

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 4.1. *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 4.2. *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 weighted automata over integral domains *cannot* be generalised to automata over commutative rings. In fact, we observe that non-minimal trim bideterministic weighted automata do exist over every *semiring* that is not zero-divisor free – and thus also over every nontrivial commutative ring that is not an integral domain. Moreover, we exhibit a class of commutative semirings S such that bideterministic weighted automata over S do not even always admit a minimal bideterministic equivalent – although this class includes many finite commutative rings as well, it does not include neither all nontrivial commutative rings other than integral domains, nor all such finite rings.

Let us start by a simple observation showing that the answer to Question 1 of Section 3 is negative whenever the semiring in consideration is not zero-divisor free.

Theorem 4.3. *Let S be a semiring that is not zero-divisor free. Then there exists a trim bideterministic weighted automaton $\mathcal{A}$ over S that is not minimal.*

Proof. As S is not zero-divisor free, one can find $s, t \in S \setminus \{0\}$ such that $st = 0$. Given such elements, consider the automaton $\mathcal{A}$ in Fig. 1.
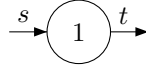
Figure 1: A trim bideterministic weighted automaton $\mathcal{A}$ over S that is not minimal.

The automaton $\mathcal{A}$ clearly is both trim and bideterministic. However, it is not minimal: $\|\mathcal{A}\| = 0$, and the minimal automaton for this series is empty. $\qquad\square$

Corollary 4.4. *Let R be a commutative ring. Then the following are equivalent:*

(i) *Every trim bideterministic weighted automaton $\mathcal{A}$ over R is minimal.*

(ii) *R is either trivial, or an integral domain.*

Classification of commutative rings with affirmative answer to Question 2 of Section 3 seems to be less straightforward. We first show that the answer to this question is negative over a large class of commutative semirings including also many finite commutative rings.

Theorem 4.5. *Let S be a commutative semiring containing elements $s, t, s', t' \in S$ such that $st \neq 0 \neq s't'$ and $st' = 0 = s't$. 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. 2. Clearly,

$$\|\mathcal{A}\| = st \cdot aba + s't' \cdot bb.$$

The automaton $\mathcal{A}$ is not minimal, as the same series is realised by a smaller automaton $\mathcal{B}$ in Fig. 3:

$$\|\mathcal{B}\| = st \cdot aba + s't' \cdot bb = \|\mathcal{A}\|.$$

This gives an alternative way to obtain the negative answer to Question 1 of Section 3 over S.

We show that $\|\mathcal{A}\|$ actually is 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 $(\|\mathcal{A}\|, aba) = st \neq 0$, 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.
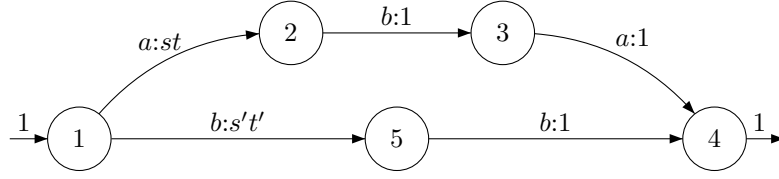
Figure 2: The trim bideterministic weighted automaton $\mathcal{A}$ over S.
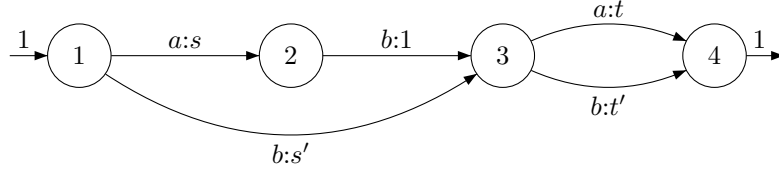
Figure 3: The four-state weighted automaton $\mathcal{B}$ over S equivalent to $\mathcal{A}$.

Similar reasoning as above 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 $(\|\mathcal{A}\|, bb) = s't' \neq 0$, 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. □

Several natural classes of semirings satisfying the assumptions of Theorem 4.5 can be readily identified.

Corollary 4.6. *Let S be a commutative semiring with elements $u, v \in S$ such that $uv = 0$ and $u^2 \neq 0 \neq v^2$. 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. Take $s = t = u$ and $s' = t' = v$ in Theorem 4.5. □

Note that the class of commutative semirings from Corollary 4.6 also includes many *finite* commutative *rings*. In particular, the ring $\mathbb{Z}/m\mathbb{Z}$ of integers modulo a positive integer m falls into this class whenever m has at least two distinct prime factors.

Corollary 4.7. *Let m be a positive integer with at least two distinct prime factors. Then there is a trim bideterministic weighted automaton $\mathcal{A}$ over $\mathbb{Z}/m\mathbb{Z}$, the finite commutative ring of integers modulo m, such that none of the minimal automata for $\|\mathcal{A}\|$ is bideterministic.*

Proof. For p a prime factor of m and $k \in \mathbb{N} \setminus \{0\}$ the highest exponent such that p^k divides m, let $q = m/p^k$. Then the ring $\mathbb{Z}/m\mathbb{Z}$ satisfies the assumptions of Corollary 4.6 with $u = p^k$ and $v = q$. □

Let us mention one more particular class of semirings satisfying the assumptions of Theorem 4.5. This class includes, e.g., the commutative ring $\mathbb{Q}[x, y]/(x^2, y^2)$.

Corollary 4.8. *Let S be a commutative semiring with elements $u, v \in S$ such that $uv \neq 0$ and $u^2 = 0 = v^2$. 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. Take $s = t' = u$ and $t = s' = v$ in Theorem 4.5. □

Remark 4.9. The class of commutative semirings satisfying Theorem 4.5 can be conveniently described in terms of zero-divisor graphs [3, 4, 13, 14] – more precisely, in terms of their looped version appearing, e.g., in [3]. Let S be a commutative semiring and $Z(S)^*$ its set of nontrivial divisors of zero, *i.e.*,

$$Z(S)^* = Z(S) \setminus \{0\}$$

for

$$Z(S) = \{a \in S \mid \exists b \in S \setminus \{0\} : ab = 0\}.$$

The *zero-divisor graph* of S then is a possibly infinite undirected graph with $Z(S)^*$ as its set of vertices such that vertices $a, b \in Z(S)^*$ are connected by an edge (a loop in case $a = b$) if and only if $ab = 0$.

