NEW DIRECTION IN FUZZY TREE AUTOMATA

S. MOGHARI, M. M. ZAHEDI AND R. AMERI

Abstract. In this paper, our focus of attention is the proper propagation of fuzzy degrees in determinization of *Nondeterministic Fuzzy Finite Tree Automata* (NFFTA). Initially, two determinization methods are introduced which have some limitations (one in behavior preserving and other in type of fuzzy operations). In order to eliminate these limitations and increasing the efficiency of FFTA, we define the notion of fuzzy complex state and *Complex FFTA* (CFFTA). Also, we define ∇-normalization operation in algebra of fuzzy complex state to solve the multi membership state problem in fuzzy automata. Furthermore, we discuss the relationship between FFTA and CFFTA. Finally, determinization of CFFTA is presented.

1. Introduction

Automata are the prime examples of general computational systems over discrete spaces [8] and have a long history both in theory and application [1, 12, 15]. Since Zadeh [18] created the theory of fuzzy sets, many applications of this theory had been grown in other areas. In 1967, Wee [16] introduced the concept of fuzzy automata and in 1995; Mateescu et al. [10] defined finite automata with fuzzy transitions and finite automata with fuzzy states. Fuzzy automata are devices which accept fuzzy languages and are able to create capabilities which are not easily achievable by other mathematical tools [17].

Finite tree automata was introduced by Doner [4, 5] and Thatcher and Wright [13, 14]. Their goal was to prove the decidability of the weak second order theory of multiple successors. A FTA accepts a set of trees called tree language. A FFTA [2, 7, 11] accepts trees with a fuzzy membership degree which is known as behavior of automaton. Processing trees by NFFTA has a potential to makes multiple derivations for the same tree with different membership degrees [6, 9]. This property is undesirable if we wish to know the total membership degree of a particular tree in a fuzzy tree language. Determinization of a FFTA can be thought of as a two-stage process. First, the structure of the automata must be determined such that a single run map exists for each recognized input tree. This is achieved by a classic power set construction, i.e., a state must be constructed in the output automaton that represents all the possible reachable destination states given an input and a label. Because we are working with tree automata, our input is a vector of states, not a single state. A comparable power set construction on non-fuzzy FTA and a proof

Received: July 2010; Revised: October 2010 and November 2010; Accepted: January 2011
Key words and phrases: Fuzzy tree automata, Complex fuzzy tree automata, Determinization.

of correctness can be found in Comon, et. al [3]. The second consideration to fuzzy determinization is proper propagation of fuzzy degrees.

Initially, we introduce two determinization methods for NFFTA. The first method solves propagation of fuzzy degrees by operating on degree of all possible derivation chains for each class of trees. Since the class of trees for a regular tree language is finite index [11], the realized automata will be finite. This method is language preserving but it is not behavior preserving. The second method constructs power set on the pair of states and all potential membership degrees and then, realizes a *Deterministic FFTA* (DFFTA) with non-fuzzy transition rules and fuzzy final states. The second method is behavior preserving (and consequently is language preserving) but the fuzzy operators in automata are limited to finite-range operators; because the automaton is finite and then, the set of potential membership degrees must be finite. Another weakness of this method is that the numbers of states of DFFTA are exponentially more than the first method.

The problem of limitations in determinization of a NFFTA can be solved by extending the definition of state in automata. We define the notion of complex symbols and use it in the definition of state in CFFTA and show that the new definition is generalization of FFTA. Another consideration in the NFFTA is the multiple membership states [6, 9]. We solve this problem by defining ∇-*normalization* operation and converting NFFTA to an equivalent *Deterministic CFFTA* (DCFFTA).

The reminder of this paper is organized as follows. Section (2) presents the preliminary notes and in section (3), our determinization methods are discussed. The notion of CFFTA and related results are explained in section (4). Finally, we reach our conclusion in section (5).

2. Preliminaries

The set of natural numbers is denoted by $\mathbb{N}$, and the set of finite strings over $\mathbb{N}$ is $\mathbb{N}^*$. The empty string is denoted by ε. A Σ-*alphabet* is a finite and nonempty set of symbols. A *ranked alphabet* is a couple $(\Sigma, Arity : \Sigma \to \mathbb{N})$, which is the disjoint union of sets of *n-ary* symbols $\Sigma_n = \{\sigma | Arity(\sigma) = n\}$ for all $n \geq 0$. The set $T_\Sigma(Q)$ of Σ-*trees* indexed by Q is inductively defined to be the smallest set such that $Q \subseteq T_\Sigma(Q)$ and $\sigma(t_1, \cdots, t_n) \subseteq T_\Sigma(Q)$ for every $\sigma \in \Sigma_n$ and $t_1, \cdots, t_n \in T_\Sigma(Q)$. We write T_Σ for $T_\Sigma(\phi)$. A Σ-tree can also be defined as a partial function $\mathbb{N} \to \Sigma \cup Q$ with domain $Pos(t)$ satisfying the following properties:

- $Pos(t)$ is nonempty and prefix-closed,
- If $t = \sigma \in \Sigma_0$, then $Pos(t) = \{\varepsilon\}$,
- If $t = \sigma(t_1, \cdots, t_n)$ where $\sigma \in \Sigma_n$and $t_1, \cdots, t_n \in T_\Sigma$, then $Pos(t) = \{\varepsilon\} \cup \bigcup_{i=1}^{n} \{ip | p \in Pos(t_i)\}$.

