© 2015 IJEDR | Volume 3, Issue 4 | ISSN: 2321-9939

Family of lattice valued Aleshin type finite state automata

[1] Dr.A.Jeyanthi, [2] B.Stalin
[1] Faculty, [2] Assistant Professor
[1] Department of Mathematics, [2] Department of Mechanical Engineering
[1, 2] Anna University, Regional Office Madurai,Tamilnadu, India

***Abstract* - In this paper, we study Aleshin type finite automata with membership values in a lattice, which are called lattice-valued Aleshin Type finite automata. The extended subset construction of lattice-valued finite automata is introduced, then the equivalences between lattice-valued Aleshin type finite automata, lattice-valued deterministic Aleshin type finite automata and lattice-valued Aleshin type finite automata with ε-moves are proved. A simple characterization of lattice-valued languages recognized by lattice-valued Aleshin type finite automata.**

***Index Terms* - Aleshin type finite automata, lattice-valued deterministic Aleshin type finite automata.**

I. INTRODUCTION

The concept of fuzzy automata was introduced in the very early age of fuzzy set theory [7,16,23,32]. Since finite automata constitute a mathematical model of computation, fuzzy finite automata may be considered as an extended model which includes notions like ‘‘vagueness’’ and ‘‘imprecision’’, i.e., notions frequently encountered in the study of natural languages. So investigating fuzzy finite-state automata might reduce the gap between formal languages as studied in classical automata theory [8] on the one hand and natural languages on the other hand. Usually, fuzzy automata took values in the unit interval [0,1]. To enhance the processing ability of fuzzy automata, the membership grades were extended to much general algebraic structures. For example, automata theory based on complete residual lattices-valued logic has been primarily established in [24–26], and automata theory based on lattice-ordered monoids has been established in [19]. In fact, fuzzy finite automata could be considered as a special instance of weighted automata considered recently [4,13,31,9], where weighted automata take values in semirings, more general structures than lattice-ordered monoids when a lattice-ordered monoid is considered as an idempotent semiring [13]. Of course, there are some distinct differences between fuzzy automata and weighted automata. In fuzzy automata, it is stressed the role of order (or hierarchy) and fuzzy logic in automata theory, while, for weighted automata, it is emphasized the weight of source used in automata theory.

We can use fuzzy rules to represent fuzzy automata [6], and fuzzy automata can be seen as a special kind of discrete fuzzy systems [18]. While weighted automata can be seen as the algebraic treatment of automata using semiring and formal power series [5,14]. When we use a lattice as the truth value domain of fuzzy automata or as a semiring involved in weighted automata, this lattice should satisfy distributive law. On the other hand, there are many kinds of lattices that are not distributive lattices. The lattices M_3 (the diamond) and N_5 (the pentagon) in Fig. 1 are two simple examples. The free lattices generated by more than two elements are the other instances. There are some deep reasons to study such automata. First, this work may provide some useful ideas and more general framework to study automata theory based on quantum logic [33, 27–29]. As we know, automata based on quantum logic used orthomodular lattices [11] as the truth values of the automata considered. In general, an orthomodular lattice as a lattice needs not to be a distributive lattice. Indeed, automata theory based on quantum logic, which maybe thought of as a logic foundation for quantum computing and quantum information, has some essential differ-ence with classical automata and forms an important direction in the study of quantum computation, see [33,27,28] for the detailed explanation. In an orthomodular lattice, orthomodular laws hold instead of distributive laws, the Chinese lantern MO2 in Fig. 1 is one of such examples. For any Hilbert space H, the closed subspaces of H under the order of subset inclusion forms the standard orthomodular lattice which can be considered as the logic of quantum mechanics, see [2,11] for the de-tailed explanation. However, there are many (ortho-) lattices which do not satisfy the orthomodular laws. The simplest one is the lattice O_6 as in Fig. 1. We want to know how about the behaviors of finite automata with membership values in a gen-eral lattice. In this paper, we will use lattices as the structures of truth values domain. We do not require the used lattice to be distributive, so the used lattice needs not to be a semiring. We would obtain much more general results than fuzzy finite automata and weighted automata. Second, these results may be useful in multi-valued model checking. Some related work has been done using multi-valued automata in Ref. [3,15], where the multivalued domain is chosen as a De Morgan distributive lattice. We shall extend multi-valued domain to a general lattice.

