University of Washington
Department of Civil and Environmental Engineering

DEVELOPMENT AND APPLICATION OF FRESH: A FUZZY REASONING SHELL

Shinji Nishida
Richard N. Palmer

Water Resources Series
Technical Report No.119
December 1989

Seattle, Washington
98195

Department of Civil Engineering
University of Washington
Seattle, Washington 98195

DEVELOPMENT AND APPLICATION OF FRESH: A FUZZY REASONING SHELL

Shinji Nishida
Richard N. Palmer

Water Resources Series
Technical Report No. 119

December 1989

Development and Application of FRESH:

A Fuzzy Reasoning Shell

by Shinji Nishida

and

Richard N. Palmer

Technical Report No. 119
December 1989

Environmental Engineering and Sciences
Department of Civil Engineering
University of Washington

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

The authors wish to express their appreciation to a number of people who aided in the completion of this research. The case study used in this report is based upon a portion of the water supply system of North West Water, which is located in Warrington, England. The authors wish to thank Dr. Susan Walker of North West Water for her help in understanding this system. The authors also wish to thank Carol Slaughterbeck, of the Department of Civil Engineering, at the University of Washington. Ms. Slaughterbeck was a constant source of useful information and help. She was extremely valuable in formulating the original crisp rule base for operation, based on interviews with Susan Walker, that was used as a point of departure for this research.

This report is based upon the Masters thesis of the first author. He would like to express his gratitude for the loving support of his wife and daughter, Chieko and Ai, during the completion of his graduate education at the University of Washington.

CHAPTER ONE

Introduction

Expert systems are user-friendly, interactive computer software that use domain specific knowledge, in the form of rules and heuristics, and inference procedures to solve complex problems. Some specific domains for which expert systems have been designed include the area of medical diagnosis, engineering, decision making, management forecast, and geological exploration. These expert systems consist of a domain specific *knowledge base* and reasoning algorithms known as an *inference engine*. Some systems consist of an inference engine *shell* to which various knowledge bases can be added.

The development of knowledge-based expert systems has become a popular topic for scientists and engineers in the late 1970's and 1980's. Among these systems are MYCIN (Shortliffe, 1974), that diagnoses infectious diseases; PROSPECTOR (Duda et al, 1978), that provides consultation to geologists for ore-grade deposits; and XCON (Kraft, 1984), a tool that configurates DEC VAX-11/780 computer systems. The development of powerful and economical personal computers has accelerated the application of expert systems.

In spite of these successes, the development, maintenance, and enhancement of expert systems are still difficult and complex tasks. Two significant obstacles to the successful development and application of expert systems are knowledge acquisition and the management of uncertainty in the rules and heuristics.

Knowledge acquisition is the process of obtaining knowledge or expertise from human experts and formulating this information into a logic structure that can be interpreted by a computer. This process of knowledge acquisition contains two key technical roles: domain experts and knowledge engineers. A domain expert is a person knowledgeable in a specific field. Domain experts often have obtained a collection of definitions, specialized facts, strategies and heuristics in a narrow domain area. They seldom have experience with expert system technology. A knowledge engineer is the individual who extracts knowledge from domain experts and translates it into expert system rules. Knowledge engineers typically have experience with computer technology, computer languages, and software. They also must have training in interviewing techniques and constructing logic structures for combining knowledge into a decision.

Knowledge acquisition is often cited as an impediment to the development of expert systems because no standard technique exists to extract knowledge from experts (Buchanan and Shortliffe, 1984). In addition, the interpretation of experts' knowledge by the knowledge engineer is always a potential source of error. Therefore, this process requires the extensive efforts of both domain experts and knowledge engineers in development of successful expert systems.

The management of uncertainty is also important in expert systems development because the human decision making process is based on imprecise, incomplete, or unreliable information and rules (Zadeh, 1983). Certainty factors (representing the confidence that one has in a piece of evidence) and propositional logic are used often in expert system shells to deal with uncertainty.

However, the combination of certainty factors and propositional logic to deal with uncertainty has serious shortcomings. It is difficult to justify their use as a truly accurate model of human thought or human heuristics. In recent years the application of *fuzzy logic* has been suggested as a framework for managing uncertainty and possibility in expert systems (Zadeh, 1983). It is around this approach that this thesis centers.

This research develops a fuzzy logic based shell, FRESH (Fuzzy Reasoning Shell) that integrates a menu-driven system, user input screens, graphics, and a fuzzy inference engine. Unlike conventional expert system shells or AI special languages, FRESH allows experts to build a fuzzy reasoning system without extensive computational skills. Therefore, FRESH has a potential to solve the two previously mentioned problems in expert systems development: knowledge acquisition and the management of uncertainty.

FRESH has been used to develop an operational rule base for NWWA (North West Water Authority) reservoir system management in England. The application result shows that FRESH provides an interactive knowledge acquisition environment to construct reservoir operating rules. The proposed operational rule base used here has a very simple production (if-then) rules without forward/backward chaining in rules. However, the present state of fuzzy logic will show that this structure can be extended to more complex settings including forward/backward chaining in the future.

The contents of this thesis are organized as follows: Chapter Two describes two shortcomings of current expert system technology. Chapter Three reviews fuzzy logic principles. Chapter Four compares propagation of uncertainty based on conventional expert systems logic and fuzzy logic. Chapter Five describes the characteristics of FRESH. Chapter Six presents an application result for the

NWWA reservoir system. Chapter Seven contains conclusions and indicates potential further research.

CHAPTER TWO

Limitations of Expert Systems

A conventional knowledge base for an expert system is constructed through the interactions of two groups, domain experts and knowledge engineers. Intuitively, the most direct approach is for an expert to place his or her knowledge directly into a knowledge base. When a knowledge engineer and an expert are not the same person, building a complex knowledge base (one embodying uncertain judgemental knowledge) is a difficult task. Generally, it is very time consuming to construct and debug such a knowledge base system, and the difficulties in constructing such a knowledge base are often underestimated.

One particularly difficult aspect of translating human decision making into an expert system is modeling uncertainty and incomplete information. This problem is fundamental to accurately modeling human decision making and it is an area of research that has many conflicting and competing approaches. Both of the problems, knowledge acquisition and modeling uncertainty, are discussed below.

Knowledge Acquisition

Knowledge acquisition is the most important and most difficult step in developing successful expert systems. This task requires a substantial investment of careful thought and analysis. Because knowledge engineers typically are not experts in the domain in which they are developing an application, the knowledge engineers must overcome communication problems during the process of developing the knowledge base of the expert system. Also, human knowledge or expertise is not always well organized. Thus, the knowledge engineer must be an effective communicator, an organizer, and a logician. The testing and validation of an expert system are also important tasks for knowledge engineers. If the knowledge engineer is unable to adequately extract information from domain experts, the expert system cannot be successful.

There are several strategies to capture knowledge acquisition for expert system development. Table 2-1 summarizes five variations of knowledge acquisition modes suggested by McGraw and Harbison-Briggs (1989).

In the first mode, the knowledge engineer extracts the domain knowledge from human experts and translates it into a knowledge base. In this mode, Parsaye (1985) suggests three major approaches: interviewing experts, learning by

Table 2-1. Variations in Possible Knowledge Acquisition Mode

	KNOWLEDGE SOURCE	KNOWLEDGE ACQUISITION MODE
(1)	Human Experts	Knowledge Engineer
(2)	Data	Knowledge Engineer
(3)	Human Experts	Editing Program
(4)	Human Experts	Learning Program
(5)	Texts	Text Understanding Program

observation, and learning by being told. In the first approach, knowledge engineers have formal or informal meeting with domain experts and extract their expertise. This is the most economical and flexible approach. In the second approach, knowledge engineers observe decision processes of domain experts without interference and extract it through various sample scenarios. Learning by observation permits human experts to concentrate on their decision-making process. The third approach requires domain experts to explain their decision-making process to knowledge engineers. This approach will be successful if human experts can express and refine their rules and heuristics precisely. However, the first two approaches require well-trained knowledge engineers, and all approaches require a great amount of time.

The second mode suggests that the knowledge engineer studies more formal sources of information and encodes it into a knowledge base. Textbooks, journal articles, experimental data, videotapes, and other such material are typical examples of where information can be found. This mode does not require the contribution of human experts; however, for many areas of interest, no sources of such information exist. Thus, this mode is often used as a supplement to the first mode.

The combination of the first two modes is often used in conventional expert system development. However, it requires the attention and efforts of both domain experts and knowledge engineers. The interpretation of the experts' knowledge by the knowledge engineer is always a potential source of error.

The third mode suggests another approach for knowledge acquisition. In this mode, the human experts construct a knowledge base directory using their rules and heuristics with the aid of an editing program. Two types of commercial editing programs are currently available for the personal computer environment: expert system shells, such as INSIGHT2+ (Level Five Research, Inc., 1986) and VP-EXPERT (Paperback Software International, 1987); and AI special languages, such as LISP and PROLOG. In the case of using conventional expert system shells or AI special languages, the human expert must have extensive computational skills to develop useful knowledge-based expert systems. This is the reason why most expert systems are developed by a combination of the first two modes. However, if an acceptably intelligent editing program is available, this approach is the most precise, cost effective, and efficient prototype because it can avoid the intermediate steps that require the knowledge engineers.

One approach to aid constructing a knowledge base is an *induction table*. An induction table represents production (IF-THEN) rules in a simple manner. Using this procedure, users can create a knowledge base. Expert system shells that have this feature include VP-EXPERT and EXSYS (Exsys, Inc., 1986). However, experts must learn a considerable amount about programming before such techniques can be used to develop useful knowledge bases.

The fourth mode represents "machine learning." In this mode, the role of human experts is to answer questions based on various scenarios and case studies presented by learning program. Then, the learning program reviews the answers, and helps to construct appropriate rules in the knowledge base. AQUINAS (Boose et al, 1988) is one of the existing systems used at the Boeing Company. AQUINAS helps human experts to organize, analyze, test, and refine their knowledge base through direct interview with human experts.

The final mode suggests that a sophisticated computer reads and understands the texts and builds a knowledge base automatically. Limitations of current computer technology do not allow the development of this type of extraordinarily sophisticated systems.

The last two modes represent a dramatically different approach. These modes require computer technology not currently available. Several laboratory systems exist in the fourth mode; however, these are in their infancy, not yet completely successful, and not available in the commercial market. Thus, knowledge acquisition is still handled in the first three modes in actual expert system development.

Uncertainty and Approximate Reasoning

Uncertainty in data and reasoning is another serious problem in expert system development. In most instances, facts, rules, and conclusions in a knowledge base contain a degree of uncertainty, and it is critical that expert systems have the capability to quantify this uncertainty and trace its influence on the decision making process.

To address this problem, most existing expert systems adopt a parameter called a certainty factor. Certainty factors are not purely probabilities, but an informal *probability-like* measure of confidence or evidence. The first use of certainty factors was in the MYCIN project (Buchanan and Shortliffe, 1984). In MYCIN, certainty factors range from -1 to +1, and rules also have a certainty factor associated with them. The inference engine of MYCIN handles the computation of certainty factors through a combination of methods which are based on predicate logic and certainty factors. The concept of certainty factors was implemented and tested in MYCIN and became a central element of this system (Buchanan and Shortliffe, 1984).

General concepts of certainty factors (CF) can be explained using the following two simple rules:

Rule 1

If: (1) x is A, AND

(2) y is B

Then: (3) z is C with CF = 0.8

Rule 2

If: (1) x is D, OR

(2) y is E

Then: (3) z is F with CF = 0.5

where (1) and (2) are antecedent objects and (3) is a consequence object.

Table 2-2 presents the truth table of these two rules under uncertain information. If antecedents are connected to each other by AND, the minimum CF must be propagated to the consequence. In this case, if the user input answers *false* in any part of the antecedent, the rule fails. On the other hand, if antecedents are connected by OR, the maximum CF can be propagated to the consequence. In this

Table 2-2. Truth Table of <u>Rule 1</u> and <u>Rule 2</u>
(T: true, F: false)

<u>Rule</u>	<u>Antecedents</u>				<u>Consequence</u>	
	(1)	CF	(2)	CF	(3)	CF
1	T	0.4	T	0.6	T	min(0.4, 0.6)*0.8 = 0.32
	F	---	T	1.0	F	---
	T	0.7	F	---	F	---
	F	---	F	---	F	---
2	T	0.4	T	0.6	T	max(0.4, 0.6)*0.5 = 0.3
	F	---	T	1.0	T	1.0*0.5 = 0.5
	T	0.7	F	---	T	0.7*0.5 = 0.35
	F	---	F	---	F	---

case, the rule may *succeed* even if the user answered *false* to the question related to one antecedent of this rule.

It is recognized that the certainty factor model has serious shortcomings, and is hard to justify their use in heuristic rules because much of the uncertainty in expert systems is a function of possibility (Zadeh, 1983) rather than just a measurement of confidence. For example, the following type of rule is often used in water resources management expert systems:

IF: Current storage is less than 50 % of net capacity
THEN: Take an action for drought management with CF = 0.5.

Since the conventional knowledge based systems, which neglect the concept of fuzzy set theory, cannot deal with *partial matching*, the uncertainty associated with the certainty factor does not affect the antecedent object. For example, this rule succeeds when storage volume is 49.9 % of net capacity, but fails when storage volume is 50.1 %. What is needed is a graduation of action that deals with partial matching through the use of the compositional rule of reasoning.

Another important issue in dealing with uncertainty in knowledge based systems is that of quantifiers such as, *very, more or less, most, many,* and *few*, when they are used in the antecedent objects and/or the consequence object of a rule. Most heuristic rules in expert systems contain these fuzzy quantifiers, and thus the inference engine of an expert system shell should handle these fuzzy propositions.

Approach to Uncertainty Management

This research develops a user friendly environment, (FRESH - Fuzzy Reasoning Shell) that integrates a menu-driven system, user input screens, graphics, and a fuzzy logic based inference engine. FRESH provides an interactive knowledge acquisition environment and a systematic framework for dealing with fuzzy quantifiers. Unlike conventional expert system shells or AI special languages, FRESH allows experts to build fuzzy reasoning systems without extensive computational skills. Therefore, FRESH has a potential to solve the two previously mentioned serious problems in expert systems development: knowledge acquisition and management of uncertainty. Before describing FRESH, a review of fuzzy logic is provided in the Chapter Three.

CHAPTER THREE

Principle of Fuzzy Mathematics

Fuzzy set theory has gained significant recognition from researchers in a wide range of disciplines since its introduction by Zadeh (1965). An important feature of fuzzy set theory is the simplification of complex decision making systems. This is accomplished by incorporating uncertainty into decision making with the use of fuzzy membership functions. These functions are only approximations of complex decision making systems, but their relevance to the original systems is fully maintained.

Many decision making situations cannot be modeled easily, or with any degree of accuracy, using conventional techniques (Zadeh, 1983). This is because the world is full of uncertainties. Since reasoning under uncertainty occupies a central role in decision making, a variety of tools are required to deal with uncertain information. Fuzzy logic is one of these approaches. Indeed, although probability theory is often used to analyze uncertain information, the knowledge or the meaning of certain empirical facts or data is never only probabilistic, but also involves some form of approximate and vague reasoning.

In this chapter, two basic concepts of fuzzy mathematics, fuzzy sets and fuzzy logic, are briefly described using a concise comparison with classic mathematics. FRESH (Fuzzy Reasoning Shell) incorporates these two principals into a decision making context.

Fuzzy Sets

Fuzzy sets are concise mathematical statements of the approximate nature of physical objects or concepts. Such sets underlie much of our ability to summarize, communicate, and make decisions under uncertain or partial information. Indeed, fuzzy sets appear to play an essential role in human cognition, especially in relation to concept formation, pattern classification, and logic reasoning (Kaufmann and Gupta, 1988).

A classic set is defined into two groups; members that certainly belong in the set and nonmembers that certainly do not. A crisp, clear distinction exists between the members and nonmembers of the class or category. Many of the classes and categories we commonly employ, however, such as the class of heavy rain, dry weather, small reservoir storage, or severe drought, do not exhibit this property. Instead, their boundaries seem vague, and the transition from member to nonmember appears gradual rather than abrupt. Thus, the fuzzy set introduces vagueness by eliminating the crisp boundary dividing members of class from nonmembers.

A fuzzy set can be defined mathematically by assigning a value representing its grade of membership to each possible individual in the fuzzy set. This grade corresponds to the degree to which that individual is similar or compatible with the concept represented by the fuzzy set. These membership grades are represented by real number values ranging in the closed interval between zero and unity.

For example, let A be a subset of the universe of discourse X. The characteristic function $M_A(x)$ of classical subset A takes its values in the two-element set $\{0, 1\}$, and is such that $M_A(x) = 1$ if $x \in A$ and $M_A(x) = 0$ otherwise. A fuzzy set A has a characteristic function $M_A(x)$ taking its values in the interval $[0, 1]$, with the grade 1 and 0 representing full membership and nonmembership, respectively. Thus, $M_A(x)$ is called the membership function of A. Since $M_A(x)$ is the grade of membership of $x \in A$, it is also possible to interpret $M_A(x)$ as the degree of possibility (Zadeh, 1983) that x is the value of a parameter fuzzily restricted by A. A is symbolically denoted $[\{x, M_A(x)\} | x \in A]$. Other convenient notations are:

$$A = \int M_A(x)/x \qquad \text{when } X \text{ is continuum.} \tag{3-1}$$

$$A = M_A(x_1)/x_1 + \ldots\ldots + M_A(x_n)/x_n = \Sigma_i\ M_A(x_i)/x_i \tag{3-2}$$

when X has n elements ($|x| = n$),
where + is the union of $M_A(x_i)/x_i$'s.

In a fuzzy set, the transition between membership and non-membership is usually gradual rather than abrupt. The shape of membership functions vary. For practical application, however, simple triangular and trapezoidal shapes are the best (Kaufmann and Gupta, 1988) because the use of nonlinear membership function leads to more complex computations. It is important to note that X is not fuzzy. This implies that X is a non-fuzzy set of the universe discourse and A is a fuzzy set derived from that universe. For instance, X could be the amount of rainfall, and A could be the linguistic value *heavy rain*.

Several fundamental structures can be defined on the interval [0, 1], such as fuzzy set union, intersection, and complement operator. These functions coincide with those used in classical set theory. Zadeh (1965) first used the operator {[0, 1], max, min, not} that are the most frequently employed, i.e.:

Union : $M_{A \cup B}(x) = \max\{M_A(x), M_B(x)\}$ (3-3)

Intersection: $M_{A \cap B}(x) = \min\{M_A(x), M_B(x)\}$ (3-4)

Complement : $\Gamma M_A(x) = 1 - M_A(x)$ (3-5)

Figure 3-1 illustrates these relationships.

Fuzzy Logic

Fuzzy logic is a natural expansion of classical logic that every proposition is either *true* or *false*. Zadeh (1973) proposed a fuzzy logic in which the antecedent involves a fuzzy conditional proposition. For instance, "If x is A then y is B", with A and B being fuzzy concepts. Such a structure allows the development of simple rules typical of those found in expert systems. However, the fuzzy logic is significantly more complex than that used in most classical propositional logic. This complexity allows the development of much more subtle and accurate management of uncertainty.

Logic is the study of the methods and principles of *reasoning* in all its possible form. One area of classical logic is *propositional logic* that requires an object to be either *true* (unity) or *false* (zero). These values are termed *truth values*. Propositional logic deals with combinations of *objects* (or variables) that represent

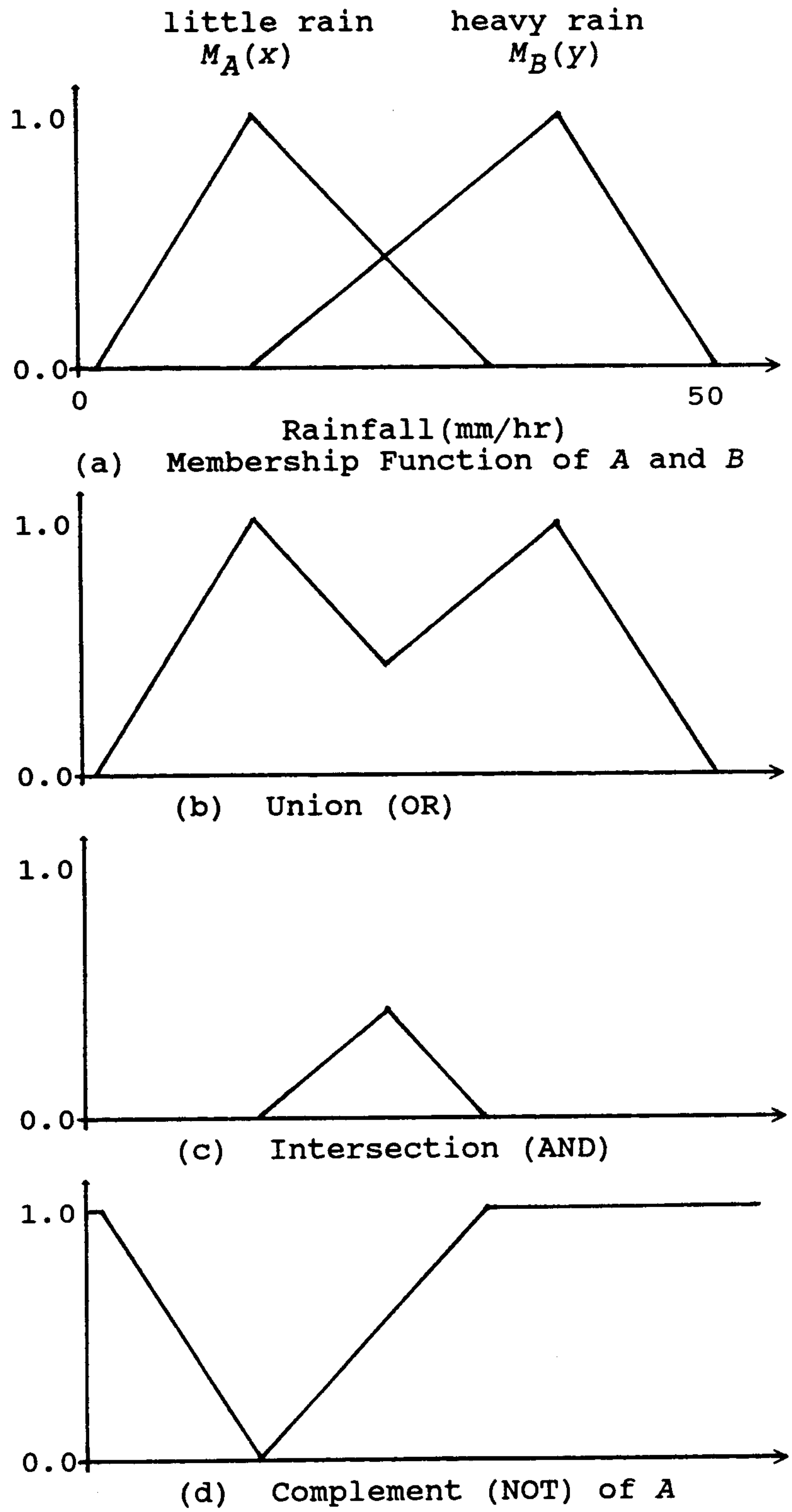


Figure 3-1. The Basic Mathematics of Fuzzy Sets

arbitrary propositions. As each object represents a hypothetical proposition, it may assume either of the two truth values. Various forms of *reasoning rules* (or inference rules) can be used for making deductive reasoning.

Modus ponens is one of the most frequently used reasoning rules, i.e.:

$$\{A \cap (A \Rightarrow B)\} \Rightarrow B$$

where ∩ is *and-function* (or conjunction) and => is *implication*. Modus ponens states that given two true propositions A and $A \Rightarrow B$ (the *antecedents*), the truth of the proposition B (the *consequence*) may be inferred.

The basic assumption of classical logic, that every proposition is either true or false, has long been questioned. This occurs because truth values of certain propositions are essentially indeterminate due to fundamental limitations of measurement. For instance, the following type of rule is often used in water resources management expert systems:

IF current storage is less than 40 % of capacity
THEN take an action for drought management.

The antecedent is true if the storage is 39.9 %, and is false if the storage is 40.1 %. However, near the 40 % of capacity, it is very difficult to distinguish that the antecedent is true or false. Moreover, this is a forced distinction based on a crisp logic.

In order to deal with more flexible propositions, a fuzzy logic was formulated by Zadeh (1973). This flexibility allows the development of much more complex and natural management of uncertainty in propositional logic. The membership grade $M_A(x)$ for $x \in A$ can be interpreted as the truth values of the proposition.

First, we shall consider the following simple form of approximate reasoning that includes a fuzzy conditional proposition:

Ant : IF current storage is *small*.
THEN drought condition is *severe* is *quite true*.
Fact: Current storage is *about* 10 % of capacity.

Cons: Drought condition is *severe* is *fairly true*.

This is a meaningful deductive reasoning. In order to deal with this type of reasoning, fuzzy logic allows the use of *fuzzy predicates* (small, large, severe, moderate, etc.), *fuzzy quantifiers* (very, less, many, few, about, etc.), and *fuzzy truth values*, equivalent to certainty factors in conventional expert systems, (quite true, very true, mostly false, etc.). More generally, Ant : IF x is A

THEN y is B WITH TV.

