Intern. J. Fuzzy Mathematical Archive
Vol. 3, 2013, 58-67
ISSN: 2320 –3242 (P), 2320 –3250 (online)
Published on 30 December 2013
www.researchmathsci.org

International Journal of
Fuzzy Mathematical Archive

Myhill Nerode Theorem for Fuzzy Automata (Min-max Composition)

***V. Ramaswamy*[1] and *K. Chatrapathy*[2]**

[1]Bapuji College of Engineering and Technology, VTU-Belgaum, India
Email: researchwork04@yahoo.com

[2]School of Engineering and Technology, JAIN University, India
Email: kcpathy@yahoo.in

Received 20 December 2013; accepted 28 December 2013

Abstract. In this paper, Myhill Nerode theorem of finite automaton has been extended to fuzzy automaton where the composition considered is min-max composition. In the case of max-min composition, it has already been proved that if L is a fuzzy regular language, then for any $\alpha \in [0, 1]$, $L_\alpha = L(D_\alpha(M))$ [3]. In the case of max-product composition L_α is only a subset of $L(D_\alpha(M))$. But still Myhill Nerode theorem has been extended to max-product composition [4]. In the case of max-average composition, L_α is not even contained in $L(D_\alpha(M))$. This lead to lots of challenges and we had to resort to splitting to prove the analogue of Myhill Nerode Theorem for max-average composition. In a similar line, an attempt has been made in this paper to study the behavior of fuzzy automata under min-max composition and to prove the analogue of Myhill Nerode Theorem for min - max composition. An algorithm to compute L(s) for any string s is also developed.

***Keywords*:** Monoid, min-max composition, finite automaton, equivalence class, fuzzy regular language, fuzzy automaton

AMS Mathematics Subject Classification (2010): 68Q45, 68Q70

1. Introduction

Let A be a finite non empty set. A *fuzzy automaton over A* is a 4-tuple M = (Q, f, I, F) where Q is a finite nonempty set, f is a fuzzy subset of Q x A x Q, I and F are fuzzy subsets of Q. In other words, f: Q x A x Q → [0, 1] and I, F: Q →[0, 1].

Let S be a free monoid with identity element e generated by A. If $s \in S$, then s can be written as $a_1a_2\ldots a_n$ where $a_i \in A$. Here n is called the length of s and we write |s| = n. We now extend f to a function f^*: Q x S x Q → [0, 1] defined as

$$f^*(q, e, p) = 0 \text{ if } q = p, \quad 1 \text{ otherwise.}$$

$$f^*(q, sa, p) = \bigwedge_{r \in Q} [\, f^*(q, s, r) \vee f(r, a, p)] \quad (s \in S, a \in A)$$

It can be shown that $f^*(q, a, p) = f(q, a, p)$ for all $p, q \in Q$ and for all $a \in A$.

Definition 1.1. Let M = (Q, f^*, I, F) be a fuzzy automaton over S. We define the *language accepted by* M denoted by L (M) to be a fuzzy subset of S defined as L (M) (s) = I o f_s^* o F for all s ∈ S. Here o denotes min-max composition.

Definition 1.2. A fuzzy subset L of S is said to be a *fuzzy regular language* if L= L(M) where M is a fuzzy automaton over S.

2. Myhill Nerode Theorem for Fuzzy Automata

Let S be a monoid with identity element e and L be a fuzzy subset of S. Then the following statements are equivalent.

(i) L is a fuzzy regular language.

(ii) L can be expressed as a fuzzy union

$L = (\delta_1)_L \cup (\delta_2)_L \cup \ldots \cup (\delta_t)_L$ where $\delta_1, \delta_2, \ldots \delta_t \in [0, 1]$. For each i = 1,2…t, $(\delta i)_L = \delta i$. $L_{\delta i}$ where $L_{\delta i} = U\, [s]_{\delta i}$.

This union is a set theoretic union and $[s]_{\delta i}$ denotes the equivalence class of s of a right invariant equivalence relation of finite index in $L_{\delta i}$.

(iii) Define a relation R_L as follows.

If s, t ε S, then s R_L t if and only if for all u ∈ S and for all α ∈ [0, 1] , L(su) ≥ α only when L(tu) ≥ α. Then R_L is a right invariant equivalence relation of finite index.

Proof: (i) → (ii)

Since L is a fuzzy regular language, we have L = L (M) where M = (Q, f*, I, F) is a fuzzy automaton. Consider any α ∈ [0, 1]. With M and α, we associate a non-deterministic automaton D_α(M) = (Q, d_α, I_α, F_α) where

d_α: Q x S $\rightarrow 2^Q$ is defined as d_α (q, s) = {p ∈ Q | f* (q, s, p) ≥ α},

I_α = {p ∈ Q | I (p) ≥ α} and

F_α = {p ∈ Q | F (p) ≥ α}.

For the sake of simplicity, we will denote L (D_α (M)) by L_α(M).

