CHAPTER 4

VAGUE REGULAR LANGUAGE

4.1 INTRODUCTION

In this chapter, we tried to generalize the grade of membership function $\mu_L(w)$ of fuzzy language. Here, we use the concept of vague sets to overcome the difficulty of single membership value of strings in fuzzy language [46]. The single membership value $\mu_L(w)$ in $[0, 1]$ combines the evidence for $w \in \Sigma^*$ and the evidence against $w \in \Sigma^*$, without indicating how much of each is there. This single number tells us nothing about its accuracy. The grade of membership value of each string in fuzzy language is replaced by a continuous subinterval of $[0, 1]$ by applying vague sets. This subinterval keeps track of both the favoring evidence and the opposing evidence. We call the favoring evidence a truth membership value and the opposing evidence a false membership value. These are obtained from the truth-membership function $t_L(w)$ and the false-membership function $f_L(w)$ respectively, to record the lower bounds on $\mu_L(w), w \in \Sigma^*$. These lower bounds are used to create a subinterval of $[0, 1]$, denoted by $[t_L(w), 1-f_L(w)]$. We call the resulting language a vague language, each string of which is having the membership $[t_L(w), 1-f_L(w)]$. Here, $t_L(w)$ and $1-f_L(w)$ represents, the lower bound and the upper bound of this subinterval respectively. These interval bounds may be derived on $\mu_L(w)$ and $\mu_L^c(w)$ (complement of $\mu_L(w)$) as follows.

1. $t_L(w) \leq \mu_L(w)$
2. $f_L(w) \leq \mu_L^c(w)$

$\Leftrightarrow f_L(w) \leq 1-\mu_L(w) \Leftrightarrow 1-f_L(w) \geq \mu_L(w)$.

From (1) and (2), we obtain $t_L(w) \leq \mu_L(w) \leq 1-f_L(w)$.

It is clear that the interval is a subinterval of $[0, 1]$ since 0 is a lower bound of all membership functions and $t_L(w) \leq \mu_L(w), f_L(w) \leq \mu_L^c(w)$ $\Leftrightarrow t_L(w) + f_L(w) \leq \mu_L(w) + \mu_L^c(w) \leq 1$.

The main contributions of this chapter are five fold.

1. The definitions of vague language and vague regular language are proposed and
studied their related properties.

2. The acceptance of vague regular language through the finite automata (DFA and NDFA) with vague transitions and vague (final) states is discussed. These automata are the pessimistic part of the extended fuzzy automata and these carry truth & false membership values in their transition. By neglecting the false membership value, one can see them working as fuzzy automata.

3. Vague regular expression is defined.

4. Myhill-Nerode theorem is extended for vague regular language and minimization of vague (final) states automata is achieved through an algorithm.

5. Relation between the membership values of intuitionistic fuzzy language, interval-valued fuzzy and vague language is obtained.

4.2 VAGUE REGULAR LANGUAGE

Definition 4.2.1: Let $\Sigma$ be an alphabet. Then we call the set

$$\tilde{L} = \{(w, [t_{\tilde{L}}(w), 1 - f_{\tilde{L}}(w)]) \mid w \in \Sigma^* \}$$

a vague language (VL) over $\Sigma$, where

$$t_{\tilde{L}}(w) : \Sigma^* \rightarrow [0, 1] \text{ and } f_{\tilde{L}}(w) : \Sigma^* \rightarrow [0, 1]$$

represents respectively, the truth membership and the false membership functions of $\tilde{L}$ such that, $0 \leq t_{\tilde{L}}(w) \leq 1 - f_{\tilde{L}}(w) \leq 1$ or $0 \leq t_{\tilde{L}}(w) + f_{\tilde{L}}(w) \leq 1$. These languages are generated by vague fuzzy grammars.

Definition 4.2.2: A vague fuzzy grammar (VFG) is defined as a 4-tuple $G = (V_N, V_T, P, S)$, where

a) $V_T$ is the finite set of alphabets called terminal symbols,

b) $V_N$ is the finite set of non-terminal symbols ($V_N \cap V_T = \emptyset$),

c) $P$ is a list of production (rewrite) rules, and
d) $S$ is an initial symbol included in $V_N$.

The elements of $P$ are expressions of the form

$$\mu(\alpha \rightarrow \beta) = [\tau, 1 - \omega], \text{ for } \tau, \omega \in [0, 1] \text{ and } 0 \leq \tau + \omega \leq 1,$$

where $\alpha$ and $\beta$ are strings in

$$(V_N \cup V_T)^*$$

and $\tau, \omega$ represents respectively, the truth membership and false membership of $\beta$ for a given $\alpha$.

For notational ease we abbreviate
\[ \mu(\alpha \rightarrow \beta) = [\tau, 1 - \omega] \text{ to } \alpha^{[\tau,1-\omega]} \rightarrow \beta \text{ or, more simply, } \alpha \rightarrow \beta. \]

**Vague fuzzy regular grammar** (VFRG) allows the productions of the form

\[ A^{[\tau,1-\omega]} \rightarrow AB \text{ or } A^{[\tau,1-\omega]} \rightarrow a, A, B \in V_N, a \in V_T, 0 \leq \tau, \omega \leq 1. \]

In addition, the production \( S \rightarrow \varepsilon \) is allowed with the truth membership and false membership values 1 & 0 respectively.

**Note:** The precision of our knowledge about the belongingness of the string ‘\( w \)’ in ‘\( \tilde{L} \)’ is noticeable with the difference \( 1 - f_{\tilde{L}}(w) - t_{\tilde{L}}(w) \) is described below:

1. If this difference is small; our knowledge about ‘\( w \)’ is relatively exact,
2. If this difference is large; our knowledge about ‘\( w \)’ is meager,
3. If \( 1 - f_{\tilde{L}}(w) = t_{\tilde{L}}(w); \) our knowledge about ‘\( w \)’ is exact and the theory reverts back to that of fuzzy languages.

**Example 4.1:** If \( \Sigma = \{a, b\} \) and \( t_{\tilde{L}}(w): \Sigma^* \rightarrow [0,1] \) and \( f_{\tilde{L}}(w): \Sigma^* \rightarrow [0,1] \) then,

\[ \tilde{L} = \{(a^*b, [0.6,0.8]), (a^*ba^*, [0.4,0.7]) \mid a^*b, a^*ba^* \in \Sigma^* \} \]

is a VL.

Let \( \tilde{L} \) be a VL over an alphabet \( \Sigma \) and \( t_{\tilde{L}}(w): \Sigma^* \rightarrow M \) and \( f_{\tilde{L}}(w): \Sigma^* \rightarrow N \) represents respectively, the truth membership and the false membership functions of \( \tilde{L} \), then for each \( m \in M \) denote by \( S_{\tilde{L}}(m) \) the set, \( S_{\tilde{L}}(m) = \{ w \mid w \in \Sigma^* \text{ and } t_{\tilde{L}}(w) = m \} \) and for each \( n \in N \), denote by \( S_{\tilde{L}}(1-n) \) the set, \( S_{\tilde{L}}(1-n) = \{ w \mid w \in \Sigma^* \text{ and } 1 - f_{\tilde{L}}(w) = 1 - n \} \).

Note that, here \( M \) and \( N \) represents the set of real numbers in \([0,1]\).

Here, \( S_{\tilde{L}}(m) = S_{\tilde{L}}(1-n) \) and \( 0 \leq m + n \leq 1 \).

**Some Algebraic Operations over Vague Languages**

Let \( \tilde{L}_1 \) and \( \tilde{L}_2 \) be two VLs over an alphabet \( \Sigma \). Let \( t_{\tilde{L}_1}, t_{\tilde{L}_2}, f_{\tilde{L}_1}, f_{\tilde{L}_2} \), be the truth and false membership functions of \( \tilde{L}_1 \) and \( \tilde{L}_2 \) respectively. Then, we can describe some of the basic algebraic operations such as union, intersection, complement, concatenation and star operations on \( \tilde{L}_1 \) and \( \tilde{L}_2 \) in the following way.
1. **Union:**

\[ \tilde{L} = \tilde{L}_1 \cup \tilde{L}_2 = \{ (w, [\max(t_{\tilde{L}_1}(w), t_{\tilde{L}_2}(w)), 1 - \min(f_{\tilde{L}_1}(w), f_{\tilde{L}_2}(w))] \mid w \in \Sigma^* \}. \]

2. **Intersection:**

\[ \tilde{L} = \tilde{L}_1 \cap \tilde{L}_2 = \{ (w, [\min(t_{\tilde{L}_1}(w), t_{\tilde{L}_2}(w)), 1 - \max(f_{\tilde{L}_1}(w), f_{\tilde{L}_2}(w))] \mid w \in \Sigma^* \}. \]

3. **Complement:** The complement of \( \tilde{L}_i \) is denoted by \( \tilde{L}_i^c \) and given as

\[ \tilde{L}_i^c = \{ (w, [f_{\tilde{L}_i}(w), 1 - t_{\tilde{L}_i}(w)) \mid w \in \Sigma^* \}. \]

4. **Concatenation:**

\[ \tilde{L} = \tilde{L}_1 \cdot \tilde{L}_2 = \{ (w, [\max\{\min(t_{\tilde{L}_1}(x), t_{\tilde{L}_2}(y))\}, 1 - \min\{\max(f_{\tilde{L}_1}(x), f_{\tilde{L}_2}(y))\}] \mid w = xy, x, y \in \Sigma^* \}, w \in \Sigma^*. \]

5. **Star:**

\[ \tilde{L} = \tilde{L}_i^* = \{ (w, [\max\{\min(t_{\tilde{L}_i}(x_1), t_{\tilde{L}_i}(x_2), \ldots, t_{\tilde{L}_i}(x_n))\}, 1 - \min\{\max(f_{\tilde{L}_i}(x_1), f_{\tilde{L}_i}(x_2), \ldots, f_{\tilde{L}_i}(x_n))\}] \mid w = x_1x_2\ldots x_n, x_1, x_2, \ldots, x_n \in \Sigma^*, n \geq 0 \}, w \in \Sigma^*. \]

assuming that \( \min \varepsilon = [0,0] \).

