

A Note on Fuzzy Tree Automata

S. R. Chaudhari

Department of Mathematics

Shivaji University,

Kolhapur (M. S.), India- 416004

Mukta N. Joshi

Department of Mathematics

Sinhgad College of Engineering,

Pune (M. S.),India

ABSTRACT

The aim of this note is to introduce fuzzy tree automaton over ranked alphabet (with and without ϵ -rules). In addition, we discuss completeness and reduction of fuzzy tree automata. Finally, we discuss closure properties of fuzzy tree languages.

Keywords:

Fuzzy tree automata, Fuzzy tree language, Complete and reduced fuzzy tree automata.

1. INTRODUCTION

Tree automaton was initially introduced by Doner and then by Thatcher and Wright for proving the decidability of the weak second order theory of multiple successors. Since then tree automaton is an important tool in computer science and have many applications in various areas of computer science. Few notable among them are i) automata on infinite words and trees are the basis of model-checking for formally verifying correctness of systems. ii) automatic verification of cryptographic protocols. iii) in compiler construction it help in compile time program analysis and to obtain information about the run-time behavior of programs iv) it is used in generating efficient code selectors v) tree automata with output are used for processing of XML documents vi) it provide the underlying guiding principles for data on the web. Fuzzy tree automaton provides a tool for dealing with vaguely defined trees. Application of fuzzy tree automaton for syntactic pattern recognition was first studied by Lee in 1982 [3]. The algebraic study of fuzzy tree automata was done by Zoltan and Liu [10]. Few more papers appear in literature on fuzzy automata dealing with various issues and applications [1,6,7,8,9]. The aim of this paper is to introduce fuzzy tree automata defined on ranked alphabet analogously to that of fuzzy automaton. We refer for tree automata and its basic properties a manograph by Comon et.al [2] and for fuzzy tree automata a manograph by Mordeson and Malik [5]. We have succeeded in defining fuzzy tree automata with ϵ -moves and proving its equivalent ness with fuzzy tree automata without having ϵ -moves. We have also tried to obtain complete and reduced fuzzy tree automaton equivalent

with the given one. We have derived two algorithms for minimal realization of fuzzy tree automata in an analogous way to that of for automata defined by Mohari et.al in [4]. The paper is arranged in four sections as described follows: Section 2 contains preliminaries related to fuzzy tree automata that are needed for remaining sections. Section 3 contains fuzzy tree automata with ϵ -rules and its equivalence with fuzzy tree automata without ϵ -rules. Further, in this section, we have obtain complete and reduced equivalent fuzzy tree automata for a given fuzzy tree automata. We have also constructed two algorithms for minimal realization for a given fuzzy tree automata. In the section 4 we have verified two closure properties namely union and intersection for fuzzy tree languages.

2. PRILIMINARIES

Let N_0 be the set of non-negative integers and N_0^* a set of finite strings over N_0 including empty string, ϵ . A ranked alphabet is a couple $(F, Arity)$, where F is nonempty finite set and $Arity : F \rightarrow N_0$ is a function. The arity of a symbol $f \in F$ is the number of arguments of the function f . Elements of arity 0, 1, 2, ..., n are respectively called constant, unary, binary, ..., n -ary symbols. For the sake of notational ease, throughout this paper, we use F for the ranked alphabet $(F, Arity)$. F_n denotes the set of all symbols of arity $n \geq 0$. We assume that F contain at least one constant. The set Y is disjoint with the set F_0 and is called the set of variables.

DEFINITION 1. Let F be the ranked alphabet and Y the set of variables. The smallest set $T(F, Y)$, containing Y , if $f \in F_n (n \geq 1)$ and $t_1, t_2, \dots, t_n \in T(F, Y)$ implies that $f(t_1, t_2, \dots, t_n) \in T(F, Y)$. If $Y = \phi$, then the set $T(F, Y)$ is called the set of all ground terms over F . In this case we shall denote it by $T(F)$.

