Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT

Fuzzy Sets and Systems 141 (2004) 415–437

FUZZY sets and systems

www.elsevier.com/locate/fss

Finite L-fuzzy machines

Ketty Peeva

Faculty of Applied Mathematics and Informatics, Technical University of Sofia, P.O.Box 384, Sofia 1000, Bulgaria

Received 9 November 2001; received in revised form 20 November 2002; accepted 13 December 2002

Abstract

For finite L-fuzzy machines we develop behaviour, reduction and minimization theory. Here L stands for $(L, \vee, \wedge, 0, 1)$, where L is a totally ordered set with universal bounds 0 and 1 and the operations are join $\vee$ and meet $\wedge$. The most essential results include investigating behaviour, equivalence, reduction and minimization problems and their algorithmical decidability.

Keywords: Finite fuzzy machine; Equivalence; Reduction and minimization of finite fuzzy machines

1. Introduction

There exist quite a few papers and monographs covering equivalence, reduction and minimization problems for various kinds of sequential machines:

- For deterministic machines these problems are completely solved [11,16,51].
- For nondeterministic machines investigations are published in [5,8,9,16,51].
- Incompletely specified machines are studied in [6,12,14,25,32,41,42,43].
- For stochastic machines the problems are investigated in [10,11,17,30,31,35,51].
- Fuzzy machines are subject of [21,23,24,33,35,37,40,45–48,52].
- A general algebraic approach to these problems is presented in [16,22,32,35].
- Categorial approach is given in [2–4,34].

Fuzzy machines were first proposed by Santos [45,46] and studied by him in [45–48], by Santos and Wee [49], by Mizumoto and co-authors [28]. During the next two decades Adamek and Wechler [1], Brunner and Wechler [7], Wechler [53,54], Arbib and Manes [3,4], Mizumoto [26], Mizumoto and co-authors [27,28], Peeva [33,35], Peeva and co-authors [40,52] were active in this area. At this time two monographs have been published: by Wechler [54], and by Kandel and Lee [19].

E-mail address: kgp@vmei.acad.bg (K. Peeva).

0165-0114/03/$ - see front matter
doi:10.1016/S0165-0114(02)00609-7

Now fuzzy sequential machines are covered in the monograph [21, Chapter 12, pp. 349–353] by Klir and Bo Yuan, (completed with suitable cross-references and notes), and in the monograph [23, Chapters 5 and 6, pp. 139–246] by Malik and Mordeson.

As soon as the concept of fuzzy sequential machine had been introduced [45,46], Santos studied equivalence, reduction and minimization problems for finite max-product fuzzy machines [48] and for finite maximin fuzzy machines [47]. Various types of equivalence relations and minimal forms have been defined. Among the minimal forms considered are those which are similar to minimal state forms of stochastic machines, cf. [10]. Most of the results presented in [47,48] have counterparts in the theory of stochastic sequential machines [10,17,30,51].

In order to investigate finite max-product or maximin fuzzy machines a type of algebra, called the max-product (maximin, respectively) algebra, has been developed by Santos [47,48]. The role played by max-product (maximin, respectively) algebra in the theory of max-product (maximin, respectively) fuzzy machines is the same as that played by linear algebra in the theory of stochastic machines. But linear algebra, max-product algebra and maximin algebra are completely unrelated.

There are several issues in [47], which are either incomplete or confused. For investigating maximin fuzzy machines by the use of maximin algebra, Santos [47, Proposition 2.2], expresses a correct statement with confused proof: if the vector $B=(b_j)$ is a convex maximin combination of the vectors $X_1,\dots,X_n$:

$$B=(a_1\wedge X_1)\vee\cdots\vee(a_n\wedge X_n) \tag{1.1}$$

and the number of the distinct elements b_j in B is finite, then B is representable as a convex maximin combination of the form

$$B=(b_1\wedge X_1)\vee\cdots\vee(b_n\wedge X_n), \tag{1.2}$$

where b_i, $i=1,\dots,n$ are among the elements of B. In the proof of Proposition 2.2, b_i is chosen to be the largest number in B such that $b_i\leqslant a_i$. Let us consider an example. For the convex maximin combination

$$\left[0.9\wedge\begin{pmatrix}0.9\\0\\0.2\end{pmatrix}\right]\vee\left[0.7\wedge\begin{pmatrix}0.4\\0.8\\0.5\end{pmatrix}\right]\vee\left[0.2\wedge\begin{pmatrix}0.2\\0.5\\0.1\end{pmatrix}\right]=\begin{pmatrix}0.9\\0.7\\0.5\end{pmatrix},$$

following the proof of Proposition 2.2 from [47], we obtain $b_1=0.9=a_1$, $b_2=0.7=a_2$, but for $a_3=0.2$ we do not have choice because 0.2 is smaller than any of 0.9, 0.7, 0.5.

If we take b_i among the elements of B, according to formulation of Proposition 2.2, we have to check all selections with repetitions b^m of order m from b elements, where b is the number of the different entries in B, m is the size (number of components) of B. This requires exponential time, and the procedure to obtain (1.2) from (1.1) will be not effective.

Formulation of Proposition 2.3 in [47] is: it is decidable whether or not a vector B is a convex maximin combination of given vectors (with finite number of components). The proof of Proposition 2.3 follows from Proposition 2.2. Again any algorithm, finding a selection as a solution, will have exponential time complexity. Hence all constructions in [47, Propositions 2.3, 2.9, 3.4, Theorems 4.5 and 4.7], based on Proposition 2.2, will not be effective. Those constructions and statements concern decidability whether or not a vector is a convex maximin combination, obtaining the behaviour matrix, establishing equivalence, and finding reduced and minimal forms.

This motivated the author to develop a theory for behaviour, reduction and minimization for finite L-fuzzy machines, with correct and complete solutions for the above-mentioned problems. In this paper we introduce and investigate finite L-fuzzy machines. They are a generalization of the Santos' max–min fuzzy machines [47]. We clarify incomplete or confused issues from [47] and develop theory for behaviour, reduction and minimization that is valid in particular for maximin machines. In order to investigate finite L-fuzzy machines, suitable fuzzy relation algebra methods and polynomial time algorithms are developed in [36,38,39] for direct and for inverse problem resolution in fuzzy relation calculus. They are presented in Section 2. The direct problem resolution (Section 2) provides computing the behaviour of finite L-fuzzy machines (Section 3). The inverse problem resolution, given in Section 2, provides theoretical and algorithmical solvability for equivalence (Section 4), reduction and minimization (Section 5) of finite L-fuzzy machines. Thus, by the use of fuzzy relation algebra, complete solutions for behaviour, reduction and minimization problems of finite L-fuzzy machines are obtained.

The most essential results include: computing the behaviour matrix and investigating algorithmical decidability of the behaviour, equivalence, reduction and minimization problems, establishing various equivalences (equivalence of states, equivalence of machines, covering of machines, etc.) and solving reduction and minimization problems (reduction of the number of states, finding reduced machine, minimization with respect to the number of states, finding minimal machine, etc.).

2. Direct and inverse problem resolution in fuzzy relation algebra

The basic notions in this section are given according to [13,20,21,44].

We use the symbol $\leqslant$ for the partial order relation on a partially ordered set (poset) P. By a *greatest element* of a poset P we mean an element $b\in P$ such that $x\leqslant b$ for all $x\in P$; the *least element* of P being defined dually. The (unique) least and greatest elements of P, when they exist, are called *universal bounds* of P and are denoted by 0 and 1, respectively.

2.1. Definition. (i) A *lattice* L is a poset L any two of whose elements x and y have a *greatest lower bound* (g.l.b.) or *meet* denoted by $x\wedge y$, and a *least upper bound* (l.u.b.) or *join* denoted by $x\vee y$.

(ii) A totally ordered poset L is called a *chain*.

(iii) A chain with universal bounds 0 and 1 is called *bounded chain*.

Notation. L stands for the bounded chain $\mathsf{L}=(L,\vee,\wedge,0,1)$, where L is a totally ordered poset with universal bounds 0 and 1 and with operations join $\vee$ and meet $\wedge$. In particular the real closed interval [0, 1] with operations $\vee=\max$ and $\wedge=\min$ is a bounded chain $\mathsf{I}=([0,1],\max,\min,0,1)$, that is very often used in references.

In the bounded chain $\mathsf{L}=(L,\vee,\wedge,0,1)$ the operation α is defined as follows:

$$a\alpha b=\begin{cases}1, & a\leqslant b,\\ b, & a>b.\end{cases} \tag{2.1}$$

Fuzzy relation compositions. Let X and Y be crisp sets. A (binary) fuzzy relation $R \subseteq X \times Y$ is defined as a fuzzy subset of the Cartesian product $X \times Y$:

$$R = \{((x, y), \mu_R(x, y)) \mid (x, y) \in X \times Y,\ \mu_R : X \times Y \to [0, 1]\}.$$

An L-fuzzy relation R, written $R \in L(X \times Y)$, is defined as an L-fuzzy subset of $X \times Y$:

$$R = \{((x, y), \mu_R(x, y)) \mid (x, y) \in X \times Y, \mu_R : X \times Y \to L\}.$$

The *inverse* [15,21,44] of $R \in L(X \times Y)$, denoted by R^{-1}, is a relation on $Y \times X$, defined as

$$R^{-1}(y,x) = R(x, y) \text{ for all pairs } (y,x) \in Y \times X.$$

The L-fuzzy relations $R \in L(X \times Y)$ and $S \in L(Y \times Z)$, with $pr_2(X \times Y) = pr_1(Y \times Z) = Y$, are called *composable*.