Hence, we have \( \tilde{L}_i^* = \bigcap_{i=0}^{\infty} \tilde{L}_i = \tilde{L}_0 \cup \tilde{L}_1 \cup \tilde{L}_2 \cup \ldots \), i.e., Kleene Closure is satisfied.

6. **The + operation on \( \tilde{L}_i \) is defined by,**

\[ \tilde{L} = \tilde{L}_i^+ = \{ (w, [\max\{\min(t_{\tilde{L}_i}(x_1), t_{\tilde{L}_i}(x_2), \ldots, t_{\tilde{L}_i}(x_n))\}, 1 - \min\{\max(f_{\tilde{L}_i}(x_1), f_{\tilde{L}_i}(x_2), \ldots, f_{\tilde{L}_i}(x_n))\}] \mid w = x_1x_2\ldots x_n, x_1, x_2, \ldots, x_n \in \Sigma^*, n \geq 1 \}, w \in \Sigma^*. \]

Thus, \( \tilde{L}_i^+ = \bigcup_{i=1}^{\infty} \tilde{L}_i = \tilde{L}_1 \cup \tilde{L}_2 \cup \ldots \) i.e., Positive Closure is satisfied.

We treat VL as a special class of vague sets, and hence we say \( \tilde{L}_1 \) is contained in \( \tilde{L}_2 \) i.e., \( \tilde{L}_1 \subseteq \tilde{L}_2 \) iff \( t_{\tilde{L}_1}(w) \leq t_{\tilde{L}_2}(w) \) and \( 1 - f_{\tilde{L}_1}(w) \leq 1 - f_{\tilde{L}_2}(w) \), \( w \in \Sigma^* \)

and \( \tilde{L}_1 \) is equal to \( \tilde{L}_2 \) i.e., \( \tilde{L}_1 = \tilde{L}_2 \) iff \( t_{\tilde{L}_1}(w) = t_{\tilde{L}_2}(w) \) and \( 1 - f_{\tilde{L}_1}(w) = 1 - f_{\tilde{L}_2}(w) \), \( w \in \Sigma^* \).

**Definition 4.3.3:** Let \( \tilde{L} \) be a VL over an alphabet \( \Sigma \) with \( t_{\tilde{L}}(w) : \Sigma^* \rightarrow M \) & \( f_{\tilde{L}}(w) : \Sigma^* \rightarrow N \) as its truth membership and false membership functions respectively.

Then, we call \( \tilde{L} \) a vague regular language (VRL) if,

i. the sets \{ \( m \mid m \in M \& S_{\tilde{L}}(m) \neq \phi \) \} \& \{ \( n \mid n \in N \& S_{\tilde{L}}(1 - n) \neq \phi \) \} are finite, and

ii. for each \( m \in M \), \( S_{\tilde{L}}(m) \) and for each \( n \in N \), \( S_{\tilde{L}}(1 - n) \) is regular.
Theorem 4.1: VRLs are closed under union, intersection, complement, concatenation and star operations.

Proof: Let $\tilde{L}_1$ and $\tilde{L}_2$ be two VRLs over an alphabet $\Sigma$.

Let $t_{\tilde{L}_1} : \Sigma^* \rightarrow M_1$, $t_{\tilde{L}_2} : \Sigma^* \rightarrow M_2$ and $f_{\tilde{L}_1} : \Sigma^* \rightarrow N_1$, $f_{\tilde{L}_2} : \Sigma^* \rightarrow N_2$ be the truth and the false membership functions of $\tilde{L}_1$ and $\tilde{L}_2$ respectively, where $M_1$, $N_1$, $M_2$ and $N_2$ are the set of real numbers in $[0, 1]$. Let $\tilde{L}$ be the resulting language after an operation (union, intersection, complement, concatenation or star) and $t_{\tilde{L}}(w) : \Sigma^* \rightarrow M$ and $f_{\tilde{L}}(w) : \Sigma^* \rightarrow N$. It is understood that $M \subseteq M_1 \cup M_2$ and $N \subseteq N_1 \cup N_2$ (in the case of union, intersection, concatenation or star operation) or $M = N_1$ and $1 - N = 1 - M_1$ or $N = M_1$ (in the case of complementation) are finite, for which the corresponding strings are regular. Let $m$ and $n$ be in $M$ and $N$ respectively. Then $S_{\tilde{L}}(m)$ and $S_{\tilde{L}}(1 - n)$ are described as follows.

1. Union:-

$$S_{\tilde{L}}(m) = \begin{cases} S_{\tilde{L}_1}(m) - \cup_{m > m'} S_{\tilde{L}_2}(m'), & \text{if } m \in M_1 - M_2, \\ S_{\tilde{L}_2}(m) - \cup_{m > m'} S_{\tilde{L}_1}(m'), & \text{if } m \in M_2 - M_1, \\ (S_{\tilde{L}_1}(m) \cup S_{\tilde{L}_2}(m)) - \cup_{m > m'} S_{\tilde{L}_1}(m') - \cup_{m > m'} S_{\tilde{L}_2}(m''), & \text{if } m \in M_1 \cap M_2. \\ \end{cases}$$

$$S_{\tilde{L}}(1 - n) = \begin{cases} S_{\tilde{L}_1}(1 - n) - \cup_{(1-n) > (1-n)} S_{\tilde{L}_2}(1 - n'), & \text{if } n \in N_1 - N_2, \\ S_{\tilde{L}_2}(1 - n) - \cup_{(1-n) > (1-n)} S_{\tilde{L}_1}(1 - n'), & \text{if } n \in N_2 - N_1, \\ (S_{\tilde{L}_1}(1 - n) \cup S_{\tilde{L}_2}(1 - n)) - \cup_{(1-n) > (1-n)} S_{\tilde{L}_1}(1 - n') - \cup_{(1-n) > (1-n)} S_{\tilde{L}_2}(1 - n''), & \text{if } n \in N_1 \cap N_2. \\ \end{cases}$$

2. Intersection:-

$$S_{\tilde{L}}(m) = \begin{cases} S_{\tilde{L}_1}(m) - \cup_{m < m'} S_{\tilde{L}_2}(m'), & \text{if } m \in M_1 - M_2, \\ S_{\tilde{L}_2}(m) - \cup_{m < m'} S_{\tilde{L}_1}(m'), & \text{if } m \in M_2 - M_1, \\ (S_{\tilde{L}_1}(m) \cup S_{\tilde{L}_2}(m)) - \cup_{m < m'} S_{\tilde{L}_1}(m') - \cup_{m < m'} S_{\tilde{L}_2}(m''), & \text{if } m \in M_1 \cap M_2. \\ \end{cases}$$

$$S_{\tilde{L}}(1 - n) = \begin{cases} S_{\tilde{L}_1}(1 - n) - \cup_{(1-n) < (1-n)} S_{\tilde{L}_2}(1 - n'), & \text{if } n \in N_1 - N_2, \\ S_{\tilde{L}_2}(1 - n) - \cup_{(1-n) < (1-n)} S_{\tilde{L}_1}(1 - n'), & \text{if } n \in N_2 - N_1, \\ (S_{\tilde{L}_1}(1 - n) \cup S_{\tilde{L}_2}(1 - n)) - \cup_{(1-n) < (1-n)} S_{\tilde{L}_1}(1 - n') - \cup_{(1-n) < (1-n)} S_{\tilde{L}_2}(1 - n''), & \text{if } n \in N_1 \cap N_2. \\ \end{cases}$$
3. Complement:-

\[ S_{\overline{L}}(m) = S_{\overline{L_1}}(n) \text{ and } S_{\overline{L}}(1-n) = S_{\overline{L_1}}(1-m). \]