Fact: x is $A_{\prime}$.

---

cons: y is B'.

where x and y are the names of objects; A, A' ("similar to" A), B, and B' ("similar to" B) are the labels of fuzzy subsets in the universes of discourse X, X, Y, and Y, respectively; and TV is the truth value of the antecedent. This reasoning scheme is called *generalized modes ponens*, and has two major differences from classical modes ponens (Bellman and Zadeh, 1977). First, A' is not required to be identical with A, as it is in the classical case. Second, A, A', and B are not required to be crisp.

The implied relation between the two objects x and y is expressed in terms of the cartesian product of the two fuzzy subsets A and B. The cartesian product, characterized by a function $M_R(x,y)$ by which each pair (x,y) is assigned a number in [0, 1] indicating the extent to which the relation R is true for (x,y), of two fuzzy subset A and B is denoted as $A*B$. Equations (3-6) and (3-7) illustrate two different definitions of cartesian product; the logical-product (Mamdani, 1977) and the algebraic-product (for example, Dubois and Prade, 1980):

Logical-Product

$$A*B = \Sigma_i \Sigma_j \min \{M_A(x_i), M_B(y_j)\} \tag{3-6}$$

Algebraic-Product

$$A*B = \Sigma_i \Sigma_j \{M_A(x_i) * M_B(y_j)\} \tag{3-7}$$

where Σ stands for union; x_i and y_j are generic elements of the universes of discourse X and Y, respectively; $i = 1, 2, \ldots, m$; and $j = 1, 2, \ldots, n$.

A typical reservoir operation rule that is previously mentioned can be written as:

IF current storage is small

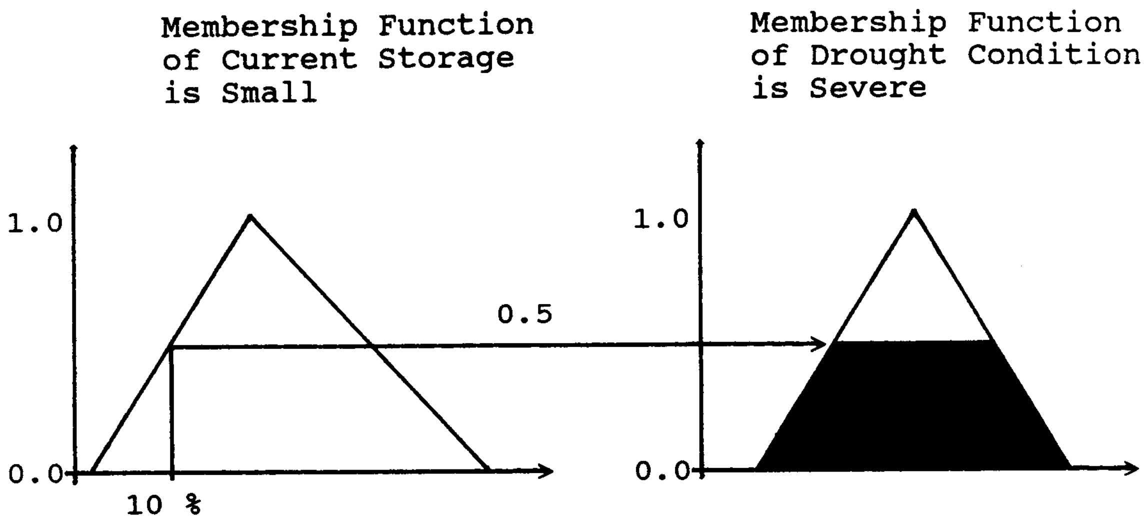
THEN drought condition is severe.

Equations (3-6) and (3-7) can be used to infer the fuzzy subset of the second object given a membership function, or a value, for the first. For example, "Current storage is 10 % of capacity. What is the drought condition ?" Suppose we denote the relationship between two objects by R_{AB} ($A \Rightarrow B$). Then, 10 % of capacity is the given membership function of current storage A' and the membership function of drought condition B' is inferred by:

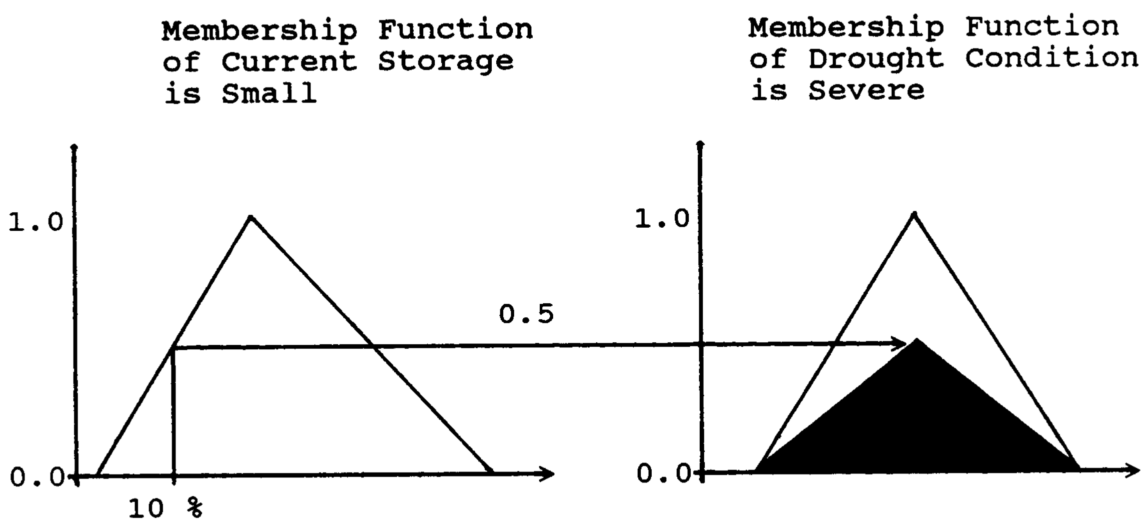
$$B' = A' \circ R_{AB} \tag{3-8}$$

where o is composition. Composition is performed by the maximum-product of A' and R_{AB} (See Appendix A for more details). In the present application, A' is chosen to be nonfuzzy vectors with only one element equal to unity, all the rest being zero. Figure 3-2 shows the reasoning results by both the logical-product and the algebraic-product. The difference between these two methods is the shape of the membership function (or possibility distribution) of the reasoning results. The result by the logical-product shows that the upper portion of the consequence membership function is *truncated* by the grade of the antecedent membership function associated with the given fact, and the shape of membership function is changed to trapezoidal. On the other hand, the result using the algebraic-product illustrates that the consequence membership function is algebraically *multiplied* by the grade of the antecedent membership function associated with the given fact, and the shape of membership function is maintained triangle.

Sugeno (1984) suggests that using the algebraic-product can improve reasoning results because of its property that maintain the original shape of the consequence membership function. Therefore, the algebraic-product is adopted as a fuzzy reasoning operator for this research.



(a) Logical Product



(b) Algebraic Product

Figure 3-2. Reasoning Results under the Fact, Current Storage is 10 % of Capacity

The implementation of truth values (*TV*) to fuzzy conditional propositions is accomplished in same manner as the implementation of certainty factors (*CF*) in conventional expert systems. Extending the previous example mentioned before:

IF current storage is small
THEN drought condition is severe WITH *TV* = 0.5.

The reasoning is "current storage is 10 % of capacity. what is drought condition ?". Figure 3-3 shows the reasoning result with *TV* using the algebraic-product. First, the grade of the membership function of the consequence object is multiplied by the truth value (*TV* = 0.5). Second, the reasoning result of the consequence membership function is multiplied again by the grade of the antecedent membership function associated with the given fact, the current storage is 10 % of capacity.

More complex rules are developed by use of two fuzzy antecedent objects, x1 and x2, that are connected by AND.

Ant : IF $x1$ is A AND $x2$ is B
THEN y is C.
Fact1: $x1$ is A'.
Fact2: $x2$ is B'.

Cons : y is C'.

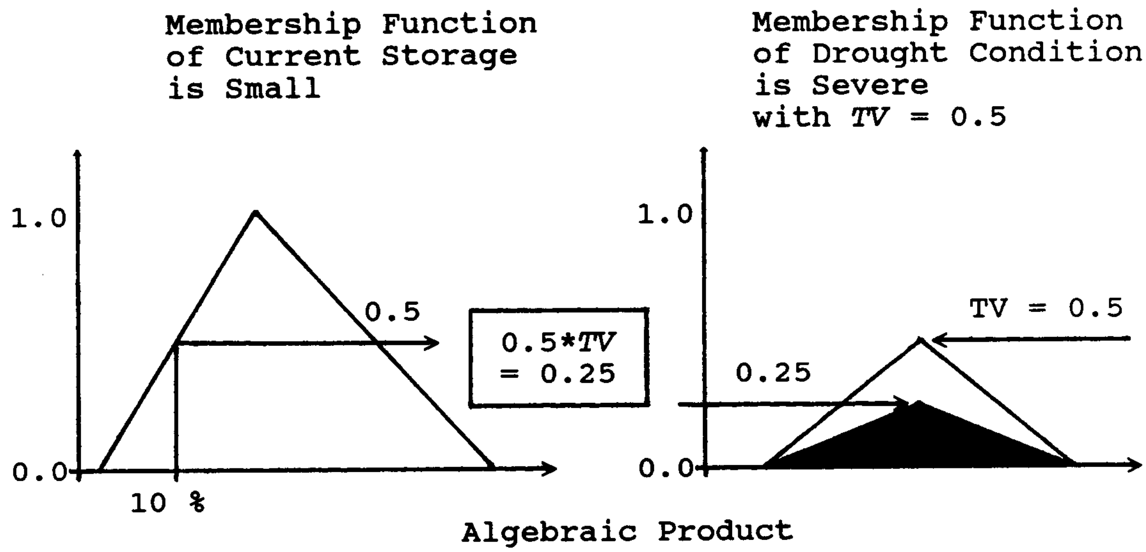


Figure 3-3. Reasoning Result with *TV* under the Fact, Current Storage is 10 % of Capacity

For instance,

IF current storage is small

AND demand is large

THEN drought condition is severe.

Current storage is 10 % of capacity.

Demand is 120 % of base demand.

What is Drought condition ?

In this example, the algebraic-product and composition can be defined as:

$$A*C \text{ AND } B*C = \Sigma_i \Sigma_k \{M_A(x1_i) * M_C(y_k)\} \cap \Sigma_j \Sigma_k \{M_B(x2_j) * M_C(y_k)\} \tag{3-9}$$

$$C' = (A' \circ R_{AC}) \cap (B' \circ R_{BC}) \tag{3-10}$$

Equation (3-10) implies that this example can be divided into the two simple propositions previously mentioned. Thus, each composition is performed by the maximum-product of the antecedents, A' *and* R_{AC}, *and* B' *and* R_{BC}, respectively. Then, the AND operator is interpreted as the *minimum* (or *intersection*) operation, and connects these two separate propositions. Figure 3-4 shows the reasoning result by the algebraic-product.

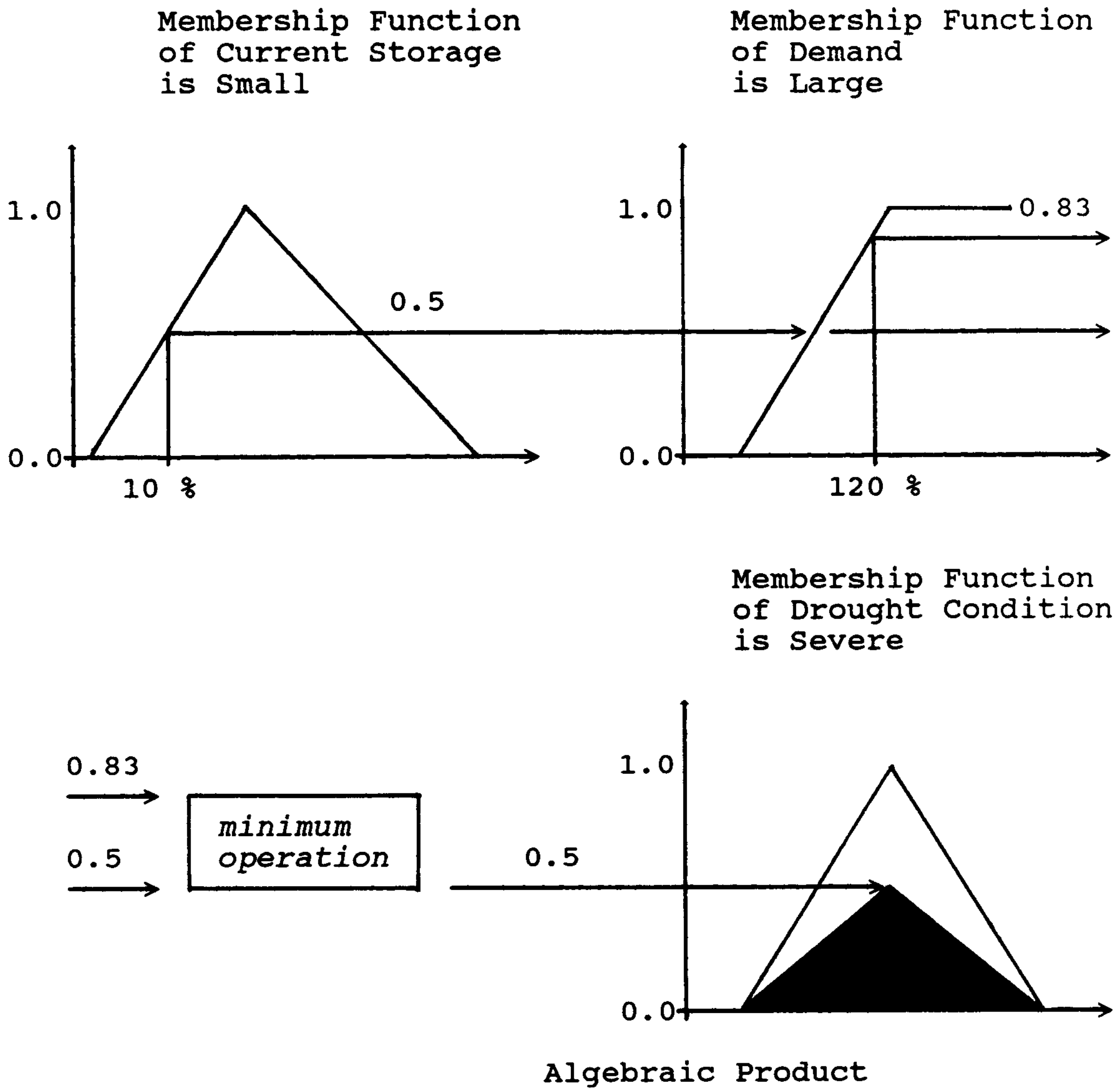


Figure 3-4. Reasoning Result under the Facts, Current Storage is 10 % of Capacity AND Demand is 120 % of Base Demand

Finally, the OR operator is considered with the AND operator.

Ant : IF ($x1$ is A AND $x2$ is B) OR ($x3$ is C)
THEN y is D.
Fact1: $x1$ is A'.
Fact2: $x2$ is B'.
Fact3: $x3$ is C'.

Cons : y is D'.

This form of fuzzy reasoning can be divided into the following two sets of fuzzy conditional propositions.

Ant 1: IF $x1$ is A AND $x2$ is B
THEN y is D.
Fact1: $x1$ is A'.
Fact2: $x2$ is B'.

Cons1: y is D'.

OR

Ant 2: IF $x3$ is C
THEN y is D.
Fact3: $x3$ is C'.

Cons2: y is D'.

Thus,

Cons: Cons1 OR Cons2.

For instance,

IF *(current storage is small* AND *demand is large)*

OR *(predicted storage is small)*

THEN *drought condition is severe.*

Current storage is 10 % of capacity.

Demand is 120 % of base demand.

Predicted storage is 20 % of capacity.

What is drought condition ?

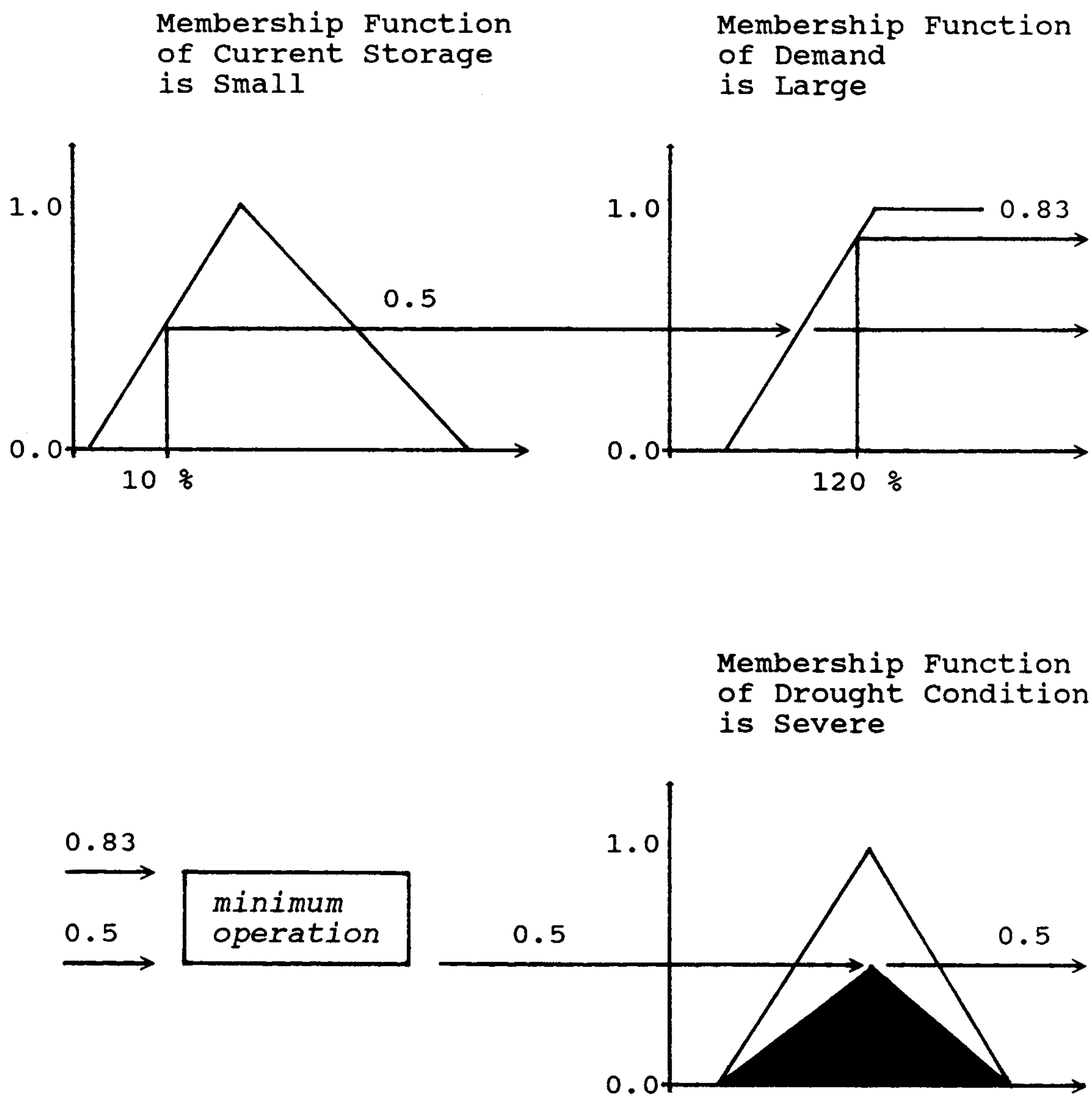
The algebraic-product and composition is defined as:

$$(A*D \text{ AND } B*D) \text{ OR } C*D$$
$$= [\Sigma_i \Sigma_l \{M_A(x1_i) * M_D(y_l)\} \cap \Sigma_j \Sigma_l \{M_B(x2_j) * M_D(y_l)\}]$$
$$\cup [\Sigma_k \Sigma_l \{M_C(x3_k) * M_D(y_l)\}] \qquad (3\text{-}11)$$

$$D' = [(A' \circ R_{AD}) \cap (B' \circ R_{BD})] \cup [(C' \circ R_{CD})] \qquad (3\text{-}12)$$

Composition and the AND operator are performed in same manner. Then, the OR operator is interpreted as the *maximum* (or *union*) operation, and connects these two sets of fuzzy conditional propositions. Figure 3-5 illustrates the reasoning result.

In general, we shall consider the following multiple fuzzy conditional propositions that have several different fuzzy propositions combined with the OR operator, such as:



(continue to next page)

Figure 3-5(1). Reasoning Result under the Facts,
Current Storage is 10 % of Capacity
AND Demand is 120 % of Base Demand
AND Predicted Storage is 20 % of Capacity

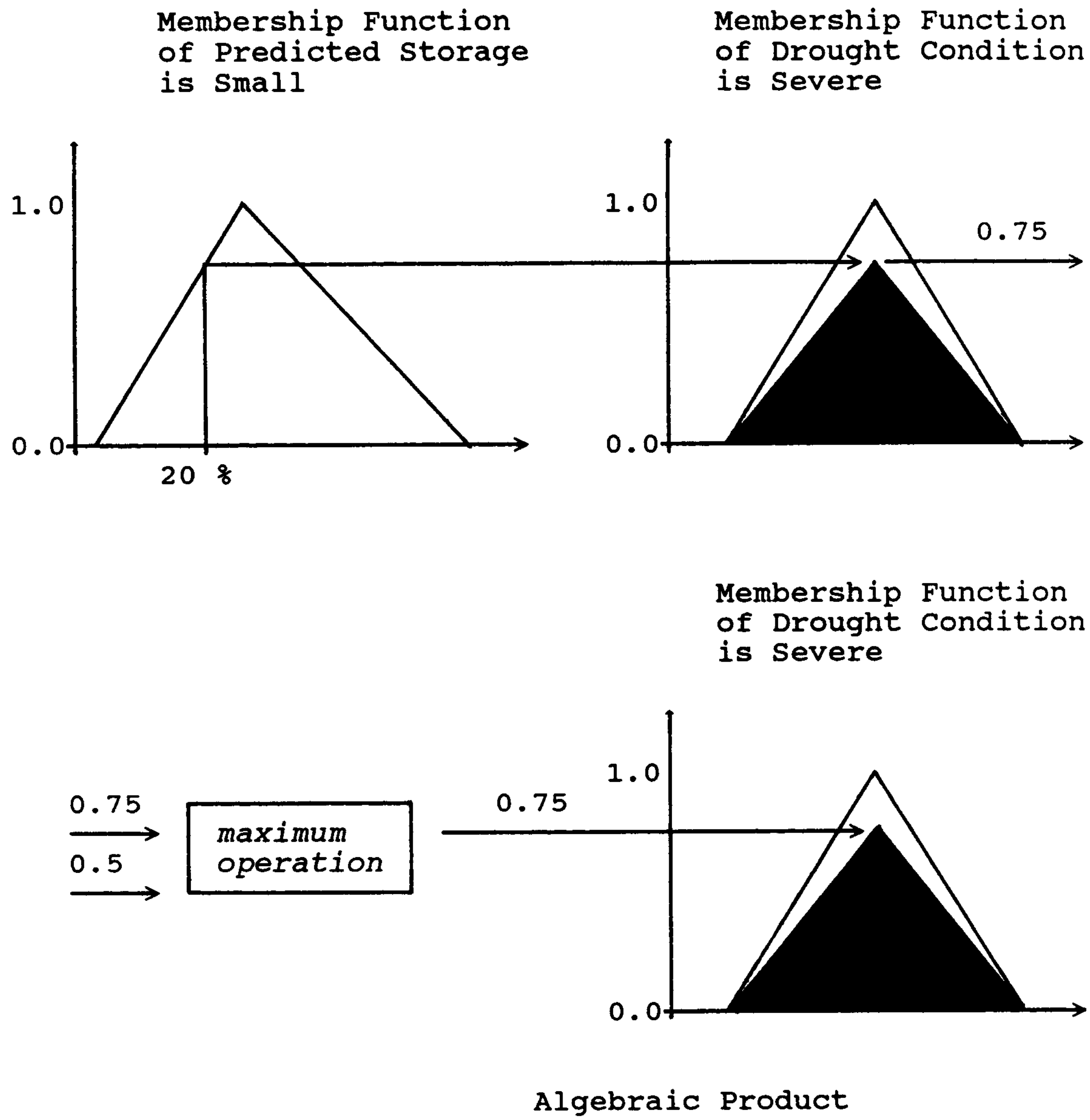


Figure 3-5(2). Reasoning Result under the Facts,
Current Storage is 10 % of Capacity
AND Demand is 120 % of Base Demand
AND Predicted Storage is 20 % of Capacity

```
Proposition 1:  A1  AND  B1  AND  C1  =>  X1  OR
Proposition 2:  A2  AND  C2           =>  X2  OR
                ...........................
Proposition n:  Jn  AND  Ln           =>  Xn.
Fact:           A', B', C', ... , and L'.
------------------------------------------------
Consequence:                              X'.
```

The consequence X' by given facts is interpreted as the union of X1', X2', ..., and Xn'. For instance,

Storage is very small AND demand is high
=> drought is very severe.
Storage is small AND inflow is small
=> drought is severe.
....
Storage is medium AND month is July
=> drought is moderate

In expert system applications, the interest in fuzzy reasoning is that under the given condition, what is the most relevant consequence? In this example, what is the drought condition, very severe, severe, or moderate? To accomplish this purpose, combined reasoning result X' must be compared with pre-defined membership functions of consequence X1, X2, and Xn, such as very severe, severe, and moderate. One can decide then which consequence is the most relevant. The following *matching degree* is proposed for this research to measure relevance of consequences.

