Minimization of L-valued Finite Sequential Machine

Chang Shu† and Zhi-wen Mo††,

College of Mathematics and Software Science, Sichuan Normal University, Chengdu, Sichuan, 610066, P. R. China

Summary

We take a finite lattice-ordered monoid as the truth-value domain for fuzzy sequential machines in this paper.The notion of L-valued finite sequential machines(L-FSMs.for short)occurring in [1]is modified and some related properties are also discussed.Moreover,we formulate the definition of behavior mapping to determine the operation of an L-FSM,and give an algorithm to compute the image of the behavior mapping for a given L-FSM. Finally, its statewise equivalence relations are defined, based on which we present a minimization algorithm for an L-FSM.

Key words:

Fuzzy finite sequential machine, Lattice-ordered monoid, Equivalence, Minimization.

1.Introduction

As soon as the concept of fuzzy sequential machines has been introduced, Santos [2],[3] investigated the minimization problems for finite max-min fuzzy machines and finite max-product fuzzy machines. It is worth of mention that Qiu [4],[5] established a fundamental framework of automata theory based on complete residuated lattice-valued logic, to a certain extent, automata theory based on lattice-ordered monoids initiated by Li and Pedrycz [6], which is the generalization of previous study. Many authors have contributed to this field such as Cheng and Mo [8]. Recently, Lei and Li [7] presented an algorithm aimed at the minimization of states in automata theory based on lattice-ordered monoids. Following the notion of L-valued sequential machines introduced by Xing and Qiu [1], in this paper, we study the minimization of this fuzzy sequential machine. Unlike L-valued sequential machines occurring in [1], whose basic structure of membership value was complete residuated lattice, the minimization problem is still not considered. For that reason, we focus our attention to efficient minimization algorithm by means of defining a behavior mapping to determine the operation of an L-FSM. In detail, we refer to the recent studies from [1], [6],[7]. In a sense, the paper is arranged in the following ways.

In Section 2 we first define L-valued finite sequential machine (L-FSM, for short) based on lattice-ordered monoids and discuss some related properties. In Theorem 2.1, two equivalence conditions about lattice-ordered monoids are displayed. In Section 3, a behavior mapping determining the operation of an L-FSM is given, and an algorithm computing the image of the mapping for a given L-FSM is presented. The algorithm with finite steps is proved by Theorem 3.2. Section 4 deals with the minimization problem of an L-FSM. We give two kinds of statewise equivalence relations and obtain its minimization algorithm with finite steps.

2. L-VALUED FINITE SEQUENTIAL MACHINE

Definition 2.1 *[6] Given a lattice L, $\bigvee$ and $\bigwedge$ represent the supermum and infimum operation on L respectively, 0 and 1 are the least and largest element. Assume that there is a binary operation $\bullet$ on L such that $(L, \bullet, \epsilon)$ is a monid with identity $\epsilon \in L$. We call L an ordered-monoid if it satisfies the following condition:*

Manuscript received September 5, 2012
Manuscript revised September 20, 2012

(i) $\forall a, b \in L, a \leq b \Longrightarrow \forall x \in L, a \bullet x \leq b \bullet x$ *and* $x \bullet a \leq x \bullet b$;

if L is an ordered-monoid and it satisfies the distributive laws:

(ii) $\forall a, b, c \in L, a \bullet (b \bigvee c) = (a \bullet b) \bigvee (a \bullet c)$, *and* $(b \bigvee c) \bullet a = (b \bullet a) \bigvee (c \bullet a)$. *Then we call L a lattice-ordered monoid.*

Remark: Without any explicit specification, in what follows, we will be referring to L as a finite lattice-ordered monoid and $(L, \bullet, e)$ as an ordered-monoid without nilfactor, that is, for any $a, b \in L$, if $a \neq 0, b \neq 0$, then $a \bullet b \neq 0$.

Definition 2.2 *Let $(L, \bullet, \bigvee)$ be a lattice-ordered monoid. A quadruple $A = (Q, X, Y, \delta)$ is called an L-valued finite sequential machine (L-FSM, for short), where $Q = \{q_1, q_2, \cdots, q_n\}$, $X = \{a_1, a_2, \cdots, a_m\}$, $Y = \{b_1, b_2, \cdots, b_l\}$ are nonempty finite sets of states, input letters and output letters, respectively. And $\delta \in L^{Q\times X\times Y\times Q}$, an L-valued subsets of $Q \times X \times Y \times Q$, is the L-valued transition-output function of A.*

In the case of single input-output:

$$\delta : Q \times X \times Y \times Q \longrightarrow L$$

$$(q_i, a_s, b_r, q_j) \longmapsto \delta(q_i, a_s, b_r, q_j) = \theta_{ij}(a_s|b_r)$$

$\delta(a_s|b_r) = [\theta_{ij}(a_s|b_r)]_{n\times n}, s = 1, 2, \cdots, m; r = 1, 2, \cdots, l.$

We regard $\theta_{ij}(a_s|b_r) \in L$ as the membership degree that L-FSM A will enter state $q_j \in Q$ and produce output $b_r \in Y$ given that the present state is $q_i \in Q$ and the input is $a_s \in X$.

