Filomat 36:8 (2022), 2775–2793
https://doi.org/10.2298/FIL2208775S

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

Certain Linear and Weakly Linear Systems of Matrix Equations Over Semirings. Applications in a State Reduction of Weighted Automata

Aleksandar Stamenković[a], Stefan Stanimirović[a], Vesa Halava[b]

[a] *University of Niš, Faculty of Sciences and Mathematics, Višegradska 33, 18000 Niš, Serbia*
[b] *University of Turku, Department of Mathematics and Statistics, FIN-20014 Turku, Finland*

Abstract. In this paper we study particular linear and weakly linear systems of matrix equations over semirings, with the aim of describing and computing functions as solutions to those systems. Our special attention is devoted to solutions whose ranks are as small as possible. We prove the existence of solutions with the smallest ranks, give their characterizations, and present a method for their computations. Regarding weakly linear systems, the method is based on the well known partition refinement algorithm by Kanellakis and Smolka, adapted to work with weighted labeled transition systems. Moreover, we give a state reduction procedure of weighted automata based on a decomposition of solutions to particular linear and weakly linear systems.

1. Introdudction

Systems of matrix equations studied in this paper, beside linear systems, are a subclass of weakly linear systems (abbr. WLS), systems of matrix inequations and equations that have been widely studied in the past. In particular, WLS have emerged in the study of fuzzy automata, and have also been applied in social network analysis. They have been used to reduce the number of states of fuzzy automata [9, 10, 23, 24], in the study of simulations and bisimulations for nondeterministic, fuzzy and weighted automata [5–7, 11], as well as in the positional analysis of social networks [12, 13]. In addition, certain types of WLS have been used to improve determinization algorithms for fuzzy automata [16, 25] and in the conflict analysis of fuzzy discrete event systems [24].

In all these cases, the underlying structures of membership values are complete residuated lattices, and the key role in those studies is played by completeness and residuation in such structures. Namely, those two properties ensure completeness and residuation in the corresponding lattices of fuzzy matrices, ensuring also the existence of the greatest solutions to WLS and providing necessary algebraic tools for their computation. Unlike complete residuated lattices, semirings are structures that are neither ordered nor complete and do not provide a residuation, making the study of WLS very challenging. As a consequence,

2020 *Mathematics Subject Classification*. Primary 15A24; Secondary 15A80, 16Y60, 68Q70
Keywords. Semirings; Matrices over semirings; Weakly linear system; Partition refinement; Weighted automata; State reduction
Received: 10 April 2019; Revised: 15 November 2019; Accepted: 14 January 2020
Communicated by Miroslav Ćirić
Research supported by Ministry of Education, Science and Technological Development, Republic of Serbia, Contract No. 451-03-68/2022-14/200124 and by the Science Fund of the Republic of Serbia, GRANT No 7750185, Quantitative Automata Models: Fundamental Problems and Applications - QUAM

Email addresses: aca@pmf.ni.ac.rs (Aleksandar Stamenković), stefan.stanimirovic@pmf.edu.rs (Stefan Stanimirović), vesa.halava@utu.fi (Vesa Halava)

in order to overcome the lack of algebraic features of semirings, we introduce and investigate functions as solutions to WLS. In addition, due to the same reason, instead of computing the greatest solution to WLS, we focus on functions whose ranks are as small as possible, with the aim to achieve the best possible state reduction of weighted finite automata. Studying such functions over semirings appear to be very useful, considering the fact that such functions can be efficiently computed (Example 3.3). Moreover, provided ranks of identity matrices on finite sets are equal to the cardinality of those sets, ranks and corresponding rank decomposition of such matrices can also be efficiently computed. In addition, let us note that computing solutions to WLS that are not functions, along with computing their ranks, is, in general, impossible (cf. [23]).

The structure of the paper is as follows. After introducing a linear system of matrix equations (ls.1), we give a characterization of functions that are its solutions. We prove that solutions to the system (ls.1) with the smallest ranks are idempotent functions whose kernels are equal to *the equivalence associated to* (ls.1). Those results lead us to the simple method for their computation. Regarding weakly linear system (wls.1), we prove the existence of solutions with the smallest ranks and characterize them as idempotent functions whose kernels are equal to *the equivalence associated to* (wls.1). We also present a method for their computation, based on calculating the equivalence associated to (wls.1). In addition, we give an algorithm for computing equivalences associated to (wls.1) whose time complexity is smaller then those presented in [3, 9, 10, 14, 15, 21, 22]. We also show how this method can be used for computing solutions with the smallest rank to the both systems. Ultimately, we implement some of our results in a state reduction of weighted finite automata.

2. Preliminaries

A *semiring* is a structure $\mathcal{S} = (S, +, \cdot, 0, 1)$ consisting of a set S, two binary operations $+$ and $\cdot$ on S, and two constants $0, 1 \in S$, such that the following is true:

(i) $(S, +, 0)$ is a commutative monoid,

(ii) $(S, \cdot, 1)$ is a monoid,

(iii) the distributivity laws $(a + b) \cdot c = a \cdot c + b \cdot c$ and $c \cdot (a + b) = c \cdot a + c \cdot b$ hold for every $a, b, c \in S$,

(iv) $0 \cdot a = a \cdot 0 = 0$, for every $a \in S$.

Throughout this paper, let S be a semiring. For arbitrary finite non-empty sets A and B, a mapping $M : A \times B \to S$ is called an $A \times B$ *matrix over* S. As usual when working with mappings, we denote the set of all $A \times B$ matrices over S by $S^{A\times B}$. The *transpose* of an $A \times B$ matrix M is a $B \times A$ matrix $M^\top$ defined by $M^\top(b, a) = M(a, b)$, for every $a \in A, b \in B$. B *row vector* (resp. A *column vector*) *over* S is an $A \times B$ matrix over S, where $|A| = 1$ (resp. $|B| = 1$). Sometimes, we identify a B row vector (resp. A column vector) with a mapping $M : B \to S$ (resp. $M : A \to S$), and accordingly denote the set of all B row vectors (resp. A column vectors) by S^B (resp. S^A). For an $A \times B$ matrix M, its *row vector corresponding to* a, denoted by aM, and its *column vector corresponding to* b, denoted by Mb, are defined as follows

$$aM(b) = M(a, b) \quad \text{and} \quad Mb(a) = M(a, b),$$

for arbitrary $a \in A$ and $b \in B$.

For finite non-empty sets A, B and C, and matrices $M \in S^{A\times B}$ and $T \in S^{B\times C}$, we define the *matrix product* $M \cdot T \in S^{A\times C}$, by

$$(M \cdot T)(a, c) = \sum_{b\in B} M(a, b) \cdot T(b, c), \tag{1}$$

for any $(a, c) \in A \times C$. If $Q = M \cdot T$, the matrix pair (M, T) is called a *decomposition* of Q, and if $|B| = k$, a decomposition (M, T) is called a *k-decomposition* of a matrix Q. Matrix $M \in S^{A\times A}$ is *idempotent* if $M^2 = M$. *Matrix sum* of matrices $M, T \in S^{A\times B}$, denoted by $M + T$, is a matrix defined by

$$(M + T)(a, b) = M(a, b) + T(a, b), \tag{2}$$

for every $a \in A$, $b \in B$.

Rank of a nonzero matrix M, denoted $\rho(M)$, is the smallest integer k such that there exists a k-decomposition of M. Rank of a zero matrix is equal to 0. Note that, not only in the Boolean matrix theory, but also in the theory of fuzzy matrices [1, 2, 18], the above defined number $\rho(M)$ is known as the *Schein rank* of a matrix. In addition, the number $\rho(M)$ is called the *fuzzy rank* of M, where M is a matrix over a semiring defined on the real unit interval $[0,1]$ with $x \cdot y = \min(x,y)$ and $x + y = \max(x,y)$ (cf. [4]). For other notions regarding row and column spaces as well as ranks of matrices refer to [23].

In this paper, matrices taking values in the set $\{0,1\}$, where $0,1 \in S$, are called *relations between A and B*. The set of all relations of A to B is denoted by $2^{A\times B}$, and for $E \in 2^{A\times B}$, expressions "$E(a,b) = 1$" and "$(a,b) \in E$" have the same meaning. For relations $E, F \in 2^{A\times B}$, the ordering $E \leqslant F$ is defined pointwise, provided $0 \leqslant 1$. In case when $E \leqslant F$, we say that E is *contained* in F. Further, a relation $E \in 2^{A\times A}$ is called *reflexive*, if $E(a,a) = 1$, for every $a \in A$; *symmetric*, if $E(a,b) = E(b,a)$, for all $a,b \in A$; and *transitive*, if $E(a,b) = 1$ and $E(b,c) = 1$ implies $E(a,c) = 1$, for all $a,b,c \in A$. A reflexive, symmetric and transitive relation is called an *equivalence relation on A*. For each $a \in A$, both row vector aE and column vector Ea are called the *equivalence class of E determined by a*. This identification is justified by the fact that due to the symmetry of E, vectors aE and Ea, considered as mappings, are equal. Moreover, equivalence classes have the same properties as classes of equivalence relations, defined in a usual way. Namely, the following is true

$$E(a,b) = 1 \iff aE = bE \iff Ea = Eb. \tag{3}$$

The set of all equivalence classes of E is denoted by A/E and called the *factor set* of A with respect to E.