A commutative semiring S satisfies the assumptions of Theorem 4.5 if and only if its zero-divisor graph contains not necessarily distinct vertices s, t, s', t' such that there are edges connecting s with t' and s' with t, while there are no edges connecting s with t and s' with t'. This "configuration" is illustrated in Fig. 4.
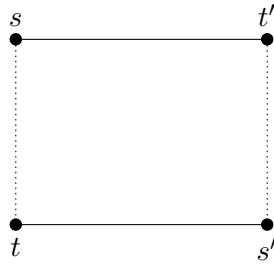
Figure 4: A "configuration" that has to be contained in the zero-divisor graph of every commutative semiring satisfying the assumptions of Theorem 4.5. Full lines in the diagram represent edges, while dotted lines should be interpreted as nonexistent edges. The vertices s, t, s', t' might not be distinct, so a line between two vertices may also indicate (non)existence of a loop.

Remark 4.10. Theorem 4.5 significantly narrows the class of commutative semirings, for which Question 2 of Section 3 may admit an affirmative answer. Although we show below that this actually also happens for some finite commutative rings other than integral domains, Theorem 4.5 implies that some special properties are needed for this to be the case. More precisely, the set of zero divisors $Z(S)$ in a commutative semiring S is obviously always closed under multiplication by an arbitrary element of S, but it may have little additive structure in general – see, e.g., [3] for the case of rings. It follows by Theorem 4.5 that $Z(S)$ is an ideal of S whenever S is a commutative semiring such that all bideterministic weighted automata over S admit an equivalent minimal automaton that is bideterministic as well. Indeed, if $a, b \in Z(S)$ are elements of S such that $a + b \notin Z(S)$ and $c, d \in S \setminus \{0\}$ satisfy $ac = bd = 0$, then

$$0 \neq (a + b)c = ac + bc = bc$$

and

$$0 \neq (a + b)d = ad + bd = ad.$$

Thus, by applying Theorem 4.5 with $s = a$, $t = d$, $s' = b$, and $t' = c$, we find out that there is a bideterministic weighted automaton over S such that none of the equivalent minimal automata is bideterministic.

The condition of $Z(S)$ forming an ideal is thus necessary in order for Question 2 of Section 3 to have an affirmative answer over a commutative semiring S. It is not a sufficient condition, as can be exemplified by the commutative ring $R = \mathbb{Q}[x, y]/(x^2, y^2)$, which satisfies the conditions of Corollary 4.8 although $Z(R)$ clearly is an ideal of R.

We leave open the full characterisation of commutative rings, over which bideterministic weighted automata always admit minimal bideterministic equivalents. However, in what follows we at least characterise the rings $\mathbb{Z}/m\mathbb{Z}$ with this property. Recall that the answer to Question 2 of Section 3 is affirmative over fields and integral domains; this includes all the fields $\mathbb{Z}/m\mathbb{Z}$ with m prime and the integral domain $\mathbb{Z} \cong \mathbb{Z}/0\mathbb{Z}$. The answer is also clearly affirmative over the trivial ring $\mathbb{Z}/1\mathbb{Z}$. On the other hand, the answer to the said question is negative by Corollary 4.7 when m contains at least two distinct prime factors. The only case left unexplored thus is the one with $m = p^k$ for p prime and $k \geq 2$ an integer. We now show that the answer to Question 2 of Section 3 is affirmative for all rings $\mathbb{Z}/m\mathbb{Z}$ with m taking this form.

In order to obtain this result, we need the following lemma about linear combinations in the module $(\mathbb{Z}/p^k\mathbb{Z})^n$ over the ring $\mathbb{Z}/p^k\mathbb{Z}$.

Lemma 4.11. *Let p be a prime, $k, n \in \mathbb{N} \setminus \{0\}$, and $\mathbf{v}_1, \ldots, \mathbf{v}_{n-1}$ vectors in $(\mathbb{Z}/p^k\mathbb{Z})^n$. Then there exists at least one index $j \in [n]$, for which there are no $a_1, \ldots, a_{n-1} \in \mathbb{Z}/p^k\mathbb{Z}$ and $b \in (\mathbb{Z}/p^k\mathbb{Z}) \setminus \{0\}$ such that*

$$a_1\mathbf{v}_1 + \ldots + a_{n-1}\mathbf{v}_{n-1} = b\mathbf{e}_j,$$

where

$$\mathbf{e}_j = (\underbrace{0, \ldots, 0}_{j-1}, 1, \underbrace{0, \ldots, 0}_{n-j}).$$

Proof. Suppose for contradiction that there is a prime number p and $k, n \in \mathbb{N} \setminus \{0\}$, for which the statement of the lemma does not hold, *i.e.*, one can find vectors $\mathbf{v}_1, \ldots, \mathbf{v}_{n-1} \in (\mathbb{Z}/p^k\mathbb{Z})^n$ such that a nonzero multiple of $\mathbf{e}_j$ belongs to

$$\langle \mathbf{v}_1, \ldots, \mathbf{v}_{n-1} \rangle = \{a_1\mathbf{v}_1 + \ldots + a_{n-1}\mathbf{v}_{n-1} \mid a_1, \ldots, a_{n-1} \in \mathbb{Z}/p^k\mathbb{Z}\}$$

for $j = 1, \ldots, n$. Note that since p^{k-1} is a multiple of every nonzero element of $\mathbb{Z}/p^k\mathbb{Z}$, we actually get

$$p^{k-1}\mathbf{e}_1, \ldots, p^{k-1}\mathbf{e}_n \in \langle \mathbf{v}_1, \ldots, \mathbf{v}_{n-1} \rangle. \tag{5}$$

This is clearly impossible for $n = 1$, so we may assume for the rest of the proof that $n \geq 2$.

We now observe that with some care, a variant of Gaussian elimination can be applied to the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_{n-1}$. There clearly are $i \in [n-1]$ and $j \in [n]$ such that all entries of the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_{n-1}$ are multiples of the j-th entry of $\mathbf{v}_i$ over $\mathbb{Z}/p^k\mathbb{Z}$ – one can take any i and j for which the highest power of p dividing the j-th entry of $\mathbf{v}_i$ is minimised. This j-th entry of $\mathbf{v}_i$ has to be nonzero, as otherwise all of the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_{n-1}$ would be zero, obviously contradicting (5). Moreover, we may assume without loss of generality that actually $j = 1$, as we can relabel the indices when this is not the case. Then, by subtracting suitable multiples of $\mathbf{v}_i$ from the remaining vectors and switching the first and the i-th vector, we obtain vectors $\mathbf{v}_1^{(1)}, \ldots, \mathbf{v}_{n-1}^{(1)}$ such that

