Note on Fuzzy Languages†

E. T. LEE

AND L. A. ZADEH
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, California

ABSTRACT

A fuzzy language is defined to be a fuzzy subset of the set of strings over a finite alphabet. The notions of union, intersection, concatenation, Kleene closure, and grammar for such languages are defined as extensions of the corresponding notions in the theory of formal languages. An explicit expression for the membership function of the language $L(G)$ generated by a fuzzy grammar G is given, and it is shown that any context-sensitive fuzzy grammar is recursive. For fuzzy context-free grammars, procedures for constructing the Chomsky and Greibach normal forms are outlined and illustrated by examples.

1. INTRODUCTION

The precision of formal languages contrasts rather sharply with the imprecision of natural languages. To reduce the gap between them, it is natural to introduce randomness into the structure of formal languages, thus leading to the concept of stochastic languages [1–3]. Another possibility lies in the introduction of fuzziness. This leads to what might be called *fuzzy languages*.

It appears that much of the existing theory of formal languages can be extended quite readily to fuzzy languages. In this preliminary note, we shall merely sketch how this can be done for a few basic concepts and results. More detailed exposition of the theory of fuzzy languages will be presented in subsequent papers.

Our notation, terminology, and reasoning parallel closely the presentation of the theory of formal languages in Hopcroft and Ullman [4].

† This work was supported in part by the Army Research Office, Durham, DAHCO-4-69-0024.

Information Sciences 1 (1969), 421–434

2. BASIC DEFINITIONS

As usual, we denote by V_T a set of terminals; by V_N a set of non-terminals, with $V_T \cap V_N = \phi$; by $V_T{}^*$ the set of finite strings composed of elements of V_T; and by $(V_T \cup V_N)^*$ the set of finite strings composed of the elements of V_T or V_N. A generic string in $V_T{}^*$ is denoted by x or, more generally, by a lowercase letter near the end of the Latin alphabet.

Fuzzy language

A *fuzzy language*, L, is a fuzzy set† in $V_T{}^*$. Thus, L is a set of ordered pairs

$$L = \{(x, \mu_L(x))\}, \qquad x \in V_T{}^*, \tag{1}$$

where $\mu_L(x)$ is the grade of membership of x in L. We assume that $\mu_L(x)$ is a number in the interval $[0, 1]$.

A trivial example of a fuzzy language is the fuzzy set

$$L = \{(0, 1.0), (1, 1.0), (00, 0.8), (01, 0.7), (10, 0.6), (11, 0.5)\}$$

in $(0, 1)^*$. It is understood that all strings in $(0, 1)^*$ other than those listed have the grade of membership 0 in L.

Union, intersection, concatenation, and Kleene closure

Let L_1 and L_2 be two fuzzy languages in $V_T{}^*$. The *union* of L_1 and L_2 is a fuzzy language denoted by $L_1 + L_2$ and defined by

$$\mu_{L_1+L_2}(x) = \max(\mu_{L_1}(x), \mu_{L_2}(x)), \qquad x \in V_T{}^*. \tag{2}$$

In effect, $L_1 + L_2$ is the union of the fuzzy sets L_1 and L_2. Employing $\vee$ as an infix operator instead of max and omitting the argument x, we can write (2) more simply as

$$\mu_{L_1+L_2} = \mu_{L_1} \vee \mu_{L_2}. \tag{3}$$

The *intersection* of L_1 and L_2 is a fuzzy language denoted by $L_1 \cap L_2$ and defined by

$$\mu_{L_1 \cap L_2}(x) = \min(\mu_{L_1}(x), \mu_{L_2}(x)), \qquad x \in V_T{}^*, \tag{4}$$

† Intuitively, a fuzzy set is a class with unsharp boundaries, that is, a class in which the transition from membership to non-membership may be gradual rather than abrupt. More concretely, a fuzzy set A in a space $X = \{x\}$ is a set of ordered pairs $\{(x, \mu_A(x))\}$, where $\mu_A(x)$ is termed the *grade of membership* of x in A. (See [5] and [6] for more detailed discussion.) We shall assume that $\mu_A(x)$ is a number in the interval $[0, 1]$; more generally, it can be a point in a lattice [7, 8]. The *union* of two fuzzy sets A and B is defined by $\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$. The *intersection* of A and B is defined by $\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$. Containment is defined by $A \subset B \Leftrightarrow \mu_A(x) \leqslant \mu_B(x)$ for all x. Equality is defined by $A = B \Leftrightarrow \mu_A(x) = \mu_B(x)$ for all x.