Definition 2.1. [3, 11] A *tree automaton* is a system $A = (Q, \Sigma, \delta, \Gamma)$, where

(1) Q is a set of *state symbols*,
(2) Σ is a finite set of ranked alphabets called *input symbols*,
(3) $\delta = \{\delta_\sigma : Q_n \times Q \times \Sigma_n \to \{0, 1\} | \sigma \in \Sigma_n, n \geq 0\}$ is a set of *transition rules*,
(4) $\Gamma \subseteq Q$ is a set of *final states*,
As well, a final state $\gamma \in \Gamma$ may be defined by $\gamma : Q \to \{0, 1\}$.

A tree automaton $A = (Q, \Sigma, \delta, \Gamma)$ called finite if Q is finite and if $\delta(q_1, \cdots, q_n, q, \sigma)$ is not defined for all but a finite number of $\delta_\sigma \in \delta$. Suppose that $t \in T_\Sigma$. A run map $\rho(t) \subseteq Q$ is defined by induction on the structure of t:

- When $t = \sigma \in \Sigma_0$, then $\rho(t) = \delta(\sigma)$,
- Assume that $t = \sigma(t_1, \cdots, t_n)$ for some $\sigma \in \Sigma_n$ and $t_1, \cdots, t_n \in T_\Sigma$, then $\rho(t) = \delta(\rho(t_1), \cdots, \rho(t_n))$.

A tree t is *Accepted* by FTA $A = (Q, \Sigma, \delta, \Gamma)$ if $\rho(t) \cap \Gamma \neq \phi$. The tree language *Recognized* by FTA A is the set of all trees accepted by A.

Definition 2.2. [3, 11] A *Deterministic FTA* is a FTA $A_d = (Q_d, \Sigma, \delta_d, \Gamma_d)$ such that for each $\sigma \in \Sigma_n$ and $q_1, \cdots, q_n \in Q_d$ there exists at most one $q \in Q_d$ such that $\delta_d(q_1, \cdots, q_n, q, \sigma) = 1$.

Theorem 2.3. *For each FTA $A = (Q, \Sigma, \delta, \Gamma)$, there exists an equivalent DFTA $A_d = (Q_d, \Sigma, \delta_d, \Gamma_d)$ such that $L(A) = L(A_d)$.*

Proof. See [3]. □

Definition 2.4. [18] Let X be a nonempty set. A fuzzy set A in X is characterized by its membership function $\mu_A : X \to [0, 1]$, and $\mu_A(x)$ is interpreted as the degree of membership of element x in fuzzy set A for each $x \in X$.

Definition 2.5. [2, 7, 11] A *Fuzzy Finite Tree Automaton* over (∇, Δ) is a system $A = (Q, \Sigma, \delta, \Gamma)$, where:

(1) Q is a set of *state symbols*,
(2) Σ is a finite set of ranked alphabets called *input symbols*,
(3) $\delta = \{\delta_\sigma : Q_n \times Q \times \Sigma_n \to [0, 1] | \sigma \in \Sigma_n, n \geq 0\}$ is a set of *transition rules*,
As well, a transition rule may be defined by $\delta_\sigma : Q_n \times \Sigma_n \times [0, 1] \to Q$.
(4) $\Gamma \subseteq Fuzzy(Q)$ is a set of *final states*,
As well, a final state $\gamma \in \Gamma$ may be defined by $\gamma : Q \to [0, 1]$.

A fuzzy tree automaton A called finite if Q is finite and if $\delta_\sigma(q_1, \ldots, q_n, q, \sigma) = 0$ for all but a finite number of $\delta_\sigma \in \delta$, and if $\Gamma(q) = 0$ for all but a finite number of states $q \in Q$.

Suppose that $t \in T_\Sigma$. A fuzzy set $\rho(t) \subseteq Fuzzy(Q)$ is defined by induction on structure of t:

- When $t = \sigma \in \Sigma_0$, then $\rho(t)(q) = \delta(q, \sigma)$ for all $q \in Q$,
- Assume that $t = \sigma(t_1, \ldots, t_n)$ for some $\sigma \in \Sigma_n$ and $t_1, \ldots, t_n \in T_\Sigma$, then

$$\rho(t)(q) = \mathop{\nabla}_{q_1, \ldots, q_n \in Q} \Big(\delta(q_1, \ldots, q_n, q, \sigma) \Delta \mathop{\Delta}_{i=1}^{n} \rho(t_i)(q_i)\Big).$$

The behavior of FFTA A is a fuzzy set $|A|$ on a set of trees $t \in T_\Sigma$ defined by:

$$|A|(t) = \mathop{\nabla}_{q \in Q} \Big(\rho(t)(q) \Delta \Gamma(q)\Big).$$

In other words, for each tree $t \in T_\Sigma$ and FFTA $A = (Q, \Sigma, \delta, \Gamma)$, $|A|(t) = \mu_{L(A)}(t)$ and t is *Accepted* by A if and only if $|A|(t) > 0$.

3. Determinization of Fuzzy Finite Tree Automata

Definition 3.1. A *Deterministic FFTA* is a FFTA $A_d = (Q_d, \Sigma, \delta_d, \Gamma_d)$ such that for each $q_1, \ldots, q_n \in Q_d$ and $\sigma \in \Sigma_n$, there exists at most one $q \in Q_d$ such that $\delta_d(q_1, \ldots, q_n, q, \sigma) > 0$.

