5

LINGUISTIC DESCRIPTIONS AND THEIR ANALYTICAL FORMS

5.1 FUZZY LINGUISTIC DESCRIPTIONS

Fuzzy linguistic descriptions (often called *fuzzy systems* or simply *linguistic descriptions*) are formal representations of systems made through fuzzy *if/then* rules. They offer an alternative and often complementary language to conventional (analytic) approaches to modeling systems (involving differential or difference equations). Informal linguistic descriptions used by humans in daily life as well as in the performance of skilled tasks, such as control of industrial facilities, troubleshooting, aircraft landing, and so on, are usually the starting point for the development of fuzzy linguistic descriptions. Although fuzzy linguistic descriptions are formulated in a human-like language, they have rigorous mathematical foundations involving fuzzy sets and relations (Zadeh, 1988). They encode knowledge about a system in statements of the form

if (a set of conditions are satisfied)

then (a set of consequences can be inferred)

For example, in process control the desirable behavior of a system may be formulated as a collection of rules combined by the connective *ELSE* such as

if	*error* is *ZERO*	*AND*	Δ*error* is *ZERO*	*then* Δu is *ZERO*	*ELSE*
if	*error* is *PS*	*AND*	Δ*error* is *ZERO*	*then* Δu is *NS*	*ELSE*
...					
if	*error* is *SMALL*	*AND*	Δ*error* is *NS*	*then* Δu is *BIG*	

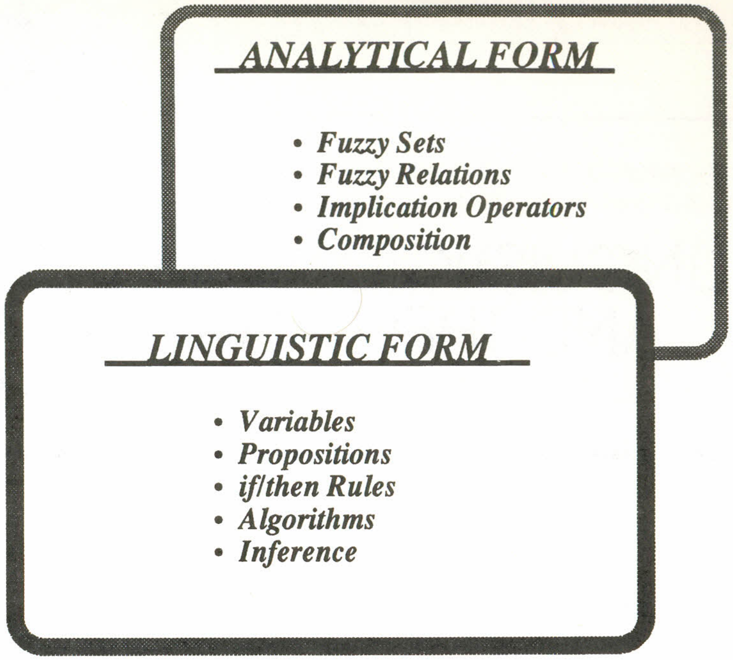


Figure 5.1 Fuzzy linguistic descriptions possess a linguistic form as well as a background analytical form involving fuzzy set operations.

where *error* and Δ*error* (change in error) are linguistic variables describing the input to a controller and Δu is a linguistic variable describing the change in output. A *linguistic variable* is a variable whose arguments are fuzzy numbers (and more generally words modeled by fuzzy sets), which we refer to as *fuzzy values*. For example, in the rules above the *fuzzy values* of the linguistic variable *error* are *ZERO*, *PS* (*positive small*), and *SMALL*, the values of Δ*error* are *ZERO* and *NS* (*negative small*) and the values of Δu are *ZERO*, *NS*, and *BIG*.[1] A specific evaluation of a fuzzy variable—for example, "*error is ZERO*"—is called *fuzzy proposition*. Individual fuzzy propositions on either *left-* (*LHS*) or *right-hand side* (*RHS*) of a rule maybe connected by connectives such as *AND* and *OR*— for example, "*error* is *PS AND* Δ*error* is *ZERO*."[2] Individual *if/then* rules are connected with the connective *ELSE* to form a *fuzzy algorithm*. Propositions and *if/then* rules in classical logic are supposed to be either true or false. In fuzzy logic they can be true or false to a degree.

[1]The convention we follow is to use lowercase italics for linguistic variables and capital italics for fuzzy values, unless otherwise specified or implied by the context.
[2]These are also called *antecedent* (*LHS*) and *consequent* (*RHS*) propositions. We find alternative designations for the *LHS* and *RHS* of a rule in different application areas. In process control, for example, the *if part* is often referred to as the *situation side* and the *then part* is often referred to as the *action* side.

Figure 5.1 shows schematically what is involved in linguistic descriptions. In the front end we find *linguistic forms* representing a system in a human-like manner. In the background we have rigorously defined *analytical forms* involving fuzzy set operations, relations, and composition procedures such as the ones we saw in Chapters 2 and 3.

Despite the difference in appearance, linguistic and conventional (analytic) descriptions are in fact equivalent to each other. Both can be used to describe the same system. However, the computational costs incurred using one or the other may be significantly different. Consider, for example, a function $y = f(x)$ shown in Figure 5.2, describing analytically a specific relation between x's and y's.[3] The same relation may be described by listing all possible, or at least a sufficiently large number of, (x, y) pairs or *points* of $f(x)$, indicating (for example), that when $x = a_1$ the value of the function is $y = b_1$, when $x = a_2$ the value of the function is $y = b_2$, when $x = a_i$ the value of the function is $y = b_i$, and so on. Knowing n such points we may alternatively represent $y = f(x)$ by listing the pairs

$$\begin{array}{c} (a_1, b_1) \\ (a_2, b_2) \\ \cdots \\ (a_i, b_i) \\ \cdots \\ (a_n, b_n) \end{array} \tag{5.1-1}$$

Of course this representation is an acceptable approximation of the analytic representation only when n becomes sufficiently large, with the precision of the approximation being controlled by choosing an appropriate n. A point (a_i, b_i) can also be thought of as a crisp *if/then* rule of the form, "*if x is* a_i *then y is* b_i." Obviously, the pairs of (5.1-1) may be expressed linguistically as crisp rules:

$$\begin{array}{llll} \textit{if} & x \text{ is } a_1 & \textit{then} & y \text{ is } b_1 \\ \textit{if} & x \text{ is } a_2 & \textit{then} & y \text{ is } b_2 \\ \cdots & & & \\ \textit{if} & x \text{ is } a_i & \textit{then} & y \text{ is } b_i \\ \cdots & & & \\ \textit{if} & x \text{ is } a_n & \textit{then} & y \text{ is } b_n \end{array} \tag{5.1-2}$$

[3]As we saw in Chapter 3, *functions* are a particular kind of *relation* allowing one and only one value of y for each x. This is also referred to as a *many-to-one mapping*.

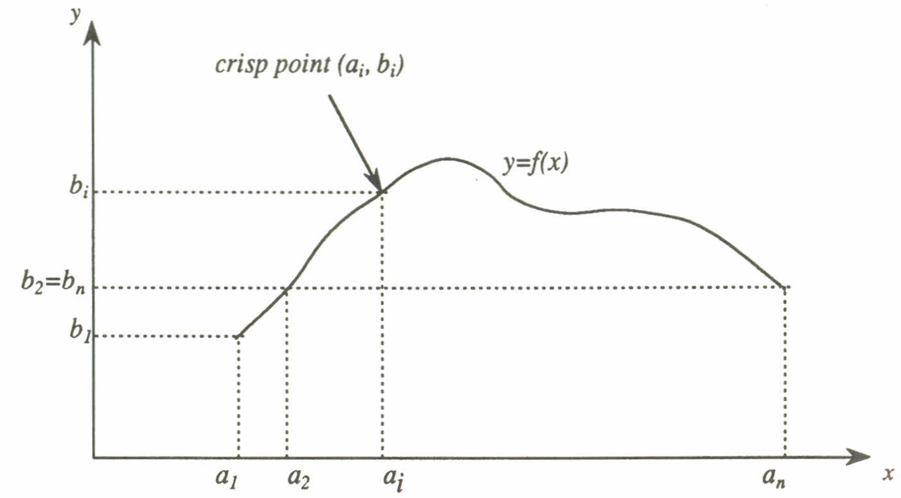


Figure 5.2 The function $y = f(x)$ may be thought of as a collection of crisp points (a_i, b_i), and each point may also be articulated as a crisp *if / then* rule.

Every representation has a cost. We can think of it as related to the number of symbols used and the complexity of operations involved, but actually it involves much more—for example, the cost of extracting the knowledge used, its realization in a machine, the cost of updating and maintaining it, and so on. When we use several crisp rules to represent $y = f(x)$ in the manner of (5.1-2), we are obviously using a more costly representation in a computational sense. By comparison, the analytical description $y = f(x)$ offers a more economical way of describing the function. In this sense the analytical description $y = f(x)$ is said to be a more *parsimonious* description than (5.1-2), in reference to the reduced cost of representation.

Intuitively we expect the crisp linguistic rendition of $y = f(x)$ to become more accurate with increasing number of rules. Having 1000 crisp rules for $f(x)$ is preferable to, say, 10 rules. However, the number of crisp *if/then* rules needed to describe a function such as the one shown in Figure 5.2 actually depends on the specific nature of $f(x)$ as well as our tolerance for approximation error. Take, for instance, a linear function, a straight line going through the origin. In this case, one crisp *if/then* rule may suffice since an additional point on the x–y plane outside the origin uniquely identifies a straight line. On the other hand, a very "noisy" function with many "spikes" and slope changes will require considerably more rules. In practical terms, however, an approximate description of $y = f(x)$ may be acceptable, sometimes even preferable. We are often interested in associations such as *if* x is "*about* a_i" *then* y is "*about* b_i"; that is to say, we are interested not in a crisp point of $f(x)$ but in an area or neighborhood around a point. This is illustrated in Figure 5.3, where instead of crisp point (a_i, b_i) we consider the

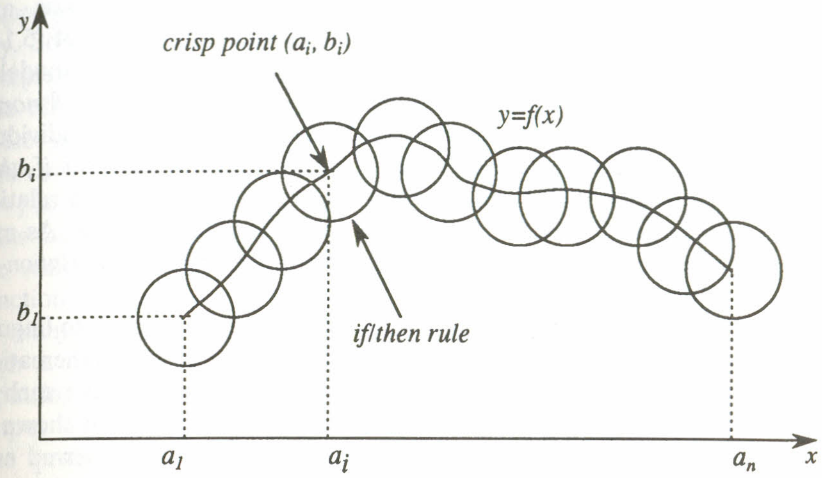


Figure 5.3 Building a linguistic description of the function $y = f(x)$.

area obtained from a point. Such an *area-cum-point* may be described by a fuzzy *if/then* rule. Let us consider "*about* a_i" to be a fuzzy number A_i on the universe of discourse of the x's and consider "*about* b_i" to be a fuzzy number B_i on the universe of discourse of the y's. As we will see later on (Section 5.2), we can define a linguistic variable x whose arguments are fuzzy numbers on the x-axis, such as A_i, and a linguistic variable y whose arguments are fuzzy numbers on the y-axis, such as B_i. Hence the *area-cum-point* "about (a_i, b_i)" can be described by a fuzzy *if/then* rule of the form

$$\textit{if} \quad x \text{ is } A_i \quad \textit{then} \quad y \text{ is } B_i \tag{5.1-3}$$

The analytical form of rule (5.1-3) is a fuzzy relation $R_i(x, y)$ called the *implication relation* of the rule. How we obtain this implication relation is a rather complicated issue which we will examine in more detail in Section 5.3. For the moment we assume that each fuzzy *if/then* rule has an *implication relation*.