or, employing $\wedge$ instead of min,

$$\mu_{L_1 \cap L_2} = \mu_{L_1} \wedge \mu_{L_2}. \tag{5}$$

The *concatenation* of L_1 and L_2 is a fuzzy language denoted by $L_1 L_2$ and defined as follows. Let a string x in $V_T{}^*$ be expressed as a concatenation of a prefix string u and a suffix string v, that is, $x = uv$. Then

$$\mu_{L_1 L_2}(x) = \sup_u \min(\mu_{L_1}(u), \mu_{L_2}(v)), \tag{6}$$

where the supremum is taken over all prefixes u of x. Using $\vee$ and $\wedge$ in place of sup and min, we may express (6) in the somewhat simpler form

$$\mu_{L_1 L_2}(x) = \bigvee_u (\mu_{L_1}(u) \wedge \mu_{L_2}(v)). \tag{7}$$

It is of interest to note that (7) takes on the appearance of a convolution when v is expressed as $x - u$, with $\vee$ corresponding to the sum (integral) and $\wedge$ to the product:

$$\mu_{L_1 L_2}(x) = \bigvee_u (\mu_{L_1}(u) \wedge \mu_{L_2}(x - u)).$$

Note also that (7) implies—by virtue of the distributivity of $\vee$ and $\wedge$—that concatenation has the associative property.

Having defined the union and concatenation, we can readily extend the notion of *Kleene closure* to fuzzy languages. Specifically, denoting the Kleene closure of L by L^*, we have as the definition of L^*,

$$L^* = \epsilon + L + LL + LLL + LLLL + \cdots, \tag{8}$$

where ϵ is the null string. Note that the meaning of the multiple concatenations $LLL, LLLL, \ldots$ is unambiguous because of the associativity of concatenation.

Fuzzy grammar

Informally, a *fuzzy grammar* may be viewed as a set of rules for generating the elements of a fuzzy set.† More concretely, a *fuzzy grammar*, or simply a *grammar*, is a quadruple $G = (V_N, V_T, P, S)$ in which V_T is a set of terminals, V_N is a set of non-terminals ($V_T \cap V_N = \phi$), P is a set of fuzzy productions, and $S \in V_N$. Essentially, the elements of V_N are labels for certain fuzzy subsets of $V_T{}^*$ called *fuzzy syntactic categories*, with S being the label for the syntactic category "sentence." The elements of P define conditioned fuzzy sets‡ in $(V_T \cup V_N)^*$.

More specifically, the elements of P are expressions of the form

$$\mu(\alpha \to \beta) = \rho, \qquad \rho > 0, \tag{9}$$

where α and β are strings in $(V_T \cup V_N)^*$ and ρ is the grade of membership of

† Note that an element of a fuzzy set, L, is an ordered pair of the form $(x, \mu_A(x))$.

‡ A fuzzy set conditioned on α is a fuzzy set whose membership function depends on α as a parameter.

β given α. Where convenient, we shall abbreviate $\mu(\alpha \to \beta) = \rho$ to $\alpha \xrightarrow{\rho} \beta$ or, more simply, $\alpha \to \beta$.

As in the case of non-fuzzy grammars, the expression $\alpha \to \beta$ represents a rewriting rule. Thus, if $\alpha \xrightarrow{\rho} \beta$ and γ and δ are arbitrary strings in $(V_T \cup V_N)^*$, then

$$\gamma\alpha\delta \xrightarrow{\rho} \gamma\beta\delta \tag{10}$$

and $\gamma\beta\delta$ is said to be *directly derivable from* $\gamma\alpha\delta$.

If $\alpha_1, \ldots, \alpha_m$ are strings in $(V_T \cup V_N)^*$ and $\alpha_1 \xrightarrow{\rho_2} \alpha_2, \ldots, \alpha_{m-1} \xrightarrow{\rho_m} \alpha_m$, $\rho_2, \ldots, \rho_m > 0$, then α_1 is said to *derive* α_m in grammar G, or, equivalently, α_m is *derivable* from α_1 in grammar G. This is expressed by $\alpha_1 \underset{G}{\Rightarrow} \alpha_m$ or simply $\alpha_1 \Rightarrow \alpha_m$. The expression

$$\alpha_1 \xrightarrow{\rho_2} \alpha_2 \cdots \alpha_{m-1} \xrightarrow{\rho_m} \alpha_m \tag{11}$$

will be referred to as a *derivation chain from* α_1 *to* α_m.

A fuzzy grammar G generates a fuzzy language $L(G)$ in the following manner. A string of terminals x is said to be *in* $L(G)$ if and only if x is derivable from S. The grade of membership of x in $L(G)$ is given by

$$\mu_G(x) = \sup \min(\mu(S \to \alpha_1), \mu(\alpha_1 \to \alpha_2), \ldots, \mu(\alpha_m \to x)), \tag{12}$$

where $\mu_G(x)$ is an abbreviation for $\mu_{L(G)}(x)$ and the supremum is taken over all derivation chains from S to x. Thus, (12) defines $L(G)$ as a fuzzy set in $(V_T \cup V_N)^*$. If $L(G_1) = L(G_2)$ in the sense of equality of fuzzy sets, then the grammars G_1 and G_2 are said to be *equivalent*.