Theorem 3.2. *For each FFTA $A = (Q, \Sigma, \delta, \Gamma)$, one can construct a DFFTA $A_d = (Q_d, \Sigma, \delta_d, \Gamma_d)$ such that $L(A) = L(A_d)$.*

Proof. For all $t \in T_\Sigma$, set $Q_t = \{q \in Q|\rho(t)(q) > 0\}$. Let $Q_d = \{Q_t|t \in T_\Sigma\}$. Since $Q_d \subseteq 2^Q$, thus Q_d is finite. Define $\Gamma_d : Q_d \to [0,1]$ by $\Gamma_d(Q_t) = \nabla\{\Gamma(q)|q \in Q\}$. Finally, define $\delta_d : (Q_d)^n \times Q_d \times T_\Sigma \to [0,1]$ by $\delta_d(Q_{t_1}, ..., Q_{t_n}, Q_t, \sigma) = \mu$ where $t = \sigma(t_1, ..., t_n)$, $\sigma \in \Sigma_n$ and $\mu = \nabla\{\delta(q_1, ..., q_n, q, \delta)|q_1 \in Q_{t_1}, ..., q_n \in Q_{t_n}, q \in Q_t\}$. Now, $t \in L(A)$ if and only if $\rho(t)(q)\Delta\Gamma(q) > 0$ for some $q \in Q$ if and only if $\Gamma(q) > 0$ for some $q \in Q_t$ if and only if $\Gamma_d(Q_t) > 0$. It suffice to show that $\rho_d(t)(Q_t)\Delta\Gamma(Q_t) > 0$ for $t \in L(A)$ then $t \in L(A_d)$ if and only if $\Gamma(Q_t) > 0$ (since $\rho_d(t)(Q_t) > 0$) if and only if $t \in L(A)$. We show that $\rho_d(t)(Q_t) > 0$ by induction on the structure of t . Suppose $t = \sigma \in \Sigma_0$. Then $\delta_d(Q_t, \sigma) = \nabla\{\delta(q, \sigma)|q \in Q_t\} > 0$. Suppose $t = \sigma(t_1, ..., t_n)$, where $t = \sigma \in \Sigma_n$ and the result is true for $t_1, ..., t_n$. Then $\delta_d(Q_{t_1}, ..., Q_{t_n}, \sigma) = \nabla\{\delta(q_1, ..., q_n, q, \sigma)|q_1 \in Q_{t_1}, ..., q_n \in Q_{t_n}, q \in Q_t\} > 0$ by induction hypothesis. Hence, $\rho_d(t)(Q_t) = \delta_d(Q_{t_1}, ..., Q_{t_n}, Q_t, \sigma)$. □

The method introduced in Theorem 3.2 is language preserving but, is not behavior preserving. The next example shows this fact.

Example 3.3. Let $A = (Q, \Sigma, \delta, \Gamma)$ be a FFTA over $(\vee, \wedge)$. Let $Q = \{q_1, q_2\}$, $\Sigma = \Sigma_0 = \{\sigma\}$, $\delta(q_1, \sigma) = 0 \cdot 2$, $\delta(q_2, \sigma) = 0 \cdot 7$, $\Gamma(q_1) = 0 \cdot 5$ and $\Gamma(q_2) = 0 \cdot 3$. Then, for $t = \sigma$ we have $|A|(t) = (0 \cdot 2 \wedge 0 \cdot 5) \vee (0 \cdot 7 \wedge 0 \cdot 3) = 0 \cdot 3$.

Applying method introduced in Theorem 3.2, we have $A_d = (Q_d, \Sigma, \delta_d, \Gamma_d)$ with $Q_d = \{\{q_1, q_2\}\}$, $\delta(\{q_1, q_2\}, \sigma) = 0 \cdot 7$ and $\Gamma(\{q_1, q_2\}) = 0 \cdot 5$. Now, we have $|A_d|(t) = 0 \cdot 7 \wedge 0 \cdot 5 = 0 \cdot 5$.

Definition 3.4. A pair of operators (∇, Δ) with domain D is called *Finite Range* if there exist a finite set R such that any combination of ∇ and $\triangle$ with domain D, be in R.

Theorem 3.5. *For each FFTA $A = (Q, \Sigma, \delta, \Gamma)$ over finite range operators (∇, Δ), one can construct an equivalent DFFTA $A_d = (Q_d, \Sigma, \delta_d, \Gamma_d)$ such that for each tree $t \in T_\Sigma$ we have $|A|(t) = |A_d|(t)$.*

Proof. For all $t \in T_\Sigma$, set $Q_t = \{(q, \rho(t)(q))|q \in Q\}$. Let $Q_d = \{Q_t|t \in T_\sigma\}$. Since Q is finite, (∇, Δ) are finite range and $Q_d \subseteq 2^{Q \times Range(\nabla, \Delta)}$, thus Q_d is finite. Define $\Gamma_d : Q_d \to [0,1]$ by $\Gamma_d(Q_t) = \nabla\{\mu\Delta\Gamma(q)|(q, \mu) \in Q_t\}$. Finally, let $\delta_d = \{\delta_\sigma(Q_{t_1}, ..., Q_{t_n}, Q_t, \sigma) = 1|\delta_\sigma(q_1, ..., q_n, q, \sigma) \in \delta, q_1 \in Q_{t_1}, ..., q_n \in Q_{t_n}, q \in Q_t\}$. It is easy to prove that for each tree $t \in T_\Sigma$ we have $|A|(t) = |A_d|(t)$. □

1 Input: FFTA $A = (Q, \Sigma, \delta, \Gamma)$
2
3 Begin
4 Set $Q_d = \phi$
5 Repeat
6 Set $t = f(t_1, \dots, t_n)$; *where* $f \in \Sigma_n$ *and* $Q_{t_1}, \dots, Q_{t_n} \in Q_d$
7 Set $Q_t = \{(q, c) | c = \underset{\delta(q_1,\dots,q_n,q,f)=c'}{\bigtriangledown} \{c' \Delta \overset{n}{\underset{i=1}{\Delta}} c_i | (q_1, c_1) \in Q_{t_1}, \dots, (q_n, c_n) \in Q_{t_n}\}\}$
8 Set $Q_d = Q_d \cup Q_t$
9 Until *no state can be added to* Q_d
10 End
11
12 Output: Q_d

FIGURE 1. Computing Deterministic State Set for a FFTA.

