Decidability of minimization of fuzzy automata

Lvzhou Li[1,*], Daowen Qiu[1,2,†]

[1]Department of Computer Science, Sun Yat-sen University, Guangzhou 510006, China

[2]SQIG–Instituto de Telecomunicações, Departamento de Matemática, Instituto Superior Técnico, Technical University of Lisbon, Av. Rovisco Pais 1049-001, Lisbon, Portugal

Abstract

State minimization is a fundamental problem in automata theory. The problem is also of great importance in the study of fuzzy automata. However, most work in the literature considered only state reduction of fuzzy automata, whereas the state minimization problem is almost untouched for fuzzy automata. Thus in this paper we focus on the latter problem. Formally, the decision version of the minimization problem of fuzzy automata is as follows:

- Given a fuzzy automaton $\mathcal{A}$ and a natural number k, that is, a pair $\langle \mathcal{A}, k\rangle$, is there a k-state fuzzy automaton equivalent to $\mathcal{A}$?

We prove for the first time that the above problem is decidable for fuzzy automata over totally ordered lattices. To this end, we first give the concept of systems of fuzzy polynomial equations and then present a procedure to solve these systems. Afterwards, we apply the solvability of a system of fuzzy polynomial equations to the minimization problem mentioned above, obtaining the decidability. Finally, we point out that the above problem is at least as hard as PSAPCE-complete.

Index Terms: Fuzzy automata; Minimization; Decidability.

1. Introduction

Fuzzy finite automata were introduced by Wee [1] and Santos [2, 3] in the late 1960s. Subsequently, the fundamentals of fuzzy language theory were established by Lee and Zadeh [4], and by Thomason and Marinos [5]. For early surveys of the theory of fuzzy finite-state machines we can refer to [6, 7, 8]. Malik and Mordeson et al have systematically established the theory of algebraic fuzzy finite automata [9, 10], and the corresponding concept of fuzzy languages was introduced in [9, 10, 11]. The idea of fuzzy finite automata valued in some

*lilvzh@mail.sysu.edu.cn (L. Li).

†issqdw@mail.sysu.edu.cn (Corresponding author, D. Qiu).

abstract sets may go back to Wechler's work [12]. Notably, fuzzy finite automata have many significant applications such as in learning systems [13], pattern recognition [6, 10], data base theory [6, 10], discrete event systems [14, 15], and neural networks [6, 7, 10, 16, 17]. To date, there are still lots of important contributions to fuzzy finite automata in the academic community, but the minimization problem of fuzzy finite automata has still not been solved and remains open.

The purpose of this paper is to study the minimization problem of fuzzy finite automata, simply *fuzzy automata*. Minimizing a fuzzy automaton means finding a fuzzy automaton that has minimal number of states among the set of all fuzzy automata equivalent to the given one. Since fuzzy automata are generalizations of nondeterministic finite automata (NFAs), we first briefly recall the history of minimizing NFAs and related problems, which is helpful for us to understand the corresponding problems of fuzzy automata.

The study of the minimization problem for finite automata dates back to the early beginnings of automata theory. Although the problem's fundamental nature is sufficient to justify its study, the problem is of practical relevance, because regular languages are used in many applications, and one may like to represent the languages succinctly. As we know, deterministic finite automata (DFAs) and nondeterministic finite automata (NFAs) are two important representations of regular languages. It is well known that there exists an efficient algorithm running in time $O(n \log n)$ for minimizing any given n-state DFA [18]. The idea of the algorithm is to merge indistinguishable states of the given automaton, obtaining a unique reduced DFA which is the minimal one we find. However in the case of NFAs, the reduced automaton obtained by merging indistinguishable states is not always minimal. Therefore, one has to find other methods for minimizing NFAs.

It is obvious that there are only finite number of NFAs which have fewer states than the given automaton. As a result, we can always resort to an exhaustive procedure to find a minimal equivalent NFA. But this procedure is too lengthy to be practical. For that reason, many attempts have been made to search for more efficient algorithms for the minimization problem of NFAs. The first non-exhaustive search algorithm for minimizing NFAs was proposed by Kameda and Weiner [19] in 1970. In 1992, Sengoku [20] proposed another algorithm which was claimed to be more effective. However, both of the two algorithms are not of polynomial-time complexity, which is contrary to the deterministic case. Indeed, the minimization problem of NFAs was proven to be computationally hard (PSPACE-complete) by Jiang and Ravikumar [21] in 1993. Nevertheless this hardness did not prevent investigators from considering this problem any furthermore (e.g. [22, 23, 24, 25]). In addition, although the problem is known to be PSPACE-complete, one may propose algorithms of not too high complexities to compute relatively good approximations.

A closely related problem to minimization is the state reduction problem which does not aim at a minimal NFA, but find "reasonably" small NFAs that can be constructed efficiently. To this end, Ilie and Yu [26] (see also [27]) first proposed the method of reducing the size of NFAs by using equivalence relations (right invariant equivalence and left invariant equivalence). This work leaded to a series of further work on this topic (e.g. [28, 29, 30, 31, 32]). The basic idea of the method is to merge indistinguishable states, which resembles the minimization algorithm for DFAs, but is more complicated. Note that right invariant equivalence was also studied from another aspect by Calude et al. [33] under the name well-behaved equivalence. In fact, right invariant equivalence shares the same idea with a central concept in theoretical computer science—"bisimulation equivalence" which originates from concurrency theory and modal logic but now is a rich concept appearing in various areas of theoretical computer science such as formal verification, model checking, etc.

Fuzzy finite automata, simply fuzzy automata, are generalizations of NFAs, and the mentioned problems concerning minimization and reduction of states are also present in work with fuzzy automata. Reduction of number of states of fuzzy automata was studied in [34, 35, 36, 37, 39, 38, 40], and the algorithms given there were also based on the idea of merging indistinguishable states. Although these algorithms were claimed to be minimization algorithms, the term "minimization" in the mentioned references does not mean state minimization in our sense, since it does not mean the usual construction of the minimal one in the set of all fuzzy automata recognizing a given fuzzy language, but just the procedure of merging indistinguishable states. Therefore, these are just state reduction algorithms. Very recently, Ćirić et al [41] made new progress on reduction of fuzzy automata. Specifically, they extended the concept of equivalence on NFAs to fuzzy equivalences on fuzzy automata. Then they studied state reduction of fuzzy automata by means of fuzzy equivalences, which parallels the work of Ilie and Yu [26] on NFAs. In all previous papers dealing with reduction of fuzzy automata (cf. [34, 35, 36, 37, 39, 38, 40]) only reductions by means of crisp equivalence were investigated, whereas Ćirić et al [41] showed that better reductions can be achieved by employing fuzzy equivalence.