The function $y = f(x)$ may be approximated by collecting several fuzzy *if/then* rules—for example,

$$\begin{array}{lllll}
\textit{if} & x \text{ is } A_1 & \textit{then} & y \text{ is } B_1 & \textit{ELSE} \\
\textit{if} & x \text{ is } A_2 & \textit{then} & y \text{ is } B_2 & \textit{ELSE} \\
\ldots & & & & \\
\textit{if} & x \text{ is } A_i & \textit{then} & y \text{ is } B_i & \textit{ELSE} \\
\ldots & & & & \\
\textit{if} & x \text{ is } A_n & \textit{then} & y \text{ is } B_n &
\end{array} \tag{5.1-4}$$

where $A_1, A_2, \ldots, A_i, \ldots, A_n$ are fuzzy numbers on the x axis and $B_1, B_2, \ldots, B_i, \ldots, B_n$ are fuzzy numbers on the y axis. The rules of (5.1-4) are combined by the connective *ELSE*, which could be analytically modeled as either *intersection* or *union* [and more generally as *T norms* or *S norms* (see Appendix A)] depending on the *implication relation* of the individual rules (we will have more on *ELSE* in Section 5.5). The collection of *if/then* rules in (5.1-4) is called a *fuzzy algorithm*, and its analytical form is a relation $R_\alpha(x, y)$ between the x's and the y's, called the *algorithmic relation*. As may be expected, the *algorithmic relation* depends on the *implication relation* of constituent rules.

The transition from conventional descriptions, such as $y = f(x)$, to linguistic descriptions addresses the fact that functions are often mathematical idealizations. In most real-world problems, we do not have a curve such as the one shown in Figure 5.2 but rather something like the region shown in Figure 5.4. For example, suppose that the function $y = f(x)$ is viewed as a control policy—that is, a prescription recommending a control action y—for each state x. In many applications, the control system changes with time (*time-varying*) and in general manifests nonlinear and complex behaviors. Hence, the control policy may actually be a more general relation $R_\alpha(x, y)$ as shown in Figure 5.4. Figure 5.2 could in fact be an idealization of the real-world control policy shown in Figure 5.4. We recall (see also Chapter 3) that a function is a special kind of relation that associates a unique y with each x. A function performs what is called a *many-to-one mapping*; that is, several values of x may have the same value of y but not vice versa. Most real-world applications, however, involve *many-to-many mappings*. Situations like the one shown in Figure 5.4—that is, relations that are *many-to-many*

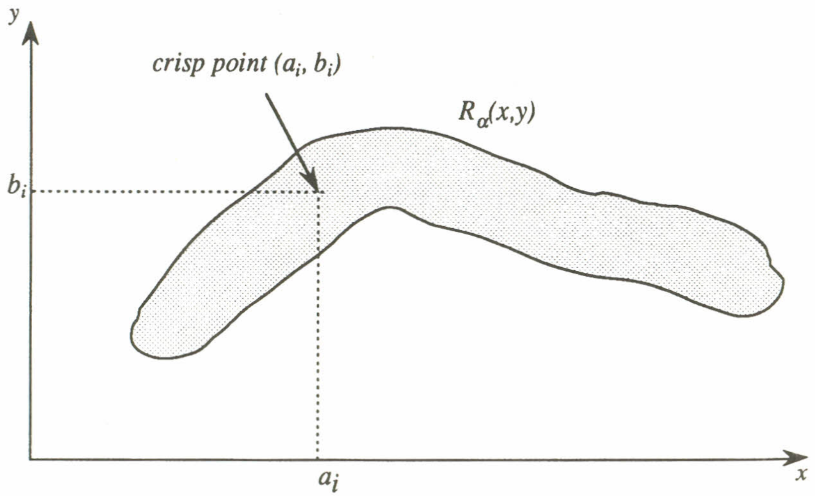


Figure 5.4 Often a real-world "function" may actually be a more general relation.

mappings—are far more common in complex engineering systems than usually. Sometimes conventional descriptions, being overly idealized models of complex systems, may suffer from lack of robustness and exhibit undesirable side effects.

Let us look again in Figure 5.3. We note that the transition from *points* to *area-cum-points* reduces the number of *if/then* rules needed to describe $y = f(x)$. For example, we could approximate $f(x)$ with only 11 fuzzy *if/then* rules (circled areas) as shown in Figure 5.3. The rules are overlapping as are the various fuzzy numbers on the x and y axes. Yet, we no longer have a function (a *many-to-one mapping*) but a more general relation $R_\alpha(x, y)$ (a *many-to-many mapping*), and the obvious question is: How do we use such a relation? In conventional descriptions we evaluate functions by inputting a crisp value of x to $f(x)$ and obtain a unique crisp value of y as output. Something similar can be done with linguistic descriptions as well. The process of evaluating a fuzzy linguistic description is called *fuzzy inference*. There are two important problems in fuzzy inference. First, given a fuzzy number A' as input to a linguistic description, we want to obtain a fuzzy number B' as its output; and, second, given B', we want to obtain A' (the inverse problem). The first problem is addressed with an inferencing procedure called *generalized modus ponens* (GMP), and the second is addressed with another inferencing procedure called *generalized modus tollens* (GMT). Both GMP and GMT have their origin in the field of logic and approximate reasoning (Section 5.4), and analytically they involve composition of fuzzy relations (Chapter 3).

In GMP, when an *if/then* rule and its antecedent are approximately matched, a consequent may be inferred. For simplicity let us consider only a generic rule of (5.1-4) having an implication relation $R(x, y)$. GMP is formally stated as

$$\begin{array}{l} \textit{if} \quad x \text{ is } A \quad \textit{then} \quad y \text{ is } B \\ \qquad x \text{ is } A' \\ \hline \qquad y \text{ is } B' \end{array} \tag{5.1-5}$$

where A' is an input value matching the antecedent A to a degree (including totally perfect and totally imperfect match). The implication relation of the rule $R(x, y)$ and the input A' above the line are considered known, whereas what is below the line—in other words B'—is considered unknown. B' is what we want to find. Analytically, GMP (5.1-5) is performed by composing A' with the implication relation $R(x, y)$ as in the max–min composition (see Chapter 3)

$$B' = A' \circ R(x, y) \tag{5.1-6}$$

We will see how this is done in detail in Section 5.4. For the moment let us simply keep in mind that we can evaluate linguistic descriptions just as we can evaluate functions and that the procedure of evaluation involves composition of fuzzy relations. GMP is related to forward-chaining or data-driven inference and is the main inferencing procedure in fuzzy control. When $A' = A$ and $B' = B$, GMP (5.1-5) reduces to an inferencing procedure of classical logic known as *modus ponens* (depending on the implication relation).

In GMT a rule and its consequent are approximately matched and from that we can obtain an antecedent. GMT is formally stated as

$$\begin{array}{l} \textit{if} \quad x \text{ is } A \quad \textit{then} \;\; y \text{ is } B \\ \qquad\qquad\qquad\qquad y \text{ is } B' \\ \hline x \text{ is } A' \end{array} \tag{5.1-7}$$

Again, everything above the line is known and we want to find out what is below the line—that is, A'. The analytical problem involved in GMT is addressed by composing the implication relation $R(x, y)$ with fuzzy number B' as

$$A' = R(x, y) \circ B' \tag{5.1-8}$$

GMT is closely related to backward-chaining or goal-driven inference, which is the main form of inference used in diagnostic expert systems. When $A' = NOT\ A$ and $B' = NOT\ B$, GMT reduces to classical *modus tollens* (depending on the implication relation used).

In general, fuzzy linguistic descriptions offer convenient tools for controlling the *granularity* of a description,[4] in the sense that they facilitate the choice of appropriate precision levels—that is, levels that application-specific considerations call for. In terms of our example, when we use fuzzy numbers and fuzzy *if/then* rules to describe $y = f(x)$, we have at our disposal a mechanism for reducing the number of rules needed and, hence, for controlling the *granularity* of this particular description and the overall cost of computation (Zadeh, 1979). In addition, the technology for computing with *if/then* rules has already advanced to the point where fuzzy microprocessors, called *fuzzy chips*, are widely available (Yamakawa, 1987; Isik, 1988; Hirota and Ozawa, 1988; Huertas et al., 1992; Shimizu et al., 1992). Fuzzy chips encoding knowledge in the form of linguistic descriptions can function as "mounted devices"—that is, dedicated processors fine-tuned to the specifics of a component and its environment, performing domain-specific computations. Such processors are already deployed in several control and robotics applications with remarkable successes (Yamakawa, 1988; Pin et al., 1992). Of course, software is a commonly used medium for the implementation of fuzzy algorithms on a variety of different computers. However, the advent of fuzzy logic hardware and the development of fuzzy computers may have a

[4] By *granularity* we roughly mean the coarseness of a description, the level of precision necessary to effectively represent a given system.

profound impact on the design and operation of engineering systems (Yamakawa, 1988). Fuzzy linguistic descriptions are of growing importance in many areas of engineering ranging from expert systems and artificial intelligence applications to process control, pattern recognition, signal analysis, reliability engineering, and machine learning (Ray and Majumder, 1988). The basic ideas, however, are rather similar and rest on the mathematics of fuzzy sets. Describing a system through a linguistic description, no matter for what purpose, involves specifying in some way *linguistic variables*, *if/then rules*, and evaluation procedures known as *fuzzy inference*.

5.2 LINGUISTIC VARIABLES AND VALUES

As we saw in the previous section, a linguistic variable is a variable whose arguments are fuzzy numbers and more generally words represented by fuzzy sets. For example, the arguments of the linguistic variable *temperature* may be *LOW*, *MEDIUM*, and *HIGH*. We call such arguments *fuzzy values*. Each and every one of them is modeled by its own membership function. The fuzzy values *LOW*, *MEDIUM*, and *HIGH* may be modeled as shown in Figure 5.5 or Figure 5.6. In Figure 5.5 we have three discrete fuzzy values, while in Figure 5.6 we have three (piecewise) continuous membership functions—$\mu_{LOW}(T)$, $\mu_{MEDIUM}(T)$, and $\mu_{HIGH}(T)$—modeling the words *LOW*,

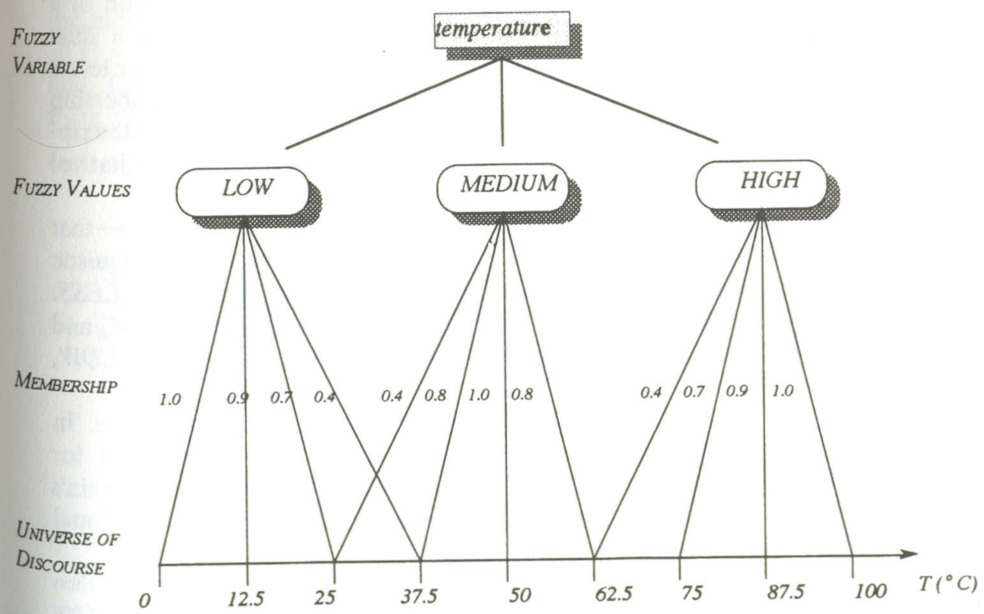


Figure 5.5 The linguistic variable *temperature* and a set of discrete fuzzy values.