The free monoid of the words over the set $X(Y)$ is denoted by $X^*(Y^*)$ with the empty word $\wedge$ as the identity element. The length of the word $x \in X^*(y \in Y^*)$ is denoted by $|x|(|y|)$. By definition $|\wedge| = 0$.

For $x \in X^*$ and $y \in Y^*$, if $|x| = |y|$, we write $(x|y) \in (X|Y)^*$, to distinguish it from the case $(x, y) \in X^* \times Y^*$.

Definition 2.3 *Let $(L, \bullet, \bigvee)$ be a lattice-ordered monoid and $A = (Q, X, Y, \delta)$ be an L-FSM. The extended transition-output function $\delta^* \in L^{Q\times X^*\times Y^*\times Q}$ is defined as follows:*

$$\delta^* : Q \times X^* \times Y^* \times Q \longrightarrow L$$

$$(q_i, x, y, q_j) \longmapsto \delta^*(q_i, x, y, q_j) = \begin{cases} e, & \text{if } x = y = \wedge \text{ and } q_i = q_j; \\ 0, & \text{otherwise.} \end{cases}$$

$$(q_i, xa, yb, q_j) \longmapsto \delta^*(q_i, xa, yb, q_j) = \bigvee_{q_o \in Q} [\delta^*(q_i, x, y, q_o) \bullet \delta(q_o, a, b, q_j)] = \bigvee_{q_o \in Q} [\theta_{io}(x|y) \bullet \theta_{oj}(a|b)]$$

$$\delta^*(\wedge|\wedge) = E_{n\times n}, E_{n\times n} = \begin{pmatrix} e & e \cdots & e \\ e & e \cdots & e \\ \vdots & \vdots & \vdots \\ e & e \cdots & e \end{pmatrix}$$

for any $(x, y) \in X^* \times Y^*$ and $(a, b) \in X \times Y$.

Proposition 2.1 Let $(L, \bullet, \bigvee)$ be a lattice-ordered monoid and $A = (Q, X, Y, \delta)$ be an L-FSM. Then $\delta = \delta^*|_{Q\times X\times Y\times Q}$.

Proof: Let $(a, b) \in X \times Y$, then $\delta^*(a|b) = \delta^*(\wedge a|\wedge b) = \delta^*(\wedge|\wedge) \circ \delta(a|b) = E_{n\times n} \circ \delta(a|b) = \delta(a|b)$.

Proposition 2.2 Let $(L, \bullet, \bigvee)$ be a lattice-ordered monoid and $A = (Q, X, Y, \delta)$ be an L-FSM. For any $q, p \in Q$ and $(x, y) \in X^* \times Y^*$, if $|x| \neq |y|$, then $\delta^*(q, x, y, p) = 0$.

Proof: Let $q, p \in Q, (x, y) \in X^* \times Y^*$ and $|x| \neq |y|$. Without loss of generality, we assume $|x| > |y|$ and $|y| = n$, and proceed by induction on n.

For the case of $n = 0$, it is obvious that $y = \wedge$ and $x \neq \wedge$. By Definition 2.3 we have $\delta^*(q, x, y, p) = 0$.

For the case of $n \geq 1$, suppose the proposition hold when $|y| = n - 1$, the proposition is also true when $|y| = n$. Indeed:

Write $x = ua, y = vb$, where $(u, v) \in X^* \times Y^*, |u| > |v|, |v| = n - 1$ and $(a, b) \in X \times Y$. By hypothesis, for any $q, p \in Q$, we have $\delta^*(q, u, v, p) = 0$. So $\delta^*(q, x, y, p) = \delta^*(q, ua, vb, p) = \bigvee_{r \in Q} [\delta^*(q, u, v, r) \bullet \delta(r, a, b, p)] = 0$. i.e. the proposition holds when $|x| > |y|$.

Similarly, we can show the result is true if $|x| < |y|$.

Remark: The above proposition indicates that the length of input word must be the same as that of output word in mathematical induction. Subsequently, we denote $(X|Y)^*$ as the set of all input-output pairs of words of the same length.

Proposition 2.3 Let $(L, \bullet, \epsilon)$ be an ordered-monoid without nilfactor and $A = (Q, X, Y, \delta)$ be an L-FSM. Then the following conditions are equivalent:

(i) $\forall q \in Q, (a, b) \in X \times Y, \exists p \in Q, \delta(q, a, b, p) > 0$;

(ii) $\forall q \in Q, (x|y) \in (X|Y)^*, \exists p \in Q, \delta^*(q, x, y, p) > 0$.