From the above introduction, one can find that so far much attention has been paid to reduction of fuzzy automata, but another important problem is almost untouched—the minimization problem. Obviously, the minimization problem looks more difficult than the state reduction problem, similar to the situations in the NFA case. However, this should not become the reason for one not trying to address the minimization problem. Indeed, in this paper we are going to attempt to consider this problem and hope this would inspire some further discussion. Formally, the decision version of the minimization problem of fuzzy automata is described as follows:

- Given a fuzzy automaton $\mathcal{A}$ and a natural number k, that is, a pair $\langle \mathcal{A}, k\rangle$, is there a k-state fuzzy automaton equivalent to $\mathcal{A}$?

Note that fuzzy automata can be defined over different underlying structures of truth values. A general model is the one defined over complete residuated lattices proposed by Qiu [42, 43]. In this paper, fuzzy automata are restricted to be over totally ordered lattices. We prove that the above problem is decidable for this model. The result will be based, on one side, on the decidability of the equivalence problem of fuzzy automata and, on the other side, on the solvability of a system of fuzzy polynomial equations. The former base is an active problem having received much attention and can be addressed successfully. The later one is a new problem and involves a new concept—systems of fuzzy polynomial equations. A system of fuzzy polynomial equations resembles the usual notion of a system of polynomial equations on the real-number field. However, the ideas for solving the two types of systems are greatly different. Therefore, in this paper we will first present a procedure to solve systems of fuzzy polynomial equations, and afterwards apply the result to address the minimization problem of fuzzy automata, obtaining the decidability result.

The reminder of this paper is organized as follows. In Section 2 some preliminary knowledge is given. In Section 3 we give the concept of systems of fuzzy polynomial equations and present a procedure to solve these systems, establishing a crucial base for the minimization problem. In Section 4 we give a condition for two fuzzy automata being equivalent, obtaining another base for the above problem. Section 5 is a main part of this paper where we prove that the minimization problem of fuzzy automata is decidable and furthermore we point out that it is at least as hard as PSPACE-complete. Finally, some conclusions are made in Section 6.

2. Preliminaries

In this paper $\mathbf{L} = (L, \wedge, \vee, 0, 1)$ means a lattice with the least element 0 and the greatest element 1, where L is a poset with the partial order relation $\leq$, and for any $x, y \in L$, $x \vee y$ and $x \wedge y$ denote the *least upper bound* and the *greatest lower bound* of $\{x, y\}$, respectively. $\mathbf{L}$ is said to be *totally ordered*, if L is a totally ordered set. From now on we assume that $\mathbf{L}$ is a totally ordered lattice.

A *fuzzy subset* of a set A over $\mathbf{L}$, or simply a *fuzzy subset* of A, is any mapping from A into L. An $n \times m$ *fuzzy matrix* over $\mathbf{L}$ is given by $A = [a_{ij}]_{n \times m}$ where $a_{ij} \in L$. $L^{n\times m}$ denotes the set of all $n \times m$ fuzzy matrices over $\mathbf{L}$. Let $A \in L^{n\times m}$ and $B \in L^{m\times l}$. Their product, denoted by $A \circ B$, is defined by

$$(A \circ B)_{ij} = \bigvee_{k=1}^{m} a_{ik} \wedge b_{kj} \tag{1}$$

where $1 \leq i \leq n$, $1 \leq j \leq l$. For two fuzzy matrices A and B, their direct sum is $A \oplus B = \begin{pmatrix} A & O \\ O & B \end{pmatrix}$ where O denotes a matrix with all elements being the least element 0.

Given $a, b \in L$ satisfying $a \leq b$, $[a, b] = \{x \in L : a \leq x \leq b\}$ is called an *interval* of L. By this, we may identify L with the interval $[0, 1]$[1]. An n-dimensional *interval vector* is an n-tuple $\mathcal{X} = (X_1, \cdots, X_n)$ where $X_i \subseteq L$. Given two n-dimensional interval vectors $\mathcal{X}$ and $\mathcal{Y}$, their intersection is defined by $\mathcal{X} \cap \mathcal{Y} = (X_1 \cap Y_1, \cdots, X_n \cap Y_n)$. Let S_1 and S_2 be two sets of interval vectors with the same dimension. The *product* of S_1 and S_2 is defined by

$$S_1 \star S_2 = \{\mathcal{X} \cap \mathcal{Y} : \mathcal{X} \in S_1, \mathcal{Y} \in S_2\}. \tag{2}$$

Note that two nonempty sets S_1 and S_2 may produce a product set with $(\emptyset, \emptyset, \cdots, \emptyset)$ as its unique element where $\emptyset$ denotes the empty set. It holds that $|S_1 \star S_2| \leq |S_1|.|S_2|$ where $|S|$ denotes the cardinality of set S.

3. Solving the systems of fuzzy polynomial equations

A *fuzzy monomial* over $\mathbf{L} = (L, \wedge, \vee, 0, 1)$ is a finite expression $M = a \wedge x_1 \wedge x_2 \wedge \cdots \wedge x_k$ where $k \geq 0$, $a \in L$, $x_1, \cdots, x_k$ are variables with values in L. When $k = 0$, it means a constant. If the coefficient a is the greatest element 1, the monomial is then simply written by $x_1 \wedge x_2 \wedge \cdots \wedge x_k$. A *fuzzy polynomial* over $\mathbf{L}$ is an expression of the form $M_1 \vee M_2 \vee \cdots \vee M_k$ where $k \geq 1$ and $M_1, \cdots, M_k$ are monomials over $\mathbf{L}$. It is usually denoted by $P = \bigvee_{i=1}^{k} M_i$.

A *system of fuzzy polynomial (in)equations (SFPI)* is of the form

$$\begin{cases} P_1 \bowtie a_1 \\ \quad\vdots \\ P_m \bowtie a_m \end{cases} \tag{3}$$

where $\bowtie \in \{=, \leq, \geq, <, >\}$, P_i's are polynomials and $a_i \in L$ for $i = 1, \cdots, m$. If $\bowtie$ is always chosen to be "$=$", then (3) is called a *system of fuzzy polynomial equations (SFPE)*. Suppose the system involves n variables $(x_1, \cdots, x_n)$. $X^0 = (x_1^0, \cdots, x_n^0)$ is called a *point solution* of (3), if (3) holds when $(x_1, \cdots, x_n)$ is assigned with X^0. If (3) has a point solution, then it is said to be *solvable*. Otherwise, it is *unsolvable*. An n-dimensional interval vector $\mathcal{X} = (X_1, \cdots, X_n)$ where $X_i \subseteq L$ is called an *interval solution* of (3), if each n-tuple $(x_1^0, \cdots, x_n^0)$ with $x_i^0 \in X_i$ is a point solution of (3) and $(X_1, \cdots, X_n)$ is maximum[2] with respect to this property.

