

Computation of the greatest simulations and bisimulations between fuzzy automata[☆]

Miroslav Ćirić^{a,*}, Jelena Ignjatović^a, Ivana Jančić^a, Nada Damljanović^b

^a University of Niš, Faculty of Sciences and Mathematics, Višegradska 33, 18000 Niš, Serbia

^b University of Kragujevac, Technical Faculty in Čačak, Svetog Save 65, P. O. Box 131, 32000 Čačak, Serbia

Received 28 March 2011; received in revised form 20 December 2011; accepted 21 May 2012

Available online 28 May 2012

Abstract

Recently, two types of simulations (forward and backward simulations) and four types of bisimulations (forward, backward, forward–backward, and backward–forward bisimulations) between fuzzy automata have been introduced. If there is at least one simulation/bisimulation of some of these types between the given fuzzy automata, it has been proved that there is the greatest simulation/bisimulation of this kind. In the present paper, for any of the above-mentioned types of simulations/bisimulations we provide an efficient algorithm for deciding whether there is a simulation/bisimulation of this type between the given fuzzy automata, and for computing the greatest one, whenever it exists. The algorithms are based on the method developed in Ignjatović et al. [On the greatest solutions to weakly linear systems of fuzzy relation inequalities and equations, *Fuzzy Sets Syst.* 161 (2010) 3081–3113], which comes down to the computing of the greatest post-fixed point, contained in a given fuzzy relation, of an isotone function on the lattice of fuzzy relations.

© 2012 Elsevier B.V. All rights reserved.

Keywords: Fuzzy automaton; Simulations; Bisimulations; Fuzzy relation; Fuzzy relation inequality; Post-fixed point; Complete residuated lattice

1. Introduction

From the very beginning of the theory of fuzzy sets, fuzzy automata and languages are studied as a means for bridging the gap between the precision of computer languages and vagueness and imprecision, which are frequently encountered in the study of natural languages. Study of fuzzy automata and languages was initiated in 1960s by Santos [77–79], Wee [85], Wee and Fu [86], and Lee and Zadeh [45]. From late 1960s until early 2000s mainly fuzzy automata and languages with membership values in the Gödel structure have been considered (cf., e.g., [22,24,55]). The idea of studying fuzzy automata with membership values in some structured abstract set comes back to Wechler [84], and in recent years researcher's attention has been aimed mostly to fuzzy automata with membership values in complete residuated lattices, lattice-ordered monoids, and other kinds of lattices. Fuzzy automata taking membership values in

[☆] Research supported by Ministry Education and Science, Republic of Serbia, Grant no. 174013.

* Corresponding author. Tel.: +381 18224492; fax: +381 18533014.

E-mail addresses: miroslav.ciric@pmf.edu.rs, mciric@pmf.ni.ac.rs (M. Ćirić), jelena.ignjatovic@pmf.edu.rs (J. Ignjatović), ivanajancic84@gmail.com (I. Jančić), nada@tfc.kg.ac.rs (N. Damljanović).

a complete residuated lattice were first studied by Qiu in [63,64], where some basic concepts were discussed, and later, Qiu and his coworkers have carried out extensive research of these fuzzy automata (cf. [65,67,87–91]). From a different point of view, fuzzy automata with membership values in a complete residuated lattice were studied by Ignjatović, Ćirić and their coworkers in [18–20,35–37,39,40,81,82]. During the decades, fuzzy automata and languages have got a wide field of applications, including lexical analysis, description of natural and programming languages, learning systems, control systems, neural networks, knowledge representation, clinical monitoring, pattern recognition, error correction, databases, discrete event systems, and many other areas.

One of the most important problems of automata theory is to determine whether two given automata are equivalent, what usually means to determine whether their behaviour is identical. In the context of deterministic, nondeterministic and fuzzy automata the behaviour of an automaton is understood to be the language or fuzzy language that is recognized by it, and two automata are considered equivalent, or more precisely language-equivalent, if they recognize the same language. For deterministic finite automata the equivalence problem is solvable in polynomial time, but for nondeterministic and fuzzy finite automata it is computationally hard (PSPACE-complete). Another important issue is to express the language-equivalence of two automata as a relation between their states, if such relationship exists, or find some kind of relations between states which would approximate the language-equivalence. The language-equivalence of two deterministic automata can be expressed in terms of relationships between their states, but in the case of non-deterministic and fuzzy automata the problem is more complicated, and we can only examine various approximations of the language-equivalence.

The most widely used concept used to model “equivalence” between states of automata is that of bisimulation. Bisimulations have been introduced by Milner [52] and Park [57] in computer science, i.e., in concurrency theory, but roughly at the same time they have been also discovered in some areas of mathematics, e.g., in modal logic and set theory. They are employed today in a many areas of computer science, such as functional languages, object-oriented languages, types, data types, domains, databases, compiler optimizations, program analysis, verification tools, etc. For more information about bisimulations we refer to [1,15,27,28,51,53,54,70,76].

Bisimulations have two main roles. First, they are used to model the equivalence between states of two different automata and approximate the language-equivalence. As we have already said, the problem of determining whether two given automata are language-equivalent is computationally hard, but we will see that the problem of determining whether two given automata are “bisimulation equivalent”, i.e., whether there is a bisimulation between them, is solvable in polynomial time. For this reason, bisimulations are generally considered as the best way to model the language-equivalence, because they give a close enough approximation of the language-equivalence and are efficiently computable.

Bisimulations are also used to reduce the number of states of automata. To illustrate the role of the state reduction, note that in real-life applications of automata we typically start from an ordinary or fuzzy regular expression, which is then converted to a nondeterministic or fuzzy finite automaton (cf. [46,81]). However, the practical implementation usually requires a deterministic finite automaton or a deterministic fuzzy finite automaton, and the obtained nondeterministic or fuzzy automaton has to be determinized. On the other hand, determinization can cause an exponential blow up in the number of states, and in the case of fuzzy finite automata over certain structures of membership values (such as the product structure), determinization can even result in an infinite automaton (cf. [16,36,39,40,46]). That is why the number of states of a fuzzy finite automaton has to be reduced prior to determinization.

Another important example that illustrates the significance of the state reduction is modeling of discrete event systems. A discrete event system is a dynamical system whose state space is described by a discrete set, and states evolve as a result of asynchronously occurring discrete events over time [15,34]. Such systems have significant applications in many fields of computer science and engineering, such as concurrent and distributed software systems, computer and communication networks, manufacturing, transportation and traffic control systems, etc. In many situations states and state transitions, as well as control strategies, are somewhat imprecise, uncertain and vague. To take this kind of uncertainty into account, Lin and Ying extended classical discrete event systems to *fuzzy discrete event systems* by proposing a fuzzy finite automaton model [47,48]. Fuzzy discrete event systems have been since studied in [12–14,42,47–50,66,68], and they have been successfully applied to biomedical control for HIV/AIDS treatment planning, robotic control, intelligent vehicle control, waste-water treatment, examination of chemical reactions, and in other fields.

Usually, a discrete event system is modeled by a deterministic or nondeterministic finite automaton, and recently by a fuzzy finite automaton, with events modeled by input letters, and the behaviour of a discrete event system is described by the language or fuzzy language generated by the automaton. Discrete event models of complex dynamical systems

are built rarely in a monolithic manner. Instead, a modular approach is used where models of individual components are built first, followed by the composition of these models to obtain the model of the overall system. In the automaton modeling formalism the composition of individual automata (that model interacting system components) is typically formalized by the parallel composition of automata. Once a complete system model has been obtained by parallel composition of a set of automata, the resulting monolithic model can be used to analyze the properties of the system, such as safety properties, blocking properties, observability, diagnosability, controllability, etc. (cf. [15,34]). The main problem that may arise here is that the size of the state set of the parallel composition may in the worst case grow exponentially in the number of automata that are composed. This process is known as the *curse of dimensionality* in the study of complex systems composed of many interacting components. The mentioned problem may be mitigated if we adopt modular reasoning, which can make it possible to replace components in the parallel composition by smaller equivalent automata that are obtained by the state reduction of the components, and then to analyze a simpler system. For example, such an approach has been applied in [82] in conflict analysis of fuzzy discrete event systems.

The most common structures on which bisimulations have been studied are labelled transition systems, but they have also been investigated in the context of deterministic, nondeterministic, weighted, probabilistic, timed and hybrid automata. Recently, bisimulations have been discussed in the context of fuzzy automata in [11,18–20,37,62,82,83]. One can distinguish two general approaches to the concept of bisimulation for fuzzy automata. The first approach, which we encounter in [11,62,83], uses ordinary crisp relations and functions. Another approach, proposed in [18–20,37,82], is based on the use of fuzzy relations, which have been shown to provide better results both in the state reduction and the modeling of equivalence of fuzzy automata. In [19,20] the state reduction of fuzzy automata was carried out by means of certain fuzzy equivalences, which are exactly bisimulation fuzzy equivalences, and in [82] it was proved that even better results can be achieved by using suitable fuzzy quasi-orders, which are nothing but simulation fuzzy quasi-orders. Moreover, it turned out in [19,20,82] that the state reduction problem for fuzzy automata is closely related to the problem of finding solutions to certain systems of fuzzy relation equations. This enabled not only to study fuzzy automata using very powerful tools of the theory of fuzzy sets, but also, it gave a great contribution to the theory of fuzzy relational equations and has led to the development of the general theory of weakly linear fuzzy relation equations and inequalities in [38].

The same approach has been used in [18], in the study of simulations and bisimulations between fuzzy automata, where simulations and bisimulations have also been defined as fuzzy relations. There have been introduced two types of simulations (forward and backward simulations) and four types of bisimulations (forward, backward, forward–backward, and backward–forward bisimulations). There has been proved that if there is at least one simulation/bisimulation of some of these types between the given fuzzy automata, then there is the greatest simulation/bisimulation of this kind. However, there has not been given any efficient algorithm for deciding whether there is a simulation/bisimulation of some of these types between the given fuzzy automata, and for computing the greatest one, if it exists. In [18] a theorem has been proved which can be used just for checking the existence of a uniform forward bisimulation (i.e., a complete and surjective forward bisimulation) between the given fuzzy automata. According to this theorem, there is a uniform forward bisimulation between fuzzy automata \mathcal{A} and \mathcal{B} if and only if there is a special isomorphism between the factor fuzzy automata of \mathcal{A} and \mathcal{B} with respect to their greatest forward bisimulation fuzzy equivalences. Using the algorithm provided in [20], in numerous cases we can efficiently compute the greatest forward bisimulation fuzzy equivalences E on \mathcal{A} and F on \mathcal{B} . Then we can construct factor fuzzy automata \mathcal{A}/E and \mathcal{B}/F , and check whether there is an isomorphism between them that satisfies an additional condition stated in [18]. But, even when we are able to efficiently compute the greatest forward bisimulations E and F and construct the factor fuzzy automata \mathcal{A}/E and \mathcal{B}/F , it may be difficult to determine whether there is an isomorphism between \mathcal{A}/E and \mathcal{B}/F that satisfies this additional condition. This problem comes down to the well-known *graph isomorphism problem*, which is one of the few important algorithmic problems whose rough computational complexity is still not known, and it is generally accepted that it lies between P and NP-complete if $P \neq NP$ (cf. [80]). Fortunately, although no worst-case polynomial-time algorithm is known, testing graph isomorphism is usually not very hard in practice.

In this paper, for any of the above-mentioned types of simulations/bisimulations we provide an efficient algorithm for deciding whether there is a simulation/bisimulation of this type between the given fuzzy automata, and for computing the greatest one, whenever it exists. The algorithms are based on the method developed in [38], which comes down to the computing of the greatest post-fixed point, contained in a given fuzzy relation, of an isotone function on the lattice of fuzzy relations. Namely, for each type of simulations and bisimulations we determine the corresponding isotone and image-localized function ϕ on the lattice of fuzzy relations, as well as the corresponding initial fuzzy relation π ,

and the computing of the greatest simulation/bisimulation of this type we reduce to the computing of the greatest post-fixed point of ϕ contained in π . This is an iterative procedure by which we successively build a decreasing sequence of relations, starting from the relation π and using the function ϕ . If this sequence is finite, then it stabilizes and its smallest member is exactly the fuzzy relation which we are searching for, the greatest post-fixed point of ϕ contained in π . We determine sufficient conditions under which this sequence is finite, when our algorithm terminates in the finite number of steps (cf. Theorem 5.3), as well as sufficient conditions under which the infimum of this sequence is exactly the fuzzy relation which we are searching for (cf. Theorem 5.5). Modifying the algorithms for computing the greatest simulations and bisimulations we provide algorithms for computing the greatest crisp simulations and bisimulations between fuzzy automata (cf. Proposition 5.6). These algorithms always terminate in a finite number of steps, independently of the properties of the underlying structure of truth values, but we show that there are fuzzy automata such that there is a simulation or bisimulation of a given type between them, and there is not any crisp simulation/bisimulation of this type (cf. Example 6.1).

The structure of the paper is as follows. In Section 2 we give definitions of basic notions and notation concerning fuzzy sets and relations, and in Section 3 we define basic notions and notation concerning fuzzy automata. In Section 4 we recall definitions of two types of simulations and four types of bisimulations introduced in [18], and for any of them we define the corresponding function on the lattice of fuzzy relations. Section 5 contains our main results on the computing of the greatest simulations and bisimulations between fuzzy automata. Examples presented in Section 6 demonstrate the application of our algorithms and clarify relationships between different types of simulations and bisimulations.

