State reduction of weighted automata using certain equivalences

Aleksandar Stamenković

Department of Computer Science
Faculty of Sciences and Mathematics
University of Niš, Serbia,
aca@pmf.ni.ac.rs

Weighted Automata: Theory and Applications,
April 20-23, 2021

State reduction problem. Formulation

For a starting automaton $\mathcal{A}$, computing smaller automaton that is **equivalent** to the starting one.

State reduction methods. Ordinary and fuzzy case

Computing and merging indistinguishable states.

Indistinguishability is modeled by particular (fuzzy) equivalences or (fuzzy) quasi-orders.

(Fuzzy) Quasi-order – idempotent matrix containing the identity matrix on the set of states of the automaton.

(Fuzzy) Equivalence – symmetric (fuzzy) quasi-order.

State reduction construction

In both cases the state reduction is done by constructing new automata – factor (fuzzy) automata and afterset (fuzzy) automata – by multiplying transition matrices of a starting automaton with appropriate (fuzzy) equivalences ((fuzzy) quasi-orders).

State reduction problem. Formulation

For a starting automaton $\mathcal{A}$, computing smaller automaton that is **equivalent** to the starting one.

State reduction methods. Ordinary and fuzzy case

Computing and merging indistinguishable states.

Indistinguishability is modeled by particular (fuzzy) equivalences or (fuzzy) quasi-orders.

(Fuzzy) Quasi-order – idempotent matrix containing the identity matrix on the set of states of the automaton.

(Fuzzy) Equivalence – symmetric (fuzzy) quasi-order.

State reduction construction

In both cases the state reduction is done by constructing new automata – factor (fuzzy) automata and afterset (fuzzy) automata – by multiplying transition matrices of a starting automaton with appropriate (fuzzy) equivalences ((fuzzy) quasi-orders).

State reduction problem. Formulation

For a starting automaton $\mathcal{A}$, computing smaller automaton that is **equivalent** to the starting one.

State reduction methods. Ordinary and fuzzy case

Computing and merging indistinguishable states.

Indistinguishability is modeled by particular (fuzzy) equivalences or (fuzzy) quasi-orders.

(Fuzzy) Quasi-order – idempotent matrix containing the identity matrix on the set of states of the automaton.

(Fuzzy) Equivalence – symmetric (fuzzy) quasi-order.

State reduction construction

In both cases the state reduction is done by constructing new automata – factor (fuzzy) automata and afterset (fuzzy) automata – by multiplying transition matrices of a starting automaton with appropriate (fuzzy) equivalences ((fuzzy) quasi-orders).

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.*

State reduction method. New approach

Instead of computing indistinguishable states, solving a **general system** – computing a solutions whose **rank** is as small as possible.

For a solution Q to the general system, finding as good as possible **decomposition** (L, R) of Q (**rank decomposition**, if possible).

Transforming transition matrices of the starting automaton using L and R.

Our goal

Implementing the new method in a state reduction of weighted automata.

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.*

State reduction method. New approach

Instead of computing indistinguishable states, solving a **general system** – computing a solutions whose **rank** is as small as possible.

For a solution Q to the general system, finding as good as possible **decomposition** (L, R) of Q (**rank decomposition**, if possible).

Transforming transition matrices of the starting automaton using L and R.

Our goal

Implementing the new method in a state reduction of weighted automata.

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.***

State reduction method. New approach

Instead of computing indistinguishable states, solving a **general system** – computing a solutions whose **rank** is as small as possible.

For a solution Q to the general system, finding as good as possible **decomposition** (L, R) of Q (**rank decomposition**, if possible).

Transforming transition matrices of the starting automaton using L and R.

Our goal

Implementing the new method in a state reduction of weighted automata.

Definitions

A *semiring* $\mathcal{S}$ – a set S equipped with two binary operations $+$ and $\cdot$, and two constants $\mathbf{0}$ and $\mathbf{1}$, such that

- $(S, +, \mathbf{0})$ is a commutative monoid,
- $(S, \cdot, \mathbf{1})$ is a monoid,
- $\cdot$ is distributive with respect to $+$,
- $\mathbf{0} \cdot k = k \cdot \mathbf{0} = \mathbf{0}$ for every $k \in S$.

