1

INTRODUCTION TO HYBRID ARTIFICIAL INTELLIGENCE SYSTEMS

As complexity rises, precise statements lose meaning and meaningful statements lose precision.

Lotfi A. Zadeh

1.1 INTRODUCTION

The term "artificial intelligence" (AI), in its broadest sense, encompasses a number of technologies that includes, but is not limited to, expert systems, neural networks, genetic algorithms, fuzzy logic systems, cellular automata, chaotic systems, and anticipatory systems. Interestingly, most of these technologies have their origins in biological or behavioral phenomena related to humans or animals, and many of these technologies are simple analogs of human and animal systems. Hybrid intelligent systems generally involve two, three, or more of these individual AI technologies that are either used in series or integrated in a way to produce advantageous results through synergistic interactions. In this book we have placed emphasis on neural networks and fuzzy systems; to a lesser extent, we have also placed emphasis on genetic algorithms where needed for optimization and expert systems where they are needed to supervise and implement the other three technologies. A major emphasis in this book will be on the integration of fuzzy and neural systems in a synergistic way.

In data and/or information processing, the objective is generally to gain an understanding of the phenomena involved and to evaluate relevant parameters quantitatively. This is usually accomplished through "modeling" of the systems, either experimentally or analytically (using mathematics and physical principles). Most hybrid systems relate experimental data to systems or models. Once we have a model of a system, we can carry out various procedures (e.g., sensitivity analysis, statistical regression, etc.) to gain a better understanding of the system. Such experimentally derived models give insight into the nature of the system behavior that can be used to enhance mathematical and physical models.

There are, however, many situations in which the phenomena involved are very complex and often not well understood and for which first principles models are not possible. Even more often, physical measurements of the pertinent quantities are very difficult and expensive. These difficulties lead us to explore the use of neural networks and fuzzy logic systems as a way of obtaining models based on experimental measurements.

1.2 NEURAL NETWORKS AND FUZZY LOGIC SYSTEMS

In the history of science and technology, new developments often come from observations made from a different perspective. Interrelationships that we take for granted today may not have been so obvious in earlier decades. For instance, we regularly gain insight into the behavior of a dynamic system by viewing it as being in the "time domain" and/or the "frequency domain." However, for the first four decades of the twentieth century, statisticians dealt with autocorrelation and cross-correlation functions (in the time domain) while electrical engineers dealt with power- and cross-spectral densities (in the frequency domain) without either group realizing that these two concepts were related to each other through Fourier transformations.

Both the statisticians and the electrical engineers have found that analysis of the fluctuations in process variables provides useful information about the variables as well as the processes involved. These fluctuations, which result in uncertainties in measured variables, often are caused by some sort of random driving function (i.e., fluid turbulence, rotational unbalance, etc.). Investigation, and the subsequent understanding of these uncertainties (fluctuations), led to the development of the field of "random noise analysis" which spawned such analytical specialties as vibration analysis, seismology, electrocardiography, oceanography, and so on.

Neural networks and fuzzy systems represent two distinct methodologies that deal with uncertainty. Uncertainties that are important include both those in the model or description of the systems involved as well as those in the variables. These uncertainties usually arise from system complexity (often including nonlinearities; we think of complexity as a property of system description—that is, related to the means of computation or language and not merely a system's complicated nature). Neural networks approach the modeling representation by using precise inputs and outputs which are used to "train" a generic model which has sufficient degrees of freedom to formulate a good approximation of the complex relationship between the inputs and the outputs. In fuzzy systems, the reverse situation prevails. The input and output variables are encoded in "fuzzy" representations, while their interrelationships take the form of well-defined *if/then* rules. Zadeh's ingenious observation that the uncritical pursuit of precision may be not only unnecessary but actually a source of error led him to the notion of a fuzzy set.

Each of these approaches has its own advantages and disadvantages. Neural networks can represent (i.e., model) complex nonlinear relationships, and they are very good at classification of phenomena into preselected categories used in the training process. On the other hand, the precision of the outputs is sometimes limited because the variables are effectively treated as analog variables (even when implemented on a digital computer), and "minimization of least squares errors" does not mean "zero error." Furthermore, the time required for proper training a neural network using one of the variations of "backpropagation" training can be substantial (sometimes hours or days). Perhaps the "Achilles heel" of neural networks is the need for substantial data that are representative and cover the entire range over which the different variables are expected to change.

Fuzzy logic systems address the imprecision of the input and output variables directly by defining then with fuzzy numbers (and fuzzy sets) that can be expressed in linguistic terms (e.g., *cold*, *warm*, and *hot*). Furthermore, they allow far greater flexibility in formulating system descriptions at the appropriate level of detail. Fuzziness has a lot to do with the parsimony and hence the accuracy and efficiency of a description. This means that complex process behavior can be described in general terms without precisely defining the complex (usually nonlinear) phenomena involved. Paraphrasing *Occam's Razor*, the philosophical principle holding that more parsimonious descriptions are more representative of nature, we may say that fuzzy descriptions are more parsimonious and hence easier to formulate and modify, more tractable, and perhaps more tolerant of change and even failure.

Neural network and fuzzy logic technologies are quite different, and each has unique capabilities that are useful in information processing. Yet, they often can be used to accomplish the same results in different ways. For instance, they can speed the unraveling and specifying the mathematical relationships among the numerous variables in a complex dynamic process. Both can be used to control nonlinear systems to a degree not possible with conventional linear control systems. They perform mappings with some degree of imprecision. However, their unique capabilities can also be combined in a synergistic way. It is this combination of the two technologies (as well as combinations with other AI technologies) with the goal of gaining the advantages of both that is the focus of this book.

1.3 THE PROGRESS IN SOFT COMPUTING

Soft computing refers to computational tools whose distinguishing characteristic is that they provide approximate solutions to approximately formulated problems (Aminzadeh, 1994). Fuzzy logic, neural networks, probabilistic reasoning, expert systems, and genetic algorithms are some of the constituents of soft computing, all having roots in the field of Artificial Intelligence. Whereas the traditional view of computing considers any imprecision and uncertainty undesirable, in soft computing some tolerance for imprecision and uncertainty is exploited in order to develop more tractable and robust models of systems, at a lower cost and greater economy of communication and computation.

Few of those who attended the historic 1956 Dartmouth Conference to discuss "the potential use of computers and simulation in every aspect of learning and any other feature of intelligence" could have envisioned the evolution and growth of the embryonic artificial intelligence field and the impact it has had on our lives. It was there that the term "artificial intelligence" was coined, perhaps because of the emphasis on learning and simulation. The term "cybernetics" was in vogue at that time with its emphasis on potential control of both man and machines. Vacuum-tube-type analog computers had reached a state of maturity that they (along with high fidelity stereo sound systems) were being marketed as "Heathkits," while the digital "supercomputer" of the time was an IBM-650 with about 2000 words of magnetic drum memory storage that operated at about 2 kHz.

It was in this environment that Frank Rosenblat developed the Perceptron by adding a learning capability to the McCulloch-Pitts model of the neuron, Marvin Minsky built the first "learning machine" (using 40 processing elements, each with six vacuum tubes and a motor/clutch/control system), and Bernard Widrow developed the "Adaline" (adaptive linear element) that even today is used in virtually every high-speed modem and telephone switching system to cancel out the echo of reflected signals. Boolean algebra was standard procedure, and John McCarthy and John von Neuman were putting forth the relative merits of symbolic (LISP) and conventional computer languages. Although there was little in the way of theoretical bases providing an understanding of these systems, work proceeded on an experimental basis that was guided primarily by the genius of the individuals involved.

Today, some 40 years later, the whole world has changed. The computing capacity of that IBM-650 is now encapsulated in a "wristwatch" computer, the Perceptron and Adaline processing elements are instantiated in neural network computing and processing methodologies, learning algorithms are routinely processed on digital computers of all sizes, *Boolean logic* and algebra are being replaced by fuzzy logic concepts, LISP is fading away in favor of object-oriented computer languages for artificial intelligence (e.g., C++), the analog computer has virtually disappeared, and the modern personal computer most of us have on our desks may have more than a gigabyte of memory, operate at a processing rate of 200 MHz or more, and be part of a vast global network of computers capable of sharing on-line information in numerical, textual, visual and audible forms.

The educational, technological, economic, and social impact and significance of the computer as a tool for computation and communication have been continuously discussed and debated in the last few decades. In the 1970's Ralph Lapp, in an interesting book called *The Logarithmic Century*, captured the ever-changing and accelerating trend in the development of technology and economics (Lapp, 1973). Yet, he did not foresee the magnitude of the impact of advanced computer technology, especially the role that communications and information processing would have on society. Perhaps our Japanese colleagues have a better grasp of the issues involved. In a book entitled *The Next Century*, Halberstam (1991) reported a conversation with a retired high official of MITI (Ministry for International Trade and Industry) who in 1987 said "... the (Japanese) educational system is in danger of... producing young people who have the intellectual capacity of computers but who will be inferior to computers in what they can actually do. The computers have caught up."

Of course, the road of technological change is by no means simple. Eloquent critics such as Neil Postman in his evocative book *Technopoly* strongly point out the dangers of subordinating culture and society to an uncritical faith in the machine (Postman, 1993). Indeed, computers cannot magically solve our problems. In today's highly integrated world, however, a diverse world population needs the multiplicity of opportunities provided by the new communications and computer technologies, and soft computing is promising to become a powerful means for obtaining quick, yet accurate and acceptable, solutions to many problems. We, the engineers who work to provide and apply these new soft computing tools, ardently hope that they will be used for the benefit of mankind.

1.4 INTELLIGENT MANAGEMENT OF LARGE COMPLEX SYSTEMS

The real challenge to *soft computing* is the intelligent management of large complex systems—that is, organizations operating on the scale of the global economy and resting on an highly globalized information infrastructure. It is perhaps the most important activity facing industrial, educational, military, and governmental organizations throughout the world today. Management decisions made today will reverberate throughout these organizations for years to come. Management decisions made in the past have shaped these organizations and have made them what they are today. In some cases, large organizations have made the "right decisions" and have been spectacularly successful. However, it is clear that the decisions of other large organizations have not been wise. Multi-hundred million and billion dollar losses, followed by layoffs, restructuring, mergers, and, all too often, bankruptcy are common as these organizations pay the price for past mistakes. Why did these organizations get into trouble or fail? What steps can be taken to ensure that decisions today are better than those in the past? The answers to these questions are as varied as the nature of the organizations. Typical responses given are as follows: incompetent management, too much attention to the next quarterly earnings, lack of vision, fierce new competition, unfair regulatory practices by governments, poor design, failure to keep up with the times, antagonism between labor and management, inadequate research an opment, and so on. The list goes on and on. All of these may be explanations in individual situations, but correcting these alleged prob will not guarantee that an organization will be successful in meeting its in the future. The successful strategies and methodologies of the 198C not work in the next century.

Large complex systems, as a general class, are often virtually out of control; indeed they are often deemed to be uncontrollable because of their complexity. The reversal of this situation is absolutely essential in a society in which systems tend to grow without bound because of the perceived ben of "economy of scale." Indeed, organizations tend to grow until they rea level of inefficiency that inhibits and impedes their growth. Only an orga tion with virtually unlimited resources or power (i.e., governmental orga tions) can continue to grow under these conditions. The finite resourc the world and of individual nations, as well as the growing populatio aspires for improved living conditions, demand improved efficiency. absolutely essential for the benefit of mankind, as well as most n nations that tend to be dominated by large complex systems, that systems be brought under intelligent control and management. The a in digital computer technology (both hardware and software) during t decade, along with the associated development of *soft computing*, ap r the first time in history, to provide a means of implementing intelligent control of complex systems which are so necessary in delivering the fruits of industrial technology and commerce to global society.

The personal computers or workstations available on the desk of engineers and managers today with its *soft computing* tools has the power of main-frame computers just a few years ago. They provide the capability of keeping track of what is going on in any organization (intelligence), they can provide the tools to examine the data in excruciating detail (analysis), they can provide models of the behavior of complex systems (synthesis) which then permits predictions into the future, at least into the short-term future, and they can provide recommendations for specific actions (intelligent management) that can be communicated to those who have a need to act in a form that they can understand (intelligent communications). To the extent that an organization's management is willing to utilize these tools correctly, significant progress in solving some of these problems by making the "right" decisions will follow.

Unfortunately, making the "right" decision under the circumstance at the time the decision is made does not guarantee success. It may have been the "right" decision at the time, but the consequences may be unpredictable because of the time lag between decision and results in a changing environment. What is needed is a form of anticipatory control as discussed in Chapter 15. In the absence of an ability to predict the future behavior of systems, many conservative organizations have elected a "minimum step" approach—that is, make a decision at the last possible moment that involves the least amount of (financial or resource) commitment and produces results at the earliest possible time. However, this can be a strategy for disaster if the basis on which the decision is made is not valid. All too often, decisions must be made in the absence of complete data, which gives rise to uncertainty in the analysis and a higher probability of an erroneous decision. Even such a "minimum step" approach requires *reliable intelligence*, *accurate analysis*, *valid synthesis*, *intelligent management*, and *intelligent communications*, because there is little margin for error. While a modern digital computer cannot guarantee the availability of these five attributes, they simply would not be available without the modern digital computers and *soft computing*.

Perhaps the single attribute that gives neurofuzzy systems an advantage in addressing the problems of large complex systems is the ability to perform what in mathematical terms would be called *many-to-many mappings*. Such mappings are an inherent part of complex systems, because every single input to a system can influence every single output; i.e., one significant input change may generate significant changes in many outputs. Most approaches to systems analysis can only deal with *one-to-one* or *many-to-one mappings*—that is, with the special class of mathematical mappings that we call *functions*, which have been the premier mathematical relation since the Newtonian revolution of the *Principia*. It is now possible and desirable, however, to effectively compute with more complex mathematical mappings than functions—that is, with *many-to-many relations* (see Section 5.1). This gives us the hope and the expectation that large complex systems can be dealt with in a flexible, reliable, and near-optimal manner.

We do not claim that *neurofuzzy* systems *per se* can bring about the control of large complex systems. It is clear to us that the integration of many technologies in a yet indiscernible manner is an essential step in the right direction. *Neurofuzzy systems* represent an integration of fuzzy logic and neural networks that have capabilities beyond either of these technologies individually (Haykin, 1994; Kartalopoulos, 1996). When we further integrate other technologies, perhaps some not yet discovered, in the decades ahead, we can look forward to tools with sufficient power to tackle problems such as intelligent control of large complex systems.

1.5 STRUCTURE OF THIS BOOK

This book is divided into four parts: Part I, entitled "Fuzzy Systems: Concepts and Fundamentals," explores the fundamentals of fuzzy logic systems and includes the following chapters:

Chapter 2. Foundations of Fuzzy Approaches
Chapter 3. Fuzzy Relations

Chapter 4. Fuzzy Numbers
Chapter 5. Linguistic Descriptions and Their Analytical Forms
Chapter 6. Fuzzy Control

Part II, entitled "Neural Networks: Concepts and Fundamentals," explores the fundamentals of neural networks and includes the following chapters:

Chapter 7. Fundamentals of Neural Networks
Chapter 8. Backpropagation and Related Training Algorithms
Chapter 9. Competitive, Associative, and Other Special Neural Networks
Chapter 10. Dynamic Systems and Neural Control
Chapter 11. Practical Aspects of Using Neural Networks

Part III, entitled "Integrated Neural–Fuzzy Technology," explores the joint use of neural networks and fuzzy logic systems. It includes the following chapters:

Chapter 12. Fuzzy Methods in Neural Networks
Chapter 13. Neural Methods in Fuzzy Systems
Chapter 14. Selected Hybrid Neurofuzzy Applications
Chapter 15. Dynamic Hybrid Neurofuzzy Systems

Part IV, entitled "Other Artificial Intelligence Systems," reviews other artificial intelligence systems that can be used with neural networks and fuzzy systems. It includes the following chapters:

Chapter 16. Expert Systems in Neurofuzzy Systems
Chapter 17. Genetic Algorithms
Chapter 18. Epilogue

1.6 MATLAB©[1] SUPPLEMENT

In this text, we have included problems for students at the end of most chapters. Generally, these problems are pedagogical in nature and are intended to be simple enough that they can be solved without the aid of computer software. To supplement these exercises, we have enlisted our colleague, Dr. J. Wesley Hines of the University of Tennessee, to prepare a *MATLAB© Supplement for Neural and Fuzzy Approaches in Engineering*, a paperback book of approximately 150 pages published by John Wiley and Sons, in which the Student Edition of MATLAB© (The MathWorks Inc.,

[1] MATLAB is copyrighted by MathWorks Inc., of Natick, MA.

1995; Hanselman, 1996) can be used for demonstrations and solving more sophisticated problems. Of course, the Professional Version of MATLAB© can also be used if it is available.

This supplement was written using the MATLAB© Notebook and Microsoft WORD.Version 6.0. The Notebook allows MATLAB© commands to be entered and evaluated while in the WORD environment, which allows the document to both briefly explain the theoretical details and also show MATLAB© implementations. It also allows the user to experiment with changing the MATLAB© code fragments in order to gain a better understanding of their application.

This supplement contains numerous examples that demonstrate the practical implementation of relevant techniques using MATLAB©. Although MATLAB© toolboxes for Fuzzy Logic and Neural Networks are available, they are not required to run the examples given. This supplement should be considered to be a brief introduction to the MATLAB© implementation of neural and fuzzy systems, and we and the author strongly recommend the use of Neural Networks and Fuzzy Logic Toolboxes for a more in-depth study of these information-processing technologies. Many of the m-files and examples are extremely general and portable while other examples will have to be altered significantly for use to solve specific problems.

The content of the *MATLAB© Supplement* is coordinated with *Fuzzy and Neural Approaches in Engineering* so that students can use it to enhance their knowledge of fuzzy systems, neural networks, and neurofuzzy systems. Indeed, it is expected that many instructors will choose to use both this book and the *MATLAB© Supplement* together in their classes. Practicing engineers and scientists in industry who want to use this text to learn about neural, fuzzy, and neurofuzzy systems will find this supplement to be a valuable aid in their self-study.

REFERENCES

Aminzadeh, F., and Jamshidi, M., *Soft Computing*, Prentice-Hall, Englewood Cliffs, NJ, 1994.

Halberstam, D., *The Next Century*, William Morrow and Company, New York, 1991.

Hanselman, D., and Littlefield, B., *Mastering MATLAB*, Prentice-Hall, Englewood Cliffs, NJ, 1996.

Haykin S., *Neural Networks: A Comprehensive Foundation*, IEEE Computer Society Press, Macmillan, New York, 1994.

Kartalopoulos, S., *Understanding Neural Networks and Fuzzy Logic*, IEEE Press, New York, 1996.

Lapp, R., *The Logarithmic Century*, Prentice-Hall, Englewood Cliffs, NJ, 1973.

Mathworks, Inc., *The Student Edition of MATLAB© Users Guide*, Natick, MA, 1995.

Postman, N., *Technopoly*, Vintage Books, New York, 1993.

FUZZY SYSTEMS CONCEPTS AND FUNDAMENTALS

2

FOUNDATIONS OF FUZZY APPROACHES

2.1 FROM CRISP TO FUZZY SETS

The mathematical foundations of fuzzy logic rest in *fuzzy set theory*, which can be thought of as a generalization of *classical set theory*. A familiarity with the novel notions, notations, and operations of fuzzy sets is useful in studying fuzzy logic principles and applications; acquiring it will be our main goal in this chapter.

Fuzziness is a property of language. Its main source is the imprecision involved in defining and using symbols. Consider, for example, the set of chairs in a room. In set theory the set of chairs may be established by pointing to every object in a room asking the question, *Is it a chair*? In classical set theory we are allowed to use only two answers: Yes or No. Let us code Yes as 1 and No as 0. Thus, our answers will be in the pair {0, 1}. If the answer is 1, an element belongs to a set; if the answer is 0, it does not. In the end we collect all the objects whose label is 1 and obtain the *set of chairs in a room*. Suppose, however, that we ask the question, *Which objects in a room may function as a chair*? Again we could point to every object and ask, *Could it function as a chair*? The answer here too could artificially be restricted to {0, 1}. Yet, the set of objects in a room that may *function* as a chair may include not only chairs but also desks, boxes, parts of the floor, and so on. It is a set not uniquely defined. It all depends on what we mean by the word *function*. Words like *function* have many shades of meaning and can be used in many different ways. Their meaning and use may vary with different persons, circumstances, and purposes; it depends on the specifics of a situation. We say therefore that the *set of objects that may function as a chair* is a *fuzzy set*, in the sense that we may not have crisply defined criteria for deciding membership into the set. Objects such as desks, boxes, and part of the floor may function as *chairs*, to a degree. It should be noted, however, that there is nothing fuzzy about the material objects themselves: Chairs, boxes, and desks are what they are. Fuzziness is a feature of their representation in a milieu of symbols and is generally a property of models, computational procedures, and language.

Let us now review some notions of classical set theory. *Classical sets* are crisply defined collections of distinct elements (numbers, symbols, objects, etc.), and for this reason we also call them *crisp sets*. The elements of all the sets under consideration in a given situation belong to an invariable, constant set, called the *universal set* or *universe* or more often the *universe of discourse*.[1] The fact that elements of a set A either belong or do not belong to a crisp set A can be formally indicated by the *characteristic function of A*, defined as

$$\chi_A(x) = \begin{cases} 1 & \text{iff} \quad x \in A \\ 0 & \text{iff} \quad x \notin A \end{cases} \tag{2.1-1}$$

where the symbols $\in$ and $\notin$ denote that x *is* and *is not* a member of A, respectively, and iff is shorthand for "if and only if." The pair of numbers $\{0, 1\}$ is called the *valuation set*. Another way of writing equation (2.1-1) is

$$\chi_A(x): X \to \{0, 1\} \tag{2.1-2}$$

The notation of equation (2.1-2) is read as follows: *There exists a function* $\chi_A(x)$ *mapping every element of the set* X *(our universe of discourse) to the set* $\{0, 1\}$. It emphasizes that the characteristic function is a mechanism for mapping the set X to the valuation set $\{0, 1\}$. Important operations in crisp sets such as *union*, *intersection*, and *complementation* are familiar to us from elementary mathematics. They are usually represented through Venn diagrams but may also be expressed in terms of the characteristic function.