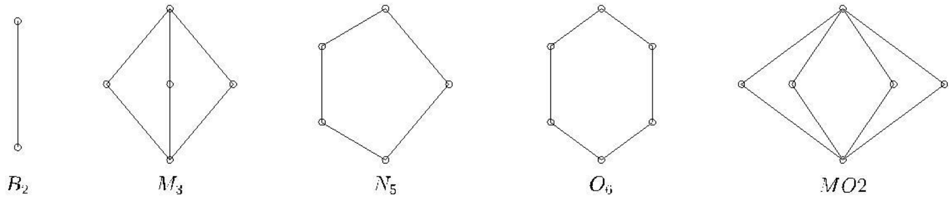

Fig. 1 Lattice diagram

The contribution of this study contains at least three aspects. First, as we just said, lattice-valued Aleshin type finite automata in this study are a common generalization of fuzzy automata and weighted automata. In this respective, the role of the distributive law for the truth valued domain of finite automata is analyzed. It is demonstrated that the distributive law is not necessary to many constructions of lattice-valued finite automata, but it indeed provides some convenience in simply processing lattice-valued finite automata. Second, the technique of extended subset construction is introduced, using this technique, the equivalence between lattice-valued Aleshin type finite automata and lattice-valued deterministic finite automata is proved.

II. FINITE AUTOMATA WITH MEMBERSHIP VALUES IN A LATTICE *L*

A lattice is a 6-tuple $l = (L, \leq, \wedge, 0,1)$, 0 and 1 are the least and largest elements of L, respectively, $\leq$ is the partial ordering in L; and for any a, b $\in$ L, a $\wedge$ b and a $\vee$ b stand for the greatest lower bound (or meet) and the least upper bound (or join) of a and b, respectively. We need lattice-valued logic (called l-valued logic in the paper, and it is indeed the geometric logic as studied in [2], which is to provide algebraic semantics for observable events) to represent automata in this paper. We define it in the following way. Since there is no negation operation and implication operation in a general lattice, we do not mention the negation connective and implication connective in lattice-valued logic. Similar to that of classical first-order logic, the syntax of l-valued logic has two primitive connectives $\wedge$ (conjunction), $\vee$ (disjunction), and two primitive quantifier $\exists$ (existential quantifier) and $\forall$ (universal quantifier). In addition, we need to use some set-theoretical formulas. Let $\in$ (mem-bership) be a binary (primitive) predicate symbol. Then $\subseteq$ and $\equiv$ (equality) can be defined with $\in$ as usual.

The semantics of *l*-valued logic is given by interpreting the connectives $\wedge$ and $\vee$ as the operations $\wedge$ and $\vee$ on *l*, respectively, and interpreting the quantifier $\exists$ and $\forall$ as the least upper bound and the greatest lower bound in l. Moreover, the truth value of set-theoretical formula $x \in A$ is $[x \in A] = A(x)$. In the *l*-valued logic, 1 is the unique designated truth value; a formula u is valid iff $[\varphi] = 1$, and denoted by $_{l}\varphi$. In order to distinguish the symbols representing languages and the symbols representing lattices, we use symbol *l* to represent a lattice, and use L to represent language. We use the symbols a, b, c, d, k to represent the elements of *l*.

Construction of 4-state Aleshin type automaton, A(S)

The Aleshin automaton is nothing but the complement of output function of Bellaterra automaton. The corresponding Aleshin automaton is as follows.

The constructed Aleshin type automaton S, [A(S)] over the alphabet X= {0, 1} with the set of internal states Q ={a, b, c, d}. The state transition function φ and the output function ψ of A(S) are defined as follows:

$$\delta(a,0)=d,\delta(a,1)=b,\delta(b,0)=b,\delta(b,1)=c,\delta(c,0)=d,\delta(c,1)=d,\delta(d,0)=a,\delta(d,1)=a;$$
$$\psi(a,0)=1,\psi(a,1)=0,\psi(b,0)=1,\psi(b,1)=0,\psi(c,0)=0,\psi(c,1)=1,\psi(d,0)=0,\psi(d,1)=1$$

