Approximate State Reduction of Fuzzy Finite Automata[*]

Miroslav Ćirić   Ivana Micić   Stefan Stanimirović

University of Niš, Faculty of Sciences and Mathematics, Višegradska 33, Niš, Serbia

miroslav.ciric@pmf.edu.rs, ivana.micic@pmf.edu.rs, stefan.stanimirovic@pmf.edu.rs

Linh Anh Nguyen

Institute of Informatics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland, and
Faculty of Information Technology, Nguyen Tat Thanh University, Ho Chi Minh City, Viet Nam

nguyen@mimuw.edu.pl

In this paper we introduce a new type of approximate state reductions where the behaviors of the reduced and the original automaton do not have to be identical, but they must match on all words of length less than or equal to some given natural number. We provide four methods for performing such reductions.

1 Introduction

Minimization and state reduction are related problems that belong to the fundamental problems of automata theory and have many significant applications. Minimization is the problem of finding an automaton with a minimal number of states equivalent to a given automaton. However, this problem cannot be solved efficiently (in polynomial time) for fuzzy finite or nondeterministic finite automata as their particular type (cf. [7] for more details). Therefore, the so-called state reduction problem for fuzzy finite automata is studied instead, where the goal is to find an automaton equivalent to the given automaton that is not necessarily minimal, but that can be treated sufficiently small or close enough to the minimal one when comparing the number of states. In turn, the state reduction algorithm can be performed efficiently. Ćirić et al. discussed this problem in [4], and then in [5], [17] and [16], where they proposed state reduction methods that construct sequences of fuzzy matrices. The drawback of these methods is that these sequences can be infinite when the underlying structure of truth values is not locally finite. However, even when the sequences of matrices are finite, the number of different elements in sequences can be high. For the reasons above, different authors have proposed an approximate approach not only in the context of state reduction but also in some other close contexts, such as containment and equivalence of fuzzy automata, as well as simulations and bisimulations between fuzzy automata (see [3, 8, 9, 10, 11, 13, 12, 15, 18, 19, 21] and articles cited there).

In the approximate state reduction problem, the main goal is to construct an automaton with a smaller number of states than a given automaton with behaviour that does not have to be identical to the behavior of a given automaton, but "close enough" to it. Dominantly, scholars have defined the closeness between the behaviors of two automata by the concept of the degree of equality of fuzzy sets [3, 21, 19, 11]. However, when we take membership values from the real unit interval, the conventional metric on that interval can also be used to define closeness [8, 20]. Furthermore, a different fuzzy similarity measure has been proposed recently in [15, 14] via relational lifting. In this paper, we approach approximate state

---

reductions differently. Namely, we require that the behaviors of a given automaton and its reduced one may not be strictly equivalent, but equivalent for all words with length not exceeding a given natural number k. This relaxation from the strict equivalence comes naturally, as when working with fuzzy automata in practical situations, one encounters words with finite (bounded) length. It is important to emphasize that Nguyen et al. recently employed a similar idea in [12] to generalize fuzzy simulations and fuzzy bisimulations for fuzzy automata [11]. In this paper, k-equivalence means the equivalence of a given fuzzy automaton and its reduced one for all words not exceeding length k. Similarly, k-reduction means a state reduction resulting in an automaton k-equivalent to the given automaton.

This paper provides four k-reduction methods based on state reduction methods developed in [17] that output a fuzzy automaton strictly equivalent to the given fuzzy automaton. Precisely, the first two methods consist of constructing a descending sequence of fuzzy quasi-order matrices. Theorems 4.1 and 4.2 prove that the fuzzy automaton formed by the different row vectors of the kth member of that sequence of matrices (the numbering starts from 0) is k-equivalent to the given fuzzy automaton. More-over, if the number of different elements in this sequence is not greater than k, then the resulting fuzzy automaton is also strictly equivalent to the given fuzzy automaton. In locally finite structures, such as the Gödel or Łukasiewicz structure, the sequence necessarily has a finite number of different elements. Therefore, we can always pick a sufficiently high $k \in \mathbb{N}$ in these structures, so the resulting fuzzy au-tomaton is also strictly equivalent to the given fuzzy automaton. On the other hand, for some non-locally structures satisfying some additional conditions [4, 17], such as the product structure, a reduced fuzzy automaton strictly equivalent to the original fuzzy automaton can be constructed from different row vectors of the infimum of this sequence. Therefore, by choosing the kth member of the sequence, the re-sulting k-equivalent reduced fuzzy automaton can be regarded as an approximation of the reduced fuzzy automaton strictly equivalent to the original fuzzy automaton.

The other two methods for performing state reduction introduced in [17] consist of constructing a family of fuzzy quasi-order matrices such that a reduced fuzzy automaton built from different row vectors of the infimum of this family is strictly equivalent to the original fuzzy automaton. These methods generally give better reductions than the first two, but their time complexity is generally higher. Here we transform that family into a sequence of fuzzy quasi-order matrices and prove that the fuzzy automaton formed by different row vectors of the kth member of that sequence (the numbering again starts from 0) is k-equivalent to the original automaton (Theorems 4.3 and 4.4). We also point out the advantages and disadvantages of the four proposed methods.

2   Preliminaries

Throughout this paper, $\mathbb{N}$ will denote the set of all natural numbers (including the zero).

A *residuated lattice* is defined as an algebra $\mathbb{L} = (\mathbb{L}, \vee, \wedge, \otimes, \rightarrow, 0, 1)$, with four binary operations and two constants 0 and 1, which satisfies the following conditions:

(R1) $(\mathbb{L}, \vee, \wedge, 0, 1)$ is a bounded lattice with the least element 0 and the greatest element 1;
(R2) $(\mathbb{L}, \otimes, 1)$ is a commutative semigroup with the identity 1;
(R3) the pair $(\otimes, \rightarrow)$ satisfies the *adjunction* or *residuation property*: for all $a, b, c \in \mathbb{L}$,

$$a \otimes b \leqslant c \;\; \Leftrightarrow \;\; a \leqslant b \rightarrow c.$$

Here $\leqslant$ stands for the ordering in the lattice from (R1). The operation $\rightarrow$ is called the *residuum*, and $\otimes$ is called the *multiplication*. If the bounded lattice from (R1) is complete, then $\mathbb{L}$ is called a *complete residuated lattice*. As it is customary in the theory of algebraic structures to use the same notation for an

algebra and its carrier set, here we denote the residuated lattice and its carrier set with the same symbol $\mathbb{L}$ as well.

The main examples of complete residuated lattices are those whose carrier set is the real unit interval $\mathbb{I} = [0,1]$ and the multiplication is some triangular norm on $\mathbb{I}$, such as, for example, the Gödel structure, product structure and Łukasiewicz structure. For more information about complete residuated lattices and the mentioned structures on $\mathbb{I}$ we refer to the books [2, 1] and other papers listed below, in the list of references.

Let $\mathbb{L}$ be an arbitrary complete residuated lattice. For arbitrary $m, n \in \mathbb{N} \setminus \{0\}$, by $\mathbb{L}^{m \times n}$ we denote the set of all $m \times n$ matrices with entries in $\mathbb{L}$, and by $\mathbb{L}^n$ the set of all vectors of size n with entries in $\mathbb{L}$ (by the size of a vector we mean the number of its coordinates). A *fuzzy subset* of a non-empty set A is defined as any function $\alpha : A \to \mathbb{L}$, and a *fuzzy relation* on A is defined as any fuzzy subset of $A \times A$, that is, as any function $M : A \times A \to \mathbb{L}$. For $a \in A$, the value $\alpha(a)$ is called the *membership degree* of a in the fuzzy set α. Here we deal mostly with fuzzy subsets of a finite set, as well as with fuzzy relations on a finite set, and then, when dealing with a finite set $A = \{a_1, a_2, \ldots, a_n\}$, a fuzzy subset α of A is identified with a vector from $\mathbb{L}^n$ whose *i*th coordinate is $\alpha(a_i)$, while a fuzzy relation M on A is identified with a matrix from $\mathbb{L}^{n \times n}$ whose (i, j)-entry is $M(a_i, a_j)$. Without risk of confusion, the vector corresponding to the fuzzy subset α is denoted by the same symbol α, and its *i*th coordinate is denoted by $\alpha(i)$, while the matrix corresponding to the fuzzy relation M is denoted by the same symbol M, and its (i, j)-entry is denoted by $M(i, j)$.