Proof:

(i)$\Longrightarrow$(ii): We prove it by induction on the length of x and y.

If $|x| = |y| = 0$, then $x = y = \wedge$, take $p = q$, we have $\delta^*(q, x, y, p) = \epsilon > 0$; if $|x| = |y| = 1$, it is obvious that (ii) holds. Assume that (ii) holds for $|x| = |y| = n-1 (n > 1)$, (ii) also holds for $|x| = |y| = n$. Indeed:

Write $x = x_1 a, y = y_1 b$, where $(x_1|y_1) \in (X|Y)^*, (a, b) \in X \times Y$ and $|x_1| = |y_1| = n - 1$. Now by Definition 2.3, $\delta^*(q, x, y, p) = \bigvee_{s\in Q} [\delta^*(q, x_1, y_1, s) \bullet \delta(s, a, b, p)] \geq \delta^*(q, x_1, y_1, r) \bullet \delta(r, a, b, p)$. By hypothesis, there exists $r \in Q$ such that $\delta^*(q, x_1, y_1, r) > 0$. Also by (i), $\delta(r, a, b, p) > 0$. Since $(L, \bullet, \epsilon)$ is an ordered-monoid without nil-factor, then $\delta^*(q, x_1, y_1, r) \bullet \delta(r, a, b, p) > 0$. i.e. $\delta^*(q, x, y, p) > 0$.

(ii)$\Longrightarrow$(i): Straightforward.

Theorem 2.1 *The following conditions are equivalent:*

(i) Let $(L, \bullet, \bigvee)$ be a lattice-ordered monoid, that is, $a \bullet (b \bigvee c) = (a \bullet b) \bigvee (a \bullet c)$ and $(b \bigvee c) \bullet a = (b \bullet a) \bigvee (c \bullet a)$.

(ii) For any L-FSM $A = (Q, X, Y, \delta), q, p \in Q, (x_1|y_1) \in (X|Y)^*$ and $(x_2|y_2) \in (X|Y)^*$, then $\delta^*(q, x_1x_2, y_1y_2, p) = \bigvee_{r\in Q} [\delta^*(q, x_1, y_1, r) \bullet \delta^*(r, x_2, y_2, p)]$. i.e. $\delta^*(x_1x_2|y_1y_2) = \delta^*(x_1|y_1) \circ \delta^*(x_2|y_2)$.

(iii) For any L-FSM $A = (Q, X, Y, \delta), q, p \in Q, (a_1a_2\cdots a_k|b_1b_2\cdots b_k) \in (X|Y)^*$, of which $(a_i, b_i) \in X \times Y, i = 1, 2, \cdots, k$. Then

$$\delta^*(q, a_1a_2\cdots a_k, b_1b_2\cdots b_k, p) = \bigvee [\delta(q, a_1, b_1, q_1) \bullet \delta(q_1, a_2, b_2, q_2) \bullet \cdots \bullet \delta(q_{k-1}, a_k, b_k, p)].$$

i.e. $\delta^*(a_1a_2\cdots a_k|b_1b_2\cdots b_k) = \delta(a_1|b_1) \circ \delta(a_2|b_2) \circ \cdots \circ \delta(a_k|b_k)$.

Proof:

(i)$\Longrightarrow$(ii): We prove it by induction on the length of x_2 and y_2.

If $|x_2| = |y_2| = 0$, then $x_2 = y_2 = \wedge$ and $\delta^*(q, x_1x_2, y_1y_2, p) = \delta^*(q, x_1\wedge, y_1\wedge, p) = \delta^*(q, x_1, y_1, p) = \delta^*(q, x_1, y_1, p) \bullet \epsilon = \delta^*(q, x_1, y_1, p) \bullet \delta^*(p, \wedge, \wedge, p) = \bigvee_{r\in Q} [\delta^*(q, x_1, y_1, r) \bullet \delta^*(r, \wedge, \wedge, p)] = \bigvee_{r\in Q} [\delta^*(q, x_1, y_1, r) \bullet \delta^*(r, x_2, y_2, p)]$, that is, (ii) holds for $|x_2| = |y_2| = 0$. If $|x_2| = |y_2| = 1$, then from Definition 2.3, we have (ii) holds obviously. Assume that (ii) holds for $|x_2| = |y_2| = n-1 (n > 1)$, (ii) also holds for $|x_2| = |y_2| = n$. In fact:

Let $(x_1|y_1) \in (X|Y)^*$ and $(x_2|y_2) = (a_1a_2\cdots a_n|b_1b_2\cdots b_n) \in (X|Y)^*$, then by Definition 2.3 and assumption, we have $\delta^*(q, x_1x_2, y_1y_2, p)$

$$= \delta^*(q, x_1a_1a_2\cdots a_n, y_1b_1b_2\cdots b_n, p) = \bigvee_{r'\in Q} [\delta^*(q, x_1a_1a_2\cdots a_{n-1}, y_1b_1b_2\cdots b_{n-1}, r') \bullet \delta(r', a_n, b_n, p)] = \bigvee_{r'\in Q} [\bigvee_{r\in Q} [\delta^*(q, x_1, y_1, r) \bullet \delta^*(r, a_1a_2\cdots a_{n-1}, b_1b_2\cdots b_{n-1}, r')] \bullet$$

$$\delta(r', a_n, b_n, p)] = \bigvee_{r'\in Q} \bigvee_{r\in Q} [\delta^*(q, x_1, y_1, r) \bullet \delta^*(r, a_1a_2\cdots a_{n-1}, b_1b_2\cdots b_{n-1}, r') \bullet \delta(r', a_n, b_n, p)] = \bigvee_{r\in Q} [\delta^*(q, x_1, y_1, r) \bullet \bigvee_{r'\in Q} [\delta^*(r, a_1a_2\cdots a_{n-1}, b_1b_2\cdots b_{n-1}, r') \bullet \delta(r', a_n, b_n, p)]] = \bigvee_{r\in Q} [\delta^*(q, x_1, y_1, r) \bullet \delta^*(r, a_1a_2\cdots a_n, b_1b_2\cdots b_n, p)] = \bigvee_{r\in Q} [\delta^*(q, x_1, y_1, r) \bullet \delta^*(r, x_2, y_2, p)].$$

By induction, (ii) holds for any $(x_2|y_2) \in (X|Y)^*$.

(ii)$\Longrightarrow$(i): We can construct an L-FSM $A = (Q, X, Y, \delta)$ as follows:

$Q = \{q_0, q_1, q_2, q_3, q_4\}, X = \{a_1, a_2, a_3\}, Y = \{b_1, b_2, b_3\}, \delta(q_0, a_1, b_1, q_1) = a, \delta(q_1, a_2, b_2, q_2) = \epsilon$, $\delta(q_1, a_2, b_2, q_3) = \epsilon, \delta(q_2, a_3, b_3, q_4) = b$, $\delta(q_3, a_3, b_3, q_4) = c$ and $\delta(q, a, b, p) = 0$ in other case.

Take $x_1 = a_1, x_2 = a_2a_3, y_1 = b_1$ and $y_2 = b_2b_3$, then $\delta^*(q_0, x_1x_2, y_1y_2, q_4) = \bigvee_{r\in Q} [\delta^*(q_0, x_1, y_1, r) \bullet \delta^*(r, x_2, y_2, q_4)] = \bigvee_{r\in Q} [\delta(q_0, a_1, b_1, r) \bullet \delta^*(r, a_2a_3, b_2b_3, q_4)] = \delta(q_0, a_1, b_1, q_1) \bullet \delta^*(q_1, a_2a_3, b_2b_3, q_4) = a \bullet \delta^*(q_1, a_2a_3, b_2b_3, q_4)$.

Since $\delta^*(q_1, a_2a_3, b_2b_3, q_4) = \bigvee_{r\in Q} [\delta(q_1, a_2, b_2, r) \bullet \delta(r, a_3, b_3, q_4)] = [\delta(q_1, a_2, b_2, q_2) \bullet \delta(q_2, a_3, b_3, q_4)] \bigvee [\delta(q_1, a_2, b_2, q_3) \bullet \delta(q_3, a_3, b_3, q_4)] = (e \bullet b) \bigvee (e \bullet c) = b \bigvee c$, then we have $\delta^*(q_0, x_1x_2, y_1y_2, q_4) = a \bullet (b \bigvee c)$.

On the other hand, $\delta^*(q_0, x_1x_2, y_1y_2, q_4) = \delta^*(q_0, a_1a_2a_3, b_1b_2b_3, q_4) = \bigvee_{r\in Q} [\delta^*(q_0, a_1a_2, b_1b_2, r) \bullet \delta(r, a_3, b_3, q_4)] = [\delta^*(q_0, a_1a_2, b_1b_2, q_2) \bullet \delta(q_2, a_3, b_3, q_4)] \bigvee [\delta^*(q_0, a_1a_2, b_1b_2, q_3) \bullet \delta(q_3, a_3, b_3, q_4)] = [\delta^*(q_0, a_1a_2, b_1b_2, q_2) \bullet b] \bigvee [\delta^*(q_0, a_1a_2, b_1b_2, q_3) \bullet c]$.

Since $\delta^*(q_0, a_1a_2, b_1b_2, q_2) = \bigvee_{r\in Q} [\delta(q_0, a_1, b_1, r) \bullet \delta(r, a_2, b_2, q_2)] = \delta(q_0, a_1, b_1, q_1) \bullet \delta(q_1, a_2, b_2, q_2) = a \bullet e = a$ and $\delta^*(q_0, a_1a_2, b_1b_2, q_3) = \bigvee_{r\in Q} [\delta(q_0, a_1, b_1, r) \bullet \delta(r, a_2, b_2, q_3)] = \delta(q_0, a_1, b_1, q_1) \bullet \delta(q_1, a_2, b_2, q_3) = a \bullet e = a$, then $\delta^*(q_0, x_1x_2, y_1y_2, q_4) = (a \bullet b) \bigvee (a \bullet c)$.