Following De Baets [13], we define three kinds of binary operations for composable L-fuzzy relations:

- *round composition* $R\,\Theta\, S$,

$$R\Theta S(x,z) = \sup_{y \in Y} C(R(x, y), S(y,z)), \tag{2.2}$$

- *subcomposition* $R \lhd S$,

$$R \lhd S(x,z) = \inf_{y \in Y} I(R(x, y), S(y,z)), \tag{2.3}$$

- *supercomposition* $R \rhd S$,

$$R \rhd S(x,z) = \inf_{y \in Y} I(S(y,z), R(x, y)). \tag{2.4}$$

Here C is conjunctor, I is implicator, as defined in [13].

If $\mathsf{L} = (L, \vee, \wedge, 0, 1)$, the *standard composition* [13] of relations, symbolized here by $\bullet$, is a kind of the round composition (2.2):

$$R \bullet S(x,z) = \bigvee_{y \in Y} (\wedge(R(x, y), S(y,z))). \tag{2.5}$$

In particular (2.5) for $\mathsf{I} = ([0, 1], \max, \min, 0, 1)$ has the form:

$$R \bullet S(x,z) = \max_{y \in Y}(\min(R(x, y), S(y,z))). \tag{2.6}$$

2.2. Theorem (Sanchez [44]). *Let $R \in L(X \times Y)$ and $T \in L(X \times Z)$ be L-fuzzy relations, $\mathsf{Q}_\bullet$ be the set of all L-fuzzy relations $Q \in L(Y \times Z)$ with $R \bullet Q = T$. Then*:

(i) $\mathsf{Q}_\bullet \neq \emptyset$ *iff* $R^{-1}\alpha T \in \mathsf{Q}_\bullet$;
(ii) *if* $\mathsf{Q}_\bullet \neq \emptyset$, *then* $R^{-1}\alpha T$ *is the greatest element in* $\mathsf{Q}_\bullet$.

Here $R^{-1}\alpha T(y,z) = \wedge_{x \in X} (R^{-1}(y,x)\alpha T(x,z))$ [44] and the operation α is according to (2.1).

Types of problems in fuzzy relation calculus. The following types of problems are among the main problems of interest for the fuzzy relation calculus [13]:

(i) *Direct problem resolution*: computing $R\Theta S$, $R \lhd S$ or $R \rhd S$, as defined in (2.2), (2.3) and (2.4), respectively, if the fuzzy relations R and S are given;
(ii) *Inverse problem resolution* means solving equation of the type $R\Theta S = T$, $R \lhd S = T$ or $R \rhd S = T$, in the unknown fuzzy relation S, if R and T are given.

Operations on binary fuzzy relations and their matrix representation. $\mathsf{L} = (L, \vee, \wedge, 0, 1)$ is a bounded chain, cf. Definition 2.1.

2.3. Definition. $A = (a_{ij})_{m\times n}$, with $a_{ij} \in L$ for each i, j, is called a *matrix over* L.

We can present the L-fuzzy relation $R \in L(X \times Y)$ by a matrix over L.

When R is represented by a matrix $R = (r_{ij})$, the matrix representation of R^{-1} is the matrix $R^{-1} = (r_{ji})$, called the *transpose* or *inverse* [15,21] of the given matrix. In this sense $R^{-1} = R^t = (r_{ji})$.

We can describe the compositions of L-fuzzy relations by corresponding matrix products. Using this fact, we consider only the matrix products, instead of compositions of L-fuzzy relations. Often we shall denote by the same letter the fuzzy relation and its representative matrix.

Two matrices $A = (a_{ij})_{m\times p}$ and $B = (b_{ij})_{p\times n}$ are called *conformable* [50], if the number of the columns in $A = (a_{ij})_{m\times p}$ equals the number of the rows in $B = (b_{ij})_{p\times n}$. This means that the product of $A = (a_{ij})_{m\times p}$ and $B = (b_{ij})_{p\times n}$, in this order, makes sense.

2.4. Definition. Let $A = (a_{ij})_{m\times p}$ and $B = (b_{ij})_{p\times n}$ be given finite conformable matrices over L.

(i) The matrix $C_{m\times n} = (c_{ij}) = A \bullet B$ is called *standard product* of A and B if

$$c_{ij} = \bigvee_{k=1}^{p} (a_{ik} \wedge b_{kj}) \text{ when } 1 \leqslant i \leqslant m,\ 1 \leqslant j \leqslant n. \tag{2.7}$$

(ii) The matrix $C_{m\times n} = (c_{ij}) = A\,\alpha\,B$ is called *α-product* of A and B if

$$c_{ij} = \bigwedge_{k=1}^{p} (a_{ik} \alpha b_{kj}) \text{ when } 1 \leqslant i \leqslant m,\ 1 \leqslant j \leqslant n. \tag{2.8}$$

Definition 2.4(i) gives the matrix representation of the standard composition of L-fuzzy relations (2.5). In particular for $\mathsf{I} = ([0,1], \max, \min, 0, 1)$ the matrix representation of (2.6) is called max–min product and (2.7) has the form

$$c_{ij} = \max_{k=1}^{p}(\min(a_{ik}, b_{kj})) \text{ when } 1 \leqslant i \leqslant m,\ 1 \leqslant j \leqslant n. \tag{2.9}$$

The direct problem is solvable in polynomial time. Computing the max–min product and α-product are among the options of the software for direct problem resolution for various matrix operations, described in [38].

2.5. Theorem. *Let* $A=(a_{ij})_{m\times p}$ *and* $C=(c_{ij})_{m\times n}$ *be matrices and* $B_\bullet$ *is the set of matrices such that* $A\bullet B=C$. *Then*:

(i) $B_\bullet\neq\emptyset$ *iff* $A^t\alpha C\in B_\bullet$.
(ii) *if* $B_\bullet\neq\emptyset$, *then* $A^t\alpha C$ *is the greatest element in* $B_\bullet$.

The proof follows from the relationship between relations and matrices and from Theorem 2.2.

Fuzzy linear systems of equations $A\bullet X=B$**.** In this subsection we give an extraction of the author's results from [36].

Let $\mathsf{L}=(L,\vee,\wedge,0,1)$ be a bounded chain, cf. Definition 2.1. Let $A=(a_{ij})_{m\times n}$, $X=(x_j)_{n\times 1}$, $B=(b_i)_{m\times 1}$ be finite matrices over L, cf. Definition 2.3.

We consider a fuzzy linear system of equations

$$\left|\begin{array}{ccccc}(a_{11}\wedge x_1) & \vee\cdots & (a_{1n}\wedge x_n) & = & b_1\\ \cdots & \cdots & \cdots & \cdots & \cdots\cdots\end{array}\right. \Leftrightarrow \quad A\bullet X=B, \tag{2.10}$$

$$\left|\begin{array}{ccccc}(a_{m1}\wedge x_1) & \vee\cdots & (a_{mn}\wedge x_n) & = & b_m\end{array}\right.$$

where $A=(a_{ij})_{m\times n}$ is the matrix of the coefficients, $X=(x_j)_{n\times 1}$ is the matrix of the unknowns and $B=(b_i)_{m\times 1}$ stands for the second member of the system.

In [36] we propose an algorithm for solving (2.10). Its computer implementation is given in [39]. In order to make the exposition clear we recall some of the results (without proofs) from [36]. They are implemented in the next sections when investigating finite L-fuzzy machines.

The terminology for algorithmical decidability is according to [18].

The matrix $X^0=(x_j^0)_{n\times 1}$ is called a *point solution* of $A\bullet X=B$ if $A\bullet X^0=B$ holds.

The set of all point solutions of $A\bullet X=B$ is denoted by X^0. If $\mathsf{X}^0\neq\emptyset$ then $A\bullet X=B$ is called *consistent*, otherwise it is *inconsistent*.

A point solution $X^0_{\mathrm{low}}\in\mathsf{X}^0$ is called a *lower point solution* of $A\bullet X=B$ if for any $X^0\in\mathsf{X}^0$ the relation $X^0\leqslant X^0_{\mathrm{low}}$ implies $X^0=X^0_{\mathrm{low}}$, where $\leqslant$ denotes the partial order, induced in X^0 by the order of L. Dually, a point solution $X^0_u\in\mathsf{X}^0$ is called an *upper point solution* of $A\bullet X=B$ if for any $X^0\in\mathsf{X}^0$ the relation $X^0_u\leqslant X^0$ implies $X^0=X^0_u$.

An interval $X_j\subseteq L$ is called *feasible* for the jth component of the solution, if the choice of any $x_j\in X_j$ results in $a_{ij}\wedge x_j\leqslant b_i$ for each $i=1,\dots,m$. An n-tuple $(X_1,\dots,X_n)$ of feasible intervals $X_j\subseteq L$, $1\leqslant j\leqslant n$ is called an *interval solution* of the system $A\bullet X=B$ if any vector $X^0=(x_j^0)_{n\times 1}$ with components $x_j^0\in X_j$, $1\leqslant j\leqslant n$, belongs to X^0, i.e. $X^0=(x_j^0)_{n\times 1}\in\mathsf{X}^0$. Any interval solution, that is maximal with respect to this feasibility property, is called a *maximal interval solution*. The interval components of any maximal interval solution are stretched on a lower point solution from the left and an upper point solution from the right [20,36].

Two fuzzy linear systems are called *equivalent*, if each solution of the first one is a solution of the second one and vice versa.

We assign to the system $A\bullet X=B$ a new system $A^*\bullet X=B$ with a coefficient matrix $A^*=(a^*_{ij})$ carried out from A with respect to B according to (2.11):

$$a^*_{ij}=\begin{cases}0 & \text{if } a_{ij}<b_i,\\ b_i & \text{if } a_{ij}=b_i,\\ 1 & \text{if } a_{ij}>b_i.\end{cases} \tag{2.11}$$

A coefficient $a^*_{ij}=0$ is called S-type coefficient (from smaller, because $a_{ij}<b_i$), a coefficient $a^*_{ij}=b_i$ is called E-type coefficient (from equal, because $a_{ij}=b_i$), a coefficient $a^*_{ij}=1$ is called G-type coefficient (from greater, because $a_{ij}>b_i$).