Fundamentally, sets are *categories*. Defining suitable categories and using operations for manipulating them is a major task of modeling and computation. From image recognition to measurement and control, the notion of *category*, or *set*, is essential in the definition of system variables, parameters, their ranges, and their interactions. The constraint to have a dual degree of membership to a set, an all-or-nothing, is a consequence of a desire to abstract a system description away from the multitude of intricacies and complexities that exist in reality and focus on factors of primary influence. Nevertheless, given our modern-day computational technologies, it may be unduly restrictive. This is particularly the case when it is desired to develop computer models *easily calibrated* to the specifics of a system and endowed with adaptive and self-organizing capabilities (Zadeh, 1973, 1988).

[1]The term *universe of discourse* is used in fuzzy logic; it comes from classical logic and describes the complete set of individual elements able to be referred to or quantified.

2.2 FUZZY SETS

As we saw in the previous section, in classical set theory there is a rather strict sense of membership to a set; that is, an element either *belongs* or *does not belong* to the set. In 1965 Lotfi A. Zadeh introduced *fuzzy sets*, where a more flexible sense of membership is possible (Zadeh, 1965). In fuzzy sets many degrees of membership are allowed. The degree of membership to a set is indicated by a number between 0 and 1—that is, a number in the interval [0, 1]. The point of departure for fuzzy sets is simply generalizing the valuation set from the pair of numbers {0, 1} to all numbers found in [0, 1]. By expanding the valuation set we alter the nature of the characteristic function, now called *membership function* and denoted by $\mu_A(x)$. We no longer have *crisp* sets but instead have *fuzzy* sets. Since the interval [0, 1] contains an infinity of numbers, infinite degrees of membership are possible. Thus, in view of equation (2.1-2) we say that *a membership function maps every element of the universe of discourse X to the interval* [0, 1], and we formally write this mapping as

$$\mu_A(x): X \rightarrow [0, 1] \tag{2.2-1}$$

Equation (2.2-1) is a generalization of the mapping shown in equation (2.1-2). Membership functions are a simple yet versatile mathematical tool for indicating flexible membership to a set and, as we shall see, for modeling and quantifying the meaning of symbols. A question often asked by people beginning the study of fuzzy sets is, How are membership functions found? Membership functions may represent an individual's (subjective) notion of a vague class—for example, *objects in a room functioning as chairs, tall people, acceptable performance, small contribution to system stability, little improvement, big benefit*, and so on. In designing and operating controllers or automatic decision-making tools, for example, modeling such notions is a very important task. Membership functions may also be determined on the basis of statistical data or through the aid of neural networks. In Part III of this book we will look at the synergistic relation between neural networks and fuzzy logic toward this end (Kosko, 1992). At this point we can simply say that membership functions are primarily subjective in nature; this does not mean that they are assigned arbitrarily, but rather on the basis of application-specific criteria (Kaufmann, 1975; Dubois and Prade, 1980; Zimmermann, 1985).

There are two commonly used ways of denoting fuzzy sets. If X is a universe of discourse and x is a particular element of X, then a fuzzy set A defined on X may be written as a collection of ordered pairs

$$A = \{(x, \mu_A(x))\}, \qquad x \in X \tag{2.2-2}$$

where each pair $(x, \mu_A(x))$ is called a *singleton* and has x first, followed by its membership in A, $\mu_A(x)$. In crisp sets a singleton is simply the element x by itself. In fuzzy sets a singleton is two things: x and $\mu_A(x)$. For example, the set of *small integers*, A, defined (subjectively) over the universe of discourse of positive integers may be given by the collection of singletons

$$A = \{(1,1.0),(2,1.0),(3,0.75),(4,0.5),(5,0.3),(6,0.3),(7,0.1),(8,0.1)\}$$

Thus the fourth singleton from the left tells us that 4 belongs to A to a degree of 0.5. A singleton is also written as $\mu_A(x)/x$—that is, by putting membership first, followed by the marker "/" separating it from x.[2] Singletons whose membership to a fuzzy set is zero may be omitted. The *support set* of a fuzzy set A is the set of its elements that have membership function other than the trivial membership of zero.

An alternative notation, used more often than equation (2.2-2), explicitly indicates a fuzzy as the *union* of all $\mu_A(x)/x$ singletons—that is,

$$A = \sum_{x_i \in X} \mu_A(x_i)/x_i \tag{2.2-3}$$

The *summation* sign in equation (2.2-3) indicates the *union* of all singletons (the union operation in set theory is like "addition"). Equation (2.2-3) assumes that we have a *discrete universe of discourse*. In this alternative notation the set of *small integers* above may be written as

$$\begin{aligned} A &= \mu_A(1)/1 + \mu_A(2)/2 + \mu_A(3)/3 + \mu_A(4)/4 + \mu_A(5)/5 \\ &\quad + \mu_A(6)/6 + \mu_A(7)/7 + \mu_A(8)/8 \\ &= 1.0/1 + 1.0/2 + 0.75/3 + 0.5/4 + 0.3/5 + 0.3/6 + 0.1/7 + 0.1/8 \end{aligned}$$

For a continuous universe of discourse, we write equation (2.2-3) as

$$A = \int_X \mu_A(x)/x \tag{2.2-4}$$

where the *integral* sign in equation (2.2-4) indicates the *union* of all $\mu_A(x)/x$ singletons.[3] Consider, for example, the fuzzy set *small numbers* defined (subjectively) over the set of non-negative real numbers through a continuous

[2] It should be noted that "/" does not indicate "division"; it is merely a marker.

[3] Note that the integral sign is not the same as the integral sign of differential and integral calculus. It is used here in the sense that the integral sign is used in set theory—that is, to indicate the *sum* or *union* of individual *singletons*.

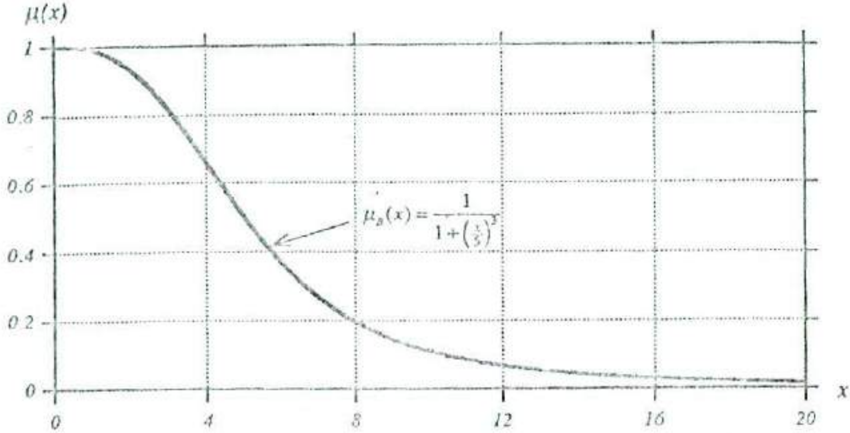


Figure 2.1 Zadeh diagram for the fuzzy set $B = \{small\ numbers\}$.

membership function $\mu_B(x)$ given by

$$\mu_B(x) = \frac{1}{1 + \left(\frac{x}{5}\right)^3} \tag{2.2-5}$$

Using the form of equation (2.2-4) the fuzzy set B is written as

$$B = \int_{x \geq 0} \mu_B(x)/x = \int_{x \geq 0} \left[\frac{1}{1 + \left(\frac{x}{5}\right)^3}\right]/x \tag{2.2-6}$$

The membership function of fuzzy set B is shown in Figure 2.1.[4] A graph like this is called a *Zadeh diagram*.

2.3 BASIC TERMS AND OPERATIONS

Many fuzzy set operations such as *intersection* and *union* are defined through the min ($\wedge$) and max ($\vee$) operators. *Min* and *max* are analogous to *product* ($\cdot$) and *sum* ($+$) in algebra (Dubois and Prade, 1980; Klir and Folger, 1988; Terano et al., 1992). Let us take a look at how they are used.

[4] Fuzzy sets are sometimes called *fuzzy subsets*, reflecting the fact that they are subsets of a larger set—that is, the *universe of discourse*. Although the term *fuzzy subsets* is factually correct, we will use the standard term *fuzzy set* for convenience.

First, min ($\wedge$) and max ($\vee$) may be used to select the minimum and maximum of two elements—for example, $2 \wedge 3 = 2$, or $2 \vee 3 = 3$. We also write $\min(2,3) = 2$, or $\max(2,3) = 3$. Formally, the minimum of two elements μ_1 and μ_2 denoted either as $\min(\mu_1, \mu_2)$, $\wedge(\mu_1, \mu_2)$, or $\mu_1 \wedge \mu_2$ is defined as

$$\mu_1 \wedge \mu_2 = \min(\mu_1, \mu_2) \equiv \begin{cases} \mu_1 & \text{iff} \quad \mu_1 \leq \mu_2 \\ \mu_2 & \text{iff} \quad \mu_1 > \mu_2 \end{cases} \tag{2.3-1}$$

where, the " $\equiv$ "symbol means "by definition"and iff is shorthand for "if and only if." Similarly the maximum of two elements μ_1 and μ_2, denoted as $\max(\mu_1, \mu_2)$ or $\mu_1 \vee \mu_2$, is defined as

$$\mu_1 \vee \mu_2 = \max(\mu_1, \mu_2) \equiv \begin{cases} \mu_1 & \text{iff} \quad \mu_1 \geq \mu_2 \\ \mu_2 & \text{iff} \quad \mu_1 < \mu_2 \end{cases} \tag{2.3-2}$$

Second, min ($\wedge$) and max ($\vee$) may operate on an entire set, selecting the least element (called *infimum* in mathematical analysis) or the greatest element (called *supremum*) of the set. For example, $\wedge\ (0.01, 0.33, 0.44, 0.999) = 0.01$ and $\vee\ (0.01, 0.33, 0.44, 0.999) = 0.999$. Formally we write this as

$$\mu = \wedge A = \inf A \tag{2.3-3}$$

and

$$\mu = \vee A = \sup A \tag{2.3-4}$$

where μ is an element of A—that is, $\mu \in A$.

In addition, min ($\wedge$) and max ($\vee$) may be used as *functions* operating on single elements or on entire sets, for example, to find the smallest element μ out of a list of elements $(\mu_1, \mu_2, \ldots, \mu_m)$—that is,

$$\mu = \wedge(\mu_1, \mu_2, \ldots, \mu_m) \tag{2.3-5}$$

which is the same as

$$\mu = \mu_1 \wedge \mu_2 \wedge \cdots \wedge \mu_m \tag{2.3-6}$$

We sometimes use a shorthand notation for equations (2.3-5) and (2.3-6) and write them as

$$\mu = \bigwedge_{k=1}^{m} (\mu_k) \tag{2.3-7}$$

This notation is analogous to finite *product* notation in algebra (or finite *summation* when $\vee$ is used). There is in fact a more general analogy between min and max and the operations of *multiplication* and *addition*. They both have the same properties of associativity and distributivity, and thus in equations that involve min and max we may employ them in the same manner as *multiplication* (·) and *addition* (+). We will see an interesting example of these properties in the composition of fuzzy relations (Chapter 3), where we treat composition as *matrix multiplication* with (∧) and (∨) in place of product (·) and sum (+).

Min (∧) and max (∨) can also operate on a collection of sets as for example in

$$A = \wedge(A_1, A_2, \ldots, A_m) \tag{2.3-8}$$

which can be succinctly written as

$$A = \bigwedge_{k=1}^{m} (A_k) \tag{2.3-9}$$

Using primarily min (∧) and max (∨), a number of useful notions and operations involving fuzzy sets can be defined.[5]

Empty Fuzzy Set

A fuzzy set A is called *empty* (denoted as $A = \varnothing$) if its membership function is zero everywhere in its universe of discourse X—that is,

$$A \equiv \varnothing \qquad \text{if } \mu_A(x) = 0, \forall x \in X \tag{2.3-10}$$

where "$\forall x \in X$" is shorthand notation indicating "for any element x in X."

Normal Fuzzy Set

A fuzzy set is called *normal* if there is at least one element x_0 in the universe of discourse where its membership function equals one—that is,

$$\mu_A(x_0) = 1 \tag{2.3-11}$$

More than one element in the universe of discourse can satisfy equation (2.3-11).[6]

[5] These operations can also be defined in terms of *T-norms* (see Appendix).

[6] It should be noted that the term *normal* does not refer to the area under the curve of the membership function. It simply means what the definition says: At least one point, maybe more, needs to have full membership value.

Equality of Fuzzy Sets

Two fuzzy sets are said to be *equal* if their membership functions are equal everywhere in the universe of discourse—that is,

$$A \equiv B \qquad \text{if } \mu_A(x) = \mu_B(x) \tag{2.3-12}$$

Union of Two Fuzzy Sets

The *union* of two fuzzy sets A and B defined over the same universe of discourse X is a new fuzzy set $A \cup B$ also on X, with membership function which is the maximum of the grades of membership of every x to A and B—that is,

$$\mu_{A \cup B}(x) \equiv \mu_A(x) \vee \mu_B(x) \tag{2.3-13}$$

The *union* of two fuzzy sets is related to the logical operation of *disjunction* (OR) in fuzzy logic. Equation (2.3-13) can be generalized to any number of fuzzy sets over the same universe of discourse.

Intersection of Fuzzy Sets

The *intersection* of two fuzzy sets A and B is a new fuzzy set $A \cap B$ with membership function which is the minimum of the grades of every x in X to the sets A and B, i.e.,

$$\mu_{A \cap B}(x) \equiv \mu_A(x) \wedge \mu_B(x) \tag{2.3-14}$$

The *intersection* of two fuzzy sets is related to *conjunction* (AND) in fuzzy logic. The definition of *intersection* in (2.3-14) can be generalized to any number of fuzzy sets over the same universe of discourse.

Complement of a Fuzzy Set

The *complement* of a fuzzy set A is a new fuzzy set, $\bar{A}$, with membership function

$$\mu_{\bar{A}}(x) \equiv 1 - \mu_A(x) \tag{2.3-15}$$

Fuzzy set *complementation* is equivalent to *negation* (NOT) in fuzzy logic.

Product of Two Fuzzy Sets

The product of two fuzzy sets A and B defined on the same universe of discourse X is a new fuzzy set, $A \cdot B$, with membership function that equals the algebraic product of the membership functions of A and B,

$$\mu_{A \cdot B}(x) \equiv \mu_A(x) \cdot \mu_B(x) \tag{2.3-16}$$

The product of two fuzzy sets can be generalized to any number of fuzzy sets on the same universe of discourse.

Multiplying a Fuzzy Set by a Crisp Number

We can multiply the membership function of a fuzzy set A by the crisp number a to obtain a new fuzzy set called *product* $a \cdot A$. Its membership function is

$$\mu_{aA}(x) \equiv a \cdot \mu_A(x) \tag{2.3-17}$$

The operations of multiplication and raising a fuzzy set to a power that we see next are useful for modifying the meaning of linguistic terms (Zadeh, 1975).

Power of a Fuzzy Set

We can raise fuzzy set A to a power α (positive real number) by raising its membership function to α. The *α power* of A is a new fuzzy set, A^α, with membership function

$$\mu_{A^\alpha}(x) \equiv [\mu_A(x)]^\alpha \tag{2.3-18}$$

Raising a fuzzy set to the second power is usually taken to be equivalent to linguistically changing it through the modifier *VERY* (Zadeh, 1983) (see Chapter 5). Thus the square of the membership function of $B = \{$*small numbers*$\}$ in Figure 2.1 is taken to represent the fuzzy set $B^2 = \{$*VERY small numbers*$\}$.

Raising a fuzzy set to the second power is a particularly useful operation and therefore has its own name. It is called *concentration* or *CON*. Taking the square root of a fuzzy set is called *dilation* or *DIL* (an operation useful for representing analytically the linguistic modifier *MORE OR LESS*).

Example 2.1 Union, Intersection, and Complement of Fuzzy Sets. Consider the Zadeh diagram of fuzzy sets A and B shown in Figure 2.2a and defined by membership functions

$$\mu_A(x) = \frac{1}{1 + 0.3(x-8)^2} \quad \text{and} \quad \mu_B(x) = \frac{1}{1 + \left(\dfrac{x}{5}\right)^3} \tag{E2.1-1}$$

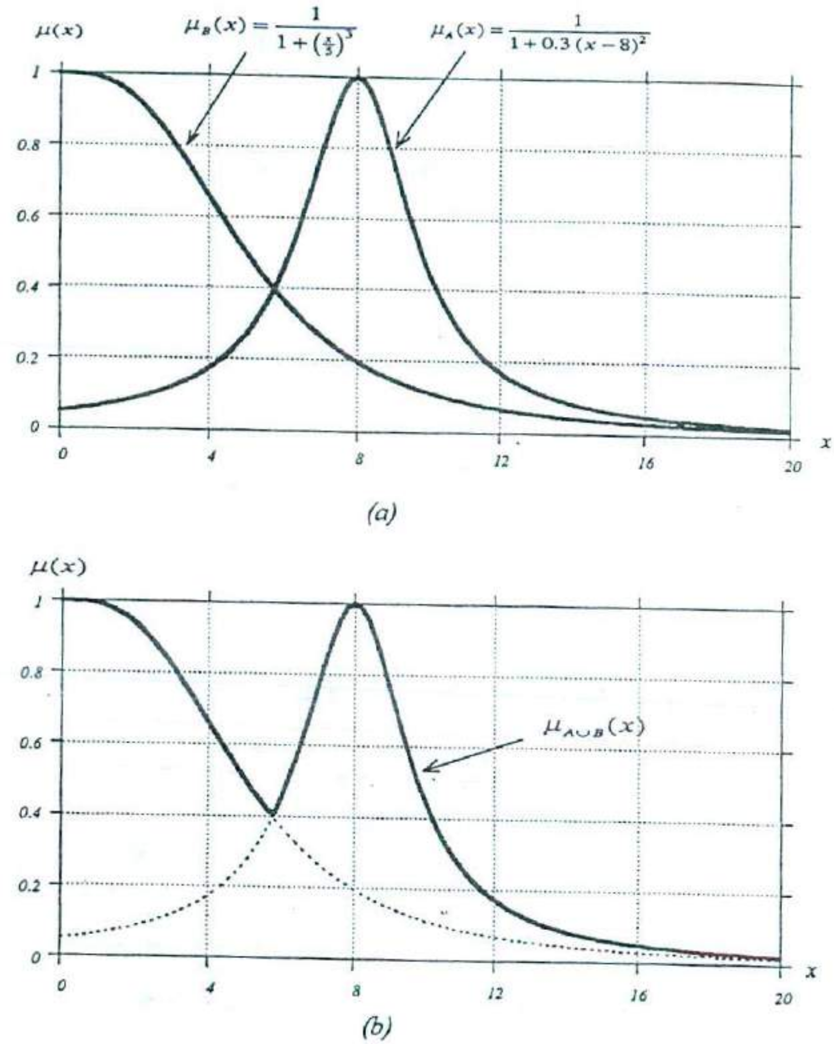


Figure 2.2 Zadeh diagram for (*a*) fuzzy sets *A* and *B* and (*b*) their *union* in Example 2.1.

Fuzzy set A may be thought of as defining the set of numbers "*about* 8," and fuzzy set B may be thought of as defining "*small numbers*." We take numbers between 0 and 20 to be the universe of discourse and, would like to find the *union* and *intersection* of A and B and the *complement* of B.