Remark 1. *First we mention that a special case of the above discussion that each monomial M involves only one variable was considered in [44]. Secondly, in the above definitions if*

[1] One should not confuse the interval with the real unit interval $[0, 1]$. In this paper 0 and 1 respectively stand for the least element and the greatest element of a lattice.

[2] Note that the inclusion "$\subseteq$" is a partial order on 2^L, and it can be extended up to be a partial order on the set of interval vectors.

$\vee$ and $\wedge$ are respectively replaced with the usual addition "+" and the multiplication ".", and L is replaced with the real number field $\mathbb{R}$, then P is the usual polynomial and (3) is a system of polynomial (in)equations. Note that deciding the solvability of a system of polynomial (in)equations is a very important problem which has wide applications. A more general problem is the decision problem for the existential theory of the reals which is the problem of deciding if the set $\{x \in \mathbb{R}^n : P(x)\}$ is non-empty, where $P(x)$ is a predicate which is a boolean combination of atomic predicates either of the form $f_i(x) \geq 0$ or $f_j(x) > 0$, f's being real polynomials (with rational coefficients). It is known that this problem can be decided in PSPACE [45, 46] and it has been applied to the complexity analysis on the reachability of Recursive Markov Chains [47] and on the minimization problem of quantum and probabilistic automata [48].

Now we consider the decision problem of SFPI, that is:

- Is it decidable whether a given system like (3) is solvable or not?

In the following we consider a special case that the relation operator $\bowtie$ in (3) is always chosen to be "=" and all polynomials take 1 as coefficients. We will show that the above problem is decidable in this case. Afterwards, we apply the result to the minimization problem of fuzzy automata. Hereafter our aim is to determine whether the following system is solvable or not:

$$\begin{cases} P_1 = a_1 \\ \quad\vdots \\ P_m = a_m \end{cases} \tag{4}$$

where all polynomials P_i's take 1 as coefficients.

First we consider the case of one equation with a monomial, i.e.

$$x_1 \wedge x_2 \wedge \cdots \wedge x_n = a. \tag{5}$$

It is easy to see that a solution of this equation has the property that one x_i equals a and all others are not less than a. More formally, the equation has n interval solutions as follows

$$\begin{array}{c} ([a,a],[a,1],[a,1],\cdots,[a,1]), \\ ([a,1],[a,a],[a,1],\cdots,[a,1]), \\ \vdots \\ ([a,1],[a,1],\cdots,[a,1],[a,a]) \end{array} \tag{6}$$

where the interval $[a,a]$ stands for the point a. Similarly, the solution of the inequality

$$x_1 \wedge x_2 \wedge \cdots \wedge x_n \leq a \tag{7}$$

has the property that one x_i is no more than a and all others can be assigned with arbitrary values. Thus, the inequality has n interval solutions given by

$$\begin{array}{c}([0,a],[0,1],[0,1],\cdots,[0,1]),\\ ([0,1],[0,a],[0,1],\cdots,[0,1]),\\ \vdots\\ ([0,1],[0,1],\cdots,[0,1],[0,a]).\end{array} \tag{8}$$

Next we consider an equation with a polynomial:

$$P = M_1 \vee M_2 \vee \cdots \vee M_k = a. \tag{9}$$

where M_k's are monomials. Eq. (9) is solvable if and only if one monomial M_i equals a and others are no more than a. Therefore there are k possible cases. For example, one case is as follows:

$$\begin{cases} M_1 = a \\ M_2 \le a \\ \quad\vdots \\ M_k \le a \end{cases} \tag{10}$$

We can search for all possible interval solutions for each equation or inequality in (10) as we dealt with (5) and (7).

Let n_i be the number of variables in M_i and let n be the total number of variables in (10). Then $n_i \le n$ for $i = 1, \cdots, k$. Here we need extend a solution on n_i variables to one on n variables. For example, the first equation in (10) has only n_1 variables and thus its interval solution is an n_1-tuple $\mathcal{X} = (X_1, \cdots, X_{n_1})$ with $X_i \subseteq L$. Assume that all variables in (10) are ordered by some order, for example the lexicographic order. Without loss of generality assume that the variables involved in M_1 are ordered at the head. Then the variables that appear in (10) but not in M_1 have no impact on the first equation in (10) and can be chosen arbitrarily from L. Therefore, $\mathcal{X}$ can be extended to the following form

$$\overline{\mathcal{X}} = (X_1, \cdots, X_{n_1}, [0,1], \cdots, [0,1]). \tag{11}$$

By this way, an interval solution for each equation or inequality of (10) will be an n-dimensional interval vector. Let S_i be the set of interval solutions of the ith formula in (10). Then the system (10) has a set of interval solutions $S = S_1 \star \cdots \star S_k$ with $|S| \le n_1.n_2 \cdots n_k \le n^k$.

Note that the equality (9) has k cases to be verified. Therefore, the number of interval solutions of Eq. (9) is upper bounded by kn^k.

Now we return to the solution of system (4). Let the ith equation have a set of interval solutions T_i where $i = 1, \cdots, m$. Then the interval solution set of (4) is $T = T_1 \star \cdots \star T_m$.

Assume that the system (4) involves n variables and each polynomial consists of at most k monomials. Then we have $|T| \leq |T_1| \cdots |T_m| \leq (kn^k)^m$. Consequently, we obtain the following theorem.

Theorem 2. *There exists an algorithm running in time $O((kn^k)^m)$ to search for all possible solutions to a system of fuzzy polynomial equations like (4) and thus it is decidable whether the system is solvable, where n is the number of variables, m is the number of equations, and k is the largest number of monomials involved in an equation.*

Proof. Using the procedure described above, we can exhaustively search for all possible solutions of the system. The number of interval solutions is at most $(kn^k)^m$. So the time is $O((kn^k)^m)$. Note that the system is solvable if and only if there exists an interval solution in the form $(X_1, \cdots, X_n)$ where $X_i \neq \emptyset$ for all i. Thus, after finding all possible interval solutions, we check if there exists a solution satisfying the above condition. If yes, then the system is solvable. Otherwise, it is unsolvable. $\square$

Remark 3. *In the above discussion, it was assumed that a system consists of only equations and all involved polynomials take* 1 *as coefficients. In fact, the above procedure can be extended to more general cases. Firstly, it can be easily adopted to deal with systems of inequalities and equations with* 1 *as coefficients. Secondly, by the similar idea, we can also deal with the most general case—systems consist of inequalities and equations with arbitrary coefficients. However, we will not discuss these general cases in any more detail, since they have no close relation with our main purpose in this paper—minimization of fuzzy finite automata.*

Theorem 2 has presented a procedure to search for all solutions to systems like (4). However, if we only want to determine whether the system is solvable or not, but not to search for all solutions, then we can present a simpler procedure for that. Let $\mathcal{V}$ denote the set consisting of distinct elements from $\{a_1, \cdots, a_m\}$. Then we get the following result.