$$p^{k-1}\mathbf{e}_1, \ldots, p^{k-1}\mathbf{e}_n \in \left\langle \mathbf{v}_1^{(1)}, \ldots, \mathbf{v}_{n-1}^{(1)} \right\rangle \tag{6}$$

and $\mathbf{v}_1^{(1)}$ is the only vector with nonzero entry in the first column. Moreover, as all entries of these vectors are multiples of the first entry of $\mathbf{v}_1^{(1)}$ over $\mathbb{Z}/p^k\mathbb{Z}$, it is clear that $p^{k-1}\mathbf{e}_1$ is the only among the vectors $p^{k-1}\mathbf{e}_1, \ldots, p^{k-1}\mathbf{e}_n$ that can be expressed by a linear combination

$$a_1\mathbf{v}_1^{(1)} + \ldots + a_{n-1}\mathbf{v}_{n-1}^{(1)}$$

of the vectors $\mathbf{v}_1^{(1)}, \ldots, \mathbf{v}_{n-1}^{(1)}$ with $a_1, \ldots, a_{n-1} \in \mathbb{Z}/p^k\mathbb{Z}$ such that $a_1\mathbf{v}_1^{(1)}$ is *not* equal to the zero vector. We may thus assume that actually $\mathbf{v}_1^{(1)} = p^{k-1}\mathbf{e}_1$, while (6) remains valid.

The remaining vectors can be dealt with similarly. For $\ell \in [n-2]$, assume that $\mathbf{v}_1^{(\ell)}, \ldots, \mathbf{v}_{n-1}^{(\ell)}$ are vectors such that

$$p^{k-1}\mathbf{e}_1, \ldots, p^{k-1}\mathbf{e}_n \in \left\langle \mathbf{v}_1^{(\ell)}, \ldots, \mathbf{v}_{n-1}^{(\ell)} \right\rangle,$$

the entries of $\mathbf{v}_{\ell+1}^{(\ell)}, \ldots, \mathbf{v}_{n-1}^{(\ell)}$ in the first ℓ columns are all zero, and

$$\mathbf{v}_1^{(\ell)} = p^{k-1}\mathbf{e}_1, \qquad \ldots, \qquad \mathbf{v}_\ell^{(\ell)} = p^{k-1}\mathbf{e}_\ell.$$

10

One can then clearly find $i \in \{\ell+1, \ldots, n-1\}$ and $j \in \{\ell+1, \ldots, n\}$ such that all entries of the vectors $\mathbf{v}_{\ell+1}^{(\ell)}, \ldots, \mathbf{v}_{n-1}^{(\ell)}$ in columns $\ell+1, \ldots, n$ are multiples of the nonzero j-th entry of $\mathbf{v}_i^{(\ell)}$ over $\mathbb{Z}/p^k\mathbb{Z}$. Without loss of generality, we may assume that $j = \ell+1$. Then, after subtracting suitable multiples of $\mathbf{v}_i^{(\ell)}$ from the remaining among the vectors $\mathbf{v}_{\ell+1}^{(\ell)}, \ldots, \mathbf{v}_{n-1}^{(\ell)}$ and switching the $(\ell+1)$-th and the i-th vector, we obtain vectors $\mathbf{v}_1^{(\ell+1)}, \ldots, \mathbf{v}_{n-1}^{(\ell+1)}$ such that

$$p^{k-1}\mathbf{e}_1, \ldots, p^{k-1}\mathbf{e}_n \in \left\langle \mathbf{v}_1^{(\ell+1)}, \ldots, \mathbf{v}_{n-1}^{(\ell+1)} \right\rangle$$

and the entries of $\mathbf{v}_{\ell+2}^{(\ell+1)}, \ldots, \mathbf{v}_{n-1}^{(\ell+1)}$ in the first $\ell+1$ columns are all zero. Moreover, similarly as above, we may assume $\mathbf{v}_{\ell+1}^{(\ell+1)} = p^{k-1}\mathbf{e}_{\ell+1}$, so that

$$\mathbf{v}_1^{(\ell+1)} = p^{k-1}\mathbf{e}_1, \qquad \ldots, \qquad \mathbf{v}_{\ell+1}^{(\ell+1)} = p^{k-1}\mathbf{e}_{\ell+1}.$$

Thus, in the end of this process, we are finally left with vectors

$$\mathbf{v}_1^{(n-1)} = p^{k-1}\mathbf{e}_1, \qquad \ldots, \qquad \mathbf{v}_{n-1}^{(n-1)} = p^{k-1}\mathbf{e}_{n-1},$$

while at the same time

$$p^{k-1}\mathbf{e}_n \in \left\langle \mathbf{v}_1^{(n-1)}, \ldots, \mathbf{v}_{n-1}^{(n-1)} \right\rangle.$$

This is a clear contradiction. □

We are now prepared to give a positive answer to Question 2 of Section 3 for automata over the rings $\mathbb{Z}/p^k\mathbb{Z}$, where p is a prime and $k \geq 1$ is an integer. This is based on an observation that although trim bideterministic automata over these rings might not be minimal when $k \geq 2$, the minimality property does hold for a class of bideterministic weighted automata that are trim in a slightly stronger sense. This is made precise by the following definition.

Definition 4.12. A weighted automaton $\mathcal{A} = (Q, \sigma, \iota, \tau)$ over a semiring S and alphabet Σ is *semantically trim* if for each state $q \in Q$, there exists a run γ_q of $\mathcal{A}$ passing through q such that $\|\gamma_q\| \neq 0$.

It is clear that every weighted automaton $\mathcal{A}$ can be transformed to an equivalent semantically trim automaton by simply removing the states q for which there is no run γ_q of $\mathcal{A}$ realising a nonzero monomial passing through q. Moreover, every semantically trim automaton is obviously trim, while the converse might not hold if the underlying semiring S is not zero-divisor free.