4. Concatenation:-

\[ S_{\overline{L}}(m) = \bigcup_{m_1 \in M_1, m_2 \in M_2} S_{\overline{L_1}}(m_1) \cdot S_{\overline{L_2}}(m_2) - \bigcup_{m_1' \in M_1, m_2' \in M_2} S_{\overline{L_1}}(m_1') \cdot S_{\overline{L_2}}(m_2'). \]

\[ S_{\overline{L}}(1-n) = \bigcup_{(1-n_1), (1-n_2)} S_{\overline{L_1}}(1-n_1) \cdot S_{\overline{L_2}}(1-n_2) - \bigcup_{(1-n_1'), (1-n_2')} S_{\overline{L_1}}(1-n_1') \cdot S_{\overline{L_2}}(1-n_2'). \]

5. Star:-

Assuming that \( M = \{m_1, m_2, \ldots, m_l\}, 1 \geq m_1 > m_2 > \ldots > m_l \geq 0, \) and

\( (1-N) = \{1-(1-n_1), \ldots, 1-(1-n_l)\}, \) \( 1 \geq (1-n_1) > (1-n_2) > \ldots > (1-n_l) \geq 0. \)

\[ S_{\overline{L}}(m_1) = \left(S_{\overline{L_1}}(m_1)\right)^* \text{ if } m_1 = 1, \]

\[ S_{\overline{L}}(m_1) = \left(S_{\overline{L_1}}(m_1)\right)^* - \{\varepsilon\} \text{ if } m_1 \neq 1, \]

\[ S_{\overline{L}}(1-n_1) = \left(S_{\overline{L_1}}(1-n_1)\right)^* \text{ if } (1-n_1) = 1, \]

\[ S_{\overline{L}}(1-n_1) = \left(S_{\overline{L_1}}(1-n_1)\right)^* - \{\varepsilon\} \text{ if } (1-n_1) \neq 1, \]

\[ S_{\overline{L}}(1-n_1) = \left(S_{\overline{L_1}}(1-n_1)\right)^* - \{\varepsilon\} \text{ if } (1-n_1) \neq 1, \]

Hence, Kleene Closure is satisfied.

4.3 FINITE AUTOMATA WITH VAGUE TRANSITIONS

Definition 4.3.1: A nondeterministic finite automaton with vague transitions (NDFA-VT) \( \tilde{A} \)

is a 6-tuple \( \tilde{A} = (Q, \Sigma, \tilde{\delta}, \tilde{\gamma}, S, F), \) where \( Q \) is the set of finite states, \( \Sigma \) is the finite set of

input alphabets, \( \tilde{\delta} \) and \( \tilde{\gamma} \) are the fuzzy subsets of \( Q \times \Sigma \times Q \) denotes respectively, the truth

membership and the false membership functions of \( \tilde{A} \), \( S \) is the starting state and \( F \subseteq Q \) is

the set of final states.

For \( x \in \Sigma^* \) and \( p, q \in Q \) define,

\[ \tilde{\delta}^*(p, x, q) = \begin{cases} 0, & \text{if } x = \varepsilon \text{ and } p \neq q, \\ 1, & \text{if } x = \varepsilon \text{ and } p = q, \\ \max_{r \in Q} \{\min(\tilde{\delta}^*(p, x', r), \tilde{\delta}(r, a, q)) \} & \text{otherwise.} \end{cases} \]

\[ \tilde{\gamma}(r, a, q) = \begin{cases} 0, & \text{if } x = \varepsilon \text{ and } p \neq q, \\ 1, & \text{if } x = \varepsilon \text{ and } p = q, \\ \max_{r \in Q} \{\min(\tilde{\delta}(r, a, q), \tilde{\gamma}(r, a, q)) \} & \text{otherwise.} \]
\[ \tilde{\gamma}^*(p,x,q) = \begin{cases} 
1, & \text{if } x = \varepsilon \text{ and } p \neq q, \\
0, & \text{if } x = \varepsilon \text{ and } p = q, \\
\min_{r \in Q} \{ \max (\tilde{\gamma}^*(p,x',r), \tilde{\gamma}(r,a,q)) \} & \text{otherwise.}
\end{cases} \]

Then, we say that \( x \in \Sigma^* \) is accepted by \( \tilde{A} \) with the truth degree \( \tilde{d}_\tilde{A}(x) \) and the false degree \( \tilde{f}_\tilde{A}(x) \) such that \( 0 \leq \tilde{d}_\tilde{A}(x) + \tilde{f}_\tilde{A}(x) \leq 1 \), where

\[
\tilde{d}_\tilde{A}(x) = \max \{ \tilde{\delta}^*(S,x,q) \mid q \in F \} \quad \text{and} \quad \tilde{f}_\tilde{A}(x) = \min \{ \tilde{\gamma}^*(S,x,q) \mid q \in F \}.
\]

We denote the language accepted by \( \tilde{A} \) as \( \tilde{L}(\tilde{A}) \), where

\[
\tilde{L}(\tilde{A}) = \{(x,[\tilde{d}_\tilde{A}(x),1-\tilde{f}_\tilde{A}(x)]) \mid x \in \Sigma^*\}.
\]

**Example 4.2:** Let \( \tilde{A}=(Q,\Sigma,\tilde{\delta},\tilde{\gamma},S,F) \) be a NDFA-VT as shown in the figure below, where

\[
Q = \{S,p,q\}, \quad \Sigma = \{a,b\}, \quad S = S, \quad F = \{q\}, \quad \tilde{\delta}(S,a,p) = 0.3, \quad \tilde{\gamma}(S,a,p) = 0.5, \quad \tilde{\delta}(p,a,q) = 0.4, \quad \tilde{\gamma}(p,a,q) = 0.7, \quad \tilde{\delta}(p,b,q) = 0.6, \quad \tilde{\gamma}(p,b,q) = 0.8, \quad \tilde{\delta}(p,b,p) = 0.3, \quad \tilde{\gamma}(p,b,p) = 0.4.
\]

![Figure 4.1: NDFA-VT](image)

The vague regular language accepted by the above NDFA-VT is;

\[
\tilde{L}(\tilde{A}) = \{(x,[0.3,0.5]) \mid x \in ab \} \cup \{(y,[0.3,0.4]) \mid y \in ab^*a\}.
\]

**Definition 4.3.2:** A deterministic finite automaton with vague transitions (DFA-VT) is a NDFA-VT \( \tilde{A}=(Q,\Sigma,\tilde{\delta},\tilde{\gamma},S,F) \) with the condition that, for each \( p,q,q' \in Q \) and \( a \in \Sigma \), if \( \tilde{\delta}(p,a,q) > 0 \) and \( \tilde{\gamma}(p,a,q) < 1 \) and \( \tilde{\delta}(p,a,q') > 0 \) and \( \tilde{\gamma}(p,a,q') < 1 \), then \( q = q' \).
Theorem 4.2: ‘\( \tilde{L} \)’ is a VRL iff it is accepted by a NDFA-VT with the exception of \( \varepsilon \) the empty string.

Proof: Let \( \tilde{L} \) be a VRL. Let \( t_{\tilde{L}}(w):\Sigma^* \rightarrow M \) and \( f_{\tilde{L}}(w):\Sigma^* \rightarrow N \) be the truth and false membership functions of \( \tilde{L} \) respectively, where \( M \) and \( N \) represents the finite set of real numbers in \([0, 1]\).

Let \( M = \{m_1, m_2, \ldots, m_l\} \) and \( N = \{n_1, n_2, \ldots, n_l\} \) be the set of truth and false membership values of \( \tilde{L} \) respectively.

Let \( S_{\tilde{L}}(m_i) \) and \( S_{\tilde{L}}(1-n_i) \) are regular for each \( m_i \in M \) and \( n_i \in N \), where \( 1 \leq i \leq l \).

Note that, \( S_{\tilde{L}}(m_i) \cap S_{\tilde{L}}(m_j) = \emptyset \) and \( S_{\tilde{L}}(1-n_i) \cap S_{\tilde{L}}(1-n_j) = \emptyset \) for \( i \neq j \).

Let \( A_i = (Q_i, \Sigma, \delta_i, \gamma_i, S_i, F_i) \) be a DFA-VT (or a NDFA-VT) such that,

\[
S_{\tilde{L}}(m_i) = S_{\tilde{L}}(1-n_i) = \tilde{L}(A_i), \text{ for } 1 \leq i \leq l.
\]

We construct, \( \tilde{A}_i = (Q_i, \Sigma, \tilde{\delta}_i, \tilde{\gamma}_i, S_i, F_i) \), where

\[
\tilde{\delta}_i(p, a, q) = \begin{cases} m_i, \text{ if } (p, a, q) \in \delta_i, \\ 0, \text{ otherwise.} \end{cases}
\]

\[
\tilde{\gamma}_i(p, a, q) = \begin{cases} n_i, \text{ if } (p, a, q) \in \gamma_i, \\ 1, \text{ otherwise.} \end{cases}
\]

represents the truth and false membership functions of \( \tilde{L} \) respectively.