Matrix $\varphi \in 2^{A\times B}$ is called a *functional matrix* if it corresponds to a relation which is a function, i.e., if for every $a \in A$ there exists $b \in B$ such that $\varphi(a,b) = 1$, and $\varphi(a,b) = \varphi(a,c) = 1$ implies $b = c$, for all $a \in A$ and $b, c \in B$. In the sequel, functional matrices will be simply called functions. For a function φ and arbitrary $a \in A$, let us denote by $\varphi(a)$ an element $b \in B$, such that $\varphi(a,b) = 1$. Let us note that a product of functions $\varphi \in 2^{A\times B}$ and $\psi \in 2^{B\times C}$ is a function $\varphi \cdot \psi \in 2^{A\times C}$, and $(\varphi \cdot \psi)(a) = \psi(\varphi(a))$, for every $a \in A$. Function φ is *injective* if $\varphi(a) = \varphi(b)$ implies $a = b$, for every $a, b \in A$. Kernel of a function $\varphi \in 2^{A\times B}$ is an equivalence relation $Ker\varphi \in 2^{A\times A}$ defined in a usual way

$$Ker\varphi(a,b) = 1 \iff \varphi(a) = \varphi(b), \tag{4}$$

for every $a, b \in A$. For a given finite set A, the *identity matrix* on A, denoted by I_A, is a function defined by

$$I_A(a,b) = \begin{cases} 1, & \text{if } a = b \\ 0, & \text{otherwise} \end{cases}, \tag{5}$$

for every $a, b \in A$. *Universal relation on A*, denoted as ∇_A is a relation defined by $\nabla_A(a,b) = 1$, for each $a, b \in A$.

3. Systems of matrix equations

In this section we deal with particular systems of matrix equations, in order to study and to compute their solutions in the set of functions.

3.1. Idempotent functions

Firstly, we focus on basic properties of functions, especially idempotent functions.

Lemma 3.1. *Let $\varphi \in 2^{A\times A}$ be a function and let $E \in 2^{A\times A}$ be an equivalence relation. The following statements are equivalent:*

(i) $\varphi \leqslant E$,
(ii) $a\varphi \leqslant aE$,
(iii) $\varphi a \leqslant Ea$,
(iv) $E(a,\varphi(a)) = E(\varphi(a),a) = 1$,

(v) $Ea = E\varphi(a)$,

(vi) $aE = \varphi(a)E$,

(vii) $\varphi \cdot E = E$,

for every $a \in A$.

Proof. Let us only prove (iv) $\iff$ (vii).

For arbitrary $a, b \in A$, since $E(a, \varphi(a)) = 1$, we have that $E(a, b) = 1$ iff $E(\varphi(a), b) = 1$. From the fact that $\varphi \cdot E(a, b) = E(\varphi(a), b)$, for every $a, b \in A$, we conclude $\varphi \cdot E = E$.

Conversely, if $\varphi \cdot E(a, b) = E(a, b)$, for every $a, b \in A$, i.e. if $\varphi \cdot E(a, b) = E(\varphi(a), b)$, for every $a, b \in A$, by putting $b = a$, we obtain (iv).

The the rest of the proof is based on simple use of the fundamental properties of both functions and equivalence relations, which is why will be left out. □

Let $\varphi \in 2^{A\times A}$ be a function and let $E \in 2^{A\times A}$ be an equivalence relation. It is clear that $\varphi \leqslant E$ results in $Ker\varphi \leqslant E$. That does not necessarily implies $\varphi \leqslant Ker\varphi$. Following lemma establishes basic properties of functions satisfying $\varphi \leqslant Ker\varphi$.

Lemma 3.2. *Let* $\varphi \in 2^{A\times A}$ *be an arbitrary function. The following statements are equivalent:*

(i) $\varphi \leqslant Ker\varphi$,

(ii) $\varphi^2 = \varphi$,

(iii) $\varphi \cdot Ker\varphi = Ker\varphi$,

(iv) $\varphi = l \cdot r$, *where* $l \in 2^{A\times A/E}$ *is the canonical map w.r.t. E, defined by* $l : a \mapsto Ea$, *for every* $a \in A$, *and* $r \in 2^{A/E\times A}$ *is an injective function satisfying* $E(r(Ea), a) = 1$, *for every* $a \in A$.

Proof. From he previous lemma we have (i) $\iff$ (iii).

(i) $\implies$ (ii): If $\varphi \leqslant E$, then, by Lemma 3.1, we have $E(a, \varphi(a)) = 1$, for every $a \in A$. Hence, $\varphi(a) = \varphi(\varphi(a)) = \varphi^2(a)$, for all $a \in A$, i.e. $\varphi^2 = \varphi$.

(ii) $\implies$ (iii): If $\varphi^2 = \varphi$, then from $(\varphi \cdot E)(a, b) = E(\varphi(a), b) = Ker\varphi(\varphi(a), b)$, for every $a, b \in A$, we conclude that $Ker\varphi(\varphi(a), b) = 1$ iff $\varphi(\varphi(a)) = \varphi(b)$, i.e. iff $\varphi(a) = \varphi(b)$. Therefore, $E(\varphi(a), b) = 1$ iff $E(a, b) = 1$, for every $a, b \in A$, which is equivalent to $E(\varphi(a), b) = E(a, b)$, for every $a, b \in A$. Hence, $\varphi \cdot E = E$.

(i) $\implies$ (iv): Let, for an idempotent function φ, a relation $r \in 2^{A/E\times A}$ be defined by $r(Ea, b) = 1$ iff $b = \varphi(a)$, for every $a, b \in A$. Since $Ea = Eb$, i.e. $E(a, b) = 1$, implies $\varphi(a) = \varphi(b)$, we have that $r(Ea, c) = r(Eb, d) = 1$ results in $c = \varphi(a) = \varphi(b) = d$. This means that r is a function, and $r(Ea) = \varphi(a)$, for every $a \in A$. If $r(Ea) = r(Eb)$, then $\varphi(a) = \varphi(b)$, i.e. $Ker\varphi(a, b) = E(a, b) = 1$, or equivalently $Ea = Eb$. Therefore, r is an injective function. Finally, $E(r(Ea), a) = E(\varphi(a), a) = 1$, and

$$(l \cdot r)(a) = r(l(a)) = r(Ea) = \varphi(a),$$

for every $a \in A$, i.e. $\varphi = l \cdot r$.

(iv) $\implies$ (i): Suppose (iv) is true. Then, we have the following

$$E(\varphi(a), a) = E((l \cdot r)(a), a) = E(r(l(a)), a) = E(r(Ea), a) = 1,$$

for every $a \in A$. By Lemma 3.1, we obtain $\varphi \leqslant E$.

□

Let us also note, that the previous lemma provides a method for constructing idempotent functions whose kernel is equal to a given equivalence $E \in 2^{A\times A}$, where A is a finite set. The method is based on choosing $r(Ea) \in A$, for every $Ea \in A/E$, provided $E(r(Ea), a) = 1$. Defined this way, the matrix $r \in 2^{A/E\times A}$ is injective. Further, $\varphi = l \cdot r$, where $l \in 2^{A\times A/E}$ is the canonical map w.r.t. E, and since $\varphi(a) = r(Ea)$, it is clear that $Ker\varphi = E$. Next example illustrates the above described construction.

Example 3.3. *Let $A = \{a_1, a_2, a_3, a_4, a_5\}$, and $E \in 2^{A\times A}$ an equivalence relation*

$$E = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}, \quad \text{where } i^{th} \text{ row of } E \text{ represent an equivalence class of } E \text{ determined by } a_i.$$

Since $Ea_1 = Ea_2 = Ea_3 \neq Ea_4 = Ea_5$, i.e. $A/E = \{Ea_1, Ea_4\}$, in order to construct an injective function $r \in 2^{A/E\times A}$, the following conditions have to be satisfied

$$r(Ea_1) = r(Ea_2) = r(Ea_3) \in \{a_1, a_2, a_3\} \quad \text{and} \quad r(Ea_4) = r(Ea_5) \in \{a_4, a_5\}.$$

Clearly, there are six possible functions $r \in 2^{A/E\times A}$ that meet those requirements:

$$r_1(Ea_1) = a_1, \quad \text{and} \quad r_1(Ea_4) = a_4, \quad \text{i.e.} \quad r_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$r_2(Ea_1) = a_2, \quad \text{and} \quad r_2(Ea_4) = a_4, \quad \text{i.e.} \quad r_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$r_3(Ea_1) = a_3, \quad \text{and} \quad r_3(Ea_4) = a_4, \quad \text{i.e.} \quad r_3 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$r_4(Ea_1) = a_1, \quad \text{and} \quad r_4(Ea_4) = a_5, \quad \text{i.e.} \quad r_4 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$r_5(Ea_1) = a_2, \quad \text{and} \quad r_5(Ea_4) = a_5, \quad \text{i.e.} \quad r_5 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$r_6(Ea_1) = a_3, \quad \text{and} \quad r_6(Ea_4) = a_5, \quad \text{i.e.} \quad r_6 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Moreover, since the canonical map $l \in 2^{A\times A/E}$ w.r.t E, is the following matrix

$$l = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}, \quad \text{and} \quad \varphi = l \cdot r, \quad \text{all possible idempotent functions } \varphi, \text{ such that } Ker\varphi = E, \text{ are:}$$

$$\varphi_1 = l \cdot r_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad \varphi_2 = l \cdot r_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad \varphi_3 = l \cdot r_3 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$\varphi_4 = l \cdot r_4 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \varphi_5 = l \cdot r_5 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \varphi_6 = l \cdot r_6 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

In the sequel, for an idempotent function $\varphi \in 2^{A\times A}$, with $E = Ker\varphi$, the canonical map $l \in 2^{A\times A/E}$ w.r.t. E, and the injective function $r \in 2^{A/E\times A}$, defined by $r(Ea) = \varphi(a)$, for every $a \in A$, are denoted by l^{φ}, and r^{φ}, respectively.