Theorem 4.13. *Let p be a prime, $k \in \mathbb{N}$, and $\mathcal{A}$ a semantically trim bideterministic weighted automaton over $\mathbb{Z}/p^k\mathbb{Z}$. Then $\mathcal{A}$ is minimal.*

Proof. The theorem is trivially true for $k = 0$; we may thus assume that $k \geq 1$. Let $\mathcal{A} = (n, \sigma, \iota, \tau)$ be a semantically trim bideterministic weighted automaton over $\mathbb{Z}/p^k\mathbb{Z}$ and alphabet Σ. The automaton is minimal if $n = 0$, so we may assume that $n \geq 1$. As $\mathcal{A}$ is semantically trim, there are words $u_1, \ldots, u_n, v_1, \ldots, v_n \in \Sigma^*$ such that

$$(_q\|\mathcal{A}\|, u_q) \neq 0, \qquad (\|\mathcal{A}\|_q, v_q) \neq 0, \qquad \text{and} \qquad (\|\mathcal{A}\|, u_q v_q) \neq 0 \tag{7}$$

for $q = 1, \ldots, n$. In particular, this implies that there is precisely one state $q_i \in [n]$ with $\iota(q_i) \neq 0$ and precisely one state $q_t \in [n]$ with $\tau(q_t) \neq 0$, and that, for $q = 1, \ldots, n$, there are runs upon u_q from q_i to q and upon v_q from q to q_t in $\mathcal{A}$. Moreover, it easily follows by bideterminism of $\mathcal{A}$ that there cannot be a run in $\mathcal{A}$ upon v_q leading from p to q_t if p and q are distinct states of $\mathcal{A}$. As a consequence,

$$(\|\mathcal{A}\|, u_p v_q) \neq 0$$

holds for $p, q \in [n]$ if and only if $p = q$.

Now, suppose for contradiction that $\mathcal{A}$ is not minimal. By possibly adding some useless states to a minimal equivalent automaton, we find out that there is a weighted automaton $\mathcal{B} = (n-1, \sigma', \iota', \tau')$ such that $\|\mathcal{B}\| = \|\mathcal{A}\|$. For $p = 1, \ldots, n-1$, let

$$\mathbf{v}_p = ((\|\mathcal{B}\|_p, v_1), \ldots, (\|\mathcal{B}\|_p, v_n)).$$

Then clearly, for $q = 1, \ldots, n$,

$$\sum_{p=1}^{n-1} ({}_p\|\mathcal{B}\|, u_q)\mathbf{v}_p = ((\|\mathcal{A}\|, u_q v_1), \ldots, (\|\mathcal{A}\|, u_q v_n)) = (\|\mathcal{A}\|, u_q v_q)\mathbf{e}_q,$$

where

$$\mathbf{e}_q = (\underbrace{0, \ldots, 0}_{q-1}, 1, \underbrace{0, \ldots, 0}_{n-q}).$$

As $(\|\mathcal{A}\|, u_q v_q) \in (\mathbb{Z}/p^k\mathbb{Z}) \setminus \{0\}$ by (7), this contradicts Lemma 4.11. $\qquad\square$

Corollary 4.14. *Let p be a prime and $k \in \mathbb{N}$. Then every bideterministic weighted automaton over $\mathbb{Z}/p^k\mathbb{Z}$ admits an equivalent minimal automaton that is bideterministic.*

Proof. Given a bideterministic weighted automaton $\mathcal{A}$ over $\mathbb{Z}/p^k\mathbb{Z}$, one can obtain a semantically trim weighted automaton $\mathcal{B}$ equivalent to $\mathcal{A}$ by possibly removing several states. This automaton $\mathcal{B}$ is clearly bideterministic as well – it is thus minimal by Theorem 4.13. $\qquad\square$

The result just established finishes the characterisation of rings $\mathbb{Z}/m\mathbb{Z}$, over which all bideterministic weighted automata admit an equivalent minimal automaton that is bideterministic as well. This is summarised by the following corollary.

Corollary 4.15. *Let $m \in \mathbb{N}$. Then the following are equivalent:*

(i) *For all bideterministic weighted automata $\mathcal{A}$ over $\mathbb{Z}/m\mathbb{Z}$, there exists an equivalent minimal automaton that is bideterministic.*

(ii) *Either $m = 0$, or $m = p^k$ for p a prime and $k \in \mathbb{N}$.*

Proof. If $m = 0$, then $\mathbb{Z}/m\mathbb{Z} \cong \mathbb{Z}$, and existence of minimal bideterministic equivalents to all bideterministic automata follows by Corollary 4.2. For $m = p^k$ with p prime and $k \in \mathbb{N}$, the same property follows by Corollary 4.14 (and for $k = 1$ also by Corollary 4.2). On the other hand, if $m \geq 2$ is not equal to p^k for p prime and $k \in \mathbb{N}$, then m necessarily has at least two different prime factors, and Corollary 4.7 implies existence of a bideterministic weighted automaton over $\mathbb{Z}/m\mathbb{Z}$ that admits no minimal bideterministic equivalent. $\qquad\square$

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 4.16. *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 the automaton $\mathcal{A}$ by "forgetting about its weights". This is a minimal nondeterministic automaton for $\mathrm{supp}(\|\mathcal{A}\|)$ by the minimality property of trim bideterministic automata without weights [44, 45].

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 $\mathcal{A}'$. $\qquad\square$

5. Decidability of Bideterminisability

Let us now consider the problem of deciding whether a given weighted automaton admits a bideterministic equivalent. We show that this *bideterminisability* problem is decidable both over effective fields and over tropical semirings (of nonnegative integers, integers, and rationals). Moreover, the decision procedure over fields is simply based on checking whether the output of the Cardon-Crochemore minimisation algorithm is bideterministic or not – it thus runs in polynomial time.

By contrast, the results known so far about the *determinisability* problem are weaker: its decidability status over tropical semirings is still open [27, 28], and its decidability over fields has only recently been established by J. P. Bell and D. Smertnig [9]. Moreover, there are no reasonable complexity bounds known for the general determinisability problem for weighted automata over fields, and a polynomial-time algorithm is known just for unary weighted automata over the rationals [26].

5.1. Fields and Integral Domains

We first prove decidability of the bideterminisability problem for automata over fields. To this end, we strengthen Theorem 4.1 by showing 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 5.1. *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 an automaton obtained as an output of the Cardon-Crochemore algorithm for input automaton $\mathcal{A}$, 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

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

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. $\square$

Theorem 5.2. *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 the automaton $\mathcal{B}$ is trim. Then $\mathcal{B}$ is minimal by Corollary 4.2, so Lemma 5.1 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 $\|\mathcal{B}\| = \|\mathcal{D}\| = \|\mathcal{A}\|$. $\square$

Corollary 5.3. *Bideterminisability of weighted automata over effective fields is decidable in polynomial time.*

We leave the decidability status of the bideterminisability problem open for weighted automata over integral domains. However, the following two examples indicate that the simple decision procedure for automata over fields is no longer sufficient when weights are taken from an integral domain.

In particular, Example 5.4 shows that a weighted automaton over an integral domain R can be bideterminisable over the fraction field of R even if it does not admit a bideterministic equivalent over R itself. The Cardon-Crochemore minimisation algorithm thus may produce a bideterministic automaton $\mathcal{B}$ over the field of fractions of R even if the input automaton $\mathcal{A}$ is not bideterminisable over R, so that simply checking whether $\mathcal{B}$ is bideterministic is insufficient to decide bideterminisability of $\mathcal{A}$.