We assume that \( Q_i \cap Q_j = \emptyset \) for \( i \neq j \).

Define, \( \tilde{A} = (Q, \Sigma, \tilde{\delta}, \tilde{\gamma}, S, F) \) such that, \( Q = Q_1 \cup Q_2 \cup \ldots \cup Q_l \cup \{S\} \),

\[
S \notin Q_1 \cup Q_2 \cup \ldots \cup Q_l, \ F = F_1 \cup F_2 \cup \ldots \cup F_l,
\]

\[
\tilde{\delta}(p, a, q) = \begin{cases} \tilde{\delta}_i(p, a, q), \text{ if } p, q \in Q_i \text{ for some } i \in \{1, 2, \ldots, l\}, \\ \delta_i(S_i, a, q), \text{ if } p = S_i, q \in Q_i \text{ for some } i \in \{1, 2, \ldots, l\}, \\ 0, \text{ otherwise.} \end{cases}
\]

&

\[
\tilde{\gamma}(p, a, q) = \begin{cases} \tilde{\gamma}_i(p, a, q), \text{ if } p, q \in Q_i \text{ for some } i \in \{1, 2, \ldots, l\}, \\ \gamma_i(S_i, a, q), \text{ if } p = S_i, q \in Q_i \text{ for some } i \in \{1, 2, \ldots, l\}, \\ 1, \text{ otherwise.} \end{cases}
\]

Clearly, \( \tilde{A} \) accepts \( \tilde{L} \) with the possible exception of \( \varepsilon \) the empty string.

Conversely,

Let \( \tilde{A} = (Q, \Sigma, \tilde{\delta}, \tilde{\gamma}, S, F) \) be a NDFA-VT.

Define, a VL \( \tilde{L} \) with \( t_{\tilde{L}}(w) = td_{\tilde{A}}(w) \) and
Define, the languages $\tilde{L}_i$, $1 \leq i \leq 1$, in the increasing order (sequence) of $i$ as follows.

\[
\tilde{L}_1 = \tilde{L}(A_1),
\]

\[
\tilde{L}_2 = \tilde{L}(A_2) - \tilde{L}(A_1),
\]

\[
\vdots
\]

\[
\tilde{L}_i = \tilde{L}(A_i) - \bigcup_{j=1}^{i-1} \tilde{L}(A_j).
\]

Then $S_{\tilde{L}}(m_i) = S_{\tilde{L}}(1 - n_i) = \tilde{L}_i$ and $\tilde{L}_i$ is regular for each $i$, $1 \leq i \leq l$.

Thus, $\tilde{L}$ represents a VRL.

**Theorem 4.3:** Let $\tilde{L}$ be a VRL, then $\tilde{L}$ is accepted by a DFA-VT iff it satisfies the following conditions: For $x, y \in \Sigma^+, u \in \Sigma^*$

\[
x = yu \text{ and } t_{\tilde{L}}(y) < 1 \text{ (or } 1 - f_{\tilde{L}}(y) > 0) \text{ implies that } t_{\tilde{L}}(x) \leq t_{\tilde{L}}(y)
\]

and $f_{\tilde{L}}(x) \geq f_{\tilde{L}}(y)$ (or $1 - f_{\tilde{L}}(x) \leq 1 - f_{\tilde{L}}(y)$).

**Proof:** Let $\tilde{L}$ be accepted by a DFA-VT $\tilde{A} = (Q, \Sigma, \tilde{\delta}, \tilde{\gamma}, S, F)$.

We have to show that $\tilde{L}$ satisfies the given condition.

Let $x = yu$ for $x, y \in \Sigma^+, u \in \Sigma^*$. If $td_{\tilde{A}}(x) = 0$ and $1 - fd_{\tilde{A}}(x) = 0$, then

\[
t_{\tilde{L}}(x) \leq t_{\tilde{L}}(y) \text{ and } f_{\tilde{L}}(x) \geq f_{\tilde{L}}(y) \text{ (or } 1 - f_{\tilde{L}}(x) \leq 1 - f_{\tilde{L}}(y))
\]

is trivially true.
Otherwise,
\[
\tilde{\delta}(x) = t_d^\tilde{A}(x) = \min \{ \tilde{\delta}^*(S, y, q), \tilde{\delta}^*(q, u, f) \} | q, f \in F \}
\]
\[
\leq \tilde{\delta}^*(S, y, q) = t_d^\tilde{A}(y) = \tilde{\delta}(y).
\]
&
\[
\tilde{\gamma}(x) = f_d^\tilde{A}(x) = \max \{ \tilde{\gamma}^*(S, y, q), \tilde{\gamma}^*(q, u, f) \} | q, f \in F \}
\]
\[
\leq \tilde{\gamma}^*(S, y, q) = f_d^\tilde{A}(y) = \tilde{\gamma}(y).
\]
Conversely,

let \( \tilde{L} \) be a VRL and let \( t_{\tilde{L}}(w): \Sigma^* \to M \) and \( f_{\tilde{L}}(w): \Sigma^* \to N \) be the truth and false membership functions of \( \tilde{L} \) respectively satisfying the given condition, where \( M \) and \( N \) are the finite set of real numbers in \([0, 1]\).

Assume that \( M = \{m_1, m_2, ..., m_l\} \) and \( N = \{n_1, n_2, ..., n_l\} \).

It is clear from the previous theorem (Theorem 4.2) that we can construct a DFA-VT \( \tilde{A}_i = (Q_i, \Sigma, \delta, \tilde{\gamma}, S_i, F_i) \) such that, \( \tilde{L}(A_i) = S_{\tilde{L}}(m_i) = S_{\tilde{L}}(1 - n_i) \) for each \( i, 1 \leq i \leq l \).

Note that for \( 1 \leq i, j \leq l \) and \( i \neq j \),
\[
\tilde{L}(A_i) \cap \tilde{L}(A_j) = S_{\tilde{L}}(m_i) \cap S_{\tilde{L}}(m_j) = S_{\tilde{L}}(1 - n_i) \cap S_{\tilde{L}}(1 - n_j) = \emptyset.
\]

Now construct a DFA \( A = (Q, \Sigma, \delta, \gamma, S, F) \), where
\[
Q = Q_1 \times Q_2 \times ... \times Q_l, \quad S = \{S_1, S_2, ..., S_l\},
\]
\( \delta: \Sigma \to Q \) is defined by \( \delta: ((q_1, q_2, ..., q_l), a) = (\delta_1(q_1, a), \delta_2(q_2, a), ..., \delta_l(q_l, a)) \),
\( \gamma: \Sigma \to Q \) is defined by \( \gamma: ((q_1, q_2, ..., q_l), a) = (\gamma_1(q_1, a), \gamma_2(q_2, a), ..., \gamma_l(q_l, a)) \) and
\( F = F'_1 \cup F'_2 \cup ... \cup F'_l \), where
\[
F'_i = \{ (q_1, q_2, ..., q_l) | (q_1, q_2, ..., q_l) \in Q, q_i \in F_i \& q_j \notin F_j \text{ for } i \neq j \}, 1 \leq i, j \leq l.
\]

It is clear that \( F'_i \cap F'_j = \emptyset \), for \( i \neq j \), \( S_{\tilde{L}}(m_i) = \{ w \in \Sigma^* \& \delta^*(S, w) \in F'_i \} \) and
\( S_{\tilde{L}}(n_i) = \{ w \in \Sigma^* \& \gamma^*(S, w) \in F'_i \} \). Hence \( S_{\tilde{L}}(m_i) = S_{\tilde{L}}(1 - n_i) \).

Based on the above DFA-VT ‘A’, we can define a DFA-VT as
\( \tilde{A} = (Q, \Sigma, \delta, \gamma, S, F) \) such that
\[
\tilde{\delta}(p, a, q) = \begin{cases} m_i, & \text{if } \delta(p, a) = q \in F'_i, \\ 1, & \text{if } \delta(p, a) = q \notin F'_i, \end{cases} \quad \& \quad \tilde{\gamma}(p, a, q) = \begin{cases} n_i, & \text{if } \delta(p, a) = q \in F'_i, \\ 0, & \text{if } \delta(p, a) = q \notin F'_i, \end{cases}
\]
0, otherwise.

Remaining part is to show that \( t_d^\tilde{A}(w) = t_{\tilde{L}}(w) \) and \( f_d^\tilde{A}(w) = f_{\tilde{L}}(w) \) for each \( w \in \Sigma^+ \).
First we show that $\tilde{A}$ has the following property:

For each $w \in \Sigma^+$ with $w = xa, x \in \Sigma^*$ and $a \in \Sigma$,
\begin{equation}
\begin{cases}
\text{if } \delta^*(S, x, p) \geq m_i \text{ and } \delta(p, a, q) = m_i \text{ for some } q \in F_i, 1 \leq i \leq l, \text{ and} \\
\text{if } \delta^*(S, x, p) \geq m_i \text{ and } \delta(p, a, q) = m_i \text{ for some } q \in F_i, 1 \leq i \leq l.
\end{cases}
\end{equation}

The ‘if’ part is true and ‘only if’ part will be true when $x = \varepsilon$.