For a matrix $M \in \mathbb{L}^{n \times n}$ and a fixed $i \in [1..n]$, where $[1..n] = \{1, 2, \ldots, n\}$, the vector whose *j*th coordinate is $M(i, j)$, for any $j \in [1..n]$, is called the *i*th *row vector* of M, and for a fixed $j \in [1..n]$, the vector whose *i*th coordinate is $M(i, j)$, for any $i \in [1..n]$, is called the *j*th *column vector* of M.

The product $M \cdot N$ of two matrices $M, N \in \mathbb{L}^{n \times n}$ (fuzzy relations on A) is a matrix from $\mathbb{L}^{n \times n}$ (a fuzzy relation on A) defined by

$$(M \cdot N)(i, j) = \bigvee_{s=1}^{n} M(i, s) \otimes N(s, j),$$

for all $i, j \in [1..n]$, the products $\alpha \cdot M$ and $M \cdot \beta$ of vectors $\alpha, \beta \in \mathbb{L}^n$ (fuzzy subsets of A) and a matrix $M \in \mathbb{L}^{n \times n}$ (fuzzy relation on A) are vectors from $\mathbb{L}^n$ (fuzzy subsets of A) defined by

$$(\alpha \cdot M)(i) = \bigvee_{s=1}^{n} \alpha(s) \otimes M(s, i), \qquad (M \cdot \beta)(i) = \bigvee_{s=1}^{n} M(i, s) \otimes \beta(s),$$

for every $i \in [1..n]$, and the product $\alpha \cdot \beta$ of two vectors from $\mathbb{L}^n$ (fuzzy subsets of A) is the element from $\mathbb{L}$ defined by

$$\alpha \cdot \beta = \bigvee_{s=1}^{n} \alpha(s) \otimes \beta(s).$$

The last product $\alpha \cdot \beta$ is called the *scalar product* or *dot product* of vectors (fuzzy subsets) α and β.

The ordering $\leqslant$ on $\mathbb{L}^{n \times n}$ is defined entrywise by

$$M \leqslant N \quad \Leftrightarrow \quad M(i, j) \leqslant N(i, j), \text{ for all } i, j \in [1..n],$$

for all $M, N \in \mathbb{L}^{n \times n}$, and similarly, the ordering $\leqslant$ on $\mathbb{L}^n$ is defined coordinatewise by

$$\alpha \leqslant \beta \quad \Leftrightarrow \quad \alpha(i) \leqslant \beta(i), \text{ for each } i \in [1..n],$$

for all $\alpha, \beta \in \mathbb{L}^n$. It is easy to verify that these orderings on $\mathbb{L}^{n\times n}$ and $\mathbb{L}^n$ are compatible with matrix products and vector-matrix products, that is,

$$\alpha \leqslant \beta \;\Rightarrow\; \alpha \cdot M \leqslant \beta \cdot M, \; M \cdot \alpha \leqslant M \cdot \beta$$
$$M \leqslant N \;\Rightarrow\; K \cdot M \leqslant K \cdot N, \; M \cdot K \leqslant N \cdot K, \; \alpha \cdot M \leqslant \alpha \cdot N, \; M \cdot \alpha \leqslant N \cdot \alpha$$

for all $\alpha, \beta \in \mathbb{L}^n$ and $K, M, N \in \mathbb{L}^{n\times n}$. The supremum and infimum of a family $\{M_s\}_{s\in I}$ of matrices from $\mathbb{L}^{n\times n}$ are respectively matrices from $\mathbb{L}^{n\times n}$ defined by

$$\left(\bigvee_{s\in I} M_s\right)(i,j) = \bigvee_{s\in I} M_s(i,j), \qquad \left(\bigwedge_{s\in I} M_s\right)(i,j) = \bigwedge_{s\in I} M_s(i,j),$$

for all $i, j \in [1..n]$.

The matrix $I_n \in \mathbb{L}^{n\times n}$ defined by $I_n(i,j) = 1$, for $i = j$, and $I_n(i,j) = 0$, for $i \neq j$, $i, j \in [1..n]$, is called the *identity matrix* of order n. A matrix $M \in \mathbb{L}^{n\times n}$ is *reflexive* if $I_n \leqslant M$, it is *transitive* if $M^2 \leqslant M$, where $M^2 = M \cdot M$, and it is *symmetric* if $M(i,j) = M(j,i)$, for all $i, j \in [1..n]$. A reflexive and transitive matrix is called a *fuzzy quasi-order matrix*, and a symmetric fuzzy quasi-order matrix is called a *fuzzy equivalence matrix*.

For matrices $M, N \in \mathbb{L}^{n\times n}$, the *right residual* of N by M is a matrix $M \backslash N \in \mathbb{L}^{n\times n}$ defined by

$$(M\backslash N)(j,k) = \bigwedge_{i=1}^{n} M(i,j) \to N(i,k), \tag{1}$$

for all $j, k \in [1..n]$, and the *left residual* of N by M is a matrix $N/M \in \mathbb{L}^{n\times n}$ defined by

$$(N/M)(i,j) = \bigwedge_{k=1}^{n} M(j,k) \to N(i,k), \tag{2}$$

for all $i, j \in [1..n]$. Matrix residuals are related with matrix multiplication by the following *residuation* (*adjunction*) *property*:

$$K \cdot M \leqslant N \;\Leftrightarrow\; M \leqslant K\backslash N \;\Leftrightarrow\; K \leqslant N/M, \tag{3}$$

for all $K, M, N \in \mathbb{L}^{n\times n}$. Next, for $\alpha, \beta \in \mathbb{L}^n$, the *right residual* of β by α is a matrix $\alpha\backslash\beta \in \mathbb{L}^{n\times n}$ given by

$$(\alpha\backslash\beta)(i,j) = \alpha(i) \to \beta(j), \tag{4}$$

for all $i, j \in [1..n]$, and the *left residual* of β by α is a matrix $\beta/\alpha \in \mathbb{L}^{n\times n}$ given by

$$(\beta/\alpha)(j,i) = \alpha(i) \to \beta(j), \tag{5}$$

for all $i, j \in [1..n]$. It is clear that $\alpha\backslash\beta = (\beta/\alpha)^\top$, that is, $\beta/\alpha = (\alpha\backslash\beta)^\top$ (here $M^\top$ denotes the *transpose* of a matrix M). The *residuation property* for these residuals is

$$\alpha \cdot M \leqslant \beta \;\Leftrightarrow\; M \leqslant \alpha\backslash\beta, \qquad M \cdot \alpha \leqslant \beta \;\Leftrightarrow\; M \leqslant \beta/\alpha, \tag{6}$$

for all $\alpha, \beta \in \mathbb{L}^n$ and $M \in \mathbb{L}^{n\times n}$.

The representation of the matrix $M \in \mathbb{L}^{m\times n}$ in the form of the product $M = L \cdot R$, where $L \in \mathbb{L}^{m\times r}$ and $R \in \mathbb{L}^{r\times n}$, is called the *r-factorization* of that matrix M. The smallest number r for which there is an

r-factorization of the matrix M is denoted by $\rho(M)$ and is called the *Schein's rank* or *factor rank* of M. The concepts of *r*-factorization and Schein's rank are defined for arbitrary matrices, but have particularly good properties when applied to fuzzy quasi-order matrices (cf. [16]). As shown in [6, 17], a fuzzy quasi-order matrix Q has the same number of different row vectors and different column vectors, which is denoted by $d(Q)$. In the general case, $\rho(Q) \leqslant d(Q)$, but for every fuzzy quasi-order matrix Q with entries in a complete residuated lattice $\mathbb{L}$ in which for all $a, b \in \mathbb{L}$ from $a \vee b = 1$ it follows $a = 1$ or $b = 1$ (for instance, this holds if $\mathbb{L}$ is linearly ordered), we have that $\rho(Q) = d(Q)$ (cf. [16]).

For undefined notions and notation we refer to [2, 1].

3   Fuzzy finite automata and the state reduction problem

Throughout this paper, if not noted otherwise, $\mathbb{L}$ will denote an arbitrary complete residuated lattice and X will denote an arbitrary non-empty alphabet. By X^* we denote the free monoid over X, whose identity, called the *empty word*, is denoted by ε, while by X^+ we denote the free semigroup over X, i.e., $X^+ = X^* \setminus \{\varepsilon\}$.