A matrix is *over a semiring* – a matrix with the entries (or the elements) belonging to a semiring. In this talk, a matrix – a matrix over a semiring.

Matrix pair (B, C) is a *decomposition* of a matrix A if $A = B \cdot C$.

The *rank* (Schein rank) of a matrix A, of type $m \times n$, denoted $\rho(A)$ – the least integer k such that $A = B \cdot C$ for some $m \times k$ matrix B and some $k \times n$ matrix C. In that case – (B, C) is a *rank decomposition of A*.

Relations

Matrices taking values in the set $\{0, 1\}$, where $0, 1 \in S$, are called *relations*.

Relation on a set A ,i.e. a matrix $E \in 2^{A\times A}$ is:

- *Reflexive*, if $E(a, a) = 1$,
- *Symmetric*, if E is symmetric matrix,
- *Transitive*, if $E(a, b) = 1$ and $E(b, c) = 1$ implies $E(a, c) = 1$,

for all $a, b, c \in A$.

Reflexive, symmetric and transitive relation on A is called an *equivalence relation on A*.

Functions

Relation $\varphi \in 2^{A\times B}$ is a *function* 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 $b, c \in B$.

If φ is a function, *kernel* of φ is an equivalence relation $Ker\varphi$ on A, defined by:

$$Ker\varphi(a, b) = 1 \quad \Leftrightarrow \quad \varphi(a) = \varphi(b), \tag{1}$$

for every $a, b \in A$.

Relations

Matrices taking values in the set $\{0,1\}$, where $0,1 \in S$, are called *relations*.

Relation on a set A ,i.e. a matrix $E \in 2^{A\times A}$ is:

- *Reflexive*, if $E(a,a) = 1$,
- *Symmetric*, if E is symmetric matrix,
- *Transitive*, if $E(a,b) = 1$ and $E(b,c) = 1$ implies $E(a,c) = 1$,

for all $a,b,c \in A$.

Reflexive, symmetric and transitive relation on A is called an *equivalence relation on* A.

Functions

Relation $\varphi \in 2^{A\times B}$ is a *function* 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 $b,c \in B$.

If φ is a function, *kernel* of φ is an equivalence relation $Ker\varphi$ on A, defined by:

$$Ker\varphi(a,b) = 1 \quad \Leftrightarrow \quad \varphi(a) = \varphi(b), \tag{1}$$

for every $a,b \in A$.

Weighted automata

$\mathcal{S}$ – semiring, X – finite set – *alphabet*

A *weighted automaton* over $\mathcal{S}$ and X – a quadruple $\mathcal{A} = (A, \delta^A, \sigma^A, \tau^A)$

- A – non-empty set – a *set of states*.
- $\delta^A : A \times X \times A \to S$ – *weighted transition function*.
- $\sigma^A : A \to S$ and $\tau^A : A \to S$.

Interpretations:

If $\mathcal{A}$ has m states, then:

δ^A_x – an $m \times m$ matrix – the *transition matrix* determined by $x \in X$.

$\delta^A_u = \delta^A_{x_1} \cdot \delta^A_{x_2} \cdot \cdots \cdot \delta^A_{x_n}$ – the *extended transition matrix* determined by the word $u = x_1 x_2 \ldots x_n \in X^+$.

$\delta^A_\varepsilon = I_A$ – the identity matrix on A.

Matrix σ^A – $1 \times m$ *initial weight vector* – row vector.

Matrix τ^A – $m \times 1$ *final weight vector* – column vector.

Weighted automata

$\mathcal{S}$ – semiring, X – finite set – *alphabet*

A *weighted automaton* over $\mathcal{S}$ and X – a quadruple $\mathcal{A} = (A, \delta^A, \sigma^A, \tau^A)$

- A – non-empty set – a *set of states*.
- $\delta^A : A \times X \times A \to S$ – *weighted transition function*.
- $\sigma^A : A \to S$ and $\tau^A : A \to S$.

Interpretations:

If $\mathcal{A}$ has m states, then:

δ^A_x – an $m \times m$ matrix – the *transition matrix* determined by $x \in X$.

