International Journal of Mathematical, Engineering and Management Sciences
Vol. 6, No. 6, 1709-1728, 2021
https://doi.org/10.33889/IJMEMS.2021.6.6.101

State Minimization of General Finite Fuzzy Automata

Ranjeet Kaur
Department of Mathematics,
Jaypee Institute of Information Technology, Noida, Uttar Pradesh, India.
ABES Engineering College, Ghaziabad, Uttar Pradesh, India.
Corresponding author: ranjeet.kaur@abes.ac.in, reetmaths@gmail.com

Alka Tripathi
Department of Mathematics,
Jaypee Institute of Information Technology, Noida, Uttar Pradesh, India.
E-mail: alka.choubey@jiit.ac.in, alka.choubey@gmail.com

(Received on May 21, 2021; Accepted on November 11, 2021)

Abstract

The minimization of automaton is important to reduce space and computational time. Reduction in number of states and transitions leads to equivalent automaton with less number of states and transitions. In this paper, state minimization of General Finite Fuzzy Automata (GFFA) is discussed. To obtain minimal equivalent GFFA we have removed redundant states and transitions using substitution property (SP) partition and quotient machine. The algorithm to find membership values of states of the GFFA is described and algorithm to associate states with quotient machine to obtain minimal machine with less number of states is discussed.

Keywords- General fuzzy automata, Minimal automata, Quotient machine, Substitution property partition.

1. Introduction

Zadeh (1965) introduced fuzzy set theory. The fuzzy automata were formalized by Wee (1967). Fuzzy Finite State Automata is used in many areas such as pattern recognition, pattern classifications, approximate string matching, lexical analysis etc. (Wee and Fu, 1969). The idea of General Fuzzy Automata (GFA) was formalized by Doostfatemeh and Kremer (2005). They identified the insufficiency in existing literature of fuzzy automata for assigning membership values to states in case of simultaneous transitions. The two main issues were to assign membership value to state after each transition and to resolve multi-membership value of a state due to simultaneous or overlapping transitions.

Researchers are continuously interested in the subject of state minimization. The minimization of states in automata reduces computational time of given automata. Several authors such as Ciric et al. (2010), Lee (2000), Li and Qiu (2015), Malik et al. (1999), Santos (1972), Stamenkovic et al. (2018), Topencharov and Peeva (1981) have given different methods of minimization to minimize number of states of fuzzy automata. Some of the existing proposed algorithms are summarized in Table 1. Malik et al. (1999) discussed the minimization of Fuzzy Finite Automata (FFA) to an equivalent automaton if states are equivalent and proved that if there exists another equivalent FFA then it is homomorphic to minimal FFA. The concept of partitioning of states by substitution property in fuzzy finite state automata using quotient machine was introduced in Basak and Gupta (2002). The minimized quotient machine was shown to be behaviourally equivalent to the given fuzzy finite state automata.

A lot of research was carried on General Fuzzy Automata, Intuitionistic Fuzzy Automata (Choubey and Ravi, 2013), Intuitionistic General Fuzzy Automata (Shamsizadeh and Zahedi, 2016), topology on General Fuzzy Automata (Horry and Zahedi, 2013), BL- General Fuzzy Automata (Abolpur and Zahedi, 2012; Shamsizadeh and Zahedi, 2019) and General Fuzzy Finite Switchboard Automata (Kavikumar et al., 2019). Shamsizadeh and Zahedi (2015) proposed Intuitionistic General Lattice valued Fuzzy Automata (IGLFA) and discussed minimal IGLFA by generalized Myhill-Nerode Theorem. Ignjatovic et al. (2018) discussed the equivalence of weighted finite automata with outputs in respect of different semantics. Recently Dubey et al. (2020) introduced Quntale valued fuzzy automata to study the existence of minimal Q-valued fuzzy automata. Balle and Rabusseau (2020) proposed approximate minimization of weighted tree automata by converting it into its canonical form. Ghorani and Moghari (2021) discussed the method for minimal lattice valued tree automata with the help of solution of system of fuzzy polynomial equations.

Table 1. Summary of minimization algorithms for fuzzy finite automata.

Author	Algorithm	Method
Santos (1972)	Reduction problem of maximin sequential like machine	Constructing state-wise and composite-wise equivalence.
Malik et al. (1999)	Minimization of Fuzzy Finite Automata (FFA)	Existence of equivalent FFA based on equivalent states and proved equivalent FFA is homomorphic to minimal FFA.
Lee (2000)	Minimization of FFA in canonical form.	Conversion of the FFA to canonical FFA and minimal canonical fuzzy finite automata using equivalence relation.
Basak and Gupta (2002)	Compute induced partition of states for FFA.	Partitioning of states by k-equivalence using substitution property (SP) partition.
Cheng and Mo (2004)	Minimization of Mealy type of FFA	Using equivalence relations of states.
Zhiwen and Xiaolei (2007)	Minimization of Mizumoto Automata with fuzzy initial and fuzzy final states.	Conversion of Mizumoto automata to equivalent canonical fuzzy automata by removing inaccessible states.
Li and Pedrycz (2007)	Minimization of lattice valued finite automata.	Obtained minimal corresponding deterministic lattice finite automata by removing inaccessible states.
Peeva and Zahariev (2008)	Minimization for max-min fuzzy finite machine.	Computing the behavior matrix to establish equivalence of states.
Li and Qiu (2015)	Minimization of FFA.	Solvability of system of fuzzy polynomial equation and condition of equivalence between two FFA.
Mendivil (2018)	Minimization of Fuzzy Finite Automata using determinization method.	Brzozowski's minimization procedure for fuzzy finite automata.

This paper proposes state minimization of the GFFA using the concept of substitution property (SP) partition (Bacon, 1964). Algorithms with pseudo code are examined to find membership value of initial state designator and to compute induced partition of states. The quotient GFFA with respect to the SP partition is obtained from induced partition. The quotient machine is behaviourally equivalent to the given GFFA with minimum number of states.

Further sections are organized as follows: Section 2 describes the preliminaries. Section 3 proposes GFFA with initial and final state designator for minimization. Section 4 explains the procedure to find minimal GFFA (Figure 1) and demonstrate the results with the help of an example. Finally, Section 5 contains conclusion.

2. Preliminaries

We can find basics of fuzzy set and automata theory in Basak and Gupta (2002), Mizumoto et al. (1969), Zadeh (1965).

The operation of maximum and minimum will be denoted by $\vee$ and $\wedge$, respectively. For any two positive numbers p and q, $p \vee q = \max\{p, q\}$, $p \wedge q = \min\{p, q\}$.

Fuzzy Matrix: For finite natural numbers m and n, a matrix C of order $m \times n$ denoted as $C = [c_{ij}]$ such that $0 \le c_{ij} \le 1$; $1 \le i \le m$; $1 \le j \le n$, is called a fuzzy matrix (Basak and Gupta, 2002; Tripathi and Kaur, 2019).

Product of Fuzzy Matrices (Basak and Gupta, 2002): Let $C = [c_{ij}]$ be any $m \times n$ fuzzy matrix and $D = [d_{jk}]$ be any $n \times p$ fuzzy matrix. The product $CD = [p_{ik}]$ is $m \times p$ fuzzy matrix, where $p_{ik} = \underset{j}{\vee}(c_{ij} \wedge d_{jk})$ for all $1 \le i \le m$, $1 \le k \le p$.

Identical Row Max of Fuzzy Matrix (Basak and Gupta, 2002): Let $C = [c_{ij}]$ be any $m \times n$ fuzzy matrix. The row max of each row is the highest value of the elements of the row and identical row max is $c = \forall i, \underset{j}{\vee} c_{ij}$.

Lemma 1 (Basak and Gupta, 2002): Let $C = [c_{ij}]$ be any $m \times n$ fuzzy matrix having identical row max c and $D = [d_{jk}]$ be any $n \times p$ fuzzy matrix having identical row max d; then CD has identical row max $c \wedge d$, *i.e.* $\min\{c, d\}$.

We have explained the product of fuzzy matrices and Lemma 1 with the help of Example 1.