The importance of idempotent matrices emerges from their applications in a state reduction of weighted finite automata, and will be carefully explained in Section 4. Namely, regarding solutions to the systems of matrix equations, we establish two main goals: the first one is to find solutions whose rank is as small as possible. The second one is to estimate their rank as well as possible, and to compute corresponding decomposition. Our next result concerns ranks of idempotent matrices.

Theorem 3.4. *Let $\varphi \in S^{A\times A}$ be an idempotent function, and $E = Ker\varphi$. Then,*

$$\rho(\varphi) = \rho(I_{A/E}) \leqslant |A/E|.$$

Proof. Let $\psi = r^{\varphi} \cdot \varphi \cdot l^{\varphi}$. The definition of ψ yields $\rho(\psi) \leqslant \rho(\varphi)$, and from the following

$$\varphi = \varphi^3 = (l^{\varphi} \cdot r^{\varphi}) \cdot \varphi \cdot (l^{\varphi} \cdot r^{\varphi}) = l^{\varphi} \cdot (r^{\varphi} \cdot \varphi \cdot l^{\varphi}) \cdot r^{\varphi} = l^{\varphi} \cdot \psi \cdot r^{\varphi},$$

we obtain the converse inequality $\rho(\varphi) \leqslant \rho(\psi)$. Thus, $\rho(\varphi) = \rho(\psi)$. Moreover, since

$$\psi(Ea) = (r^{\varphi} \cdot \varphi \cdot l^{\varphi})(Ea) = l^{\varphi}(\varphi(r^{\varphi}(Ea))) = l^{\varphi}(\varphi^2(Ea)) = l^{\varphi}(\varphi(a)) = E\varphi(a) = Ea,$$

for every $Ea \in A/E$, we conclude that $\psi = I_{A/E}$. Thus, $\rho(\varphi) = \rho(\psi) = \rho(I_{A/E})$. □

3.2. *System $X \cdot M_i = M_i, \quad i \in I$*

The first system studied here is an instance of ordinary linear systems of matrix equations, i.e. a system consisting of the following equations

$$X \cdot M_i = M_i, \quad i \in I, \tag{ls.1}$$

where I is a finite index set, $M_i \in S^{A\times B}$, $i \in I$ are given, and $X \in S^{A\times A}$ is an unknown matrix. In the further text, when aiming to emphasize the set of matrices $\mathcal{M} = \{\mathcal{M}_\rangle \in \mathcal{S}^{\mathcal{A}\times\mathcal{B}} \mid \rangle \in \mathcal{I}\}$, the above system will be called *the system (ls.1) determined by* $\mathcal{M}$. Since for all the results concerning the system (ls.1), corresponding results for the system

$$M_i \cdot X = M_i, \quad i \in I, \tag{ls.2}$$

not only exist, but can be analogously proven, their statements and proofs will be omitted. Let E^1 be an equivalence relation, called *the equivalence relation associated to (ls.1)*, defined by

$$E^1(a_1, a_2) = 1 \quad\iff\quad M_i(a_1, b) = M_i(a_2, b), \tag{6}$$

for every $a_1, a_2 \in A$, $b \in B$, and $i \in I$. The equivalence associated to the system (ls.1) play an important role in this subsection, since they are used for characterization of functions that are solutions to (ls.1). Our next result confirms this claim.

Theorem 3.5. *Let $\varphi \in 2^{A\times A}$ be a function and let $E^1 \in 2^{A\times A}$ be the equivalence relation associated to the system (ls.1). Then, φ is a solution to (ls.1) if and only if $\varphi \leqslant E^1$.*

Proof. Since, $(\varphi \cdot M_i)(a, b) = \sum_{c\in A} \varphi(a, c) \cdot M_i(c, b) = M_i(\varphi(a), b)$, for every $a \in A$, $b \in B$, and $i \in I$, we have that function φ is a solution to the system (ls.1) iff $M_i(\varphi(a), b) = M_i(a, b)$, for every $a \in A$, $b \in B$, and $i \in I$. From the definition of E^1 and Lemma 3.1, the last assertion is equivalent to $E^1(\varphi(a), a) = 1$, for every $a \in A$, i.e. to $\varphi \leqslant E^1$. □

Our last result in this subsection is the following one.

Theorem 3.6. *Let $\varphi \in 2^{A\times A}$ be a function such that $Ker\varphi = E^{l}$, where $E^{l} \in 2^{A\times A}$ is the equivalence relation associated to the system (ls.1). Then, φ is a solution to (ls.1) if and only if φ is idempotent.*

Moreover, $\rho(\varphi) \leqslant \rho(\psi)$, for an arbitrary function ψ which is a solution to (ls.1).

Proof. By the previous theorem, φ is a solution to (ls.1) if and only if $\varphi \leqslant E^{l}$, i.e. if and only if $\varphi \leqslant Ker\varphi$. By Lemma 3.2, the last condition is equivalent to φ being idempotent.

If ψ is a solution to (ls.1), then, by Theorem 3.5 and Lemma 3.1, we obtain $E^{l}(a, \psi(a)) = 1$ for arbitrary $a \in A$. Thus, from the fact that $Ker\varphi = E^{l}$, we conclude that $\varphi(a) = \varphi(\psi(a))$, for every $a \in A$, i.e. $\varphi = \psi \cdot \varphi$. Hence, $\rho(\varphi) \leqslant \rho(\psi)$. □

By the previous result, along with Theorem 3.4, we achieve both goals we pursued: in the set of functions that are solutions to the system (ls.1), idempotent functions whose kernel is equal to E^{l} have the smallest rank. Its rank is equal to the rank of the identity matrix $I_{A/E^{l}}$ and is smaller or equal to the cardinality of the set A/E^{l}.

3.3. System $X \cdot M_i \cdot X = M_i \cdot X, \quad i \in I$

The second system of matrix equations we study here is the system that consists of the equations

$$X \cdot M_i \cdot X = M_i \cdot X, \quad i \in I, \tag{wls.1}$$

where I is a finite index set, $M_i \in S^{A\times A}$, $i \in I$ are given, and $X \in S^{A\times A}$ is an unknown matrix. Similarly to the system (ls.1), in the further text, in order to emphasize the set of matrices $\mathcal{M} = \{\mathcal{M}_\rangle \in \mathcal{S}^{\mathcal{A}\times\mathcal{A}} \mid \rangle \in \mathcal{I}\}$, the above system will be called *the system (wls.1) determined by* $\mathcal{M}$. Moreover, for all the results regarding the system (wls.1), there are corresponding results for the system

$$X \cdot M_i \cdot X = X \cdot M_i, \quad i \in I, \tag{wls.2}$$

but their statements and proofs will be omitted.

Let E be an equivalence relation satisfying the following condition

$$E(a, b) = 1 \quad \Longrightarrow \quad (M_i \cdot E)(a, c) = (M_i \cdot E)(b, c), \tag{7}$$

for every $a, b, c \in A$, and $i \in I$.

Let us recall that, in the automata theory, equivalence relations that are solution to (wls.1), where matrices M_i, $i \in I$ represent transition matrices of automata, are called: right invariant [9, 10, 14, 15, 22] – in theory of NFAs and fuzzy automata, congruences [21] – in fuzzy automata theory, while in [3], bisimulations – in the theory of weighted automata. Moreover, in the above mentioned papers, the existence of the greatest right invariant equivalence (congruence, bisimulation) is proven, and efficient algorithms for their computations are given. Thus, the following definition is justified: the greatest equivalence relation that satisfies (7) is called *the equivalence relation associated to (wls.1)*, and is denoted by E^{wl}.

Lemma 3.7. *Let A and I be finite sets, $M_i \in S^{A\times A}$, $i \in I$, and $E \in 2^{A\times A}$ an equivalence relation. If $l : a \mapsto Ea$ is the canonical map w.r.t. E, then*

(i) *$(M_i \cdot E)(a, c) = (M_i \cdot l)(a, Ec)$, for every $a, c \in A$, and $i \in I$.*

(ii) *If E satisfies (7), then*

$$E(a, b) = 1 \quad \Longrightarrow \quad (M_i \cdot \varphi)(a, c) = (M_i \cdot \varphi)(b, c), \tag{8}$$

for every $a, b, c \in A$, and $i \in I$, where $\varphi = l \cdot r$, where l is the canonical map w.r.t. E, and $r \in 2^{A/E\times A}$ is an arbitrary relation.

