Filomat 33:9 (2019), 2809–2831
https://doi.org/10.2298/FIL1909809S

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Improved Algorithms for Computing the Greatest Right and Left Invariant Boolean Matrices and Their Application

Stefan Stanimirović[a], Aleksandar Stamenković[a], Miroslav Ćirić[a]

[a]*University of Niš, Faculty of Sciences and Mathematics, Višegradska 33, 18000 Niš, Serbia*

Abstract. We define right and left invariant matrices as Boolean matrices that are solutions to certain systems of matrix equations and inequalities over additively idempotent semirings. We provide improved algorithms for computing the greatest right and left invariant equivalence and quasi-order matrices. The improvements are based on the usage of the well-known partition refinement technique. Afterwards, we present the application of right invariant matrices in the determinization of weighted automata over additively idempotent, commutative and zero-divisor free semirings. In particular, we provide improvements of the well-known determinization method of weighted automata over tropical semirings given by Mohri [Computational Linguistics 23 (2) (1997) 269–311].

1. Introduction

Algorithms for checking and computing the behavioural equivalences of various systems have been investigated in many areas of mathematics and computer science under different names, most commonly as simulations and bisimulations. Under the later notions they have been investigated in numerous settings, e.g. in concurrency theory by Park [40] and Milner [37], in modal logic by van Benthem [3], in set theory by Forti and Honsell [19], etc. They have also taken an important place in the automata theory, or more precisely, in the context of nondeterministic [12, 17], weighted [4, 16], fuzzy [11, 13, 42, 46] and probabilistic automata [2], etc. When observed between two identical nondeterministic automata, bisimulations have been studied under the name right and left invariant relations [5–8, 25–28], and have been used for modeling the indistinguishability of states of nondeterministic finite automata.

Right and left invariant fuzzy relations were studied in [14, 15, 36, 43]. Their definitions are based on the corresponding crisp counterparts. Precisely, they are defined as solutions to certain systems of fuzzy relation equations and inequalities (see also [21–24]). They have been used to model the indistinguishability of states of fuzzy finite automata, and have been applied in the state reduction and determinization of fuzzy automata (cf. above mentioned references and also [31, 35, 44]). The greatest right and left invariant fuzzy quasi-orders and equivalences were calculated in [14, 15, 43] by algorithms that approximate the greatest fixpoint by the means of the Kleene fixpoint theorem, and in [36] by partition refinement algorithms. Both

2010 *Mathematics Subject Classification*. Primary 15A24; Secondary 15A80, 16Y60, 68Q70
Keywords. Matrices over semirings, Boolean matrices; Additively idempotent semirings; Partition refinement; Weighted automata; Determinization;
Received: 28 September 2018; Revised: 28 February 2019; Accepted: 26 March 2019
Communicated by Dragan S. Djordjević
Research supported by Ministry of Education, Science and Technological Development, Republic of Serbia, Grant No. 174013
Email addresses: stefan.stanimirovic@pmf.edu.rs (Stefan Stanimirović), aca@pmf.ni.ac.rs (Aleksandar Stamenković), miroslav.ciric@pmf.edu.rs (Miroslav Ćirić)

variants run in the same, polynomial-time complexity, and rely on the usage of the right and left residuals of fuzzy relations and fuzzy subsets.

Previous approach to defining and computing right and left invariant relations cannot be directly applied to weighted automata over arbitrary semirings. As pointed out in [16], two issues should be addressed: we need to provide an ordering in the semiring, and to provide a way to calculate right and left residuals in order to compute the solutions to certain matrix equations and inequalities in the semiring. To overcome these problems, in the same paper, the authors have defined several types of simulations and bisimulations for weighted automata over additively idempotent semirings as Boolean matrices that satisfy ceratin systems of matrix equations and inequalities. This is due to the fact that in additively idempotent semirings a natural ordering exists. In addition, right and left residuals exist and can be computed in the class of Boolean matrices. The latter is due to the fact that the zero and one of an additively idempotent semiring form a subsemiring isomorphic to the Boolean semiring, thus matrices with entries in this subsemiring can be treated as Boolean matrices, that is, as ordinary binary relations. Also, the authors in [16] have provided algorithms for testing the existence and computing the greatest simulations and bisimulations, if such exist.

Motivated by previous results, we define right and left invariant matrices for weighted automata over additively idempotent semirings. They are in a tight connection with certain types of simulations defined in [16]. Therefore, the polynomial-time algorithms from [16] for computing the greatest simulations can be used for computing the greatest right and left invariant quasi-order and equivalence matrices. We further improve those algorithms by using the well-employed technique of partition refinement. Originally defined in the relational setting by Kanellakis and Smolka [32], and further improved by Paige and Tarjan [39], the partition refinement technique is used in many different area of computer science dealing with graphs, strings, boolean matrices or automata [20]. While the direct method from [16] can calculate the greatest right and left invariant quasi-order and equivalence matrices in $O(mn^5)$, we show that our methods for computing the greatest right and left invariant equivalence matrices run in $O(mn^3)$ time, with m being the size of the alphabet, and n being the size of the set of states of the weighted automaton.

Further, we study weakly right and left invariant quasi-order and equivalence matrices, and give algorithms for their computation similar to those given for fuzzy automata in [43]. In the end, we show that right and left invariant matrices can be used in the determinization of weighted automata over additively idempotent, commutative and zero-divisor free semirings, as it was done for fuzzy automata in [44]. Determinization of automata is a well-elaborated problem with a long history, starting back from the paper of Rabin and Scott [41] in 1959. The original determinization algorithm, also known as the *subset construction*, meant the construction of the deterministic automaton in which every state represents the set of states that are reached in the original nondeterministic automaton at the current position. The resulting automaton has size 2^n, given a nondeterministic automaton of size n. The determinization has a wise range of applications, including lexical analysis [45, Chapter 8], pattern matching based on regular expressions [29, Chapter 4], analysis of protein sequences [34], and so forth. It is of a special interest in the content of weighted finite automata (WFAs, for short), where the exponential blow up of states is even more present, resulting sometimes even in deterministic WFAs with infinite number of states (see a classical reference on weighted automata [18], as well as particular results on determinization of weighted automata over tropical semirings [38], arbitrary semirings [33], strong bimonoids [10, 30]). As a particular case, our algorithm outperforms the determinization method by Mohri [38] for weighted automata over tropical semirings.

In Section 2 we recall basic notions and notations regarding additively idempotent semirings, matrices over them, Boolean matrices and their right and left residuals. Further, we recall notion of WFAs and deterministic WFAs, as well as factorizations. In Section 3, we define right and left invariant matrices and their weakly counterparts. We present improved algorithms for computing the greatest right and left invariant equivalence matrices, as well as their generalized versions for computing the greatest right and left invariant quasi-order matrices. In Section 4 we show the applications of right and left invariant matrices and their weakly counterparts in the determinization of weighted automata over commutative, additively idempotent, zero-divisor free semirings. As shown by examples in the same section, our method can produce finite deterministic weighted automata in some cases when the previous algorithms result in infinite ones.

2. Preliminaries

Throughout the paper, $\mathbb{N}$ denotes the set of natural numbers (without zero) and $\mathbb{R}$ denotes the set of real numbers. Also, for an arbitrary assertion φ in the classical Boolean logic let $\lceil \varphi \rceil$ denote its truth value, i.e., $\lceil \varphi \rceil = 1$ if φ is true, and $\lceil \varphi \rceil = 0$ if φ is false.

2.1. Semirings and matrices

A *semiring* is a five-tuple $(S, +, \cdot, 0, 1)$ consisting of a set S together with two binary operations *addition* $+$ and *multiplication* $\cdot$ on S, along with two constants $0, 1 \in S$ such that:

(i) $(S, +, 0)$ is a commutative monoid,
(ii) $(S, \cdot, 1)$ is a monoid,
(iii) Multiplication left and right distributes over addition, i.e. $a \cdot (b + c) = a \cdot b + a \cdot c$ and $(a + b) \cdot c = a \cdot c + b \cdot c$ for every $a, b, c \in S$,
(iv) Multiplication by 0 annihilates S, i.e. $a \cdot 0 = 0 \cdot a = 0$ for every $a \in S$.

As usual, we identify the semiring with its carrier set. A semiring S is called *zero-divisor free* if $a \cdot b = 0$ implies $a = 0$ or $b = 0$ for every $a, b \in S$, and it is called *commutative* if $(S, \cdot, 1)$ is a commutative monoid. If, in addition to (i)-(iv), S satisfies the condition

(v) $a + a = a$ for every $a \in S$,

then S is called *additively idempotent*. On an additively idempotent semiring S we define a partial order by

$$a \leqslant b \quad \Leftrightarrow \quad a + b = b, \tag{1}$$

for every $a, b \in S$. If S is additively idempotent semiring, then the monoid $(S, +, 0)$ is a join-semilattice (upper semilattice) with zero.

Let A and B be finite nonempty sets. An $A \times B$-matrix over S is any mapping $\mu : A \times B \to S$. The set of all $A \times B$-matrices over S is denoted with $S^{A\times B}$. An A-column vector over S is any $A \times C$-matrix over S with $|C| = 1$. Similarly, an A-row vector over S is any $C \times A$-matrix over S with $|C| = 1$. We denote the set of all A-column vectors over S with $S^{A\times 1}$, and the set of all A-row vectors over S with $S^{1\times A}$. For the sake of simplicity, we identify any A-column vector or A-row vector with the mapping $\mu : A \to S$, and call it simply an A-vector. Similarly, with S^A we denote the set of all A-vectors. For a matrix $\mu \in S^{A\times B}$ and elements $a \in A$ and $b \in B$, we define the *ath row-vector* $a\mu \in S^{1\times B}$ and the *bth column-vector* $\mu b \in S^{A\times 1}$ with $a\mu(y) = \mu(a, y)$ and $\mu b(x) = \mu(x, b)$ for every $y \in B$ and $x \in A$. For any matrix $\mu \in 2^{A\times B}$, its *transposed* matrix $\mu^{\mathrm{T}} \in 2^{B\times A}$ is defined in a usual way, i.e. $\mu^{\mathrm{T}}(b, a) = \mu(a, b)$, for every $a \in A$ and $b \in B$.

Let A and B be finite nonempty sets. For two $A\times B$-matrices $\mu_1, \mu_2 \in S^{A\times B}$, their *matrix sum* $\mu_1 + \mu_2 \in S^{A\times B}$ and the *Hadamard product* $\mu_1 \odot \mu_2 \in S^{A\times B}$ are defined pointwise by

$$(\mu_1 + \mu_2)(a, b) = \mu_1(a, b) + \mu_2(a, b), \quad (\mu_1 \odot \mu_2)(a, b) = \mu_1(a, b) \cdot \mu_2(a, b),$$

for every $a \in A$ and $b \in B$. Also, let C be a finite nonempty set, and $\mu_1 \in S^{A\times B}$ and $\mu_2 \in S^{B\times C}$. Then, the *matrix product* $\mu_1 \cdot \mu_2 \in S^{A\times C}$ is defined for every $a \in A$ and $c \in C$ with

$$(\mu_1 \cdot \mu_2)(a, c) = \sum_{b\in B} \mu_1(a, b) \cdot \mu_2(b, c). \tag{2}$$

When $\mu_1 \in S^{1\times B}$ or $\mu_2 \in S^{B\times 1}$, then (2) defines *matrix-vector products*, and when $\mu_1 \in S^{1\times B}$ and $\mu_2 \in S^{B\times 1}$, then (2) defines the *scalar product* of μ_1 and μ_2. Note that the matrix product and matrix-vector products are associative (whenever they are defined), due to the distributivity of the multiplication over the addition. In addition, $(S^{A\times A}, \cdot, I_A)$ is a monoid, where I_A is the *identity matrix* defined with $I_A(a, a) = 1$ for every $a \in A$, and $I_A(a, b) = 0$ for every $a, b \in A$ with $a \neq b$. We also define the *universal matrix* $U_A \in S^{A\times A}$ with $U_A(a, b) = 1$ for every $a, b \in A$. For $n \in \mathbb{N}_0$, the nth matrix power of a matrix $\varrho \in S^{A\times A}$ is a matrix $\varrho^n \in S^{A\times A}$ defined inductively by $\varrho^0 = I_A$ and $\varrho^{n+1} = \varrho^n \cdot \varrho$.

Let S be an additively idempotent semiring. For two $A \times B$-matrices $\mu_1, \mu_2 \in S^{A\times B}$, their *ordering* $\mu_1 \leqslant \mu_2$ is defined pointwise, i.e.

$$\mu_1 \leqslant \mu_2 \quad \text{if} \quad \mu_1(a,b) \leqslant \mu_2(a,b), \tag{3}$$

for every $a \in A$ and $b \in B$. This ordering is compatible with the matrix sum, product and transposition, i.e., for every $\mu_1, \mu_2, \mu_1', \mu_2' \in S^{A\times B}$ and $\nu_1, \nu_2 \in S^{B\times C}$ we have:

$$\begin{aligned}
&\mu_1 \leqslant \mu_2 \text{ and } \mu_1' \leqslant \mu_2' \text{ implies } \mu_1 + \mu_1' \leqslant \mu_2 + \mu_2',\\
&\mu_1 \leqslant \mu_2 \text{ and } \nu_1 \leqslant \nu_2 \text{ implies } \mu_1 \cdot \nu_1 \leqslant \mu_2 \cdot \nu_2,\\
&\mu_1 \leqslant \mu_2 \text{ implies } \mu_1^{\mathrm{T}} \leqslant \mu_2^{\mathrm{T}}.
\end{aligned}$$

For any additively idempotent semiring S and a nonempty set A, the structure $(S^{A\times A}, +, \cdot, 0_A, I_A)$ forms an additively idempotent semiring, where 0_A denotes the zero matrix defined with $0_A(a,a) = 0$ for every $a \in A$. In addition, the natural ordering of this additively idempotent semiring, defined by the rule (1), coincides with the pointwise ordering of matrices (3).