It is worth noting that algorithms for deciding whether there are simulations and bisimulations between nondeterministic automata, and the computing of the greatest simulations and bisimulations, if they exist, were given in [17,44]. Various algorithms for the computing of the greatest bisimulation equivalences on labelled transition systems can be found in [27,28,41,56,69,72].

2. Preliminaries

The terminology and basic notions in this section are according to [6,7,22,23,43].

We will use complete residuated lattices as the structures of membership (truth) values. Residuated lattices are a very general algebraic structure and generalize many algebras with very important applications (see for example [6,7,29,33]). A *residuated lattice* is an algebra $\mathcal{L} = (L, \wedge, \vee, \otimes, \rightarrow, 0, 1)$ such that

- (L1) $(L, \wedge, \vee, 0, 1)$ is a lattice with the least element 0 and the greatest element 1,
- (L2) $(L, \otimes, 1)$ is a commutative monoid with the unit 1,
- (L3) \otimes and \rightarrow form an *adjoint pair*, i.e., they satisfy the *adjunction property*: for all $x, y, z \in L$,

$$x \otimes y \leq z \Leftrightarrow x \leq y \rightarrow z. \quad (1)$$

If, in addition, $(L, \wedge, \vee, 0, 1)$ is a complete lattice, then \mathcal{L} is called a *complete residuated lattice*. Emphasizing their monoidal structure, in some sources residuated lattices are called integral, commutative, residuated ℓ -monoids [33].

The operations \otimes (called *multiplication*) and \rightarrow (called *residuum*) are intended for modeling the conjunction and implication of the corresponding logical calculus, and supremum (\bigvee) and infimum (\bigwedge) are intended for modeling of the existential and general quantifier, respectively. An operation \leftrightarrow defined by

$$x \leftrightarrow y = (x \rightarrow y) \wedge (y \rightarrow x), \quad (2)$$

called *biresiduum* (or *biimplication*) is used for modeling the equivalence of truth values.

The most studied and applied structures of truth values, defined on the real unit interval $[0, 1]$ with $x \wedge y = \min(x, y)$ and $x \vee y = \max(x, y)$, are the *Lukasiewicz structure* (where $x \otimes y = \max(x + y - 1, 0)$, $x \rightarrow y = \min(1 - x + y, 1)$), the *Goguen (product) structure* ($x \otimes y = x \cdot y$, $x \rightarrow y = 1$ if $x \leq y$, and $= y/x$ otherwise), and the *Gödel structure* ($x \otimes y = \min(x, y)$, $x \rightarrow y = 1$ if $x \leq y$, and $= y$ otherwise). More generally, an algebra $([0, 1], \wedge, \vee, \otimes, \rightarrow, 0, 1)$ is a complete residuated lattice if and only if \otimes is a left-continuous t-norm and the residuum is defined by $x \rightarrow y = \bigvee \{u \in [0, 1] \mid u \otimes x \leq y\}$ (cf. [7]). Another important set of truth values is the set $\{a_0, a_1, \dots, a_n\}$, $0 = a_0 < \dots < a_n = 1$, with $a_k \otimes a_l = a_{\max(k+l-n, 0)}$ and $a_k \rightarrow a_l = a_{\min(n-k+l, n)}$. A special case of the latter algebras is the two-element Boolean algebra of classical logic with the support $\{0, 1\}$. The only adjoint pair on the two-element Boolean algebra

consists of the classical conjunction and implication operations. This structure of truth values we call the *Boolean structure*. A residuated lattice \mathcal{L} satisfying $x \otimes y = x \wedge y$ is called a *Heyting algebra*, whereas a Heyting algebra satisfying the prelinearity axiom $(x \rightarrow y) \vee (y \rightarrow x) = 1$ is called a *Gödel algebra*. If any finitely generated subalgebra of a residuated lattice \mathcal{L} is finite, then \mathcal{L} is called *locally finite*. For example, every Gödel algebra, and hence, the Gödel structure, is locally finite, whereas the product structure is not locally finite.

If \mathcal{L} is a complete residuated lattice, then for all $x, y, z \in L$ and any $\{y_i\}_{i \in I} \subseteq L$ the following holds:

$$x \leq y \text{ implies } x \otimes z \leq y \otimes z, \quad (3)$$

$$x \leq y \Leftrightarrow x \rightarrow y = 1, \quad (4)$$

$$x \otimes \bigvee_{i \in I} y_i = \bigvee_{i \in I} (x \otimes y_i), \quad (5)$$

$$x \otimes \bigwedge_{i \in I} y_i \leq \bigwedge_{i \in I} (x \otimes y_i), \quad (6)$$

For other properties of complete residuated lattices we refer to [6,7].

In the further text \mathcal{L} will be a complete residuated lattice. A *fuzzy subset* of a set A over \mathcal{L} , or simply a *fuzzy subset* of A , is any function from A into L . Ordinary crisp subsets of A are considered as fuzzy subsets of A taking membership values in the set $\{0, 1\} \subseteq L$. Let f and g be two fuzzy subsets of A . The *equality* of f and g is defined as the usual equality of functions, i.e., $f = g$ if and only if $f(x) = g(x)$, for every $x \in A$. The *inclusion* $f \leq g$ is also defined pointwise: $f \leq g$ if and only if $f(x) \leq g(x)$, for every $x \in A$. Endowed with this partial order the set $\mathcal{F}(A)$ of all fuzzy subsets of A forms a complete residuated lattice, in which the meet (intersection) $\bigwedge_{i \in I} f_i$ and the join (union) $\bigvee_{i \in I} f_i$ of an arbitrary family $\{f_i\}_{i \in I}$ of fuzzy subsets of A are functions from A into L defined by

$$\left(\bigwedge_{i \in I} f_i\right)(x) = \bigwedge_{i \in I} f_i(x), \quad \left(\bigvee_{i \in I} f_i\right)(x) = \bigvee_{i \in I} f_i(x),$$

and the *product* $f \otimes g$ is a fuzzy subset defined by $(f \otimes g)(x) = f(x) \otimes g(x)$, for every $x \in A$.

Let A and B be non-empty sets. A *fuzzy relation between sets A and B* (or a *fuzzy relation from A to B*) is any function from $A \times B$ into L , that is to say, any fuzzy subset of $A \times B$, and the equality, inclusion (ordering), joins and meets of fuzzy relations are defined as for fuzzy sets. In particular, a *fuzzy relation on a set A* is any function from $A \times A$ into L , i.e., any fuzzy subset of $A \times A$. To highlight the difference between fuzzy relations between two sets and those on a set, fuzzy relations between two sets will be usually denoted by small Greek letters, and fuzzy relations on a set by capital Latin letters. The set of all fuzzy relations from A to B will be denoted by $\mathcal{R}(A, B)$, and the set of all fuzzy relations on a set A will be denoted by $\mathcal{R}(A)$. The *converse* (in some sources called *inverse* or *transpose*) of a fuzzy relation $\varphi \in \mathcal{R}(A, B)$ is a fuzzy relation $\varphi^{-1} \in \mathcal{R}(B, A)$ defined by $\varphi^{-1}(b, a) = \varphi(a, b)$, for all $a \in A$ and $b \in B$. A *crisp relation* is a fuzzy relation which takes values only in the set $\{0, 1\}$, and if φ is a crisp relation of A to B , then expressions “ $\varphi(a, b) = 1$ ” and “ $(a, b) \in \varphi$ ” will have the same meaning.

For non-empty sets A, B and C , and fuzzy relations $\varphi \in \mathcal{R}(A, B)$ and $\psi \in \mathcal{R}(B, C)$, their *composition* $\varphi \circ \psi$ is a fuzzy relation from $\mathcal{R}(A, C)$ defined by

$$(\varphi \circ \psi)(a, c) = \bigvee_{b \in B} \varphi(a, b) \otimes \psi(b, c), \quad (7)$$

for all $a \in A$ and $c \in C$. If φ and ψ are crisp relations, then $\varphi \circ \psi$ is an ordinary composition of relations, i.e.,

$$\varphi \circ \psi = \{(a, c) \in A \times C \mid (\exists b \in B)(a, b) \in \varphi \ \& \ (b, c) \in \psi\},$$

and if φ and ψ are functions, then $\varphi \circ \psi$ is an ordinary composition of functions, i.e., $(\varphi \circ \psi)(a) = \psi(\varphi(a))$, for every $a \in A$. Next, if $f \in \mathcal{F}(A)$, $\varphi \in \mathcal{R}(A, B)$ and $g \in \mathcal{F}(B)$, the *compositions* $f \circ \varphi$ and $\varphi \circ g$ are fuzzy subsets of B and A , respectively, which are defined by

$$(f \circ \varphi)(b) = \bigvee_{a \in A} f(a) \otimes \varphi(a, b), \quad (\varphi \circ g)(a) = \bigvee_{b \in B} \varphi(a, b) \otimes g(b), \quad (8)$$

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

In particular, for $f, g \in \mathcal{F}(A)$ we write

$$f \circ g = \bigvee_{a \in A} f(a) \otimes g(a). \quad (9)$$

The value $f \circ g$ can be interpreted as the “degree of overlapping” of f and g . In particular, if f and g are crisp sets and φ is a crisp relation, then

$$f \circ \varphi = \{b \in B | (\exists a \in A) f(a) \otimes \varphi(a, b) \neq 0\}, \quad \varphi \circ g = \{a \in A | (\exists b \in B) \varphi(a, b) \otimes g(b) \neq 0\}.$$

Let A, B, C and D be non-empty sets. Then for any $\varphi_1 \in \mathcal{R}(A, B)$, $\varphi_2 \in \mathcal{R}(B, C)$ and $\varphi_3 \in \mathcal{R}(C, D)$ we have

$$(\varphi_1 \circ \varphi_2) \circ \varphi_3 = \varphi_1 \circ (\varphi_2 \circ \varphi_3), \quad (10)$$

and for $\varphi_0 \in \mathcal{R}(A, B)$, $\varphi_1, \varphi_2 \in \mathcal{R}(B, C)$ and $\varphi_3 \in \mathcal{R}(C, D)$ we have that

$$\varphi_1 \leq \varphi_2 \text{ implies } \varphi_1^{-1} \leq \varphi_2^{-1}, \quad \varphi_0 \circ \varphi_1 \leq \varphi_0 \circ \varphi_2, \quad \text{and} \quad \varphi_1 \circ \varphi_3 \leq \varphi_2 \circ \varphi_3. \quad (11)$$

Further, for any $\varphi \in \mathcal{R}(A, B)$, $\psi \in \mathcal{R}(B, C)$, $f \in \mathcal{F}(A)$, $g \in \mathcal{F}(B)$ and $h \in \mathcal{F}(C)$ we can easily verify that

$$(f \circ \varphi) \circ \psi = f \circ (\varphi \circ \psi), \quad (f \circ \varphi) \circ g = f \circ (\varphi \circ g), \quad (\varphi \circ \psi) \circ h = \varphi \circ (\psi \circ h) \quad (12)$$

and consequently, the parentheses in (12) can be omitted, as well as the parentheses in (10).

Finally, for all $\varphi, \varphi_i \in \mathcal{R}(A, B)$ ($i \in I$) and $\psi, \psi_i \in \mathcal{R}(B, C)$ ($i \in I$) we have that

$$(\varphi \circ \psi)^{-1} = \psi^{-1} \circ \varphi^{-1}, \quad (13)$$

$$\varphi \circ \left(\bigvee_{i \in I} \psi_i \right) = \bigvee_{i \in I} (\varphi \circ \psi_i), \quad \left(\bigvee_{i \in I} \varphi_i \right) \circ \psi = \bigvee_{i \in I} (\varphi_i \circ \psi), \quad (14)$$

$$\left(\bigvee_{i \in I} \varphi_i \right)^{-1} = \bigvee_{i \in I} \varphi_i^{-1}. \quad (15)$$

We note that if A, B and C are finite sets of cardinality $|A| = k$, $|B| = m$ and $|C| = n$, then $\varphi \in \mathcal{R}(A, B)$ and $\psi \in \mathcal{R}(B, C)$ can be treated as $k \times m$ and $m \times n$ fuzzy matrices over \mathcal{L} , and $\varphi \circ \psi$ is the matrix product. Analogously, for $f \in \mathcal{F}(A)$ and $g \in \mathcal{F}(B)$ we can treat $f \circ \varphi$ as the product of a $1 \times k$ matrix f and a $k \times m$ matrix φ (vector–matrix product), $\varphi \circ g$ as the product of a $k \times m$ matrix φ and an $m \times 1$ matrix g^t , the transpose of g (matrix–vector product), and $f \circ g$ as the scalar product of vectors f and g .

3. Fuzzy automata

In this paper we study fuzzy automata with membership values in complete residuated lattices.