On the other hand, in Example 5.5 we observe that although the Cardon-Crochemore algorithm applied to a bideterminisable automaton over an integral domain R always outputs a bideterministic automaton over the fraction field of R, it may not output a bideterministic automaton over R itself. This means that running the Cardon-Crochemore algorithm and checking whether the result is a bideterministic automaton *over R* is not appropriate either.

Example 5.4. Consider the subring of the integral domain $\mathbb{Q}[x, y]$ generated over $\mathbb{Q}$ by the set of all monomials of degree at least two, *i.e.*, the integral domain $\mathbb{Q}[X] \subseteq \mathbb{Q}[x, y]$ for $X = \{x^m y^n \mid m + n \geq 2\}$. As one can obtain both x and y as quotients of monomials from X – e.g., $x = x^3/x^2$ and $y = y^3/y^2$ – the fraction field of $\mathbb{Q}[X]$ equals $\mathbb{Q}(x, y)$.

Let $\mathcal{A}$ be a weighted automaton over the integral domain $\mathbb{Q}[X]$ and alphabet $\Sigma = \{a\}$ given in Fig. 5. Clearly,

$$\|\mathcal{A}\| = \sum_{t \in \mathbb{N}} (x + y)^{t+2} a^t.$$
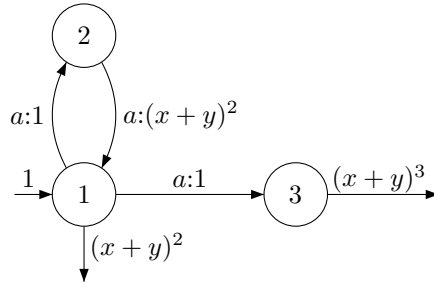
Figure 5: The weighted automaton $\mathcal{A}$ over $\mathbb{Q}[X]$ and $\Sigma = \{a\}$.

It is easy to see that $\mathcal{A}$ is equivalent to the bideterministic weighted automaton $\mathcal{B}$ over $\mathbb{Q}(x, y)$ in Fig. 6. Hence, $\mathcal{A}$ is bideterminisable over $\mathbb{Q}(x, y)$, the fraction field of $\mathbb{Q}[X]$.

In spite of this, we now show that the automaton $\mathcal{A}$ is not bideterminisable over $\mathbb{Q}[X]$. Assume for contradiction that $\|\mathcal{A}\| = \|\mathcal{C}\|$ for some bideterministic weighted automaton $\mathcal{C}$ over $\mathbb{Q}[X]$. As $\varepsilon \in \mathrm{supp}(\|\mathcal{A}\|)$,

---
[4]This is a straightforward extension of a well-known property of bideterministic finite automata without weights – see, e.g., L. Polák [35, Section 5].
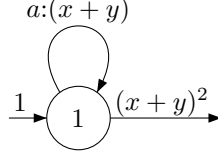
Figure 6: The bideterministic weighted automaton $\mathcal{B}$ over $\mathbb{Q}(x, y)$ equivalent to $\mathcal{A}$.

the automaton $\mathcal{C} = (Q, \sigma, \iota, \tau)$ contains a state $q \in Q$ such that both $\iota(q) \neq 0$ and $\tau(q) \neq 0$, while $\iota(p) = \tau(p) = 0$ for all $p \in Q \setminus \{q\}$. Moreover,

$$\iota(q)\tau(q) = (\|\mathcal{A}\|, \varepsilon) = (x + y)^2. \tag{8}$$

Next, as $a \in \mathrm{supp}(\|\mathcal{A}\|)$ as well, there has to be a transition upon a leading from q to a state with nonzero terminal weight, *i.e.*, an a-labelled self-loop at q. Moreover, by commutativity of $\mathbb{Q}[x, y]$,

$$\iota(q)\tau(q)\sigma(q, a, q) = \iota(q)\sigma(q, a, q)\tau(q) = (\|\mathcal{A}\|, a) = (x + y)^3,$$

which together with (8) implies

$$(x + y)^2 \sigma(q, a, q) = (x + y)^3. \tag{9}$$

However, the only possible $\sigma(q, a, q) \in \mathbb{Q}[x, y]$ satisfying (9) is $\sigma(q, a, q) = x + y$, and this is not an element of $\mathbb{Q}[X]$. This contradicts our assumption that $\mathcal{C}$ is a bideterministic weighted automaton over $\mathbb{Q}[X]$.
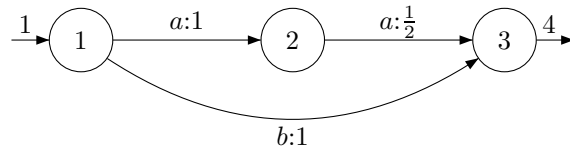
We thus see that the Cardon-Crochemore minimisation algorithm applied to $\mathcal{A}$ outputs a bideterministic weighted automaton over the fraction field $\mathbb{Q}(x, y)$ although the automaton $\mathcal{A}$ is not bideterminisable over the original integral domain $\mathbb{Q}[X]$.

Example 5.5. Consider the weighted automaton $\mathcal{A}$ over the integral domain $\mathbb{Z}$ and alphabet $\Sigma = \{a, b\}$ depicted in Fig. 7. This automaton is clearly bideterministic, so it is trivially bideterminisable.
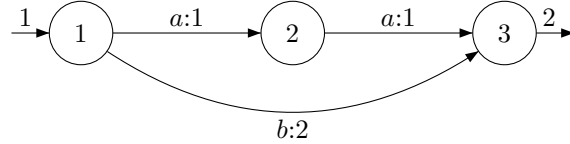
Figure 7: The bideterministic weighted automaton $\mathcal{A}$ over $\mathbb{Z}$.

Now, the output of the Cardon-Crochemore algorithm depends on how precisely the left and right basic languages are computed. Although the exact procedure may vary, a common outcome can be that the right basic language R used in the reduction step (1) and the left basic language L used in the reduction step (2) are both equal to $\{\varepsilon, a, b\}$. In that case, the Cardon-Crochemore algorithm produces the automaton $\mathcal{B}$ in Fig. 8.

Figure 8: The output automaton $\mathcal{B}$ of the Cardon-Crochemore algorithm applied to $\mathcal{A}$.