A dynamic algorithm for computing the set Q_d which is defined in the proof of Theorem 3.5, is introduced. This algorithm is shown in Figure 1; lines 6 to 8 compute elements of Q_d and firstly start with Σ_0. Then, these instructions repeat until end up with the set Q_d. Since the set Q_d is finite and in each repetition at least one state adds to it, the repeat-until loop is finite.

Example 3.6. Let $A = (Q, \Sigma, \delta, \Gamma)$ be a FFTA over finite range operators $(\vee, \wedge)$. Now, Let $Q = \{q_1, q_2, q_3\}$, $\Sigma_0 = \{\alpha, \beta\}$, $\Sigma_2 = \{\sigma\}$, $\delta(q_1, \alpha) = 0 \cdot 5$, $\delta(q_2, \alpha) = 0 \cdot 7$, $\delta(q_1, \beta) = 0 \cdot 8$, $\delta(q_1, q_1, q_2, \sigma) = 0 \cdot 6$, $\delta(q_1, q_1, q_3, \sigma) = 0 \cdot 4$, $\delta(q_2, q_1, q_2, \sigma) = 0 \cdot 2$, $\delta(q_3, q_3, q_3, \sigma) = 0 \cdot 4$, $\Gamma(q_1) = 0 \cdot 9$, $\Gamma(q_2) = 0 \cdot 3$ and $\Gamma(q_3) = 0 \cdot 1$. By method introduced in Theorem 3.5, $A_d = (Q_d, \Sigma, \delta_d, \Gamma_d)$ is realized by

$$\begin{array}{llll}
t_1 = \alpha & Q_{[t_1]} = \{(q_1, 0 \cdot 5), (q_2, 0 \cdot 7)\} & t_2 = \beta & Q_{[t_2]} = \{(q_1, 0 \cdot 8)\} \\
t_3 = (t_1, t_1) & Q_{[t_3]} = \{(q_2, 0 \cdot 5), (q_3, 0 \cdot 4)\} & t_4 = (t_3, t_1) & Q_{[t_4]} = \{(q_2, 0 \cdot 2)\} \\
t_5 = (t_2, t_2) & Q_{[t_5]} = \{(q_2, 0 \cdot 6), (q_3, 0 \cdot 4)\} & t_6 = (t_3, t_3) & Q_{[t_6]} = \{(q_3, 0 \cdot 4)\}
\end{array}$$

Thus $Q_d = \{Q_{[t_1]}, Q_{[t_2]}, Q_{[t_3]}, Q_{[t_4]}, Q_{[t_5]}, Q_{[t_6]}\}$, $\Gamma(Q_{t_1}) = 0 \cdot 5$, $\Gamma(Q_{t_2}) = 0 \cdot 8$, $\Gamma(Q_{t_3}) = 0 \cdot 3$, $\Gamma(Q_{t_4}) = 0 \cdot 3$, $\Gamma(Q_{t_5}) = 0 \cdot 2$ and $\Gamma(Q_{t_6}) = 0 \cdot 1$. Furthermore, δ_d is the following set of transition rules:

$$\begin{array}{lll}
\{\delta_d(Q_{[t_1]}, \alpha), & \delta_d(Q_{[t_2]}, \beta), & \delta_d(Q_{[t_1]}, Q_{[t_1]}, Q_{[t_3]}, \sigma), \\
\delta_d(Q_{[t_1]}, Q_{[t_2]}, Q_{[t_3]}, \sigma), & \delta_d(Q_{[t_2]}, Q_{[t_1]}, Q_{[t_3]}, \sigma), & \delta_d(Q_{[t_2]}, Q_{[t_2]}, Q_{[t_4]}, \sigma), \\
\delta_d(Q_{[t_3]}, Q_{[t_1]}, Q_{[t_5]}, \sigma), & \delta_d(Q_{[t_3]}, Q_{[t_2]}, Q_{[t_5]}, \sigma), & \delta_d(Q_{[t_3]}, Q_{[t_3]}, Q_{[t_6]}, \sigma), \\
\delta_d(Q_{[t_3]}, Q_{[t_6]}, Q_{[t_6]}, \sigma), & \delta_d(Q_{[t_4]}, Q_{[t_3]}, Q_{[t_6]}, \sigma), & \delta_d(Q_{[t_4]}, Q_{[t_4]}, Q_{[t_6]}, \sigma), \\
\delta_d(Q_{[t_4]}, Q_{[t_6]}, Q_{[t_6]}, \sigma), & \delta_d(Q_{[t_5]}, Q_{[t_1]}, Q_{[t_5]}, \sigma), & \delta_d(Q_{[t_5]}, Q_{[t_2]}, Q_{[t_5]}, \sigma), \\
\delta_d(Q_{[t_6]}, Q_{[t_3]}, Q_{[t_6]}, \sigma), & \delta_d(Q_{[t_6]}, Q_{[t_4]}, Q_{[t_6]}, \sigma), & \delta_d(Q_{[t_6]}, Q_{[t_6]}, Q_{[t_6]}, \sigma)\}.
\end{array}$$

Suppose that $t = \sigma(\alpha, \beta)$. By FFTA A we have

$t = \sigma(\alpha, \beta) \overset{0 \cdot 5}{\Rightarrow} t = \sigma(q_1, \beta) \overset{0 \cdot 8}{\Rightarrow} t = \sigma(q_1, q_1) \overset{0 \cdot 6}{\Rightarrow} t = q_2$,
$t = \sigma(\alpha, \beta) \overset{0 \cdot 5}{\Rightarrow} t = \sigma(q_1, \beta) \overset{0 \cdot 8}{\Rightarrow} t = \sigma(q_1, q_1) \overset{0 \cdot 4}{\Rightarrow} t = q_3$,
$t = \sigma(\alpha, \beta) \overset{0 \cdot 7}{\Rightarrow} t = \sigma(q_2, \beta) \overset{0 \cdot 8}{\Rightarrow} t = \sigma(q_2, q_1) \overset{0 \cdot 2}{\Rightarrow} t = q_2$.