In the rest of the paper, if not noted otherwise, let \mathcal{L} be a complete residuated lattice and let X be an (finite) alphabet. A *fuzzy automaton* over \mathcal{L} and X , or simply a *fuzzy automaton*, is a quadruple $\mathcal{A} = (A, \delta^A, \sigma^A, \tau^A)$, where A is a non-empty set, called the *set of states*, $\delta^A : A \times X \times A \rightarrow \mathcal{L}$ is a fuzzy subset of $A \times X \times A$, called the *fuzzy transition function*, and $\sigma^A : A \rightarrow \mathcal{L}$ and $\tau^A : A \rightarrow \mathcal{L}$ are the fuzzy subsets of A , called the *fuzzy set of initial states* and the *fuzzy set of terminal states*, respectively. We can interpret $\delta^A(a, x, b)$ as the degree to which an input letter $x \in X$ causes a transition from a state $a \in A$ into a state $b \in A$, whereas we can interpret $\sigma^A(a)$ and $\tau^A(a)$ as the degrees to which a is respectively an input state and a terminal state.

Many results presented in this paper can also be proved for fuzzy automata with infinite sets of states. However, here we deal with efficient algorithms for computing the greatest simulations and bisimulations, and for that reason we will assume that the sets of states of the considered fuzzy automata are always *finite*.

Let X^* denote the free monoid over the alphabet X , and let $\varepsilon \in X^*$ be the empty word. The function δ^A can be extended up to a function $\delta_*^A : A \times X^* \times A \rightarrow \mathcal{L}$ as follows: If $a, b \in A$, then

$$\delta_*^A(a, \varepsilon, b) = \begin{cases} 1 & \text{if } a = b, \\ 0 & \text{otherwise,} \end{cases} \quad (16)$$

and if $a, b \in A$, $u \in X^*$ and $x \in X$, then

$$\delta_*^A(a, ux, b) = \bigvee_{c \in A} \delta_*^A(a, u, c) \otimes \delta^A(c, x, b). \quad (17)$$

By (5) and Theorem 3.1 [46] (see also [63,64,67]), we have that

$$\delta_*^A(a, uv, b) = \bigvee_{c \in A} \delta_*^A(a, u, c) \otimes \delta_*^A(c, v, b), \quad (18)$$

for all $a, b \in A$ and $u, v \in X^*$, i.e., if $w = x_1 \cdots x_n$, for $x_1, \dots, x_n \in X$, then

$$\delta_*^A(a, w, b) = \bigvee_{(c_1, \dots, c_{n-1}) \in A^{n-1}} \delta^A(a, x_1, c_1) \otimes \delta^A(c_1, x_2, c_2) \otimes \cdots \otimes \delta^A(c_{n-1}, x_n, b). \quad (19)$$

Intuitively, the product $\delta^A(a, x_1, c_1) \otimes \delta^A(c_1, x_2, c_2) \otimes \cdots \otimes \delta^A(c_{n-1}, x_n, b)$ represents the degree to which the input word w causes a transition from a state a into a state b through the sequence of intermediate states $c_1, \dots, c_{n-1} \in A$, and $\delta_*^A(a, w, b)$ represents the supremum of degrees of all possible transitions from a into b caused by w . Also, we can visualize a fuzzy finite automaton \mathcal{A} representing it as a labelled directed graph whose nodes are states of \mathcal{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 $x \in X$.

It is worth noting that fuzzy automata are a natural generalization of nondeterministic and deterministic automata. Namely, if δ^A is a crisp subset of $A \times X \times A$, that is, $\delta^A : A \times X \times A \rightarrow \{0, 1\}$, and σ^A and τ^A are crisp subsets of A , then \mathcal{A} is an ordinary *nondeterministic automaton*. In other words, nondeterministic automata are fuzzy automata over the Boolean structure. They will also be called *Boolean automata*. If δ^A is a function of $A \times X$ into A , σ^A is a one-element crisp subset of A , that is, $\sigma^A = \{a_0\}$, for some $a_0 \in A$, and τ^A is a fuzzy subset of A , then \mathcal{A} is called a *deterministic fuzzy automaton*, and it is denoted by $\mathcal{A} = (A, \delta^A, a_0, \tau^A)$. In [16,21] the name *crisp-deterministic* was used. For more information on deterministic fuzzy automata we refer to [5,35,36,39,40,46]. Evidently, if δ^A is a crisp subset of $A \times X \times A$, or a function of $A \times X$ into A , then δ_*^A is also a crisp subset of $A \times X^* \times A$, or a function of $A \times X^*$ into A , respectively. A deterministic fuzzy automaton $\mathcal{A} = (A, \delta^A, a_0, \tau^A)$, where τ^A is a crisp subset of A , is an ordinary deterministic automaton.

If for any $u \in X^*$ we define a fuzzy relation δ_u^A on A by

$$\delta_u^A(a, b) = \delta_*^A(a, u, b), \quad (20)$$

for all $a, b \in A$, called the *fuzzy transition relation* determined by u , then (18) can be written as

$$\delta_{uv}^A = \delta_u^A \circ \delta_v^A, \quad (21)$$

for all $u, v \in X^*$.

The *reverse fuzzy automaton* of a fuzzy automaton $\mathcal{A} = (A, \delta^A, \sigma^A, \tau^A)$ is defined as the fuzzy automaton $\bar{\mathcal{A}} = (A, \bar{\delta}^A, \bar{\sigma}^A, \bar{\tau}^A)$ whose fuzzy transition function and fuzzy sets of initial and terminal states are defined by $\bar{\delta}^A(a_1, x, a_2) = \delta^A(a_2, x, a_1)$ for all $a_1, a_2 \in A$ and $x \in X$, $\bar{\sigma}^A = \tau^A$ and $\bar{\tau}^A = \sigma^A$. In other words, $\bar{\delta}_x^A = (\delta_x^A)^{-1}$, for each $x \in X$.

A *fuzzy language* in X^* over \mathcal{L} , or briefly a *fuzzy language*, is any fuzzy subset of X^* , i.e., any function from X^* into L . A *fuzzy language recognized by a fuzzy automaton* $\mathcal{A} = (A, \delta^A, \sigma^A, \tau^A)$, denoted as $L(\mathcal{A})$, is a fuzzy language in $\mathcal{F}(X^*)$ defined by

$$L(\mathcal{A})(u) = \bigvee_{a, b \in A} \sigma^A(a) \otimes \delta_*^A(a, u, b) \otimes \tau^A(b), \quad (22)$$

or equivalently,

$$\begin{aligned} L(\mathcal{A})(e) &= \sigma^A \circ \tau^A, \\ L(\mathcal{A})(u) &= \sigma^A \circ \delta_{x_1}^A \circ \delta_{x_2}^A \circ \cdots \circ \delta_{x_n}^A \circ \tau^A, \end{aligned} \quad (23)$$

for any $u = x_1 x_2 \cdots x_n \in X^+$, where $x_1, x_2, \dots, x_n \in X$. In other words, the equality (22) means that the membership degree of the word u to the fuzzy language $L(\mathcal{A})$ is equal to the degree to which \mathcal{A} recognizes or accepts the word u . Using notation from (8), and the second equality in (12), we can state (22) as

$$L(\mathcal{A})(u) = \sigma^A \circ \delta_u^A \circ \tau^A. \quad (24)$$

Fuzzy automata \mathcal{A} and \mathcal{B} are called *language-equivalent*, or sometimes just *equivalent*, if $L(\mathcal{A}) = L(\mathcal{B})$.

For more information on fuzzy automata with membership values in complete residuated lattices we refer to the recent papers [19,20,36,37,39,63,64,67,82,87–91], and for information on fuzzy automata over related types of lattices we refer to [35,46,58–61]. The most complete overview of the results concerning the “classical” fuzzy automata taking membership values in the Gödel or the Goguen structure can be found in the book [55].

4. Simulations and bisimulations between fuzzy automata

First we recall definitions of two types of simulations and four types of bisimulations introduced in [18].

Let $\mathcal{A} = (A, \delta^A, \sigma^A, \tau^A)$ and $\mathcal{B} = (B, \delta^B, \sigma^B, \tau^B)$ be fuzzy automata, and let $\gamma \in \mathcal{R}(A, B)$ be a non-empty fuzzy relation. We call γ a *forward simulation* if it satisfies

$$\sigma^A \leq \sigma^B \circ \gamma^{-1}, \quad (fs-1)$$

$$\gamma^{-1} \circ \delta_x^A \leq \delta_x^B \circ \gamma^{-1} \quad \text{for every } x \in X, \quad (fs-2)$$

$$\gamma^{-1} \circ \tau^A \leq \tau^B, \quad (fs-3)$$

and a *backward simulation* if

$$\tau^A \leq \gamma \circ \tau^B, \quad (bs-1)$$

$$\delta_x^A \circ \gamma \leq \gamma \circ \delta_x^B \quad \text{for every } x \in X, \quad (bs-2)$$

$$\sigma^A \circ \gamma \leq \sigma^B. \quad (bs-3)$$

Furthermore, we call γ a *forward bisimulation* if both γ and γ^{-1} are forward simulations, i.e., if γ satisfies

$$\sigma^A \leq \sigma^B \circ \gamma^{-1}, \quad \sigma^B \leq \sigma^A \circ \gamma, \quad (fb-1)$$

$$\gamma^{-1} \circ \delta_x^A \leq \delta_x^B \circ \gamma^{-1}, \quad \gamma \circ \delta_x^B \leq \delta_x^A \circ \gamma \quad \text{for every } x \in X, \quad (fb-2)$$

$$\gamma^{-1} \circ \tau^A \leq \tau^B, \quad \gamma \circ \tau^B \leq \tau^A, \quad (fb-3)$$

and a *backward bisimulation*, if both γ and γ^{-1} are backward simulations, i.e., if γ satisfies

$$\tau^A \leq \gamma \circ \tau^B, \quad \tau^B \leq \gamma^{-1} \circ \tau^A, \quad (bb-1)$$

$$\delta_x^A \circ \gamma \leq \gamma \circ \delta_x^B, \quad \delta_x^B \circ \gamma^{-1} \leq \gamma^{-1} \circ \delta_x^A \quad \text{for every } x \in X, \quad (bb-2)$$

$$\sigma^A \circ \gamma \leq \sigma^B, \quad \sigma^B \circ \gamma^{-1} \leq \sigma^A. \quad (bb-3)$$

Also, if γ is a forward simulation and γ^{-1} is a backward simulation, i.e., if γ satisfies

$$\sigma^A \leq \sigma^B \circ \gamma^{-1}, \quad \tau^B \leq \gamma^{-1} \circ \tau^A, \quad (fbb-1)$$

$$\gamma^{-1} \circ \delta_x^A = \delta_x^B \circ \gamma^{-1} \quad \text{for every } x \in X, \quad (fbb-2)$$

$$\sigma^B \circ \gamma^{-1} \leq \sigma^A, \quad \gamma^{-1} \circ \tau^A \leq \tau^B, \quad (fbb-3)$$

then γ is called a *forward–backward bisimulation*, and if γ is a backward simulation and γ^{-1} is a forward simulation, i.e., if

$$\sigma^B \leq \sigma^A \circ \gamma, \quad \tau^A \leq \gamma \circ \tau^B, \quad (bfbb-1)$$

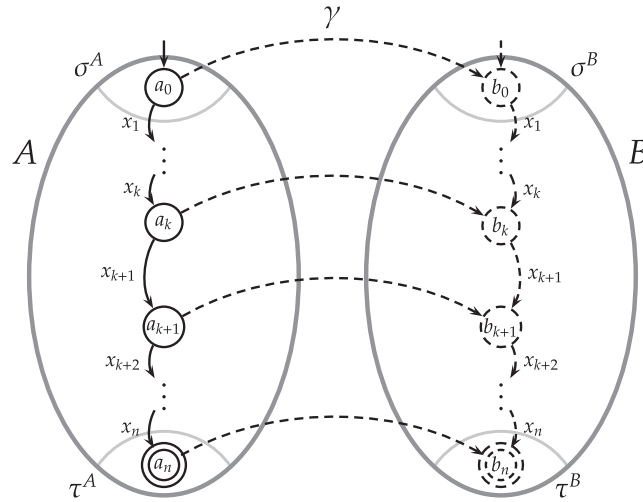


Fig. 1. Forward and backward simulation.

$$\delta_x^A \circ \gamma = \gamma \circ \delta_x^B \quad \text{for every } x \in X, \quad (bfb-2)$$

$$\sigma^A \circ \gamma \leq \sigma^B, \quad \gamma \circ \tau^B \leq \tau^A. \quad (bfb-3)$$

then γ is called a *backward-forward bisimulation*.

For the sake of simplicity, we will call γ just a *simulation* if γ is either a forward or a backward simulation, and just a *bisimulation* if γ is any of the four types of bisimulations defined above. Moreover, forward and backward bisimulations will be called *homotypic*, whereas backward-forward and forward-backward bisimulations will be called *heterotypic*. Moreover, for any $w \in \{fs, bs, fb, bb, fbb, bfb\}$, a fuzzy relation γ which satisfies (w-1), (w-2) and (w-3) will be called simply a *simulation/bisimulation of type w*.