Theorem 4. *The system (4) is solvable if and only if there exists $X^0 \in \mathcal{V}^n$ such that X^0 is a point solution of (4) where n is the number of variables involved in the system.*

Proof. First the "if" part is ready. Thus we focus on the "only if" part. Suppose that the system (4) is solvable. Then it necessarily has an interval solution $(X_1, \cdots, X_n)$ where $X_i \neq \emptyset$ for all i. More specifically, since every interval X_i is the intersection of intervals like the ones given in (6) and (8), every interval X_i is finally among the following forms $[a, b], [0, c], [d, 1], [0, 1]$ where $a, b, c, d \in \mathcal{V}$. Thus, we get a point solution $(x_1^0, \cdots, x_n^0)$ where each x_i^0 is given by

$$x_i = \begin{cases} a \text{ or } b, & \text{if } X_i = [a, b]; \\ c, & \text{if } X_i = [0, c]; \\ d, & \text{if } X_i = [d, 1]; \\ y \in \mathcal{V}, & \text{if } X_i = [0, 1] \end{cases} \tag{12}$$

where y is an arbitrary value from $\mathcal{V}$. It is obvious that $(x_1^0, \cdots, x_n^0) \in \mathcal{V}^n$. Hence, we have completed the proof. $\square$

Having in mind Theorem 4 we propose the following algorithm (Algorithm I) for establishing the solvability of systems like (4).

Input: a system like (4) with n variables
output: give a solution or return "unsolvable"
Step 1:
find the set $\mathcal{V}$ from $\{a_1, \cdots, a_m\}$.
Step 2:
While (take an element X^0 from $\mathcal{V}^n$) **do**
If (X^0 is a solution of (4)) return X^0
Step 3:
return "unsolvable"

Algorithm I: determine whether a system of fuzzy polynomial equations like (4) is solvable.

In Section 5, we will apply the above results to address the minimization problem of fuzzy automata.

4. Equivalence of fuzzy automata

Below we first give the definition of fuzzy automata and then present a condition for two fuzzy automata being equivalent which is an indispensable base for the minimization problem. Some notations used in the sequel are firstly explained here. Let Σ^* denote the free monoid over the alphabet Σ, and let $\lambda \in \Sigma^*$ be the empty word. For $x \in \Sigma^*$, $|x|$ denotes its length. Let $\Sigma^{\leq k} = \{x \in \Sigma^* : |x| \leq k\}$.

Definition 5. *A fuzzy finite automaton over* $\mathbf{L} = (L, \wedge, \vee, 0, 1)$*, simply fuzzy automaton, is a five-tuple* $\mathcal{A} = (S, \Sigma, \pi, \delta, \eta)$ *where* S *is a finite state set,* Σ *is a finite alphabet,* π *is a fuzzy subset of* S*, called the fuzzy set of initial states,* $\delta : S \times \Sigma \times S \to L$ *is a fuzzy subset of* $S \times \Sigma \times S$*, called the fuzzy transition function,* η *is a fuzzy subset of* S*, called the fuzzy set of terminal states.*

We can interpret $\delta(s, \sigma, s')$ as the degree to which an input letter $\sigma \in \Sigma$ causes a transition from a state $s \in S$ into a state $s' \in S$. The mapping δ can be extended up to a mapping $\delta^* : S \times \Sigma^* \times S \to L$ as follows:

$$\delta^*(s, \lambda, s') = \begin{cases} 1, & \text{if } s = s', \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

where $s, s' \in S$, and

$$\delta^*(s, x\sigma, s') = \bigvee_{t \in S} \delta^*(s, x, t) \wedge \delta(t, \sigma, s') \tag{14}$$

where $s, s' \in S$, $x \in \Sigma^*$ and $\sigma \in \Sigma$.

A *fuzzy language* in Σ^* over $\mathbf{L}$, or briefly a fuzzy language, is a fuzzy subset of Σ^*, i.e., a mapping from Σ^* into L. $\mathfrak{F}(\Sigma^*)$ stands for the set of all fuzzy languages in Σ^*. A fuzzy automaton $\mathcal{A} = (S, \Sigma, \pi, \delta, \eta)$ *recognizes* a fuzzy language $f \in \mathfrak{F}(\Sigma^*)$ if for any $x \in \Sigma^*$

$$f(x) = \bigvee_{s,t \in S} \pi(s) \wedge \delta^*(s, x, t) \wedge \eta(t). \tag{15}$$

In other words, the equality (15) means that the membership degree of the word x belonging to the fuzzy language f is equal to the degree to which $\mathcal{A}$ recognizes or accepts the word x. The unique fuzzy language recognized by a fuzzy automaton $\mathcal{A}$ is denoted by $f_{\mathcal{A}}$.

Let $|S| = n$. For each word $x \in \Sigma^*$, we define an $n \times n$ fuzzy matrix δ_x by

$$\delta_x(s, s') = \delta^*(s, x, s') \tag{16}$$

where $s, s' \in S$. Then it is readily seen that

$$\delta_x = \delta_{x_1} \circ \cdots \circ \delta_{x_k} \tag{17}$$

for $x = x_1 \cdots x_k \in \Sigma^*$. By using this notation, the equality (15) can be rewritten as

$$f(x) = \pi \circ \delta_x \circ \eta \tag{18}$$

where and hereafter we identify the fuzzy subsets π, η with a $1 \times n$ and an $n \times 1$ fuzzy matrices, respectively.

A basic problem concerning fuzzy automata is the equivalence problem, that is, deciding whether two given fuzzy automata recognize the same fuzzy language. Formally the definition is given as follows.

Definition 6. *Two fuzzy automata $\mathcal{A}_1$ and $\mathcal{A}_2$ are said to be k-equivalent, denoted by $\mathcal{A}_1 \approx_k \mathcal{A}_2$, if $f_{\mathcal{A}_1}(x) = f_{\mathcal{A}_2}(x)$ holds for all $x \in \Sigma^{\leq k}$. Furthermore, they are said to be equivalent, denoted by $\mathcal{A}_1 \approx \mathcal{A}_2$, if $f_{\mathcal{A}_1}(x) = f_{\mathcal{A}_2}(x)$ holds for all $x \in \Sigma^*$.*

In the following, we give a condition for two fuzzy automata being equivalent.