The membership function of the *union* of fuzzy sets A and B is the maximum grade of membership of each element x of the universe of discourse to either A or B in accordance with equation (2.3-13). Figure 2.2b shows the membership function of the *union* $A \cup B$. The interpretation of $A \cup B$ is "*about* 8 *OR small number.*" Similarly the membership function of the *intersection* of fuzzy sets A and B, shown in Figure 2.3a, represents the new fuzzy set "*about* 8 *AND small number.*" We observe that although the union of A and B is a *normal* fuzzy set, the intersection shown in Figure 2.3a is not, because fuzzy set $A \cap B$ has no point in the universe of

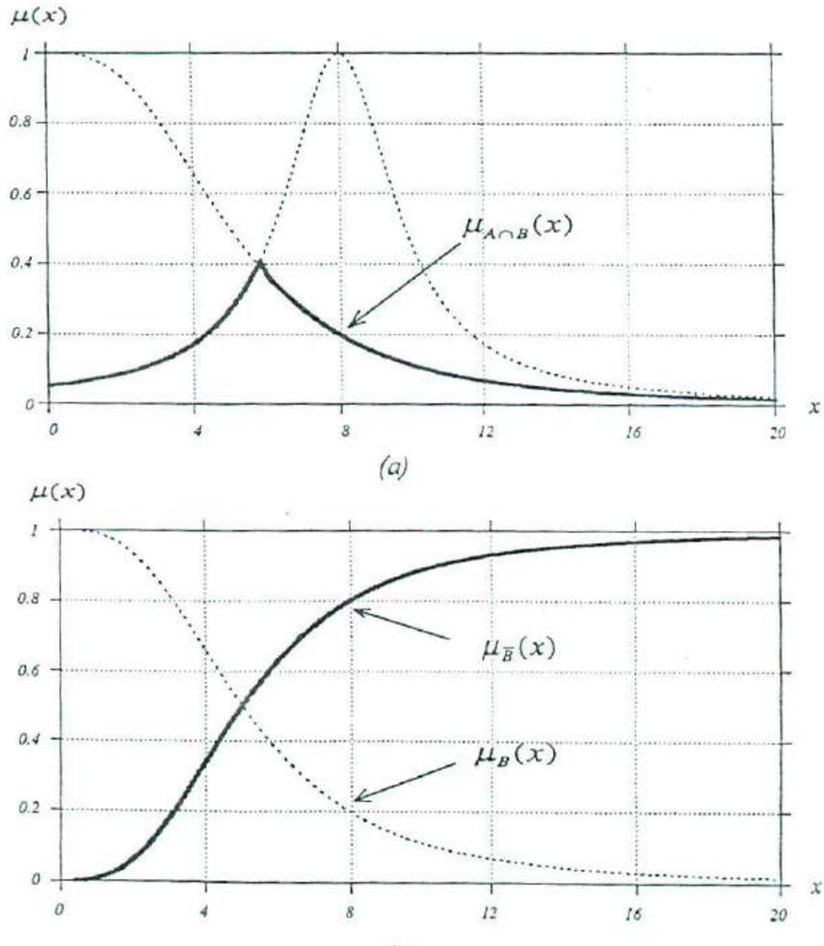


Figure 2.3 Zadeh diagram for (*a*) the *intersection* of fuzzy sets *A* and *B* and (*b*) the *complement* of *B* in Example 2.1.

discourse with grade of membership equal to 1. The complement of fuzzy set B is a new fuzzy set with membership function given by equation (2.3-15). Figure 2.3*b* shows the membership function of the *complement* $\overline{B}$. The complement $\overline{B}$ represents the logical negation (*NOT*) of B—that is, the set "*NOT small numbers.*" □ [7]

Concentration

The *concentration* of a fuzzy set A defined over a universe of discourse, X, is denoted as $CON(A)$ and it is a new fuzzy set with membership function given by

$$\mu_{CON(A)}(x) \equiv (\mu_A(x))^2 \tag{2.3-19}$$

As we said in the previous paragraph, squaring or *concentrating* a fuzzy set is equivalent to linguistically modifying it by the term *VERY*. Figure 2.4 shows the concentration operation applied to the fuzzy set B = {*small numbers*}. The membership function of the new fuzzy set $CON(B) = B^2$ = {*VERY small numbers*} is

$$\mu_{CON(B)}(x) = (\mu_B(x))^2 = \frac{1}{\left[1 + \left(\frac{x}{5}\right)^3\right]^2}$$

Dilation

The *dilation* of a fuzzy set A, denoted as $DIL(A)$, produces a new fuzzy set in X, with membership function defined as the square root of the membership function of A—that is,

$$\mu_{DIL(A)}(x) \equiv \sqrt{\mu_A(x)} \tag{2.3-20}$$

Dilation (*DIL*) and *concentration* (*CON*) are operations with opposing effects. Concentrating a fuzzy set reduces its fuzziness while dilating it increases its fuzziness. The *dilation* operation corresponds to linguistically modifying the meaning of a fuzzy set by the term "*MORE OR LESS.*" Figure 2.4 shows the dilation of B = {*small numbers*}, resulting in a new fuzzy set $DIL(B) = B^{1/2}$ = {*MORE OR LESS small numbers*}.

[7] Here and throughout this book, the end of an example is indicated by the symbol "□."

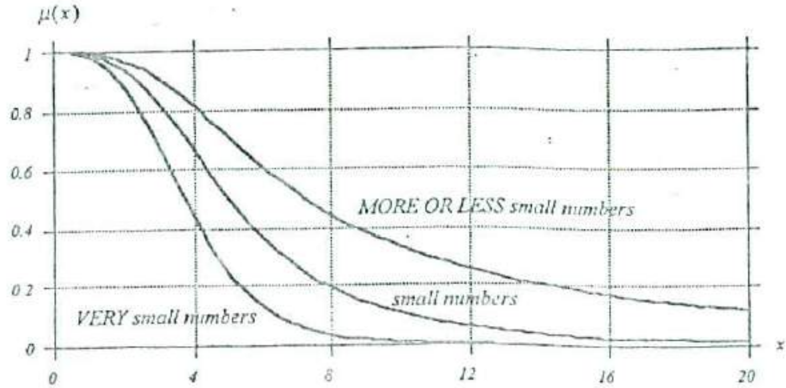


Figure 2.4 The fuzzy sets *VERY small numbers* and *MORE OR LESS small numbers* obtained by *concentrating* and *dilating* the fuzzy set *small numbers*.

Contrast Intensification

In certain applications it is desirable to control the *fuzziness* of a fuzzy set A by modifying the contrast between low and high grades of membership. For instance, we may want to increase the membership function on that part of A where membership values are higher than 0.5, and decrease it for values lower than 0.5. We define the *contrast intensification of A* as

$$\mu_{INT(A)}(x) \equiv 2[\mu_A(x)]^2 \qquad \text{for} \quad 0 \le \mu_A(x) \le 0.5$$
$$\mu_{INT(A)}(x) \equiv 1 - 2[1 - \mu_A(x)]^2, \qquad \text{for } 0.5 \le \mu_A(x) \le 1.0 \qquad (2.3\text{-}21)$$

Contrast intensification may be repeatedly applied to a fuzzy set. In the extreme, when the maximum possible contrast is achieved we no longer have a fuzzy set. We are back to a crisp set. The opposite effect—that is, going from a crisp set to fuzzy set—may be achieved through *fuzzification*.

Fuzzyfication

Fuzzification is used to transform a crisp set into a fuzzy set or simply to increase the fuzziness of a fuzzy set. For fuzzification we use a *fuzzyfier function* F that controls the fuzziness of a set. F may be one or more simple parameters. For instance, consider the fuzzy set A that describes *large* *numbers*. We define it (subjectively) through the membership function

$$\mu_{large\ numbers}(x) = \frac{1}{1 + \left(\frac{x}{F_2}\right)^{-F_1}} \tag{2.3-22}$$

where x is any positive real number. The membership function in equation (2.3-22) has two fuzzifying parameters: an *exponential fuzzyfier*, F_1, and a *denominational fuzzyfier*, F_2. Through them the fuzzy set $A = \{large\ numbers\}$ can be written as

$$A \equiv \int_X \left[\frac{1}{1 + \left(\frac{x}{F_2}\right)^{-F_1}}\right] / x \tag{2.3-23}$$

The membership function inside the brackets of equation (2.3-23) can be adjusted when needed in order to better represent the meaning of the term *large numbers*. Consider the case when we fix the value of denominational fuzzifier as $F_2 = 50$ and vary the exponential fuzzyfier F_1. The result is a family of fuzzy sets with decreasing fuzziness as F_1 increases. Figure 2.5 shows membership functions that result from such a variation. Note that when F_1 becomes very large, the set A appears almost like a crisp set. The effect of varying the denominational fuzzyfier F_2 while keeping the exponen-

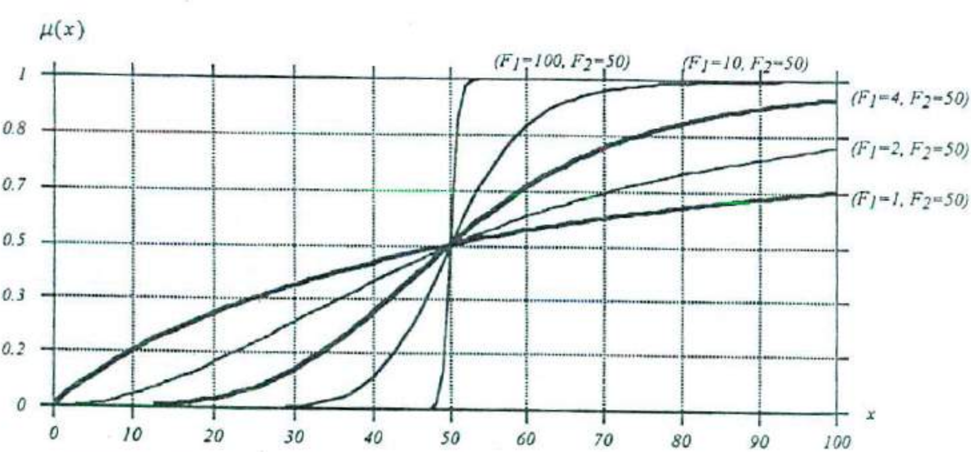


Figure 2.5 The effect of varying the exponential fuzzifier F_1 while keeping the denominational fuzzifier F_2 constant in fuzzifying the set A.

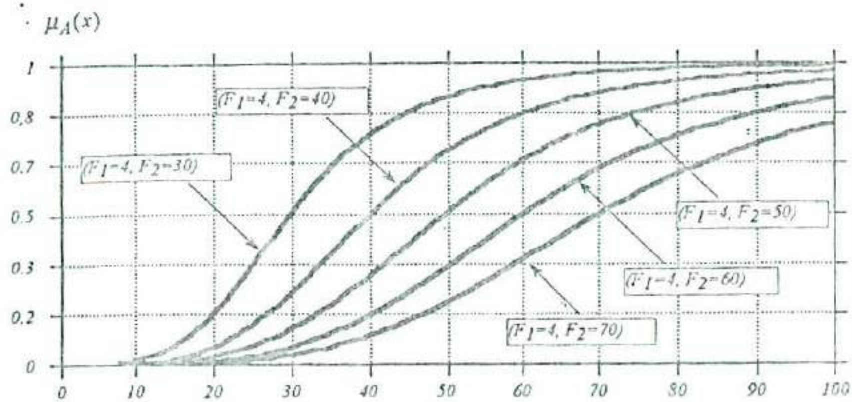


Figure 2.6 The effect of varying the denominational fuzzifier F_2 while keeping the exponential fuzzifier F_1 constant in fuzzifying the set A.

tial fuzzyfier at $F_1 = 4$ is shown in Figure 2.6. Varying F_2 results primarily in translating the membership function left and right, and to a lesser extent it affects the fuzziness of A. Such fuzzifiers are often used in fuzzy pattern recognition and image analysis in defining, for instance, the meaning of the words *vertical*, *horizontal*, and *oblique lines* (Pal and Majumder, 1986).

Fuzzification may be used more systematically by associating a *fuzzyfier F* with another function, namely a *fuzzy kernel*, $K(x)$, which is the fuzzy set that results from the application of F to a singleton x. This is often done in control applications where the input to an on-line control or diagnostic system comes from sensors and is therefore crisp, usually a real number. In order to use it in fuzzy algorithms (see Chapters 5 and 6), it is often necessary to convert a crisp number to a fuzzy set, a step known as *fuzzification*. As a result of the application of K to a fuzzy set A, we have

$$F(A;K) = \int_X \mu_A(x) \cdot \mu_{K(x)}(x)/x \qquad (2.3\text{-}24)$$

where $F(A;K)$ is a fuzzy set that results from changing the fuzziness of A in accordance with K. The *fuzzy kernel* $K(x)$ is simply a fuzzy set imposed on a singleton. It functions as a "mask" that covers the singleton to produce a fuzzy set. For example, suppose that we have the universe of discourse $X = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ and a fuzzy kernel $K(x)$ that centers a triangular fuzzy set around 5 given by

$$K(5) = 0.33/3 + 0.67/4 + 1.0/5 + 0.67/6 + 0.33/7 \qquad (2.3\text{-}25)$$

with all other elements of the universe of discourse having trivial (zero) membership. Now suppose that we have the value of 3, which may be a crisp measurement taken at a certain time. We write it as a singleton A given by

$$A = \mu_A(3)/3 = 1.0/3 \qquad (2.3\text{-}26)$$

We fuzzify A using equation (2.3-24) as follows:

$$\begin{aligned} F(A;K) &= \int_X \mu_A(x) \cdot \mu_{K(x)}(x)/x \\ &= \int_X \left[\mu_A(3) \cdot \mu_{K(3)}(x) \right]/x \\ &= 0.33/1 + 0.67/2 + 1.0/3 + 0.67/4 + 0.33/5 \end{aligned} \qquad (2.3\text{-}27)$$

which results in shifting the fuzzy kernel of (2.3-25) so that its peak is located at the singleton '3'. In other words, the effect of equations (2.3-27) is to *mask* the crisp value '3' by the fuzzy set $K(5)$, shifting its peak from '5' to '3'.

2.4 PROPERTIES OF FUZZY SETS

Fuzzy set properties are useful in performing operations involving membership functions. The properties we list here are valid for crisp and fuzzy sets as well, but some of them are specific to fuzzy sets only; more detailed treatment of properties may be found in Dubois and Prade (1980) and in Klir and Folger (1988). Consider sets A, B, C defined over a common universe of discourse X. We indicate the complement of a set by a bar over it. The following properties are true:

Double Negation Law: $$\overline{(\overline{A})} = A \qquad (2.4\text{-}1)$$

Idempotency: $$A \cup A = A \qquad A \cap A = A \qquad (2.4\text{-}2)$$

Commutativity: $$A \cap B = B \cap A \qquad A \cup B = B \cup A \qquad (2.4\text{-}3)$$

Associative Property: $$(A \cup B) \cup C = A \cup (B \cup C) \qquad (A \cap B) \cap C = A \cap (B \cap C) \qquad (2.4\text{-}4)$$

Distributive Property: $$A \cup (B \cap C) = (A \cup B) \cap (A \cup C) \qquad A \cap (B \cup C) = (A \cap B) \cup (A \cap C) \qquad (2.4\text{-}5)$$

Absorption: $$A \cap (A \cup B) = A \qquad A \cup (A \cap B) = A \qquad (2.4\text{-}6)$$

De Morgan's Laws: $$\overline{A \cup B} = \overline{A} \cap \overline{B} \qquad \overline{A \cap B} = \overline{A} \cup \overline{B} \qquad (2.4\text{-}7)$$

In fuzzy sets all these properties can be expressed using the membership function of the sets involved and the definitions of *union*, *intersection*, and *complement*. For example, consider the *associative property* given by equations (2.4-4). In terms of membership functions the associative property is written as

$$(\mu_A(x) \vee \mu_B(x)) \vee \mu_C(x) = \mu_A(x) \vee (\mu_B(x) \vee \mu_C(x))$$
$$(\mu_A(x) \wedge \mu_B(x)) \wedge \mu_C(x) = \mu_A(x) \wedge (\mu_B(x) \wedge \mu_C(x))$$

Similarly, the *distributive property*, equations (2.4-5), in terms of membership functions is written as

$$\mu_A(x) \vee (\mu_B(x) \wedge \mu_C(x)) = (\mu_A(x) \vee \mu_B(x)) \wedge (\mu_A(x) \vee \mu_C(x))$$
$$\mu_A(x) \wedge (\mu_B(x) \vee \mu_C(x)) = (\mu_A(x) \wedge \mu_B(x)) \vee (\mu_A(x) \wedge \mu_C(x))$$

De Morgan's law, equation (2.4-7), is written as

$$\overline{\mu_A(x) \vee \mu_B(x)} = \mu_{\overline{A}}(x) \wedge \mu_{\overline{B}}(x)$$

where the bar over the membership functions indicates that we take the complement. *De Morgan's law* says that the intersection of the complement of two fuzzy sets equals the complement of their union; in terms of membership functions, this is the same as saying that the minimum of two membership functions equals the complement of their maximum. There are also some properties generally not valid for fuzzy sets (although valid in crisp sets), such as the *law of contradiction*,

$$A \cap \overline{A} \neq \varnothing \tag{2.4-8}$$

and the *law of the excluded middle*,

$$A \cup \overline{A} \neq X \tag{2.4-9}$$

The law of the excluded middle in crisp sets states that the union of a set with its complement results in the universe of discourse. This is generally not true in fuzzy sets. A property unique to fuzzy sets is

$$A \cap \varnothing = \varnothing \tag{2.4-10}$$

Equation (2.4-10) says that the intersection of a fuzzy set with the empty set —that is, a set with a membership function equal to zero everywhere on the universe of discourse—is also the empty set. In terms of membership functions equation (2.4-10) is written as

$$\mu_A(x) \wedge 0 = 0$$

Also, the union of a fuzzy set A with the empty set, $\varnothing$, is A itself; that is, $A \cup \varnothing = A$ or, equivalently, $\mu_A(x) \vee 0 = \mu_A(x)$. The intersection of a fuzzy set A with the universe of discourse is the fuzzy set A itself; that is, $A \cap X = A$ or, equivalently, $\mu_A(x) \wedge 1 = \mu_A(x)$. The union of a fuzzy set A with the universe of discourse X is the universe of discourse; that is, $A \cup X = X$, which, in terms of the membership function, is written as $\mu_A(x) \vee 1 = 1$. The universe of discourse may be viewed as a fuzzy set whose membership function equals 1 everywhere; that is, $\mu_X(x) = 1$ for all x in X.

2.5 THE EXTENSION PRINCIPLE

While fuzzification operations such as the ones we saw in Section 2.3 are useful for fuzzifying individual sets or singletons, more general mathematical expressions may also be fuzzified when the quantities they involve are fuzzyfied. For example, the output of arithmetic operations when their arguments are fuzzy sets becomes also a fuzzy quantity. The *extension principle* is a mathematical tool for extending crisp mathematical notions and operations to the milieu of fuzziness. It provides the theoretical warranty that fuzzifying the parameters or arguments of a function results in computable fuzzy sets. It is an important principle, and we will use it on several occasions, particularly in conjunction with fuzzy relations (Chapter 3) and fuzzy arithmetic (Chapter 4). We give here an informal heuristic description of the extension principle; detailed formulations may be found in (Zadeh (1975), and in Dubois and Prade (1980).

Suppose that we have a function f that maps elements $x_1, x_2, \ldots, x_n$ of a universe of discourse X to another universe of discourse Y—that is,

$$\begin{aligned} y_1 &= f(x_1) \\ y_2 &= f(x_2) \\ &\cdots \\ y_n &= f(x_n) \end{aligned} \tag{2.5-1}$$

Now suppose that we have a fuzzy set A defined on $x_1, x_2, x_3, \ldots, x_n$ (the input to the function f). A is given by

$$A = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \cdots + \mu_A(x_n)/x_n \tag{2.5-2}$$

We then ask the question, If the input to our function f becomes fuzzy—for example, the set A of equation (2.5-2)—what happens to the output? Is the output also fuzzy? In other words, is there an output fuzzy set B that can be computed by inputting A to f. Well, the extension principle tells us that there is indeed such an output fuzzy set B and that it is given by

$$B = f(A) = \mu_A(x_1)/f(x_1) + \mu_A(x_2)/f(x_2) + \cdots + \mu_A(x_n)/f(x_n) \tag{2.5-3}$$

where every single image of x_i under f—that is, $y_i = f(x_i)$—becomes fuzzy to a degree $\mu_A(x_i)$. Recalling that functions are generally *many-to-one* mappings, it is conceivable that several x's may map to the same y. Thus for a certain y_0 we may have more than one x: Let us say that both x_2 and x_{13} in (2.5-1) are mapping to y_0. Hence, we have to decide which of the two membership values, $\mu_A(x_2)$ or $\mu_A(x_{13})$, we should take as the membership value of y_0. The extension principle says that the *maximum* of the membership values of these elements in the fuzzy set A ought to be chosen as the grade of membership of y_0 to the set B—that is,

$$\mu_B(y_0) = \mu_A(x_2) \vee \mu_A(x_{13}) \tag{2.5-4}$$

If, on the other hand, no element x in X is mapped to y_0—that is, no inverse image of y_0 exists—then the membership value of the set B at y_0 is zero. Having accounted for these two special cases (many x's mapping to the same y and no inverse image for a certain y), we can compute the set B—that is, the grades of membership of elements y in Y produced by the mapping $f(A)$—using equation (2.5-3).

In a more general case where we have several variables, $u, v, \ldots, w$, from different universes of discourse $U, V, \ldots, W$ and m different fuzzy sets $A_1, A_2, \ldots, A_m$ defined on the product space $U \times V \times \cdots \times W$, the multivariable function, $y = f(u, v, \ldots, w)$, may also be used to fuzzify the space Y through the extension principle. In this case, the grade of membership of any y equals the minimum of the membership values of $u, v, \ldots, w$ in $A_1, A_2, \ldots, A_m$, respectively. The membership function of B is given by

$$\mu_B(y) = \int_{U\times V\times \cdots \times Y} \left[\mu_{A_1}(u) \wedge \mu_{A_2}(v) \wedge \cdots \wedge \mu_{A_m}(w) \right] / f(u, v, \ldots, w) \tag{2.5-5}$$

where there is also a max ($\vee$) operation implicit in the union operation [the integral sign in equation (2.5-5) indicates a union ($\vee$) operation]. The max operation is performed over all $u, v, \ldots, w$ such that $y = f(u, v, \ldots, w)$. This is indicated by the union over the product space $U \times V \times \cdots \times W$ of all the universes on which the m-tuples $u, v, \ldots, w$ are defined under the integral sign. If the inverse image does not exist, then the membership function is simply zero.

In many engineering applications, the interpretation of numerical data may not be precisely known. We consider this type of data to be fuzzy. Using the extension principle, it is quite possible to adapt ordinary algorithms, which are used with precise data, to the case where the data are fuzzy. Example 2.2 is a mathematical illustration of the extension principle.

Example 2.2 Using the Extension Principle. As an illustration of how the extension principle may be used, consider the function f that maps points from the x axis to y axis in the Cartesian plane according to the equation

$$y = f(x) = \sqrt{1 - \frac{x^2}{4}} \tag{E2.2-1}$$

Figure 2.7*a* shows the function y of equation (E2.2-1). It is the upper half of an ellipse located on the center of the plane with major axis, $a = 2$, and minor axis (height), $b = 1$. The general equation of the ellipse shown in Figure 2.7*a* is

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{E2.2-2}$$

In our case with $a = 2$ and $b = 1$, equation (E2.2-2) becomes

$$\frac{x^2}{4} + y^2 = 1 \tag{E2.2-3}$$

Equation (E2.2-1) is one of the two solutions of equation (E2.2-3).

Now suppose that we define a fuzzy set A on X as shown in Figure 2.7*b*: We fuzzify the x's of equation (E2.2-1) by specifying a grade of membership $\mu_A(x)$ for each x to fuzzy set A—that is, $\mu_A(x) = \frac{1}{2}|x|$ and

$$A = \int_{-2 \le x \le 2} \left[\tfrac{1}{2}|x|\right] / x \tag{E2.2-4}$$

where $|x|$ is the absolute value of x, and we limit the support of A between -2 and $+2$ as indicated by the limits under the *integration sign* (*union*) of equation (E2.2-4).