The system $A^*\bullet X=B$ is said to be *associated* to $A\bullet X=B$. Any fuzzy linear system is equivalent with its associated system. The *augmented matrix* for $A^*\bullet X=B$ is denoted by $(A^*:B)$.

Remark. We suppose that the conventions given below are valid for the system $A\bullet X=B$:

(i) $b_1\geqslant b_2\geqslant\cdots\geqslant b_n$ is satisfied for the second member of the system $A\bullet X=B$;
(ii) the associated system $A^*\bullet X=B$ is obtained;
(iii) for any column j, $1\leqslant j\leqslant n$, we denote by k the greatest number of the row with G-type coefficient in it and by r the smallest number of the row with E-type coefficient in it.

2.6. Theorem. *Let the system $A\bullet X=B$ be given.*
(i) *If the jth column of A^* contains a G-type coefficient $a^*_{kj}=G$, then*

  (a) *$X_j=[0,b_k]$ is a feasible interval for the jth component*;
  (b) *$x_j=b_k$ implies $a_{ij}\wedge x_j=b_i$ for*:
   - *$i=k$*;
   - *for each $i<k$ with $a^*_{ij}\geqslant b_i=b_k$*;
   - *for each $i>k$ with $a^*_{ij}=b_i$.*

(ii) *If the jth column of A^* does not contain any G-type coefficient, but it contains an E-type coefficient $a^*_{rj}=b_r$, then*
  (a) *$X_j=L$ is the feasible interval for the jth component*;
  (b) *$x_j\in[b_r,1]$ implies $a_{ij}\wedge x_j=b_i$ for each $i\geqslant r$ with $a^*_{ij}=b_i$.*

(iii) *If the jth column of A^* contains neither a G-type nor an E-type coefficient, then the feasible interval is $X_j=L$ and $a_{ij}\wedge x_j<b_i$ holds for any $x_j\in L$.*

We denote by $\mathtt{G}$ and $\mathtt{E}$ all G and E coefficients, respectively selected to satisfy the ith equation, $1\leqslant i\leqslant n$, by the term $a_{ij}\wedge x_j=b_i$ according to Theorem 2.6(i)(b), (ii)(b). Those $\mathtt{G}$, $\mathtt{E}$ coefficients are called *selected.*

From Theorem 2.6 we obtain the following properties:

(1) Let the system $A\bullet X=B$ be given.

  (i) It is consistent iff there exists at least one selected coefficient $a^*_{ij}\in\{\mathtt{G},\mathtt{E}\}$ for each i, $1\leqslant i\leqslant m$.
  (ii) It is inconsistent, if there exists an equation with no selected coefficient. Then this equation is in contradiction with the other equations.
  (iii) The time complexity function for establishing the consistency of the system is $O(mn)$.

(2) Let the system $A\bullet X=B$ be consistent. Then:

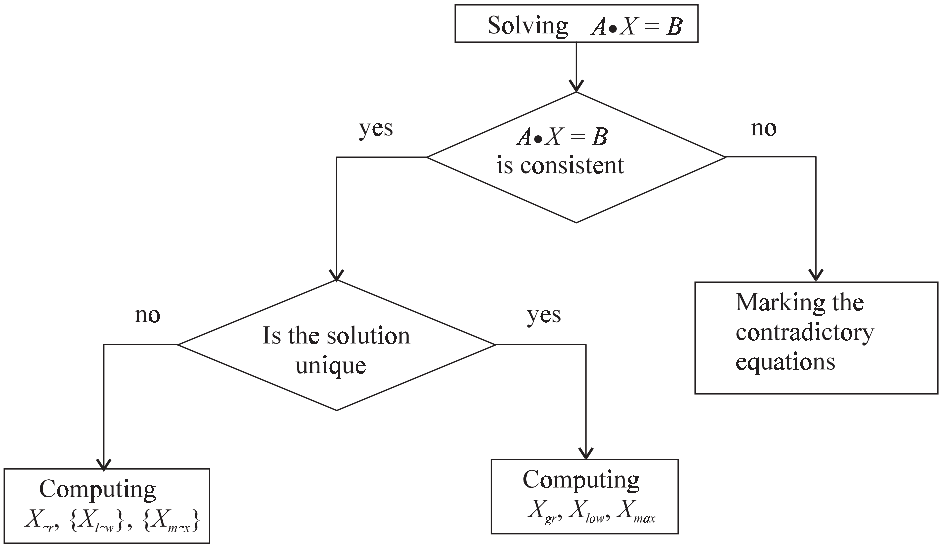

Fig. 1. Principal steps in solving the fuzzy linear system $A \bullet X = B$.

(i) It has a unique greatest solution $X_{\mathrm{gr}} = (x_j)_{n\times 1}$, where for $j = 1, \ldots, n$ the component $x_j = b_k$ if the jth column of A^* contains G type coefficient and $x_j = 1$ otherwise.
(ii) The time complexity function for computing X_{gr} is $\mathrm{O}(mn)$.

For computing X_{gr} we may also use the formula $X_{\mathrm{gr}} = A^t \alpha B$ according to Theorem 2.5.

If the system $A \bullet X = B$ is consistent, the lower solutions are computable and the set of all lower solutions is finite. The details for finding lower solutions are given in [36]. We omit computing the lower solutions of the system, because it is not essential for the subject of the paper.

We introduce a *help matrix* $H = (h_{ij})_{m\times n}$ with elements

$$h_{ij} = \begin{cases} b_i & \text{if } a^*_{ij} = \mathsf{G} \text{ or } a^*_{ij} = \mathsf{E}, \\ 0 & \text{otherwise.} \end{cases}$$

For any i, $1 \leqslant i \leqslant n$, the elements $h_{ij} = b_i \neq 0$ in H mark different ways to satisfy the ith equation of the system.

The vector *IND* is used to indicate the consistency of the system.

Theorem 2.6 provides the next polynomial time algorithm for solving the system $A \bullet X = B$, cf. also Fig. 1.

2.7. Algorithm for solving the system $A \bullet X = B$.

(1) Enter the matrices $A_{m\times n}$, $B_{m\times 1}$.
(2) Compute the matrix $(A^* : B)$ with elements a^*_{ij} according to (2.11).
(3) Initialize the vector *IND* and the help matrix $H = (h_{ij})_{m\times n}$.
(4) For each $i = 1, \ldots, n$ put $X_{\mathrm{gr}}(i) := 1$ and $X_{\mathrm{low}}(i) := 0$.
(5) $j = 0$.

(6) $j = j + 1$.
(7) If $j > n$ go to 10.
(8) If the jth column in A^* does not contain any G-type coefficient, go to 9.
Otherwise take the greatest number k of the row with G-type coefficient in the jth column of A^*. Put:

- $X_{\rm gr}(j) := b_k$,
- $IND(k) := IND(k) + 1$ and $h_{kj} := b_i$;
- $IND(i) := IND(i) + 1$ for each $i < k$ if $a^*_{ij} \geqslant b_i = b_k$;
- $IND(i) := IND(i) + 1$ and $h_{ij} := b_i$ for each $i > k$ if $a^*_{ij} = b_i$.

Go to 6.

(9) If the jth column in A^* does not contain any E-type coefficient, go to 6.
Otherwise take the smallest number r of the row with E-type coefficient in the jth column of A^*. Put $IND(i) := IND(i) + 1$ and $h_{ij} := b_i$ for $i \geqslant r$ with $a^*_{ij} = b_i$.
Go to 6.
(10) If $IND(i) = 0$ for some $i = 1, \ldots, m$, then the system is inconsistent and the ith equation is in contradiction with the others.
Go to 13.
(11) Compute the lower point solutions according to Theorem 2.6 and [36].
(12) If $IND(i) = 1$ for each $i = 1, \ldots, m$ the system is consistent with a unique maximal interval solution stretched from $X_{\rm low}$ to $X_{\rm gr}$.
Go to 14.
(13) The system is consistent, $X_{\rm gr}$ contains the greatest point solution. The lower point solutions are according to step 11. Obtain the maximal interval solutions by the lower and the greatest solutions.
(14) End.

Hence, for the system (2.10):

(i) It is algorithmically decidable in polynomial time whether the system is consistent or not;
(ii) if the system is consistent: the greatest point solution, the lower point solutions and the maximal interval solutions are computable;
(iii) if the system is inconsistent we can determine the contradictory equations.

2.8. Definition. We shall say that the vector $B = (b_i)_{m\times 1}$ is a *convex linear combination* (CLC) of the vectors $A_1, \ldots, A_n$ with coefficients $x_j \in L$, $1 \leqslant j \leqslant n$, if

$$B = (A_1 \wedge x_1) \vee \cdots \vee (A_n \wedge x_n),$$

where $A_1 = (a_{i1})_{m\times 1}$, $A_2 = (a_{i2})_{m\times 1}, \ldots, A_n = (a_{in})_{m\times 1}$, i.e., $b_i = \vee_{1\leqslant j\leqslant n}(a_{ij} \wedge x_j)$ for each $1 \leqslant i \leqslant m$.

2.9. Corollary. *Let* $A_1 = (a_{i1})_{m\times 1}$, $A_2 = (a_{i2})_{m\times 1}, \ldots, A_n = (a_{in})_{m\times 1}$, $B_{m\times 1} = (b_i)_{m\times 1}$ *be given. It is algorithmically decidable whether* $B_{m\times 1}$ *is a CLC of* $A_1, \ldots, A_n$.