Theorem 7. *Two fuzzy automata $\mathcal{A}_i = (S^{(i)}, \Sigma, \pi^{(i)}, \delta^{(i)}, \eta^{(i)})$ $(i = 1, 2)$ are equivalent if and only if they are $(d^{(n_1+n_2)} - 1)$-equivalent, where n_1 and n_2 are numbers of states of $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively, and d is the cardinality of the following set:*

$$\begin{aligned} V &= \{\delta^{(1)}(s, \sigma, t) : s, t \in S^{(1)}, \sigma \in \Sigma\} \cup \{\eta^{(1)}(s) : s \in S^{(1)}\} \\ &\cup \{\delta^{(2)}(s, \sigma, t) : s, t \in S^{(2)}, \sigma \in \Sigma\} \cup \{\eta^{(2)}(s) : s \in S^{(2)}\}. \end{aligned}$$

Proof. Let $M(\sigma) = \delta_\sigma^{(1)} \oplus \delta_\sigma^{(2)}$ for all $\sigma \in \Sigma$, and $\eta = \begin{pmatrix} \eta^{(1)} \\ \eta^{(2)} \end{pmatrix}$, being an $(n_1+n_2)\times 1$ matrix. Then we have

$$f_{\mathcal{A}_1}(x) = \pi^{(1)} \circ \delta_x^{(1)} \circ \eta^{(1)} = (\pi^{(1)}, \mathbf{0}) \circ M(x) \circ \eta, \tag{19}$$
$$f_{\mathcal{A}_2}(x) = \pi^{(2)} \circ \delta_x^{(2)} \circ \eta^{(2)} = (\mathbf{0}, \pi^{(2)}) \circ M(x) \circ \eta \tag{20}$$

where $M(x) = M(x_1) \circ \cdots \circ M(x_k)$ for $x = x_1 \cdots x_k \in \Sigma^*$ and $\mathbf{0}$ denotes a row vector with all elements being 0. Let

$$\varphi(k) = \{M(x) \circ \eta : x \in \Sigma^{\leq k}\}$$

and

$$\varphi = \{M(x) \circ \eta : x \in \Sigma^*\}.$$

Then the following result holds.

Proposition 8. *There exists an integer $l \leq d^{n_1+n_2} - 1$ such that*

$$\varphi(0) \subset \varphi(1) \subset \cdots \subset \varphi(l) = \varphi(l+1) = \varphi(l+2) = \cdots = \varphi$$

where $\subset$ denotes the proper inclusion.

Proof of Proposition 8. First it is obvious that $\varphi(i) \subseteq \varphi(i+1) \subseteq \varphi$ holds for $i = 0, 1, \cdots$. Note that $|\varphi| \leq d^{n_1+n_2}$. Thus, there exists $l \leq |\varphi| - 1 \leq d^{n_1+n_2} - 1$ such that $\varphi(l) = \varphi(l+1)$. Next we prove that $\varphi(l) = \varphi(l+j)$ holds for $j = 2, 3, \cdots$. In fact we only need prove $\varphi(l) = \varphi(l+2)$. Take $v \in \varphi(l+2)$. We have

$$\begin{array}{rcll} v &=& M(x) \circ \eta, & x \in \Sigma^{\leq (l+2)} \\ &=& M(\sigma) \circ M(x') \circ \eta, & \sigma \in \Sigma, x' \in \Sigma^{\leq (l+1)} \\ &=& M(\sigma) \circ v', & v' \in \varphi(l) \\ &\in& \varphi(l+1) = \varphi(l). & \end{array}$$

This completes the proof of Proposition 8.

By the above results we obtain

$$\begin{array}{rcl} \mathcal{A}_1 \approx \mathcal{A}_2 &\Leftrightarrow& f_{\mathcal{A}_1}(x) = f_{\mathcal{A}_2}(x), \forall x \in \Sigma^* \\ &\Leftrightarrow& (\pi^{(1)}, \mathbf{0}) \circ M(x) \circ \eta = (\mathbf{0}, \pi^{(2)}) \circ M(x) \circ \eta, \forall x \in \Sigma^* \\ &\Leftrightarrow& (\pi^{(1)}, \mathbf{0}) \circ v = (\mathbf{0}, \pi^{(2)}) \circ v, \forall v \in \varphi \\ &\Leftrightarrow& (\pi^{(1)}, \mathbf{0}) \circ v' = (\mathbf{0}, \pi^{(2)}) \circ v', \forall v' \in \varphi(k) \\ &\Leftrightarrow& f_{\mathcal{A}_1}(x) = f_{\mathcal{A}_2}(x), \forall x \in \Sigma^{\leq k} \\ &\Leftrightarrow& \mathcal{A}_1 \approx_k \mathcal{A}_2 \end{array}$$

where $k = d^{n_1+n_2} - 1$. Thus we have completed the proof of Theorem 7. $\square$

5. Minimization of fuzzy automata

The study of the minimization problem for finite automata dates back to the early beginnings of automata theory. This problem is of practical relevance, because regular languages are used in many applications, and one may like to represent the languages succinctly. In the past over forty years, fuzzy automata have received much attention. In the study of fuzzy automata, the minimization problem is also a very important problem and worthy of serious consideration. Note that a closely related problem to minimization is the state reduction problem which does not aim at a minimal one, but find "reasonable" small automata which can be efficiently constructed. As mentioned in Section 1, while much attention has been paid to reduction of fuzzy automata, the minimization problem is almost untouched. Thus in this section we focus on the minimization problem. Formally, the decision version of the minimization problem of fuzzy automata, is as follows:

- Given a fuzzy automaton $\mathcal{A}$ and a natural number k, that is, a pair $\langle \mathcal{A}, k \rangle$, is there a k-state fuzzy automaton equivalent to $\mathcal{A}$?

In this section, we prove the above problem is decidable, based on the solvability of systems of fuzzy polynomial equations (Theorem 4) and on the condition of equivalence between fuzzy automata (Theorem 7). Furthemore, we point out that the minimization problem is at least as hard as PSPACE-complete.