$\delta^A_u = \delta^A_{x_1} \cdot \delta^A_{x_2} \cdot \cdots \cdot \delta^A_{x_n}$ – the *extended transition matrix* determined by the word $u = x_1 x_2 \ldots x_n \in X^+$.

$\delta^A_\varepsilon = I_A$ – the identity matrix on A.

Matrix σ^A – $1 \times m$ *initial weight vector* – row vector.

Matrix τ^A – $m \times 1$ *final weight vector* – column vector.

Behavior of weighted automata

Let L be an arbitrary language over an alphabet X ($L \subseteq X^*$).

The *L-behavior of a weighted automaton* $\mathcal{A} = (A, X, \delta^A, \sigma^A, \tau^A)$, denoted as $[\![\mathcal{A}]\!]_L$ – a mapping from L to S defined by

$$[\![\mathcal{A}]\!]_L(u) = \sigma^A \cdot \delta^A_u \cdot \tau^A, \quad (2)$$

for any $u \in L$.

Weighted automata $\mathcal{A}$ and $\mathcal{B}$ are *L-equivalent* – $\mathcal{A}$ and $\mathcal{B}$ have the same L-behavior.

Weighted automata $\mathcal{A}$ and $\mathcal{B}$ *equivalent* – $\mathcal{A}$ and $\mathcal{B}$ are X^*-equivalent weighted automata.

General system

System consisting of matrix equations

$$\begin{aligned} \sigma^A \cdot \tau^A &= \sigma^A \cdot U \cdot \tau^A, \\ \sigma^A \cdot \delta^A_{x_1} \cdots \delta^A_{x_n} \cdot \tau^A &= \sigma^A \cdot U \cdot \delta^A_{x_1} \cdot U \cdots U \cdot \delta^A_{x_n} \cdot U \cdot \tau^A, \end{aligned} \quad (3)$$

where U is an unknown $m \times m$ matrix, $n \in \mathbb{N}$ and $u = x_1 x_2 \dots x_n \in L$.

The system (3) – the *L-general system*.

Behavior of weighted automata

Let L be an arbitrary language over an alphabet X ($L \subseteq X^*$).

The *L-behavior of a weighted automaton* $\mathcal{A} = (A, X, \delta^A, \sigma^A, \tau^A)$, denoted as $[\![\mathcal{A}]\!]_L$ – a mapping from L to S defined by

$$[\![\mathcal{A}]\!]_L(u) = \sigma^A \cdot \delta^A_u \cdot \tau^A, \tag{2}$$

for any $u \in L$.

Weighted automata $\mathcal{A}$ and $\mathcal{B}$ are *L-equivalent* – $\mathcal{A}$ and $\mathcal{B}$ have the same L-behavior.

Weighted automata $\mathcal{A}$ and $\mathcal{B}$ *equivalent* – $\mathcal{A}$ and $\mathcal{B}$ are X^*-equivalent weighted automata.

General system

System consisting of matrix equations

$$\begin{aligned} \sigma^A \cdot \tau^A &= \sigma^A \cdot U \cdot \tau^A, \\ \sigma^A \cdot \delta^A_{x_1} \cdots \delta^A_{x_n} \cdot \tau^A &= \sigma^A \cdot U \cdot \delta^A_{x_1} \cdot U \cdots U \cdot \delta^A_{x_n} \cdot U \cdot \tau^A, \end{aligned} \tag{3}$$

where U is an unknown $m \times m$ matrix, $n \in \mathbb{N}$ and $u = x_1 x_2 \ldots x_n \in L$.

The system (3) – the *L-general system*.

Definition. (L, R)**–transformations of weighted automata**

- $\mathcal{A} = (A, X, \delta^A, \sigma^A, \tau^A)$ – weighted automaton with m states.
- L – an $k \times m$ matrix. R – an $m \times k$ matrix, for an arbitrary $k \in \mathbb{N}$.

Weighted automaton $\mathcal{B} = (B, X, \delta^B, \sigma^B, \tau^B)$ is an (L, R)–*transformation* of an automaton $\mathcal{A}$ if it is defined by:

$$\begin{aligned} \delta^B_x &= L \cdot \delta^A_x \cdot R, \\ \sigma^B &= \sigma^A \cdot L, \\ \tau^B &= R \cdot \tau^A, \end{aligned} \tag{4}$$

and B is its set of states.