It is helpful to observe that if a production $\alpha \xrightarrow{\rho} \beta$ is visualized as a chain link of strength ρ, then the strength of a derivation chain $\alpha_1 \xrightarrow{\rho_2} \alpha_2 \cdots \xrightarrow{\rho_m} \alpha_m$ is the strength of its weakest link, that is $\min(\rho_2, \ldots, \rho_m)$. Then, the defining equation (12) may be expressed in words as:

$\mu_G(x)$ = grade of membership of x in the language generated by grammar G

= strength of the strongest derivation chain from S to x.

Let $\mu(S \to \alpha_1) = \rho_1$, $\mu(\alpha_1 \to \alpha_2) = \rho_2$, $\ldots$, $\mu(\alpha_m \to x) = \rho_{m+1}$. Then, on writing (12) in the form

$$\mu_G(x) = \vee(\rho_1 \wedge \rho_2 \cdots \wedge \rho_{m+1}) \tag{13}$$

it follows at once from the associativity of $\wedge$ that (12) is equivalent to a more general expression in which the successive α's are derivable from their immediate predecessors rather than *directly* derivable from them, as in (12).

Example. Suppose that $V_T = \{0, 1\}$, $V_N = \{A, B, S\}$, and P is given by

$$\begin{array}{rl} \mu(S \to AB) = 0.5 & \mu(A \to 0) = 0.5 \\ \mu(S \to A) = 0.8 & \mu(A \to 1) = 0.6 \\ \mu(S \to B) = 0.8 & \mu(B \to A) = 0.4 \\ \mu(AB \to BA) = 0.4 & \mu(B \to 0) = 0.2. \end{array}$$

Consider the terminal string $x = 0$. The possible derivation chains for this string are $S \xrightarrow{0.8} A \xrightarrow{0.5} 0$, $S \xrightarrow{0.8} B \xrightarrow{0.2} 0$, and $S \xrightarrow{0.8} B \xrightarrow{0.4} A \xrightarrow{0.5} 0$. Hence

$$\mu_L(0) = \max(\min(0.8, 0.5),\ \ \min(0.8, 0.2),\ \ \min(0.8, 0.4, 0.5)) = 0.5.$$

Similarly, the possible derivation chains for the terminal string $x = 01$ are $S \xrightarrow{0.5} AB \xrightarrow{0.5} 0B \xrightarrow{0.4} 0A \xrightarrow{0.6} 01$, $S \xrightarrow{0.5} AB \xrightarrow{0.4} AA \xrightarrow{0.5} 0A \xrightarrow{0.6} 01$, $S \xrightarrow{0.5} AB \xrightarrow{0.4} BA \xrightarrow{0.2} 0A \xrightarrow{0.6} 01$, and $S \xrightarrow{0.5} AB \xrightarrow{0.4} BA \xrightarrow{0.4} AA \xrightarrow{0.5} 0A \xrightarrow{0.6} 01$. Hence,

$$\mu_G(01) = \max(0.4, 0.4, 0.2, 0.4) = 0.4.$$

Given a grammar G, an important question which arises in connection with the definition of $L(G)$ is whether or not there exists an algorithm for computing $\mu_G(x)$ by the use of the defining equation (12). If such an algorithm exists, then G is said to be *recursive*.

3. TYPES OF GRAMMARS

Paralleling the standard classification of non-fuzzy grammars, we can distinguish four principal types of fuzzy grammars.

Type 0 grammar

In this case, productions are of the general form $\alpha \xrightarrow{\rho} \beta$, $\rho > 0$, where α and β are strings in $(V_T \cup V_N)^*$.

Type 1 grammar (context-sensitive)

Here the productions are of the form $\alpha_1 A \alpha_2 \xrightarrow{\rho} \alpha_1 \beta \alpha_2$, $\rho > 0$, with α_1, α_2, and β in $(V_T \cup V_N)^*$, A in V_N, and $\beta \neq \epsilon$. In addition, the production $S \to \epsilon$ is allowed.

Type 2 grammar (context-free)

The allowable productions are of the form $A \xrightarrow{\rho} \beta$, $\rho > 0$, $A \in V_N$, $\beta \in (V_T \cup V_N)^*$, $\beta \neq \epsilon$, and $S \to \epsilon$.

Type 3 grammar (regular)

In this case the allowable productions are of the form $A \xrightarrow{\rho} aB$ or $A \xrightarrow{\rho} a$, $\rho > 0$, where $a \in V_T$, A, $B \in V_N$. In addition, $S \to \epsilon$ is allowed.

In what follows, we shall focus our attention on context-free grammars. However, there is one basic property of context-sensitive grammars that needs stating at this point. Specifically, in defining $L(G)$ we have mentioned that G is recursive if there exists an algorithm for computing $\mu_G(x)$. It is easy to demonstrate that context-sensitive—and hence also context-free and regular—grammars are recursive. This can be stated as an extension of Theorem 2.2 in Hopcroft and Ullman [4].