Let s ∈ L_α. Then L(s) = L(M)(s) ≥ α. ie (I o f_s* o F) ≥ α which means

$$\bigwedge_{p \in Q} [(f_s^* \text{ o } F)(p) \vee I(p)] \geq \alpha$$

This means for any state p ∈ Q, I (p) ≥ α OR (f_s* o F) (p) ≥ α. This leads to the following three cases:

Case A: I (p) ≥ α and (f_s* o F) (p) ≥ α

Case B: I (p) < α and (f_s* o F) (p) ≥ α

Case C: I (p) ≥ α and (f_s* o F) (p) < α

We now consider each case separately.

Case A: I (p) ≥ α and (f_s* o F) (p) ≥ α. In this case p ∈ I_α.

Now (f_s* o F) (p) ≥ α means

$$\bigwedge_{r \in Q} [(f_s^*(p, r) \vee F(r)] \geq \alpha$$

This leads to the following three cases:

Case A_1: f_s* (p, r) ≥ α and F (r) ≥ α

Case A_2: f_s* (p, r) ≥ α and F (r) < α

Case A_3: f_s* (p, r) < α and F (r) ≥ α

Case A_1: f_s* (p, r) = f* (p, s, r) ≥ α and F(r) ≥ α

First alternative means $r \in d_\alpha(p, s)$. $F(r) \geq \alpha$ means $r \in F_\alpha$.
Thus $r \in d_\alpha(p, s) \cap F_\alpha$. Hence $d_\alpha(p, s) \cap F_\alpha \neq \phi$ where $p \in I_\alpha$. This proves that $s \in L(D\alpha(M)) = L\alpha(M)$. (1)

Case A₂: $f_s^*(p, r) \geq \alpha$ and $F(r) < \alpha$
Let $F(r) = \beta < \alpha$. Then $r \in F_\beta$. $I(p) \geq \alpha > \beta$ means $p \in I_\beta$. Also $f_s^*(p, r) \geq \alpha > \beta$ means $r \in d_\beta(p, s)$.
Thus $r \in d_\beta(p, s)$ and $r \in F_\beta$ so that $d_\beta(p, s) \cap F_\beta \neq \phi$ where $p \in I_\beta$.
This proves that $s \in L(D_\beta(M)) = L_\beta(M)$. (2)

Case A₃: $f_s^*(p, r) < \alpha$ and $F(r) \geq \alpha$
Let $f_s^*(p, r) = \gamma < \alpha$. Then $f^*(p, s, r) = \gamma$ so that $r \in d_\gamma(p, s)$
$F(r) \geq \alpha > \gamma$ means $r \in F_\gamma$. $I(p) \geq \alpha > \gamma$ means $p \in I_\gamma$
Thus $r \in d_\gamma(p, s) \cap F_\gamma$ so that $d_\gamma(p, s) \cap F_\gamma \neq \phi$ where $p \in I_\gamma$. This proves that $s \in L(D_\gamma(M)) = L_\gamma(M)$ (3)

Case B: $I(p) < \alpha$ and $(f_s^* \circ F)(p) = \bigwedge_{r \in Q} [(f_s^*(p, r) \vee F(r)] \geq \alpha$.

This leads to the following three cases.

Case B₁: $f_s^*(p, r) \geq \alpha$ and $F(r) \geq \alpha$
Case B₂: $f_s^*(p, r) \geq \alpha$ and $F(r) < \alpha$
Case B₃: $f_s^*(p, r) < \alpha$ and $F(r) \geq \alpha$

Case B₁: $f_s^*(p, r) \geq \alpha$ and $F(r) \geq \alpha$. We already have $I(p) < \alpha$.
Let $I(p) = \lambda < \alpha$. This implies $p \in I_\lambda$. Now $F(r) \geq \alpha > \lambda$ means $r \in F_\lambda$ and $f_s^*(p, r) = f^*(p, s, r) \geq \alpha > \lambda$ means $r \in d_\lambda(p, s)$.
Thus $r \in d_\lambda(p, s) \cap F_\lambda$ so that $d_\lambda(p, s) \cap F_\lambda \neq \phi$ where $p \in I_\lambda$. This proves that $s \in L(D_\lambda(M)) = L_\lambda(M)$. (4)

.**Case B₂**: $f_s^*(p, r) \geq \alpha$ and $F(r) < \alpha$. We already have $I(p) < \alpha$.
Let $I(p) = \rho < \alpha$. Then $p \in I_\rho$. Let $F(r) = \varphi < \alpha$. Then $r \in F_\varphi$.
If $\rho > \varphi$, then $I_\rho \subseteq I_\varphi$ so that $p \in I_\varphi$. Also $f_s^*(p, r) = f^*(p, s, r) \geq \alpha > \rho > \varphi$ which means $r \in d_\varphi(p, s)$. Thus there exists $p \in I_\varphi$ such that $d_\varphi(p, s) \cap F_\varphi \neq \phi$. This proves that $s \in L(D_\varphi(M)) = L_\varphi(M)$. (5)
If $\rho < \varphi$, then $r \in F_\varphi \subseteq F_\rho$ Also $f^*(p, s, r) \geq \alpha > \rho$ implies $r \in d_\rho(p, s)$. $I(p) = \rho$ means $p \in I_\rho$. Thus $r \in d_\rho(p, s) \cap F_\rho$ so that $d_\rho(p, s) \cap F_\rho \neq \phi$ where $p \in I_\rho$. This proves that $s \in L(D_\rho(M)) = L_\rho(M)$. (6)