EXAMPLE 1. Let $F = \{g(), f(), a, b\}$ and $Y = \{x, y\}$. Then g is a unary symbol, f is a binary symbol and a, b are constants and $g(f(x, a)), f(x, b)$ are just the terms over (F, Y) , the terms $f(a, g(f(b, a)))$ and $g(g(f(b, a)))$ are the ground terms over F .

DEFINITION 2. A finite subset U of N_0^* is called a finite tree domain, if the following conditions hold

- (1) $w \in U$ and $w = uv \Rightarrow u \in U$, where $u, v, w \in N_0^* - \{\epsilon\}$ (i.e. U is prefix closed).
- (2) $wn \in U$ and $m \leq n$ implies $wm \in U$, where $w \in N_0^* - \{\epsilon\}$ and $m, n \in N_0$

Let U be a finite tree domain. Then $\bar{U} = \{w \in U / w.1 \notin U\}$ is called the leaf node set.

EXAMPLE 2. Let $U = \{\epsilon, 0, 1, 2, 11, 12, 21, 22, 23\}$ be a finite tree domain. Then $\bar{U} = \{0, 11, 12, 21, 22, 23\}$ is the set of leaf nodes. (Note that 01 is not same as 1)

DEFINITION 3. A (rooted) tree on a ranked alphabet F and set of variables Y is a function $t : U \rightarrow F \cup Y$ such that $t(w) \in F$ (arity > 0), for $w \in U - \bar{U}$, $t(w) \in F \cup Y$, for $w \in \bar{U}$.

REMARK 4. Every term over F and Y can be treated as a rooted tree whose vertices are labeled by symbols of the term from left to right and whose decedents equal to the arity of the symbol.

EXAMPLE 3. A term $f(y, g(h(b, x)))$ is a rooted tree as shown below in figure 1.

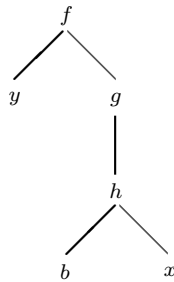


Fig. 1.

DEFINITION 5. A fuzzy tree automata (or FTA) over a ranked alphabet F is a triplet $A = (X, P, x_0)$, where X is called the set of non terminal nodes, $x_0 \in X$ and $P : X \times T(F; X) \rightarrow [0, 1]$. The elements of P are called fuzzy rewriting rules. For simplicity if $P(x, t) = c$, then we shall write $x \xrightarrow{c} t$.

EXAMPLE 4. Let $X = \{x_0, x, y\}$, $F = \{B, C, a, b\}$ and $P : x_0 \xrightarrow{c_1} B(x, a), y \xrightarrow{c_2} C(a, b), x \xrightarrow{c_3} b, y \xrightarrow{c_4} a$. Then $A = (X, P, x_0)$ is a FTA over a ranked alphabet F .

REMARK 6. If $t \in T(F; X)$ is a tree, then $p(t)$ denotes the term corresponds to the tree t . For the tree t in figure 1, the term $P(t)$ is $f(y, g(h(b, x)))$.

For all $\alpha, \beta \in T(F, X)$, we say that α derives β with degree $c \in [0, 1]$, symbolically $\alpha \xrightarrow{c} \beta$, if

- (1) $p(\alpha) = \lambda x \gamma$
- (2) $x \xrightarrow{c} t$ is in P and
- (3) $p(\beta) = \lambda p(t) \gamma$,

where $\lambda, \gamma \in [F \cup X \cup \{(\cdot, \cdot)\}]^*$, $x \in X$ and $t \in T(F; X)$. The transitive closure $\xrightarrow{*c}$ of \xrightarrow{c} is defined as follows:

- (1) $\alpha \xrightarrow{*1} \alpha$, for all $\alpha \in T(F; X)$
- (2) $\alpha \xrightarrow{*c} \beta$
iff $c = \bigvee_{c', c'' \in [0,1]} \left\{ c' \wedge c'' \mid \alpha \xrightarrow{c'} \gamma, \gamma \xrightarrow{c''} \beta \right\}$ Here onward, without any ambiguity we write $x \xrightarrow{c} t$ for both $x \xrightarrow{c} t$ and $x \xrightarrow{*c} t$.