THEOREM. *If* $G = (V_N, V_T, P, S)$ *is a fuzzy context-sensitive grammar, then* G *is recursive.*

Proof. First we show that for any type of grammar the supremum in (12) may be taken over a subset of the set of all derivation chains from S to x, namely, the subset of all *loop-free* derivation chains. These are chains in which no α_i, $i = 1, \ldots, m$, occurs more than once.

For suppose that in a derivation chain C,

$$C = S \xrightarrow{\rho_1} \alpha_1 \xrightarrow{\rho_2} \alpha_2 \cdots \xrightarrow{\rho_m} \alpha_m \xrightarrow{\rho_{m+1}} x,$$

α_i, say, is the same as α_j, $j > i$. Now consider the chain C' resulting from replacing the subchain $\alpha_i \xrightarrow{\rho_{i+1}} \cdots \xrightarrow{\rho_j} \alpha_j \xrightarrow{\rho_{j+1}} \alpha_{j+1}$ in C by $\alpha_i \xrightarrow{\rho_{j+1}} \alpha_{j+1}$. Clearly, if C is a derivation chain from S to x, so is C'. But

$$\min(\rho_1, \ldots, \rho_i, \rho_{i+1}, \ldots, \rho_{j+1}, \ldots, \rho_{m+1}) \leqslant \min(\rho_1, \ldots, \rho_i, \rho_{j+1}, \ldots, \rho_{m+1})$$

and hence C may be deleted without affecting the supremum in (12). Consequently, we can replace the definition (12) for $\mu_L(x)$ by

$$\mu_G(x) = \sup \min(\mu(S \to \alpha_1), \mu(\alpha_1 \to \alpha_2), \ldots, \mu(\alpha_m \to x)), \tag{14}$$

where the supremum is taken over all loop-free derivation chains from S to x.

Next we show that for context-sensitive grammars the set over which the supremum is taken in (14) can be further restricted to derivation chains of bounded length l_0, where l_0 depends on $|x|$ (length of x) and the number of symbols in $V_T \cup V_N$.

Specifically, if G is context-sensitive, then because of the non-contracting character of productions in P, we have

$$|\alpha_j| \geqslant |\alpha_i| \qquad \text{if } j > i. \tag{15}$$

Now let k be the number of symbols in $V_T \cup V_N$. Since there are at most k^l distinct strings in $(V_T \cup V_N)^*$ of length l, and since the derivation chain is loop-free, (15) implies that the total length of the chain is bounded by

$$l_0 = 1 + k + \cdots + k^{|x|}.$$

To complete the proof, we have to exhibit a way for generating all finite derivation chains from S to x of length $\leqslant l_0$. For this purpose, we start with S

and using P generate the set Q_1 of all strings in $(V_T \cup V_N)^*$ of length $\leqslant|x|$ which are derivable from S in one step (that is, are directly derivable from S). Then, we construct Q_2—the set of all strings in $(V_T \cup V_N)^*$ of length $\leqslant|x|$ which are derivable from S in two steps—by noting that Q_2 is identical with the set of all strings in $(V_T \cup V_N)^*$ of length $\leqslant|x|$ which are directly derivable from strings in Q_1. Continuing this process, we construct consecutively Q_3, Q_4, …, Q_r until $r = l_0$ or $Q_r = \phi$, whichever happens first. Since the Q_λ, $\lambda = 1, \ldots, r$, are finite sets, their knowledge enables us to find in a finite number of steps all loop-free derivation chains from S to x of length $\leqslant l_0$ and thus to compute $\mu_G(x)$ by the use of (14). This, then, constitutes an algorithm—though not necessarily an efficient one—for the computation of $\mu_G(x)$. Consequently, G is recursive.

4. FUZZY CONTEXT-FREE GRAMMARS

As was stated in the introduction, many of the basic results in the theory of formal languages can readily be extended to fuzzy languages. As an illustration, we shall sketch—without giving proofs—such extensions in the case of the Chomsky and Greibach normal forms for context-free languages.

Chomsky normal form for fuzzy context-free languages

Let G be a fuzzy context-free grammar. Then, any such grammar is equivalent to a grammar G_c in which all productions are of the form $A \xrightarrow{\rho} BC$ or $A \xrightarrow{\rho} a$, where $\rho > 0$; A, B, C are non-terminals, and a is a terminal.

It is convenient to effect the construction of G_c in three stages, as follows.

First, we construct a grammar G_1 equivalent to G in which there are no productions of the form $A \to B$, A, $B \in V_N$.