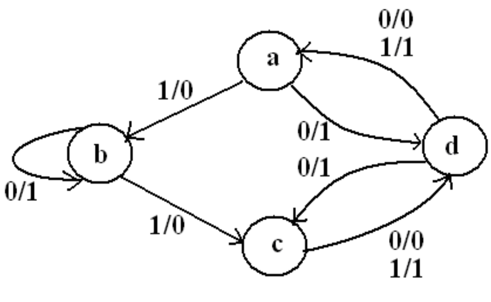

Fig. 2 Aleshin type automaton

Definition

An *l*-valued Aleshin finite automaton (*l*-VFA for short) is a 5-tuple A = (Q, Σ, δ, I, F), where Q denotes a finite set of states, Σ a finite input alphabet, and δ,an *l*-valued subset of Q x ΣxQ; that is, a mapping from Q xΣ xQ into *l*, and it is called the *l*-valued transition relation. Intuitively, δ is an *l*-valued (ternary) predicate over Q, Σ and Q, and for any p, q $\in$ Q and r $\in$Σ, d(p, r, q) stands for the truth value of the proposition that input r causes state p to become q. I and F are *l*-valued subsets of Q; that is, a mapping from Q into l, which represent the initial state and final state, respectively. For each q $\in$ Q, I(q) indicates the truth value (in the underlying lattice-valued logic) of the proposition that q is an initial state, F(q) expresses the truth value of the proposition that q is a finial state.

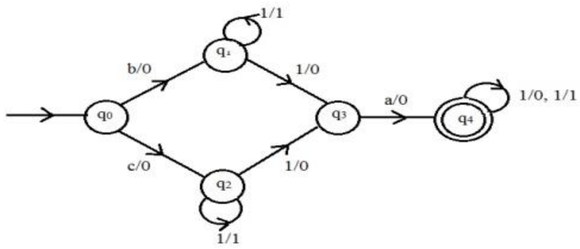

Fig. 3 Lattice valued Aleshin type automaton

The propositions of the form

"q is an initial state " written $q \in I$, q is a final state , written $q \in F$,

input r causes state q to become p, according to the specification given by δ, written by $(q,r,p) \in \delta$

Denote the atomic propositions in our logical languages designated for describing *l*-valued Aleshin type automaton A. The truth values of the above three propositions are respectively I(q), F(q) and $\delta(q, r, p)$. We use the symbols r, s to represent the elements in Σ, use the symbols x, h to denote the strings over Σ, and use e to represent the empty string over Σ. Let Σ^* denote the set of all strings over Σ. We use the symbols A; B to denote the l-valued finite automata.

For an *l*-VAFA A, the *l*-valued unary recognizability predicate rec_A over Σ^* is defined as a mapping from Σ^* into l: for each $x \in \Sigma^*$, let $x = r_1 \ldots\ldots r_n$ for some $n \geq 0$,

$$x \in rec_A =_{def} (\exists q_0 \in Q) \ldots. (\exists q_n \in Q) \ldots..(q_0 \in I \wedge q_n \in F \wedge (q_0, r_1, q_1) \in \delta \wedge \ldots\ldots \wedge (q_{n-1}, r_n, q_n) \in \delta$$

In other words, the truth value of the proposition that x is recognizable by A is given by

$$rec_A(x) = \vee \quad \{I(q_0) \wedge \delta(q_0, r_1, q_1) \wedge \ldots. \wedge \ \delta(q_{n\_1}, r_n, q_n) \wedge F(q_n) : q_0, \ldots, q_n \in Q\}$$

We call rec_A the *l*-valued language recognized or accepted by *l*-VAFA A. We use $l(\Sigma^*)$ to denote the set of all *l*-valued languages over Σ^*, which is an *l*-valued subset of Σ^*; that is, a mapping from Σ^* to *l*. For an $A \in l(\Sigma^*)$, if there is an *l*-VAFA A such that A = rec_A, then we call A an *l*-valued regular language or *l*-regular language on R, which is also called lattice-valued regular language without mentioning the truth-valued lattice.

We give an example to illustrate the definition of *l*-VAFA.

