FUZZY GRAMMARS AND LANGUAGES

Dr. Komejwar Digambar Dattoba

Reg.No.U74120 MH2013 PTC 251205
Harshwardhan Publication Pvt.Ltd.
At.Post.Limbaganesh,Tq.Dist.Beed
Pin-431126 (Maharashtra) Cell:07588057695,09850203295
harshwardhanpubli@gmail.com, vidyawarta@gmail.com
All Types Educational & Reference Book Publisher & Distributors
www.vidyawarta.com

FUZZY GRAMMARS AND LANGUAGES

❖ **Publisher :**

Harshwardhan Publication Pvt.Ltd.
Limbaganesh, Dist. Beed (Maharashtra)
Pin-431126, vidyawarta@gmail.com

❖ **Printed by :**

Harshwardhan Publication Pvt.Ltd.
Limbaganesh, Dist. Beed, Pin-431126
www.vidyawarta.com

❖ **Page design & Cover :**

Shaikh Jahuroddin, Parli-V

❖ **Edition: January 2017**

ISBN 978-93-90618-39-2

❖ **Price : 200/ -**

FUZZY GRAMMARS AND LANGUAGES

CONTENTS

1. Preliminaries
  1.1 Introduction
  1.2 Automata, Grammars and their languages
  1.3 Fuzzy Sets
  1.4 Fuzzy Automata
  1.5 Fuzzy Grammars
2. Fuzzy Regular Grammars and Their Forms
  2.1 Introduction
  2.2 Fuzzy Regular Grammars
  2.3 Forms of Fuzzy Grammars
  2.4 Reduction of Fuzzy Grammars
3. Fuzzy Automaton Languages
  3.1 Introduction
  3.2 Fuzzy Automaton Crisp Languages and their closure properties
  3.3 Fuzzy Languages of Fuzzy Automaton
  3.4 Fuzzy Regular Languages

Preface

The present thesis is the study of some aspects of fuzzy automata, fuzzy languages and fuzzy grammars. Introduction of fuzzy sets, by Zadeh in 1965, offers very well-built tools for description of an uncertainty in common human reasoning. Therefore, this concept is applied in wide range of scientific areas including theory of computations. Lee and Zadeh [38] first introduced fuzzy grammars and fuzzy languages as an extension of their crisp counterpart. There after fuzzy grammars and fuzzy languages have been studied extensively [6, 19, 30, 49, 50, 63, 68]. In 1969, Wee and Fu [66] formulated a class of fuzzy automata and discussed an application of fuzzy automata as a model of learning systems. The advantage of employing fuzzy automaton as a learning model is its simplicity in design and computation. Inspired with this, Santos theoretically developed fuzzy automata in his series of papers [60 - 63] and their applications have discussed by many [18, 20, 29, 46, 54, 65]. Recently, researchers have shown lot of interest in studying fuzzy automata from different angles [2, 5, 10, 11, 15, 24, 28, 31, 51, 69], due to their potentiality of applications in vast areas. Also, there is an increasing interest on research in the fields such as image[105], speech and handwriting recognition [12, 13, 58] virus gene, protein and crystal structure prediction [52], neural networks [53] and clinical monitoring [9, 13, 59, 64], string matching [2], signal processing [57] etc. A simple technique for dealing such issues is a grammatical inference. An extraction of grammatical rules from the set example is of prime importance and which is relatively complex as well as vague in nature. Fuzzy grammars provide necessary frameworks for dealing this issue [6].

Chapter 1 entitled " Preliminaries" contains brief introduction of automata theory, formal grammars and languages. Here, we also report introductory definitions and results from fuzzy sets, fuzzy automata, fuzzy grammars and fuzzy pushdown automaton. These definitions and results will surve as the basic foundation of the thesis. Further, we elaborate the history of the problem at appropriate places.

Chapter 2 entitled " Fuzzy Regular grammars and their forms" is the study of fuzzy regular grammars with initial variables and their various equivalent forms including fuzzy automata with initial states discussed by Mordeson [42] and Kumbhojkar and Chaudhari [34]. We introduce fuzzy left, fuzzy right linear grammars, fuzzy grammars in normal form and show that they are all equivalent to fuzzy regular grammars expect for the empty string. We show that the reduction of fuzzy regular grammar leads to obtain Chomsky normal form for every fuzzy regular grammar. At the end of the chapter we discuss pumping lemma for fuzzy regular languages.

In the third chapter entitled " Fuzzy automaton languages", we discuss languages both crisp and fuzzy) generated by fuzzy automaton. We begin with few closure properties of *crisp languages* generated by fuzzy automaton [26, 42]. We point out that *fuzzy languages* generated by fuzzy automata also satisfies similar closure properties. To establish them we use the mechanism of extension principle [32].We show that only the closure property for the complement is not preserved. We end this chapter by finding equivalent fuzzy regular grammar that generates fuzzy automata language and conversely. In sum generalization of crisp languages generated by fuzzy automata to fuzzy languages generated by fuzzy automaton is computationally easy but one may lose (few) closure properties.

Preliminaries

1.1 Introduction

In this chapter elementary concept and results from fuzzy set theory, automata theory and grammars are given. These concepts and results are necessary for the remaining part of the thesis. Few elementary concepts needed for the developments of the theory are also discussed at appropriate places throughout the thesis.

1.2 Automata, Grammars and their Languages

Let Σ be a non-empty finite set of *alphabet*. A *word* is a finite sequence of elements over Σ. The set of all words over Σ is denoted by Σ^{+}. The symbol Λ denotes the word without any elements, called the *empty word* .Let $\Sigma^{*} = \Sigma^{+} \cup \{\Lambda\}$. It is obvious that Σ^* is a monoid under usual juxtaposition of words.

Definition 1.2.1 [21] A *finite state machine* is a triplet $M = (Q, \Sigma, P)$, , where Q, Σ are non-empty finite sets and $P : Q \times \Sigma \to Q$ is a partial function. M is called *non-deterministic* finite state machine, when P is a relation.

Definition 1.2.2 [21, 22] A *finite state automaton* is a five tuple $M = (Q, \Sigma, \delta, q_0, F)$, where

i) Q is a finite nonempty set of *states*.

ii) Σ is a finite nonempty set, called the *input alphabet*.

iii) δ is a function from $Q \times \Sigma$ into Q and is usually called the *state transition function*.

iv) $q_0 \in Q$ is called the *initial state*.

v) $F \subseteq Q$ is a set called of *final states*.

The transition $\delta(q, a) = p$ means that p is the next state, when δ is in state q, and the input is a. The state transition function δ is naturally extended to a δ^* function $\delta^* : Q \times \Sigma^* \to Q$ as follows:

1) $\delta^*(q, \Lambda) = q, \forall q \in Q.$

2) $\delta^*(q, xa) = \delta(\delta^*(q, x), a), \forall q \in Q, x \in \Sigma^*$ and $a \in \Sigma.$

When no confusion arises we shall write δ for δ^*.

Definition 1.2.3 A string $x \in \Sigma^*$ is said to be *accepted* by a finite automaton $M = (Q, \Sigma, \delta, q_0, F)$, if $\delta(q_0, x) = q,$ for some $q \in F$.

This is basically the acceptability of a string by the final state.

Definition 1.2.4 Let $M=(Q, \Sigma, \delta, q_0, F)$ be a finite automaton. Then the set $L(M)=\{x \in \Sigma^* \mid \delta(q_0, x)=q, \text{ for some } q \in F\}$ is called the *language* of M.

Definition 1.2.5 A subset $L \subseteq \Sigma^*$ is called *finite automaton language*, if there exists a finite automaton $M=(Q, \Sigma, \delta, q_0, F)$ such that $L(M)=L.$

We may note that

(1) The class of finite automaton languages is closed under union, intersection and complementation.

(2) The class of finite automaton languages is closed under the quotient with arbitrary sets.

Definition 1.2.6 A function $\rho : \Sigma^* \to \Sigma^*$ is called the *reversal, if* $\rho(\Lambda) = \Lambda$, $\rho(a) = a$ and $\rho(xa) = \rho(a)\rho(x), \forall a \in \Sigma, x \in \Sigma^*$.

Clearly, $\rho(\rho(x)) = x$ and

$\rho(xy) = \rho(y)\rho(x), \ \forall x, y \in \Sigma^*$.

Let $L \subseteq \Sigma^*$. Then the set $L^\rho = \{\rho(x) : x \in L\}$ is called the *reversal* of L.

Theorem 1.2.7 Let L be finite automaton language, then so is L^ρ.

Wee and Fu [66] have first extended finite automata by using fuzzy sets and discussed their applications for learning model. Many researchers have studied fuzzy finite automata from different angles [2,5,10,15,24,28,31,51,69]. Few introductory concepts are discussed in section 1.5. Here, in this thesis we discuss few closure properties of fuzzy automata languages in chapter 2 (for crisp languages) and chapter 3 (for fuzzy languages).

Definition 1.2.8 [22] A *phrase-structure grammar* or *grammar* is a four tuple $G = (V, \Sigma, S, P)$, where

i) V is a finite nonempty set whose elements are called *variables*.

ii) Σ is a finite nonempty set whose elements are called *terminals*.

iii) S is a special symbol called the *start symbol*.

iv) P is a finite set whose elements are $\alpha \to \beta$, where α and β are strings over $V \cup \Sigma$, α has at least one symbol from V.

The elements of P are called *productions* or *rules* of the grammar G.

Definition 1.2.9 [22] A grammar $G = (V, \Sigma, S, P)$ is called

(1) *regular*, if all its productions are of the type $A \to aB$ or $A \to a$,

(2) *context free*, if all its productions are of the type $A \to \alpha$, where $A \in V$ and

$\alpha \in (V \cup \Sigma)^*$.

(3) *context sensitive*, if all its productions are of the type $\phi A \psi \to \phi \alpha \psi$ and $\alpha \neq \Lambda$. The productions $S \to \Lambda$ is allowed in a context sensitive grammar, but then S does not appear on the right side of any production.

(4) *unrestricted*, if its productions are of the type $\alpha \to \beta$, while $\alpha, \beta \in (V \cup \Sigma)^*$.

where $A, B \in V$, $a \in \Sigma \cup \{\Lambda\}$ and $\alpha, \beta \in (V \cup \Sigma)^*$.

If we denote $\mathcal{G}_{\mathfrak{R}}$ ($\mathcal{G}_{\mathfrak{CF}}$, $\mathcal{G}_{\mathfrak{CS}}$ and $\mathcal{G}_{\mathfrak{U}}$) the class of all regular (context free, context sensitive and unrestricted respectively) grammars, then $\mathcal{G}_{\mathfrak{R}} \subseteq \mathcal{G}_{\mathfrak{CF}} \subseteq \mathcal{G}_{\mathfrak{CS}} \subseteq \mathcal{G}_{\mathfrak{U}}$.

Following can be established easily

Theorem 1.2.10 For given finite automaton $M = (Q, \Sigma, \delta, q_0, F)$, there is a regular grammar G such that $L(G) = L(M) - \{\Lambda\}$.

Theorem 1.2.11 For given regular grammar $G = (V, \Sigma, S, P)$, there is a finite automaton M such that $L(M) = L(G)$.

Fuzzy grammars with crisp initial state and fuzzy languages were first discussed by Lee and Zadeh [38]. Many researches have studied fuzzy grammars from different perspectives [3, 4, 30, 50]. In the chapter 2 (crisp languages) and chapter 3 (fuzzy languages), we will establish that fuzzy grammar and fuzzy automaton are equivalent in the sense of their language generation.

1.4 Fuzzy Sets

Definition 1.4.1 [18] A *fuzzy set* A in U is a function A: $U \rightarrow$ [0, 1]. For any $x \in U$, $A(x)$ is called the *grade membership* of x in A.

Clearly, if we replace the closed interval [0, 1] by the set {0, 1}, then the set A of U is a characteristic function of the subset $\{x \in U \mid A(x) = 1\}$.

Thus, fuzzy set is generalization of a crisp set.

Definition 1.4.2 Let A and B be two fuzzy sets in U. Then *union* and *intersection* of A and B are respectively defined by membership functions

$A \cup B(x) = \max\{A(x), B(x)\}$ and

$A \cap B(x) = \min\{A(x), B(x)\}$.

Definition 1.4.3 A fuzzy set $R: U \times V \to [0, 1]$ is called a *fuzzy relation* from U to V.

Definition 1.4.4 Let $R_1: U \times V \to [0, 1]$ and $R_2: V \times W \to [0, 1]$ be two fuzzy relations. Then *the composition*, $R_1 \circ R_2$, of R_1 and R_2 is a fuzzy relation from U to W, which is defined as follows:

$$R_1 \circ R_2(x, z) = \bigvee \{R_1(x, y) \wedge R_2(y, z) \mid y \in V\}$$

Clearly,

Theorem 1.4.5 Let $R_1: U \times V \to [0, 1]$, $R_2: V \times W \to [0, 1]$ and $R_3: W \times Z \to [0, 1]$ be fuzzy relations. Then $R_1 \circ (R_2 \circ R_3) = (R_1 \circ R_2) \circ R_3$.

1.5 Fuzzy Automata

Now, we shall recall fuzzy automata and languages generated by them :

Definition 1.5.1 [43, 44, 45] A *fuzzy finite state machine* is a triplet $m = (Q, \Sigma, \mu)$, where Q and Σ are non-empty finite sets of states and inputs respectively and $\mu : Q \times \Sigma \times Q \to [0, 1]$ is a fuzzy relation from $Q \times \Sigma$ to Q.

Remark 1.5.2 The fuzzy set μ induces a fuzzy set $\mu^* : Q \times \Sigma^* \times Q \to [0, 1]$ as follows

$$\mu^*(q, \Lambda, p) = \begin{cases} 1, & \text{if } q = p \\ 0, & \text{if } q \neq p \end{cases} \quad \text{and}$$

Clearly,

$\mu^*(q, xa, p)$ =

$\vee\{\mu^*(q, x, r)\wedge\mu(r, a, p) \mid r\in Q\}, \forall x\in\Sigma^*, a\in\Sigma$ *and* $p, q\in Q$
[ ]

Definition 1.5.3 [42] A *fuzzy automaton* is a five tuple $M=(Q, \Sigma, \mu, I, F)$, where

Q is a finite non-empty set of states,

Σ is a finite non-empty set of inputs,

$\mu: Q \times \Sigma \times Q\to[0, 1]$ is a fuzzy set, called the *fuzzy transition function*,

I is a fuzzy set of Q, called the *fuzzy initial state* and

F is a fuzzy subset of Q, called the *fuzzy final stae.*

Remark 1.5.4 The above definition is similar to that of Santos [60], except for fuzzy sets which are probabilistic there. Therefore, above fuzzy finite automaton is an extension of non-deterministic finite automaton rather than deterministic one. Recently, Belolavek [5] discussed L-fuzzy finite automaton M, where [0, 1] is extended to lattice ordered monoid, he proved that this L-fuzzy automation is equivalent to deterministic fuzzy finite automaton, $M = (Q, \Sigma, \mu, I, F)$, where $\mu : Q \times \Sigma \to Q$ and I is a crisp initial state rather than a fuzzy set of states. Initially, Lan and Zheiwen [37] have discussed closure properties of languages accepted by fuzzy finite automaton, whose initial state and final states are crisp in nature. In [42] Malik and Mordeson have established various algebraic properties of fuzzy regular languages for deterministic fuzzy finite automaton of [41] along with initial fuzzy set. Very recently, Li and Li [39] have studied algebraic properties of $L-$ fuzzy finite automaton.

. Then the set $L(M)=\{x \in \Sigma^{*} / x \text{ is accepted by } M\}$ is called the *language* of M .

In literature some researchers have denote this language of M by fuzzy regular language [37, 56]. However, this will be justified only after the Corollary 2.2.5 and 2.2.7.

Definition 1.5.7 A language $L \subseteq \Sigma^{*}$ is called *fuzzy automaton language*, if there exists a fuzzy automaton M such that $L(M)=L$.

As discussed in the section 1.2, we will establish various closure properties of fuzzy automaton languages in the chapter 2. We also extend them for fuzzy languages generated by fuzzy automaton in the chapter 3.

1.6 Fuzzy Grammar

In this subsection we will discuss elementary concepts of fuzzy grammars which were established by Lee and Zadeh in [38].