Thus $\rho(t)(q_2) = 0{\cdot}5$, $\rho(t)(q_3) = 0{\cdot}4$ and then, $|A|(t) = (0{\cdot}5 \wedge 0{\cdot}3) \vee (0{\cdot}4 \wedge 0{\cdot}1) = 0{\cdot}3$. Now, by DFFTA A_d; $t = \sigma(\alpha, \beta) \Rightarrow t = \sigma(Q_{[t_1]}, \beta) \Rightarrow t = \sigma(Q_{[t_1]}, Q_{[t_2]} \Rightarrow t = Q_{[t_3]}$. Hence, $|A|(t) = 1 \wedge 0 \cdot 3 = 0 \cdot 3$.

Note that $2^{Q \times Range(\Delta, \nabla)}$ is not an appropriate bound for Q_d. For example, considering a FFTA on $(\vee, \wedge)$ with two states and five transition rules; its equivalent DFFTA may have 10^3 states and 10^6 transition rules.

4. **Complex Fuzzy Finite Tree Automata**

In order to increase the efficiency in determinization process, we generalize the concept of state in automata and call it general state. Also we define the algebra of fuzzy complex state to solve some problems in FFTA (such as above limitations in determinization and multi membership states [6]) by using this algebra in definition of complex fuzzy finite tree automata and its determinization method.

Definition 4.1. A *Complex Symbol* is defined by

- A symbol is a complex symbol.
- A k-tuple $\langle c_1, ...c_k \rangle$ of symbols $c_1, ...c_k$ is a complex symbol, where $k > 1$. In other words, the only single-component complex symbols are symbols.

The set of components of a complex symbol $c = \langle c_1, ...c_k \rangle$ is denoted by $Cp(c) = \{c_1, ..., c_k\}$ and the *Rank* of c is defined by $R(c) = |Cp(c)|$.

Convention 4.2. When c is a symbol, then $Cp(c) = \{c\}$.

Convention 4.3. The set of components of a complex symbol set S is denoted by $Cp(S) = \bigcup_{c \in S} Cp(c)$.

Definition 4.4. An ordering relation on complex symbol set S is defined as follows:

- If c and c' are symbols, then $c < c'$ means that c, c' is in the lexicographic ordering of elements of set S.
- Otherwise, $c < c'$ means that for $c' = \langle c'_1, ...c'_k \rangle$ and $c = \langle c_1, ...c_k \rangle$ there exist $i \in \{1, ..., min(k, k')\}$ such that $c_i < c'_i$ and $c_j = c'_j$ for $j \in \{1, ..., i-1\}$, and if such i is not exist, then $k < k'$.

A complex symbol $c = \langle c_1, ..., c_k \rangle$ is *Normalized* if $i < j$ implies that $c_i < c_j$ for $i, j \in \{1, ..., k\}$. A normalized complex symbol c denote by $c = \langle Cp(c) \rangle$.

Definition 4.5. A *Uset* $\breve{S}$ on the set S of complex symbols is defined as follows: $S \subseteq \breve{S}$, and if $c = \langle c_1, ..., c_k \rangle$, $k > 1$ and $c \in \breve{S} \setminus S$, then $\exists c'_1, ..., c'_{k-1} \in S$ such that $c_1 \in Cp(c'_1), ..., c_{k-1} \in Cp(c'_{k-1})$ and $c_k \in S$.

Definition 4.6. A *Fuzzy Complex Symbol* is denoted by $\tilde{c} = \langle (c_1, \mu_1), ..., (c_k, \mu_k) \rangle$, where $\mu_1, ..., \mu_k \in [0, 1]$, and two k-tuples $c = \langle c_1, ..., c_k \rangle$ and $\mu = \langle \mu_1, ..., \mu_k \rangle$ are called its corresponding complex symbol and corresponding membership degree, respectively.

Denote by $\tilde{S}$ the set of fuzzy complex symbols, while S points to the set of corresponding complex symbols and μ^S points to the set of corresponding membership degrees. Ordering relation on a fuzzy complex symbol set is the same as crisp one, and a fuzzy complex symbol is normalized if its corresponding complex symbol is normalized.

Definition 4.7. Let $\tilde{c} = \langle (c_1, \mu_1), ..., (c_k, \mu_k) \rangle$ be a non normalized fuzzy complex symbol and ∇ be an operator or a function with domain $\{\mu_1, ..., \mu_k\}$. Fuzzy complex symbol $\tilde{c'} = \langle (c'_1, \mu'_1), ..., (c'_{k'}, \mu'_{k'}) \rangle$ denoted by $\tilde{c'} = |\tilde{c}|_\nabla$ is ∇-normalization of $\tilde{c}$ if $\{c'_1, ..., c'_k\} = \{c_1, ..., c_k\}$, and $\mu'_i = \underset{c'_j = c_j}{\nabla} \mu_j$ for $i = 1, ..., k'$ and $j = 1, ..., k$.