The proof follows from Theorem 2.6 and Algorithm 2.7.

Corollary 2.9 provides a precise answer of Proposition 2.3 from [47]. It is implemented in Section 3 (Theorems 3.4, 3.5) and in Section 4 (Theorem 4.4).

Let A and B be conformable matrices with $A \bullet B = C$, A and C are given matrices, B is unknown. Then $A \bullet B = C$ is called a *matrix equation*.

We can solve $A \bullet B = C$ for the unknown matrix B. In order to do this, we split B and C by columns and use the following equivalence:

$$A_{m\times n} \bullet B_{n\times p} = C_{m\times p} \Leftrightarrow \left| \begin{array}{c} A_{m\times n} \bullet B^{(1)} = C^{(1)} \\ A_{m\times n} \bullet B^{(2)} = C^{(2)} \\ \dots \\ A_{m\times n} \bullet B^{(p)} = C^{(p)} \end{array} \right. . \tag{2.12}$$

Here $B^{(j)}$, $j = 1,\dots,p$, stands for the jth column of B and $C^{(j)}$, $j = 1,\dots,p$, for the jth column of C. It means that instead of $A_{m\times n} \bullet B_{n\times p} = C_{m\times p}$ we solve with Algorithm 2.7 p fuzzy linear systems of the same type—they have the same matrix $A_{m\times n}$, taking for any equation the corresponding columns (with the same indices) of B and C.

We propose (cf. also [36,39]) the following:

2.10. Algorithm for solving the matrix equation $A_{m\times n} \bullet B_{n\times p} = C_{m\times p}$.

(1) Enter the matrices $A_{m\times n}$, $C_{m\times p}$.
(2) Solve the equivalent system (2.12) for B by Algorithm 2.7.
(3) If the system is consistent, form the solution for $B_{n\times p}$.
(4) End.

Algorithm 2.10 is used in Sections 4 and 5 (Theorems 4.5, 5.5 and Corollary 5.3).

Remark. If we are interested in greatest solution of $A \bullet B = C$, we apply Theorem 2.5. We can always compute $A^t \alpha C$. If $A \bullet (A^t \alpha C) = C$ is true, then $A^t \alpha C$ is the greatest solution of $A \bullet B = C$. Otherwise the matrix equation is inconsistent.

3. Behaviour of finite L-fuzzy machines

We define the basic notions for finite L-fuzzy machines in accordance with similar notions for deterministic, nondeterministic and stochastic machines [11,16,22,25,30,51].

3.1. Definition (Peeva [33]). A *finite* L-*fuzzy machine* is a quadruple $A = (X, Q, Y, M)$, where:

(i) X, Q, Y are finite nonempty sets of *input letters*, *states* and *output letters*, respectively;

(ii)

$$M = \{M(x \mid y) \mid M(x \mid y) = (m_{qq'}(x|y)), x \in X, y \in Y, q, q' \in Q, m_{qq'}(x \mid y) \in L\} \tag{3.1}$$

is the set of (the transition-output) matrices of A, that determines its *stepwise behaviour*.

$\mathsf{L} = (L, \vee, \wedge, 0, 1)$ is a bounded chain, cf. Definition 2.1.

In particular, for $\mathsf{I}=([0,1],\max,\min,0,1)$ we obtain max–min machines as defined by Santos [45,47].

We interpret $m_{qq'}(x \mid y)$ as the degree of membership that the finite L-fuzzy machine will enter state q' and produce output y given that the present state is q and the input is x.

We consider here finite L-fuzzy machines, denoted by FFMs.

In this section we study the input–output behaviour of FFMs. Computing this behaviour is provided by the direct problem resolution methods, given in Section 2. An infinite matrix describes the complete input–output behaviour of an FFM. Here the aim is to find the upper bound for a finite (behaviour) matrix, extracted from the complete behaviour matrix. This finite (behaviour) matrix is expected to have sufficiently good properties for solving equivalence, reduction and minimization problems.

The *free monoid of the words* over the input set X (output set Y) is denoted by X^* (by Y^*, respectively) with the *empty word* e as the identity element. The *length of the word* u is denoted by $|u|$. Obviously $|u| \in \mathsf{N}$. By definition $|e|=0$. If $X \neq \emptyset$ and $Y \neq \emptyset$ then X^*, or Y^*, respectively, are countably infinite.

We denote by $(X|Y)^* = \{(u|v) \mid u \in X^*, v \in Y^*, |u|=|v|\}$ the set of all input–output pairs of words of the same length. For $u \in X^*$, $v \in Y^*$, if $|u|=|v|$, we write $(u|v) \in (X|Y)^*$, to distinguish it from the case $(u,v) \in X^* \times Y^*$.

Let $A=(X,Q,Y,M)$ be an FFM. We extend its stepwise behaviour to the *extended input–output behaviour*: for any pair $(u \mid v) \in (X \mid Y)^*$ we multiply the corresponding stepwise behaviour matrices from the set M, cf. (3.1), and compute $M(u \mid v)$ as defined in Definition 2.4(i):

3.2. Definition. Let $A=(X,Q,Y,M)$ be an FFM. The *extended input–output behaviour of* A for $(u \mid v) \in (X \mid Y)^*$ is defined by the square matrix $M(u \mid v)$ of order $|Q|$:

$$M(u|v) = \begin{cases} M(x_1 \mid y_1) \bullet \cdots \bullet M(x_k \mid y_k) & \text{if } (u|v) = (x_1 \dots x_k | y_1 \dots y_k), \quad k \geqslant 1, \\ U & \text{if } (u \mid v) = (e \mid e) \end{cases} \tag{3.2}$$

where $U=(\delta_{ij})$ is the square unit matrix of order $|Q|$ with elements

$$\delta_{ij} = \begin{cases} 1 & \text{if } i=j, \\ 0 & \text{if } i \neq j. \end{cases}$$

The *interpretation* of the elements of $M(u \mid v)$ is as follows: each element $m_{qq'}(u \mid v)$ in $M(u \mid v)$ determines the operation of A under the input word u beginning at state q, producing the output word v and reaching the state q' after $|u|=|v|$ consecutive steps.

Example 3.1. Let the finite max–min fuzzy machine $A=(X,Q,Y,M)$ over $I=([0,1],\max,\min,0,1)$ be given with the following data:

$$X=\{x\}, \quad Y=\{y_1,y_2\}, \quad Q=\{q_1,q_2\},$$

$$M(x \mid y_1) = \begin{pmatrix} 0.1 & 0.5 \\ 0.2 & 0.3 \end{pmatrix}, \quad M(x \mid y_2) = \begin{pmatrix} 0 & 0.15 \\ 0.4 & 1 \end{pmatrix}.$$

Table 1
The matrix T

	$T(e \mid e)$	$T(x_1 \mid y_1)$	$\dots$	$T(u \mid v)$	$\dots$
q_1	1	$t_{q_1}(x_1 \mid y_1)$	$\dots$	$t_{q_1}(u \mid v)$	$\dots$
q_2	1	$t_{q_2}(x_1 \mid y_1)$	$\dots$	$t_{q_2}(u \mid v)$	$\dots$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
q_n	1	$t_{q_n}(x_1 \mid y_1)$	$\dots$	$t_{q_n}(u \mid v)$	$\dots$
length l	$l=0$	$l=1$	$\dots$	$l=\lvert u\rvert$	$\dots$

Its extended input–output behaviour for $(u \mid v)=(xx \mid y_1 y_2)$ according to Definition 3.2 and (3.2) is

$$M(u \mid v) = M(xx \mid y_1 y_2) = M(x \mid y_1) \bullet M(x \mid y_2) = \begin{pmatrix} 0.1 & 0.5 \\ 0.2 & 0.3 \end{pmatrix} \bullet \begin{pmatrix} 0 & 0.15 \\ 0.4 & 1 \end{pmatrix} = \begin{pmatrix} 0.4 & 0.5 \\ 0.3 & 0.3 \end{pmatrix}.$$

For instance, the element $m_{21}(xx \mid y_1 y_2)$ is computed as follows:

$$m_{21}(xx/y_1 y_2) = \max(\min(0.2, 0),\ \min(0.3, 0.4)) = 0.3.$$

The other elements of $M(u \mid v)=M(xx \mid y_1 y_2)$ are obtained similarly.

If we are not interested in the next state q', we consider the input–output behaviour of A:

3.3. Definition. The *input–output behaviour* of A for $(u \mid v) \in (X \mid Y)^*$, is defined by the column-matrix

$$T(u \mid v) = (t_q(u|v))_{|Q|\times 1} = \begin{cases} M(u|v) \bullet E, & \text{if } (u|v) \neq (e|e), \\ E, & \text{if } (u|v) = (e|e), \end{cases}$$

where $E=(1\ 1\ \dots\ 1)^t_{|Q|\times 1}$ is the column-matrix with all elements equal to 1. Here

$$t_q(u|v) = \bigvee_{q' \in Q} (m_{qq'}(u|v))$$

means that each element $t_q(u|v)$ of $T(u|v)$ determines the operation of A under the input word u beginning at state q and producing the output word v after $|u|=|v|$ consecutive steps. This is the way of achieving maximal degree of membership.

We use the column-matrices $T(u|v)$, $(u|v) \in (X|Y)^*$ to describe the complete input–output behaviour of A. To this aim we suppose that $(X|Y)^*$ is ordered lexicographically [23,47]. Let T be the semi-infinite matrix with $|Q|$ rows and with columns $T(u|v)$, ordered according to the lexicographical order in $(X|Y)^*$. The *complete behaviour matrix* T describes the *complete input–output behaviour* of A, Table 1.