Example

Let l be the lattice N_s as shown in Fig. 1. Write the elements of l as 0, 1, a, b, c, where b < a and c cannot be comparable with a and b. An l-VFA A = (Q , Σ, δ, I, F) is defined as follows, where Q = $\{q_0, q_1, q_2, q_3, q_4\}$, $\Sigma = \{0,1\}$, $\delta(q_0,0, q_1) = b$, $\delta(q_0,0, q_2) = c$, $\delta(q_1, 1,q_1) = 1$,

$$\delta(q_2,0, q_2) = 1,\ \delta(q_1,0, q_3) = l,\ \delta(q_3,0, q_4) = a,\ \delta(q_4, 0, q_4) = 1 ,\ \delta(q_4, l , q_4) = l \qquad (1)$$

and I $= \frac{l}{q0}$, F $= \frac{l}{q4}$ This l-VFA is represented as in Fig. 2, where we use $\Sigma_{t\in T} \frac{at}{xt}$ to denote

the *l*- valued subset with nonzero a_t, at the element x_t , for $t \in T$ and T is an index set and if $\delta(q, r,p) = \sigma$ then there is an edge

label from the node with name q to the node p in the graphic representation of the *l*- valued VFA A. By the simple calculations, we can see that

$rec_A(x) = (a \wedge b) \vee (a \wedge c) =$ b if $x \in 01^*00(0 + 1)^*$, and $rec_A(x) = 0$ in the other cases.

The extension of d, denoted δ^*, is defined as follows,

(i) $\forall\, p \in Q$, if p=q, then $\delta^*(q,\varepsilon, p) = 1$, otherwise $\delta^*(q,\varepsilon,p) = 0$

(ii) $\forall \theta = r_1 \ldots\ldots r_n \in \Sigma^*$, $\delta^*(q, r_1 \ldots\ldots r_n, p) = \vee$ $\{\delta(q , r_1 , q_1) \wedge \ldots. \wedge \delta(q_{n-1}, r_n, q_n) : q_1 \ldots\ldots q_{n-1} \in Q\}$

As far as the extension δ^* is concerned , for any $\theta \in \Sigma^*$, if $\theta = \theta_1 \theta_2$, then it wish to satisfy the following equations

$$\delta^*(q , \theta_1 \theta_2 ,p) = \vee_{\sigma\in Q}[\delta^*(q, \theta_1,\sigma) \wedge \delta^*(\sigma,\theta_2 ,p)] \qquad (2)$$

$$a \wedge (b \vee c) = (a \vee b) \vee (a \wedge c); \quad a \vee (b \wedge c) = (a \wedge b) \wedge (a \vee c) \qquad (3)$$

Proposition 1

The following conditions are equivalent:

(i) l is a distributive lattice.

(ii) For any l-VAFA, A = (Q , Σ, δ, I, F), and for any $p,q \in Q$, $\theta_1, \theta_2 \in \Sigma^*$, the Eq. (2) holds, i.e., $\delta^*(q , \theta_1 \theta_2 ,p) = \vee_{\sigma\in Q}[\delta^*(q, \theta_1,\sigma) \wedge \delta^*(\sigma,\theta_2 ,p)]$

The related result similar to Proposition 2.1 is also presented in [19,24], and the proofs are very similar, we omit the proof here.

Let us define the operations of l-valued languages (c.f. [19,21,33]): for A, B $\in$ l(Σ^*) and $\sigma \in$ l, the union A $\vee$ B, the intersec-tion A $\wedge$ B, the scalar product σA, the concatenation AB, the Kleene closure A* are defined as follows: for any x $\in \Sigma^*$,

$$A\vee B(x) = A(x)\vee B(x), A\wedge B(x) = A(x)\wedge B(x),\ \sigma A(x) = \sigma\wedge A(x)\ ,\ AB(x) = \vee\ \ A(x_1)\wedge B(x_2){:}x_1 x_2 = x,$$
$$A^*(x) = \vee\ \{ A(x_1)\wedge\ldots A(x_n){:}\ n\geq 0,\ x_1\ldots\ldots x_n = x\}$$

The following proposition claims the relationship between the associativity of concatenation operation and the distributive of lattice l.

Proposition 2

The following statements are equivalent.

(i) *l* is a distributive lattice.

(ii) For any A,B,C $\in$ l(Σ^*), (AB)C = A(BC).