The following common notations are well-known in the recent literature (cf. [16]). We use $\sum$ and $\prod$ to denote the addition and multiplication of family of matrices in $S^{A\times B}$, respectively, as well as $\bigodot$ to denote the Hadamard product of family of matrices in $S^{A\times B}$. We comply with the following standard conventions for square matrices in $S^{A\times A}$: $\sum_\emptyset = 0_A$ and $\prod_\emptyset = \bigodot_\emptyset = I_A$.

Matrices over the Boolean semiring are called *Boolean matrices*. The set of all $A \times B$-Boolean matrices is denoted with $2^{A\times B}$. For an additively idempotent semiring S, the set $\{0,1\}$ forms a Boolean subsemiring of S. Thus, matrices over S taking values in $\{0,1\}$ can be identified with Boolean matrices, i.e. we can say that $2^{A\times B}$ is a subsemiring of an additively idempotent semiring $S^{A\times B}$.

Moreover, Boolean matrices from $2^{A\times B}$ can be identified with binary relations between sets A and B. From this point of view, the ordering of matrices corresponds to the set-theoretical inclusion, the matrix sum and the Hadamard product correspond to the set-theoretical union and intersection, and the matrix product corresponds to the composition of binary relations. In addition, we define the *difference* between $\varrho \in 2^{A\times B}$ and $\theta \in 2^{A\times B}$ as the Boolean matrix $\phi = \theta - \varrho \in 2^{A\times B}$ that satisfies $\phi + \varrho = \theta$. Note that the difference between matrices correspond to the set-theoretical difference.

In addition, we say that a Boolean matrix $\varrho \in 2^{A\times A}$ is *reflexive* if $\varrho(a,a) = 1$ for every $a \in A$, *symmetric* if $\varrho(a,b) = \varrho(b,a)$ for every $a, b \in A$, and *transitive* if $\varrho(a,b) \cdot \varrho(b,c) \leqslant \varrho(a,c)$ for every $a,b,c \in A$. A reflexive and transitive Boolean matrix is called a *quasi-order matrix*, and a symmetric quasi-order matrix is called an *equivalence matrix*. Note that every quasi-order matrix, and hence, every equivalence matrix, is additively idempotent, i.e. it satisfies $\varrho \cdot \varrho = \varrho$. Also, it is easy to verify that for every two quasi-order (resp. equivalence) matrices $\varrho, \theta \in 2^{A\times A}$ we have that $\varrho \odot \theta \in 2^{A\times A}$ is also a quasi-order (resp. equivalence) matrix. In addition, the following lemma states the basic property of quasi-order (and also equivalence) matrices.

Lemma 2.1. *Let $\varrho, \theta \in 2^{A\times A}$ be two quasi-order matrices such that $\varrho \leqslant \theta$. Then $\varrho \cdot \theta = \theta \cdot \varrho = \theta$.*

For an equivalence matrix $\varrho \in 2^{A\times A}$, we have that $a\varrho = \varrho a$, therefore, we use the common notion $\varrho^a = a\varrho = \varrho a$ to denote the *equivalence class* of ϱ by a. The set of all equivalence classes of ϱ is denoted with A/ϱ and called the *factor set* of A with respect to ϱ. As in the case of the equivalence relations, we have that $\varrho^a \odot \varrho^b$ is the zero vector when $\varrho(a,b) = 0$, $\varrho^a = \varrho^b$ when $\varrho(a,b) = 1$, and $\sum_{a\in A} \varrho^a = 1^A$, where 1^A is the A-vector whose all values are 1.

Let $\varrho, \theta \in 2^{A\times B}$ be quasi-order matrices. We say that ϱ is a *refinement* of θ if $\varrho \leqslant \theta$, i.e., if ϱ is included in θ. It is easy to check that ϱ is a refinement of θ if and only if every row vector of ϱ is included in some row vector of θ (or equivalently, every column vector of ϱ is included in some column vector of θ).

Let S be an additively idempotent semiring, and let $\alpha \in S^{A\times C}$ and $\beta \in S^{B\times C}$. Then the *Boolean left residual* of β by α is the Boolean matrix $\beta/\alpha \in 2^{B\times A}$ defined by

$$(\beta/\alpha)(b,a) = \lceil a\alpha \leqslant b\beta \rceil, \tag{4}$$

for every $b \in B$ and $a \in A$. The Boolean left residual $\beta/\alpha \in 2^{B\times A}$ of β by α is the greatest solution in the set of all Boolean $B \times A$-matrices to the matrix inequality $\chi \cdot \alpha \leqslant \beta$, with $\chi \in 2^{B\times A}$ being an unknown matrix. Dually, for $\alpha \in S^{C\times A}$ and $\beta \in S^{C\times B}$, we define the *Boolean right residual* of β by α as the Boolean matrix $\alpha\backslash\beta \in 2^{A\times B}$ defined by

$$(\alpha\backslash\beta)(a,b) = \lceil \alpha a \leqslant \beta b \rceil, \tag{5}$$

for every $a \in A$ and $b \in B$. The Boolean right residual $\alpha\backslash\beta \in 2^{A\times B}$ of β by α is the greatest solution in the set of all Boolean $A \times B$-matrices to the matrix inequality $\alpha \cdot \chi \leqslant \beta$, with $\chi \in 2^{A\times B}$ being an unknown matrix. In addition, for $\alpha \in S^{A\times A}$ and $\beta \in S^{A\times A}$, we define a Boolean matrix $\alpha|\beta \in 2^{A\times A}$ as $\alpha|\beta = (\alpha\backslash\beta) \odot (\alpha/\beta)$.

Also, let $\nu \in S^A$ and $\eta \in S^B$. Then the *Boolean left residual* of η by ν and the *Boolean right residual* of η by ν are Boolean matrices $\eta/\nu \in 2^{B\times A}$ and $\nu\backslash\eta \in 2^{A\times B}$ defined by

$$(\eta/\nu)(b,a) = (\nu\backslash\eta)(a,b) = \lceil \nu(a) \leqslant \eta(b) \rceil,$$

for every $b \in B$ and $a \in A$, i.e. we have $(\eta/\nu) = (\nu\backslash\eta)^{\mathrm{T}}$. The Boolean left residual $\eta/\nu \in 2^{B\times A}$ of η by ν is the greatest solution in the set of all Boolean $B \times A$-matrices to the matrix inequality $\chi \cdot \nu \leqslant \eta$, with $\chi \in 2^{B\times A}$ being an unknown matrix, while the Boolean right residual $\nu\backslash\eta \in 2^{A\times B}$ of η by ν is the greatest solution in the set of all Boolean $A \times B$-matrices to the matrix inequality $\nu \cdot \chi \leqslant \eta$, with $\chi \in 2^{A\times B}$ being an unknown matrix. For more information on Boolean right and left residuals we refer to [16]. In addition, for $\nu, \eta \in S^A$ we define an $A \times A$-Boolean matrix $\nu|\eta \in 2^{A\times A}$ with $\nu|\eta = (\nu\backslash\eta) \odot (\nu/\eta)$, or equivalently,

$$(\nu|\eta)(a,b) = \lceil \nu(a) = \eta(b) \rceil, \tag{6}$$

for every $a, b \in A$.

The following properties related to Boolean residuals are used through the rest of the paper.

Lemma 2.2. *a) Let $\varrho, \theta \in 2^{A\times A}$ be quasi-order matrices such that ϱ is a refinement of θ. Then ϱ is also a refinement of $\theta a/\theta a$, for every $a \in A$.*

b) Let $\varrho, \theta \in 2^{A\times A}$ be equivalence matrices such that ϱ is a refinement of θ. Then ϱ is also a refinement of $\theta^a|\theta^a$, for every $a \in A$.

Proof. a) By $\varrho \leqslant \theta$ and the fact that θ is a quasi-order matrix, it follows that $\varrho \cdot \theta \leqslant \theta \cdot \theta = \theta$. Choose arbitrary $a, b \in A$. Then we have

$$(\varrho \cdot \theta)(b,a) = \sum_{c\in A} \varrho(b,c) \cdot \theta(c,a) = \sum_{c\in A} \varrho(b,c) \cdot \theta a(c) = (\varrho \cdot \theta a)(b), \tag{7}$$

thus we have $\varrho \cdot \theta a \leqslant \theta a$, or equivalently, $\varrho \leqslant \theta a/\theta a$, for every $a \in A$.

b) Since ϱ and θ are equivalence matrices, by a) we have $\varrho \leqslant \theta^a/\theta^a$, for every $a \in A$. In addition, by the symmetry of ϱ we obtain that, for every $a \in A$, $\varrho = \varrho^{\mathrm{T}} \leqslant (\theta^a/\theta^a)^{\mathrm{T}} = \theta^a\backslash\theta^a$. Therefore, $\varrho \leqslant \theta^a|\theta^a$ for every $a \in A$. □

2.2. Weighted automata

In the rest of the paper, X^+ and X^* denote, respectively, the free semigroup and the free monoid over an alphabet X, and ε denotes the *empty word* in X^*.

Let S be an arbitrary semiring. A *weighted finite automaton* (WFA, for short) over X and S, or simply just a weighted finite automaton, is a quadruple $\mathcal{A} = (A, \delta, \sigma, \tau)$, where A is a finite nonempty *set of states*, $\delta : A \times X \times A \to S$ is a *weighted transition function*, $\sigma \in S^A$ is an *initial weighted vector* and $\tau \in S^A$ is a *final weighted vector*. For each $x \in X$ we define a *weighted transition matrix* $\delta_x \in S^{A\times A}$ with $\delta_x(a,b) = \delta(a,x,b)$, for all $a, b \in A$. In addition, for every $u \in X^*$ we define an $A \times A$-matrix $\delta_u \in S^{A\times A}$ inductively as follows: $\delta_\varepsilon = I_A$, and for every $u \in X^*$ and $x \in X$ we set $\delta_{ux} = \delta_u \cdot \delta_x$. Moreover, for every $u \in X^*$ we define matrices $\sigma_u \in S^{A\times A}$ and $\tau_u \in S^{A\times A}$ with $\sigma_u = \sigma \cdot \delta_u$ and $\tau_u = \delta_u \cdot \tau$.

We visualize a WFA $\mathcal{A}$ by a labelled directed graph whose nodes are states of A, and an edge from a node a into a node b is labelled by pairs of the form $x/\delta(a, x, b)$, for any $x \in X$. Also, we represent σ by drawing for each node a the ingoing arrow labelled by $\sigma(a)$, and represent τ by double-circling every node a such that $\tau(a) \neq 0$, and putting a label $\tau(a)$ by that node. We call this graph the *transition graph* of $\mathcal{A}$. Usually, edges and ingoing arrows labelled by 0 are not shown in the transition graph. Also, we do not explicitly show the label on the ingoing arrow or double-circled node if it is equal to 1.

A *formal power series* over X and S, or simply just a *series*, is any mapping $\varphi : X^* \to S$. The *behaviour* of a weighted finite automaton $\mathcal{A} = (A, \delta, \sigma, \tau)$ is the series $[\![\mathcal{A}]\!]$ defined by

$$[\![\mathcal{A}]\!](u) = \sigma \cdot \delta_u \cdot \tau = \sum_{a,b \in A} \sigma(a) \cdot \delta_u(a, b) \cdot \tau(b), \tag{8}$$

for every $u \in X^*$. Weighted automata $\mathcal{A}$ and $\mathcal{B}$ are *equivalent* if they have the same behaviour, i.e., if $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!]$.

A *complete deterministic* (or in some sources, *sequential*) weighted finite automaton (CDWFA, for short) is a quadruple $\mathcal{A} = (A, \delta, \{a_0/\sigma(a_0)\}, \tau)$, where A is a finite nonempty set of states, $\delta : A \times X \times A \to S$ is a weighted transition function, such that for all $x \in X$ and $b, c_1, c_2 \in A$, if $\delta_x(b, c_1) \neq 0$ and $\delta_x(b, c_2) \neq 0$ then $c_1 = c_2$, and for every $x \in X$ and $b \in A$ there exists $c \in A$ such that $\delta_x(b, c) \neq 0$, $a_0 \in A$ is the initial state with the initial weight $\sigma(a_0) \neq 0$, and $\tau \in S^A$ is a final weighted vector. Also, we allow the set A to be infinite, and then $\mathcal{A}$ is called a complete deterministic weighted automaton (for short: CDWA).