Table 2
An initial fragment from T

	$T_{e/e}$	$T_{1/1}$	$T_{1/2}$	$T_{2/1}$	$T_{2/2}$	$T_{11/11}$	$T_{11/12}$	$T_{11/21}$	...
q_1	1	0.3	0.1	0.8	0.9	0.3	0.2	0.1	...
q_2	1	0.5	0.3	0.3	0.6	0.5	0.5	0.3	...
q_3	1	0.6	0.6	0.5	0.4	0.6	0.6	0.6	...
l	$l=0$	$l=1$				$l=2$			

Example 3.2. Let the finite max–min fuzzy machine $A=(X,Q,Y,M)$ over $\mathsf{I}=([0,1],\max,\min,0,1)$ be given with the following data:

$$X=\{x_1,x_2\},\quad Y=\{y_1,y_2\},\quad Q=\{q_1,q_2,q_3\},$$

$$M(x_1\mid y_1)=\begin{pmatrix}0.3&0&0.2\\0&0.1&0.5\\0.5&0&0.6\end{pmatrix},\quad M(x_1\mid y_2)=\begin{pmatrix}0&0&0.1\\0.2&0&0.3\\0.6&0&0.6\end{pmatrix},$$

$$M(x_2\mid y_1)=\begin{pmatrix}0.8&0&0\\0&0.3&0\\0.2&0.1&0.5\end{pmatrix},\quad M(x_2\mid y_2)=\begin{pmatrix}0.9&0.7&0\\0.1&0.6&0\\0.4&0.3&0.2\end{pmatrix}.$$

Let $T_{1/1}=T(x_1\mid y_1)$, $T_{1/2}=T(x_1\mid y_2)$, $T_{2/1}=T(x_2\mid y_1)$, $T_{2/2}=T(x_2\mid y_2)$.

We multiply the matrices according to Definition 2.4(ii) and (2.8) and we obtain (Table 2):

$$M(x_1x_1|y_1y_1)=\begin{pmatrix}0.3&0&0.2\\0.5&0.1&0.5\\0.5&0&0.6\end{pmatrix},\quad T_{11/11}=T(x_1x_1|y_1y_1)=\begin{pmatrix}0.3\\0.5\\0.6\end{pmatrix},$$

$$M(x_1x_1|y_1y_2)=\begin{pmatrix}0.2&0&0.2\\0.5&0&0.5\\0.6&0&0.6\end{pmatrix},\quad T_{11/12}=T(x_1x_1|y_1y_2)=\begin{pmatrix}0.2\\0.5\\0.6\end{pmatrix},$$

$$M(x_1x_1|y_2y_1)=\begin{pmatrix}0.1&0&0.1\\0.3&0&0.3\\0.5&0&0.6\end{pmatrix},\quad T_{11/21}=T(x_1x_1|y_2y_1)=\begin{pmatrix}0.1\\0.3\\0.6\end{pmatrix},$$

etc.

T has $|Q|$ rows and countably infinite number of columns that is essential for all algorithms using the matrix T: all of them will be infinite. This leads to the main problem: *does there exist a finite submatrix of T, providing finite investigations and polynomial time algorithms for solving equivalence, reduction and minimization problems*?

In order to answer this question we introduce some notations.

We denote by T_i the finite submatrix of T containing the columns $T(u\mid v)$ for the words of length not greater than i, $i\in\mathsf{N}$.

Let B_i be a submatrix of T_i obtained by omitting all columns from T_i that are CLC (max–min or $\vee$–$\wedge$ linear combination), cf. Definition 2.8, of the previous columns. Similarly, let B be the matrix obtained from T by omitting all columns from T that are CLC (max–min or $\vee$–$\wedge$ linear combination) of the previous columns. The matrix B is called the *behaviour matrix* for A.

For any FFM we can obtain the matrix B_i from T_i for any $i \in \mathsf{N}$, according to Corollary 2.9.

The next Theorem is essential for the equivalence, reduction and minimization.

3.4. Theorem. *Let A be an FFM with complete behaviour matrix T. If the $(k+1)$-column T_{k+1} in T is a CLC of the previous columns T_j, $1 \leqslant j \leqslant k$, i.e.*

$$T_{k+1} = (\alpha_1 \wedge T_1) \vee \cdots \vee (\alpha_k \wedge T_k), \tag{3.3}$$

then the following properties hold:

(i) *If the elements in the* sth *and pth rows of any column T_j, $1 \leqslant j \leqslant k$, are equal, then the sth and pth elements in the column $T_{k+1} = T_{k+1}(u \mid v)$ are also equal, i.e.*

$$t_{sj} = t_{pj}, \quad 1 \leqslant j \leqslant k \Rightarrow t_{s,k+1}(u \mid v) = t_{p,k+1}(u \mid v). \tag{3.4}$$

(ii) *If the sth row in the matrix T_k is a CLC of the other rows in T_k, i.e.*

$$t_{sj} = (\beta_1 \wedge t_{1j}) \vee (\beta_2 \wedge t_{2j}) \vee \cdots \vee (\beta_n \wedge t_{nj}), \quad 1 \leqslant j \leqslant k, \tag{3.5}$$

then the sth element in the column $T_{k+1} = T_{k+1}(u \mid v)$ is the same CLC of the elements from the column T_{k+1}, i.e.

$$t_{s,k+1}(u \mid v) = [\beta_1 \wedge t_{1,k+1}(u \mid v)] \vee [\beta_2 \wedge t_{2,k+1}(u \mid v)] \vee \cdots \vee [\beta_n \wedge t_{n,k+1}(u \mid v)].$$

Proof. (i) From (3.3) we have

$$t_{s,k+1}(u \mid v) = (\alpha_1 \wedge t_{s1}) \vee (\alpha_2 \wedge t_{s2}) \vee \cdots \vee (\alpha_k \wedge t_{sk}). \tag{3.6}$$

Since $t_{sj} = t_{pj}$, $1 \leqslant j \leqslant k$, cf. (3.4), we substitute t_{sj} with its equal t_{pj} in (3.6) and we obtain

$$t_{s,k+1}(u \mid v) = (\alpha_1 \wedge t_{p1}) \vee (\alpha_2 \wedge t_{p2}) \vee \cdots \vee (\alpha_k \wedge t_{pk}) = t_{p,k+1}(u \mid v).$$

(ii) Substituting t_{sj}, $1 \leqslant j \leqslant k$ according to (3.5) in (3.6), we have

$$\begin{aligned} t_{s,k+1}(u \mid v) = {} & (\alpha_1 \wedge [(\beta_1 \wedge t_{11}) \vee (\beta_{21} \wedge t_{21}) \vee \cdots \vee (\beta_n \wedge t_{n1})]) \\ & \vee \cdots \vee (\alpha_k \wedge [(\beta_1 \wedge t_{1k}) \vee (\beta_2 \wedge t_{21k}) \vee \cdots \vee (\beta_n \wedge t_{nk})]) \\ = {} & (\beta_1 \wedge [(\alpha_1 \wedge t_{11}) \vee (\alpha_2 \wedge t_{12}) \vee \cdots \vee (\alpha_k \wedge t_{1k})]) \\ & \vee \cdots \vee (\beta_n \wedge [(\alpha_1 \wedge t_{n1}) \vee (\alpha_2 \wedge t_{n2}) \vee \cdots \vee (\alpha_k \wedge t_{nk})]) \\ = {} & [\beta_1 \wedge t_{1,k+1}(u \mid v)] \vee [\beta_2 \wedge t_{2,k+1}(u \mid v)] \vee \cdots \vee [\beta_n \wedge t_{n,k+1}(u \mid v)]. \end{aligned}$$

Theorem 3.4 enables the behaviour matrix B to be extracted from T. For this aim we can remove from T:

- a column, which is a CLC of the previous columns;
- a row, which is a CLC of the previous rows.

In order to obtain a numerical evaluation of the time complexity function for computing the behaviour matrix B we introduce the following notations.

For arbitrary matrices C and D, we write $C \subseteq D$, if each column of C is a column of D. If each column in D is a CLC of the columns from C, we write $C \sim D$. Obviously for each $i \in \mathsf{N}$, we have

(i) $T_i \subseteq T_{i+1} \subseteq \cdots \subseteq T$,
(ii) $B_i \subseteq B_{i+1} \subseteq \cdots \subseteq B$,
(iii) $B_i \subseteq T_i$.

Let $A = (X, Q, Y, M)$ be given FFM. According to (3.1), we have

$$M = \{M(x \mid y) | M(x \mid y) = (m_{qq'}(x \mid y)), x \in X, y \in Y, q, q' \in Q, m_{qq'}(x \mid y) \in L\}.$$

We denote by DM the set of the (distinct) transition-output degrees of membership in M, i.e. the set of distinct entries for the matrices $M(x \mid y) \in M$:

$$DM = \{m_{qq'}(x \mid y) | x \in X, y \in Yq, q' \in Q\}.$$

The cardinality of DM is denoted by a, i.e. $a = |DM|$. Let us denote by DT the set of the (distinct) input–output degrees of membership in the matrix T:

$$DT = \{t_q(u \mid v) | (u \mid v) \in (X \mid Y)^*\}.$$

The cardinality of DT is denoted by b, i.e. $b = |DT|$. The elements of DT (excluding eventually 1) are a subset of the elements of DM because of the chain properties and Definition 2.4(i) of the standard matrix multiplication. It means that $b \leqslant a + 1$.

3.5. Theorem. *For any FFM the following statements hold*:

(i) *There exists an integer* $k \leqslant b^{|Q|} - 1$, *such that* $T_k \sim T_{k+1}$ *and* $B_k = B$.
(ii) *If* $T_i \sim T_{i+1}$, *then*:
  - $T_i \sim T_{i+p} \sim \cdots \sim T$ *for each* $p = 0, 1, 2, \ldots$,
  - $B_i = B_{i+p} = \cdots = B$ *for each* $p = 0, 1, 2, \ldots$.