DEFINITION 7. A ground term $t \in T(F)$ is said to be recognized by a FTA $A = (X, P, x_0)$ over a ranked alphabet F , if $x_0 \xrightarrow{c} t$, for some $c \in [0, 1]$. The fuzzy set $\{(t, e) / x_0 \xrightarrow{e} t, t \in T(F; X), e \in [0, 1]\}$ is called the fuzzy tree language generated by the FTA A over the ranked alphabet F . We shall denote it by $L(A)$.

EXAMPLE 5. Let $A = (X, P, x_0)$ be a FTA over a ranked alphabet F , where $X = \{x_0, y, z\}$, $F = \{B, a, b\}$ and $P : x_0 \xrightarrow{0.3} B(x_0, y), x_0 \xrightarrow{0.4} a, y \xrightarrow{0.5} B(a, b)$. Then $L(A) = \{(a, 0.4), (B(a, B(a, b)), 0.3), (B(B(a, B(a, b))), 0.3), (B(B(B...B(a, B(a, b))...)), 0.3)\}$.

DEFINITION 8. A fuzzy set L of $T(F)$ is said to be recognizable, if there exists a FTA A over a ranked alphabet F such that $L = L(A)$.

DEFINITION 9. Two FTA's over the same ranked alphabet F are said to be equivalent, if they generates the same fuzzy tree language.

EXAMPLE 6. Let $A = (X, P, x_0)$ be a FTA over a ranked alphabet F , where $X = \{x_0, y, z\}$, $F = \{B, a, b\}$ and $P : x_0 \xrightarrow{0.3} B(x_0, y), x_0 \xrightarrow{0.4} a, y \xrightarrow{0.5} B(a, b)$. Then $L(A) = \{(a, 0.4), (B(a, B(a, b)), 0.3), (B(B(a, B(a, b))), 0.3), (B(B(B...B(a, B(a, b))...)), 0.3)\}$.

Let $A_1 = (X_1, P_1, x_0)$ be a FTA over a ranked alphabet F , where $X_1 = \{x_0, y_1, z_1\}$ and $P_1 : x_0 \xrightarrow{0.4} B(y_1, z_1), y_1 \xrightarrow{0.3} a, x_0 \xrightarrow{0.4} a, y_1 \xrightarrow{0.5} B(y_1, z_1), z_1 \xrightarrow{0.5} B(a, b)$. Then $L(A_1) = \{(a, 0.4), (B(a, B(a, b)), 0.3), (B(B(a, B(a, b))), 0.3), (B(B(B...B(a, B(a, b))...)), 0.3)\}$. This shows that $A \cong A_1$.

3. FTA WITH ϵ -RULES AND THEIR MINIMIZATION

FTA with ϵ -rules In the section we introduce a form of fuzzy tree automata namely fuzzy tree automata with ϵ -rules and prove its equivalence with fuzzy tree automata without ϵ -rules, in the sense of same fuzzy language generation. We also obtain equivalent, minimal in the sense of non-terminal nodes and rules, form of given fuzzy tree automata namely complete reduced as well as minimal realization.

DEFINITION 10. A fuzzy tree automaton with ϵ -rules over a ranked alphabet F is a triplet $A = (X, P, x_0)$, where $P : X \times [T(F; X) \cup X] \rightarrow [0, 1]$ and X, x_0 are as in the Definition 2.8. Fuzzy rules $x \xrightarrow{c} y; x, y \in X$ and $c \in [0, 1]$ are called ϵ -rules.

DEFINITION 11. A ground term t is said to be recognized by FTA $A = (X, P, x_0)$ with ϵ -rule over a ranked alphabet F , if $x_0 \xrightarrow{c} t$, for some $c \in [0, 1]$. The fuzzy set $\{(t, e)/x_0 \xrightarrow{c} t, t \in T(F; X), e \in [0, 1]\}$ is called *the fuzzy tree language generated by FTA A with ϵ -rule over a ranked alphabet F* . we shall denote it by $L_\epsilon(A)$.