The behaviour of the CDWA is the series defined for every $u = x_1 x_2 \ldots x_n \in X^*$ with

$$[\![\mathcal{A}]\!](u) = \sigma(a_0) \cdot \left(\prod_{i=1}^{n} \delta_{x_i}(a_{i-1}, a_i) \right) \cdot \tau(a_n), \tag{9}$$

where $a_1, a_2 \ldots, a_n \in A$ is a unique sequence of states such that $\delta_{x_i}(a_{i-1}, a_i) > 0$ for every $i \in \{1, 2, \ldots, n\}$.

2.3. Factorizations

Let A be a nonempty finite set and S be an arbitrary zero-divisor free semiring. We call an ordered pair $D = (f, g)$ of functions $f : S^A \to S^A$ and $g : S^A \to S$ a *factorization of dimension* A, or simply just a *factorization* when A is clear from the context, if the following two properties hold:

$$\mu = g(\mu) \cdot f(\mu), \quad \text{for every } \mu \in S^A,$$
$$g(0_A) = 1.$$

For every $\mu \in S^A$ we have $g(\mu) \neq 0$ and $\mu = 0_A$ if and only if $f(\mu) = 0_A$. The factorization $D_N = (f_N, g_N)$ of A, where $g_N(\mu) = 1$ and $f_N(\mu) = \mu$, for every $\mu \in S^A$, is called the *trivial factorization*. Every zero-divisor free semiring admits a factorization, namely the trivial factorization. A factorization $D = (f, g)$ is called *maximal* if $f(a \cdot \mu) = f(\mu)$ for every $a \in S$ and $\mu \in S^A$ such that $a \cdot \mu \neq 0_A$. As noted in [33], the restriction "$a \cdot \mu \neq 0_A$" is necessary, since otherwise we have that $f(\mu) = f(0 \cdot \mu) = f(0_A)$, for every $\mu \in S^A$, and the notion of the maximal factorization becomes meaningless. If $D = (f, g)$ is a maximal factorization, then $f(f(\mu)) = f(g(\mu) \cdot f(\mu)) = f(\mu)$ for every $\mu \in S^A$.

Example 2.3. *In this example, we present maximal factorizations for some additively idempotent semirings. Let A be a nonempty finite set.*

1. *The* Boolean semiring *is the structure* $(\{0, 1\}, \max, \min, 0, 1)$. *Any two-element additively idempotent semiring is isomorphic to the Boolean semiring. The Boolean semiring admits only the trivial factorization, i.e.* $g(\mu) = 1$ *and* $f(\mu) = \mu$, *for every* $\mu \in \{0, 1\}^A$, *which is also a maximal factorization.*

2. *The* tropical semiring *is the semiring* $(\mathbb{R}_+ \cup \{\infty\}, \min, +, \infty, 0)$. *In the tropical semiring, we define the* Mohri's factorization *with* $g(\mu) = \min_{a \in A} \mu(a)$ *and* $f(\mu) = \mu'$, *with* $\mu' \in (\mathbb{R}_+ \cup \{\infty\})^A$ *defined with* $g(\mu) + \mu' = \mu$, *for every* $\mu \in (\mathbb{R}_+ \cup \{\infty\})^A$. *The A-vector μ' is uniquely determined. Then the Mohri's factorization is a maximal factorization.*

3. *The* Viterbi semiring *(also known as the* probabilistic semiring*) is the semiring* $([0,1], \max, \cdot, 0, 1)$. *In the Viterbi semiring, we also define the* Mohri's factorization *with* $g(\mu) = \max_{a\in A} \mu(a)$ *and* $f(\mu) = \mu'$, *with* $\mu' \in [0,1]^A$ *defined with* $g(\mu) \cdot \mu' = \mu$, *for every* $\mu \in [0,1]^A$. *The A-vector* μ' *is also uniquely determined, and the Mohri's factorization is a maximal factorization.*

Let us note that previous examples of additively idempotent semirings are all also examples of commutative and zero-divisor free semirings.

3. Right and left invariant Boolean matrices and weakly counterparts

Let $\mathcal{A} = (A, \delta, \sigma, \tau)$ be a WFA over an additively idempotent semiring S. A Boolean $A\times A$-matrix $\varrho \in 2^{A\times A}$ over S is called *r-stable* if

$$\varrho \cdot \delta_x \cdot \varrho \leqslant \delta_x \cdot \varrho, \quad \text{for every } x \in X, \tag{10}$$

and is called *right invariant* if it is r-stable and it satisfies

$$\varrho \cdot \tau \leqslant \tau. \tag{11}$$

Dually, a Boolean $A \times A$-matrix $\varrho \in 2^{A\times A}$ over S is called *l-stable* if

$$\varrho \cdot \delta_x \cdot \varrho \leqslant \varrho \cdot \delta_x, \quad \text{for every } x \in X, \tag{12}$$

and is called *left invariant* if it is l-stable and it satisfies

$$\sigma \cdot \varrho \leqslant \sigma. \tag{13}$$

A Boolean $A \times A$-matrix $\varrho \in 2^{A\times A}$ is called stable if it is both r-stable and l-stable. Furthermore, a Boolean $A \times A$-matrix $\varrho \in 2^{A\times A}$ is called *weakly right invariant* if

$$\varrho \cdot \tau_u \leqslant \tau_u, \quad \text{for every } u \in X^*. \tag{14}$$

and is called *weakly left invariant* if

$$\sigma_u \cdot \varrho \leqslant \sigma_u, \quad \text{for every } u \in X^*. \tag{15}$$

By the induction on the length of the word $u \in X^*$, one can easily verify that every right (resp. left) invariant matrix is weakly right (resp. weakly left) invariant. Note that, if ϱ is a reflexive matrix, then all systems of inequalities (10)–(15) become systems of equations.

Right and left invariant matrices are closely related to simulations for WFAs over additively idempotent semirings introduced in [16]. Namely, the following lemma holds.

Lemma 3.1. *Let* $\mathcal{A} = (A, \delta, \sigma, \tau)$ *be a WFA over an additively idempotent semiring S, and let* $\varrho \in 2^{A\times A}$ *be a quasi-order matrix. Then the following two conditions are equivalent for every* $x \in X$:

a) $\varrho \cdot \delta_x \cdot \varrho = \delta_x \cdot \varrho$,
b) $\varrho \cdot \delta_x \leqslant \delta_x \cdot \varrho$.

Proof. Choose an arbitrary $x \in X$. If $\varrho\cdot\delta_x\cdot\varrho = \delta_x\cdot\varrho$, then by the reflexivity of ϱ we obtain $\varrho\cdot\delta_x \leqslant \varrho\cdot\delta_x\cdot\varrho = \delta_x\cdot\varrho$. Conversely, if $\varrho \cdot \delta_x \leqslant \delta_x \cdot \varrho$, since ϱ is a quasi-order matrix we have $\varrho \cdot \delta_x \cdot \varrho \leqslant \delta_x \cdot \varrho \cdot \varrho = \delta_x \cdot \varrho$. The other inequality follows from the reflexivity of ϱ. □

By Lemma 3.1 we have that a quasi-order matrix $\varrho \in 2^{A\times A}$ is right invariant if its transpose matrix ϱ^{T} is a forward simulation on $\mathcal{A}$. Similarly, we prove that a quasi-order matrix is left invariant if it is a backward simulation on $\mathcal{A}$.

Theorem 3.2. *Let $\mathcal{A} = (A, \delta, \sigma, \tau)$ be a WFA over an additively idempotent semiring S. Then there exists the greatest right and left invariant (resp. weakly right and weakly left invariant) quasi-order matrix.*

Proof. Since the identity matrix I_A is right invariant quasi-order matrix, the family $\{\varrho_i\}_{i\in I}$ of all right invariant quasi-order matrices is nonempty. By employing Lemma 3.1, we easily conclude that the sum of elements from the family $\{\varrho_i\}_{i\in I}$ is also right invariant matrix, but not necessarily a quasi-order matrix. On the other hand, since the matrix square (and thus, an arbitrary matrix power) of the right invariant quasi-order matrix is also a right invariant quasi-order matrix, we conclude that the transitive closure of the sum of the family $\{\varrho_i\}_{i\in I}$, defined by

$$\varrho^{\mathrm{ri}} = \sum_{n\in\mathbb{N}} \Big(\sum_{i\in I} \varrho_i\Big)^n,$$

is also a right invariant quasi-order matrix. Therefore, ϱ^{ri} is the greatest right invariant quasi-order matrix. Similarly we prove the assertion for the greatest left invariant matrix, and the same arguments are valid for weakly right and weakly left invariant matrices. □

The algorithms for computing the greatest quasi-order matrices can be easily derived from the algorithms for computing the greatest simulations between weighted automata [16]. The following theorem provides a method for constructing the greatest solution to the system of matrix inequalities (10)–(11). The method is based on computing Boolean left residuals of matrices defined by rule (4). Since it represents a variation of [16, Theorem 5.4], we omit its proof.

Theorem 3.3. *Let $\mathcal{A} = (A, \delta, \sigma, \tau)$ be a WFA over an additively idempotent semiring, and let $\{\varrho_k\}_{k\in\mathbb{N}} \subseteq 2^{A\times A}$ be a sequence of Boolean $A \times A$-matrices defined inductively with*

$$\varrho_1 = \tau/\tau \quad \text{and} \quad \varrho_{k+1} = \varrho_k \odot \bigodot_{x\in X} (\delta_x \cdot \varrho_k)/(\delta_x \cdot \varrho_k), \quad \text{for every } k \in \mathbb{N}.$$

Then the sequence $\{\varrho_k\}_{k\in\mathbb{N}}$ is finite and descending, and it stabilizes for some $m \in \mathbb{N}$, i.e. there is the least $m \in \mathbb{N}$ such that $\varrho_m = \varrho_{m+1}$. In addition, ϱ_m is the greatest right invariant quasi-order matrix.

In a similar way we can compute the greatest left invariant quasi-order matrix, i.e. the only difference is that we compute the sequence $\{\varrho_k\}_{k\in\mathbb{N}} \subseteq 2^{A\times A}$ by employing the Boolean right residual of matrices (5) in the following way

$$\varrho_1 = \tau\backslash\tau \quad \text{and} \quad \varrho_{k+1} = \varrho_k \odot \bigodot_{x\in X} (\varrho_k \cdot \delta_x)\backslash(\varrho_k \cdot \delta_x), \quad \text{for every } k \in \mathbb{N},$$

as well as the greatest right and left invariant equivalence matrices, where the sequence $\{\varrho_k\}_{k\in\mathbb{N}} \subseteq 2^{A\times A}$ is build in the corresponding of the following ways

$$\varrho_1 = \tau|\tau \quad \text{and} \quad \varrho_{k+1} = \varrho_k \odot \bigodot_{x\in X} (\delta_x \cdot \varrho_k)|(\delta_x \cdot \varrho_k), \quad \text{for every } k \in \mathbb{N}, \text{ or}$$
$$\varrho_1 = \tau|\tau \quad \text{and} \quad \varrho_{k+1} = \varrho_k \odot \bigodot_{x\in X} (\varrho_k \cdot \delta_x)|(\varrho_k \cdot \delta_x), \quad \text{for every } k \in \mathbb{N}.$$

The corresponding algorithms for computing the greatest right or left invariant quasi-order or equivalence matrices can be easily derived from previous formulae. They all finish in a finite number of steps and in polynomial time, or more precisely, in $O(mn^5c_+)$ time complexity, where $|\mathcal{A}| = n$, $|X| = m$, and c_+ denotes the computation cost for performing the operation $+$ in the semiring S. (cf. [16]).

As noted in the introduction, the algorithm for computing the greatest right or left invariant equivalence matrix can be improved by using the well-exploited idea of partition refinement. Precisely, we employ the idea behind the algorithm by Paige and Tarjan [39] and, in contrast to the direct method given by Theorem 3.3, we maintain the additional partition alongside to the current partition such that some additional properties are satisfied. In what follows, we give improved algorithms for computing the greatest right or left invariant equivalence matrices.

Let $\mathcal{A} = (A, \delta, \sigma, \tau)$ be a WFA over an additively idempotent semiring S, and let $\varrho, \theta \in 2^{A\times A}$ be two equivalence matrices. Then we say that ϱ is *stable with respect to* θ if

$$\varrho \leqslant \bigodot_{x\in X} (\delta_x \cdot \theta^a) | (\delta_x \cdot \theta^a), \quad \text{for every } a \in A. \tag{16}$$

In case when $\mathcal{A}$ is a WFA over the Boolean semiring, the previous definition coincides with the one given for partitions on the set of states of a nondeterministic automaton (cf. [1, 9, 39]).