Example 1: Consider two fuzzy matrices C and D of order 3×3;

$$C = \begin{bmatrix} 0.8 & 0.6 & 0.4 \\ 0.7 & 0.5 & 0.8 \\ 0.4 & 0.8 & 0.7 \end{bmatrix}, \quad D = \begin{bmatrix} 0.7 & 0.3 & 0.5 \\ 0.7 & 0.6 & 0.5 \\ 0.4 & 0.7 & 0.6 \end{bmatrix}.$$

identical row max for matrix C is $\forall i, \underset{j}{\vee} c_{ij} = 0.8$, identical row max for matrix D is $\forall j, \underset{k}{\vee} d_{jk} = 0.7$.

Product of fuzzy matrices C and D is, $CD = \begin{bmatrix} 0.7 & 0.6 & 0.5 \\ 0.7 & 0.7 & 0.6 \\ 0.7 & 0.7 & 0.6 \end{bmatrix}$.

By, Lemma 1 identical row max for matrix CD is $\forall i, \underset{k}{\vee} p_{ik} = \underset{k}{\vee}\underset{j}{\vee}(c_{ij} \wedge d_{jk}) = 0.8 \wedge 0.7 = 0.7$.

Mizumoto et al. (1969) introduced fuzzy distribution in fuzzy automata and defined finite fuzzy automata. They proved that efficiency of fuzzy automata can be improved by defining initial state designator and assigning membership values to transition matrix.

Let $\sum$ be a finite non-empty set of alphabets and $\sum^*$ be the set of all finite strings including null string over $\sum$. $|x|$ denotes the length of string, where $x \in \sum^*$.

Finite Fuzzy Automaton (FFA) (Basak and Gupta, 2002; Mizumoto et al., 1969; Zhiwen and Xiaolei, 2007): An FFA over an input alphabet $\sum$ is defined to be an algebraic system $\mathcal{A} = \left(S, \alpha, \{M_\sigma : \sigma \in \sum\}, \eta^{\mathcal{F}}\right)$ where, $S = \{s_1, s_2, ..., s_n\}$ is set of n states. $\alpha = \{\alpha_{s_1}, \alpha_{s_2}, ..., \alpha_{s_n}\}$ is fuzzy row-vector of dimension n where, $0 \le \alpha_{s_k} \le 1, 1 \le k \le n$, called the initial state designator. Every state of fuzzy finite automata $\mathcal{A}$ has membership value which is represented by components of fuzzy row-vector α. For each $\sigma \in \sum$, $M_\sigma = (m_{ij}(\sigma))$, is fuzzy transition matrix of order n on input σ. The transition function is a fuzzy subset of $S \times S$ whose membership function has the value $m_{ij}(\sigma)$ at (s_i, s_j), where $s_i, s_j \in S$. $\mathcal{F} \subseteq S$ denotes set of final states. $\eta^{\mathcal{F}} = (\eta_1, \eta_2, ..., \eta_n)^T$ is fuzzy column-vector of dimension n, where η_i has value 1 if $s_i \in \mathcal{F}$ and 0 if $s_i \in S - \mathcal{F}$, called the final state designator.

Doostfatemeh and Kremer (2005) proposed General Fuzzy Automata (GFA). They addressed the issue of assigning membership value to the active states and states having simultaneous transitions.

General Fuzzy Automaton (GFA) (Doostfatemeh and Kremer, 2005; Doostfatemeh and Kremer, 2006; Shamsizadeh and Zahedi, 2016): A GFA is considered as $\tilde{A} = (S, \sum, \mathcal{R}, Z, \tilde{\delta}, \omega, F_1, F_2)$, where,

(i) $S = \{s_1, s_2, ..., s_n\}$ is a finite non-fuzzy set of states.

(ii) $\sum$ is a finite non-fuzzy set of input symbols.

(iii) $\mathcal{R} \subseteq P(S)$ is set of fuzzy initial states, where $P(S)$ is fuzzy power set of S.

(iv) $Z = \{b_1, b_2, ..., b_p\}$ is a finite non-fuzzy set of output symbols.

(v) $\tilde{\delta}$ is the augmented transition function that assigns membership values to next state in the interval [0,1] using membership assignment function $F_1(\mu, \delta)$.

(vi) $\omega : S \to Z$ is the output function.

(vii) $F_1 : [0,1] \times [0,1] \to [0,1]$ is membership assignment function processed by augmented transition function $\tilde{\delta}$ to assign membership values to the active states. $F_1(\mu, \delta)$ depends on membership value μ of predecessor state and weight of transition δ such that $\delta : S \times \sum \times S \to [0,1]$. If there is a transition at time t from state s_i to state s_j on input a then its membership value is denoted as $\mu^t(s_j) = \tilde{\delta}((s_i, \mu^{t-1}(s_i)), a, s_j) = F_1(\mu^{t-1}(s_i), \delta(s_i, a, s_j))$.

(viii) $F_2 : [0,1]^* \to [0,1]$ is multi-membership resolution function that determines the active states with simultaneous transitions and assigns a single membership value to them.

Generally, $F_1(\mu,\delta)$ can be taken as $\max(\mu,\delta)$, $\min(\mu,\delta)$ or $(\mu+\delta)/2$. Similar to F_1 many operations are applicable to F_2. Δ denotes the set of all transitions of GFA. The states are called active states at time t if there is at least one transition on input symbol. $Q_{act}(t=i), \forall i \geq 0$ is the fuzzy set of active states. Hence, $Q_{act}(t=0)=\mathcal{R}$ is the active state set of fuzzy start states.

$$Q_{act}(t=i)=\{(s,u^{t=i}(s)):\exists\ s'\in Q_{act}(t=i-1),\exists\ a\in\sum,\ \delta(s',a,s)\in\Delta\},\forall i\geq 1.$$

Doostfatemeh and Kremer (2005) proposed the multi membership resolution algorithm to find membership value of the state s_m if simultaneous transition occurs at time t, using combined operations of F_1 and F_2.

Algorithm 1 (Doostfatemeh and Kremer, 2005): For a GFA $\tilde{A}=(S,\sum,\mathcal{R},Z,\tilde{\delta},\omega,F_1,F_2)$ with simultaneous transitions the following algorithm assigns a single membership value to the active state s_m at time t.

1. F_1 produces a membership value v_i by processing transition weight $\delta(s_i,a_k,s_m)$ and $\mu^{t-1}(s_i);t\geq 1$, *i.e.* $v_i=\tilde{\delta}((s_i,\mu^{t-1}(s_i)),a_k,s_m)=F_1(\mu^{t-1}(s_i),\delta(s_i,a_k,s_m))$.
2. If the membership values v_i obtained at time $t-1$, $t\geq 1$ are not equal, they are determined by F_2.
3. F_2 assigns final membership value to the active state s_m at time t with n overlapping transitions i.e. $\mu^t(s_m)=F_{2\,i=1}^{\,n}[v_i]=F_{2\,i=1}^{\,n}[F_1(\mu^{t-1}(s_i),\delta(s_i,a_k,s_m))]$.

3. General Finite Fuzzy Automata with Initial and Final State Designator

In this section we have extended GFA with the concept of fuzzy finite automata defined by Mizumoto et al. (1969) and discussed minimization of General Finite Fuzzy Automata.

General Finite Fuzzy Automata with Initial and Final State Designator (GFFA): A GFFA $\tilde{\mathcal{M}}=\{S,\sum,\mathcal{R},\mathcal{Z},\tilde{\delta},\omega,\{A(a):a\in\sum\},\mathcal{F}_1,\mathcal{F}_2,\alpha,\eta^{\mathcal{F}}\}$ is an eleven-tuple machine, where

(i) $S=\{s_1,s_2,\ldots,s_n\}$ is a finite non-fuzzy set of states.

(ii) $\sum$ is a finite non-fuzzy set of input symbols.

(iii) $\mathcal{R}\subseteq\mathcal{P}(S)$ denotes set of fuzzy initial states, where $\mathcal{P}(S)$ is fuzzy power set of S.

(iv) $\mathcal{Z}=\{b_1,b_2,\ldots,b_p\}$ is a finite non-fuzzy set of output symbols.