EXAMPLE 7. Let $A = (X, P, x_0)$ be a FTA with ϵ -rule over a ranked alphabet F , where $X = \{x_0, y\}$, $F = \{B, a, b\}$ and fuzzy rewriting rules $P : x_0 \xrightarrow{0.4} B(x_0, a)$, $x_0 \xrightarrow{0.5} y$, $y \xrightarrow{0.5} b$, $y \xrightarrow{0.6} B(a, a)$. Then the language $L_\epsilon(A) = \{(B(b, a), 0.4), (B(B(b, a), a), 0.4), (B(B(a, a), a), 0.4), (B(B(B...B(b, a), a)...), 0.4), (B(B(B...B(B(a, a), a)...), 0.4)\}$

THEOREM 12. Let L be a fuzzy tree language of a FTA A with ϵ -rule over a ranked alphabet F . Then there is a FTA A' over a ranked alphabet F such that $L(A') = L$.

PROOF. Let $A = (X, P, x_0)$ be FTA with ϵ -rule over a ranked alphabet F . Denote P_ϵ by a set of all ϵ -rules ($x \xrightarrow{c} y, x, y \in X, c \in [0, 1]$). Define $A' = (X', P', x_0)$, where $P' = P - P_\epsilon \cup \{z \xrightarrow{d \wedge c} t' / p(t') = \lambda x \gamma, z \xrightarrow{d} t, p(t) = \lambda y \gamma$ and $y \xrightarrow{c} x\}$ and X' is the set of nonterminals existing in P' , clearly A' is a FTA, (without ϵ -rules) over a ranked alphabet F . We now prove that $L_\epsilon(A) = L(A')$

Let $(t, e) \in L_\epsilon(A)$, then $x_0 \xrightarrow{c_1} t_1 \xrightarrow{c_2} t_2 \dots \xrightarrow{c_n} t_n \xrightarrow{c_{n+1}} t_{n+1} \dots \xrightarrow{c_r} t_r = t$ and $e = c_1 \wedge c_2 \wedge \dots \wedge c_r$. Suppose in the $(n+1)^{th}$ step $t_n \xrightarrow{c_{n+1}} t_{n+1}$ an ϵ -rule is used say $y \xrightarrow{c_n} x$. Then $p(t_{n-1}) = \gamma_1 y \gamma_2, p(t_n) = \lambda y \gamma, p(t_{n+1}) = \lambda x \gamma$. But then $x_0 \xrightarrow{c_1} t_1 \xrightarrow{c_2} t_2 \dots \xrightarrow{c_{n-2}} t_{n-2} \xrightarrow{c_{n-1} \wedge c_n} t'_{n-1} \xrightarrow{c_{n+1}} t_{n+1} \dots \xrightarrow{c_r} t_r = t$, where $p(t'_{n-1}) = \gamma_1 x \gamma_2, p(t_{n+1}) = \lambda x \gamma$ ie t is derived from x without ϵ -rules with degree e . Therefore $(t, e) \in L(A')$. Hence $L_\epsilon(A) \subseteq L(A')$.

Conversly, let $(t, e) \in L(A')$

Suppose $x \xrightarrow{c_1} t_1 \xrightarrow{c_2} t_2 \dots \xrightarrow{c_r} t_r = t$ and $e = c_1 \wedge c_2 \wedge \dots \wedge c_r$. If the rule used in the n^{th} step $t_{n-1} \xrightarrow{c_n} t_n$ is not present in A and $p(t_{n-1}) = \gamma_1 x \gamma_2, p(t_n) = \lambda x \gamma$, then by definition of P , there will be a derivation $t_{n-1} \xrightarrow{c} t'_n \xrightarrow{d} t_n$, where $p(t_{n-1}) = \gamma_1 y \gamma_2, p(t'_n) = \gamma_1 x \gamma_2$ and $p(t_n) = \lambda x \gamma$. Thus, there is an ϵ rule $y \xrightarrow{c} x \in A$ over a ranked alphabet F and $c_n = d \wedge c$. This proves that $(t, e) \in L_\epsilon(A)$. Hence $L_\epsilon(A) = L(A')$. \square