Lemma 3.4. *Let $\mathcal{A} = (A, \delta, \sigma, \tau)$ be a WFA over an additively idempotent semiring S and $\varrho \in 2^{A\times A}$ an equivalence matrix. Then ϱ is stable if and only if it is stable with respect to itself.*

Proof. The equivalence matrix $\varrho \in 2^{A\times A}$ is stable iff it is both r-stable and l-stable. By (7), $\varrho \in 2^{A\times A}$ is stable iff

$$\varrho \cdot \delta_x \cdot \varrho^a \leqslant \delta_x \cdot \varrho^a \quad \text{and} \quad \varrho^a \cdot \delta_x \cdot \varrho \leqslant \varrho^a \cdot \delta_x, \quad \text{for every } a \in A \text{ and } x \in X,$$

or equivalently,

$$\varrho \leqslant (\delta_x \cdot \varrho^a)/(\delta_x \cdot \varrho^a) \quad \text{and} \quad \varrho \leqslant (\varrho^a \cdot \delta_x)\backslash(\varrho^a \cdot \delta_x), \quad \text{for every } a \in A \text{ and } x \in X,$$

which was to be proved. □

Lemma 3.5. *Let $\mathcal{A} = (A, \delta, \sigma, \tau)$ be a WFA over an additively idempotent semiring, and let $\varrho, \theta \in 2^{A\times A}$ be two equivalence matrices. If $\varrho \leqslant \theta$ and ϱ is r-stable, then ϱ is stable w.r.t θ.*

Proof. Since ϱ satisfies (10), we have

$$\varrho \cdot \delta_x \cdot \varrho \cdot \theta \leqslant \delta_x \cdot \varrho \cdot \theta, \quad \text{for every } x \in X,$$

and according to Lemma 2.1 we further have

$$\varrho \cdot \delta_x \cdot \theta \leqslant \delta_x \cdot \theta, \quad \text{for every } x \in X.$$

By employing (7), we conclude that

$$\varrho \cdot \delta_x \cdot \theta^a \leqslant \delta_x \cdot \theta^a, \quad \text{for every } x \in X \text{ and every } a \in A,$$

which is equivalent to

$$\varrho \leqslant (\delta_x \cdot \theta^a)/(\delta_x \cdot \theta^a), \quad \text{for every } x \in X \text{ and every } a \in A.$$

Furthermore, by the symmetry of ϱ we get

$$\varrho = \varrho^{\mathrm{T}} \leqslant ((\delta_x \cdot \theta^a)/(\delta_x \cdot \theta^a))^{\mathrm{T}} = (\delta_x \cdot \theta^a)\backslash(\delta_x \cdot \theta^a), \quad \text{for every } x \in X \text{ and every } a \in A.$$

Hence, $\varrho \leqslant (\delta_x \cdot \theta^a)|(\delta_x \cdot \theta^a)$, for every $x \in X$ and $a \in A$, thus (16) follows. □

Theorem 3.6. *Let $\mathcal{A} = (A, \delta, \sigma, \tau)$ be a WFA over an additively idempotent semiring, and let $\{\theta_k\}_{k\in\mathbb{N}}$ and $\{\varrho_k\}_{k\in\mathbb{N}}$ be two sequences of Boolean $A \times A$-matrices defined inductively in the following way: For $k = 1$, set*

$$\theta_1 = U_A, \tag{17}$$

$$\varrho_1 = \tau|\tau \odot \bigodot_{x\in X} (\delta_x \cdot \theta_1^a)|(\delta_x \cdot \theta_1^a), \quad \textit{for some } a \in A. \tag{18}$$

Assume that we have computed θ_k and ϱ_k in the current step k. In the next step, if $\theta_k \neq \varrho_k$, then select some $a \in A$ such that $\theta_k^a \neq \varrho_k^a$, and compute θ_{k+1} and ϱ_{k+1} in the following manner

$$\theta_{k+1} = \theta_k \odot (\varrho_k^a | \varrho_k^a), \tag{19}$$

$$\varrho_{k+1} = \varrho_k \odot \bigodot_{x\in X}\Big((\delta_x \cdot \varrho_k^a)|(\delta_x \cdot \varrho_k^a) \odot (\delta_x \cdot (\theta_k^a - \varrho_k^a))|(\delta_x \cdot (\theta_k^a - \varrho_k^a))\Big). \tag{20}$$

In other case, when $\theta_k = \varrho_k$, set $\theta_k = \theta_{k+1}$ and $\varrho_k = \varrho_{k+1}$. Then the following properties hold:

a) Sequences $\{\theta_k\}_{k\in\mathbb{N}}$ and $\{\varrho_k\}_{k\in\mathbb{N}}$ are descending;
b) ϱ_k and θ_k are equivalence matrices, for every $k \in \mathbb{N}$;
c) ϱ_k is a refinement of θ_k (i.e. $\varrho_k \leqslant \theta_k$), for every $k \in \mathbb{N}$;
d) ϱ_k is stable w.r.t. θ_k, for every $k \in \mathbb{N}$;
e) There exists $s \in \mathbb{N}$ such that $\theta_s = \varrho_s$, and ϱ_s is the greatest right invariant equivalence matrix on A.

Proof. a) Since the Hadamard product $\bigodot$ is applied on Boolean matrices, as well as that 0 is the least element in the additively idempotent semiring, by (19) it follows that $\theta_{k+1} \leqslant \theta_k$, and by (20) we have that $\varrho_{k+1} \leqslant \varrho_k$.

b) Follow directly from definitions of θ_k and ϱ_k.

c) We prove that $\varrho_k \leqslant \theta_k$ holds for every $k \in \mathbb{N}$ by induction on k. For $k = 1$ we have $\varrho_1 \leqslant U_A = \theta_1$. Assume now that $\varrho_m \leqslant \theta_m$ for some $k = m$. According to part a), we have that $\varrho_{m+1} \leqslant \varrho_m$. On the other hand, according to Lemma 2.2 we have $\varrho_m \leqslant \varrho_m^c|\varrho_m^c$ for every $c \in A$. Combining this with the induction assumption $\varrho_m \leqslant \theta_m$, as well that θ_m and $\varrho_m^c|\varrho_m^c$ are Boolean matrices, for every $c \in A$, we obtain:

$$\varrho_m \leqslant \theta_m \odot (\varrho_m^a|\varrho_m^a) = \theta_{m+1}.$$

Hence, $\varrho_{m+1} \leqslant \varrho_m \leqslant \theta_{m+1}$, which was to be proven.

d) We prove that

$$\varrho_k \leqslant \bigodot_{x\in X}(\delta_x \cdot \theta_k^a)|(\delta_x \cdot \theta_k^a), \quad \text{for every } a \in A, \tag{21}$$

holds for every $k \in \mathbb{N}$ by induction on k. We can conclude directly from (18) that (21) is valid for $k = 1$.

In case when $k = 2$ we have $\theta_2 = \theta_1 \odot (\varrho_1^a|\varrho_1^a)$ for some $a \in A$. Since θ_1 has exactly one equivalence class $\theta_1^a = 1^A$, this means that θ_2 is obtained from θ_1 by splitting the equivalence class θ_1^a into two equivalence classes ϱ_1^a and $\theta_1^a - \varrho_1^a$. According to (20) we have

$$\varrho_2 = \varrho_1 \odot \bigodot_{x\in X}\Big((\delta_x \cdot \varrho_1^a)|(\delta_x \cdot \varrho_1^a) \odot (\delta_x \cdot (\theta_1^a - \varrho_1^a))|(\delta_x \cdot (\theta_1^a - \varrho_1^a))\Big),$$

which means that

$$\varrho_2 \leqslant \bigodot_{x\in X}(\delta_x \cdot \varrho_1^a)|(\delta_x \cdot \varrho_1^a)$$

and

$$\varrho_2 \leqslant \bigodot_{x\in X}(\delta_x \cdot (\theta_1^a - \varrho_1^a))|(\delta_x \cdot (\theta_1^a - \varrho_1^a)),$$

i.e., (21) is valid for both equivalence classes of θ_2, thus (21) follows in case when $k = 2$.

Assume now that (21) is valid for some $k = m$. Consider the equivalence matrix $\theta_{m+1} = \theta_m \odot (\varrho_m^a|\varrho_m^a)$ for some $a \in A$ such that $\theta_m^a \neq \varrho_m^a$. From c) it follows that $\varrho_m^a < \theta_m^a$, which means that θ_{m+1} is obtained from θ_m by splitting the equivalence class θ_m^a into two equivalence classes ϱ_m^a and $\theta_m^a - \varrho_m^a$. Thus, for every equivalence class $\theta^b_{m+1} \in \theta_{m+1}$, $b \in A$, such that $\theta^b_{m+1} \neq \varrho^a_m$ and $\theta^b_{m+1} \neq \theta^a_m - \varrho^a_m$, according to a) and the induction assumption we have

$$\varrho_{m+1} \leqslant \varrho_m \leqslant \bigodot_{x\in X} (\delta_x \cdot \theta^b_m)|(\delta_x \cdot \theta^b_m) = \bigodot_{x\in X} (\delta_x \cdot \theta^b_{m+1})|(\delta_x \cdot \theta^b_{m+1}).$$

In other words, (21) holds for every $b \in A$ such that $\theta^b_{m+1} \neq \varrho^a_m$ and $\theta^b_{m+1} \neq \theta^a_m - \varrho^a_m$. On the other hand, directly from (20) we have

$$\varrho_{m+1} = \varrho_m \odot \bigodot_{x\in X} \Big((\delta_x \cdot \varrho^a_m)|(\delta_x \cdot \varrho^a_m) \odot (\delta_x \cdot (\theta^a_m - \varrho^a_m))|(\delta_x \cdot (\theta^a_m - \varrho^a_m))\Big)$$

which means that

$$\varrho_{m+1} \leqslant \bigodot_{x\in X} (\delta_x \cdot \varrho^a_m)|(\delta_x \cdot \varrho^a_m)$$

and

$$\varrho_{m+1} \leqslant \bigodot_{x\in X} (\delta_x \cdot (\theta^a_m - \varrho^a_m))|(\delta_x \cdot (\theta^a_m - \varrho^a_m)),$$

i.e., (21) holds in the case $k = m + 1$, which was to be proven.

e) Since A is a finite set, we have that $2^{A\times A}$ is finite, thus both sequences $\{\theta_k\}_{k\in\mathbb{N}}$ and $\{\varrho_k\}_{k\in\mathbb{N}}$ stabilize. But, if for some $s \in \mathbb{N}$ we have $\theta_s = \theta_{s+1}$, by the construction of these two sequences it follows that $\theta_s = \varrho_s$.

In fact, if $\theta_s = \theta_{s+1}$ for some $s \in \mathbb{N}$, we have that every matrix θ_{m+1}, with $m < s$, is obtained from θ_m by splitting some equivalence class of θ_m into two classes. Since θ_1 has exactly one equivalence class, we have that θ_s can have at most $|A|$ equivalence classes. In other words, it follows that $s \leqslant |A|$.

It remains to prove that ϱ_s is the greatest right invariant equivalence matrix. Since by d) ϱ_s is stable w.r.t. θ_s, and $\theta_s = \varrho_s$, we obtain that ϱ_s is stable, i.e.

$$\varrho_s \leqslant \bigodot_{x\in X} (\delta_x \cdot \varrho^c_s)|(\delta_x \cdot \varrho^c_s),$$

for every $c \in A$, which means that ϱ_s satisfies (10). Also, by a) we have $\varrho_s \leqslant \varrho_1 \leqslant \tau|\tau$, therefore, ϱ_s also satisfies (11) and ϱ_s is a right invariant equivalence matrix. To prove that ϱ_s is also the greatest one, we prove that $\omega \leqslant \varrho_k$ for every right invariant equivalence matrix $\omega \in 2^{A\times A}$ by induction on k.

For $k = 1$, we have that $\omega \leqslant U_A = \theta_1$, and since ω is r-stable, by Lemma 3.5 we have that ω is stable w.r.t. θ. In addition, ω satisfies (11), which means that

$$\omega \leqslant \tau/\tau \quad \text{and} \quad \omega = \omega^{\mathrm{T}} \leqslant (\tau/\tau)^{\mathrm{T}} = \tau\backslash\tau,$$

therefore, $\omega \leqslant \tau|\tau$, thus we have

$$\omega \leqslant \tau|\tau \odot \bigodot_{x\in X} (\delta_x \cdot \theta^a_1)|(\delta_x \cdot \theta^a_1) = \varrho_1.$$

Assume now that $\omega \leqslant \varrho_m$ for some $k = m$. By b) we have that $\omega \leqslant \theta_m$, and by Lemma 2.2 we also have that $\omega \leqslant \varrho^a_m|\varrho^a_m$, where $a \in A$ such that $\varrho^a_m \neq \theta^a_m$. Therefore,

$$\omega \leqslant \theta_m \odot (\varrho^a_m|\varrho^a_m) = \theta_{m+1}.$$

Since ω is r-stabile, from Lemma 3.5 it follows that ω is stable w.r.t. θ_{m+1}, i.e. (21) holds for all equivalence classes of θ_{m+1}, and particularly, for equivalence classes ϱ^a_m and $\theta^a_m - \varrho^a_m$. That means that

$$\omega \leqslant \varrho_m \odot \bigodot_{x\in X} \Big((\delta_x \cdot \varrho^a_m)|(\delta_x \cdot \varrho^a_m) \odot (\delta_x \cdot (\theta^a_m - \varrho^a_m))|(\delta_x \cdot (\theta^a_m - \varrho^a_m))\Big) = \varrho_{m+1},$$

which was to be proven. $\square$

Algorithm 1 Computation of the greatest right invariant equivalence matrix

```
Input: Weighted automaton $\mathcal{A} = (A, \delta, \sigma, \tau)$ over alphabet $X$ and additively idempotent semiring $S$
Output: The greatest right invariant equivalence matrix
1: $\theta \leftarrow U_A$
2: $\varrho \leftarrow \tau|\tau \odot \bigodot_{x\in X}(\delta_x \cdot \theta^a)|(\delta_x \cdot \theta^a)$,     for some $a \in A$
3: while $\theta \neq \varrho$ do
4:     Choose $a \in A$ such that $\theta^a \neq \varrho^a$
5:     $\theta_1 \leftarrow \theta \odot (\varrho^a|\varrho^a)$
6:     $\varrho_1 \leftarrow \varrho \odot \bigodot_{x\in X}\big((\delta_x \cdot \varrho^a)|(\delta_x \cdot \varrho^a) \odot (\delta_x \cdot (\theta^a - \varrho^a))|(\delta_x \cdot (\theta^a - \varrho^a))\big)$
7:     $\theta \leftarrow \theta_1$,   $\varrho \leftarrow \varrho_1$
8: end while
9: return $\varrho$
```

The previous theorem can be transformed into the algorithm for computing the greatest right invariant equivalence matrix on a WFA over the additively idempotent semiring.

