FUZZY GRAPHS IN FUZZY AUTOMATA MODEL

JITENDRA KUMAR JAISWAL*

ARINDAM DEY*

ANITA PAL*

Abstract

An automaton is a mathematical model of computing. It is an abstract model of digital computer. Classical automata are formal models of computing with values. Fuzzy automata are generalizations of classical automata where the knowledge about the systems next state is vague or uncertain. It is worth noting that like classical automata, fuzzy automata can only process strings of input symbols. Therefore, such fuzzy automata are still (abstract) devices for computing with values, although a certain vagueness or uncertainty is involved in the process of computation value. In this paper, we observe that, the fuzzy automata architecture is isomorphic to the fuzzy graph model and using fuzzy graph we will describe the three basic models of fuzzy automata.

Keywords: Fuzzy Graph, Fuzzy Automata, Classical automata

* Department of Mathematics, National Institute of Technology, Durgapur, West Bengal, India

1. Introduction

Finite automata are simple but important models of computation and have great practical importance, with significat applications such as in software engineering, lexical analysis, the description of natural languages and, the programming languages.

As the complexity of a system increases, our ability to make precise and yet significant statements about its behavior diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics. The real world is fuzzy, and fuzziness in the world generally arises from uncertainty in the form of ambiguity. Usually we accept the happening either with acceptability or for denies. In mathematical terms either 0 or 1, either yes or no. but we can't refuse with the fact that there are a lot of cases rather than crispness. For example if we say a person is tall. It is not precise to say only yes or no. it also matters what height he has and up to what height he can be verified him as a tall man.

Problems featuring complexity and ambiguity have been addressed subconsciously by humans since they could think. These ubiquitous features pervade most social, technical and economic problems faced by the human race. Why there are computers, which have been designed by humans', not capable of addressing complex and ambiguous issues. Humans have the capacity to reason approximately, a capability that computers currently do not have thoroughly. With the help of fuzzy logic we now try to make computers more intelligent in solving complexity and ambiguity. Due to this fuzzy automata play important role in today life.

We are first introduced with the notion of fuzzy finite automata. Afterwards, there were many authors such as Santos, Thomason et al., and Zadeh who have contributed to this field. Fuzzy finite automata have many important applications such as in learning systems, pattern recognition, and data base theory. In addition, they have been used to solve meaningful problems such as intelligent interface design, clinical monitoring and, neural networks.

2. Preliminaries

In section basic concepts of fuzzy set and fuzzy graph are discussed with finite automata.

Definition 2.1: A fuzzy set A defined on a non empty set X is the family $A = \{(x, \mu_A(x))/x \epsilon X\}$ where μ_A: $x \rightarrow I$ is the membership function. In fuzzy set theory the set I is usually defined as the interval [0, 1] such that $\mu_A(x) = 0$ if x does not belong to A, $\mu_A(x) = 1$ if x strictly belongs to A and for any intermediate value represents the degree in which x could belong to A, the set I can have any value between 0 and 1. $\mu_A(x) < \mu_A(x_1)$ indicates that the degree of membership of x to A is lower than the degree of membership of x_1.

Definition 2.2: Let us define three fuzzy sets $\tilde{A}$, $\tilde{B}$ and $\tilde{C}$ on the universe X. For a given element x of the universe,

Union: $\mu_{\tilde{A} \cup \tilde{B}}(x) = \mu_{\tilde{A}}(x) \vee \mu_{\tilde{B}}(x)$

Intersection: $\mu_{\tilde{A} \cap \tilde{B}}(x) = \mu_{\tilde{A}}(x) \wedge \mu_{\tilde{B}}(x)$

Complement: $\mu_{\tilde{\tilde{A}}}(x) = 1 - \mu_{\tilde{A}}(x)$

Where $\vee$ represents maximum and $\wedge$ represents minimum value.

Definition 2.3: Suppose the input universe is composed of the Cartesian product of many universes. The mapping f is defined on the power set of this universe as

f: $P(X_1 \times X_2 \times \cdots \times X_n) \rightarrow P(Y)$.