The meaning of forward and backward simulations can be best explained in the case when \mathcal{A} and \mathcal{B} are nondeterministic (Boolean) automata. For this purpose we will use the diagram shown in Fig. 1. Let γ be a forward simulation between \mathcal{A} and \mathcal{B} and let a_0, a_1, \dots, a_n be an arbitrary successful run of the automaton \mathcal{A} on a word $u = x_1x_2 \dots x_n$ ($x_1, x_2, \dots, x_n \in X$), i.e., a sequence of states of \mathcal{A} such that $a_0 \in \sigma^A$, $(a_k, a_{k+1}) \in \delta_{x_{k+1}}^A$, for $0 \leq k \leq n-1$, and $a_n \in \tau^A$. According to (fs-1), there is an initial state $b_0 \in \sigma^B$ such that $(a_0, b_0) \in \gamma$. Suppose that for some k , $0 \leq k \leq n-1$, we have built a sequence of states b_0, b_1, \dots, b_k such that $(b_{i-1}, b_i) \in \delta_{x_i}^B$ and $(a_i, b_i) \in \gamma$, for each i , $1 \leq i \leq k$. Then $(b_k, a_{k+1}) \in \gamma^{-1} \circ \delta_{x_{k+1}}^A$, and by (fs-2) we obtain that $(b_k, a_{k+1}) \in \delta_{x_{k+1}}^B \circ \gamma^{-1}$, so there exists $b_{k+1} \in B$ such that $(b_k, b_{k+1}) \in \delta_{x_{k+1}}^B$ and $(a_{k+1}, b_{k+1}) \in \gamma$. Therefore, we have successively built a sequence b_0, b_1, \dots, b_n of states of \mathcal{B} such that $b_0 \in \sigma^B$, $(b_k, b_{k+1}) \in \delta_{x_{k+1}}^B$, for every k , $0 \leq k \leq n-1$, and $(a_k, b_k) \in \gamma$, for each k , $0 \leq k \leq n$. Moreover, by (fs-3) we obtain that $b_n \in \tau^B$. Thus, the sequence b_0, b_1, \dots, b_n is a successful run of the automaton \mathcal{B} on the word u which simulates the original run a_0, a_1, \dots, a_n of \mathcal{A} on u . In contrast to forward simulations, where we build the sequence b_0, b_1, \dots, b_n moving forward, starting with b_0 and ending with b_n , in the case of backward simulations we build this sequence moving backward, starting with b_n and ending with b_0 . In a similar way we can understand forward and backward simulations between arbitrary fuzzy automata, taking into account degrees of possibility of transitions and degrees of relationship.

In numerous papers dealing with simulations and bisimulations mostly forward simulations and forward bisimulations have been studied. They have been usually called just simulations and bisimulations, or *strong simulations* and *strong bisimulations* (cf. [53,54,70]), and the greatest bisimulation equivalence has been usually called a *bisimilarity*. Distinction between forward and backward simulations, and forward and backward bisimulations, has been made, for instance, in [10,30,51] (for various kinds of automata), but less or more these concepts differ from the concepts having the same name which are considered here. More similar to our concepts of forward and backward simulations and bisimulations are those studied in [9], and in [31,32] (for tree automata). Moreover, backward-forward bisimulations have been discussed in the context of weighted automata in [2–4,8,10,25,26].

Given fuzzy automata $\mathcal{A} = (A, \delta^A, \sigma^A, \tau^A)$ and $\mathcal{B} = (B, \delta^B, \sigma^B, \tau^B)$, a fuzzy relation $\gamma \in \mathcal{R}(A, B)$ is a backward simulation between fuzzy automata \mathcal{A} and \mathcal{B} if and only if it is a forward simulation between the reverse fuzzy automata $\bar{\mathcal{A}}$ and $\bar{\mathcal{B}}$. Therefore, forward and backward simulations, forward and backward bisimulations, and backward–forward and forward–backward bisimulations, are mutually dual concepts, what means that for any statement on some of these concepts which is universally valid (valid for all fuzzy automata) there is the corresponding universally valid statement on its dual concept.

Now we introduce several other notions and notation that will be used in the rest of the paper.

For non-empty sets A and B and fuzzy subsets $\eta \in \mathcal{F}(A)$ and $\xi \in \mathcal{F}(B)$, fuzzy relations $\eta \rightarrow \xi \in \mathcal{R}(A, B)$ and $\eta \leftarrow \xi \in \mathcal{R}(A, B)$ are defined as follows:

$$(\eta \rightarrow \xi)(a, b) = (\eta(a) \rightarrow \xi(b)), \quad (25)$$

$$(\eta \leftarrow \xi)(a, b) = (\xi(b) \rightarrow \eta(a)), \quad (26)$$

for arbitrary $a \in A$ and $b \in B$. Let us note that $\eta \leftarrow \xi = (\xi \rightarrow \eta)^{-1}$.

We have the following.

Lemma 4.1. *Let A and B be non-empty sets and let $\eta \in \mathcal{F}(A)$ and $\xi \in \mathcal{F}(B)$.*

- (a) *The set of all solutions to the inequality $\eta \circ \chi \leq \xi$, where χ is an unknown fuzzy relation between A and B , is the principal ideal of $\mathcal{R}(A, B)$ generated by the fuzzy relation $\eta \rightarrow \xi$.*
- (b) *The set of all solutions to the inequality $\chi \circ \xi \leq \eta$, where χ is an unknown fuzzy relation between A and B , is the principal ideal of $\mathcal{R}(A, B)$ generated by the fuzzy relation $\eta \leftarrow \xi$.*

Proof. These are the well-known results by Sanchez (cf. [73–75]). \square

Note that $(\eta \rightarrow \xi) \wedge (\eta \leftarrow \xi) = \eta \leftrightarrow \xi$, where $\eta \leftrightarrow \xi$ is a fuzzy relation between A and B defined by

$$(\eta \leftrightarrow \xi)(a, b) = (\eta(a) \leftrightarrow \xi(b)), \quad (27)$$

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

Next, let A and B be non-empty sets and let $\alpha \in \mathcal{R}(A)$, $\beta \in \mathcal{R}(B)$ and $\gamma \in \mathcal{R}(A, B)$. The *right residual* of γ by α is a fuzzy relation $\alpha \backslash \gamma \in \mathcal{R}(A, B)$ defined by

$$(\alpha \backslash \gamma)(a, b) = \bigwedge_{a' \in A} (\alpha(a', a) \rightarrow \gamma(a', b)), \quad (28)$$

for all $a \in A$ and $b \in B$, and the *left residual* of γ by β is a fuzzy relation $\gamma / \beta \in \mathcal{R}(A, B)$ defined by

$$(\gamma / \beta)(a, b) = \bigwedge_{b' \in B} (\beta(b, b') \rightarrow \gamma(a, b')), \quad (29)$$

for all $a \in A$ and $b \in B$. We think of the right residual $\alpha \backslash \gamma$ as what remains of γ on the right after “dividing” γ on the left by α , and of the left residual γ / β as what remains of γ on the left after “dividing” γ on the right by β . In other words,

$$\alpha \circ \gamma' \leq \gamma \Leftrightarrow \gamma' \leq \alpha \backslash \gamma, \quad \gamma' \circ \beta \leq \gamma \Leftrightarrow \gamma' \leq \gamma / \beta, \quad (30)$$

for all $\alpha \in \mathcal{R}(A)$, $\beta \in \mathcal{R}(B)$ and $\gamma', \gamma \in \mathcal{R}(A, B)$. In the case when $A = B$, these two concepts become the well-known concepts of right and left residuals of fuzzy relations on a set (cf. [38]).

We also have the following.

Lemma 4.2. *Let A and B be non-empty sets and let $\alpha \in \mathcal{R}(A)$, $\beta \in \mathcal{R}(B)$ and $\gamma \in \mathcal{R}(A, B)$.*

- (a) *The set of all solutions to the inequality $\alpha \circ \chi \leq \gamma$, where χ is an unknown fuzzy relation between A and B , is the principal ideal of $\mathcal{R}(A, B)$ generated by the right residual $\alpha \backslash \gamma$ of γ by α .*
- (b) *The set of all solutions to the inequality $\chi \circ \beta \leq \gamma$, where χ is an unknown fuzzy relation between A and B , is the principal ideal of $\mathcal{R}(A, B)$ generated by the left residual γ / β of γ by β .*

Proof. These are also results by Sanchez (cf. [73–75]). \square

As we said in the Introduction, the problem of deciding whether there is a simulation or bisimulation of a given type between fuzzy automata, and the problem of computing the greatest simulation/bisimulation of this type, will be reduced to the problem of computing the greatest post-fixed point, contained in a given fuzzy relation, of an appropriate isotone function on the lattice of fuzzy relations. For this purpose we define the following fuzzy relations and functions on the lattice of fuzzy relations.

Let $\mathcal{A} = (A, \delta^A, \sigma^A, \tau^A)$ and $\mathcal{B} = (B, \delta^B, \sigma^B, \tau^B)$ be fuzzy automata. We define fuzzy relations $\pi^w \in \mathcal{R}(A, B)$, for $w \in \{fs, bs, fb, bb, fbb, bfb\}$, in the following way:

$$\pi^{fs} = \tau^A \rightarrow \tau^B, \quad (31)$$

$$\pi^{bs} = \sigma^A \rightarrow \sigma^B, \quad (32)$$

$$\pi^{fb} = (\tau^A \rightarrow \tau^B) \wedge (\tau^A \leftarrow \tau^B) = \tau^A \leftrightarrow \tau^B, \quad (33)$$

$$\pi^{bb} = (\sigma^A \rightarrow \sigma^B) \wedge (\sigma^A \leftarrow \sigma^B) = \sigma^A \leftrightarrow \sigma^B, \quad (34)$$

$$\pi^{fbb} = (\tau^A \rightarrow \tau^B) \wedge (\sigma^A \leftarrow \sigma^B), \quad (35)$$

$$\pi^{bfb} = (\sigma^A \rightarrow \sigma^B) \wedge (\tau^A \leftarrow \tau^B). \quad (36)$$

Moreover, we define functions $\phi^w : \mathcal{R}(A, B) \rightarrow \mathcal{R}(A, B)$, for $w \in \{fs, bs, fb, bb, fbb, bfb\}$, as follows:

$$\phi^{fs}(\gamma) = \bigwedge_{x \in X} [(\delta_x^B \circ \gamma^{-1}) / \delta_x^A]^{-1}, \quad (37)$$

$$\phi^{bs}(\gamma) = \bigwedge_{x \in X} \delta_x^A \setminus (\gamma \circ \delta_x^B), \quad (38)$$

$$\phi^{fb}(\gamma) = \bigwedge_{x \in X} [(\delta_x^B \circ \gamma^{-1}) / \delta_x^A]^{-1} \wedge [(\delta_x^A \circ \gamma) / \delta_x^B] = \phi^{fs}(\gamma) \wedge [\phi^{fs}(\gamma^{-1})]^{-1}, \quad (39)$$

$$\phi^{bb}(\gamma) = \bigwedge_{x \in X} [\delta_x^A \setminus (\gamma \circ \delta_x^B)] \wedge [\delta_x^B \setminus (\gamma^{-1} \circ \delta_x^A)]^{-1} = \phi^{bs}(\gamma) \wedge [\phi^{bs}(\gamma)]^{-1}, \quad (40)$$

$$\phi^{fbb}(\gamma) = \bigwedge_{x \in X} [(\delta_x^B \circ \gamma^{-1}) / \delta_x^A]^{-1} \wedge [\delta_x^B \setminus (\gamma^{-1} \circ \delta_x^A)]^{-1} = \phi^{fs}(\gamma) \wedge [\phi^{bs}(\gamma^{-1})]^{-1}, \quad (41)$$

$$\phi^{bfb}(\gamma) = \bigwedge_{x \in X} [\delta_x^A \setminus (\gamma \circ \delta_x^B)] \wedge [(\delta_x^A \circ \gamma) / \delta_x^B] = \phi^{bs}(\gamma) \wedge [\phi^{fs}(\gamma^{-1})]^{-1}, \quad (42)$$

for any $\gamma \in \mathcal{R}(A, B)$. Notice that in the expression “ $\phi^w(\alpha^{-1})$ ” ($w \in \{fs, bs\}$) we denote by ϕ^w a function from $\mathcal{R}(B, A)$ into itself.

The next theorem provides equivalent forms of the second and third conditions in the definitions of simulations and bisimulations.

Theorem 4.3. Let $\mathcal{A} = (A, \delta^A, \sigma^A, \tau^A)$ and $\mathcal{B} = (B, \delta^B, \sigma^B, \tau^B)$ be fuzzy automata and $w \in \{fs, bs, fb, bb, fbb, bfb\}$. A fuzzy relation $\gamma \in \mathcal{R}(A, B)$ satisfies conditions (w-2) and (w-3) if and only if it satisfies

$$\gamma \leq \phi^w(\gamma), \quad \gamma \leq \pi^w. \quad (43)$$

Proof. We will prove only the case $w = fs$. The assertion concerning the case $w = bs$ follows by the duality, and according to Eqs. (33)–(36) and (39)–(42), all other assertions can be obtained by the first two.

Consider an arbitrary $\gamma \in \mathcal{R}(A, B)$. According to Lemma 4.1(b), γ satisfies condition (fs-3) if and only if $\gamma^{-1} \leq \tau^B \leftarrow \tau^A = (\tau^A \rightarrow \tau^B)^{-1}$, which is equivalent to $\gamma \leq \tau^A \rightarrow \tau^B = \pi^{fs}$. Therefore, γ satisfies (fs-3) if and only if $\gamma \leq \pi^{fs}$.