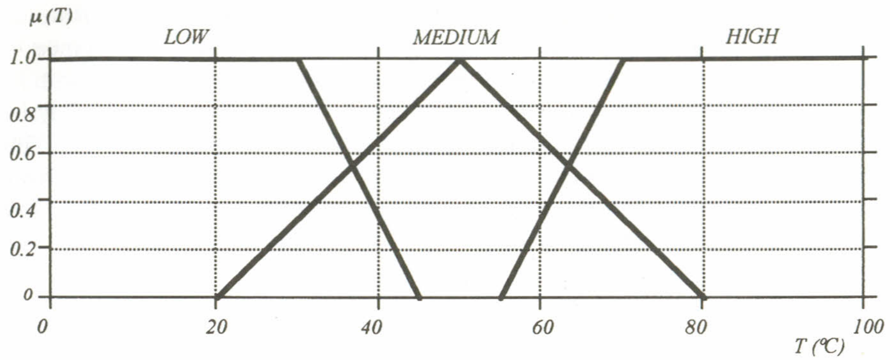


Figure 5.6 Membership functions $\mu(T)$ used for describing the primary values, *LOW, MEDIUM*, and *HIGH*, of the linguistic variable *temperature*.

MEDIUM, and *HIGH*, respectively. Any crisp value of temperature e.g., 60°C) has a unique degree of membership to each fuzzy value of *temperature*. In Figure 5.6, for example, crisp temperature 60°C is *LOW* to a degree zero, *MEDIUM* to a degree 0.65, and *HIGH* to a degree 0.35.

We distinguish four different levels in the definition of a linguistic variable as shown in Figure 5.5. At the top level we have the name of the variable (e.g., *temperature*). At the level below it we have the labels of *fuzzy values* (starting with an initial set of values called *primary values* or *term set*[5]). Further down we have *membership functions*, and at the bottom we have the *universe of discourse*. All four levels are indispensable in the definition of a variable. It is important to observe that linguistic variables have a dual nature; at higher levels we have a symbolic linguistic form, and at lower levels we have a well-defined quantitative analytical form—that is, the membership function. This double identity is a general feature of fuzzy linguistic descriptions rendering them convenient for performing both symbolic (qualitative) and numerical (quantitative) computations (Zadeh, 1975).

Generally, the values of a linguistic variable may be *compound values*—that is, values constructed through the use of *primary values* and linguistic modifiers such as *NOT*, *VERY*, *RATHER*, *ALMOST*, and *MORE OR LESS*. For example, out of the initial set of primary values *LOW*, *MEDIUM*, and *HIGH* for *temperature*, compound values such as *NOT LOW*, *VERY LOW*, *RATHER MEDIUM*, and *ALMOST HIGH* may be formed.

Fuzzy values are essentially *aggregations* or *categories* of crisp values. In fuzzy logic the flexibility of adjusting membership functions is useful for categorizing the parameters of a domain in accordance with the domain's own unique features. If temperature is considered in a conventional sense—that is, as a numerical variable—its arguments are simply the crisp numbers of a universe of discourse (e.g., natural numbers between 1°C and 100°C). We may think of each number as a crisp category of temperature; in this case we could have 100 different categories. For certain applications this may be an acceptable categorization of the values of temperature. For others we may need 1000 categories, and still for others 3 categories may suffice. Fuzzy values provide this kind of flexibility. They allow for adjustable categories and explicitly acknowledge the ambiguous and application-dependent nature of this or the other categorization.

[5]The *values* of a linguistic variable are referred to by a variety of names in the literature. Often they are called *fuzzy variables*, *primary terms*, *set of prototypes*, or the *term set*. We will mostly use the name *fuzzy value* or simply *value*. Linguistic variables are also called *fuzzy variables*.

Primary Values

The words which function as the initial values of a linguistic variable are called *primary values*. They are the principal categorization of a universe of discourse—for example, the values *LOW*, *MEDIUM*, and *HIGH* shown in Figures 5.5 and 5.6. To model them we often use functions whose shape is adjusted through a finite set of parameters. For example, the function

$$\mu(x) = \frac{1}{1 + a(x - c)^b} \tag{5.2-1}$$

has parameters a, b, and c which may be used to adjust the overall form of $\mu(x)$. Parameter a adjusts the width of the membership function, b determines the extent of fuzziness, and c describes the location of the "peak" of the membership function. This is the point in the universe of discourse where $\mu(x) = 1$. Consider the primary values of *temperature*, *SMALL*, *MEDIUM*, and *LARGE* shown in Figure 5.7. Their membership functions are of the form of Equation (5.2-1) with $a = 0.0005$, $b = 3$, and $c = 20$, 50, and 80,

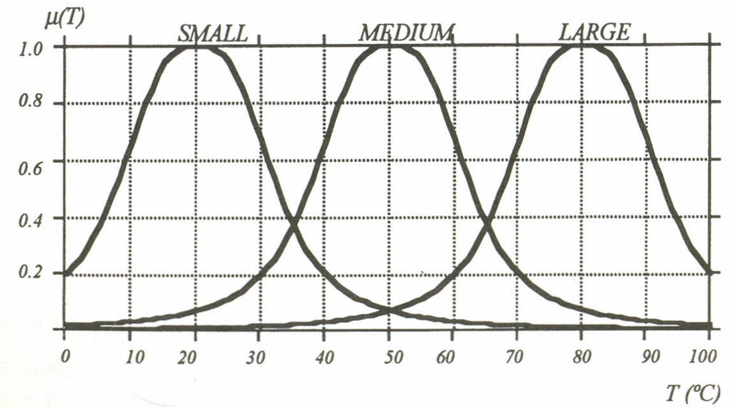


Figure 5.7 Adjustable membership functions for modeling primary values.

respectively—that is,

$$\mu_{SMALL}(T) = \frac{1}{1 + 0.0005(|T - 20|)^3}$$

$$\mu_{MEDIUM}(T) = \frac{1}{1 + 0.0005(|T - 50|)^3} \qquad (5.2\text{-}2)$$

$$\mu_{LARGE}(T) = \frac{1}{1 + 0.0005(|T - 80|)^3}$$

In many control applications, continuous membership functions such as the trapezoidal/triangular functions of Figure 5.6 are used. Fuzzy values defined through trapezoidal/triangular membership functions have adjustable parameters as well, namely the "corners" of the function—that is, the points where the monotonicity changes. We recall their use in Chapter 4 in connection with fuzzy numbers. In fuzzy arithmetic, however, we required that fuzzy sets be normalized—that is, that there be at least one point of the universe of discourse where the membership function reaches unity, whereas in fuzzy linguistic descriptions this requirement is relaxed. Fuzzy values ought to be convex, just as fuzzy numbers, but not necessarily normal.

Primary values can also be modeled through *S-shaped* and *Π-shaped* functions named by their general form (Zimmermann, 1985; Kandel, 1986). S-shaped and Π-shaped membership functions may be adjusted to suit various application needs merely by altering a limited number of parameters as in the case of trapezoidal and triangular membership functions. S-shaped functions are defined through three parameters α, β, and γ as follows:

$$
\begin{aligned}
S(x;\alpha,\beta,\gamma) &= 0 && \text{for } x \le \alpha \\
S(x;\alpha,\beta,\gamma) &= 2\left(\frac{x-\alpha}{\gamma-\alpha}\right)^2 && \text{for } \alpha \le x \le \beta \\
S(x;\alpha,\beta,\gamma) &= 1 - 2\left(\frac{x-\gamma}{\gamma-\alpha}\right)^2 && \text{for } \beta \le x \le \gamma \\
S(x;\alpha,\beta,\gamma) &= 1 && \text{for } x \ge \gamma
\end{aligned}
\qquad (5.2\text{-}3)
$$

where x is any real number and α, β, and γ are appropriately chosen parameters. For continuity of slope at $x = \beta$, the two intervals $(\beta - \alpha)$ and $(\gamma - \beta)$ must be equal.

A Π-shaped function may be thought of as two S-shaped functions put together "back-to-back" and can be expressed as

$$
\begin{aligned}
\Pi(x;\delta,\gamma) &= S\left(x;\gamma-\delta,\frac{\gamma-\delta}{2},\gamma\right) && \text{for } x \le \gamma \\
\Pi(x;\delta,\gamma) &= 1 - S\left(x;\gamma,\frac{\gamma+\delta}{2},\gamma+\delta\right) && \text{for } x \ge \gamma
\end{aligned}
\qquad (5.2\text{-}4)
$$

The parameter δ in Π-shaped functions is called the *bandwidth.* It is the distance between the crossover (inflection) points—that is, the points where the function equals 0.5. The parameter γ is the point where the Π-shaped function reaches unity. Fuzzy values modeled by S-shaped and Π-shaped functions are more often encountered in software than in hardware realizations of fuzzy linguistic descriptions. Triangular/trapezoidal membership functions are the preferred shapes for fuzzy values used in hardware realizations.

Compound Values

Using the connectives *AND* and *OR* and a collection of linguistic modifiers such as *NOT*, *VERY*, *MORE OR LESS*, *RATHER*, and so on, we can generate compound values from primary values. Modifiers and connectives are modeled by fuzzy set operations as well. For example, *AND* and *OR* are modeled by the fuzzy set operations of *intersection* and *union*, respectively, while *NOT* is modeled by *complementation.* More generally they are modeled by T and S norms (see Appendix A). Through linguistic modifiers we may easily construct a larger, potentially infinite set of values from a relatively small and finite set of primary values. Some modifiers are also called *linguistic hedges* due to the property of semantically constraining (*hedging*) the general meaning of a word by operating on the fuzzy set that represents it (Zadeh, 1983).

The connective *OR* generates a compound value with membership function equal to the max ($\vee$) of the membership functions of other values. Consider the values A and B defined over the same universe of discourse X as

$$A = \int_X \mu_A(x)/x, \qquad B = \int_X \mu_B(x)/x$$

The compound value "A *OR* B" is defined as

$$A\ OR\ B \equiv \int_X [\mu_A(x) \vee \mu_B(x)]/x \qquad (5.2\text{-}5)$$

The connective *AND* uses the min operator ($\wedge$) to generate the membership function of the compound value out of the membership functions of two (or more) other values. The compound value constructed through the connective *AND* is defined as

$$A \; AND \; B \equiv \int_X [\mu_A(x) \wedge \mu_B(x)]/x \tag{5.2-6}$$

The *AND* connective has to be used with caution when generating compound values because it may lead to nonsensical words such as in the proposition "*temperature* is (*HIGH AND LOW*)." As shown in Figure 5.8a, this compound value has zero membership function and may be thought of as meaningless. The connective *AND* can produce correct compound values when used with the complement of primary values as, for example, in the proposition "*temperature* is ((*NOT LOW*) *AND* (*NOT HIGH*))," whose membership function can be seen in Figure 5.8b.

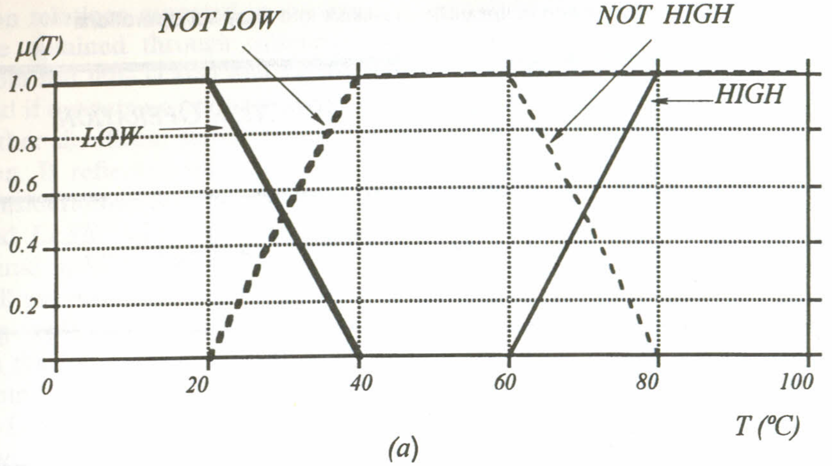
The membership function of a compound value produced by negating another value is the complement of the membership function of the original value—that is,

$$NOT \; A \equiv \int_X [1 - \mu_A(x)]/x \tag{5.2-7}$$

The semantics of the modifier *NOT* are fairly straightforward, and it may be used very much as negation is used in natural language—for example, "*temperature* is (*NOT HIGH*)."