Thus, suppose that in G we have productions of the form $A \to B$, which lead to derivation chains of the form

$$A \xrightarrow{\rho_1} B_1 \xrightarrow{\rho_2} B_2 \cdots B_m \xrightarrow{\rho_{m+1}} B \xrightarrow{\rho_{m+2}} \alpha,$$

where $\alpha \notin V_N$. Then we replace all such productions: $A \xrightarrow{\rho_1} B_1$, $B_1 \xrightarrow{\rho_2} B_2$, …, $B_m \xrightarrow{\rho_{m+1}} B$ in G by single productions of the form $A \xrightarrow{\rho} \alpha$, in which

$$\rho = \min(\mu(A \Rightarrow B), \mu(B \to \alpha)) \tag{16}$$

where

$$\mu(A \Rightarrow B) = \sup \min(\mu(A \to B_1), \ldots, \mu(B_m \to B)) \tag{17}$$

with the supremum taken over all loop-free derivation chains from A to B. It can readily be shown that the resultant grammar, G_1, is equivalent to G.

Second, we construct a grammar G_2 equivalent to G_1 in which there are no productions of the form $A \xrightarrow{\rho} B_1 B_2 \cdots B_m$, $\rho > 0$, $m > 2$, in which one or more of the B's are terminals. Thus, suppose that B_i, say, is a terminal a. Then B_i in $B_1 B_2 \cdots B_m$ is replaced by a new non-terminal C_i which does not appear on the right-hand side of any other production, and we set

$$\mu(A \to B_1 B_2 \cdots B_i \cdots B_m) = \mu(A \to B_1 B_2 \cdots C_i \cdots B_m) \tag{18}$$

Furthermore, we add to the productions of G the production $C_i \xrightarrow{1} a$. On doing this for all terminals in $B_1 \cdots B_m$ in all productions of the form $A \to B_1 \cdots B_m$, we arrive at a grammar G_2 in which all productions are of the form $A \to a$ or $A \to B_1 \cdots B_m$, $m \geqslant 2$, where all the B's are non-terminals. It is evident that G_2 is equivalent to G_1.

Third, we construct a grammar G_3 equivalent to G_2 in which all productions are of the form $A \to a$ or $A \to BC$, $A,B,C \in V_N$, $a \in V_T$. To this end, consider a typical production in G_2 of the form $A \xrightarrow{\rho} B_1 \cdots B_m$, $\rho > 0$, $m > 2$. Replace this production by the productions

$$\begin{aligned} A &\xrightarrow{\rho} B_1 D_1 \\ D_1 &\xrightarrow{1} B_2 D_2 \\ D_{m-2} &\xrightarrow{1} B_{m-1} B_m, \end{aligned} \tag{19}$$

where the D's are new non-terminals which do not appear on the right-hand side of any production in G_2. On performing such replacements for all productions in G_2 of the form $A \xrightarrow{\rho} B_1 \cdots B_m$, we obtain a grammar G_3 which is equivalent to G_2. This establishes that G_3—which is in Chomsky normal form—is equivalent to G.

Example. Consider the following fuzzy grammar in which $V_T = \{a,b\}$ and $V_n = \{A, B, S\}$.

$$\begin{array}{ll} S \xrightarrow{0.8} bA & B \xrightarrow{0.4} b \\ S \xrightarrow{0.6} aB & A \xrightarrow{0.3} bSA \\ A \xrightarrow{0.2} a & B \xrightarrow{0.5} aSB. \end{array}$$

To find the equivalent grammar in Chomsky normal form, we proceed as follows.

First, $S \xrightarrow{0.8} bA$ is replaced by $S \xrightarrow{0.8} C_1 A$, $C_1 \xrightarrow{1} b$. Similarly, $S \xrightarrow{0.6} aB$ is replaced by $S \xrightarrow{0.6} C_2B$ and $C_2 \xrightarrow{1} a$. Also $A \xrightarrow{0.3} bSA$ is replaced by $A \xrightarrow{0.3} C_3SA$, $C_3 \xrightarrow{1} b$; $B \xrightarrow{0.5} aSB$ is replaced by $B \xrightarrow{0.5} C_4SB$ and $C_4 \xrightarrow{1} a$.

Second, the production $A \xrightarrow{0.3} C_3SA$ is replaced by $A \xrightarrow{0.3} C_3D_1$, $D_1 \xrightarrow{1} SA$; and the production $B \xrightarrow{0.5} C_4SB$ is replaced by $B \xrightarrow{0.5} C_4D_2$, $D_2 \xrightarrow{1} SB$. Thus, the productions in the equivalent Chomsky normal form read:

$$
\begin{array}{ll}
S \xrightarrow{0.8} C_1A & A \xrightarrow{0.3} C_3D_1 \\
C_1 \xrightarrow{1} b & D_1 \xrightarrow{1} SA \\
S \xrightarrow{0.6} C_2B & C_3 \xrightarrow{1} b \\
C_2 \xrightarrow{1} a & B \xrightarrow{0.5} C_4D_2 \\
A \xrightarrow{0.2} a & D_2 \xrightarrow{1} SB \\
B \xrightarrow{0.4} b & C_4 \xrightarrow{1} a.
\end{array}
$$

Greibach normal form

As in the case of the Chomsky normal form, let G be any fuzzy context-free grammar. Then G is equivalent to a fuzzy grammar G_G in which all productions are of the form $A \to a\alpha$, where A is a non-terminal, a is a terminal and α is a string in V_N^*. The fuzzy grammar G_G is in *Greibach normal form.*

Paralleling the approach used in Ullman and Hopcroft, it is convenient to state two lemmas which are of use in constructing G_G. We shall omit proofs of these lemmas since their validity is reasonably evident and their detailed proofs fairly long.