Case B₃: $f_s^*(p, r) < \alpha$ and $F(r) \geq \alpha$. We already have $I(p) < \alpha$.
Let $I(p) = \pi < \alpha$. Then $p \in I_\pi$ and $F(r) \geq \alpha > \pi$ implies $r \in F_\pi$.
Let $f_s^*(p, r) = f^*(p, s, r) = \mu < \alpha$.
If $\mu \leq \pi$, then $F(r) \geq \alpha > \mu$ implies $r \in F\mu$ and $f^*(p, s, r) = \mu$ means $r \in d_\mu(p, s)$.
Also $I(p) = \pi \geq \mu$ means $p \in I_\mu$. Thus $r \in d_\mu(p, s) \cap F\mu$ so that $d_\mu(p, s) \cap F\mu \neq \phi$ where $p \in I_\mu$. This proves that $s \in L(D_\mu(M)) = L_\mu(M)$. (7)
If $\mu > \pi$, then $f^*(p, s, r) = \mu > \pi$ means $r \in d_\pi(p, s)$. Thus $r \in d_\pi(p, s) \cap F_\pi$ so that $d_\pi(p, s) \cap F_\pi \neq \phi$ where $p \in I\pi$.
This proves that $s \in L(D_\pi(M)) = L_\pi(M)$. (8)

Case C: $I(p) \geq \alpha$ and $(f_s^* \circ F)(p) < \alpha$.
This implies $f_s^*(p, r) < \alpha$ and $F(r) < \alpha$.

We already have $I(p) \geq \alpha$.

Let $f_s^*(p, r) = \nu < \alpha$ and $F(r) = \Omega < \alpha$.

First assume that $\nu > \Omega$. Now $F(r) = \Omega$ means $r \in F_\Omega$. Also

$f^*(p, s, r) = \nu \geq \Omega$ means $r \in d_\Omega(p, s)$. Hence $d_\Omega(p, s) \cap F_\Omega \neq \phi$. Also

$I(p) \geq \alpha > \nu > \Omega$ means $p \in I_\Omega$. Hence

$s \in L(D_\Omega(M)) = L_\Omega(M)$. (9)

. Suppose $\nu < \Omega$. Now $F(r) = \Omega > \nu$ means $r \in F_\nu$. $f^*(p, s, r) = \nu$ means

$r \in d_\nu(p, s)$ so that $d_\nu(p, s) \cap F_\nu \neq \phi$. Also $I(p) \geq \alpha > \nu$ means $p \in I_\nu$.

Hence $s \in L(D_\nu(M)) = L_\nu(M)$. (10)

From (1), (2), (3), (4), (5), (6), (7), (8), (9), and (10) it follows that

$L_\alpha \subseteq L_\alpha(M) \cup L_\beta(M) \cup L_\gamma(M) \cup L_\lambda(M) \cup L\varphi(M) \cup L\rho(M) \cup L\mu(M) \cup L_\pi(M) \cup L_\Omega(M) \cup L_\nu(M) \cup L_\sigma(M) \cup L_\eta(M)$ where $\alpha, \beta, \gamma, \lambda, \varphi, \rho, \pi, \nu, \Omega, \mu, \sigma, \eta \in [0,1]$.

Since each of the languages $L_\alpha(M)$, $L_\beta(M)$, $L_\gamma(M)$, …… $L_\pi(M)$ are fuzzy regular languages accepted by non-deterministic automata $D_\alpha(M)$, $D_\beta(M)$, $D_\gamma(M)$, …… $D_\pi(M)$ respectively, Myhill Nerode theorem for finite automata is applicable for each automaton. Let $Q = \{q_0, q_1, q_2 ….. q_n\}$. For every $s \in S$, the possible values of $L(s)$ are $I(q_0)$, $I(q_1)$,…$I(q_n)$, $f(q_i, a_j, q_k)$ ($q_i, q_k \in Q$, $a_j \in A$), $F(q_0)$, $F(q_1)$,…$F(q_n)$. Denote these fixed values (after arranging them in non decreasing order) by $\delta_1, \delta_2….\delta_t$. So, there can be only finitely many values of $L(s)$ ($s \in S$). Then $\delta_1, \delta_2… \delta_t \in [0, 1]$ and for each δi ($1 \leq i \leq t$),

$L_{\delta i} \subseteq (L_\alpha(M) \cup L_\beta(M) \cup L_\gamma(M) \cup L_\lambda(M) \cup L\varphi(M) \cup L\rho(M) \cup L\mu(M) \cup L_\pi(M) \cup L_\Omega(M) \cup L_\nu(M) \cup L_\sigma(M) \cup L_\eta(M))$