Every linguistic modifier is associated with a corresponding fuzzy set operation involving membership functions. Table 5.1 lists some of these associations. The *PLUS* and *MINUS* modifiers in Table 5.1 offer a smaller degree of concentration and dilation than do the concentration *CON* and dilation *DIL* operations which we saw in Chapter 2. Modifiers may be connected in series in order to form larger compound values. Suppose, for example, that we start with the primary values *SMALL* and *LARGE*. We form compound values such as (*VERY SMALL*) and (*NOT VERY SMALL*) by logically multiplying *SMALL* by *VERY* and *NOT*. We can go on in this manner obtaining more compound values—for example, C = ((*NOT VERY SMALL*) *AND* (*NOT LARGE*)). Using the operations in Table 5.1 we model C by the following membership function:

$$\mu_C(x) = [1 - \mu^2_{SMALL}(x)] \wedge [1 - \mu_{LARGE}(x)] \tag{5.2-8}$$



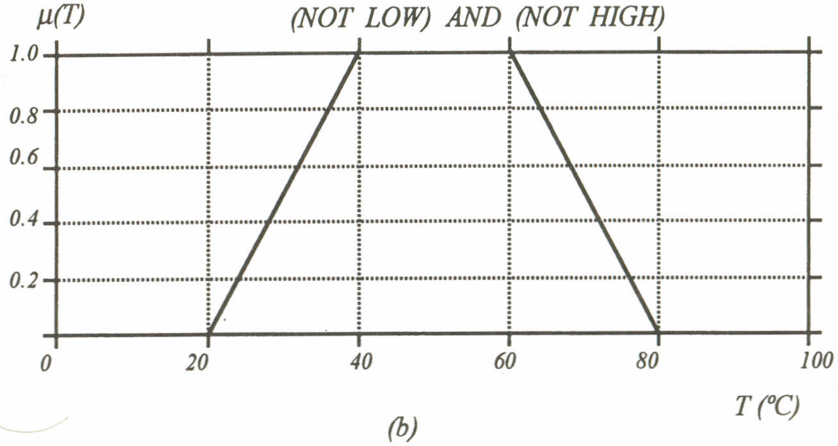


Figure 5.8 The semantics of compound terms generated by *AND* ought to be carefully examined. In (*a*) the compound term *LOW AND HIGH* has trivial membership function, while in (*b*) the compound term (*NOT LOW*) *AND* (*NOT HIGH*) is well defined.

It should be noted that compound fuzzy values may not be arbitrarily generated. We need to examine their semantics—that is, their meaning in the context of a specific application. An interesting quantitative guide to the semantics of compound values is provided by their membership function. When the new membership function becomes uniformly 1 or 0 we may have a semantically suspect compound value.

Table 5.1 Translation of linguistic modifiers into fuzzy set operations

MODIFIER	*MEMBERSHIP FUNCTION OPERATION*
VERY A	$\mu_{CON(A)}(x) \equiv [\mu_A(x)]^2$
MORE OR LESS A	$\mu_{DIL(A)}(x) \equiv [\mu_A(x)]^{1/2}$
INDEED A	$\mu_{INT(A)}(x)$ [see Equation (2.3–21)]
PLUS A	$[\mu_A(x)]^{1.25}$
MINUS A	$[\mu_A(x)]^{0.75}$
OVER A	$1-\mu_A(x), \quad x \geq x_{max}$ $0, \quad x < x_{max}$
UNDER A	$1-\mu_A(x), \quad x \leq x_{min}$ $0, \quad x > x_{min}$

5.3 IMPLICATION RELATIONS

Fuzzy *if/then* rules are conditional statements that describe the dependence of one (or more) linguistic variable on another. As we already alluded to earlier, the underlying analytical form of an *if/then* rule is a fuzzy relation called the *implication relation*. There are over 40 different forms of implication relations reported in the literature (Lee, 1990a, b). Implication relations are obtained through different *fuzzy implication operators* ϕ. Information from the left- (LHS) and right-hand side (RHS) of a rule is inputted to ϕ, and it outputs an implication relation. The choice of implication operator is a rather significant step in the overall development of a fuzzy linguistic description. It reflects application-specific criteria, as well as logical and intuitive considerations focusing on the interpretation of the connectives *AND*, *OR*, and *ELSE*. An extensive discussion of different implication relations may be found in Mizumoto (1988), Lee (1990a, b), and Ruan and Kerre (1993).[6] We will examine here the most common implication operators used in engineering applications, particularly in fuzzy control (Chapter 6). Our focus will be on the implication relation of a simple *if/then* rule and on how to obtain it from LHS and RHS membership functions.

Let us consider a generic *if/then* rule involving two linguistic variables, one on each side of the rule—for example,

$$\textit{if} \quad x \text{ is } A \quad \textit{then} \quad y \text{ is } B \tag{5.3-1}$$

where linguistic variables x and y take the values A and B, respectively. The underlying analytical form of rule (5.3-1) is the *implication relation*

$$R(x, y) = \int_{(x,y)} \mu(x, y)/(x, y) \tag{5.3-2}$$

where $\mu(x, y)$ is the membership function of the implication relation, the thing we want to obtain. When the linguistic variables in (5.3-1) are defined over discrete universes of discourse, an implication relation is written as

$$R(x_i, y_j) = \sum_{(x_i, y_j)} \mu(x_i, y_j)/(x_i, y_j) \tag{5.3-3}$$

There are several options for obtaining the membership function of the implication relation. We explore them through the implication operator notion. For the rule of (5.3-1) an implication operator ϕ takes as input the membership functions of the antecedent and consequent parts, namely, $\mu_A(x)$ and $\mu_B(y)$, and takes as outputs $\mu(x, y)$, namely

$$\mu(x, y) = \phi[\mu_A(x), \mu_B(y)] \tag{5.3-4}$$

[6]Implication operators can also be expressed through T and S norms (see Appendix). It should be noted that the term "implication" is somewhat of a misnomer (since strictly speaking there is no logical implication in a rule); nonetheless it is widely used in the literature.

We distinguish the following implication operators:

Zadeh Max–Min Implication Operator

The *Zadeh max–min* implication operator (Zadeh, 1973) is

$$\phi_m[\mu_A(x), \mu_B(y)] \equiv (\mu_A(x) \wedge \mu_B(y)) \vee (1 - \mu_A(x)) \quad (5.3\text{-}5)$$

Thus the membership function of the implication relation (5.3-2) is

$$\mu(x, y) = (\mu_A(x) \wedge \mu_B(y)) \vee (1 - \mu_A(x))$$

Mamdani Min Implication Operator

The *Mamdani min* implication operator is a simplified version of Zadeh max–min proposed by Mamdani in the 1970s in connection with fuzzy control (Mamdani, 1977) and is defined as

$$\phi_c[\mu_A(x), \mu_B(y)] \equiv \mu_A(x) \wedge \mu_B(y) \quad (5.3\text{-}6)$$

Larsen Product Implication Operator

The *Larsen product* implication operator uses arithmetic product (Larsen, 1980) and is defined as

$$\phi_p[\mu_A(x), \mu_B(y)] \equiv \mu_A(x) \cdot \mu_B(y) \quad (5.3\text{-}7)$$

Arithmetic Implication Operator

The *arithmetic* implication operator is based in multivalued logic (Zadeh, 1975) and is defined as

$$\phi_a[\mu_A(x), \mu_B(y)] \equiv 1 \wedge (1 - \mu_A(x) + \mu_B(y)) \quad (5.3\text{-}8)$$

Boolean Implication Operator

The *Boolean* implication operator is based on classical logic and has been used in control and decision-making applications. It is defined as

$$\phi_b[\mu_A(x), \mu_B(y)] \equiv (1 - \mu_A(x)) \vee \mu_B(y) \quad (5.3\text{-}9)$$

The Bounded Product Implication Operator

The *bounded product* fuzzy implication operator has been used in fuzzy control and is defined as

$$\phi_{bp}[\mu_A(x), \mu_B(y)] \equiv 0 \vee (\mu_A(x) + \mu_B(y) - 1) \quad (5.3\text{-}10)$$

The Drastic Product Implication Operator

The *drastic product* implication operator has also been used in the field of control. As the name implies, it involves a more drastic (crisp) decision as to the form of the implication relation and is defined as

$$\phi_{dp}[\mu_A(x), \mu_B(y)] \equiv \begin{cases} \mu_A(x), & \mu_B(y) = 1 \\ \mu_B(y), & \mu_A(x) = 1 \\ 0, & \mu_A(x) < 1, \mu_B(y) < 1 \end{cases} \quad (5.3\text{-}11)$$

The Standard Sequence Implication Operator

The *standard sequence* implication operator has crisp logic features. It is defined as

$$\phi_s[\mu_A(x), \mu_B(y)] \equiv \begin{cases} 1, & \mu_A(x) \leq \mu_B(y) \\ 0, & \mu_A(x) > \mu_B(y) \end{cases} \quad (5.3\text{-}12)$$

Gougen Implication Operator

The *Gougen* implication operator considers the fuzzy implication relation to be strong, reaching unity, if the membership function of the antecedent $\mu_A(x)$ is smaller than the membership function of the consequent $\mu_B(x)$. Otherwise, the greater $\mu_A(x)$ becomes, relative to $\mu_B(y)$, the more the membership function of the implication relation $\mu(x, y)$ comes to resemble that of the consequent. The Gougen implication relation is in a way a more tempered version of the standard sequence operator. It is formally defined as

$$\phi_\Delta[\mu_A(x), \mu_B(y)] \equiv \begin{cases} 1, & \mu_A(x) \leq \mu_B(y) \\ \dfrac{\mu_B(y)}{\mu_A(x)}, & \mu_A(x) > \mu_B(y) \end{cases} \quad (5.3\text{-}13)$$

Gödelian Implication Operator

The *Gödelian* implication operator is defined as

$$\phi_g[\mu_A(x), \mu_B(y)] \equiv \begin{cases} 1, & \mu_A(x) \leq \mu_B(y) \\ \mu_B(y), & \mu_A(x) > \mu_B(y) \end{cases} \quad (5.3\text{-}14)$$

These fuzzy implication operators are listed in Table 5.2. They are frequently encountered in engineering applications particularly in fuzzy control (Chapter 6). One interesting issue arises in connection with whether or not some of

Table 5.2 Some fuzzy implication operators

NAME	IMPLICATION OPERATOR $\phi[\mu_A(x), \mu_B(y)] =$
ϕ_m, *Zadeh Max-Min*	$(\mu_A(x) \wedge \mu_B(y)) \vee (1-\mu_A(x))$
ϕ_c, *Mamdani min*	$\mu_A(x) \wedge \mu_B(y)$
ϕ_p, *Larsen Product*	$\mu_A(x) \cdot \mu_B(y)$
ϕ_a, *Arithmetic*	$1 \wedge (1-\mu_A(x)+\mu_B(y))$
ϕ_b, *Boolean*	$(1-\mu_A(x)) \vee \mu_B(y)$
ϕ_{bp}, *Bounded Product*	$0 \vee (\mu_A(x)+\mu_B(y)-1)$
ϕ_{dp}, *Drastic Product*	$\mu_A(x),$ *if* $\mu_B(y)=1$ $\mu_B(y),$ *if* $\mu_A(x)=1$ $0,$ *if* $\mu_A(x)<1, \mu_B(y)<1$
ϕ_s, *Standard Sequence*	$1,$ *if* $\mu_A(x) \leq \mu_B(y)$ $0,$ *if* $\mu_A(x) > \mu_B(y)$
ϕ_Δ, *Gougen*	$1,$ *if* $\mu_A(x) \leq \mu_B(y)$ $\frac{\mu_B(y)}{\mu_A(x)},$ *if* $\mu_A(x) > \mu_B(y)$
ϕ_g, *Gödelian*	$1,$ *if* $\mu_A(x) \leq \mu_B(y)$ $\mu_B(y),$ *if* $\mu_A(x) > \mu_B(y)$

TAKAGI – SUGENO

these operators satisfy *classical modus ponens* and *modus tollens*. Another issue has to do with the manner that they satisfy certain intuitive criteria about inferencing such as, for example, the expectation that evaluating "*if x is A then y is B*" by "*x is VERY A*" ought to result in "*y is VERY B*." A good discussion of these issues is found in Mizumoto (1988) and Lee (1990a, b).