(iii) $B \sim T$.

Proof. (i) follows from the properties of the operations in L and from Definition 2.4(i): for arbitrary $a, c \in L$, $a \vee c \in \{a, c\}$ and $a \wedge c \in \{a, c\}$, the number k equals the number of the selections with repetitions $b^{|Q|}$ of order $|Q|$ from b elements. The term -1 corresponds to the empty word and the column $T(e \mid e)$ in T.

Conditions (ii) and (iii) follow from the construction of the corresponding matrices.

In particular, if we take $\mathsf{I}=([0,1],\max,\min,0,1)$ in Theorem 3.5, we receive Santos' Propositions 3.2 and 3.3 for max–min fuzzy machines, published in [47].

3.6. Corollary. *For any finite nondeterministic machine we have*:

(i) *There exists an integer* $k \leqslant 2^{|Q|}-1$, *such that* $T_k \sim T_{k+1}$ *and* $B_k = B$.
(ii) *If* $T_i \sim T_{i+1}$, *then*:
- $T_i \sim T_{i+p} \sim \cdots \sim T$, *for each* $p=0,1,2,\ldots$,
- $B_i = B_{i+p} = \cdots = B$, *for each* $p=0,1,2,\ldots$.

(iii) $B \sim T$.

Proof. If in Theorem 3.5(i) we take $L=\{0,1\}$, then the machine is nondeterministic. Substituting $b=2$ in $k \leqslant b^{|Q|}-1$ we obtain $k \leqslant 2^{|Q|}-1$.

Corollary 3.6 gives a new proof of a well-known result for nondeterministic machines [8,9] as a partial case of fuzzy machines.

For solving the next questions for equivalence, reduction and minimization of FFMs the behaviour matrix is very important. According to Theorem 3.5, any algorithm for computing the behaviour matrix will contain the following steps:

3.7. Algorithm for computing the behaviour matrix B of an FFM $A=(X,Q,Y,M)$.
(1) Enter the set of matrices M due to (3.1), the cardinality $|Q|$ of the set of states Q and the number $b=|DT|$ of the different elements in the matrix T.
(2) Compute $l=b^{|Q|}-1$.
(3) Determine the smallest natural number $k \leqslant l$, such that $T_k \sim T_{k+1}$.
(4) Obtain $B_k = B$ from T_k according to Theorem 3.5.
(5) End.

The natural question now is: what is the time complexity of this algorithm. The answer is included in the next corollary.

3.8. Corollary. *For any FFM*:

(i) *Its behaviour matrix B is finite* [33];
(ii) *The time complexity function for determining* $T_k \sim T_{k+1}$ *and* $B_k = B$, *where* $k \leqslant b^{|Q|}-1$, *is exponential.*

Corollary 3.8 disproves Santos' Proposition 3.4 from [47] for the time complexity function for obtaining the behaviour matrix B from T.

3.9. Remark. This result means that depending on $k \leqslant b^{|Q|}-1$ in some cases we shall be able to solve the problems for equivalence, reduction and minimization, in other cases we shall prefer to solve those problems for the words of a fixed length not exceeding a natural number $l=1,2,\ldots,l<k$. There exist also situations where α-cut over the set of matrices $M(x \mid y) \in M$ makes sense—it will reduce the number a (b, respectively) of distinct membership degrees and so decrease the time to compute B.

Comments. A comparison between the behaviour matrices for FFMs and the behaviour matrices for other kinds of machines makes sense:

(1) The behaviour matrix B for finite deterministic and finite stochastic machines is computable in polynomial time because there exists an integer i with $i \leqslant |Q| - 1$, such that $B_i = B$ [17,35]. The natural question is why we do not use this nice property for fuzzy machines. And the simple answer is—due to the properties of L and fuzzy relation algebra, we cannot generate free Noetherian modules [35] over the set of states Q for FFM. In particular, the same is valid for nondeterministic machines, cf. Corollary 3.6.
(2) The estimation $k \leqslant 2^{|Q|} - 1$ is well known for nondeterministic machines: Burkhard [8,9] has shown that the upper bound $2^{|Q|} - 1$ for the length of the words cannot be improved. This comment shows the importance of Theorem 3.5: its results include as a particular case a fundamental result for nondeterministic machines, strengthening its range of validity to fuzzy machines.

4. Equivalence of finite L-fuzzy machines

The idea for equivalence is first proposed for deterministic machines by Moore [29].

Historical remarks for equivalence, reduction and minimization are given in [32,51].

In this section we study various equivalence problems and their algorithmical decidability. Although the terminology for equivalence problems for finite L-fuzzy machines is introduced in accordance with the classical definitions for deterministic, nondeterministic and stochastic machines [30,51], we shall give the necessary definitions.

4.1. Definition. Let $A = (X, Q, Y, M)$ be a finite L-fuzzy machine, $f : Q \to L$ be a map and $F = (f(q))_{q \in Q}$ be the corresponding row matrix of type $1 \times |Q|$ with elements defined by the map f.

(i) F is called *initial distribution* of the membership degrees over the set of states Q.
(ii) (A, F) is said to be *initialized finite* L-*fuzzy machine* (IFFM) associated to the finite L-fuzzy machine $A = (X, Q, Y, M)$ with initial distribution F.

The complete input–output behaviour of (A, F) is described by $F \bullet T$ and the behaviour of (A, F) is described by $F \bullet B$ (here T stands for the complete input–output behaviour of A and B stands for the input–output behaviour of A).

We will be interested in two special kinds of initial distributions, respectively IFFMs:

4.2. Definition. For an FFM $A = (X, Q, Y, M)$:

(i) The row matrix I_q of type $1 \times |Q|$ such that

$$I_q(q') = \begin{cases} 1 & \text{if } q = q', \\ 0 & \text{if } q \neq q' \end{cases}$$

is called a *selecting initial distribution* of membership degrees over Q. (A, I_q) is called an *initial* L-*fuzzy machine*. Often we shall write (A, q) for (A, I_q).

(ii) The nonzero row matrix 0_q of type $1 \times |Q|$ such that

$$0_q(q') = \begin{cases} 0 & \text{if } q = q', \\ \text{arbitrary} & \text{if } q \neq q' \end{cases}$$

is called an *isolating initial distribution* of membership degrees over Q.

Remark. (1) I_q from Definition 4.2(i) selects the state $q \in Q$ as *initial state* for A. The complete input–output behaviour of the initial FFM (A, I_q) is described by $I_q \bullet T$, its behaviour—by $I_q \bullet B$.

(2) 0_q from Definition 4.2(ii) isolates the state $q \in Q$ when A begins with 0_q. The complete input–output behaviour of the IFFM $(A, 0_q)$ is described by $0_q \bullet T$ and its behaviour—by $0_q \bullet B$.

4.3. Definition. Let $A = (X, Q, Y, M)$ and $A' = (X, Q', Y, M')$ be given finite L-fuzzy machines with the same X and Y.

(i) The IFFMs (A, F) and (A', F') are called:
- *equivalent*, written $(A, F) \sim (A', F')$, if the complete input–output behaviour of (A, F) coincides with that of (A', F'),
- *k-equivalent*, written $(A, F) \sim_k (A', F')$, if the input–output behaviour of (A, F) coincides with that of (A', F') for all words $(u \mid v) \in (X \mid Y)^*$, with $|u| = |v| \leqslant k$.

(ii) $A = (X, Q, Y, M)$ is called:
- *weakly covered* by $A' = (X, Q', Y, M')$, written $A \underset{\approx}{\subset} A'$, if for each F there exists F' such that $(A, F) \sim (A', F')$,
- *k-weakly covered* by $A' = (X, Q', Y, M')$, written $A \underset{\approx}{\subset}_k A'$ if for each F there exists F' such that $(A, F) \sim_k (A', F')$.

(iii) $A = (X, Q, Y, M)$ and $A' = (X, Q', Y, M')$ are called:
- *weakly equivalent*, written $A \approx A'$, if $A \underset{\approx}{\subset} A'$ and $A' \underset{\approx}{\subset} A$,
- *k-weakly equivalent*, written $A \approx_k A'$, if $A \underset{\approx}{\subset}_k A'$ and $A' \underset{\approx}{\subset}_k A$.

(iv) The state $q \in Q$ is called:
- *equivalent* to the state $q' \in Q'$, written $q \sim q'$, if $(A, q) \sim (A', q')$,
- *k-equivalent* to the state $q' \in Q'$, written $q \sim_k q'$, if $(A, q) \sim_k (A', q')$.

(v) $A = (X, Q, Y, M)$ is called:
- *covered* by $A' = (X, Q', Y, M')$, written $A \lesssim A'$, if for each $q \in Q$ there exists an equivalent $q' \in Q'$,
- *k-covered* by $A' = (X, Q', Y, M')$, written $A \lesssim_k A'$ if for each $q \in Q$ there exists a k-equivalent $q' \in Q'$.

(vi) $A = (X, Q, Y, M)$ is called:
- *equivalent* to $A' = (X, Q', Y, M')$, written $A \sim A'$, if $A \lesssim A'$ and $A' \lesssim A$,
- *k-equivalent* to $A' = (X, Q', Y, M')$, written $A \sim_k A'$, if $A \lesssim_k A'$ and $A' \lesssim_k A$.

All the notions for k-equivalence are obtained as a restriction of the corresponding equivalence notions for fixed $k \in \mathsf{N}$. For this reason we shall list only equivalence statements.