A *fuzzy finite automaton* over $\mathbb{L}$ and X is defined as a tuple $\mathcal{A} = (A, \sigma, \delta, \tau)$, where A is a non-empty finite set, while σ, τ and δ are functions such that $\sigma, \tau : A \to \mathbb{L}$ and $\delta : A \times X \times A \to \mathbb{L}$. The function δ is often replaced by the family of functions $\{\delta_x\}_{x \in X}$, where $\delta_x : A \times A \to \mathbb{L}$ is given by $\delta_x(a, b) = \delta(a, x, b)$, for all $a, b \in A$ and $x \in X$. We call A the *set of states*, σ the *fuzzy set of initial states*, τ the *fuzzy set of terminal states*, and δ and δ_x, $x \in X$, the *fuzzy transition functions*. The number of states of $\mathcal{A}$ will be denoted by $|\mathcal{A}|$.

The *behavior* of the fuzzy finite automaton $\mathcal{A}$ is a function $[\![\mathcal{A}]\!] : X^* \to \mathbb{L}$ (i.e., a fuzzy subset of X^*) defined by

$$[\![\mathcal{A}]\!](u) = \bigvee_{(a_0, a_1, \ldots, a_k) \in A^{k+1}} \sigma(a_0) \otimes \delta(a_0, x_1, a_1) \otimes \delta(a_1, x_2, a_2) \cdots \delta(a_{k-1}, x_k, a_k) \otimes \tau(a_k), \tag{7}$$

for $u = x_1 x_2 \ldots x_k \in X^+$, $x_1, x_2, \ldots, x_k \in X$, and

$$[\![\mathcal{A}]\!](\varepsilon) = \bigvee_{a \in A} \sigma(a) \otimes \tau(a). \tag{8}$$

We say that $[\![\mathcal{A}]\!]$ is the *fuzzy language recognized (accepted) by* the fuzzy finite automaton $\mathcal{A}$, or in short just the *fuzzy language of* $\mathcal{A}$.

As we noted in the previous section, fuzzy subsets of a finite set with n elements can be treated as fuzzy vectors. i.e., as vectors from $\mathbb{L}^n$, while fuzzy relations on such a set can be treated as fuzzy matrices, i.e. as matrices from $\mathbb{L}^{n \times n}$. When such a way of viewing is applied to the fuzzy finite automaton $\mathcal{A}$, then σ and τ are treated as vectors from $\mathbb{L}^n$, called respectively the *initial fuzzy vector* and *terminal fuzzy vector*, while δ_x, $x \in X$, are treated as matrices from $\mathbb{L}^{n \times n}$, called the *transition fuzzy matrices*. Then the tuple is called the *linear representation* of the fuzzy finite automaton $\mathcal{A}$, while n is called the *dimension* of $\mathcal{A}$. Such a way of looking at σ, τ and δ_x's will be applied here as well. In the vector-matrix form, the behavior of the fuzzy finite automaton $\mathcal{A}$ is represented as follows:

$$[\![\mathcal{A}]\!](u) = \sigma \cdot \delta_{x_1} \cdot \delta_{x_2} \cdots \delta_{x_s} \cdot \tau = \sigma \cdot \delta_u \cdot \tau, \tag{9}$$

for $u = x_1 x_2 \ldots x_s \in X^+$, $x_1, x_2, \ldots, x_s \in X$, where $\delta_u = \delta_{x_1} \cdot \delta_{x_2} \cdots \delta_{x_k}$, and

$$[\![\mathcal{A}]\!](\varepsilon) = \sigma \cdot \tau. \tag{10}$$

As can be seen, here we treat σ as a row vector and τ as a column vector.

Two fuzzy finite automata $\mathcal{A}$ and $\mathcal{B}$ over $\mathbb{L}$ and X are said to be *equivalent* if

$$[\![\mathcal{A}]\!](u) = [\![\mathcal{B}]\!](u), \qquad \text{for every } u \in X^*, \tag{11}$$

i.e., if $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!]$, and for an arbitrary $k \in \mathbb{N}$, we say that $\mathcal{A}$ and $\mathcal{B}$ are *k-equivalent* if

$$[\![\mathcal{A}]\!](u) = [\![\mathcal{B}]\!](u), \qquad \text{for every } u \in X^* \text{ such that } |u| \leqslant k. \tag{12}$$

If $\mathcal{A}$ and $\mathcal{B}$ are equivalent or *k*-equivalent, we also say that $\mathcal{A}$ is *equivalent to* or *k-equivalent to* $\mathcal{B}$, and vice versa.

Let $Q \in \mathbb{L}^{n \times n}$ be an arbitrary fuzzy quasi-order matrix. Let $1 \leq i_1 \leq \ldots \leq i_k \leq n$ be the smallest indices of all pairwise distinct rows of Q. As mentioned earlier [6, 17], they are also the smallest indices of all pairwise distinct columns of Q. Let $Q_r \in \mathbb{L}^{k \times n}$ (respectively, $Q_c \in \mathbb{L}^{n \times k}$) be the matrix consisting of the rows (respectively, columns) $i_1, \ldots, i_k$ of Q. Let $\mathcal{A} = (A, \sigma, \delta, \tau)$ be a fuzzy finite automaton over $\mathbb{L}$ and X with n states. Then, a new fuzzy finite automaton, with the linear representation

$$\mathcal{A}_Q = (k, \sigma^Q, \{\delta_x^Q\}_{x \in X}, \tau^Q),$$

is defined by putting that

$$\sigma^Q = \sigma \cdot Q_c,$$
$$\delta_x^Q = Q_r \cdot \delta_x \cdot Q_c, \quad \text{for all } x \in X,$$
$$\tau^Q = Q_r \cdot \tau.$$

It is clear that $\sigma^Q, \tau^Q \in \mathbb{L}^k$ and $\delta_x^Q \in \mathbb{L}^{k \times k}$, for each $x \in X$, so $\mathcal{A}_Q$ is a well-defined fuzzy finite automaton. Since $Q = Q_c \cdot Q_r$ (cf. [16, Theorem 4.1], the behavior of $\mathcal{A}_Q$ is given in the vector-matrix form by

$$[\![\mathcal{A}_Q]\!](u) = \sigma \cdot Q \cdot \delta_{x_1} \cdot Q \cdot \delta_{x_2} \cdot \cdots \cdot Q \cdot \delta_{x_s} \cdot Q \cdot \tau, \tag{13}$$

for every $u = x_1 x_2 \ldots x_s \in X^+$, where $x_1, x_2, \ldots, x_s \in X$, and

$$[\![\mathcal{A}_Q]\!](\varepsilon) = \sigma \cdot Q \cdot \tau. \tag{14}$$

As we mentioned earlier, fuzzy matrices can be identified with fuzzy relations between finite sets. Since in the theory of fuzzy sets, the "rows" of fuzzy relations are known as *aftersets*, and the "columns" are known as *foresets*, the fuzzy finite automaton $\mathcal{A}_Q$ was called in [17] the *afterset automaton* of $\mathcal{A}$ corresponding to the fuzzy quasi-order matrix Q, but it can also be rightly called the *row automaton*. It was also shown in [17] that if in the construction of the automaton $\mathcal{A}_Q$ instead of the rows (aftersets) of the matrix Q we use its columns (foresets), then essentially nothing changes, because an isomorphic fuzzy finite automaton is obtained.

Let us note that the number of states of $\mathcal{A}_Q$ is $d(Q) \leqslant n$, that is, $\mathcal{A}_Q$ has less than or equal number of states with $\mathcal{A}$. Therefore, by constructing the automaton $\mathcal{A}_Q$ we can reduce the number of states of the automaton $\mathcal{A}$, provided that these two automata are equivalent. Consequently, the main question here is *under what conditions on Q the automaton $\mathcal{A}_Q$ is equivalent to $\mathcal{A}$?* This question can also be formulated as: *under what conditions on Q the construction of $\mathcal{A}_Q$ preserves the fuzzy language of the automaton*

$\mathcal{A}$? More answers to these questions were provided in [5] and [17]. *Right invariant matrices* were defined in [5, 17] as solutions of the system of matrix equations

(ri-1) $U \cdot \tau \leqslant \tau$;
(ri-2) $U \cdot \delta_x \leqslant \delta_x \cdot U$,   for all $x \in X$,

where U is an unknown matrix taking values in $\mathbb{L}^{n \times n}$, and *left invariant matrices* were defined as solutions of the system

(li-1) $\sigma \cdot U \leqslant \sigma$;
(li-2) $\delta_x \cdot U \leqslant U \cdot \delta_x$,   for all $x \in X$.