Result

Let $\mathcal{A} = (A, X, \delta^A, \sigma^A, \tau^A)$ be a weighted automaton and let $\mathcal{B} = (B, X, \delta^B, \sigma^B, \tau^B)$ be an (L, R)–*transformation* of $\mathcal{A}$ for some matrices L and R. Automata $\mathcal{A}$ and $\mathcal{B}$ are equivalent if and only if a matrix $Q = L \cdot R$ is a solution to the general system of matrix equations.

Definition. (L, R)**–transformations of weighted automata**

- $\mathcal{A} = (A, X, \delta^A, \sigma^A, \tau^A)$ – weighted automaton with m states.
- L – an $k \times m$ matrix. R – an $m \times k$ matrix, for an arbitrary $k \in \mathbb{N}$.

Weighted automaton $\mathcal{B} = (B, X, \delta^B, \sigma^B, \tau^B)$ is an (L, R)–*transformation* of an automaton $\mathcal{A}$ if it is defined by:

$$\begin{aligned} \delta^B_x &= L \cdot \delta^A_x \cdot R, \\ \sigma^B &= \sigma^A \cdot L, \\ \tau^B &= R \cdot \tau^A, \end{aligned} \tag{4}$$

and B is its set of states.

Result

Let $\mathcal{A} = (A, X, \delta^A, \sigma^A, \tau^A)$ be a weighted automaton and let $\mathcal{B} = (B, X, \delta^B, \sigma^B, \tau^B)$ be an (L, R)–*transformation* of $\mathcal{A}$ for some matrices L and R. Automata $\mathcal{A}$ and $\mathcal{B}$ are equivalent if and only if a matrix $Q = L \cdot R$ is a solution to the general system of matrix equations.

Problems with (L, R)-transformations of weighted automata

Prior to make the (L, R)-transformations of a given weighted automaton, one has to compute a matrix $Q = L \cdot R$ which is solution to the system (3) and has as small as possible rank.

- Dealing with the L-general system is a problem – generally, it consists of infinitely many equations.
- Even if a solution with the desirable (small) rank is computed – computing its rank decomposition may be a problem.

Resolutions (goals):

- Solving instances of the L-general system effectively, or even efficiently.
- Computing functions that are solutions to the L-general system. Its rank decomposition can be efficiently done.

Definition. L-weakly right invariant system

Let $\mathcal{A} = (A, X, \delta^A, \sigma^A, \tau^A)$ be a weighted automaton over an alphabet X and L an arbitrary language over X.

Let $\tau_u^A = \delta_u^A \cdot \tau^A$ ($\sigma_u^A = \sigma^A \cdot \delta_u^A$), for every $u \in X^*$.

System of matrix equations:

$$X \cdot \tau_u^A = \tau_u^A, \tag{5}$$

where $X \in S^{A\times A}$ is an unknown matrix and $u \in L$ is an arbitrary word.

System (5) – *L-weakly right invariant system.*

Definition. L-weakly left invariant system

System of matrix equations, dual to the previous one:

$$\sigma_u^A \cdot X = \sigma_u^A, \tag{6}$$

where $X \in S^{A\times A}$ is an unknown matrix and $u \in L$ is an arbitrary word.

System (6) – *L-weakly left invariant system.*

Definition. L-weakly right invariant system

Let $\mathcal{A} = (A, X, \delta^A, \sigma^A, \tau^A)$ be a weighted automaton over an alphabet X and L an arbitrary language over X.

Let $\tau_u^A = \delta_u^A \cdot \tau^A$ ($\sigma_u^A = \sigma^A \cdot \delta_u^A$), for every $u \in X^*$.

System of matrix equations:

$$X \cdot \tau_u^A = \tau_u^A, \tag{5}$$

where $X \in S^{A\times A}$ is an unknown matrix and $u \in L$ is an arbitrary word.

System (5) – *L-weakly right invariant system.*

Definition. L-weakly left invariant system

System of matrix equations, dual to the previous one:

$$\sigma_u^A \cdot X = \sigma_u^A, \tag{6}$$

where $X \in S^{A\times A}$ is an unknown matrix and $u \in L$ is an arbitrary word.