Since $L(D_{\delta i}(M))$ is the language accepted by a finite automaton, by Myhill Nerode theorem for finite automata, it follows that there exists a right invariant equivalence relation R_i of finite index. Let R_i' denotes it's restriction on $L_{\delta i}$. Similarly, we obtain other restrictions like M_i', N_i', O_i', P_i', Q_i', S_i', T_i', U_i', V_i', W_i', X_i', and Y_i' from $L(D_{\alpha i}(M))$, $L(D_{\beta i}(M))$, $L(D_{\gamma i}(M))$, $L(D_{\lambda i}(M))$, $L(D_{\varphi i}(M))$, $L(D_{\rho i}(M))$, $L(D_{\pi i}(M))$, $L(D_{\nu i}(M))$, $L(D_{\Omega i}(M))$, $L(D_{\mu i}(M))$, $L(D_{\sigma i}(M))$, $L(D_{\eta i}(M))$ respectively. Note that M_i', N_i', O_i', P_i', Q_i', S_i', T_i', U_i', V_i', W_i', X_i', and Y_i' are all right invariant equivalence relations of finite index. Hence $Z_i' = M_i' \cap N_i' \cap O_i' \cap P_i' \cap Q_i' \cap S_i' \cap T_i' \cap U_i' \cap V_i' \cap W_i' \cap X_i' \cap Y_i'$ is a right invariant equivalence relation in $L_{\delta i}$ of finite index. Let $[s]_{\delta i}$ denote the equivalence class of S under this equivalence relation. Since the equivalence classes partition $L_{\delta i}$, it follows that $L_{\delta i} = \cup [s]_{\delta i}$.

Next we will prove the fact that $L = (\delta_1)_L \cup (\delta_2)_L \cup … \cup (\delta_t)_L$.

Define $(\delta_i)_L = \delta_i . L_{\delta i}$. If $s \in S$ such that $L(s) \geq \delta_i$ ($s \in L_{\delta i}$), then $(\delta_i)_L(s) = \delta_i$. Otherwise, $(\delta_i)_L(s) = 0$. We note that each $(\delta_i)_L$ is a fuzzy set. Let $s \in S$ and assume that $L(s) = \delta_i$. Now $L(s) = \delta_i \leq \delta_{i+1} \leq … \leq \delta_t$. Again, $L(s) = \delta_i \geq \delta_{i-1} \geq … \geq \delta_1$.

Hence $((\delta_1)_L \cup (\delta_2)_L \cup … \cup (\delta_t)_L)(s) = (\delta_1)_L(s) \vee (\delta_2)_L(s) \vee … (\delta_t)_L(s) = \delta_1 \vee \delta_2 \vee … \vee \delta_i = \delta_i = L(s)$. This proves that $L = (\delta_1)_L \cup (\delta_2)_L \cup … \cup (\delta_t)_L$.

Proof: (ii) → (iii).

If $s \in S$, then $s R_L s$ because for all $u \in S$ and for all $\alpha \in [0,1]$, $L(su) \geq \alpha$ only when $L(su) \geq \alpha$ is obviously true. This proves that R_L is reflexive. Clearly, R_L is symmetric. If $s R_L t$ and $t R_L v$, then for all $u \in S$ and for all $\alpha \in [0,1]$, $L(su) \geq \alpha$ only when $L(tu) \geq \alpha$ only when $L(vu) \geq \alpha$ proving that s R_L v. Hence R_L is transitive. R_L is thus an equivalence relation.

To prove R_L is right invariant, assume that s R_L t and $u \in S$. We have to prove that su R_L tu. For this, we have to prove that for all $v \in S$ and $\alpha \in [0,1]$, $L(suv) \geq \alpha$ only when $L(tuv) \geq \alpha$ which is the same as saying that $L(sz) \geq \alpha$ only when $L(tz) \geq \alpha$ where z = uv. But this is true since s R_L t.

We will now prove that R_L is of finite index. For i = 1,2,…,t, let R_i denote the right invariant equivalence relation of finite index in $L_{\delta i}$. Let $R = R_1 \cap R_2 \cap \ldots \cap R_t$. Then R is an equivalence relation of finite index. We will prove that s R t implies s R_L t. This will mean that index (R_L) ≤ index (R). Since index (R) is finite, this will prove that index(R_L) is also finite.

Assume that s R t. Consider any $u \in S$ and any $\alpha \in [0, 1]$. Suppose $su \in L_\alpha$. We have to prove that $tu \in L_\alpha$. Now $\alpha \leq L(su) = \delta_j$ (say). Then $su \in L_{\delta j}$ which is a subset of L_α. By definition of R, we have s R_j t. Since R_j is right invariant, su R_j tu. Since $L_{\delta j} = U [v]_{\delta j}$, it follows that su belongs to one of the equivalence classes of R_j and hence tu also belongs to the same equivalence class. Hence $tu \in L_{\delta j}$ and since $L_{\delta j}$ is a subset of L_α, we have $tu \in L_\alpha$.

Proof: (iii) → (i)

We have to define a fuzzy automaton M such that L = L (M). For every element $s \in S$, let [s] denote the equivalence class of s under the equivalence relation R_L.