Proof. (i) Since $E(a,b)=1$ iff $Ea=Eb$ iff $l(a,Eb)=1$, the following equalities hold

$$(M_i\cdot E)(a,c)=\sum_{d\in A}M_i(a,d)\cdot E(d,c)=\sum_{d\in A,\ E(d,c)=1}M_i(a,d)\cdot E(d,c)=\sum_{d\in A}M_i(a,d)\cdot l(d,Ec)=(M_i\cdot l)(a,Ec),$$

for every $a,c\in A$, and $i\in I$.

(ii) Let $E(a,b)=1$, for $a,b\in A$, and let $r\in 2^{A/E\times A}$ be an arbitrary relation. Since, by (i) of this lemma, $(M_i\cdot E)(a,c)=(M_i\cdot l)(a,Ec)$, for every $c\in A$, we have that

$$(M_i\cdot l)(a,Ec)=(M_i\cdot E)(a,c)=(M_i\cdot E)(b,c)=(M_i\cdot l)(b,Ec),$$

for every $c\in A$. Moreover, if $\varphi=l\cdot r$, we have

$$\begin{aligned}(M_i\cdot\varphi)(a,c)&=(M_i\cdot(l\cdot r))(a,c)=((M_i\cdot l)\cdot r)(a,c)=\sum_{Ed\in A/E}(M_i\cdot l)(a,Ed)\cdot r(Ed,c)\\&=\sum_{Ed\in A/E}(M_i\cdot l)(b,Ed)\cdot r(Ed,c)=(M_i\cdot\varphi)(b,c)\end{aligned}$$

for every $c\in A$, and $i\in I$. □

By the following result we give some necessary and sufficient conditions for functions to be solutions to (wls.1).

Theorem 3.8. *Let $\varphi\in 2^{A\times A}$ be a function. The following statements are true*

(i) *If φ is a solution to (wls.1), then the equivalence $Ker\varphi$ satisfies (7).*

(ii) *If φ is idempotent and $\varphi\leqslant E$, for some equivalence relation $E\in 2^{A\times A}$ that satisfies (7), then φ is a solution to (wls.1).*

Proof. (i) Function φ is a solution to (wls.1) iff $(\varphi\cdot M_i\cdot\varphi)(a,b)=(M_i\cdot\varphi)(a,b)$, for every $a,b\in A$, and $i\in I$, or equivalently

$$(M_i\cdot\varphi)(\varphi(a),b)=(M_i\cdot\varphi)(a,b), \tag{9}$$

for every $a,b\in A$, and $i\in I$. Let φ be a solution to (wls.1), and let $E=Ker\varphi$. From the previous consideration, we have

$$(M_i\cdot\varphi)(a,c)=(M_i\cdot\varphi)(\varphi(a),c),\quad\text{and}\quad(M_i\cdot\varphi)(b,c)=(M_i\cdot\varphi)(\varphi(b),c),$$

for every $a,b,c\in A$, and $i\in I$. If $E(a,b)=1$, for $a,b\in A$, or equivalently if $\varphi(a)=\varphi(b)$, then $(M_i\cdot\varphi)(a,c)=(M_i\cdot\varphi)(b,c)$, thus, as a consequence of the following equalities

$$\begin{aligned}(M_i\cdot E)(a,c)&=\sum_{d\in A}M_i(a,d)\cdot E(d,c)=\sum_{d\in A,\ E(d,c)=1}M_i(a,d)=\sum_{d\in A,\ \varphi(d)=\varphi(c)}M_i(a,d)\\&=\sum_{d\in A}M_i(a,d)\cdot\varphi(d,\varphi(c))=(M_i\cdot\varphi)(a,\varphi(c))=(M_i\cdot\varphi)(b,\varphi(c))\\&=(M_i\cdot E)(b,c),\end{aligned}$$

for every $c\in A$, we have that E is an equivalence satisfying (7).

(ii) Let $\varphi\leqslant E$, for some equivalence E that satisfies (7). Then, $E(\varphi(a),a)=1$, for every $a\in A$, and since φ is idempotent, by (iv) of Lemma 3.2, we have that $\varphi=l^{\varphi}\cdot r^{\varphi}$. By (ii) of Lemma 3.7, it is $(M_i\cdot\varphi)(a,b)=(M_i\cdot\varphi)(\varphi(a),b)$, for every $a,b\in A$, and $i\in I$. Ultimately, from (9) we obtain that φ is a solution to (wls.1). □

We are ready now for result regarding (wls.1) that corresponds to Theorem 3.6.

Theorem 3.9. *Let $\varphi \in 2^{A\times A}$ be a function such that $Ker\varphi = E^{\mathrm{wl}}$, where $E^{\mathrm{wl}} \in 2^{A\times A}$ is an equivalence relation associated to the system (wls.1). Then, φ is a solution to (wls.1) if and only if φ is idempotent.*

Moreover, $\rho(\varphi) \leqslant \rho(\psi)$, for an arbitrary function ψ which is a solution to (wls.1).

Proof. Let φ be a solution to the system $X \cdot M_i \cdot X = M_i \cdot X$, $i \in I$, such that $Ker\varphi = E^{\mathrm{wl}}$, and let F be an equivalence relation associated to the system $X \cdot N_i = N_i$, $i \in I$, where X is an unknown matrix, and $N_i = M_i \cdot \varphi$, $i \in I$. If ψ is an idempotent function, such that $Ker\psi = F$, then, by Theorem 3.6, we have $\psi \cdot N_i = N_i$, $i \in I$, and since $\varphi \cdot N_i = N_i$, $i \in I$, by Theorem 3.5, we have $\varphi \leqslant F$. Therefore, $F(a, \varphi(a)) = Ker\psi(a, \varphi(a)) = 1$, for every $a \in A$, which implies $\psi(\varphi(a)) = \psi(a)$, $a \in A$. In conclusion, $\psi = \varphi \cdot \psi$. Further, from

$$\psi \cdot N_i = N_i \quad\Longrightarrow\quad \psi \cdot N_i \cdot \psi = N_i \cdot \psi \quad\Longrightarrow\quad \psi \cdot M_i \cdot \varphi \cdot \psi = M_i \cdot \varphi \cdot \psi \quad\Longrightarrow\quad \psi \cdot M_i \cdot \psi = M_i \cdot \psi,$$

for every $i \in I$, by (i) of Theorem 3.8, we obtain that F satisfies (7), end therefore $F \leqslant E^{\mathrm{wl}}$. The fact that $F \leqslant E^{\mathrm{wl}}$, along with $\psi \leqslant F$ lead to the conclusion that $\psi \leqslant E^{\mathrm{wl}}$, which results in $\varphi = \psi \cdot \varphi$. Finally, since

$$\varphi^2 = \varphi \cdot (\psi \cdot \varphi) = (\varphi \cdot \psi) \cdot \varphi = \psi \cdot \varphi = \varphi,$$

a function φ is idempotent.

Conversely, if $\varphi^2 = \varphi$ and $Ker\varphi = E^{\mathrm{wl}}$, then $\varphi \leqslant E^{\mathrm{wl}}$, and by (ii) of Theorem 3.8, we obtain that φ is a solution to (wls.1).

Let ψ be an arbitrary solution to (wls.1), and let $\Psi = \{\psi^k \mid k \in \mathbb{N}\}$. Obviously, all elements of Ψ are solutions to the system (wls.1). Moreover, since $(\Psi, \cdot)$ is a finite semigroup, there exist $k \in \mathbb{N}$ such that ψ^k is an idempotent function. By Lemma 3.2 and by (i) of Theorem 3.8, $\psi^k \leqslant Ker\psi^k$ and $Ker\psi^k$ is an equivalence satisfying (7). Hence, $Ker\psi^k \leqslant E^{\mathrm{wl}}$, and therefore $E^{\mathrm{wl}}(a, \psi^k(a)) = 1$, for every $a \in A$. Thus, $\varphi(a) = \varphi(\psi^k(a))$, for every $a \in A$, i.e. $\varphi = \psi^k \cdot \varphi$. In conclusion, we have that $\rho(\varphi) \leqslant \rho(\psi^k) \leqslant \rho(\psi)$. □

Let $\mathcal{M} = \{\mathcal{M}_\rangle \in \mathcal{S}^{\mathcal{A}\times\mathcal{A}} \mid \rangle \in \mathcal{I}\}$ and $\mathcal{N} = \{\mathcal{N}_| \in \mathcal{S}^{\mathcal{A}\times\mathcal{A}} \mid | \in \mathcal{J}\}$ be finite sets of matrices, and let (ls.1) and (wls.1) be systems determined by $\mathcal{M}$ and $\mathcal{N}$, respectively. If E^{l} and E^{wl} are equivalences associated to the appropriate systems, then we have the following result.

Theorem 3.10. *The following statements are true:*

(i) *There exists E^{lwl}, the greatest equivalence that is contained in E^{l} and satisfies (7).*

(ii) *Every idempotent function $\varphi \in 2^{A\times A}$ with $Ker\varphi = E^{\mathrm{lwl}}$ is a solution to both the system (ls.1) and the system (wls.1).*

Moreover, if ψ is an arbitrary function that is a solution to both systems, then $\rho(\varphi) \leqslant \rho(\psi)$.

Proof. (i) Proof of the first assertion was the subject of many papers, most of which are mentioned in this paper with regard to right invariant equivalences, congruences and bisimulations, and therefore will be left out.