So the above proof indicates that $a \bullet (b \bigvee c) = (a \bullet b) \bigvee (a \bullet c)$.

In addition, we construct another L-FSM $A' = (Q, X, Y, \delta')$, of which Q, X, Y are identical with those in A, $\delta'(q_0, a_1, b_1, q_1) = b, \delta'(q_0, a_1, b_1, q_2) = c, \delta'(q_1, a_2, b_2, q_3) = e, \delta'(q_2, a_2, b_2, q_3) = e, \delta'(q_3, a_3, b_3, q_4) = a$ and $\delta'(q, a, b, p) = 0$ in other case. We also get $(b \bigvee c) \bullet a = (b \bullet a) \bigvee (c \bullet a)$ in a similar way.

These tell us that the distributive laws of multiplication on $\bigvee$ is satisfied.

Likewise, we may show that (i)$\Longrightarrow$(iii) and (iii)$\Longrightarrow$(i).

Corollary Let $(L, \bullet, \bigvee)$ be a lattice-ordered monoid and $A = (Q, X, Y, \delta)$ be a single L-FSM, where $Q = \{q\}, X = \{a\}$ and $Y = \{b\}$. Then $\delta^*(q, a^k, b^k, q) = \delta^k(q, a, b, q)$. i.e. $\delta^*(a^k|b^k) = \delta(a|b) \circ \cdots \circ \delta(a|b)$.

Proof: The result is straightforward from Theorem 2.1(iii), so the proof is omitted.

Remark: If $\bullet = \wedge$, then $\delta^*(q, a^k, b^k, q) = \delta(q, a, b, q)$ in Corollary.

3. BEHAVIOR MAPPING OF AN L-FSM

Let $ImL^Q = \{(u_1, u_2, \cdots, u_n)^T | u_i \in L, i = 1, 2, \cdots, n\}$, column matrix $(u_1, u_2, \cdots, u_n)^T$ corresponds to Q with which elements u_i stands for the membership degree that the current state is $q_i, i = 1, 2, \cdots, n$.

We modify the definition in [1] as follows:

Definition 3.1 *Let $A = (Q, X, Y, \delta)$ be an L-FSM with $|Q| = n$, we define a behavior mapping $\tau_A : (X|Y)^* \longrightarrow L^Q$ as $\tau_A(x|y) = [t_j(x|y)]_{n\times 1}$*

$$= \begin{cases} E_{n\times 1}, & if\ (x|y) = (\wedge|\wedge); \\ \delta^*(x|y) \circ E_{n\times 1}, & if\ (x|y) \neq (\wedge|\wedge). \end{cases}$$

for any $(x|y) \in (X|Y)^*$, where $E_{n\times 1} = (e\ e\ \cdots\ e)^T_{n\times 1}$.

Each element $t_j(x|y) = \bigvee_{q_k\in Q} \theta_{jk}(x|y)$ of $\tau_A(x|y)$ determines the operation of A under the input word x, beginning at state q_j and producing the output word y after $|x| = |y|$ consecutive steps.

Theorem 3.1 *Let $(L, \bullet, e)$ be a finite ordered-monoid without nilfactor and $A = (Q, X, Y, \delta)$ be an L-FSM, where $|L| = k$ and $|Q| = n$. Then there exists at most k^{n^2} different matrices in $\{\delta^*(x|y)|(x|y) \in (X|Y)^*\}$.*

Proof: Let $Q = \{q_1, q_2, \cdots, q_n\}, X = \{a_1, a_2, \cdots, a_m\}$ and $Y = \{b_1, b_2, \cdots, b_l\}$. Let $L = \{u_1, u_2, \cdots, u_k\}$ be the set of all the elements which occur in the matrices $\{\delta(a_1|b_1), \cdots, \delta(a_1|b_l), \delta(a_2|b_1), \cdots, \delta(a_m|b_l)\}$. Hence, the number of different matrices which can be obtained in $\{\delta^*(x|y)|(x|y) \in (X|Y)^*\}$ is at most k^{n^2}.