Although $\mathcal{B}$ is a bideterministic weighted automaton over $\mathbb{Q}$ – which is the fraction field of the integral domain $\mathbb{Z}$ – it contains a weight $1/2$, so it is not a bideterministic weighted automaton over $\mathbb{Z}$. Of course, one could construct similar examples for other algorithms of computing the basic languages as well.

We may thus conclude that the Cardon-Crochemore algorithm applied to a bideterminisable weighted automaton over an integral domain might not always produce a bideterministic weighted automaton over the same domain – although it necessarily outputs a bideterministic weighted automaton over its field of fractions.

5.2. Tropical Semirings

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

In order to obtain these decidability results, we need the following lemma showing that a very specific form of determinisability is decidable for tropical automata.

Lemma 5.6. *Let $\mathbb{T}$ be one of the semirings $\mathbb{N}_{\min}$, $\mathbb{Z}_{\min}$, or $\mathbb{Q}_{\min}$. Let $\mathcal{A}$ be a weighted automaton over $\mathbb{T}$, and $\mathcal{B}$ a deterministic finite automaton without weights. Then it is decidable whether $\mathcal{A}$ is equivalent to some deterministic weighted automaton $\mathcal{B}'$ over $\mathbb{T}$ obtained by assigning weights to $\mathcal{B}$.*

Proof. Positivity of tropical semirings implies that the automaton $\mathcal{B}'$ can only exist when $\mathcal{B}$ recognises the language $\mathrm{supp}(\|\mathcal{A}\|)$. This condition is clearly decidable by removing the weights of $\mathcal{A}$ and deciding equivalence of the resulting nondeterministic finite automaton with $\mathcal{B}$. We may therefore assume that $\mathcal{B}$ indeed recognises $\mathrm{supp}(\|\mathcal{A}\|)$ in what follows.

Moreover, the task is trivial when $\mathcal{B}$ contains no initial state. We therefore assume that $\mathcal{B}$ contains precisely one initial state.

Denote the unknown weights assigned to $\mathcal{B}$ by $x, y_1, \ldots, y_M, z_1, \ldots, z_N$ for some $M, N \in \mathbb{N}$, where x corresponds to the unknown initial weight, $y_1, \ldots, y_M$ correspond to the unknown transition weights, and $z_1, \ldots, z_N$ correspond to the unknown terminal weights. Let $\mathbf{x} = (x, y_1, \ldots, y_M, z_1, \ldots, z_N)$. Given $w \in \mathrm{supp}(\|\mathcal{A}\|)$, let η_i denote, for $i = 1, \ldots, M$, the number of times the unique successful run of $\mathcal{B}$ upon w goes through the transition corresponding to the unknown weight y_i. Moreover, for $j = 1, \ldots, N$, let $\nu_j = 1$ if this unique successful run on w ends in the state corresponding to the unknown weight z_j, and let $\nu_j = 0$ otherwise. Finally, set $\Psi(w) = (1, \eta_1, \ldots, \eta_M, \nu_1, \ldots, \nu_N)$.

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

$$\Psi(w_i) \cdot \mathbf{x}^T = (\|\mathcal{A}\|, w_i) \qquad \text{for } i = 1, \ldots, K, \tag{10}$$

where $w_1, \ldots, w_K \in \mathrm{supp}(\|\mathcal{A}\|)$ are such that $(\Psi(w_1), \ldots, \Psi(w_K))$ 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, $w_1, \ldots, w_K$ can be found as well.

We may thus solve the system (10) 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 [39], and integer linear programming in the latter case [39].

If there is no solution, the automaton $\mathcal{B}'$ does not exist. Otherwise, any solution $\mathbf{x}$ gives us a deterministic 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 $\mathcal{B}' = \mathcal{B}_{\mathbf{x}}$. Otherwise, $\mathcal{B}'$ does not exist. $\qquad\square$

Remark 5.7. The *determinisability* problem for tropical automata – whose decidability status is still open [27, 28] – is clearly recursively enumerable, as one can decide equivalence of a nondeterministic tropical automaton with a deterministic one [2], as well as enumerate all deterministic tropical automata. Lemma 5.6 implies that a computable bound on the state complexity of determinisation of tropical automata would actually be sufficient to establish decidability of this problem: one could simply enumerate all deterministic finite automata up to the size given by the computable bound, and check whether weights can be assigned to any of them such that one obtains a tropical automaton equivalent to the original one.

We now use Lemma 5.6 to establish decidability of the *bideterminisability* problem for weighted automata over tropical semirings.

Theorem 5.8. *Bideterminisability of weighted automata over the tropical 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 – and existence of such an automaton $\mathcal{B}'$ is decidable by Lemma 5.6. □

Note that the above described decision algorithm makes use of deciding equivalence of a nondeterministic tropical automaton with a deterministic one, which is **PSPACE**-complete [2]. Nevertheless, this does not rule out existence of a more effective algorithm. We leave the complexity of the bideterminisability problem for tropical automata open.

5.3. An Undecidability Result

We now show that the decidability result of Theorem 5.8 *does not* generalise to all effective positive semirings.

To this end, let us consider a semiring arising from the semiring of formal languages 2^{Σ^*} over some alphabet Σ by identifying all languages containing at least two different words. In this way, we obtain a semiring $(S_\Sigma, \sqcup, \cdot, \emptyset, \{\varepsilon\})$ containing all singleton languages over Σ, the empty language, and an element $\top$ representing the identified languages with two or more words:

$$S_\Sigma = \{\emptyset\} \cup \binom{\Sigma^*}{1} \cup \{\top\}.$$

The operations of S_Σ are based upon the usual operations on formal languages, *i.e.*,

$$
\begin{aligned}
\emptyset \sqcup L = L \sqcup \emptyset = L \quad & \text{for all } L \in S_\Sigma; \\
\top \sqcup L = L \sqcup \top = \top \quad & \text{for all } L \in S_\Sigma; \\
\{w\} \sqcup \{w\} = \{w\} \quad & \text{for all } w \in \Sigma^*; \\
\{u\} \sqcup \{v\} = \top \quad & \text{for all distinct } u, v \in \Sigma^*; \\
\emptyset \cdot L = L \cdot \emptyset = \emptyset \quad & \text{for all } L \in S_\Sigma; \\
\top \cdot L = L \cdot \top = \top \quad & \text{for all } L \in \binom{\Sigma^*}{1} \cup \{\top\}; \\
\{u\} \cdot \{v\} = \{uv\} \quad & \text{for all } u, v \in \Sigma^*.
\end{aligned}
$$

More concisely, S_Σ can be described as a factor semiring $2^{\Sigma^*}/\equiv$, where $\equiv$ is a congruence on 2^{Σ^*} such that $K, L \subseteq \Sigma^*$ satisfy $K \equiv L$ if and only if either $K = L$, or $|K| \geq 2$ and $|L| \geq 2$. It is easy to see that S_Σ actually is a *positive* semiring.