On the other hand, γ satisfies (fs-2) if and only if

$$\gamma^{-1}(b, a) \otimes \delta_x^A(a, a') \leq (\delta_x^B \circ \gamma^{-1})(b, a'),$$

for all $a, a' \in A, b \in B$ and $x \in X$. According to the adjunction property, this is equivalent to

$$\gamma^{-1}(b, a) \leq \bigwedge_{a' \in A} [\delta_x^A(a, a') \rightarrow (\delta_x^B \circ \gamma^{-1}(b, a'))] = ((\delta_x^B \circ \gamma^{-1})/\delta_x^A)(b, a),$$

for all $a \in A, b \in B$ and $x \in X$, which is further equivalent to

$$\gamma(a, b) \leq \bigwedge_{x \in X} [(\delta_x^B \circ \gamma^{-1})/\delta_x^A]^{-1}(a, b) = (\phi^{fs}(\gamma))(a, b),$$

for all $a \in A$ and $b \in B$. Therefore, γ satisfies (fs-2) if and only if $\gamma \leq \phi^{fs}(\gamma)$.

Now we conclude that a fuzzy relation $\gamma \in \mathcal{R}(A, B)$ satisfies (fs-2) and (fs-3) if and only if it satisfies (43) (for $w = fs$), which was to be proved. \square

5. Computation of the greatest simulations and bisimulations

We will provide a method for computing the greatest simulations and bisimulations between fuzzy automata adapting the method developed in [38] for computing the greatest post-fixed points of an isotone function on the lattice of fuzzy relations on a single set.

Let A and B be non-empty sets and let $\phi : \mathcal{R}(A, B) \rightarrow \mathcal{R}(A, B)$ be an isotone function, i.e., let $\alpha \leq \beta$ implies $\phi(\alpha) \leq \phi(\beta)$, for all $\alpha, \beta \in \mathcal{R}(A, B)$. A fuzzy relation $\alpha \in \mathcal{R}(A, B)$ is called a *post-fixed point* of ϕ if $\alpha \leq \phi(\alpha)$. The well-known Knaster–Tarski fixed point theorem (stated and proved in a more general context, for complete lattices) asserts that the set of all post-fixed points of ϕ form a complete lattice (cf. [71]). Moreover, for any fuzzy relation $\pi \in \mathcal{R}(A, B)$ we have that the set of all post-fixed points of ϕ contained in π is also a complete lattice. According to Theorem 4.3, our main task is to find an efficient procedure for computing the greatest post-fixed point of the function ϕ^w contained in the fuzzy relation π^w , for each $w \in \{fs, bs, fb, bb, fbf, bfb\}$.

It should be noted that the set of all post-fixed points of an isotone function on a complete lattice is always non-empty, because it contains the least element of this complete lattice. However, this set may consist only of that single element. In our case, when we are working with a lattice of fuzzy relations, the empty relation may be the only post-fixed point, whereas we have defined simulations and bisimulations to be non-empty fuzzy relations. This requirement is necessary because the empty relation cannot satisfy the condition (w-1), unless the fuzzy set of initial states or the fuzzy set of terminal states is also empty. Therefore, our task is actually to find an efficient procedure for deciding whether there is a non-empty post-fixed point of ϕ^w contained in π^w , and if it exists, then find the greatest one.

Let $\phi : \mathcal{R}(A, B) \rightarrow \mathcal{R}(A, B)$ be an isotone function and $\pi \in \mathcal{R}(A, B)$. We define a sequence $\{\gamma_k\}_{k \in \mathbb{N}}$ of fuzzy relations from $\mathcal{R}(A, B)$ by

$$\gamma_1 = \pi, \quad \gamma_{k+1} = \gamma_k \wedge \phi(\gamma_k) \text{ for each } k \in \mathbb{N}. \quad (44)$$

The sequence $\{\gamma_k\}_{k \in \mathbb{N}}$ is obviously descending. If we denote by $\hat{\gamma}$ the greatest post-fixed point of ϕ contained in π , we can easily verify that

$$\hat{\gamma} \leq \bigwedge_{k \in \mathbb{N}} \gamma_k. \quad (45)$$

Now two very important questions arise. First, under what conditions the equality holds in (45)? Even more important question is: Under what conditions the sequence $\{\gamma_k\}_{k \in \mathbb{N}}$ is finite? If this sequence is finite, then it is not hard to show that there exists $k \in \mathbb{N}$ such that $\gamma_k = \gamma_m$, for every $m \geq k$, i.e., there exists $k \in \mathbb{N}$ such that the sequence stabilizes on γ_k . We can recognize that the sequence has stabilized when we find the smallest $k \in \mathbb{N}$ such that $\gamma_k = \gamma_{k+1}$. In this case $\hat{\gamma} = \gamma_k$, and we have an algorithm which computes $\hat{\gamma}$ in a finite number of steps.

Some conditions under which equality holds in (45) or the sequence is finite were found in [38], for fuzzy relations on a single set. It is easy to show that the same results are also valid for fuzzy relations between two sets. In the sequel we present these results.

First we note that a sequence $\{\gamma_k\}_{k \in \mathbb{N}}$ of fuzzy relations from $\mathcal{R}(A, B)$ is finite if and only if it is *image-finite*, which means that the set $\bigcup_{k \in \mathbb{N}} \text{Im}(\gamma_k)$ is finite. Next, the function $\phi : \mathcal{R}(A, B) \rightarrow \mathcal{R}(A, B)$ is called *image-localized* if there

exists a finite $K \subseteq L$ such that for each fuzzy relation $\gamma \in \mathcal{R}(A, B)$ we have

$$\text{Im}(\phi(\gamma)) \subseteq \langle K \cup \text{Im}(\gamma) \rangle, \quad (46)$$

where $\langle K \cup \text{Im}(\gamma) \rangle$ denotes the subalgebra of \mathcal{L} generated by the set $K \cup \text{Im}(\gamma)$. Such K will be called a *localization set* of the function ϕ .

Theorem analogous to the following theorem was proved in [38] for fuzzy relations on a single set, but its proof can be easily transformed into the proof of the corresponding theorem concerning fuzzy relations between two sets.

Theorem 5.1. *Let the function ϕ be image-localized, let K be its localization set, let $\pi \in \mathcal{R}(A, B)$, and let $\{\gamma_k\}_{k \in \mathbb{N}}$ be the sequence of fuzzy relations in $\mathcal{R}(A, B)$ defined by (44). Then*

$$\bigcup_{k \in \mathbb{N}} \text{Im}(\gamma_k) \subseteq \langle K \cup \text{Im}(\pi) \rangle. \quad (47)$$

If, moreover, $\langle K \cup \text{Im}(\pi) \rangle$ is a finite subalgebra of \mathcal{L} , then the sequence $\{\gamma_k\}_{k \in \mathbb{N}}$ is finite.

Going back now to the functions ϕ^w , for $w \in \{fs, bs, fb, bb, fbb, bfb\}$, we prove the following.

Theorem 5.2. *Let $\mathcal{A} = (A, \delta^A, \sigma^A, \tau^A)$ and $\mathcal{B} = (B, \delta^B, \sigma^B, \tau^B)$ be arbitrary automata.*

For any $w \in \{fs, bs, fb, bb, fbb, bfb\}$ the function ϕ^w is isotone and image-localized.

Proof. We will prove only the case $w = fs$. The assertion concerning the case $w = bs$ follows by the duality, and according to Eqs. (33)–(36) and (39)–(42), all other assertions can be derived from the first two.

Let $\gamma_1, \gamma_2 \in \mathcal{R}(A, B)$ be fuzzy relations such that $\gamma_1 \leq \gamma_2$, and consider the following systems of fuzzy relation inequalities:

$$\chi^{-1} \circ \delta_x^A \leq \delta_x^B \circ \gamma_1^{-1}, \quad x \in X, \quad (48)$$

$$\chi^{-1} \circ \delta_x^A \leq \delta_x^B \circ \gamma_2^{-1}, \quad x \in X, \quad (49)$$

where $\chi \in \mathcal{R}(A, B)$ is an unknown fuzzy relation. Using Lemma 4.2(b) and the definition of an inverse relation, it can be easily shown that the set of all solutions to system (48) (resp. (49)) form a principal ideal of $\mathcal{R}(A, B)$ generated by $\phi^{fs}(\gamma_1)$ (resp. $\phi^{fs}(\gamma_2)$). Since for each $x \in X$ we have that $\delta_x^B \circ \gamma_1^{-1} \leq \delta_x^B \circ \gamma_2^{-1}$, we conclude that every solution to (48) is a solution to (49). Consequently, $\phi^{fs}(\gamma_1)$ is a solution to (49), so $\phi^{fs}(\gamma_1) \leq \phi^{fs}(\gamma_2)$. Therefore, we have proved that ϕ^{fs} is an isotone function.

Next, let $K = \bigcup_{x \in X} (\text{Im}(\delta_x^A) \cup \text{Im}(\delta_x^B))$ and let $\gamma \in \mathcal{R}(A, B)$ be an arbitrary fuzzy relation. It is evident that $\text{Im}(\phi^{fs}(\gamma)) \subseteq \langle K \cup \text{Im}(\gamma) \rangle$, and since the input alphabet X is finite, then K is also finite. This confirms that the function ϕ^{fs} is image-localized. \square

Now we are ready to prove the main result of this paper, which provides algorithms for deciding whether there is a simulation or bisimulation of a given type between fuzzy automata, and for computing the greatest simulations and bisimulations, when they exist.

Theorem 5.3. *Let $\mathcal{A} = (A, \delta^A, \sigma^A, \tau^A)$ and $\mathcal{B} = (B, \delta^B, \sigma^B, \tau^B)$ be fuzzy automata, let $w \in \{fs, bs, fb, bb, fbb, bfb\}$, and let a sequence $\{\gamma_k\}_{k \in \mathbb{N}}$ of fuzzy relations from $\mathcal{R}(A, B)$ be defined by*

$$\gamma_1 = \pi^w, \quad \gamma_{k+1} = \gamma_k \wedge \phi^w(\gamma_k) \text{ for each } k \in \mathbb{N}. \quad (50)$$

If $\langle \text{Im}(\pi^w) \cup \bigcup_{x \in X} (\text{Im}(\delta_x^A) \cup \text{Im}(\delta_x^B)) \rangle$ is a finite subalgebra of \mathcal{L} , then the following is true:

- (a) *the sequence $\{\gamma_k\}_{k \in \mathbb{N}}$ is finite and descending, and there is the least natural number k such that $\gamma_k = \gamma_{k+1}$;*
- (b) *γ_k is the greatest fuzzy relation in $\mathcal{R}(A, B)$ which satisfies (w-2) and (w-3);*
- (c) *if γ_k satisfies (w-1), then it is the greatest fuzzy relation in $\mathcal{R}(A, B)$ which satisfies (w-1), (w-2) and (w-3);*
- (d) *if γ_k does not satisfy (w-1), then there is not any fuzzy relation in $\mathcal{R}(A, B)$ which satisfies (w-1), (w-2) and (w-3).*

Proof. We will prove only the case $w = fs$. All other cases can be proved in a similar manner.

So, let $\langle \text{Im}(\pi^{fs}) \cup \bigcup_{x \in X} (\text{Im}(\delta_x^A) \cup \text{Im}(\delta_x^B)) \rangle$ be a finite subalgebra of \mathcal{L} .

(a) According to Theorems 5.2 and 5.1, the sequence $\{\gamma_k\}_{k \in \mathbb{N}}$ is finite and descending, so there are $k, m \in \mathbb{N}$ such that $\gamma_k = \gamma_{k+m}$, whence $\gamma_{k+1} \leq \gamma_k = \gamma_{k+m} \leq \gamma_{k+1}$. Thus, there is $k \in \mathbb{N}$ such that $\gamma_k = \gamma_{k+1}$, and consequently, there is the least natural number having this property.

(b) By $\gamma_k = \gamma_{k+1} = \gamma_k \wedge \phi^{fs}(\gamma_k)$ we obtain that $\gamma_k \leq \phi^{fs}(\gamma_k)$, and also, $\gamma_k \leq \gamma_1 = \pi^{fs}$. Therefore, by Theorem 4.3 it follows that γ_k satisfies (fs-2) and (fs-3).

Let $\alpha \in \mathcal{R}(A, B)$ be an arbitrary fuzzy relation which satisfies (fs-2) and (fs-3). As we have already noted, α satisfies (fs-3) if and only if $\alpha \leq \pi^{fs} = \gamma_1$. Next, suppose that $\alpha \leq \gamma_n$, for some $n \in \mathbb{N}$. Then for every $x \in X$ we have that $\alpha^{-1} \circ \delta_x^A \leq \delta_x^B \circ \alpha^{-1} \leq \delta_x^B \circ \gamma_n^{-1}$, and according to Lemma 4.2(b), $\alpha^{-1} \leq (\delta_x^B \circ \gamma_n^{-1}) / \delta_x^A$, i.e., $\alpha \leq [(\delta_x^B \circ \gamma_n^{-1}) / \delta_x^A]^{-1} = \phi^{fs}(\gamma_n)$. Therefore, $\alpha \leq \gamma_n \wedge \phi^{fs}(\gamma_n) = \gamma_{n+1}$. Now, by induction we obtain that $\alpha \leq \gamma_n$, for every $n \in \mathbb{N}$, and hence, $\alpha \leq \gamma_k$. This means that γ_k is the greatest fuzzy relation in $\mathcal{R}(A, B)$ satisfying (fs-2) and (fs-3).

(c) This follows immediately from (b).