We construct the sequence $I_0 \subset I_1 \subset \cdots \subset I$ of subsets of $I = (X|Y)^*$ as follows: [1]

$I_0 = \{\wedge|\wedge\}$

$I_1 = I_0 \bigcup\{(x|y) : (x|y) \in (X|Y)^* \text{ and } |x| = |y| = 1\}$

$\vdots$

$I_i = I_{i-1} \bigcup\{(x|y) : (x|y) \in (X|Y)^* \text{ and } |x| = |y| = i\}$

Evidently, $\tau_A(I_0) \subset \tau_A(I_1) \subset \cdots \subset \tau_A(I)$. Then we have:

Theorem 3.2 *Let $(L, \bullet, \bigvee)$ be a finite lattice-ordered monoid and $A = (Q, X, Y, \delta)$ be an L-FSM with its behavior mapping τ_A.*

(i) If $\tau_A(I_i) = \tau_A(I_{i+1})$, then $\tau_A(I_i) = \tau_A(I_{i+s})$, for each $s = 0, 1, 2, \cdots$.

(ii) $\tau_A(I_{k^{n^2}-1}) = \tau_A(I_{k^{n^2}}) = \cdots = \tau_A(I)$.

Proof: (i) We only show that $\tau_A(I_i) = \tau_A(I_{i+2})$.

Since $\tau_A(I_i) \subset \tau_A(I_{i+1}) \subset \tau_A(I_{i+2})$, we need to prove $\tau_A(I_{i+2}) \subset \tau_A(I_i)$. For any $(x|y) \in (X|Y)^*$,

$\tau_A(I_i) = \{[t_j(x|y)] \,|\, |x| = |y| \le i\}$

$\tau_A(I_{i+1}) = \{[t_j(xa_1|yb_1)] \,|\, |x| = |y| \le i \text{ and } (a_1, b_1) \in X \times Y\}$

$\tau_A(I_{i+2}) = \{[t_j(xa_1a_2|yb_1b_2)] \,|\, |x| = |y| \le i \text{ and } |a_1a_2| = |b_1b_2| = 2\}$

Next, for any $[t_j(xa_1a_2|yb_1b_2)] \in \tau_A(I_{i+2})$, we have $t_j(xa_1a_2|yb_1b_2) = \bigvee_{q_k \in Q} \theta_{jk}(xa_1a_2|yb_1b_2)$

$= \bigvee_{q_k \in Q} \delta^*(q_j, xa_1a_2, yb_1b_2, q_k)$

$= \bigvee_{q_k \in Q} \bigvee_{q_r \in Q} [\delta^*(q_j, xa_1, yb_1, q_r) \bullet \delta(q_r, a_2, b_2, q_k)]$

$= \bigvee_{q_k \in Q} [\bigvee_{q_r \in Q} \delta^*(q_j, xa_1, yb_1, q_r) \bullet \bigvee_{q_r \in Q} \delta(q_r, a_2, b_2, q_k)]$.

By $\tau_A(I_i) = \tau_A(I_{i+1})$, $\tau_A(I_{i+1}) = \tau_A(I_i) \bigcup\{[t_j(xa_1|yb_1)] \,|\, |x| = |y| = i$ and $(a_1, b_1) \in X \times Y\}$ implies that $[t_j(xa_1|yb_1)] \in \tau_A(I_i)$, i.e. there exists $(x_1|y_1) \in (X|Y)^*$, of which $|x_1| = |y_1| \le i$ such that $[t_j(xa_1|yb_1)] = [t_j(x_1|y_1)]$. So $\bigvee_{q_k \in Q} \theta_{jk}(xa_1|yb_1) = \bigvee_{q_k \in Q} \theta_{jk}(x_1|y_1), j = 1, 2, \cdots, n$.

Hence $t_j(xa_1a_2|yb_1b_2)$

$= \bigvee_{q_k \in Q} [\bigvee_{q_r \in Q} \delta^*(q_j, x_1, y_1, q_r) \bullet \bigvee_{q_r \in Q} \delta(q_r, a_2, b_2, q_k)]$

$= \bigvee_{q_k \in Q} \bigvee_{q_r \in Q} [\delta^*(q_j, x_1, y_1, q_r) \bullet \delta(q_r, a_2, b_2, q_k)]$

$= \bigvee_{q_k \in Q} \delta^*(q_j, x_1a_2, y_1b_2, q_k)$

$= \bigvee_{q_k \in Q} \theta_{jk}(x_1a_2|y_1b_2) = t_j(x_1a_2|y_1b_2), j = 1, 2, \cdots, n$.

for $|x_1a_2| = |y_1b_2| \le i + 1$, then $[t_j(x_1a_2|y_1b_2)] \in \tau_A(I_{i+1})$.

Thus $\tau_A(I_{i+2}) \subset \tau_A(I_{i+1}) = \tau_A(I_i)$.

(ii) If $\tau_A(I_i) = \tau_A(I_{i+1})$, the result is obvious. If $\tau_A(I_i) \subset \tau_A(I_{i+1})$, the result is also true. Indeed:

If $\tau_A(I_{i+1})$ contains only one new vector different from $\tau_A(I_i)$, then we have $\tau_A(I_0) \subset \tau_A(I_1) \subset \cdots \subset \tau_A(I_{k^{n^2}-1}) = \tau_A(I_{k^{n^2}}) = \cdots = \tau_A(I)$; if $\tau_A(I_{i+1})$ contains more than one new vector different from $\tau_A(I_i)$, then there exists $j \le k^{n^2} - 1$ such that $\tau_A(I_0) \subset \tau_A(I_1) \subset \cdots \subset \tau_A(I_j) = \cdots = \tau_A(I_{k^{n^2}-1}) = \tau_A(I_{k^{n^2}}) = \cdots = \tau_A(I)$.