Let $Q = \{[s] / s \in S\}$. Since R_L is of finite index, it follows that Q is a finite set. Define

$I: Q \rightarrow [0, 1]$, $f^*: Q \times S \times Q \rightarrow [0, 1]$ and $F: Q \rightarrow [0, 1]$ as follows.

I ([s]) = 0 if [s] = [e]

= 1 otherwise.

f*([s], t, [u]) = 1 if [u] = [st] , 0 otherwise.

F ([s]) = L(s).

We will first prove that F is well defined. For this, we have to prove that if [s] = [t], then L (s) = L (t). Assume that L (s) = β. We will prove that L (t) = β. Since [s] = [t], s R_L t so that $L(s) = L(se) \geq \beta$ only when $L(t) = L(te) \geq \beta$. Since $L(s) \geq \beta$, it follows that $L[t] \geq \beta$.

Assume $L[t] = \gamma > \beta$. Take $\eta = (\beta + \gamma) / 2$. Clearly, $\beta < \eta < \gamma = L[t]$. Since s R_L t, $L[t] > \eta$ implies that $L[s] \geq \eta > \beta$. But this contradicts the fact that L(s) = β. Hence our assumption that L[t] > β is wrong. Since $L[t] \geq \beta$, it follows that L[t] = β.

Take M = (Q, I, f*, F). Then M is a fuzzy automaton and it remains to prove that L = L (M). For this, we have to prove that for all $s \in S$, L (s) = L (M) (s).

We have

$$L(M)(s) = I \circ f_s^* \circ F = \bigwedge_{[t]}\{I([t]) \vee (f^*_s \circ F)([t])\}$$

$$(f^*_s \circ F)([t]) = \bigwedge_{[u]}\{f^*_s([t],[u]) \vee F([u])\} = \bigwedge_{[u]}\{f^*([t], s, [u]) \vee F([u])\}$$

Note that f*([t], s, [u]) = 0 if [ts] = [u] and 1 otherwise. Therefore, in the above expression f*([t], s, [u]) = 0 only when [ts] = [u] . In all remaining cases (ie. whenever [ts] ≠ [u]) the term $f^*([t], s, [u]) \vee F([u])$ becomes 1. Thus the above equation becomes

$(f^*_s \text{ o } F)([t]) = F([u])$
$= L(ts)$ (since $F([s]) = L(s)$).

Hence $L(M)(s) = \wedge_{[t]} \{I([t]) \vee (f^*_s \text{ o } F)([t])\}$

Note that $I([t]) = 0$ only when $[t] = [e]$, $I([t]) = 1$ whenever $[t] \neq [e]$. Therefore, $\{I([t]) \vee (f^*_s \text{ o } F)([t])\} = 1$ whenever $[t] \neq [e]$ and $\{I([t]) \vee (f^*_s \text{ o } F)([t])\} = (f^*_s \text{ o } F)([t])$ when $[t] = [e]$. Thus the above equation becomes

$L(M)(s) = (f^*_s \text{ o } F)([t])$ where $[t] = [e]$.
$= L(ts)$ (by the above result)
$= L(es)$ (since $I[t] = 0$ when $[t]=[e]$ and R_L is a right invariant relation, $[ts] = [es]$)
$= L(s)$

Thus for all s all $s \in S$, $L(s) = L(M)(s)$. This proves that $L = L(M)$.

3. Implementation

The algorithm to compute $L(s) = L(M)(s)$ for any string s of arbitrary length and any fuzzy automata M with any number of states is developed and implemented in C++. Following procedures are used to compute $f^*(q_{i,} s, q_j)$ and $L(s)$ for all $s \in S$ and $q_{i,}, q_j \in Q$.

Procedure MinMax(i,j,X,Y). This procedure computes and returns the min-max composition value of row-I of matrix X and column-j of matrix Y. X and Y are the n x n transition matrices, min, temp and r are temporary variables.

1. min = ∞
2. for r = 0 to n-1 do
   2.1 if (X[i][r] ≥ Y[r][j]) then
         temp = X[i][j]
       else
         temp = Y[i][j]
   2.2 if (min>temp) then
       min=temp
3. return min

Procedure computeFstar (s). This procedure computes f* - matrix for the input string s and stores it in n x n matrix A. F0 and F1 are the transition matrices for the input symbols 0 and 1 respectively. The procedure call **COPY(X, Y)** copies the matrix X to matrix Y. B is the temporary matrix of size n x n. The procedure call computeFstar(X, Y, Z) computes the f*-value for each pair $(q_i, q_j) \in Q \times Q$ using transition matrices X , Y and stores the result in the matrix Z.

1. if (s[0]='0') then
      COPY (A, F0)
   else
      COPY (A, F1)
2. for i = 1 to (length(s) – 1) do
   if (s[i] = '0')
      computeFstar(A, F0, B)
   else
      computeFstar(A, F1, B)

else
3. COPY(A, B).
4. Exit

Procedure computeFstarCompF(q). This procedure computes and returns (f^*_s o F)(q) value for a given state q ∈ Q . A is the f^*_s- matrix for the string s.