Definition 4.8. Let $\mu = \langle \mu_1, ..., \mu_k \rangle$, $\mu' = \langle \mu'_1, ..., \mu'_k \rangle$ where $\mu_i, \mu'_i \in [0, 1]$ for $i = 1, ..., k$, and $\alpha \in [0, 1]$. Then, define operation ∇ for tuples by

- $\nabla \mu = \underset{i \in \{1, ..., k\}}{\nabla} \mu_i$,
- $\mu' \nabla \mu = \mu \nabla \mu' = \langle \mu_1 \nabla \mu'_1, ..., \mu_k \nabla \mu'_k \rangle$
- $\alpha \nabla \mu = \mu \nabla \alpha = \langle \mu_1 \nabla \alpha, ..., \mu_k \nabla \alpha \rangle$

Definition 4.9. A *Complex FFTA* (CFFTA) is a system $A = (Q, \Sigma, \delta, \Gamma)$, where:

(1) Q is a finite set of finite complex symbols called *states*,

(2) Σ is a finite set of ranked alphabets called *input symbols*,

(3) $\delta = \{\delta_\sigma : (\tilde{Q})^n \times \breve{Q} \times \Sigma_n \rightarrow \tilde{Q} | \sigma \in \Sigma_n, |\breve{Q}| = n + 1, n > 0\}$ is a finite set of *transition rules*,
A transition rule may be defined as $\delta_\sigma : (\tilde{Q})^n \times \breve{Q} \times \Sigma_n \times Q \rightarrow \mu^Q$ or as $\delta_\sigma : (\tilde{Q})^n \times Fuzzy(\breve{Q}) \times \Sigma_n \rightarrow Q$.

(4) $\Gamma \subseteq \tilde{Q}$ is a finite set of *final states*,
A final state may be defined as a function $\gamma : Q \rightarrow \mu^Q$, where $\gamma \in \Gamma$.

(5) $\rho : (\nabla, \Delta) \times T_\Sigma(\tilde{Q}) \rightarrow \tilde{Q}$ is a finite set of *partial run maps* over a pair of operators (∇, Δ),

Suppose that $t \in T_\Sigma$. The partial run map ρ over (∇, Δ) is defined by induction on the structure of t

– When $t = \sigma \in \Sigma_0$, then $\rho_{\nabla\Delta}(t)(q) = \underset{\delta_\sigma(q,\sigma) \in \delta}{\nabla} \delta_\sigma(q, \sigma)$, where $q \in Q$,

– Assume that $t = \sigma(t_1, ..., t_n)$ for some $\sigma \in \Sigma_n$ and $t_1, ..., t_n \in T_\Sigma(Q)$. Then $\rho_{\nabla\Delta}(t)(q) = \langle (\mu_{\tilde{q}}(\bar{q}_1), \ldots, \mu_{\tilde{q}}(\bar{q}_k)) \rangle$, where $q = \langle \bar{q}_1, ..., \bar{q}_k \rangle$ and

$$\mu_{\tilde{q}}(\bar{q}_i) = \underset{\substack{\tilde{q}_1 \in \rho_{\nabla\Delta}(t_1), \ldots, \tilde{q}_n \in \rho_{\nabla\Delta}(t_n) \\ \breve{q}_\sigma \in \breve{Q}, \delta_\sigma(\tilde{q}_1, \ldots, \tilde{q}_n, \breve{q}_\sigma, q, \sigma)}}{\nabla} \Big(\mu_{\breve{q}_\sigma}(\langle q'_1, \ldots, q'_n, \bar{q}_i \rangle) \Delta \overset{\mathrm{n}}{\underset{\mathrm{j=1}}{\Delta}} \mu_{|\tilde{q}_j|_\nabla}(q'_j) \Big).$$

for $i = 1, \ldots, k$.

(6) $\beta \subseteq Fuzzy(T_\Sigma(Q))$ is called the *behavior* of automaton,
The behavior of CFFTA A is a fuzzy set on a set of trees $t \in T_\Sigma$ defined by:

$$\beta = \nabla \Big(\underset{q \in Q}{\Delta} \big(\rho(t)(q)\Delta\Gamma(q)\big)\Big).$$

Moreover, $\beta(t) = \mu$ means that $\mu_{L(A)}(t) = \mu$, for each $t \in T_\Sigma$.

Definition 4.10. A *Primitive FFTA* (PFFTA) is a CFFTA such that its states are symbols.

Proposition 4.11. *Each FFTA is equivalent to a PFFTA.*

Proof. Let FFTA $A = (Q, \Sigma, \delta, \Gamma)$ and PFFTA $A' = (Q, \Sigma, \delta', \Gamma', \rho', \beta')$. Assume that $\delta_\sigma(q_1, ..., q_n, q, \sigma) \in \delta$ and $\delta'_\sigma(q_1, ..., q_n, \check{q}_\sigma, q, \sigma) \in \delta'$ be equivalent transition rules and $\Gamma'(q) = \Gamma(q)$, where $\sigma \in \Sigma_n$ $q_1, ..., q_n, q \in Q$ and $\check{q}_\sigma = \{\langle q_1, ..., q_n, q\rangle\}$. According to the definition of behavior in CFFTA, it is easy to prove that for all $t \in T_\sigma$ we have $\beta'(t) = |A|(t)$. □

Corollary 4.12. *Each FFTA is equivalent to a CFFTA.*

Proposition 4.13. *For each CFFTA $A = (Q, \Sigma, \delta, \Gamma, \rho, \beta)$ one can construct an equivalent FFTA $A' = (Q', \Sigma, \delta', \Gamma')$ such that for all $t \in T_\Sigma$ we have $|A'|(t) = \beta(t)$.*

Proof. Set $Q' = Cp(Q)$. For each $\delta_\sigma(q_1, ..., q_n, \check{q}_\sigma, q, \sigma) \in \delta$ set

$$Split(\delta_\sigma) = \{\delta'_\sigma(q'_1, ..., q'_n, q', \sigma) = \mu' | \check{q}_\sigma = \{(\langle q'_1, ..., q'_n, q'\rangle, \mu')\}\}.$$