Let the fuzzy sets $A_1, A_2, \ldots, A_n$ be defined on $X_1, X_2, \ldots, X_n$, then $B = f(A_1, A_2, \ldots, A_n)$.

The membership function of B is defined as

$$\mu_B(y) = \max_{y = f(x1, x2, \ldots, xn)} \{\min [\mu_{A1}(x_1), \mu_{A2}(x_2), \ldots, \mu_{An}(x_n)]\}$$

This equation is usually called the Zadeh's extension principle.

Definition 2.4: α cut set of fuzzy set A is defined as A_α is made up of members whose membership value is not less than $\alpha A_\alpha = \{ x \in X \mid \mu_A(x) \geq \alpha. \}$. α cut set of fuzzy set is crisp set. In this paper, α cut set depend on vertex and edge membership value.

Definition 2.5: Blue et al. have given five types of graph fuzziness .Fuzzy graph is a graph G_F satisfying one of the following types of fuzziness (G_F of the i^{th} type) or any of its combination:

(i) $G_{F1} = \{G_1, G_2, G_3, \ldots, G_F\}$ where fuzziness is on each graph G_i.

(ii) $G_{F2} = \{V, E_F\}$ where the edge set is fuzzy.

iii) $G_{F3} = \{V, E(t_F, h_F)\}$ where both the vertex and edge sets are crisp, but the edges have fuzzy heads $h(e_i)$ and fuzzy tails $t(e_i)$

iv) $G_{F4} = \{V_F, E\}$ where the vertex set is fuzzy.

(v) $G_{F5} = \{V, E(w_F)\}$ where both the vertex and edge sets are crisp but the edges have fuzzy weights.

Let V be a finite nonempty set. The triple $G= (V, \sigma, \mu)$ is called a fuzzy graph on V where μ and σ are fuzzy sets on V and E $(V \times V)$, respectively, such that $\mu(\{u, v\}) \leq \min\{\sigma(u), \sigma(v)\}$ for all $u, v \in V$.

Note that a fuzzy graph is a generalization of crisp graph in which

$\mu(v) = 1$ for all $v \in V$

$=0$ otherwise

and $\rho(i, j) = 1$ if $(i, j) \in E$

$=0$ otherwise

so all the crisp graph are fuzzy graph but all fuzzy graph are not crisp graph.

Now we will define some basic concept of Automata models those are used in this paper.

A finite automaton has a set of finite states and its control moves from state to state in response to external inputs. It is called finite automata since it has finite number of states in its system. Finite automata are of two types, deterministic finite automata (DFA) and non-deterministic finite automata (NFA). In deterministic finite automata the control cannot be in more than one state at one time where the NFA has the additional choice of making a transition from one state

to another spontaneously, i.e. it may be in several states at once. Adding non-determinism does not let us define any language that cannot be defined by deterministic finite automata, but there can be substantial efficiency in describing an application using a non-deterministic automaton. In effect, non-determinism allows us to "program" solutions to problems using higher level languages. These automata accept nothing but regular languages.

Definition 2.6: A DFA can be analyzed by 5-tuple M= (Q, Σ, δ, q_0, F), where

(i) Q is the finite non-empty set of states

(ii) Σ is the finite set of input symbols

(iii)δ is the function which maps Q x Σ into Q and is usually called direct transition function, that takes as arguments a state and an initial symbol and returns a state.

(iv) $q_0 \in Q$, is the start state.

(v) $F \subseteq Q$ is the set of final states.

Definition 2.7: An NFA is a 5-tuple M= (Q, Σ, δ, q_0, F), where all symbols have their usual meanings as in DFA but δ differs as it is the transition function mapping from Q x Σ into 2^Q which is the power set of Q, the set of all subset of Q.

Using a crisp graph we can describe any automata. This type of graph is known as transition graph.