1. min = ∞
2. for r = 0 to n-1 do
   2.1 temp = MAX(A[p][r], F[r])
   2.2 if (temp < min) then
       min = temp
3. return min

Procedure computeLs. This procedure computes and returns L(s) value for a given string s.

1. min = ∞
2. for p = 0 to n-1 do
   2.1 temp = computeFstarCompF(p)
   2.2 if (I [p] > temp) then
       temp = I [p]
   2.3 if (temp < min) then
       min = temp
3. return min

Procedure main( ). This procedure inputs the fuzzy automaton M = (Q, f, I, F), computes and returns L(s) value for a given input string s. F0, F1, n are transition matrix for 0, transition matrix 1 and number of states in Q respectively. Fe is the f*-matrix for e. I and F are array of size n. Ls stores the L(s) value of the input string s.

1. read number of states n
2. read arrays I and F
3. set f^*_e – matrix Fe
4. read transition matrices F0, F1
5. ch = ‘y’
6. while (ch = ‘y’) do
   6.1 Read input string s
   6.2 A = computeFstar(s)
   6.3 Ls = computeLs( )
   6.4 Print transition matrix A
   6.5 Print Ls
   6.6 read input character ch = ‘y’ to continue, ch = ‘n’ to stop
7. Exit

The program is tested for large number of fuzzy automata and strings of arbitrary length.

4. Example

Let $\Sigma = \{0, 1\}$ and $S = \Sigma^*$, the set of all strings over the alphabet Σ. Consider the fuzzy automaton $M = (Q, f, I, F)$ where $Q = \{q_0, q_1, q_2\}$, f is the fuzzy subset $f: Q \times \Sigma \times Q \rightarrow [0, 1]$ defined as

$f(q_0, 0, q_0) = 0.0$, $f(q_0, 0, q_1) = 0.8$, $f(q_0, 0, q_2) = 0.6$
$f(q_1, 0, q_0) = 0.5$, $f(q_1, 0, q_1) = 0.0$, $f(q_1, 0, q_2) = 0.7$
$f(q_2, 0, q_0) = 0.3$, $f(q_2, 0, q_1) = 0.6$, $f(q_2, 0, q_2) = 0.0$
$f(q_0, 1, q_0) = 0.0$, $f(q_0, 1, q_1) = 0.6$, $f(q_0, 1, q_2) = 0.7$
$f(q_1, 1, q_0) = 0.5$, $f(q_1, 1, q_1) = 0.0$, $f(q_1, 1, q_2) = 0.8$
$f(q_2, 1, q_0) = 0.4$, $f(q_2, 1, q_1) = 0.2$, $f(q_2, 1, q_2) = 0.0$

$I = \{q_0\}$ and F is the fuzzy subset of Q defined as $F(q_1) = 0.4$ and $F(q_2) = 0.9$.
For any string w = sa of length two or more we will calculate $f^*(q_i, w, q_j)$ as follows:

$$f^*(q, sa, p) = \bigwedge_{r\in Q} [\, f^*(q, s, r) \vee f(r, a, p)] \quad (s \in S, a \in A, q_i, q_j \in Q)$$

After computing f*-matrix for a given string s, we will compute L(M)(s) as follows:

$$\begin{aligned} L(M)(s) &= I \circ f_0^* \circ F \\ &= \bigwedge_{p\in Q} [\, I(p) \vee (f_s^* \circ F)(p)\,] \\ &= [\, I(q_0) \vee (f_s^* \circ F)(q_0)\,] \wedge [\, I(q_1) \vee (f_s^* \circ F)(q_1)\,] \wedge [\, I(q_2) \vee (f_s^* \circ F)(q_2)\,] \\ &= (f_s^* \circ F)(q_1) \wedge (f_s^* \circ F)(q_2) \end{aligned}$$

Therefore,for any string $s\in S$ $L(M)(s) = (f_s^* \circ F)(q_1) \wedge (f_s^* \circ F)(q_2)$ (11)

$$\begin{aligned} (f_s^* \circ F)(q_1) &= \bigwedge_{r\in Q} [F(r) \vee f_s^*(q_1, r)] \\ &= [F(q_0) \vee f_s^*(q_1, q_0)] \wedge [F(q_1) \vee f_s^*(q_1, q_1)] \wedge [F(q_2) \vee f_s^*(q_1, q_2)] \\ &= 0.4 \wedge f_s^*(q_1, q_0) \wedge [0.9 \vee f_s^*(q_1, q_2)] \end{aligned}$$

Therefore, for any string $s \in S$, $(f_s^* \circ F)(q_1) = 0.4 \wedge f_s^*(q_1, q_0) \wedge [0.9 \vee f_s^*(q_1, q_2)])$ (12)

$$\begin{aligned} (f_s^* \circ F)(q_2) &= \bigwedge_{r\in Q} [F(r) \vee f_s^*(q_{2,}, r)] \\ &= [F(q_0) \vee f_s^*(q_2, q_0)] \wedge [F(q_1) \vee f_s^*(q_2, q_1) \wedge [F(q_2) \vee f_s^*(q_2, q_2)] \\ &= 0.9 \wedge f_s^*(q_2, q_0)] \wedge [0.4 \vee f_s^*(q_2, q_1)\,] \end{aligned}$$