For $x \neq \varepsilon$, we assume the contrary,

\begin{align*}
\delta^*(S, x, p) &= m_i, \text{ and } \delta(p, a, q) = m_j > m_i \text{ for } 1 \leq i, j \leq l, \\
\gamma^*(S, x, p) &= n_i, \text{ and } \gamma(p, a, q) = n_j < n_i \text{ for } 1 \leq i, j \leq l.
\end{align*}

Then, there exists a decomposition of $x = ybz, y, z \in \Sigma^*$ and $b \in \Sigma$, such that

\begin{align*}
\delta^*(S, y, r) &\geq m_i, \text{ and } \delta(r, b, t) = m_i, \text{ and } \delta(t, z, p) \geq m_i, \text{ also} \\
\gamma^*(S, y, r) &\leq n_i, \text{ and } \gamma(r, b, t) = n_i, \text{ and } \gamma(t, z, p) \leq n_i.
\end{align*}

By the definition of $\tilde{A}$, we know that $t \in F_i$ and $q \in F_j$.

Thus, we have $t_{\tilde{L}}(yb) = m_i, f_{\tilde{L}}(yb) = n_i$ and $t_{\tilde{L}}(w) = m_j, f_{\tilde{L}}(w) = n_j$.

Since we assume that $m_j > m_i$ and $n_j < n_i$, this is a contradiction to the given condition.

So, (1) holds. Furthermore, the right hand side of (1) implies that $xa \in S_{\tilde{L}}(m_i)$ i.e., $t_{\tilde{L}}(w) = m_i$ and $xa \in S_{\tilde{L}}(1-n_i)$ i.e., $f_{\tilde{L}}(w) = n_i$.

This completes the proof.

4.4 FINITE AUTOMATA WITH VAGUE (FINAL) STATES

In this section, the models of finite automaton with vague (final) states (NDFA-VS and DFA-VS) are proposed and the acceptance of vague regular language through them is discussed. Unlike previous models (NDFA-VT and DFA-VT proposed in previous section), in this model, NDFA-VS and DFA-VS are equivalent in accepting vague regular language. Vague regular language is accepted by NDFA-VS and DFA-VS without any restrictions and vice versa. So, this model is more suitable for the study of vague regular languages.

Definition 4.4.1: A nondeterministic finite automaton with vague (final) states (NDFA-VS) ‘$\tilde{A}$’ is a 7-tuple $\tilde{A}=(Q, \Sigma, \delta, \gamma, S, \tilde{T}_{\tilde{F}A}, \tilde{F}_{\tilde{F}A})$, where $Q$ is the finite set of states, $\Sigma$ is the finite set of input alphabets, $\delta, \gamma : Q \times \Sigma \rightarrow 2^Q$ are the transition functions, $S$ is the vague starting
state, and \( \tilde{T}_{F_A}, \tilde{F}_{F_A} : Q \rightarrow [0,1] \) called respectively, the truth and false membership functions of vague (final) state set.

Define, \( t d_{\tilde{A}}(x) = \max\{ \tilde{T}_{F_A}(q) \mid (S, x, q) \in \delta^* \} \) and

\[
fd_{\tilde{A}}(x) = \min\{ \tilde{F}_{F_A}(q) \mid (S, x, q) \in \gamma^* \} \quad \text{or} \quad fd_{\tilde{A}}(x) = \max\{ 1 - \tilde{F}_{F_A}(q) \mid (S, x, q) \in \gamma^* \},
\]

where \( \delta^*, \gamma^* : Q \times \Sigma^* \rightarrow 2^Q \) called respectively, the reflexive and transitive closure of \( \delta \) and \( \gamma \).

The string \( 'x' \) is accepted by \( 'A' \) with the truth degree \( t d_{\tilde{A}}(x) \) and the false degree \( fd_{\tilde{A}}(x) \) respectively with the condition \( 0 \leq t d_{\tilde{A}}(x) + fd_{\tilde{A}}(x) \leq 1 \).

The vague language accepted by \( 'A' \) is denoted by \( \tilde{L}(A) \), and is given by the set,

\[
\tilde{L}(A) = \{(x, [t d_{\tilde{A}}(x), 1 - fd_{\tilde{A}}(x)]) \mid x \in \Sigma^* \}.
\]

**Example 4.3:**

Let \( \tilde{A} = (Q, \Sigma, \delta, \gamma, S, \tilde{T}_{F_A}, \tilde{F}_{F_A}) \) be a NDFA-VS as shown in the figure below.

Here, \( t d_{\tilde{A}}(b) = 0.6, fd_{\tilde{A}}(b) = 0; t d_{\tilde{A}}(ba) = 0.5, fd_{\tilde{A}}(ba) = 0.2; t d_{\tilde{A}}(bb) = 0.7, fd_{\tilde{A}}(bb) = 0.1; t d_{\tilde{A}}(bbba) = 0.8, fd_{\tilde{A}}(bbba) = 0 \), and so on.

![NDFA-VS diagram](image)

**Figure 4.2: NDFA-VS**
**Definition 4.4.2:** A deterministic finite automaton with vague (final) states (DFS-VS)

\[ \tilde{A} = (Q, \Sigma, \delta, \gamma, S, \tilde{T}_{F\tilde{A}}, \tilde{F}_{F\tilde{A}}) \]

is a NDFA-VS with \( \delta \) and \( \gamma \) being functions.

For each \( x \in \Sigma^* \),

\[ td_{\tilde{A}}(x) = \tilde{T}_{F\tilde{A}}(q), \text{ where } q = \delta^*(S, x) \] and \( fd_{\tilde{A}}(x) = \tilde{F}_{F\tilde{A}}(q), \text{ where } q = \gamma^*(S, x) \).

Define, \( td_{\tilde{A}}(x) = 0 \) and \( fd_{\tilde{A}}(x) = 1 \) if \( \delta^*(S, x) \) and \( \gamma^*(S, x) \) are not defined.

**Theorem 4.4:** Let \( \tilde{L} \) be a VL. Then \( \tilde{L} \) is a VRL iff it is accepted by a DFA-VS.

**Proof:** Let \( \tilde{L} \) be a VL having truth and false membership functions

\[ t_L(w): \Sigma^* \to M \] and \( f_L(w): \Sigma^* \to N \) respectively, where \( M \) and \( N \) denotes unit interval.

Assume that \( \tilde{L} \) is a VRL. So, for each \( m \in M \) and \( n \in N \), the sets \( S_L(m) \) and \( S_L(1-n) \) are regular.

Assume that, \( M = \{m_1, m_2, ..., m_l\} \) and \( N = \{n_1, n_2, ..., n_l\} \).

For each \( i, 1 \leq i \leq l \), construct a DFA-VS, \( \tilde{A}_i = (Q, \Sigma, \delta_i, \gamma_i, S_i, \tilde{T}_{F\tilde{A}_i}, \tilde{F}_{F\tilde{A}_i}) \) such that,

\[ \tilde{L}(\tilde{A}_i) = S_L(m_i) = S_L(1-n_i). \]

Now define a DFA-VS,

\[ \tilde{A} = (Q, \Sigma, \delta, \gamma, S, \tilde{T}_{F\tilde{A}}, \tilde{F}_{F\tilde{A}}) \]

to be the cross product of \( \tilde{A}_i, 1 \leq i \leq l \), with

\[ \tilde{T}_{F\tilde{A}}(q^{(1)}, q^{(2)}, ..., q^{(l)}) = \begin{cases} m_i, & q^{(i)} \in \tilde{F}_{F\tilde{A}_i} \text{ for some } 1 \leq i \leq l \text{ and } q^{(j)} \notin F_i \forall j \neq i, \\ 0, & \text{otherwise.} \end{cases} \]

\&

\[ \tilde{F}_{F\tilde{A}}(q^{(1)}, q^{(2)}, ..., q^{(l)}) = \begin{cases} n_i, & q^{(i)} \in \tilde{F}_{F\tilde{A}_i} \text{ for some } 1 \leq i \leq l \text{ and } q^{(j)} \notin F_i \forall j \neq i, \\ 1, & \text{otherwise.} \end{cases} \]

Note that if \( (q^{(1)}, q^{(2)}, ..., q^{(l)}) \) is reachable from \( (S_1, S_2, ..., S_l) \), in \( \tilde{A} \), then it is not possible to get \( q^{(i)} \in \tilde{F}_{F\tilde{A}_i} \) and \( q^{(j)} \in \tilde{F}_{F\tilde{A}_j} \) also, \( q^{(i)} \notin F_i \forall j \neq i \), \( q^{(j)} \notin F_j \forall i \neq j \) because \( \tilde{L}(\tilde{A}_i) \cap \tilde{L}(\tilde{A}_j) = \phi \) for \( i \neq j, 1 \leq i, j \leq l \).