(ii) The first part of the claim is the direct consequence of Lemma 3.2, Theorem 3.5 and Theorem 3.8.

Let ψ be an arbitrary function that is a solution to both the system (ls.1) and the system (wls.1). Since ψ^k is also the solution to both the system (ls.1) and the system (wls.1), for every $k \in \mathbb{N}$, there exists $k \in \mathbb{N}$ such that ψ^k is idempotent function. The rest of the proof is similar to the proof of Theorem 3.9. □

3.4. Computing solutions to the system (wls.1)

By Theorem 3.9, idempotent functions whose kernels are equal to E^{wl} are solutions to (wls.1) with the smallest rank. As it is illustrated by Example 3.3, computation of those solutions is based on reckoning the equivalence associated to (wls.1). As mentioned above, there are many algorithms for computing E^{wl}, that work for systems (wls.1) over different underlining structures. The proof of the next theorem is similar to proofs of corresponding theorems in [3, 9, 10, 14, 15, 21, 22], and will be omitted.

Theorem 3.11. *Consider the system (wls.1) determined by* $\mathcal{M} = \{\mathcal{M}_\rangle \in \mathcal{S}^{\mathcal{A}\times\mathcal{A}} \mid \rangle \in \mathcal{I}\}$*, and let* $\{E_k\}_{k\in\mathbb{N}} \subseteq 2^{A\times A}$ *be a sequence of equivalences on A inductively defined as follows*

$$E_1 = \nabla_A,$$
$$E_{k+1}(a,b) = \begin{cases} 1, & \textit{if } E_k(a,b) = 1 \textit{ and } (M_i \cdot E_k)(a,c) = (M_i \cdot E_k)(b,c), \textit{ for all } i \in I \textit{ and } c \in A \\ 0, & \textit{otherwise} \end{cases},$$

for every $a, b \in A$*. Then, the sequence* $\{E_k\}_{k\in\mathbb{N}}$ *is finite and descending. Moreover, there exists the least* $k \in \mathbb{N}$ *such that* $E_k = E_{k+m}$*, for each* $m \in \mathbb{N}$*, and* E_k *is the equivalence associated to (wls.1).*

The time complexity of this algorithm is $O(|I||A|^5)$ ([11]), if the computation cost for performing operations $+$ and $\cdot$ in S are equal to 1. However, time complexity of this algorithm is $O(|I||A|^5 c_+ c_\cdot)$, where c_+ and $c_\cdot$ denote the computation costs for performing operations $+$ and $c_\cdot$ in S respectively. Recall that $1 \leqslant c_+$ $(1 \leqslant c_\cdot)$, and therefore, $O(n) \leqslant O(nc_+)$ $(O(n) \leqslant O(nc_\cdot))$, since in most semirings the addition is derived from elementary functions, for which the computational cost is at least 1.

Let us also note that the above algorithm can be easily modified with the purpose of computing solutions to both a system (ls.1) and a system (wls.1). This modification is based on changing the starting equivalence relation E_1. It is easy to verify, that if E_1 is an equivalence other than ∇_A, then the algorithm computes the greatest equivalence relation that both satisfies (7) and is contained in E_1. In other words, if $E_1 = E^1$, where E^1 is an equivalence relation associated to a certain system (ls.1), then by Theorem 3.10, the algorithm can be used to compute functions with the smallest rank, that are solutions to both systems (ls.1) and (wls.1). Our next example illustrates those computations.

Example 3.12. *Let a semiring* $\mathbb{R}$ *be the field of real numbers and let* $A = \{a_1, a_2, a_3, a_4\}$*. Observe a system of matrix equations over* $\mathbb{R}$*, consisting of the system* (*ls*.1)*:*

$$X \cdot M = M,$$

where $M = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$, $X \in 2^{A\times A}$ *is an unknown matrix, and the system* (*wls*.1)*:*

$$X \cdot N_i \cdot X = N_i \cdot X,$$

where $I = \{1, 2\}$*, and* N_1*,* N_2 *are matrices:*

$$N_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{bmatrix}, \quad N_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

Since, $E_1 = E^1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$*, in order to compute* E_2*, we have to compute* $N_i \cdot E_1$*, for* $i \in \{1, 2\}$*.*

$$N_1 \cdot E_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & -1 & -1 \end{bmatrix},$$

$$N_2 \cdot E_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & -1 & -1 \end{bmatrix}.$$

By Theorem 3.11, $E_2(a_i,a_j) = 1$ iff $E_1(a_i,a_j) = 1$ and $a_i(N_i \cdot E_1) = a_j(N_i \cdot E_1)$, for $i \in \{1,2\}$. Therefore, since $a_1(N_i \cdot E_1) = a_2(N_i \cdot E_1)$, for $i \in \{1,2\}$, and $E_1(a_1,a_2) = 1$, we have $E_2(a_1,a_2) = 1$. In a similar way, we obtain $E_2(a_3,a_4) = 1$. It is also clear that $E_2(a_1,a_3) = 0$. In conclusion,

$$E_2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

Obviously, since $E_2 = E_1$, the algorithm stops and leads us to the conclusion that E_2 is the greatest equivalence that contains E^1 and satisfies (7). Finally, by Theorem 3.9, and Theorem 3.8, functions with the smallest ranks that are solutions to the system consisting of systems (ls.1) and (wls.1) are idempotent functions whose kernels are equal to E_2:

$$\varphi_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \ \varphi_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \ \varphi_3 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \ \varphi_4 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

However, the algorithm, given in Theorem 3.11, can be improved. The improvement is based on using successive refinements of an initial partition determined by a starting equivalence E_1. Refinements computed by the algorithm ultimately converge to the greatest equivalence satisfying (7) contained in E_1. In other words, the improvement can be done by adapting the well known partition refinement algorithms given by Kanellakis and Smolka [8, 19], Paige and Tarjan [20], etc.

With this purpose, let us introduce the following notions: *Partition* π of a nonempty set A, is a set $\pi = \{B_1, B_2, \ldots, B_k\}$, of nonempty mutually disjoint sets, such that $\bigcup_{i=1}^{k} B_i = A$. Elements of a given partition are called *blocks*. Let us recall that every partition π of A defines an equivalence relation on A, denoted as E_π, in the following way

$$E_\pi(a,b) = 1 \qquad \Longleftrightarrow \qquad a,b \in B_i, \tag{10}$$

for some $i \in \{1,2,\ldots,k\}$. Conversely, an equivalence relation E on a set A defines a partition of that set, denoted as π_E, as follows

$$\pi_E = \{B_1, B_2, \ldots, B_k\}, \tag{11}$$

where for every $i \in \{1,\ldots,k\}$ there exists $a \in A$, such that $B_i = \{b \in A \mid E(a,b) = 1\}$, and for every $a \in A$ there exists $B_i \in \pi_E$ that contains a. Partition π_E is a *refinement* of a partition π_F, if an equivalence E is contained in F.

Let $\mathcal{M} = \{\mathcal{M}\rangle \in \mathcal{S}^{\mathcal{A}\times\mathcal{A}} \mid \rangle \in \mathcal{I}\}$ be an arbitrary finite set of matrices. Block B_j is a *splitter* for a block B_i w.r.t. $\mathcal{M}$, if for some $a,b \in B_i$, there exists $l \in I$ such that $\sum_{c\in B_j} M_l(a,c) \neq \sum_{c \in B_j} M_l(b,c)$. In case when a block B_i has a splitter w.r.t. $\mathcal{M}$ it can be split into two sets

$$B_i^{a,1} = \Big\{c \in B_i \mid \sum_{d\in B_j} M_l(c,d) = \sum_{d\in B_j} M_l(a,d), \text{ for every } B_j \in \pi_E\Big\}, \quad B_i^{a,2} = B_i \setminus B_i^{a,1}.$$

for any $a \in B_i$, and any $l \in I$. This splitting results in a new partition $\pi_{E'} = \{B_1, \ldots, B_i^{a,1}, B_i^{a,2} \ldots, B_k\}$, which is a refinement of π_E. Partition π is *stable w.r.t.* $\mathcal{M}$, if no block of π has splitters.

Lemma 3.13. *Let $\mathcal{M} = \{\mathcal{M}\rangle \in \mathcal{S}^{\mathcal{A}\times\mathcal{A}} \mid \rangle \in \mathcal{I}\}$ be an arbitrary finite set of matrices and E an equivalence relation on A. Partition π_E is stable w.r.t. $\mathcal{M}$ if and only if E satisfies (7).*

Proof. Partition π_E is stable w.r.t. $\mathcal{M}$ iff for every $B_i, B_j \in \pi_E$, block B_j is not a splitter for a block B_i, or equivalently

$$\sum_{c\in B_j} M_l(a,c) = \sum_{c\in B_j} M_l(b,c),$$

for every $a, b \in B_i, B_j \in \pi_E$ and every $l \in I$. Thus, if $E(a, b) = 1$, i.e. if $a, b \in B_i$, for some $B_i \in \pi_E$, then for every $c \in A$ and $i \in I$ we have

$$
\begin{aligned}
(M_i \cdot E)(a,c) &= \sum_{d\in A} M_i(a,d)\cdot E(d,c) = \sum_{d\in A,\ E(d,c)=1} M_i(a,d) = \sum_{d\in B_j} M_i(a,d) = \sum_{d\in B_j} M_i(b,d) \\
&= \sum_{d\in A,\ E(d,c)=1} M_i(b,d) = \sum_{d\in A} M_i(b,d)\cdot E(d,c) = (M_i\cdot E)(b,c),
\end{aligned}
$$

where $B_j \in \pi_E$ is a block that contains c.