EXAMPLE 8. Let $A = (X, P, x_0)$ be FTA with ϵ -rule over a ranked alphabet F , where $X = \{x_0, x_1, y_1\}$, $F = \{B, C, a, b\}$, and $P : x_0 \xrightarrow{0.7} B(x_1, y_1), y_1 \xrightarrow{0.6} B(a, b), y_1 \xrightarrow{0.5} b$ and $x_1 \xrightarrow{0.4} y_1$. Let $A' = (X', P', x_0)$ be a FTA (without ϵ -rule) over a ranked alphabet F , where $X' = \{x_0, y_1\}$ and the production rules are $P' : x_0 \xrightarrow{0.4} B(y_1, y_1), y_1 \xrightarrow{0.6} B(a, b), y_1 \xrightarrow{0.5} b$ Then A' is equivalent to A .

DEFINITION 13. A FTA $A = (X, P, x_0)$ over a ranked alphabet F is said to be *complete*, if for every $x \in X$ there is at least one fuzzy rewriting rule of the form $x \xrightarrow{c} t$, for some $t \in T(F; X)$ and $c \in [0, 1]$.

The FTA over a ranked alphabet F defined in Example 3.3 is complete, whereas Example 2.12 is incomplete.

THEOREM 14. Let $L(A)$ be a fuzzy tree language of a FTA A over a ranked alphabet F . Then there exists a complete FTA A' over a ranked alphabet F that geneates $L(A)$.

PROOF. Let $A = (X, P, x_0)$ be an incomplete FTA over a ranked alphabet F . For each $z \in X$, for which no fuzzy production rule is exists in P , add an extra nonterminal node π_z , a node symbol C_z and fuzzy production rules (exactly two) namely $z \xrightarrow{c} C_z(\pi_z), \pi_z \xrightarrow{d} C_z(\pi_z)$, for some $c, d \in (0, 1]$. Now, constuct a FTA $A' = (X_1, P', x_0)$, where $X_1 = X \cup \{\pi\}$, $P' = P \cup \{z \xrightarrow{c} C_z(\pi_z), \pi_z \xrightarrow{d} C_z(\pi_z)\}$. Clearly A' is complete and $L(A) = L(A')$. \square

DEFINITION 15. A non terminal node $y \in X$ is said to be *accessible*, if y is one of the non terminal node of a term that is derivable from the initial nonterminal node.

DEFINITION 16. A non terminal node $y \in X$ is said to be *coaccessible*, if there exists a ground term $t \in T(F)$ such that $y \xrightarrow{c} t, c \in [0, 1]$.

EXAMPLE 9. Let $A = (X, P, x_0)$ be a FTA over a ranked alphabet F , where $X = \{x_0, y, z\}$, $F = \{B, C, a, b, c\}$ and $P : x_0 \xrightarrow{0.5} B(x_0, z), y \xrightarrow{0.6} C(c, b), z \xrightarrow{0.8} B(z, z), x_0 \xrightarrow{0.7} C(a, x_0), x_0 \xrightarrow{0.4} a, z \xrightarrow{0.6} c$. Here, x_0, z are accessible, whereas y is coaccessible.

DEFINITION 17. A node symbol is said to be accessible, if it exists in a term derivable from the initial non terminal node.

Notations:

- (1) N_{Ass} = Non accessible nonterminals.
- (2) N_{Coass} = Non coaccessible nonterminals.
- (3) $N_{Ass(y)}$ = Set of all non accessible nodes from nonterminal nodes y .
- (4) $P_{N_{Ass}}$ = Set of fuzzy production rules begin with non accessible nonterminal nodes.
- (5) $P_{N_{Coass}}$ = Set of fuzzy production rules begin with non coaccessible nonterminal nodes.

DEFINITION 18. A fuzzy tree automaton is said to be *reduced*, if its every nonterminal node is accessible as well as coaccessible.