According to Theorem 3.6, sequences $\{\theta_k\}_{k\in\mathbb{N}}$ and $\{\varrho_k\}_{k\in\mathbb{N}}$ are descending and stabilizing, therefore Algorithm 1 always terminates in a finite number of steps, regardless of the underlying additively idempotent semiring.

We now analyse the running time of Algorithm 1. To that effort, as before, denote with $|\mathcal{A}| = n$ and $|X| = m$. Also, denote with c_+ and $c_\cdot$, respectively, the computation costs for performing operations $+$ and $\cdot$ in S. For example, when S is the Boolean, tropical or Viterbi semiring, then we have $c_+ = c_\cdot = 1$.

We analyse only those steps of Algorithm 1 that consume more than $O(1)$ computation time. Step 1 takes $O(n^2)$ time, while Step 2 requires computing the Hadamard product of $m + 1$ Boolean matrices, where each Boolean matrix is computed by means of formula (6). Since the elements of the Boolean matrix can only be zero or one from the semiring, the multiplication can be done in the constant time, thus the computation of every A-vector $\delta_x \cdot \theta^a$ takes $O(n^2 c_+)$ time. The computation of the matrix defined by formula (6) requires n^2 equality checking, thus taking $O(n^2)$ time. Therefore, Step 2 is performed in $O(mn^2 c_+)$ time.

In Step 4 we have, in the worst case, to iterate through all classes of equivalence matrices θ and ϱ, and they can have at most n equivalence classes. Checking if two equivalence classes are equal takes $O(n)$ time, thus Step 4 takes $O(n^2)$ time. Similarly as in the analysis of Step 2, we conclude that Step 5 is performed in $O(n^2)$ time, and Step 6 in $O(mn^2 c_+)$ time. Step 7 obviously requires $O(n^2)$ time.

According to Theorem 3.6 e), the total number of repetition of Steps 4-7 is at most $n - 1$. Therefore, the total computation time of Algorithm 1 is $O(mn^3 c_+)$. In conclusion, it runs in polynomial time and outperforms the direct method given by Theorem 3.3.

The algorithm for computing the greatest left invariant equivalence matrix can be derived analogously.

Algorithm 2 Computation of the greatest left invariant equivalence matrix

```
Input: Weighted automaton $\mathcal{A} = (A, \delta, \sigma, \tau)$ over alphabet $X$ and additively idempotent semiring $S$
Output: The greatest left invariant equivalence matrix
1: $\theta \leftarrow U_A$
2: $\varrho \leftarrow \tau|\tau \odot \bigodot_{x\in X}(\theta^a \cdot \delta_x)|(\theta^a \cdot \delta_x)$,     for some $a \in A$
3: while $\theta \neq \varrho$ do
4:     Choose $a \in A$ such that $\theta^a \neq \varrho^a$
5:     $\theta_1 \leftarrow \theta \odot (\varrho^a|\varrho^a)$
6:     $\varrho_1 \leftarrow \varrho \odot \bigodot_{x\in X}\big((\varrho^a \cdot \delta_x)|(\varrho^a \cdot \delta_x) \odot ((\theta^a - \varrho^a) \cdot \delta_x)|((\theta^a - \varrho^a) \cdot \delta_x)\big)$
7:     $\theta \leftarrow \theta_1$,   $\varrho \leftarrow \varrho_1$
8: end while
9: return $\varrho$
```

The following example illustrates the work of previous algorithms.

Example 3.7. *Let $\mathcal{A} = (A, \delta, \sigma, \tau)$ be a WFA over the tropical semiring with $A = \{a_1, a_2, a_3, a_4, a_5\}$ and $X = \{x\}$ represented by the transition graph given by Figure 1. We have that the initial weighted vector, the final weighted vector and the weighted transition matrix are given by*

$$\sigma = \begin{bmatrix} 0 & \infty & \infty & \infty & \infty \end{bmatrix}, \quad \tau = \begin{bmatrix} \infty \\ \infty \\ 0 \\ \infty \\ 0 \end{bmatrix}, \quad d_x = \begin{bmatrix} \infty & 0 & \infty & 0 & \infty \\ 0 & 0 & \infty & 0 & \infty \\ 0 & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & 0 & 0 \\ 0 & 0 & \infty & \infty & \infty \end{bmatrix}.$$

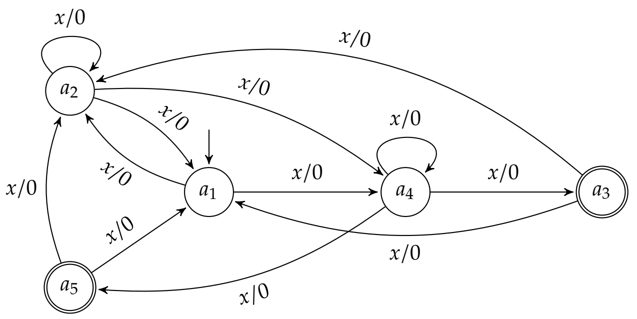

Figure 1: The transition graph of the WFA from Example 3.7

In the first step, Algorithm 1 produces matrices

$$\theta_1 = U_A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \ \varrho_1 = \tau|\tau \odot (\delta_x \cdot \theta_1^a)|(\delta_x \cdot \theta_1^a) = \begin{bmatrix} 0 & 0 & \infty & 0 & \infty \\ 0 & 0 & \infty & 0 & \infty \\ \infty & \infty & 0 & \infty & 0 \\ 0 & 0 & \infty & 0 & \infty \\ \infty & \infty & 0 & \infty & 0 \end{bmatrix}.$$

Next, for $a_1 \in A$ we have that $\varrho_1^{a_1} \neq \theta_1^{a_1}$, thus we choose $a_1 \in A$ and obtain

$$\theta_2 = \theta_1 \odot (\varrho_1^{a_1}|\varrho_1^{a_1}) = \begin{bmatrix} 0 & 0 & \infty & 0 & \infty \\ 0 & 0 & \infty & 0 & \infty \\ \infty & \infty & 0 & \infty & 0 \\ 0 & 0 & \infty & 0 & \infty \\ \infty & \infty & 0 & \infty & 0 \end{bmatrix},$$

$$\varrho_2 = \varrho_1 \odot (\delta_x \cdot \varrho_1^{a_1})|(\delta_x \cdot \varrho_1^{a_1}) \odot (\delta_x \cdot (\theta_1^{a_1} - \varrho_1^{a_1}))|(\delta_x \cdot (\theta_1^{a_1} - \varrho_1^{a_1})) = \begin{bmatrix} 0 & 0 & \infty & \infty & \infty \\ 0 & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 0 \\ \infty & \infty & \infty & 0 & \infty \\ \infty & \infty & 0 & \infty & 0 \end{bmatrix}.$$

Again, choose $a_1 \in A$ since $\varrho_2^{a_1} \neq \theta_2^{a_1}$, and obtain

$$\theta_3 = \theta_2 \odot (\varrho_2^{a_1} | \varrho_2^{a_1}) = \begin{bmatrix} 0 & 0 & \infty & \infty & \infty \\ 0 & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 0 \\ \infty & \infty & \infty & 0 & \infty \\ \infty & \infty & 0 & \infty & 0 \end{bmatrix},$$

$$\varrho_3 = \varrho_2 \odot (\delta_x \cdot \varrho_2^{a_1}) | (\delta_x \cdot \varrho_2^{a_1}) \odot (\delta_x \cdot (\theta_2^{a_1} - \varrho_2^{a_1})) | (\delta_x \cdot (\theta_2^{a_1} - \varrho_2^{a_1})) = \begin{bmatrix} 0 & 0 & \infty & \infty & \infty \\ 0 & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 0 \\ \infty & \infty & \infty & 0 & \infty \\ \infty & \infty & 0 & \infty & 0 \end{bmatrix}.$$

Since $\theta_3 = \varrho_3$, we obtain that the greatest right invariant equivalence matrix is equal to $\theta_3 = \varrho_3$.

The algorithm for computing the greatest right (resp. left) invariant equivalence matrix can be generalized to compute the greatest right (resp. left) invariant quasi-order matrix for a given weighted automaton over an additively idempotent semiring.

Let $\mathcal{A} = (A, \delta, \sigma, \tau)$ be a weighted automaton over an additively idempotent semiring S, and let $\varrho, \theta \in 2^{A \times A}$ be two quasi-order matrices. Then we say that ϱ is *r-stable with respect to θ* if

$$\varrho \leqslant \bigodot_{x \in X} (\delta_x \cdot \theta a)/(\delta_x \cdot \theta a), \qquad \text{for every } a \in A,$$

and dually, ϱ is *l-stable with respect to θ* if

$$\varrho \leqslant \bigodot_{x \in X} (a\theta \cdot \delta_x) \backslash (a\theta \cdot \delta_x), \qquad \text{for every } a \in A.$$

The following two Lemmas are generalizations of Lemmas 3.4 and 3.5.

Lemma 3.8. *Let $\mathcal{A} = (A, \delta, \sigma, \tau)$ be a WFA over an additively idempotent semiring, and let $\varrho \in 2^{A \times A}$ be a quasi-order matrix. Then ϱ is r-stable (resp. l-stable) iff it is r-stable (resp. l-stable) with respect to itself.*

Lemma 3.9. *Let $\mathcal{A} = (A, \delta, \sigma, \tau)$ be a WFA over an additively idempotent semiring, and let $\varrho, \theta \in 2^{A \times A}$ be two quasi-order matrices. If $\varrho \leqslant \theta$ and ϱ is r-stable (resp. l-stable), then ϱ is r-stable (resp. l-stable) w.r.t θ.*

The following Theorems formalize the way to compute the greatest right invariant matrix.

Theorem 3.10. *Let $\mathcal{A} = (A, \delta, \sigma, \tau)$ be a WFA over an additively idempotent semiring, and let $\{\theta_k\}_{k \in \mathbb{N}}$ and $\{\varrho_k\}_{k \in \mathbb{N}}$ be two sequences of Boolean $A \times A$-matrices defined inductively in the following way: For $k = 1$, set*

$$\theta_1 = U_A,$$
$$\varrho_1 = \tau/\tau \odot \bigodot_{x \in X} (\delta_x \cdot \theta_1 a)/(\delta_x \cdot \theta_1 a), \qquad \textit{for some } a \in A.$$

Assume that we have computed θ_k and ϱ_k in the current step k. In the next step, if $\varrho_k \neq \theta_k$, select some $a \in A$ such that $\theta_k a \neq \varrho_k a$, and compute the set $B = \{b \in A | \varrho_k a(b) \neq 0\}$. Then compute θ_{k+1} and ϱ_{k+1} in the following way

$$\theta_{k+1} = \theta_k \odot (\varrho_k a / \varrho_k a),$$
$$\varrho_{k+1} = \varrho_k \odot \bigodot_{x \in X} \bigodot_{b \in B} (\delta_x \cdot \theta_{k+1} b)/(\delta_x \cdot \theta_{k+1} b).$$

Otherwise, set $\theta_{k+1} = \theta_k$ and $\varrho_{k+1} = \varrho_k$. Then the following holds:

a) Sequences $\{\theta_k\}_{k\in\mathbb{N}}$ and $\{\varrho_k\}_{k\in\mathbb{N}}$ are descending;
b) ϱ_k and θ_k are quasi-order matrices, for every $k \in \mathbb{N}$;
c) ϱ_k is a refinement of θ_k, for every $k \in \mathbb{N}$;
d) ϱ_k is r-stable w.r.t. θ_k, for every $k \in \mathbb{N}$;
e) There exists $s \in \mathbb{N}$ such that $\theta_s = \varrho_s$, and ϱ_s is the greatest right invariant quasi-order matrix on A.