Definition2.8: A transition graph or a transition system is a finite directed labeled graph in which each vertex (or node) represents a state and the directed edges indicate the transition of a state and the edges are labeled with input/output.

A typical transition system is shown in fig.1. In the figure the initial state is represented by a circle with an arrow pointed towards it, final state by two concentric circles, and the other states are represented by just a circle. Edges are labeled by input/output (e.g. by 1/0 or 1/1).

For example , if the system is in state q_0 the input 1 is applied, the system moves to state q_1 as there is a directed from q_0 to q_1 with label 1/0. It outputs 0.

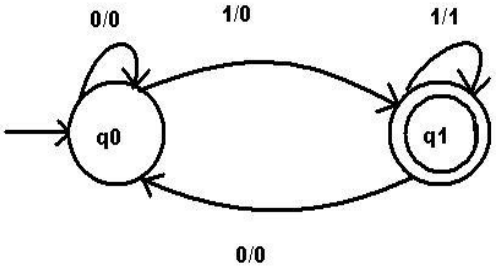

Fig.1: Finite Automata

3. Fuzzy-finite automata:

Definition 3.1:

A fuzzy-finite automaton has 5-tuple M= (Q, Σ, δ, q_0, F), where Q, Σ and q_0 are as in finite automaton and δ and F appear as follows:

(i). F is the fuzzy subset of Q, called the fuzzy set of final states and for each q Є Q, F (q) indicates intuitively the degree to which q is a final state.

(ii). δ is a transition function mapping from Q x Σ in F (Q), the set of all fuzzy subsets of Q. For any q Є Q, δ (q, a) is fuzzy subset of Q and it may be seen as the possibility distribution of the states that the automaton in the state q and with input a can enter. More generally, for each q' Є Q, δ (q, a)(q') is the possibility degree to which the automaton in state q and with input a may inter in the state q'.

Definition 3.2:

Let M= (Q, Σ, δ, q_0, F), be a fuzzy-finite automaton.

(i). the transition function δ is extended to δ: Q x Σ*→F (Q)

$$\delta(q, \varepsilon) = \frac{1}{q}$$

$$\delta(q, w\,a) = \bigcup_{p \in Q} [\delta(q,w)(p).\delta(p,a)]$$

for w Є Σ* and a Є Σ, where $\frac{1}{q}$ is a singleton in Q, i.e., the fuzzy subset of Q with membership 1 at q and 0 membership value for all other elements of Q. In addition, $\delta(q,w)(p).\delta(p,a)$ stands for the scalar product of the fuzzy set $\delta(p,a)$ with parameter $\delta(q,w)(p)$.

(ii). for any w Є Σ*, the length to which w is accepted by M is

L(M, w) =height($\delta(q_0,w)\cap F$).

(iii). The language L (M) accepted by M is a fuzzy subset of Σ* and it is defined by L(M)(w)=L(M, w) for all w Є Σ*.

A much more convenient way to calculate δ(q, w) for q Є Q and w Є Σ* is presented in the following lemma.

Lemma1. Let M= (Q, Σ, δ, q_0, F), be the fuzzy finite automaton, q_0ЄQ and $a_1, a_2, \ldots\ldots, a_n$ Є Σ. Then, for each p Є Q

$$\delta(q, a_1, a_2, \ldots\ldots, a_n) = \vee_{q1, q2, \ldots.., qn\text{-}1 \in Q} . \min [\delta(q, a_1)(q_1), \delta(q_1, a_2)(q_2), \ldots., \delta(q_{n-1}, a_n)(p)]$$

Proof: It is easy by induction on n.

Definition 3.3: Let M= (Q, Σ, δ, q_0, F), an FFA, then we may extend δ to a function from Q x F (Σ) into F (Q), which is still denoted as δ, with Zadeh's extension principle:

$$\delta(q,A) = \bigcup_{a \in \Sigma} [A(a).\delta(q,a)]$$

i.e,

$$\delta(q, a)(q') = \vee_{a \in \Sigma} \min[A(a), \delta(q,a)(q')]$$

For any q Є Q, A Є F(Σ) and q' Є Q.