It was proved in [5] that both right and left invariant fuzzy equivalence matrices provide equivalence between $\mathcal{A}$ and the corresponding row automaton, while the same for fuzzy quasi-order matrices was proved in [17]. At the same time, it has been proven that fuzzy quasi-order matrices give better reductions than fuzzy equivalence matrices, in the sense that they produce fuzzy finite automata with a smaller number of states.

Procedures for computing the greatest right and left invariant matrices, which are necessarily fuzzy quasi-order matrices, were provided in [17]. For right invariant matrices, the procedure consists of building a decreasing sequence of matrices, which is defined in the next section by formula (17). When there are two equal consecutive members of that sequence, the sequence is finite and stabilizes at some member which is the greatest right invariant matrix. For instance, if $\mathbb{L}$ is the Gödel structure or Boolean algebra, then every such sequence is finite and the greatest right invariant matrix can be computed in a finite number of steps. However, there are also cases when this sequence is infinite and the greatest right invariant matrix can not be computed in a finite number of steps. For example, this may happen when $\mathbb{L}$ is the product structure. All this also applies to left-invariant matrices, where the procedure for computing the greatest such matrix is based on the decreasing sequence of matrices defined in the next section by formula (20).

In paper [17], another way was also provided to get a fuzzy quasi-order matrix Q for which the construction of $\mathcal{A}_Q$ will preserve the language of $\mathcal{A}$. *Weakly right invariant matrices* were defined in as solutions of the system of matrix equations

(wri) $U \cdot \tau_u \leqslant \tau_u$,   for all $u \in X^*$,

where U is an unknown matrix taking values in $\mathbb{L}^{n \times n}$ and $\tau_u = \delta_u \cdot \tau$, and *weakly left invariant matrices* were defined as solutions of the system

(wli) $\sigma_u \cdot U \leqslant \sigma_u$,   for all $u \in X^*$,

where $\sigma_u = \sigma \cdot \delta_u$. As shown in [17], both for any weakly right invariant or weakly left invariant fuzzy quasi-order matrix Q, the row automaton $\mathcal{A}_Q$ is equivalent to $\mathcal{A}$, and the greatest weakly right invariant fuzzy quasi-order, i.e., the greatest solution of (wri), can be expressed as

$$\bigwedge_{u \in X^*} \tau_u / \tau_u, \tag{15}$$

while the greatest weakly left invariant fuzzy quasi-order, i.e., the greatest solution of (wli), can be expressed as

$$\bigwedge_{u \in X^*} \sigma_u \backslash \sigma_u. \tag{16}$$

In general, the greatest weakly right invariant fuzzy quasi-order matrix provides better reduction than the greatest right invariant one, but its computation may be significantly more complex. There may also be a problem of efficient computation of these matrices, because the families $\{\tau_u \mid u \in X^*\}$ and $\{\sigma_u \mid u \in X^*\}$ can be infinite, and even when they are finite, the number of their members can be too large.

All the mentioned problems that concern the computation of the greatest invariant and weakly invariant fuzzy quasi-order matrices actualize the issue of approximate state reductions of fuzzy finite automata, which will be discussed in the next section.

Note that in [16] a method for additional reduction of the number of states of the automaton $\mathcal{A}_Q$ is offered, which is based on the *r*-factorization of the fuzzy quasi-order matrix Q. Namely, let $Q = L \cdot R$ be an *r*-factorization of Q, i.e., $L \in \mathbb{L}^{n \times r}$ and $R \in \mathbb{L}^{r \times n}$. Then we can construct a fuzzy finite automaton $\mathcal{A}_{L|R}$ with the linear representation

$$\mathcal{A}_{L|R} = (r, \sigma^{L|R}, \{\delta_x^{L|R}\}_{x \in X}, \tau^{L|R}),$$

where

$$\sigma^{L|R} = \sigma \cdot L,$$
$$\delta_x^{L|R} = R \cdot \delta_x \cdot L, \quad \text{for all } x \in X,$$
$$\tau^{L|R} = R \cdot \tau.$$

Then $\sigma^{L|R}, \tau^{L|R} \in \mathbb{L}^r$ and $\delta_x^{L|R} \in \mathbb{L}^{r \times r}$, so $\mathcal{A}_{L|R}$ is a well-defined fuzzy finite automaton with r states. According to (13), (14) and $Q = L \cdot R$ we have that $[\![\mathcal{A}_{L|R}]\!] = [\![\mathcal{A}_Q]\!]$. Therefore, when we reduce the number of states of the fuzzy finite automaton $\mathcal{A}$ using the greatest fuzzy quasi-order matrix Q, an additional reduction of the number of states could be performed with the help of an *r*-factorization of Q. Clearly, the smallest number of states we can obtain in this way is $\rho(Q)$, the Schein's rank of Q.

4   Approximate state reduction: *k*-reduction

As we already noted in the introduction, there were already several articles dealing with approximate state reduction, mainly in the context of studying approximate simulations and bisimulations between fuzzy finite automata. The approach used in those articles was to, starting from a given fuzzy finite automaton, construct a new fuzzy finite automaton with a smaller number of states, whose fuzzy language does not have to be identical to the fuzzy language of the original automaton, but is sufficiently similar to that fuzzy language, in relation to a certain measure of similarity. In most papers, that measure was based on subsethood and equality degrees between fuzzy sets.

In this paper, we use a different approach. We require that between fuzzy languages of the original and the reduced fuzzy finite automaton there is an exact match for all words of length less than or equal to *k*, while membership degrees for longer words do not matter to us. In the terminology from the previous section, this means that these fuzzy languages are *k*-equivalent. Such an approach is quite natural if we keep in mind that in practical applications of automata the length of input words is always limited, so the only thing that matters to us is that the membership degrees match for words whose length does not exceed that limit.

A procedure for reduction of the number of states of a fuzzy finite automaton $\mathcal{A}$, which results in a fuzzy finite automaton that is *k*-equivalent to $\mathcal{A}$, will be called a *k-reduction*. The following theorem provides such a procedure that is based on the procedure for reduction of the number of states, provided in [17], that results in an automaton which is strictly equivalent to the original automaton.

Theorem 4.1 *Let $\mathcal{A} = (A, \sigma, \delta, \tau)$ be a fuzzy finite automaton over $\mathbb{L}$ and X with n states.*
Let us inductively define a sequence of matrices $\{Q_k\}_{k \in \mathbb{N}} \subset \mathbb{L}^{n \times n}$ as follows:

$$Q_0 = \tau/\tau, \qquad Q_{k+1} = Q_k \wedge \bigwedge_{x \in X} [(\delta_x \cdot Q_k)/\delta_x], \quad \text{for every } k \in \mathbb{N}. \tag{17}$$

Then for an arbitrary $k \in \mathbb{N}$ the following statements hold:

(a) *Q_k is a fuzzy quasi-order matrix;*
(b) *$\mathcal{A}_{Q_k}$ is k-equivalent to $\mathcal{A}$;*
(c) *if $Q_k = L \cdot R$ is an r-factorization of Q_k, for some $r \leqslant d(Q_k)$, then $\mathcal{A}_{L|R}$ is k-equivalent to $\mathcal{A}$;*
(d) *if $Q_s = Q_{s+1}$, for some $s \leqslant k$, then $Q_k = Q_s$ and both $\mathcal{A}_{Q_k}$ and $\mathcal{A}_{L|R}$ are equivalent to $\mathcal{A}$.*

Proof. (a) By its definition, Q_0 is the greatest solution of the inequation $U \cdot \tau \leqslant \tau$, where U is an unknown matrix taking values in $\mathbb{L}^{n \times n}$. Since I_n is also a solution to this inequation, we conclude that $I_n \leqslant Q_0$. Moreover, we have that

$$Q_0^2 \cdot \tau = Q_0 \cdot Q_0 \cdot \tau \leqslant Q_0 \cdot \tau \leqslant \tau,$$

which means that Q_0^2 is also a solution of $U \cdot \tau \leqslant \tau$, and since Q_0 is the greatest solution to this inequation, we conclude that $Q_0^2 \leqslant Q_0$. This proves that Q_0 is a fuzzy quasi-order matrix.