Proof. (i)) (ii) is obvious. Conversely, for any a, b, c $\in$ l, we want to prove the distributive law a $\wedge$(b $\vee$ c) = (a $\wedge$b) (a $\wedge$ c) holds. For r $\in \Sigma$, we take A, B, C $\in$ l(Σ^*) as follows,

$$A(x) = \begin{cases} b, \text{ if } x = r \\ 1, \text{ if } x = \varepsilon, \\ 0, \text{ otherwise} \end{cases} \qquad B(x) = \begin{cases} c, \text{ if } x = r \\ 1, \text{ if } x = \varepsilon \\ 0, \text{ otherwise} \end{cases} \qquad C(x) = \begin{cases} a, \text{ if } x = r \\ 0, \text{ otherwise} \end{cases}$$

That is, A = $\frac{l}{\varepsilon} + \frac{b}{r}$, B = $\frac{l}{\varepsilon} + \frac{c}{r}$ and C = $\frac{a}{r}$, Then (AB)C(rr) = AB(r)$\wedge$a = (A(r)$\wedge$B(ε))$\vee$(A(ε)$\wedge$B(r))$\wedge$a = ((b$\wedge$l)$\vee$(l$\wedge$c))$\wedge$a = a$\wedge$(b$\vee$c), since (AB)C(rr) = (A(r) $\wedge$BC(r))$\vee$(A(ε) $\wedge$BC(rr)) = (b$\wedge$a)$\vee$(l$\wedge$B(r) $\wedge$C(r)) = (b$\wedge$a)$\vee$(c$\wedge$a) = (a$\wedge$b) $\vee$ (a$\vee$c), by associativity law, we have (AB)C = A(BC). It follows that the distributive law a $\wedge$(b $\vee$ c) = (a $\wedge$b) (a $\wedge$ c) holds.

III. Determinization of *L*-valued Aleshin type finite automata and extended subset constructions

First, we show that the image set of each lattice-valued regular langauge is always a finite set of *l*.

Lemma [17]

Let *l* be a lattice, and X a finite subset of *l*. Then the $\wedge$-semilattice of l generated by X, written as $X_\wedge$, is finite, the $\vee$-semilattice of l generated by X, denoted $X_\vee$, is also finite, where $X_\wedge = \{x_1 \wedge\ldots\ldots\wedge x_k{:}k\ \geq 1, x_1, \ldots, x_k \in X\}\cup\ \{1\}$, and X_$_\vee$ = $\{x_{1\vee}\ldots\ldots x_k{:}k \geq 1, x_1, \ldots, x_k \in X\} \cup\{0\}$.

Proposition

Let A = (Q, Σ, δ, I, F) be an *l*-VFA. Then the image set of the *l*-valued language rec_A, as a mapping from Σ^* to *l*, is finite; that is, the subset Im(rec_A)= r $\in$ *l* : $\exists$x $\in \Sigma^*$, rec_A (x) = σ of *l* is finite.

Proof

For any x = $r_1\ldots\ldots r_k \in\Sigma^*$, observing that $rec_A(x) = \vee[I(q_0 \wedge\delta(q_0, \sigma_1, q_1) \wedge\ldots\ldots \wedge\delta(q_{k-1}, \sigma_k, q_k) \wedge F(q_k) : q_0; \ldots ; q_k \in Q]$

On input x = $r_1\ldots\ldots r_k \in \Sigma^*$, there are only finite accepting paths, assumed as m, causing an initial state $q_0 \in$ I to become a final state $q_k \in$ F. For the i-th accepting path, we let $a_{i0} = I(q_0)$, $a_{i1} = \delta(q_0, \sigma_1, q_1), \ldots, a_{ik} = \delta(q_{k-1}, \sigma_k, q_k)$ and $a_{i,k+1} = F(q_k)$. Then the truth value of $rec_A(x)$ can be calculated as, $rec_A(x)= (a_{10}\wedge\ldots\ldots\wedge a_{1k} \wedge a_{1;,+1})\vee\ldots\ldots\vee(a_{m0} \wedge\ldots\ldots\wedge a_{mk} \wedge a_{m,k+1})$. Let X = Im($\delta$) U Im(I) U Im(F), then X is obviously a finite subset of *l* and $a_{ij} \in$ X for any 1 $\leq$ i $\leq$ m and 0 $\leq$ j $\leq$ k + 1. For any x $\in \Sigma^*$, by the above observation, it follows that $rec_A(x)\in (X_\wedge)_\vee$, so Im($rec_A$)$\subseteq (X_\wedge)_\vee$. By Lemma 3.1, $(X_\wedge)_\vee$, is a finite subset of l, and thus Im(rec_A), as a subset of $(X_\wedge)_\vee$, is also a finite subset of *l*.