(v) $\tilde{\delta}:(S\times[0,1])\times\sum\times S\overset{\mathcal{F}_1(\mu,\delta)}{\longrightarrow}[0,1]$ is an augmented transition function.

(vi) $\omega:S\rightarrow\mathcal{Z}$ is the output function.

(vii) For each $a\in\sum$, $A(a)=(a_{ij}(a))$ is a fuzzy transition matrix of order n on input a. The transition function is a fuzzy subset of $S\times S$ whose membership function has the value $\delta(s_i,a,s_j)=a_{ij}(a)$, where $s_i,s_j\in S$.

(viii) $\mathcal{F}_1 : [0,1] \times [0,1] \to [0,1]$ is membership assignment function applied through $\tilde{\delta}$ to assign membership values to the active states.

(ix) $\mathcal{F}_2 : [0,1]^* \to [0,1]$ is multi-membership resolution function which determines active states with simultaneous transitions and assigns a final membership value to them.

(x) $\alpha = \{\alpha_{s_1}, \alpha_{s_2}, ..., \alpha_{s_n}\}$ is a fuzzy row-vector of dimension n, where $0 \le \alpha_{s_k} \le 1, 1 \le k \le n$, called the initial state designator. Every state of $\tilde{\mathcal{M}}$ has membership value which is represented by components of α.

(xi) $\eta^{\mathcal{F}} = (\eta_1, \eta_2, ..., \eta_n)^T$ is a fuzzy column-vector of dimension n, where $\mathcal{F}$ is final states set such that η_i has value 1 if $s_i \in \mathcal{F}$ and 0 if $s_i \in S - \mathcal{F}$, called the final state designator.

For any string $x \in \sum^*$, the transition matrix $A(x)$ is defined recursively as:

1. $A(\lambda) = I_n$, unit matrix and λ is null string.
2. $\forall x \in \sum^*, a \in \sum, A(xa) = A(x)A(a).$

The behaviour of a GFFA $\tilde{\mathcal{M}}$ refers to a fuzzy subset of $\sum^*$ denoted as $B_{\tilde{\mathcal{M}}}$, with membership function $\forall x \in \sum^*, \mu_{B_{\tilde{\mathcal{M}}}}(x) = \alpha A(x)\eta^{\mathcal{F}} = \alpha\eta^{\mathcal{F}}(x)$ where $\eta^{\mathcal{F}}(x) = A(x)\eta^{\mathcal{F}}$.

Two GFFA's $\tilde{\mathcal{M}}$ and $\tilde{\mathcal{M}}'$ are said to be behaviourally equivalent if they accept same string with same membership value i.e. $\mu_{B_{\tilde{\mathcal{M}}}}(x) = \mu_{B_{\tilde{\mathcal{M}}'}}(x)$.

Partition: Consider a GFFA $\tilde{\mathcal{M}}$ with finite state set S. A partition π is a set that contains all non-empty subsets of S such that every state of S is in exactly one of these subsets (Bacon, 1964). These non-empty subsets are called the blocks of the partition.

Product of Partitions (Bacon, 1964): The product of two set partitions π_1 and π_2 is defined as the set partition whose parts are the non-empty intersections between each part of π_1 and each part of π_2.

Substitution Property (SP) (Bacon, 1964): A partition π on the state set S in GFFA $\tilde{\mathcal{M}} = \{S, \sum, \mathcal{R}, \mathcal{Z}, \tilde{\delta}, \omega, \{A(a) : a \in \sum\}, \mathcal{F}_1, \mathcal{F}_2, \alpha, \eta^{\mathcal{F}}\}$ is said to have substitution property if for every $a \in \sum$, every fuzzy sub-matrix obtained from transition matrix $A(a)$ by blocks of partition π has identical row max. A partition π_r refines the set of final states $\mathcal{F}$, if each block of π_r belongs to either $\mathcal{F}$ or $S - \mathcal{F}$.

Equivalent States: The states can be categorized based on their equivalence (Basak and Gupta, 2002). Let $\tilde{\mathcal{M}} = \{S, \sum, \mathcal{R}, \mathcal{Z}, \tilde{\delta}, \omega, \{A(a) : a \in \sum\}, \mathcal{F}_1, \mathcal{F}_2, \alpha, \eta^{\mathcal{F}}\}$ be a GFFA. Two states s_i and s_j of $\tilde{\mathcal{M}}$ are said to be equivalent, written as $s_i \equiv s_j$, iff for every string $x \in \sum^*$, $\eta^{\mathcal{F}}(x)$ has the same i^{th} and j^{th} entries. The partition induced by this equivalence relation will be denoted as $\pi_{\mathcal{F}}$, called the induced partition for $\tilde{\mathcal{M}}$.

k-Equivalent States (Basak and Gupta, 2002):
Let $\tilde{\mathcal{M}} = \{S, \sum, \mathcal{R}, \mathcal{Z}, \tilde{\delta}, \omega, \{A(a) : a \in \sum\}, \mathcal{F}_1, \mathcal{F}_2, \alpha, \eta^{\mathcal{F}}\}$ be a GFFA. A state s_i is said to be k-equivalent to a state s_j, where k is a non-negative integer iff for any string $x \in \sum^*$ such that $|x| \le k$; $\eta^{\mathcal{F}}(x)$ has the same i^{th} and j^{th} entries and the partition of state set S induced by this k-equivalence relation is denoted as π_k.

Quotient General Finite Fuzzy Automata: Consider a General Finite Fuzzy automata $\tilde{\mathcal{M}} = \{S, \sum, \mathcal{R}, \mathcal{Z}, \tilde{\delta}, \omega, \{A(a) : a \in \sum\}, \mathcal{F}_1, \mathcal{F}_2, \alpha, \eta^{\mathcal{F}}\}$ and π be a SP partition that refines final state set $\mathcal{F}$. Then the quotient general finite fuzzy automata $\tilde{\mathcal{M}}/\pi = \{S', \sum, \mathcal{R}', \mathcal{Z}, \tilde{\delta}', \omega, \{A'(a) : a \in \sum\}, \mathcal{F}_1', \mathcal{F}_2', \alpha', \eta^{\mathcal{F}'}\}$ is a GFFA, where: $S' = \{B_1, B_2, \ldots B_m\}$ is the set of blocks of partition π such that $m = rank(\pi)$, *i.e.* number of non-zero rows in partition π. For every $a \in \sum$, $A'(a)$ is an $m \times m$ fuzzy matrix obtained by replacing each of the submatrices into which $A(a)$ is partitioned by the blocks of π with its constant row max. α' is the fuzzy row *m*-vector obtained by replacing each of the sub-vectors into which α is partitioned by the blocks of π with its max. $\mathcal{F}'$ is the set of blocks of π partitioning $\mathcal{F}$. For any $x \in \sum^*$, $\eta^{\mathcal{F}}(x)$ has identical elements corresponding to each block of partition π and $\eta^{\mathcal{F}'}(x)$ is a fuzzy column vector obtained by replacing each of column sub-vectors with its constant element.

Theorem 3.1: If for a given General Finite Fuzzy Automata $\tilde{\mathcal{M}} = \{S, \sum, \mathcal{R}, \mathcal{Z}, \tilde{\delta}, \omega, \{A(a) : a \in \sum\}, \mathcal{F}_1, \mathcal{F}_2, \alpha, \eta^{\mathcal{F}}\}$, π is a partition satisfying substitution property on state set S, then for any string $x \in \sum^*$ all the possible fuzzy sub-matrices of fuzzy transition matrix $A(x)$ partitioned by the blocks of partition π has identical row max.

Proof. We prove result using induction for length of string $|x| = n$.
Let $A(a) = (a_{ij})$ be an $n \times n$ fuzzy transition matrix of $\tilde{\mathcal{M}}$ on input $a \in \sum$, π be a partition satisfying substitution property on state set S.

Basis step: If $n = 0$, then $x = \lambda$, $\lambda \in \sum^*$, the transition matrix $A(\lambda) = \begin{cases} 1\,;\ i = j \\ 0;\ i \ne j \end{cases}$.

Clearly $A(\lambda)$ is partitioned by the blocks of π has identical row max 1.