Proof. Parts a), b), c) and e) can be proved in an analogue manner as in Theorem 3.6 by using Lemmas 3.8 and 3.9.

d) We prove that

$$\varrho_k \leqslant \bigodot_{x\in X}(\delta_x \cdot \theta_k a)/(\delta_x \cdot \theta_k a), \qquad \text{for every } a \in A, \tag{22}$$

holds for every $k \in \mathbb{N}$ by induction on k. In the case $k = 1$, (22) follows directly from the definition of ϱ_1 since A/θ_1 contain only one foreset $\theta_1^a = 1^A$. Assume that (22) is valid for some $k = m$ and prove that it is also valid for $m + 1$. According to the definition of ϱ_{m+1} we have that

$$\varrho_k \leqslant \bigodot_{x\in X}(\delta_x \cdot \theta_k c)/(\delta_x \cdot \theta_k c), \qquad \text{for every } c \in B. \tag{23}$$

We now prove that $\theta_m c = \theta_{m+1} c$ for every $c \in A - B$. If $c \in A - B$ then $\varrho_m a(c) = 0$ and therefore, for every $d \in A$ we have $\varrho_m a(c) \leqslant \varrho_m a(d)$ and

$$\theta_{m+1}c(d) = \theta_{m+1}(d,c) = \theta_m(d,c)\cdot(\varrho_m a/\varrho_m a)(d,c) = \theta_m(d,c)\cdot 1 = \theta_m(d,c) = \theta_m c(d).$$

The previous is valid for every $d \in A$, thus by the induction hypothesis we have

$$\varrho_{m+1} \leqslant \varrho_m \leqslant \bigodot_{x\in X}(\delta_x \cdot \theta_m c)/(\delta_x \cdot \theta_m c) = \bigodot_{x\in X}(\delta_x \cdot \theta_{m+1} c)/(\delta_x \cdot \theta_{m+1} c), \qquad \text{for every } c \in A - B. \tag{24}$$

By combining (23) and (24), (22) follows in the case $k = m + 1$, and the proof is completed. □

The previous theorem can be transformed into the algorithm for computing the greatest right invariant quasi-order matrix on a WFA over an additively idempotent semiring.

Algorithm 3 Computation of the greatest right invariant quasi-order matrix

Input: Weighted automaton $\mathcal{A} = (A, \delta, \sigma, \tau)$ over alphabet X and additively idempotent semiring S
Output: The greatest right invariant quasi-order matrix

```
 1: $\theta \leftarrow U_A$
 2: $\varrho \leftarrow \tau/\tau \odot \bigodot_{x\in X}(\delta_x \cdot \theta a)/(\delta_x \cdot \theta a)$,    for some $a \in A$
 3: while $\theta \neq \varrho$ do
 4:     Choose $a \in A$ such that $\theta a \neq \varrho a$
 5:     Compute $B = \{b \in A | \varrho a(b) \neq 0\}$
 6:     $\theta_1 \leftarrow \theta \odot (\varrho a/\varrho a)$
 7:     $\varrho_1 \leftarrow \varrho \odot \bigodot_{x\in X} \bigodot_{b\in B}(\delta_x \cdot \theta_1 b)/(\delta_x \cdot \theta_1 b)$
 8:     $\theta \leftarrow \theta_1$,   $\varrho \leftarrow \varrho_1$
 9: end while
10: return $\varrho$
```

Similarly as in the case of the greatest right invariant equivalence matrix, Algorithm 3 always terminates in a finite number of steps, regardless of the underlying additively idempotent semiring.

We now analyse the running time of Algorithm 3. Step 1 requires $O(n^2)$ time. Next, in order to compute the left residual of two vectors of dimension n, we need to perform n^2 inequality checks, and since the ordering in S is defined by (1), the computation time of the single inequality check is equal to the cost of the addition c_+. Thus, computation of the right residual is done in $O(n^2c_+)$ time. By applying the rest of the formula, we conclude that Step 2 takes $O(mn^2c_+)$ time.

Step 4 takes $O(n^2)$ time, Step 5 $O(n)$ time, and Step 6 $O(n^2c_+)$ time. In Step 7, we compute the product $\delta_x \cdot \theta b$ in $O(n^2c_+)$ time, and the number of right residuals that need to be calculated is $m \cdot n$. Thus, Step 7 takes $O(mn^3c_+)$ time.

In contrast to the case of equivalence matrices, the total number of repetitions of Steps 3-9 can be n^2, making the total time complexity of Algorithm 3 proportional to $O(mn^5c_+)$. However, the theoretical upper bound n^2 is reached only in extreme cases, which means that Algorithm 3 performs faster than the direct method given by Theorem 3.3 in majority of practical examples.

The algorithm for computing the greatest left invariant quasi-order matrix can be derived analogously.

Algorithm 4 Computation of the greatest left invariant quasi-order matrix

```
Input: Weighted automaton $\mathcal{A} = (A, \delta, \sigma, \tau)$ over alphabet $X$ and additively idempotent semiring $S$
Output: The greatest left invariant quasi-order matrix
 1: $\theta \leftarrow U_A$
 2: $\varrho \leftarrow \tau\backslash\tau \odot \bigodot_{x\in X}(a\theta \cdot \delta_x)\backslash(a\theta \cdot \delta_x)$,     for some $a \in A$
 3: while $\theta \neq \varrho$ do
 4:     Choose $a \in A$ such that $a\theta \neq a\varrho$
 5:     Compute $B = \{b \in A | a\varrho(b) \neq 0\}$
 6:     $\theta_1 \leftarrow \theta \odot (a\varrho\backslash a\varrho)$
 7:     $\varrho_1 \leftarrow \varrho \odot \bigodot_{x\in X} \bigodot_{b\in B}(b\theta_1 \cdot \delta_x)\backslash(b\theta_1 \cdot \delta_x)$
 8:     $\theta \leftarrow \theta_1$,   $\varrho \leftarrow \varrho_1$
 9: end while
10: return $\varrho$
```

Example 3.11. *Let $\mathcal{A} = (A, \delta, \sigma, \tau)$ be the WFA from Example 3.7. We aim to compute the greatest right invariant quasi-order matrix by means of Algorithm 3. In the first step we calculate matrices*

$$\theta_1 = U_A = \begin{bmatrix} 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0 \end{bmatrix}, \ \varrho_1 = \tau/\tau \odot (\delta_x \cdot \theta_1 a)/(\delta_x \cdot \theta_1 a) = \begin{bmatrix} 0&0&\infty&0&\infty\\ 0&0&\infty&0&\infty\\ 0&0&0&0&0\\ 0&0&\infty&0&\infty\\ 0&0&0&0&0 \end{bmatrix}.$$

In the next step we choose, for example, $a_3 \in A$, because $\theta_1 a_3 \neq \varrho_1 a_3$, and compute the set $B = \{b \in A | \varrho_1 a_3(b) \neq 0\} = \{a_3, a_5\}$. Then we have

$$\theta_2 = \theta_1 \odot (\varrho_1 a_3/\varrho_1 a_3) = \begin{bmatrix} 0&0&\infty&0&\infty\\ 0&0&\infty&0&\infty\\ 0&0&0&0&0\\ 0&0&\infty&0&\infty\\ 0&0&0&0&0 \end{bmatrix}.$$

Since $\theta_2 a_3 = \theta_2 a_5$, we further have

$$\varrho_2 = \varrho_1 \odot (\delta_x \cdot \theta_2 a_3)/(\delta_x \cdot \theta_2 a_3) = \begin{bmatrix} 0&0&\infty&\infty&\infty\\ 0&0&\infty&\infty&\infty\\ 0&0&0&\infty&0\\ 0&0&\infty&0&\infty\\ 0&0&0&\infty&0 \end{bmatrix}.$$

In the next step, since for $a_4 \in A$ we have $\theta_1 a_4 \neq \varrho_1 a_4$, we choose $a_4 \in A$, and compute the set $B = \{b \in A | \varrho_2 a_4(b) \neq 0\} = \{a_4\}$. Then we have

$$\theta_3 = \theta_2 \odot (\varrho_2 a_4 / \varrho_2 a_4) = \begin{bmatrix} 0 & 0 & \infty & \infty & \infty \\ 0 & 0 & \infty & \infty & \infty \\ 0 & 0 & 0 & \infty & 0 \\ 0 & 0 & \infty & 0 & \infty \\ 0 & 0 & 0 & \infty & 0 \end{bmatrix}, \ \varrho_3 = \varrho_2 \odot (\delta_x \cdot \theta_3 a_4)/(\delta_x \cdot \theta_3 a_4) = \begin{bmatrix} 0 & 0 & \infty & \infty & \infty \\ 0 & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 0 \\ 0 & 0 & \infty & 0 & \infty \\ \infty & \infty & 0 & \infty & 0 \end{bmatrix}.$$

In the end, we choose $a_1 \in A$ and obtain

$$\theta_4 = \theta_3 \odot (\varrho_3 a_1 / \varrho_3 a_1) = \begin{bmatrix} 0 & 0 & \infty & \infty & \infty \\ 0 & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 0 \\ 0 & 0 & \infty & 0 & \infty \\ \infty & \infty & 0 & \infty & 0 \end{bmatrix}, \ \varrho_4 = \varrho_3 \odot (\delta_x \cdot \theta_4 a_1)/(\delta_x \cdot \theta_4 a_1) = \begin{bmatrix} 0 & 0 & \infty & \infty & \infty \\ 0 & 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 0 \\ 0 & 0 & \infty & 0 & \infty \\ \infty & \infty & 0 & \infty & 0 \end{bmatrix}.$$

The algorithms stops and we obtain that $\theta_4 = \varrho_4$ is the greatest right invariant quasi-order matrix for $\mathcal{A}$.

We now turn our attention to the computation of the greatest weakly right and weakly left invariant matrix. According to the corresponding definitions, we first need to compute all A-column vectors τ_u from the collection $\{\tau_u\}_{u \in X^*}$. The procedure for computation of the family $\{\tau_u\}_{u \in X^*}$ can be derived in a similar way as in [31, 43], and is given by Algorithm 5. We use the well-known data structures and operations on them. In particular, we use a queue equipped with the standard operations:

- `Enqueue`(q, v), which adds an item v onto the end of the queue q,
- `Dequeue`(q), which removes the item from the front of the queue q and returns it as a result,
- `IsEmpty`(q), which returns true if no more items can be dequeued and there is no front item in the queue q, and false otherwise.

All operations can be performed in $O(1)$ time. In addition, we use a tree as a data structure together with operations

- `Lookup`(T, v), which returns true if there is a node with the value v in the tree T, and false otherwise,
- `Insert`(T, n, v), which adds a new node with the value n to the tree T with an edge from the existing node with value v,

We also use the notation `Insert`(T, n) = `Insert`$(T, n, null, null)$ for adding a node with value n as a root of an empty tree T. If we denote with t the number of nodes of the tree T, and with c the computation cost for checking whether values of two nodes from the tree T are equal, then the `Lookup`(T, v) operation requires going through, in the worst case, all nodes from the tree T (which requires $O(t)$ time), and for each node checking whether its value is equal to v (which is done in $O(c)$ time). Thus, the total running time of `Lookup` operation is $O(tc)$. Similarly, we conclude that `Insert` operation also run in $O(tc)$ time.

Unfortunately, this procedure does not necessarily terminate in a finite number of steps, since the collection $\{\tau_u\}_{u \in X^*}$ may be infinite. In cases when the mentioned collection is finite, then the procedure terminates in a finite number of steps, after computing all its members. For example, this holds when the subsemiring $S(\delta, \tau)$ of S generated by all values taken by weighted transition matrices $\{\delta_x\}_{x \in X}$ and the final weighted column vector τ is finite (but not only in this case). Denote with k the number of elements of the subsemiring $S(\delta, \tau)$. Then the collection $\{\tau_u\}_{u \in X^*}$ can have at most k^n different members. Recall that $|\mathcal{A}| = n$ and $|X| = m$.

We now analyse the running time of Algorithm 5. Step 8 requires computing the composition of the $A \times A$-matrix δ_x and the A-column vector τ_u, which is done in $O(n^2(c_+ + c_\cdot))$ time. Since the transition tree can have at most k^n vertices, and checking whether two column vectors are equal takes $O(n)$ time, we conclude that Steps 9 and 12 each take $O(nk^n)$ time, i.e. they take exponential time. Thus, Steps 8-12 take $O(nk^n)$ time

Algorithm 5 Computation of all members of the family $\{\tau_u\}_{u\in X^*}$

Input: Weighted automaton $\mathscr{A} = (A, \delta, \sigma, \tau)$ over alphabet X and additively idempotent semiring S
Output: Family of A-column vectors $\{\tau_u\}_{u\in X^*}$

```
 1: Initialize an empty queue $q$ of $A$-column vectors and an empty tree $T$ whose nodes are also $A$-column
    vectors
 2: Calculate $\tau_\varepsilon = \tau$
 3: Insert($T, \tau_\varepsilon$)
 4: Enqueue($q, \tau_\varepsilon$)
 5: while not IsEmpty($q$) do
 6:     $\tau_u \leftarrow$ Dequeue($q$)
 7:     for all $x \in X$ do
 8:         Calculate $\tau_{x\cdot u} = \delta_x \cdot \tau_u$
 9:         if not Lookup($T, \tau_{x\cdot u}$) then
10:             Enqueue($q, \tau_{x\cdot u}$)
11:         end if
12:         Insert($T, \tau_{x\cdot u}, \tau_u$)
13:     end for
14: end while
15: return Internal vertices of the tree $T$ (all different members of the family $\{\tau_u\}_{u\in X^*}$)
```

to compute. Since $|X| = m$, we conclude that the loop forming Steps 7-13 takes $O(mnk^n)$ time. In the end, by the facts that the elements of the queue are the elements of the collection $\{\tau_u\}_{u\in X^*}$, and that by Step 9 every element from that collection is exactly once present in the queue in some point in time, we conclude that Steps 5-14 are performed in $O(mnk^{2n})$ time. Since Step 15 does not exceed that time, we conclude that the total computation time for Algorithm 5 is equal to $O(mnk^{2n})$.