System (6) – *L-weakly left invariant system.*

Result

Let $\mathcal{A} = (A, X, \delta^A, \sigma^A, \tau^A)$ be a weighted automaton over an alphabet X and L an arbitrary prefix-closed language over X.

Then, every solution to the L-weakly right (left) invariant system is a solution to the L-general system.

Equivalences related to L-weakly right (left) invariant equivalences

Let $\mathcal{A} = (A, X, \delta^A, \sigma^A, \tau^A)$ be a weighted automaton over an alphabet X and L an arbitrary language over X.

Equivalence relation satisfying:

$$E(a,b) = 1 \quad \Rightarrow \quad \tau_u^A(a) = \tau_u^A(b), \tag{7}$$

for every $a, b \in A$ and every $u \in L$, is called an *L-weakly right invariant equivalence*.

An L-weakly right invariant equivalence, defined by

$$E_L(a,b) = 1 \quad \Leftrightarrow \quad \tau_u^A(a) = \tau_u^A(b), \quad \text{for every } u \in L \tag{8}$$

is the greatest one.

Result

Let $\mathcal{A} = (A, X, \delta^A, \sigma^A, \tau^A)$ be a weighted automaton over an alphabet X and L an arbitrary prefix-closed language over X.

Then, every solution to the L-weakly right (left) invariant system is a solution to the L-general system.

Equivalences related to L-weakly right (left) invariant equivalences

Let $\mathcal{A} = (A, X, \delta^A, \sigma^A, \tau^A)$ be a weighted automaton over an alphabet X and L an arbitrary language over X.

Equivalence relation satisfying:

$$E(a,b) = 1 \quad \Rightarrow \quad \tau_u^A(a) = \tau_u^A(b), \tag{7}$$

for every $a, b \in A$ and every $u \in L$, is called an *L-weakly right invariant equivalence*.

An L-weakly right invariant equivalence, defined by

$$E_L(a,b) = 1 \quad \Leftrightarrow \quad \tau_u^A(a) = \tau_u^A(b), \quad \text{for every } u \in L \tag{8}$$

is the greatest one.

Results

Let $\mathcal{A} = (A, X, \delta^A, \sigma^A, \tau^A)$ be a weighted automaton over an alphabet X, L a prefix-closed language over X and E_L the greatest L-weakly right invariant equivalence.

Let $\varphi : A \to A$ be an arbitrary function. Then:

- φ is a solution to the L-weakly right invariant system with the smallest rank iff φ is idempotent and $Ker\varphi = E_L$.
- $\rho(\varphi) = \rho(I_{A/E_L})$.

Some consequences

- There can be many different functions that are solutions to the system (5) with the smallest rank.
- Any such function can be efficiently computed.
- If φ is one them, then (l^φ, r^φ), a rank decomposition of φ, can be efficiently computed. Moreover, an (l^φ, r^φ)–transformation of $\mathcal{A}$ has at most $|A/E_L|$ states.

Results

Let $\mathcal{A} = (A, X, \delta^A, \sigma^A, \tau^A)$ be a weighted automaton over an alphabet X, L a prefix-closed language over X and E_L the greatest L-weakly right invariant equivalence.

Let $\varphi : A \to A$ be an arbitrary function. Then:

- φ is a solution to the L-weakly right invariant system with the smallest rank iff φ is idempotent and $Ker\varphi = E_L$.
- $\rho(\varphi) = \rho(I_{A/E_L})$.

Some consequences

- There can be many different functions that are solutions to the system (5) with the smallest rank.
- Any such function can be efficiently computed.
- If φ is one them, then (l^φ, r^φ), a rank decomposition of φ, can be efficiently computed. Moreover, an (l^φ, r^φ)–transformation of $\mathcal{A}$ has at most $|A/E_L|$ states.

Problems with the computation of the greatest L-weakly right equivalences

In order to construct a function that is a solution to (5) with the smallest rank, one has to compute the greatest L-weakly right invariant equivalence E_L.

In case L is an infinite language, an equivalence relation E_L has to satisfy infinitely many particular conditions.

Resolution

Efficiently solving some instances of L-weakly right general system.