If $n = 1$, then $x \in \sum$, π is a partition satisfying substitution property on state set S *i.e.* $\pi = \{S - \mathcal{F}, \mathcal{F}\}$,

$$A(x)=\begin{cases}1\,;\, i=j\\ \mu_{ij}\,;\, i\neq j,\mu_{ij}\in[0,1]\end{cases}.$$

Clearly $A(x)$ is partitioned by the blocks of π has identical row max.

Induction Hypothesis: Assume the result is true for string of length $|x|=n-1; x=x_1x_2...x_{n-1}$, $\forall\ x\in\sum^*$ and $n>1$.

The fuzzy transition matrix $A(x)=A(x_1x_2...x_{n-1})=A(x_1)A(x_2)...A(x_{n-1})=(b_{ij})$; $1\leq i\leq n,\ 1\leq j\leq n$ is partitioned by the blocks of π has identical row max.

Induction Step: Now, we will prove theorem is true for string of length n ; $|x|=n$, $\forall\, x\in\sum^*$ and $n>1$.

Let $x=ya$; $y=x_1x_2....x_{n-1}$ with $y\in\sum^*$; $a\in\sum$, such that $|y|=n-1, |a|=1$.

$A(x)=A(ya)=A(y)A(a)=(b_{ij})(a_{ij})=(c_{ij})$, $1\leq i\leq n,\ 1\leq j\leq n$ (using lemma 1) is partitioned by the blocks of π has identical row max.

Hence the proof.

Theorems for induced partition proved in Basak and Gupta (2002) can easily be proved for General Finite Fuzzy Automata.

Theorem 3.2: For GFMA, $\tilde{\mathcal{M}}=\{S,\sum,\mathcal{R},\mathcal{Z},\tilde{\delta},\omega,\{A(a):a\in\sum\},\mathcal{F}_1,\mathcal{F}_2,\alpha,\eta^{\mathcal{F}}\}$.

1. Induced partition $\pi_{\mathcal{F}}$ refines set $\mathcal{F}$ of final states.
2. Induced partition $\pi_{\mathcal{F}}$ satisfies substitution property.
3. Induced partition $\pi_{\mathcal{F}}$ is the largest SP partition that refines $\mathcal{F}$.

Minimal Machine: If $\pi_{\mathcal{F}}$ is the induced partition for GFFA $\tilde{\mathcal{M}}$, then the quotient general finite fuzzy automata $\tilde{\mathcal{M}}/\pi_{\mathcal{F}}$ is called minimal machine of given GFFA $\tilde{\mathcal{M}}$, denoted by $\tilde{\mathcal{M}}_{\min}$. Minimal Machine is equivalent to given GFFA denoted as $\tilde{\mathcal{M}}\equiv\tilde{\mathcal{M}}_{\min}$.

4. Procedure

In this section, we have given procedure to find membership values of the states of General Finite Fuzzy Automata. In Basak and Gupta (2002), the membership values of initial state designator were taken as fixed values, but it may not be same every time. To find membership values of initial state designator for General Finite Fuzzy Automata we have proposed Algorithm 4.1.

Algorithm 4.1: For a given General Finite Fuzzy Automata with n states $\tilde{\mathcal{M}} = \{S, \Sigma, \mathcal{R}, \mathcal{Z}, \tilde{\delta}, \omega, \{A(a): a \in \Sigma\}, \mathcal{F}_1, \mathcal{F}_2, \alpha, \eta^{\mathcal{F}}\}$, algorithm to compute membership values of initial state designator α is given as:

Step1: Consider all the elements of starting state set $\mathcal{R} = Q_{act}(t=0)$, where $Q_{act}(t=0) = \{(s_k, \mu^{t=0}(s_k) = 1)\}$ such that $1 \le k \le n$.
Step2: Assign $i = 1$. Find length of string $|y|$. If $|y| = |y_1 y_2 ... y_i ... y_l|$ then $|y| = l$.
Step3: Find all possible active states for input y_i at $t = t+1; 1 \le i \le l$ and compute $\mu^{t+1}(s_j) = \mathcal{F}_2[\mathcal{F}_1(\mu^t(s_k), \delta(s_k, y_i, s_j))]$ where, s_j is active state at time $t+1$ and s_k is active state at time t.
Step4: Compute $Q_{act}(t = t+1) = \{(s_j, \mu^{t+1}(s_j)) \mid \exists\, \delta(s_k, y_i, s_j)\}; 1 \le j \le n$ and $k \ne j$.
Step5: Update membership values of states at time $t = i$ in initial state designator $\alpha(y_1 y_2 ... y_l)[j] = \mu^{t+1}(s_j)$.
Step6: If $i = l$ end, else assign $i = i+1$ and go to step 3.

Algorithm 4.1 Pseudo Code
Input: Set of all states S, set of final states $\mathcal{F}$, set of fuzzy initial states $\mathcal{R}$, number of total states n, alphabet Σ, transition matrices $A(a), \forall a \in \Sigma$, augmented transition function $\tilde{\delta}$, mapping function $\mathcal{F}_1$, mapping function $\mathcal{F}_2$, input string y.

Output: $\alpha(y)$
initialize $t \leftarrow 0$
initialize $\alpha(\lambda) \leftarrow$ *new Array of size n*
for $m \leftarrow 1$ to n do
    if $\mathcal{R}$ contains s_m then
        $\alpha(\lambda)[m] \leftarrow 1$
    else
        $\alpha(\lambda)[m] \leftarrow 0$
    end if
end for

initialize $activeStateSet_t \leftarrow \mathcal{R}$
set $Q_{act}(t=0) = \{(s_k, \mu^{t=0}(s_k) = 1), \forall s_k \in \mathcal{R}\}$
initialize $l \leftarrow |y|$
initialize $i \leftarrow 1$
while true do
    for each transition $\delta(s_k, y_i, s_j)$ do

if $k \neq j$ then
$activeStateSet_{t+1}.add(s_j)$
end if
end for
for s_j in $activeStateSet_{t+1}$ do
initialize $\mu^{t+1}\_intermediateSet \leftarrow new empty set$
for s_k in $activeStateSet_t$ do
$\mu^{t+1}\_intermediateSet.add\left(\mathcal{F}_1\left(\mu^t(s_k), \delta(s_k, y_i, s_j)\right)\right)$
end for
$\mu^{t+1}(s_j) = \mathcal{F}_2(\mu^{t+1}\_intermediateSet)$
$Q_{act}(t=t+1) \leftarrow \{(s_j, \mu^{t+1}(s_j))\}$
end for

for $j \leftarrow 1$ to n do
$\alpha(y_1 y_2 ... y_i)[j] \leftarrow \mu^{t+1}(s_j)$
end for

if $i = l$ then
return $\alpha(y)$
\\end
else
$i \leftarrow i+1$
$activeStateSet_t \leftarrow activeStateSet_{t+1}$
$t \leftarrow t+1$
end if
end while

Basak and Gupta (2002) proposed an algorithm to compute induced partition for given fuzzy finite automata. We have extended the algorithm to compute induced partition for General Finite Fuzzy Automata with initial and final state designator.

Algorithm 4.2: To compute the induced partition $\pi_{\mathcal{F}}$ for a given General Finite Fuzzy Automata $\tilde{\mathcal{M}} = \{S, \Sigma, \mathcal{R}, \mathcal{Z}, \tilde{\delta}, \omega, \{A(a): a \in \Sigma\}, \mathcal{F}_1, \mathcal{F}_2, \alpha, \eta^{\mathcal{F}}\}$:

Step1: Compute initial partition from final and non-final states, $\pi_0 = \{S - \mathcal{F}, \mathcal{F}\}$ where $\mathcal{F}$ is set of final states and $S - \mathcal{F}$ is set of non-final states.

Step2: If $|x| = 0$, for final state designator $\eta^{\mathcal{F}}$ assign value 1 to final states and 0 to non-final states. Assign $l = 1$.

Step3: If $|x| = l$; $|x| = |x_1 x_2 \ldots. x_l|$, $x \in \sum^*$, compute $\eta^{\mathcal{F}}(x_1 x_2 \ldots x_l) = A(x_1)\, \eta^{\mathcal{F}}(x_2 x_3 \ldots x_l)$ for all possible string of length l.