Now we can compute $\tau_A(I)$ in the following algorithm by Theorem 3.1 and Theorem 3.2, where L is a finite lattice-ordered monoid.

Now we can compute $\tau_A(I)$ in the following algorithm by Theorem 3.1 and Theorem 3.2, where L is a finite lattice-ordered monoid.

Algorithm 1

*Step*1: We obtain $\tau_A(I_1)$ by $E_{n\times 1}$ and $\delta(a_1|b_1) \circ E_{n\times 1}, \cdots, \delta(a_1|b_l) \circ E_{n\times 1}, \delta(a_2|b_1) \circ E_{n\times 1}, \cdots, \delta(a_m|b_l) \circ E_{n\times 1}$;

*Step*2: For $i = 2$ to k^{n^2}

Compute $\delta^*(a_1a_2\cdots a_i|b_1b_2\cdots b_i)$ by $\delta^*(a_1a_2\cdots a_i|b_1b_2\cdots b_i) = \delta(a_1|b_1) \circ \delta(a_2|b_2) \circ \cdots \circ \delta(a_i|b_i)$;

We obtain $\tau_A(I_2), \cdots, \tau_A(I_{k^{n^2}})$ by the same way as *Step*1.

*Step*3: Output $\tau_A(I) = \tau_A(I_{k^{n^2}})$.

4 MINIMIZATION OF AN L-FSM

At first, two kinds of statewise equivalence relations can be introduced.

Definition 4.1 *Let $A = (Q, X, Y, \delta)$ and $A' = (Q', X, Y, \delta')$ be two L-FSMs with behavior mapping τ_A and $\tau_{A'}$, respectively. Let $q \in Q$ and $q' \in Q'$, then:*

(i) q and q' are equivalent ($q \equiv q'$) $\Longleftrightarrow$ the rows corresponding to the state q in $\tau_A(I)$ and the state q' in $\tau_{A'}(I)$ are identical.

(ii) for each positive integer k, q and q' are $k-$equivalent ($q \equiv_k q'$) $\Longleftrightarrow$ the rows corresponding to the state q in $\tau_A(I_k)$ and the state q' in $\tau_{A'}(I_k)$ are identical.

(iii) A and A' are equivalent ($A \equiv A'$) $\Longleftrightarrow \forall q \in Q, \exists q' \in Q'$ such that $q \equiv q'$ and $\forall q' \in Q', \exists q \in Q$ such that $q' \equiv q$.

(iv) for each positive integer k, A and A' are $k-$equivalent ($A \equiv_k A'$) $\Longleftrightarrow \forall q \in Q, \exists q' \in Q'$ such that $q \equiv_k q'$ and $\forall q' \in Q', \exists q \in Q$ such that $q' \equiv_k q$.

We denote the partition corresponding to $\equiv$ and $\equiv_k$ by $Q/\equiv$ and $Q/\equiv_k$ respectively. By Definition 3.1 and 4.1 we have $\tau_A = \tau_{A'} \Longrightarrow A \equiv A'$.

Theorem 4.1 *Let $A = (Q, X, Y, \delta)$ be an L-FSM. Then there exists a minimal L-FSM A' equivalence to A.*

Definition 4.2 *Let $A = (Q, X, Y, \delta)$ be an L-FSM with behavior mapping τ_A. Then A is said to be minimal $\Longleftrightarrow$ any two rows in $\tau_A(I)$ are not identical.*

Next, we shall prove the existence of the minimal L-FSM.

Proof: If any two rows in $\tau_A(I)$ are not identical, the claim is true by Definition 4.2, of which $A' = A$.

We can get a minimal L-FSM A' equivalent to A by the above constructed method.

Theorem 4.2 *Let $(L, \bullet, e)$ be a finite ordered-monoid without nilfactor and $A = (Q, X, Y, \delta)$ be an L-FSM with τ_A, where $|L| = k$ and $|Q| = n$. Then there exists at most k^n steps to distinguish all states in Q.*

Proof: Since $\tau_A(I)$ is a semi-infinite matrix with n rows and numberable columns, and $L = \{u_1, u_2, \cdots, u_k\}$ is the set of all the elements which appear in each column vector from $\tau_A(I)$, thus the number of different columns which can be obtained in $\tau_A(I)$ is at most k^n.

Now we can obtain $Q/\equiv$ in the following minimization algorithm for A by Theorem 4.1 and 4.2.