Definition 1.6.1 A *fuzzy grammar* is a four tuple $G = (V, \Sigma, S, P)$, where V is a non-empty finite set of *variables*, Σ is a non-empty finite set of *terminals* such that $V \cap \Sigma = \phi$, S is a fuzzy set of V, i.e. $S : V \to [0,\ 1]$, called the set of *fuzzy initial variables* and P is a fuzzy set of $(V \cup \Sigma)^* \times (V \cup \Sigma)^*$ such that the following condition holds

If $P(\alpha,\ \beta) = \sigma > 0,\ for\ \alpha,\ \beta \in (V \cup \Sigma)^*$, then the string α contains at least one variable from V.

Remark 1.6.2 This definition is different from that of fuzzy grammar introduced in [38] in two respects. Firstly, it allows any variable as an initial with grade than a crisp variable considered in [38]. The importance of doing so is explained by many authors. Secondly, whenever $P(\alpha, \beta) = \sigma > 0$, then α must contains at least one variable from V.

Throughout this thesis, $P(\alpha, \beta) = \sigma > 0$, we shall mean α *derives* β with grade membership σ and we shall write it as $\alpha \xrightarrow{\sigma} \beta$. We call $\alpha \xrightarrow{\sigma} \beta$ a *fuzzy production* and hence the set P of fuzzy *productions*.

Definition 1.6.3 [38] Let $\alpha \xrightarrow{\sigma} \beta$ be a fuzzy production of G and $\gamma, \delta \in (\Sigma \cup V)^*$. Then

$\gamma\alpha\delta \xrightarrow{\sigma} \gamma\beta\delta$ is also a fuzzy production. In this case we say that $\gamma\beta\delta$ is *directly derivable* from $\gamma\alpha\delta$ or $\gamma\alpha\delta$ *directly derives* $\gamma\beta\delta$.

Let $\alpha, \beta \in (V \cup \Sigma)^*$. Then β is said to be *derivable* from α, if there exist $\alpha_1, \alpha_2, \cdots, \alpha_n \in (V \cup \Sigma)^*$ and $\sigma_0, \sigma_1, \ldots, \sigma_n \in (0, 1]$ such that $\alpha \xrightarrow{\sigma_0} \alpha_1$, $\alpha_1 \xrightarrow{\sigma_1} \alpha_2$, ..., $\alpha_n \xrightarrow{\sigma_n} \beta$ are elements of P.

Definition 1.6.4 [38] An expression $\alpha_1 \xrightarrow{\sigma_1} \alpha_2 \xrightarrow{\sigma_2} \alpha_3 \ldots \alpha_{n-1} \xrightarrow{\sigma_{n-1}} \alpha_n$ is called a *derivation chain* from α_1 to α_n.

The *degree of derivability* of α_n from α_1 denoted by $d(\alpha_1 \Rightarrow \alpha_n)$, will be defined as $d(\alpha_1 \Rightarrow \alpha_n) = \sup\{\min(\sigma_1, \sigma_2, \ldots, \sigma_{n-1})\}$, where the supremum is taken over all derivation chains $\alpha_1 \xrightarrow{\sigma_1} \alpha_2 \xrightarrow{\sigma_2} \alpha_3 \ldots \alpha_{n-1} \xrightarrow{\sigma_{n-1}} \alpha_n$ from α_1 to α_n.

Definition 1.6.5 A string x of terminals is said to be *derivable* from a variable $A \in V$, symbolically $A \Rightarrow x$, if there is at least one derivation chain from A to x.

The set of all strings derivable from $A \in V$ with $S(A) > 0$ is called *the language generated by* G. We shall denote it by $L(G)$.

Two fuzzy grammars G_1 *and* G_2 are said to be equivalent, if $L(G_1) = L(G_2)$, we shall write them as $G_1 \square G_2$. We use the concepts developed in this section to establish equivalence between fuzzy automata and fuzzy grammars in the chapter 2 and chapter 3.

Chapter 2

Fuzzy Regular Grammars and their Forms

2.1 Introduction

A kind of ambiguity (due to vagueness) that exists in natural language can partially be tackled with the help of fuzzy set theory. One such attempt was made by **Lee and Zadeh** [38] by fuzzyfying a formal grammar. Many researchers have then contributed in developing the fuzzy theory of grammars from different angles [8, 19, 30, 47-50, 63]. Lee and Zadeh have proved that the context sensitive fuzzy grammar is always recursive. They have also obtained the Chomsky and the Greibach normal form for a given fuzzy context free grammar. **Gerla** [19] established that, a fuzzy language is recursive if and only if it is generated by a fuzzy grammar. Generalization of fuzzy grammar namely $L-$grammars was studied mainly by **Li and Pedrycz** [40], **Li and Li** [56] etc. **Lan and Zhiwen** [28] have discussed a relationship between a fuzzy grammar and a fuzzy finite automaton. **Malik and Mordeson** [41] have introduced a max-min fuzzy language and find a fuzzy automaton that generates this language. **Asveld** [3, 4] generalized fuzzy context free grammar and used them to model grammatical errors occurs in robust recognizing and parsing algorithm.

In this chapter, we study fuzzy regular grammars and their various forms. We show that fuzzy max min finite automaton (stationary) introduced by **Santos** [60] as well as **Mordeson [41]** is equivalent to the fuzzy regular grammar. We begin with the motivation for introducing fuzzy regular grammar that generates a crisp language. Sometimes in a sentence fuzziness occurs due to insertion or deletion or incorrect spelling of words (letters). For example, a sentence “I am a boy “ may sometimes be typed as “I am am a boy“ or “I am a by “ or “I am a booy “ etc. To deal with such fuzziness we do not deem necessary to consider a fuzzy language, but a crisp language with some degree of acceptance. To derive such sentences we consider a fuzzy regular grammar with production rules:

$A \xrightarrow{1} I\,B, B \xrightarrow{n_1} am\ C / am\,am\ C / aam\ C / amm\ C$ and

$C \xrightarrow{n_2} a\, boy / a\, by / a\, booy$, where

$$n_1 = \begin{cases} \dfrac{3}{\begin{array}{c}\text{no. of a's} + \text{no. of b's in the antisedent} \\ \text{part of the second production rule}\end{array}}, & \begin{array}{l}\text{if length of antisedent part of} \\ \text{the second production rule is} \geq 3\end{array} \\ \dfrac{1}{|x|^2}, & \begin{array}{l}\text{if } |x| = |\text{no. of symbols of antisedent} \\ \text{part of the second production rule is} < 3\end{array} \end{cases}$$

and

$$n_2 = \begin{cases} \dfrac{4}{\begin{array}{c}\text{no. of a's} + \text{no. of b's} + \text{no. of } o\text{'s} + \text{no. of } y\text{'s in the} \\ \text{antisedent part of the third production rule}\end{array}}, & \begin{array}{l}\text{if length of antisedent part of the} \\ \text{third production rule is} \geq 4\end{array} \\ \dfrac{1}{|y|^2}, & \begin{array}{l}\text{if } |y| = |\text{ no. of symbols of antisedent} \\ \text{part of the third production rule is} < 4\end{array} \end{cases}$$

Clearly, the sentence “I am a boy “ is derived from this fuzzy grammar with degree $1 \wedge n_1 \wedge n_2 = 1$ and “I am a booy “is derived from this grammar with degree $1 \wedge n_1 \wedge n_2 = 0.8$ etc.

Thus, we introduce fuzzy regular grammar with fuzzy initial variables and the forms of fuzzy grammar such as fuzzy right grammar, fuzzy left grammar, fuzzy grammar in normal form. We show that they all are equivalent to fuzzy automaton with fuzzy initial states, in the sense that they generate the same language.

2.2 Fuzzy Regular Grammar

As we have stated in the introduction, we introduce fuzzy regular grammar with fuzzy initial variables and accordingly we define an acceptance of a string. We prove that fuzzy regular grammar and fuzzy finite automaton are equivalent (except for empty word Λ), in the sense that they generate the same language. We also provide examples to illustrate this. In the next section, We introduce various forms of fuzzy grammar and prove that all these forms are equivalent to fuzzy regular grammar. Recall that a fuzzy grammar is a quintuple $G=(V, \Sigma, S, P)$, where V and Σ are non-empty sets, S is a fuzzy set of V and P is a fuzzy set of $(V \cup \Sigma)^* \times (V \cup \Sigma)^*$ such that the following condition holds

If $P(\alpha, \beta)=\sigma>0,\ for\ \alpha, \beta \in (V \cup \Sigma)^*$, then the string α contains at least one variable from V.

We now define fuzzy regular grammar below

Definition 2.2.1 A fuzzy grammar $G=(V, \Sigma, S, P)$ is said to be *regular,* if each of its fuzzy production is either of the form $A \xrightarrow{\sigma} aB$ or $A \xrightarrow{\sigma} a$, where $A, B \in V$, $a \in \Sigma$ and $\sigma > 0$.

Definition 2.2.2 A string $x \in \Sigma^*$ is said to be *generated* by the fuzzy regular grammar G, if, $\bigvee_{A_0 \in V} \{S(A_0) \wedge d(A_0 \Rightarrow x)\} > 0.$

Equivalently, there exists $A_0 \in V$ such that $S(A_0) > 0$ and $d(A_0 \Rightarrow x) > 0.$

Definition 2.2.3 The set of all strings $x \in \Sigma^*$ that are generated by a fuzzy regular grammar G is called the *fuzzy regular language*. We shall denote it by $L(G)$.

Example 2.2.4 Suppose $G = (V, \Sigma, S, P)$ is a fuzzy regular grammar, where $V = \{q_0, q_1, q_2\}$, $\Sigma = \{0, 1\}$, $S = \{0.1/q_0, 0.2/q_1\}$,

$$P = \{q_0 \xrightarrow{0.1} 1q_2, q_0 \xrightarrow{0.3} 0q_0, q_1 \xrightarrow{0.6} 1q_2,$$
$$q_1 \xrightarrow{0.5} 0q_1,\ q_2 \xrightarrow{0.4} 0q_1,$$
$$q_1 \xrightarrow{0.2} 0, q_2 \xrightarrow{0.7} 1\}.$$

Then

$L(G) = \left(0^{*}1(1+0^{+}0+(0^{+}1)^{*}1)\right) + \left(0^{*}(1\ 0^{+})^{*}0+0^{*}1(0^{+}1)^{*}1\right)$,

where strings of $0^{*}1\left(1+0^{+}0+(0^{+}1)^{*}1\right)$ are having degree 0.1 and strings of $0^{*}(1\ 0^{+})^{*}0+0^{*}1(0^{+}1)^{*}1$ are having degree 0.2.

The above three definitions coincides with the definitions of [16], if the fuzzy initial state S is treated as *a crisp* initial state and $S \xrightarrow{1} \wedge$ is allowed in P. This slight change in the above definition allows to think of the equivalentness of a fuzzy finite automaton discussed in [7, 44] and fuzzy regular grammar defined above.

Theorem 2.2.5 Given a fuzzy finite automaton $M = (Q,\ \Sigma,\ \mu,\ I,\ F)$, there is a fuzzy regular grammar G such that $L(G) = L(M) - \{\Lambda\}$.

Proof Consider $V = Q,\ S = I$ and define P as:

$$P = \{p \xrightarrow{\sigma} aq \mid \mu(p, a, q) = \sigma > 0\} \cup \{p \xrightarrow{\sigma} a \mid \mu(p, a, q) = \sigma \text{ and } F(q) > 0\}$$

Clearly, $G = (V,\ \Sigma,\ S,\ P)$ is a required fuzzy regular grammar.

We construct an example to depict the above theorem.

Example 2.2.6 Consider a fuzzy finite automata $M = (Q, \Sigma, \mu, I, F)$, where $Q = \{q_0, q_1, q_2\}$ $I = \{0.7/q_0\}$, $F = \{1/q_1, 0.8/q_2\}$, $\Sigma = \{0,\ 1\}$ and

$\mu(q_0, 0, q_0) = 0.2$, $\mu(q_0, 0, q_1) = 0.5$, $\mu(q_0, 1, q_2) = 0.4$,
$\mu(q_1, 0, q_1) = 0.3$, $\mu(q_1, 1, q_2) = 0.6$, $\mu(q_1, 0, q_0) = 0.7$.

Then, $L(M) = 0^*1 + 0^+$ with string 1 is of degree 0.4, string 0 is of degree 0.5, strings of 0^+1 are of degree 0.5, strings of $(00)^{+}$ are of degree 0.3 and strings of $(000)^{+}$ are of degree 0.5.

By the use of above theorem one can construct an equivalent fuzzy regular grammar G as : $G=(\{q_0,q_1,q_2\},\{0,1\},\{0.7/q_0\},P)$, where

$$p=\{q_0\xrightarrow{0.2}0q_0, q_0\xrightarrow{0.5}0, q_0\xrightarrow{0.4}1, q_1\xrightarrow{0.3}0, q_1\xrightarrow{0.7}0q_0, q_1\xrightarrow{0.6}1\}.$$

We now establish the converse of the above theorem.

Theorem 2.2.7 Given a fuzzy regular grammar $G=(V, \Sigma, S, P)$, there is a fuzzy finite automaton M such that $L(M)=L(G)$

Proof Construction of the required fuzzy finite automaton will be as follows:

$Q = V \cup \{q_f\}$, where q_f not in V , $I = S$, $F = \{1/q_f\}$

and μ is defined as:

$$\mu(q,a,p) = \sigma \quad \text{if and only if} \quad q \xrightarrow{\sigma} ap$$
$$\mu(q,a,q_f) = \sigma \quad \text{if and only if} \quad q \xrightarrow{\sigma} a$$

Theorem 2.2.5 and theorem 2.2.7 combinely establishes the equivalence between the fuzzy finite automaton and the fuzzy regular grammar.

2.3 Forms of Fuzzy Grammars

In this section we introduce various forms of fuzzy grammars namely fuzzy grammar in normal form, fuzzy linear grammar, fuzzy left linear grammar and fuzzy right linear grammar. We prove that all these forms of fuzzy grammars are equivalent to a fuzzy regular grammar in the sense that, they generate the same language.

Definition 2.3.1 A fuzzy grammar $G = (V, \Sigma, S, P)$ is said to be in *normal form* if it has either of the following types fuzzy production $A \xrightarrow{\sigma} aB$ or $A \xrightarrow{\sigma} \Lambda$, where $A, B \in V$, $a \in \Sigma$, and $\sigma > 0$

Remark 2.3.2 Every fuzzy regular grammar can be made fuzzy grammar in normal form by changing its fuzzy production $A \xrightarrow{\sigma} a$ by two fuzzy productions $A \xrightarrow{\sigma} aC$, $C \notin V$ and $\mathrm{C} \xrightarrow{\sigma} \Lambda$

Theorem 2.3.3 For every fuzzy grammar $G = (V, \Sigma, S, P)$ in normal form, there is a fuzzy finite automaton M such that $L(M) = L(G)$.

Proof Consider $Q = V$, $I = S$. Define $F: Q \to [0, 1]$

by $F(q) = \sigma$ if $q \xrightarrow{\sigma} \Lambda \in P$

and $F(q) = 0$, otherwise; and define $\mu: Q \times \Sigma \times Q \to [0, 1]$

by $\mu(q,\ a,\ p)=\sigma$, if $q \xrightarrow{\sigma} ap \in P$ and

$\mu(q, a, p)=0$, otherwise.

We prove that fuzzy finite automaton $M = (\ Q,\ \Sigma,\ \mu,\ I,\ F\ )$ is equivalent to G.

Let $x \in L(G)$ and $x = a_1 a_2 \ldots a_{n-1} a_n$, implies that for $a_i \in \Sigma$ $S(q_0) > 0$ and $d(q_0 \Rightarrow x) > 0$, for some $q_0 \in V$.

Since $d(q_0 \Rightarrow x) > 0$ we have there exist $q_1,\ q_2,\ \ldots,\ q_n \in V$ and $\sigma_1,\ \sigma_2,\ \ldots,\ \sigma_{n+1} \in (0,\ 1]$ such that

$$q_0 \xrightarrow{\sigma_1} a_1 q_1 \xrightarrow{\sigma_2} a_1 a_2 q_2 \ \ldots \ \xrightarrow{\sigma_n} a_1 a_2 \ldots a_n q_n \xrightarrow{\sigma_{n+1}} a_1 a_2 \ldots a_n \Lambda = x$$

Then P must have following fuzzy productions:

$q_0 \xrightarrow{\sigma_1} a_1 q_1$, $q_1 \xrightarrow{\sigma_2} a_2 q_2$, ... , $q_{n-1} \xrightarrow{\sigma_n} a_n q_n$ and

$q_n \xrightarrow{\sigma_{n+1}} \Lambda$

Therefore, $\mu(q_0, a_1, q_1)=\sigma_1$, $\mu(q_1, a_2, q_2)=\sigma_2$, ... ,$\mu(q_{n-1}, a_n, q_n)=\sigma_n$ and $F(q_n)=\sigma_{n+1}$

But then,

$\mu(q_0, a_1, q_1) \wedge \mu(q_1, a_2, q_2) \wedge \ldots \wedge \mu(q_{n-2}, a_{n-1}, q_{n-1}) \wedge \mu(q_{n-1}, a_n, q_n) > 0$

and $F(q_n) > 0$

Thus, $\bigvee \{I(q) \wedge \mu(q, x, p) \wedge F(p) \mid q, p \in Q\} > 0$

i.e. $x \in L(M)$

Converse is similar.