5.4 FUZZY INFERENCE AND COMPOSITION

Fuzzy inference refers to computational procedures used for evaluating fuzzy linguistic descriptions. There are two important inferencing procedures: *generalized modus ponens* (GMP) and *generalized modus tollens* (GMT). For simplicity let us consider a linguistic description involving only a simple *if/then* rule with known implication relation $R(x, y)$ and a fuzzy value A' approximately matching the antecedent of the rule. GMP allows us to compute (infer) the consequent B'. It is formally stated as

$$\begin{array}{l} \textit{if} \quad x \text{ is } A \quad \textit{then} \quad y \text{ is } B \\ \quad x \text{ is } A' \\ \hline \quad y \text{ is } B' \end{array} \tag{5.4-1}$$

where everything above the line is analytically known, and what is below is analytically unknown. Suppose, for example, that we have the rule "*if temperature* is *HIGH then humidity* is *ZERO*." Given that "*temperature* is *VERY HIGH*," GMP allows us to evaluate the rule and infer a value for *humidity*. The inferred value B' is computed through the composition of A' with the implication relation $R(x, y)$. Let us look at what is involved analytically in (5.4-1). We know the implication relation $R(x, y)$ of the rule "*if x is A then y is B*" (obtained by using one of the operators shown in Table 5.2) and the membership function of A'. To compute the membership function of B' in (5.4-1), we use *max–min composition* of fuzzy set A' with $R(x, y)$—that is,

$$B' = A' \circ R(x, y) \tag{5.4-2}$$

In terms of membership functions, equation (5.4-2) is (see Chapter 3)

$$\mu_{B'}(y) = \bigvee_x [\mu_{A'}(x) \wedge \mu(x, y)] \tag{5.4-3}$$

where $\mu_{A'}(x)$ is the membership function of A', $\mu(x, y)$ is the membership function of the implication relation, and $\mu_{B'}(y)$ the membership function of B'. We recall from Chapter 3 that max–min composition ($\circ$) is analogous to matrix multiplication with max ($\vee$) and min ($\wedge$) in place of *addition* ($+$) and *multiplication* ($\times$).

In GMT a rule and a fuzzy value approximately matching its consequent are given and it is desired to infer the antecedent—that is,

$$\begin{array}{l} \text{if} \quad x \text{ is } A \quad \text{then } y \text{ is } B \\ \qquad\qquad\qquad\qquad y \text{ is } B' \\ \hline x \text{ is } A' \end{array} \tag{5.4-4}$$

In GMT we know $R(x, y)$ and the consequent B'. To compute the membership function of A' in (5.4-2), we can use max–min composition of $R(x, y)$ with fuzzy set B'—that is,

$$A' = R(x, y) \circ B' \tag{5.4-5}$$

In terms of membership functions, equation (5.4-5) is (see Chapter 3)

$$\mu_{A'}(x) = \bigvee_y [\mu(x, y) \wedge \mu_{B'}(y)] \tag{5.4-6}$$

Of course, other compositions may be used in place of max–min. For example, using max-product composition the membership function of B' in (5.4-2) is given by

$$\mu_{B'}(y) = \bigvee_x [\mu_{A'}(x) \cdot \mu_R(x, y)] \tag{5.4-7}$$

where we take the maximum with respect to x of all the products of the pairs inside the brackets (see Example 5.2). In general max-$*$ composition may be used to infer the membership function of B':

$$\mu_{B'}(y) = \bigvee_x [\mu_{A'}(x) * \mu_R(x, y)] \tag{5.4-8}$$

Using composition of relations to infer consequents—that is, to draw conclusions on the basis of imprecise premises—is known as the *compositional rule of inference*, since logical inferencing such as GMP is performed analytically through composition. As shown in Figure 5.9, GMP works in a manner analogous to *evaluating* a function and GMT is analogous to finding the *inverse* (Pappis and Sugeno, 1985). When a fuzzy value A' is given as input to a linguistic description (single rule or fuzzy algorithm) we can obtain B' through GMP; conversely, if we know B' we can obtain A' through GMT. Generally, we have several overlapping rules, and more than one may contribute a nontrivial B' (or A'). The *union* or *intersection* (depending on the implication operator used as we will see in the next section) of all contributions is the output of the linguistic description for a given A' (or B'). Often the fuzzy values used are not symmetric or of the same form, and

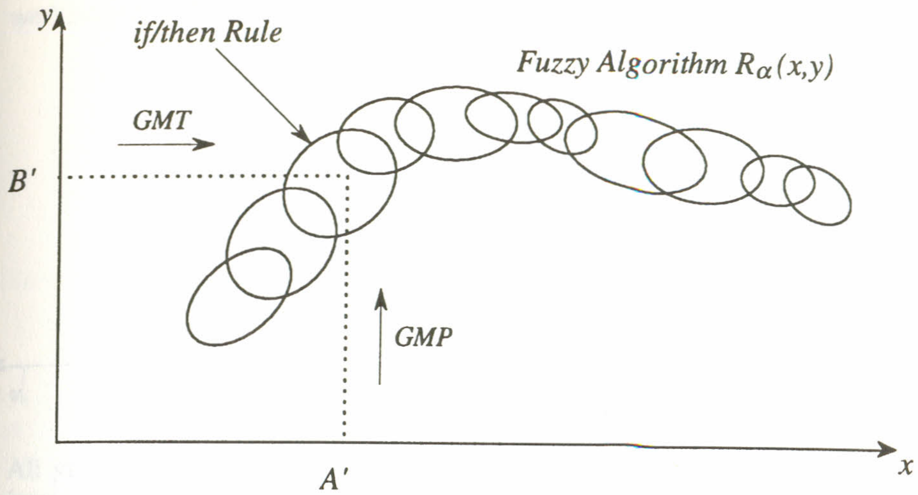


Figure 5.9 GMP and GMT are procedures for evaluating fuzzy linguistic descriptions.

hence we may not have circular *area-cum-points* as in Figure 5.3 but instead have the more general shapes shown in Figure 5.9. Of course in order to use composition we must have available *implication* and *algorithmic relations*.

Logical operations other than GMP or GMT may also be performed analytically through composition—for example, by combining two or more rules in a *syllogism* (Zimmermann, 1985). Consider the following rules:

$$\begin{array}{l} \text{if} \quad x \text{ is } A \quad \text{then } y \text{ is } B \\ \text{if} \quad y \text{ is } B \quad \text{then } z \text{ is } C \end{array} \tag{5.4-9}$$

from which we can infer another rule: "*If* x *is* A *then* z *is* C" through *syllogism*. Each rule in (5.4-9) is analytically described by a fuzzy relation, the first by $R_1(x, y)$ and the second by $R_2(y, z)$. From these relations we may infer a new relation $R_{12}(x, z)$ for the rule "*if* x is A *then* z is C" using max–min composition of $R_1(x, y)$ and $R_2(y, z)$—that is, $R_{12}(x, z) = R_1(x, y) \circ R_2(y, z)$. Again, max–min, max-product, or max-$*$ composition may also be used to obtain $R_{12}(x, z)$.

Example 5.1 GMP and Mamdani Min Implication. In this example we use GMP to evaluate a linguistic description comprised of a single rule "*if* x is A *then* y is B" with LHS and RHS membership functions $\mu_A(x)$ and $\mu_B(y)$, as shown in Figures 5.10a and 5.10b. The implication relation of the rule is modeled through Mamdani min implication operator. Fuzzy number A' (a singleton) shown in Figure 5.10c is the input to the rule. From Figure 5.10

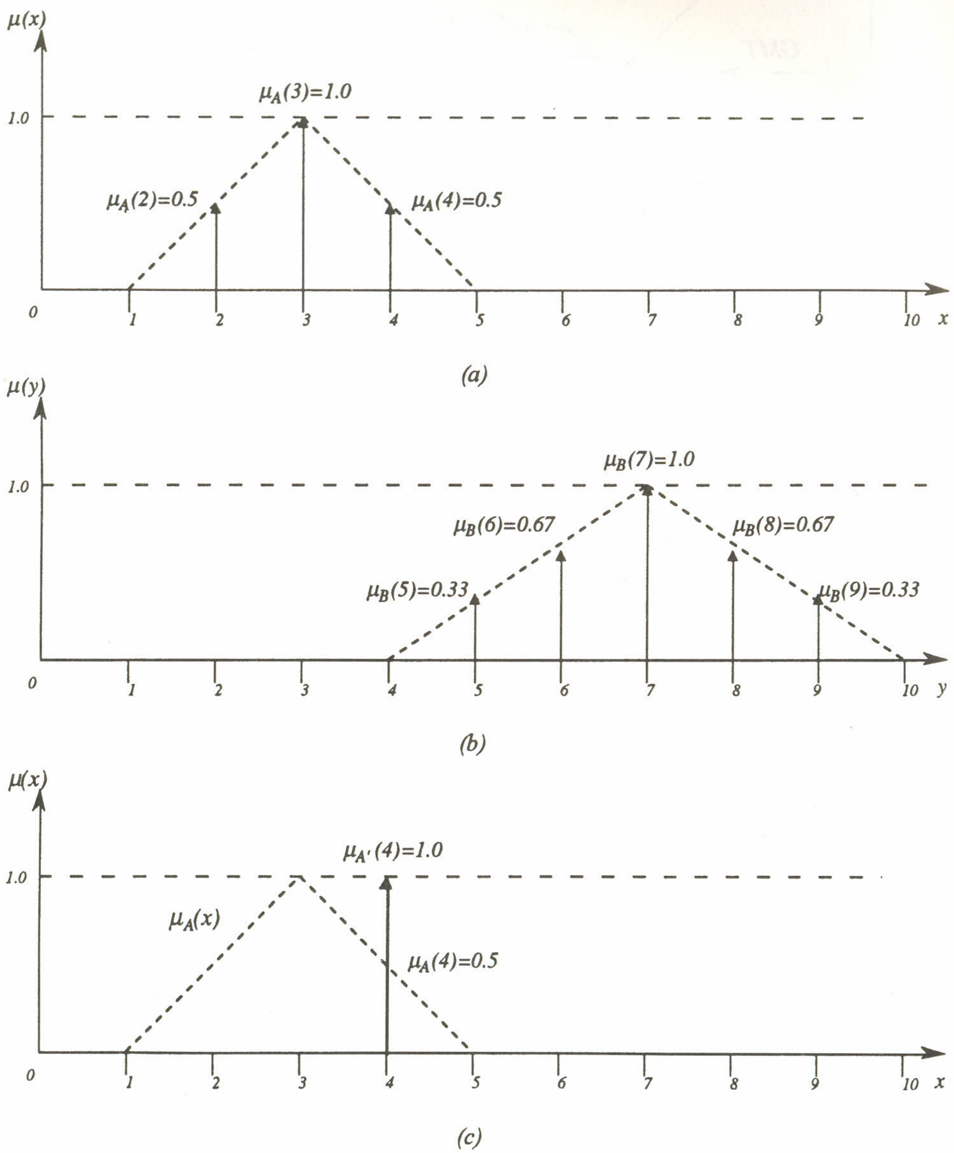


Figure 5.10 (*a*) The membership function of the antecedent *A*. (*b*) The fuzzy value *B* of the consequent. (*c*) A fuzzy *A'* that approximately matches the antecedent in Example 5.1.

we have

$$A = \sum_{i=0}^{10} \mu_A(x_i)/x_i$$
$$= 0.5/2 + 1.0/3 + 0.5/4 \qquad \text{(E5.1-1)}$$
$$B = \sum_{i=0}^{10} \mu_B(y_i)/y_i$$
$$= 0.33/5 + 0.67/6 + 1.0/7 + 0.67/8 + 0.33/9 \qquad \text{(E5.1-2)}$$
$$A' = \sum_{i=0}^{10} \mu_A(x_i)/x_i = 1.0/4 \qquad \text{(E5.1-3)}$$

All variables are defined over the same universe of discourse, the set of integers from 0 to 10; and, as is customary, zero membership singletons are omitted.

First, let us compute the membership function $\mu(x_i, y_j)$ of the implication relation $R(x_i, y_j)$ that analytically describes the rule using the Mamdani min implication operator. Having discrete fuzzy values we use discretized fuzzy relations as well. From Table 5.2 we see that the membership function of the implication relation is given by

$$\mu(x_i, y_j) = \phi_c[\mu_A(x_i), \mu_B(y_j)]$$
$$= \mu_A(x_i) \wedge \mu_B(y_j) \qquad \text{(E5.1-4)}$$

Thus the analytical form of the rule is given by the implication relation (5.3-3)—that is,

$$R(x_i, y_j) = \sum_{(x_i, y_j)} \mu(x_i, y_j)/(x_i, y_j)$$
$$= 0.33/(2,5) + 0.5/(2,6) + 0.5/(2,7) + 0.5/(2,8)$$
$$+ 0.33/(2,9) + 0.33/(3,5) + 0.67/(3,6) + 1.0(3,7)$$
$$+ 0.67/(3,8) + 0.33/(3,9) + 0.33/(4,5) + 0.5/(4,6)$$
$$+ 0.5/(4,7) + 0.5/(4,8) + 0.33/(4,9) \qquad \text{(E5.1-5)}$$

where the membership function is computed using (E5.1-4). The implication relation $R(x_i, y_j)$ of (E5.1-5) is shown in Table 5.3. We note that the full relation is defined over the Cartesian product of the discrete universe of discourse of LHS and RHS variables. Since each universe of discourse is the set of integers from 0 to 10, the Cartesian product is the 11×11 product

Table 5.3 The fuzzy implication relation in Example 5.1

x_i \ y_j	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0.33	0.50	0.5	0.5	0.33	0
3	0	0	0	0	0	0.33	0.67	1.0	0.67	0.33	0
4	0	0	0	0	0	0.33	0.5	0.5	0.5	0.33	0
5	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0

space shown in Table 5.3. The nontrivial part of the relation is found in the shaded cells of Table 5.3.