5.1. Decidability of minimization

The main idea for addressing the minimization problem of fuzzy automata is briefly depicted as follows:

1. First, for a given pair $\langle \mathcal{A}, k \rangle$, define the set

$$\mathbb{S}(\mathcal{A}, k) = \{\mathcal{A}' : \mathcal{A}' \approx \mathcal{A}, \text{ is a fuzzy automaton with } k \text{ states}\}. \tag{21}$$

2. Next, we prove that $\mathbb{S}(\mathcal{A}, k)$ can be described by a system of fuzzy polynomial equations. Then, by Theorem 4, there exists an algorithm to decide whether $\mathbb{S}(\mathcal{A}, k)$ is nonempty or not, and furthermore, if it is nonempty, a k-state fuzzy automaton $\mathcal{A}'$ equivalent to $\mathcal{A}$ is given.

Specifically, we have the following result.

Theorem 9. *The minimization problem of fuzzy automata is decidable.*

Proof. For a given n-state fuzzy automaton $\mathcal{A} = (S, \Sigma, \pi, \delta, \eta)$ and a natural number k, define the set $\mathbb{S}(\mathcal{A}, k)$ given by (21). Suppose that a k-state fuzzy automaton has the form $\mathcal{A}' = (S', \Sigma, \pi', \delta', \eta')$ with $|S'| = k$. Equivalently, $\mathbb{S}(\mathcal{A}, k)$ can be represented as follows:

$$\mathbb{S}(\mathcal{A}, k) = \{X \in L^{2k+|\Sigma|k^2} : P(X)\} \tag{22}$$

where:

- X is a vector consisting of $(2k + |\Sigma|k^2)$ variables which take values from L. More specifically, X is decomposed into several parts:
  - $X_{\pi'}, X_{\eta'} \in L^k$ are respectively used to represent the initial state π' and the final state η' of $\mathcal{A}'$.
  - $X_{\delta_\sigma} \in L^{k^2}$ for each $\sigma \in \Sigma$ is used to represent a fuzzy transition matrix δ_σ of $\mathcal{A}'$.
- $P(X)$ is a system of fuzzy polynomial equations given by:

$$X_{\pi'} \circ X_{\delta_x} \circ X_{\eta'} = \pi \circ \delta_x \circ \eta, \forall x \in \Sigma^* \tag{23}$$

  where $X_{\delta_x} = X_{\delta_{x_1}} \circ \cdots \circ X_{\delta_{x_m}}$ for $x = x_1 \cdots x_m \in \Sigma^*$.

It is obvious that a point $X^0 \in L^{2k+|\Sigma|k^2}$ is a solution to the system (23) if and only if it represents a k-state fuzzy automaton $\mathcal{A}'$ equivalent to $\mathcal{A}$. However, since (23) consists of infinitely many equations, there exists no algorithmic procedure to determine whether it is solvable. Thereby we need reduce (23) to a system of finitely many equations.

Let

$$V = \{\pi(s), \eta(s), \delta_\sigma(s, s') : s, s' \in S, \sigma \in \Sigma\}$$

and let

$$d = |V| + |\Sigma|k^2 + k.$$

We consider the following finite system of equations:

$$X_{\pi'} \circ X_{\delta_x} \circ X_{\eta'} = \pi \circ \delta_x \circ \eta, \forall x \in \Sigma^* \text{ with } |x| \leq d^{n+k} - 1. \tag{24}$$

Indeed, we obtain the following result.

Claim 1. *$X^0 \in L^{2k+|\Sigma|k^2}$ is a solution to the system (23) if and only if it is a solution to the system (24).*

Proof. The "only if " part is obvious. Thus we prove the "if" part. Suppose that $X^0 \in L^{2k+|\Sigma|k^2}$ is a solution to the system (24). Assume that the number of different values involved by $\{X^0_{\eta'}, \eta, \delta_\sigma, X^0_{\delta_\sigma} : \sigma \in \Sigma\}$ is d'. Then we have $d' \leq d$. Now it follows from Theorem 7 that if $X^0_{\pi'} \circ X^0_{\delta_x} \circ X^0_{\eta'} = \pi \circ \delta_x \circ \eta$ holds for $x \in \Sigma^*$ with $|x| \leq (d')^{n+k} - 1$, then it also holds for all $x \in \Sigma^*$. Remember that X^0 is a solution to the system (24) and $d' \leq d$. Thus by the above discussion X^0 is also a solution to the system (23). This completes the proof of Claim 1. $\square$

Furthermore, we can reduce the system (24) to a more simple one. Formally, we have

Claim 2. *The system (24) is solvable if and only if the following system is solvable:*

$$X_{\pi'} \circ X_{\delta_x} \circ X_{\eta'} = \pi \circ \delta_x \circ \eta, \forall x \in \Sigma^* \text{ with } |x| \leq |V|^{n+k} - 1. \tag{25}$$

Proof. First, the "only if" part is obvious, since $|V| < d$. Now suppose that the system (25) is solvable. Note that the values of the right side of (25) are necessarily to be in the set V. Therefore, by Theorem 4, the system (25) is solvable in the set V, which means that there exists $X^0 \in V^{2k+|\Sigma|k^2}$ such that $X^0_{\pi'} \circ X^0_{\delta_x} \circ X^0_{\eta'} = \pi \circ \delta_x \circ \eta$ holds for $x \in \Sigma^*$ with $|x| \leq |V|^{n+k} - 1$. Then, by Theorem 7, the equality holds for all $x \in \Sigma^*$, and thus it necessarily holds for $x \in \Sigma^*$ with $|x| \leq d^{n+k} - 1$, which means that X^0 is a solution to the system (24). □

At length, $P(X)$ has been reduced to be a finite one given by (25). Correspondingly, determining whether the set $\mathbb{S}(\mathcal{A}, k)$ is nonempty is equivalent to determining whether the system (25) is solvable. The later problem can be addressed in terms of Theorem 4.

In conclusion the above procedure is refined to a minimization algorithm (Algorithm II). This completes the proof of Theorem 9. □

Input: a fuzzy automaton $\mathcal{A}$ and a natural number k
output: either return a fuzzy automaton $\mathcal{A}'$ equivalent to $\mathcal{A}$ with k states or return "empty"
Step 1:

construct the set $\mathbb{S}(\mathcal{A}, k)$ given in (22) where $P(x)$ is given by (25)

Step 2:

invoke Algorithm I to decide whether $\mathbb{S}(\mathcal{A}, k)$ is nonempty

Step 3:

If $\mathbb{S}(\mathcal{A}, k) = \emptyset$ return "empty"; **else** return $X^0 \in \mathbb{S}(\mathcal{A}, k)$

Algorithm II: the minimization algorithm

Complexity analysis. Here we take a complexity analysis of the above procedure. In the following we assume that the two operations $\vee$ and $\wedge$ can be done in constant time. For simplicity, let $c = |V|^{n+k} - 1$. The number of equations in (25) is

$$N = 1 + |\Sigma| + |\Sigma|^2 + \cdots + |\Sigma|^c.$$

To solve the system (25) we need to compute the values in the right side which can be efficiently computed by the following two steps:

- first compute the set $\varphi(c) = \{\delta_x \circ \eta : |x| \leq c\}$;
- then compute the value $\pi \circ v$ for each $v \in \varphi(c)$.