Theorem 2.2.5 enables us to conclude that

Corollary 2.3.4 For every fuzzy grammar G in normal form, there is a fuzzy regular grammar G_1 such that $L(G_1) = L(G) - \{\Lambda\}$.

Definition 2.3.5 A fuzzy grammar $G = (V, \Sigma, S, P)$ is called *linear grammar*, if it has either of the following productions:

(i) $A \xrightarrow{\sigma} x$ or

(ii) $A \xrightarrow{\sigma} x_1 B x_2$, where $A, B \in V, x_1, x_2, x \in \Sigma^*$ and $\sigma > 0$

If $x_1 = \Lambda$ then G is called the *left linear grammar* and if $x_2 = \Lambda$ then it is called the *right linear grammar*.

Definition 2.3.6 A language $L \subseteq \Sigma^*$ is *fuzzy linear* (*left linear* , *right linear*), if there is fuzzy linear (left linear , right linear respectively) grammar G such that $L(G) = L.$

Remark 2.3.7 By the definition of fuzzy linear grammar, one can obviously guess that the class of fuzzy regular languages is contained in the class of fuzzy linear languages.

Theorem 2.3.8 Fuzzy left linear grammar and fuzzy right linear grammar generates the same language. (i.e. they are equivalent)

Proof Let $G=(V, \Sigma, S, P)$ is a fuzzy left linear grammar. Let us construct a fuzzy right linear grammar $G'=(V, \Sigma, S, P')$ with the fuzzy productions set P' as follows:

1. $q_0 \xrightarrow{\sigma} x$ in P' if and only if $q_0 \xrightarrow{\sigma} x$ in P, $S(q_0)>0$
2. $q_0 \xrightarrow{\sigma} xA$ in P', for $S(q_0)>0$ if and only if $A \xrightarrow{\sigma} x$ in P, $S(A)=0$
3. $A \xrightarrow{\sigma} x$ and $A \xrightarrow{\sigma} xq_0$ in P' if and only if $q_0 \xrightarrow{\sigma} Ax$ in P, $S(q_0)>0$
4. $A \xrightarrow{\sigma} xB$ in P' if and only if $B \xrightarrow{\sigma} Ax$ in P, $S(B)=0$

We prove that $L(G') = L(G)$

Let $x \in L(G)$, where $x = x_1 x_2 \ldots x_{n-1} x_n$; $x_i \in \Sigma^*$

Then, $\bigvee_{q \in V} \{S(q) \wedge d(q \Rightarrow x)\} > 0$

Thus, there exists $q_0 \in V$ such that $S(q_0) > 0$ *and* $d(q_0 \Rightarrow x) > 0$ in G

If $q_0 \xrightarrow{\sigma} x$ is a productions in P for some $\sigma \in (0,\ 1]$ then $q_0 \xrightarrow{\sigma} x$ is in P' and $x \in L(G')$.

Otherwise there exist $A_n,\ A_{n-1},\ \ldots,\ A_2 \in V$ and $\sigma_1, \sigma_2, \ldots, \sigma_n \in (0,\ 1]$ such that

$$q_0 \xrightarrow{\sigma_n} A_n x_n \xrightarrow{\sigma_{n-1}} A_{n-1} x_{n-1} x_n \xrightarrow{\sigma_{n-2}} \cdots \xrightarrow{\sigma_2} A_2 x_2 x_3 \ldots x_{n-1} x_n \xrightarrow{\sigma_1} x_1 x_2 \ldots x_{n-1} x_n = x$$

But then, corresponding to above derivation chain, P must have following fuzzy productions:

$q_0 \xrightarrow{\sigma_n} A_n x_n$, $A_n \xrightarrow{\sigma_{n-1}} A_{n-1} x_{n-1}$, ... , $A_3 \xrightarrow{\sigma_2} A_2 x_2$

and $A_2 \xrightarrow{\sigma_1} x_1$

Therefore, P' should have following fuzzy productions

$A_n \xrightarrow{\sigma_n} x_n$, $A_{n-1} \xrightarrow{\sigma_{n-1}} x_{n-1} A_n$, ... , $A_2 \xrightarrow{\sigma_2} x_2 A_3$,

$q_0 \xrightarrow{\sigma_1} x_1 A_2$

Thus, there is a derivation chain for x in G′:

$$q_0 \xrightarrow{\sigma_1} x_1 A_2 \xrightarrow{\sigma_2} x_1 x_2 A_3 \quad \ldots \quad \xrightarrow{\sigma_{n-2}} x_1 x_2 \ldots x_{n-2} A_{n-1}$$

$$\xrightarrow{\sigma_{n-1}} x_1 x_2 \ldots x_{n-1} A_n \xrightarrow{\sigma_n} x_1 x_2 \ldots x_{n-1} x_n = x$$

Therefore, there exist $q_0 \in V$ such that $S(q_0) > 0$

and $d(q_0 \Rightarrow x) > 0$

i.e. $\bigvee_{q \in V} \{S(q) \wedge d(q \Rightarrow x)\} > 0$

Hence, $x \in L(G')$

On the similar line one can prove the other part.

Theorem 2.3.9 Every fuzzy right linear language can be generated by a fuzzy grammar in normal form.

Proof We first construct a fuzzy regular grammar $G_1 = (V', \Sigma, S, P_1)$ such that $G' \square G_1$. Then construct a fuzzy grammar $G' = (V', \Sigma, S, P')$ in normal form from G_1 equivalent to G_1 proves the theorem.

Case 1. Let $G = (V, \Sigma, S, P)$ be given fuzzy right linear grammar.

Any fuzzy production $A \xrightarrow{\sigma} xB$ or $A \xrightarrow{\sigma} x$ of P with $|x| \leq 1$, is a fuzzy production in P_1, and for any fuzzy production $A \xrightarrow{\sigma} xB$ in P with $|x| > 1$, $x = a_1 a_2 \ldots a_{n-1} a_n$, we add in P_1 a set of fuzzy productions $A \xrightarrow{\sigma} a_1 Z_1$, $Z_1 \xrightarrow{\sigma} a_2 Z_2$, … , $Z_{n-1} \xrightarrow{\sigma} a_n B$, where $Z_1, Z_2, \ldots, Z_{n-1}$ are new variables not belonging to V. Similarly, for a fuzzy production $A \xrightarrow{\sigma} a_1 a_2 \ldots a_m$, m ≥ 2 and $\sigma > 0$, we add in P_1 a set of fuzzy productions $A \xrightarrow{\sigma} a_1 Y_1$, $Y_1 \xrightarrow{\sigma} a_2 Y_2$, … , $Y_{m-1} \xrightarrow{\sigma} a_m Y_m$, $Y_m \xrightarrow{\sigma} \Lambda$, where and $Y_1, Y_2, \ldots, Y_{m-1}, Y_m$ are new variables not belonging to V.

Let V' be the set of all variables in V and new variables added in the above process.

Thus, we have fuzzy grammar $G_1 = (V', \Sigma, S, P_1)$ with having following types of fuzzy productions:

1. $A \xrightarrow{\sigma} aB$
2. $A \xrightarrow{\sigma} B$

3. $A \xrightarrow{\sigma} \Lambda$, where $A, B \in V', \Lambda \in \Sigma^*, a \in \Sigma$ and $\sigma > 0.$

We now prove that $G \square G_1$.

Let $x \in L(G)$. Then $\bigvee\{S(q) \wedge d(q \Rightarrow x)\} > 0$

i.e. there exists

$q_0 \in V$ such that $S(q_0) > 0$ *and* $d(q_0 \Rightarrow x) > 0$

If $q_0 \xrightarrow{\sigma} x$ is a fuzzy production in P and $|x| = 1$,

then clearly $q_0 \xrightarrow{\sigma} x$ in P_1.

Now, if $|x| > 1$ and $x = a_1 a_2 \ldots a_{n-1} a_n$, $a_i \in \Sigma$, then

there exist $q_1, q_2, \ldots, q_{n-1} \in V'$ such that

$$q_0 \xrightarrow{\sigma} a_1 q_1 \xrightarrow{\sigma} a_1 a_2 q_2, \ \cdots \ , q_{n-1} \xrightarrow{\sigma} a_1 a_2 \cdots a_n = x$$

i.e. $q_0 \overset{\sigma}{\Longrightarrow} x$ in G_1

Therefore, there exists

$q_0 \in V'$ such that $S(q_0) > 0$ *and* $d(q_0 \Rightarrow x) > 0$ in G_1

Hence, $x \in L(G_1)$

Consider $x \in L(G_1)$. Then

$$\bigvee_{q \in V'} \{S(q) \wedge d(q \Rightarrow x)\} > 0$$

i.e. there exists

$q_0 \in V'$ such that $S(q_0) > 0$ and $d(q_0 \Rightarrow x) > 0$ in G_1

Now $d(q_0 \Rightarrow x) > 0$

Therefore, there is a derivation chain of x in G_1 as:

$$q_0 \xrightarrow{\sigma_1} a_1 A_1' \xrightarrow{\sigma_2} a_1 a_2 A_2' \xrightarrow{\sigma_3} \cdots \xrightarrow{\sigma_s} a_1 a_2 \cdots a_n A_n' \xrightarrow{\sigma_{s+1}} x_1 q_1$$

$$\xrightarrow{\delta_1} x_1 b_1 B_1' \xrightarrow{\delta_2} x_1 b_1 b_2 B_2' \xrightarrow{\delta_3} \cdots \xrightarrow{\delta_m} x_1 b_1 b_2 \cdots b_m B_m' \xrightarrow{\delta_{m+1}} x_1 x_2 q_2$$

$$\xrightarrow{\vartheta_1} \ldots \xrightarrow{\tau_{r+1}} x_1 x_2 \cdots x_{n-1} q_{n-1} \xrightarrow{\eta_1} x_1 x_2 \cdots x_{n-1} c_1 C_1' \xrightarrow{\eta_2} \ldots$$

$$\xrightarrow{\eta_n} x_1 x_2 \cdots x_n q_n \xrightarrow{\eta_{n+1}} x_1 x_2 \cdots x_n \Lambda = x$$

Thus, the productions

$$q_0 \xrightarrow{\sigma_1} a_1 A_1',\ A_1' \xrightarrow{\sigma_2} a_2 A_2',\ \cdots, A_{i-1}' \xrightarrow{\sigma_n} a_i A_i',\ A_{i+1}' \xrightarrow{\sigma_{i+1}} a_{i+1} A_{i+2}',\ \ldots, A_n' \xrightarrow{\sigma_{s+1}} q_1,$$

$$q_1 \xrightarrow{\delta_2} b_1 B_1', B_1' \xrightarrow{\delta_2} b_2 B_2',\ \cdots, B_m' \xrightarrow{\delta_{m+1}} q_2,\ \ldots, C_{n-1}' \xrightarrow{\eta_n} x_n q_n,\ q_n \xrightarrow{\eta_{n+1}} \Lambda$$

are in P_1

Thus, there is a derivation chain of x in G as:

$$q_0 \xrightarrow{\sigma} x_1 q_1 \xrightarrow{\delta} x_1 x_2 q_2 \xrightarrow{\vartheta} \cdots \xrightarrow{\eta} x_1 x_2 \cdots x_n q_n \xrightarrow{\eta_{n+1}} x_1 x_2 \cdots x_n \Lambda = x$$

, where

$$\sigma = \sigma_1 \wedge \sigma_2 \wedge \ldots \wedge \sigma_{s+1},\ \delta = \delta_1 \wedge \delta_2 \wedge \ldots \wedge \delta_{m+1},\ \vartheta = \vartheta_1 \wedge \vartheta_2 \wedge \ldots \wedge \vartheta_{r+1},\ \ldots, \eta = \eta_1 \wedge \eta_2 \wedge \ldots \wedge \eta_n$$

Thus there exists

$q_0 \in V$ such that $S(q_0) > 0$ and $d(q_0 \Rightarrow x) > 0$ in G

Therefore, $\bigvee_{q \in V} \{S(q) \wedge d(q \Rightarrow x)\} > 0$

Hence, $x \in L(G)$.

Thus,G is equivalent to G_1

Case 2. . We note that V' contains all variables in

V and new variables introduced in the above procedure for finding G_1. Fuzzy production like $A \xrightarrow{\sigma} B$ is called fuzzy chain rule.

We describe an algorithm to eliminate all such fuzzy chain rules as follows.

For this we construct the set $U_i(A) = \{A\}$, for $A \in V'$, and

$U_{i+1} = U_i(A) \cup \{B \mid B \xrightarrow{\sigma} Z \in P_1 \text{ for some } Z \in U_i(A), \sigma > 0\}$

Since V' is finite, there exist an integer K such that $U_{K+j}(A) = U_K(A)$, $j = 1, 2, \ldots$.

We denote this $U_K(A)$ by $U(A)$, for all $A \in V'$

We now construct the required fuzzy grammar $G' = (V', \Sigma, S, P')$.

1. $A \xrightarrow{\delta \wedge \sigma} aB$, in P' if and only if $\exists$ Z$\in$ V' such that $A \in U(Z)$ and $Z \xrightarrow{\sigma} aB$ in P_1, $\delta = d(A \Rightarrow Z)$ (Here, $\delta \wedge \sigma$ denotes the minimum of δ and σ)

2. $A \xrightarrow{\delta \wedge \sigma} \Lambda$, in P' if and only if $\exists$ Z$\in$V$'$ such that $A \in U(Z)$ and $Z \xrightarrow{\sigma} \Lambda$ in P_1, $\delta = d(A \Rightarrow Z)$

Clearly, the fuzzy grammar $G' = (V', \Sigma, S, P')$ is in the normal form. It is now easy to see that G_1 is equivalent to G'.

Example 2.3.10 Consider a fuzzy right linear grammar $G = (V, \Sigma, S, P)$, where $V = \{q_0, q_1, q_2, q_3\}$,

$S = \{0.2/q_0, 0.3/q_1\}$,

$\Sigma = \{0, 1\}$,

$P = \{q_0 \xrightarrow{0.1} 110q_1,\ q_1 \xrightarrow{0.7} 00q_2, q_1 \xrightarrow{0.3} 11q_3,$

$q_2 \xrightarrow{0.4} 01q_3, q_2 \xrightarrow{0.5} 00, q_1 \xrightarrow{0.4} q_2\}$ .This grammar generates string 11000 of degree 0.1, string 1100000 of degree 0.1, string 00 of degree 0.3 and string 0000 of degree 0.3.

Construct $G' = (V', \Sigma, S, P')$, where $V' = V \cup \{A_1, A_2, B_1, C_1, C_2, D_1, D_2\}$ and

$$P' = \{ q_0 \xrightarrow{0.1} 1A_1, A_1 \xrightarrow{0.1} A_2, A_2 \xrightarrow{0.1} 0q_1, A_2 \xrightarrow{0.1} 0q_1, q_2 \xrightarrow{0.4} 0q_3, C_1 \xrightarrow{0.3} 1q_3$$

$$q_1 \xrightarrow{0.4} 0q_3, q_1 \xrightarrow{0.5} 0D_1, C_2 \xrightarrow{1} 0, q_1 \xrightarrow{0.7} 0B_1, B_1 \xrightarrow{0.7} 0q_2, q_1 \xrightarrow{0.3} 1C_1$$
$$, q_2 \xrightarrow{0.5} 0D_1, D_1 \xrightarrow{0.5} 0D_2, D_2 \xrightarrow{0.5} \Lambda \}.$$

Then by above theorem $L(G') = L(G)$.

This enables us to write

Corollary 2.3.11 Fuzzy right linear grammar is equivalent to fuzzy regular grammar except for Λ.

We can summaries all these theorems into a theorem

Theorem 2.3.12 The followings are equivalent except for Λ.

(1) Fuzzy regular grammar

(2) Fuzzy left linear grammar

(3) Fuzzy right linear grammar

(4) Fuzzy grammar in normal form.