Step4: Denote the partition τ_x of state set S by dividing into blocks for which the entries of $A(x_1)\eta^{\mathcal{F}}(x_2 x_3 \ldots x_l)$ are constant and compute partition $\tau_{l-1} = \prod_{x_1 \in \sum} \prod_{|x|=l} \tau_x$; $x = x_1 x_2 \ldots x_l$.

Step5: Compute $\pi_l = \pi_{l-1}\tau_{l-1}$.

Step6: If $\pi_{l-1} = \pi_l$ then stop, it is the final partition $\pi_{\mathcal{F}}$ of state set S; else replace l by $l+1$ and go to step 3.

Algorithm 4.2 Pseudo Code

Input: Set of all states S, set of final states $\mathcal{F}$, set of fuzzy initial states $\mathcal{R}$, number of total states n, alphabet $\sum$, transition matrices $A(a)$, $\forall a \in \sum$, input string x.

Output: Induced state partition $\pi_{\mathcal{F}}$.

initialize $\pi_0 = \{S - \mathcal{F}, \mathcal{F}\}$

initialize $\eta^{\mathcal{F}}(\lambda) \leftarrow$ *new Array of size n*

for $i \leftarrow 1$ to n do

if $\mathcal{F}$ contains s_i then

$\eta^{\mathcal{F}}(\lambda)[i] \leftarrow 1$

else

$\eta^{\mathcal{F}}(\lambda)[i] \leftarrow 0$

end if

end for

$l \leftarrow 1$

while true do

for all $x \in \sum^l$ do

$\eta^{\mathcal{F}}(x) \leftarrow ProductOfFuzzyMatrices(A(x_1), \eta^{\mathcal{F}}(x_2 x_3 \ldots x_l))$ \\ see definition in section 2.

end for

initialize *etaHash* $\leftarrow$ *new hashmap*

for $s \leftarrow 1$ to n do

// length of each column is equal to number of possible x strings

initialize $columnMemberValue \leftarrow$ *Empty set of length* $|\sum|^l$

initialize $i \leftarrow 0$

for all $x \in \sum^l$ do

$columnMemberValue[i] \leftarrow \eta^{\mathcal{F}}(x)[s]$

$i \leftarrow i+1$

end for

if *etaHash* does not contains *columnMemberValue* then
initialize *etaHash[columnMemberValue]* $\leftarrow$ *Empty set*
etaHash[columnMemberValue].add({ s })
else
etaHash[columnMemberValue].add({ s })
end if
end for
$\tau_{l-1} \leftarrow$ *etaHash.values()*
$\pi_l = ProductOfPartitions(\pi_{l-1}, \tau_{l-1})$ \\ see definition in Section 2
if $\pi_{l-1} \neq \pi_l$ then
$l \leftarrow l+1$
else
$\pi_{\mathcal{F}} \leftarrow \pi_l$
end
end if
end while

4.1 Example

Consider a General Finite Fuzzy Automata with initial and final state designator $\tilde{\mathcal{M}} = \{S, \Sigma, \mathcal{R}, \mathcal{Z}, \tilde{\delta}, \omega, \{A(a) : a \in \Sigma\}, \mathcal{F}_1, \mathcal{F}_2, \alpha, \eta^{\mathcal{F}}\}$ specified as:

$S = \{s_1, s_2, ..., s_{10}\}$, $\Sigma = \{a, b\}$, $\mathcal{R} = \{(s_1,1)\ (s_2,1)\}$, $\mathcal{Z} = \varphi$, $\tilde{\delta}$: augmented transition function, ω : no output function is considered, $\mathcal{F} = \{s_4, s_5, s_6, s_8, s_9, s_{10}\}$ with transition matrices:

$$A(a) = \begin{bmatrix} 1 & 0.3 & 0.4 & 0.5 & 0 & 0.9 & 0.7 & 0 & 0 & 0 \\ 0.3 & 1 & 0.5 & 0 & 0.6 & 0.6 & 0 & 0.9 & 0 & 0 \\ 0.4 & 0.5 & 1 & 0.4 & 0.5 & 0 & 0.8 & 0 & 0.9 & 0 \\ 0.5 & 0 & 0.4 & 1 & 0.6 & 0.8 & 0.7 & 0.5 & 0 & 0 \\ 0 & 0.6 & 0.5 & 0.6 & 1 & 0 & 0 & 0.8 & 0.5 & 0 \\ 0.9 & 0.6 & 0 & 0.8 & 0 & 1 & 0.5 & 0 & 0.7 & 0.8 \\ 0.7 & 0 & 0.8 & 0.7 & 0 & 0.5 & 1 & 0.5 & 0 & 0 \\ 0 & 0.9 & 0 & 0.5 & 0.8 & 0 & 0.5 & 1 & 0.3 & 0.7 \\ 0 & 0 & 0.9 & 0 & 0.5 & 0.7 & 0 & 0.3 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.8 & 0 & 0.7 & 0 & 1 \end{bmatrix},$$

$$A(b)=\begin{bmatrix} 1 & 0.4 & 0.5 & 0.6 & 0.7 & 0.8 & 0.6 & 0 & 0.3 & 0 \\ 0.4 & 1 & 0.8 & 0 & 0.3 & 0.7 & 0.8 & 0 & 0 & 0.6 \\ 0.5 & 0.8 & 1 & 0.6 & 0.7 & 0.8 & 0.5 & 0.4 & 0.7 & 0 \\ 0.6 & 0 & 0.6 & 1 & 0.8 & 0.6 & 0.5 & 0.4 & 0.5 & 0.6 \\ 0.7 & 0.3 & 0.7 & 0.8 & 1 & 0.4 & 0.3 & 0.6 & 0 & 0.4 \\ 0.8 & 0.7 & 0.8 & 0.6 & 0.4 & 1 & 0.6 & 0 & 0.5 & 0.4 \\ 0.6 & 0.8 & 0.5 & 0.5 & 0.3 & 0.7 & 1 & 0.7 & 0 & 0.4 \\ 0 & 0 & 0.4 & 0.4 & 0.7 & 0 & 0.7 & 1 & 0.8 & 0 \\ 0.3 & 0 & 0.7 & 0.5 & 0 & 0.8 & 0 & 0.8 & 1 & 0.3 \\ 0 & 0.6 & 0 & 0.6 & 0.4 & 0.9 & 0.9 & 0 & 0.3 & 1 \end{bmatrix}.$$

$A(a)=(a_{ij}(a));a_{ij}(a)=\delta(s_i,a,s_j)$, $A(b)=(b_{ij}(b));b_{ij}(b)=\delta(s_i,b,s_j)$. Consider $\mathcal{F}_1$: minimum function *i.e.* $\mathcal{F}_1(\mu,\delta)=\min(\mu,\delta)$, $\mathcal{F}_2$: maximum function *i.e.* ${}_{i=1}^{k}\mathcal{F}_2[\mu(s_i)]=\max(\mu(s_1),\mu(s_2),...,\mu(s_k))$, k denotes number of simultaneous transitions to the active state, input string $y=a^2b$.

The active states and their membership values for initial state designator α can be calculated using Algorithm 4.1. Here, we are considering the input string $y=a^2b$ and $|y|=3$. The membership values of states at different time for $y=a^2b$ is calculated: At time $t=0$ active states are starting states, $Q_{act}(t=0)=\{(s_1,\mu^{t=0}(s_1)),(s_2,\mu^{t=0}(s_2))\}$.(Table 2).