Due to Proposition, for any *l*-VFA, the image set of its recognizable lattice-valued language is always finite. Then we have the following observation: the lattice *l* may be infinite as a set, but for a given *l*-VFA A, only a finite subset of *l* is employed in the operating of A. This observation is the core in the introducing of extended subset construction in this section.

The notion of nondeterminism plays a central role in the theory of computation. The nondeterministic mechanism enables a device to change its states in a way that is only partially determined by the current state and the input symbol. The concept of *l*-VFA is obviously a generalization of nondeterministic finite automaton (NFA for short). In classical theory of automata, each nondeterministic finite automaton is equivalent to a deterministic one; more precisely, there exists a deterministic finite automaton (DFA for short) which accepts the same language as the originally given nondeterministic one does. The construction of DFA from an NFA is the well-known subset construction introduced by Rabin and Scott [30]. With respect to the case of *l*-VFA, the situation is more complex. In fact, as shown in [33], the subset construction does not work well for *l*-VFA even for an orthomodular lattice *l*. That is, for an *l*-VFA A, one can construct an l-valued deterministic finite automaton B, as defined in [33] using the modified subset construction. However, B is not necessarily equivalent to A, i.e., the equality rec_A = rec_B does not hold in general. Some conditions that guarantee the equivalence between A and B are given in [33]. Therefore, it is an open problem whether an *l*-VFA can always be determinizable. We shall show that the answer is affirmative for a general lattice *l*. We shall introduce subset construction in lattice setting which we call it the extended subset construction. First, we define the notion of deterministic *l*-VFA.

Definition [19]

An *l*-valued deterministic finite automaton (*l*-VDFA for short) is a 5-tuple A = (Q , Σ, δ, q_0, F), where Q, Σ and F are the same as those in an *l*-valued automaton, $q_0 \in Q$ is the initial state, and the lattice-valued transition relation δ is crisp and deterministic; that is, δ is a mapping from Q xΣ into Q.

Note that our definition differs from the usual definition of a deterministic finite automaton only in that the final states form an *l*-valued subset of Q. This, however, makes it possible to accept words to certain truth degrees (in the lattice setting), and thus to recognize lattice-valued languages.

For an *l*-VDFA, A = (Q , Σ, δ, q_0 F), its corresponding *l*-valued recognizability predicate $rec_A \in l(\Sigma^*)$ is defined as: for all x = $r_1 \ldots\ldots r_n \in \Sigma^*$,

$x \in recA = def\ (\exists q_1 \in Q)\ldots\ldots(\exists q_n \in Q).(q_n \in F \wedge \delta(q_0, r_1) = q_1 \wedge \ldots\ldots \wedge \delta(q_{n-1}, r_n) = q_n)$

Write δ* the extension of transition relation δ by putting δ*(q, ε) = q and δ*(q, xr) = δ(δ*(q, x),r) for any $q \in Q$ and $x \in \Sigma^*$

and $r \in \Sigma$, then the truth value of the proposition $x \in recA$ is given by, recA(x) = F(δ*(q_0,x)). For any l-VFA, A =(Q , Σ, δ, q_0, F), we now introduce the extended subset construction an equivalent l-VDFA, A^d = =(Q^d , Σ, η, S, E) from A.