Hence \( \tilde{A} \) accepts \( \tilde{L} \).

Conversely,

Let \( \tilde{A} = (Q, \Sigma, \delta, \gamma, S, \tilde{T}_{F\tilde{A}}, \tilde{F}_{F\tilde{A}}) \) be a DFA-VS.
Define, \( M = \{ m | \tilde{T}_{F\tilde{A}}(q) = m \text{ for some } q \in Q \} \) and \( N = \{ n | \tilde{F}_{F\tilde{A}}(q) = n \text{ for some } q \in Q \} \).

Thus \( M \) and \( N \) are finite sets.

For each \( m \in M \) and \( n \in N \), define \( A_{[m,n]} = (Q, \Sigma, \delta, \gamma, S, T_m, F_n) \), where

\[
T_m = \{ q | \tilde{T}_{F\tilde{A}}(q) = m \} \quad \text{and} \quad F_n = \{ q | \tilde{F}_{F\tilde{A}}(q) = n \}.
\]

Let \( \tilde{L} = \tilde{L}(\tilde{A}) \) i.e., \( t_{\tilde{L}}(w) = td(\tilde{A}) \) and \( f_{\tilde{L}}(w) = fd(\tilde{A}) \). Let \( \tilde{L}(\tilde{A}) = S_{\tilde{L}}(m_i) = S_{\tilde{L}}(1-n_i) \)

Thus for each \( m \in M \) the string \( S_{\tilde{L}}(m) \) and for each \( n \in N \) the string \( S_{\tilde{L}}(1-n) \) are regular.

Hence \( \tilde{L} \) is a VRL. This completes the proof.

**Theorem 4.5:** A VL is accepted by a NDFA-\( VS \) iff it is accepted by a DFA-\( VS \).

**Proof:** Let ‘\( \tilde{L} \)’ be a VL. Here we need to show if ‘\( \tilde{A} \)’ is a NDFA-\( VS \) and \( \tilde{L} = \tilde{L}(\tilde{A}) \) then \( \tilde{L} = \tilde{L}(\tilde{A}') \), where \( \tilde{A}' \) is a DFA-\( VS \).

Let \( \tilde{A} = (Q, \Sigma, \delta, \gamma, S, T_{F\tilde{A}}, F_{F\tilde{A}}) \) be a NDFA-\( VS \). We can construct a DFA-\( VS \)

\( \tilde{A}' = (Q', \Sigma, \delta', \gamma', S', T'_{F\tilde{A}}, F'_{F\tilde{A}}) \) using the method of standard subset construction.

For each \( P \in Q' (P \subseteq Q) \), define \( T'_{F\tilde{A}}(P) = \max \{ m | \tilde{T}_{F\tilde{A}}(q) = m, q \in P \} \) and

\( F'_{F\tilde{A}}(P) = \min \{ n | \tilde{F}_{F\tilde{A}}(q) = n, q \in P \} \), where \( m \) and \( n \) represents respectively, the truth

and false membership values of the strings in the language.

Hence \( \tilde{L} = \tilde{L}(\tilde{A}') \).

### 4.5 Vague Regular Expressions

Every string in VRL has finite truth membership and finite false membership value in [0, 1].

The set of finite strings associated with these values forms a regular language. As fuzzy regular expressions expresses fuzzy regular language, VRL may also be represented in the form of a modified vague regular expression.

These modified vague regular expressions can be used in lexical analysis of VRL.

For example: Let \( \Sigma = \{ a, b, c \} \) be an alphabet.

Then, \((a + b)^* ab/[0.4, 0.7] + bc/[0.7, 0.9] \) represents a vague regular expression.

We give a prescribed definition of vague regular expression below.

**Definition 4.5.1:** Let \( \Sigma \) be an alphabet and \( M, N \) be finite sets of real numbers in [0, 1].
1. Let ‘e’ be a regular expression over $\Sigma$ and $m \in M, n \in N$. Then, we call $e = \emptyset / [m, 1-n]$ a vague regular expression (VRE), where $m$ and $n$ denotes respectively, the truth and false membership value of ‘e’ such that $0 \leq m + n \leq 1$.

2. Let $\tilde{e}_1$ and $\tilde{e}_2$ be two vague regular expressions then, the following holds.
   i. $\emptyset \in VRE$ with membership value $[1, 1]$,
   ii. $\varepsilon \in VRE$ with membership value $[1, 1]$,
   iii. $a \in VRE$ with membership value $[m, 1-n]$, where $m, n \in [0,1]$ called the truth and false membership value of the string ‘a’ for $a \in \Sigma$ respectively,
   iv. $\forall \tilde{e}_1 \in VRE, (\tilde{e}_1 + \tilde{e}_2) \in VRE, (\tilde{e}_1 \cdot \tilde{e}_2) \in VRE$, and $(\tilde{e}_1)^* \in VRE$.

By applying above mentioned steps ((1) and (2)) finite number of times, a vague regular expression can be obtained.

**Definition 4.5.2:** Let $\tilde{e}$ be a VRE. Then, the vague regular language (VRL) $L(\tilde{e})$ corresponding to VRE ‘$\tilde{e}$’ is defined as; 
$L = L(\tilde{e}) = \{(x,[m,1-n]) \mid x \in L(e)\}$.

Here, $L(e)$ represents the language for regular expression ‘e’ and $m, n \in [0,1]$ denotes respectively, the truth and false membership value of the string ‘x’. If

1. $\tilde{e} = \tilde{e}_1 + \tilde{e}_2$
2. $\tilde{e} = \tilde{e}_1 \cdot \tilde{e}_2$, and
3. $\tilde{e} = (\tilde{e}_1)^*$ or $\tilde{e} = (\tilde{e}_2)^*$, then the corresponding language will be
   i. $L(\tilde{e}) = L(\tilde{e}_1) \cup L(\tilde{e}_2)$
   ii. $L(\tilde{e}) = L(\tilde{e}_1) \cdot L(\tilde{e}_2)$
   iii. $L(\tilde{e}) = (L(\tilde{e}_1))^*$ or $L(\tilde{e}) = (L(\tilde{e}_2))^*$ respectively.

**Definition 4.5.3:** A VRE $\tilde{e}$ over an alphabet $\Sigma$ is normalized vague regular expression if it is of the form $\tilde{e} = e_1 / [m_1, 1-n_1] + e_2 / [m_2, 1-n_2] + ... + e_l / [m_l, 1-n_l]$, where $e_1, e_2, ..., e_l$ represents the regular expressions over $\Sigma$.

Here, $m_1, m_2, ..., m_l$ and $n_1, n_2, ..., n_l$ are real numbers in $[0, 1]$ and represents respectively, the truth membership and the false membership values of $e_1, e_2, ..., e_l$, such that
0 ≤ m_i + n_j ≤ 1, ∀ l ≥ 1.

Note that, if [m, 1 − n] = [1, 1] then, e / [m, 1 − n] can simply be written as ‘e’.

We assume that ‘·’ and ‘*’ have higher priorities than ‘/’. So, certain pairs of parenthesis can be omitted.

**Example 4.4:**

1. \( \overline{e} = (b + ab + aab)^* (e + a + a)/(0.6, 0.9) + (a + e)(a + ab)^* \),
2. \( \overline{e} = (ba + (a + bb)a^* b)/(0.4, 0.7) + ab^* a/(0.8, 1) + a^* \),
3. \( \overline{e} = (b^* (a + b)^* [0.5, 0.9]) \cdot (a^* ba^* /[0.8, 1]) + a^* + b \),
4. \( \overline{e} = ((a + b)^* /[0.5, 0.6]) /[0.6, 0.8] + a^* ba^* (b + a) /[1, 1] \),
5. \( \overline{e} = ((a + b)^* /[0.7, 0.9] + (b + a)^* /[0.5, 0.8])^* /[0.8, 1] + ab(aba^*) /[0.7, 0.9] \).

Above (1) to (3) are valid VREs, where (1) and (2) are normalized and (3) is not normalized. Also, (4) and (5) are not valid VREs.

**Definition 4.5.4:** A VRE \( \overline{e} \) is called a **strictly normalized vague regular expression**, if it is normalized. i.e., \( \overline{e} = e_1 /[m_1, 1 − n_1] + e_2 /[m_2, 1 − n_2] + ... + e_l /[m_l, 1 − n_l] \) and for any \( m_i \neq m_j \) and \( n_i \neq n_j \), \( L(e_i) \cap L(e_j) = \phi \).

**Example 4.5:**

1. \( \overline{e} = abc /[0.4, 0.9] + bc /[0.5, 0.8] \),
2. \( \overline{e} = ((aba)^* a + (a^* b)^*) /[0.5, 0.9] + (a^* ba^*) /[0.3, 0.9] \).

Above VREs are strictly normalized.