Note that in this first step upon computation of $\delta_x \circ \eta$, computation of $\delta_\sigma \circ \delta_x \circ \eta$ requires $O(n^2)$ operations. Thus, we compute all $\delta_x \circ \eta$ from $l = 0$ to $l = c$ where $l = |x|$. Then the number of operations required is give as follows:

length	number of operations
$l = 0$	$O(1)$
$l = 1$	$O(\|\Sigma\|n^2)$
$l = 2$	$O(\|\Sigma\|^2 n^2)$
$\cdots$	$\cdots$
$l = c$	$O(\|\Sigma\|^c n^2)$

Thus, the number of operations required in the first step is $O(Nn^2)$. The second step requires $O(Nn)$ operations. Therefore the computation of the right side of (25) requires $O(Nn^2)$ operations.

Now in order to determine whether the system (25) is solvable or not, by Theorem 4 we need to check all $X \in V^{2k+|\Sigma|k^2}$. For each X, it requires $O(Nk^2)$ operations to compute the left side.

Therefore, the total number of operations required is

$$O(|V|^{2k+|\Sigma|k^2} Nk^2) + O(Nn^2).$$

5.2. The minimization problem is at least as hard as PSPACE-complete

In the last section we have presented a procedure to address the minimization problem of fuzzy automata, but the complexity is not good enough. Then one may ask whether there exists a more efficient algorithm to the minimization problem. The answer may be yes, but at present what we are sure is that the minimization problem is at least as hard as PSPACE-complete.

We recall that in Definition 5, a fuzzy automaton $\mathcal{A} = (S, \Sigma, \pi, \delta, \eta)$ is defined on a general totally ordered lattice $\mathbf{L} = (L, \vee, \wedge, 0, 1)$. Now we consider a special case where $L = \{0, 1\}$. In this case, $\mathcal{A}$ reduces to an NFA, which follows from the following observations:

- $\pi(s_i) = 1$ stands for $s_i \in S$ being an initial state, and $\eta(s_i) = 1$ indicates s_i being an accepting state.
- $\delta(s_i, \sigma, s_j) = 1$ stands for that the current state s_i will change to s_j when the automaton is fed with the symbol σ.

- The language accepted by $\mathcal{A}$ is $\mathcal{L} = \{x \in \Sigma^* : f_{\mathcal{A}}(x) = 1\}$.

The minimization problem of NFA attracted much attention from the academic community, for example [19, 20, 21, 22, 23, 24, 25]. It has been proved that the minimization problem of NFA is PSPACE-complete [21]. On can refer to a survey of finite automata [49] for learning more information. Therefore, the minimization problem of fuzzy automata is at least as hard as PSPACE-complete.

6. Conclusions

In this paper we have proved for the first time that the minimization problem of fuzzy automata over totally order lattices is decidable. This result depends heavily on, the solvability of a system of fuzzy polynomial equations, a result established in this paper. Two problems worthy of further consideration are listed as follows:

- Are there more efficient algorithms for the minimization problem? Although the problem is known to be at least as hard as PSPACE-complete, it is still interesting to search for more efficient algorithms for the problem as in the case of NFAs.
- In this paper we have considered only minimizing fuzzy automata over totally ordered lattices. In the further study, one can consider the problem for more general cases, for example, for fuzzy automata over complete residuated lattices.

Acknowledgments

This work is supported in part by the National Natural Science Foundation (Nos. 61272058, 61073054, 60873055, 61100001), the Natural Science Foundation of Guangdong Province of China (No. 10251027501000004), the Fundamental Research Funds for the Central Universities (No. 11lgpy36), the Specialized Research Fund for the Doctoral Program of Higher Education of China (Nos. 20100171110042, 20100171120051).

References

[1] W.G. Wee, "On generalizations of adaptive algorithm and application of the fuzzy sets concept to pattern classification," Ph.D. Thesis, Purdue University, West lafayett, Indiana, USA, June 1967.

[2] E.S. Santos, "Maximin automata," *Inform. Control*, vol. 12, pp. 367-377, 1968.

[3] E.S. Santos, “On reduction of max-min machines,” *J. Math. Anal. Appl.*, vol. 37, pp. 677-686, 1972.

[4] E. T. Lee and L. A. Zadeh, “Note on fuzzy languages,” *Inform. Sci.* , vol. 1, pp. 421-434, 1969.

[5] M.G. Thomason and P.N. Marinos, “Deterministic acceptors of regular fuzzy languages,” *IEEE Trans. Syst., Man, Cybern.*, vol. 4, pp. 228-230, 1974.

[6] D. Dubois and H. Prade, “Fuzzy Sets and Systems: Theory and Applications,” New York, USA: Academic Press, 1980.

[7] M. M. Gupta, G. N. Saridis, and B. R. Gaines, “Fuzzy Automata and Decision Processes,” New York, USA: North-Holland, 1977.

[8] A. Kandel and S.C. Lee, “Fuzzy Switching and Automata: Theory and Applications,” New York, USA: Crane Russak, 1997.

[9] D.S. Malik and J.N. Mordeson, “Fuzzy Discrete Structures,” New York, USA: Physica-Verlag, 2000.

[10] J.N. Morderson, D.S. Malik, “Fuzzy Automata and Languages: Theory and Applications,” London, Boca Raton, FL: Chapman & Hall, CRC, 2002.

[11] D.W. Qiu, “Characterizations of fuzzy finite automata,” *Fuzzy Sets Syst.*, vol. 141, pp. 391-414, 2004.

[12] W. Wechler, “The Concept of Fuzziness in Automata and Language Theory,” Berlin, Germany: Akademie-Verlag, 1978.

[13] W.G. Wee and K.S. Fu, “A formulation of fuzzy automata and its application as a model of learning systems,” *IEEE Trans. Syst., Man, Cybern.*, vol. 5, pp. 215-223, 1969.

[14] F. Lin and H. Ying, “Modeling and control of fuzzy discrete event systems,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.* vol. 32 , pp. 408-415, 2002.

[15] D.W. Qiu, “Supervisory control of fuzzy discrete event systems: a formal approach,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.* vol. 35, pp. 72-88, 2005.