Converse assertion can be easily derived in a similar manner. □

Prior to present the Kanellakis and Smolka's algorithm adapted to work with weights over semirings, let us recall some facts and notions related to both the original and our algorithm. The basic idea that underlines the algorithm given by Kanellakis and Smolka is to iterate splitting of blocks of a starting partition π w.r.t. some finite set of matrices over Boolean semiring, until no further refinement of the current partition is possible. The resulting partition is often called the *coarsest stable partition* and coincides with the greatest equivalence contained in E_π satisfying (7) (Theorem 3.14). Kanellakis and Smolka's algorithm plays an important role in the study of labelled systems, showing that the problem of deciding bisimilarity of labelled transition systems can be solved efficiently. Since our version of this algorithm works with weights over semirings we will introduce first the notion of weighted labelled transition system.

Let A be an arbitrary finite set and $\mathcal{M} = \{\mathcal{M}_\rangle \in \mathcal{S}^{\mathcal{A}\times\mathcal{A}} \mid \rangle \in \mathcal{I}\}$ be a finite set set of matrices. *Weighted labelled transition system determined by* $\mathcal{M}$ (abbr. *WLTS determined by* $\mathcal{M}$) is a triple (A, I, w), where A is a *set of states*, I is a *set of labels* and $w : A \times I \times A \to S$ is a *weight function* defined by

$$
w(a,i,b) = M_i(a,b), \tag{12}
$$

for every $a, b \in A$ and $i \in I$. In case $w(a, i, b) \neq 0$, we say that there is a *transition* from a state a to a state b labeled by i with weight $s = w(a, i, b)$, denoted as (a, i, s, b). In that case, a is an *outging* state and b is a *target* state. Otherwise, i.e if $w(a, i, b) = 0$, we say that there is no transition from a state a to b labeled by i. Obviously, WLTS are an extension of the concept of labelled transition system, and for other notions and notations concerning WLTS refer to [17].

Pseudo code given by Algorithm 1 presents the crucial function of the adapted Kanellakis and Smolka's algorithm, called Split(B, i, π). The purpose of this procedure is to detect if the block $B \in \pi$ has a splitter w.r.t. a label $i \in I$. With this purpose the procedure chooses a state $a \in B$ and compares sums $\sum_{d\in C} w(a, i, d)$ and $\sum_{d\in C} w(b, i, d)$, where $C \in \pi$ and $b \in B$. In case when a splitter exists, i.e. the sums are not equal, the procedure returns blocks $B^{a,1}$ and $B^{a,2}$ as a result of the splitting. Otherwise, block B is returned. Since the efficiency of a splitting is crucial for the complexity of the algorithm, we will discuss it in more detail. In order to efficiently compare sums $\sum_{d\in C} w(a, i, d)$ and $\sum_{d\in C} w(b, i, d)$, where $C \in \pi$ and $b \in B$, one needs to order transitions outgoing from all states belonging to the block B. Prior to discuss that ordering, we will assume that the set of labels I is linearly ordered. Since I is the index set, the easiest way is to set $I = \{1, 2, \ldots, k\}$, where $k \in \mathbb{N}$. In addition, we impose an ordering to the blocks of π. The ordering of transitions will be done in the following way: transitions are ordered by their labels, while transitions with the same label are ordered by the blocks containing their target states.

Algorithm 1

```
function SPLIT($B, i, \pi$)    ▷ Function that returns split block $B$ of $\pi$ w.r.t. transitions labeled by i
    choose some $a \in B$
    $B^{a,1} := \emptyset, B^{a,2} := \emptyset$
    for all $b \in B$ do
        $splitting := false$
        for all $C \in \pi$ do
            if $\sum_{d \in C} w(a, i, d) \neq \sum_{d \in C} w(b, i, d)$ then $splitting := true$    ▷ C is a splitter for B
            end if
        end for
        if $splitting$ then $B^{a,2} := B^{a,2} \cup \{b\}$
        else $B^{a,1} := B^{a,1} \cup \{b\}$
        end if
    end for
    if $B^{2,a} = \emptyset$ then return $\{B\}$
    else return $\{B^{a,1}, B^{a,2}\}$
    end if
end function
```

When a block is split, the ordering of transitions whose target states belong to that block can be contravened. Therefore, one has to sort transitions outgoing from all states of a given block before attempting to split it. Procedure *TransitionsSort*(B, i) uses lexicographic sorting to reorder the *i*-labelled transitions outgoing from the block B, e.g. using the classic algorithm from Aho, Hopcroft and Ullman, 1974.

Algorithm 2

```
$\pi := \pi_E$
$changed := true$
while $changed$ do
    $changed := false$
    $\pi_c := \pi$
    for all $B \in \pi_c$ do
        for all $i \in I$ do
            $TransitionsSort(B, i)$
            if $Split(B, i, \pi_c) \neq \{B\}$ then
                $\pi := (\pi \setminus \{B\}) \cup Split(B, i, \pi_c)$
                $changed := true$
                Break
            end if
        end for
    end for
end while
```

As noted above, the purpose of Algorithm 2 is to perform successive refinements of partitions, starting with the partition π_E, by continuous splitting of their blocks, until no further refinements are possible.

Theorem 3.14. *Let A be a finite set, $\mathcal{M} = \{\mathcal{M}_\rangle \in \mathcal{S}^{\mathcal{A}\times\mathcal{A}} \mid \rangle \in \mathcal{I}\}$ a finite set set of matrices, and π_E a partition on A. If π_F is the coarsest stable partition, then F is the greatest equivalence contained in E that satisfies (7).*

Moreover, if $|A| = m$ and n is the number of transitions of a WLTS determined by $\mathcal{M}$, then Algorithm 2. takes $O(mnc_+)$ time.

Proof. Obviously, π_F is stable partition, and since π_F is a refinement of π_E, by Lemma 3.13, the equivalence

F is contained in E and satisfies (7). Let us prove that F is the greatest equivalence contained in E satisfying (7).

Evidently, $F \leqslant E$. Let G be an arbitrary equivalence relation contained in E satisfying (7), and let $k \in \mathbb{N} \cup \{0\}$ be the number of repetition of the main loop of Algorithm 2. Denote by F_l, $l \in \{0, 1, \dots, k\}$, an equivalence relation obtained by Algorithm 2, after l repetition of the main loop. Since $F_0 = E$ and $G \leqslant E$, there exists the greatest $s \in \{0, 1, \dots, k\}$, such that $G \subseteq F_s$. If $\pi_{F_s} = \{B_1, B_2, \dots, B_p\}$ and $s < k$ then π_{F_s} is not stable, and therefore there exists a block $B_i \in \pi_{F_s}$ that can be split into blocks $B_i^{a,1}$ and $B_i^{a,2}$, for some $a \in B_i$. Let $B_l \in \pi_{F_s}$ be a splitter for B_i and let $\pi_{F'_s} = \{B_1, B_2, \dots, B_i^{a,1}, B_i^{a,2}, \dots, B_p\}$. Since $G \leqslant F_s$, we have $B_l = \bigcup_{t=1}^{m} C_t$, for some blocks $C_t \in \pi_G$, $t \in \{1, 2, \dots, m\}$. If $G(b, c) = 1$, for some $b, c \in A$, the fact that G satisfies (7) results in the following

$$\begin{aligned}\sum_{d \in B_l} w(b, j, d) &= \sum_{t=1}^{m} \sum_{d \in C_t} w(b, j, d) = \sum_{t=1}^{m} \sum_{d \in C_t} M_j(b, d) = \sum_{t=1}^{m} (M_j \cdot G)(b, d_t) \\ &= \sum_{t=1}^{m} (M_j \cdot G)(c, d_t) = \sum_{t=1}^{m} \sum_{d \in C_t} M_j(c, d) = \sum_{t=1}^{m} \sum_{d \in C_t} w(c, j, d) \\ &= \sum_{d \in B_l} w(c, j, d),\end{aligned}$$

for any $j \in I$, where $d_t \in C_t$, for every $t \in \{1, 2, \dots, m\}$. Thus, if $\sum_{d \in B_l} w(a, j, d) = \sum_{d \in B_l} w(b, j, d)$, then $b, c \in B_i^{a,1}$, otherwise, $b, c \in B_i^{a,2}$. Both cases lead to $F'_s(b, c) = 1$, and therefore $G \leqslant F'_s$. From the previous consideration and the fact that F_{s+1} is obtained from F_s by splitting some of the blocks of π_{F_s}, one conclude that $G \leqslant F_{s+1}$, which contradicts to the assumption that $s < k$. Accordingly, $s = k$, i.e. $G \leqslant F_k = F$.