1. \( \overline{e} = a^* b /[0.7, 0.9] + (a + b)^* a(a + b)^* b /[0.4, 0.7] \) shows a VRE which is not strictly normalized.

One can see that the families of languages as represented by VREs, normalized VREs, and strictly normalized VREs, respectively, are same as the family of vague regular languages.

### 4.6 MYHILL-NERODE THEOREM FOR VRL

This section deals with the construction of the minimal DFA-VS. In the fuzzy framework, Myhill-Nerode theorem has been extended for fuzzy regular language and an algorithm is given for minimization of deterministic finite automaton with fuzzy (final) states [66]. Here, we have extended Myhill-Nerode theorem for VRL and an algorithm is given to minimize
DFA-VS by computing and merging indistinguishable states of it using the property of right invariant equivalence. This results in reducing the number of redundant states of DFA-VS. So that the output of constructed automaton will be approximately the same as the original one. For classical automaton this problem was generally solved in a procedure of minimization that uses a notion of right invariant equivalence classes.

Theorem (Myhill-Nerode theorem for VRL) 4.6:

**Statement:** The following three statements are equivalent to one another:

(i) Some finite automaton with vague (final) states can accept a VRL $\tilde{L}$ over $\Sigma$.

(ii) $\tilde{L}$ is the union of some of the equivalence classes of a right invariant equivalence relation of finite index.

(iii) Let the relation $R_L \subseteq \Sigma^* \times \Sigma^*$ be defined by $xR_L y \iff \forall z \in \Sigma^*, t_L(xz) = t_L(yz)$ and $f_L(xz) = f_L(yz)$, then $R_L$ is an equivalence relation of finite index.

**Proof.** (i)$\Rightarrow$(ii) Let $\tilde{L}$ be a VRL over an alphabet $\Sigma$. Assume that $\tilde{L}$ is accepted by some DFA-VS $\tilde{A} = (Q, \Sigma, \delta, \gamma, q_0, \tilde{T}_F, \tilde{F}_A)$. Let $R_\tilde{A}$ be the equivalence relation $xR_\tilde{A} y$ iff $\delta(q_0, x) = \delta(q_0, y)$ and $\gamma(q_0, x) = \gamma(q_0, y)$. $R_\tilde{A}$ is right invariant since, for any $z$, $\delta(q_0, xz) = \delta(q_0, yz)$, if $\delta(q_0, x) = \delta(q_0, y)$ and $\gamma(q_0, xz) = \gamma(q_0, yz)$, if $\gamma(q_0, x) = \gamma(q_0, y)$. Then the index of $R_\tilde{A}$ is finite, since the index is at most the number of states in Q. Furthermore, $\tilde{L}$ is the union of those equivalence classes having a string ‘x’ such that $\delta(q_0, x)$ and $\gamma(q_0, x)$ are respectively in $\tilde{T}_F$ and $\tilde{F}_A$ (i.e., the equivalence classes corresponding to the final states).

(ii)$\Rightarrow$(iii) We show that any equivalence relation ‘E’ satisfying (ii) is a refinement of $R_\tilde{L}$; i.e., some equivalence class of $R_\tilde{L}$ will be the superset of every equivalence class ‘E’. Thus, the index of $R_\tilde{L}$ cannot be greater than the index of ‘E’ and so is finite. Assume that ‘xEy’.

For each $z \in \Sigma^*$, ‘xzEy$z’ and thus $\tilde{L}(xz) = \tilde{L}(yz)$ (since ‘E’ is right invariant). Hence $xR_\tilde{L} y$. We conclude that each equivalence class of ‘E’ is the subset of some equivalence class of $R_\tilde{L}$.
(iii) \( \Rightarrow (i) \) To show that \( R_L \) is right invariant, suppose \( x R_L y \), and let \( w \in \Sigma^* \), we must prove that \( xw R_L yw \); i.e., for any \( z \), \( \overline{L}(xz) = \overline{L}(yw) \). Since \( x R_L y \), for any \( v \), \( \overline{L}(xv) = \overline{L}(yw) \) (by the definition of \( R_L \)). Consider \( v = wz \) to prove \( R_L \) is right invariant.

Now we present the minimized DFA-VS by constructing equivalence classes of \( R_L \):

Let \( Q' \) be the finite set of equivalence classes of \( R_L \) and \( [x] \in Q' \) containing \( 'x' \).

Define, \( \delta'([x], a) = [xa] \) and \( \gamma'([x], a) = [xa] \). This definition is consistent as \( R_L \) is right invariant. If we choose \( 'y' \) instead of \( 'x' \) from \( [x] \), we will have \( \delta'([x], a) = [ya] \) and \( \gamma'([x], a) = [ya] \). But \( x R_L y \), so \( \overline{L}(xz) = \overline{L}(yz) \). In particular, if \( z = az' \), \( \overline{L}(zaz') = \overline{L}(yaz') \), so \( xa R_L ya \) and \( [xa] = [ya] \).

Let \( q_0 = [\varepsilon] \) \( \overline{F}_{\overline{A}}' = \{ [x] \mid x \in \overline{L} \} \) and \( \overline{F}_{\overline{A}} = \{ [x] \mid x \in \overline{L} \} \). The finite automaton \( \overline{A}' = (Q', \Sigma, \delta', \gamma', q_0, \overline{F}_{\overline{A}}, \overline{F}_{\overline{A}}') \) accepts \( \overline{L} \), since \( \delta'(q_0, x) = \delta'(\varepsilon, x) = [x] \) and \( \gamma'(q_0, x) = \gamma'(\varepsilon, x) = [x] \). Thus \( \overline{L}(\overline{A}') = \overline{L}(\overline{A}) \).

### 4.7 ALGORITHM FOR MINIMIZING DFA-VS

Let \( \overline{A} = (Q, \Sigma, \delta, \gamma, q_0, \overline{F}_\overline{A}, \overline{F}_\overline{A}) \) be a DFA-VS. Assume that \( Q = \{ q_0, q_1, \ldots, q_n \} \), \( n \geq 0 \) and let \( P = \{ (q_i, q_j) \mid q_i, q_j \in Q \) and \( 0 \leq i < j \leq n \} \).

**begin**

\[ \text{Step 1: for each pair } (q_i, q_j) \in P, \text{ and } \overline{F}_\overline{A}(q_i) \neq \overline{F}_\overline{A}(q_j) \text{ or } \overline{F}_\overline{A}'(q_i) \neq \overline{F}_\overline{A}'(q_j) \] do mark \( (q_i, q_j) \);

**Step 2: for each unmarked pair } (q_i, q_j) \in P \) do

if for some \( x \in \Sigma, (\delta(q_i, x), \delta(q_j, x)) \) and \( (\gamma(q_i, x), \gamma(q_j, x)) \) is marked then

**Step 2.1:** mark \( (q_i, q_j) \);

**Step 2.2:** recursively mark all unmarked pairs on the list of \( (q_i, q_j) \)

and on the list of other pairs that are marked at this step.

**else**

**Step 2.3:** for all input symbols ‘x’ do
put \( (q_i, q_j) \) on the list for \( (\delta(q_i, x), \delta(q_j, x)) \) and
\( (\gamma(q_i, x), \gamma(q_j, x)) \) unless \( \delta(q_j, x) = \delta(q_j, x) \) and \( \gamma(q_i, x) = \gamma(q_j, x) \).

**Step 3:** Equivalence classes of \( Q \) are constructed as follows:

For \( i = 0 \text{ to } n - 1 \) do

For \( j = i + 1 \text{ to } n \) do

if \( (q_i, q_j) \) is unmarked, \( q_j \) is in \( [q_i] \), the equivalence class containing \( q_i \).

**Step 4:** Define a minimum DFA-VS \( \tilde{A}' = (Q', \Sigma', \gamma', q_0', \tilde{F}_{F_A}', \tilde{F}_{F_A}') \) as follows;

\[
Q' = \{ [q_i] \mid q_i \in Q \}, \quad \delta'(\{q_i\}, a) = [\delta(q_i, a)] \quad \text{and} \quad \gamma'(\{q_i\}, a) = [\gamma(q_i, a)],
\]

\[
q_0' = [q_0], \quad \tilde{F}_{F_A}'([q_i]) = \tilde{F}_{F_A}(q_i) \quad \text{and} \quad \tilde{F}_{F_A}'([q_i]) = \tilde{F}_{F_A}(q_i).
\]

end.

The complexity of the above algorithm is \( O(n^2) + 2O(a n^2) \), where \( n \) is the number of states of given DFA-VS and \( a \) the element in the assumed set of input alphabet.