Having the x values fuzzyfied by the fuzzy set A, we want to know the effect of fuzzification on y. The extension principle tells us that the fuzziness of A will be extended to y as well. In other words, we will have a fuzzy set B on Y derived by equations (2.5-3) or (2.5-5). To avoid the case where more than one x will map to the same y, we consider first the function f in the first quadrant of the plane (where both x and y are positive). Later we will look at the entire function. The fuzzy set, B, defined on Y is

$$B = f(A) = \int_Y \mu_B(y)/y \tag{E2.2-5}$$

We need to find $\mu_B(y)$ in equation (E2.2-5). In terms of the membership function of A and according to the extension principle, equation (2.5-3), the

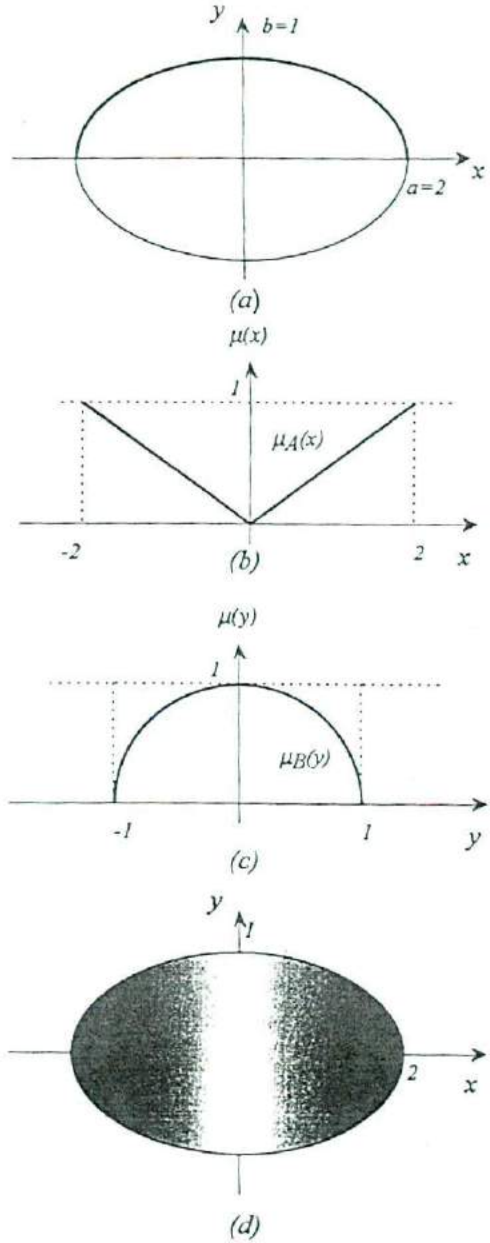


Figure 2.7 Graphs for Example 2.2. (*a*) The function *y*, which is the upper part of the ellipse shown. (*b*) The membership function of the set *A*. (*c*) The membership function of *B*. (*d*) Fuzzifying the interior of the ellipse.

set B will be

$$B = f(A) = \int_Y \mu_A(x)/f(x) \qquad \text{(E2.2-6)}$$

Of course we want to transform the x variable to y in equation (E2.2-6) since the union (integration) is formed with respect to Y, the universe of discourse for B. We use equation (E2.2-1) to solve for x:

$$x = 2\sqrt{1 - y^2} \qquad \text{(E2.2-7)}$$

Then we substitute (E2.2-7) in (E2.2-6), noting that $f(x) = y$ and that $\mu_A(x)$ is given by (E2.2-4). Thus we obtain the fuzzy set B:

$$B = \int_{0 \le y \le 1} \sqrt{1 - y^2} \Big/ y \qquad \text{(E2.2-8)}$$

Now if we consider negative values for x as well, we would have to take the maximum of the membership value of A at (x) and $(-x)$ in accordance with equation (2.5-5). Due to the symmetry of the problem these values are actually the same and therefore B is still as derived in (E2.2-8). The membership function of B is,

$$\mu_B(y) = \sqrt{1 - y^2} \qquad \text{(E2.2-9)}$$

as shown in Figure 2.7*c*. Figure 2.7*d* shows the geometric interpretation of fuzzyfying the interior of the ellipse in accordance with the fuzzy sets A and B above. The result is a kind of fuzzy elliptic region, strongest near the x axis and particularly at its $x = \pm 2$ sides and weakest near the origin and the $y = \pm 1$ sides. □

2.6 ALPHA-CUTS

With any fuzzy set A we can associate a collection of crisp sets known as *α-cuts* (*alpha-cuts*) or *level sets* of A. An α-cut is a crisp set consisting of elements of A which belong to the fuzzy set at least to a degree α. As we shall see in the next section, α-cuts offer a method for resolving any fuzzy set in terms of constituent crisp sets (something analogous to resolving a vector into its components). In Chapter 4 we will see that α-cuts are indispensable in performing arithmetic operations with fuzzy sets that represent various qualities of numerical data. It should be noted that α-cuts are crisp, *not* fuzzy, sets.[8]

[8] Formally, a distinction is made between two types of α-cuts, the *strong* and the *weak* α-cut (Dubois and Prade, 1980). We use the weak α-cut, simply calling it α-cut.

The α-cut of a fuzzy set A denoted as A_α is the crisp set comprised of all the elements x of a universe of discourse X for which the membership function of A is *greater than or equal to* α; that is,

$$A_\alpha = \{x \in X \mid \mu_A(x) \geq \alpha\} \tag{2.6-1}$$

where α is a parameter in the range $0 < \alpha \leq 1$; the vertical bar "|" in equation (2.6-1) is shorthand for "such that."

Consider, for example, a fuzzy set A with trapezoidal membership function as shown in Figure 2.8. The 0.5-cut of A is simply the part of its support where its membership function is greater than 0.5. In Figure 2.8 we can see the 0.5-cut of A. Reflecting the fact that the α-cut is a *crisp set*, its membership function appears like a characteristic function. As another example consider the set A of *small integers* given by

$$A = 1.0/1 + 1.0/2 + 0.75/3 + 0.5/4 + 0.3/5 + 0.3/6 + 0.1/7 + 0.1/8$$

The 0.5-cut of A is simply the crisp set $A_{0.5} = \{1, 2, 3, 4\}$.

In the next section we will see that α-cuts provide a useful way both for resolving a membership function in terms of constituent crisp sets as well as for synthesizing a membership function out of crisp sets.

A fuzzy set can have an extensive support since its membership function can be zero or nearly zero, or very small. In order to deal with situations where small degrees of membership are not worthy of consideration, *level*

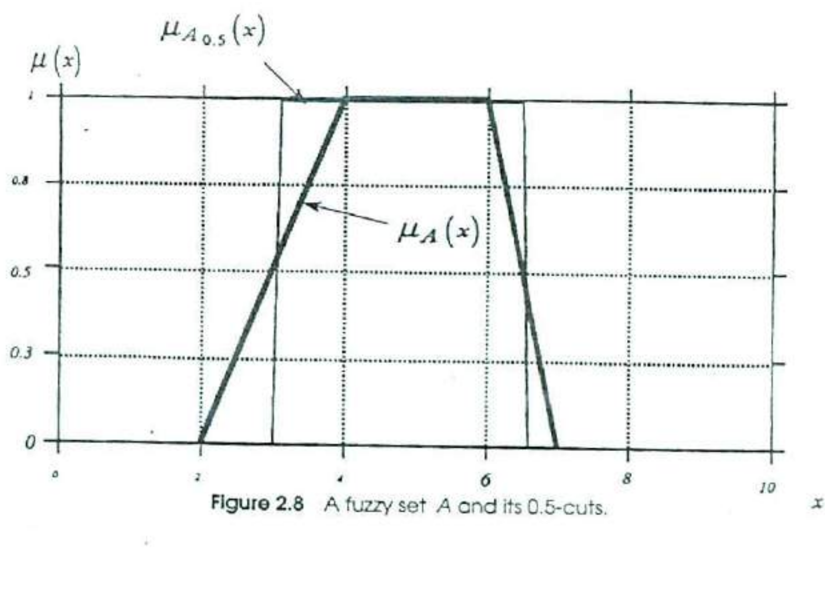


Figure 2.8 A fuzzy set A and its 0.5-cuts.

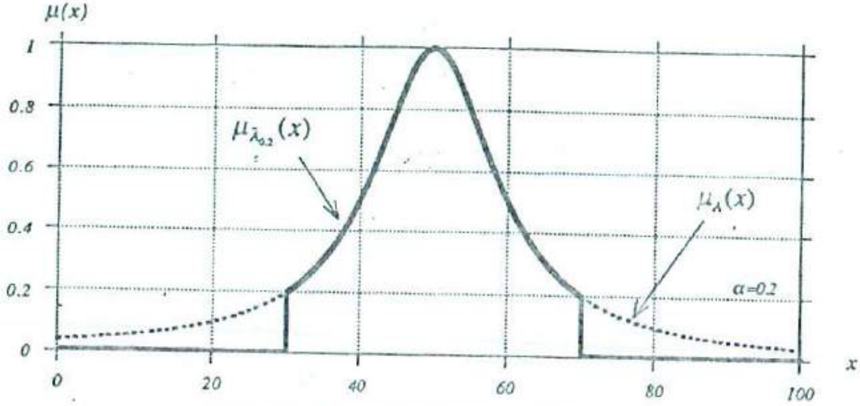


Figure 2.9 The 0.2-level fuzzy set of fuzzy set A.

fuzzy sets were introduced to exclude undesirable grades of membership (Radecki, 1977). We define the level fuzzy sets of a fuzzy set A as fuzzy sets $\bar{A}_\alpha$ whose membership values are greater than α, where $0 < \alpha < 1$. Formally

$$\bar{A}_\alpha \equiv \{(x, \mu_A(x)) | x \in A_\alpha\} \tag{2.6-2}$$

where A_α is the α-cut of A. Equation (2.6-2) indicates that for a given α we have a level fuzzy set which is the part of A that has membership greater than α. Let us consider, for example, a fuzzy set A whose membership function is

$$\mu_A(x) = \frac{1}{1 + 0.01(x - 50)^2} \tag{2.6-3}$$

as shown in Figure 2.9 (dotted curve). Suppose that we are not interested in the part of the support that has membership less than 0.2. We obtain the 0.2-level fuzzy set of A by chopping the part of the membership function which is less than 0.2 as shown in the figure. Its membership function $\mu_{\bar{A}_{0.2}}(x)$ is shown by the solid curve. It is the same as $\mu_A(x)$ between $x = 30$ and $x = 70$ and zero everywhere else. Level fuzzy sets should not be confused with level sets, which is a synonym for α-cuts. Level fuzzy sets are indeed fuzzy sets, whereas α-cuts are crisp sets. They provide a useful way of considering fuzzy sets in the significant part of their support, and hence they save on computing time and storage requirements.

2.7 THE RESOLUTION PRINCIPLE

There are several ways of representing fuzzy sets, and we have already seen a few of them. They all involve two things: identifying a suitable universe of discourse and defining membership functions. One way to represent a fuzzy set would be to list all the elements of the universe of discourse together with the grade of membership of each element (omitting the possibly infinite elements that have zero membership). Alternatively, we can just provide an analytical representation of the membership function. The *resolution principle* offers another way of representing membership to a fuzzy set, namely through its α-cuts. It asserts that the membership function of a fuzzy set A can be expressed in terms of its α-cuts as follows:

$$\mu_A(x) = \bigvee_{0<\alpha\leq 1} \left[\alpha \cdot \mu_{A_\alpha}(x)\right] \tag{2.7-1}$$

where the maximum is taken over all α's. Equation (2.7-1) indicates that the membership function of A is the union (notice the max operator) of all α-cuts, after each one of them has been multiplied by α.

Consider, for example, the fuzzy set A with triangular membership function shown in Figure 2.10. Several α-cuts of A, each multiplied by α, are also shown. Knowing many α's and the α-cuts of A, we can form their products and put them together (in the sense of taking their union) to approximate the function. For example, we multiply the 0.25-cut by 0.25 to get the 0.25-cut pushed down to 0.25, and similarly we multiply the 0.5-cut by 0.5, the 0.75-cut by 0.75, and so on. When put together we have an approximation of the membership function of A as shown in Figure 2.10.

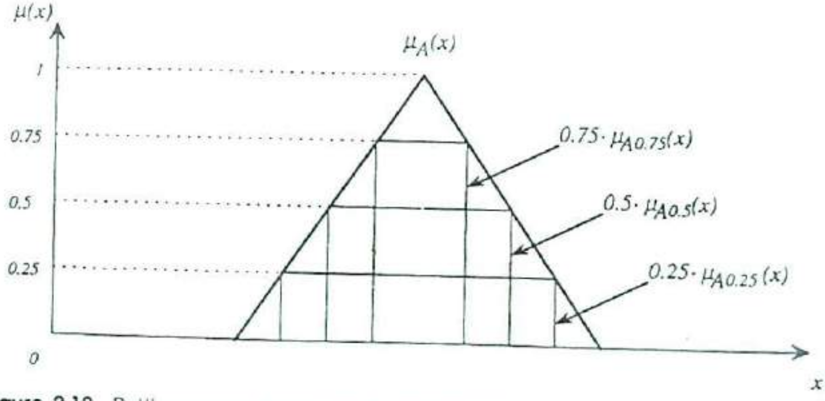


Figure 2.10 Putting many α-cuts of A multiplied by α together approximates the membership function of A.

Thus, a large enough family of α-cuts provides another way of representing a fuzzy set. Although we often know the membership function exactly, in some applications only α-cuts are known and out of them we need to approximate the membership function (see Chapter 4).

2.8 POSSIBILITY THEORY AND FUZZY PROBABILITIES

In the late 1970s Zadeh advanced a theoretical framework for information and knowledge analysis, called *possibility theory*, emphasizing the quantification of the semantic, context-dependent nature of symbols—that is, *meaning* rather than measures of information. The theory of possibility is analogous, and yet conceptually different from the theory of probability. Probability is fundamentally a measure of the frequency of occurrence of an event. Although there are several interpretations of probability (*subjectivistic*, *axiomatic*, and *frequentistic*), probabilities generally have a physical event basis. They are tied to statistical experiments and are primarily useful for quantifying how frequently a sample occurs in a population. Possibility theory, on the other hand, attempts to quantify how accurately a sample resembles an *ideal* element of a population. The ideal element is a prototypical class or a category of the population which we think of as a fuzzy set. In a sense, possibility theory may be viewed as a generalization of the theory of probability with the *consistency principle*, which we will see later on, providing a heuristic connection between the two. Possibility theory focuses more on the *imprecision* intrinsic in language, whereas probability theory focuses more on events that are *uncertain* in the sense of being random in nature. In natural language processing, automatic speech recognition, knowledge-based diagnosis, image analysis, robotics, analysis of rare events, information retrieval, and related areas, major problems are encountered on quantifying the meaning of events—that is, the efficacious and accurate interpretation of their significance and consequence and not the extent of their occurrence. Let us illustrate with a simple example.

In the field of reliability analysis, probabilistic methods have been the basic instrument for quantifying equipment and human reliability as well. Two very important concepts used are the *failure rate* and the *error rate*. Knowing the failure rate of a component amounts to knowing the duration of time that the component may be trusted to operate safely, and thus a schedule for replacement and maintenance activities can be devised. It is not unusual, however, that after a component is fixed or replaced, the entire system breaks down, a problem particularly acute with electronic components. Indeed, such general failures sometimes cause extremely negative consequences, leading to catastrophic accidents. The problem here is that failure rates are not sufficiently meaningful to account for the complex interactions that a human being, such as a maintenance technician or an operator, may have with a machine. In addition, the correct estimation of failure rate and error rate requires a large amount of data, which is often not practically possible to obtain. Is is obviously impractical to melt nuclear reactors to collect failure rate data. Thus, in practice, the failure rate and error rate are estimated by experts based on their engineering judgment (Onisawa, 1990); from this point of view, fuzzy possibilities and probabilities (which we will examine momentarily) can be used to model such judgments in a flexible and efficient way. Engineering judgment enters many areas of systems and reliability analysis including estimating the effect of environmental factors, operator stress, dependence between functions or units, selection of sequence of events, expressing the degree of uncertainty involved in the formulation of safety criteria, assuming parameter ranges, and so on (Shinohara, 1976). Alternatives to failure and error rates have been developed employing the notion of possibility measures, called *failure* and *error possibilities*, and have been applied to the reliability analysis of nuclear power plants, structural damage assessments, and earthquake engineering. Failure possibilities and error possibilities are essentially fuzzy sets on the interval [0, 1] that employ the notions we examine in this section.

Over the years, two views, or schools of thought, of the definition of fuzziness have emerged. The first view, which we implicitly held in the previous sections, has to do with categorizing or grouping the elements of a universe of discourse into classes or sets whose boundaries of membership are fuzzy. Thus when we defined the set of *small numbers* in Example 2.1 we identified a category of numbers within the universe of all numbers. Implicitly, what we dealt with in the example was the problem of *imprecision*. Our main problem was to find the membership function that most appropriately or accurately described the category of *small numbers*. The other view of fuzziness has to do with the problem of *uncertainty*. Here our main concern is to quantify the certainty of an assertion such as "a number x is a *small number*," where x is an element of the universe of discourse X of numbers (whose location on X is not known in advance) and is therefore called a *nonlocated element*. Possibility theory was advanced in order to address this type of problem. Possibility is more generally known as a *fuzzy measure*, which is a function assigning a value between 0 and 1 to each crisp set of the universe of discourse, signifying the degree of evidence or belief that a particular element belongs to the set. Other types of fuzzy measure are *belief measures*, *plausibility measures*, *necessity measures*, and *probability measures*. The theory of fuzzy measures was advanced in 1974 by Sugeno as part of his Ph.D. dissertation at Tokyo University. Fuzzy measures subsume probability measures as well as belief and plausibility measures used in what is known as the *Dempster–Shafer Theory of Evidence*.

Let us now take a closer look at *possibility*. Possibility is a *fuzzy measure*, which means that possibility is a function with a value between 0 and 1, indicating the degree of evidence or belief that a certain element x belongs to a set (Zadeh, 1978; Dubois and Prade, 1988). A possibility of 0.3 for element x, for example, may indicate a 0.3 degree of evidence or belief that

x belongs to a certain set. How this belief is distributed to elements other that x is quantified through a *possibility distribution*. In possibility theory, the concept of *possibility distribution* is analogous to the notion of *probability distribution* in probability theory. A possibility distribution is viewed as a fuzzy restriction acting as an elastic constraint on the values that may be assigned to a variable. What does this mean? Well, it is best to review the notion of a *variable*, first. Let A be a crisp set defined on a universe X and let V be a variable taking values on some element x of X, a situation illustrated in Figure 2.11. The crisp set A is what in the parlance of probability we call an *event*. Events are comprised out of one or more basic events. Thus, the element x may be thought of as a basic event. If x is within A and x occurs, then we say that the event A has occurred as well. For example, in reliability analysis, equipment failure and human error are considered to be events whose occurrence is based on the occurrence of basic events known as *initiating events*. To say that *V takes its values in A* is to indicate that any element (basic event) of event A could *possibly* be a value of V and that any element outside of A, the complement of A, cannot be a value of V. Thus, the statement *V takes its value in A* can be viewed as inducing a possibility, Π over X, associating with each value x the possibility that x is a value of V. This can be written as

$$\Pi(V := x) = \pi_V(x) = \chi_A(x) = \begin{cases} 1 & \text{if} \quad x \in A \\ 0 & \text{if} \quad x \notin A \end{cases} \tag{2.8-1}$$

where " := " is an assignment symbol indicating that x is assigned to the

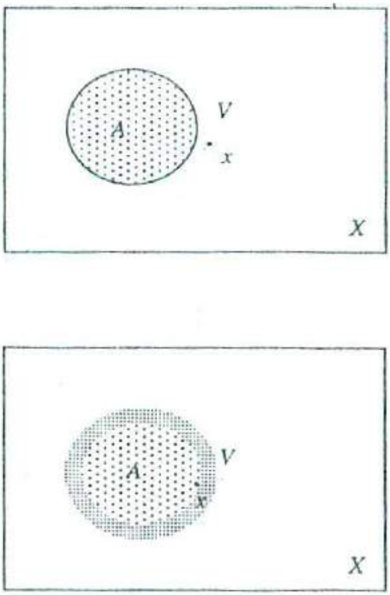


Figure 2.11 The statement about a variable V, "*V takes its values in A*," has a different meaning when the set A is crisp (*top*) than when the set A is fuzzy (*bottom*).

variable V, and $\pi_V(x)$ is the *possibility distribution* associated with V (or the *possibility distribution function* of Π). In equation (2.8-1), $\chi_A(x)$ is the characteristic function of A (see Section 2.1). Mathematically, Π is considered a *measure* which is a special function mapping the universe to the interval [0, 1]. Knowing that the values that V may take are members of A is the same as knowing which values of the universe X are restricted to be values of V and which are restricted not to be values of V. We indicated this in equation (2.8-1) by using the *characteristic function* of the crisp set A. We think of the crisp set A as a *restriction* on the values of the variable V, and in view of the nonfuzzy nature of A this type of restriction is called a *crisp restriction*.

Next, suppose that A is a fuzzy set and that its boundary no longer crisp (i.e., does not sharply divide members from nonmembers) but is instead a fuzzy boundary allowing an element x to be a member of A to some degree. As with any fuzzy set, A is uniquely identified by its membership function $\mu_A(x)$. In terms of events we think of A as a fuzzy event, and we can associate with each basic event x a membership function indicating its membership to A. Let us again consider a variable V whose arguments are elements of X[9]. Now suppose that V is constrained to take values on X. The fuzzy set A also restricts the possible values that the variable V may take, but in a fuzzy manner—that is, to a degree. In such a case we consider the fuzzy set A to act as a *fuzzy restriction* on the possible values of V. Generalizing equation (2.8-1) to the fuzzy case we say that the fuzzy set A induces a possibility Π. The associated possibility distribution $\pi_V(x)$ on the values that V may assume is defined to be equal to the membership function of A, $\mu_A(x)$ and is written as

$$\Pi(V := x) = \pi_V(x) = \mu_A(x) \tag{2.8-2}$$

Thus, the possibility that V is assigned x—that is, $V := x$, which is sometimes indicated as "V is x"—is postulated to be equal to the membership function of A evaluated at x—that is, $\mu_A(x)$. It is important to observe in equation (2.8-2) that *possibility distributions* are fuzzy sets, while *possibilities* are just numbers between 0 and 1. The possibility Π in (2.8.2) is a measure of the compatibility of a given crisp value x that V may take with an *a priori* defined set A. In this way, V becomes a variable associated with the *possibility distribution* $\pi_V(x)$ in much the same way as a random variable is associated with the probability distribution.