Furthermore δ can be extended as follows:

δ: F(Q) x F(Σ) → F(Q)

$\delta(p, a) = \bigcup_{q \in Q} [P(a).\delta(q,a)]$

For any P Є F (Q) and A Є F(Σ) and

δ: F(Q) x F(Σ*) → F(Q)

δ (p, ε) = P

δ (P,WA) = δ(δ(P,W),A)

where PЄF(Q), WЄF(Σ*) and AЄF(Σ).

Definition 3.4 . If M= (Q, Σ, δ, q_0, F), is a fuzzy automaton and W Є F(Σ*), then the degree to which M accepts W is defined as follows :

L (M, W) = height (δ(q_0,w) ∩ F).

The word language L_w(M) accepted by M is a finite subset of F(Σ*) and it is defined by L_w(M)(W) = L(M,W) for any W Є F(Σ*).

Note: Definition 2 deals with strings of values whereas Definition 3 &4 are for strings of words.

We now present an example to explain the notions that we have used so far.

Example 3.1 . Assume that Q = {q_0, q_1} and Σ = {0, 1, 2}. Consider FFA M= (Q, Σ, δ, q_0, F), in which F = $\frac{0.3}{q0} + \frac{1}{q1}$

and δ is given as

δ(q_0,0) = (q_0 ,0.7)

δ(q_0,1) = {(q_0,0.3),(q_1,0.4)}

$\delta(q_0,2) = (q_1,0.2)$

$\delta(q_1,0) = (q_0,0.2)$

$\delta(q_1,1) = (q1,0.3)$

$\delta(q_1,2) == (q_0,0.4)$

Now we solve for the string 120

$\delta(q_0,1) = \{(q_0,0.3),(q_1,0.4)\}$

$\delta(q_0,12) = 0.3.\delta(q_0,2) \bigcup_{0.4} \delta(q_1,2)$

$= \{(q_1,0.2),(q_0,0.4)\}$

$\delta(q_0,120) = 0.2.\ \delta(q_1,0) \bigcup^{0.4} .\ \ \delta(q_0,0)$

$= \{(q_1,0.4),(q_0,0.2)\}$

$\delta(q_0,120) \cap F = q_0,0.2),(q_1,0.4)\}$

and the degree to which string 120 is accepted by M is $L(M,120) = 0.2 \vee 0.4 = 0.4$

Fuzzy transition table: For any fuzzy automata finite state and input both can be a fuzzy set. But in this example state are fuzzy set but input are not fuzzy. We can represent this transition function as transition table which is not crisp.

Present State	Input a=0	Input a=1	Input a=2
$\rightarrow q_0$	$(q_0, 0.7)$	$\{(q_0,0.3),(q_1,0.4)\}$	$(q_1, 0.2)$
q_1	$(q_0, 0.2)$	$(q1, 0.3)$	$(q_0, 0.4)$

Table1: Fuzzy transition table

Fuzzy graph model for finite fuzzy automata: Graphs are simply model of relation. A graph is a convenient way of representing information involving relationship between objects. The objects are represented by vertices and relations by edges. In many real world problems, we get partial information about that problem. So there is vagueness in the description of the objects or in its relationships or in both. To describe this type of relation, we need to design fuzzy graph model.

Automata can be represented by a directed graph. The vertices (denoted by single circles) of a transition diagram represent the states of the Automata and the arcs labeled with an input symbol correspond to the transitions. An arc (p, q) from vertex p to vertex q with label σ represents the transition $\delta(p, \sigma) = q$

In fuzzy automata can be represented by directed fuzzy graph (G). G= (V, σ, μ) is called a fuzzy graph on V where σ and μ are fuzzy sets on V and E ($V \times V$), respectively, such that $\mu(\{u, v\}) \leq \min\{\sigma(u), \sigma(v)\}$ for all $u, v \in V$.