Problems with the computation of the greatest L-weakly right equivalences

In order to construct a function that is a solution to (5) with the smallest rank, one has to compute the greatest L-weakly right invariant equivalence E_L.

In case L is an infinite language, an equivalence relation E_L has to satisfy infinitely many particular conditions.

Resolution

Efficiently solving some instances of L-weakly right general system.

Definition. L-right invariant system

Let $\mathcal{A} = (A, X, \delta^A, \sigma^A, \tau^A)$ be a weighted automaton over an alphabet X and L an arbitrary language over X.

System of matrix equations:

$$
\begin{aligned}
X \cdot \delta_x^A \cdot X &= \delta_x^A \cdot X, \\
X \cdot \tau^A &= \tau^A,
\end{aligned}
\tag{9}
$$

where $X \in S^{A\times A}$ is an unknown matrix and $x \in X$ is an arbitrary symbol.

System (5) – *L-right invariant system.*

Definition. L-left invariant system

System of matrix equations, dual to the previous one:

$$
\begin{aligned}
X \cdot \delta_x^A \cdot X &= X \cdot \delta_x^A, \\
\sigma^A \cdot X &= \sigma^A,
\end{aligned}
\tag{10}
$$

where $X \in S^{A\times A}$ is an unknown matrix and $x \in X$ is an arbitrary symbol.

System (6) – *L-left invariant system.*

Definition. L-right invariant system

Let $\mathcal{A} = (A, X, \delta^A, \sigma^A, \tau^A)$ be a weighted automaton over an alphabet X and L an arbitrary language over X.

System of matrix equations:

$$\begin{aligned} X \cdot \delta_x^A \cdot X &= \delta_x^A \cdot X, \\ X \cdot \tau^A &= \tau^A, \end{aligned} \tag{9}$$

where $X \in S^{A\times A}$ is an unknown matrix and $x \in X$ is an arbitrary symbol.

System (5) – *L-right invariant system.*

Definition. L-left invariant system

System of matrix equations, dual to the previous one:

$$\begin{aligned} X \cdot \delta_x^A \cdot X &= X \cdot \delta_x^A, \\ \sigma^A \cdot X &= \sigma^A, \end{aligned} \tag{10}$$

where $X \in S^{A\times A}$ is an unknown matrix and $x \in X$ is an arbitrary symbol.

System (6) – *L-left invariant system.*

Results

Let $\mathcal{A} = (A, X, \delta^A, \sigma^A, \tau^A)$ be a weighted automaton over an alphabet X and L a prefix-closed language over X. The following is true:

- Every solution to the L-right (resp. L-left) invariant system is a solution to the L-weakly right (resp. L-weakly left) invariant system.
- There exists a function φ that is a solution to the L-right invariant system with the smallest rank.
- Function φ, its rank and rank decomposition can be efficiently computed.
- Time complexity of the algorithm that computes φ is $O(mnc_+)$, where m is the number of states of $\mathcal{A}$, n is the number of nonzero transitions of $\mathcal{A}$, and c_+ and denotes the computation cost for performing the semiring operation $+$.

All the presented results are in

A. Stamenković, S. Stanimirović, Vesa Halava, **Certain linear and weakly linear systems of matrix equations over semirings. Applications in a state reduction of weighted automata,** *Filomat, in editing*

Results

Let $\mathcal{A} = (A, X, \delta^A, \sigma^A, \tau^A)$ be a weighted automaton over an alphabet X and L a prefix-closed language over X. The following is true:

- Every solution to the L-right (resp. L-left) invariant system is a solution to the L-weakly right (resp. L-weakly left) invariant system.
- There exists a function φ that is a solution to the L-right invariant system with the smallest rank.
- Function φ, its rank and rank decomposition can be efficiently computed.
- Time complexity of the algorithm that computes φ is $O(mnc_+)$, where m is the number of states of $\mathcal{A}$, n is the number of nonzero transitions of $\mathcal{A}$, and c_+ and denotes the computation cost for performing the semiring operation $+$.

All the presented results are in

A. Stamenković, S. Stanimirović, Vesa Halava,* Certain linear and weakly linear systems of matrix equations over semirings. Applications in a state reduction of weighted automata, *Filomat, in editing

THANK YOU