The main loop (while loop) of Algorithm 2 is repeated at most m times. Within one iteration of the main loop, procedure *Split* is called for each block at most once for each label $i \in I$. In turn, procedure *Split*, while calculating appropriate sums, considers each transition of every state in the block at most once. Therefore, the calls to split within one iteration of the main loop take $O(nc_+)$ time. The calls to *TransitionsSort* collectively take $O(|I| + n)$ time, or $O(n)$ when the set of labels is bounded. Since $O(n) \leqslant O(nc_+)$, the running time of Algorithm 2 is $O(mnc_+)$. □

Let us observe that the time complexity of the algorithm given by Theorem 3.11 is greater or equal to $O(m^3 nc_+ c_\cdot)$ since $n \leqslant |I||A|^2$, and therefore is less efficient then one given in the previous theorem.

4. Applications in a state reduction of weighted finite automata

Let S be a semiring and X an alphabet. A *weighted finite automaton* (WFA, for short) over X and S, is a quadruple $\mathcal{A} = (A, X, \sigma^A, \delta^A, \tau^A)$, where A is a finite nonempty *set of states*, $\delta^A : A \times X \times A \to S$ is a *transition function*, $\sigma^A \in S^A$ is an *initial vector* and $\tau^A \in S^A$ is a *final vector*. For each $x \in X$ we define a *transition matrix* $\delta_x \in S^{A \times A}$ with $\delta_x^A(a, b) = \delta^A(a, x, b)$, for all $a, b \in A$. In addition, for every $u \in X^*$ we define δ_u^A, an $A \times A$ matrix over S inductively as follows: $\delta_\varepsilon = I_A$, and for every $u \in X^*$ and $x \in X$ we set $\delta_{ux} = \delta_u \cdot \delta_x$. Let us also note that σ^A is considered as a row, and τ^A as a column vector.

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

$$\begin{aligned}&[\![\mathcal{A}]\!](\varepsilon) = \sigma^A \cdot \tau^A, \\ &[\![\mathcal{A}]\!](u) = \sigma^A \cdot \delta_{x_1}^A \cdot \delta_{x_2}^A \cdot \dots \cdot \delta_{x_n}^A \cdot \tau^A,\end{aligned} \tag{13}$$

for any $u = x_1 x_2 \dots x_n \in X^+$, where $x_1, x_2, \dots, x_n \in X$ and ε is an empty word. Weighted finite automata $\mathcal{A}$ and $\mathcal{B}$ are *isomorphic* if there exists a bijection $\alpha : A \to B$, such that

$$\sigma^A(a) = \sigma^B(\alpha(a)), \quad \delta_x^A(a, b) = \delta_x^B(\alpha(a), \alpha(b)), \quad \tau^A(a) = \tau^B(\alpha(a)) \tag{14}$$

for every $a, b \in A$, and every $x \in X^*$. Weighted finite automata $\mathcal{A}$ and $\mathcal{B}$ are *equivalent* if they have the same behaviour, i.e. if the following stands

$$[\![\mathcal{A}]\!](u) = [\![\mathcal{B}]\!](u), \tag{15}$$

for every $u \in X^*$.

Let $\mathcal{A} = (A, X, \sigma^A, \delta^A, \tau^A)$ be an arbitrary WFA. Let l be an $A \times B$ matrix and r be an $B \times A$ matrix, for an arbitrary finite set B. Weighted finite automaton $\mathcal{B} = (B, X, \delta^B, \sigma^B, \tau^B)$ is an (l, r)*-transformation* of a weighted finite automaton $\mathcal{A}$ if B is the set of states of $\mathcal{B}$ and matrices δ^B_x, δ^B_y, σ^B and τ^B are defined by

$$\begin{aligned} \delta^B_x &= r \cdot \delta^A_x \cdot l, \\ \sigma^B &= \sigma^A \cdot l, \\ \tau^B &= r \cdot \tau^A. \end{aligned} \tag{16}$$

The following result is presented and proven in ([23]), and also stand for weighted finite automata.

Theorem 4.1. *Let $\mathcal{A} = (A, X, \sigma^A, \delta^A, \tau^A)$ be a weighted finite automaton and let $\mathcal{B} = (B, X, \sigma^B, \delta^B, \tau^B)$ be an (l, r)-*transformation *of $\mathcal{A}$ for a $B \times A$ matrix l and an $A \times B$ matrix r, where B is a finite set. Automata $\mathcal{A}$ and $\mathcal{B}$ are equivalent if and only if a matrix $Q = l \cdot r$ is a solution to the system of matrix equations*

$$\begin{aligned} &\sigma^A \cdot \tau^A = \sigma^A \cdot U \cdot \tau^A, \\ &\sigma^A \cdot \delta^A_{x_1} \cdot \delta^A_{x_2} \cdot \dots \cdot \delta^A_{x_n} \cdot \tau^A = \sigma^A \cdot U \cdot \delta^A_{x_1} \cdot U \cdot \delta^A_{x_2} \cdot U \cdot \dots \cdot U \cdot \delta^A_{x_n} \cdot U \cdot \tau^A, \end{aligned} \tag{17}$$

for all $n \in \mathbb{N}$ and $x_1, x_2, \dots, x_n \in X$, where U is an uknown $m \times m$ matrix.

In the sequel the system (17) will be called the *general system*.

State reduction of a weighted finite automaton $\mathcal{A}$ by using arbitrary (l, r)-transformation of $\mathcal{A}$ is performed in the following way. Efficiently computing a solution Q to the general system, whose rank $\rho(Q)$ is as small as possible. Further, efficiently decomposing Q into k-decomposition (l, r), where k is also as small as possible (with $k = \rho(Q)$, if possible). Finally, we simply construct an (l, r)-transformation of $\mathcal{A}$, by computing all its transition matrices. An (l, r)-transformation of $\mathcal{A}$ is a weighted finite automaton that is, by Theorem 4.1, equivalent to $\mathcal{A}$ and has k states.

However, implementing the above state reduction method, includes dealing with several issues. The first one is solving the general system that consists of infinitely many equations. Thus, instead of solving the general system we solve its instance, i.e. a system whose solutions are also solutions to the general. With this purpose, for a weighted finite automaton $\mathcal{A} = (A, X, \delta^A, \sigma^A, \tau^A)$ over X and S, we introduce a *system right-generated by* $\mathcal{A}$, consisting of one (ls.1) system and one (wls.1) system of matrix equation as follows

$$\begin{aligned} &U \cdot \tau^A = \tau^A, \\ &U \cdot \delta^A_x \cdot U = \delta^A_x \cdot U, \quad x \in X, \end{aligned} \tag{18}$$

where $U \in S^{A \times A}$ is an unknown matrix. According to Theorem 3.10., Theorem 3.14. and Example 3.3, functions with the smallest ranks that are solutions to the above system, can be efficiently computed. Moreover, if φ is such a function, its k-decomposition (l^φ, r^φ) can be easily obtained, and by Theorem 3.4, k is a good estimation of $\rho(\varphi)$.

Let us note that, for all the results regarding the system (18), there are corresponding results for the system

$$\begin{aligned} &\sigma^A \cdot U = \sigma^A, \\ &U \cdot \delta^A_x \cdot U = U \cdot \delta^A_x, \quad x \in X, \end{aligned} \tag{19}$$

where $U \in S^{A \times A}$ is an unknown matrix. The above system is called a *system left-generated by* $\mathcal{A}$, and all statements and proofs regarding (19) will be omitted.

At the end of this section, let us discuss about one more question regarding the state reduction of weighted finite automata using (l, r)-transformations. Namely, as shown in Example 3.3 and Example 3.12, for a given equivalence relation E there are exactly $|B_1| \cdot |B_1| \cdot \ldots \cdot |B_k|$ idempotent functions whose kernels are equal to E, where $B_1, B_2, \ldots, B_k$ are all the blocks of π_E. Consequently, there can be many different functions with the smallest ranks that are solutions to the system (18), and accordingly, there can be many different (l, r)-transformations of a given weighted finite automaton $\mathcal{A}$. Our next result pertains to the relations between those (l, r)-transformations.

Theorem 4.2. *Let $\mathcal{A} = (A, X, \delta^A, \sigma^A, \tau^A)$ be a weighted finite automaton over an alphabet X and a semiring S. If $\varphi, \psi \in 2^S$ are functions with the smallest ranks that are solutions to the system right-generated by $\mathcal{A}$, then (l^φ, r^φ)-transformation and (l^ψ, r^ψ)-transformation are isomorphic.*

Proof. Theorem 3.10 implies $Ker\varphi = Ker\psi$. If E is a kernel of φ and ψ, then, using Lemma 3.7, we have the following

$$(r^\varphi \cdot \delta_x \cdot l^\varphi)(Ea, Eb) = (\delta_x \cdot l^\varphi)(\varphi(a), Eb) = (\delta_x \cdot E)(\varphi(a), b),$$

for every $Ea, Eb \in A/E$, and $x \in X$. In a similar manner we obtain

$$(r^\psi \cdot \delta_x \cdot l^\psi)(Ea, Eb) = (\delta_x \cdot l^\psi)(\psi(a), Eb) = (\delta_x \cdot E)(\psi(a), b),$$

for every $Ea, Eb \in A/E$, and $x \in X$. On the other hand, φ and ψ are idempotent functions, and therefore $E(a, \varphi(a)) = E(a, \psi(a)) = 1$, for every $a \in A$. As a result, we have that $E(\varphi(a), \psi(a)) = 1$, for every $a \in A$. Since E satisfies a condition (7), we also have

$$E(\varphi(a), \psi(a)) = 1 \quad \Longrightarrow \quad (\delta_x \cdot E)(\varphi(a), b) = (\delta_x \cdot E)(\psi(a), b),$$

for every $a, b \in A$ and $x \in X$. In conclusion, we have that $r^\varphi \cdot \delta_x \cdot l^\varphi = r^\psi \cdot \delta_x \cdot l^\psi$.