For our example, we consider the fuzzy graph whose vertices are fuzzy but edges are in crisp.

$\mu(\{u, v\})=1$

and $1 \geq \sigma(V) \geq 0$

Graph for above Fuzzy-automata transition can be given as follows

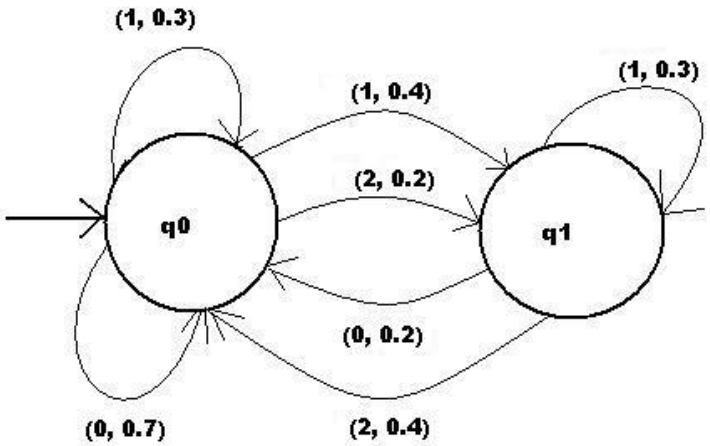

Fig.2: Fuzzy-finite automata

As we can observe from the above graph that there is no concept of final state since every string is accepted by some fuzzy membership value. There may be more than one transition for same input from the same state and all these transitions should be considered simultaneously.

4. Fuzzy Mealy Machine and Fuzzy Moore Machine:

The finite automata which we consider earlier have binary output, i.e. they accept the string or the do not accept the string, and in FFA the y accept the strings with some membership values. The acceptability was decided on the basis of reaching to the final state by starting from the initial state. Now we remove this restriction and consider the model where the output can be chosen from some other alphabet. The value of the output function Z(t) in the most general case if the function of the present state q(t) and present input x(t), i.e. ,

$$Z(t) = \lambda\,(q(t)\,,\, x(t))$$

where λ is called the output function. This generalized model is usually called Mealy machine. If the output function Z(t) depends only on the present state and is independent of the current input, the output function may be written as

$$Z(t) = \lambda(q(t))$$

This restricted model is called Moore machine.

Fuzzy-Mealy machine

The Fuzzy-Mealy machine has 6-tuple $M= (Q, \Sigma, \Delta, \delta, \lambda, q_0)$, where

(i) Q is a finite set of state ;

(ii) Σ is the input alphabet ;

(iii) Δ is the output alphabet ;

(iv) δ is the transition function from Σ x Q into Q ;

(v) λ is the output function from Σ x Q into F(Δ) where F(Δ) is the fuzzy transition function into Δ ; and

(vi) q_0 is the start state .

Now let us present an example to solve a fuzzy Mealy machine problem as follows:

Example 4.1. Assume $Q = \{q_1, q_2, q_3, q_4\}$, $\Sigma = \{0, 1\}$, and $\Delta = \{0, 1\}$. Consider a fuzzy Mealy machine $M= (Q, \Sigma, \Delta, \delta, \lambda, q_0)$, which has an output fuzzy transition function given as follows:

$$\lambda(q_1,0) = \frac{0.2}{0} + \frac{0.3}{1}$$

$$\lambda(q_1,1) = \frac{0.2}{1}$$

$$\lambda(q_2,0) = \frac{0.4}{0}$$

$$\lambda(q_2,1) = \frac{0.7}{1}$$

$$\lambda(q_3,0) = \frac{0.4}{1}$$

$$\lambda(q_3,1) = \frac{0.4}{0} + \frac{0.3}{1}$$

$$\lambda(q_4,0) = \frac{0.6}{1}$$

$$\lambda(q_4,1) = \frac{0.4}{0} + \frac{0.5}{1}$$

And the transition table is given as follows:

Present State	**Next State**			
	$a=0$		$a=1$	
	State	Output	State	Output
$\rightarrow q_1$	q_3	$\frac{0.2}{0} + \frac{0.3}{1}$	q_2	$\frac{0.2}{1}$
q_2	q_1	$\frac{0.4}{0}$	q_4	$\frac{0.7}{1}$
q_3	q_2	$\frac{0.4}{1}$	q_1	$\frac{0.4}{0} + \frac{0.3}{1}$

q_4	q_4	$\frac{0.6}{1}$	q_3	$\frac{0.4}{0}+\frac{0.5}{1}$

Table2: A Fuzzy Mealy Machine

And the transition graph can be given as follows

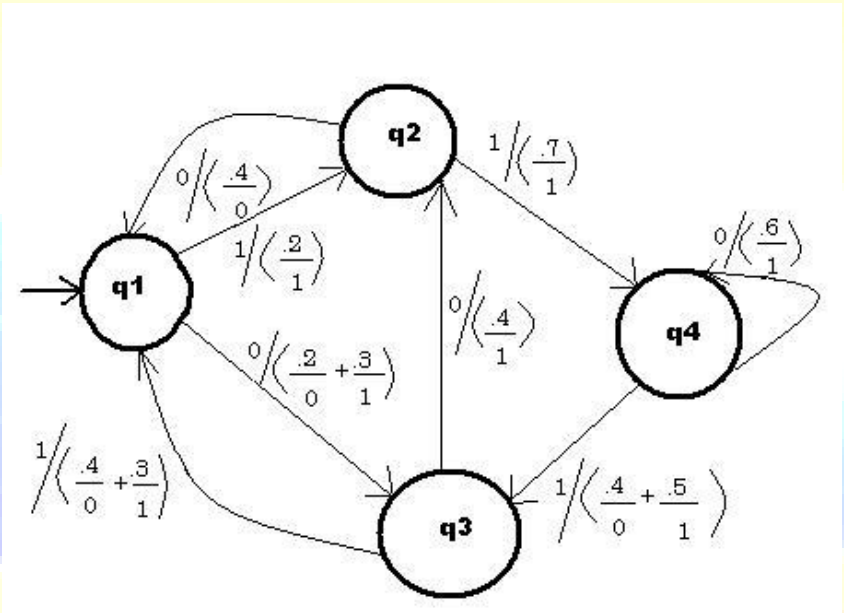

Fig.3: Fuzzy Mealy Machine

Now let us solve for a string 01110.

As we can observe from the above transition table, the transition of state is given by $q_1 \rightarrow q_3 \rightarrow q_1 \rightarrow q_2 \rightarrow q_4 \rightarrow q_4$ and the output can be given by the fuzzy transition

$$\max[\min\{(\frac{0.2}{0}+\frac{0.3}{1}),(\frac{0.4}{0}+\frac{0.3}{1}),(\frac{0.2}{1}),(\frac{0.7}{1}),(\frac{0.6}{1})\}]$$

$$=\max[\min\{(\frac{0.2}{0}+\frac{0.3}{1}),(\frac{0.4}{0}+\frac{0.3}{1}),(\frac{0.2}{1}),(\frac{0.7}{1}),(\frac{0.6}{1})\}]$$

$$=\max\left[\frac{\min(0.2,0.4,)}{0}+\frac{\min(0.3,0.3,0.2,0.7,0.6)}{1}\right]$$

$$=\max\left[\frac{0.2}{0}+\frac{0.2}{1}\right]$$

$$=0.2 \vee 0.2$$

$$=0.2$$

The output string is accepted by the fuzzy membership value 0.2.

Fuzzy Moore machine

The fuzzy Moore machine has 6-tuple M= (Q, Σ, Δ, δ, λ, q_0), where

(i) Q is a finite set of state ;

(ii) Σ is the input alphabet ;

(iii) Δ is the output alphabet ;

(iv) δ is the transition function from Σ x Q into Q ;

(v) λ is the output function from Q into F(Δ) where F(Δ) is the fuzzy transition function into Δ ; and

(vi) q_0 is the start state .