Let X = Im(δ)∪ Im(I)∪ Im(F), then X is clearly a finite subset of l. Let l_1 = X∧.By lemma, l_1 is a semi lattice of l generated by X and also finite subset of *l*. Choose

$Q^d = 2^{Qx(l_1-(0))}$ where $2^{Qx(l_1-(0))}$ denotes the set of all subsets of Qx{ l_1- 0}. Then Q^d is clearly a finite set. Take S= {(q,I(q)): q∈Q and I(q)≠0} then S∈ Qd . The state transition η: Q^d x Σ→ Q^d is defined as , for any (q,σ)∈ Qx l_1- {0} and σ∈Σ,

η((q,σ),r) = { (p,δ(q,r,p)∧σ):p∈Q and δ(q, r,p) ∧σ≠0 and for Z∈ Q^d}

η(Z,r) = ∪ {η(q,σ), r): (q,σ)∈Z } . By the definition of l_1 , l_1 is closed under finite meet operation, for any a,b,∈ l_1 , a∧b ∈ l_1 , follows that , for any σ∈ l_1 and foe any (p,r,q)∈QxΣxQ , σ∧δ(p,r,q)∈ l_1 , and thus η((q,σ , r)∈ Q^d for any (q,r)∈ Qx l_1-{0}.Then the mapping η is well defined. The *l*- valued final state E: Q^d→ *l* is defined for any Z∈ Q^d , E(Z) = **V** {σ∧F(q) : (q,σ)∈Z}. Then A^d is an *l*-VDFA.

Theorem

For any *l*-VAFA, A = (Q, Σ,δ, I, F), the *l*-VDFA A^d = (Q^d, Σ, η,S,E) constructed above is equivalent to A, i.e., $rec_A = rec_A{}^d$.

Proof

To prove that by induction on the length |x| of input string x that

η* (S,x) = (q_n, I(q_0)∧δ(q_0,r_1, q_1)∧……∧δ(q_{n-1}, r_n, ,q_n)): q_0,…… q_n∈Q and

σ_n = I(q_0)∧δ(q_0,r_1, q_1)∧……∧δ(q_{n-1}, r_n, ,q_n) ≠0 } where x = r_1, ……r_n for n≥0. The result is trivial for |x| =0, since x =ε and

η*(S,x) = {(q_0, I(q_0)): q_0∈Q and I(q_0)≠0} .

Suppose that the hypothesis is true for inputs of length n or less. Let x = r_1, ……r_{n+1} be a string of length n+1,

write y = r_1, ……r_n, then x =y r_{n+1}, then

η*(S, yr_{n+1}) = η(η*(S,y), yr_{n+1})

By the inductive hypothesis,

η*(S, y) = (q_n, I(q_0) ∧ δ(q_0, r_1, q_1) ∧…..∧δ(q_{n-1}, r_n, q_n)) : q_0, . . . ,q_n ∈ Q and r_n = = I(q_0) ∧ δ(q_0, r_1, q_1) ∧……∧δ(q_{n-1}, r_n, q_n) ≠0

By the definition of η,

$$\eta(\eta^*(S,y), r_{n+1}) = \bigcup_{(qn,rn)\in\eta^*(S,y)} \eta(\{(q_n,\sigma_n)\}, \sigma_{n+1}) = \bigcup_{(qn,rn)\in\eta^*(S,y)} \{(q_{n+1}, r_n \wedge \delta(q_n, \sigma_{n+1}, q_{n+1})) : q_{n+1}$$

∈Q and r_n ∧δ(q_n,σ_{n+1},q_{n+1})≠0 }={(q_{n+1},I(q_0)∧δ(q_0,σ_1,q_1)∧…. ∧δ(q_{n-1},σ_n, q_n) ∧δ(q_n,σ_{n+1},q_{n+1})) : q_0……… q_{n+1}∈Q and r_{n+1} =I(q_0)∧δ(q_0,σ_1,q_1)∧…∧δ(q_n,σ_{n+1},q_{n+1}) ≠0},

which establishes the inductive hypothesis.

By the definition of l-valued final state E, for any input ω =$\sigma_1 \ldots \sigma_n \in \Sigma^*$(n≥0), we have

$rec_A{}^d(\omega)$ = E(η^*(S, ω) = **V**{r_n∧F(q_n): (q_n,r_n) (η^*(S, ω)}={I(q_0, ,σ_1,q_1)… ∧δ(q_{n-1},σ_n, q_n) F(q_n) : q_0 ,…., q_n ∈ Q and I(q_0)∧δ(q_0,σ_1,q_1)∧…. ∧δ(q_{n-1},σ_n, q_n) ≠0}

=**V** {I(q_0))∧δ(q_0,σ_1,q_1)∧…. ∧δ(q_{n-1},σ_n, q_n) F(q_n) : q_0 ,…., q_n ∈ Q = rec_A(ω).