EXAMPLE 10. Let $X = \{x_0, y, z\}, F = \{B, C, a, b, c\}$ and $P : x_0 \xrightarrow{0.5} B(x_0, z), x_0 \xrightarrow{0.4} C(y, z), y \xrightarrow{0.6} C(c, b), z \xrightarrow{0.8} B(z, z), x_0 \xrightarrow{0.7} C(a, x_0), x_0 \xrightarrow{0.4} a, z \xrightarrow{0.6} c$. Then $A = (X, P, x_0)$ is a reduced FTA over a ranked alphabet F .

THEOREM 19. Let $L(A)$ be a fuzzy tree language of a fuzzy tree automaton over a ranked alphabet F . Then there exists a reduced fuzzy tree automaton over the same rank alphabet F that accepts $L(A)$.

PROOF. let $A = (X, P, x_0)$ be a fuzzy tree automaton over a ranked alphabet F . Define $A' = (X_1, P_1, x_0)$ a reduced fuzzy tree automaton over the ranked alphabet F , where

$X' = X - \{N_{Ass} \cup N_{Coass}\}$, $y \in N_{Ass} \cup N_{Coass}$ and $P_1 = P - \{P_{N_{Ass}} \cup P_{N_{Coass}}\}$. If $(t, e) \in L(A')$, then $(t, e) \in L(A)$, since all the fuzzy production rules used in the derivation of $(t, e) \in L(A')$ are present in A . Conversely if $(t, e) \in L(A)$, then $x_0 \xrightarrow{c_1} t_1 \xrightarrow{c_2} t_2 \dots \xrightarrow{c_r} t_r = t$, where $c_1 \wedge c_2 \wedge \dots \wedge c_r = c$, for some $t_1, t_2, \dots, t_r \in T(F; X)$. Obviously t_1, t_2, \dots, t_r contains only accessible nonterminal nodes which is in P'_1 Therefore $(t, e) \in L(A')$. Hence, $L(A') = L(A)$. \square

DEFINITION 20. Two nonterminal node symbols are said to be *equivalent*, if they generate the same tree with same degree. That is if $x_1 \xrightarrow{c} t$ and $x_2 \xrightarrow{c} t$, then x_1, x_2 are equivalent.

DEFINITION 21. A FTA $A_m = (X_2, P_2, x_2)$ over a ranked alphabet F is minimal realization of FTA $A = (X, P, x_0)$ over a ranked alphabet F , if $L(A_m) = L(A)$ and there exist no equivalent nonterminal nodes in X .

ALGORITHM 22. Algorithm for minimal realization of a complete and reduced FTA $A = (X_1, P_1, x_0)$ over a ranked alphabet F

Step1: $R = \{x \in X_1/x \xrightarrow{c} t \in P_1, \text{ for some } c > 0\}$
 Step2: $P = \phi$
 Step3: $x \in R$
 Step4: $[x] = \{y \in R/x \xrightarrow{c} t_i \text{ and } y \xrightarrow{c} t_i, t_i \in T(F, F_0), i \in N\}$
 Add $[x]$ to P
 Step 5: $R = R - [x]$
 Goto step 3 until $R = \phi$
 $X_2 = P$
 Step 6: $P_2 : [x] \xrightarrow{c} t_i$
 where $y \xrightarrow{c} t_i, i = 1, 2, \dots, r$ and $y \in [x]$
 Step 7: $A_m = (X_2, P_2, [x_0])$

LEMMA 23. For each reduced and complete FTA A over a ranked alphabet F , there is a minimal realization A_m such that $L(A_m) = L(A)$.