Suppose that Q_s is a fuzzy quasi-order matrix, for some $s \in \mathbb{N}_0$. Let us observe that for every $x \in X$ the matrix $M_x = (\delta_x \cdot Q_s)/\delta_x$ is the greatest solution of the inequation $U \cdot \delta_x \leqslant \delta_x \cdot Q_s$, with an unknown matrix U. We also have that $I_n \leqslant Q_s$, whence

$$I_n \cdot \delta_x = \delta_x \cdot I_n \leqslant \delta_x \cdot Q_s,$$

which means that I_n is also a solution to the inequation $U \cdot \delta_x \leqslant \delta_x \cdot Q_s$, and consequently, $I_n \leqslant M_x$, since M_x is the greatest solution of this inequation.

On the other hand, we have that

$$M_x^2 \cdot \delta_x = M_x \cdot M_x \cdot \delta_x \leqslant M_x \cdot \delta_x \cdot Q_s \leqslant \delta_x \cdot Q_s \cdot Q_s = \delta_x \cdot Q_s^2 = \delta_x \cdot Q_s,$$

which means that M_x^2 is also a solution of $U \cdot \delta_x \leqslant \delta_x \cdot Q_s$, and thus, $M_x^2 \leqslant M_x$. Hence, $M_x = (\delta_x \cdot Q_s)/\delta_x$ is a fuzzy quasi-order matrix, for each $x \in X$, and the matrix Q_{s+1} is also a fuzzy quasi-order matrix, as the intersection of the family of fuzzy quasi-order matrices $M_x = (\delta_x \cdot Q_s)/\delta_x$, $x \in X$, and the fuzzy quasi-order matrix Q_s.

Let us also note that Q_{s+1} is the greatest solution of the system of linear matrix inequations

$$U \cdot \delta_x \leqslant \delta_x \cdot Q_s, \quad x \in X, \tag{18}$$

contained in Q_s, where U is an unknown matrix taking values in $\mathbb{L}^{n \times n}$.

(b) According to (17), we have that $Q_0 \cdot \tau \leqslant \tau$ and $Q_{t+1} \cdot \delta_x \leqslant \delta_x \cdot Q_t$, for arbitrary $t \in \mathbb{N}_0$ and $x \in X$. On the other hand, from $Q_t \leqslant Q_0$ it follows that $Q_t \cdot \tau \leqslant Q_0 \cdot \tau \leqslant \tau$, whereas from $I_n \leqslant Q_t$ we obtain that $\tau = I_n \cdot \tau \leqslant Q_t \cdot \tau$. Furthermore, since $Q_{t+1} \cdot \delta_x \leqslant \delta_x \cdot Q_t$, $Q_t^2 = Q_t$ and $I_n \leqslant Q_{t+1}$, we have that

$$Q_{t+1} \cdot \delta_x \cdot Q_t \leqslant \delta_x \cdot Q_t^2 = \delta_x \cdot Q_t, \qquad \delta_x \cdot Q_t = I_n \cdot \delta_x \cdot Q_t \leqslant Q_{t+1} \cdot \delta_x \cdot Q_t.$$

Therefore, we have proved that

$$Q_t \cdot \tau = \tau, \quad Q_{t+1} \cdot \delta_x \cdot Q_t = \delta_x \cdot Q_t, \tag{19}$$

for all $t \in \mathbb{N}_0$ and $x \in X$.

Now we have that

$$[\![\mathcal{A}_{Q_k}]\!](\varepsilon) = \sigma \cdot Q_k \cdot \tau = \sigma \cdot \tau = [\![\mathcal{A}]\!](\varepsilon),$$

and for $u = x_1 x_2 \ldots x_s \in X^+$, where $x_1, x_2, \ldots, x_s \in X$ and $s \leqslant k$, we have that

$$[\![\mathcal{A}]\!](u) = \sigma \cdot \delta_{x_1} \cdot \delta_{x_2} \cdot \ldots \cdot \delta_{x_s} \cdot \tau \leqslant \sigma \cdot Q_k \cdot \delta_{x_1} \cdot Q_k \cdot \delta_{x_2} \cdot \ldots \cdot Q_k \cdot \delta_{x_s} \cdot Q_k \cdot \tau = [\![\mathcal{A}_{Q_k}]\!](u),$$

and

$$
\begin{aligned}
[\![\mathcal{A}_{Q_k}]\!](u) &= \sigma \cdot Q_k \cdot \delta_{x_1} \cdot Q_k \cdot \delta_{x_2} \cdot \ldots \cdot Q_k \cdot \delta_{x_s} \cdot Q_k \cdot \tau \leqslant \sigma \cdot Q_s \cdot \delta_{x_1} \cdot Q_{s-1} \cdot \delta_{x_2} \cdot \ldots \cdot Q_1 \cdot \delta_{x_s} \cdot Q_0 \cdot \tau \\
&= \sigma \cdot \delta_{x_1} \cdot Q_{s-1} \cdot \delta_{x_2} \cdot \ldots \cdot Q_1 \cdot \delta_{x_s} \cdot Q_0 \cdot \tau = \sigma \cdot \delta_{x_1} \cdot \delta_{x_2} \cdot \ldots \cdot Q_1 \cdot \delta_{x_s} \cdot Q_0 \cdot \tau \\
&= \ldots = \sigma \cdot \delta_{x_1} \cdot \delta_{x_2} \cdot \ldots \cdot \delta_{x_s} \cdot Q_0 \cdot \tau = \sigma \cdot \delta_{x_1} \cdot \delta_{x_2} \cdot \ldots \cdot \delta_{x_s} \cdot \tau = [\![\mathcal{A}]\!](u).
\end{aligned}
$$

Therefore, we have proved that $[\![\mathcal{A}_{Q_k}]\!](u) = [\![\mathcal{A}]\!](u)$, for every $u \in X^*$ such that $|u| = s \leqslant k$, which means that the fuzzy automaton $\mathcal{A}_{Q_k}$ is k-equivalent to $\mathcal{A}$.

(c) This follows by the fact that $[\![\mathcal{A}_{Q_k}]\!] = [\![\mathcal{A}_{L|R}]\!]$ (cf. [16]).

(d) Assume that $Q_s = Q_{s+1}$, for some $s \leqslant k$. As shown in [17], then $Q_s = Q_t$, for every $t \geqslant s$, $t \in \mathbb{N}_0$, and $[\![\mathcal{A}_{Q_s}]\!] = [\![\mathcal{A}]\!]$ holds. Therefore, we have that $Q_k = Q_s$, so $[\![\mathcal{A}_{Q_k}]\!] = [\![\mathcal{A}_{L|R}]\!] = [\![\mathcal{A}_{Q_s}]\!] = [\![\mathcal{A}]\!]$. $\square$

It is worth noting that in assertion (c) of Theorem 4.1 we are talking about the r-decomposition of the matrix Q_k for some $r \in \mathbb{N}$ for which $\rho(Q_k) \leqslant r \leqslant d(Q_k)$, rather than on the $\rho(k)$-decomposition, which would give a smallest row automaton. The reason is that the r-decomposition could be done using an algorithm provided in [16], which consists in removing those row vectors of Q_k that can be represented as linear combinations of other row vectors. The result of that procedure strongly depends on the choice and order of the vectors we remove, so that one can get an r-decomposition for any r such that $\rho(Q_k) \leqslant r \leqslant d(Q_k)$.

Let us also note that the use of the r-decompositions in k-reductions (and reductions in general) is useless if the underlying complete residuated lattice $\mathbb{L}$ satisfies the condition that $a \vee b = 1$ implies that $a = 1$ or $b = 1$, since in this case we have that $\rho(Q_k) = d(Q_k)$ (as shown in [16]), so there is no reduction whatsoever. For instance, this holds when $\mathbb{L}$ is linearly ordered, what includes all the cases when $\mathbb{L}$ is the structure defined on the real unit interval by means of triangular norms.

Similarly to Theorem 4.1, we can prove the following theorem, which provides an alternative way for k-reduction.