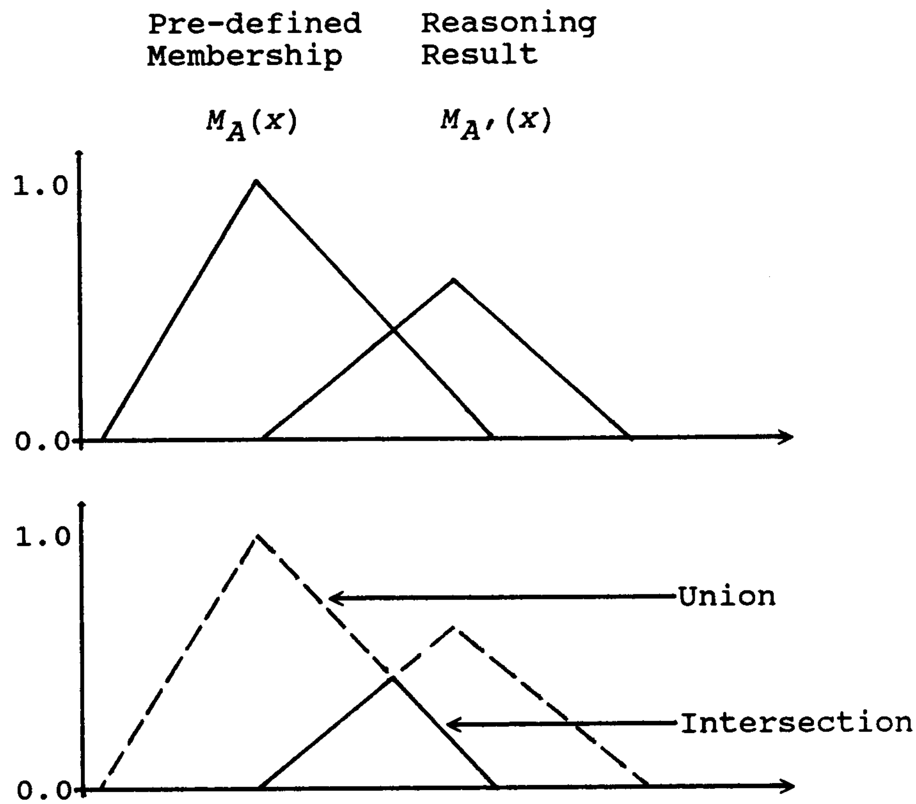
$$\text{Matching degree} = \text{Intersection} \,/\, \text{Union} \tag{3-13}$$

Figure 3-6 illustrates the definition of the matching degree. Under this criterion, perfect match must be 1.0 because such a case the union and intersection is identical.

Summary

The principals of fuzzy logic have been discussed in this chapter. The application of the principals to a simple water resource example also has been demonstrated. In addition, the development of simple rule based logic similar to that used in expert systems has been described and the use of fuzzy reasoning operators has been explained. This logic allows a single conceptual framework to deal with different types of uncertainty. Probability theory was long thought to be the only numerical approach to deal with uncertain information. However, an extensive amount of recent applications of fuzzy logic shows that this logic has changed the perceptions of many scientific researchers.

The following chapter describe differences between propagation of uncertainty using certainty factors in conventional expert systems and membership functions using fuzzy logic.



$$\text{Matching Degree} = \frac{\text{Intersection}}{\text{Union}}$$

Figure 3-6. Definition of Matching Degree

CHAPTER FOUR

Comparison Between Conventional Expert System's Logic and Fuzzy Logic

Interest in expert systems that are based on fuzzy set theory and fuzzy logic has recently increased. The differences between the propagation of uncertainty using certainty factors in expert systems and membership functions using fuzzy logic are of particular interest. Few comparative studies exist that evaluate this important topic.

This chapter describes the conceptual differences of these two techniques for uncertainty propagation. It also describes the propagation of uncertainty using a simple multi-layer reasoning (chaining of the rules) example.

Conceptual Differences Between Certainty Factor Model and Fuzzy Logic

Zadeh (1983) suggests that the most important issues associated with uncertainty information and rules in expert systems are 1) the fuzziness of antecedents and/or consequences in rules, and 2) the natural way of partial matching between the antecedent of rule and a fact supplied by the user. These are discussed below.

1) The fuzziness of antecedents and/or consequence in rules.

IF x is A THEN y is B WITH $TV = \alpha$.

where the antecedent (x is A), and the consequence (y is B) are fuzzy propositions and α is a truth value, TV (equivalent to a certainty factor, CF, in conventional expert systems). For example, the following type of rule is often used in water resources management expert systems:

IF storage is about less than 30 % of net capacity
THEN drought condition is severe
WITH TV = 0.7,

in which the antecedent "storage is about less than 30 % of net capacity" and the consequence "drought condition is severe" are fuzzy propositions because these objects involve the concept of fuzzy set theory.

Perhaps the best known and documented expert system is MYCIN (Buchanan and Shortliffe, 1984). These researchers chose not to use fuzzy logic to manage uncertainty propagation, despite their familiarity with the fuzzy approaches. They suggest that, "Although an analogy between our diagnostic problem and fuzzy set theory can be made, the statement of diagnostic decision criteria in terms of set membership does not appear to be a natural concept for the experts who must formulate our rules." However, this does not lead necessarily to the conclusion that fuzzy set theory cannot contribute in expert system applications.

2) The natural way of partial matching between the antecedent of rule and a fact supplied by the user.

Often there are situations in which a fact such as "x is A'" may not match exactly the antecedent of any rules of the form "IF x is A THEN y is B WITH $CF = \alpha$", where A' can also be a membership function in fuzzy logic. Expert systems using fuzzy set theory can deal naturally with partial matching using continuous membership functions.

The certainty factor model in conventional expert systems emulates partially the fuzzy set approach, but has several limitations. For example the user can imply uncertainty to the statement "storage is somewhat less than 30 % of net capacity" by assigning a certainty factor of 0.5 to be associated with the rule "IF storage is somewhat less than 30 % of net capacity THEN drought condition is severe". This rule is a fuzzy proposition. However, in this example, expertise is shared by the users as well as the expert system because the user must assume the responsibility in answering this question using a fuzzy set concept (under a given situation, the user must imagine the shape of membership functions and enter a *CF* according to his expertise). Also, the certainty factor model does not deal with membership as input by the user. This requires that the propagation of uncertainty in the certainty factor model be done with a single number rather than the entire shape of membership function (distribution of possibility).

Comparison of Propagation of Uncertainty in Multi-Layer Reasoning

The propagation of uncertainty in multi-layer reasoning (chaining of rules) using fuzzy logic is an active area of research (Dubois and Prade, 1989). There are several procedures available to define the functions of union and intersection as well as combining rules for OR and AND in fuzzy logic. In this research, the procedures suggested by Mamdani (1977) are used to define the OR and AND operators.

The fundamental techniques described in the previous chapter have been implemented in a simple two-layer forward chaining over propositions. In this section, an example of its application and the results concerning an efficient procedure for a generalized modus ponens are compared with the certainty factor model. The objectives for presenting this example are to demonstrate: 1) the difference between fuzzy logic and certainty factors in propagating uncertainty, and 2) how certainty factor models can emulate fuzzy logic.

The illustrative example concerns determination of release during drought. The following rules are represented by the tree structure in Figure 4-1.

Rule 1:

IF predicted rainfall is small
THEN future inflow is small
WITH TV = 1.0.

Rule 2:

IF predicted rainfall is large
THEN future inflow is large
WITH TV = 1.0.

Rule 3:

IF future inflow is small and storage is small
THEN release is small
WITH TV = 1.0.

Rule 4:

IF future inflow is large and storage is small

THEN release is medium

WITH TV = 1.0

Rule 5:

IF future inflow is small and storage is large

THEN release is medium

WITH TV = 0.5

Rule 6:

IF future inflow is large and storage is large

THEN release is large

WITH TV = 1.0

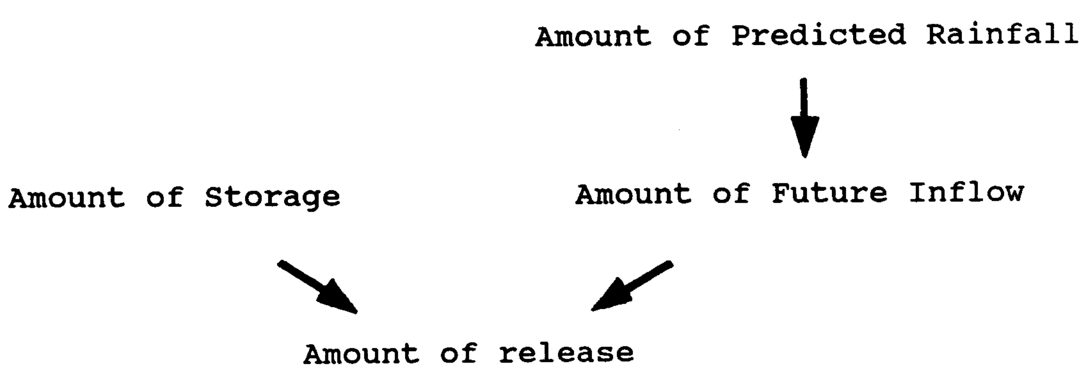


Figure 4-1 Tree Structure of Rules

Fuzzy Logic Approach

In the application of fuzzy logic, one must define the shape of membership functions for each object, predicted rainfall, future inflow, storage, and release. These membership functions can be continuous functions, however, in this example they are treated as discrete functions for simplicity. The shape of membership functions for each object are defined in Figure 4-2. Each object takes its value from zero to 100 with 25 intervals. This is relative scale and values have no units.

Using the notation for membership functions and cartesian products described in Chapter Three, the rules can be expressed in the following matrices:

	value:	0	25	50	75	100
Predicted Rainfall is small:	M_{A1} =	[1.0	0.6	0.3	0.1	0.0]
Predicted Rainfall is large:	M_{A2} =	[0.0	0.1	0.3	0.6	1.0]
Future Inflow is small:	M_{B1} =	[1.0	0.8	0.5	0.2	0.0]
Future Inflow is large:	M_{B2} =	[0.0	0.2	0.5	0.8	1.0]
Storage is small:	M_{C1} =	[1.0	0.9	0.5	0.1	0.0]
Storage is large:	M_{C2} =	[0.0	0.1	0.5	0.9	1.0]
Release is small:	M_{D1} =	[0.5	1.0	0.5	0.0	0.0]
Release is medium:	M_{D2} =	[0.0	0.5	1.0	0.5	0.0]
Release is large:	M_{D3} =	[0.0	0.0	0.5	1.0	0.5]

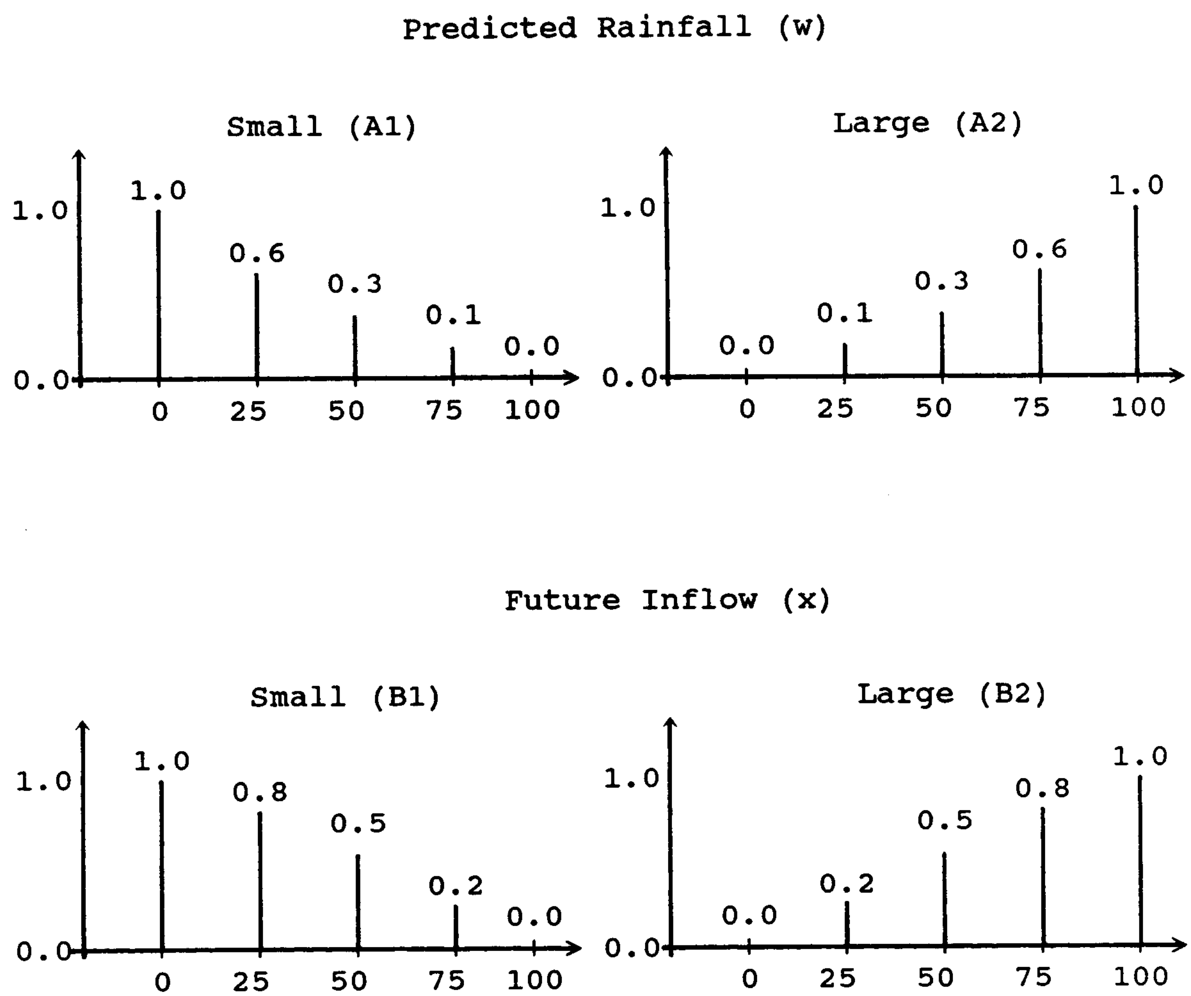


Figure 4-2(1). Membership Functions for Each Object

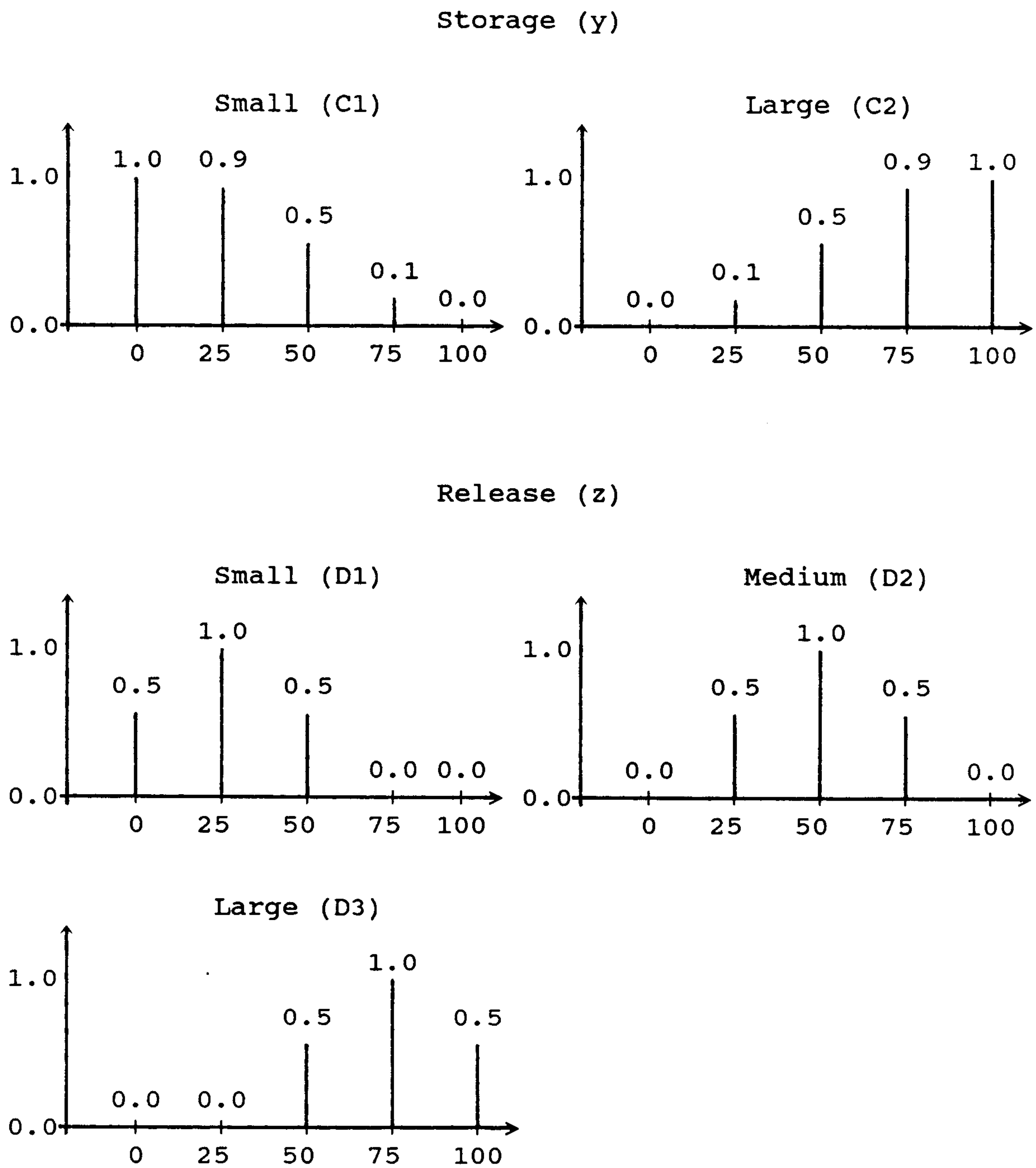


Figure 4-2(2). Membership Functions for Each Object

Rule 1: $R_{A1B1} = \begin{vmatrix} 1.0 & 0.6 & 0.3 & 0.1 & 0.0 \\ 0.8 & 0.6 & 0.3 & 0.1 & 0.0 \\ 0.5 & 0.5 & 0.3 & 0.1 & 0.0 \\ 0.2 & 0.2 & 0.2 & 0.1 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{vmatrix}$

Rule 2: $R_{A2B2} = \begin{vmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.1 & 0.2 & 0.2 & 0.2 \\ 0.0 & 0.1 & 0.3 & 0.5 & 0.5 \\ 0.0 & 0.1 & 0.3 & 0.6 & 0.8 \\ 0.0 & 0.1 & 0.3 & 0.6 & 1.0 \end{vmatrix}$

Rule 3-1 $R_{B1D1} = \begin{vmatrix} 0.5 & 0.5 & 0.5 & 0.2 & 0.0 \\ 1.0 & 0.8 & 0.5 & 0.2 & 0.0 \\ 0.5 & 0.5 & 0.5 & 0.2 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{vmatrix}$

Rule 3-2 $R_{C1D1} = \begin{vmatrix} 0.5 & 0.5 & 0.5 & 0.1 & 0.0 \\ 1.0 & 0.9 & 0.5 & 0.1 & 0.0 \\ 0.5 & 0.5 & 0.5 & 0.1 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{vmatrix}$

Rule 4-1 $R_{B2D2} = \begin{vmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.2 & 0.5 & 0.5 & 0.5 \\ 0.0 & 0.2 & 0.5 & 0.8 & 1.0 \\ 0.0 & 0.2 & 0.5 & 0.5 & 0.5 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{vmatrix}$

Rule 4-2 $R_{C1D2} = \begin{vmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.5 & 0.5 & 0.5 & 0.1 & 0.0 \\ 1.0 & 0.9 & 0.5 & 0.1 & 0.0 \\ 0.5 & 0.5 & 0.5 & 0.1 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{vmatrix}$

Rule 5-1 $R_{B1D2} = \begin{vmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.5 & 0.5 & 0.5 & 0.2 & 0.0 \\ 1.0 & 0.8 & 0.5 & 0.2 & 0.0 \\ 0.5 & 0.5 & 0.5 & 0.2 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{vmatrix}$

Rule 5-2 $R_{C2D2} = \begin{vmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.1 & 0.5 & 0.5 & 0.5 \\ 0.0 & 0.1 & 0.5 & 0.9 & 1.0 \\ 0.0 & 0.1 & 0.5 & 0.5 & 0.5 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{vmatrix}$

Rule 6-1 $R_{B2D3} = \begin{vmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.8 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.2 & 0.5 & 0.5 & 0.5 \\ 0.0 & 0.2 & 0.5 & 0.8 & 1.0 \\ 0.0 & 0.2 & 0.5 & 0.5 & 0.5 \end{vmatrix}$

Rule 6-2 $R_{C2D3} = \begin{vmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.1 & 0.5 & 0.5 & 0.5 \\ 0.0 & 0.1 & 0.5 & 0.9 & 1.0 \\ 0.0 & 0.1 & 0.5 & 0.5 & 0.5 \end{vmatrix}$

Modus ponens to infer the consequence can be written as:

$$M_B' = \{ (A' \circ R_{A1B1}) \cup (A' \circ R_{A2B2}) \mid TV = 1.0 \}$$

$$\begin{aligned} M_D' = & \{ (M_B' \circ R_{B1D1}) \cap (C' \circ R_{C1D1}) \mid TV = 1.0 \} \\ & \cup \{ (M_B' \circ R_{B2D2}) \cap (C' \circ R_{C1D2}) \mid TV = 1.0 \} \\ & \cup \{ (M_B' \circ R_{B1D2}) \cap (C' \circ R_{C2D2}) \mid TV = 0.5 \} \\ & \cup \{ (M_B' \circ R_{B2D3}) \cap (C' \circ R_{C2D3}) \mid TV = 1.0 \} \end{aligned}$$

where o is composition, U is OR (maximum) operator, and ∩ is AND (minimum) operator. Since linear relationships exist between the cartesian product and the truth value, the composition with the truth value can be performed by simple multiplication of truth value by the membership functions and the cartesian products.

As described earlier, the user response can be a membership function. To simplify the comparison with a certainty factor model in this example, however, the user's responses are chosen to be nonfuzzy vectors with only one element equal to unity and the remaining equal to zero. Fuzzy logic based expert systems convert a value to a nonfuzzy vector; therefore, the user specifies one value for each object.