Therefore for any string $s \in S$, $(f_s^* \circ F)(q_2) = 0.9 \wedge f_s^*(q_0, q_2) \wedge [0.4 \vee f_s^*(q_1, q_2)\,]$ (13)

$$\begin{aligned} L(0) = L(M)(0) &= I \circ f_0^* \circ F \\ &= (f_0^* \circ F)(q_1) \wedge (f_0^* \circ F)(q_2) \\ &= \{0.4 \wedge f_0^*(q_1, q_0) \wedge [0.9 \vee f_0^*(q_1, q_2)]\, \} \wedge \{\, 0.9 \wedge f_0^*(q_2, q_0) \wedge \\ &\quad [0.4 \vee f_0^*(q_2, q_1)]\} = 0.3 \end{aligned}$$

Similarly, $L(1) = 0.4$

$f_{00}^*(q_0, q_0) = f^*(q_0, 00, q_0)$

$= [f(q_0, 0, q_0) \vee f(q_0, 0, q_0)] \wedge [f(q_0, 0, q_1) \vee f(q_1, 0, q_0)] \wedge [f(q_0, 0, q_2) \vee f(q_2, 0, q_0)] = 0$

$f_{00}^* (q_0, q_1) = f^* (q_0, 00, q_0)$
$= [f (q_0, 0, q_0) \vee f (q_0,0,q_1)] \wedge [f (q_0,0,q_1) \vee f (q_1,0,q_1)] \wedge$
$[f (q_0,0,q_2) \vee f (q_2,0,q_1)] = 0.6$

Similarly, the f_{00}^{*} - matrix is computed as follows;

$f_{00}^* (q_0, q_0) = 0$	$f_{00}^* (q_0, q_1) = 0.6$	$f_{00}^* (q_0, q_2) = 0.6$
$f_{00}^* (q_1, q_0) = 0.5$	$f_{00}^* (q_1, q_1) = 0$	$f_{00}^* (q_1, q_2) = 0.6$
$f_{00}^* (q_2, q_0) = 0.3$	$f_{00}^* (q_2, q_1) = 0.6$	$f_{00}^* (q_2, q_2) = 0$

$L(00) = (f_{00}^* \text{ o } F) (q_1) \wedge (f_{00}^* \text{ o } F) (q_2)$
$= \{0.4 \wedge f_{00}^*(q_1,q_0) \wedge [0.9 \vee f_{00}^*(q_1,q_2)] \} \wedge \{ 0.9 \wedge f_{00}^*(q_2,q_0) \wedge [0.4 \vee f_{00}^*(q_2,q_1)]\}$
$= \{0.4 \wedge 0.5 \wedge [0.9 \vee 0.6] \} \wedge \{ 0.9 \wedge 0.3 \wedge [0.4 \vee 0.6] \} = 0.3$

Using the program for the example fuzzy automata, f^*_s – matrix and L(s) values are computed for various strings and the same values are checked using manual calculations. Both manually calculated values and computer results are tallied. Some of the L(s) values are as follows.

L(0) = 0.3, L(1)=0.4, L(00)=L(01)=L(10)=0.3, L(11) = 0.4, L(000)=L(001)= . . . L(110) = 0.3, L(111) = 0.4,
L(0000)=...L(1110)=0.3,L(1111)=0.4,
L(00000000)=0.3, L(00001111)=0.3, L(11110000)=0.3, L(01011100)=0.3,
L(11010110101)=0.3, L(0010101000011)=0.3,
L(1101010100100110101 0001)=0.3, L(111111111111111)=0.4,
L(1111111111111111)=0.4.

It is found that L(s)=0.4 only when every symbol in s is 1. Otherwise, L(s)=0.3. The possible values of δ_i (after arranging them in nondecreasing order) are 0.3, 0.4.