Now let us present an example to solve a fuzzy Mealy machine problem as follows:

Example 4.2. Assume Q = {q_0, q_1, q_2, q_3}, Σ = {0, 1}, and Δ = {0, 1}. Consider a fuzzy Mealy machine M= (Q, Σ, Δ, δ, λ, q_0), which has an output fuzzy transition function given as follows:

$$\lambda(q_0)=\frac{0.4}{0}+\frac{0.3}{1}$$

$$\lambda(q_1) = \frac{0.8}{0}$$

$$\lambda(q_2) = \frac{0.4}{1} + \frac{0.2}{0}$$

$$\lambda(q_3) = \frac{0.7}{1}$$

and the transition table is given as follows :

Present State	Next State		Output (λ)
	$a=0$	$a=1$	
$\rightarrow q_0$	q_3	q_1	$\frac{0.4}{0} + \frac{0.3}{1}$
q_1	q_1	q_2	$\frac{0.8}{0}$
q_2	q_2	q_3	$\frac{0.4}{1} + \frac{0.2}{0}$
q_3	q_3	q_0	$\frac{0.7}{1}$

Table3: A Fuzzy Moore Machine

And the transition graph for Fuzzy-Moore machine can be given as follows:

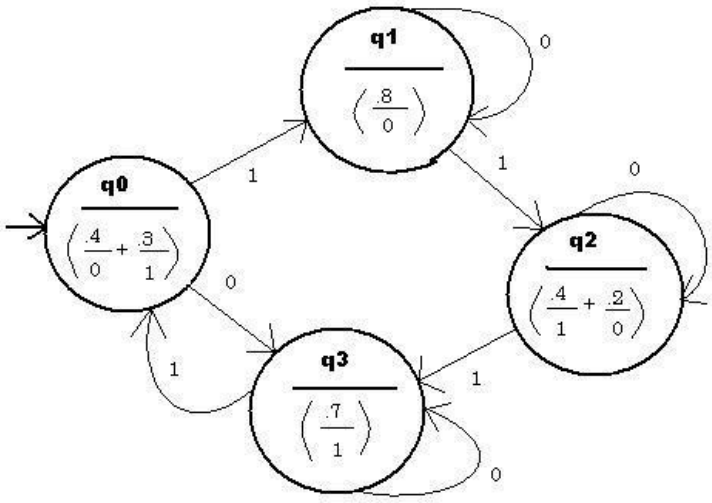

Fig.4: Fuzzy Moore Machine

Now let us solve for a string 00111.

As we can observe from the above transition table, the transition of state is given by $q_0 \rightarrow q_3 \rightarrow q_3 \rightarrow q_0 \rightarrow q_1 \rightarrow q_2$ and the output can be given by the fuzzy transition

$$\max[\min\{(\frac{0.4}{0}+\frac{0.3}{1}),(\frac{0.7}{1}),(\frac{0.7}{1}),(\frac{0.4}{0}+\frac{0.3}{1}),(\frac{0.8}{0}),(\frac{0.4}{1}+\frac{0.2}{0})\}]$$

$$=\max[\min\{(\frac{0.4}{0}+\frac{0.3}{1}),(\frac{0.7}{1}),(\frac{0.7}{1}),(\frac{0.4}{0}+\frac{0.3}{1}),(\frac{0.8}{0}),(\frac{0.4}{1}+\frac{0.2}{0})\}]$$

$$= \max[\frac{\min(0.4,0.4,0.8,0.2)}{0}+\frac{\min(0.3,0.7,0.7,0.3,0.4)}{1}]$$

$$=\max[\frac{0.2}{0}+\frac{0.3}{1}]$$

$$=0.2 \vee 0.3$$

$$=0.3$$

The output string is accepted by the fuzzy membership value 0.3.