Theorem 4.2 *Let $\mathcal{A} = (A, \sigma, \delta, \tau)$ be a fuzzy finite automaton over $\mathbb{L}$ and X with n states.*

Let us inductively define a sequence of matrices $\{P_k\}_{k \in \mathbb{N}} \subset \mathbb{L}^{n \times n}$ as follows:

$$P_0 = \sigma \backslash \sigma, \qquad P_{k+1} = P_k \wedge \bigwedge_{x \in X} [\delta_x \backslash (P_k \cdot \delta_x)], \quad \text{for every } k \in \mathbb{N}. \tag{20}$$

Then for an arbitrary $k \in \mathbb{N}$ the following statements hold:

   (a) *P_k is a fuzzy quasi-order matrix;*
   (b) *$\mathcal{A}_{P_k}$ is k-equivalent to $\mathcal{A}$;*
   (c) *if $P_k = L \cdot R$ is an r-factorization of P_k, for some $r \leqslant d(P_k)$, then $\mathcal{A}_{L|R}$ is k-equivalent to $\mathcal{A}$;*
   (d) *if $P_s = P_{s+1}$, for some $s \leqslant k$, then $P_k = P_s$ and both $\mathcal{A}_{P_k}$ and $\mathcal{A}_{L|R}$ are equivalent to $\mathcal{A}$.*

The third way to perform the k-reduction is given by the next theorem.

Theorem 4.3 *Let $\mathcal{A} = (A, \sigma, \delta, \tau)$ be a fuzzy finite automaton over $\mathbb{L}$ and X with n states.*
Let us define a sequence of matrices $\{\widehat{Q}_k\}_{k \in \mathbb{N}} \subset \mathbb{L}^{n \times n}$ as follows:

$$\widehat{Q}_k = \bigwedge_{|u| \leqslant k} \tau_u / \tau_u, \tag{21}$$

for each $k \in \mathbb{N}$. Then for an arbitrary $k \in \mathbb{N}_0$ the following statements hold:
  (a) *$\widehat{Q}_k$ is a fuzzy quasi-order matrix and $Q_k \leqslant \widehat{Q}_k$, where Q_k is as in Theorem 4.1;*
  (b) *$\mathcal{A}_{\widehat{Q}_k}$ is k-equivalent to $\mathcal{A}$;*
  (c) *if $\widehat{Q}_k = L \cdot R$ is an r-factorization of $\widehat{Q}_k$, for some $r \leqslant d(\widehat{Q}_k)$, then $\mathcal{A}_{L|R}$ is k-equivalent to $\mathcal{A}$.*

Proof. (a) We have that $\widehat{Q}_k$ is a fuzzy quasi-order matrix as the infimum of a family of fuzzy quasi-order matrices τ_u / τ_u, $u \in X^*$, $|u| \leqslant k$.

Consider an arbitrary $u \in X^*$ such that $|u| \leqslant k$. Then $u = x_1 x_2 \ldots x_s$, where $x_1, x_2, \ldots, x_s \in X$ and $s \leqslant k$, and we get

$$\begin{aligned}
Q_k \cdot \tau_u &= Q_k \cdot \delta_{x_1} \cdot \delta_{x_2} \cdot \ldots \cdot \delta_{x_s} \cdot \tau \leqslant Q_s \cdot \delta_{x_1} \cdot \delta_{x_2} \cdot \ldots \cdot \delta_{x_s} \cdot \tau \leqslant \delta_{x_1} \cdot Q_{s-1} \cdot \delta_{x_2} \cdot \ldots \cdot \delta_{x_s} \cdot \tau \\
&\leqslant \delta_{x_1} \cdot \delta_{x_2} \cdot Q_{s-2} \cdot \ldots \cdot \delta_{x_s} \cdot \tau \leqslant \ldots \leqslant \delta_{x_1} \cdot \delta_{x_2} \cdot \ldots \cdot Q_1 \cdot \delta_{x_s} \cdot \tau \leqslant \delta_{x_1} \cdot \delta_{x_2} \cdot \ldots \cdot \delta_{x_s} \cdot Q_0 \cdot \tau \\
&\leqslant \delta_{x_1} \cdot \delta_{x_2} \cdot \ldots \cdot \delta_{x_s} \cdot \tau = \tau_u.
\end{aligned}$$

Therefore, $Q_k \cdot \tau_u \leqslant \tau_u$. i.e., $Q_k \leqslant \tau_u / \tau_u$, for every $u \in X^*$ such that $|u| \leqslant k$, so

$$Q_k \leqslant \bigwedge_{|u| \leqslant k} \tau_u / \tau_u = \widehat{Q}_k.$$

(b) According to the definition of $\widehat{Q}_k$ we obtain that $\widehat{Q}_k \cdot \tau_u \leqslant \tau_u$, for every $u \in X^*$, $|u| \leqslant k$, and since $\tau_u = I_n \cdot \tau_u \leqslant \widehat{Q}_k \cdot \tau_u$, we conclude that

$$\widehat{Q}_k \cdot \tau_u = \tau_u, \quad \text{for every } u \in X^*, |u| \leqslant k. \tag{22}$$

Consider again an arbitrary $u \in X^*$ such that $|u| \leqslant k$. According to (22) we get

$$\begin{aligned}
[\![\mathcal{A}_{\widehat{Q}_k}]\!](u) &= \sigma \cdot \widehat{Q}_k \cdot \delta_{x_1} \cdot \widehat{Q}_k \cdot \delta_{x_2} \cdot \widehat{Q}_k \cdot \ldots \cdot \widehat{Q}_k \cdot \delta_{x_s} \cdot \widehat{Q}_k \cdot \tau = \sigma \cdot \widehat{Q}_k \cdot \delta_{x_1} \cdot \widehat{Q}_k \cdot \delta_{x_2} \cdot \widehat{Q}_k \cdot \ldots \cdot \widehat{Q}_k \cdot \delta_{x_s} \cdot \tau \\
&= \sigma \cdot \widehat{Q}_k \cdot \delta_{x_1} \cdot \widehat{Q}_k \cdot \delta_{x_2} \cdot \widehat{Q}_k \cdot \ldots \cdot \widehat{Q}_k \cdot \tau_{x_s} = \sigma \cdot \widehat{Q}_k \cdot \delta_{x_1} \cdot \widehat{Q}_k \cdot \delta_{x_2} \cdot \widehat{Q}_k \cdot \ldots \cdot \tau_{x_s} = \ldots \\
&= \sigma \cdot \widehat{Q}_k \cdot \delta_{x_1} \cdot \widehat{Q}_k \cdot \delta_{x_2} \cdot \widehat{Q}_k \cdot \tau_{x_3 \ldots x_s} = \sigma \cdot \widehat{Q}_k \cdot \delta_{x_1} \cdot \widehat{Q}_k \cdot \delta_{x_2} \cdot \tau_{x_3 \ldots x_s} = \\
&= \sigma \cdot \widehat{Q}_k \cdot \delta_{x_1} \cdot \widehat{Q}_k \cdot \tau_{x_2 x_3 \ldots x_s} = \sigma \cdot \widehat{Q}_k \cdot \delta_{x_1} \cdot \tau_{x_2 x_3 \ldots x_s} = \sigma \cdot \widehat{Q}_k \cdot \tau_{x_1 x_2 x_3 \ldots x_s} \\
&= \sigma \cdot \tau_{x_1 x_2 x_3 \ldots x_s} = \sigma \cdot \tau_u = [\![\mathcal{A}]\!](u).
\end{aligned}$$

Therefore, we have proved that $\mathcal{A}_{\widehat{Q}_k}$ is k-equivalent to $\mathcal{A}$.

(c) This is proved in the same way as the statement (c) in Theorem 4.1. $\square$

In a similar way we can prove the theorem that provides the fourth way to perform the k-reduction.

Theorem 4.4 *Let $\mathcal{A} = (A, \sigma, \delta, \tau)$ be a fuzzy finite automaton over $\mathbb{L}$ and X with n states.*
Let us define a sequence of matrices $\{\widehat{P}_k\}_{k \in \mathbb{N}} \subset \mathbb{L}^{n \times n}$ as follows:

$$\widehat{P}_k = \bigwedge_{|u| \leqslant k} \sigma_u \backslash \sigma_u, \tag{23}$$

for every $k \in \mathbb{N}$.