For example, the conclusion that results from the given facts, predicted rainfall is 25 and storage is 25, is calculated as:

$$M_B' = \{\ (A' \circ R_{A1B1}) \cup (A' \circ R_{A2B2})\ |\ TV = 1.0\ \}$$

$$= [\ 0\ \ 1\ \ 0\ \ 0\ \ 0\ ] \circ \begin{vmatrix} 1.0 & 0.6 & 0.3 & 0.1 & 0.0 \\ 0.8 & 0.6 & 0.3 & 0.1 & 0.0 \\ 0.5 & 0.5 & 0.3 & 0.1 & 0.0 \\ 0.2 & 0.2 & 0.2 & 0.1 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{vmatrix}$$

$$\cup\ [\ 0\ \ 1\ \ 0\ \ 0\ \ 0\ ] \circ \begin{vmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.1 & 0.2 & 0.2 & 0.2 \\ 0.0 & 0.1 & 0.3 & 0.5 & 0.5 \\ 0.0 & 0.1 & 0.3 & 0.6 & 0.8 \\ 0.0 & 0.1 & 0.3 & 0.6 & 1.0 \end{vmatrix}$$

$$= [0.6\ 0.6\ 0.5\ 0.2\ 0.0] \cup [0.0\ 0.1\ 0.1\ 0.1\ 0.1]$$

$$= [0.6\ 0.6\ 0.5\ 0.2\ 0.1]$$

$$\begin{aligned} M_D' = &\ \{\ (M_B' \circ R_{B1D1}) \cap (C' \circ R_{C1D1})\ |\ TV = 1.0\ \} \\ &\cup \{\ (M_B' \circ R_{B2D2}) \cap (C' \circ R_{C1D2})\ |\ TV = 1.0\ \} \\ &\cup \{\ (M_B' \circ R_{B1D2}) \cap (C' \circ R_{C2D2})\ |\ TV = 0.5\ \} \\ &\cup \{\ (M_B' \circ R_{B2D3}) \cap (C' \circ R_{C2D3})\ |\ TV = 1.0\ \} \end{aligned}$$

$$= \{ \ [0.6\ 0.6\ 0.5\ 0.2\ 0.1] \circ \begin{vmatrix} 0.5 & 0.5 & 0.5 & 0.2 & 0.0 \\ 1.0 & 0.8 & 0.5 & 0.2 & 0.0 \\ 0.5 & 0.5 & 0.5 & 0.2 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{vmatrix}$$

$$\cap [\ 0\ \ 1\ \ 0\ \ 0\ \ 0\ ] \circ \begin{vmatrix} 0.5 & 0.5 & 0.5 & 0.1 & 0.0 \\ 1.0 & 0.9 & 0.5 & 0.1 & 0.0 \\ 0.5 & 0.5 & 0.5 & 0.1 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{vmatrix} \ \}$$

$$\cup \{ [0.6\ 0.6\ 0.5\ 0.2\ 0.1] \circ \begin{vmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.2 & 0.5 & 0.5 & 0.5 \\ 0.0 & 0.2 & 0.5 & 0.8 & 1.0 \\ 0.0 & 0.2 & 0.5 & 0.5 & 0.5 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{vmatrix}$$

$$\cap [\ 0\ \ 1\ \ 0\ \ 0\ \ 0\ ] \circ \begin{vmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.5 & 0.5 & 0.5 & 0.1 & 0.0 \\ 1.0 & 0.9 & 0.5 & 0.1 & 0.0 \\ 0.5 & 0.5 & 0.5 & 0.1 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{vmatrix} \ \}$$

$$\cup \{ [0.30\ 0.30\ 0.25\ 0.10\ 0.05] \circ \begin{vmatrix} 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.25 & 0.25 & 0.25 & 0.10 & 0.00 \\ 0.50 & 0.40 & 0.25 & 0.10 & 0.00 \\ 0.25 & 0.25 & 0.25 & 0.10 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \end{vmatrix}$$

$$\cap [\ 0\ \ 0.5\ \ 0\ \ 0\ \ 0\ ] \circ \begin{vmatrix} 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.05 & 0.25 & 0.25 & 0.25 \\ 0.00 & 0.05 & 0.25 & 0.45 & 0.50 \\ 0.00 & 0.05 & 0.25 & 0.25 & 0.25 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \end{vmatrix} \ \}$$

$$\cup \{ \ [0.6\ 0.6\ 0.5\ 0.2\ 0.1] \circ \begin{vmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.2 & 0.5 & 0.5 & 0.5 \\ 0.0 & 0.2 & 0.5 & 0.8 & 1.0 \\ 0.0 & 0.2 & 0.5 & 0.5 & 0.5 \end{vmatrix}$$

$$\cap [\ 0\ \ 1\ \ 0\ \ 0\ \ 0\ ] \circ \begin{vmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.1 & 0.5 & 0.5 & 0.5 \\ 0.0 & 0.1 & 0.5 & 0.9 & 1.0 \\ 0.0 & 0.1 & 0.5 & 0.5 & 0.5 \end{vmatrix} \ \}$$

$$= \{[0.50\ 0.60\ 0.50\ 0.00\ 0.00] \cap [0.50\ 0.90\ 0.50\ 0.00\ 0.00]\}$$
$$\cup \{[0.00\ 0.50\ 0.50\ 0.50\ 0.00] \cap [0.00\ 0.50\ 0.90\ 0.50\ 0.00]\}$$
$$\cup \{[0.00\ 0.25\ 0.30\ 0.25\ 0.00] \cap [0.00\ 0.05\ 0.05\ 0.05\ 0.00]\}$$
$$\cup \{[0.00\ 0.00\ 0.50\ 0.50\ 0.50] \cap [0.00\ 0.00\ 0.10\ 0.10\ 0.10]\}$$

$$= [0.50\ 0.60\ 0.50\ 0.00\ 0.00]$$
$$\cup [0.00\ 0.50\ 0.50\ 0.50\ 0.00]$$
$$\cup [0.00\ 0.05\ 0.05\ 0.05\ 0.00]$$
$$\cup [0.00\ 0.00\ 0.10\ 0.10\ 0.10]$$

$$= [0.50\quad 0.60\quad 0.50\quad 0.50\quad 0.10]$$

Figure 4-3 illustrates the membership function of the consequence object, release, using fuzzy reasoning.

Membership Function of the Consequence, Release

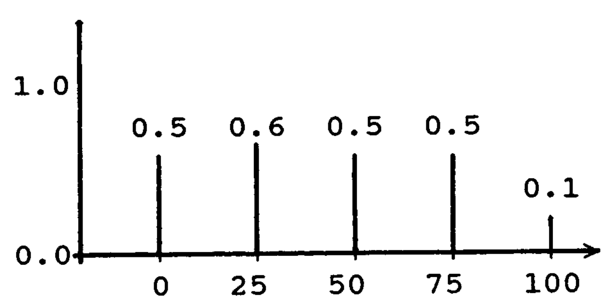


Figure 4-3. Fuzzy Reasoning Result

We can interpret this reasoning result in several ways. First, the simplest implication is that the release is 25 because that value is associated the maximum membership function. Second, the release could be established at 40 based on a calculation of the horizontal centroid of the membership function. The horizontal centroid (HG) can be calculated by:

$$HG = \frac{0*0.5 + 25*0.6 + 50*0.5 + 75*0.5 + 100*0.1}{0.5 + 0.6 + 0.5 + 0.5 + 0.1} = 40$$

These two implications are often used in the applications of dynamic system's controller. Finally, an implication can be drawn based on the calculation of the matching degree as defined in Chapter Tree. In this example, the calculation results of matching degree for each consequences are:

Release is small with matching degree 62 %.

Release is medium with matching degree 56 %.

Release id large with matching degree 32 %.

These fuzzy matrix manipulations appear complicated. However, they are performed by maximum and minimum operators, and are not computationally intensive.

Certainty Factor Model Approach

The certainty factor model is now examined using the same problem. Several approaches can be taken to deal with *quasi-fuzzy* logic in the certainty factor model applications; however, the following simple assumptions are incorporated:

1) A fuzzy set is used to represent each object; rainfall, inflow, storage, and release.

2) The rules and the tree structure presented in Figure 4-1 are maintained.

3) The goal of this system is to determine release.

4) The user has the responsibility to enter appropriate certainty factors into the system.

5) The conclusion of the system is a qualitative expressions, such as "Release is small with CF = 0.8", etc.

6) The system has the capability to deal with multi-attribute objects.

These conditions are similar to those in the example of fuzzy logic; thus, predicted rainfall is 25 and storage is 25. In this example, the user cannot enter these two values directly. Instead, the user must answer the following questions with certainty factors.

Q1: Is predicted rainfall small ?

A1: True with CF = 0.6.

Q2: Is predicted rainfall large ?

A2: True with CF = 0.1.

Q3: Is storage small ?

A3: True with CF = 0.9.

Q4: Is storage large ?

A4: True with CF = 0.1.

Based on these user inputs, the conclusions of future inflow and release are:

Rule 1: Future inflow is small with CF = 0.6.

Rule 2: Future inflow is large with CF = 0.1.

Rule 3: Release is small with CF = 0.6.

$CF = \min(0.9, 0.6) * 1.0.$

Rule 4: Release is medium with CF = 0.1.

$CF = \min(0.9, 0.1) * 1.0.$

Rule 5: Release is medium with CF = 0.05.

$CF = \min(0.1, 0.6) * 0.5$

Rule 6: Release is large with CF = 0.1.

$CF = \min(0.1, 0.1) * 1.0$

If Rule 4 and Rule 5 are connected with OR, the final conclusions are:

Release is small with *CF* = 0.6.

Release is medium with *CF* = 0.1.

Release is large with *CF* = 0.1.

It is obvious from these conclusions that the certainty factor model does not capture the same type of information as the fuzzy logic. The certainty factor model can emulate fuzzy logic only in the most certain conclusion (*CF* of "release is small" is the same as the maximum membership in Figure 4-3). However, the certainty factor model cannot emulate the distribution of possibility. In this example, the certainty factor of "release is medium" is the same as the certainty factor of "release is large". This conclusion is intuitively contradictory.

One approach to emulate the possibility distribution in rule based systems with certainty factors is to develop a series of rules that replicate the membership function. For instance, Rule 3 can be divided into the following five rules.

Rule 3-1:

IF future inflow is small and storage is small

THEN release is 0

WITH *CF* = 0.5.

Rule 3-2:

IF future inflow is small and storage is small

THEN release is 25

WITH *CF* = 1.0.

Rule 3-3:

IF future inflow is small and storage is small

THEN release is 50

WITH *CF* = 0.5.

Rule 3-4:

IF future inflow is small and storage is small

THEN release is 75

WITH *CF* = 0.

Rule 3-5:

IF future inflow is small and storage is small

THEN release is 100

WITH *CF* = 0.

If each rule that has the same consequence is connected to the other rules with OR, the final conclusions are:

Release is 0 with $CF = 0.3$.

Release is 25 with $CF = 0.6$.

Release is 50 with $CF = 0.1$.

Release is 75 with $CF = 0.1$.

Release is 100 with $CF = 0.05$.

This approach also cannot emulate fuzzy logic correctly because of the technique used to propagate uncertainty. In fuzzy logic, uncertainty is propagated by membership functions. By contrast in the uncertainty factor model, it is performed based on one value. Therefore, even though rules are divided into small rules, the certainty factor model cannot capture fuzzy information precisely.

Summary

This chapter discusses the differences between fuzzy logic and the certainty factor model in terms of propagation of uncertainty. The important conclusions are:

1) The concept of fuzzy sets is useful for water resources management expert systems because rules such as, "Storage is Small", are often used in practical applications.

2) The certainty factor model of conventional expert systems can emulate fuzzy propositions partially; however, expertise must be in the user's hand as well as the expert system.

3) Propagation of uncertainty cannot be performed correctly by the certainty factor model, even if rules are divided into small rules.

Table 4-1 summarizes the differences between the certainty factor model and fuzzy logic.

Table 4-1. Certainty Factor Model vs Fuzzy Logic

Subject	Certainty Factor Model	Fuzzy Logic
Propositions	Can be Fuzzy	Fuzzy
Antecedent	Can be Fuzzy	Fuzzy
Consequence	Not Fuzzy But Uncertain	Fuzzy Membership
User Input	True with CF	Value or Membership (can be in system)
Expertise	System and User	System
Vehicle of Uncertainty	One Value	Membership Function
Form of Rules	Wordy	Concise

CHAPTER FIVE

Development of FRESH (<u>F</u>uzzy <u>Re</u>asoning <u>Sh</u>ell)

FRESH, (<u>F</u>uzzy <u>Re</u>asoning <u>Sh</u>ell) provides an interactive knowledge acquisition environment and a fuzzy logic based framework for the management of uncertainty. FRESH integrates a menu-driven system, user input screens, graphics, and a fuzzy logic based inference engine. With FRESH, a user develops fuzzy logic based decision making systems (or dynamic system controllers) that use a knowledge base, and provide solutions to problems that require approximate reasoning.

FRESH is written in Turbo C (Borland International, 1988) for the IBM Personal Computer and its compatibles. The C language was chosen because it provides a structured skeleton without limiting the creativity of the programmer, while C compilers consistently produce extremely fast and efficient executable program. C also provides powerful string manipulation capabilities. Most currently advanced computer software including word processors, expert system shells, and UNIX are written in C.

Since FRESH is an executable system, the programmer can link FRESH with another system that was written in an AI language. This means that it is possible to develop *fuzzy logic* based expert systems that incorporate backward chaining, database management, and fuzzy reasoning capabilities in the future.

The remainder of this chapter provides an introduction to the organization and functions of FRESH. Appendix B describes the installation and hardware requirements of FRESH.

FRESH Functions

The use of FRESH is simple. Since FRESH uses a menu-driven system, the user can begin by using the sample knowledge base provided that is explained in Chapter Five. After using this sample knowledge base, the user can easily create his/her knowledge bases to run under FRESH.

All of the major functions of FRESH are accessed from a main menu that lists five primary options. These options include 1) editing membership functions and fuzzy propositions, 2) graphics of membership functions, 3) graphics of fuzzy conditional propositions, 4) fuzzy inference engine including graphics presentation of fuzzy reasoning results, and 5) print a list of membership functions and fuzzy propositions.

With menu screens, the user selects the desired FRESH function by either a pointer (using the upward and downward cursor arrow keys followed by the <ENTER> key) or the first red colored letter of the entries, called *hot keys*.

FRESH has on-line HELP available from every menu screen. HELP is restricted to provide information pertinent to the specific context from which HELP was called. When HELP is activated by pressing <?> key, a pop-up screen of topics is presented that explains the feature of FRESH. To exit HELP, pressing any key will return the user to the previously active screen. Figure 5-1 shows the main menu of FRESH.

FRESH Editor

Choose "Edit Membership Functions or Propositions" from the main menu with either the cursor allow followed by <ENTER> key or the hot key. FRESH shows the next menu and asks the user whether to "Edit New File" or "Edit Existing File" (Figure 5-2). The FRESH editor allows the user to create new membership functions and fuzzy conditional propositions (the equivalent to a knowledge base of conventional expert systems). Also, the user can modify, add, and delete existing membership functions and fuzzy propositions using this editor.

```
Options Menu

E Edit Membership Functions and Conditional Propositions
M View User Defined Membership Functions
F View User Defined Fuzzy Conditional Propositions
R Run Fuzzy Reasoning Based on User Input
P Print a List
Q Quit Program
```

```
Press desired hot key or use arrow keys and press Enter; ? for help
```

Figure 5-1. Options Menu

```
File Menu

E Edit Existing File
N Edit New File
R Return to Option Menu
Q Quit Program
```

```
Press desired hot key or use arrow keys and press Enter; ? for help
```

Figure 5-2. File Menu

If the user chooses "Edit Existing File", FRESH automatically accesses 'C:\FRESH' directory and shows data files that are currently available on screen (Figure 5-3). In this screen, three hot keys (N, P, and C) are used to choose the desired file. After obtaining the input data file name using the combination of hot keys, the FRESH editor provides the next edit menu that allows the user to edit membership functions or fuzzy conditional propositions (Figure 5-4).

The user can define only one consequence object, but up to nineteen antecedent objects. Each object can have up to ten different membership functions. For example, the user can define fuzzy quantifiers, such as *very small, small, medium, large,* and *very large*, for the object, reservoir storage.

If the selection is "Modify Membership Function of Existing Object", FRESH requires the user to select the object name using three hot keys (N, P, and C) that are shown in Figure 5-5. After obtaining the object name, FRESH will ask the user to enter the number of membership functions (Figure 5-6) and the minimum and the maximum values of the object (Figure 5-7). Then, the FRESH editor requests the user to enter the name and the shape of membership functions (Figure 5-8).

```
Existing File Name Selection Screen

Choose desired filename to store membership functions and
propositions using the following three HOT characters.

 N : Next file,      P : Previous file,      C : Choose file

The files shown in the window below are available at present.
NOTE: Only the first 15 files in the 'C:FRESH' directory can be seen.

NWWA_JAN.DAT
```

```
NWWA_JAN.DAT   NWWB_FEB.DAT   NWWC_MAR.DAT   NWWD_APR.DAT
NWWE_MAY.DAT   NWWF_JUN.DAT   NWWG_JUL.DAT   NWWH_AUG.DAT
NWWI_SEP.DAT   NWWJ_OCT.DAT   NWWK_NOV.DAT   NWWL_DEC.DAT
```

Figure 5-3. Existing File Mane Selection Screen

```
Edit Menu

1 Modify Membership Function of Existing Object
2 Add New Object
3 Delete Existing Object
4 Edit Fuzzy Conditional Propositions
5 Change Fuzzy Reasoning Accuracy
6 Draw Existing Membership Functions
7 Draw Existing Conditional Propositions
R Return to Option Menu
Q Quit Program
```

```
Press desired hot key or use arrow keys and press Enter; ? for help
```

Figure 5-4. Edit Menu

```
                    Object Name Selection Screen

 Choose desired object name to modify membership functions using the
 following three HOT keys.
  N : Next object,    P : Previous object,    C : Choose object
 The following objects shown in the window are avairable at present.

                         Operating_policy
```

```
 Operating_policy      Storage_in_Jan.(Ml)   Sim_Result(%_of_Cap)
 Lake_Dist(%fall/wk)
```

Figure 5-5. Object Name Selection Screen

```
      Number of Membership Functions of Operating_policy

 Please enter the number of membership functions as an INTEGER value.

 Maximum number of membership functions is 10.

                              4
```

Figure 5-6. Number of Membership Functions Entering Screen

```
              Input Screen for Operating_policy

Please enter the minimum and the maximum value of this object.

         The minimum value :  0.00000

         The maximum value :  1.00000
```

Figure 5-7. Minimum and Maximum Values of Objects Entering Screen

```
              Input Screen for Operating_policy

Please enter the name of this membership function (maximum length
of characters is 20) and four x-axis points.
   The minimum value : 0.00000   The maximum value : 1.00000

The name of this membership function : Normal_Operation
          The x-axis of bottom left  : 0.00000
          The x-axis of top left     : .200000
          The x-axis of top right    : .200000
          The x-axis of bottom right : .400000
```

Figure 5-8. Name and X-Axis Points of Membership Functions Entering Screen

Membership functions must be piecewise-linear in FRESH. However, they can be non-symmetric triangle or trapezoidal. Bottom left, top left, top right, and bottom right points of membership functions are specified by the user. Ordinates of bottom and top of membership functions are always zero and unity, respectively. Thus, the user's input is only these four x-axis. Figure 5-9 illustrates the location of these four points. If the x-axis of top left is equal to the x-axis of top right, the membership function is triangle.

When the user selects "Add New Object" in Figure 5-4, FRESH requires the user to enter the new object name. The user must type it in response to the questions by FRESH. FRESH then evaluates the user's responses to avoid errors in the knowledge base. In this case, the user cannot choose the name of existing objects. The result for this is "You cannot redeclare the name of existing objects. Please enter new object name !". After obtaining the new object name, FRESH will ask the same questions as the previous example.

If the user chooses "Delete Existing Object" in Figure 5-4, FRESH requires the user to select the desired object name using three hot keys. After obtaining the desired object name, FRESH will present the influence of this action and confirm it (Figure 5-10).

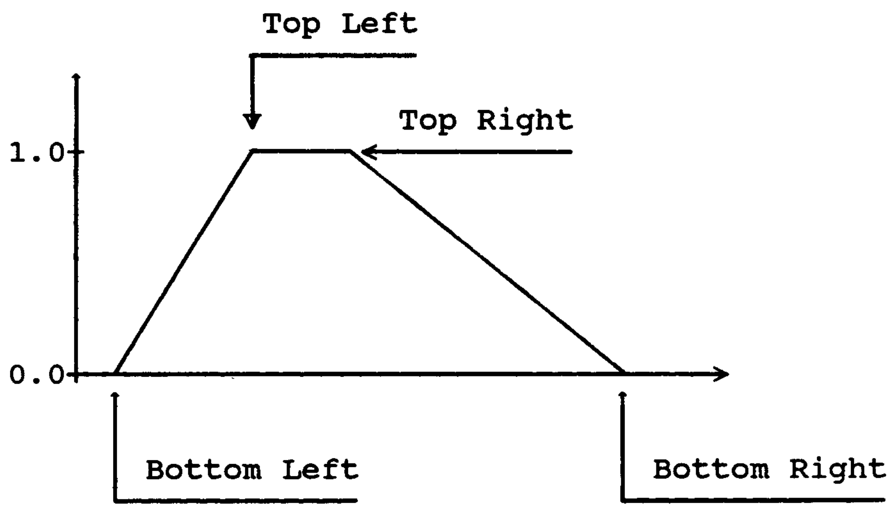


Figure 5-9. Location of Four X-Axis Points of Membership Functions

```
You will lose 14  propositions that contain this object.
                  Are you sure ? (Y/N)
```

```
                    Object Name Selection Screen

Choose desired object name to modify membership functions using the
following three HOT keys.
 N : Next object,     P : Previous object,     C : Choose object
The following objects shown in the window are avairable at present.

                        Storage_in_Jan.(Ml)
```

```
Operating_policy      Storage_in_Jan.(Ml)    Sim_Result(%_of_Cap)
Lake_Dist(%fall/wk)
```

Figure 5-10. Confirmation Screen of "Delete Existing Object"

The user can define up to one hundred fuzzy conditional propositions. Each fuzzy proposition can have up to three antecedent objects. Therefore, one hundred fuzzy conditional propositions, or rules, can be connected by OR operators, and three antecedent objects can be connected by AND operators in each fuzzy conditional proposition.

When the user's selection is "Edit Fuzzy Conditional Propositions" in Figure 5-4, FRESH shows the editing screen (Figure 5-11). In this screen, the user uses six hot keys (N, P, M, A, D, and R) to choose the desired editing functions. Instructions for these hot keys are available from FRESH. The user can edit propositions with the hot keys. Figures 5-12 and 5-13 show this environment. The user can select appropriate object names and membership function names using three hot keys (N, P, and C). The bottom window shows available object names and membership function names.

FRESH uses truth values (TV), from zero to unity, for each fuzzy proposition. The truth values are equivalent to certainty factors (CF) of conventional expert systems. For example, the user can define the following two rules that have different TV:

IF *Current storage* is *Small* AND *Demand* is *Large*
THEN *Drought condition* is *Severe*
WITH TV = 1.0.

OR

IF *Predicted storage* is *Small*
THEN *Drought condition* is *Severe*
WITH TV = 0.8.

These two rules imply that the consequences of rules are the same, *Drought condition* is *Severe*, but the confidence or certainty of rules is different. This capability exists because users commonly have more confidence in some rules than others.

If the user selects "Change Fuzzy Reasoning Accuracy" in Figure 5-4, FRESH will request for input values of reasoning accuracy (Figure 5-14). This is necessary because FRESH calculates matching degree using numerical integration. FRESH asks the user for the number of subintervals between minimum and maximum values of consequence object.

```
                Fuzzy Conditional Proposition Editing Screen

          Choose desired option using the following HOT keys.
 N: Next,   P: Previous,  M: Modify,  A: Add,  D: Delete,  R: Return

              Proposition Name : Rule_Normal_2
   IF Storage_in_Jan.(Ml)  IS Zone_B
  AND Sim_Result(%_of_Cap) IS More_than_15%
  AND                      IS
 THEN Operating_policy     IS Normal_Operation     WITH TV = 1.00.
```

Figure 5-11. Fuzzy Conditional Propositions Editing Screen

```
                Fuzzy Conditional Proposition Editing Screen

          Choose desired option using the following HOT keys.
             N: Next,      P: Previous     C: Choose

              Proposition Name : Rule_Normal_2
   IF Storage_in_Jan.(Ml)  IS Zone_B
  AND Sim_Result(%_of_Cap) IS More_than_15%
  AND                      IS
 THEN Operating_policy     IS Normal_Operation     WITH TV = 1.00.
```

```
 Operating_policy     Storage_in_Jan.(Ml)    Sim_Result(%_of_Cap)
 Lake_Dist(%fall/wk)
```

Figure 5-12. Propositions Editing Environment (1)

```
              Fuzzy Conditional Proposition Editing Screen

        Choose desired option using the following HOT keys.
           N: Next,       P: Previous    C: Choose

            Proposition Name : Rule_Normal_2
  IF  Storage_in_Jan.(Ml)   IS Zone_B
 AND  Sim_Result(%_of_Cap)  IS More_than_15%
 AND                        IS
THEN  Operating_policy      IS Normal_Operation     WITH TV = 1.00.
```

```
Zone_C                  Zone_B                  Zone_A
```

Figure 5-13. Propositions Editing Environment (2)

```
                         Accuracy Input Screen

Enter desired accuracy of Fuzzy Reasoning as an INTEGER value.

The accuracy is the number of subintervals for numerical integration.

The value should be at least 10 and less than 100.

NOTE: High accuracy leads to long execution time.  20 is recommended.

                                 20
```

Figure 5-14. Accuracy Input Screen

FRESH Graphics

When the user selects "View User Defined Membership Functions" or "View User Defined Fuzzy Conditional Propositions" from the main menu FRESH automatically accesses 'C:\FRESH' directory and provides the data files that are currently available on screen (Figure 5-3). After a data file name is selected, the FRESH graphics will plot this information.

The user can also access the FRESH graphics from the edit menu (Figure 5-4). The user can, therefore, modify membership functions or fuzzy conditional propositions using an interactive environment.

FRESH automatically determines whether the computer has a EGA or VGA color monitor and presents the appropriate graphics. Since the C language allows good video graphics image on color display, a VGA or high resolution EGA color monitor is recommended.

FRESH Inference Engine

When the user selects the "Run Fuzzy Reasoning Based on User Input" from the main, FRESH automatically accesses 'C:\FRESH' directory and presents the data files that are currently available on screen (Figure 5-3). After selection of the appropriate file name, FRESH requests the user to enter values of antecedent objects (Figure 5-15). The user can, then, choose "Show Reasoning Results for Each Conditional Propositions", "Show Matching Result and Summary of Fuzzy Reasoning", or both from menu screen (Figure 5-16). The FRESH graphics has the capability to present these results during a fuzzy reasoning session. The FRESH inference engine, then, performs fuzzy reasoning to solve the specified problem.