(d) Suppose that γ_k does not satisfy (fs-1). Let $\gamma \in \mathcal{R}(A, B)$ be an arbitrary fuzzy relation which satisfies (fs-1), (fs-2) and (fs-3). According to (b) of this theorem, $\gamma \leq \gamma_k$, so we have that $\sigma^A \leq \sigma^B \circ \gamma^{-1} \leq \sigma^B \circ \gamma_k^{-1}$. But, this contradicts our starting assumption that γ_k does not satisfy (fs-1). Hence, we conclude that there is not any fuzzy relation in $\mathcal{R}(A, B)$ which satisfies (fs-1), (fs-2) and (fs-3). \square

Algorithm 5.4 (Construction of the greatest simulation/bisimulation). The input of this algorithm are fuzzy finite automata $\mathcal{A} = (A, \delta^A, \sigma^A, \tau^A)$ and $\mathcal{B} = (B, \delta^B, \sigma^B, \tau^B)$. The algorithm decides whether there is a simulation or bisimulation between \mathcal{A} and \mathcal{B} of a given type $w \in \{fs, bs, fb, bb, fbb, bfb\}$, and when it exists, the output of the algorithm is the greatest simulation/bisimulation of the type w .

The procedure is to construct the sequence of fuzzy relations $\{\gamma_k\}_{k \in \mathbb{N}}$, in the following way:

- (A1) In the first step we compute π^w and we set $\gamma_1 = \pi^w$.
- (A2) After the k th step let a fuzzy relation γ_k have been constructed.
- (A3) In the next step we construct the fuzzy relation γ_{k+1} by means of the formula $\gamma_{k+1} = \gamma_k \wedge \phi^w(\gamma_k)$.
- (A4) Simultaneously, we check whether $\gamma_{k+1} = \gamma_k$.
- (A5) The first time we find the smallest number k such that $\gamma_{k+1} = \gamma_k$, the procedure of constructing the sequence $\{\gamma_k\}_{k \in \mathbb{N}}$ terminates, and we check whether γ_k satisfies (w-1).
If γ_k satisfies (w-1), then it is the greatest simulation/bisimulation between \mathcal{A} and \mathcal{B} of type w , and if γ_k does not satisfy (w-1), then there is not any simulation/bisimulation between \mathcal{A} and \mathcal{B} of type w .

If the underlying structure of membership values \mathcal{L} is locally finite, then the algorithm terminates in a finite number of steps, for any fuzzy finite automata over \mathcal{L} . On the other hand, if \mathcal{L} is not locally finite, then the algorithm terminates in a finite number of steps under conditions determined by Theorems 5.1 and 5.3.

Note that the claim of Theorem 5.3 referring to forward bisimulations, i.e., the related part of Algorithm 5.4, is a straightforward generalization of a well-known result concerning forward bisimulations between ordinary nondeterministic automata, given in Kozen's book [44, p. 106]. Theorem 5.3 also generalizes recent results given in [17], which refer to all four types of bisimulations between nondeterministic automata.

Next, we will consider the case when $\mathcal{L} = (L, \wedge, \vee, \otimes, \rightarrow, 0, 1)$ is a complete residuated lattice satisfying the following conditions:

$$x \vee \left(\bigwedge_{i \in I} y_i \right) = \bigwedge_{i \in I} (x \vee y_i), \quad (51)$$

$$x \otimes \left(\bigwedge_{i \in I} y_i \right) = \bigwedge_{i \in I} (x \otimes y_i), \quad (52)$$

for all $x \in L$ and $\{y_i\}_{i \in I} \subseteq L$. Let us note that if $\mathcal{L} = ([0, 1], \wedge, \vee, \otimes, \rightarrow, 0, 1)$, where $[0, 1]$ is the real unit interval and \otimes is a left-continuous t-norm on $[0, 1]$, then (51) follows immediately by linearity of \mathcal{L} , and \mathcal{L} satisfies (52) if and only if \otimes is a continuous t-norm, i.e., if and only if \mathcal{L} is a BL-algebra (cf. [6,7]). Therefore, conditions (51) and

(52) hold for every BL -algebra on the real unit interval. In particular, the Łukasiewicz, Goguen (product) and Gödel structures fulfill (51) and (52).

Under these conditions we have the following.

Theorem 5.5. Let $\mathcal{A} = (A, \delta^A, \sigma^A, \tau^A)$ and $\mathcal{B} = (B, \delta^B, \sigma^B, \tau^B)$ be fuzzy automata, let $w \in \{fs, bs, fb, bb, fbb, bfb\}$, let $\{\gamma_k\}_{k \in \mathbb{N}}$ be the sequence of fuzzy relations from $\mathcal{R}(A, B)$ defined by (50), and let

$$\gamma = \bigwedge_{k \in \mathbb{N}} \gamma_k. \quad (53)$$

If \mathcal{L} is a complete residuated lattice satisfying (51) and (52), then the following is true:

- (a) γ is the greatest fuzzy relation in $\mathcal{R}(A, B)$ which satisfies (w-2) and (w-3);
- (b) if γ satisfies (w-1), then it is the greatest fuzzy relation in $\mathcal{R}(A, B)$ which satisfies (w-1), (w-2) and (w-3);
- (c) if γ does not satisfy (w-1), then there is not any fuzzy relation in $\mathcal{R}(A, B)$ which satisfies (w-1), (w-2) and (w-3).

Proof. We will prove only the case $w = fs$. All other cases can be proved similarly.

So, let \mathcal{L} be a complete residuated lattice satisfying (51) and (52). First, notice that it has been shown in [20] that if (51) holds, then for all descending sequences $\{x_k\}_{k \in \mathbb{N}}$, $\{y_k\}_{k \in \mathbb{N}} \subseteq L$ we have

$$\bigwedge_{k \in \mathbb{N}} (x_k \vee y_k) = \left(\bigwedge_{k \in \mathbb{N}} x_k \right) \vee \left(\bigwedge_{k \in \mathbb{N}} y_k \right). \quad (54)$$

- (a) For arbitrary $x \in X$, $a \in A$ and $b \in B$ we have that

$$\begin{aligned} \left(\bigwedge_{k \in \mathbb{N}} (\delta_x^B \circ \gamma_k^{-1}) \right) (b, a) &= \bigwedge_{k \in \mathbb{N}} (\delta_x^B \circ \gamma_k^{-1})(b, a) = \bigwedge_{k \in \mathbb{N}} \left(\bigvee_{b' \in B} \delta_x^B(b, b') \otimes \gamma_k^{-1}(b', a) \right) \\ &= \bigvee_{b' \in B} \left(\bigwedge_{k \in \mathbb{N}} \delta_x^B(b, b') \otimes \gamma_k^{-1}(b', a) \right) \quad \text{by (54)} \\ &= \bigvee_{b' \in B} \left(\delta_x^B(b, b') \otimes \left(\bigwedge_{k \in \mathbb{N}} \gamma_k^{-1}(b', a) \right) \right) \quad \text{(by (52))} \\ &= \bigvee_{b' \in B} \left(\delta_x^B(b, b') \otimes \gamma^{-1}(b', a) \right) = (\delta_x^B \circ \gamma^{-1})(b, a), \end{aligned}$$

which means that

$$\bigwedge_{k \in \mathbb{N}} \delta_x^B \circ \gamma_k^{-1} = \delta_x^B \circ \gamma^{-1},$$

for every $x \in X$. The use of condition (54) is justified by the facts that B is finite, and that $\{\gamma_k^{-1}(b', a)\}_{k \in \mathbb{N}}$ is a descending sequence, so $\{\delta_x^B(b, b') \otimes \gamma_k^{-1}(b', a)\}_{k \in \mathbb{N}}$ is also a descending sequence.

Now, for all $x \in X$ and $k \in \mathbb{N}$ we have that

$$\gamma \leq \gamma_{k+1} \leq \phi^{fs}(\gamma_k) = [(\delta_x^B \circ \gamma_k^{-1}) / \delta_x^A]^{-1},$$

which is equivalent to

$$\gamma^{-1} \circ \delta_x^A \leq \delta_x^B \circ \gamma_k^{-1}.$$

As the last inequality holds for every $k \in \mathbb{N}$, we have that

$$\gamma^{-1} \circ \delta_x^A \leq \bigwedge_{k \in \mathbb{N}} \delta_x^B \circ \gamma_k^{-1} = \delta_x^B \circ \gamma^{-1},$$

for every $x \in X$. Therefore, γ satisfies (fs-2). Moreover, $\gamma \leq \gamma_1 = \pi^{fs}$, so γ also satisfies (fs-3).

Next, let $\alpha \in \mathcal{R}(A, B)$ be an arbitrary fuzzy relation satisfying (fs-2) and (fs-3). According to Theorem 4.3, $\alpha \leq \phi^{fs}(\alpha)$ and $\alpha \leq \pi^{fs} = \gamma_1$. By induction we can easily prove that $\alpha \leq \gamma_k$ for every $k \in \mathbb{N}$, and therefore, $\alpha \leq \gamma$. This means that γ is the greatest fuzzy relation in $\mathcal{R}(A, B)$ which satisfies (fs-2) and (fs-3).

The assertion (b) follows immediately from (a), whereas the assertion (c) can be proved in the same way as the assertion (d) of Theorem 5.3. \square

In some situations we do not need simulations and bisimulations that are fuzzy relations, but those that are ordinary crisp relations. Moreover, in cases where our algorithms for computing the greatest simulations and bisimulations fail to terminate in a finite number of steps, we can search for the greatest crisp simulations and bisimulations. They can be understood as a kind of “approximations” of the greatest fuzzy simulations and bisimulations. Here we show that the above given algorithms for computing the greatest fuzzy simulations and bisimulations can be modified to compute the greatest crisp simulations and bisimulations. The new algorithms terminate in a finite number of steps, independently of the properties of the underlying structure of truth values. In the next section we will give an example which shows that the greatest crisp simulations and bisimulations cannot be obtained simply by taking the crisp parts of the greatest fuzzy simulations and bisimulations. In fact, our example shows that there may be a fuzzy simulation/bisimulation of a given type between two fuzzy automata, but there is not any crisp simulation/bisimulation of this type between them.

Let A and B be non-empty finite sets, and let $\mathcal{R}^c(A, B)$ denote the set of all crisp relations from $\mathcal{R}(A, B)$. It is not hard to verify that $\mathcal{R}^c(A, B)$ is a complete sublattice of $\mathcal{R}(A, B)$, i.e., the meet and the join in $\mathcal{R}(A, B)$ of an arbitrary family of crisp relations from $\mathcal{R}^c(A, B)$ are also crisp relations (in fact, they coincide with the ordinary intersection and union of crisp relations). Moreover, for each fuzzy relation $\gamma \in \mathcal{R}(A, B)$ we have that $\gamma^c \in \mathcal{R}^c(A, B)$, where γ^c denotes the *crisp part* of a fuzzy relation γ (in some sources called the *kernel* of γ), i.e., a function $\gamma^c : A \times B \rightarrow \{0, 1\}$ defined by $\gamma^c(a, b) = 1$, if $\gamma(a, b) = 1$, and $\gamma^c(a, b) = 0$, if $\gamma(a, b) < 1$, for arbitrary $a \in A$ and $b \in B$. Equivalently, γ^c is considered as an ordinary crisp relation between A and B given by $\gamma^c = \{(a, b) \in A \times B \mid \gamma(a, b) = 1\}$.

For each function $\phi : \mathcal{R}(A, B) \rightarrow \mathcal{R}(A, B)$ we define a function $\phi^c : \mathcal{R}^c(A, B) \rightarrow \mathcal{R}^c(A, B)$ by

$$\phi^c(\gamma) = (\phi(\gamma))^c \quad \text{for any } \gamma \in \mathcal{R}^c(A, B).$$

If ϕ is isotone, then it can be easily shown that ϕ^c is also an isotone function.

We have that the following is true.

Proposition 5.6. *Let A and B be non-empty finite sets, let $\phi : \mathcal{R}(A, B) \rightarrow \mathcal{R}(A, B)$ be an isotone function and let $\pi \in \mathcal{R}(A, B)$ be a given fuzzy relation. A crisp relation $\varrho \in \mathcal{R}^c(A, B)$ is the greatest crisp solution in $\mathcal{R}(A, B)$ to the system*

$$\chi \leq \phi(\chi), \quad \chi \leq \pi, \tag{55}$$

if and only if it is the greatest solution in $\mathcal{R}^c(A, B)$ to the system

$$\xi \leq \phi^c(\xi), \quad \xi \leq \pi^c, \tag{56}$$

where χ is an unknown fuzzy relation and ξ is an unknown crisp relation.

Furthermore, a sequence $\{\varrho_k\}_{k \in \mathbb{N}} \subseteq \mathcal{R}(A, B)$ defined by

$$\varrho_1 = \pi^c, \quad \varrho_{k+1} = \varrho_k \wedge \phi^c(\varrho_k) \quad \text{for every } k \in \mathbb{N} \tag{57}$$

is a finite descending sequence of crisp relations, and the least member of this sequence is the greatest solution to the system (56) in $\mathcal{R}^c(A, B)$.