2.4 Reduction of Fuzzy Grammars

By a reduction of a fuzzy grammar, we mean an elimination of those variables that do not take part in the generation of the language. This is precisely the aim of the present section. We do this with the help of elimination of null and unit productions. Finally, we use these reduced grammars to find an equivalent Chomsky normal form for a given fuzzy grammar. This Chomsky normal form of fuzzy grammar will have an important role in compilation of computer languages.

Definition 2.4.1 Let $G = (V, \Sigma, S, P)$ be a fuzzy right linear grammar. A variable $A \in V$ is called *nullable variable*, if Λ derives empty string. A fuzzy production of the form $B \xrightarrow{\sigma} \Lambda$, if exists in P is called a Λ - *production*.

Algorithm 2.4.2 {To find nullable variables of the given fuzzy regular grammar G}

$N_0 = \{A \in V | P \text{ contains a fuzzy production } A \xrightarrow{\sigma} \Lambda, \ \sigma > 0\};$

$i = 0$

do

$i = \text{i} + 1$

$N_i = N_{i-1} \cup \{B \in V | P \text{ contains a fuzzy production } B \xrightarrow{\sigma} C, C \in N_{i-1}, \ \sigma > 0\};$

While

$N_i \neq N_{i-1}$

N_i is the set of nullable variables.

Algorithm 2. 4.3 [To find an equivalent fuzzy right linear grammar with no $\wedge$ production]

Given a fuzzy right linear grammar $G = (V, \Sigma, S, P)$. Construct a fuzzy right linear grammar $G_1 = (V, \Sigma, S, P_1)$ with no $\wedge$ production as follows:

1. Initialize P_1 to P
2. Find all nullable variables in V using Algorithm 2.3.2.
3. For every production $A \xrightarrow{\sigma} yB$ such that B is nullable in P and $\sigma > 0$, add to P_1 the set of fuzzy productions

   $\{A \xrightarrow{\sigma} y\} \cup \{A \xrightarrow{\sigma \wedge d} yxC \mid B \xrightarrow{d} xC, x \in \Sigma^*, d > 0\}.$

Delete all null productions and repeated ones from P_1

Theorem 2.4.4 For a fuzzy right linear grammar G, there is an equivalent fuzzy right linear grammar G_1 without Λ productions except for the empty string.

Proof: Let $G = (V, \Sigma, S, P)$ be a fuzzy right linear grammar.

We construct a fuzzy right linear grammar $G_1 = (V_1, \Sigma, S, P_1)$, where V_1 be the set of non-nullable variables in G

$A \xrightarrow{\sigma} x \in P_1$ if $A \xrightarrow{\sigma} x \in P$

$A \xrightarrow{\sigma} xB \in P_1$ if $A \xrightarrow{\sigma} xB \in P$ and B is not nullable.

$A \xrightarrow{\sigma} x \in P_1$ if $A \xrightarrow{\sigma} xB \in P$ and B is nullable.

$A \xrightarrow{\sigma} x \in P_1$ and $A \xrightarrow{\sigma \wedge \delta} xyC \in P_1$ if $A \xrightarrow{\sigma} xB,\ B \xrightarrow{\delta} yC \in P,\ C \in V \cup \{\Lambda\}$ and B is nullable

Assume that P_1 has no Λ productions and repetitions.

Let $x \in L(G)$ and $x \neq \Lambda$

Then, $\bigvee_{q \in V} \{S(q) \wedge d(q \Rightarrow x)\} > 0$

i.e. there exists $q_0 \in V$ such that $S(q_0) \wedge d(q_0 \Rightarrow x) > 0$

We use the notation $q_0 \overset{\sigma}{\underset{G}{\Rightarrow}}{}^n x$ to mean that there is a n-step derivation of x of degree σ from q_0 in G

We claim that , if $q_0 \overset{\sigma}{\underset{G}{\Rightarrow}}{}^n x$, then $q_0 \overset{\sigma}{\underset{G_1}{\Rightarrow}}{}^* x$.

For this, if $q_0 \overset{\sigma}{\underset{G}{\Rightarrow}}{}^1 x$, then $q_0 \xrightarrow{\sigma} x$ must be a fuzzy production in P and hence a fuzzy production in P_1, since $x \neq \Lambda$. Thus, $q_0 \overset{\sigma}{\underset{G_1}{\Rightarrow}}{}^* x$.

Let $q_0 \overset{\sigma}{\underset{G}{\Rightarrow}}{}^{n+1} x$. Suppose the $(n+1)-\text{step}$ derivation of x in G is as follows:

$$q_0 \xrightarrow{\sigma_1} x_1q_1 \xrightarrow{\sigma_2} x_1x_2q_2 \xrightarrow{\sigma_3} \ldots \xrightarrow{\sigma_n} x_1x_2\ldots x_nq_n \xrightarrow{\sigma_{n+1}} x_1x_2\ldots x_nx_{n+1} = x$$

$$\text{where } \sigma_i \in (0, 1] \qquad \text{.......(i)}$$

Corresponding to this derivation chain of x, P must have following fuzzy productions

$$q_0 \xrightarrow{\sigma_1} x_1q_1 \text{ , } q_1 \xrightarrow{\sigma_2} x_2q_2 \text{ , } \ldots, \; q_{n-1} \xrightarrow{\sigma_n} x_nq_n, \;\; q_n \xrightarrow{\sigma_{n+1}} x_{n+1}$$

Then either no q_i is nullable or some of the q_i's are nullable.

If no q_i is nullable in G, then the above chain is the required derivation chain for x in G_1

Suppose q_i is a null variable for some i, then corresponding to fuzzy production $q_{i-1} \xrightarrow{\sigma_i} x_i q_i$ in P , we have following fuzzy productions in P_1

$q_{i-1} \xrightarrow{\sigma_i} x_i$ and $q_{i-1} \xrightarrow{\sigma_i \wedge \sigma_{i+1}} x_i x_{i+1} q_{i+1}$

Thus, the derivation chain for x in G_1 takes the following form:

$$q_0 \xrightarrow{\sigma_1} x_1 q_1 \xrightarrow{\sigma_2} x_1 x_2 q_2 \xrightarrow{\sigma_3} \ldots \xrightarrow{\sigma_{i-1}} x_1 x_2 \ldots x_{i-1} q_{i-1} \xrightarrow[p']{\sigma_i \wedge \sigma_{i+1}} x_1 x_2 \ldots x_{i-1} x_i x_{i+1} q_{i+1}$$

$$\xrightarrow{\sigma_{i+2}} x_1 x_2 \ldots x_{i+1} x_{i+2} q_{i+2} \xrightarrow{\sigma_{i+3}} \ldots \xrightarrow{\sigma_n} x_1 x_2 \ldots x_n q_n \xrightarrow{\sigma_{n+1}} x_1 x_2 \ldots x_n x_{n+1} = x$$

Therefore , $q_0 \overset{\sigma}{\underset{G_1}{\Rightarrow}}{}^* x$

Thus, there exists $q_0 \in V_1$ such that $S(q_0) > 0$ and $q_0 \underset{G_1}{\Rightarrow}{}^* x$

i.e. $\bigvee_{q \in V_1} \{S(q) \wedge d(q \Rightarrow x)\} > 0$

Hence, $x \in L(G_1)$. The converse can be proved similarly.

Definition 2.4.5 For a variable A of a fuzzy regular grammar $G = (V, \Sigma, S, P)$, A*-derivable variable* is recursively defined as follows:

(1) If $A \xrightarrow{\sigma} B$ is a fuzzy production, then B is called A-derivable

(2) If C is A-derivable and $C \xrightarrow{\sigma} B$ is a fuzzy production and $B \neq A$, then B is

A-derivable

(3) No other variable in V is A- derivable.

A fuzzy production of the form

$A \xrightarrow{\sigma} B,\ \mathrm{A, B} \in \mathrm{V}$ is called an unit production

Algorithm 2.4.6 To Find A-derivable variables

Let $G = (V, \Sigma, S, P)$ be a fuzzy regular grammar.

$N_0 = \{B \in V | \, P \text{ contains a fuzzy production } A \xrightarrow{\sigma} B,\ \sigma > 0\};$

$i = 0$

do

$i = \mathrm{i} + 1$

$N_i = N_{i-1} \cup \{C \in V \mid P \text{ contains a fuzzy production } B \xrightarrow{\sigma} C, \text{ for some } \mathrm{B} \in \mathrm{N}_{i-1}, \ \sigma > 0\};$

While

$N_i \neq N_{i-1}$

N_i is the set of A-derivable variables.

Algorithm 2.4.7 To find an equivalent fuzzy regular grammar with non unit production

Given a fuzzy regular grammar $G = (V, \ \Sigma, \ S, \ P)$, we construct a fuzzy regular grammar $G_1 = (V, \ \Sigma, \ S, \ P_1)$ as follows

1. Initialize P_1 to P
2. For each $A \in V$, Find the set of A - derivable variables using Find A- derivable Algorithm

3. For every pair (A, B) such that B is A-derivable and every non unit fuzzy production $B \xrightarrow{\sigma} \alpha$, add the production $A \xrightarrow{\sigma} \alpha$ to P_1, if not already present in P_1.

Delete all unit productions from P_1

Theorem 2.4.8 For a fuzzy right linear grammar G without Λ- productions, there is fuzzy right linear grammar G_1 with no unit productions such that $L(G_1) = L(G)$

Proof Let V_1 be the set of variables which is not A- derivable

Construct fuzzy right linear grammar $G_1 = (V_1, \Sigma, S, P_1)$, where

$$P_1 = P \cup \{A \xrightarrow{\sigma \wedge \delta} \alpha \in P_1 \mid B \xrightarrow{\sigma} \alpha \in P \text{ is a non unit fuzzy production and } A \xrightarrow{\delta} B \in P\}$$

We assume that P_1 has no repetitions.

Let $x \in L(G)$, where $x = x_1 x_2 \ldots x_{n+1}$, $x_i \in \Sigma^*$

Then, $\bigvee_{q \in V} \{S(q) \wedge d(q \Rightarrow x)\} > 0$

Thus, there exists $q_0 \in V$ such that $S(q_0) \wedge d(q_0 \Rightarrow x) > 0$

We claim that $q_0 \Rightarrow^n x$ in G iff $q_0 \Rightarrow^* x$ in G_1

If $q_0 \overset{\sigma}{\underset{G}{\Rightarrow}}{}^1 x$, then $q_0 \xrightarrow{\sigma} x$ must be a fuzzy a production in P and hence $q_0 \xrightarrow{\sigma} x$ a fuzzy production in P_1. Thus, $q_0 \overset{\sigma}{\underset{G_1}{\Rightarrow}}{}^1 x$

Let $q_0 \overset{\sigma}{\underset{G}{\Rightarrow}}{}^{n+1} x$. Suppose that the (n+1)th step derivation of x in G is as follows:

$$q_0 \xrightarrow{\sigma_1} x_1 q_1 \xrightarrow{\sigma_2} x_1 x_2 q_2 \xrightarrow{\sigma_3} \ldots \xrightarrow{\sigma_n} x_1 x_2 \ldots x_n q_n \xrightarrow{\sigma_{n+1}} x_1 x_2 \ldots x_n x_{n+1} = x,$$

where

$\sigma_i \in (0,\ 1]$

If no q_i is $q_{i-1}-$ derivable variable in the above chain, then the above chain is the required derivation chain for x in G_1

If some q_i is $q_{i-1}-$ derivable of degree $\delta \in (0,\ 1]$ in the above chain, then P must have following fuzzy productions

$$q_0 \xrightarrow{\sigma_1} x_1 q_1\ , q_1 \xrightarrow{\sigma_2} x_2 q_2\ , \ldots, q_{i-2} \xrightarrow{\sigma_{i-1}} x_{i-1} q_{i-1}, q_{i-1} \xrightarrow{\sigma_i} q_i,$$

$$q_i \xrightarrow{\sigma_{i+1}} x_i x_{i+1} q_{i+1},\ q_{i+1} \xrightarrow{\sigma_{i+2}} x_{i+2} q_{i+2}, \ldots,\ q_n \xrightarrow{\sigma_{n+1}} x_{n+1}$$

But then P_1 must contains following type of fuzzy productions

$$q_0 \xrightarrow{\sigma_1} x_1 q_1\ , q_1 \xrightarrow{\sigma_2} x_2 q_2\ , \ldots, q_{i-2} \xrightarrow{\sigma_{i-1}} x_{i-1} q_{i-1}, q_{i-1} \xrightarrow{\delta \wedge \sigma_{i+1}} x_i x_{i+1} q_{i+1},$$

$$q_{i+1} \xrightarrow{\sigma_{i+2}} x_{i+2} q_{i+2}, \ldots,\ q_n \xrightarrow{\sigma_{n+1}} x_{n+1}$$

Thus, the derivation chain for x in G_1 takes the following form:

$$q_0 \xrightarrow{\sigma_1} x_1 q_1 \xrightarrow{\sigma_2} x_1 x_2 q_2 \xrightarrow{\sigma_3} \ldots \xrightarrow{\sigma_{i-1}} x_1 x_2 \ldots x_{i-1} q_{i-1} \xrightarrow[p']{\delta \wedge \sigma_{i+1}} x_1 x_2 \ldots x_i x_{i+1} q_{i+1}$$

$$\xrightarrow{\sigma_{i+2}} x_1 x_2 \ldots x_{i+1} x_{i+2} q_{i+2} \xrightarrow{\sigma_{i+3}} \cdots \xrightarrow{\sigma_{n+1}} x_1 x_2 \ldots x_n x_{n+1} = x$$

Therefore, $q_0 \Rightarrow^* x$ in G_1

Thus, there exists $q_0 \in V_1$ such that $S(q_0) > 0$ and

$q_0 \underset{G_1}{\Rightarrow^*} x$

i.e. $\bigvee_{q \in V_1} \{S(q) \wedge d(q \Rightarrow x)\} > 0$

Hence, $x \in L(G_1)$.

Conversely, let $x \in L(G_1)$

Thus, $\bigvee_{q \in V_1} \{S(q) \wedge d(q \Rightarrow x)\} > 0$

i.e. there exists $q_0 \in V_1$ such that $S(q_0) > 0$ and

$d(q_0 \Rightarrow x) > 0$

We now claim that $q_0 \Rightarrow^n x$ in G_1 iff $q_0 \Rightarrow^* x$ in G

Suppose $q_0 \overset{\sigma}{\Rightarrow}^1 x$, then $q_0 \xrightarrow{\sigma} x$ must be a fuzzy a production in P_1. Then P have following types of fuzzy productions

(1) $q_0 \xrightarrow{\sigma} x$ or

(2)

$$q_0 \xrightarrow{\delta_1} x_1 q_1,\ q_1 \xrightarrow{\delta_2} x_2 q_2,\ \ldots,\ q_{n-1} \xrightarrow{\delta_n} x_n q_n,\ q_n \xrightarrow{\delta_1} x, \text{ for some } n \geq 1 \text{ and } \delta_1 \wedge \delta_2 \wedge \ldots \delta_{n+1} = \sigma$$

But in either the case $q_0 \Rightarrow^* x$ in G

Let $q_0 \Rightarrow^{n+1} x$. Suppose the $(n+1)-\text{step}$ derivation of x in G_1 is as follows

$$q_0 \xrightarrow{\sigma_1} x_1 q_1 \xrightarrow{\sigma_2} x_1 x_2 q_2 \xrightarrow{\sigma_3} \ldots q_{i-1} \xrightarrow[p_1]{\sigma_i \wedge \sigma_{i+1}} x_1 x_2 \ldots x_{i-1} x_i q_{i+1}$$

$$\xrightarrow{\sigma_{i+2}} x_1 x_2 \ldots x_i x_{i+1} q_{i+2} \xrightarrow{\sigma_{i+3}} \ldots \xrightarrow{\sigma_{n+1}} x_1 x_2 \ldots x_n q_{n+1} \xrightarrow{\sigma_{n+2}} x_1 x_2 \ldots x_n x_{n+1} = x$$

Then P_1 has following types of fuzzy productions

$$q_0 \xrightarrow{\sigma_1} x_1 q_1,\ q_1 \xrightarrow{\sigma_2} x_2 q_2,\ \ldots,\ q_{i-2} \xrightarrow{\sigma_{i-1}} x_{i-1} q_{i-1},\ q_{i-1} \xrightarrow[p_1]{\sigma_i \wedge \sigma_{i+1}} x_i x_{i+1} q_{i+1}$$

$$q_{i+1} \xrightarrow{\sigma_{i+2}} x_{i+2} q_{i+2},\ \ldots,\ q_n \xrightarrow{\sigma_{n+1}} x_{n+1}$$

For every fuzzy production $q_i \xrightarrow{\sigma_{i+1}} x_{i+1}q_{i+1}$ in P_1, we have following types of fuzzy productions in P

(1) $q_i \xrightarrow{\sigma_{i+1}} x_{i+1}q_{i+1}$

(2) $q_i \xrightarrow{\alpha_1} p_1, p_1 \xrightarrow{\alpha_2} p_2, \ldots, p_{m-1} \xrightarrow{\alpha_m} p_m$ and $p_m \xrightarrow{\alpha_{m+1}} x_{i+1}q_{i+1}$, for some $m \geq 1$ and

$\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_{m+1} = \sigma_{i+1}$

But in either the case $q_0 \overset{\sigma_{i+1}}{\Rightarrow^*} x_{i+1}q_{i+1}$ in G

Therefore, the derivation chain for x in G will be as follows

$$q_0 \xrightarrow{\sigma_1} x_1q_1 \xrightarrow{\sigma_2} x_1x_2q_2 \xrightarrow{\sigma_3} \ldots \xrightarrow{\sigma_n} x_1x_2\ldots x_nq_n \xrightarrow{\sigma_{n+1}} x_1x_2\ldots x_nx_{n+1} = x$$

i.e. $q_0 \Rightarrow^* x$ in G

Therefore $q_0 \Rightarrow^* x$ in G

Thus, there exists $q_0 \in V$ such that $S(q_0) > 0$ and $d(q_0 \Rightarrow x) > 0$

i.e. $\bigvee_{q \in V} \{S(q) \wedge d(q \Rightarrow x)\} > 0$

Hence, $x \in L(G)$

Definition 2.4.9 A fuzzy grammar G is said to be in Chomsky normal form, if every production is of the form $A \xrightarrow{\sigma} a$ or $A \xrightarrow{\sigma} AB$, $\forall\, A,\, B \in V,\ a \in \Sigma$ and $\sigma \in (0,\ 1]$

Theorem 2.4.10 For fuzzy right linear grammar G, there is a fuzzy grammar G_1 in Chomsky normal form such that $L(G_1) = L(G) - \{\Lambda\}$

Proof Let $G = (V,\ \Sigma,\ S,\ P)$ be any fuzzy right linear grammar.