Algorithm 2

*Step*1: The equivalence class $\equiv_1$ can be obtained by $\tau_A(I_1)$;

*Step*2: Repeat for $i = 2, 3, \cdots$

Compute $\tau_A(I_2), \tau_A(I_3), \cdots$ and obtain equivalence classes $Q/\equiv_2, Q/\equiv_3, \cdots$.

Until $|Q/\equiv_i| = n$ or $i = k^n$.

*Step*3: Output $Q/\equiv_i = Q/\equiv$. i.e. $Q/\equiv$ is the equivalence class of the L-FSM A.

Example. Let $L = (\{0, 0.3, 0.8, 1\}, \wedge, \vee)$ with $e = 1, \bullet = \wedge$ and $A = (Q, X, Y, \delta)$ be an L-FSM with $Q = \{q_1, q_2, q_3\}, X = \{0\}, Y = \{0, 1\}$. A is given by

$$\delta(0|0) = \begin{pmatrix} 1 & 0 & 0.3 \\ 1 & 0.8 & 0 \\ 0.3 & 0 & 0 \end{pmatrix},$$

$$\delta(0|1) = \begin{pmatrix} 0 & 1 & 0.3 \\ 0.8 & 0 & 1 \\ 0 & 0 & 0.8 \end{pmatrix}.$$

Then $\tau_A(0|0) = \begin{pmatrix} 1 \\ 1 \\ 0.3 \end{pmatrix}$ and $\tau_A(0|1) = \begin{pmatrix} 1 \\ 1 \\ 0.8 \end{pmatrix}$,

so $\tau_A(I_1) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0.3 & 0.8 \end{pmatrix}$.

*Step*1: We can see $Q/\equiv_1 = \{\{q_1, q_2\}, \{q_3\}\}$ by $\tau_A(I_1)$;

*Step*2: Compute $\tau_A(I_2)$:

$$\tau_A(00|00) = \begin{pmatrix} 1 \\ 1 \\ 0.3 \end{pmatrix}, \tau_A(00|01) = \begin{pmatrix} 1 \\ 1 \\ 0.3 \end{pmatrix},$$

We can obtain

$$\tau_A(I_2) = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0.8 & 0.8 \\ 1 & 0.3 & 0.8 & 0.3 & 0.3 & 0.3 & 0.8 \end{pmatrix}$$ and

know $Q/\equiv_2 = \{\{q_1\}, \{q_2\}, \{q_3\}\}$, stop.

*Step*3: Output $Q/\equiv = Q/\equiv_2 = \{\{q_1\}, \{q_2\}, \{q_3\}\}$.

$$\tau_A(00|10) = \begin{pmatrix} 1 \\ 0.8 \\ 0.3 \end{pmatrix}, \tau_A(00|11) = \begin{pmatrix} 1 \\ 0.8 \\ 0.8 \end{pmatrix},$$

5. CONCLUSIONS

In this paper, we use a generalized truth values algebraic structure–finite lattice-ordered monoid as the basic structure of membership values and formulate the definition of behavior mapping to determine the whole operation of an L-valued finite sequential machine (L-FSM).

The concept of L-FSM in [1] has been modified, next, we have discussed the related properties. Furthermore, we have presented Algorithm 1 to obtain the image of the behavior mapping for a given L-FSM. In the sequel, the statewise equivalence relations have been given, in order to obtain the minimal L-FSM, we has got Algorithm 2 within finite steps.

Acknowledgements

This work is supported by National Natural Science Foundation of China (No.11071178) and National Science Foundation of China (No. 61203285).

References

[1] Xing HY, Qiu DW, Liu FC, and Fan ZJ, Equivalence in automata theory based on complete residuated lattice-valued logic, *Fuzzy Set and Systems*, Vol. 158, 1407-1422,2007.

[2] Santos ES, On reduction of maxi-min machines, *J.Math.Anal.Appl*, Vol. 40, 60-78, 1972.

[3] Santos ES, Max-product machines, *J.Math.Anal.Appl*, Vol. 37, 667-686, 1972.

[4] Qiu DW, Automata theory based on complete residuated lattice-valued logic(I), *Sci. China Ser.F44*,Vol. 6, 419-429,2001.

[5] Qiu DW, Automata theory based on complete residuated lattice-valued logic(II), *Sci. China Ser.F45*,Vol. 6, 442-452,2002.

[6] Li YM , and Pedrycz W, Fuzzy finite automata and fuzzy regular expressions with membership values in lattice-ordered monoids, *Fuzzy Sets and Systems*, Vol. 156, 68-92, 2005.

[7] Lei HX, and Li YM, Minimization of states in automata theory based on finite lattice-ordered monoids, *Information Science* ,doi: 10.1016/j.ins.2006.07.015.

[8] Cheng W, and Mo ZW, Minimization algorithm of fuzzy finite automata, *Fuzzy Sets and Systems*, Vol. 141, 439-448, 2004.