LEMMA 1. *Let G be a fuzzy context-free grammar. Let $A \to \alpha_1 B\alpha_2$ be a production in P, $A, B \in V_N$, and $\alpha_1, \alpha_2 \in (V_T \cup V_N)^*$. Furthermore, let $B \to \beta_1, \ldots, B \to \beta_r$ be the set of all B-productions (that is, all productions with B on the left-hand side). Let G_1 be the grammar resulting from the replacement of each of the productions of the form $A \to \alpha_1 B\alpha_2$ with the productions $A \to \alpha_1\beta_1\alpha_2, \ldots, A \to \alpha_1\beta_r\alpha_2$, in which*

$$\mu(A \to \alpha_1\beta_i\alpha_2) = \min(\mu(A \to \alpha_1 B\alpha_2), \mu(B \to \beta_i)), \qquad i = 1, \ldots, r. \quad (20)$$

Then G_1 is equivalent to G.

LEMMA 2. *Let G be a fuzzy context-free grammar. Let $A \to A\alpha_i$, $i = 1, \ldots, r$, be the set of A-productions for which A is also the leftmost symbol on the right-hand side. Furthermore, let $A \to \beta_j$, $j = 1, \ldots, s$ be the remaining A-productions,* *with $\alpha_i, \beta_j \in (V_T \cup V_N)^*$, $i = 1, \ldots, r, j = 1, \ldots, s$. Let G_2 be the grammar resulting from the replacement of the $A \to A\alpha_i$ in G with the productions:*

$$A \to \beta_j Z, \qquad j = 1, \ldots, s \tag{21}$$

$$Z \to \alpha_i, Z \to \alpha_i Z, \qquad i = 1, \ldots, r, \tag{22}$$

where

$$\begin{aligned} \mu(A \to \beta_j Z) &= \mu(A \to \beta_j), \qquad i = 1, \ldots, r \\ \mu(Z \to \alpha_i) &= \mu(A \to A\alpha_i) \\ \mu(Z \to \alpha_i Z) &= \mu(A \to A\alpha_i). \end{aligned} \tag{23}$$

Making use of these lemmas, the Greibach normal form for G can be derived as follows.

First, G is put into the Chomsky normal form. Let the non-terminals in this form be denoted by $A_1, \ldots, A_m$.

Second, the productions of the form $A_i \to A_j\gamma$, $\gamma \in (V_T \cup V_N)^*$, are modified in such a way that for all such productions $j \geqslant i$. This is done in stages. Thus, suppose that it has been done for $i \leqslant k$, that is, if

$$A_i \to A_j \gamma \tag{24}$$

is a production with $i \leqslant k$, then $j > i$. To extend this to A_{k+1}-productions suppose that $A_{k+1} \to A_j\gamma$ is any production with $j < k+1$. Using Lemma 1 and substituting for A_j the right-hand side of each A_j-production, we obtain by repeated substitution productions of the form

$$A_{k+1} \to A_l \gamma, \qquad l \geqslant k+1. \tag{25}$$

In (25), those productions in which l is equal to $k+1$ are replaced by the application of Lemma 2, resulting in a new non-terminal Z_{k+1}. Then, by repetition of this process all productions are put into the form

$$A_k \to A_l \gamma, \qquad l > k, \gamma \in (V_N \cup \{Z_1, \ldots, Z_n\})^* \tag{26}$$

$$A_k \to a\gamma, \qquad a \in V_T \tag{27}$$

$$Z_k \to \gamma \tag{28}$$

with membership grades given by Lemmas 1 and 2.

In view of (26) and (27), the leftmost symbol on the right-hand side of any production for A_m must be a terminal. Similarly, for A_{m-1}, the leftmost symbol on the right-hand side must be either A_m or a terminal. Substituting for A_m by Lemma 1, we obtain productions whose right-hand sides start with terminals. Repeating this process for $A_{m-2}, \ldots, A_1$, we eventually put all productions for the A_i, $i = 1, \ldots, m$, into a form where their right-hand sides start with terminals.

At this stage, only the productions in (28) may not be in the desired form.

We observe that the leftmost symbol in γ in (28) may be either a terminal or one of the A_i, $i = 1, \ldots, m$. If it is the latter, application of Lemma 1 to each Z_i production results in productions of the desired form and thus completes the construction.