PROOF. Follows due to the definition of the algorithm 3.7. \square

EXAMPLE 11. Let $A = (X, P_1, x)$ be a FTA, where $X = \{x, y, z\}$, $F = \{B, a, b\}$ and $P : x \xrightarrow{0.8} B(a, x), x \xrightarrow{0.8} a, x \xrightarrow{0.7} B(a, y), y \xrightarrow{0.6} B(a, y), y \xrightarrow{0.6} a, z \xrightarrow{0.8} B(z, a), z \xrightarrow{0.6} b$ Now, $x \xrightarrow{0.8} B(a, x) \xrightarrow{0.7} B(a, B(a, y)) \xrightarrow{0.6} B(a, B(a, a))$ ie $x \xrightarrow{0.6} B(a, B(a, a))$, where $0.6 = \min(0.8, 0.7, 0.6)$ and $y \xrightarrow{0.5} B(a, y) \xrightarrow{0.6} B(a, B(a, y)) \xrightarrow{0.6} B(a, B(a, a))$ ie $y \xrightarrow{0.5} B(a, B(a, a))$, where $0.6 = \min(0.6, 0.6)$ Then we get a FTA $A_m = (X_2, P_2, [x])$ over a ranked alphabet F , where $[x] = \{x, y\}$, $X_2 = \{[x], [z]\}$ and $P_2: [x] \xrightarrow{0.8} B(a, [x]), [x] \xrightarrow{0.8} a, [z] \xrightarrow{0.8} B(z, a)$ and $[z] \xrightarrow{0.6} b$.

If one allows to consider the language of a FTA $A = (X, P, x_0)$ over a ranked alphabet F is a crisp set $\{t/x_0 \xrightarrow{c} t, t \in T(F) \text{ and } e \in [0, 1]\}$, then the algorithm for minimal realization will be modified as follows:

ALGORITHM 24. Algorithm for minimal Realization of complete and reduced FTA $A = (X_1, P_1, x_0)$

Step1: $R = \{x \in X_1/x \xrightarrow{c} t \in P_1, \text{ for some } c > 0\}$
 Step2: $P = \phi$
 Step3: $x \in R$
 Step4: $[x] = \{y \in R/x \xrightarrow{c_1} t \text{ and } y \xrightarrow{c_2} t\}$
 Add $[x]$ to P
 Step5: $R = R - [x]$
 Goto step 3 until $R = \phi$
 $X_2 = P$
 Step6: $P_2 : [x] \xrightarrow{e} t$
 where $e = \max\{d_1, d_2, \dots, d_r\}$
 and $y_i \xrightarrow{e} t, i = 1, 2, \dots, r$ and $y_i \in [x]$
 Step7: $A_m = (X_2, P_2, [x_0])$

By redefining the equivalent non terminal nodes as : two non-terminal node symbols are said to be equivalent, if they generate same tree. ie $x_1 \xrightarrow{c_1} t$ and $x_2 \xrightarrow{c_2} t$, where $x_1, x_2 \in X$ and $c_1, c_2 \in [0, 1]$; One can prove the lemma 3.17 in more general setting by the use of the Algorithm 3.20.

4. CLOSURE PROPERTIES OF FTL'S

closure properties In this section we only verify two closure properties namely union and intersection for fuzzy tree languages.

THEOREM 25. The class of fuzzy tree languages is closed under union and intersection.

PROOF. UNION: Let $A = (X_1, P_1, x_1)$ and $B = (X_2, P_2, x_2)$ are FTA over a ranked alphabet F with fuzzy tree languages $L(A)$ and $L(B)$. Let us consider the fuzzy tree automaton $C = (X_1 \cup X_2 \cup \{x\}, P, x)$, where $x \notin X_1 \cup X_2$, and

$$P(y, t) = \begin{cases} P_1(y, t) & , \text{ if } y \in X_1 \\ P_2(y, t) & , \text{ if } y \in X_2 \\ 1 & , \text{ if } y = x \text{ and } t \in \{x_1, x_2\} \end{cases}$$

Claim: $L(C) = L(A) \cup L(B)$ Let $(t, e) \in L(C)$ that is $x \xrightarrow{e} t$ That is $x \xrightarrow{1} x_1 \xrightarrow{c_1} t_1 \xrightarrow{c_2} t_2 \dots \xrightarrow{c_n} t$ or $x \xrightarrow{1} x_2 \xrightarrow{d_1} t'_1 \xrightarrow{d_2} t'_2 \dots \xrightarrow{d_n} t$ Therefore $x_1 \xrightarrow{c} t$ or $x_2 \xrightarrow{e} t$ ie $(t, e) \in L(A)$ or $L(B)$ ie $(t, e) \in L(A) \cup L(B)$. This implies $L(C) \subseteq L(A) \cup L(B)$ Similarly $L(A) \cup L(B) \subseteq L(C)$.