With the reasoning results for each conditional fuzzy proposition, the user reviews the rules of the knowledge base, the input provided to FRESH, and the reasoning results. This facility provides transparency to the knowledge base; that is, the user can review the rule base, logic, and truth values. During the development of a knowledge base, this capability is a powerful debugging tool that enables the user to test the results of different input.

The FRESH inference engine computes the matching degrees of pre-defined membership functions of consequence object and graphs the results. FRESH, then, provides the number of active rules under given conditions, and suggests

```
                    Input Screen for Storage_in_Jan.(Ml)

Please enter the antecedent object value to use in Fuzzy Reasoning.

   The minimum value of this object :       0.00000
   The maximum value of this object :       4355.00

                                  4355.00
```

Figure 5-15. Antecedent Object Value Input Screen

```
                 Results of Fuzzy Reasoning Menu

B Show Both Results Listed Below
P Show Reasoning Results for Each Conditional Proposition
S Show Matching Result and Summary of Fuzzy Reasoning
R Return to Option Menu
Q Quit Program
```

```
Press desired hot key or use arrow keys and press Enter; ? for help
```

Figure 5-16. Results of Fuzzy Reasoning Menu the best three matched consequence in the summary of fuzzy reasoning.

Summary

FRESH is an interactive knowledge acquisition environment and a fuzzy logic based shell. FRESH provides a menu-driven system to allow a user to create his/her knowledge bases easily. Syntax error evaluation is also provided. Figure 5-17 summarizes the structure of FRESH. FRESH is written in the C language and designed to run on IBM Personal Computer and its compatibles.

The following chapter illustrates the application of FRESH to a North West Water Authority reservoir management system.

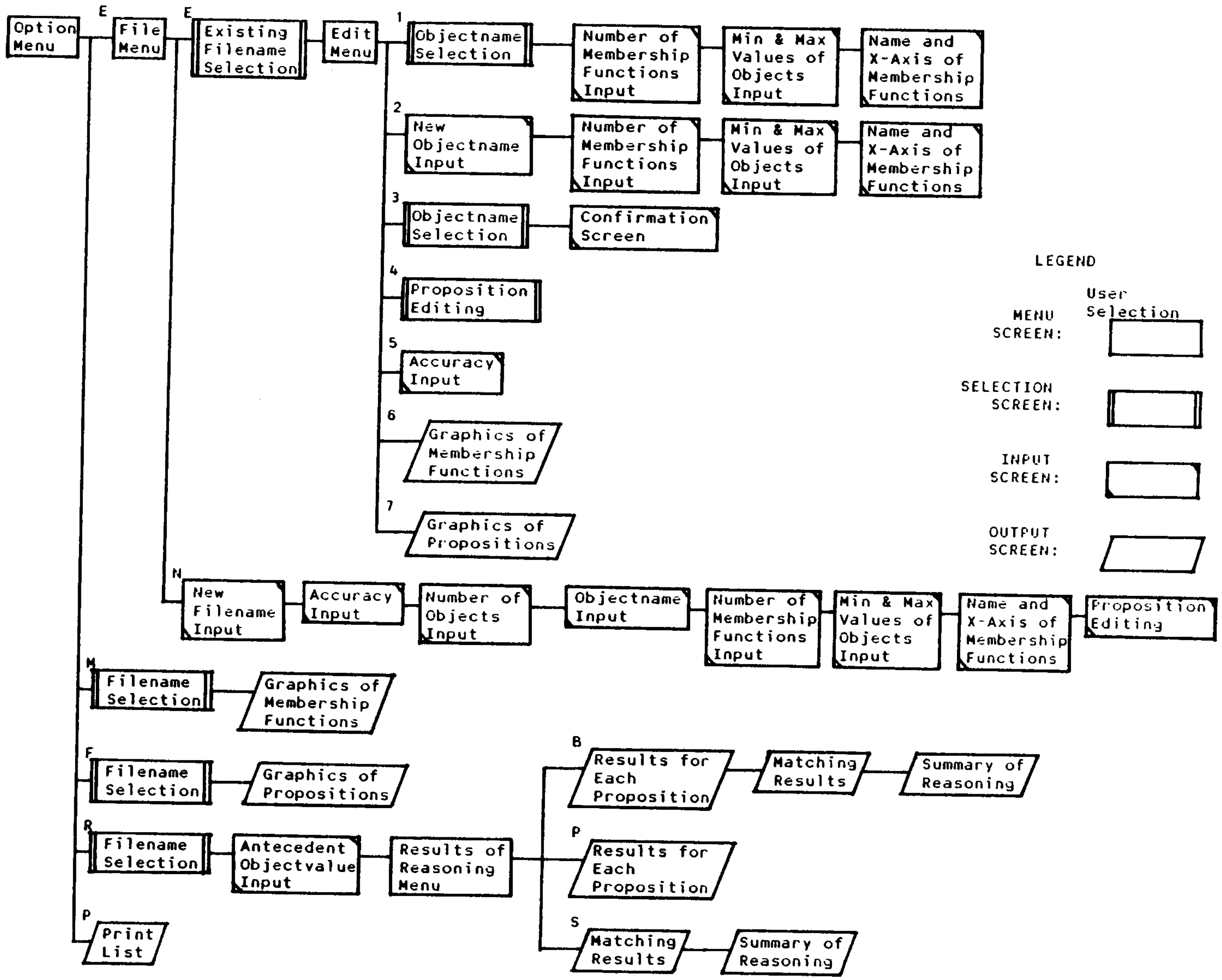


Figure 5-17. Structure of FRESH

CHAPTER SIX

Application of FRESH to North West Water Authority Reservoir Operation

This chapter describes the application of FRESH to the development of reservoir operating policies. In this example, operating policies are developed for the North West Water Authority (NWWA) of Warrington, England. The operating policy is implemented as an interpreter of a set of rules expressed as fuzzy conditional propositions. Since rules used in practice contain linguistic vagueness, fuzzy propositional logic is used to synthesize the linguistic operational rules of an expert. This chapter begins with a brief introduction to the NWWA and their development of operating policies followed by the application. Much of this introductory material was abstracted from Slaughterbeck (1989).

North West Water Authority

NWWA is one of the largest water supply authorities in Great Britain. The NWWA was established by the Water Act 1973 that required aggregation of 231 small public water supply systems in northwest portion of England.

The NWWA supplies water to 7,000,000 people stretching from the Scottish border in the north to Chesire in the south. The service area, approximately 14,500 km^2, is divided into 7 districts that have responsibility for water supply, sewage, and recreation, and receives about 1,200 mm of precipitation per year. The total supply is approximately 2,500 megalitres per day (Ml/d). The NWWA operates about 120 reservoirs as well as pumped transfer systems of lake and river water.

The most severe drought of record occurred in England in 1976. This drought led the Drought Act of 1976 that allowed the NWWA to prohibit or limit water usage, such as recreation grounds, golf courses, lawns, swimming pools, washing cars, and cleaning buildings (Parker, 1980).

Water has been inexpensive and abundant in England and this has resulted in a high per capita use (Parker, 1980). Until the 1976 drought, most people had not been aware that water is finite resource and that drought could negatively impact their lifestyles.

Project Area and Operating Policies

The case study in this thesis deals with three reservoirs; Clowbridge, Cloughbottom, and Haslingden Grain, located in the Rossendale District. These three reservoirs provide water to Manchester and the surrounding area. They are operated as one system by the NWWA; therefore, this system can be conceptualized from the multi-site reservoir system to a single reservoir system. Active storages of these reservoirs are 1229 megalitres (Ml) at Clowbridge, 785 Ml at Cloughbottom, and 1876 Ml at Haslingden Grane, respectively. They supply approximately 28 Ml/d under normal condition.

Like other water authorities in England, the NWWA uses control curves as their primary approach to operate reservoir systems. The NWWA defines a reservoir control curve as the set of values of storage required at the beginning of each month to meet a specified demand during a design drought without failure (Walsh, 1971). These control curves are determined without directly considering economic efficiency but by restricting the probability of reservoir storage being exhausted.

The derivation of control curve ordinates can be divided into two steps. First, the inflow that might occur during the design drought are calculated from a probabilistic analysis. Second, these probabilistic inflows are converted to values of storage required by simple continuity equations balancing inflow and release.

The minimum historic and probabilistic (1, 2, 5, 10, and 20 %) inflows for periods of 1 through 36 consecutive months starting in each month of the calendar year are calculated based on Stocks reservoir (a pilot reservoir of the NWWA) inflow data. These 432 (12 calendar month * 36 different consecutive months) sets of data are examined and fit to a 2-parameter log-normal distribution. Based on the continuity equation, the storage requirement at the beginning of each calendar month is calculated (Walsh, 1971).

The reservoir operational rules at the NWWA are simple. Two control curves are used to make decisions. The upper control curve is calculated based on minimum historic records and safe yield for the reservoir. They assume that the safe yield is a constant value throughout the year. The maximum ordinate of this curve is equal to active storage capacity, called *top water level* in the NWWA. The lower curve is calculated based on minimum historic events and 85 % of the safe yield value. Figure 6-1 and Table 6-1 show the control curves at three sites and conceptualized reservoir.

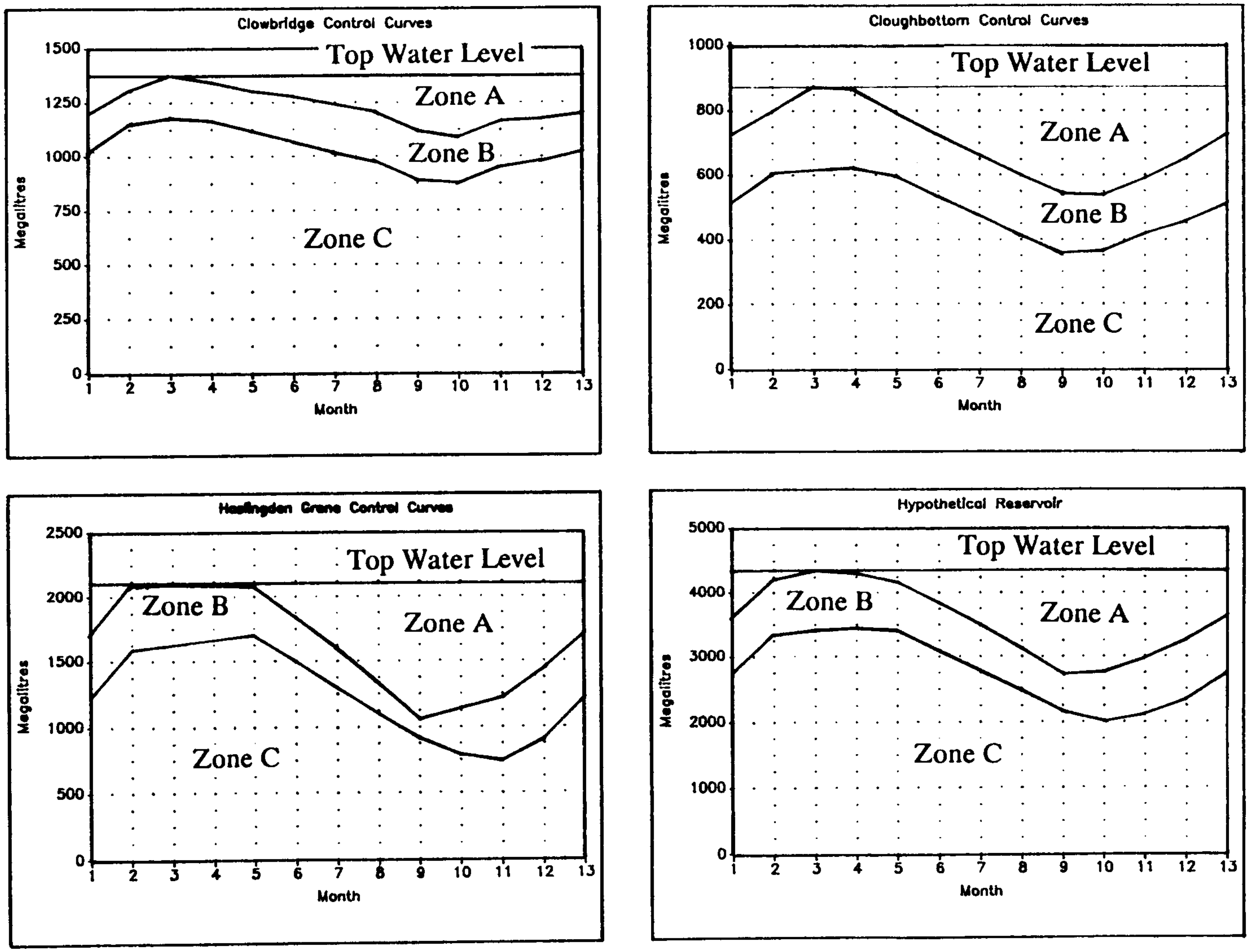


Figure 6-1. Control Curves for Three Sites and Conceptualized Reservoir

Table 6-1. Ordinates of Control Curves

for Three Sites and Conceptualized Reservoir

Month	Clowbridge		Cloughbottom		Haslingden Grain		Hypothetical Reservoir	
	upper	lower	upper	lower	upper	lower	upper	lower
1	1200	1024	727	515	1718	1224	3645	2763
2	1309	1152	799	606	2107	1589	4215	3347
3	1376	1182	872	615	2087	1625	4335	3422
4	1341	1165	864	622	2080	1661	4285	3448
5	1300	1117	792	593	2073	1697	4165	3407
6	1275	1065	724	533	1836	1494	3835	3092
7	1237	1015	660	473	1599	1291	3496	2779
8	1201	970	596	413	1328	1094	3125	2477
9	1118	887	541	358	1058	914	2717	2159
10	1089	875	536	368	1140	785	2765	2001
11	1166	953	588	420	1222	743	2976	2116
12	1175	981	655	458	1445	913	3275	2352
Top Water Level	1376		872		2107		4355	

Three simple reservoir conditions are derived from these control curves. If the storage level remains above the upper control curve, the reservoir is *healthy* (Zone A). If the storage is between the two control curves, the reservoir is *satisfactory* (Zone B). If the storage level is below the lower control curve, the reservoir is *concern* (Zone C).

When reservoir stays in Zone A, supplemental releases can be made. This implies that abstraction is higher than the safe yield. Further analysis is unnecessary in this position.

When reservoir stays in Zone B, only the safe yield can be released. Further analysis is unnecessary in this position. Instead, bringing water from other sources or other districts (Lake District in this case study) might be considered (called *rezoning*).

If the storage falls below Zone C, further analysis is required to determine if drought actions are necessary. Abstraction already equals to the safe yield. If simulation results, based on the safe yield and minimum historic inflows, indicate failure, the abstraction is reduced to the critical yield calculated by the simulation program. In this case, bringing water from other district is required, and/or the other drought actions may be necessary, such as advertisement of urgent restraint in water use, compensation water reductions, and hosepipe bans.

However, these three simple rules do not govern all of practical situations. The other important and practical operating policies are (Slaughterbeck, 1989):

1) The operating policy based on control curves is not strict. In *some cases*, overdraft will continue against the operating policy because rezoning, for example, will cause water quality problems and field work.

2) If the storage of Lake District is less than 30 % of net capacity and falling more than 4 % per week, bringing in water from this source *may not be possible*.

3) If the simulation result shows that the future minimum storage will be less than 15 % of active storage, the manager *starts to be concerned* this situation is "failure".

These practical rules contain linguistic vagueness, such as *some cases*, *may not be possible*, and *starts to be concerned*, and implies the necessity of approximate reasoning.

Approximate Reasoning Using Linguistic Synthesis by FRESH

The linguistic synthesis approach is applied to make decisions concerning drought management policy for the conceptualized reservoir in the NWWA. The point of view adopted here is that objects are associated with universes of discourse that are nonfuzzy sets. These objects take on specific linguistic values that are expressed as fuzzy subsets of the corresponding universes. For instance, an universe of discourse is *storage volume*, an object is *current storage*, and a linguistic value is *Zone A*.

The rules obtained from the expert in this case study consist of one consequence object and three antecedent objects. For instance, if a rule can be stated as "IF *A* AND *B* AND *C* THEN *D*", the antecedent objects are *A*, *B*, and *C* and the consequence object is *D*. The consequence object is Operating Policy that has four linguistic values: 1) Normal Operation, 2) Draw from Lake District, 3) Consider Bans, and 4) Need Bans. The three antecedent objects are 1) Current Storage, 2) Simulated Future Minimum Storage, and 3) Condition of Lake District (storage falling rate per week). They also have linguistic values like the consequence object. As previously mentioned, the transitions of these linguistic values are gradual rather than abrupt. The operating policy consists of a set of rules in terms of specific linguistic values of these objects.

A list of fuzzy subsets (including their linguistic values) and rules for June by FRESH is shown in Appendix C. FRESH does not have a capability to deal with dimensional arrays; therefore, a user must make twelve different data files associated with each calendar month in this application. These results have been obtained after several modifications of the fuzzy subsets and operating rules. The trial and error method was used in order to obtain an appropriate set of rules. These rules are in the form of fuzzy conditional propositions and can be expressed by fuzzy relations that are discussed in Chapter Three.

Figure 6-2 illustrates the shape of membership functions of each object by FRESH. For instance, the consequence object, Operating Policy, has four membership functions (linguistic values). They overlap each other and the transitions are gradual. For this object, the conceptual meaning of *x-axis* can be interpreted as potential of drought. Small *x* value means drought potential is minor, and large *x* means drought potential is severe.

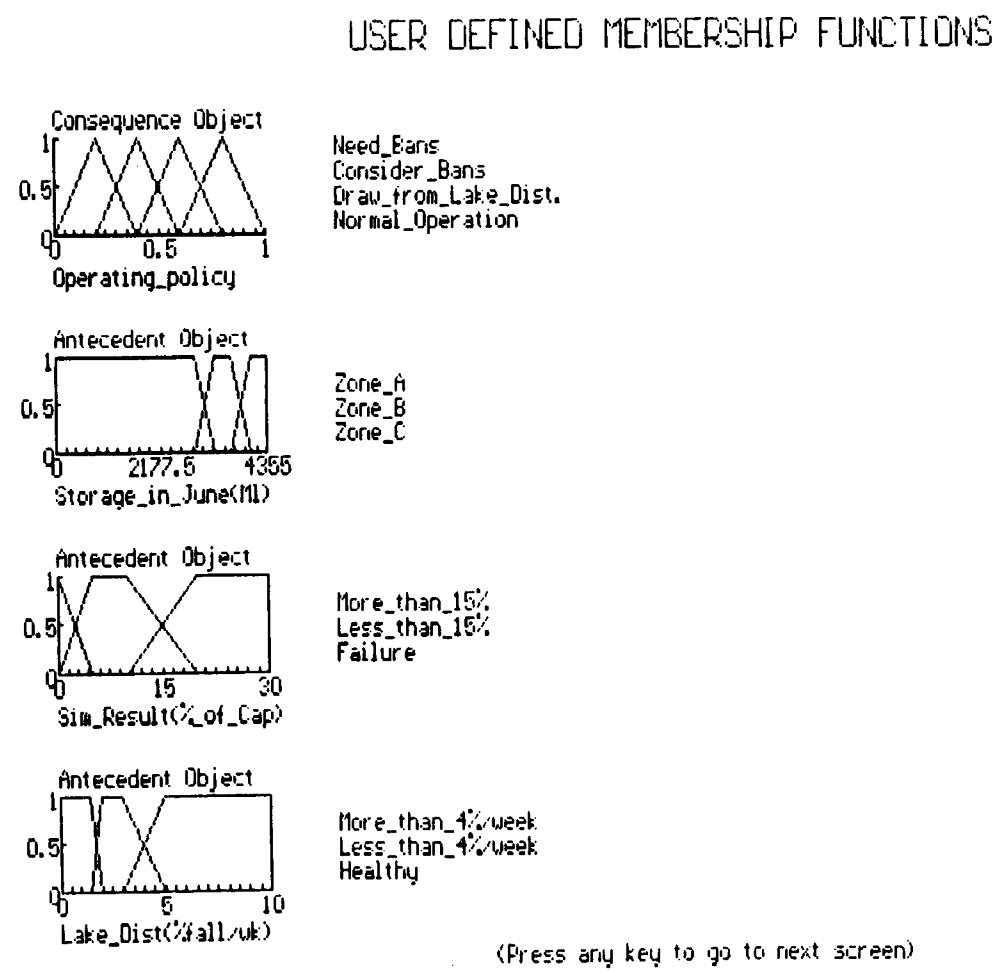


Figure 6-2. The Shape of Membership Functions of Each Object by FRESH

The other example is the first antecedent object, Current Storage. This object has three membership functions: 1) Zone A, 2) Zone B, and 3) Zone C. The crisp operating policy says that if Current Storage is 3093 then it is in Zone B, and if Current Storage is 3091 then it is in Zone C. As previously mentioned, however, this crisp rule is not accurate because if Current Storage is around 3092 (the lower control curve) it is sometimes Zone B, and it is sometimes Zone C. The implementation of this vague condition by fuzzy set theory is that:

1) If Current Storage is 3092 then the possibility of "in Zone C" is 0.5, and the possibility of "in Zone B" is 0.5.

2) If Current Storage is 2992 then the possibility of "in Zone B" is 0.25, and the possibility of "in Zone C" is 0.75.

3) If Current Storage is 2892 then the possibility of "in Zone B" is zero, and the possibility of "in Zone C" is unity.

Figure 6-3 illustrates the first four linguistic rules interpreted as fuzzy conditional propositions by FRESH. Each row represents one rule. These rules have up to three antecedent object connected with AND operators. All rules are then connected with OR operators. The comparison with Appendix C describes the output of FRESH.

Table 6-2 illustrates the case of experimental design undertaken to investigate the effect of the input values of antecedent objects and their reasoning results. This experimental design contains twenty different combinations of antecedent objects. These are:

1) Current Storage

(1) just on the lower control curve and (2) in Zone C.

2) Simulated Future Minimum Storage

(1) 7.5 % and (2) 2.5 %.

3) Condition of Lake District (storage falling rate per week)

(1) 0 %, (2) 1.75 %, (3) 2.5 %, (4) 4 %, and (5) 10 %.

Figure 6-4 illustrates reasoning results for each fuzzy conditional proposition by FRESH under the case (1)-(2)-(4) (Current Storage is just on the lower control curve, Simulated Future Minimum Storage is 2.5 % of net capacity, and Condition of Lake District is 4 % storage falling rate per week) in Table 6-2.