Since $Q_{act}(t=0)=\mathcal{R}$, therefore $\mu^{t=0}(s_1)=1,\mu^{t=0}(s_2)=1.$

$$\alpha(\lambda)=[1\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0].$$

time $t=1$: At time $t=1$ on input a active states are based on transitions from state s_1 and s_2. $s_1\to s_2,s_3,s_4,s_6,s_7$ and $s_2\to s_1,s_3,s_5,s_6,s_8$, hence active states are $\{s_1,s_2,s_3,s_4,s_5,s_6,s_7,s_8\}$, therefore;

$$\mu^{t=1}(s_1)=\mathcal{F}_1\{\mu^{t=0}(s_2),\ \delta(s_2,a,s_1)\}=\min\{1,0.3\}=0.3$$

$$\mu^{t=1}(s_2)=\mathcal{F}_1\{\mu^{t=0}(s_1),\ \delta(s_1,a,s_2)\}=\min\{1,0.3\}=0.3$$

$$\mu^{t=1}(s_3)=\mathcal{F}_2[\mathcal{F}_1\{\mu^{t=0}(s_1),\ \delta(s_1,a,s_3)\},\mathcal{F}_1\{\mu^{t=0}(s_2),\ \delta(s_2,a,s_3)\}]$$

$$=max[min\{1,0.4\},min\{1,0.5\}]=0.5$$

$$\mu^{t=1}(s_4)=\mathcal{F}_1\{\mu^{t=0}(s_1),\ \delta(s_1,a,s_4)\}=\min\{1,0.5\}=0.5$$

$$\mu^{t=1}(s_5)=\mathcal{F}_1\{\mu^{t=0}(s_2),\ \delta(s_2,a,s_5)\}=\min\{1,0.6\}=0.6$$

$$\mu^{t=1}(s_6) = \mathcal{F}_2[\mathcal{F}_1\{\mu^{t=0}(s_1),\ \delta(s_1,a,s_6)\},\mathcal{F}_1\{\mu^{t=0}(s_2),\ \delta(s_2,a,s_6)\}]$$

$$= max[min\{1,0.9\}, min\{1,0.6\}] = 0.9$$

$$\mu^{t=1}(s_7) = \mathcal{F}_1\{\mu^{t=0}(s_1),\ \delta(s_1,a,s_7)\} = \min\{1,0.7\} = 0.7$$

$$\mu^{t=1}(s_8) = \mathcal{F}_1\{\mu^{t=0}(s_2),\ \delta(s_2,a,s_8)\} = \min\{1,0.9\} = 0.9$$

Initial state designator at time $t = 1$ is
$\alpha(a) = [0.3 \quad 0.3 \quad 0.5 \quad 0.5 \quad 0.6 \quad 0.9 \quad 0.7 \quad 0.9 \quad 0 \quad 0]$.

Similarly, the membership values of states at time $t = 2$ for input a and at time $t = 3$ for input b is calculated and given in Table 3 and Table 4 respectively:

Table 2. Initial state designator α for λ and string a.

time	0		1							
input	λ		a							
$Q_{act}(t)$	s_1	s_2	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
Membership value	1	1	0.3	0.3	0.5	0.5	0.6	0.9	0.7	0.9

Table 3. Initial state designator α for string a^2.

time	2									
input	a^2									
$Q_{act}(t)$	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}
Membership value	0.9	0.9	0.7	0.8	0.8	0.5	0.5	0.6	0.7	0.8

Table 4. Initial state designator α for string a^2b.

time	3									
input	a^2b									
$Q_{act}(t)$	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}
Membership value	0.7	0.7	0.8	0.8	0.8	0.8	0.8	0.7	0.7	0.6

$$\alpha(a^2) = [0.9 \quad 0.9 \quad 0.7 \quad 0.8 \quad 0.8 \quad 0.5 \quad 0.5 \quad 0.6 \quad 0.7 \quad 0.8].$$

$$\alpha(a^2b) = [0.7 \quad 0.7 \quad 0.8 \quad 0.8 \quad 0.8 \quad 0.8 \quad 0.8 \quad 0.7 \quad 0.7 \quad 0.6].$$

The final state designator is calculated using available tools in Python for calculation of fuzzy matrices. The induced partition for given GFFA is computed using algorithm 4.2.

$$\pi_0 = \{[s_1, s_2, s_3, s_7], [s_4, s_5, s_6, s_8, s_9, s_{10}]\}.$$

$$\eta^{\mathcal{F}} = [0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1]^T.$$

The final state designator for string of length 1, *i.e.*, for $|x| = 1$.

$\eta^{\mathcal{F}}(a) = A(a)\eta^{\mathcal{F}} = [0.9 \quad 0.9 \quad 0.9 \quad 1 \quad 1 \quad 1 \quad 0.7 \quad 1 \quad 1 \quad 1]^T$.

$\eta^{\mathcal{F}}(b) = A(b)\eta^{\mathcal{F}} = [0.8 \quad 0.7 \quad 0.8 \quad 1 \quad 1 \quad 1 \quad 0.7 \quad 1 \quad 1 \quad 1]^T$.

$\tau_0 = \{[s_1, s_3], [s_2], [s_7], [s_4, s_5, s_6, s_8, s_9, s_{10}]\}$.

$\pi_1 = \pi_0\tau_0 = \{[s_1, s_3], [s_2], [s_7], [s_4, s_5, s_6, s_8, s_9, s_{10}]\}$.

The final state designator for all possible strings of length 2, *i.e.*, $|x| = 2$.

$\eta^{\mathcal{F}}(a^2) = A(a)\eta^{\mathcal{F}}(a) = [0.9 \quad 0.9 \quad 0.9 \quad 1 \quad 1 \quad 1 \quad 0.8 \quad 1 \quad 1 \quad 1]^T$.

$\eta^{\mathcal{F}}(ab) = A(a)\eta^{\mathcal{F}}(b) = [0.9 \quad 0.9 \quad 0.9 \quad 1 \quad 1 \quad 1 \quad 0.8 \quad 1 \quad 1 \quad 1]^T$.

$\eta^{\mathcal{F}}(ba) = A(b)\eta^{\mathcal{F}}(a) = [0.9 \quad 0.9 \quad 0.9 \quad 1 \quad 1 \quad 1 \quad 0.8 \quad 1 \quad 1 \quad 1]^T$.

$\eta^{\mathcal{F}}(b^2) = A(b)\eta^{\mathcal{F}}(b) = [0.8 \quad 0.8 \quad 0.8 \quad 1 \quad 1 \quad 1 \quad 0.7 \quad 1 \quad 1 \quad 1]^T$.

$\tau_1 = \{[s_1, s_2, s_3], [s_7], [s_4, s_5, s_6, s_8, s_9, s_{10}]\}$.

$\pi_2 = \pi_1\tau_1 = \{[s_1, s_3], [s_2], [s_7], [s_4, s_5, s_6, s_8, s_9, s_{10}]\} = \pi_1 = \pi_{\mathcal{F}}$.

Since $\pi_2 = \pi_1$ it is the induced partition $\pi_{\mathcal{F}}$ of state set S satisfying all the conditions of theorem 3.2.

To verify we can partition state set S for all possible strings of length 3, *i.e.* $|x| = 3$.

$\eta^{\mathcal{F}}(a^3) = A(a)\eta^{\mathcal{F}}(a^2) = [0.9 \quad 0.9 \quad 0.9 \quad 1 \quad 1 \quad 1 \quad 0.8 \quad 1 \quad 1 \quad 1]^T$,

$\eta^{\mathcal{F}}(aba) = A(a)\eta^{\mathcal{F}}(ba) = [0.9 \quad 0.9 \quad 0.9 \quad 1 \quad 1 \quad 1 \quad 0.8 \quad 1 \quad 1 \quad 1]^T$,

$\eta^{\mathcal{F}}(baa) = A(b)\eta^{\mathcal{F}}(aa) = [0.9 \quad 0.9 \quad 0.9 \quad 1 \quad 1 \quad 1 \quad 0.8 \quad 1 \quad 1 \quad 1]^T$,

$\eta^{\mathcal{F}}(b^3) = A(b)\eta^{\mathcal{F}}(b^2) = [0.8 \quad 0.8 \quad 0.8 \quad 1 \quad 1 \quad 1 \quad 0.8 \quad 1 \quad 1 \quad 1]^T$,

$\eta^{\mathcal{F}}(bab) = A(b)\eta^{\mathcal{F}}(ab) = [0.9 \quad 0.9 \quad 0.9 \quad 1 \quad 1 \quad 1 \quad 0.8 \quad 1 \quad 1 \quad 1]^T$,

$\eta^{\mathcal{F}}(aab) = A(a)\eta^{\mathcal{F}}(ab) = [0.9 \quad 0.9 \quad 0.9 \quad 1 \quad 1 \quad 1 \quad 0.8 \quad 1 \quad 1 \quad 1]^T$,

$\eta^{\mathcal{F}}(bba) = A(b)\eta^{\mathcal{F}}(ba) = [0.9 \quad 0.9 \quad 0.9 \quad 1 \quad 1 \quad 1 \quad 0.8 \quad 1 \quad 1 \quad 1]^T$,

$\eta^{\mathcal{F}}(abb) = A(a)\eta^{\mathcal{F}}(b^2) = \begin{bmatrix}0.9 & 0.9 & 0.9 & 1 & 1 & 1 & 0.8 & 1 & 1 & 1\end{bmatrix}^T$.
$\tau_2 = \{[s_1, s_2, s_3],[s_7],[s_4, s_5, s_6, s_8, s_9, s_{10}]\}$

$\pi_3 = \pi_2\tau_2 = \{[s_1, s_3],[s_2],[s_7],[s_4, s_5, s_6, s_8, s_9, s_{10}]\} = \pi_2 = \pi_{\mathcal{F}}$. Thus, the induced partition $\pi_{\mathcal{F}}$ is the largest SP partition for $\tilde{\mathcal{M}}$ refining $\mathcal{F}$. By definition, the quotient machine will be the minimal machine for given GFFA.