By using Theorems 2.3.4 and 2.3.8, one can obtain a fuzzy right linear grammar say $G' = (V',\ \Sigma,\ S,\ P')$, without any $\Lambda -$production and any unit production.

We now construct a fuzzy grammar $G_1 = (V_1, \Sigma, S, P_1)$

If $A \xrightarrow{\sigma} a_1 a_2 ... a_n B$, $n \geq 2$ is a fuzzy production in P', then add in P_1 a set of fuzzy productions

$A \xrightarrow{\sigma} C_1 D_1,\ D_1 \xrightarrow{\sigma} C_2 D_2,\ ...,\ D_{n-2} \xrightarrow{\sigma} C_{n-1} D_{n-1},\ D_{n-1} \xrightarrow{\sigma} C_n B, C_1 \xrightarrow{1} a_1,$

$C_2 \xrightarrow{1} a_2,\ ...,\ C_n \xrightarrow{1} a_n$, where C_i's and D_j's are new variables not belongs to V′.

Similarly, If $A \xrightarrow{\sigma} a_1 a_2 ... a_n$, $n \geq 2$ is a fuzzy production in P', then we replace it by the set of fuzzy productions

$A \xrightarrow{\sigma} Y_1 Z_1,\ Z_1 \xrightarrow{\sigma} Y_2 Z_2,\ ...,\ Z_{n-2} \xrightarrow{\sigma} Y_{n-1} Z_{n-1},\ Y_1 \xrightarrow{1} a_1,\ Y_2 \xrightarrow{1} a_2$

$,\ ...,\ Y_{n-1} \xrightarrow{1} a_{n-1},\ Z_{n-1} \xrightarrow{1} a_n$ in P_1, where Y_i's and Z_j's are new variables.

And all other fuzzy productions in P' are taken as it is in P_1

We claim that G_1 is the required grammar

Let $x \in L(G')$, $x = a_1 a_2 \dots a_n$, $a_i \in \Sigma$

Then, $\bigvee_{q \in V'} \{S(q) \wedge d(q \Rightarrow x)\} > 0$

i.e. there exists $q_0 \in V'$ such that $S(q_0) \wedge d(q_0 \Rightarrow x) > 0$

We set a claim, if $q_0 \Rightarrow^n x$ in G', then $q_0 \Rightarrow^* x$ in G_1

If $q_0 \Rightarrow^1 x$ in G', then $q_0 \xrightarrow{\sigma} x$ must be a fuzzy production in P'. But then there is a set of fuzzy productions in P_1

$$A \xrightarrow{\sigma} Y_1 Z_1,\ Z_1 \xrightarrow{\sigma} Y_2 Z_2,\ \dots ,\ Z_{m-2} \xrightarrow{\sigma} Y_{m-1} Z_{m-1},$$

$$Y_1 \xrightarrow{1} a_1,\ Y_2 \xrightarrow{1} a_2,,\ \dots ,\ Y_{m-1} \xrightarrow{1} a_{n-1},$$

$$Z_{m-1} \xrightarrow{1} a_n,\ \text{for some } m \geq 1$$

This proves that, $q_0 \Rightarrow x$ in G_1 also

Let $q_0 \Rightarrow^n x$, for $n \geq 2$. Suppose the n^{th} step derivation of x in G' is

$$q_0 \xrightarrow{\sigma_1} x_1 q_1 \xrightarrow{\sigma_2} x_1 x_2 q_2 \xrightarrow{\sigma_3} \ldots \xrightarrow{\sigma_{n\text{-}1}} x_1 x_2 \ldots x_{n-1} q_{n-1} \xrightarrow{\sigma_n} x_1 x_2 \ldots x_n = x$$

Corresponding to this derivation chain of x P' must have following fuzzy productions

$$q_0 \xrightarrow{\sigma_1} x_1 q_1,\ q_1 \xrightarrow{\sigma_2} x_2 q_2,\ \ldots,\ q_{n-2} \xrightarrow{\sigma_{n\text{-}1}} x_{n-1} q_{n-1},\ q_{n-1} \xrightarrow{\sigma_n} x_n$$

By definition, to each fuzzy production $q_i \xrightarrow{\sigma} x_{i+1} q_{i+1}$ in P', we have the following set fuzzy productions in P_1

$$q_0 \xrightarrow{\sigma_1} X_{a_{i+1,1}} q_1,\ X_{a_{i+1,1}} \xrightarrow{1} a_{i+1,1},\ q_1 \xrightarrow{\sigma_2} X_{a_{i+1,2}} q_2,\ X_{a_{i+1,2}} \xrightarrow{1} a_{i+1,2},\ \ldots,$$

$$q_{n-2} \xrightarrow{\sigma_{n\text{-}1}} X_{a_{i+1,n-1}} q_n,\ X_{a_{i+1,n-1}} \xrightarrow{1} a_{i+1,n-1},\ q_n \xrightarrow{\sigma_n} a_{i+1,n}$$

Clearly, $q_0 \Rightarrow^* x$ in G_1

Therefore, by induction $q_0 \underset{G_1}{\Rightarrow^*} x$

Thus, there exists $q_0 \in V_1$ such that $S(q_0) > 0$ and $d(q_0 \Rightarrow x) > 0$

i.e. $\bigvee_{q \in V_1} \{S(q) \wedge d(q \Rightarrow x)\} > 0$

Hence, $x \in L(G_1)$. The converse can be traced back easily.

Example 2.4.11 Consider a fuzzy right linear grammar $G = (V, \Sigma, S, P)$ with $V = \{q_0, q_1, q_2\}$, $S = \{1/q_0\}$, $\Sigma = \{0, 1\}$,

$p = \left\{q_0 \xrightarrow{0.3} 01q_2, q_0 \xrightarrow{0.4} 00q_1, q_1 \xrightarrow{0.5} 11q_2, q_2 \xrightarrow{1.0} 10q_2, q_2 \xrightarrow{0.9} 0\right\}$. Then

$L(G) = \left((01)^+ 0011(10)^* 0\right) + \left(0011(10)^* 0\right)$ and every strings of $(01)^+ 0011(10)^* 0$ are of degree 0.3 and strings of $0011(10)^* 0$ are of degree 0.4

Construct $G_1 = (V', \Sigma, S, P')$, where $V' = V \cup \{C_1, D_1, C_2, D_2, C_3, D_3, C_4, D_4\}$ and P' contains following fuzzy productions

$q_0 \xrightarrow{0.3} C_1D_1, C_1 \xrightarrow{1} 0$,

$D_1 \xrightarrow{0.3} 1q_0, q_0 \xrightarrow{0.4} C_2D_2, C_2 \xrightarrow{1} 0, D_2 \xrightarrow{0.4} 0q_1$

$q_0 \xrightarrow{0.5} C_3D_3, C_3 \xrightarrow{1} 1$,

$D_3 \xrightarrow{0.5} 1q_2, q_2 \xrightarrow{1} C_4D_4, C_4 \xrightarrow{1} 1, D_4 \xrightarrow{1} 0q_2$,

$q_2 \xrightarrow{0.9} 0$.

One can verify that G_1 is in Chomsky normal form and $L(G_1) = L(G)$.

Corollary 2.4.12 For fuzzy regular grammar G, there is a fuzzy grammar G_1 in Chomsky normal form such that $L(G_1) = L(G)$.

We conclude this chapter by proving the pumping lemma for fuzzy regular languages, by the use of fuzzy regular grammar. We show that the pumping lemma is useful mainly for showing that certain fuzzy languages are not regular.

Theorem 2.4.13 Let L be a fuzzy regular language. Then there is a positive integer n such that if x is any word in L with $|x| \geq n$, then $x = uvw$, $|\mathrm{uv}| \leq n,\ |v| \geq 1$ and $\mathrm{uv}^m w \in L,$ for any $m \geq 0$.

Proof Let $G = (V,\ \Sigma,\ S,\ P)$ be a fuzzy regular grammar having n variables such that $L(G) = L.$

Let $x \in L$ such that $|x| \geq n.$ Suppose $x = a_1 a_2 \ldots a_p,\ a_i \in \Sigma.$

Then, $\bigvee_{q \in Q} \{S(q) \wedge d(q \Rightarrow x)\} > 0.$

This implies there exists $q_0 \in V$ such that $S(q_0) > 0$

and $d(q_0 \Rightarrow x) > 0.$

This implies there exists $q_0, q_1, \cdots, q_{p-1} \in V$ such that

$S(q_0) > 0$ and $q_0 \xrightarrow{\sigma_1} a_1 q_1,$

$q_1 \xrightarrow{\sigma_2} a_2 q_2, \cdots, q_{p-1} \xrightarrow{\sigma_n} a_n,$ where

$\sigma_i \in (0,\ 1],\ 1 \le i \le p.$

Since $|V| = n,$ there exists $i \ne j$ such that $0 \le i < j \le n$

and $q_i = q_j$. Let $u = a_1\ a_2\ \cdots\ a_i,$

$v = a_{i+1}\ a_{i+2}\ \cdots\ a_j\ ,\ w = a_{j+1}\ a_{j+2}\ \cdots\ a_p.$ Then,

$x = uvw,\ |\mathrm{uv}| \le n$ and $|v| \ge 1.$

Let $q_0 \overset{\rho_1}{\Rightarrow} uq_i\ ,\ q_i \overset{\rho_2}{\Rightarrow} vq_j$ and $q_j \overset{\rho_3}{\Rightarrow} w.$

Since, $q_i \Rightarrow vq_i \Rightarrow vvq_i$ and $q_i \overset{\rho_2}{\Rightarrow} vq_i,$ we have ,

$d\left(q_i \Rightarrow v^2 q_i\right) \ge \rho_2.$

Let, if , possible $d\left(q_i \Rightarrow v^2 q_i\right) > \rho_2.$

Then, $d(q_i \Rightarrow vq_r) \wedge d(q_r \Rightarrow vq_i) \geq \rho_2$, for some $q_r \in V$.

But then, $q_r \in \{q_{i+1}, q_{i+2}, \cdots, q_j = q_i\}$, since $v = a_{i+1}a_{i+2}\cdots a_j$

Therefore, $d(q_i \Rightarrow vq_i) \geq d(q_i \Rightarrow vq_r) > \rho_2$, which is a contradiction to $d(q_i \Rightarrow vq_j) = \rho_2$.

Hence, $d(q_i \Rightarrow v^2 q_i) = \rho_2$.

Inductively one can prove that $d(q_i \Rightarrow v^m q_i) = \rho_2$,, for all $m \geq 0$.

Hence,

$$d(q_0 \Rightarrow uv^m w) \geq d(q_0 \Rightarrow uq_i) \wedge d(q_i \Rightarrow v^m q_j) \wedge d(q_j \Rightarrow w) > 0.$$

Therefore, $S(q_0) > 0$ and $d(q_0 \Rightarrow uv^m w) > 0$.

Hence, $uv^m w \in L$

We now use pumping lemma to show that the given language is not regular.

Example 2.4.14

$L = \{0^r \mid r \text{ is a perfect square of an integer } \geq 1\}$ is not a fuzzy regular language.

Solution: Suppose L is fuzzy regular language. Then there exists a fuzzy regular grammar $G = (V, \Sigma, S, P)$ such that $L = L(G)$. Let $|V| = n$ and Let $x = 0^{n^2}$, then $|x| = n^2 > n.$

By Pumping Lemma, $x = uvw$, where $1 \leq |v| < n$, $|\mathrm{uv}| \leq n$ and $uv^i w \in L(G)$, for all $i \geq 1$. For $i = 2$, we have $n^2 = |x| < |uv^2 w| = |uvw| + |v| \leq n^2 + n < (n+1)^2$

i.e. $n^2 < |uv^2 w| < (n+1)^2$

Hence, $uv^2 w \notin L = L(G)$

Therefore, L is not regular.

Chapter 3

Fuzzy Automaton Languages

3.1 Introduction:

Lee and Zadeh [38] discussed fuzzy language generalized by fuzzy grammar which is an extension of crisp language generated by (fuzzy) grammar. In this chapter we first discuss few closure properties of fuzzy automaton languages and then introduce fuzzy languages generated by fuzzy automata with fuzzy initial states. To establish closure properties of homomorphic and an inverse homomorphic images of fuzzy language generated by fuzzy automaton, we use extension principle. Further, we point out that a class of fuzzy languages generated by fuzzy automata is not closed with respect to complementation. At the end, we introduce an equivalent fuzzy regular grammar for a given fuzzy automaton with fuzzy initial states.

3.2 Fuzzy Automaton Crisp Languages and their Closure Properties

In this section, we discuss fuzzy automaton languages and discuss their closure properties with respect to union, intersection, reversal, homomorphic and inverse homomorphic image.

Theorem 3.2.1 [33] Let Σ be any non-empty set. Then

(a) $\{x\},\ x \in \Sigma^*$ (b) ϕ (c) Σ and (d) Σ^* are fuzzy automaton languages.

Theorem 3.2.2 [34, 42] The class of fuzzy automaton languages is closed under union, intersection, complement, kleen closure and product.

We now establish closure properties for languages namely homomorphic and inverse homomorphic image properties with the help of extension principle.

Definition 3.2.3 An onto mapping $f:\Sigma \to \Delta^*$ is called a *homomorphism*, if $f(ab) = f(a)f(b)$, $\forall a,\ b \in \Sigma$.

The homomorphism *f* is extended to $f:\Sigma^* \to \Delta^*$ as $f(\wedge) = \wedge,$ $f(\sigma) = \sigma,$ and $f(\sigma x) = f(\sigma)f(x),$ for all $\sigma \in \Sigma$ and $x \in \Sigma^*$.

Thus, the following theorem is immediate by induction on the length of y, i.e. $|y|$.

Theorem. 3.2.4 If $f:\Sigma \to \Delta^*$ is a homomorphism, then for any $x, y \in \Sigma^*$, we have $f(xy) = f(x)f(y).$

It can be noted that, for each $y \in \Delta^*, \exists x \in \Sigma^*$ such that $f(x) = y.$

Theorem. 3.2.5 The class of fuzzy automaton languages is closed under homomorphism and inverse homomorphism.

Proof Let $f:\Sigma \to \Delta^*$ be a homomorphism and let $M = (Q, \Sigma, \mu, I, F)$ be a fuzzy finite automaton such that $L(M) = L$.