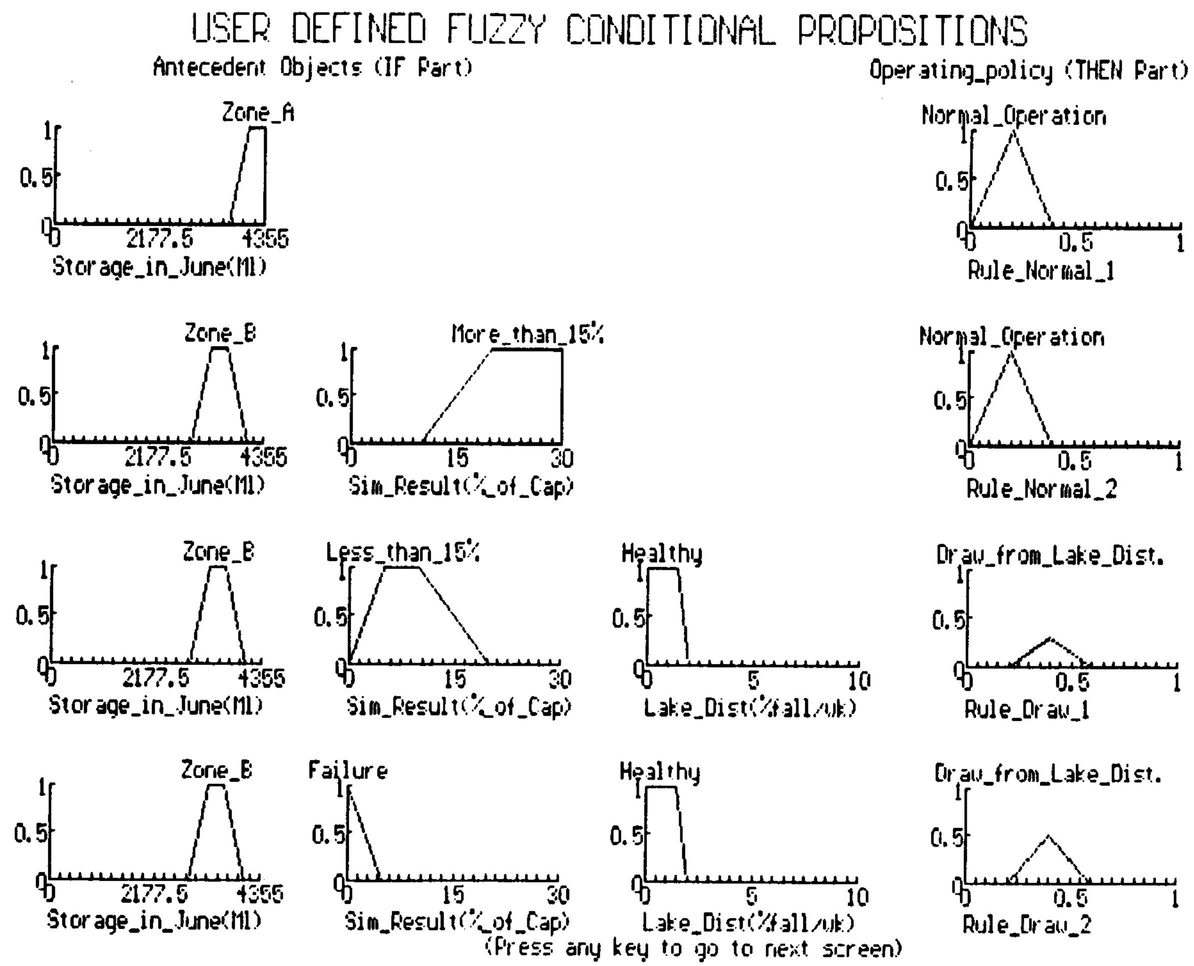


Figure 6-3. The First Four Fuzzy Conditional Propositions by FRESH

Table 6-2 Experimental Design Cases

and Matching Degrees

Current Storage	Simulated Future Minimum Storage	Condition of Lake District	Normal Operation	Draw from Lake District	Consider Bans	Need Bans
Lower Control Curve	7.5 %	0 %	10 %	40 %	10 %	0 %
		1.75 %	8 %	34 %	34 %	8 %
		2.5 %	0 %	10 %	40 %	10 %
		4 %	0 %	8 %	34 %	34 %
		10 %	0 %	0 %	10 %	40 %
	2.5 %	0 %	11 %	50 %	11 %	0 %
		1.75 %	9 %	39 %	39 %	9 %
		2.5 %	0 %	11 %	50 %	11 %
		4 %	0 %	9 %	39 %	39 %
		10 %	0 %	0 %	11 %	50 %
Zone C	7.5 %	0 %	13 %	80 %	13 %	0 %
		1.75 %	8 %	34 %	34 %	8 %
		2.5 %	0 %	13 %	80 %	13 %
		4 %	0 %	8 %	34 %	34 %
		10 %	0 %	0 %	13 %	80 %
	2.5 %	0 %	11 %	50 %	11 %	0 %
		1.75 %	9 %	39 %	39 %	9 %
		2.5 %	0 %	11 %	50 %	11 %
		4 %	0 %	9 %	39 %	39 %
		10 %	0 %	0 %	11 %	50 %

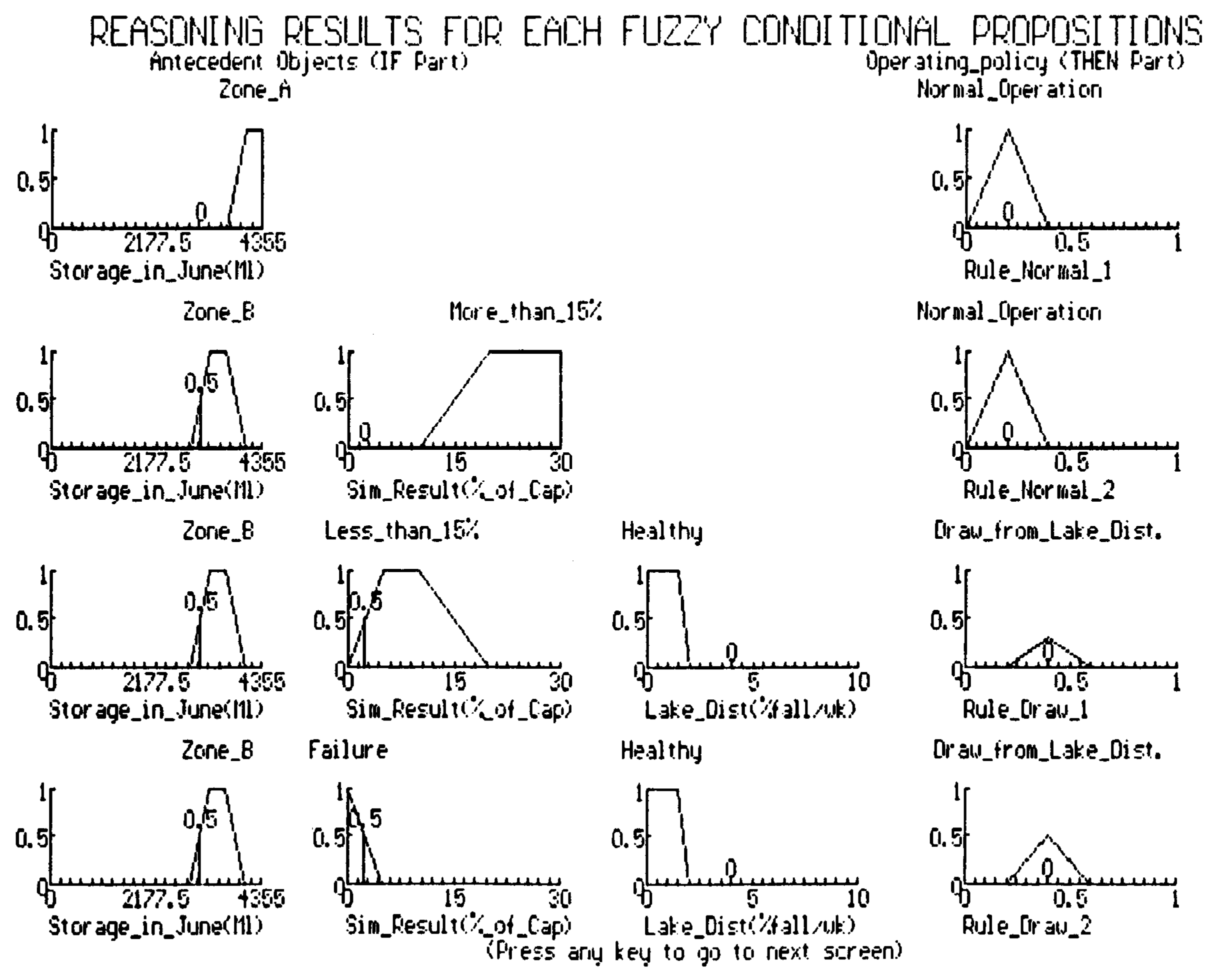


Figure 6-4 (1). Reasoning Results for Each Fuzzy Conditional Propositions by FRESH

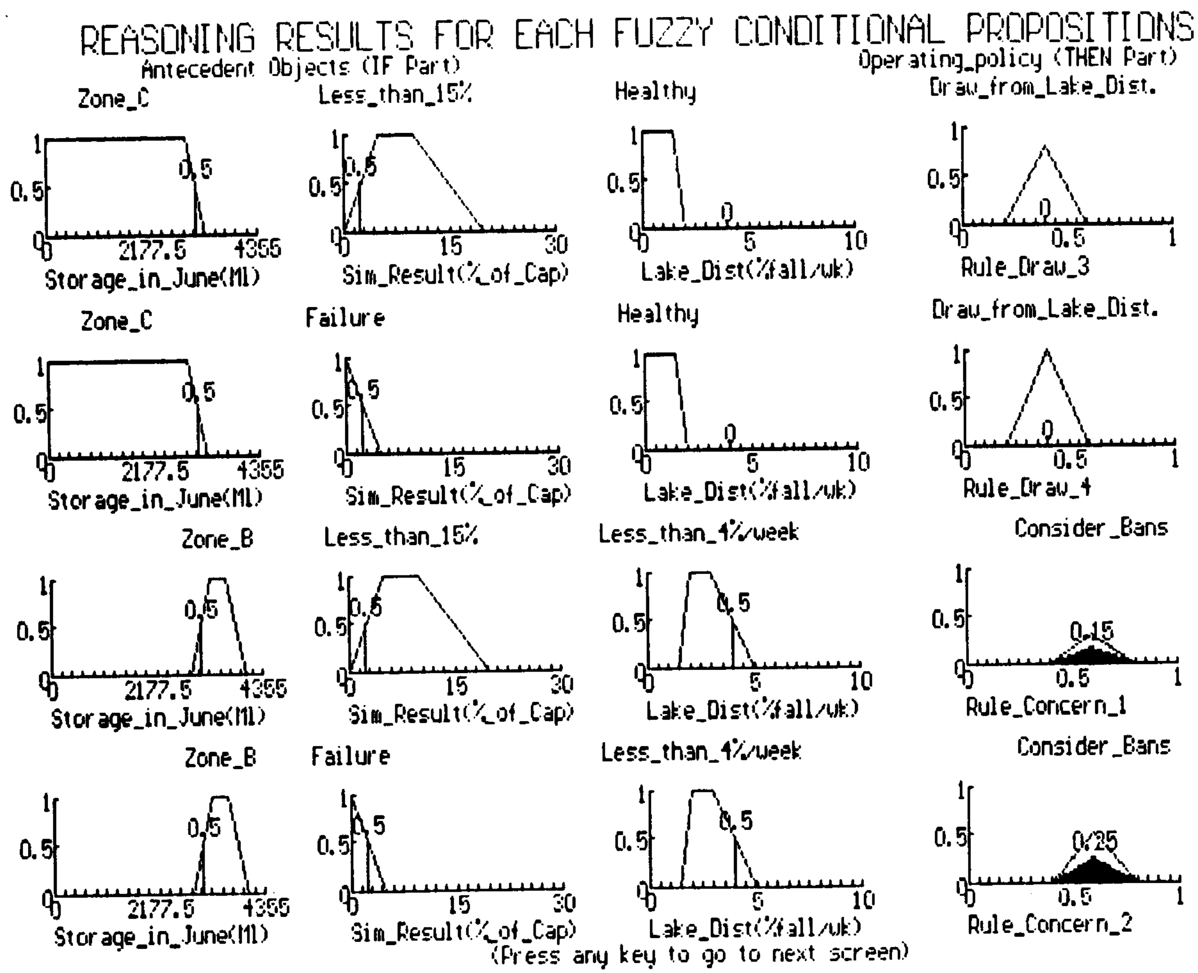


Figure 6-4 (2). Reasoning Results for Each Fuzzy Conditional Propositions by FRESH

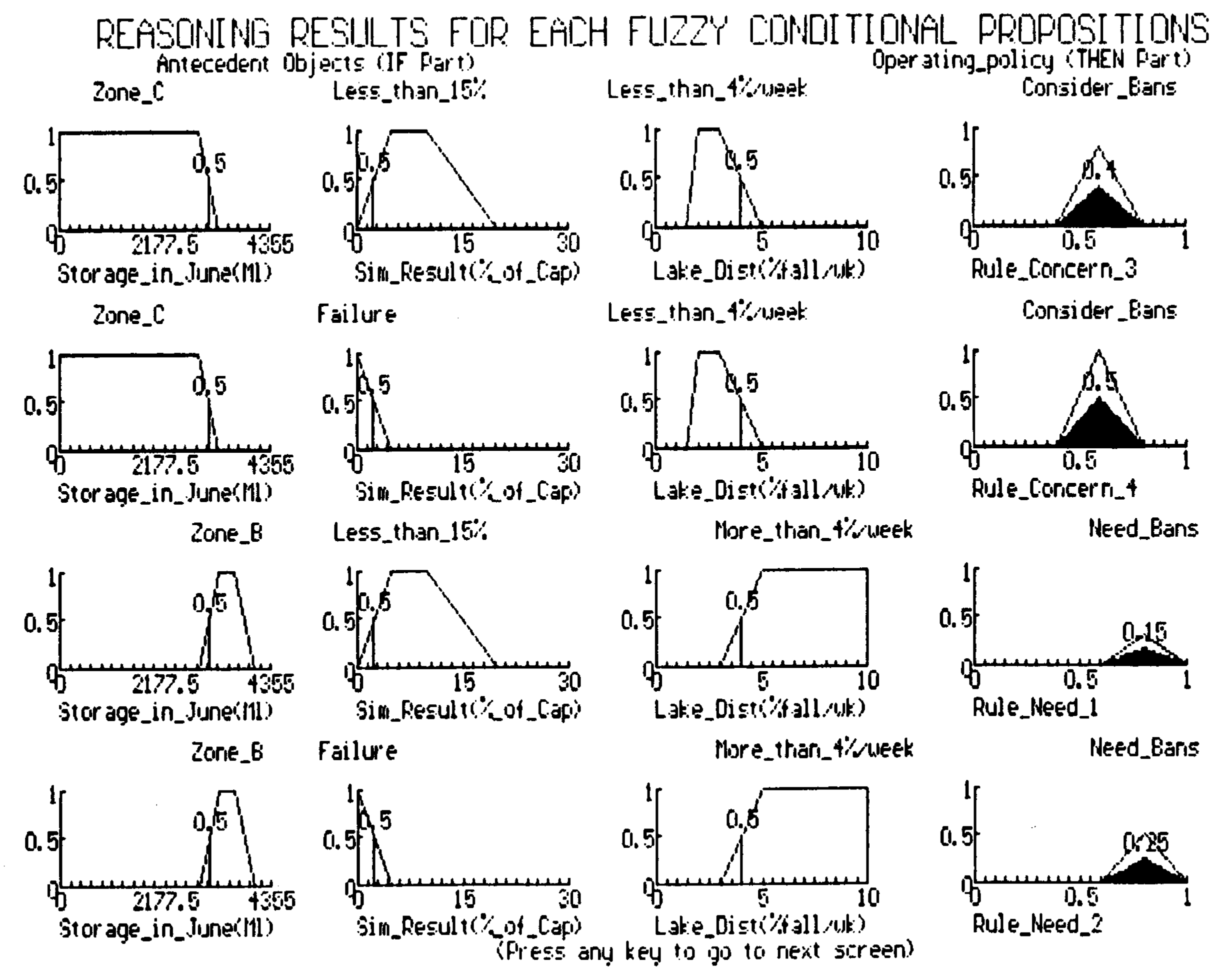


Figure 6-4 (3). Reasoning Results for Each Fuzzy Conditional Propositions by FRESH

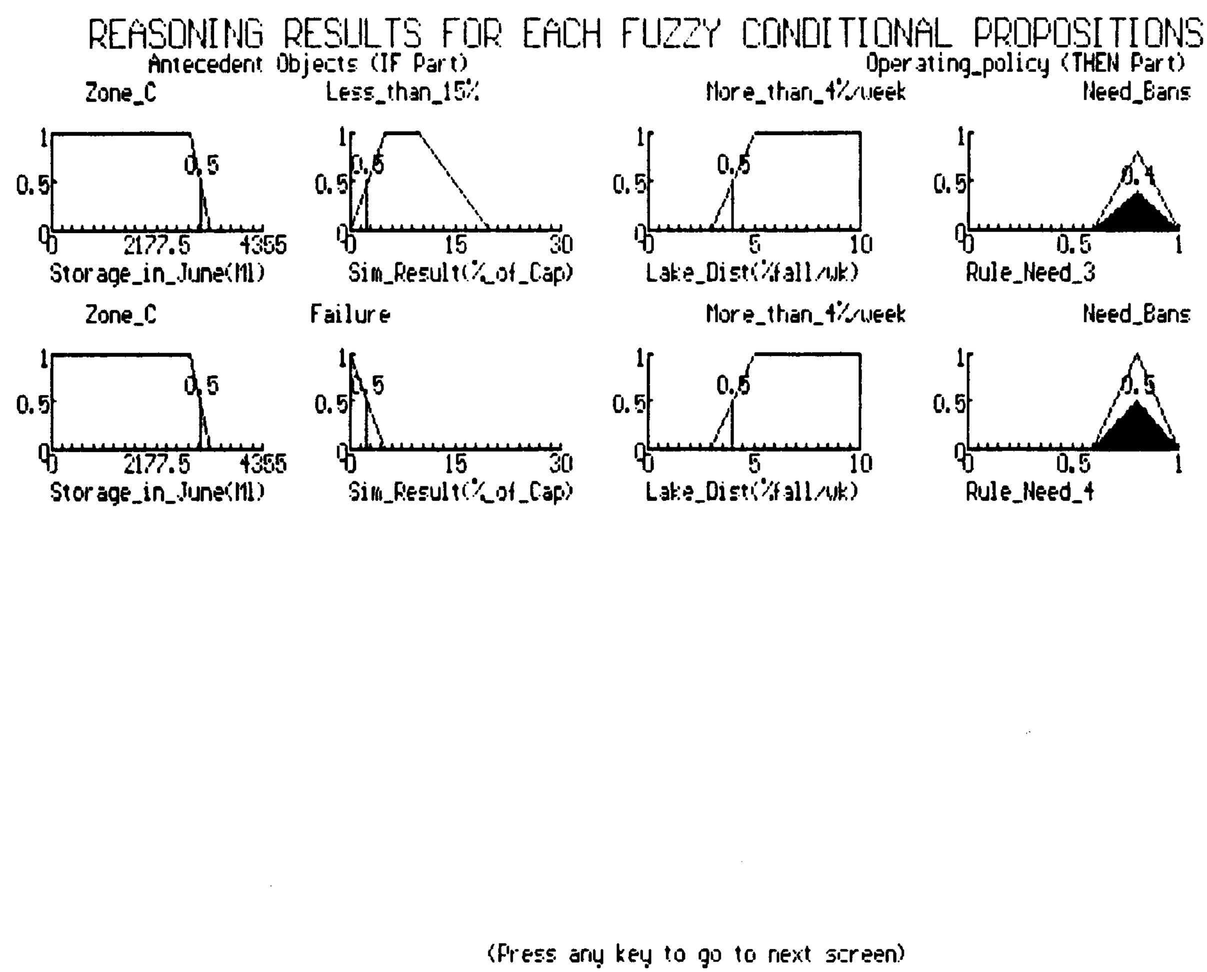


Figure 6-4 (4). Reasoning Results for Each Fuzzy Conditional Propositions by FRESH

For instance, the result of the second rule (Rule_Normal_2) is that the possibility of "Current Storage is in Zone B" is 0.5, and the possibility of "Simulated Future Minimum Storage is More Than 15 % of net capacity" is zero; thus, the possibility of "Operating Policy is Normal Operation" is zero because the AND operator is interpreted as a minimum operation.

The seventh rule (Rule_Concern_1) can be interpreted as: The minimum possibility of antecedent objects is 0.5. Thus, the possibility of consequence may also be 0.5. However, the truth value of this rule is 0.3. Therefore, the possibility of consequence is 0.15 (= 0.5 * 0.3). The filled portion of consequence membership function represents the possibility distribution of reasoning result under given condition.

Figures 6-5 and 6-6 illustrate the matching results and a summary of fuzzy reasoning by FRESH. These reasoning results imply that under the given condition, it is difficult to justify only one consequence. Instead, FRESH suggests that 1) the best relevant consequence is Consider Bans with 39 % matching degree, 2) the second best relevant consequence is Need Bans with also 39 % matching degree, and 3) the third best relevant consequence is Draw from Lake

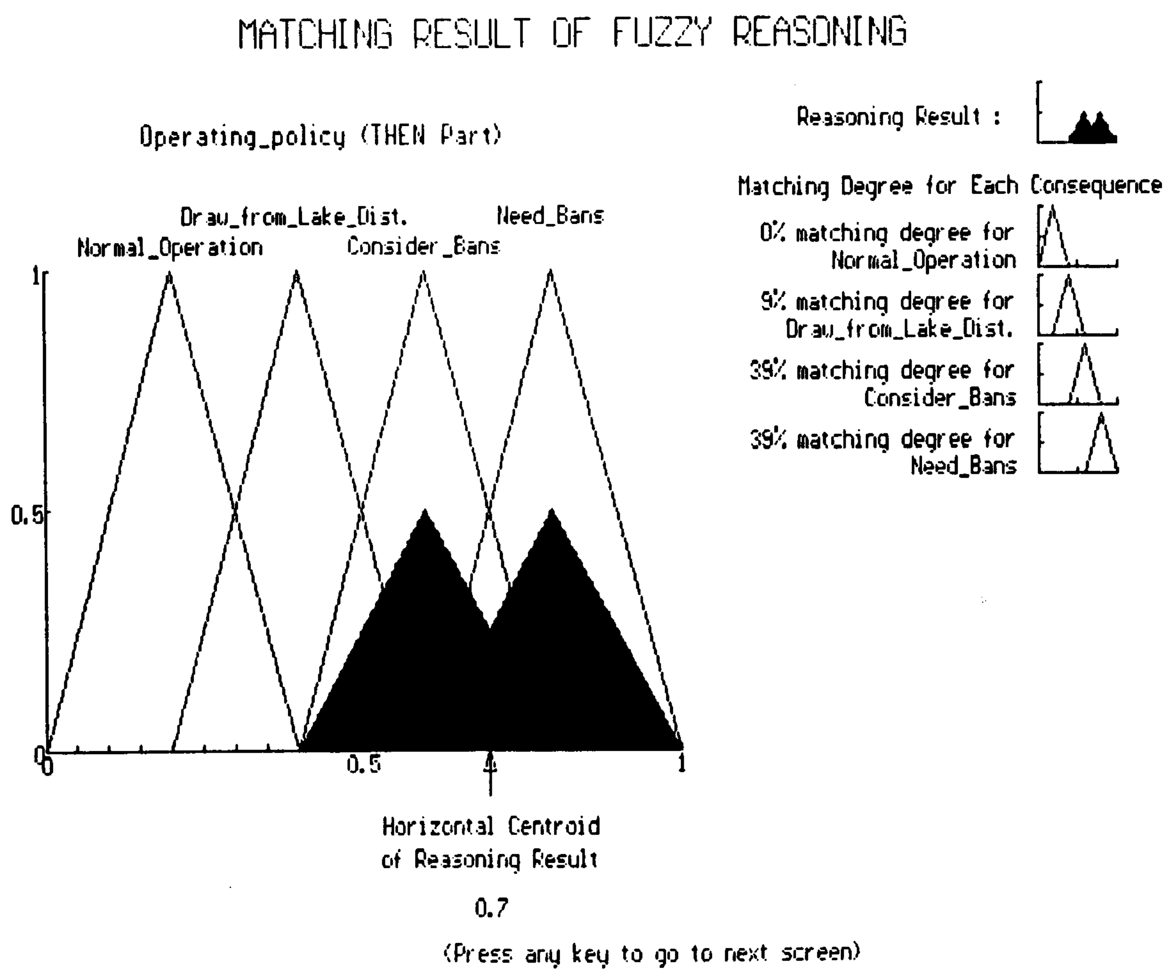


Figure 6-5. Matching Result of Fuzzy Reasoning by FRESH

Summary of Fuzzy Reasoning

=== User Iuput ===

Storage_in_June(Ml) : **3092**   Sim_Result(%_of_Cap) : **2.5**   Lake_Dist(%fall/wk) : **4**

=== Reasoning Results ===

The number of total propositions : **14**

The number of active propositions : **8**

The best three matched consequences :

Consider_Bans with **39** % matching degree.

Need_Bans with **39** % matching degree.

Draw_from_Lake_Dist. with **9** % matching degree.

Horizontal centroid of reasoning result : **0.7**

(Press any key to go to next screen)

Figure 6-6. Summary of Fuzzy Reasoning by FRESH

District with 9 % matching degree. In this example, one of the first two pieces of advice might be used in practical operation because their matching degrees are the same.

Figure 6-7 illustrates the sensitivity analysis results regarding matching degrees of each consequence based on the different conditions shown in Table 6-2. These results suggest that if the given conditions are on the *fuzzy boundaries*, such as just on the control curves, crisp consequence may not be obtained. For the other cases, however, the best relevant consequence is obvious.

Summary

This chapter has presented the results of an application of FRESH to the NWWA water resources management in England. The application illustrates that FRESH can produce results that are considered to approximate the reasoning of an expert.

The final chapter discusses the conclusions of this research and recommendations for future direction.