Construct Quotient machine $\tilde{\mathcal{M}}/\pi_{\mathcal{F}} = \{S', \Sigma, \mathcal{R}', \mathcal{Z}, \tilde{\delta}', \omega, \{A'(a) : a \in \Sigma\}, \mathcal{F}_1', \mathcal{F}_2', \alpha', \eta^{\mathcal{F}'}\}$ using induced partition, where

$$S' = \{B_1, B_2, B_3, B_4\},\ B_1 = [s_1, s_3],\ B_2 = [s_2],\ B_3 = [s_7],\ B_4 = [s_4, s_5, s_6, s_8, s_9, s_{10}].$$

Since $\alpha(a^2b) = \begin{bmatrix}0.7 & 0.7 & 0.8 & 0.8 & 0.8 & 0.8 & 0.8 & 0.7 & 0.7 & 0.6\end{bmatrix}$, $\alpha'(a^2b)$ is obtained by each of the sub-vectors into which α is partitioned by the blocks of $\pi_{\mathcal{F}}$ using row max.
$\alpha'(a^2b) = \begin{bmatrix}0.8 & 0.7 & 0.8 & 0.8\end{bmatrix}$, $\mathcal{F}' = \{B_4\}$ *i.e.*, set of blocks of $\pi_{\mathcal{F}}$ partitioning $\mathcal{F}$.

$$A'(a) = \begin{bmatrix} 1 & 0.5 & 0.8 & 0.9 \\ 0.5 & 1 & 0 & 0.9 \\ 0.8 & 0 & 1 & 0.7 \\ 0.9 & 0.9 & 0.7 & 1 \end{bmatrix}$$

$$A'(b) = \begin{bmatrix} 1 & 0.8 & 0.6 & 0.8 \\ 0.8 & 1 & 0.8 & 0.7 \\ 0.6 & 0.8 & 1 & 0.7 \\ 0.8 & 0.7 & 0.9 & 1 \end{bmatrix}$$

$$A'(a^2) = \begin{bmatrix} 1 & 0.9 & 0.8 & 0.9 \\ 0.9 & 1 & 0.7 & 0.9 \\ 0.8 & 0.7 & 1 & 0.8 \\ 0.9 & 0.9 & 0.8 & 1 \end{bmatrix} \qquad A'(a^2b) = A'(a^2)A'(b) = \begin{bmatrix} 1 & 0.9 & 0.9 & 0.9 \\ 0.9 & 1 & 0.9 & 0.9 \\ 0.8 & 0.8 & 1 & 0.8 \\ 0.9 & 0.9 & 0.9 & 1 \end{bmatrix}$$

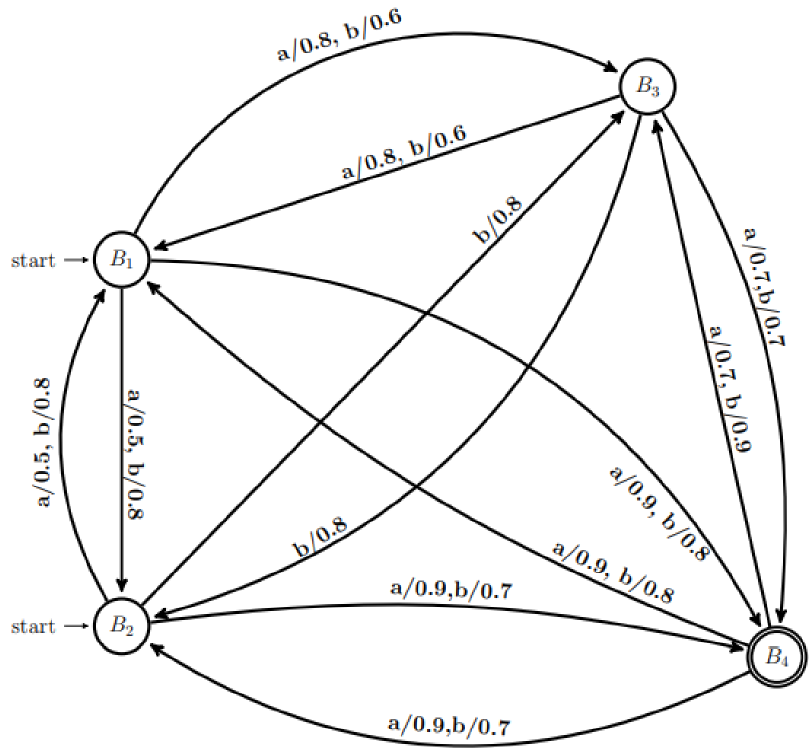

Figure 1. Minimal general finite fuzzy automata.

The membership value of string a^2b for given GFFA $\tilde{\mathcal{M}}$ is $\mu_{B_{\tilde{\mathcal{M}}}}(a^2b) = \alpha(a^2b)\eta^{\mathcal{F}}(a^2b)$ $= [0.7 \;\; 0.7 \;\; 0.8 \;\; 0.8 \;\; 0.8 \;\; 0.8 \;\; 0.8 \;\; 0.7 \;\; 0.7 \;\; 0.6][0.9 \;\; 0.9 \;\; 0.9 \;\; 1 \;\; 1 \;\; 1 \;\; 0.8 \;\; 1 \;\; 1 \;\; 1]^T = 0.8$

The membership value of language a^2b for quotient machine $\tilde{\mathcal{M}}/\pi_{\mathcal{F}}$ is $\mu_{B_{\tilde{\mathcal{M}}'}}(a^2b) = \alpha'(a^2b)\eta^{\mathcal{F}'}(a^2b) = [0.8 \;\; 0.7 \;\; 0.8 \;\; 0.8][0.9 \;\; 0.9 \;\; 0.8 \;\; 1]^T = 0.8$ or membership value can also be calculated as;

$$\mu_{B_{\tilde{\mathcal{M}}'}}(a^2b) = \alpha'(a^2b)A'(a^2b)\eta^{\mathcal{F}'}$$

$$= [0.8 \;\; 0.7 \;\; 0.8 \;\; 0.8]\begin{bmatrix} 1 & 0.9 & 0.9 & 0.9 \\ 0.9 & 1 & 0.9 & 0.9 \\ 0.8 & 0.8 & 1 & 0.8 \\ 0.9 & 0.9 & 0.9 & 1 \end{bmatrix}[0 \;\; 0 \;\; 0 \;\; 1]^T = 0.8.$$

Since $\mu_{B_{\tilde{\mathcal{M}}}}(a^2b) = \mu_{B_{\tilde{\mathcal{M}}'}}(a^2b)$, GFFA $\tilde{\mathcal{M}}$ and quotient machine $\tilde{\mathcal{M}}/\pi_{\mathcal{F}}$ are behaviourally equivalent.

Hence, minimal machine $\tilde{\mathcal{M}}_{\min} \equiv \tilde{\mathcal{M}} / \pi_{\mathcal{F}}$.