What equation (2.8-2) indicates is that in certain situations, such as in the definition of failure and error possibilities, it is of interest to interpret the membership function $\mu_A(x)$ of a fuzzy set as a *possibility distribution* of a variable V. In this sense the fuzzy set A is viewed as the set of more or less possible values for V.

[9] In Chapter 5 the variable V will be generalized to a *fuzzy variable*, which is a variable that takes fuzzy sets as values.

Given a possibility distribution $\pi_V(x)$, the possibility that x may belong to another crisp set B is defined as

$$\Pi(V \subset B) = \bigvee_{x \in B} \pi_V(x) \tag{2.8-3}$$

What equation (2.8-3) indicates is that the possibility of B is the possibility of the most possible elementary event x of B. Generalizing this relationship, it can be shown (Dubois and Prade, 1988; Kandel, 1986) that the possibility measure of the union of two crisp sets B and C is the maximum of the possibilities of B and C and can be written as

$$\Pi(B \cup C) = \Pi(B) \vee \Pi(C) \tag{2.8-4}$$

Given a fuzzy set A and a possibility distribution function, $\pi_V(x)$, the possibility of A, denoted as $\Pi(A)$, is given by

$$\Pi(A) = \bigvee_{x \in X} [\mu_A(x) \wedge \pi_V(x)] \tag{2.8-5}$$

Consider two fuzzy events A and B defined over the universe of discourse X. The possibility of A with respect to B is defined as

$$\Pi(A|B) = \bigvee_{x \in X} [\mu_A(x) \wedge \mu_B(x)] \tag{2.8-6}$$

The possibility measure of A with respect to B reflects the extent to which A and B coincide or overlap. Thus, possibility may be viewed as a measure of comparison of fuzzy sets.

Conditional possibilities have been defined in analogy with conditional probabilities; an entire body of theoretical results has been achieved, known generally as *possibility theory*. It is finding an increasing number of applications in the fields of knowledge representation and applied artificial intelligence (Ragheb and Tsoukalas, 1988). A very comprehensive treatment of possibility may be found in the book entitled *Possibility Theory* by Dubois and Prade (1988). The theory of possibility has assumed particular significance in the field of natural language processing due to the inherent fuzziness of natural language. In the late 1970s Zadeh constructed a universal language called *PRUF*, in which the translation of a proposition expressed in natural language takes the form of a procedure for computing the possibility distribution of a set of fuzzy relations in a database. The procedure, then, may be interpreted as a semantic computation transforming the meaning of a proposition to a computed possibility distribution quantifying the information conveyed by the proposition (Zadeh, 1983).

There are certain differences between probability and possibility measures worth pointing out. Possibility measures are "softer" than probability measures, and the interpretation of probability and possibility is quite different. Probability is used to quantify the frequency of occurrence of an event, while possibility (along with fuzzy tools) is used to quantify the *meaning* of an event. Consider the following example offered by Zadeh (1978). Suppose that we have the proposition "*Hans ate V eggs for breakfast,*" where $V = \{1, 2, 3, \ldots\}$. A *possibility distribution* and a *probability distribution* may be associated with V, as shown in the following table:

x	1	2	3	4	5	6	7	8	9
$\pi_V(x)$	1	1	1	1	0.8	0.6	0.4	0.2	0.1
$p_V(x)$	0.1	0.8	0.1	0	0	0	0	0	0

The possibility distribution is interpreted as the *degree of ease* with which Hans can eat x eggs, while the probability distribution might have been determined by observing Hans at breakfast for 100 days. Note that the probability distribution function $p_V(x)$ is given a *frequentistic* interpretation and that it sums to '1', while the possibility distribution function $\pi_V(x)$ is imputed with a *situation* or *context-dependent* interpretation and does not have to sum to '1'.

Possibility is an upper bound for probability: A high degree of possibility does not imply a higher degree of probability. If, however, an event is not possible, it is not also probable. This is referred to as the *probability/possibility consistency principle* (Zadeh, 1978). This heuristic principle is useful for drawing a distinction between the *objectivistic* use of probability measures and the *subjectivist* use of possibility or fuzzy measures. When we attempt to use the two to describe a similar thing, we can use the *possibility/probability consistency principle* as a guide. Possibility measures are more flexible measures useful for epistemic (i.e., cognitive) or context-dependent descriptions. In general, according to Zadeh a variable may be associated with both a possibility distribution and a probability distribution, with the weak connection between the two given by the consistency principle (Zadeh, 1978).

In the language of probability theory the set A in Figure 2.11 may be viewed as a *fuzzy event*. Such a fuzzy event induces a distribution on the values of a variable which we called the possibility distribution function and defined in equation (2.8-2). We can also define the *probability of a fuzzy event A*. Suppose that a fuzzy event A is comprised of elementary events x, and with each x we associate a basic probability $p(x)$.

Zadeh defined the *probability of fuzzy event A* as the mathematical expectation (the first moment) of its membership function, that is,

$$P(A) = \frac{\int_X \mu_A(x) p(x)\, dx}{\int_X p(x)\, dx} \tag{2.8-7}$$

where A is a fuzzy event on the universe X, x is an element of X, also called an elementary event, and $p(x)$ is a probability distribution (Zadeh, 1968). When A is not a fuzzy event, equation (2.8-7) reduces back to the usual crisp probability $P(A)$. In equation (2.8-7) we assume that the probability measure on the entire universe of discourse must equal unity—that is, $\int_X p(x)\,dx = 1$.

In addition, given equation (2.8-7) we can define a *fuzzy mean* as

$$m_A = \frac{1}{P(A)} \int_X x \mu_A(x) p(x)\,dx \tag{2.8-8}$$

and a *fuzzy variance* as

$$\sigma_A^2 = \frac{1}{P(A)} \int_X (x - m_A)^2 \mu_A(x) p(x)\,dx \tag{2.8-9}$$

The probability of a fuzzy event as defined in equation (2.8-7) has been an extremely useful notion with wide application in the field of quantification theory (Terano et al., 1992). Quantification methods are useful in analyzing data involving human judgments which are not normally given numerical expression, as well as in interpreting and understanding such data.

Example 2.3 Possibility Measures and Distributions. Let us illustrate the distinction between *possibility measure* or *possibility* and *possibility distribution*. We consider a possibility distribution induced by the proposition "*V is a small integer*" where the possibility distribution is (subjectively) defined as

$$\pi_V(x) = 1.0/1 + 1.0/2 + 0.75/3 + 0.5/4 + 0.3/5 + 0.3/6 + 0.1/7 + 0.1/8 \tag{E2.3-1}$$

We also consider the crisp set $A = \{3, 4, 5\}$ which we can write as

$$A = \sum_{x \in X} \mu_A(x)/x = 1/3 + 1/4 + 1/5 \tag{E2.3-2}$$

What is the possibility of A? The *possibility measure* $\Pi(A)$ is found using equation (2.8-5); that is,

$$\Pi(A) = \bigvee_{x \in X} [\mu_A(x) \wedge \pi_V(x)] \tag{E2.3-3}$$

Using equations (E2.3-1) and (E2.3-2) in (E2.3-3), we can obtain the possibility of A:

$$\Pi(A) = 0.75 \vee 0.5 \vee 0.3 = 0.75 \tag{E2.3-4}$$

For another fuzzy set B = {*integers that are not small*} given by

$$B = 0.2/3 + 0.3/4 + 0.6/5 + 0.8/6 + 1.0/7$$

using equation (E2.3-3), we could obtain that the possibility of B is

$$\Pi(B) = 0.2 \vee 0.3 \vee 0.3 \vee 0.3 \vee 0.1 = 0.3 \qquad \text{(E2.3-5)}$$

It should be noted in equations (E2.3-4) and (E2.3-5) that the possibility is simply a number between 0 and 1, whereas the possibility distribution is a fuzzy set—for example, equation (E2.3-1).

Let us now consider a simple instance of how to generate the possibility distribution itself. Let $C = 1/1 + 1/2 + 0.8/3 + 0.6/4 + 0.4/5 + 0.2/6$ be a fuzzy set that represents *small numbers*. Then the proposition "*V is a small number*" associates with V the possibility distribution, $\pi_V(x)$, taken in view of equation (2.8-2) to be equal to the membership function of C—that is,

$$\pi_V(x) = 1/1 + 1/2 + 0.8/3 + 0.6/4 + 0.4/5 + 0.2/6 \quad \text{(E2.3-6)}$$

In equation (E2.3-6) a singleton such as 0.6/4 indicates that the possibility that x is 4, given that x is a *small integer*, is 0.6. □

REFERENCES

Dubois, D., and Prade, H., *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, Boston, 1980.

Dubois, D., and Prade, H., *Possibility Theory*, Plenum Press, New York, 1988.

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

Kaufmann, A., *Introduction to the Theory of Fuzzy Subsets*, Volume I, Academic Press, New York, 1975.

Klir, G. J., and Folger, T. A., *Fuzzy Sets, Uncertainty, and Information*, Prentice-Hall, Englewood Cliffs, NJ, 1988.

Kosko, B., *Neural Networks and Fuzzy Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1992.

Onisawa, T., An Application of Fuzzy Concepts to Modelling of Reliability Analysis, *Fuzzy Sets and Systems*, No. 37, pp. 267–286, North-Holland, Amsterdam, 1990.

Pal, S. K., and Majumder, D. K. D., *Fuzzy Mathematical Approach to Pattern Recognition*, John Wiley & Sons, New York, 1986.

Radecki, T., Level Fuzzy Sets, *Journal of Cybernetics*, Vol. 7, pp. 189–198, 1977.

Ragheb, M., and Tsoukalas, L. H., Monitoring Performance of Devices Using a Coupled Probability–Possibility Method, *International Journal of Expert Systems*, Vol. 1, pp. 111–130, 1988.

Shinohara, Y., Fuzzy Set Concepts for Risk Assessment, *International Institute for Applied Systems Analysis*, Report WP-76-2, Laxenburg, Austria, January 1976.

Terano, T., Asai, K., and Sugeno, M., *Fuzzy Systems Theory and Its Applications*, Academic Press, Boston, 1992.

Zadeh, L. A., Fuzzy Sets, *Information and Control*, Vol. 8, pp. 338–353, 1965.

Zadeh, L. A., Probability Measure of Fuzzy Events, *Journal of Mathematical Analysis and Applications*, Vol. 23, pp. 421–427, 1968.

Zadeh, L. A., Outline of a New Approach to the Analysis of Complex Systems and Decision Processes, *IEEE Transactions on Systems, Man and Cybernetics*, SMC-3, 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 as a Basis for Theory of Possibility, *Fuzzy Sets and Systems*, Vol. 1, pp. 3–28, 1978.

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. What happens to the curves in Figure 2.5 if we set $F_2 = 40$ and vary F_1 as in the figure?

2. In Figure 2.6, what is the significance of the intersection between the $\mu_A = 0.5$ line and the curves?

3. In Example 2.2, substitute $y = \sin x$ for equation (E2.2-1) and utilize the extension principal in the same way as in the example. Choose an appropriate range for x and assume any additional information needed as in the example.

4. The fuzzy variable of Figure 2.9 is given by the equation $\mu_A(x) = 1/[1 + 0.3(x - 50)^2]$. Show that the 0.2 level fuzzy set of fuzzy set A can be represented by α-cuts using the resolution principal.

5. The fuzzy sets A and B are given by

$$A = 0.33/6 + 0.67/7 + 1.00/8 + 0.67/9 + 0.33/10$$
$$B = 0.20/3 + 0.60/4 + 1.00/5 + 0.60/6 + 0.20/7$$

   (a) Write an expression for $A \vee B$.

   (b) Write an expression for $A \wedge B$.

6. Different fuzzy symbols are often used to mean similar things.

   (a) Write all symbols or terms that have the same general meaning as *max* ($\vee$).

   (b) Write all symbols or terms that have the same general meaning as *min* ($\wedge$).

7. Given fuzzy set A, describing *pressure p is higher than* 15 *mPa*, through the membership function:

$$\mu_A(x) = \frac{1}{1+(x-15)^{-2}} \quad x > 15,$$
$$= 0 \quad x \le 15,$$

and fuzzy set B, describing *pressure p is approximately equal to* 17 *mPa*, with membership function:

$$\mu_B(x) = \frac{1}{1+(x-17)^4}.$$

Find the membership function of the fuzzy set C, describing *pressure p is higher than* 15 *mPa and approximately equal to* 17 *mPa*. Use at least four different norms for interpreting *AND* (see Appendix) and draw all membership functions.

8. Using the data given in Problem 7, find the membership function of the fuzzy set D, describing *pressure p is higher than* 15 *mPa or approximately equal to* 17 *mPa*. Use at least four different norms for interpreting *OR* (see Appendix) and draw all membership functions.

9. Using the data given in Problem 7, find the membership function of the fuzzy set E, describing *pressure p is not higher than* 15 *mPa and approximately equal to* 17 *mPa*. Use four different norms for interpreting *AND* (see Appendix) and draw all membership functions.

10. Determine all α-cuts for the following fuzzy sets, given that $\alpha = 0.0, 0.1, 0.2, \ldots 0.9, 1.0$.

    I. $A = 0.1/3 + 0.2/4 + 0.3/5 + 0.4/6 + 0.5/7 + 0.6/8 + 0.7/9 + 0.8/10 + 1.0/11 + 0.8/12$

    II. $B = \int_{-\infty<x<+\infty} \left[\frac{1}{1+(x-15)^{-2}}\right] / x$

    Write a MATLAB program that takes a number of α-cuts (minimum 10) and reconstructs the membership function.

11. Let $X = N \times N$, and the fuzzy sets:

$$\mu_A(x) = \frac{1}{1+10(x-2)^2}$$

$$\mu_B(y) = \frac{1}{1+2y^2}$$

Let the mappings $z = f(x, y)$, $f\colon N \times N \to N$ be the following quadric surfaces

(a) $$z = \sqrt{\frac{x^2}{4} + \frac{y^2}{2}}, \; x \in A, \; y \in B.$$

(b) $$\frac{x^2}{9} + \frac{y^2}{15} - \frac{z^2}{8} = 1$$

(c) $$2y^2 + 12z^2 = x^2$$

Sketch the surfaces and determine the image $f(A \times B)$ by the extension principle, for each of the above.

3

FUZZY RELATIONS

3.1 INTRODUCTION

In fuzzy approaches, *relations* possess the computational potency and significance that *functions* possess in conventional approaches. Fuzzy *if/then* rules and their aggregations, known as *fuzzy algorithms*, both of central importance in engineering applications, are fuzzy relations in linguistic disguise. Fuzzy relations may be thought of as fuzzy sets defined over high-dimensional universes of discourse. As the name indicates, a relation implies the presence of an association between elements of different sets. If the degree of association is either 0 or 1, we have *crisp relations*. If the degree of association is between 0 and 1, we have *fuzzy relations*; a number between 0 and 1 is taken to indicate partial absence or presence of association. In this chapter we begin by reviewing crisp relations and various ways for representing them. Next, we look at fuzzy relations and properties used to classify them, and finally we come to *composition of fuzzy relations*, a very important tool for approximate reasoning with applications in the fields of expert systems, control, and diagnosis.

On what basis do we associate various elements in a relation? The association may be due to a common property, a quality, a reference, a condition, or a rule, satisfied by pairs of elements (e.g., objects, numbers, words, variables, etc.). For example, the statements "***is greater than***" or "***is a component of***" indicate an association between two elements. The order of the elements is important. For instance, if the relation "***is a component of***" holds for the pair of elements (*u-tube, steam-generator*)—that is, if the statement "*u-tube* ***is a component of*** *steam-generator*" is true—the relation may no longer be true when the elements are interchanged. The relation "*steam* *generator* ***is a component of*** *u-tube*" is not true. Thus, this is an important point to observe: In relations, *order* is important!

A relation such as "***is a component of***" may also be expressed as an *if/then* rule. We can say "*if an object is a u-tube, then it is a component of a steam generator*." Any ambiguity as to what degree an object is known to be a *u-tube* or a *steam generator*, or any ambiguity as to the degree of truth in such an association, results in a fuzzy relation.

When two elements belong to a relation R, we refer to them as an *ordered pair* denoted as $(a, b) \in R$, or aRb, with element a being distinguished as the first element and b as the second. With two elements in association, we have *binary relations*. With three elements wë have *tertiary relations*, and when n elements are in association we have *n-ary relations*. An association of n elements in an *n-ary* relation is called *n-tuple*. A relation is any *set* of ordered *n-tuples*. The keyword here is "*set*." Relations are formed out of sets of elements, and they are sets themselves.

Crisp relations are defined over the *Cartesian product* or *product space* of two or more sets. The *Cartesian product* $X \times Y$ of two sets X and Y is the set of all ordered pairs (x, y) with x in X and y in Y. The product $X \times X$ is often abbreviated as X^2, the product $X^2 \times X$ as X^3, and so on.

We saw that relations are sets where order is important. But relations may also be thought of as *mappings*, with the process of association in mathematics being called a *mapping*. *Functions* are *mappings* as well. Relations, however, are a more general type of mapping. A function performs what is called a *many-to-one mapping*; that is, many elements are associated with one (and only one) element but not vice versa. For example, if the mapping is done between x's and y's in the $X \times Y$ plane, we may have more than one x mapped to the same y but not the other way around. Relations, however, perform *many-to-many* mappings. Many x's can be associated with a single y and vice versa. Many y's can also be associated with a single x. The importance of this abstract-sounding distinction in terms of engineering and computational applications cannot possibly be overstated, as we will see in later chapters. But for the moment let us turn our attention to an example of a crisp relation in order to see some of the ways that relations may be represented.

Example 3.1 A Crisp Relation. Let us consider a *divisibility relation*, R_d, on the set $S = \{1, 2, 3, 4, 6\}$ defined by the statement "*x* ***divides*** *y*." R_d is a *binary* relation because it involves two elements, x and y, drawn from the Cartesian product of the set S with itself—that is, $S \times S$. Furthermore, it is a *crisp* relation since a number either divides another number or not (assuming integer division only). It is easy to list all the pairs of the relation and to see that the relation itself is a set, namely, the crisp set of all the pairs

$$R_d = \{(1,1), (1,2), (1,3), (1,4), (1,6), (2,2), (2,4), (2,6), (3,3), (3,6), (4,4), (6,6)\} \qquad \text{(E3.1-1)}$$

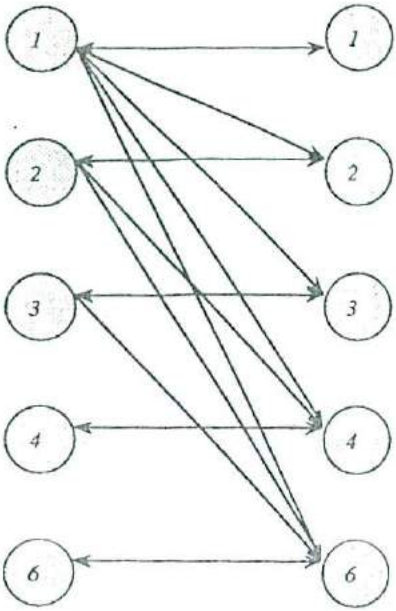


Figure 3.1 The directed graph of the divisibility relation R_d defined on the Cartesian product $S \times S$ of the set $S = \{1, 2, 3, 4, 6\}$.

where the meaning of the elements inside the parentheses is "*1 divides 1*," and so on. The relation R_d can also be represented through a graph as shown in Figure 3.1. The individual elements are represented by circles, called the *vertices* of the graph. If R_d is true for two elements, we connect them by an arrow, with the direction of the arrow indicating the order of the elements in the relation. For example, given that 3 divides 6, there is an arrow going from 3 to 6; and since 6 does not divide 3, there is no arrow going from 6 to 3. Reflecting the fact that the order of elements or the directions of the arrows is important, we call this a *directed graph.*

The binary relation R_d may also be represented by a table or a matrix. Table 3.1 shows the tabular representation of R_d. When a table entry is 1, it indicates that x (row entry) divides the corresponding y (column entry); for example, in the fourth row and fourth column we simply have that the element 4 divides itself. A 0 indicates the absence of such a relation. Should the divisibility relation have been a fuzzy relation, the table entries would be numbers between 0 and 1 as we will see later on.

R_d can also be represented by a matrix obtained from Table 3.1 by removing the column of x's on the side and the row of y's from the top; that is,

$$R_d = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{(E3.1-2)}$$

Table 3.1 A tabular representation of the divisibility relation in Example 3.1

R_d:

x \ y	1	2	3	4	6
1	1	1	1	1	1
2	0	1	0	1	1
3	0	0	1	0	1
4	0	0	0	1	0
6	0	0	0	0	1

Thus we have seen five different ways for representing R_d:

1. *Linguistically*, through the statement "*x divides y*"
2. By listing the *set* of all ordered pairs as in equation (E3.1-1)
3. As a *directed graph* (Figure 3.2)
4. As a *table* (Table 3.1)
5. As a *matrix*, equation (E3.1-2)

It should be noted that the last two ways are generally convenient only for *binary* relations. For *tertiary* relations, for example, we would need a three-dimensional table or matrix (for *n-ary* relations *n*-dimensional tables and matrices), and therefore tables and matrices may be conveniently used only with *binary* relations. □

3.2 FUZZY RELATIONS

In fuzzy relations we consider *pairs* of elements, and more generally *n-tuples*, that are related to *a degree*. Just as the question of whether some element belongs to a set may be considered a matter of degree, whether some elements are associated may also be a matter of degree (Zadeh, 1971; Dubois and Prade, 1980). For example, suppose we have a diagnosis problem involving vibration data with a set of faults $F = \{f_1, \ldots, f_n\}$ associated to a set of symptoms $S = \{s_1, \dots, s_m\}$. First we need to establish how symptoms relate to faults—that is, establish a relation from F to S. One of these symptoms, let's say s_i, may be "excessive vibration." Knowing whether a machine vibrates depends on the interpretation of vibration data. If the concept of "excessive vibration" has been crisply defined—that is, it can be readily determined whether the machine vibrates and we can associate a symptom s_i with a fault f_j—we have a crisp relation from F to S. In reality, however, it may be rather difficult to crisply define such associations and hence all faults $F = \{f_1, \dots, f_n\}$ and all symptoms $S = \{s_1, \dots, s_m\}$ may be associated to a degree, giving us a fuzzy relation from F to S. What is important in such cases is to compute these degrees. Having established the fuzzy relation from F to S, we can subsequently use it to identify the highest degrees of association given a symptom s_i so that it may be linked to faults f_k, f_j, and so forth (Kaufmann, 1975).