The following properties are valid:

(i) $A \lesssim A' \Rightarrow A \underset{\approx}{\subset} A'$,
(ii) $A \sim A' \Rightarrow A \approx A'$,
(iii) The relations $\lesssim$ and $\underset{\approx}{\subset}$ are reflexive and transitive,
(iv) The relations $\sim$ and $\approx$ are equivalence relations.

4.4. Theorem. *For the FFMs $A=(X,Q,Y,M)$ and $A'=(X,Q',Y,M')$ we have*:

(i) $(A,F) \sim (A',F')$ *iff* $F \bullet T_A = F' \bullet T_{A'}$ (*iff* $F \bullet B_A = F' \bullet B_{A'}$),
(ii) *for* $q \in Q$, $q' \in Q'$: $(A,I_q) \sim (A',I_{q'})$ *iff* $I_q \bullet T_A = I_{q'} \bullet T_{A'}$ (*iff* $I_q \bullet B_A = I_{q'} \bullet B_{A'}$),
(iii) $A \underset{\approx}{\subset} A'$ *iff for each* F *there exists* F' *such that* $F \bullet T_A = F' \bullet T_{A'}$ (*iff* $F \bullet B_A = F' \bullet B_{A'}$),
(iv) $A \approx A'$ *iff for each* F *there exists* F' *such that* $F \bullet T_A = F' \bullet T_{A'}$ (*iff* $F \bullet B_A = F' \bullet B_{A'}$) *and vice versa*,
(v) $A \lesssim A'$ *iff for each* $q \in Q$ *there exists* $q' \in Q'$ *such that* $I_q \bullet T_A = I_{q'} \bullet T_{A'}$ (*iff* $I_q \bullet B_A = I_{q'} \bullet B_{A'}$),
(vi) $A \sim A'$ *iff for each* $q \in Q$ *there exists* $q' \in Q'$ *such that* $I_q \bullet T_A = I_{q'} \bullet T_{A'}$ (*iff* $I_q \bullet B_A = I_{q'} \bullet B_{A'}$) *and vice versa.*

The proof follows from construction of the complete input–output behaviour matrix T and Definition 4.3 (for equivalence of machines and equivalence of states). The assertions in brackets follow from Theorem 3.5.

Remark. Theorem 4.4 is valid for k-equivalence, $k \in \mathsf{N}$.

4.5. Theorem. *The following relations are algorithmically decidable for arbitrary finite* L-*fuzzy machines* $A=(X,Q,Y,M)$ *and* $A'=(X,Q',Y,M')$, *for any fixed* $k \in \mathsf{N}$:

(i) *k-equivalence of states*,
(ii) *k-equivalence of IFFMs*,
(iii) *k-weakly covering of machines*,
(iv) *k-covering of machines*,
(v) *k-equivalence of machines.*

Proof. Let $A=(X,Q,Y,M)$ and $A'=(X,Q',Y,M')$ be given. Since the behaviour matrices are computable for any $k \in \mathsf{N}$, we obtain (cf. Theorem 4.4):

(i) For any $q \in Q$, $q' \in Q'$: $(A,I_q) \sim_k (A',I_{q'})$ *iff* $I_q \bullet T_k = I_{q'} \bullet T'_k$. We can check directly the last equality. (Note that if $(A,I_q) \sim_k (A',I_{q'})$, then the rows, corresponding to $q \in Q$ and $q' \in Q'$ in T_k and T'_k are equal.)
(ii) $(A,F) \sim_k (A',F')$ *iff* $F \bullet T_k = F' \bullet T'_k$. For the last equality we apply direct problem resolution.
(iii) According to Theorem 4.4(iii) $A \underset{\approx}{\subset} A'$ if for each F there exists F' such that $F \bullet B_A = F' \bullet B_{A'}$. Obviously, we cannot list all F, but any initial distribution F is representable as a CLC:

$$F = (f(q))_{q \in Q} = (f(q_1) \wedge I_{q_1}) \vee \cdots \vee (f(q_n) \wedge I_{q_n}).$$

Bearing this in mind, we can establish $A \underset{\approx}{\subset} A'$: for each $I_q, q \in Q$ we have to find an F' such that $I_q \bullet B_A = F' \bullet B_{A'}$. It means that we have to solve the fuzzy matrix equation $U \bullet T_k = F' \bullet T_k'$ for the unknown F' using Algorithm 2.10 (here U is the square unit matrix.),
(iv) follows from the definition of k-covering of machines and Theorem 4.4(i),
(v) follows from the definition of k-equivalence of machines and Theorem 4.4(i).

4.6. Corollary. *For any FFM $A = (X, Q, Y, M)$, it is algorithmically decidable whether $q' \sim_k q''$, for arbitrary q', $q'' \in Q$.*

Note that if $q' \sim q''$ then the rows corresponding to q' and q'' are equal in T (in T_k, respectively), as well as in B (in B_k, respectively).

Remark. In any of those cases establishing the corresponding kind of equivalence requires computing the behaviour matrix B, but the time complexity for this is exponential, cf. Theorem 3.5 and Corollary 3.8. According to Remark 3.9, it makes sense again to consider k-equivalence or α-cuts.

5. Reduction and minimization of finite L-fuzzy machines

In this section we investigate reduction and minimization problems and their algorithmical decidability.

Although the terminology for reduction and minimization is like this for deterministic, nondeterministic and stochastic machines [12,30,51], we shall give here the necessary definitions.

5.1. Definition. Let $A = (X, Q, Y, M)$ be a finite L-fuzzy machine. We shall say that

- A is in *reduced form* if for each $q', q'' \in Q$ the relation $q' \sim q''$ implies $q' = q''$,
- A' is a *reduct* of A if A' is in reduced form and $A \sim A'$.

5.2. Theorem. *Let $A = (X, Q, Y, M)$ be a finite L-fuzzy machine with complete input–output matrix T. If T contains two identical rows, then there exist a finite L-fuzzy machine A' with $|Q| - 1$ states, such that $A \sim A'$.*

Proof. Let the ith and the jth rows in T are identical. According to Theorem 4.4(ii) the states q_i and q_j are equivalent. We construct a machine $A' = (X, Q', Y, M')$ with $Q' = Q \backslash \{q_i\}$ and

$$M' = \{M'(x \mid y) | M'(x \mid y) = (m'_{qq'}(x \mid y)), x \in X, y \in Y, q, q' \in Q', m'_{qq'}(x \mid y) \in L\},$$

where

$$m'_{qq'}(x \mid y) = \begin{cases} m_{qq'}(x \mid y) & \text{if } q' \neq q_i, \\ m_{qq_i}(x \mid y) \vee m_{qq_j}(x \mid y) & \text{if } q' = q_i. \end{cases}$$

Obviously, $|Q'| = |Q| - 1$. Each state $q \in Q$, $q \neq q'$ from A has $q \in Q'$ from A' as its equivalent; $q_i \in Q$ from A is equivalent to $q_j \in Q'$ from A'. Hence $A \sim A'$.

The above constructed machine A' with $A \sim A'$ is called *natural reduct* of A.

According to Theorem 5.2 the matrix T for A indicates whether A is in reduced form: if no two (or more) rows in T are identical, then A is in reduced form. Otherwise the equivalence classes over Q with respect to the relation 'equivalence of states' are determined to obtain the set of the natural reducts for A.

5.3. Corollary. *If for a finite* L*-fuzzy machine A we know its behaviour matrix B, then*:

(i) *it is algorithmically decidable whether A is in reduced form*,
(ii) *all natural reducts of A are constructable*,
(iii) *all natural reducts of A have sets of states with the same cardinality*,
(iv) *there exist finite number of natural reducts of A.*

Proof. (i) We have to check whether there exist identical rows in B (cf. Corollary 4.6).

(ii) and (iii) follow from Theorem 5.2.

(iv) Let $A=(X,Q,Y,M)$ be an FFM and $A'=(X,Q',Y,M')$ be a natural reduct of A. Let the corresponding behaviour matrices B and B' be known. We can obtain all reducts of A using the following method: from each natural reduct $A'=(X,Q',Y,M')$ we compute a set of reducts, associated to A': $\{A_r=(X,Q',Y,M_r)\}$, where $M_r=\{M_r(x\mid y)\mid x\in X, y\in Y\}$ and any $M_r(x\mid y)$ is a solution of the matrix equation $M'(x\mid y)\bullet B'=M_r(x\mid y)\bullet B'$ (cf. Algorithm 2.10). All reducts of A have the same number of states according to Theorem 5.2.

But if a fuzzy linear system of equations (fuzzy relational equation, respectively) is consistent, it has interval solutions. Hence the number of all reducts of A is not finite.

Remark. This result and Remark 3.9 mean that depending on $k\leqslant b^{|Q|}-1$ in some cases we shall be able to solve the complete reduction problem, in other cases we shall prefer to solve it for the words in fixed length not exceeding a natural number $l=1,2,\dots,l<k$. There exist also situations where an α-cut over the set of matrices $M(x\mid y)\in M$ makes sense—it will reduce the number a (b, respectively) of distinct membership degrees, and it will affect on the time to compute B.

We shall consider now the minimization problem.

5.4. Definition. Let $A=(X,Q,Y,M)$ and $A'=(X,Q',Y,M')$ be FFMs. $A=(X,Q,Y,M)$ is in *minimal form* if for each I_q, $q\in Q$ there does not exists 0_q, $q\in Q$, such that $(A,I_q)\sim(A,0_q)$.

5.5. Theorem. *For any finite* L*-fuzzy machine A with known behaviour matrix B, it is algorithmically decidable whether A is in minimal form.*

Proof. We have to solve the relational equation $U\bullet B=0_q\bullet B$ for 0_q, cf. Algorithm 2.10. If it has no solutions, A is in minimal form.