5. Conclusion

In this paper we have proposed General Finite Fuzzy Automata with initial and final state designator and proposed algorithm to find the membership value of states of initial state designator of the GFFA. The membership values calculated for initial state designator are more accurate as it depends on membership value of states and weight of transition. We have applied substitution property for partitioning the states of GFFA. Algorithm with its pseudo code is proposed to calculate induced partition for Quotient General Finite Fuzzy Automata and the minimal behaviourally equivalent GFFA is obtained. The method can be extended to minimize General Fuzzy Automata with outputs, Intuitionistic General Fuzzy Automata, Multiset Fuzzy Automata in the near future. A comparison of minimization can be done with existing methods in detail. The complexity of algorithm can be compared with other existing minimization algorithms proposed in the literature.

Conflict of Interest

The authors confirm that there is no conflict of interest to declare for this publication

Acknowledgments

The authors are very grateful to the reviewers whose valuable remarks and suggestions that have made a significant contribution in improving the quality of the manuscript. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

Abolpur, K., & Zahedi, M.M. (2012). Isomorphism between two BL-general fuzzy automata. *Soft Computing*, *16*, 729-736. https://doi.org/10.1007/s00500-011-0782-4.

Bacon, G.C. (1964). The decomposition of stochastic automata. *Information and Control*, *7*(3), 320-329. https://doi.org/10.1016/S0019-9958(64)90374-2.

Balle, B., & Rabusseau, G. (2020). Approximate minimization of weighted tree automata. *Information and Computation*. https://doi.org/10.1016/j.ic.2020.104654.

Basak, N.C., & Gupta, A. (2002). On quotient machines of a fuzzy automaton and the minimal machine. *Fuzzy Sets and Systems*, *125*(2), 223-229. https://doi.org/10.1016/S0165-0114(01)00064-1.

Cheng, W., & Mo, Z. (2004). Minimization algorithm of fuzzy finite automata. *Fuzzy Sets and Systems*, *141*(3), 439-448. https://doi.org/10.1016/S0165-0114(02)00607-3.

Choubey, A., & Ravi, K.M. (2013). Minimization of deterministic finite automata with vague (final) states and intuitionistic fuzzy (final) states. *Iranian Journal of Fuzzy Systems*, *10*(1), 75-88.

Ciric, M., Stamenkovic, A., Ignjatovic, J., & Petkovic, T. (2010). Fuzzy relation equations and reduction of fuzzy automata. *Journal of Computer and System Sciences*, *76*(7), 609-633.

Doostfatemeh, M., & Kremer, S.C. (2005). New directions in fuzzy automaton. *International Journal of Approximate Reasoning*, *38*(2), 175-214. https://doi.org/10.1016/j.ijar.2004.08.001.

Doostfatemeh, M., & Kremer, S.C. (2006). General fuzzy automata, new efficient acceptors for fuzzy languages. *IEEE International Conference on Fuzzy Systems* (pp. 2097-2103). IEEE. Vancouver, BC, Canada. doi: 10.1109/FUZZY.2006.1681991.

Dubey, M.K., Tiwari, S.P., & Sostak, A. (2020). Categories of quantale-valued fuzzy automata: determinization and minimization. *Journal of Applied Mathematics and Computing*, *63*, 771-785. https://doi.org/10.1007/s12190-020-01338-3.

Ghorani, M., & Moghari, S. (2021). Decidability of the minimization of fuzzy tree automata with membership values in complete lattices. *Journal of Applied Mathematics and Computing*. http://doi.org/10.1007/s12190-021-01529-6.

Horry, M., & Zahedi, M.M. (2013). Some (fuzzy) topologies on general fuzzy automata. *Iranian Journal of Fuzzy Systems*, *10*(6), 73-89. https://dx.doi.org/10.22111/ijfs.2013.1317.

Ignjatovic, J., Ciric, M., & Jancic, Z. (2018). Weighted finite automata with output. *Soft Computing*, *22*(4), 1121-1138. https://doi.org/10.1007/s00500-017-2493-y.

Kavikumar, J., Tiwari, S.P., Ebas, N.A., & Shamsidah, A.H.N. (2019). General fuzzy finite switchboard automata. *New Mathematics and Natural Computation*, *15*(2), 283-305. https://doi.org/10.1142/S1793005719500157.

Lee, H.S. (2000). Minimizing fuzzy finite automata. *Ninth IEEE International Conference on Fuzzy Systems* (Vol. 1, pp. 65-70). IEEE. San Antonio, TX, USA. https://dx.doi.org/10.1109/FUZZY.2000.838635.

Li, L., & Qiu, D. (2015). On the state minimization of fuzzy automata. *IEEE Transactions on Fuzzy Systems*, *23*(2), 434-443. https://doi.org/10.1109/TFUZZ.2014. 2315620.

Li, Y., & Pedrycz, W. (2007). Minimization of lattice finite automata and its application to the decomposition of lattice languages. *Fuzzy Sets and Systems*, *158*(13), 1423-1436. http://dx.doi.org/10.1016/j.fss.2007.03.003.

Malik, D.S., Mordeson, J.N., & Sen, M.K. (1999). Minimization of fuzzy finite automata. *Information Sciences*, *113*(3-4), 323-330. https://doi.org/10.1016/S0020-0255(98)10073-7.

Mendivil, J.R.G. (2018). Condition for minimal fuzzy deterministic finite automata via brzozowski's procedure. *IEEE Transactions on Fuzzy Systems*, *26*(4), 2409-2420. https://doi.org/10.1109/TFUZZ.2017.2775601.

Mizumoto, M., Toyoda, J., & Tanaka, K. (1969). Some considerations on fuzzy automata. *Journal of Computer and System Sciences*, *3*(4), 409-422. https://doi.org/10.1016/S0022-0000(69)80029-2.

Peeva, K.G., & Zahariev, Z. (2008). Computing behavior of finite fuzzy machines-algorithm and its application to reduction and minimization. *Information Sciences*, *178*(21), 4152-4165.

Santos, E.S. (1972). On reduction of maximin machines. *Journal of Mathematical Analysis and Applications*, *40*(1), 60-78.

Shamsizadeh, M., & Zahedi, M.M. (2019). Bisimulation of type 2 for BL-general fuzzy automata. *Soft Computing*, *23*, 9843-9852. https://doi.org/10.1007/s00500-019-03812-y.

Shamsizadeh, M., & Zahedi, M.M. (2015). Minimal intuitionistic general L-fuzzy automata. *Italian Journal of Pure and Applied Mathematics*, *35*, 155-186.

Shamsizadeh, M., & Zahedi, M.M. (2016). Intuitionistic general fuzzy automata. *Soft Computing*, *20*, 3505-3519. https://doi.org/10.1007/s00500-015-1969-x.

Stamenkovic, A., Ciric, M., & Basic, M. (2018). Ranks of fuzzy matrices. applications in state reduction of fuzzy automata. *Fuzzy Sets and Systems*, *333*, 124-139. https://doi.org/10.1016/j.fss.2017.05.028.

Topencharov, V.V., & Peeva, K.G. (1981). Equivalence, reduction and minimization of finite fuzzy-automata. *Journal of Mathematical Analysis and Applications*, *84*(1), 270-281.

Tripathi, A., & Kaur, R. (2019). A review of state minimization and state reduction techniques in fuzzy automata. *AIP Conference Proceedings* (Vol. 2061, No. 1, p. 020033). https://doi.org/10.1063/1.5086655.

Wee, W.G., & Fu, K.S. (1969). A formulation of fuzzy automata and its application as a model of learning systems. *IEEE Transaction on Systems Science and Cybernetics*, *5*(3), 215-223, https://doi.org/10.1109/TSSC.1969.300263.

Wee, W.G. (1967). *On generalizations of adaptive algorithm and application of fuzzy sets concept to pattern classification*. Ph D. thesis, Purdue University.

Zadeh, L.A. (1965). Fuzzy sets. *Information and Control*, *8*(3), 338-353. https://doi.org/10.1016/S0019-9958(65)90241-X.

Zhiwen, M., & Xiaolei, H. (2007). Minimization of mizumoto automata. *Fuzzy Information and Engineering*, *40*, 739-743. https://doi.org/10.1007/978-3-540-71441-5_79.