We construct a fuzzy finite automaton $M' = (Q, \Delta, \mu', I, F)$, where $\mu' : Q \times \Delta \times Q \to [0, 1]$

is defined by $\mu'(q_0, \alpha, q) = \sigma$, whenever $\exists\ x \in \Sigma^*$

such that $f(x) = \alpha$ and $\mu(q_0, x, q) = \sigma > 0.$

Let

$\alpha \in L(M')$. Then $I(q_0) \wedge \mu'(q_0, \alpha, q) \wedge F(q) > 0$, for some $q_0, q \in Q$

Thus, $\exists\, x \in \Sigma^*$ such that $f(x) = \alpha$ and $\mu(q_0, x, q) = \mu'(q_0, \alpha, q)$

i.e. $\exists\, x \in \Sigma^*$ such that $f(x) = \alpha$ and $I(q_0) \wedge \mu(q_0, x, q) \wedge F(q) > 0$

Hence, $x \in L(M) = L$

Therefore, $f(x) \in f(L)$

i.e $\alpha \in f(L)$

Conversely, if $\alpha \in f(L)$, then $\exists\, x \in L$ such that

$\alpha = f(x)$ and

$I(q_0) \wedge \mu(q_0, x, q) \wedge \mathrm{F(q)} > 0$

Since $\mu'(q_0, \alpha, q) = \mu(q_0, x, q)$, we have $I(q_0) \wedge \mu'(q_0, \alpha, q) \wedge F(q) > 0.$

Thus, $\alpha \in L(M')$

Therefore, $f(L)$ is a fuzzy automaton language.

We now prove that the class of fuzzy automaton languages is closed under inverse homomorphism.

For closure property under inverse homomorphism, let $M = (Q, \Delta, \mu, I, F)$ be a fuzzy finite automaton such that $L(M) = L.$

Then $M' = (Q, \Sigma, \mu', I, F)$ is a fuzzy finite automaton, where $\mu': Q \times \Sigma^* \times Q \to [0, 1]$ is defined by $\mu'(q_0, x, q) = \mu(q_0, f(x), q),$ for all $q_0, q \in Q$ all $x \in \Sigma^*$.

Now,

$$x \in L(M') \Leftrightarrow I(q_0) \wedge \mu'(q_0, x, q) \wedge F(q) > 0$$

$$\Leftrightarrow I(q_0) \wedge \mu(q_0, f(x), q) \wedge F(q) > 0$$

$$\Leftrightarrow f(x) \in L(M) = L$$

$$\Leftrightarrow x \in f^{-1}(L)$$

This proves that $f^{-1}(L)$ is also a fuzzy automaton language.

Theorem 3.2.6 The class of fuzzy automaton languages is closed under the quotient with arbitrary set.

Proof Let $L_1 \subseteq \Sigma^*$ be a fuzzy automaton language and $L_2 \subseteq \Sigma^*$ be any set. Let $M = (Q, \Sigma, \mu, I, F)$ be a fuzzy finite automaton such that $L(M) = L_1$. Construct $M' = (Q, \Sigma, \mu, I, F')$, where $F' : Q \to [0, 1]$ is defined by

$$F'(q) = \bigvee_{y \in L_2} \{\, \mu(\, q,\, y,\, q'\,) \wedge F(\, q'\,)\,\}.$$

If $x \in L(M')$, then there exist $q_0, q \in Q$ such that

$I(q_0) \wedge \mu(q_0, x, q) \wedge F'(q) > 0$

Thus, there is $y \in L_2$ such that $\mu(q,\, y,\, q') \wedge F(q'\,) > 0,$

since $F'(q) > 0.$

Hence, $I(q_0) \wedge\ \mu(q_0,\, x,\, q) \wedge\ \mu(q,\, y,\, q') \wedge\ F(q') >\ 0$,

for some $y \in L_2$

i.e.

$I(q_0) \wedge\ \ \mu(q_0,\, xy,\, q') \wedge\ F(q') >\ 0$, for some $q_0,\ q \in Q$ and $y \in L_2$

Thus, $xy \in L(M) =\ L_1$, for some $y \in L_2$

Hence, $x \in\ L_1\ L_2^{-1}.$

Converse is similar in nature.

3.3 Fuzzy Languages of Fuzzy Automaton

We recall that, fuzzy language L over the alphabet Σ is a fuzzy sets of Σ^*. i.e. $L:\Sigma^* \to [0,1]$. Also, fuzzy language of fuzzy automaton $M = (Q, \Sigma, \mu, I, F)$ is a fuzzy set $L(M) \neq \Sigma^*$ defined as $L(M)(x) = \bigvee_{p,q \in Q} \{I(p) \wedge \mu(p,x,q) \wedge F(q)\}$. Here, we discuss closure properties of fuzzy languages of fuzzy automata namely union, intersection, Cartesian product and reversal. We also established closure properties of homomorphic and inverse homomorphic images with the help of extension principal. We show that the class of fuzzy languages generated by fuzzy automata is not closed with respect to complementation.

Definition 3.3.1 If $f:\Sigma \to \Delta^*$ is a homomorphism and $L:\Sigma^* \to [0,\ 1]$ is a fuzzy language over Σ, then (due to extension principle) [3], $f(L)$ is a fuzzy language over Δ defined as follows: For all $\forall y \in \Delta^*$.

$$f(L)(y) = \begin{cases} \bigvee_{x \in f^{-1}(y)} L(x), & \text{if } f^{-1}(y) \neq \phi \\ 0, & \text{otherwise} \end{cases}$$

We call this fuzzy language, $f(L)$, the *homomorphic image* of the fuzzy language L.

Similarly, if $K:\Delta^* \to [0,\ 1]$ is a fuzzy language over Δ, then $f^{-1}(K)$ is a fuzzy language over Σ defined as

$[f^{-1}(K)](x) = K[f(x)]$, $\forall x \in \Delta^*$.

This fuzzy language $f^{-1}(K)$ is called the *inverse homomorphic* image of the fuzzy language K.

We now recall fuzzy language of a fuzzy finite automaton discussed in [34]:

Definition 3.3.2 [60] Let $M = (Q, \Sigma, \mu, I, F)$ be a fuzzy finite automaton. Then the fuzzy language $L(M)$ of Σ^* defined as $L(M)(x) = \bigvee_{p, q \in Q} \{I(p) \wedge \mu(p,x,q) \wedge F(q)\}$ is called the *fuzzy language generated* by M.

We now discuss closure properties of fuzzy languages accepted by fuzzy automata.

Theorem 3.3.3 Let L_1 and L_2 are fuzzy languages generated by fuzzy automaton. Then $L_1 \cup L_2$, $L_1 \cap L_2$, $L_1 \times L_2$, L^ρ are fuzzy languages generated by some fuzzy automaton.

Proof (1) Let $M_1 = (Q_1,\ \Sigma,\ \mu_1,\ I_1,\ F_1)$ and $M_2 = (Q_2,\ \Sigma,\ \mu_2,\ I_2,\ F_2)$ be fuzzy finite automaton accepting L_1 and L_2 respectively.

Construct a fuzzy automaton $M = (Q_1 \cup Q_2,\ \Sigma,\ \mu_1 \vee \mu_2,\ I_1 \vee\ I_2,\ F_1 \vee F_2)$, by making $Q_1 \cap Q_2 = \phi$ (possibly by renaming elements of either Q_1 or Q_2), where

$$(\mu_1 \vee \mu_2)(q,\ a,\ p) = \begin{cases} \mu_1(q,\ a,\ p), & \text{if } q,\ p \in Q_1 \\ \mu_2(q,\ a,\ p), & \text{if } q,\ p \in Q_2 \\ 0 & \text{otherwise;} \end{cases}$$

$$(I_1 \vee I_2)(q) \quad = \quad \begin{cases} I_1(q), & \text{if} \quad q \in Q_1 \\ I_2(q), & \text{if} \quad q \in Q_2 \end{cases}$$

and

$$(F_1 \vee F_2)(p) \quad = \quad \begin{cases} F_1(p), & \text{if} \quad p \in Q_1 \\ F_2(p), & \text{if} \quad p \in Q_2 \end{cases}$$

Now,

$$L(M)(x) = \bigvee_{q,p\ \in Q_1 \cup Q_2} \{(I_1 \vee I_2)(q) \wedge (\mu_1 \vee \mu_2)(q,x,p) \wedge (F_1 \vee F_2)(p)\}$$

$$= \left(\bigvee_{q,\ p\ \in Q_1} \{(I_1 \vee I_2)(q) \wedge (\mu_1 \vee \mu_2)(q,x,p) \wedge (F_1 \vee F_2)(p)\} \right) \vee$$

$$\left(\bigvee_{q,\ p\ \in Q_2} \{ (I_1 \vee I_2)(q) \wedge (\mu_1 \vee \mu_2)(q,x,p) \wedge (F_1 \vee F_2)(p)\} \right)$$

$$\vee \left(\bigvee_{p\ \in Q_1 \text{ and } q\ \in Q_2} \{(I_1 \vee I_2)(q) \wedge (\mu_1 \vee \mu_2)(q,\ x,\ p) \wedge (F_1 \vee F_2)(p)\} \right) \vee$$

$$\left(\bigvee_{p\ \in Q_2 \text{ and } q\ \in Q_1} \{(I_1 \vee I_2)(q) \wedge (\mu_1 \vee \mu_2)(q,\ x,\ p) \wedge (F_1 \vee F_2)(p)\} \right)$$

$$= \left(\bigvee_{q,\ p\ \in Q_1} \{I_1\ (q) \wedge \mu_1\ (q,x,p) \wedge F_1(p)\} \right) \vee$$

$$\left(\bigvee_{q,\ p\ \in Q_2} \{ I_2(q) \wedge \ \mu_2(q,x,p) \wedge F_2(p)\} \right), \text{ since } Q_1 \cap Q_2 = \varnothing$$

$$= L(M_1)(x) \vee L(M_2)(x)$$

$= (L_1 \cup L_2)(x)$

Thus, $L_1 \cup L_2$ is fuzzy automaton language generated by fuzzy automaton M.

(2) Let $M_1 = (Q_1, \Sigma, \mu_1, I_1, F_1)$ and $M_2 = (Q_2, \Sigma, \mu_2, I_2, F_2)$ be fuzzy finite automaton.

Construct $M = (Q_1 \times Q_2, \Sigma, \mu_1 \times \mu_2, I_1 \times I_2, F_1 \times F_2)$, where

$(\mu_1 \times \mu_2)((q_1, q_2), a, (p_1, p_2)) = \mu_1(q_1, a, p_1) \wedge \mu_2(q_2, a, p_2)$,
$(I_1 \times I_2)(q_1, q_2) = I_1(q_1) \wedge I_2(q_2)$ and

$(F_1 \times F_2)(p_1, p_2) = F_1(p_1) \wedge F_2(p_2)$.

Then,

$L(M)(x)$

$$= \bigcup_{(q_1, q_2), (p_1, p_2) \in Q_1 \times Q_2} \{(I_1 \times I_2)(q_1, q_2) \wedge (\mu_1 \times \mu_2)((q_1, q_2), x, (p_1, p_2)) \wedge (F_1 \times F_2)(p_1, p_2)\}$$

$$= \left(\bigvee_{q_1, p_1 \in Q_1} (I_1(q_1) \wedge \mu_1(q_1, x, p_1) \wedge F_1(p_1))\right) \wedge \left(\bigvee_{q_2, p_2 \in Q_2} (I_2(q_2) \wedge \mu_2(q_2, x, p_2) \wedge F_2(p_2))\right)$$

$= L_1(x) \wedge L_2(x)$

$= \left(L_1 \cap L_2\right)(x)$

Thus, $L_1 \cap L_2$ is a fuzzy language generated by fuzzy automaton M.

(3) Let $M = (Q, \Sigma, \mu, I, F)$ and $M' = (Q', \Sigma', \mu', I', F')$ be fuzzy finite automaton such that $L_1 = L(M_1)$ and $L_2 = L(M_2)$

Then $M \times M' = \left(Q \times Q', \Sigma \times \Sigma', \mu \times \mu', I \times I', F \times F'\right)$ is a fuzzy finite automaton, where

$(\mu \times \mu')\left((q, q'), (x, x'), (p, p')\right) = \mu(q, x, p) \wedge \mu'(q', x', p'),$
$I \times I'\left(q, q'\right) = I\left(q\right) \wedge I'\left(q'\right)$ and

$I \times I'\left(q, q'\right) = I\left(q\right) \wedge I'\left(q'\right)$

Then it is evident that $L\left(M \times M'\right)(x, x')$
$= L(M)(x) \times L(M')(x')$

Thus, $L_1 \times L_2$ is a fuzzy language generated by fuzzy automaton $M \times M'$.

(4) Let $M = (Q, \Sigma, \mu, I, F)$ be a fuzzy finite automaton such that $L = L(M)$.

Construct a fuzzy finite automaton $M^{\rho} = (Q, \Sigma, \mu^{\rho}, F, I)$, where $\mu^{\rho}(q, x, p) = \mu(q, \rho(x), p)$.

Then,

$\left[L(M^{\rho})\right](x)$

$= \bigvee_{p, q \in Q} \{F(p) \wedge \mu^{\rho}(p, x, q) \wedge I(q)\}$

$= \bigvee_{q, p \in Q} \{I(q) \wedge \mu(q, \rho(x), p) \wedge F(p)\}$

$= L(M)\rho(x) = [L(M)]^{\rho}(x)$

Thus, L^{ρ} is fuzzy language generated by fuzzy finite automaton M^{ρ}.

Theorem 3.3.4 Let L be fuzzy automaton language by fuzzy finite automaton and $f : \Sigma \rightarrow \Sigma'$ be a homomorphism. Then $f(L)$ and $f^{-1}(L)$ are fuzzy languages generated by fuzzy finite automaton.

Proof (1) Let $f:\Sigma\to\Sigma'$ be a homomorphim and $M=(Q,\ \Sigma,\mu,\ I,F)$ be a fuzzy finite automaton such that $L=L(M)$.

Then $f(M)=(Q,\ \Sigma',\ \mu',\ I,\ F)$ be a fuzzy finite automaton, where

$$\mu'(q,\ x',\ p)=\bigvee_{x:\,x'=f(x)}\{\mu(q,\ x,\ p)\}$$

Now $f\big(L(M)\big)(x')=\bigvee\big\{\big(L(M)\big)(x)/\ x:x'=f(x)\big\},$

since f is onto.

$$=\bigvee_{x:\,x'=f(x)}\left\{\bigvee_{p,\,q\,\in\,Q}\big(I(q)\wedge\mu(q,\ x,\ p)\wedge F(p)\big)\right\}$$

$$= \bigvee_{p,\, q\, \in Q} \left\{ \left(I(q) \wedge \left(\bigvee_{x:\, x' = f(x)} \mu(q,\, x,\, p) \right) \wedge F(p) \right) \right\}$$

$$= \bigvee_{p,\, q\, \in Q} \left\{ I(q) \wedge \mu'(q,\, x',\, p) \wedge F(p) \right\}$$

$$= L\big(f(M) \big)(x')$$

Thus, $f(L)$ is a fuzzy language generated by fuzzy finite automaton $f(M)$.

(2) Let $M = (Q,\ \Sigma,\ \mu,\ I, F)$ be a fuzzy finite automaton such that $L = L(M)$

Construct a fuzzy finite automaton

$f^{-1}(M) = (Q,\ \Sigma',\ \mu',\ I, F)$, where

$\mu' : Q \times\ \Sigma' \times Q \rightarrow [0,\ 1]$ such that

$\mu'(p, \sigma', q) = \mu(p, f(\sigma'), q)$.

Now, $L\big(f^{-1}(M) \big)(x)$

$$= \bigvee_{q,p \in Q} \{I(q) \wedge \mu'(q, x, p) \wedge F(p)\}$$

$$= \bigvee_{q,p \in Q} \{I(q) \wedge \mu(q, f(x), p) \wedge F(p)\}$$

$$= L(M)\big(f(x)\big)$$

$$= f^{-1}\big(L(M)\big)(x)$$

Thus, $f^{-1}(L)$ is a fuzzy language generated by fuzzy finite automaton $f^{-1}(M)$.

The complement of the fuzzy automaton crisp language is a fuzzy automaton crisp language, in reference to definitions given in [7, 22, 25, 26, 35], but the following example shows that this is not true for *fuzzy languages generated by fuzzy finite automaton*.