Therefore, Algorithm 5 is exponential in the number of states of $\mathscr{A}$ from a theoretical point of view. This is due to the fact that the number of members of the collection $\{\tau_u\}_{u\in X^*}$ may grow exponentially. However, this number is usually much smaller than its theoretical upper bound k^n, therefore, making Algorithm 5 of practical use.

We now provide the algorithm for computing the greatest weakly right invariant quasi-order matrix. Other types of weakly invariant matrices can be computed in a similar way.

Algorithm 6 Computation of the greatest weakly right invariant quasi-order matrix

Input: Weighted automaton $\mathscr{A} = (A, \delta, \sigma, \tau)$ over alphabet X and additively idempotent semiring S
Output: The greatest weakly right invariant quasi-order matrix

```
1: Compute the family of $A$-column vectors $\{\tau_u\}_{u\in X^*}$ by means of Algorithm 5
2: $\varrho \leftarrow \bigodot_{u\in X^*} \tau_u/\tau_u$
3: return $\varrho$
```

Again, if the subsemiring $S(\delta, \tau)$ of S has k elements, then the collection $\{\tau_u\}_{u\in X^*}$ can contain at most k^n different members, with n being the number of states of the weighted automaton $\mathscr{A}$. Then, the computation of the left Boolean residual τ_u/τ_u takes $O(n^2c_+)$ time, the computation of the whole collection $\{\tau_u/\tau_u\}_{u\in X^*}$ takes $O(k^n n^2 c_+)$ time, and the computation of the Hadamard product of all matrices from this collection also requires $O(k^n n^2 c_+)$ time. Thus, Step 2 runs in $O(k^n n^2 c_+)$ time, which does not exceed the computation time of Step 1 (i.e. of Algorithm 5). Therefore, the running time of Algorithm 6 is $O(mnk^{2n})$, the same as for Algorithm 5.

Summing up, the algorithm for computing the greatest weakly right (resp. left) invariant quasi-order matrix runs in exponential time, in contrast to the algorithm for computing the greatest right (resp. left) invariant quasi-order matrix which runs in polynomial time. In other words, the greatest weakly right (resp. left) invariant quasi-order matrix is harder to compute. On the other hand, the greatest weakly right (resp. left) invariant quasi-order matrix, when employed in the determinization of weighted automata, can result in deterministic weighted automata with smaller number of states than deterministic weighted automata obtained by using right (resp. left) invariant quasi-order matrix, as it is formally explained by Remark 4.5.

4. Determinization algorithm

Let $\mathcal{A} = (A, \delta, \sigma, \tau)$ be a WFA over a zero-divisor free semiring S, and $D = (f, g)$ a factorization of dimension A. For each $u \in X^*$ define an A-vector $\sigma_u^D \in S^A$ inductively as follows: $\sigma_\varepsilon^D = f(\sigma)$ and for every $u \in X^*$ and $x \in X$ we set $\sigma_{ux}^D = f(\sigma_u^D \cdot \delta_x)$. Let us set $A^D = \{\sigma_u^D | u \in X^*\}$, and let us define functions $\delta^D : A^D \times X \times A^D \to S$ and $\tau^D : A^D \to S$ with

$$\delta^D(\mu, x, \nu) = \begin{cases} g(\sigma_u^D \cdot \delta_x), & \mu = \sigma_u^D \text{ and } \nu = \sigma_{ux}^D \\ 0, & \text{otherwise} \end{cases}, \quad \text{for every } u \in X^* \text{ and } x \in X,$$

$$\tau^D(\mu) = \mu \cdot \tau, \quad \text{for every } \alpha \in A^D.$$

The following two theorems come from Kirsten and Mäurer [33].

Theorem 4.1. *Let $\mathcal{A} = (A, \sigma, \delta, \tau)$ be a WFA over a zero-divisor free semiring, and $D = (f, g)$ a factorization of dimension A. Then $\mathcal{A}^D = (A^D, \{\sigma_\varepsilon^D / g(\sigma)\}, \delta^D, \tau^D)$ is a well-defined CDWA equivalent to $\mathcal{A}$.*

Theorem 4.2. *Let $\mathcal{A}$ be a WFA over a zero-divisor free semiring, $N = (f_N, g_N)$, $M = (f_M, g_M)$ and $D = (f, g)$ be the trivial, a maximal and an arbitrary factorization of dimension A, with $\mathcal{A}^N$, $\mathcal{A}^M$ and $\mathcal{A}^D$ the corresponding CDWAs of $\mathcal{A}$, respectively. Then $|\mathcal{A}^M| \leqslant |\mathcal{A}^D|$ and $|\mathcal{A}^M| \leqslant |\mathcal{A}^N|$.*

Now, let S be a zero-divisor free and additively idempotent semiring, and let $\varrho \in 2^{A\times A}$ be a Boolean $A \times A$-matrix. Define a WFA $\mathcal{A}_\varrho = (A, \delta_\varrho, \sigma_\varrho, \tau_\varrho)$ with

$$\sigma_\varrho = \sigma \cdot \varrho, \quad \tau_\varrho = \tau,$$
$$\delta_\varrho(a, x, b) = (\delta_x \cdot \varrho)(a, b), \quad \text{for every } x \in X \text{ and } a, b, \in A.$$

Denote a family $(A_\varrho)^D = \{(\sigma_\varrho)_u^D\}_{u\in X^*}$ with $A_\varrho^D = \{\varrho_u^D\}_{u\in X^*}$, and denote a weighted automaton $(\mathcal{A}_\varrho)^D = ((A_\varrho)^D, \{(\sigma_\varrho)_\varepsilon^D / g(\sigma \cdot \varrho)\}, (\delta_\varrho)^D, (\tau_\varrho)^D)$ simply with $\mathcal{A}_\varrho^D = (A_\varrho^D, \{\varrho_\varepsilon^D / g(\sigma \cdot \varrho)\}, \delta_\varrho^D, \tau_\varrho^D)$. According to Theorem 4.1, $\mathcal{A}_\varrho^D$ is a CDWA (not necessarily finite), thus by (9) we have that the language accepted by this weighted automaton is equal to

$$[\![\mathcal{A}_\varrho^D]\!](u) = c_u^{\varrho,D} \cdot \tau_\varrho^D(\varrho_u^D),$$

for every word $u = x_1 x_2 \ldots x_n \in X^*$, where $c_u^{\varrho,D} \in S$ is equal to

$$c_u^{\varrho,D} = g(\sigma \cdot \varrho) \cdot \prod_{i=1}^{n} \delta_\varrho^D(\varrho_{x_1x_2\ldots x_{i-1}}^D, x_i, \varrho_{x_1x_2\ldots x_i}^D).$$

In addition, one can easily prove that

$$\sigma \cdot \varrho \cdot \prod_{i=1}^{n} (\delta_{x_i} \cdot \varrho) = c_{x_1x_2\ldots x_n}^{\varrho,D} \cdot \varrho_{x_1x_2\ldots x_n}^D. \tag{25}$$

We firstly determine the conditions under which the equivalence of $\mathcal{A}$ and $\mathcal{A}_\varrho^D$ follows.

Theorem 4.3. *Let $\mathcal{A} = (A, \delta, \sigma, \tau)$ be a WFA over an additive idempotent, zero-divisor free semiring, and $\varrho \in 2^{A\times A}$ a reflexive weakly right invariant matrix, and let $\mathcal{A}_\varrho = (A, \delta_\varrho, \sigma_\varrho, \tau_\varrho)$. Then we have $[\![\mathcal{A}]\!] = [\![\mathcal{A}_\varrho^D]\!]$.*

Proof. Let $u = x_1x_2\dots x_n \in X^*$ with $n \in \mathbb{N}_0$ be an arbitrary word. Then by (8) and the successive application of (14), together with the fact that ϱ is a reflexive matrix, we get

$$[\![\mathcal{A}_\varrho]\!](u) = \sigma_\varrho \cdot \left(\prod_{i=1}^{n}(\delta_\varrho)_{x_i}\right)\cdot \tau_\varrho = \sigma\cdot\varrho\cdot\left(\prod_{i=1}^{n}\delta_{x_i}\cdot\varrho\right)\cdot\tau = \sigma\cdot\left(\prod_{i=1}^{n}\varrho\cdot\delta_{x_i}\right)\cdot\varrho\cdot\tau = \sigma\cdot\left(\prod_{i=1}^{n}\delta_{x_i}\right)\cdot\tau = [\![\mathcal{A}]\!](u).$$

Since the previous holds for every $n \in \mathbb{N}_0$, the equivalence of weighted automata $\mathcal{A}$ and $\mathcal{A}_\varrho$ follows. On the other hand, the equivalence of $\mathcal{A}_\varrho$ and $\mathcal{A}_\varrho^D$ follows from Theorem 4.1. The proof is therefore completed. □

With the following theorem, we show that further improvements in the determinization can be made for WFAs over commutative, additively idempotent, zero-divisor free semirings by using right invariant quasi-order matrices.

Theorem 4.4. *Let $\mathcal{A} = (A, \delta, \sigma, \tau)$ be a WFA over a commutative, additively idempotent, zero-divisor free semiring, $D = (f, g)$ a maximal factorization of dimension A, and let $\varphi, \phi \in 2^{A\times A}$ be right invariant quasi-order matrices. Let $\mathcal{A}_\varphi^D = (A_\varphi^D, \delta_\varphi^D, \{\varphi_\varepsilon^D/g(\sigma\circ\varphi)\}, \tau_\varphi^D)$ and $\mathcal{A}_\phi^D = (A_\phi^D, \delta_\phi^D, \{\phi_\varepsilon^D/g(\sigma\circ\phi)\}, \tau_\phi^D)$ be the corresponding CDWAs of $\mathcal{A}$. If $\varphi \leqslant \phi$ holds, then $|\mathcal{A}_\phi^D| \leqslant |\mathcal{A}_\varphi^D|$ follows.*

Proof. Define a function $\xi : A_\varphi^D \to A_\phi^D$ with $\xi(\varphi_u^D) = \phi_u^D$ for every $u \in X^*$. We prove that ξ is a well-defined function. Indeed, let $u = x_1x_2\dots x_n \in X^*$ and $v = y_1y_2\dots y_m \in X^*$ be two arbitrary words such that $\varphi_u^D = \varphi_v^D$ holds. Then by (25) we get

$$c_v^{\varphi,D}\cdot\left(\sigma\cdot\varphi\cdot\prod_{i=1}^{n}(\delta_{x_i}\cdot\varphi)\right) = c_v^{\varphi,D}\cdot c_u^{\varphi,D}\cdot\varphi_u^D = c_u^{\varphi,D}\cdot c_v^{\varphi,D}\cdot\varphi_v^D = c_u^{\varphi,D}\cdot\left(\sigma\cdot\varphi\cdot\prod_{i=1}^{m}(\delta_{y_i}\cdot\varphi)\right).$$

By using the previous relation with Lemma 2.1, we obtain the following

$$\begin{aligned}
c_v^{\varphi,D}\cdot c_u^{\phi,D}\cdot\phi_u^D &= c_v^{\varphi,D}\cdot\left(\sigma\cdot\phi\cdot\prod_{i=1}^{n}(\delta_{x_i}\cdot\phi)\right) = c_v^{\varphi,D}\cdot(\sigma\cdot\delta_u\cdot\phi) = c_v^{\varphi,D}\cdot(\sigma\cdot\delta_u\cdot\varphi\cdot\phi)\\
&= c_v^{\varphi,D}\cdot\left(\sigma\cdot\varphi\cdot\left(\prod_{i=1}^{n}(\delta_{x_i}\cdot\varphi)\right)\cdot\phi\right) = \left(c_v^{\varphi,D}\cdot\left(\sigma\cdot\varphi\cdot\prod_{i=1}^{n}(\delta_{x_i}\cdot\varphi)\right)\right)\cdot\phi\\
&= \left(c_u^{\varphi,D}\cdot\left(\sigma\cdot\varphi\cdot\prod_{i=1}^{m}(\delta_{y_i}\cdot\varphi)\right)\right)\cdot\phi = c_u^{\varphi,D}\cdot\left(\sigma\cdot\varphi\cdot\left(\prod_{i=1}^{m}(\delta_{y_i}\cdot\varphi)\right)\cdot\phi\right)\\
&= c_u^{\varphi,D}\cdot(\sigma\cdot\delta_v\cdot\varphi\cdot\phi) = c_u^{\varphi,D}\cdot(\sigma\cdot\delta_v\cdot\phi) = c_u^{\varphi,D}\cdot\left(\sigma\cdot\phi\cdot\prod_{i=1}^{m}(\delta_{y_i}\cdot\phi)\right).
\end{aligned}$$

Again, by (25) we get

$$c_v^{\varphi,D}\cdot c_u^{\phi,D}\cdot\phi_u^D = c_u^{\varphi,D}\cdot c_v^{\phi,D}\cdot\phi_v^D,$$

and since $D = (f, g)$ is the maximal factorization, we finally obtain

$$\phi_u^D = f(\phi_u^D) = f(c_v^{\varphi,D}\cdot c_u^{\phi,D}\cdot\phi_u^D) = f(c_u^{\varphi,D}\cdot c_v^{\phi,D}\cdot\phi_v^D) = f(\phi_v^D) = \phi_v^D.$$

In conclusion, we proved that $\varphi_u^D = \varphi_v^D$ implies $\phi_u^D = \phi_v^D$, which ensures that ξ is a well-defined function. In addition, it is obvious that for every $\phi_u^D \in A_\phi^D$ there exists $\varphi_u^D \in A_\varphi^D$ such that $\xi(\varphi_u^D) = \phi_u^D$, for an arbitrary $u \in X^*$, which makes ξ a surjective mapping. Thus, $|A_\phi^D| \leqslant |A_\varphi^D|$ follows. □

Remark 4.5. *Let us compare the size of CDWAs obtained by using weakly right invariant quasi-order matrices with those obtained by using right invariant ones. Recall from Section 3 that the set of all right invariant quasi-order matrices on a WFA $\mathcal{A}$ is contained in the set off all all weakly right invariant quasi-order matrices of $\mathcal{A}$. Therefore, for the greatest weakly right invariant quasi-order matrix ϱ^{wri} on $\mathcal{A}$ and the greatest right invariant quasi-order matrix ϱ^{ri} on $\mathcal{A}$ we have that $\varrho^{ri} \leqslant \varrho^{wri}$. Thus, according to Theorem 4.4, it follows that $|\mathcal{A}^D_{\varrho^{wri}}| \leqslant |\mathcal{A}^D_{\varrho^{ri}}|$, with D being the maximal factorization of A. The previous inequality may be strict, which means that in general, weakly right invariant quasi-order matrices provide better results than the right invariant ones when used in the determinization. However, as we have seen, from the aspect of computation time, the advantage is on the side of right invariant quasi-order matrices.*

With the following example we show the case when $\mathcal{A}^D$ is infinite, but $\mathcal{A}^D_\varrho$ is a finite CDWA, with ϱ being the greatest right invariant quasi-order matrix, and D being the maximal factorization of dimension A.