Set $\delta' = \bigcup_{\delta_\sigma \in \delta} Split(\delta_\sigma)$. For each $q' \in Q'$ set $\Gamma'(q') = \nabla \mu_{\tilde{q}}(q') | \tilde{q} \in \Gamma, q' \in Cp(q)\}$. It is easy to prove that for all $t \in T_\Sigma$ we have $\beta'(t) = \beta(t)$. □

Corollary 4.14. *The classes of language for CFFTA, PFFTA and FFTA are equivalent.*

Example 4.15. Let $A = (Q, \Sigma, \delta, \Gamma, \rho, \beta)$ be a CFFTA, where $Q = \{\langle q_1, q_2\rangle, \langle q_1, q_3\rangle, \langle q_2, q_3\rangle\}$, $\Sigma_0 = \{\alpha, \tau\}$, $\Sigma_2 = \{\sigma\}$, $\Gamma(\langle q_1, q_2\rangle) = \langle 0 \cdot 7, 0 \cdot 3\rangle$, $\Gamma(\langle q_1, q_3\rangle) = \langle 0 \cdot 1, 0 \cdot 6\rangle$, $\Gamma(\langle q_2, q_3\rangle) = \langle 0 \cdot 9, 0 \cdot 5\rangle$ and δ is

$\delta(\{0 \cdot 3, 0 \cdot 3\}, \alpha) = \langle q_1, q_2\rangle$
$\delta(\{0 \cdot 5, 0 \cdot 4\}, \alpha) = \langle q_1, q_3\rangle$
$\delta(\{0 \cdot 2, 0 \cdot 6\}, \tau) = \langle q_2, q_3\rangle$
$\delta(\langle q_1, q_2\rangle, \langle q_2, q_3\rangle, \{(\langle q_1, q_2, q_1\rangle, 0.5), (\langle q_1, q_3, q_2, 0.8)\}, \sigma) = \langle q_1, q_2\rangle$
$\delta(\langle q_1, q_2\rangle, \langle q_2, q_3\rangle, \{(\langle q_1, q_2, q_1\rangle, 0.2), (\langle q_2, q_2, q_3, 0.7), (\langle q_2, q_3, q_3\rangle, 0.9)\}, \sigma) = \langle q_1, q_3\rangle$
$\delta(\langle q_1, q_3\rangle, \langle q_1, q_3\rangle, \{(\langle q_1, q_1, q_2\rangle, 0.3), (\langle q_3, q_1, q_2, 0.7), (\langle q_3, q_3, q_3\rangle, 0.7)\}, \sigma) = \langle q_2, q_3\rangle$
$\delta(\langle q_2, q_3\rangle, \langle q_1, q_3\rangle, \{(\langle q_2, q_1, q_1\rangle, 0.4), (\langle q_2, q_3, q_2, 0.6)\}, \sigma) = \langle q_1, q_2\rangle$

The FFTA $A' = (Q', \Sigma, \delta', \Gamma')$ equivalent to CFFTA A is defined by $Q' = \{q_1, q_2, q_3\}$, $\Gamma'(q_1) = 0.7$,$\Gamma'(q_2) = 0.9$,$\Gamma'(q_3) = 0.6$, and δ' is the following set of transition rules:

$\delta(q_1, \alpha) = 0.3 \quad \delta(q_2, \alpha) = 0.3 \quad \delta(q_1, \alpha) = 0.5 \quad \delta(q_3, \alpha) = 0.4$
$\delta(q_2, \tau) = 0.2 \quad \delta(q_3, \tau) = 0.6 \quad \delta(q_1, q_2, q_1, \sigma) = 0.5$
$\delta(q_1, q_3, q_2, \sigma) = 0.8 \quad \delta(q_2, q_2, q_3, \sigma) = 0.7 \quad \delta(q_2, q_3, q_3, \sigma) = 0.9$
$\delta(q_1, q_1, q_2, \sigma) = 0.3 \quad \delta(q_1, q_2, q_1, \sigma) = 0.2 \quad \delta(q_3, q_1, q_2, \sigma) = 0.7$
$\delta(q_3, q_3, q_3, \sigma) = 0.7 \quad \delta(q_2, q_1, q_1, \sigma) = 0.4 \quad \delta(q_2, q_3, q_2, \sigma) = 0.6$

Note that some rules such as $\delta(q_1, \alpha) = 0.3$ and $\delta(q_1, q_2, q_1, \sigma) = 0.2$ may be removed from δ'.

Definition 4.16. A *Deterministic CFFTA* is a CFFTA $A_d = (Q_d, \Sigma, \Gamma_d, \rho_d, \beta_d)$ such that for each $\sigma \in \Sigma_n$ and $q_1, ..., q_n \in Q_d$, there exists at most one $q \in Q_d$ and $\breve{q} \subseteq Fuzzy(\breve{Q})$ such that $\delta_\sigma(q_1, ..., q_n, \breve{q}, q, \sigma) \in \delta_d$.