Example. As a simple illustration, we shall convert into the Greibach normal form a fuzzy grammar G in which $V_T = \{a,b\}$, $V_N = \{A_1,A_2,A_3\}$ and the productions are in the Chomsky normal form:

$$\begin{array}{ll} A_1 \xrightarrow{0.8} A_2A_3 & A_3 \xrightarrow{0.2} A_1A_2 \\ A_2 \xrightarrow{0.7} A_3A_1 & A_3 \xrightarrow{0.5} a. \\ A_2 \xrightarrow{0.6} b & \end{array}$$

Step 1. Since the right-hand sides of the productions for A_1 and A_2 start with terminals or higher-numbered variables, we begin with the production $A_3 \to A_1A_2$ and substitute A_2A_3 for A_1. Note that $A_1 \to A_2A_3$ is the only production with A_1 on the left.

The resulting set of productions is

$$\begin{array}{ll} A_1 \xrightarrow{0.8} A_2A_3 & A_2 \xrightarrow{0.7} A_3A_1 \\ A_2 \xrightarrow{0.6} b & A_3 \xrightarrow{0.2} A_2A_3A_2 \\ A_3 \xrightarrow{0.5} a & \end{array}$$

Note that in $A_3 \xrightarrow{0.2} A_2A_3A_2$, 0.2 is $0.8 \wedge 0.2$.

Since the right-hand side of the production $A_3 \to A_2A_3A_2$ begins with a lower-numbered variable, we substitute for the first occurrence of A_2 either A_3A_1 or b.

The new set is

$$\begin{array}{ll} A_1 \xrightarrow{0.8} A_2A_3 & A_2 \xrightarrow{0.7} A_3A_1 \\ A_2 \xrightarrow{0.6} b & A_3 \xrightarrow{0.2} A_3A_1A_3A_2. \\ A_3 \xrightarrow{0.2} bA_3A_2 & A_3 \xrightarrow{0.5} a \end{array}$$

We now apply Lemma 2 to the productions $A_1 \to A_3A_1A_3A_2$, $A_3 \to bA_3A_2$, and $A_3 \to a$. We introduce Z_3 and replace the production $A_3 \to A_3A_1A_3A_2$ by $A_3 \to bA_3A_2Z_3$, $A_3 \to aZ_3$, $Z_3 \to A_1A_3A_2$, and $Z_3 \to A_1A_3A_2Z_3$.

The resulting set is

$$A_1 \xrightarrow{0.8} A_2 A_3 \qquad A_2 \xrightarrow{0.7} A_3 A_1 \qquad A_3 \xrightarrow{0.2} bA_3 A_2 Z_3.$$

$$A_2 \xrightarrow{0.6} b \qquad A_3 \xrightarrow{0.2} bA_3 A_2$$

$$A_3 \xrightarrow{0.5} a \qquad A_3 \xrightarrow{0.5} aZ_3$$

$$Z_3 \xrightarrow{0.2} A_1 A_3 A_2 Z_3 \qquad Z_3 \xrightarrow{0.2} A_1 A_3 A_2$$

Step 2. Now all productions with A_3 on the left have right-hand sides that start with terminals. These are used to replace A_3 in the production $A_2 \to A_3 A_1$ and then the productions with A_2 on the left are used to replace A_2 in the production $A_1 \to A_2 A_3$. The resulting set is

$$A_3 \xrightarrow{0.2} bA_3 A_2 \qquad A_3 \xrightarrow{0.2} bA_3 A_2 Z_3$$

$$A_3 \xrightarrow{0.5} a \qquad A_3 \xrightarrow{0.5} aZ_3$$

$$A_2 \xrightarrow{0.2} bA_3 A_2 A_1 \qquad A_2 \xrightarrow{0.2} bA_3 A_2 Z_3 A_1$$

$$A_2 \xrightarrow{0.5} aA_1 \qquad A_2 \xrightarrow{0.5} aZ_3 A_1$$

$$A_2 \xrightarrow{0.6} b \qquad A_1 \xrightarrow{0.2} bA_3 A_2 A_1 A_3$$

$$A_1 \xrightarrow{0.2} bA_3 A_2 Z_3 A_1 A_3 \qquad A_1 \xrightarrow{0.5} aA_1 A_3$$

$$A_1 \xrightarrow{0.6} bA_3 \qquad Z_3 \xrightarrow{0.2} A_1 A_3 A_2 Z_3.$$

$$Z_3 \xrightarrow{0.2} A_1 A_3 A_2$$

Note that in $A_2 \xrightarrow{0.2} bA_3 A_2 A_1$, 0.2 is $0.7 \wedge 0.2$. Likewise, in $A_1 \xrightarrow{0.5} aA_1 A_3$, 0.5 is $0.5 \wedge 0.8$. The same holds for the way in which the membership grades of other productions are determined (by the use of Lemmas 1 and 2).