Thus $rec_A{}^d$ = rec_A A and A^d are equivalent.

IV. CONCLUSION AND FUTURE WORK

The notion of lattice-valued Aleshin type finite automata was introduced in this paper. It generalized the notion of fuzzy automata with membership values in a distributive lattice. Some general results were obtained here. In particular, finite automata under quantum logic could be considered as a special case of lattice-valued finite automata with orthomodular lattice as truth values domain. Some results in were strengthened here. In particular, the extended subset construction of lattice valued finite automata was introduced, then the equivalence between lattice-valued finite automata, lattice-valued deterministic finite automata and lattice-valued finite automata with e-moves was established. The extended subset construction and the determinization method considered in were compared. A simple characterization of lattice-valued languages recognized by lattice-valued finite automata was presented.

REFERENCES

[1] R. Beˇlohlavek, "Determinism and fuzzy automata," Information Sciences, vol.142, pp.205–209, 2002.

[2] G. Birkhoff, Lattice Theory, third ed., Amer. Math. Soc., Providence, Rhode Island, USA, 1973.

[3] M. Chechik, B. Devereux, A. Gurfinkel, S. Easterbrook, "Multi-valued symbolic model-checking," ACM Transactions on Software Engineering and Methodology, vol.12, Issue 4, pp. 1–38, 2004.

[4] M. Droste, P. Gastin, "Weighted automata and weighted logics," Theoretical Computer Science, vol.380, pp. 69–86, 2007.

[5] S. Eilenberg, Automata, Languages and Machines, vols. A, B, Academic Press, New York, 1974.

[6] C.A. Gama, A.G. Evsukoff, P. Weber, N.F.F. Ebecken, "Parameter identification of recurrent fuzzy systems with fuzzy finite-state automata representation," IEEE Transactions on Fuzzy Systems, vol.16, Issue1, pp.213–224, 2008.

[7] M.M. Gupta, G.N. Saridis, Fuzzy Automata and Decision Processes, North-Holland, New York, 1977.

[8] J.E. Hopcroft, J.D. Ullman, Introduction to Automata Theory, Languages and Computation, Addison-Wesley, New York, 1979.

[9] J. Ignjatovic, M. C´ iric´, "Formal power series and regular operations on fuzzy languages," Information Sciences, vol.180, Issue 7, pp.1104–1120, 2010.

[10] J. Ignjatovic´, M. C´ iric´ , S. Bogdanovic´ , Determinization of fuzzy automata with membership values in complete residuated lattices, Information Sciences 178 (1) (2008) 164–180.

[11] G. Kalmbach, Orthomodular Lattices, Academic Press, London, 1983.

[12] S.C. Kleene, Representation of events in nerve nets and finite automata, in: C.E. Shannon, J. McCarthy (Eds.), Automata Studies, Princeton University Press, Princeton, NJ, 1956, pp. 3–42.

[13] W. Kuich, G. Rahonis, "Fuzzy regular languages over finite and infinite words," Fuzzy Sets and Systems, vol.157, pp.1532–1549, 2006.

[14] W. Kuich, A. Salomaa, Semirings, "Automata Languages, EATCS Monographs," Theoretical Computer Science, vol. 5, Springer-Verlag, Berlin/Heidelberg/ New York/Tokyo, 1986.

[15] O. Kupferman, Y. Lustig, Lattice automata, in: Proceedings of VWCAI2007, LNCS, vol. 4349, pp. 199–213, 2007.

[16] E.T. Lee, L.A. Zadeh, "Note on fuzzy languages," Information Sciences, vol. 1, pp. 421–434, 1969.

[17] Y.M. Li, Z.H. Li, "Free semilattices and strongly free semilattices generated by partially ordered sets," Northeastern Mathematical Journal, vol. 9, Issue 3, pp. 359–366,1993.

[18] Y.M. Li, Analysis of Fuzzy Systems, Academic Press, Beijing, 2005 (in Chinese).

[19] Y.M. Li, W. Pedrycz, "Fuzzy finite automata and fuzzy regular expressions with membership values in lattice-ordered monoids," Fuzzy Sets and Systems, vol.156, pp. 68–92, 2005.