We now establish undecidability of the bideterminisability problem for weighted automata over the semiring S_Σ in case Σ contains at least two letters. To do so, we reduce the *Post's correspondence problem* (PCP) to our problem. Recall that a PCP instance over an alphabet Σ can be described by a triple (Γ, f, g), where Γ is an alphabet and $f, g \colon \Gamma^* \to \Sigma^*$ are homomorphisms, and the task is to decide whether there exists a nonempty word $w \in \Gamma^+$ such that $f(w) = g(w)$. Post's correspondence problem is famous for being undecidable when Σ contains at least two different letters [38, Section 5.1].

Theorem 5.9. *Bideterminisability of weighted automata over S_Σ is undecidable for any alphabet Σ containing at least two letters.*

Proof. Consider an arbitrary Post's correspondence problem instance (Γ, f, g) over the alphabet Σ. Let $\Gamma = \{c_1, \ldots, c_m\}$, and let us construct a weighted automaton $\mathcal{A}_{\Gamma, f, g}$ over the semiring S_Σ and alphabet Γ depicted in Fig. 9.
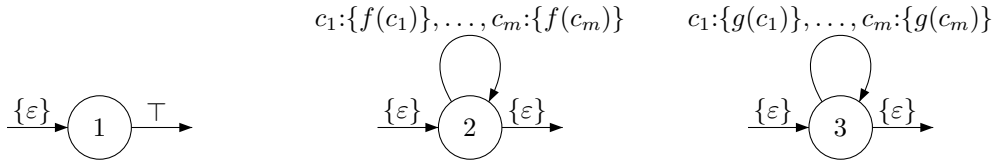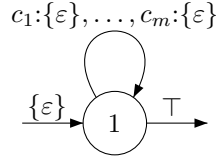

Figure 9: The weighted automaton $\mathcal{A}_{\Gamma, f, g}$ over S_Σ and Γ corresponding to the PCP instance (Γ, f, g).

Given $w \in \Gamma^*$, clearly

$$(\|\mathcal{A}_{\Gamma, f, g}\|, w) = \begin{cases} \top & \text{if } w = \varepsilon, \\ \{x\} & \text{if } w \in \Gamma^+ \text{ and } f(w) = g(w) = x, \\ \top & \text{if } w \in \Gamma^+ \text{ and } f(w) \neq g(w). \end{cases} \tag{11}$$

We prove undecidability of our problem by showing that the automaton $\mathcal{A}_{\Gamma, f, g}$ is bideterminisable if and only if there is no solution to the PCP instance (Γ, f, g), *i.e.*, there is no $w \in \Gamma^+$ such that $f(w) = g(w)$.

Figure 10: The bideterministic weighted automaton $\mathcal{B}$ equivalent to $\mathcal{A}_{\Gamma, f, g}$ in case (Γ, f, g) has no solution.

Indeed, if there is no solution to (Γ, f, g), then (11) implies that $(\|\mathcal{A}_{\Gamma, f, g}\|, w) = \top$ for all $w \in \Gamma^*$. The automaton $\mathcal{A}_{\Gamma, f, g}$ is equivalent to the bideterministic weighted automaton $\mathcal{B}$ in Fig. 10 as a result.

Now, suppose that there is some solution $w \in \Gamma^+$ to the instance (Γ, f, g), and assume for contradiction that $\mathcal{A}_{\Gamma, f, g}$ is equivalent to some bideterministic weighted automaton $\mathcal{C} = (Q, \sigma, \iota, \tau)$ over S_Σ and Γ. As $\mathrm{supp}(\|\mathcal{A}_{\Gamma, f, g}\|) \neq \emptyset$, the automaton $\mathcal{C}$ contains precisely one state with initial weight different from $\emptyset$ – denote this state by q. As it is implied by (11) that $(\|\mathcal{A}_{\Gamma, f, g}\|, w) \in \binom{\Sigma^*}{1}$, necessarily

$$\iota(q) \in \binom{\Sigma^*}{1},$$

and as $(\|\mathcal{A}_{\Gamma, f, g}\|, \varepsilon) = \top$, this also implies that

$$\tau(q) = \top.$$

However, the bideterministic weighted automaton $\mathcal{C}$ cannot have any other state with terminal weight different from $\emptyset$, which means that $(\|\mathcal{A}_{\Gamma, f, g}\|, w) \in \{\emptyset, \top\}$, contradicting the fact that $(\|\mathcal{A}_{\Gamma, f, g}\|, w)$ is a singleton language. □

18

6. Conclusions

The concept of *bideterminism* has been generalised to weighted automata over a semiring. We have seen that trim bideterministic weighted automata over integral domains and over positive semirings are always minimal, generalising the well-known property of bideterministic finite automata without weights [44, 45]. On the contrary, we have observed that this property does not hold over other than zero-divisor free semirings, and thus also over nontrivial commutative rings other than integral domains. For a large class of commutative semirings S including also many finite commutative rings, we have shown that a bideterministic weighted automaton over S might not even admit an equivalent minimal automaton that is bideterministic as well, and we have completely characterised all $m \in \mathbb{N}$ such that this is the case over the ring $\mathbb{Z}/m\mathbb{Z}$.

We also studied the *bideterminisability* problem for weighted automata, in which the task is to decide whether a given weighted automaton over a semiring S admits an equivalent bideterministic automaton over S. We have seen that this problem is decidable over fields by simply running the Cardon-Crochemore minimisation algorithm and checking whether its output is bideterministic or not. On the other hand, we have noted that this simple decision procedure is no longer sufficient for weighted automata over integral domains. Moreover, we have established decidability of the bideterminisability problem for tropical automata – and observed that this result does not generalise to automata over effective positive semirings.

We have left several questions open. For instance: Can one characterise commutative rings over which bideterministic weighted automata always admit equivalent minimal automata that are bideterministic as well? Is the bideterminisability problem decidable for automata over integral domains – or at least over, e.g., completely integrally closed domains? What can one say about the computational complexity of the bideterminisability problem over tropical semirings? And can one characterise positive semirings with decidable bideterminisability problem?

References

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

[2] S. Almagor, U. Boker, and O. Kupferman. What's decidable about weighted automata? *Information and Computation*, 282, 2022. Article 104651.