Fuzzy relations are fuzzy sets defined on Cartesian products. Whereas the fuzzy sets we encountered in the previous chapter were defined on a single universe of discourse (e.g., X), fuzzy relations are defined on higher-dimensional universes of discourse (e.g., $X \times X$ or $X \times Y \times Z$). A Cartesian product for us is simply a higher-dimensional universe of discourse. Suppose that we have a binary fuzzy relation R defined on $X \times Y$. As with any fuzzy set, we can list all pairs of the relation explicitly as we did in equation (2.2-2); that is,

$$R = \{((x, y), \mu_R(x, y))\} \tag{3.2-1}$$

where every individual pair (x, y) belongs to the Cartesian product $X \times Y$. Alternatively, we can use the notation of equation (2.2-3) to form the union of all $\mu_R(x, y)/(x, y)$ singletons of $X \times Y$. For a discrete Cartesian product we would have

$$R = \sum_{(x_i, y_j) \in X \times Y} \mu_R(x_i, y_j)/(x_i, y_j) \tag{3.2-2}$$

while for a continuous Cartesian product we have

$$R = \int_{X \times Y} \mu_R(x, y)/(x, y) \tag{3.2-3}$$

The same notation is used for any *n-ary fuzzy relation*.

So much for the fuzzy set nature of fuzzy relations and notation. Let us now take a look at alternative ways of representing them. One of them, which is particularly useful for the composition of relations (see Section 3.5), is to form a matrix of grades of membership in a manner analogous to (E3.1-2), only now we have instead of 0's and 1's various numbers between 0 and 1.

The membership matrix of an $n \times m$ binary fuzzy relation has the general form

$$R = \begin{bmatrix} \mu_R(x_1, y_1) & \mu_R(x_1, y_2) & \cdots & \mu_R(x_1, y_n) \\ \mu_R(x_2, y_1) & \mu_R(x_2, y_2) & \cdots & \mu_R(x_2, y_n) \\ & \vdots & & \\ \mu_R(x_m, y_1) & \mu_R(x_m, y_2) & \cdots & \mu_R(x_m, y_n) \end{bmatrix} \qquad (3.2\text{-}4)$$

Let us take a look at some special relations and their membership matrices. The *identity fuzzy relation*, R_I, is a special type of relation which has 1 in all diagonal elements and 0 in all off-diagonal elements—that is,

$$R_I = \begin{bmatrix} 1 & 0 & & 0 \\ 0 & 1 & & 0 \\ & & \ddots & \\ 0 & 0 & & 1 \end{bmatrix} \qquad (3.2\text{-}5)$$

Another special relation is the *universe relation*, R_E, namely a relation with 1 everywhere in its membership matrix—that is,

$$R_E = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & & 1 \\ \vdots & & \ddots & \\ 1 & 1 & & 1 \end{bmatrix} \qquad (3.2\text{-}6)$$

The *null relation*, R_0, has a membership matrix with 0 everywhere—that is,

$$R_0 = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & & 0 \\ \vdots & & \ddots & \\ 0 & 0 & & 0 \end{bmatrix} \qquad (3.2\text{-}7)$$

The transpose of a membership matrix gives the membership matrix of the *inverse relation* of R denoted by R^{-1} and defined by

$$\mu_{R^{-1}}(y, x) \equiv \mu_R(x, y) \qquad (3.2\text{-}8)$$

Thus the *inverse* of the relation represented by the matrix of equation (3.2-4) has the membership matrix

$$R^{-1} = \begin{bmatrix} \mu_R(x_1, y_1) & \mu_R(x_2, y_1) & \cdots & \mu_R(x_m, y_1) \\ \mu_R(x_1, y_2) & \mu_R(x_2, y_2) & \cdots & \mu_R(x_m, y_2) \\ & \vdots & & \\ \mu_R(x_1, y_n) & \mu_R(x_2, y_n) & \cdots & \mu_R(x_m, y_n) \end{bmatrix} \qquad (3.2\text{-}9)$$

which is the transpose of the matrix found by interchanging the rows of R to produce the columns of R^{-1}, and the columns of R have become the rows of R^{-1} (Klir and Folger, 1988; Terano et al., 1992). The inverse of an inverse relation is the original relation just as the inverse of the inverse of a matrix is the original matrix—that is,

$$(R^{-1})^{-1} = R \tag{3.2-10}$$

So far we defined fuzzy relations on crisp Cartesian products. However, fuzzy relations can also be defined on fuzzy Cartesian products (Kandel, 1986; Klir and Folger, 1988). Although fuzzy relations defined over fuzzy sets are of interest, particularly in connection with decision making under uncertainty, we will make no actual use of them in this book. Unless otherwise indicated, fuzzy relations in this book are assumed to be defined over crisp Cartesian products.

Example 3.2 Representing a Fuzzy Relation. Let us take two discrete sets $X = \{x_1, x_2, x_3, x_4\}$ and $Y = \{y_1, y_2, y_3, y_4\}$ and define (subjectively) on their Cartesian product the fuzzy relation R = "*x is similar* to *y*," shown by the directed graph of Figure 3.2. R may be represented through the five different ways we saw in Example 3.1 with regard to crisp relations:

1. Linguistically, for example by the statement "*x is similar to y*"
2. By listing (or taking the union of) all fuzzy singletons
3. As a *directed graph* (Figure 3.2)
4. In *tabular* form
5. As a *matrix*

Let us represent the relation as a *fuzzy set* by taking the union of all singletons—that is, all ordered pairs and their membership values:

$$R = \int_{X \times Y} \mu_R(x, y)/(x, y) \tag{E3.2-1}$$

Using the data of Figure 3.2, equation (E3.2-1) gives

$$\begin{aligned} R = {} & 1.0/(x_1, y_1) + 0.3/(x_1, y_2) + 0.9/(x_1, y_3) + 0.0/(x_1, y_4) \\ & + 0.3/(x_2, y_1) + 1.0/(x_2, y_2) + 0.8/(x_2, y_3) + 1.0/(x_2, y_4) \\ & + 0.9/(x_3, y_1) + 0.8/(x_3, y_2) + 1.0/(x_3, y_3) + 0.8/(x_3, y_4) \\ & + 0.0/(x_4, y_1) + 1.0/(x_4, y_2) + 0.8/(x_4, y_3) + 1.0/(x_4, y_4) \end{aligned} \tag{E3.2-2}$$

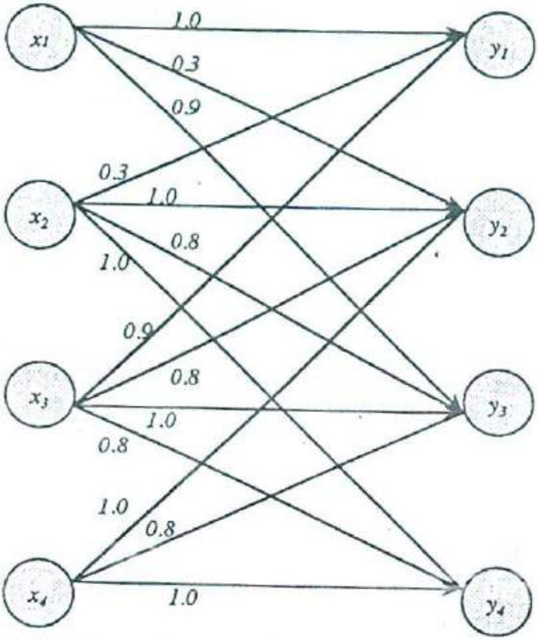


Figure 3.2 The directed graph of the fuzzy relation R in Example 3.2.

The relation R may also be represented in tabular form as

R:

	y_1	y_2	y_3	y_4
x_1	1.0	0.3	0.9	0.0
x_2	0.3	1.0	0.8	1.0
x_3	0.9	0.8	1.0	0.8
x_4	0.0	1.0	0.8	1.0

Note that compared to Table 3.1, where we only used 0's and 1's, in the tabular representation of R we find grades of membership between 0 and 1. Consider the pair (x_3, y_4). From the table of R we see that "x_3 ***is similar to*** y_4" is true to a 0.8 degree.

In matrix form, R is given by

$$R = \begin{bmatrix} 1.0 & 0.3 & 0.9 & 0.0 \\ 0.3 & 1.0 & 0.8 & 1.0 \\ 0.9 & 0.8 & 1.0 & 0.8 \\ 0.0 & 1.0 & 0.8 & 1.0 \end{bmatrix} \tag{E3.2-3}$$

The inverse of R, which we denote as R^{-1}, is the transpose of the membership matrix of equation (E3.2-3), given by

$$R^{-1} = \begin{bmatrix} 1.0 & 0.3 & 0.9 & 0.0 \\ 0.3 & 1.0 & 0.8 & 1.0 \\ 0.9 & 0.8 & 1.0 & 0.8 \\ 0.0 & 1.0 & 0.8 & 1.0 \end{bmatrix} \tag{E3.2-4}$$

Of course the inverse fuzzy relation R^{-1} in this case has the same membership matrix due to the fact that R is a symmetric relation (see next section).
□

3.3 PROPERTIES OF RELATIONS

Crisp and fuzzy relations alike are classified on the basis of the mathematical properties they possess. We present here a brief introduction to the subject of properties mostly for the sake of reference. We look first at properties of crisp relations and then examine the properties of fuzzy relations. In fuzzy relations, different properties call for different requirements for the membership function of a relation.

Let S be a Cartesian product (e.g., $S = X \times Y$, with x being an element of X and y being an element of Y) and let R be a relation on S. The relation R could have the following properties:

Reflexive. We say that a relation R is *reflexive* if for any arbitrary element x in S we have that xRx is *valid*—that is, the pair (x, x) also belongs to the relation R.

Antireflexive. A relation R is *antireflexive* if there is no x in S for which xRx is valid.

Symmetric. A relation R is *symmetric* if for all x and y in S, the following is true: If xRy holds, then yRx is valid also.

Asymmetric. A relation R is *asymmetric* if there are no elements x and y in S such that both xRy and yRx are valid.

Antisymmetric. A relation R is *antisymmetric* if for all x and y in S when xRy is valid and yRx is also valid, then $x = y$.

Transitive. A relation R is called *transitive* if the following is true for all x, y, z in S: If xRy is valid and yRz is also valid, then xRz is valid as well.

Connected. A relation R is *connected* when for all x, y in S the following is true: If $x \neq y$, then either xRy is valid or yRx is valid.

Left Unique. A relation R is called *left unique* when for all x, y, z in S the following is true: If xRz is valid and yRz is also valid, then we can infer that $x = y$.

Right Unique. A relation R is *right unique* when for all x, y, z in S the following is true: If xRy and xRz hold true, then $y = z$.

Right Biunique. A relation R which is both *left unique* and *right unique* is called *biunique*.

Relations are classified into different groups on the basis of these properties. For example, an important type of crisp relation is the so-called *equivalence relation*. An equivalence relation is a relation that is *reflexive, symmetric*, and *transitive* (Klir and Folger, 1988). Equivalence relations are found in every corner of mathematics and are particularly useful in engineering fields such as pattern recognition, measurement, and control. Other important relations are the so-called *order relations*. For example, a relation R is called a *partial ordering* if it is *reflexive, transitive*, and *antisymmetric*. If R is also *connected*, then it is called a *total linear ordering*. Order relations are very important in fuzzy arithmetic (Kaufmann and Gupta, 1991).

The properties of fuzzy relations are described in terms of various requirements for their membership function. In a pioneering paper on the subject, Zadeh (1971) showed that most of the important properties of crisp relations stated above are extended to fuzzy relations as well. Let a relation R be a fuzzy relation on the Cartesian product $S = X \times X$. *Reflexivity, symmetry*, and *transitivity* are the three most important properties that help us properly categorize fuzzy relations. R is a *reflexive* relation if for all x in X we have that

$$\mu_R(x, x) = 1 \tag{3.3-1}$$

If for at least one x in X but not for all x's, equation (3.3-1) is not true the relation R is called *irreflexive*. If equation (3.3-1) is not satisfied for any x, then R is called *antireflexive*.

A fuzzy relation R is *symmetric* if order is not important—that is, if we can interchange x's and y's. In terms of the membership function of R, this is equivalent to saying that

$$\mu_R(x, y) = \mu_R(y, x) \tag{3.3-2}$$

If equation (3.3-2) is not satisfied for some pairs (x, y), then we say that R is *antisymmetric*; if it is not satisfied for all pairs (x, y), then we say that the relation R is *asymmetric*.

A fuzzy relation R on the Cartesian product $X \times X$ is *max–min transitive* if for two pairs (x, y) and (y, z) both in $X \times X$ we have

$$\mu_R(x, y) \geq \bigvee_z [\mu_R(x, z) \wedge \mu_R(z, y)] \tag{3.3-3}$$

where all the maxima with respect to z are taken for all the mimima inside the brackets in equation (3.3-3). *Transitivity* can be defined for other operations such as *product* $(\cdot)$ instead of min $(\wedge)$ in equation (3.3-3); in such a case we have what is called *max-product transitivity*. A relation that does not satisfy equation (3.3-3) for all pairs is called *nontransitive*, and if it fails to satisfy (3.3-3) for all pairs, then it is called *antitransitive*.

A fuzzy relation that is *reflexive* and *symmetric* is called a *proximity* or *tolerance relation*. A fuzzy relation that is *reflexive*, *symmetric*, and *transitive* is called a *similarity* relation, which is the fuzzy generalization of the *equivalence* property of crisp relations (Zadeh, 1971). Similarity relations are very important in fuzzy logic, and together with proximity relations they are crucially important in the field of fuzzy diagnosis. A *fuzzy ordering* is a fuzzy transitive relation. If a fuzzy relation is *reflexive*, *transitive*, and *antisymmetric*, then we call it a *fuzzy partial ordering*. Fuzzy orderings and similarity relations may be resolved into nonfuzzy partial orderings, in a manner analogous to the way we used the resolution principle in Chapter 2. Let us now look at an example of a fuzzy similarity relation.

Example 3.3 A Similarity Relation. Consider a fuzzy relation R indicating that two points on the $X \times Y$ plane are near the origin. This is a relation we would expect to have a membership function equal to 1 exactly at the origin and to have gradually diminishing membership as we move away from the origin. We can indicate the relation by a statement such as "*x is near the origin with y*" or analytically as a fuzzy set with an appropriately chosen (subjectively) membership function—for example,

$$\mu_R(x, y) = e^{-(x^2+y^2)} \tag{E3.3-1}$$

Thus the relation R is the fuzzy set

$$R = \int_{X \times Y} \mu_R(x, y)/(x, y) \tag{E3.3-2}$$

which using equation (E3.3-1) we can write as

$$R = \int_{X \times Y} e^{-(x^2+y^2)}/(x, y) \tag{E3.3-3}$$

The membership function of R is shown in Figure 3.3. It can be shown that R is a *fuzzy similarity relation*; that is, it is *reflexive*, *symmetric*, and *transitive*.

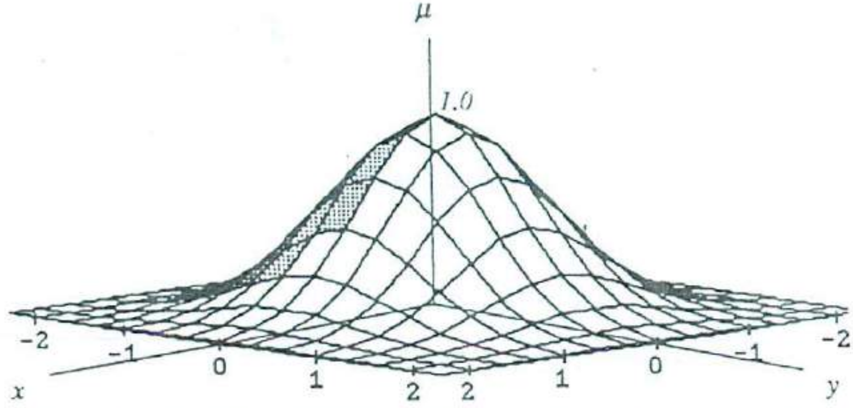


Figure 3.3 The membership function of the relation R indicating that an (x, y) point of the Cartesian plane, $X \times Y$, is close to the origin (0, 0).

Figure 3.3 also illustrates that fuzzy relations are fuzzy sets on high-dimensional universes of discourse. In this case the universe of discourse is the x–y plane—that is, the Cartesian product $X \times Y$. □

3.4 BASIC OPERATIONS WITH FUZZY RELATIONS

Fuzzy relations are fundamentally fuzzy sets defined over higher-dimensional universes of discourse—that is, Cartesian products. All the fuzzy set operations we saw in Chapter 2, such as *union*, *intersection*, *α-cuts*, and so on, are also applicable to fuzzy relations. Here we take a look at the *union*, *intersection*, *inclusion*, *α-cuts*, and *resolution* as well as some operations specific to relations such as *projection* and *cylindrical extension* (Dubois and Prades, 1980; Zimmermann, 1985).

Suppose that we have two fuzzy relations R_1 and R_2. Their *union* is a new relation

$$R_1 \cup R_2 = \int_{X \times Y} \left[\mu_{R_1}(x, y) \vee \mu_{R_2}(x, y) \right] \Big/ (x, y) \qquad (3.4\text{-}1)$$

where the membership function of $R_1 \cup R_2$, as indicated in equation (3.4-1), is

$$\mu_{R_1 \cup R_2}(x, y) \equiv \mu_{R_1}(x, y) \vee \mu_{R_2}(x, y) \qquad (3.4\text{-}2)$$

for every (x, y) pair of the Cartesian product.

The intersection of fuzzy relations R_1 and R_2 is a new fuzzy relation whose membership function is the minimum of the membership functions of R_1 and R_2 taken at every point (x, y) of the Cartesian product,

$$R_1 \cap R_2 = \int_{X \times Y} \left[\mu_{R_1}(x, y) \wedge \mu_{R_2}(x, y) \right] / (x, y) \tag{3.4-3}$$

where the membership function of $R_1 \cap R_2$ is

$$\mu_{R_1 \cap R_2}(x, y) \equiv \mu_{R_1}(x, y) \wedge \mu_{R_2}(x, y) \tag{3.4-4}$$

We define the α-cut of a fuzzy relation in a manner similar to the way we defined in Section 2.6 the α-cuts of one-dimensional fuzzy sets. The *resolution principle* applied to fuzzy relations offers us an alternative way of representing the membership function of a fuzzy relation. It says that the membership function of a fuzzy relation can be represented through its α-cuts. More specifically, the *resolution principle* asserts that the membership function of a fuzzy relation R is expressed in terms of its α-cuts in a manner analogous to equation (2.7-1) as

$$\mu_A(x) = \bigvee_{0 < \alpha \leq 1} \left[\alpha \cdot \mu_{R_\alpha}(x, y) \right] \tag{3.4-5}$$

where the maximum is taken over all α's and $\mu_{R_\alpha}(x, y)$ is the α-cut of the membership function of the relation R at level α.

We say that a relation R_1 is *included* in R_2 if both are defined over the same product space and we have everywhere

$$\mu_{R_1}(x, y) \leq \mu_{R_2}(x, y) \tag{3.4-6}$$

Note that the *union* and *intersection* of fuzzy relations are meaningful in the context of relations defined over the same Cartesian product. When the product spaces of two relations are different, these operations have no meaning and instead the important and useful operations become the various *composition operations* which we examine later.

Example 3.4 Union and Intersection of Fuzzy Relations. Suppose that we have the following two relations R_1 and R_2 described by the tables below:

$R_1 =$ "*x is larger than y*":

	y_1	y_2	y_3	y_4
x_1	0.0	0.0	0.1	0.8
x_2	0.0	0.8	0.0	0.0
x_3	0.1	0.8	1.0	0.8

R_2 = "*y is much bigger than x*":

	y_1	y_2	y_3	y_4
x_1	0.4	0.4	0.2	0.1
x_2	0.5	0.0	1.0	1.0
x_3	0.5	0.1	0.2	0.6

The union of the two relations, $R_1 \cup R_2$, is formed by taking the maximum of the two grades of membership for the corresponding elements of the two tables. The table of the new relation is as follows:

$R_1 \cup R_2$:

	y_1	y_2	y_3	y_4
x_1	0.4	0.4	0.2	0.8
x_2	0.5	0.8	1.0	1.0
x_3	0.5	0.8	1.0	0.8

For the intersection, $R_1 \cap R_2$, we take the minimum of the two grades of membership in each cell of the tables of the two relations, and the resulting table is as follows:

$R_1 \cap R_2$:

	y_1	y_2	y_3	y_4
x_1	0.0	0.0	0.1	0.1
x_2	0.0	0.0	0.0	0.0
x_3	0.4	0.1	0.2	0.6

Some caution is needed when we interpret the new relations produced by *union* and *intersection*. For example, the union $R_1 \cup R_2$ can be interpreted as a proposition of the form: "*x is quite different than y*." The intersection, however, is not very meaningful, since x cannot be simultaneously larger than y and y cannot be larger than x (Zimmermann, 1985). □

In relations, when it is desired to go to a space of lower dimension we use *projection*. Starting with a fuzzy relation defined on a two-dimensional space, we can take the *first* and *second projection* and go to one-dimensional universe of discourse, with each projection eliminating the first and second dimension, respectively. The *total projection* takes us to a zero-dimensional singleton, eliminating both dimensions. *Projections* are also called *marginal fuzzy restrictions*. The inverse of projection—that is, going toward higher dimensions—is called *cylindrical extension* (Zadeh, 1971).