INTERSECTION: Consider the fuzzy tree automaton $C = (X_1 \times X_2, P, (x_1, x_2))$, where $(x, y) \xrightarrow{e} t$ iff $x \xrightarrow{e} t \in P_1$ and $y \xrightarrow{e} t \in P_2$. **Claim:** $L(C) = L(A) \cap L(B)$. Let $(t, e) \in L(C)$. That is $(x_1, x_2) \xrightarrow{e} t$. Thus $x_1 \xrightarrow{e} t$ and $x_2 \xrightarrow{e} t$ i.e. $(t, e) \in L(A)$ and $(t, e) \in L(B)$. Thus $(t, e) \in L(A) \cap L(B)$. Therefore $L(C) \subseteq L(A) \cap L(B)$. Similarly $L(A) \cap L(B) \subseteq L(C)$. \square

REMARK 26. Since complement of fuzzy tree language can not be defined, we are unable to discuss closure property for complement.

Conclusion: The presently existing results that hold for (fuzzy) finite automata are shown to hold for fuzzy tree automata in this paper. Some of the results relating complements of fuzzy tree do not have any meaning, due the fact that complement for fuzzy tree can not be defined properly. Somehow, we have the additional advantage that associated results of fuzzy tree automata and languages can be used for handling fuzzy trees that appearing in various branches of computer science. By using these results (consecutive combination of them is as well possible) one can study fuzzy tree automata algebraically in varied direction such as decomposition of fuzzy tree automata, topological study of source and successors in fuzzy tree automata, generalized tree automata having continuous time variable which works parallel to fuzzy neural network etc. The resulting theories may be of interest to researchers in lexical analysis, code optimization and code generation phases of fuzzy compiler.

5. REFERENCES

- [1] Chaudhari S.R.,Komejwar D.D., On Fuzzy Pushdown Automata and their covering , *International Journal of Computer Applications* Vol.46(8),2012 pp 27-33
- [2] Comon H.,Dauchet M.,Gilleron R.,Jacquemard F., Lugiez D.,Loding C.,Tison S., and Tommasi M., *Tree Automata: techniques and applications.* (2007)
- [3] Lee E.T. Fuzzy tree automata and syntactic pattern recognition, *IEEE Trans Pattern Anal Mach Intell.* 1982 , 4(4):445-449.
- [4] Moghari S., Zahedi M.,and Ameri R., Minimization of Fuzzy Finite Tree Automata, *Proc.of 6th IMT-GT conference on mathematics, statistics and its applications, University of Tunku Abdul Rahman, Kuala Lumpur, Malaysia* pp 1044-1052.
- [5] Mordeson J.N. and Malik D., *Fuzzy Automata and Languages: Theory and Applications*,Chapman and Hill, 2002.
- [6] Qing E. W. , Wang T., Huang Y. X. , Li J. S.,State fusion of fuzzy automata with application on target tracking, *Comput. Math. with Appli.*, 57(2009), 949-960.
- [7] Qing E. W., Cui G., Application of Fuzzy Automata to Fuzzy Signal, *Processing of Intern.Conf. on Comput. Sci. and Software Engg.*, 5 (2008), 1283-1286.
- [8] Ramaswami V. , Girijamma H, Conversion of Finite Automata to Fuzzy Automata for String Comparison, *Int. J. of Computer Applications*, 37(8) (2012), 1-4.
- [9] Wee W.G, Fu K.S., A formulation of fuzzy automata and its application as a model of learning systems, *IEEE Trans. Syst. Sci. Cyber*, 5 (1969), 215-23.
- [10] Zoltan E. and Liu G. Fuzzy tree automata, *Fuzzy Sets and Systems* Vol. 158, Issue 13, 2007, Pp 1450-1460.