[3] D. F. Anderson, M. C. Axtell, and J. A. Stickles. Zero-divisor graphs in commutative rings. In M. Fontana, S.-E. Kabbaj, B. Olberding, and I. Swanson, editors, *Commutative Algebra: Noetherian and Non-Noetherian Perspectives*, pages 23–45. Springer, 2011.

[4] D. F. Anderson and P. S. Livingston. The zero-divisor graph of a commutative ring. *Journal of Algebra*, 217(2):434–447, 1999.

[5] D. Angluin. Inference of reversible languages. *Journal of the ACM*, 29(3):741–765, 1982.

[6] M.-P. Béal, S. Lombardy, and J. Sakarovitch. On the equivalence of $\mathbb{Z}$-automata. In *Automata, Languages and Programming, ICALP 2005*, pages 397–409. Springer, 2005.

[7] M.-P. Béal, S. Lombardy, and J. Sakarovitch. Conjugacy and equivalence of weighted automata and functional transducers. In *Computer Science – Theory and Applications, CSR 2006*, pages 58–69. Springer, 2006.

[8] J. Bell and D. Smertnig. Noncommutative rational Pólya series. *Selecta Mathematica*, 27(3):article 34, 2021.

[9] J. P. Bell and D. Smertnig. Computing the linear hull: Deciding Sequential? and Unambiguous? for weighted automata over fields. Available at `https://arxiv.org/abs/2209.02260`, 2022.

[10] J. Berstel and C. Reutenauer. *Noncommutative Rational Series with Applications*. Cambridge University Press, 2011.

[11] A. Cardon and M. Crochemore. Détermination de la représentation standard d'une série reconnaissable. *Informatique Théorique et Applications*, 14(4):371–379, 1980.

[12] M. Ćirić, M. Droste, J. Ignjatović, and H. Vogler. Determinization of weighted finite automata over strong bimonoids. *Information Sciences*, 180:3497–3520, 2010.

[13] F. DeMeyer and L. DeMeyer. Zero divisor graphs of semigroups. *Journal of Algebra*, 283(1):190–198, 2005.

[14] D. Dolžan and P. Oblak. The zero-divisor graphs of rings and semirings. *International Journal of Algebra and Computation*, 22(4), 2012. Article 1250033.

[15] M. Droste, W. Kuich, and H. Vogler, editors. *Handbook of Weighted Automata*. Springer, 2009.

[16] M. Droste and D. Kuske. Weighted automata. In J.-É. Pin, editor, *Handbook of Automata Theory, Vol. 1*, chapter 4, pages 113–150. European Mathematical Society, 2021.

[17] S. Eilenberg. *Automata, Languages, and Machines, Vol. A*. Academic Press, 1974.

[18] J. S. Golan. *Semirings and their Applications*. Kluwer Academic Publishers, 1999.

[19] H. Gruber. Digraph complexity measures and applications in formal language theory. *Discrete Mathematics & Theoretical Computer Science*, 14(2):189–204, 2012.

[20] U. Hebisch and H. J. Weinert. *Semirings*. World Scientific, 1998.

[21] D. Janin. Free inverse monoids up to rewriting. Technical report, LaBRI – Laboratoire Bordelais de Recherche en Informatique, 2015. Available at `https://hal.archives-ouvertes.fr/hal-01182934`.

[22] D. Kirsten. An algebraic characterization of semirings for which the support of every recognizable series is recognizable. *Theoretical Computer Science*, 534:45–52, 2014.

[23] D. Kirsten and S. Lombardy. Deciding unambiguity and sequentiality of polynomially ambiguous min-plus automata. In *Symposium on Theoretical Aspects of Computer Science, STACS 2009*, pages 589–600, 2009.

[24] D. Kirsten and I. Mäurer. On the determinization of weighted automata. *Journal of Automata, Languages and Combinatorics*, 10(2–3):287–312, 2005.

[25] P. Kostolányi. Bideterministic weighted automata. In *Algebraic Informatics, CAI 2022*, pages 161–174, 2022.

[26] P. Kostolányi. Determinisability of unary weighted automata over the rational numbers. *Theoretical Computer Science*, 898:110–131, 2022.

[27] S. Lombardy and J. Mairesse. Max-plus automata. In J.-É. Pin, editor, *Handbook of Automata Theory, Vol. 1*, chapter 5, pages 151–188. European Mathematical Society, 2021.

[28] S. Lombardy and J. Sakarovitch. Sequential? *Theoretical Computer Science*, 356:224–244, 2006.

[29] R. McNaughton. The loop complexity of pure-group events. *Information and Control*, 11(1–2):167–176, 1967.

[30] R. McNaughton. The loop complexity of regular events. *Information Sciences*, 1(3):305–328, 1969.

[31] M. Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–311, 1997.

[32] M. Mohri. Weighted automata algorithms. In M. Droste, W. Kuich, and H. Vogler, editors, *Handbook of Weighted Automata*, chapter 6, pages 213–256. Springer, 2009.

[33] R. S. R. Myers, S. Milius, and H. Urbat. Nondeterministic syntactic complexity. In *Foundations of Software Science and Computation Structures, FOSSACS 2021*, pages 448–468, 2021.

[34] J.-É. Pin. On reversible automata. In *Latin American Symposium on Theoretical Informatics, LATIN 1992*, pages 401–416, 1992.

[35] L. Polák. Minimalizations of NFA using the universal automaton. *International Journal of Foundations of Computer Science*, 16(5):999–1010, 2005.

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

[37] J. Sakarovitch. Rational and recognisable power series. In M. Droste, W. Kuich, and H. Vogler, editors, *Handbook of Weighted Automata*, chapter 4, pages 105–174. Springer, 2009.

[38] A. Salomaa. *Computation and Automata*. Cambridge University Press, 1985.

[39] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, 1986.

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

[41] P. Shankar, A. Dasgupta, K. Deshmukh, and B. S. Rajan. On viewing block codes as finite automata. *Theoretical Computer Science*, 290(3):1775–1797, 2003.

[42] J. B. Stephen. Presentations of inverse monoids. *Journal of Pure and Applied Algebra*, 63(1):81–112, 1990.

[43] H. Tamm. On transition minimality of bideterministic automata. *International Journal of Foundations of Computer Science*, 19(3):677–690, 2008.

[44] H. Tamm and E. Ukkonen. Bideterministic automata and minimal representations of regular languages. In *Implementation and Application of Automata, CIAA 2003*, pages 61–71, 2003.

[45] H. Tamm and E. Ukkonen. Bideterministic automata and minimal representations of regular languages. *Theoretical Computer Science*, 328(1–2):135–149, 2004.