Then for an arbitrary $k \in \mathbb{N}$ the following statements hold:

(a) $\widehat{P_k}$ *is a fuzzy quasi-order matrix and $P_k \leqslant \widehat{P_k}$, where P_k is as in Theorem 4.2;*

(b) $\mathcal{A}_{\widehat{P_k}}$ *is k-equivalent to $\mathcal{A}$;*

(c) *if $\widehat{P_k} = L \cdot R$ is an r-factorization of $\widehat{P_k}$, for some $r \leqslant d(\widehat{P_k})$, then $\mathcal{A}_{L|R}$ is k-equivalent to $\mathcal{A}$.*

The previous four theorems provide four different methods for *k*-reduction of fuzzy finite automata. The question naturally arises: Do we really need all four methods? In other words, are any of these methods better than others, so that others are not necessary? In the sequel, we will show that each of these methods has some advantages, but also some disadvantages in relation to the others, as well as in relation to methods for full state reduction (reduction that ensures full equivalence) provided in [17].

First we note that the sequence of matrices we use in computing the greatest right invariant (and also left invariant) fuzzy quasi-order matrix may be infinite, and in such cases, the efficiency of reductions using such matrices becomes questionable. In contrast, *k*-reductions by means of the *k*th members of sequences defined by (17) and (20) can always be realized in a finite number of steps. In addition, members of those arrays with smaller indices produce automata with fewer states than members with larger indices (see example below), which means that *k*-reductions generally yield fewer fuzzy finite automata than reductions that result in strictly equivalent fuzzy finite automata.

Regarding *k*-reductions by means of the *k*th members of sequences defined by (17) and (21), by $Q_k \leqslant \widehat{Q_k}$ it follows that $\widehat{Q_k}$ generally yields better reduction than Q_k (see Example 4.5), for each $k \in \mathbb{N}_0$. However, computing the matrix $\widehat{Q_k}$ can be significantly more difficult than computing the matrix Q_k, because it requires computing the vectors τ_u, for $u \in X^*$, $|u| \leqslant k$, and their the number can be more than m^k, where *m* is the number of input letters.

Finally, as far as *k*-reductions by means of the *k*th members of sequences defined by (17) and (20) are concerned, they are not comparable. In some cases one of them will give better results, and in other cases the another one. In Example 4.5 we have a case where P_k, for each $k \geqslant 1$, does not perform reduction, while Q_k does, but the opposite would happen on the reverse automaton of the automaton considered in that example. The same can be said for *k*-reductions by means of the *k*th members of sequences defined by (21) and (23).

In the following example, we assumed the structure of membership values to be the two-element Boolean algebra $\mathbb{B}$. The reason why we decided so is that $\mathbb{B}$ is a subalgebra of every complete residuated lattice, and every automaton over the two-element Boolean algebra $\mathbb{B}$ can also be considered an automaton over an arbitrary complete residuated lattice.

Example 4.5 Let $\mathcal{A} = (A, \sigma, \delta, \tau)$ be a fuzzy finite automaton over the two-element Boolean algebra $\mathbb{B} = \{0, 1\}$ and an inpit alphabet $X = \{x, y\}$ given by

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

Applying formula (17) we get

$$Q_0 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad Q_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad Q_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix},$$

$$Q_3 = Q_4 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix},$$

with $d(Q_0) = 2$, $d(Q_1) = 3$, $d(Q_2) = 4$ and $d(Q_3) = 5$, where $Q_3 = Q_4$ is the greatest right invariant fuzzy quasi-order matrix, and applying (21) we get

$$\widehat{Q}_0 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad \widehat{Q}_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix},$$

with $d(\widehat{Q}_0) = 2$ and $d(\widehat{Q}_1) = 3$, where $\widehat{Q}_1$ is the greatest weakly right invariant fuzzy quasi-order matrix.

We have that $\mathcal{A}_{Q_k}$ is k-equivalent to $\mathcal{A}$, for each $k \in \{0,1,2\}$, while $\mathcal{A}_{Q_3}$ is strictly equivalent to $\mathcal{A}$, and we also have that $|\mathcal{A}_{Q_0}| < |\mathcal{A}_{Q_1}| < |\mathcal{A}_{Q_2}| < |\mathcal{A}_{Q_3}|$. On the other hand, $\mathcal{A}_{\widehat{Q}_0}$ is 0-equivalent to $\mathcal{A}$, while $\mathcal{A}_{\widehat{Q}_1}$ is strictly equivalent to $\mathcal{A}$. We see that $\widehat{Q}_1$ provides better reduction than Q_3.

The sequence computed according to formula (20) is the following:

$$P_0 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad P_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad P_2 = P_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

with $d(P_0) = 2$, $d(P_1) = 6$ and $d(P_2) = 6$, $P_2 = P_3$ is the greatest left invariant fuzzy quasi-order matrix, while the sequence computed using (23) is the following:

$$\widehat{P}_0 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \widehat{P}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \widehat{P}_2 = \widehat{P}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

with $d(\widehat{P}_0) = 2$, $d(\widehat{P}_1) = 6$ and $d(\widehat{P}_2) = 6$, while $\widehat{P}_2$ is the greatest weakly left invariant fuzzy quasi-order matrix. Although $\mathcal{A}_{P_1}$ and $\mathcal{A}_{\widehat{P}_1}$ are 1-equivalent to $\mathcal{A}$, and $\mathcal{A}_{P_2}$ and $\mathcal{A}_{\widehat{P}_2}$ are strictly equivalent to $\mathcal{A}$, this is of no significance as there is no any reduction.

5   Complexity issues

Let n denote the number of states of a fuzzy finite automaton $\mathcal{A} = (A, \sigma, \delta, \tau)$ and m the number of letters in the input alphabet X, and let $c_\vee$, $c_\wedge$, $c_\otimes$ and $c_\rightarrow$ be respectively computational costs of the operations $\vee$, $\wedge$, $\otimes$ and $\rightarrow$ in the underlying complete residuated lattice $\mathbb{L}$. If $\mathbb{L}$ is linearly ordered, we can assume that $c_\vee = c_\wedge = 1$, and if $\mathbb{L}$ is the Gödel structure, we can also assume that $c_\otimes = c_\rightarrow = 1$.

First we consider the computational time of the procedure from Theorem 4.1 (or Theorem 4.2), for a given $k \in \mathbb{N}$. It is clear that the time required to compute Q_0 is $O(n^2 c_\rightarrow)$. When we have computed Q_s, for some $s \in \mathbb{N}$, $s < k$, and we are computing Q_{s+1} from Q_s, we have the following:

1) For a fixed $x \in X$, the time required to compute the product $\delta_x \cdot Q_s$ is $O(n^3(c_\otimes + c_\vee))$, and when this product is computed, we need an additional time $O(n^3(c_\rightarrow + c_\wedge))$ to compute the residual $(\delta_x \cdot Q_s)/\delta_x$. Thus, the total time required to compute $(\delta_x \cdot Q_s)/\delta_x$ is $O(n^3(c_\rightarrow + c_\wedge + c_\otimes + c_\vee))$.

2) Now, to compute $(\delta_x \cdot Q_s)/\delta_x$ for all $x \in X$ we need time $O(mn^3(c_\rightarrow + c_\wedge + c_\otimes + c_\vee))$.

3) Next, when all matrices from 2) have been computed, to compute all infima in $Q_s \wedge \bigwedge_{x \in X}(\delta_x \cdot Q_s)/\delta_x$ and obtain Q_{s+1} we need time $O(n^2 m c_\wedge)$.

4) Finally, the total time required to compute Q_{s+1} from Q_s is $O(mn^3(c_\rightarrow + c_\wedge + c_\otimes + c_\vee))$.

Therefore, the time required to compute all matrices $Q_0, Q_1, \ldots, Q_k$, i.e., the total computational time of the procedure from Theorem 4.1, amounts $O(kmn^3(c_\rightarrow + c_\wedge + c_\otimes + c_\vee))$. This is also the total computational time of the procedure from Theorem 4.2. That computational time can be even better in cases where $Q_s = Q_{s+1}$, for some $s < k$, because then we do not have to compute all the matrices between Q_{s+1} and Q_k which are all equal to each other. However, to achieve that better computational time, after computing the matrix Q_{s+1} we need to check whether $Q_{s+1} = Q_s$. The time required for such a check is $O(n^2)$, and for all such checks it is at most $O(kn^2)$, which obviously does not affect the total computational time order of our procedure.