To find B' we compose A' with $R(x_i, y_j)$ in accordance with equation (5.4-2). It is sufficient to consider the nonzero part of the relation—that is, the shaded part of Table 5.3. We use matrix notation and remind ourselves (see Chapter 3) that max–min composition ($\circ$) is analogous to matrix multiplication with max ($\vee$) and min ($\wedge$) in place of addition ($+$) and multiplication ($\times$), respectively. From Equation (5.4-2) we have

$$B'(y_j) = A'(x_i) \circ R(x_i, y_j)$$

$$= [0 \quad 0 \quad 1] \circ \begin{bmatrix} 0.33 & 0.50 & 0.50 & 0.50 & 0.33 \\ 0.33 & 0.67 & 1.00 & 0.66 & 0.33 \\ 0.33 & 0.50 & 0.50 & 0.50 & 0.33 \end{bmatrix} \qquad \text{(E5.1-6)}$$

where the column vector for A' ranges from $x = 2$ to $x = 4$ (see Figure 5.10) which is the same as the row range of the implication matrix. The columns of the implication matrix range from $y = 5$ to $y = 9$ (see Table 5.3). From equation (5.4-3) the membership function of the first element of the consequent—that is, at $y = 5$—is computed as follows:

$$\mu_{B'}(5) = \bigvee_x [0 \wedge 0.33, 0 \wedge 0.33, 1 \wedge 0.33]$$
$$= \bigvee_x [0, 0, 0.33]$$
$$= 0.33 \qquad \text{(E5.1-7)}$$

Similarly we compute the rest of B'. The result is

$$B' = 0.33/5 + 0.50/6 + 0.50/7 + 0.5/8 + 0.33/9 \qquad \text{(E5.1-8)}$$

as shown in Figure 5.11. It should be noted in Figure 5.11 that the membership function of B' is essentially the membership function of B clipped at a height equal to the degree that A' matches A (see Figure 5.10c). This value is called the *degree of fulfillment* (DOF) of the rule. It is a measure of the degree of similarity between the input A' and the antecedent of the rule A. In the present case we have that

$$\text{DOF} = 0.5 \qquad \text{(E5.1-9)}$$

Clipping the membership function of the consequent by DOF is a feature of ϕ_c, the Mamdani min implication operator. Whenever we use ϕ_c to model the implication relation involved in GMP we get such a clipping transformation of the consequent. The situation is shown in general in Figure 5.12. We shall encounter clipping in Chapter 6 when dealing with control applications of linguistic descriptions. We should keep in mind that clipping depends on the Mamdani min implication operator (not to be confused with max–min composition). Using different implication operators to model the implication relation leads to different shape transformations of the RHS of a rule evaluated under GMP. □

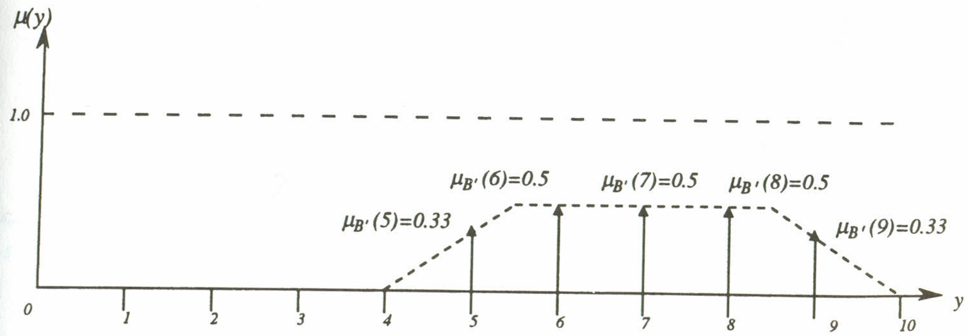


Figure 5.11 The fuzzy set B' produced by evaluating the linguistic description of Example 5.1.

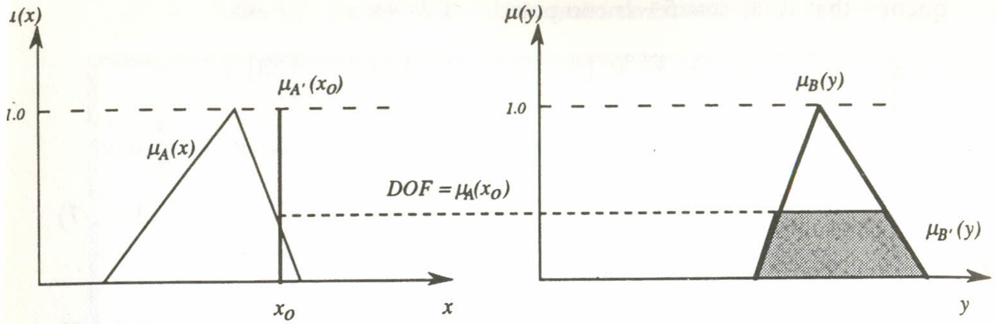


Figure 5.12 When the Mamdani min implication operator is used to model an implication relation, GMP clips the membership function of the consequent by the DOF of the rule.

Example 5.2 GMP with Larsen Product Implication. In this example we evaluate a fuzzy *if/then* rule, whose implication relation is modeled by the Larsen product fuzzy implication operator ϕ_p (see Table 5.2) using GMP. The antecedent and consequent variables of rule *if* x is A *then* y is B are shown in Figures 5.13a and 5.13b. The membership function of the input value A' is shown in Figure 5.13c. From Figure 5.14 we have

$$A = \sum_{i=-5}^{5} \mu_A(x_i)/x_i$$

$$= 0.33/(-1) + 0.67/0 + 1.0/1 + 0.75/2 + 0.5/3 + 0.25/4 \quad \text{(E5.2-1)}$$

$$B = \sum_{i=-5}^{5} \mu_B(y_i)/y_i$$

$$= 0.50/(-4) + 1.0/(-3) + 0.67/(-2) + 0.33/(-1) \quad \text{(E5.2-2)}$$

$$A' = \sum_{i=-5}^{5} \mu_A(x_i)/x_i = 1.0/3 \quad \text{(E5.2-3)}$$

Using equations (5.4-2) and (5.4-3) for GMP we can compute the membership function of value B'. First, however, we have to obtain the membership function of the implication relation, $\mu(x, y)$, using the Larsen product fuzzy implication operator ϕ_p (see Table 5.2). The implication relation has the membership function

$$\mu(x_i, y_j) = \phi_p\big[\mu_A(x_i), \mu_B(y_j)\big] = \mu_A(x_i) \cdot \mu_B(y_j) \quad \text{(E5.2-4)}$$

and plugging in numbers from equations (E5.2-1) to (E5.2-3) we obtain the

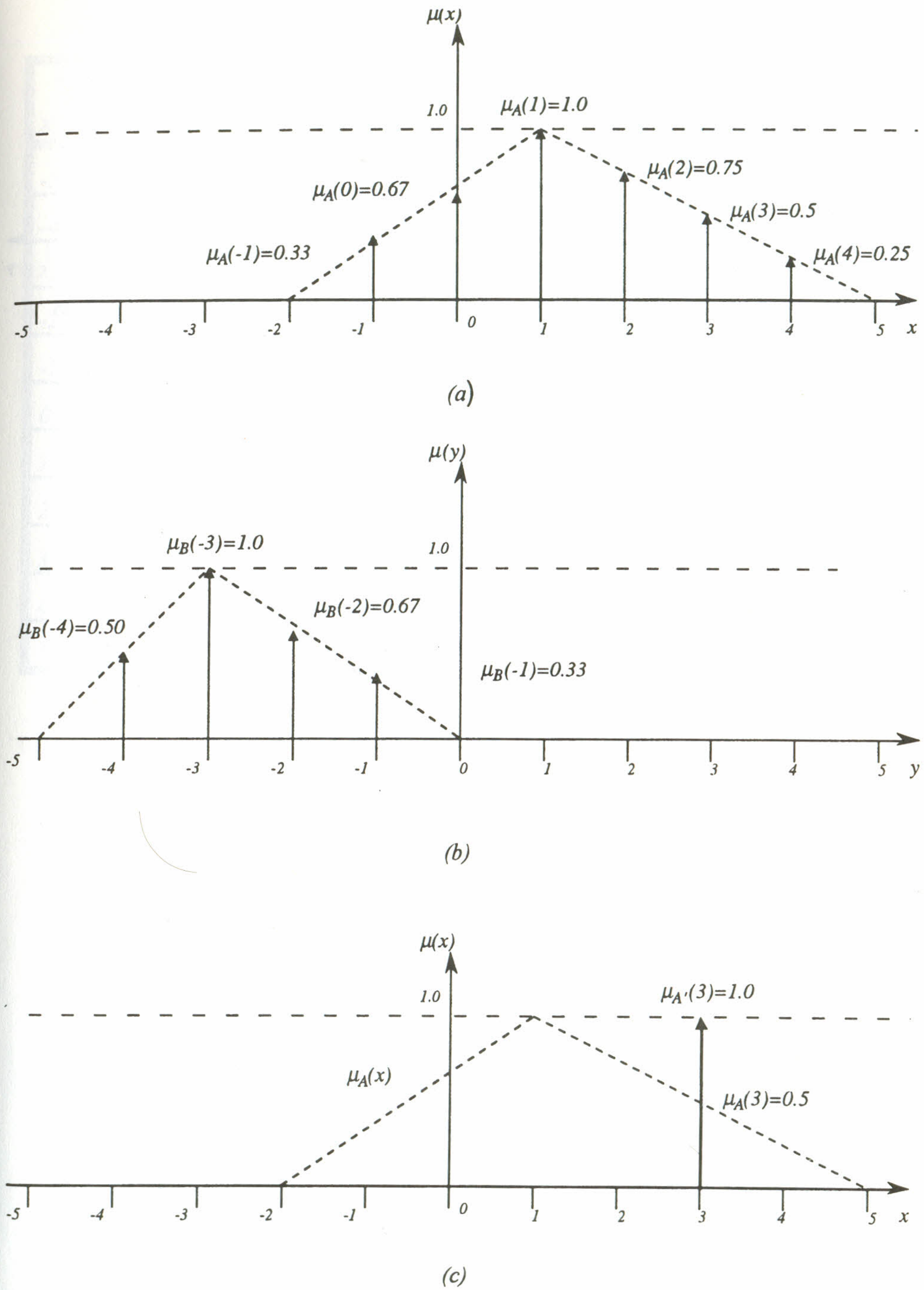


Figure 5.13 (*a*) The fuzzy value A of the antecedent. (*b*) The fuzzy value B of the consequent. (*c*) The singleton A' that approximately matches the antecedent in Example 5.2.