It is easy to verify that $\sigma^A \cdot l^\varphi = \sigma^B \cdot l^\psi$ and $r^\psi \cdot \sigma^A = r^\psi \cdot \sigma^B$. In conclusion, (l^φ, r^φ)-transformation and (l^ψ, r^ψ)-transformation are isomorphic. □

In the next example we illustrate the above described state reduction method. Prior to do that, let us recall that a WFA $\mathcal{A}$ can be presented by the 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 $x/\delta^A(a, x, b)$, for any $x \in X$. Also, for each node a we represent its initial value $\sigma^A(a)$ by drawing the ingoing arrow to a labelled by $\sigma^A(a)$, and represent $\tau^A(a)$ by double-circling node a, provided $\tau^A(a) \neq 0$, 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. In addition, we do not explicitly show the label on the ingoing arrow or double-circled node if it is equal to 1.

Example 4.3. *Let a semiring $\mathbb{R}$ be the field of real numbers and let $\mathcal{A} = (A, X, \sigma^A, \delta^A, \tau^A)$ be a WFA automaton over $X = \{x, y\}$ and $\mathbb{R}$, whose graph is presented by Figure 1.*

Matrices δ^A_x, δ^A_y, σ^A and τ^A are given as follows

$$\delta^A_x = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \delta^A_y = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \quad \sigma^A = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}, \quad \tau^A = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}.$$

By Theorem 3.10, the first step in computation of function with the smallest rank that is a solution to the system right-generated by $\mathcal{A}$ is to compute an equivalence E^1 associated to the matrix equation $U \cdot \tau^A = \tau^A$, where U is an unknown matrix. Obviously,

$$E^1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

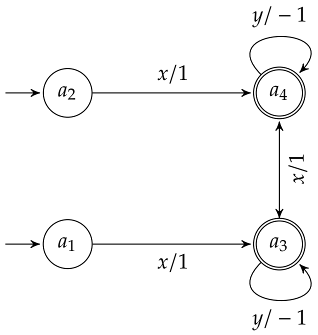

Figure 1: Transition graph of the WFA $\mathcal{A}$

The second step is to compute E, the greatest equivalence contained in E^1 that satisfies (7). By using iterative method of Theorem 3.11, one has to compute an equivalence E_2, using products $\delta_x^A \cdot E_1$ and $\delta_y^A \cdot E_1$, where $E_1 = E^1$.

$$\delta_x^A \cdot E_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix},$$

$$\delta_y^A \cdot E_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & -1 & -1 \end{bmatrix}.$$

It is easy to verify (see Theorem 3.11 and Example 3.12.) that

$$E_2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

Since $E_2 = E_1$, the algorithm stops, and functions with the smallest ranks that are solutions to the system right-generated by $\mathcal{A}$ are

$$\varphi_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \varphi_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \varphi_3 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \varphi_4 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

By Theorem 3.11, the (l,r)-transformation of $\mathcal{A}$ does not depend on the choice of φ_i, $i \in \{1,2,3,4\}$, thus we decompose φ_1. Obviously, $\varphi_1 = l^{\varphi_1} \cdot r^{\varphi_1}$, where

$$l^{\varphi_1} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}, \quad \textit{and} \quad r^{\varphi_1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

The final step is building the resulting automaton $\mathcal{B} = (\mathcal{B}, \mathcal{X}, \sigma^{\mathcal{B}}, \delta^{\mathcal{B}}, \tau^{\mathcal{B}})$, by calculating matrices δ^B, σ^B and τ^B:

$$\delta_x^B = r^{\varphi_1} \cdot \delta_x^A \cdot l^{\varphi_1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix},$$

$$\delta_y^B = r^{\varphi_1} \cdot \delta_y^A \cdot l^{\varphi_1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix},$$

$$\sigma^B = \sigma^A \cdot l^{\varphi_1} = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 \end{bmatrix},$$

$$\tau^B = r^{\varphi_1} \cdot \tau^A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

In conclusion, the $(l^{\varphi_1}, r^{\varphi_1})$-transformation of $\mathcal{A}$ is the weighted finite automaton $\mathcal{B}$ with two states whose graph is presented by Figure 2.

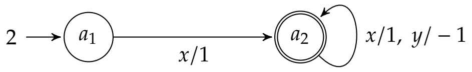

Figure 2: Transition graph of the WFA $\mathcal{B}$

5. Concluding remarks

In this paper we have dealt with functions over semirings that are solutions to the systems (ls.1) and (wls.1) , with the special attention on solutions whose ranks are as small as possible. We have proved the existence of functions that are solutions to the systems (ls.1) and (wls.1) with the smallest ranks, and gave their characterization: as idempotent functions whose kernels satisfy certain conditions. Those results have led us to the method for their construction. The method is based on computing particular equivalences, and afterwards on computing idempotent functions whose kernels are exactly above-mentioned equivalences. Regarding the system (wls.1), it appeared that kernels of solutions with the smallest ranks are equivalences that satisfy condition (7), also known as right-invariant, congruences, bisimulations, coarsest stable, etc.

Using the fact that algorithms for computing equivalences satisfying (7) are based on partition refinement algorithms, we have built one by adapting the well known Kanellakis and Smolka's partition refinement algorithm. Our algorithm works for matrices over semirings and its time complexity is less then the time complexity of all known algorithms performing that task.

In the last section of our paper, we have implemented some of our results in a state reduction problem. Namely, we have proved that decompositions of solutions to the system (wls.1) can be used for constructing (l, r)-transformations of WFA. We have introduced a state reduction method, based on (l, r)-transformations, and obtained state reduction of a starting WFA equal to those given by Peter Buchholz in [3]. In particular, WFA produced by our method is isomorfic to one computed in [3]. However, since the rank of solutions to the system (17) play the crucial role in a state reduction, our method gives potentially better results. More precisely, we have proved that if there exists a semiring S such that $\rho(I_A) < |A|$, for some finite set A, then our method produces WFA with smaller number of states than the aggregated automaton introduced in [3]. To the best of our knowledge, the existence of a semiring S such that $\rho(I_A) < |A|$, for some finite set A, is an open problem.

References

[1] R. Bělohlávek, *Optimal decompositions of matrices with entries from residuated lattices*, Journal of Logic and Computation 22 (2012) 1405–1425.

[2] R. Bělohlávek, J. Konecny, *Row and Column Spaces of Matrices Over Residuated Lattices*, Fundamenta Informaticae 4 (2012) 279–295.

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

[4] H. H. Cho, *Regular fuzzy matrices and fuzzy equations*, Fuzzy Sets and Systems 105 (1999) 445–451.

[5] Ćirić M, Ignjatović J, Bašić M, Jančić I (2014) Nondeterministic automata: equivalence, bisimulations, and uniform relations. Inf Sci 261:185–218

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

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

[8] R. Cleaveland, O. Sokolsky,*Equivalence and Preorder Checking for Finite-State Systems*, In: J.A. Bergstra, A. Ponse, S. Smolka (eds.), Elsevier Science, Handbook of Process Algebra (2001) 391—424.

[9] M. Ćirić, A. Stamenković, J. Ignjatović, T. Petković, *Factorization of fuzzy automata*, in: E. Csuhaj-Varju, and Z. Ésik (eds.), FCT 2007, Lecture Notes in Computer Science 4639 (2007) 213–225.

[10] 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.

[11] Damljanović N, Ćirić M, Ignjatović, J (2014) Bisimulations for weighted automata over an additively idempotent semiring. Theor Comput Sci 534:86–100

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

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

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

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

[16] Jančić Z, Micić I, Ignjatović J, Ćirić M, *Further improvements of determinization methods for fuzzy finite automata*, Fuzzy Sets Syst 301 (2016) 79–102.

[17] B. Klin, *Structural operational semantics for weighted transition systems*, In: J. Palsberg (editor), Semantics and Algebraic Specification, Lecture Notes in Computer Science 5700 (2009) 121-–139.

[18] K. H. Kim and F. W. Roush, *Generalized Fuzzy Matrices*, Fuzzy Sets and Systems 4 (1980) 293–315.

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

[20] R. Paige, R. E. Tarjan, *Three partition refinement algorithms*, SIAM Journal on Computing 16 (1987) 973–989.

[21] T. Petković, *Congruences and homomorphisms of fuzzy automata*, Fuzzy Sets and Systems 157 (2006) 444–458.

[22] A. Stamenković, M. Ćirić, *Construction of fuzzy automata from fuzzy regular expressions*, Fuzzy Sets and Systems 199 (2012) 1–27.

[23] A. Stamenković, M. Ćirić, M. Bašić, *Ranks of fuzzy matrices. Applications in state reduction of fuzzy automata*, Fuzzy Sets and systems 333 (2018), 124–139.

[24] A. Stamenković, M. Ćiric, J. Ignjatović, *Reduction of fuzzy automata by means of fuzzy quasi-orders*, Information Sciences 275 (2014) 168–198.

[25] 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.