Next we consider the computational time of the procedure from Theorem 4.3 (or Theorem 4.4), for a given $k \in \mathbb{N}$. To compute $\widehat{Q}_k$ we have to compute the family of vectors $\{\tau_u\}_{|u| \leqslant k}$. That family forms a perfect m-ary tree with the root corresponding to the vector τ, and computing the members of the family is reduced to filling that tree level by level, starting from the root. Consequently, the number of members of that family is at most $O(m^k)$. When a vector τ_u, for some $u \in X^*$, $|u| < k$, is computed, then for each $x \in X$ we compute τ_{xu} according to the formula $\tau_{xu} = \delta_x \cdot \tau_u$. Therefore, the time needed to compute this product, i.e., the time nedded to compute each member of the considered family, is $O(n^2(c_\otimes + c_\vee))$. Moreover, when τ_u, for some $u \in X^+$, $|u| \leqslant k$, is computed, the residual τ_u/τ_u can be computed in time $O(n^2 c_\rightarrow)$, so the time required to compute a single τ_u and the residual τ_u/τ_u is $O(n^2(c_\otimes + c_\vee + c_\rightarrow))$, and the total time required to compute the whole family $\{\tau_u\}_{|u| \leqslant k}$ and all residuals τ_u/τ_u, for $u \in X^*$, $|u| \leqslant k$, amounts $O(m^k n^2(c_\otimes + c_\vee + c_\rightarrow))$. Finally, to compute $\widehat{Q}_k$ we have to apply the operation $\wedge$ between matrices τ_u/τ_u, $u \in X^*$, $|u| \leqslant k$, at most $O(m^k)$ times, and since the computational time for a single application of the operation $\wedge$ is $O(n^2 c_\wedge)$, the total computational time for all such operations is $O(m^k n^2 c_\wedge)$. Hence, the total time required to compute $\widehat{Q}_k$, i.e., the total computational time of the procedure from Theorem 4.3 is $O(m^k n^2(c_\otimes + c_\vee + c_\rightarrow + c_\wedge))$. This is also the total computational time of the procedure from Theorem 4.4.

Applying the technique of [12], the procedures from Theorems 4.1–4.4 can be improved so that the factor n^2 in their complexity estimate is reduced to the sum of n and the number of non-zero fuzzy transitions of the input fuzzy automaton $\mathcal{A}$.

References

[1] Radim Bělohlávek & Vilém Vychodil (2005): *Fuzzy Equational Logic*. In: *Fuzzy Equational Logic, Studies in Fuzziness and Soft Computing* 186, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 139–170, doi:`10.1007/11376422_3`.

[2] Radim Bělohlávek (2002): *Fuzzy Relational Systems: Foundations and Principles*. Kluwer Academic Publishers, USA, doi:`10.1007/978-1-4615-0633-1`.

[3] Radim Bělohlávek & Michal Krupka (2009): *On Approximate Minimization of Fuzzy Automata*. Journal of Multiple-Valued Logic and Soft Computing 15(2-3), pp. 125–135, doi:`10.1142/9789812709677_0194`.

[4] Miroslav Ćirić, Aleksandar Stamenković, Jelena Ignjatović & Tatjana Petković (2007): *Factorization of Fuzzy Automata*. In Erzsébet Csuhaj-Varjú & Zoltán Ésik, editors: *Fundamentals of Computation Theory*, Lecture Notes in Computer Science 4639, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 213–225, doi:`10.1007/978-3-540-74240-1_19`.

[5] Miroslav Ćirić, Aleksandar Stamenković, Jelena Ignjatović & Tatjana Petković (2010): *Fuzzy relation equations and reduction of fuzzy automata*. Journal of Computer and System Sciences 76, pp. 609–633, doi:`10.1016/j.jcss.2009.10.015`.

[6] Jelena Ignjatović, Miroslav Ćirić, Branimir Šešelja & Andreja Tepavčević (2015): *Fuzzy relational inequalities and equations, fuzzy quasi-orders, closures and openings of fuzzy sets*. Fuzzy Sets and Systems 260, pp. 1–24, doi:`10.1016/j.fss.2014.05.006`.

[7] Lvzhou Li & Daowen Qiu (2015): *On the State Minimization of Fuzzy Automata*. IEEE Transactions on Fuzzy Systems 23(2), pp. 434–443, doi:`10.1109/TFUZZ.2014.2315620`.

[8] Yongming Li (2008): *Approximation and robustness of fuzzy finite automata*. International Journal of Approximate Reasoning 47(2), pp. 247–257, doi:`10.1016/j.ijar.2007.05.004`.

[9] Ivana Micić, Zorana Jančić & Stefan Stanimirović (2022): *Computation of solutions to certain nonlinear systems of fuzzy relation inequations*. In Dimitros Poulakis & George Rahonis, editors: *Algebraic Informatics, 9th International Conference, CAI 2022*, Lecture Notes in Computer Science 13706, Thessaloniki, Greece, pp. 192–202, doi:`10.1007/978-3-031-19685-0_14`.

[10] Ivana Micić, Linh Anh Nguyen & Stefan Stanimirović (2022): *Characterization and computation of approximate bisimulations for fuzzy automata*. Fuzzy Sets and Systems 442, pp. 331–350, doi:`10.1016/j.fss.2022.05.003`.

[11] Linh Anh Nguyen (2023): *Fuzzy simulations and bisimulations between fuzzy automata*. International Journal of Approximate Reasoning, pp. 113–131, doi:`10.1016/j.ijar.2023.02.002`.

[12] Linh Anh Nguyen, Ivana Micić & Stefan Stanimirović (2023): *Depth-Bounded Fuzzy Simulations and Bisimulations between Fuzzy Automata*, doi:`10.48550/arXiv.2307.03318`. arXiv:`2307.03318`.

[13] Linh Anh Nguyen, Ivana Micić & Stefan Stanimirović (2023): *Fuzzy minimax nets*. IEEE Transactions on Fuzzy Systems 31(8), pp. 2799–2808, doi:`10.1109/TFUZZ.2023.3237936`.

[14] Sha Qiao, Ping Zhu & Jun e Feng (2022): *Fuzzy bisimulations for nondeterministic fuzzy transition systems*. IEEE Transactions on Fuzzy Systems 31(7), pp. 2450–2463, doi:`10.1109/TFUZZ.2022.3227400`.

[15] Sha Qiao, Ping Zhu & Witold Pedrycz (2023): *Approximate bisimulations for fuzzy-transition systems*. Fuzzy Sets and Systems, p. 108533, doi:`10.1016/j.fss.2023.108533`.

[16] Aleksandar Stamenković, Miroslav Ćirić & Milan Bašić (2018): *Ranks of fuzzy matrices. Applications in state reduction of fuzzy automata*. Fuzzy Sets and Systems 333, pp. 124–139, doi:`10.1016/j.fss.2017.05.028`.

[17] Aleksandar Stamenković, Miroslav Ćirić & Jelena Ignjatović (2014): *Reduction of fuzzy automata by means of fuzzy quasi-orders*. Information Sciences 275, pp. 168–198, doi:`10.1016/j.ins.2014.02.028`.

[18] Stefan Stanimirović & Ivana Micić (2022): *On the solvability of weakly linear systems of fuzzy relation equations*. Information Sciences 607, pp. 670–687, doi:`10.1016/j.ins.2022.05.111`.

[19] Stefan Stanimirović, Ivana Micić & Miroslav Ćirić (2022): *Approximate bisimulations for fuzzy automata over complete Heyting algebras*. IEEE Transactions on Fuzzy Systems 30(2), pp. 437–447, doi:10.1109/TFUZZ.2020.3039968.

[20] Chao Yang & Yongming Li (2018): *ε-bisimulation relations for fuzzy automata*. IEEE Transactions on Fuzzy Systems 26(4), pp. 2017–2029, doi:10.1109/TFUZZ.2017.2760278.

[21] Chao Yang & Yongming Li (2020): *Approximate bisimulations and state reduction of fuzzy automata under fuzzy similarity measures*. Fuzzy Sets and Systems 391, pp. 72–95, doi:10.1016/j.fss.2019.07.010.