Suppose $0 < \alpha \leq 0.3$.
Let $D_\alpha(M) = M_\alpha$ denote the nondeterministic automaton corresponding to α.
Then $I_\alpha = \{ q_0 \}$, $F_\alpha = \{ q1, q2 \}$, $d_\alpha(q_0 , s) = \{ p \in Q / f_0^*(q_0, s) \geq 0.3 \} = \{q_1, q_2\}$
$L (D_\alpha (M)) = \{s \in S / \text{there exists } q \in I_\alpha \text{ such that } (d_\alpha (q, s) \cap F_\alpha) \neq \phi\}$
$= \{s \in S / \text{there exists } q \in I_{0.3} \text{ such that } (d_{0.3} (q, s) \cap F_{0.3}) \neq \phi\}$
$= \{0, 1\}^+$
$L_\alpha = \{s \in S / L(s) \geq \alpha\}$
$= \{s \in S / L(s) \geq 0.3\}$
$= \{0,1\}^+$
$L (D_\alpha (M)) = L_\alpha.$
Furthermore, $[0]_\alpha$ = {0, 01, 10, 000, 001, 010, 110, 0000, 1110, 00000,…… 11110. ….}
$[1]_\alpha$ = { 1, 11, 111, 1111, ……..}
$L_\alpha = \cup [s]_\alpha = [0]_\alpha \cup [1]_\alpha$.
Suppose $0.3 < \alpha \leq 0.4$.
Let $D_\alpha(M) = M_\alpha$ denote the nondeterministic automaton corresponding to α.
Then $I_\alpha = \{ q_0 \}$, $F_\alpha = \{ q1, q2 \}$, $d_\alpha(q_0 , s) = \{ p \in Q / f_0^*(q_0, s) \geq 0.4 \} = \{q1, q2\}$
$L (D_\alpha (M)) = \{s \in S / \text{there exists } q \in I_\alpha \text{ such that } (d_\alpha (q, s) \cap F_\alpha) \neq \phi\}$
$= \{s \in S / \text{there exists } q \in I_{0.4} \text{ such that } (d_{0.4} (q, s) \cap F_{0.4}) \neq \phi\} = \{0, 1\}^+$
$L_\alpha = \{s \in S / L(s) \geq \alpha\}$

$= \{s \in S / L(s) \geq 0.4\} = \{1\}^+$

$L(D_\alpha(M)) \neq L_\alpha$ and also $L_\alpha \subseteq L(D_\alpha(M))$.

Furthermore, $[1]_{\alpha =} \{1, 11, 111, 1111 \ldots\}$

$$L_\alpha = \cup [s]_\alpha = [1]_\alpha .$$

If $\alpha > 0.4$, then there exists no corresponding nondeterministic automaton and $L(D_\alpha(M)) = L_\alpha = \phi$.

When $\alpha = 0.3$

$\alpha_L(s) = \alpha$ if $L(s) \geq \alpha$, 0 otherwise.

$\alpha_L(0) = \alpha_L(00) = \alpha_L(01) = \alpha_L(10) = \alpha_L(000) = \ldots \alpha_L(110) = \alpha_L(0000) = \ldots \alpha_L(1110) = \ldots.. = 0.3$

$\alpha_L(1) = \alpha_L(11) = \alpha_L(111) = \alpha_L(1111) = \ldots \alpha_L(11111\ldots111) = 0$

When $\alpha = 0.4$

$\alpha_L(s) = \alpha$ if $L(s) \geq \alpha$, 0 otherwise.

$\alpha_L(0) = \alpha_L(00) = \alpha_L(01) = \alpha_L(10) = \alpha_L(000) = \ldots \alpha_L(110) = \alpha_L(0000) = \ldots \alpha_L(1110) = 0$

$\alpha_L(1) = \alpha_L(11) = \alpha_L(111) = \alpha_L(1111) = \ldots \alpha_L(11111\ldots111) = 0.4$

$L = \bigcup_{\alpha \in [0,1]} \alpha_L$ where $\cup$ denotes fuzzy union.

$(\cup \alpha_L)(0) = \vee \alpha_L(0) = 0.3 \vee 0 = 0.3 = L(0)$

$(\cup \alpha_L)(1) = \vee \alpha_L(1) = 0 \vee 0.4 = 0.4 = L(1)$

Similarly, $(\cup \alpha_L)(s) = \vee \alpha_L(s) = 0.3 \vee 0 = 0.3 = L(s)$ for all $s \in S$.

This verifies $L = \cup \alpha_L$

5. Results and Conclusions

In this paper, Myhill Nerode theorem of finite automaton has been extended to fuzzy automaton where the composition considered is min-max composition. The algorithm to compute $f^*(q_i, s, q_j)$ and $L(s)$ is developed and implemented in C++. The program is tested with different fuzzy automata and strings of different lengths. In min-max composition, it is found that L_α need not even be contained in $L(D_\alpha(M))$. Anyway, we have been able to prove the analogue of Myhill Nerode Theorem for fuzzy automata even for min-max composition.

REFERENCES

1. Jiri Mockr, Fuzzy and non-deterministic automata, Research Report No. 8, University of Ostrava, Czech Republic, 1997.
2. V. Ramaswamy and H.A. Girijamma, An extension of Myhill Nerode theorem for fuzzy automata, *Advances in Fuzzy Mathematics*, 4(1) (2009) 41- 47
3. V. Ramaswamy and H.A. Girijamma, Characterization of fuzzy regular languages, *Intern. J. Computer Science and Network Security*, 8(12) (2008) 306 -308.
4. W.Cheng and Z.-W.Mo, Minimization algorithm of fuzzy finite automata, *Fuzzy Sets and Systems*, 141 (2004) 439-448.
5. J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley Publishing, Reading, MA,1979.
6. Vivek Raich, Archana Gawande and Seema Modi, Fuzzy matrix solution for the study of teacher's evaluation, *Intern. J. Fuzzy Mathematical Archive*, 3 (2013) 9-15.
7. J.Boobalan and S.Sriram, The semi inverse of max-min product of fuzzy matrices, *Intern. J. Fuzzy Mathematical Archive*, 3 (2013) 23-27.