Proof. The proof of this proposition can be obtained simply by translating the proof of Theorem 5.8 [38] to the case of relations between the two sets. \square

Taking ϕ to be any of the functions ϕ^w , for $w \in \{fs, bs, fb, bb, fbb, bfb\}$, Proposition 5.6 gives algorithms for deciding whether there is a crisp simulation or bisimulation of a given type between fuzzy automata, and for computing the greatest crisp simulations and bisimulations, when they exist. As we have seen in Proposition 5.6, these algorithms always terminate in a finite number of steps, independently of the properties of the underlying structure of

truth values. However, as we have already mentioned, in the next section we will give an example which shows that there may be a fuzzy simulation/bisimulation of a given type between two fuzzy automata, and there is not any crisp simulation/bisimulation of this type between them.

It is worth noting that functions $(\phi^w)^c$, for all $w \in \{fs, bs, fb, bb, fbb, bfb\}$, can be characterized as follows:

$$\begin{aligned}(a, b) \in (\phi^{fs})^c(q) &\Leftrightarrow (\forall x \in X)(\forall a' \in A)\delta_x^A(a, a') \leq (\delta_x^B \circ q^{-1})(b, a'), \\(a, b) \in (\phi^{bs})^c(q) &\Leftrightarrow (\forall x \in X)(\forall a' \in A)\delta_x^A(a', a) \leq (q \circ \delta_x^B)(a', b), \\(\phi^{fb})^c(q) &= (\phi^{fs})^c(q) \wedge [(\phi^{fs})^c(q^{-1})]^{-1}, \quad (\phi^{bb})^c(q) = (\phi^{bs})^c(q) \wedge [(\phi^{bs})^c(q^{-1})]^{-1}, \\(\phi^{fbb})^c(q) &= (\phi^{fs})^c(q) \wedge [(\phi^{bs})^c(q^{-1})]^{-1}, \quad (\phi^{bfb})^c(q) = (\phi^{bs})^c(q) \wedge [(\phi^{fs})^c(q^{-1})]^{-1},\end{aligned}$$

for all $q \in \mathcal{R}^c(A, B)$, $a \in A$ and $b \in B$.

6. Computational examples

In this section we give examples which demonstrate the application of our algorithms and clarify relationships between different types of simulations and bisimulations.

The first example demonstrates the case when there are all types of simulations and bisimulations, but there is not any crisp bisimulation between two given automata.

Example 6.1. Let $\mathcal{A} = (A, \delta^A, \sigma^A, \tau^A)$ and $\mathcal{B} = (B, \delta^B, \sigma^B, \tau^B)$ be automata over the Gödel structure and an alphabet $X = \{x, y\}$, with $|A| = 3$, $|B| = 2$, and fuzzy transition relations and fuzzy sets of initial and terminal states which are represented by the following fuzzy matrices and vectors:

$$\sigma^A = [1 \ 1 \ 1], \quad \delta_x^A = \begin{bmatrix} 1 & 0.3 & 0.4 \\ 0.5 & 1 & 0.3 \\ 0.4 & 0.6 & 0.7 \end{bmatrix}, \quad \delta_y^A = \begin{bmatrix} 0.5 & 0.6 & 0.2 \\ 0.6 & 0.3 & 0.4 \\ 0.7 & 0.7 & 1 \end{bmatrix}, \quad \tau^A = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad (58)$$

$$\sigma^B = [1 \ 1], \quad \delta_x^B = \begin{bmatrix} 1 & 0.6 \\ 0.6 & 0.7 \end{bmatrix}, \quad \delta_y^B = \begin{bmatrix} 0.6 & 0.6 \\ 0.7 & 1 \end{bmatrix}, \quad \tau^B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \quad (59)$$

Using algorithms based on Theorem 5.3 we obtain that there are all types of simulations and bisimulations between fuzzy automata \mathcal{A} and \mathcal{B} , and the greatest simulations and bisimulations are given by matrices

$$\begin{aligned}\gamma^{fs} &= \begin{bmatrix} 1 & 0.7 \\ 1 & 0.7 \\ 0.6 & 1 \end{bmatrix}, \quad \gamma^{bs} = \begin{bmatrix} 1 & 0.7 \\ 1 & 0.7 \\ 0.7 & 1 \end{bmatrix}, \quad \gamma^{fb} = \begin{bmatrix} 1 & 0.6 \\ 1 & 0.6 \\ 0.6 & 1 \end{bmatrix}, \quad \gamma^{bb} = \begin{bmatrix} 1 & 0.7 \\ 1 & 0.7 \\ 0.7 & 1 \end{bmatrix}, \\ \gamma^{fbb} &= \begin{bmatrix} 1 & 0.7 \\ 1 & 0.7 \\ 0.6 & 1 \end{bmatrix}, \quad \gamma^{bfb} = \begin{bmatrix} 1 & 0.6 \\ 1 & 0.6 \\ 0.7 & 1 \end{bmatrix}.\end{aligned}$$

On the other hand, using the version of the algorithms for crisp simulations and bisimulations, we obtain that there is not any crisp bisimulation between fuzzy automata \mathcal{A} and \mathcal{B} .

The second example demonstrates the case when there is a forward bisimulation, but there is not any other type of bisimulations between two given automata.

Example 6.2. Let us change σ^A in (58) and σ^B in (59) to

$$\sigma^A = [0 \ 1 \ 0], \quad \sigma^B = [1 \ 0.5].$$

Then the greatest forward bisimulation between fuzzy automata \mathcal{A} and \mathcal{B} is given by

$$\gamma^{fb} = \begin{bmatrix} 1 & 0.6 \\ 1 & 0.6 \\ 0.6 & 1 \end{bmatrix},$$

but there are no bisimulations of any other type between \mathcal{A} and \mathcal{B} .

Due to the duality between forward and backward bisimulations, it is possible to construct automata between which there is a backward bisimulation, but there is not any other type of bisimulations.

We can also give an example which demonstrates the case when there is a backward–forward bisimulation, but there is not any other type of bisimulations between two given automata.

Example 6.3. Let us change σ^A in (58) and σ^B in (59) to

$$\sigma^A = [0 \ 0 \ 1], \quad \sigma^B = [0.7 \ 1].$$

Then the greatest backward–forward bisimulation between fuzzy automata \mathcal{A} and \mathcal{B} is given by

$$\gamma^{bfb} = \begin{bmatrix} 1 & 0.6 \\ 1 & 0.6 \\ 0.7 & 1 \end{bmatrix},$$

but there are no bisimulations of any other type between \mathcal{A} and \mathcal{B} .

Due to the duality between backward–forward and forward–backward bisimulations, it is possible to construct automata between which there is a backward bisimulation, but there is not any other type of bisimulations.

Next, we give an example which demonstrates the case when there is not any type of simulations and bisimulations between two given automata.

Example 6.4. Let us change σ^A in (58) and σ^B in (59) to

$$\sigma^A = [1 \ 0 \ 0], \quad \sigma^B = [0.5 \ 1].$$

Then there are no simulations and bisimulations of any type between \mathcal{A} and \mathcal{B} .

The following example considers the case (over the product structure) when the sequence of fuzzy relations defined by (50) is infinite, and its intersection is the greatest forward bisimulation between the given fuzzy automata.

Example 6.5. Let $\mathcal{A} = (A, \delta^A, \sigma^A, \tau^A)$ and $\mathcal{B} = (B, \delta^B, \sigma^B, \tau^B)$ be automata over the Goguen (product) structure and an alphabet $X = \{x\}$, with $|A| = 3$, $|B| = 2$, and fuzzy transition relations and fuzzy sets of initial and terminal states which are represented by the following fuzzy matrices and vectors:

$$\sigma^A = [1 \ 1 \ 1], \quad \delta_x^A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix}, \quad \tau^A = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad \sigma^B = [1 \ 1], \quad \delta_x^B = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{2} \end{bmatrix}, \quad \tau^B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Computing the sequence $\{\gamma_k\}_{k \in \mathbb{N}}$ for forward bisimulations by the formula (50) ($w = fb$) we obtain that

$$\gamma_k = \begin{bmatrix} 1 & \frac{1}{2^{k-1}} \\ 1 & \frac{1}{2^{k-1}} \\ \frac{1}{2^{k-1}} & 1 \end{bmatrix} \text{ for each } k \in \mathbb{N}, \quad \gamma = \bigwedge_{k \in \mathbb{N}} \gamma_k = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

According to Theorem 5.5, γ is the greatest forward bisimulation between fuzzy automata \mathcal{A} and \mathcal{B} .

The last example shows that the finiteness of the subalgebra $\langle \text{Im}(\pi^w) \cup \bigcup_{x \in X} (\text{Im}(\delta_x^A) \cup \text{Im}(\delta_x^B)) \rangle$ of \mathcal{L} , which appears as an assumption in Theorem 5.3, is sufficient for the finiteness of the sequence defined by (50), but it is not necessary.

Example 6.6. Let us change σ^A , σ^B , τ^A and τ^B in the previous example to

$$\sigma^A = [1 \ 1 \ 0], \quad \sigma^B = [1 \ 0], \quad \tau^A = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad \tau^B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Computing the fuzzy relations γ_k , $k \in \mathbb{N}$, using the formula (50), we obtain that

$$\gamma_1 = \gamma_2 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},$$

and it is the greatest forward bisimulation between fuzzy automata \mathcal{A} and \mathcal{B} .

On the other hand, we have that the subalgebra $\langle \text{Im}(\pi^w) \cup \text{Im}(\delta_x^A) \cup \text{Im}(\delta_x^B) \rangle$ of \mathcal{L} is not finite, since it contains $1/2^k$, for every $k \in \mathbb{N}$.

7. Concluding remarks

Simulations and bisimulations between fuzzy automata have recently been introduced and studied in [18]. However, this paper did not provide an efficient algorithm for deciding whether there is a simulation or bisimulation of a given type between two fuzzy automata, and for computing the greatest one, when it exists. Although the mentioned paper has offered a method for checking the existence of a uniform forward (resp. backward) bisimulation between two fuzzy automata, we cannot consider this method efficient because it boils down to the well-known graph isomorphism problem, whose rough computational complexity is still not known and the general opinion is that it lies between P and NP -complete if $P \neq NP$. The present paper has filled this gap by providing, for any of the six types of simulations and bisimulations, an efficient procedure for deciding whether there is a simulation/bisimulation of this type between two fuzzy automata, and for computing the greatest one, when it exists. This procedure is based on ideas developed in [20,38], where the problem of solving related systems of fuzzy relation inequalities and equations has been reduced to the problem of computing the greatest post-fixed points of particular isotone functions on lattices of fuzzy relations.

Note that fuzzy relation inequalities and equations discussed in this paper will be studied from a more general point of view in our further research.

References

- [1] L. Aceto, A. Ingolfssdottir, K.G. Larsen, J. Srba, *Reactive Systems: Modelling, Specification and Verification*, Cambridge University Press, Cambridge, 2007.
- [2] M.P. Béal, S. Lombardy, J. Sakarovitch, On the equivalence of \mathbb{Z} -automata, in: L. Caires et al. (Eds.), *ICALP 2005, Lecture Notes in Computer Science*, vol. 3580, Springer, Heidelberg, 2005, pp. 397–409.
- [3] M.P. Béal, S. Lombardy, J. Sakarovitch, Conjugacy and equivalence of weighted automata and functional transducers, in: D. Grigoriev, J. Harrison, E.A. Hirsch (Eds.), *CSR 2006, Lecture Notes in Computer Science*, vol. 3967, Springer, Heidelberg, 2006, pp. 58–69.
- [4] M.P. Béal, D. Perrin, On the generating sequences of regular languages on k symbols, *J. ACM* 50 (2003) 955–980.
- [5] R. Bělohlávek, Determinism and fuzzy automata, *Inf. Sci.* 143 (2002) 205–209.
- [6] R. Bělohlávek, *Fuzzy Relational Systems: Foundations and Principles*, Kluwer, New York, 2002.
- [7] R. Bělohlávek, V. Vychodil, *Fuzzy Equational Logic*, Springer, Berlin, Heidelberg, 2005.
- [8] S.L. Bloom, Z. Ésik, *Iteration Theories: The Equational Logic of Iterative Processes*, EATCS Monographs on Theoretical Computer Science, Springer, Berlin, Heidelberg, 1993.
- [9] T. Brihaye, Words and bisimulations of dynamical systems, *Discrete Math. Theor. Comput. Sci.* 9 (2) (2007) 11–32.
- [10] P. Buchholz, Bisimulation relations for weighted automata, *Theor. Comput. Sci.* 393 (2008) 109–123.
- [11] Y. Cao, G. Chen, E. Kerre, Bisimulations for fuzzy transition systems, *IEEE Trans. Fuzzy Syst.* 19 (2011) 540–552.
- [12] Y.Z. Cao, M.S. Ying, Supervisory control of fuzzy discrete event systems, *IEEE Trans. Syst. Man Cybern. Part B* 35 (2005) 366–371.
- [13] Y.Z. Cao, M.S. Ying, Observability and decentralized control of fuzzy discrete-event systems, *IEEE Trans. Fuzzy Syst.* 14 (2006) 202–216.