Example 4.6. *Let $\mathcal{A} = (A, \delta, \sigma, \tau)$ be a WFA over the alphabet $X = \{x, y\}$ and the tropical semiring, with the transition graph given by Figure 2, or in other words, with the initial weighted row vector, final weighted column vector and weighted transition matrices given by*

$$\sigma = \begin{bmatrix} 0 & \infty & \infty & \infty \end{bmatrix}, \ \tau = \begin{bmatrix} \infty \\ \infty \\ \infty \\ 0 \end{bmatrix}, \ \delta_x = \begin{bmatrix} \infty & 1 & 0 & \infty \\ \infty & 1 & \infty & \infty \\ \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & \infty \end{bmatrix}, \ \delta_y = \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 0 \\ \infty & \infty & \infty & 0 \\ \infty & \infty & \infty & \infty \end{bmatrix}.$$

One can easily verify that $\mathcal{A}$ does not satisfy the twins property *(cf. [38]), therefore, $\mathcal{A}^D$ is an infinite CDWA. On the other hand, by means of Algorithm 3 we can compute the greatest right invariant quasi-order matrix $\varrho \in 2^{A\times A}$, which is given by*

$$\varrho = \begin{bmatrix} 0 & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty \\ 0 & 0 & 0 & \infty \\ \infty & \infty & \infty & 0 \end{bmatrix}.$$

In this way, we can obtain the WFA $\mathcal{A}_\varrho$ which is determinizable. The transition graph of $\mathcal{A}^D_\varrho$ is depicted in Figure 3.

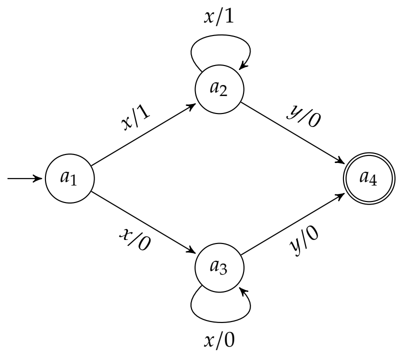

Figure 2: The transition graph of the WFA $\mathcal{A}$ from Example 4.6.

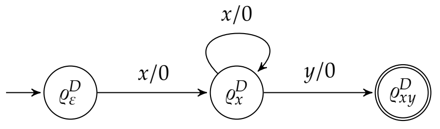

Figure 3: The transition graph of the CDWFA $\mathcal{A}^D_\varrho$ equivalent to $\mathcal{A}$.

References

[1] L. Aceto, A. Ingólfsdóttir, J. Srba, The Algorithmics of Bisimilarity, In: D. Sangiorgi & J. Rutten (Eds.), Advanced Topics in Bisimulation and Coinduction, Chapter 3, Cambridge: Cambridge University Press 52 (2011) 100–172.

[2] C. Baier, B. Engelen, M. Majster-Cederbaum, Deciding Bisimilarity and Similarity for Probabilistic Processes, Journal of Computer and System Sciences 60 (1) (2000) 187–231.

[3] J. van Benthem, Modal correspondence theory, Ph.D. Thesis, Universiteit van Amsterdam, Instituut voor Logica en Grondslagenonderzo ek van Exacte Wetenschappen, 1976.

[4] P. Buchholz, Bisimulation relations for weighted automata, Theoretical Computer Science, 393 (2008) 109–123.

[5] C.S. Calude, E. Calude, B. Khoussainov, Finite nondeterministic automata: Simulation and minimality, Theoretical Computer Science 242 (2000) 219–235.

[6] C. Câmpeanu, N. Sântean, S. Yu, Mergible states in large NFA, Theoretical Computer Science 330 (2005) 23–34.

[7] J.-M. Champarnaud, F. Coulon, NFA reduction algorithms by means of regular inequalities, in: Z. Ésik, and Z. Fülöp (eds.), DLT 2003, Lecture Notes in Computer Science 2710 (2003) 194–205.

[8] J.-M. Champarnaud, F. Coulon, NFA reduction algorithms by means of regular inequalities, Theoretical Computer Science 327 (2004) 241–253 (erratum: Theoretical Computer Science 347 (2005) 437–40).

[9] R. Cleaveland and O. Sokolsky, Equivalence and Preorder Checking for Finite-State Systems, In: J.A. Bergstra & A. Ponse & S.A. Smolka (Eds.), Handbook of Process Algebra, Chapter 6, Elsevier Science, Amsterdam (2001) 391 – 424.

[10] M. Ćirić, M. Droste, J. Ignjatović, H. Vogler, Determinization of weighted finite automata over strong bimonoids, Information Sciences 180 (2010) 3497-3520.

[11] M. Ćirić, J. Ignjatović, N. Damljanović, M. Bašić, Bisimulations for fuzzy automata, Fuzzy Sets and Systems 186 (2012) 100–139.

[12] M. Ćirić, J. Ignjatović, M. Bašić, I. Jančić, Nondeterministic automata: Equivalence, bisimulations, and uniform relations, Information Sciences 261 (2014) 185 – 218.

[13] M. Ćirić, J. Ignjatović, I. Jančić, N. Damljanović, Computation of the greatest simulations and bisimulations between fuzzy automata, Fuzzy Sets and Systems 208 (2012) 22–42.

[14] M. Ćirić, A. Stamenković, J. Ignjatović, T. Petković, Factorization of fuzzy automata, in: E. Csuhaj-Varju, Z. Ésik (Eds.), FCT 2007, in: Lecture Notes in Comput. Sci., vol. 4639, Springer, Heidelberg, 2007, pp. 213–225.

[15] M. Ćirić, A. Stamenković, J. Ignjatović, T. Petković, Fuzzy relation equations and reduction of fuzzy automata, Journal of Computer and System Sciences 76 (2010) 609–633.

[16] N. Damljanović, M. Ćirić, J. Ignjatović, Bisimulations for weighted automata over an additively idempotent semiring, Theoretical Computer Science 534 (2014) 86–100.

[17] A. Dovier, C. Piazza, A. Policriti, An efficient algorithm for computing bisimulation equivalence, Theoretical Computer Science 311 (2004) 221–256.

[18] M. Droste, W. Kuich, H. Vogler (Eds.), Handbook of Weighted Automata, Monographs in Theoretical Computer Science, An EATCS Series, Springer-Verlag, 2009.

[19] M. Forti, F. Honsell, Set theory with free construction principles, Annali Scuola Normale Superiore Pisa, Cl. Sc. IV (10) (1983) 493522.

[20] M. Habib, C. Paul, L. Viennoti, A synthesis on partition refinement: A useful routine for strings, graphs, boolean matrices and automata, In: M. Morvan, C. Meinel, D. Krob (Eds.) STACS 98. STACS 1998, Lecture Notes in Computer Science, vol 1373. Springer, Berlin, Heidelberg (1998).

[21] J. Ignjatović, M. Ćirić, Weakly linear systems of fuzzy relation inequalities and their applications: A brief survey, Filomat 26:2 (2012), 207–241.

[22] J. Ignjatović, M. Ćirić, S. Bogdanović, On the greatest solutions to weakly linear systems of fuzzy relation inequalities and equations, Fuzzy Sets and Systems 161 (2010) 3081–3113.

[23] J. Ignjatović, M. Ćirić, N. Damljanović, I. Jančić, Weakly linear systems of fuzzy relation inequalities: The heterogeneous case, Fuzzy Sets and Systems 199 (2012) 64–91.

[24] J. Ignjatović, M. Ćirić, B. Šešelja, A. Tepavčević, Fuzzy relational inequalities and equations, fuzzy quasi-orders, closures and openings of fuzzy sets, Fuzzy Sets and Systems 260 (2015) 1–24.

[25] L. Ilie, S. Yu, Algorithms for computing small NFAs, in: K. Diks et al. (eds): MFCS 2002, Lecture Notes in Computer Science 2420 (2002) 328–340.

[26] L. Ilie, S. Yu, Reducing NFAs by invariant equivalences, Theoretical Computer Science 306 (2003) 373–390.

[27] L. Ilie, G. Navarro, S. Yu, On NFA reductions, in: J. Karhumäki et al. (eds): Theory is Forever, Lecture Notes in Computer Science 3113 (2004) 112–124.

[28] L. Ilie, R. Solis-Oba, S. Yu, Reducing the size of NFAs by using equivalences and preorders, in: A. Apostolico, M. Crochemore, and K. Park (eds): CPM 2005, Lecture Notes in Computer Science 3537 (2005) 310–321.

[29] J. Friedl, Mastering Regular Expressions, O'Reilly Media, Sebastopol, CA, 3rd edn (2006).

[30] Z. Jančić, J. Ignjatović, M. Ćirić, An improved algorithm for determinization of weighted and fuzzy automata, Information Sciences 181 (2011) 1358–1368.

[31] Z. Jančić, I. Micić, J. Ignjatović, M. Ćirić, Further improvement of determinization methods for fuzzy finite automata, Fuzzy Sets and Systems 301(2015) 79–102.

[32] P. Kanellakis, S. Smolka, CCS expressions, finite state processes, and three problems of equivalence. Information and Computation 86(1) (1990) 43–68.

[33] D. Kirsten, L. Mäurer, On the determinization of weighted automata, Journal of Automata, Languages and Combinatorics 10 (2005) 287–312.

[34] F. Marass, C. Upton, Sequence Searcher: A Java tool to perform regular expression and fuzzy searches of multiple DNA and protein sequences, BMC Research Notes 2:14 (2009).

[35] I. Micić, Z. Jančić, J. Ignjatović, M. Ćirić, Determinization of fuzzy automata by means of the degrees of language inclusion, IEEE Transactions on Fuzzy Systems 23 (6) (2015) 2144–2153.

[36] I. Micić, Z. Jančić, S. Stanimirović, Computation of the greatest right and left invariant fuzzy quasi-orders and fuzzy equivalences, Fuzzy Sets and Systems 339 (2018) 99-118.
[37] R. Milner, A calculus of communicating systems, in: G. Goos, J. Hartmanis (Eds.), Lecture Notes in Computer Science, Springer (1980) vol. 92.
[38] M. Mohri, Finite-state transducers in language and speech processing, Computational Linguistics 23 (2) (1997) 269–311.
[39] R. Paige, R. E. Tarjan, Three partition refinement algorithms, SIAM Journal of Computing 16 (1987) 973-989.
[40] D. Park, Concurrency and automata on infinite sequences, in: P. Deussen (Ed.), Proc. 5th GI Conf., Karlsruhe, Germany, in: Lecture Notes in Computer Science, vol.104, Springer-Verlag, 1981, pp. 167–183.
[41] M. O. Rabin, D. Scott, Finite automata and their decision problems, IBM J. Res. Dev. (1959).
[42] M. Shamsizadeh, M. M. Zahedi, K. Abolpour, Bisimulation For BL-general Fuzzy Automata, Iranian Journal of Fuzzy Systems 13 (4) (2016) 35–50.
[43] A. Stamenković, M. Ćirić, J. Ignjatović, Reduction of fuzzy automata by means of fuzzy quasi-orders, Information Sciences 275 (2014) 168–198.
[44] S. Stanimirović, M. Ćirić, J. Ignjatović, Determinization of fuzzy automata by factorizations of fuzzy states and right invariant fuzzy quasi-orders, Information Sciences 469 (2018) 79–100.
[45] P. Terry, Compiling with C# and Java, Pearson/Addison-Wesley (2005).
[46] H. Wu, Y. Deng, Logical Characterizations of Simulation and Bisimulation for Fuzzy Transition Systems, Fuzzy Sets and Systems 301 (2016) 19–36.