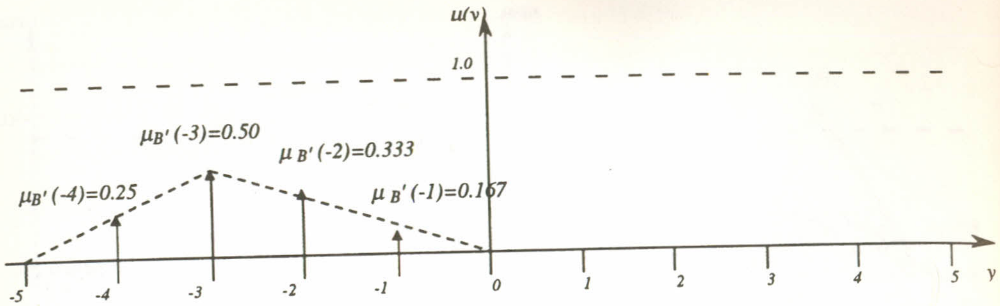


Figure 5.14 The fuzzy set B' produced by evaluating the linguistic description in Example 5.2.

implication relation

$$R(x_i, y_j) = \sum_{(x_i, y_j)} \mu(x_i, y_j)/(x_i, y_j)$$

$$\begin{aligned}
&= 0.167/(-1,-4) + 0.333/(-1,-3) + 0.222/(-1,-2) \\
&\quad + 0.111/(-1,-1) + 0.333/(0,-4) + 0.667/(0,-3) \\
&\quad + 0.445/(0,-2) + 0.222/(0,-1) + 0.500/(1,-4) \\
&\quad + 1.000/(1,-3) + 0.667/(1,-2) + 0.333/(1,-1) \\
&\quad + 0.375/(2,-4) + 0.750/(2,-3) + 0.500/(2,-2) \\
&\quad + 0.250/(2,-1) + 0.250/(3,-4) + 0.500/(3,-3) \\
&\quad + 0.333/(3,-2) + 0.167/(3,-1) + 0.125/(4,-4) \\
&\quad + 0.250/(4,-3) + 0.167/(4,-2) + 0.083/(4,-1)
\end{aligned} \tag{E5.2-5}$$

The implication relation of (E5.2-5) can also be seen as the shaded part of Table 5.4, where we use a similarly scaled discrete universe of discourse for both antecedent and consequent variables, namely, integers from -5 to $+5$. Thus, the implication relation is taking values on an 11×11 Cartesian product space as shown.

We find B' through GMP—that is, max–min composition of A' with $R(x_i, y_j)$. Again we need only consider the nonzero part of the relation—that is, the shaded part of Table 5.4. We use matrix notation and remind ourselves (see Chapter 3) that max–min composition is analogous to matrix multiplication with the max ($\vee$) and min ($\wedge$) in the role of addition ($+$) and

Table 5.4 Implication relation in Example 5.2

x_i \ y_j	-5	-4	-3	-2	-1	0	1	2	3	4	5
-5	0	0	0	0	0	0	0	0	0	0	0
-4	0	0	0	0	0	0	0	0	0	0	0
-3	0	0	0	0	0	0	0	0	0	0	0
-2	0	0	0	0	0	0	0	0	0	0	0
-1	0	0.165	0.333	0.222	0.111	0	0	0	0	0	0
0	0	0.333	0.667	0.445	0.222	0	0	0	0	0	0
1	0	0.500	1.00	0.667	0.333	0	0	0	0	0	0
2	0	0.375	0.750	0.495	0.250	0	0	0	0	0	0
3	0	0.250	0.50	0.333	0.166	0	0	0	0	0	0
4	0	0.125	1.00	0.167	0.083	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0

multiplication ($\times$). Thus we have

$$B' = A' \circ R = \begin{bmatrix} 0 & 0 & 0 & 0 & 1.0 & 0 \end{bmatrix} \circ \begin{bmatrix} 0.165 & 0.333 & 0.222 & 0.111 \\ 0.333 & 0.667 & 0.444 & 0.222 \\ 0.50 & 1.000 & 0.667 & 0.333 \\ 0.375 & 0.750 & 0.500 & 0.250 \\ 0.250 & 0.500 & 0.333 & 0.167 \\ 0.125 & 0.250 & 0.167 & 0.083 \end{bmatrix} \tag{E5.2-6}$$

The column vector to the left—that is, the discrete membership function of A'—ranges from $x = -1$ to $x = 4$, matching the x dimension (which is the row dimension) of the matrix. The y dimension (the columns) of the matrix ranges from $y = -4$ to $y = -1$ as shown in Table 5.4. The result is the fuzzy value

$$B' = 0.25/(-4) + 0.50/(-3) + 0.33/(-2) + 0.167/(-1) \tag{E5.2-7}$$

B' is shown in Figure 5.14. Its membership function is essentially the

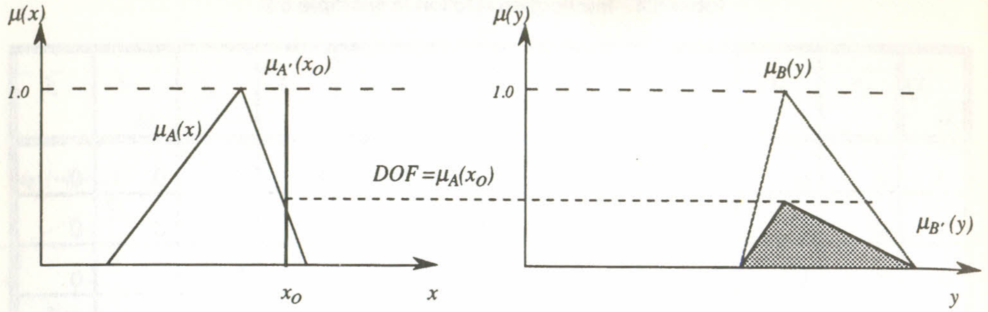


Figure 5.15 When GMP is used to evaluate a rule whose implication relation is modeled by the Larsen product, the membership function of the consequent is scaled by the DOF.

membership function of B scaled (multiplied) by the degree that A' matches the membership function of A at $x = 5$—that is, the DOF of the rule by A'. Scaling the membership function of the consequent by DOF is a feature of the Larsen product fuzzy implication operator ϕ_p. Schematically this property of ϕ_p is shown in Figure 5.15. Other fuzzy implication operators (Table 5.2) result in different shape transformations of the consequent. □

5.5 FUZZY ALGORITHMS

A *fuzzy algorithm* is a procedure for performing a task formulated as a collection of fuzzy *if/then* rules. The rules are defined over the same product space and are connected by the connective *ELSE* which may be interpreted either as *union* or *intersection* depending on the implication operator used for the individual rules.[7] Consider for example the algorithm

$$\begin{aligned} &\textit{if} \quad x \text{ is } A_1 \quad \textit{then} \quad y \text{ is } B_1 \quad \textit{ELSE} \\ &\textit{if} \quad x \text{ is } A_2 \quad \textit{then} \quad y \text{ is } B_2 \quad \textit{ELSE} \\ &\ldots \\ &\textit{if} \quad x \text{ is } A_n \quad \textit{then} \quad y \text{ is } B_n \end{aligned} \tag{5.5-1}$$

We recall that analytically each rule in (5.5-1) is represented by an implication relation $R(x, y)$ and that the form of $R(x, y)$ depends on the implication operator used (see Table 5.2). Table 5.5 lists the most common interpretation

[7]*ELSE* can also be interpreted as *arithmetic sum* and *product* (as well as other T and S norms), which we do not use in this book.

Table 5.5 Interpretation of *ELSE* under various implications

IMPLICATION	INTERPRETATION OF ELSE
ϕ_m, Zadeh Max-Min	AND ($\wedge$)
ϕ_c, Mamdani Min	OR ($\vee$)
ϕ_p, Larsen Product	OR ($\vee$)
ϕ_a, Arithmetic	AND ($\wedge$)
ϕ_b, Boolean	AND ($\wedge$)
ϕ_{bp}, Bounded Product	OR ($\vee$)
ϕ_{dp}, Drastic Product	OR ($\vee$)
ϕ_s, Standard Sequence	AND ($\wedge$)
ϕ_Δ, Gougen	AND ($\wedge$)
ϕ_g, Gödelian	AND ($\wedge$)

of the connective *ELSE* for the implication operators shown in Table 5.2 (in the next chapter we will see more on this). The relation of the entire collection of rules (5.5-1) is called the *algorithmic relation*

$$R_\alpha(x, y) = \int_{(x,y)} \mu_\alpha(x, y)/(x, y) \tag{5.5-2}$$

and is either the *union* ($\vee$) or the *intersection* ($\wedge$) of the implication relations of the individual rules. A fuzzy algorithm is a linguistic description evaluated analytically using composition operations just as we did in the case of single-rule linguistic descriptions. Given a new fuzzy value A' we evaluate

Mamdani, E. H., Applications of Fuzzy Set Theory to Control Systems: A Survey, in *Fuzzy Automata and Decision Processes*, M. M. Gupta, G. N. Saridis and B. R. Gaines, eds., North-Holland, New York, 1977, pp. 1–13.

Mizumoto, M., Fuzzy Controls Under Various Reasoning Methods, *Information Sciences*, Vol. 45, pp. 129–141, 1988.

Pappis, C. P., and Sugeno, M., Fuzzy Relational Equations and the Inverse Problem, *Fuzzy Sets and Systems*, Vol. 15, pp. 79–90, 1985.

Pin, G. F., Watanabe, H., Symon, J., Pattay, R. S., Autonomous Navigation of Mobile Robot Using Custom-Designed Qualitative Reasoning VLSI Chips and Boards, *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, Nice, France, May 1992, pp. 123–128.

Ray, S. K., and Majumder, D. D., Fuzzy Rule Based Approach to Image Segmentation, in *Fuzzy Computing Theory, Hardware, and Applications*, M. M. Gupta and T. Yamakawa, eds., Elsevier/North-Holland, Amsterdam, 1988, pp. 375–397.

Ruan, D., and Kerre, E. E., Fuzzy Implication Operators and Generalized Fuzzy Method of Cases, *Fuzzy Sets and Systems*, Vol. 54, pp. 23–37, 1993.

Shimizu, K., Osumi, M., and Imae, F., Digital Fuzzy Processor FP-5000, *Proceedings of the 2nd International Conference on Fuzzy Logic and Neural Networks, IIZUKA '92*, pp. 539–542, Iizuka, Japan, July 17–22, 1992.

Yamakawa, T., Fuzzy Hardware Systems of Tomorrow, in *Approximate Reasoning in Intelligent Systems, Decision and Control*, E. Sanchez, and L. A. Zadeh, eds., Pergamon Press, Oxford, 1987, pp. 1–20.

Yamakawa, T., Intrinsic Fuzzy Electronic Circuits for Sixth Generation Computers, in *Fuzzy Computing Theory, Hardware, and Applications*, M. M. Gupta and T. Yamakawa, eds., Elsevier/North-Holland, Amsterdam, 1988, pp. 157–173.

Zadeh, L. A., Outline of a New Approach to the Analysis of Complex Systems and Decision Processes, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 1, pp. 28–44, 1973.

Zadeh, L. A., The Concept of a Linguistic Variable and Its Application to Approximate Reasoning, *Information Sciences*, Vol. 8, pp. 199–249, 1975.

Zadeh, L. A., Fuzzy Sets and Information Granularity, *Advances in Fuzzy Set Theory and Applications*, M. M. Gupta, R. K. Ragade, and R. R. Yager, eds., North-Holland, Amsterdam, 1979, pp. 3–18.

Zadeh, L. A., A Computational Approach to Fuzzy Quantifiers in Natural Languages, *Computer and Mathematics*, Vol. 9, pp. 149–184, 1983.

Zadeh, L. A., Fuzzy Logic, *IEEE Computer*, pp. 83–93, April 1988.

Zimmermann, H. J., *Fuzzy Set Theory and Its Applications*, Kluwer-Nijhoff, Boston, 1985.

PROBLEMS

1. The Mamdani min implication operator given by Equation (5.3-6) is alleged to be a simplification of the Zadeh max–min implication operator given by Equation (5.3-5). Explain what simplifications were made and

discuss how these influence implication operations in fuzzy operations such as control. Illustrate your discussion with sketches.

2. A linguistic description is comprised of a single rule

$$\text{if} \quad x \text{ is } A \quad \text{then } y \text{ is } B$$

where A and B are the fuzzy numbers

$$A = 0.33/6 + 0.67/7 + 1.00/8 + 0.67/9 + 0.33/10$$
$$B = 0.33/1 + 0.67/2 + 1.00/3 + 0.67/4 + 0.33/5$$

The implication relation of the rule is modeled through the Larsen product implication operator. If a fuzzy number $x = A'$ is a premise, use generalized modus ponens to infer a fuzzy number $y = B'$ as the consequent. A' is defined by

$$A' = 0.5/5 + 1.00/6 + 0.5/7$$

3. Using the data given in Problem 2, Mamdani min implication operator, and generalized modus ponens, evaluate the rule.

4. Using the data given in Problem 2, arithmetic implication operator, and generalized modus ponens, evaluate the rule.

5. Using the data given in Problem 2, Boolean implication operator, and generalized modus ponens, evaluate the rule.

6. Using the data given in Problem 2, bounded product implication operator, and generalized modus ponens, evaluate the rule.

7. Using the data given in Problem 2, Zadeh max–min implication operator and generalized modus ponens, evaluate the rule.

8. Given the rule and fuzzy values for A and B as well as the B' that you found in Problem 2, use generalized modus tollens to infer an A'.