Theorem 4.17. *For each CFFTA $A_n = (Q_n, \Sigma, \delta_n, \Gamma_n, \rho_n, \beta_n)$, one can construct an equivalent DCFFTA $A_d = (Q_d, \Sigma, \delta_d, \Gamma_d, \rho_d, \beta_d)$, such that for each tree $t \in T_\Sigma$ we have $\beta_d(t) = \beta_n(t)$.*

Proof. We assume without loss of generality that FFTA $A = (Q, \Sigma, \delta, \Gamma)$ is equivalent to A_n. For all $t \in T_\Sigma$, set $Q_t = \langle \{q \in Q | \rho(t)(q) \rangle 0\} \rangle$. Let $Q_d = \{Q_t | t \in T_\Sigma\}$. Define $\Gamma_d : Q_d \to \mu^{Q_d}$ by

$$\Gamma_d(Q_t) = \langle \Gamma(q_1), ..., \Gamma(q_k) \rangle$$

where $k = R(Q_t)$ and $Cp(Q_t) = \{q_1, ..., q_k\}$. Define $\delta_d : (\tilde{Q}_d)^n \times \breve{Q}_t \times T_\Sigma \times Q_d \to \mu^{Q_d}$ by $\delta_d(Q_{t_1}, ..., Q_{t_1}, \breve{Q}_t, Q_t, \sigma) = \langle \rho(t)(q_1), ..., \rho(t)(q_k) \rangle$ where $\sigma \in \Sigma_n$, $t = \sigma(t_1, ...t_n)$, $k = R(Q_t) \geq 1$ $Cp(Q_t) = \{q_1, ..., q_k\}$, $\rho(t)(q_i) > 0$ for $i = 1, ..., k$, and

$\breve{Q}_t = \{(\langle q'_1, ..., q'_n, q' \rangle, \delta(q'_1, ..., q'_n, q', \sigma)) | q'_1 \in Cp(Q_{t_1}), ..., q'_n \in Cp(Q_{t_n}), q' \in Cp(Q_t)\}$.

It is easy to prove that for all $t \in T_\Sigma$ we have $\beta(t) = \beta_d(t)$. □

5. Conclusions

We propose two different scenarios as solution of determinization problem in FFTA. The first scenario includes two determinization algorithms. The first algorithm which is presented in Theorem 3.2 is appropriate for language preserving but the membership degree of terms in DFFTA and in its corresponding NFFTA may be different. The second algorithm which is presented in Theorem 3.5 is behavior preserving because the membership degrees of each term in NFFTA and its corresponding DFFTA are the same. The later algorithm works with specified type of fuzzy operators and exponentially increases the number of states of automaton rather than the first algorithm. In the second scenario, we define the algebra of fuzzy complex symbols to generalize the definition of state in automata. Then, we define the CFFTA and show that the CFFTA capability solves the determinization problem in FFTA. Another problem in fuzzy automata is that a state may have multiple-membership degree [6] in a position while processing a tree with a FFTA. As well, the ∇-normalization operation is defined in algebra of fuzzy complex symbols to solve the multiple-membership problem.

References

[1] M. A. Arbib, *From automata theory to brain theory*, Int. J. Man-Machine Studies, **7(3)** (1975), 279-295.

[2] S. Bozapalidis and O. L. Bozapalidoy, *Fuzzy tree language recognizability*, Fuzzy Sets and Systems, **161(5)** (2010), 716-734.

[3] H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, C. Loding, S. Tison and M. Tommasi, *Tree automata: techniques and applications*, Available: http://tata.gforge.inria.fr, 2007.

[4] J. E. Doner, *Decidability of the weak second-order theory of two successors*, Notices American Mathematic Society, **12** (1965), 365-486.

[5] J. E. Doner, *Tree acceptors and some of their applications*, Journal of Computer and System Science, **4** (1970), 406-451.

[6] M. Doostfatemeh and S. C. Kremer, *New directions in fuzzy automata*, Int. J. Approximate Reasoning, **38** (2005), 175-214.

[7] Z. Esik and L. Guangwu, *Fuzzy tree automata*, Fuzzy Sets and Systems, **158** (2007), 1450-1460.

[8] B. Gaines and L. J. Kohout, *The logic of automata*, Int. J. General Systems, **2** (1976), 191-208.

[9] M. Horry and M. M. Zahedi, *Hypergroups and general fuzzy automata*, Iranian Journal of Fuzzy Systems, **6(2)** (2009), 61-74.

[10] A. Mateescu, A. Salomaa, K. Salomaa and S. Yu, *Lexical analysis with a simple finite fuzzy automata model*, Journal of Universal Computing, **1(5)** (1995), 288-307.

[11] J. Mordeson and D. Malik, *Fuzzy automata and languages: theory and applications*, Chapman & Hall, London, 2002.

[12] M. O. Rabin, *FHandbook of mathematical logic*, North Holland, 1977.

[13] J. W. Thatcher and J. B. Wright, *Generalized finite automata with an application to a decision problem of second-order logic*, Mathematical System Theory, **2** (1968), 57-82.

[14] J. W. Thatcher and J. B. Wright, *Generalized finite automata*, Notices American Mathematic Society, **820** (1965).

[15] W. Thomas, *Handbook of theoretical computer science*, Elsevier, **B** (1990).

[16] W. G. Wee, *On generalization of adaptive algorithm and application of the fuzzy sets concept to pattern classification*, Ph.D. Dissertation, Purdue University, Lafayette, 1967.

[17] M. Ying, *A formal model of computing with words*, IEEE Transactions on Fuzzy Systems, **10(5)** (2002), 640-652.

[18] L. A. Zadeh, *Fuzzy sets*, Information and Control, **8** (1965), 338-353.

S. Moghari*, Department of Mathematics, Science Faculty, Alzahra University, Vanak, Tehran, Iran

E-mail address: `s_moghari@student.alzahra.ac.ir`

M. M. Zahedi, Department of Mathematics, Shahid Bahonar University of Kerman, Kerman, Iran

E-mail address: `zahedi_mm@mail.uk.ac.ir`

R. Ameri, College of Sciences, Tehran University, P.O. Box 14155-6455, Tehran, Iran

E-mail address: `ameri@umz.ac.ir`

*Corresponding author