Concluding remark. Although fuzzy linear systems of equations and fuzzy relational equations are solvable in polynomial time and although the notions for equivalence, reduction and minimization are introduced in accordance with the classical machines theory [11,30,51], the time complexity function for solving any of those problems for finite L-fuzzy machines is exponential. It is sensible to consider equivalence, reduction and minimization for words of smaller length (smaller than $k\leqslant b^{|Q|}-1$) or for suitable α-cuts on the transition-output matrices.

Acknowledgements

The author is grateful to the referees for their valuable suggestions.

References

[1] J. Adamek, W. Wechler, Minimization of R-fuzzy automata, in: Studien zur Algebra und Ihre Anwendungen, Akademie-Verlag, Berlin, 1976.
[2] B.D. Anderson, M.A. Arbib, E.G. Manes, Foundations of System Theory: Finitary and Infinitary Conditions, in: Lecture Notes in Economics and Mathematical Systems, Vol. 115, Springer, Berlin, Heidelberg, New York, 1975.
[3] M.A. Arbib, E.G. Manes, Fuzzy Morphisms in Automata Theory, Lecture Notes in Computer Science, Vol. 25, Springer, Berlin, 1975, pp. 80–86.
[4] M.A. Arbib, E.G. Manes, Fuzzy machines in a category, Bull. Austral. Math. Soc. 13 (1975) 169–210.
[5] A. Arnold, A. Dicky, M. Nivat, A note about minimal nondeterministic automata, Bull. EATCS 47 (1992) 166–169.
[6] N.N. Bismas, State minimization of incompletely specified sequential machines, IEEE Trans. Comput. 23 (1) (1974) 80–84.
[7] J. Brunner, W. Wechler, On behaviour of R-fuzzy automata, Proc. 5th Symp. on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science, Vol. 45, 1976, pp. 210–215.
[8] H.-D. Burkhard, Über Experimente an nicht-deterministischen Automaten, Elektron. Inform. Kybernet. 5 (1969) 347–376.
[9] H.-D. Burkhard, Über Experimente an nicht-deterministischen Automaten, Elektron. Inform. Kybernet. 6 (1970) 3–14.
[10] J.W. Carlyle, Reduced forms for stochastic sequential machines, J. Math. Anal. Appl. 7 (2) (1963) 167–175.
[11] M.K. Chirkov, Fundamentals of general finite machines theory (Russian), Izd. Leningradskogo Universiteta, Leningrad, 1975.
[12] M.K. Chirkov, On the minimization of incomplete p-machines (Russian), Bvichislitel'naja technika I vopr. kibernetiki, Leningrad No. 16, 1979, pp. 109–137.
[13] B. De Baets, Analytical solution methods for fuzzy relational equations, in: D. Dubois, H. Prade (Eds.), The Series: Fundamentals of Fuzzy Sets, The Handbooks of Fuzzy Sets Series, Vol. 1, Kluwer Academic Publishers, Dordrecht, 2000, pp. 291–340.
[14] P.D. Dien, On the complexity of the problem of state minimization for incompletely specified sequential automata, Elektron. Inform. Kybernet. 11 (1/2) (1975) 3–17.
[15] D. Dubois, H. Prade (Eds.), Fundamentals of Fuzzy Sets, in: The Handbooks of Fuzzy Sets Series, Vol. 7, Kluwer Academic Publishers, Dordrecht, 2000.
[16] Eilenberg, Automata, Languages and Machines, Vol. A, Academic Press, New York, 1974.
[17] S. Even, Comments on the minimization of stochastic machines, IEEE Trans. Electron. Comput. EC-14 (4) (1965) 634–637.
[18] M.R. Garey, D.S. Johnson, Computers and Intractability, A Guide to the Theory of NP-Completeness, Freeman, San Francisco, 1979.
[19] A. Kandel, S.C. Lee, Fuzzy Switching and Automata: Theory and Applications, Crane& Russak, 1979.
[20] G. Klir, U.H. St. Clair, Bo Yuan, Fuzzy Set Theory Foundations and Applications, Prentice-Hall PRT, Englewood Cliffs, NJ, 1997.
[21] G. Klir, Bo Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice-Hall PTR, NJ, 1995.
[22] W. Kuich, A. Salomaa, Semirings, Automata, Languages, Springer, Berlin, 1986.
[23] D.S. Malik, J.N. Mordeson, Fuzzy Discrete Structures, Springer, Berlin, 2000.
[24] S. Malik, J.N. Mordeson, M.K. Sen, Minimization of fuzzy finite automata, Inform. Sci. 113 (1999) 323–330.
[25] A. Meduna, Automata and Languages: Theory and Applications, Springer, Berlin, 2000.
[26] M. Mizumoto, Fuzzy automata and fuzzy grammars, Ph.D. Thesis, Faculty of Engineering Science, Osaka University, Osaka, Japan, 1971.
[27] M. Mizumoto, K. Tanaka, Fuzzy-fuzzy automata, Kybernetes 5 (1976) 107–112.

[28] M. Mizumoto, J. Toyoda, K. Tanaka, Some consideration on fuzzy automata, J. Comput. System Sci. 3 (4) (1969) 409–422.
[29] E.F. Moore, Gedanken-experiments of sequential machines, in: Automata Studies, Princeton University Press, Princeton, NJ, 1956, pp. 129–153.
[30] A. Paz, Introduction to Probabilistic Automata, Academic Press, New York, London, 1971.
[31] K. Peeva, On equivalence, reduction and minimization of stochastic automata (Bulgarian English, Russian summary), Godishnik Viss. Uchebn. Zaved. Prilozhna Mat. T.15 book 2, 1979, pp. 49–58.
[32] K. Peeva, A general algebraic approach to equivalence, reduction and minimization of finite automata, Report TI-1/82, Technische Hochschule Darmstadt, Fachbereich Informatik, Darmstadt, 1982.
[33] K. Peeva, Behaviour, reduction and minimization of finite L-automata, Fuzzy Sets and Systems 28 (2) (1988) 171–181.
[34] K. Peeva, Categories of stochastic and fuzzy acceptors for pattern recognition, in: J. Kacprzyk, A. Straszak (Eds.), Proc. 1st Joint IFSA—EG and EURO-WC Workshop on Progress in Fuzzy Sets in Europe, November 25–27 1986, Warsaw, Polska Akademia Nauk, Warsaw, 1988, pp. 289–298.
[35] K. Peeva, Equivalence, reduction and minimization of finite automata over semirings, Theoret. Comput. Sci. 88 (1991) 269–285.
[36] K. Peeva, Fuzzy linear systems, Fuzzy Sets and Systems 49 (1992) 339–355.
[37] K. Peeva, Equivalence, reduction and minimization of finite fuzzy automata, in: Preproc. DCAGRS 2000, Descriptional Complexity of Automata, Grammars and Related Structures, Report No. 555, 2000, Department of Computer Science, UWO, Canada.
[38] K. Peeva, B. Deliiska, V. Hristov, Computer code for direct problem resolution in fuzzy relation calculus, in: D. Ivanchev, M. Todorov (Eds.), 27th Summer School Applications of Mathematics in Engineering and Economics, Sozopol, 2001, Heron Press, 2002, pp. 507–515.
[39] K. Peeva, I. Gueorguieva, J. Kjossev, Fuzzy linear systems for adequate diagnosis and prediction, in: B. Cheshankov, M. Todorov (Eds.), 25th Jubilee Summer School Applications of Mathematics in Engineering and Economics, Sozopol, 1999, Heron Press, 2001, pp. 178–184.
[40] K. Peeva, P. Nedyalkova, Reduction and minimization of fuzzy automata (Bulgarian English, Russian summary), Godishnik Viss. Uchebn. Zaved. Prilozhna Mat t. 17 book. 1, 1981, pp. 29–39.
[41] C.P. Pfleger, State reduction in incompletely specified finite-state machines, IEEE Trans. Comput. C-22 (12) (1973) 1099–1102.
[42] S. Przybylo, Relations entre les morphismes et l'équivalence des automates indéterminés, Problems Control Inform. Theory 2 (2) (1973) 163–171.
[43] C.V. Rao, N.N. Bismas, Minimization of incompletely specified sequential machines, IEEE Trans. Comput. C-24 (11) (1975) 1089–1100.
[44] E. Sanchez, Resolution of composite fuzzy relation equations, Inform. and Control 30 (1976) 38–48.
[45] S. Santos, Maximin automata, Inform. and Control 13 (1968) 363–377.
[46] E.S. Santos, Maximin, minimax and composite sequential machines, J. Math. Anal. Appl. 24 (1968) 246–259.
[47] E.S. Santos, On reduction of maxi-min machines, J. Math. Anal. Appl. 40 (1972) 60–78.
[48] E. Santos, Max-product machines, J. Math. Anal. Appl. 37 (1972) 677–686.
[49] E.S. Santos, W.G. Wee, General formulation of sequential machines, Inform. and Control 12 (1) (1968) 5–10.
[50] St. Schwartzman, The Words of Mathematics. An Etymological Dictionary of Mathematical Terms Used in English, Mathematical Association of America, Washington, 1994.
[51] P. Starke, Abstract Automata, Academic Press, New York (1978, 2nd Edition), Birkhauser, Basel, 1998.
[52] V. Topencharov, K. Peeva, Equivalence, reduction and minimization of finite fuzzy automata, J. Math. Anal. Appl. 84 (1981) 270–281.
[53] W. Wechler, R-Fuzzy Automata with a time-variant structure, in: A. Blikle (Ed.), 3rd Symposium on Mathematical Foundations of Computer Science, 1974, Lecture Notes in Computer Science, Vol. 28, Springer, Berlin, 1975, pp. 73–76.
[54] W. Wechler, The Concept of Fuzziness in Automata and Language Theory, Akademie-Verlag, Berlin, 1978.