Example 3.3.5 Let

$L = \{(y, L(y)) \mid y \in \{0, 1\}^* \text{ and y ends with 01}\}$, where

$$L(y)=\begin{cases} \dfrac{\text{no. of } 0's \text{ in y}}{\text{no. of } 1's \text{ in y}} & , \text{ if no. of 1's are more than no. of 0's} \\ \dfrac{\text{no. of } 1's \text{ in y}}{\text{no. of } 0's \text{ in y}} & , \text{ if no. of 0's are more than no. of 1's} \\ 1 & , \text{ otherwise} \end{cases}$$

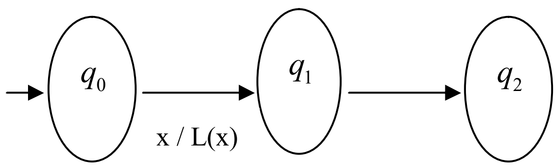
Consider a fuzzy finite automaton $M=(Q,\ \Sigma,\ \mu,\ I,\ F)$, given below $I(q_0)=F(q_f)=1$

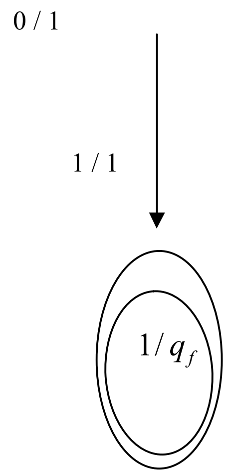
and μ is defined as

$$\mu(q_0,x,q_1)=\begin{cases} \dfrac{(\text{no. of } 0's \text{ in} x)+1}{(\text{no. of } 1's \text{ in} x)+1} & , \text{ if no. of 1's are more than no. of 0's} \\ \dfrac{(\text{no. of 1's in} x)+1}{(\text{no. of 0's in} x)+1} & , \text{ if no. of 0's are more than no. of 1's} \\ 1 & , \text{ otherwise,} \end{cases}$$

$\mu(q_1,\ 0,\ q_2)=1$ and $\mu(q_2,\ 1,\ q_f)=1.$

The diagrammatic representation is as follows:





Now,

$L(M)(y) = L(y)$, $\forall y \in \{0,\ 1\}^*$ that ends with 01

But for fuzzy finite automaton $M^c = (Q, \Sigma, \mu, I, F^c)$,

where $F^c = Q - F$ in general , we have

$L(M^c)(y) \neq L^c(y)$.

For this, let $y = 00$.

Then

$L(M^c)(00) = \bigvee_{p,\, q \in Q} \{I(p) \wedge \mu(p, 00, q) \wedge F^c(q)\} = \frac{1}{2}$ and

$L^c(00) = 1 - L(00) = 1$.

Thus, the fuzzy language generated by fuzzy automaton is not necessarily closed under complementation.

We now define fuzzy quotient language and discuss its closure property

Definition 3.3.6 Let L_1 and L_2 be two fuzzy languages over Σ^*. The *quotient of* L_1 *by* L_2, denoted by $L_1L_2^{-1}$ (or L_1/L_2), is a fuzzy language over Σ^* which is defined as follows: $L_1L_2^{-1}(x) = \bigvee_y \{L_1(xy) \wedge L_2(y)\}, \forall x \in \Sigma^*$.

Theorem 3.3.7 If L_1 is fuzzy language generated by fuzzy finite automaton and L_2 is a fuzzy languages over Σ^*, then $L_1L_2^{-1}$ is a fuzzy language generated by fuzzy finite automaton.

Proof Let $M = (Q, \Sigma, \mu, I, F)$ be a fuzzy finite automaton such that $L(M) = L_1$. Construct a fuzzy finite automaton $M' = (Q, \Sigma, \mu, I, F')$, where $F' : Q \to [0, 1]$ defined by $F'(q) = \bigvee_{y \in \Sigma^*} \{ \mu(q, y, q') \wedge F(q') \wedge L_2(\mathrm{y}) \}$.

Now, $L_1 L_2^{-1}(x) = \bigvee_{y} \{ L_1(xy) \wedge L_2(y) \}$

$$= \bigvee_{y} \left\{ \bigvee_{q, p \in Q} \{ I(q) \wedge \mu(q, xy, p) \wedge F(p) \} \wedge L_2(y) \right\}$$

$$= \bigvee_{y} \left\{ \bigvee_{q, p \in Q} \{ I(q) \wedge \left(\bigvee_{r} \{ \mu(q, x, r) \wedge \mu(r, y, p) \} \right) \wedge F(p) \} \wedge L_2(y) \right\}$$

$$= \bigvee_{p, q, r \in Q} \left\{ I(q) \wedge \mu(q, x, r) \wedge \left[\bigvee_{y} \{ \mu(r, y, p) \} \wedge F(p) \wedge L_2(y) \right] \right\}$$

$$= \bigvee_{q, r \in Q} \{ I(q) \wedge \mu(q, x, r) \wedge F'(r) \}$$

$$= L(M')(x).$$

Hence, $L_1L_2^{-1}$ is a fuzzy language generated by fuzzy automaton M'.

Slight modification in the above definition, enable us to define

Definition 3.3.8 Let L_1 and L_2 be two fuzzy languages over Σ^*. The *right quotient of* L_1 *by* L_2, denoted by $L_2^{-1}L_1$ (or L_1/L_2), is a fuzzy language over Σ^* which is defined as follows: $L_2^{-1}L_1(x) = \bigvee_y \{L_1(yx) \wedge L_2(y)\}, \forall x \in \Sigma^*$.

Theorem 3.3.9 If L_1 is fuzzy language generated by fuzzy finite automaton and L_2 is a fuzzy languages over Σ^*, then $L_2^{-1}L_1$ is a fuzzy language generated by fuzzy finite automaton.

Proof Let $M = (Q, \Sigma, \mu, I, F)$ be a fuzzy finite automaton such that $L(M) = L_1$. Construct a fuzzy finite automaton $M' = (Q, \Sigma, \mu, I', F)$, where

$I' : Q \to [0,\ 1]$ defined by

$$I'(q') = \bigvee_{y \in \Sigma^*} \{\ I(\ q\ ) \wedge \mu(\ q,\ y,\ q'\ ) \wedge L_2(y)\ \}.$$

Now, $L_2^{-1} L_1(x) = \bigvee_y \{L_1(yx) \wedge L_2(y)\}$

$$= \bigvee_y \left\{ \bigvee_{q,\,p \in Q} \{I(q) \wedge \mu(q, yx, p) \wedge F(p)\} \wedge L_2(y) \right\}$$

$$= \bigvee_y \left\{ \bigvee_{q,\,p \in Q} \{I(q) \wedge \left(\bigvee_r \{\mu(q, y, r) \wedge \mu(r, x, p)\} \right) \wedge F(p)\} \wedge L_2(y) \right\}$$

$$= \bigvee_{p,q,r \in Q} \left\{ \left[\bigvee_y \{I(q) \wedge \mu(q, y, r) \wedge L_2(y)\} \right] \wedge \mu(r, x, p) \wedge F(p) \right\}$$

$$= \bigvee_{r,\,p \in Q} \{I'(r) \wedge \mu(r, x, p) \wedge F(p)\}$$

$$= L(M')(x).$$

Hence, $L_2^{-1} L_1$ is a fuzzy language generated by fuzzy automaton M'.

3.4 Fuzzy Regular Languages In this section we introduce fuzzy regular grammars with fuzzy initial variables and prove that they are equivalent to fuzzy finite automata, in the sense of same fuzzy language generation.

Definition 3.4.1 Let $G = (V, \Sigma, S, P)$ be a fuzzy regular grammar. Then the fuzzy language $L(G)$ of Σ^* defined by $L(G)(x) = \bigvee_{A_0 \in V} \{S(A_0) \wedge d(A_0 \Rightarrow x)\}$ is called the *fuzzy language generated* by G.

Definition 3.4.2 A fuzzy language L of Σ^* is called a *regular language*, if their exists a fuzzy regular grammar G such that $L = L(G)$.

Following theorems establish equivalence between fuzzy language generated by fuzzy finite automaton and fuzzy regular language.

Theorem 3.4.3 Every *non-null* fuzzy language generated by fuzzy finite automaton is a fuzzy regular language.

Proof Let L be a fuzzy finite automaton language generated by fuzzy finite automaton

$M=(Q,\ \Sigma,\ \mu,\ I,\ F)$.

Let

$$P=\{p\xrightarrow{\sigma}aq \mid \mu(p,\ a,\ q)=\sigma\}\cup\{p\xrightarrow{\sigma}a \mid \mu(p,\ a,\ q)=\sigma \text{ and } F(q)>0\}$$

Then $G=(Q,\ \Sigma,\ I,\ P)$ is a fuzzy regular grammar.

Let $x=a_1a_2\ldots a_{n-1}a_n$; $a_i\in\Sigma$. We have

$L(M)(x)=\vee\{I(q)\wedge\mu(q,x,p)\wedge F(p)\mid q,p\in Q\}$ if and only if there exist $q_0,p_0\in Q$

such that $L(M)(x)=I(q_0)\wedge\ \mu(q_0,\ x,\ p_0)\wedge\ F(p_0)$

$\mu(q_0,x,p_0)=\mu(q_0,a_1,q_1)\wedge\mu(q_1,a_2,q_2)\wedge\ldots\wedge\mu(q_{n-2},a_{n-1},q_{n-1})\wedge\mu(q_{n-1},a_n,p_0)$, for some $q_1,\ q_2,\ \ldots\ ,\ q_{n-1}\in Q$

If

$\mu(q_0,a_1,q_1)=d_1,\mu(q_1,a_2,q_2)=d_2,\ldots,\mu(q_{n-2},a_{n-1},q_{n-1})=d_{n-1},\mu(q_{n-1},a_n,p_0)=d_n,$

where

$d_i\in[0,\ 1]$, then by definition of P

$q_0\xrightarrow{d_1}a_1q_1$, $q_1\xrightarrow{d_2}a_2q_2$, ..., $q_{n-2}\xrightarrow{d_{n-1}}a_{n-1}q_{n-1}$, $q_{n-1}\xrightarrow{d_n}a_np_0$ and $q_{n-1}\xrightarrow{d_n}a_n$ for all $p\in V$ such that $F(p_0)>0.$

Now,

$q_0\xrightarrow{d_1}a_1q_1\xrightarrow{d_2}a_1a_2q_2\xrightarrow{d_3}\ldots\xrightarrow{d_{n-1}}a_1a_2\ldots a_{n-1}q_{n-1}\xrightarrow{d_n}a_1a_2\ldots a_{n-1}a_n=x$

implies that

$q_0\Rightarrow x$ in G; since $F(p_0)>0.$

Thus, $\mu(q_0,x,p_0)=d(q_0\Rightarrow x)$

Therefore, $L(M)(x) = S(q_0) \wedge d(q_0 \Rightarrow x)$, for some $q_0 \in V$.

$$\leq \vee_{q \in V} \{S(q) \wedge d(q \Rightarrow x)\}$$

$$= L(G)(x)$$

Thus, $L(M) \subseteq L(G)$

Now, $L(G)(x) = \vee_{q \in V} \{S(q) \wedge d(q \Rightarrow x)\}$

i.e. there exists $q_0 \in V$ such that

$L(G)(x) = S(q_0) \wedge d(q_0 \Rightarrow x)$

Since $d(q_0 \Rightarrow x) = \mathrm{Sup}\{\min(\sigma_1, \sigma_2, \ldots, \sigma_n)\}$, where supremum is taken over all derivation chains of x from q_0, there exist

$q_1, q_2, \ldots, q_{n-1} \in V$ and $d_1, d_2, \ldots, d_n \in [0, 1]$ such that

$$q_0 \xrightarrow{d_1} a_1 q_1 \xrightarrow{d_2} a_1 a_2 q_2 \xrightarrow{d_3} \ldots \xrightarrow{d_{n-2}} q_{n-2} \xrightarrow{d_{n-1}} a_1 a_2 \ldots a_{n-1} q_{n-1} \xrightarrow{d_n} a_1 a_2 \ldots a_{n-1} a_n = x$$

Then, P must have following fuzzy productions:

$q_0 \xrightarrow{d_1} a_1 q_1$, $q_1 \xrightarrow{d_2} a_2 q_2$, … , $q_{n-2} \xrightarrow{d_{n-1}} a_{n-1} q_{n-1}$ and

$q_{n-1} \xrightarrow{d_n} a_n$ Whence,

$\mu(q_0, a_1, q_1) = d_1,\ \mu(q_1, a_2, q_2) = d_2\ ,....,\ \mu(q_{n-2}, a_{n-1}, q_{n-1}) = d_{n-1},\ \mu(q_{n-1}, a_n, p_0) = d_n$
for some $p_0 \in V$ such that $F(p_0) > 0$

Clearly by vary choice of $d_1, d_2, \ldots, d_n$, we have

$\mu(q_0, x, p_0) = d(q_0 \Rightarrow x)$

Therefore,

$L(G)(x) = I(q_0) \wedge \mu(q_0, x, p_0) \wedge F(p_0)$ for some $q_0, p_0 \in V = Q$
$\leq \bigvee_{q_0, p_0 \in Q} \{I(q_0) \wedge \mu(q_0, x, p_0) \wedge F(p_0)\}$

$= L(M)(x)$

Example 3.4.4 Consider a fuzzy automaton M given below

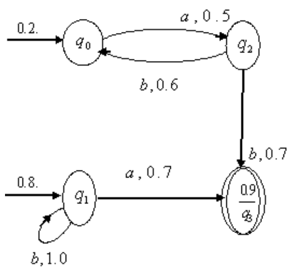


Diagram : Fuzzy finite automaton

Here, $L(M)=\left\{\frac{0.2}{(ab)^{*}ab},\frac{0.7}{b^{*}a}\right\}$

Then equivalent fuzzy grammar will be as follows:

$q_0\xrightarrow{0.5}aq_2, q_1\xrightarrow{0.7}aq_3, q_1\xrightarrow{1.0}bq_1, q_1\xrightarrow{0.7}a, q_2\xrightarrow{0.6}bq_0, q_2\xrightarrow{0.7}bq_3$ and

$q_2\xrightarrow{0.7}b.$

Theorem 3.4.5 Every fuzzy regular language is generated by a fuzzy finite automaton.

Proof Consider $Q= V\cup\{q_f\}$, where q_f not in V,

$I=S,\ F=\{1/q_f\}.$

Define $\mu:Q\times\Sigma\times Q\to[0,1]$ by

$$\mu(q,a,p)=\sigma \quad \text{if and only if} \quad q\xrightarrow{\sigma}ap$$

$$\mu(q,a,q_f)=\sigma \quad \text{if and only if} \quad q\xrightarrow{\sigma}a$$

Then, $M=(Q,\Sigma,\mu,I,F)$ is a fuzzy finite automaton

Let $x=a_1a_2...a_{n-1}a_n;\ a_i\in\Sigma$

$$L(G)(x) = \vee_{q \in V} \{S(q) \wedge d(q \Rightarrow x)\}$$

i.e. there exists $q_0 \in V$ such that

$L(G)(x) = S(q_0) \wedge d(q_0 \Rightarrow x)$

Since $d(q_0 \Rightarrow x) = \text{Sup}\{\min(\sigma_1, \sigma_2, \ldots, \sigma_n)\}$, where the supremum is taken over all derivation chains of x from q_0, there exist $q_1, q_2, \ldots, q_{n-1} \in V$ and $d_1, \text{d}_2, \ldots, d_n \in [0, 1]$ such that

$$q_0 \xrightarrow{d_1} a_1 q_1 \xrightarrow{d_2} a_1 a_2 q_2 \xrightarrow{d_3} \ldots \xrightarrow{d_{n-1}} a_1 a_2 \ldots a_{n-1} q_{n-1} \xrightarrow{d_n} a_1 a_2 \ldots a_{n-1} a_n = x$$

Then, corresponding to above derivation chain , P must have following fuzzy productions:

$q_0 \xrightarrow{d_1} a_1 q_1$, $q_1 \xrightarrow{d_2} a_2 q_2$, …, $q_{n-2} \xrightarrow{d_{n-1}} a_{n-1} q_{n-1}$ and $q_{n-1} \xrightarrow{d_n} a_n$

Whence,

$\mu(q_0, a_1, q_1)=d_1, \mu(q_1, a_2, q_2)=d_2, \ldots, \mu(q_{n-2}, a_{n-1}, q_{n-1})=d_{n-1}$ and $\mu(q_{n-1}, a_n, q_f)=d_n$

Since

$$\mu(q_0, a_1a_2, p) = \vee\{\mu(q_0, a_1, r)\mu(r, a_2, p) \mid r \in Q\} ,$$

we have

$$\mu(q_0, x, q_f) =$$
$$\mu(q_0, a_1, q_1) \wedge \mu(q_1, a_2, q_2) \wedge \ldots \wedge \mu(q_{n-2}, a_{n-1}, q_{n-1}) \wedge \mu(q_{n-1}, a_n, q_f)$$