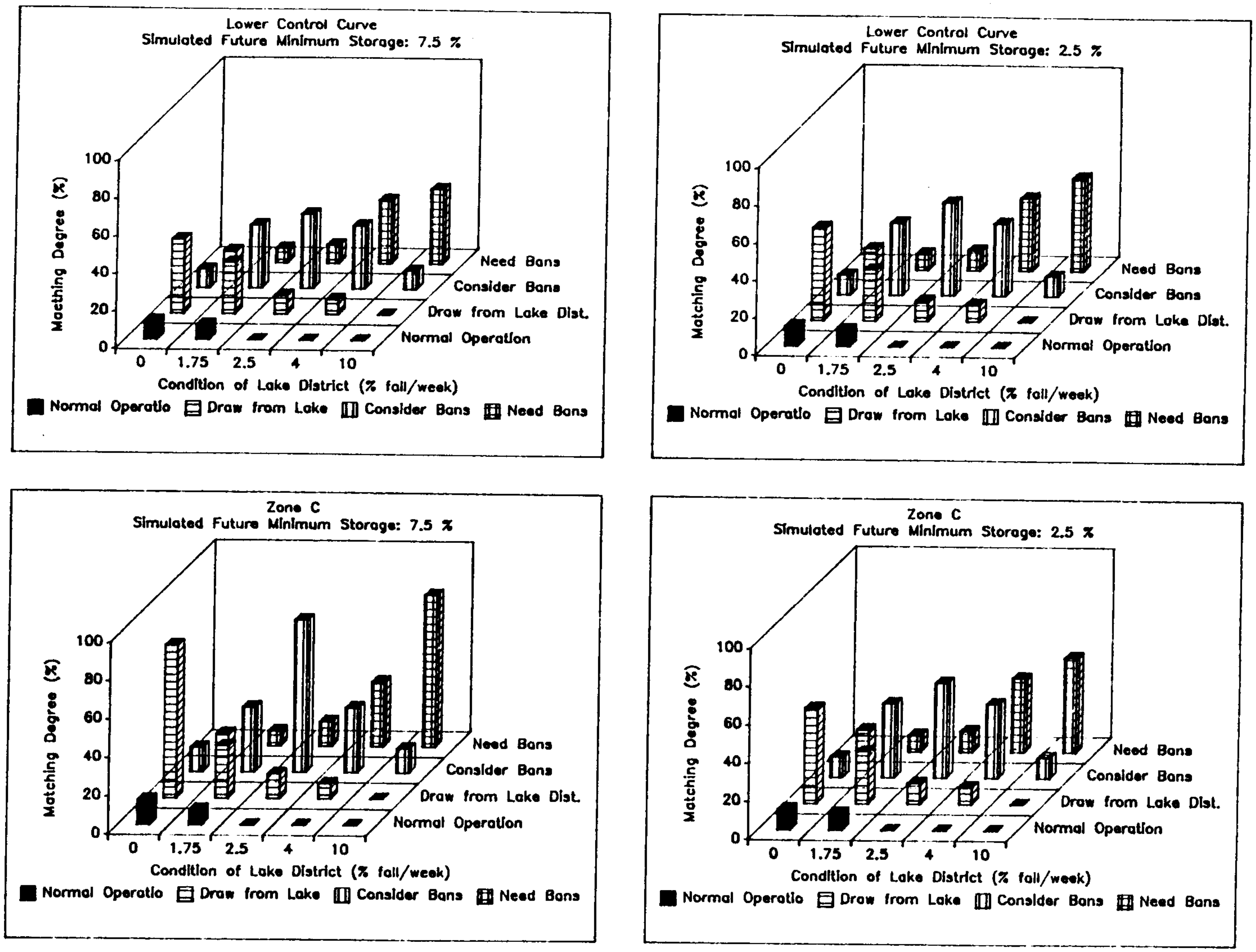


Figure 6-7. Sensitivity Analysis Results of Matching Degrees

CHAPTER SEVEN

Conclusions and Recommendations

Conclusions

This research discusses the use of fuzzy sets in propagating uncertainty in expert system knowledge bases. Also, the research describes software that has an interactive knowledge acquisition environment and a fuzzy logic based inference engine. This software is denoted as FRESH (Fuzzy Reasoning Shell).

FRESH integrates a menu-driven system, user input screens, graphics, and a fuzzy logic based inference engine. With these features and a syntax error evaluation system, an expert can create his/her knowledge base without extensive computing skills. Much of uncertainty in the decision making process contains elements of possibility; therefore, FRESH can be a useful tool to develop knowledge-based systems.

Approximate reasoning using linguistic synthesis by FRESH is applied to the North West Water Authority in England. The example also illustrates an effective application of fuzzy propositional logic.

It is important to note that fuzzy propositional logic allows one to simplify complex decision making systems using linguistic rules. Because of this, it is easier for a programmer to synthesize and implement approximate reasoning in linguistic terms. This fact implies that development of the set of linguistic rules can be concise (only 14 rules for this example).

The *fuzziness* occupies the important role in the human decision making process. Thus, it is critical that knowledge-based systems have the capability to quantify the uncertainty and trace its influence on decision making process. Since FRESH allows one to incorporate all expertise in a system, a novice user can use this system and obtain an expert's approximate reasoning results without being familiar with the full complexity of the decision making process.

The proposed operational rule base used in the linguistic decision making system has a very simple structure. It can be characterized as a collection of conditional statements, all of the type "IF *A* AND *B* AND *C* THEN *D*". However, this structure can be extended to more complex settings including forward/backward chaining in the future.

Recommendations

A major limitation of FRESH is that it can only deal with one-layer reasoning (no forward/backward chaining). In Chapter Four, the example of multi-layer reasoning by fuzzy logic is described. However, fuzzy logic is an active area of research, and the definitions of intersection (the logical AND or minimum operator) and union (the logical OR or minimum operator) are still an arena of active debate for the propagation of uncertainty in multi-layer reasoning (Dubois and Prade, 1989). It is hoped that further theoretical and experimental work will be done on this problem.

Another limitation of FRESH in the application phase is that a programmer has the responsibility of defining the shape of membership functions and propositions based on a trial and error method in order to build successful knowledge bases. Although FRESH provides an interactive knowledge acquisition environment, this process remains a difficult task in the development of knowledge base. Therefore, induction or the machine learning approach should be considered in further research.

The facts, relations, and judgments of human experts contain degree of impression and uncertainty. Thus, the management of possibility in the design of expert systems is a key factor for the successful modeling of reasoning process. The utility of fuzzy set theory and fuzzy logic for this purpose has been and continues to be extensively studied.

BIBLIOGRAPHY

Barkakati, N. (1988). The Waite Group's: Turbo C bible. Indianapolis, IN: Howard W. Sams & Company.

Bellman, R. E. and Zadeh, L. A. (1977). Local and fuzzy logics. In Dunn, J. M. and Epstein, G. (Ed.), Modern uses of multiple-valued logic. (pp. 103-165). Dordrecht, Netherlands: Reidel Publishing Company.

Boose, J. H., Shema, D. B., and Bradshaw, J. M. (1989). Recent progress in AQUINAS: A knowledge acquisition workbench. International Journal of Man-machine Studies, in press.

Borland International. (1988). Turbo C: Reference guide, version 2.0. Scotts Valley, CA.

Buchanan, B. G. and Shortliffe, E. H. (Ed.). (1984). Rule-based expert systems: The MYCIN experiments of the Stanford Heuristic Programming Project. Reading, MA: Addison-Wesley Publishing Company.

Buckley, J. and Siler, W. (1987). Fuzzy operators for possibility interval sets. Fuzzy Sets and Systems, 22(X), 215-227.

Dubois, D. and Prade, H. (1980). Fuzzy Sets and Systems. New York, NY: Academic Press, Inc.

Dubois, D. and Prade, H. (1988). Possibility theory. New York, NY: Plenum Press.

Dubois, D. and Prade, H. (1989). A typology of fuzzy "If... then..." rules. Proceedings of the 3rd IFSA congress. Seattle, WA. 782-785.

Duda, R. O. Hart, P.E., Barrett, P., Gaschnig, J., Konolige, K., Reboh, R, and Slocum, J. (1978). Development of the PROSPECTOR consultant system for mineral exploration. Final report for SRI project 5821 and 6415, Artificial Intelligent Center, SRI International. Menlo Park, CA.

Duda, R. O. and Reboh, R. (1984). AI and decision making: The PROSPECTOR experience. In Reitman, W. (Ed.), Artificial intelligence applications for business. Norwood, NJ: Ablex Publishing Corporation.

Exsys, Inc. (1986). EXSYS demo manual: Expert System development software, version 3.0. Albuqurque, NM: Author.

Jamsa, K. (1988). Turbo C: Programmer's library. Berkeley, CA: Osborne McGraw-Hill.

Kaufmann A. and Gupta, M. M. (1988). Fuzzy mathematical models in engineering and management science. Amsterdam, Netherlands: Elsevier Science Publishers B. S.

Kandel, A. (1986). Fuzzy mathematical techniques with applications. Reading, MA: Addison-Wesley Publishing Company.

Kraft, A. (1984). XCON: An expert configuration system at Digital Equipment Corporation. In Winston, P. H. and Prendergast, K. A.(Ed.), The AI business: The commercial uses of artificial intelligence. Cambridge, MA: The MIT Press.

Klir, G. and Folger, T. A. (1988). Fuzzy sets, uncertainty, and information. Englewood Cliffs, NJ: Prentice Hall.

Level Five Research, Inc. (1986). INSIGHT 2+ reference manual, version 1.0. Indialantic, FL: Author.

Maher, M. L. (Ed.). (1987). Expert systems for civil engineers: Technology and application. New York, NY: American Society of Civil Engineers.

Mamdani, E. H. (1974). Application of fuzzy algorithms for control of simple dynamic plant. Proc. IEE, 121(12), 1585-1588.

Mamdani, E. H. (1977). Application of fuzzy logic to approximate reasoning using linguistic synthesis. IEEE Transactions on Computers, C-26(12), 1182-1191.

McGraw, K. L., and Harbison-Briggs, K. (1989). Knowledge acquisition: Principles and Guidelines. Englewood Criffs, NJ: Prentice Hall.

Mizumoto, M. and Zimmermann, H. J. (1982). Comparison of fuzzy reasoning methods. Fuzzy Sets and Systems, 8(X), 253-283.

Negoita, C. V. (1985). Expert systems and fuzzy systems. Menlo Park, CA: The Benjamin/Cummings Publishing Company, Inc.

Paperback Software International. (1987). VP-Expert: Rule-based expert system development tool. Berkeley, CA: Author.

Pappis, C. P. and Mamdani, E. M. (1977). A fuzzy logic controller for a traffic junction. IEEE Transactions on Systems, Man, and Cybernetics, SMC-7(10), 707-717.

Parker, D. J. and Edmund, C. P. (1980). Water planning in Britain, George Allen & Unwin, London.

Parsaye (1985). In McGraw, K. L., and Harbison-Briggs, K. Knowledge acquisition: Principles and Guidelines (pp. 9-11). Englewood Criffs, NJ: Prentice Hall.

Schmucker, K. J. (1984). Fuzzy sets, natural language computations, and risk analysis. Rockville, ML: Computer Science Press, Inc.

Shortliffe, E. H. (1974). MYCIN: A rule-based computer program for advising physicians regarding antimicrobial therapy selection. Ph.D. dissertation, Stanford University.

Slaughterbeck, C. D. and Palmer, R. N. (1989). Knowledge acquisition for a drought management, MSCE Thesis, University of Washington.

Sugeno, M. (1988). Fuzzy controls. Tokyo, Japan: Nikkan Kogyo Publishing Company. (in Japanese).

Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision process. IEEE Transactions on Systems, Man, and Cybernetics, SMC-3(1), 28-44.

Zadeh, L. A. (1978). Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets and Systems, 1(1), 3-28.

Zadeh, L. A. (1983). The role of fuzzy logic in the management of uncertainty in expert systems. Fuzzy Sets and Systems, 11(3), 199-228.

APPENDIX A

Mathematical Example of Fuzzy Reasoning

This appendix illustrates that both single and multi-layer fuzzy reasoning satisfy modus ponens. In the single-layer reasoning example, the logical-product (Mamdani, 1977) and the algebraic-product (for example, Dubois and Prade, 1980) of fuzzy sets are examined, and fuzzy reasoning under given facts is illustrated. In the multi-layer reasoning example, only the logical-product is examined.

Background of Fuzzy Sets and Fuzzy Logic

Let A be a subset of the universe of discourse X. The characteristic function $M_A(x)$ of classical subset A takes its values in the two-element set $\{0, 1\}$, and is such that $M_A(x) = 1$ if $x \in A$ and $M_A(x) = 0$ otherwise. A fuzzy set A has a characteristic function $M_A(x)$ taking its values in the interval $[0, 1]$, with the grade 1 and 0 representing full membership and nonmembership, respectively. Thus, $M_A(x)$ is called the membership function of A. Since $M_A(x)$ is the grade of membership of $x \in A$, it is also possible to interpret $M_A(x)$ as the degree of possibility (Zadeh, 1983) that x is the value of a parameter fuzzily restricted by A.

A is symbolically denoted $[\{x, M_A(x)\} | x \in A]$.

Several fundamental structures can be defined on the interval [0, 1], such as fuzzy set union, intersection, and complement operator. These functions coincide with those used in classical set theory. Zadeh (1965) first used the operator {[0, 1], max, min, not} that are the most frequently employed, i.e.:

Union : $M_{A \cup B}(x) = \max\{M_A(x), M_B(x)\}$

Intersection: $M_{A \cap B}(x) = \min\{M_A(x), M_B(x)\}$

Complement : $\Gamma M_A(x) = 1 - M_A(x)$

Fuzzy logic is a natural expansion of classical logic that every proposition is either *true* or *false*. Zadeh (1973) proposed a fuzzy logic in which the antecedent involves a fuzzy conditional proposition. For instance, "If x is A then y is B", with A and B being fuzzy concepts.

Logic is the study of the methods and principles of *reasoning* in all its possible form. One area of classical logic is *propositional logic* that requires an object to be either *true* (unity) or *false* (zero). These values are termed *truth values*. Propositional logic deals with combinations of *objects* (or variables) that represent arbitrary propositions. As each object represents a hypothetical proposition, it may assume either of the two truth values. Various forms of *reasoning rules* (or inference rules) can be used for making deductive reasoning.

Modus ponens is one of the most frequently used reasoning rules, i.e.:

$$\{A \cap (A \Rightarrow B)\} \Rightarrow B$$

where $\cap$ is *and-function* (or conjunction) and => is *implication*. Modus ponens states that given two true propositions A and $A \Rightarrow B$ (the *antecedents*), the truth of the proposition B (the *consequence*) may be inferred.

Single-Layer Reasoning

First, we shall consider the following simple form of approximate reasoning that includes a fuzzy conditional proposition:

Ant : IF x is A
THEN y is B WITH TV.
Fact: x is $A_{\prime}$.
----
cons: y is B'.

where x and y are the names of objects; A, A' ("similar to" A), B, and B' ("similar to" B) are the labels of fuzzy subsets in the universes of discourse X, X, Y, and Y, respectively; and TV is the truth value of the antecedent.

This reasoning scheme is called *generalized modes ponens*, and has two major differences from classical modes ponens (Bellman and Zadeh, 1977). First, A' is not required to be identical with A, as it is in the classical case. Second, A, A', and B are not required to be crisp.

The implied relation between the two objects x and y is expressed in terms of the cartesian product of the two fuzzy subsets A and B. The cartesian product, characterized by a function $M_R(x,y)$ by which each pair (x,y) is assigned a number in [0, 1] indicating the extent to which the relation R is true for (x,y), of two fuzzy subset A and B is denoted as $A*B$. The following two equations illustrate two different definitions of cartesian product, the logical-product and the algebraic-product.

Logical-Product

$$A*B = \Sigma_i \ \Sigma_j \ \min \ \{M_A(x_i), \ M_B(y_j)\}$$

Algebraic-Product

$$A*B = \Sigma_i \ \Sigma_j \ \{M_A(x_i) \ * \ M_B(y_j)\}$$

where Σ stands for union; x_i and y_j are generic elements of the universes of discourse X and Y, respectively; $i = 1, 2, \ldots, m$; and $j = 1, 2, \ldots, n$.

These equations can be used to infer the fuzzy subset of the second object given a membership function, or a value, for the first. Suppose we denote the relationship between two objects by R_{AB} (A=>B). A' is given membership function, then, the fuzzy subset y is inferred by:

$$B' = A' \text{ o } R_{AB}$$

where o is composition. Composition is performed by the maximum-product of A' and R_{AB}. For example, let:

$$M_A(x_i) = [\ 0.0 \quad 0.5 \quad 1.0 \quad 0.5 \quad 0.0\ ]$$
$$M_B(y_j) = [\ 0.6 \quad 0.8 \quad 0.4 \quad 0.0\ ]$$

Figure A-1 illustrates the membership functions of A and B. The conditional inference of this form is:

"If x is nearly equal 3 then y is nearly equal 2."

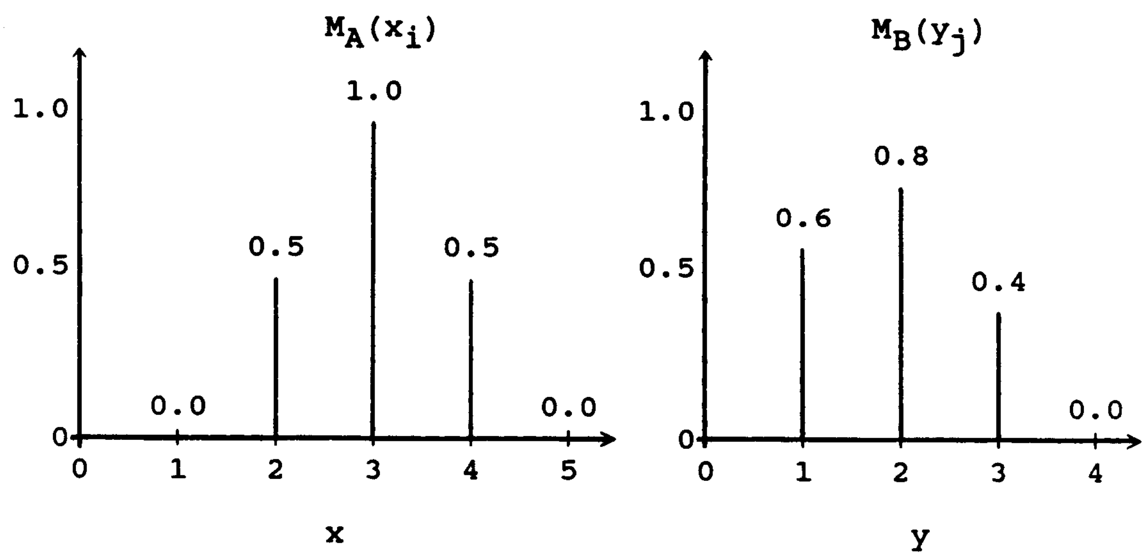


Figure A-1. Membership Functions of A and B

Logical-Product:

$$R_{AB}(i, j) = \min_{i,j} \{M_A(x_i), M_B(y_j)\}$$

$$= \begin{vmatrix} 0.0 & 0.0 & 0.0 & 0.0 \\ 0.5 & 0.5 & 0.4 & 0.0 \\ 0.6 & 0.8 & 0.4 & 0.0 \\ 0.5 & 0.5 & 0.4 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{vmatrix}$$

If A' is equal to A, B' must be equal to B. This relationship is *modus ponens*.

$$M_B'(y_j) = \max_i[\min_j\{ M_A(x_i), R_{AB}(i, j) \}]$$

$$= \max_i \begin{vmatrix} 0.0 & 0.0 & 0.0 & 0.0 \\ 0.5 & 0.5 & 0.4 & 0.0 \\ 0.6 & 0.8 & 0.4 & 0.0 \\ 0.5 & 0.5 & 0.4 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{vmatrix}$$

$$= [\ 0.6 \quad 0.8 \quad 0.4 \quad 0.0\ ]$$

$$= M_B(y_j)$$

Thus, the logical-product satisfies modus ponens.

In the FRESH application, A' and B' were chosen to be nonfuzzy vectors and with only one element equal to unity and the remaining equal to zero. For example, let:

$$M_A'(x_i) = [\ 0.0 \quad 1.0 \quad 0.0 \quad 0.0 \quad 0.0\ ]$$

Then, the consequence $M_B'(y_j)$ will be:

$$M_B'(y_j) = \max_i[\ \min_j\{\ M_A'(x_i),\ R_{AB}(i,\ j)\ \}\ ]$$

$$= \max_i \begin{vmatrix} 0.0 & 0.0 & 0.0 & 0.0 \\ 0.5 & 0.5 & 0.4 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{vmatrix}$$

$$= [\ 0.5 \quad 0.5 \quad 0.4 \quad 0.0\ ]$$

This result implies that even though the given fact is not exactly identical with the condition of the relationship, fuzzy logic infers something according to the relationship.

Algebraic-Product:

$$R_{AB}(i,\ j) = \{M_A(x_i) * M_B(y_j)\}$$

$$= \begin{vmatrix} 0.0 & 0.0 & 0.0 & 0.0 \\ 0.3 & 0.4 & 0.2 & 0.0 \\ 0.6 & 0.8 & 0.4 & 0.0 \\ 0.3 & 0.4 & 0.2 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{vmatrix}$$

If A' is equal to A:

$$M_B'(y_j) = \max_i[\ \{\ M_A(x_i) * R_{AB}(i,\ j)\ \}\ ]$$

$$= \max_i \begin{vmatrix} 0.00 & 0.00 & 0.00 & 0.00 \\ 0.15 & 0.20 & 0.10 & 0.00 \\ 0.60 & 0.80 & 0.40 & 0.00 \\ 0.15 & 0.20 & 0.10 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 \end{vmatrix}$$

$$= [\ 0.6 \quad 0.8 \quad 0.4 \quad 0.0\ ]$$

$$= M_B(y_j)$$

Thus, the algebraic-product also satisfies modus ponens.

It should be noted that the maximum *possibility* of antecedent object must be equal to unity to satisfy *modus ponens;* however, the maximum *possibility* of a consequence object can assume a value less than unity. This fact implies that the management of truth value (TV) can be dealt under the same framework of fuzzy logic.

Under the same condition as logical-product, let:

$$M_A'(x_i) = [\ 0.0 \quad 1.0 \quad 0.0 \quad 0.0 \quad 0.0\ ]$$

Then, the consequence $M_B'(y_j)$ will be:

$$M_B'(y_j) = \max_i[\ M_A'(x_i) * R_{AB}(i,\ j)\ \}\ ]$$

$$= \max_i \begin{vmatrix} 0.0 & 0.0 & 0.0 & 0.0 \\ 0.3 & 0.4 & 0.2 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{vmatrix}$$

$$= [\ 0.3 \quad 0.4 \quad 0.2 \quad 0.0\ ]$$

Even though the same fact is used, the reasoning result of the algebraic-product is different from the result of the logical-product. The result by the logical-product shows that the upper portion of the consequence membership function is truncated, and the shape of membership function is changed. By contrast, the result by the algebraic product illustrates that the consequence membership function is algebraically multiplied by the grade of the antecedent membership function associated with the given fact, and the original shape of membership is maintained.

Sugeno (1984) suggests that using the algebraic-product can improve reasoning results because of its property that maintain the original shape of the consequence membership function. Therefore, algebraic-product is adopted as a fuzzy reasoning operator for this research.

More complex rules are developed by use of two fuzzy antecedent objects, *x1* and *x2*, that are connected by AND.

Ant : IF *x1* is *A* AND *x2* is *B*
THEN *y* is *C*.
Fact1: *x1* is *A'*.
Fact2: *x2* is *B'*.

Cons : *y* is *C'*.

In this example, the algebraic-product and composition can be defined as:

$$A*C \text{ AND } B*C = \Sigma_i \Sigma_k (M_A(x1_i) * M_C(y_k)) \cap \Sigma_j \Sigma_k (M_B(x2_j) * M_C(y_k))$$

$$C' = (A' \circ R_{AC}) \cap (B' \circ R_{BC})$$

For example, let:

$$M_A(x1_i) = [\ 0.0 \quad 0.5 \quad 1.0 \quad 0.5 \quad 0.0\ ]$$

$$M_B(x2_j) = [\ 1.0 \quad 0.6 \quad 0.0\ ]$$

$$M_C(Y_k) = [\ 0.6 \quad 0.8 \quad 0.4 \quad 0.0\ ]$$

The AND operator can be interpreted as the *minimum* (or *intersection*) operation. Thus,

$$R_{AC}(i,\ k) = \{M_A(x1_i) * M_C(Y_k)\}$$

$$= \begin{vmatrix} 0.0 & 0.0 & 0.0 & 0.0 \\ 0.3 & 0.4 & 0.2 & 0.0 \\ 0.6 & 0.8 & 0.4 & 0.0 \\ 0.3 & 0.4 & 0.2 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{vmatrix}$$

$$R_{BC}(j,\ k) = \{M_B(x2_j) * M_C(Y_k)\}$$

$$= \begin{vmatrix} 0.60 & 0.80 & 0.40 & 0.00 \\ 0.36 & 0.48 & 0.24 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 \end{vmatrix}$$

If *A'* and *B'* are equal to *A* and *B*,

$$M_C'(Y_k) = \max_i[\ \{\ M_A(x1_i) * R_{AC}(i,\ k)\ \}\ ]$$
$$\cap \max_j[\ \{\ M_B(x2_j) * R_{BC}(j,\ k)\ \}\ ]$$

$$= \max_i \begin{vmatrix} 0.00 & 0.00 & 0.00 & 0.00 \\ 0.15 & 0.20 & 0.10 & 0.00 \\ 0.60 & 0.80 & 0.40 & 0.00 \\ 0.15 & 0.20 & 0.10 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 \end{vmatrix}$$

$$\cap \max_j \begin{vmatrix} 0.60 & 0.80 & 0.40 & 0.00 \\ 0.22 & 0.28 & 0.14 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 \end{vmatrix}$$

$$= [\ 0.6 \quad 0.8 \quad 0.4 \quad 0.0\ ]$$

$$\cap\ [\ 0.6 \quad 0.8 \quad 0.4 \quad 0.0\ ]$$

$$= M_C(y_k)$$

Thus, also in this case, algebraic-product satisfies modus ponens.