Consider the fuzzy relation R defined over the Cartesian product $X \times Y$ —that is,

$$R = \int_{X \times Y} \mu_R(x, y)/(x, y) \tag{3.4-7}$$

The first projection is a fuzzy set that results by eliminating the second set Y of the Cartesian product, $X \times Y$, hence projecting the relation on the universe of discourse of the first set X. We write the first projection as

$$R^1 = \int_X \mu_{R^1}(x)/x \tag{3.4-8}$$

The membership function of the first projection is defined as

$$\mu_{R^1}(x) \equiv \bigvee_y [\mu_R(x, y)] \tag{3.4-9}$$

To obtain $\mu_{R^1}(x)$, equation (3.4-9) indicates that we take the maximum of $\mu_R(x, y)$ with respect to y. Similarly the second projection (projecting on the Y universe of discourse) is a fuzzy set:

$$R^2 = \int_Y \mu_{R^2}(y)/y \tag{3.4-10}$$

with membership function defined as

$$\mu_{R^2}(y) \equiv \bigvee_x [\mu_R(x, y)] \tag{3.4-11}$$

where we take the maximum of $\mu_R(x, y)$ with respect to x. The *total projection* of R simply identifies the peak point of the relation—that is, a singleton (x_0, y_0) where the membership function of the original relation reaches its highest value.

$$R^T = \bigvee_x \bigvee_y \mu_R(x_0, y_0)/(x_0, y_0) \tag{3.4-12}$$

The opposite of projection is called the *cylindrical extension*. Through cylindrical extension we go from a fuzzy relation defined over a lower-dimensional space to a fuzzy relation on a higher-dimensional space. If a relation R is defined on a subsequence of a product space $X = X_1 \times X_2 \times X_3 \times \cdots \times X_n$, call it $X_{i1} \times X_{i2} \times X_{i3} \times \cdots \times X_{ik}$, then the cylindrical extension of R, denoted as CE(R), is defined as

$$\mathrm{CE}(R) \equiv \int_{X_1 \times \cdots \times X_n} \mu_R(x_{i1}, \ldots, x_{ik}) / (x_1, \ldots, x_n) \qquad (3.4\text{-}13)$$

Let us look at an example of projection and cylindrical extension.

Example 3.5 Projection and Cylindrical Extension. Consider the relation R defined over the Cartesian product $X \times Y$ of the sets $X = \{x_1, x_2, x_3\}$ and $Y = \{y_1, y_2, y_3, y_4, y_5, y_6\}$ as shown in Table 3.2. The membership functions for the first and second projection are indicated by the column to the right of the table and the row below the table, respectively. The first projection is what the relation would look like if seen from the direction of the arrow on the left side of the table. Imagine that we look in the direction that the arrow on the left indicates. We see in front of us three rows of the relation and select the highest value in each row. As a result, we obtain the first projection, namely,

$$R^1 = \sum_X \mu_{R^1}(x_i)/x_i = 1.0/x_1 + 0.9/x_2 + 1.0/x_3 \qquad (\mathrm{E}3.5\text{-}1)$$

Equation (E3.5-1) indicates that the first projection of the binary fuzzy relation R is simply a fuzzy set on a one-dimensional universe of discourse.

Table 3.2 Fuzzy relation and projections

⇓

⇒	y_1	y_2	y_3	y_4	y_5	y_6	$\mu_{R^1}(x)$	
x_1	0.1	0.2	0.4	0.8	1.0	0.6	1.0	
x_2	0.2	0.4	0.8	0.9	0.8	0.6	0.9	
x_3	0.5	0.9	1.0	0.8	0.4	0.2	1.0	
$\mu_{R^2}(y)$	0.5	0.9	1.0	0.9	1.0	0.6	1.0	μ_{R^T}

The second projection is what the relation would look like if seen from the direction of the arrow on top of the table.

$$R^2 = \sum_Y \mu_{R^2}(y_j)/y_j = 0.5/y_1 + 0.9/y_2 + 1.0/y_3 + 0.9/y_4 + 0.6/y_5 + 0.8/y_6 \tag{E3.5-2}$$

The total projection is the single cell in the corner and represents the highest grade of membership that the relation has, namely, 1.

Let us next take a look at the cylindrical extension of the second projection. In a way the cylindrical extension is the *opposite* of projection. We expect therefore to obtain a relation on $X \times Y$ somewhat similar to the original relation R. As equation (E3.5-2) indicates, the second projection is defined on the Y universe of discourse. The generalization of this to the $X \times Y$ two-dimensional space is given by the *cylindrical extension*. Using equation (3.4-7) we obtain that the cylindrical extension of the second projection of the relation R^2 is simply the fuzzy set of the second projection extended in one more dimension, namely,

CE(R^2):

	y_1	y_2	y_3	y_4	y_5	y_6
x_1	0.5	0.9	1.0	0.9	1.0	0.6
x_2	0.5	0.9	1.0	0.9	1.0	0.6
x_3	0.5	0.9	1.0	0.9	1.0	0.6

Note that although the cylindrical extension of the second projection R^2 results in a relation of higher dimensionality, it did not recover the original relation R. Some information was lost through the operation of the cylindrical extension. □

3.5 COMPOSITION OF FUZZY RELATIONS

Fuzzy relations defined on different Cartesian products can be combined with each other in a number of different ways through *composition*. Composition may be thought of metaphorically as a bridge that allows us to connect one product space to another, provided that there is a common boundary. Figure 3.4 illustrates the notion. Given two fuzzy relations—one in $X \times Y$ and another on $Y \times Z$—we want to associate directly elements of X with elements of Z. The set Y is the common boundary. Composition results in a

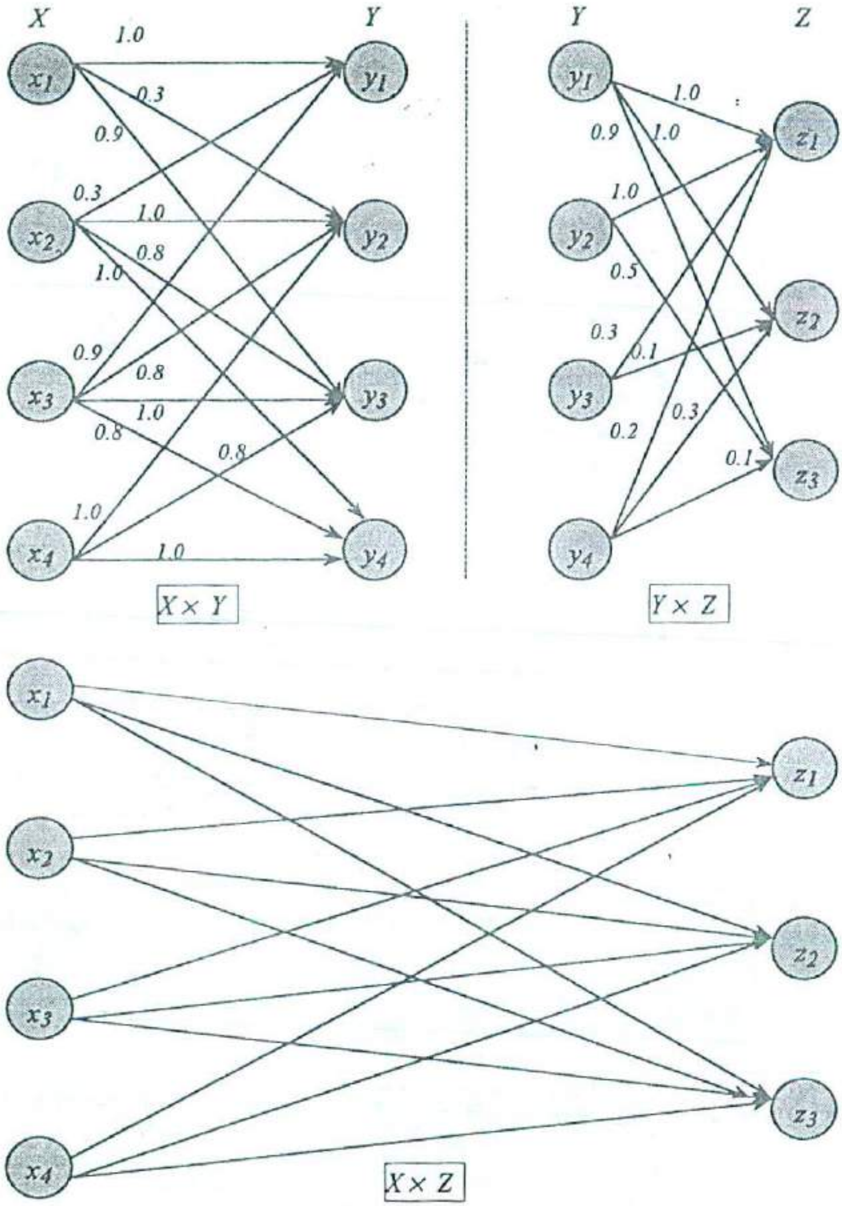


Figure 3.4 The composition of two fuzzy relations is a new relation directly associating elements from X and Z.

new relation shown at the bottom of Figure 3.4 that directly relates X to Z. Our main task in composition is to compute the grades of membership of the pairs (x, z) in the composed relation, namely, $\mu(x, z)$ (not shown in Figure 3.4).

Composition is very important for inferencing procedures used in linguistic descriptions of systems and is particularly useful in fuzzy controllers and expert systems (Klir and Folger, 1988). As we shall see in Chapters 5 and 6, collections of fuzzy *if/then* rules or *fuzzy algorithms* are mathematically equivalent to fuzzy relations, and the problem of *inferencing* or (*evaluating* them with specific inputs) is mathematically equivalent to *composition*. There are several types of composition. By far the most common in engineering applications is *max–min composition*, but we will also look at *max-star*, *max-product*, and *max-average*. In general, different types of composition result in different composed relations.

Max–Min Composition

The max–min composition of two fuzzy relations uses the familiar operators of fuzzy sets, max ($\vee$) and min ($\wedge$) (see Section 2.3). Suppose that we have two fuzzy relations $R_1(x, y)$ and $R_2(y, z)$ defined over the Cartesian products $X \times Y$ and $Y \times Z$, respectively. The max–min composition of R_1 and R_2 is a new relation $R_1 \circ R_2$ defined on $X \times Z$ as

$$R_1 \circ R_2 \equiv \int_{X \times Z} \bigvee_y \left[\mu_{R_1}(x, y) \wedge \mu_{R_2}(y, z) \right] \Big/ (x, z) \qquad (3.5\text{-}1)$$

where the symbol "$\circ$" stands for max–min composition of relations R_1 and R_2. When the Cartesian product $X \times Y$ is discrete, then the integral (union) sign in (3.5-1) is replaced by summation. From equation (3.5-1) we see that the grade of membership of each (x, z) pair in the new relation is

$$\mu_{R_1 \circ R_2}(x, z) = \bigvee_y \left[\mu_{R_1}(x, y) \wedge \mu_{R_2}(y, z) \right] \qquad (3.5\text{-}2)$$

where the outer maximum is taken with respect to the elements y of the common boundary. The operation on the right-hand side of equation (3.5-2) is actually very similar to matrix multiplication, with max ($\vee$) being analogous to *summation* ($+$) and min ($\wedge$) being analogous to *multiplication* ($\cdot$), as we will see in the examples that follow. Interchanging min and max in (3.5-1) is known as the *min–max composition*. In this book, however, we will mostly use max–min composition and compositions where the final (outer) operand is max ($\vee$). Max–min composition is used extensively in diagnostic and control applications of fuzzy logic.

Max–Star Composition

We can use *multiplication*, *summation*, or some other *binary operation* ($*$) in place of min ($\wedge$) in equations (3.5-1) and (3.5-2) while still performing maximization with respect to y. This type of composition of two fuzzy relations is generally known as the "*max-star*" or "*max-$*$ composition.*"[1]

Suppose that we have two fuzzy relations R_1 and R_2 defined over the Cartesian products $X \times Y$ and $Y \times Z$, respectively. The max-$*$ composition of R_1 and R_2 is the new relation

$$R_1 * R_2 \equiv \int_{X\times Z} \bigvee_y \left[\mu_{R_1}(x, y) * \mu_{R_2}(y, z)\right] \Big/ (x, z) \qquad (3.5\text{-}3)$$

We see from equation (3.5-3) that the membership function of the new relation is

$$\mu_{R_1 * R_2}(x, z) = \bigvee_y \left[\mu_{R_1}(x, y) * \mu_{R_2}(y, z)\right] \qquad (3.5\text{-}4)$$

When the Cartesian product is discrete the integral sign in equation (3.5-3) is replaced by summation. Again as we shall see in the examples that follow this is essentially a computational procedure very similar to matrix multiplication. Two special cases of the max-star composition are the *max-product* (or *max-prod*) and the *max-average* composition.

Max-Product Composition

In max-product composition we use product ($\cdot$) in place of ($*$) in equations (3.5-3) and (3.5-4). Thus the max-product composition of two relations R_1 and R_2 is

$$R_1 \cdot R_2 \equiv \int_{X\times Z} \bigvee_y \left[\mu_{R_1}(x, y) \cdot \mu_{R_2}(y, z)\right] \Big/ (x, z) \qquad (3.5\text{-}5)$$

For discrete product spaces we use the summation sign in equation (3.5-5). The membership function of the composed relation is given by

$$\mu_{R_1 \cdot R_2}(x, z) = \bigvee_y \left[\mu_{R_1}(x, y) \cdot \mu_{R_2}(y, z)\right] \qquad (3.5\text{-}6)$$

[1] The name "star" refers to the star symbol that stands for a number of operations such as *average* and *product*.

Max–Average Composition

In the max-average composition of fuzzy relations we use the arithmetic sum (+) divided by 2 in place of (∗) in equations (3.5-3) and (3.5-4). Thus the max-average composition of R_1 with R_2 is a new relation $R_1\langle + \rangle R_2$ given by

$$R_1\langle + \rangle R_2 \equiv \int_{X\times Z} \bigvee_y \left[\tfrac{1}{2}\left(\mu_{R_1}(x,y) + \mu_{R_2}(y,z)\right)\right] \Big/ (x,z) \quad (3.5\text{-}7)$$

with membership function

$$\mu_{R_1\langle + \rangle R_2}(x,z) = \bigvee_y \left[\tfrac{1}{2}\left(\mu_{R_1}(x,y) + \mu_{R_2}(y,z)\right)\right] \quad (3.5\text{-}8)$$

Let us take a look at a few examples of composition.

Example 3.6 Max–Min Composition of Fuzzy Relations. Let's use max–min composition with the two relations shown in the upper part of Figure 3.4. The membership matrices of the relations R_1 on $X \times Y$ and R_2 on $Y \times Z$ are

$$R_1 = \begin{bmatrix} \mu_{R_1}(x_1,y_1) & \mu_{R_1}(x_1,y_2) & \mu_{R_1}(x_1,y_3) & \mu_{R_1}(x_1,y_4) \\ \mu_{R_1}(x_2,y_1) & \mu_{R_1}(x_2,y_2) & \mu_{R_1}(x_2,y_3) & \mu_{R_1}(x_2,y_4) \\ \mu_{R_1}(x_3,y_1) & \mu_{R_1}(x_3,y_2) & \mu_{R_1}(x_3,y_3) & \mu_{R_1}(x_3,y_4) \\ \mu_{R_1}(x_4,y_1) & \mu_{R_1}(x_4,y_2) & \mu_{R_1}(x_4,y_3) & \mu_{R_1}(x_4,y_4) \end{bmatrix}$$

$$= \begin{bmatrix} 1.0 & 0.3 & 0.9 & 0.0 \\ 0.3 & 1.0 & 0.8 & 1.0 \\ 0.9 & 0.8 & 1.0 & 0.8 \\ 0.0 & 1.0 & 0.8 & 1.0 \end{bmatrix} \quad (\text{E3.6-1})$$

and

$$R_2 = \begin{bmatrix} \mu_{R_2}(y_1,z_1) & \mu_{R_2}(y_1,z_2) & \mu_{R_2}(y_1,z_3) \\ \mu_{R_2}(y_2,z_1) & \mu_{R_2}(y_2,z_2) & \mu_{R_2}(y_2,z_3) \\ \mu_{R_2}(y_3,z_1) & \mu_{R_2}(y_3,z_2) & \mu_{R_2}(y_3,z_3) \\ \mu_{R_2}(y_4,z_1) & \mu_{R_2}(y_4,z_2) & \mu_{R_2}(y_4,z_3) \end{bmatrix}$$

$$= \begin{bmatrix} 1.0 & 1.0 & 0.9 \\ 1.0 & 0.0 & 0.5 \\ 0.3 & 0.1 & 0.0 \\ 0.2 & 0.3 & 0.1 \end{bmatrix} \quad (\text{E3.6-2})$$

We want to compute the membership matrix of the max–min composition of R_1 and R_2. We can use equation (3.5-2) to obtain the membership function of the composed relation. The operations in (3.5-2) are similar to matrix multiplication, with ($\vee$) being treated like *summation* (+) and ($\wedge$) being treated like *multiplication* ($\cdot$). With this in mind, instead of using

$$\mu_{R_1 \circ R_2}(x, z) = \bigvee_y \left[\mu_{R_1}(x, y) \wedge \mu_{R_2}(y, z) \right]$$

we can use the matrix form of max–min composition, namely,

$$R_1 \circ R_2 = \begin{bmatrix} 1.0 & 0.3 & 0.9 & 0.0 \\ 0.3 & 1.0 & 0.8 & 1.0 \\ 0.9 & 0.8 & 1.0 & 0.8 \\ 0.0 & 1.0 & 0.8 & 1.0 \end{bmatrix} \circ \begin{bmatrix} 1.0 & 1.0 & 0.9 \\ 1.0 & 0.0 & 0.5 \\ 0.3 & 0.1 & 0.0 \\ 0.2 & 0.3 & 0.1 \end{bmatrix} \qquad \text{(E3.6-3)}$$

To evaluate equation (E3.6-3) we proceed, like in matrix multiplication, by forming the pairs of minima of each element in the first row of membership matrix R_1 with every element in the first column of membership matrix R_2. For example, to obtain the first element, (x_1, z_1), of the composition we perform the following operations:

$$\begin{aligned} &\begin{bmatrix} 1.0 & 0.3 & 0.9 & 0.0 \end{bmatrix} \circ \begin{bmatrix} 1.0 \\ 1.0 \\ 0.3 \\ 0.2 \end{bmatrix} \\ &\quad = [1.0 \wedge 1.0] \vee [0.3 \wedge 1.0] \vee [0.9 \wedge 0.3] \vee [0.0 \wedge 0.2] \\ &\quad = 1.0 \vee 0.3 \vee 0.3 \vee 0.0 \\ &\quad = 1.0 \end{aligned}$$

We repeat this procedure for all rows and columns and the result is the membership matrix of the composed relation $R_1 \circ R_2$ given by

$$R_1 \circ R_2 = \begin{bmatrix} 1.0 & 1.0 & 0.9 \\ 1.0 & 0.3 & 0.5 \\ 0.9 & 0.9 & 0.9 \\ 1.0 & 0.3 & 0.5 \end{bmatrix} \qquad \text{(E3.6-4)}$$

The new relation is a fuzzy set over the Cartesian product $X \times Z$ which may also be written as

$$\begin{aligned} R_1 \circ R_2 = {} & 1.0/(x_1, z_1) + 1.0/(x_1, z_2) + 0.9/(x_1, z_3) \\ & + 1.0/(x_2, z_1) + 0.3/(x_2, z_2) + 0.5/(x_2, z_3) \\ & + 0.9/(x_3, z_1) + 0.9/(x_3, z_2) + 0.9/(x_3, z_3) \\ & + 1.0/(x_4, z_1) + 0.3/(x_4, z_2) + 0.5/(x_4, z_3) \qquad \text{(E3.6-5)} \end{aligned}$$

□

Example 3.7 Max–Min, Max-Product, and Max-Average Composition of Fuzzy Relations. Suppose we have the two relations R_1 and R_2, shown below, and we want to compute a new relation which is the max–min composition of the two, $R = R_1 \circ R_2$. We will also find the max-product and max-average compositions. We perform max–min composition using the tabular representation of the relations and the definition of max–composition given in equations (3.5-1) or (3.5-2). The relations to be composed are described by the following membership tables:

R_1:

	y_1	y_2	y_3	y_4	y_5
x_1	0.1	0.2	0.0	1.0	0.7
x_2	0.3	0.5	0.0	0.2	1.0
x_3	0.8	0.0	1.0	0.4	0.3

R_2:

	z_1	z_2	z_3	z_4
y_1	0.9	0.0	0.3	0.4
y_2	0.2	1.0	0.8	0.0
y_3	0.8	0.0	0.7	1.0
y_4	0.4	0.2	0.3	0.0
y_5	0.0	1.0	0.0	0.8

To find the new relation $R = R_1 \circ R_2$ we use equation (3.5-2), the definition of max–min composition, namely,

$$\mu_{R_1 \circ R_2}(x, z) = \bigvee_y \left[\mu_{R_1}(x, y) \wedge \mu_{R_2}(y, z) \right] \qquad \text{(E3.7-1)}$$