Thus there exist $q_0, q_f \in Q$ such that

$$\mu(q_0, x, q_f) = d(q_0 \Rightarrow x)$$

$$L(G)(x) = I(q_0) \wedge \mu(q_0, x, q_f) \wedge F(q_f) \text{ for some } q_0 \in V$$

$$\leq \vee_{q, q_f \in Q}\{I(q) \wedge \mu(q, x, q) \wedge F(q_f)\} = L(M)(x)$$

Thus, $L(G) \subseteq L(M)$

Now, $L(M)(x) = \vee_{q, p \in Q}\{I(q) \wedge \mu(q, x, p) \wedge F(p)\}$

i.e. there exist $q_{0,} q_f \in Q$ such that

$$L(M)(x) = I(q_0) \wedge \mu(q_0, x, q_f) \wedge F(q_f)$$

By definition of μ, there exist $q_1, q_2, \dots, q_{n-1} \in Q$

such that

$$\mu(q_0, x, q_f) =$$
$$\mu(q_0, a_1, q_1) \wedge \mu(q_1, a_2, q_2) \wedge \dots \wedge \mu(q_{n-2}, a_{n-1}, q_{n-1}) \wedge \mu(q_{n-1}, a_n, q_f)$$

Let

$\mu(q_0, a_1, q_1) = d_1, \mu(q_1, a_2, q_2) = d_2, \dots, \mu(q_{n-2}, a_{n-1}, q_{n-1}) = d_{n-1}$ and $\mu(q_{n-1}, a_n, q_f) = d_n$, where $d_i \in [0,\ 1]$

Whence,

$q_0 \xrightarrow{d_1} a_1 q_1$, $q_1 \xrightarrow{d_2} a_2 q_2$, … , $q_{n-2} \xrightarrow{d_{n-1}} a_{n-1} q_{n-1}$ and

$q_{n-1} \xrightarrow{d_n} a_n$, since $F(q_f) > 0$

Clearly by vary choice of $d_1, d_2, \dots, d_n$, we have

$$\mu(q_0, x, p_0) = d(q_0 \Rightarrow x)$$

Therefore,

$$L(M)(x) = S(q_0) \ \wedge \ d(q_0 \Rightarrow x) \ \text{ for some } q_0 \in V$$

$$\leq \bigvee_{q_0 \in V} \{S(q_0) \ \wedge \ d(q_0 \Rightarrow x)\}$$

$$= L(G)(x).$$

Example 3.4.6 Consider a fuzzy grammar

$$G = \left(\{A, B, C\}, \{0, 1\}, \left\{ \frac{0.6}{A}, \frac{0.4}{B} \right\}, \{ A \xrightarrow{0.5} 1, B \xrightarrow{0.6} 0, B \xrightarrow{0.7} 0C, C \xrightarrow{0.3} 0A, C \xrightarrow{0.7} 1C \} \right)$$

Then, $L(G) = \left\{ \frac{0.5}{1}, \frac{0.4}{0}, \frac{0.3}{01^{*}01} \right\}.$

An equivalent fuzzy finite automaton is :

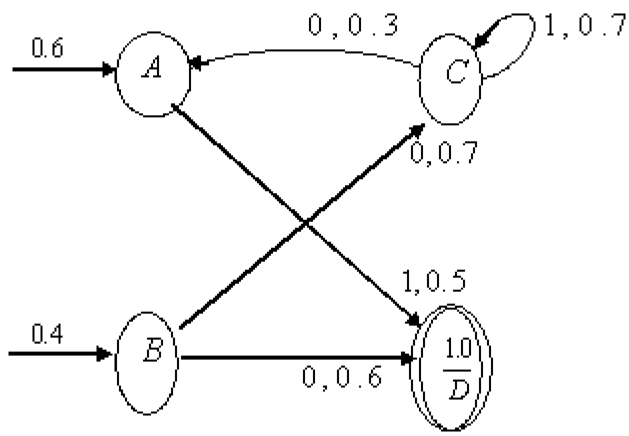


Diagram : Fuzzy finite automaton

Corollary 3.4.7 Fuzzy finite automaton and fuzzy regular grammar are equivalent.

Theorem 3.3.3, 3.3.4 and 3.3.7 enable us to consider following remarks

Remarks 3.4.8

(1) Union, intersection, Cartesian product and reversal of fuzzy regular languages is a fuzzy regular language.

(2) If $L_1 \subseteq \Sigma^*$ be a fuzzy automaton language and $L_2 \subseteq \Sigma^*$ be any set, then $L_1 L_2^{-1}$ and $L_2^{-1} L_1$ are fuzzy regular languages.

(3) Let $f : \Sigma \rightarrow \Delta^*$ be a homomorphism. Then

(1) If $L \subseteq \Sigma^*$ is a fuzzy regular language, then $f(L)$ is a fuzzy regular language.

(2) If $K \subseteq \Delta^*$ is a fuzzy regular language, then $f^{-1}(K)$ is a fuzzy regular language

Bibliography

[1] Abolpour K. and Zahedi M., Isomorphism between two BL- general fuzzy automata, *Soft Comput.,* **16(4)** (2012) pp. 729-736

[2] Astrain J., Garitagoitia J., Gonzalez D., Mendivil J. , Villadangos J. and Fariña F., Approximate String Matching Using Deformed Fuzzy Automata, *Fuzzy Optimization and Decision Making,* **3(2)** (2004) pp. 141-155.

[3] Asveld P. R. J., Fuzzy context-free languages-Part 1: Generalized fuzzy context-free grammars, *Theoretical Computer Science* **347(1-2)** (2005) pp. 167-190.

[4] Asveld P. R. J., Fuzzy context-free languages-Part 2: Generalized fuzzy context-free grammars, *Theoretical Computer Science* **347(1-2)** (2005) pp.191-213.

[5] Belohlavek R., Determinism and fuzzy automata, *Information Sciences* **143** (2002) pp. 205-209.

[6] Blanco A., Delgado M., Pegalajar M., *Fuzzy automaton induction using neural network*, Int. J. of Appro. Reasoning, **27** (2001) pp. 1 – 26.

[7] Bucurescu I. and Pascu A., Fuzzy pushdown automata, *J. Comput.. Math*., **10** (1981) pp. 109 – 119

[8] Bucurescu I. and Pascu A., On the languages generated by context-sensitive fuzzy grammars, *International*

Journal of Computer Mathematics **9(4)** (1981) pp. 277 – 285.

[9] Cao Y. and Ezawa Y., Non-deterministic fuzzy automata, *Inf.. Sci.,* **191** (2012) pp. 86-97.

[10] Cheng W. and Mo Z., Minimization algorithm of fuzzy finite automata, *Comput. Math. with Appli.*, **57** (2009) pp. 949-960.

[11] Ćirić M., Stamenković A., Ignjatović J. and Petković T., Fuzzy relation equations and reduction of fuzzy automata , *J. Comput. System Sci.,* **76** (*2009*) pp. 609-633.

[12] Dal Cin M., Fuzzy State Automata: Their Stability and Fault Tolerance, *International Journal of Computer and Inf. Sci.,* **4(1)** (1975) pp. 63-80.

[13] Dal Cin M., Modification Tolerance of Fuzzy Automata, *International Journal of Computer and Inf. Sci*., **4(1)** (1975) pp. 63-80.

[14] Doostfatemeh M. and Kremer S., *Representing Generalised fuzzy automata in recurrent neural networks*, proc. Canadin Con. On Electrical and Computer Engineering, **4** (2004), pp. 1901-1906.

[15] Doostfatemeh M. and Kremer S.C., New directions in fuzzy automata, *Int. J. of approximate reasoning*, **38** (2005), pp. 175-214.

[16] Doostfatemeh M. and Kremer S. Biological Sequence Prediction using General Fuzzy Automata, *CIBCB* (2005) pp. 406-413.

[17] Doostfatemeh M**.** and Kremer S**.,** General fuzzy automata, New Efficient Accepters for Fuzzy Languages**,** *proc. IEEE Int. Con. On Fuzzy Syst.,* (2006) pp. 2097**-**2103.

[18] Dubois D. and Prade H., *Fuzzy Sets and Systems: Theory and Applications*, Academic Press Inc. 1980

[19] Gerla G., Fuzzy grammars and recursively enumerable fuzzy languages*, Inform. Sciences* **60** (1992) pp. 137 – 143.

[20] Gupta M., Saridas G. and Gaines B. , *Fuzzy Automata and Decision Processes*. North Holland, New York, 1977.

[21] Holcombe W., *Algebraic Automata Theory*, Cambridge University press, 1982.

[22] Hopcroft J., Ullman J., *Introduction to automata theory, languages and computation*, Addson-Wesley, New York, 1979.

[23] Horry M. and Zahedi M., Hypergroups and general fuzzy automata, *Iranian J. of fuzzy systems,* **6(2)** (2009) pp. 61 – 74.

[24] Horry M. and Zahedi M., On general fuzzy recognizers, *Iranian J. of Fuzzy Syst.* **8(3)** (2011), 125-135

[25] Horry M. and Zahedi M., Uniform and semi-uniform topology on general fuzzy automata, Iranian J. of fuzzy Systems **6(2)** (2009) pp. 19 – 29.

[26] Jahromi A., Karimaghaee P., Haghighi H., Doostfatemeh M**.** and Meng J. Intelligent selective packet discarding using general fuzzy Automata, *Proc. of IEEE Int. Con. On Syst., Man and Cybernetics* (2008) pp. 3714 – 3720.

[27] Jun L., Zhiwen M., Dong Q. and Yang W.**,** Products of Mealy-type fuzzy finite state Machines, *Fuzzy Sets and Syst.,* **160** (2009) pp. 2401-2415.

[28] Jun Y. and Kavikumar J., Bipolar fuzzy finite state machines, *Bull. Malays. Math. Soc.* **(2) 34(1)** (2011) pp. 181 – 188.

[29] Kandel A., and Lee S., *Fuzzy Switching and Automata: Theory and its Application*, Crane Rusak, 1980.

[30] Kim H., Mizumoto M., Toyoda J. and Tanaka K., L-fuzzy grammars*, Inf.. Sci.* **8** (1975) pp 123 – 140.

[31] Kim Y., Kim J. and Cho S., Product of T-generalized state machines and Transformation semigroups, *Int. J. of Fuzzy Sets and Syst*. **93** (1998) pp. 87-97.

[32] Klir G. J. and Yuan B., *Fuzzy Sets and Fuzzy Logic: Theory and Applications,* Prentice Hall, New York, 1995.

[33] Kumbhojkar H. and Chaudhari S. Advances in Pattern Recognition and Digital Techniques, *Proce. Narosa Publishing House,* (2000) pp. 261-264.

[34] Kumbhojkar H. and Chaudhari S., Fuzzy recognizers and recognizable sets, *Fuzzy Sets and Syst.* **131(3)** (2002) pp. 381 – 392.

[35] Kumbhojkar H. and Chaudhari S., On covering of Products of fuzzy finite state machines, *Int. J. of Fuzzy Sets and Syst.* **125** (2002), pp. 215-222.

[36] Lan S., *Fuzzy Pushdown automata*, http:// www. polytech. univ-savoie.fr / fileadmin / polytech_autres_sites / sites/listic/busefal/Papers/49.zip/49_10.pdf.

[37] Lan S. and Zhiwen M., Closure of finite-state automaton languages, *Fuzzy Sets and Syst.* **75** (1995) pp. 393-397.

[38] Lee E. T. and Zadeh L. A., Note on fuzzy languages, *Information Sciences* **1(4),** (1969) pp. 421-434.

[39] Lei H. and Li Y., Minimization of states in automata theory based on finite lattice-ordered monoids, *Inf.. Sci.,* **177**(2007) pp. 1413-1421.

[40] Li Y. and Pedrycz W., Minimization of lattice finite automata and its application to the decomposition of

lattice languages, *Fuzzy Sets and Syst.* **158** (2007) pp. 1423-1436.

[41] Malik D., Mordeson J. and Sen M., On Fuzzy Regular Languages. *Inf. Sci.* **88(1-4)** (1996), pp. 263- 298**.**

[42] Malik D. S. and Mordeson J. N., On fuzzy Recognizers, *Kybernetics* **28(1)** (1999) pp. 47-60

[43] Malik D., Mordeson J., and Sen M., On Subsystem of fuzzy finite state machines, *Int. J. of Fuzzy Sets and Syst***.** **68** (1994) pp. 83-92.

[44] Malik D., Mordeson J. and Sen M., Product of fuzzy finite state machines, *Int. J. of Fuzzy Sets and Syst.* **92**(1997) pp. 95-102.

[45] Malik D., Mordeson J., and Sen M., The Cartesian Composition of fuzzy Finite State Machines, *Kybernetics* **24(4)** (1995) pp. 98-110

[46] Mateescu A., Salomaa A., Salomaa K. and Yu S., Lexical Analysis with a Simple Finite-Fuzzy-Automaton Model, *Journal of Universal Computer Science*, **1(5)** (1995) pp. 292-311

[47] Mizumoto M., Tanaka J. and Tanaka K., Some consideration on fuzzy automata, *J. Comput. System Sci.* **3** (1969) pp. 409–422.

[48] Mizumoto M., Toyada J. and Tanaka K., Various kinds of automata, with weights, *J. Comput. System Sci.***10(2)** (1975) pp. 219–236.

[49] Mizumoto M., Toyoda J. and Tanaka K., General formulation of formal grammars, *Inf.. Sci.* **4** (1972), pp. 87–100.

[50] Mizumoto M., Toyoda J. and Tanaka K., N-fold fuzzy grammars, *Inf. Sci*. **5** (1973) pp. 25-43.

[51] Mordeson J. N. and Malik D. S., *Fuzzy Automata and Languages: Theory and Applications*, Chapman & Hall/CRC, 2002.

[52] Ning Q. and Terrence J. S., Predecting the secondary structure of globular protein Using Neural Network Model, *J. Molucular Biology* **202** (1988) pp. 865-884.

[53] Omlin C. and Giles C., Extraction of rules from discrete-time recurrent Neural Network, *Neural Network* **9(1)**, (1996) pp. 41-52.

[54] Paz A., Fuzzy Star Functions, Probabilistic Automata and Their Approximation by Non-probabilistic Automata, *J. Comput. Syst. Sci.***1** (1967) pp. 280-290.

[55] Petkovic T. Congruences and homomorphisms of fuzzy automata, *Fuzzy Sets and Syst.* **157**(2006) pp. 444-458.

[56] Ping Li. and Yong-ming Li, Algebraic properties of LA-languages, *Inf. Sci.* **176 (21)** (2006) pp. 3232-3255.

[57] Qing E. and Cui G. Application of Fuzzy Automata to Fuzzy Signal, *Processing of Intern. Conf. on Comput. Sci. and Software Engg.,* **5** (2008) pp. 1283-1286

[58] Qing E., Wang T., Huang Y. and Li S., *State fusion of fuzzy automata with application on target tracking*, Comput. Math. with Appli., **57** (2009) pp. 949 – 960.

[59] Rigatos G. and Tzafestas S., Fuzzy Automata for Fault Diagnosis, A Syntactic Analysis Approach, *In: Lecture Notes in Comput. Sci.* **3025** (2004) pp. 301-310.

[60] Santos E., Maximin automata, *Inf. and Control* **13** (1968) pp. 363–377.

[61] Santos E., On reduction of max-min machines*, J. Math. and Anal. Appl.,* **40** (1972) pp. 60-78.

[62] Santos E.,General formulation of sequential machines, *Inf.. and Control* **12** (1968) pp. 5–10.

[63] Santos, E., Max-product grammars and languages, *Inf. Sci.* **9**(1975b) pp. 1-23.

[64] Steimann F. and Adlassnig K., Clinical monitoring with fuzzy automata**,** *Fuzzy Sets and Syst.*, **61** (1994) pp. 37 – 42.

[65] Wechler W., *The Concept of Fuzziness in Automata and Language Theory*, Academic Varlag, Berlin. 1978.

[66] Wee W. and Fu K., A formulation of fuzzy automata and its application as a model of learning systems, *IEEE Trans. Syst. Sci. Cyber,* **5** (1969) pp. 215-23.

[67] Xing H., Fuzzy pushdown automata, *Fuzzy Sets and Syst* **158** (2007) pp. 1437 – 1449.

[68] Xiuhong G., Grammar theory based on lattice-ordered monoid, *Fuzzy Sets and Syst.* **160(8),** (2009) pp. 1152-1161.

[68] Zhang X. and Li Y., Intuitionistic fuzzy recognizers and intuitionistic fuzzy finite automata, *Soft computing*, **13(6)**, (2009) pp. 611-616.