If A' and B' are not equal to A and B, let:

$$M_A'(x1_i) = [\ 0.0 \quad 1.0 \quad 0.0 \quad 0.0 \quad 0.0\ ]$$

$$M_B'(x2_j) = [\ 0.0 \quad 1.0 \quad 0.0\ ]$$

Then, the consequence $M_C'(y_k)$ will be:

$$M_C'(y_k) = \max_i[\ \{\ M_A'(x1_i),\ R_{AC}(i,\ k)\ \}\ ]$$

$$\cap \max_j[\ \{\ M_B'(x2_j),\ R_{BC}(j,\ k)\ \}\ ]$$

$$= \max_i \begin{vmatrix} 0.0 & 0.0 & 0.0 & 0.0 \\ 0.3 & 0.4 & 0.2 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{vmatrix}$$

$$\cap \max_j \begin{vmatrix} 0.00 & 0.00 & 0.00 & 0.00 \\ 0.36 & 0.48 & 0.24 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 \end{vmatrix}$$

$$= [\ 0.3 \quad 0.4 \quad 0.2 \quad 0.0\ ]$$

$$\cap\ [\ 0.36 \quad 0.48 \quad 0.24 \quad 0.0\ ]$$

$$= [\ 0.3 \quad 0.4 \quad 0.2 \quad 0.0\ ]$$

Finally, the OR operator is considered with the AND operator.

Ant : IF (*x1* is *A* AND *x2* is *B*) OR (*x3* is *C*)
THEN *y* is *D*.
Fact1: *x1* is *A'*.
Fact2: *x2* is *B'*.
Fact3: *x3* is *C'*.

Cons : *y* is *D'*.

This form of fuzzy reasoning can be divided into the following two sets of fuzzy conditional propositions.

Ant 1: IF *x1* is *A* AND *x2* is *B*
THEN *y* is *D*.
Fact1: *x1* is *A'*.
Fact2: *x2* is *B'*.

Cons1: *y* is *D'*.

OR

Ant 2: IF *x3* is *C*
THEN *y* is *D*.
Fact3: *x3* is *C'*.

Cons2: *y* is *D'*.

Thus,

Cons: Cons1 OR Cons2.

The algebraic-product and composition is defined as:

$$(A{*}D \text{ AND } B{*}D) \text{ OR } C{*}D$$

$$= [\Sigma_i \Sigma_l \{M_A(x1_i) * M_D(y_l)\} \cap \Sigma_j \Sigma_l \{M_B(x2_j) * M_D(y_l)\}] \cup [\Sigma_k \Sigma_l \{M_C(x3_k) * M_D(y_l)\}]$$

$$D' = [(A' \circ R_{AD}) \cap (B' \circ R_{BD})] \cup [(C' \circ R_{CD})]$$

Composition and the AND operator are performed in same manner. Then, the OR operator is interpreted as the *maximum* (or *union*) operation, and connects these two sets of fuzzy conditional propositions.

For example, let:

$$M_A(x1_i) = [\ 0.0 \quad 0.5 \quad 1.0 \quad 0.5 \quad 0.0\ ]$$

$$M_B(x2_j) = [\ 1.0 \quad 0.6 \quad 0.0\ ]$$

$$M_C(y_k) = [\ 0.6 \quad 0.8 \quad 0.4 \quad 0.0\ ]$$

$$R_{AD}(i,\ l) = \{M_A(x1_i) * M_D(y_l)\}$$

$$= \begin{vmatrix} 0.0 & 0.0 & 0.0 & 0.0 \\ 0.3 & 0.4 & 0.2 & 0.0 \\ 0.6 & 0.8 & 0.4 & 0.0 \\ 0.3 & 0.4 & 0.2 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{vmatrix}$$

$$R_{BD}(j, l) = \{M_B(x2_j) * M_D(y_l)\}$$

$$= \begin{vmatrix} 0.60 & 0.80 & 0.40 & 0.00 \\ 0.36 & 0.48 & 0.24 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 \end{vmatrix}$$

$$R_{CD}(k, l) = \{M_C(x3_k) * M_D(y_l)\}$$

$$= \begin{vmatrix} 0.60 & 0.80 & 0.40 & 0.00 \\ 0.12 & 0.16 & 0.08 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 \end{vmatrix}$$

If A', B' and C' are equal to A, B and C,

$$M_D'(y_l) = [\ \max_i[\ \{\ M_A(x1_i) * R_{AD}(i, l)\ \}\ ]$$
$$\cap \max_j[\ \{\ M_B(x2_j) * R_{BD}(j, l)\ \}\ ]\ ]$$
$$\cup\ [\ \max_k[\ \{\ M_C(x3_k) * R_{CD}(k, l)\ \}\ ]\ ]$$

$$= [\ 0.6 \quad 0.8 \quad 0.4 \quad 0.0\ ] \text{ (See previous example)}$$

$$\cup \max_j \begin{vmatrix} 0.60 & 0.80 & 0.40 & 0.00 \\ 0.02 & 0.03 & 0.02 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 \end{vmatrix}$$

$$= [\ 0.6 \quad 0.8 \quad 0.4 \quad 0.0\ ]$$
$$\cup\ [\ 0.6 \quad 0.8 \quad 0.4 \quad 0.0\ ]$$

$$= [\ 0.6 \quad 0.8 \quad 0.4 \quad 0.0\ ]$$

$$= M_D(y_l)$$

Thus, algebraic-product satisfies modus ponens in this case also.

If A', B', and C' are not equal to A, B and C, let:

$$M_A'(x1_i) = [\ 0.0 \quad 1.0 \quad 0.0 \quad 0.0 \quad 0.0\ ]$$

$$M_B'(x2_j) = [\ 0.0 \quad 1.0 \quad 0.0\ ]$$

$$M_C'(x3_k) = [\ 0.0 \quad 1.0 \quad 0.0\ ]$$

Then, the consequence $M_D'(y_1)$ will be:

$$\begin{aligned} M_D'(y_1) = [\ & \max_i[\ \{\ M_A'(x1_i) * R_{AD}(i, 1)\ \}\ ] \\ & \cap \max_j[\ \{\ M_B'(x2_j) * R_{BD}(j, 1)\ \}\ ]\ ] \\ & \cup\ [\ \max_k[\ \{\ M_C'(x3_k) * R_{CD}(k, 1)\ \}\ ]\ ] \end{aligned}$$

$$= [\ 0.3 \quad 0.4 \quad 0.2 \quad 0.0\ ] \text{ (See previous example)}$$

$$\cup \max_j \begin{vmatrix} 0.00 & 0.00 & 0.00 & 0.00 \\ 0.12 & 0.16 & 0.08 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 \end{vmatrix}$$

$$= [\ 0.3 \quad 0.4 \quad 0.2 \quad 0.0\ ]$$

$$\cup\ [\ 0.12 \quad 0.16 \quad 0.08 \quad 0.00\ ]$$

$$= [\ 0.3 \quad 0.4 \quad 0.2 \quad 0.0\ ]$$

Multi-Layer Reasoning

Multi-layer reasoning by fuzzy logic is an active area of research and controversial topic. Therefore, this appendix illustrates the simplest two-layer reasoning using the logical-product. Let,

$$\begin{array}{lccc} \text{value:} & 1 & 2 & 3 \\ M_{A1} = [& 0.2 & 0.5 & 1.0], \end{array} \qquad \begin{array}{lccc} \text{value:} & 2 & 4 & 6 \\ M_{B1} = [& 0.1 & 0.3 & 1.0] \end{array}$$

$$R_{AB1} = \begin{vmatrix} 0.1 & 0.1 & 0.1 \\ 0.2 & 0.3 & 0.3 \\ 0.2 & 0.5 & 1.0 \end{vmatrix}$$

$$\begin{array}{lccc} \text{value:} & 3 & 6 & 9 \\ M_{C1} = [& 0.8 & 1.0 & 0.1] \end{array}$$

$$R_{BC1} = \begin{vmatrix} 0.1 & 0.3 & 0.8 \\ 0.1 & 0.3 & 1.0 \\ 0.1 & 0.1 & 0.1 \end{vmatrix}$$

The example of this reasoning is:

If rainfall (A) is about 3 then inflow (B) is about 6.

If inflow (B)is about 6 then release (C) is about 6.

We can also have the other rules, like:

$$\begin{array}{lccc} \text{value:} & 1 & 2 & 3 \\ M_{A2} = [& 0.6 & 1.0 & 0.8], \end{array} \qquad \begin{array}{lccc} \text{value:} & 2 & 4 & 6 \\ M_{B2} = [& 0.4 & 1.0 & 0.6] \end{array}$$

$$R_{AB2} = \begin{vmatrix} 0.4 & 0.4 & 0.4 \\ 0.6 & 1.0 & 0.8 \\ 0.6 & 1.0 & 0.6 \end{vmatrix}$$

$$\begin{array}{lccc} \text{value:} & 3 & 6 & 9 \\ M_{C2} = [& 1.0 & 0.5 & 0.2] \end{array}$$

$$R_{BC2} = \begin{vmatrix} 0.4 & 1.0 & 0.6 \\ 0.4 & 0.5 & 0.5 \\ 0.2 & 0.2 & 0.2 \end{vmatrix}$$

The example of this reasoning is:

If rainfall (A) is about 2 then inflow (B) is about 4.

If inflow (B)is about 4 then release (C) is about 3.

Figure A-2 illustrates these membership functions.

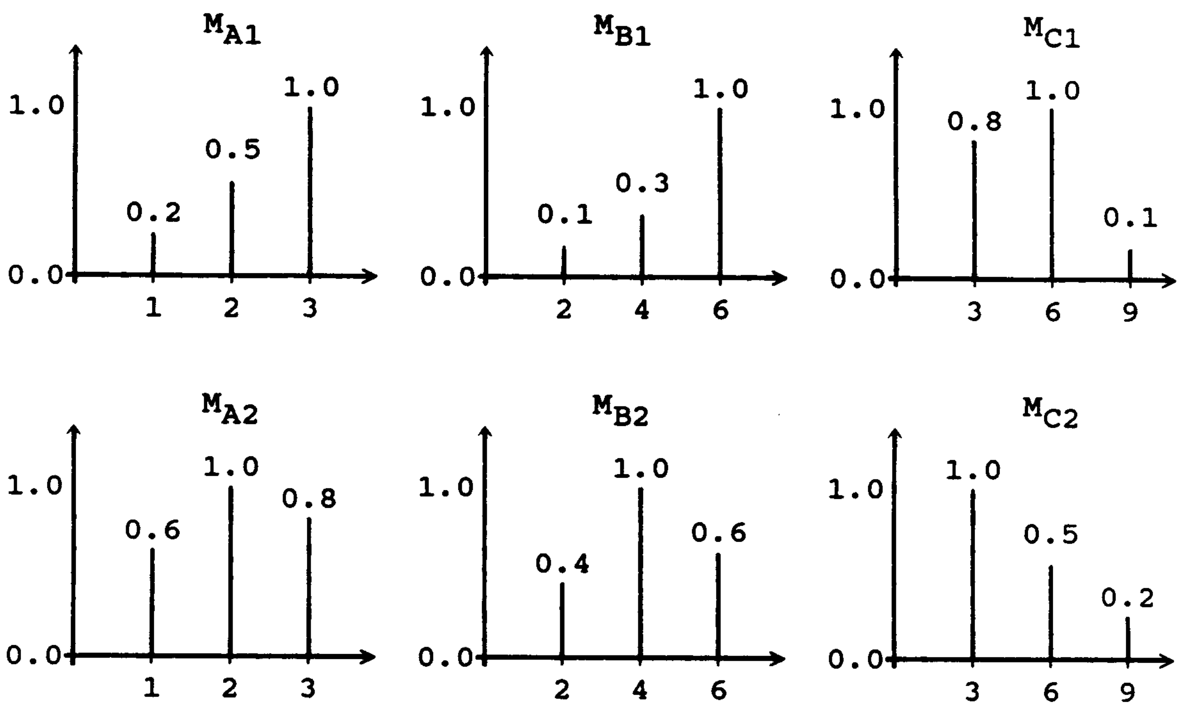


Figure A-2. Membership Functions of *A1, B1, C1, A2, B2,* and *C2*

If the rainfall is about 3 (A' = A1), then the consequence membership function of inflow (M_{B1}') must be equal to "inflow is about 6" (M_{B1}).

$$M_{B1}' = M_A' \circ R_{AB1}$$

$$= [\ 0.2 \quad 0.5 \quad 1.0\ ] \circ \begin{vmatrix} 0.1 & 0.1 & 0.1 \\ 0.2 & 0.3 & 0.3 \\ 0.2 & 0.5 & 1.0 \end{vmatrix}$$

$$= \begin{vmatrix} 0.1 & 0.1 & 0.1 \\ 0.2 & 0.3 & 0.3 \\ 0.2 & 0.5 & 1.0 \end{vmatrix} \quad \text{Minimum Operator}$$

$$= \begin{vmatrix} 0.1 \\ 0.3 \\ 1.0 \end{vmatrix} \quad \text{Maximum Operator}$$

$$= M_{B1}$$

where o is composition. Thus, modus Ponens is satisfied.

At the same time:

$$M_{B2}' = M_A' \circ R_{AB2}$$

$$= [\ 0.2 \quad 0.5 \quad 1.0\ ] \circ \begin{vmatrix} 0.4 & 0.4 & 0.4 \\ 0.6 & 1.0 & 0.8 \\ 0.6 & 1.0 & 0.6 \end{vmatrix}$$

$$= \begin{vmatrix} 0.2 & 0.4 & 0.4 \\ 0.2 & 0.5 & 0.8 \\ 0.2 & 0.5 & 0.6 \end{vmatrix} \quad \text{Minimum Operator}$$

$$= \begin{vmatrix} 0.4 \\ 0.8 \\ 0.6 \end{vmatrix} \quad \text{Maximum Operator}$$

Therefore,

$$M_B' = M_{B1}' \text{ OR } M_{B2}'$$
$$= [\ 0.1 \quad 0.3 \quad 1.0\ ] \cap [\ 0.4 \quad 0.8 \quad 0.6\ ]$$
$$= [\ 0.4 \quad 0.8 \quad 1.0\ ]$$

This is not equal to M_{B1} because OR operator takes maximum operation of M_{B1}' and M_{B2}'.

The next layer's reasoning result will be:

$$M_{C1}' = M_B' \circ R_{BC1}$$
$$= [\ 0.4 \quad 0.8 \quad 1.0\ ] \circ \begin{vmatrix} 0.1 & 0.3 & 0.8 \\ 0.1 & 0.3 & 1.0 \\ 0.1 & 0.1 & 0.1 \end{vmatrix}$$
$$= \begin{vmatrix} 0.1 & 0.3 & 0.8 \\ 0.1 & 0.3 & 1.0 \\ 0.1 & 0.1 & 0.1 \end{vmatrix} \quad \text{Minimum Operator}$$
$$= \begin{vmatrix} 0.8 \\ 1.0 \\ 0.1 \end{vmatrix} \quad \text{Maximum Operator}$$
$$= M_{C1}$$

Thus, modus Ponens is also satisfied.

At the same time:

$$M_{C2}' = M_B' \circ R_{BC2}$$
$$= [\ 0.4 \quad 0.8 \quad 1.0\ ] \circ \begin{vmatrix} 0.4 & 1.0 & 0.6 \\ 0.4 & 0.5 & 0.5 \\ 0.2 & 0.2 & 0.2 \end{vmatrix}$$
$$= \begin{vmatrix} 0.4 & 0.8 & 0.6 \\ 0.4 & 0.5 & 0.5 \\ 0.2 & 0.2 & 0.2 \end{vmatrix} \quad \text{Minimum Operator}$$

$$= \begin{vmatrix} 0.8 \\ 0.5 \\ 0.2 \end{vmatrix} \quad \text{Maximum Operator}$$

Therefore,

$$M_C' = M_{C1}' \text{ OR } M_{C2}'$$
$$= [\ 0.8 \quad 1.0 \quad 0.1\ ] \cap [\ 0.8 \quad 0.5 \quad 0.2\ ]$$
$$= [\ 0.8 \quad 1.0 \quad 0.2\ ]$$

This is not equal to M_{C1} because OR operator takes maximum operation of M_{C1}' and M_{C2}'.

This example illustrates that modes ponens is also satisfied in multi-layer reasoning using the logical-product.

APPENDIX B

Installation of FRESH

FRESH runs on the IBM PC or any IBM compatible computer that has a color EGA or VGA monitor, a hard disk drive, 640 K RAM, and PC-DOS/MS-DOS operating system.

Installation of FRESH is very straightforward, and requires a minimal understanding of DOS to perform. FRESH files must be copied to a hard disk. There must be over 600 K of storage space remaining on the hard disk.

After the computer has turned-on, create a 'FRESH' directory in the C (hard disk) drive to contain FRESH files. This is accomplished by typing MD FRESH at the C: prompt and pressing the <ENTER> key. This will create the FRESH subdirectory on the C drive. Next, move to the new directory by typing:

```
C:\> CD FRESH
```

followed by <ENTER> key. At this point the user is ready to copy FRESH files to the hard disk.

Place the FRESH program disk in the A drive. Enter at the DOS prompt:

```
C:\FRESH> COPY A:*.*
```

then press <ENTER> key. The screen will display the name of each file as it is copied to the hard disk.

After installation of FRESH, enter at the DOS prompt:

C:\FRESH> FRESH

then press <ENTER> key. Learning to user FRESH is a very simple process. Since FRESH adopts a menu-driven system, the user can immediately begin using the sample knowledge base provided that is explained in Chapter Five.

APPENDIX C

Listing of a Sample Knowledge Base by FRESH

```
============================
=                          =
=     List of Data File    =
=                          =
============================

             File Name : C:\FRESH\NWWF_JUN.DAT
 Number of Subintervals :     20
      Number of Objects :      4
 Number of Propositions :     14

          =====  Object Data  =====

           Object Name :    Operating_policy
      Minimum Boundary :   0.00
      Maximum Boundary :   1.00
Number of Menbership Functions :   4

   Membership Function Name :    Normal_Operation

  X-Axis 1      X-Axis 2      X-Axis 3      X-Axis 4
     0.00          0.20          0.20          0.40

   Membership Function Name : Draw_from_Lake_Dist.

  X-Axis 1      X-Axis 2      X-Axis 3      X-Axis 4
     0.20          0.40          0.40          0.60

   Membership Function Name :        Consider_Bans

  X-Axis 1      X-Axis 2      X-Axis 3      X-Axis 4
     0.40          0.60          0.60          0.80

   Membership Function Name :            Need_Bans

  X-Axis 1      X-Axis 2      X-Axis 3      X-Axis 4
     0.60          0.80          0.80          1.00

           Object Name :  Storage_in_June(Ml)
      Minimum Boundary :   0.00
      Maximum Boundary : 4355.00
Number of Menbership Functions :   3

   Membership Function Name :               Zone_C

  X-Axis 1      X-Axis 2      X-Axis 3      X-Axis 4
     0.00          0.00       2892.00       3292.00

   Membership Function Name :               Zone_B

  X-Axis 1      X-Axis 2      X-Axis 3      X-Axis 4
  2892.00       3292.00       3635.00       4035.00
```

Membership Function Name : Zone_A

X-Axis 1	X-Axis 2	X-Axis 3	X-Axis 4
3635.00	4035.00	4355.00	4355.00

Object Name : Sim_Result(%_of_Cap)
Minimum Boundary : 0.00
Maximum Boundary : 30.00
Number of Menbership Functions : 3

Membership Function Name : Failure

X-Axis 1	X-Axis 2	X-Axis 3	X-Axis 4
0.00	0.00	0.00	5.00

Membership Function Name : Less_than_15%

X-Axis 1	X-Axis 2	X-Axis 3	X-Axis 4
0.00	5.00	10.00	20.00

Membership Function Name : More_than_15%

X-Axis 1	X-Axis 2	X-Axis 3	X-Axis 4
10.00	20.00	30.00	30.00

Object Name : Lake_Dist(%fall/wk)
Minimum Boundary : 0.00
Maximum Boundary : 10.00
Number of Menbership Functions : 3

Membership Function Name : Healthy

X-Axis 1	X-Axis 2	X-Axis 3	X-Axis 4
0.00	0.00	1.50	2.00

Membership Function Name : Less_than_4%/week

X-Axis 1	X-Axis 2	X-Axis 3	X-Axis 4
1.50	2.00	3.00	5.00

Membership Function Name : More_than_4%/week

X-Axis 1	X-Axis 2	X-Axis 3	X-Axis 4
3.00	5.00	10.00	10.00

```
                     =====  Proposition Data  =====

              Proposition Name :          Rule_Normal_1

   IF    Storage_in_June(Ml)  IS                   Zone_A
 THEN       Operating_policy  IS        Normal_Operation   WITH   TV =   1.00


              Proposition Name :          Rule_Normal_2

   IF    Storage_in_June(Ml)  IS                   Zone_B
  AND   Sim_Result(%_of_Cap)  IS             More_than_15%
 THEN       Operating_policy  IS        Normal_Operation   WITH   TV =   1.00


              Proposition Name :            Rule_Draw_1

   IF    Storage_in_June(Ml)  IS                   Zone_B
  AND   Sim_Result(%_of_Cap)  IS             Less_than_15%
  AND    Lake_Dist(%fall/wk)  IS                  Healthy
 THEN       Operating_policy  IS  Draw_from_Lake_Dist.     WITH   TV =   0.30


              Proposition Name :            Rule_Draw_2

   IF    Storage_in_June(Ml)  IS                   Zone_B
  AND   Sim_Result(%_of_Cap)  IS                  Failure
  AND    Lake_Dist(%fall/wk)  IS                  Healthy
 THEN       Operating_policy  IS  Draw_from_Lake_Dist.     WITH   TV =   0.50


              Proposition Name :            Rule_Draw_3

   IF    Storage_in_June(Ml)  IS                   Zone_C
  AND   Sim_Result(%_of_Cap)  IS             Less_than_15%
  AND    Lake_Dist(%fall/wk)  IS                  Healthy
 THEN       Operating_policy  IS  Draw_from_Lake_Dist.     WITH   TV =   0.80


              Proposition Name :            Rule_Draw_4

   IF    Storage_in_June(Ml)  IS                   Zone_C
  AND   Sim_Result(%_of_Cap)  IS                  Failure
  AND    Lake_Dist(%fall/wk)  IS                  Healthy
 THEN       Operating_policy  IS  Draw_from_Lake_Dist.     WITH   TV =   1.00


              Proposition Name :         Rule_Concern_1

   IF    Storage_in_June(Ml)  IS                   Zone_B
  AND   Sim_Result(%_of_Cap)  IS             Less_than_15%
  AND    Lake_Dist(%fall/wk)  IS         Less_than_4%/week
 THEN       Operating_policy  IS             Consider_Bans   WITH   TV =   0.30


              Proposition Name :         Rule_Concern_2

   IF    Storage_in_June(Ml)  IS                   Zone_B
  AND   Sim_Result(%_of_Cap)  IS                  Failure
  AND    Lake_Dist(%fall/wk)  IS         Less_than_4%/week
 THEN       Operating_policy  IS             Consider_Bans   WITH   TV =   0.50
```

```
                  Proposition Name :        Rule_Concern_3

IF     Storage_in_June(Ml)     IS                Zone_C
AND    Sim_Result(%_of_Cap)    IS        Less_than_15%
AND    Lake_Dist(%fall/wk)     IS    Less_than_4%/week
THEN      Operating_policy     IS        Consider_Bans    WITH   TV =   0.80

                  Proposition Name :        Rule_Concern_4

IF     Storage_in_June(Ml)     IS                Zone_C
AND    Sim_Result(%_of_Cap)    IS               Failure
AND    Lake_Dist(%fall/wk)     IS    Less_than_4%/week
THEN      Operating_policy     IS        Consider_Bans    WITH   TV =   1.00

                  Proposition Name :           Rule_Need_1

IF     Storage_in_June(Ml)     IS                Zone_B
AND    Sim_Result(%_of_Cap)    IS        Less_than_15%
AND    Lake_Dist(%fall/wk)     IS    More_than_4%/week
THEN      Operating_policy     IS            Need_Bans    WITH   TV =   0.30

                  Proposition Name :           Rule_Need_2

IF     Storage_in_June(Ml)     IS                Zone_B
AND    Sim_Result(%_of_Cap)    IS               Failure
AND    Lake_Dist(%fall/wk)     IS    More_than_4%/week
THEN      Operating_policy     IS            Need_Bans    WITH   TV =   0.50

                  Proposition Name :           Rule_Need_3

IF     Storage_in_June(Ml)     IS                Zone_C
AND    Sim_Result(%_of_Cap)    IS        Less_than_15%
AND    Lake_Dist(%fall/wk)     IS    More_than_4%/week
THEN      Operating_policy     IS            Need_Bans    WITH   TV =   0.80

                  Proposition Name :           Rule_Need_4

IF     Storage_in_June(Ml)     IS                Zone_C
AND    Sim_Result(%_of_Cap)    IS               Failure
AND    Lake_Dist(%fall/wk)     IS    More_than_4%/week
THEN      Operating_policy     IS            Need_Bans    WITH   TV =   1.00
```