9. What happens if you repeat Problem 8, having used bounded product implication operator to model the rule?

10. Which of the fuzzy implication operators given in Table 5.2 reduce to classical modus ponens under max–min composition? Examine each operator and show an example of what happens using the data found in Example 5.1.

11. This problem requires an investigation on your part of the concept of fuzzy functions. Generally, a fuzzy function can be understood as a mapping between fuzzy sets and the extension principle can serve as a tool for generalizing ordinary mappings. Depending on where fuzziness

(5.5-1) through GMP formally stated as

$$
\begin{array}{llll}
\textit{if} & x \text{ is } A_1 & \textit{then} \quad y \text{ is } B_1 & \textit{ELSE} \\
\textit{if} & x \text{ is } A_2 & \textit{then} \quad y \text{ is } B_2 & \textit{ELSE} \\
\cdots & & & \\
\textit{if} & x \text{ is } A_n & \textit{then} \quad y \text{ is } B_n & \\
 & x \text{ is } A' & & \\
\hline
 & y \text{ is } B' & &
\end{array}
\tag{5.5-3}
$$

The output value B' in (5.5-3) is computed by max–min composition (and more generally max-$*$) of A' and $R_\alpha(x, y)$—that is,

$$B' = A' \circ R_\alpha(x, y) \tag{5.5-4}$$

The membership function of B' is

$$\mu_{B'}(y) = \bigvee_x \left[\mu_{A'}(x) \wedge \mu_\alpha(x, y) \right] \tag{5.5-5}$$

Then inverse problem is solved through GMT, stated as

$$
\begin{array}{llll}
\textit{if} & x \textit{ is } A_1 & \textit{then} \quad y \textit{ is } B_1 & \textit{ELSE} \\
\textit{if} & x \textit{ is } A_2 & \textit{then} \quad y \textit{ is } B_2 & \textit{ELSE} \\
\cdots & & & \\
\textit{if} & x \textit{ is } A_n & \textit{then} \quad y \textit{ is } B_n & \\
 & & y \textit{ is } B' & \\
\hline
 & x \textit{ is } A' & &
\end{array}
\tag{5.5-6}
$$

The membership function of A' in (5.5-4) can be computed by max–min composition (and more generally max-$*$) of $R_\alpha(x, y)$ and B'—that is,

$$A' = R_\alpha(x, y) \circ B' \tag{5.5-7}$$

with the membership function of A' given by

$$\mu_{A'}(x) = \bigvee_x \left[\mu_\alpha(x, y) \wedge \mu_{B'}(y) \right] \tag{5.5-8}$$

In the elementary fuzzy algorithm of (5.5-1) there is only one variable in the antecedent side of each implication and one on the consequent side. Generally, we are interested in linguistic descriptions that may have more than one variable in either side, which we refer to as *multivariate fuzzy algorithms*. The interpretations of the connective *ELSE* are the same as for the elementary algorithm of (5.5-1). Consider an *if/then* rule of the form

$$\textit{if} \quad x_1 \textit{ is } A_1 \quad \textit{AND} \quad x_2 \textit{ is } A_2 \quad \textit{AND} \cdots \textit{AND} \quad x_m \textit{ is } A_m \quad \textit{then} \quad y \textit{ is } B \tag{5.5-9}$$

where $x_1, \ldots, x_m$ are antecedent linguistic variables with $A_1, \ldots, A_m$ their respective fuzzy values and y is the consequent linguistic variable with B its fuzzy value. The connective *AND* in the LHS of rule (5.5-9) can be analytically modeled either as min or as arithmetic product. In such cases we can combine the propositions in the LHS either through min ($\wedge$) or through product ($\cdot$) and use an appropriate implication operator ϕ (Table 5.2) to obtain the membership function of implication relation of (5.5-9). Thus we have

$$\mu(x_1, x_2, \ldots, x_m, y) = \phi\left[\mu_{A_1}(x_1) \wedge \mu_{A_2}(x_2) \wedge \cdots \wedge \mu_{A_m}(x_m), \mu_B(y) \right] \tag{5.5-10}$$

In case *AND* is analytically modeled as product, the implication relation has membership function

$$\mu(x_1, x_2, \ldots, x_m, y) = \phi\left[\mu_{A_1}(x_1) \cdot \mu_{A_2}(x_2) \cdot \cdots \cdot \mu_{A_m}(x_m), \mu_B(y) \right] \tag{5.5-11}$$

where ϕ is an appropriate implication operator from Table 5.2. In a similar manner the connective *OR* can be interpreted as max ($\vee$) or as sum ($+$) or other S norms (see Appendix A)).

Less frequently we encounter *multivariate fuzzy implications* involving m nested fuzzy implications, each having one antecedent variable, of the form

$$\textit{if} \quad x_1 \textit{ is } A_1 \quad \textit{then} \left(\textit{if } x_2 \textit{ is } A_2 \textit{ then } \cdots \left(\textit{if } x_m \textit{ is } A_m \textit{ then } y \textit{ is } B \right) \cdots \right) \tag{5.5-12}$$

The membership function of a multivariate fuzzy implication of equation (5.5-12) is obtained through repeated application of an implication operator (see Table 5.2), once for each nested *if/then* rule:

$$\mu(x_1, x_2, \ldots, x_m, y) = \phi\left[\mu_{A_1}(x_1), \phi\left[\mu_{A_2}(x_2), \ldots, \phi\left[\mu_{A_m}(x_m), \mu_B(y) \right] \right] \right] \tag{5.5-13}$$

When we have several rules of the form of (5.5-9) or (5.5-12) the overall algorithmic relation depends upon the implication operator used and the related interpretation of the connective *ELSE*.

Let us consider a fuzzy algorithm consisting of n multivariate fuzzy implications of the form shown in (5.5-9). We have m variables $x_1, \ldots, x_m$ on the antecedent side of the jth *if/then* rule taking values $A_{1j}, \ldots, A_{mj}$ ($j = 1, \ldots, n$) and only one consequent linguistic variable y, taking values $B_1, B_2, \ldots, B_n$. Our fuzzy algorithm is the collection of rules

$$\begin{array}{llllllll}
\textit{if} & x_1 \textit{ is } A_{11} & \textit{AND} & x_2 \textit{ is } A_{21} & \textit{AND} \cdots \textit{AND} & x_{1m} \textit{ is } A_{m1} & \textit{then } y \textit{ is } B_1 & \textit{ELSE} \\
\textit{if} & x_1 \textit{ is } A_{12} & \textit{AND} & x_2 \textit{ is } A_{22} & \textit{AND} \cdots \textit{AND} & x_m \textit{ is } A_{m2} & \textit{then } y \textit{ is } B_2 & \textit{ELSE} \\
\cdots & & & & & & & \\
\textit{if} & x_1 \textit{ is } A_{1n} & \textit{AND} & x_2 \textit{ is } A_{2n} & \textit{AND} \cdots \textit{AND} & x_m \textit{ is } A_{mn} & \textit{then } y \textit{ is } B_n &
\end{array} \tag{5.5-14}$$

The fuzzy algorithm of (5.5-14) is analytically described by an algorithmic relation of the form

$$R_\alpha(x_1, x_2, \ldots, x_m, y) = \int_{(x_1, x_2, \ldots, x_m, y)} \mu_\alpha(x_1, x_2, \ldots, x_m, y) / (x_1, x_2, \ldots, x_m, y) \tag{5.5-15}$$

and when discrete fuzzy sets are used we obtain

$$R_\alpha(x_{1i}, x_{2i}, \ldots, x_{mi}, y_j) = \sum_{(x_{1i}, x_{2i}, \ldots, x_{mi}, y_j)} \mu_\alpha(x_{1i}, x_{2i}, \ldots, x_{mi}, y_j) / (x_{1i}, x_{2i}, \ldots, x_{mi}, y_j) \tag{5.5-16}$$

The membership function in (5.5-15) or (5.5-16) can be obtained from the implication relation of the individual rules and appropriate interpretation of the connectives *AND* and *ELSE*. Once the algorithmic relation is known, GMP may be used to obtain an output B' given inputs $A'_1, A'_2, \ldots, A'_m$— that is,

$$\begin{array}{llllllll}
\textit{if} & x_1 \textit{ is } A_{11} & \textit{AND} & x_2 \textit{ is } A_{21} & \textit{AND} \cdots \textit{AND} & x_{1m} \textit{ is } A_{m1} & \textit{then } y \textit{ is } B_1 & \textit{ELSE} \\
\textit{if} & x_1 \textit{ is } A_{12} & \textit{AND} & x_2 \textit{ is } A_{22} & \textit{AND} \cdots \textit{AND} & x_m \textit{ is } A_{m2} & \textit{then } y \textit{ is } B_2 & \textit{ELSE} \\
\cdots & & & & & & & \\
\textit{if} & x_1 \textit{ is } A_{1n} & \textit{AND} & x_2 \textit{ is } A_{2n} & \textit{AND} \cdots \textit{AND} & x_m \textit{ is } A_{mn} & \textit{then } y \textit{ is } B_n & \\
 & x_1 \textit{ is } A'_1 & & x_2 \textit{ is } A'_2 & \cdots & x_m \textit{ is } A'_m & & \\
\hline
 & & & & & & y \textit{ is } B' &
\end{array} \tag{5.5-17}$$

Let A'_1 be a new input to (5.5-17). The membership function of B' is given by max–min composition of the fuzzy set $A'_1 = A'(x_1)$ and $R_\alpha(x_1, x_2, \ldots, x_m, y)$— that is,

$$B'(y) = A'_1 \circ R_\alpha(x_1, x_2, \ldots, x_m, y) \tag{5.5-18}$$

When m inputs are offered to the algorithm and the connective *AND* in the LHS of each rule is interpreted as min, GMP will give an output value

$$B'(y) = \left(\bigwedge_{j=1}^{m} A'(x_j) \right) \circ R_\alpha(x_1, x_2, \ldots, x_m, y) \tag{5.5-19}$$

with membership function

$$\mu_{B'}(y) = \bigvee_{x_1} \bigvee_{x_2} \cdots \bigvee_{x_m} \left[\left(\bigwedge_{j=1}^{m} \mu_{A'}(x_j) \right) \wedge \mu_\alpha(x_1, x_2, \ldots, x_m, y) \right] \tag{5.5-20}$$

Other compositions may be used as well, such as the max-product or, more generally max-$*$, to obtain the membership function of the new consequent B' (see Chapter 3).

REFERENCES

Hirota, K., and Ozawa, K., Fuzzy Flip-Flop as a Basis of Fuzzy Memory Modules, in *Fuzzy Computing Theory, Hardware, and Applications*, M. M. Gupta and T. Yamakawa, eds., Elsevier/North-Holland, Amsterdam, 1988, pp. 173–183.

Huertas, J. L., Sanchez-Solano, S., Barriga, A., and Baturone, I., Serial Architecture for Fuzzy Controllers: Hardware Implementation Using Analog/Digital VLSI Techniques, in *Proceedings of the 2nd International Conference on Fuzzy Logic and Neural Networks, IIZUKA '92*, Iizuka, Japan, July 17–22, 1992, pp. 535–539.

Isik, C., Inference Hardware for Fuzzy Rule-Based Systems, in *Fuzzy Computing Theory, Hardware, and Applications*, M. M. Gupta and T. Yamakawa, eds., Elsevier/North-Holland, Amsterdam, 1988, pp. 185–194.

Kandel, A., *Fuzzy Mathematical Techniques with Applications*, Addison-Wesley, Reading, MA, 1986.

Larsen, P. M., Industrial Applications of Fuzzy Logic Control, *International Journal of Man–Machine Studies*, Vol. 12, No. 1, 1980, pp. 3–10.

Lee, C. C., Fuzzy Logic in Control Systems: Fuzzy Logic Controller—Part-I, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20, No. 2, pp. 404–418, 1990a.

Lee, C. C., Fuzzy Logic in Control Systems: Fuzzy Logic Controller—Part-II, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20, No. 2, pp. 419–435, March/April 1990b.

occurs one gets different types of fuzzy functions. The problem is this: Set up a fuzzy function that will take as input ambient temperatures and will produce as output energy demand to a power plant. There are no unique solutions, but rather, different approaches to formulating the solution. State clearly, what could be fuzzy in this problem; what assumptions you need to make; what crisp function, if any, you start with. Also, give the functional form and test it. Does it make sense? Could you get higher energy demand for lower temperatures from your model?

12. Given the assumptions made in Problem 11, find a fuzzy algorithm that describes the same general relation as the fuzzy function you developed in Problem 11.