As we can observe from both above tables, for a Moore machine and for a fuzzy Moore machine as well, if the input string of length n, for output it takes n+1 transitions, i.e., the output string is of length n+1. The first output is $\lambda(q_0)$ for all output strings. In the case of Mealy machine and for fuzzy Mealy machine if the input string of length n, for output it takes also n transitions, i.e., the output string is also of length n.

Conclusion:

Until now the power of fuzzy graphs has not exploited in fuzzy automata theory. Also different fuzzy automata have different underlying topologies, so it is more convenient and flexible to utilize fuzzy graph theory when analyzing and designing fuzzy automata. Once the fuzzy graph model for a specific problem is obtained, we can directly go for the corresponding topology of fuzzy automata model. In this way fuzzy graph can be drawn for fuzzy Mealy and fuzzy Moore machine and transition table can also be represented as well. We try to apply those fuzzy automata in real world problem.

References:

1. Mingsheng Ying ," A Formal Model of Computing With Words", IEEE TRANSACTIONS ON FUZZY SYSTEMS, VOL. 10, NO. 5, OCTOBER 2002.
2. B. Gaines and L. Kohout, "The logic of automata," *Int. J. Gen. Syst* .,vol. 2, pp. 191–208, 1976.
3. C. L. Giles, C. W. Omlin, and K. K. Thornber, "Equivalence in knowledge representation: Automata, recurrent neural networks, and dynamical fuzzy systems," *Proc. IEEE, vol. 87, pp. 1623–1640, Sept.* 1999.
4. J. Grantner and M. J. Paytyra, "VLSI implementations of fuzzy logic finite state machines," in *Proc. 5th IFSA Congr., 1993, pp. 781–784.*

5. "Synthesis and analysis of fuzzy logic finite state machine models," in *Proc. 3rd IEEE Conf. Fuzzy Systems, vol. I, 1994, pp.* 205–210.

6. J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation. Reading, MA: Addison-Wesley, 1979.*

7. E. B. Kosmatopoulos and M. A. Christodoulou, "Neural networks for identification of fuzzy dynamic systems: An application to identification of vehicle highway systems," in *Proc. 4th IEEE Mediterranean Symp. New Direction in Control and Automation*, 1996, pp. 23–28.

8. S. I. Mensch and H. M. Lipp, "Fuzzy specification of finite state machines," in *Proc. Eur. Design Automation Conf.*, 1990, pp. 622–626.

9. C. W. Omlin, K. K. Thornber, and C. L. Giles, "Fuzzy finite-state automata can be deterministically encoded into recurrent neural networks," *IEEE Trans. Fuzzy Syst.*, vol. 6, pp. 52–75, Feb. 1998.

10. E. S. Santos, "Maximin automata," *Inform. Control*, vol. 12, pp.363–377, 1968.

11. M. G. Thomason and P. N. Marinos, "Deterministic acceptors of regular fuzzy languages," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-4, pp.228–230, 1974.

12. F. A. Unal and E. Khan, "A fuzzy finite state machine implementation based on a neural fuzzy system," in *Proc. 3rd Int. Conf. Fuzzy Systems*, vol. 3, 1994, pp. 1749–1754.

13. L. A. Zadeh, "Fuzzy Languages and Their Relation to Human and Machine Intelligence," Electron. Res. Lab., Univ. California, Berkeley, CA, Tech. Rep. ERL-M302, 1971.

14. "Fuzzy logic, neural networks, and soft computing," *Commun.ACM*, vol. 37, pp. 77–84, 1994.

15. "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic," *Fuzzy Sets Syst.*, vol. 90, pp. 111–127, 1997.

16. "Fuzzy logic = computing with words," *IEEE Trans. Fuzzy Syst.*,vol. 4, pp. 103–111, May 1996.

17. K.L.P. Mishra and N. Chandrashekharan , Theory of Computer Science (Automata, Languages and Computation) 2nd edition.

18. FUZZY SET THEORY and its applications, second edition, H.J.Zimmermann.