**Example 4.6:**

Let \( \tilde{A} = (Q, \Sigma, \delta, \gamma, q_0, \tilde{F}_{F_A}, \tilde{F}_{F_A}) \) be a DFA-VS (Figure 4.3 (a)). Here, \( Q = \{a, b, c, d, e, f\}, \Sigma = \{0, 1\}, q_0 = \{a\} \) the vague starting state with truth membership value \( \tilde{F}_{F_A}(a) = 0.4 \) and false membership value \( \tilde{F}_{F_A}(a) = 0.5 \). \( \delta, \gamma : Q \times \Sigma \rightarrow Q \) are the transition functions given as

\[
\delta(a, 0) = \gamma(a, 0) = c, \quad \delta(a, 1) = \gamma(a, 1) = b, \quad \delta(b, 0) = \gamma(b, 0) = d, \quad \delta(b, 1) = \gamma(b, 1) = a,
\]

\[
\delta(c, 0) = \gamma(c, 0) = f, \quad \delta(c, 1) = \gamma(c, 1) = e, \quad \delta(d, 0) = \gamma(d, 0) = f, \quad \delta(d, 1) = \gamma(d, 1) = e,
\]

\[
\delta(e, 0) = \gamma(e, 0) = f, \quad \delta(e, 1) = \gamma(e, 1) = e, \quad \delta(f, 0) = \gamma(f, 0) = f, \quad \delta(f, 1) = \gamma(f, 1) = f, \quad \text{and}
\]

\[
\tilde{F}_{F_A}(b) = 0.4, \quad \tilde{F}_{F_A}(c) = 0.6, \quad \tilde{F}_{F_A}(c) = 0.6, \quad \tilde{F}_{F_A}(d) = 0.7, \quad \tilde{F}_{F_A}(d) = 0.7, \quad \tilde{F}_{F_A}(e) = 0.7, \quad \tilde{F}_{F_A}(e) = 0.9 \quad \text{and} \quad \tilde{F}_{F_A}(f) = 0.2, \quad \tilde{F}_{F_A}(f) = 0.3 \quad \text{shows the truth and false membership values of the states \{b\}, \{c\}, \{d\}, \{e\} and \{f\} respectively.}
Figure 4.3 (a): DFA-VS

Figure 4.3 (b): Minimized DFA-VS of Figure 4.3 (a)

Above DFA-VS (Figure 4.3 (a)) and its minimized DFA-VS (Figure 4.3 (b)) will accept the vague regular language;
If vague sets are reduced to fuzzy sets, we will consider only the truth-membership value. In this case, DFA-VS becomes deterministic finite automaton with fuzzy (final) states (DFA-FS). Depending on the state transition and membership value of each state this DFA-FS may reduce further. For reducing DFA-FS we apply the algorithm given in [66]. Again, if fuzzy set is reduced to crisp set, we consider each state with membership value zero in DFA-FS as non-final states and all other states as final states in deterministic finite automaton (DFA). Thus DFA-FS becomes DFA. Again this DFA may get reduced further depending on its states transition and for doing so, we apply the algorithm given in [5].

Furthermore, if vague set is reduced to fuzzy set, vague regular language becomes fuzzy regular language, where each string is having membership value in unit interval. This language can be recognized by DFA-FS & NDFA-FS [66]. Again, on reduction of fuzzy sets to crisp sets, fuzzy regular language becomes regular language and is recognized by DFA & NDFA. Here we will consider only strings of fuzzy regular language having non zero membership as strings of regular language.

DFA-VS (Figure 4.3 (a)) is changed to DFA-FS. This DFA-FS and its minimized DFA-FS recognizing the fuzzy regular language

\[
\tilde{L} = \{1(1)^* /[0.4,0.6], (11)^* /[0.4,0.5], 1^*0/[0.6,0.7], 1^*01^+ /[0.7,0.9], 1^*00(0+1)^* /[0.2,0.3], \\
1^*01^+0(0+1)^* /[0.2,0.3] \}
\]

are given in Figure 4.4 (a) & Figure 4.4 (b) respectively.
DFA-FS (Figure 4.4 (a)) can be changed to DFA. The DFA so changed and its minimized one are given in Figure 4.5 (a) & Figure 4.5 (b) respectively and accepting the regular language $\tilde{L} = \{ 1^*, 1^*01^*, 1^*01^*0(0+1)^*, 1^*00(0+1)^* \}$. 

**Figure 4.4 (a):** DFA-FS

**Figure 4.4 (b):** Minimized DFA-FS of Figure 4.4 (a)
4.8 RELATION BETWEEN INTUITIONISTIC FUZZY LANGUAGE, INTERVAL-VALUED FUZZY LANGUAGE AND VAGUE LANGUAGE

Intuitionistic fuzzy sets, interval-valued fuzzy sets and vague sets are the extensions of fuzzy set. According to Atanassov and Gargov [41], the interval-valued fuzzy sets can be expressed in the form of intuitionistic fuzzy sets and according to Bustince and Burillo [45], the vague sets are equivalent to intuitionistic fuzzy sets. In this section, an attempt has been made to
compare three models that extend fuzzy language theory: intuitionistic fuzzy language (IFL) (discussed in chapter 2), interval-valued fuzzy language (IVFL) (discussed in chapter 3) and vague language (VL) (discussed in this chapter) theory. Our exposition recalls the concept of their membership values resulting in some relations among them. IFL, IVFL and VL when traced back to the underlying mathematical structure they are defined on, subside to the same syntactical entity. Some would consider this equivalence sufficient evidence to dismiss IFS theory as redundant and giving cause to unnecessary confusion. We wish to counter that allegation by demonstrating that the theories are at the crossroads of three important, different traditions. The relations among the membership values of these languages are studied. These relations will facilitate to study the property of one language when the property of another is given.

The difference between IFL and IVFL is due to the definition of their membership values. In IFL, we have \((f_L(w), g_L(w))\) as the membership value of a string ‘w’, where each of \(f_L(w)\) and \(g_L(w)\) represents a value of ‘w’ in \([0, 1]\). These are respectively, the membership and the non-membership values of ‘w’, with the condition that \(0 \leq f_L(w) + g_L(w) \leq 1\). In IVFL the membership value of a string ‘w’ is given by \([f^L_L(w), f^U_L(w)]\), where each of \(f^L_L(w)\) and \(f^U_L(w)\) represents a value in \([0, 1]\). These represents respectively, the lower membership and the upper membership values of ‘w’, with the condition that \(0 \leq f^L_L(w) + (1 - f^U_L(w)) \leq 1\). It has been observed that the semantics of the membership value \(f_L(w)\) of a string ‘w’ in IFL is same as the lower membership value \(f^L_L(w)\) of the string ‘w’ in IVFL and the non-membership value \(g_L(w)\) of the string ‘w’ in IFL is same as \((1 - f^U_L(w))\) (i.e., complement of the upper membership value) of the string ‘w’ in IVFL.

Also, the membership value plays an important role while differentiating IFL and VL. The membership value of a string ‘w’ in IFL is explained above. The membership value of a string ‘w’ in VL is given by \([t_L(w), 1 - f_L(w)]\), where each of \(t_L(w)\) and \(f_L(w)\) represents a value in \([0, 1]\). These are respectively, the truth membership and the false membership values of ‘w’, with the condition that \(0 \leq t_L(w) + f_L(w) \leq 1\). It has been observed that the semantics of the membership value \(f_L(w)\) of ‘w’ in IFL is same as that of the truth membership value
From the above discussion, one can obtain an equality relation between the membership values of the string ‘w’ in aforementioned languages (in sequence IFL, IVFL and VL) as;

\[ f_L(w) = f^{L'}_L(w) = t_L(w) , \text{ and } g_L(w) = 1 - f^U_L(w) = f^{-}_L(w). \]

The above three extensions/generalizations of fuzzy language are having the virtue of complementing fuzzy language, with an ability to model uncertainty as well. Some authors proved IFS, IVFS and VS are equipollent (in logical and philosophical sense). We have tried to equate only the membership values in their respective resultant languages in our study. Intuitionistic fuzzy sets and interval-valued fuzzy sets (intuitionistic fuzzy sets and vague sets) are two approaches of fuzzy sets which are mathematically equivalent; however, they have arisen on different grounds and have different symantics [26], [39], [45]. This is true with their respective languages also. Hence, we have studied the proposed languages separately.

### 4.9 CONCLUSION

This chapter explores another extension of fuzzy language in view of vague sets i.e., one more extension of fuzzy sets. This language keeps the track of both the favouring evidence and opposing evidence of belongingness of a string to a language. In this thesis vague finite automaton to recognize this language is proposed. Vague regular expressions will help us describing the proposed language. It discusses the extended Myhill-Nerode theorem in the framework of VRL also, it explains the method of minimizing DFA-VS through an algorithm. These may contribute to a better understanding of the role of vague (final) states automaton in lexical analysis, decision making, pattern recognition, learning systems and other processes involving the manipulation of imprecise data.

The extensive research being done on intuitionistic fuzzy sets, interval-valued fuzzy sets and vague sets shows a mounting interest in these models. This chapter has attempted to mend the situation by obtaining a relation between the membership values of IFL, IVFL and VL. The theory of IFL, VL, and IVFL may prove to be of relevance in the construction of better models for natural languages.