Step 3. The two Z_3 productions, $Z_3 \xrightarrow{0.2} A_1 A_3 A_2$ and $Z_3 \xrightarrow{0.2} A_1 A_3 A_2 Z_3$, are converted to desired form by substituting the right-hand side of each of the five productions with A_1 on the left for the first occurrence of A_1. Thus, $Z_3 \xrightarrow{0.2} A_1 A_3 A_2$ is replaced by

$$Z_3 \xrightarrow{0.2} bA_3 A_3 A_2 \qquad Z_3 \xrightarrow{0.2} bA_3 A_2 A_1 A_3 A_3 A_2$$

$$Z_3 \xrightarrow{0.2} aA_1 A_3 A_3 A_2 \qquad Z_3 \xrightarrow{0.2} bA_3 A_2 Z_3 A_1 A_3 A_3 A_2.$$

$$Z_3 \xrightarrow{0.2} aZ_3 A_1 A_3 A_3 A_2$$

The other production for Z_3 is converted similarly. The final set of productions reads

$$
\begin{array}{ll}
A_3 \xrightarrow{0.2} bA_3A_2 & A_3 \xrightarrow{0.2} bA_3A_2Z_3 \\
A_3 \xrightarrow{0.5} a & A_3 \xrightarrow{0.5} aZ_3 \\
A_2 \xrightarrow{0.2} bA_3A_2A_1 & A_2 \xrightarrow{0.2} bA_3A_2Z_3A_1 \\
A_2 \xrightarrow{0.5} aA_1 & A_2 \xrightarrow{0.5} aZ_3A_1 \\
A_2 \xrightarrow{0.6} a & A_1 \xrightarrow{0.2} bA_3A_2A_1A_3 \\
A_1 \xrightarrow{0.2} bA_3A_2Z_3A_1A_3 & A_1 \xrightarrow{0.5} aA_1A_3 \\
A_1 \xrightarrow{0.5} aZ_3A_1A_3 & A_1 \xrightarrow{0.6} bA_3 \\
Z_3 \xrightarrow{0.2} bA_3A_3A_2 & Z_3 \xrightarrow{0.2} bA_3A_3A_2Z_3 \\
Z_3 \xrightarrow{0.2} bA_3A_2A_1A_3A_3A_2 & Z_3 \xrightarrow{0.2} bA_3A_2A_1A_3A_3A_2Z_3 \\
Z_3 \xrightarrow{0.2} aA_1A_3A_3A_2 & Z_3 \xrightarrow{0.2} aA_1A_3A_3A_2Z_3 \\
Z_3 \xrightarrow{0.2} bA_3A_2Z_3A_1A_3A_3A_2 & Z_3 \xrightarrow{0.2} bA_3A_2Z_3A_1A_3A_3A_2Z_3 \\
Z_3 \xrightarrow{0.2} aZ_3A_1A_3A_3A_2 & Z_3 \xrightarrow{0.2} aZ_3A_1A_3A_3A_2Z_3.
\end{array}
$$

This concludes our brief description of the construction of the Chomsky and Greibach normal forms for a fuzzy context-free grammar.

As was stated in the introduction, the purpose of the present note is merely to illustrate by a few examples how some of the basic concepts and results in the theory of formal languages can be extended to fuzzy languages. As the reader can see, the extensions are, for the most part, quite straightforward and require hardly any modifications in the statements of lemmas and theorems. The proofs, however, are generally somewhat longer since they involve not just the positivity of membership functions but their values in the interval [0, 1].

The theory of fuzzy languages offers what appears to be a fertile field for further study. It may prove to be of relevance in the construction of better models for natural languages and may contribute to a better understanding of the role of fuzzy algorithms and fuzzy automata in decision making, pattern recognition, and other processes involving the manipulation of fuzzy data.

REFERENCES

1 P. Turakainen, On stochastic languages, *Information and Control*, **12** (April 1968), 304–313.
2 M. M. Kherts, Entropy of languages generated by automated or context-free grammars with a single-valued deduction, *Nauchno-Tekhnicheskaya Informatsya*, Ser. 2, No. 1 (1968).
3 K. S. Fu and T. J. Li, On stochastic automata and languages, *3rd Conference on Information and System Sciences, Princeton University*, 1969, pp. 338–377.
4 J. E. Hopcroft and J. D. Ullman, *Formal Languages and Their Relation to Automata*, Addison-Wesley, Reading, Mass., 1969.
5 L. A. Zadeh, Fuzzy sets, *Information and Control*, **8** (June 1965), 338–353.
6 L. A. Zadeh, Toward a Theory of Fuzzy Systems, *ERL Rept*. No. 69-2, *Electronics Research Laboratories, University of California*, Berkeley, June 1969.
7 J. Goguen, L-fuzzy sets, *J. Math. Anal. Appl.*, **18** (April 1967), 145–174.
8 J. G. Brown, Fuzzy sets on Boolean lattices, *Rept*. No. 1957, *Ballistic Research Laboratories*, Aberdeen, Md., Jan. 1969.

Received August 11, 1969