[16] C.L. Giles, C.W. Omlin, and K.K. Thornber, “Equivalence in knowledge representation: automata, recurrent neural networks, and dynamical fuzzy systems,” *Proceedings of IEEE* , vol. 87, 1623-1640, 1999.

[17] C.W. Omlin, K.K. Thornber, and C.L. Giles, “Fuzzy finite-state automata can be deterministically encoded in recurrent neural networks,” *IEEE Trans. Fuzzy Syst.* , vol. 5, pp. 76-89, 1998.

[18] J. E. Hopcroft, “An $n \log n$ algorithm for minimizing the state in a finite automaton,” in *The Theory of Machines and Computations*, USA: Academic Press, 1971, pp. 189-196.

[19] T. Kameda and P. Weiner, “On the State Minimization of Nondeterministic Finite Automata,” *IEEE Trans. Comput.*, vol. C-19, no.7, pp. 617-627, July 1970.

[20] H. Sengoku, “Minimization of nondeterministic finite automata,” Master thesis, Kyoto University, Japan, 1992.

[21] T. Jiang and B. Ravikumar, “Minimal NFAproblems are hard,” *SIAM J. Comput.*, vol. 22, pp. 1117-1141, 1993.

[22] B. F. Melnikov, “A new algorithm of the state-minimization for the nondeterministic finite automata,” *Korean J. Comput. & Appl. Math.*, vol.6, no. 2, pp. 277- 290, 1999.

[23] B. F. Melnikov, “Once more about the state-minimization of the nondeterministic finite automata,” *Korean J. Comput. & Appl. Math.*, vol.7, no.3, pp. 655-662, 2000.

[24] L. Polak, “Minimalizations of NFA using the universal automaton,” *Int. J. Found. Comput. Sci.*, vo.16, no. 5, pp. 999-1010, 2005.

[25] J.-M. Champarnaud and F. Coulon, “Theoretical study and implementation of the canonical automaton,” *Fund. Inform.*, vol. 55, pp. 23-38, 2003.

[26] L. Ilie and S. Yu, “Algorithms for computing small NFAs,” in *Proceedings of 27th International Symposium on Mathematical Foundations of Computer Science*, Lecture Notes in Comput. Sci., vol. 2420, pp. 328-340, 2002.

[27] L. Ilie and S. Yu, “Reducing NFAs by invariant equivalences,” *Theoret. Comput. Sci.* , vol. 306, pp. 373-390, 2003.

[28] L. Ilie and S. Yu, “Follow automata,” *Inform. Comput.*, vol. 186, 140-62, 2003.

[29] L. Ilie, G. Navarro, and S. Yu, “On NFA reductions,” in *Theory is Forever*, Lecture Notes in Comput. Sci., vol. 3113, pp. 112-24, 2004.

[30] L. Ilie, R. Solis-Oba, S. Yu, “Reducing the size of NFAs by using equivalences and preorders, in *Proceedings of 16th Annual Symposium on Combinatorial Pattern Matching*, Lecture Notes in Comput. Sci., vol. 3537, pp. 310-21, 2005.

[31] C. Campeanu, N. Santean, and S. Yu, “Mergible states in large NFA,” *Theoret. Comput. Sci.* , vol. 330, pp. 23-34, 2005.

[32] J.-M. Champarnaud and F. Coulon, “NFA reduction algorithms by means of regular inequalities,” *Theoret. Comput. Sci.*, vol. 327, pp. 241-253, 2004.

[33] C. S. Calude, E. Calude, and B. Khoussainov, "Finite nondeterministic automata: Simulation and minimality," *Theoret. Comput. Sci.*, vol. 242, pp. 219-235, 2000.

[34] K. Peeva, "Equivalence, reduction and minimization of finite automatao versemirings," *Theoret. Comput. Sci.* , vol. 88, pp. 269-285, 1991.

[35] D.S. Malik, J.N. Mordeson, and M.K. Sen, "Minimization of fuzzy finite automata," *Inform. Sci.*, vol. 113, pp. 323-330, 1999.

[36] N.C. Basak, A. Gupta, "On quotient machines of a fuzzy automaton and the minimal machine," *Fuzzy Sets Syst.*, vo. 125, pp. 223-229, 2002.

[37] W. Cheng and Z. Mo, "Minimization algorithm of fuzzy finite automata," *Fuzzy Sets Syst.*, vol. 141, pp. 439-448, 2004.

[38] T. Petković, "Congruences and homomorphisms of fuzzy automata," *Fuzzy Sets Syst.*, vol. 157, pp. 444-458, 2006.

[39] H. Lei and Y. M. Li, "Minimization of states in automata theory based on finite lattice-ordered monoids," *Inform. Sci.*, vol. 177, pp. 1413-1421, 2007.

[40] L. H. Wu and D. W. Qiu, "Automata theory based on complete residuated lattice-valued logic: Reduction and minimization," *Fuzzy Sets Syst.*, vol. 161, pp. 1635-1656, 2010.

[41] M. Ćirić, A. Stamenković, J. Ignjatović, and T. Petković, "Fuzzy relation equations and reduction of fuzzy automata," *J. Comput. Syst. Sci.*, vol. 76, pp. 609-633, 2010.

[42] D.W. Qiu, "Automata theory based on completed residuated lattice-valued logic (I)," *Sci. China Ser. F* , vol. 44, no. 6, pp. 419-429, 2001.

[43] D.W. Qiu, "Automata theory based on completed residuated lattice-valued logic (II)," *Sci. China Ser. F*, vol. 45, no. 6, pp. 442-452, 2002.

[44] K. Peeva, "Fuzzy linear systems," *Fuzzy Sets and Syst.* , vol. 49, no. 3, pp. 339-355, 1992.

[45] J. Renegar, "A faster PSPACE algorithm for deciding the existential theory of the reals," *in Proceedings of the 29th Annual Symposium on Foundations of Computer Science*, IEEE Computer Society Press, 1988, pp. 291-295.

[46] J. Canny, "Some algebraic and geometric computations in PSPACE," in *Proceedings of the twentieth annual ACM Symposium on Theory of Computing*, New York, USA, 1988, pp. 460–469.

[47] K. Etessami and M. Yannakakis, "Recursive Markov Chains, Stochastic Grammars, and Monotone Systems of Nonlinear Equations," *J. ACM*, vol. 56, no. 1, pp. 1-66, 2009.

[48] P. Mateus, D. W. Qiu, and L. Z. Li , "On the complexity of minimizing probabilistic and quantum automata," *Inform. Comput.*, vol. 218, pp. 36-53, 2012.

[49] M. Holzer, and M. Kutrib, "Descriptional and computational complexity of finite automata-A survey," *Inform. Comput.*, vol. 209, pp. 456-470, 2011.