- [14] Y.Z. Cao, M.S. Ying, G.Q. Chen, State-based control of fuzzy discrete-event systems, *IEEE Trans. Syst. Man Cybern. Part B* 37 (2007) 410–424.
- [15] C.G. Cassandras, S. Lafortune, *Introduction to Discrete Event Systems*, Springer, 2008.
- [16] M. Ćirić, M. Droste, J. Ignjatović, H. Vogler, Determinization of weighted finite automata over strong bimonoids, *Inf. Sci.* 180 (2010) 3497–3520.
- [17] M. Ćirić, J. Ignjatović, M. Bašić, I. Jančić, Nondeterministic automata: simulation, bisimulation and structural equivalence, *Inf. Sci.*, submitted for publication.
- [18] M. Ćirić, J. Ignjatović, N. Damjanović, M. Bašić, Bisimulations for fuzzy automata, *Fuzzy Sets Syst.* 186 (2012) 100–139.
- [19] M. Ćirić, A. Stamenković, J. Ignjatović, T. Petković, Factorization of fuzzy automata, in: E. Csuhaj-Varju, Z. Ésik (Eds.), *FCT 2007, Lecture Notes in Computer Science*, vol. 4639, Springer, Heidelberg, 2007, pp. 213–225.
- [20] M. Ćirić, A. Stamenković, J. Ignjatović, T. Petković, Fuzzy relation equations and reduction of fuzzy automata, *J. Comput. Syst. Sci.* 76 (2010) 609–633.
- [21] M. Droste, T. Stüber, H. Vogler, Weighted finite automata over strong bimonoids, *Inf. Sci.* 180 (2010) 156–166.
- [22] D. Dubois, H. Prade, *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, New York, 1980.
- [23] D. Dubois, H. Prade (Eds.), *Fundamentals of Fuzzy Sets, The Handbooks of Fuzzy Sets Series*, vol. 1, Kluwer Academic Publishers, 2000.
- [24] M.M. Gupta, G.N. Saridis, B.R. Gaines, *Fuzzy Automata and Decision Processes*, North-Holland, New York, 1977.
- [25] Z. Ésik, W. Kuich, A generalization of Kozen’s axiomatization of the equational theory of the regular sets, in: *Words, Semigroups, and Transductions*, World Scientific, River Edge, NJ, 2001, pp. 99–114.
- [26] Z. Ésik, A. Maletti, Simulation vs. Equivalence, *CoRR* abs/1004.2426, 2010.
- [27] A. Dovier, C. Piazza, A. Policriti, An efficient algorithm for computing bisimulation equivalence, *Theor. Comput. Sci.* 311 (2004) 221–256.
- [28] R. Gentilini, C. Piazza, A. Policriti, From bisimulation to simulation: coarsest partition problems, *J. Autom. Reason.* 31 (2003) 73–103.
- [29] P. Hájek, *Mathematics of Fuzzy Logic*, Kluwer, Dordrecht, 1998.
- [30] T.A. Henzinger, P.W. Kopke, A. Puri, P. Varaiya, What’s decidable about hybrid automata?, *J. Comput. Syst. Sci.* 57 (1998) 94–124.
- [31] J. Högborg, A. Maletti, J. May, Backward and forward bisimulation minimisation of tree automata, in: J. Holub, J. Ždárek (Eds.), *IAA07, Lecture Notes in Computer Science*, vol. 4783, Springer, Heidelberg, 2007, pp. 109–121.
- [32] J. Högborg, A. Maletti, J. May, Backward and forward bisimulation minimisation of tree automata, *Theor. Comput. Sci.* 410 (2009) 3539–3552.
- [33] U. Höhle, Commutative, residuated ell-monoids, in: U. Höhle, E. Klement (Eds.), *Non-Classical Logics and Their Applications to Fuzzy Subsets*, Kluwer Academic Publishers, Boston, Dordrecht, 1995, pp. 53–106.
- [34] B. Hruz, M.C. Zhou, *Modeling and Control of Discrete-event Dynamical Systems: with Petri Nets and other Tools*, Springer, 2007.
- [35] J. Ignjatović, M. Ćirić, Formal power series and regular operations on fuzzy languages, *Inf. Sci.* 180 (2010) 1104–1120.
- [36] J. Ignjatović, M. Ćirić, S. Bogdanović, Determinization of fuzzy automata with membership values in complete residuated lattices, *Inf. Sci.* 178 (2008) 164–180.
- [37] J. Ignjatović, M. Ćirić, S. Bogdanović, Fuzzy homomorphisms of algebras, *Fuzzy Sets Syst.* 160 (2009) 2345–2365.
- [38] J. Ignjatović, M. Ćirić, S. Bogdanović, On the greatest solutions to weakly linear systems of fuzzy relation inequalities and equations, *Fuzzy Sets Syst.* 161 (2010) 3081–3113.
- [39] J. Ignjatović, M. Ćirić, S. Bogdanović, T. Petković, Myhill–Nerode type theory for fuzzy languages and automata, *Fuzzy Sets Syst.* 161 (2010) 1288–1324.
- [40] Z. Jančić, J. Ignjatović, M. Ćirić, An improved algorithm for determinization of weighted and fuzzy automata, *Inf. Sci.* 181 (2011) 1358–1368.
- [41] P.C. Kannelakis, S.A. Smolka, CCS expressions, finite state processes, and three problems of equivalence, *Inf. Comput.* 86 (1990) 43–68.
- [42] E. Kilic, Diagnosability of fuzzy discrete event systems, *Inf. Sci.* 178 (2008) 858–870.
- [43] G.J. Klir, B. Yuan, *Fuzzy Sets and Fuzzy Logic, Theory and Application*, Prentice-Hall, Englewood Cliffs, NJ, 1995.
- [44] D.C. Kozen, *Automata and Computability*, Springer, 1997.
- [45] E.T. Lee, L.A. Zadeh, Note on fuzzy languages, *Inf. Sci.* 1 (1969) 421–434.
- [46] Y.M. Li, W. Pedrycz, Fuzzy finite automata and fuzzy regular expressions with membership values in lattice ordered monoids, *Fuzzy Sets Syst.* 156 (2005) 68–92.
- [47] F. Lin, H. Ying, Fuzzy discrete event systems and their observability, in: *Proceedings of the 2001 IFSA/NAFIP Conference*, 2001, pp. 1271–1276.
- [48] F. Lin, H. Ying, Modeling and control of fuzzy discrete event systems, *IEEE Trans. Man Syst. Cybern. Part B* 32 (2002) 408–415.
- [49] F. Lin, H. Ying, R.D. MacArthur, J.A. Cohn, D. Barth-Jones, L.R. Crane, Decision making in fuzzy discrete event systems, *Inf. Sci.* 177 (2007) 3749–3763.
- [50] J.P. Liu, Y.M. Li, The relationship of controllability between classical and fuzzy discrete-event systems, *Inf. Sci.* 178 (2008) 4142–4151.
- [51] N. Lynch, F. Vaandrager, Forward and backward simulations: Part I. Untimed systems, *Inf. Comput.* 121 (1995) 214–233.
- [52] R. Milner, A calculus of communicating systems, in: G. Goos, J. Hartmanis (Eds.), *Lecture Notes in Computer Science*, vol. 92, Springer, 1980.
- [53] R. Milner, *Communication and Concurrency*, Prentice-Hall International, 1989.
- [54] R. Milner, *Communicating and Mobile Systems: the π -Calculus*, Cambridge University Press, Cambridge, 1999.
- [55] J.N. Mordeson, D.S. Malik, *Fuzzy Automata and Languages: Theory and Applications*, Chapman & Hall, CRC, Boca Raton, London, 2002.
- [56] R. Paige, R.E. Tarjan, Three partition refinement algorithms, *SIAM J. Comput.* 16 (1987) 973–989.
- [57] D. Park, Concurrency and automata on infinite sequences, in: P. Deussen (Ed.), *Proceedings of the 5th GI Conference*, Karlsruhe, Germany, *Lecture Notes in Computer Science*, vol. 104, Springer-Verlag, 1981, pp. 167–183.
- [58] K. Peeva, Finite L-fuzzy acceptors, *Int. J. Uncertainty Fuzziness Knowl. Based Syst.* 12 (2004) 89–104.
- [59] K. Peeva, Finite L-fuzzy machines, *Fuzzy Sets Syst.* 141 (2004) 415–437.
- [60] K. Peeva, Y. Kyosev, Fuzzy relational calculus: theory, applications, and software (with CD-ROM), in: *Advances in Fuzzy Systems—Applications and Theory*, vol. 22, World Scientific, 2004.

- [61] K. Peeva, Z. Zahariev, Computing behavior of finite fuzzy machines—algorithm and its application to reduction and minimization, *Inf. Sci.* 178 (2008) 4152–4165.
- [62] T. Petkovic, Congruences and homomorphisms of fuzzy automata, *Fuzzy Sets Syst.* 157 (2006) 444–458.
- [63] D.W. Qiu, Automata theory based on completed residuated lattice-valued logic (I), *Sci. China Ser. F* 44 (6) (2001) 419–429.
- [64] D.W. Qiu, Automata theory based on completed residuated lattice-valued logic (II), *Sci. China Ser. F* 45 (6) (2002) 442–452.
- [65] D.W. Qiu, Characterizations of fuzzy finite automata, *Fuzzy Sets Syst.* 141 (2004) 391–414.
- [66] D. Qiu, Supervisory control of fuzzy discrete event systems: a formal approach, *IEEE Trans. Man Syst. Cybern. Part B* 35 (2005) 72–88.
- [67] D.W. Qiu, Pumping lemma in automata theory based on complete residuated lattice-valued logic: a note, *Fuzzy Sets Syst.* 157 (2006) 2128–2138.
- [68] D.W. Qiu, F.C. Liu, Fuzzy discrete-event systems under fuzzy observability and a test algorithm, *IEEE Trans. Fuzzy Syst.* 17 (3) (2009) 578–589.
- [69] F. Ranzato, F. Tapparo, Generalizing the Paige–Tarjan algorithm by abstract interpretation, *Inf. Comput.* 206 (2008) 620–651.
- [70] M. Roggenbach, M. Majster-Cederbaum, Towards a unified view of bisimulation: a comparative study, *Theor. Comput. Sci.* 238 (2000) 81–130.
- [71] S. Roman, *Lattices and Ordered Sets*, Springer, New York, 2008.
- [72] D. Saha, An incremental bisimulation algorithm, in: V. Arvind, S. Prasad (Eds.), *FSTTCS 2007, Lecture Notes in Computer Science*, vol. 4855, Springer, Heidelberg, 2007, pp. 204–215.
- [73] E. Sanchez, Resolution of composite fuzzy relation equations, *Inf. Control* 30 (1976) 38–48.
- [74] E. Sanchez, Solutions in composite fuzzy relation equations: application to medical diagnosis in Brouwerian logic, in: M.M. Gupta, G.N. Saridis, B.R. Gaines (Eds.), *Fuzzy Automata and Decision Processes*, North-Holland, Amsterdam, 1977, pp. 221–234.
- [75] E. Sanchez, Resolution of eigen fuzzy sets equations, *Fuzzy Sets Syst.* 1 (1978) 69–74.
- [76] D. Sangiorgi, On the origins of bisimulation and coinduction, *ACM Trans. Programming Lang. Syst.* 31 (4) (2009) 111–151.
- [77] E.S. Santos, Maximin automata, *Inf. Control* 12 (1968) 367–377.
- [78] E.S. Santos, On reduction of max–min machines, *J. Math. Anal. Appl.* 37 (1972) 677–686.
- [79] E.S. Santos, Fuzzy automata and languages, *Inf. Sci.* 10 (1976) 193–197.
- [80] S.S. Skiena, *The Algorithm Design Manual*, Springer, London, 2008.
- [81] A. Stamenković, M. Ćirić, Construction of fuzzy automata from fuzzy regular expressions, *Fuzzy Sets Syst.* 199 (2012) 1–27.
- [82] A. Stamenković, M. Ćirić, J. Ignjatović, Reduction of fuzzy automata by means of fuzzy quasi-orders, *Inf. Sci.*, submitted for publication.
- [83] D.D. Sun, Y.M. Li, W.W. Yang, Bisimulation relations for fuzzy finite automata, *Fuzzy Syst. Math.* 23 (2009) 92–100 (in Chinese).
- [84] W. Wechler, *The Concept of Fuzziness in Automata and Language Theory*, Akademie-Verlag, Berlin, 1978.
- [85] W.G. Wee, On Generalizations of Adaptive Algorithm and Application of the Fuzzy Sets Concept to Pattern Classification, Ph.D. Thesis, Purdue University, June 1967.
- [86] W.G. Wee, K.S. Fu, A formulation of fuzzy automata and its application as a model of learning systems, *IEEE Trans. Syst. Man Cybern.* 5 (1969) 215–223.
- [87] L.H. Wu, D.W. Qiu, Automata theory based on complete residuated lattice-valued logic: reduction and minimization, *Fuzzy Sets Syst.* 161 (2010) 1635–1656.
- [88] H. Xing, D.W. Qiu, Pumping lemma in context-free grammar theory based on complete residuated lattice-valued logic, *Fuzzy Sets Syst.* 160 (2009) 1141–1151.
- [89] H. Xing, D.W. Qiu, Automata theory based on complete residuated lattice-valued logic: a categorical approach, *Fuzzy Sets Syst.* 160 (2009) 2416–2428.
- [90] H. Xing, D.W. Qiu, F.C. Liu, Automata theory based on complete residuated lattice-valued logic: pushdown automata, *Fuzzy Sets Syst.* 160 (2009) 1125–1140.
- [91] H. Xing, D.W. Qiu, F.C. Liu, Z.J. Fan, Equivalence in automata theory based on complete residuated lattice-valued logic, *Fuzzy Sets Syst.* 158 (2007) 1407–1422.