To use (E3.7-1) we proceed in the following manner. First, we fix x and z—for example, $x = x_1$ and $z = z_1$—and vary y. Next, we evaluate the following pairs of minima, using the numbers from the shaded cells in the tables of the two relations:

$$\begin{aligned}
\mu_{R_1}(x_1, y_1) \wedge \mu_{R_2}(y_1, z_1) &= 0.1 \wedge 0.9 = 0.1 \\
\mu_{R_1}(x_1, y_2) \wedge \mu_{R_2}(y_2, z_1) &= 0.2 \wedge 0.2 = 0.2 \\
\mu_{R_1}(x_1, y_3) \wedge \mu_{R_2}(y_3, z_1) &= 0.0 \wedge 0.8 = 0.0 \\
\mu_{R_1}(x_1, y_4) \wedge \mu_{R_2}(y_4, z_1) &= 1.0 \wedge 0.4 = 0.4 \\
\mu_{R_1}(x_1, y_5) \wedge \mu_{R_2}(y_5, z_1) &= 0.7 \wedge 0.0 = 0.0
\end{aligned} \qquad \text{(E3.7-2)}$$

We take the maximum of all these terms and obtain the value of the (x_1, z_1) element of the relation, namely,

$$\mu_{R_1 \circ R_2}(x_1, z_1) = 0.1 \vee 0.2 \vee 0.0 \vee 0.4 = 0.4 \qquad \text{(E3.7-3)}$$

This is the value in the shaded cell in the table of the composed relation shown below. In a similar manner, we determine the grades of membership for all other pairs and finally we have

$R = R_1 \circ R_2$:

	z_1	z_2	z_3	z_4
x_1	0.4	0.7	0.3	0.7
x_2	0.3	1.0	0.5	0.8
x_3	0.8	0.3	0.7	1.0

Let us now compose these two relations using max-product composition as defined by equation (3.5-6)—that is,

$$\mu_{R_1 \cdot R_2}(x, z) = \bigvee_y \left[\mu_{R_1}(x, y) \cdot \mu_{R_2}(y, z) \right] \qquad \text{(E3.7-4)}$$

Again we fix x and z and vary y—for example, $x = x_1$, $z = z_1$, and $y = y_i$ for $i = 1, \ldots, 5$. We form and evaluate the products of the shaded cells in the relation tables—that is,

$$
\begin{aligned}
\mu_{R_1}(x_1, y_1) \cdot \mu_{R_2}(y_1, z_1) &= 0.1 \times 0.9 = 0.09 \\
\mu_{R_1}(x_1, y_2) \cdot \mu_{R_2}(y_2, z_1) &= 0.2 \times 0.2 = 0.04 \\
\mu_{R_1}(x_1, y_3) \cdot \mu_{R_2}(y_3, z_1) &= 0.0 \times 0.8 = 0.0 \\
\mu_{R_1}(x_1, y_4) \cdot \mu_{R_2}(y_4, z_1) &= 1.0 \times 0.4 = 0.4 \\
\mu_{R_1}(x_1, y_5) \cdot \mu_{R_2}(y_5, z_1) &= 0.7 \times 0.0 = 0.0
\end{aligned}
\qquad \text{(E3.7-5)}
$$

Taking the maximum of these terms, we obtain the grade of membership of the (x_1, z_1) pair in the composed relation, namely,

$$\mu_{R_1 \cdot R_2}(x_1, z_1) = 0.09 \vee 0.04 \vee 0.0 \vee 0.4 \vee 0.0 \qquad \text{(E3.7-6)}$$

which coincidentally evaluates also to $\mu_{R_1 \cdot R_2}(x_1, z_1) = 0.4$. This is the number in the shaded cell of the table below. Similarly, we obtain the membership of all other pairs and finally we get the membership table of the composition as

$R_1 \cdot R_2$:

	z_1	z_2	z_3	z_4
x_1	0.4	0.7	0.3	0.56
x_2	0.27	1.0	0.4	0.8
x_3	0.8	0.3	0.7	1.0

For the max-average composition of the two relations, again we fix x and z and vary y in order to find the max with respect to y in equation (3.5-8) for each (x, z) pair. Thus first we form and evaluate the sums of the shaded cells as before:

$$
\begin{aligned}
\mu_{R_1}(x_1, y_1) + \mu_{R_2}(y_1, z_1) &= 0.1 + 0.9 = 1.0 \\
\mu_{R_1}(x_1, y_2) + \mu_{R_2}(y_2, z_1) &= 0.2 + 0.2 = 0.4 \\
\mu_{R_1}(x_1, y_3) + \mu_{R_2}(y_3, z_1) &= 0.0 + 0.8 = 0.8 \\
\mu_{R_1}(x_1, y_4) + \mu_{R_2}(y_4, z_1) &= 1.0 + 0.4 = 1.4 \\
\mu_{R_1}(x_1, y_5) + \mu_{R_2}(y_5, z_1) &= 0.7 + 0.0 = 0.0
\end{aligned}
\qquad \text{(E3.7-7)}
$$

Thus, using equation (3.5-8), the grade of membership of the (x_1, z_1) pair is

$$\mu_{R_1\langle+\rangle R_2}(x_1, z_1) = \tfrac{1}{2}[1.0 \vee 0.4 \vee 0.8 \vee 1.4 \vee 0.0] = 0.7 \quad \text{(E3.7-8)}$$

This is the grade of membership of the shaded cell in the table shown below. In a similar manner the membership function for each pair is computed, and finally we get the max-average composition of the two relations in the table:

$R_1\langle+\rangle R_2$:

	z_1	z_2	z_3	z_4
x_1	0.7	0.85	0.65	0.75
x_2	0.6	1.0	0.65	0.9
x_3	0.9	0.65	0.85	1.0

We observe from the tables of the composed relations that max–min, max-product, and max-average compositions of R_1 and R_2 may result in different relations. □

REFERENCES

Dubois, D., and Prade, H., *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, Boston, 1980.

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

Kaufmann, A., *Introduction to the Theory of Fuzzy Subsets*, Vol. I, Academic Press, New York, 1975.

Kaufmann, A., and Gupta, M. M., *Introduction to Fuzzy Arithmetic*, Van Nostrand Reinhold, New York, 1991.

Klir, G. J., and Folger, T. A., *Fuzzy Sets, Uncertainty, and Information*, Prentice-Hall, Englewood Cliffs, NJ, 1988.

Terano, T., Asai, K., and Sugeno, M., *Fuzzy Systems Theory and its Applications*, Academic Press, Boston, 1992.

Zadeh, L., Similarity Relations and Fuzzy Orderings, *Information Sciences*, Vol. 3, Elsevier, Amsterdam, 1971, pp. 177–200.

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

PROBLEMS

1. A fuzzy "diagnostic relation" R_d for an automobile relates the system set S to the fault set F. These sets are given below.

$$S = [x_1 \text{ (low gas mileage)}, x_2 \text{ (excessive vibration)}, x_3 \text{ (loud noise)}, x_4 \text{ (high collant temperature)}, x_5 \text{ (steering instability)}]$$

$$F = [y_1 \text{ (bad spark plugs)}, y_2 \text{ (wheel imbalance)}, y_3 \text{ (bad muffler)}, y_4 \text{ (thermostat stuck closed)}]$$

   Assume reasonable numerical values $(0 \rightarrow 1)$ for membership values relating members of sets S and F and use them. Give all five representations of this fuzzy diagnostic relationship R_d in terms of x_i and y_j.

2. Give the max–min composition, max-star composition, and the max-average composition of the relation fuzzy "diagnostic relation" of Problem 1.

3. Repeat Example 3.3 for a fuzzy relation R indicating that "x is near the perimeter of a circle having a radius 1 with y".

4. In Example 3.4, give a table for $[R_1 \cap R_2] \cup [R_1 \cap R_2]$.

5. Find the first, second, and total projection as well as the cylindrical extension of the fuzzy relation R given by Equation (E3.2-2).

6. Find the max-product and max-average composition of relations R_1 and R_2 given by Equations (E3.6-1) and (E3.6-2), respectively.

7. Find the max–min composition of relations R_1 and R_2 given in Example 3.7.

8. a. Show that the max–min composition of fuzzy relations is *associative*. Illustrate with an example of your own.

   b. Consider the max-min composition and a relation R which is *reflexive*. Show that:

$$R \circ R = R.$$

9. Suppose that we have three relations involved in max–min composition

$$P \circ Q = R$$

When two of the components in the above equation are given and the other is unknown we have a set of equations known as *fuzzy relation equations*. Solve the following fuzzy relation equations:

(a) $P \circ \begin{bmatrix} .9 & .6 & 1 \\ .8 & .8 & .5 \\ .6 & .4 & .6 \end{bmatrix} = [.6 \quad .6 \quad .5]$

(b) $P \circ \begin{bmatrix} .2 & .4 & .5 & .7 \\ .3 & .1 & .6 & .8 \\ .1 & .4 & .6 & .7 \\ 0 & .3 & 0 & 1 \end{bmatrix} = \begin{bmatrix} .2 & .4 & .6 & .7 \\ .1 & .1 & .2 & .2 \end{bmatrix}$

10. Consider two probability distributions that are independent and described by

$$dP(x_1) = e^{-x_1}\,dx_1 \text{ and } dP(x_2) = x_2 e^{-x_2}\,dx_2,\ x_1, x_2 \geq 0$$

How can we model the *similarity* of x_1, x_2 through a fuzzy set and what would be the probability of occurrence of such a set?

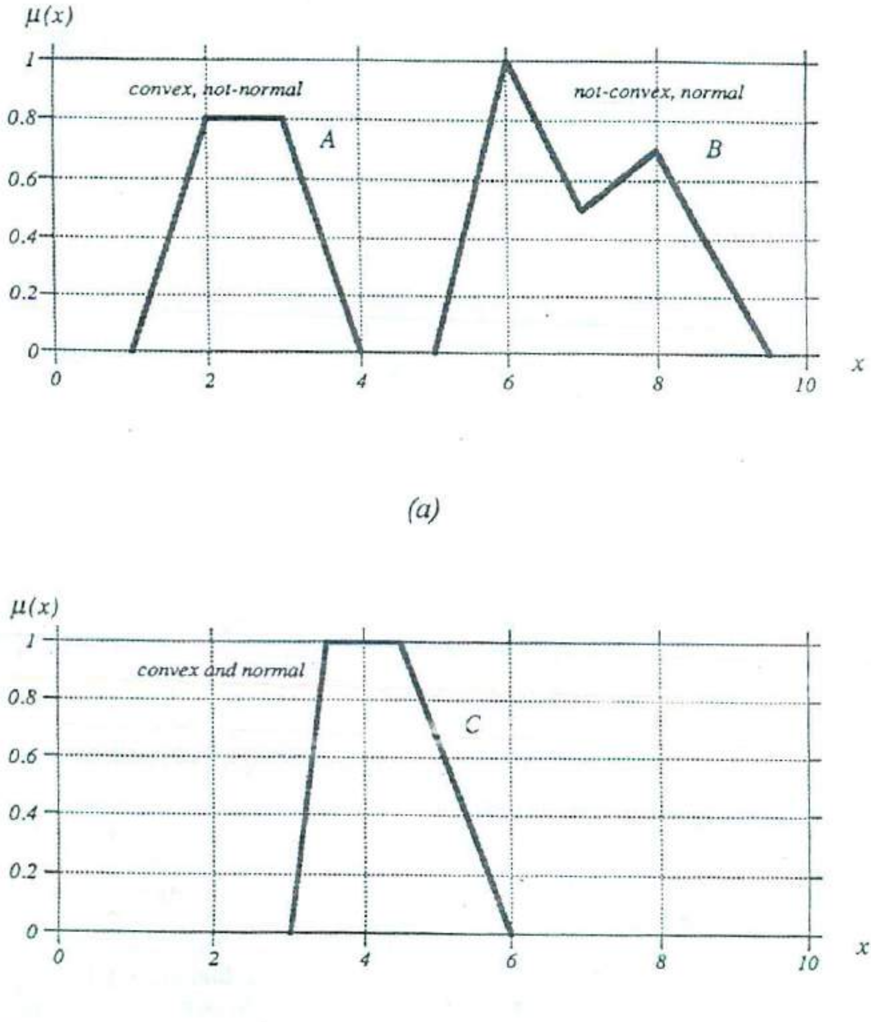
4

FUZZY NUMBERS

4.1 INTRODUCTION

Fuzzy numbers are fuzzy sets used in connection with applications where an explicit representation of the ambiguity and uncertainty found in numerical data is desirable. In an intuitive sense, they are fuzzy sets representing the meaning of statements such as "*about* 3" or "*nearly five and a half*." In other words, fuzzy numbers take into account the "about," "almost," and "not quite" qualities of numerical labels. Fuzzy set operations such as *union* and *intersection*, as well as the notions of *α-cuts*, *resolution*, and the *extension principle* (Chapter 2), are all applicable to fuzzy numbers. In addition, a set of operations very similar to the familiar operations of arithmetic, *addition*, *subtraction*, *multiplication*, and *division* can be defined for fuzzy numbers as well. In this chapter we look at such operations and examples of their use. Fuzzy numbers have been successfully applied in expert systems, fuzzy regression, and fuzzy data analysis methodologies (Kaufmann and Gupta, 1991; Terano et al., 1992). Fuzzy numbers have also been used in connection with fuzzy equations, and alternative operations of fuzzy arithmetic have been introduced for the purpose of reducing fuzziness in successive computations (Sanchez, 1993).

The universe of discourse on which fuzzy numbers are defined is the set of real numbers and its subsets (e.g., integers or natural numbers), and their membership functions ought to be *normal* and *convex*. We recall from Section 2.3 that a fuzzy set is called *normal* if there is at least one point in the universe of discourse where the membership function reaches unity [equation (2.3-11)]. But what is a "convex" fuzzy set? The intuitive meaning



(a)

(b)

Figure 4.1 (*a*) Two fuzzy sets that *cannot* be used as fuzzy numbers. (*b*) A fuzzy set that may be used as fuzzy number.

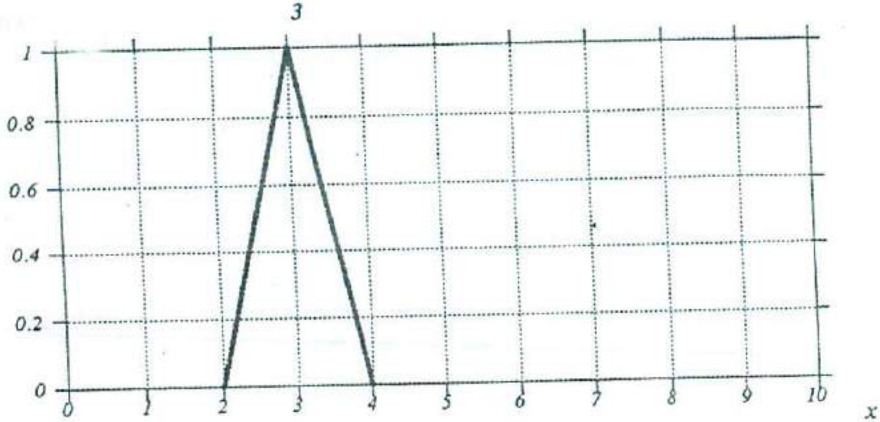
of *convexity*[1] is that the membership function of a convex fuzzy set does not go "up-and-down" more than once. Consider, for example, the fuzzy sets A and B shown in Figure 4.1*a*. Fuzzy set A is *convex* but *not normal* since nowhere in the universe of discourse does its membership function reach unity. Therefore it is not a fuzzy number. Fuzzy set B is *normal* but *not convex* since its membership function goes "up-and-down" twice, and hence it is also not a fuzzy number. On the other hand, consider the set C shown in Figure 4.1*b*. It is both *normal* and *convex* and therefore may be considered a fuzzy number. We will see in following sections that changing the shape of a membership function results in a different number. "Shape" is what fuzzy numbers are all about, and fuzzy arithmetic may be thought of as a way of computing with "shapes" (areas) instead of "points" (we consider crisp numbers as "points").

Fuzzy numbers may also be defined on a multidimensional universe of discourse that is a Cartesian product. Such fuzzy numbers are used, for example, in connection with scene analysis and robotics to define the meaning of a region in space, or a domain on the x–y plane, and also to add, subtract, and multiply regions (Pal and Majumder, 1986). In this chapter, however, we consider fuzzy numbers defined on a simple, one-dimensional universe of discourse. A very comprehensive treatment of fuzzy numbers, including multidimensional ones, may be found in the book entitled *Introduction to Fuzzy Arithmetic* by Kaufmann and Gupta (1991).

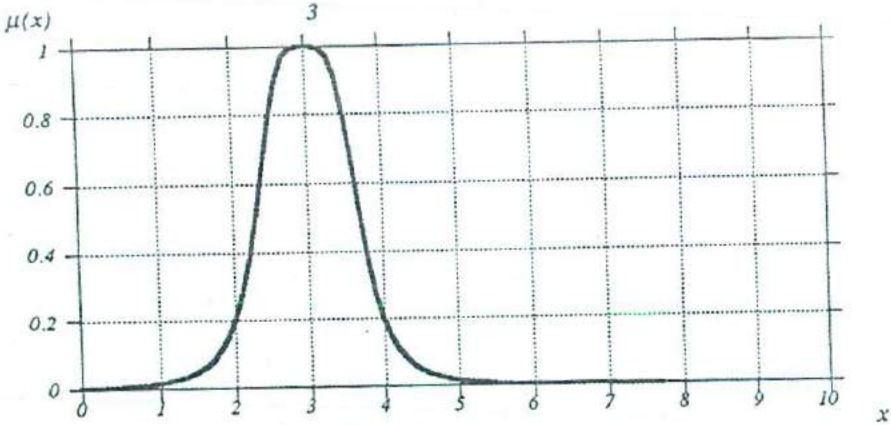
4.2 REPRESENTING FUZZY NUMBERS

We denote fuzzy numbers by boldfaced italics—for example, ***3*** or ***A***—or by referring to their membership function. As we said earlier, fuzzy numbers are fuzzy sets used to represent the "*about*," "*almost*," or "*nearly*" qualities of numerical data. We observe, however, that there are many possible meanings to a statement such as "*about* 3." Therefore, several different sets may be used to represent "*about* 3." In the context of fuzzy arithmetic operations, however, at any given time we use only one meaning, chosen on the basis of application-specific criteria and needs. Figure 4.2*a* shows a triangular membership function representing the fuzzy number ***3***. Another possible representation is the bell-shaped membership function in Figure 4.2*b*. These are two different ***3***'s. If we start a computation using the triangular ***3***, we cannot halfway through switch to the bell-shaped ***3***. Note that on both instances the shape of the membership function meets our *normal* and *convex* require-

[1]The notion of convexity is derived through references to geometrical objects. A body Ω in Euclidean space is called *convex* if the line segment joining any two points of Ω lies in Ω. Examples of convex bodies in three-dimensional space are the sphere, the ellipsoid, a cylinder, a cube, and a cone.



(a)



(b)

Figure 4.2 Two different fuzzy numbers: (*a*) triangular **3** and (*b*) bell-shaped **3**.

Table 4.1 Tabular representation of a fuzzy number *3*

	0.4	*0.7*	*1*	*0.7*	*0.4*	*0.2*	*0.1*	*0*	*0*
$\alpha=1$			1						
$\alpha=0.9$			1						
$\alpha=0.8$			1						
$\alpha=0.7$		1	1	1					
$\alpha=0.6$		1	1	1					
$\alpha=0.5$		1	1	1					
$\alpha=0.4$	1	1	1	1	1				
$\alpha=0.3$	1	1	1	1	1				
$\alpha=0.2$	1	1	1	1	1	1			
$\alpha=0.1$	1	1	1	1	1	1			
$\alpha=0$	1	1	1	1	1	1	1	1	1
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*

ments. Another possible fuzzy number ***3*** is shown in Table 4.1, where the shaded cells, the 1's, indicate the shape of the number. Here ***3*** is defined over the universe of natural numbers shown at the bottom of the table. In the leftmost column we list the values of a parameter, α, ranging between 0 and 1, used to parametrize the shape of the function (Kaufmann and Gupta, 1991). In fact, this is the same α we saw in connection with α-cuts (see Section 2.6). The α-cuts of fuzzy numbers are very useful in fuzzy arithmetic operations. Looking at Table 4.1 we see that the grade of membership of crisp number 4 to the fuzzy number ***3*** is 0.7, and the grade of membership of crisp 3 is 1.0. Although the fuzzy numbers shown in Figure 4.2 and Table 4.1 are all different, we designate them with the same symbol (i.e., ***3***) since they all peak at crisp 3 (Zimmermann, 1985; Kandel, 1986).

Fuzzy numbers, like any fuzzy set, may be represented by its α-cuts. We saw in Chapter 2 that a membership function may be parameterized by a parameter α in a manner similar to the tabular representation of number ***3*** shown in Table 4.1. The parameter α is a number between 0 and 1 (i.e., in the interval [0, 1]). Parameterizing the shape of a fuzzy number by α offers a ient way for computing with fuzzy numbers because it essentially trans ns fuzzy arithmetic operations into operations of interval arithmetic. It is easy to see what we are talking about by looking at Table 4.1. At each level α we have a horizontal "slice," or interval of the membership function, which is its α-cut. For example, at $\alpha = 0.5$, the α-cut is the interval from 2 to 4, and at $\alpha = 0.2$ it is the interval from 1 to 6. The tabular representation exemplifies the length of each α-cut; that is, it shows the number of cells and thus the length of the membership function at level α.

Consider the fuzzy number A shown in Figure 4.3. The membership function of A is parameterized by the parameter α. With each α we identify an interval $[a_1^{(\alpha)}, a_2^{(\alpha)}]$. As may be seen from the figure, we indicate by $a_1^{(\alpha)}$ the left endpoint of the interval ("left" is denoted by the subscript "1") and by $a_2^{(\alpha)}$ the right endpoint of the interval ("right" is denoted by the subscript "2"). Requiring that the membership function of a fuzzy number be *convex* and *normal* is another way of saying that the intervals that comprise the interval representation of A should be nested into one another as we move from the bottom of the membership function to the top (Klir and Folger, 1988; Terano et al., 1992). In other words, when $\alpha_1 < \alpha_2$, as shown in Figure 4.3, we have

$$\left[a_1^{(\alpha_2)}, a_2^{(\alpha_2)}\right] \subset \left[a_1^{(\alpha_1)}, a_2^{(\alpha_1)}\right] \tag{4.2-1}$$

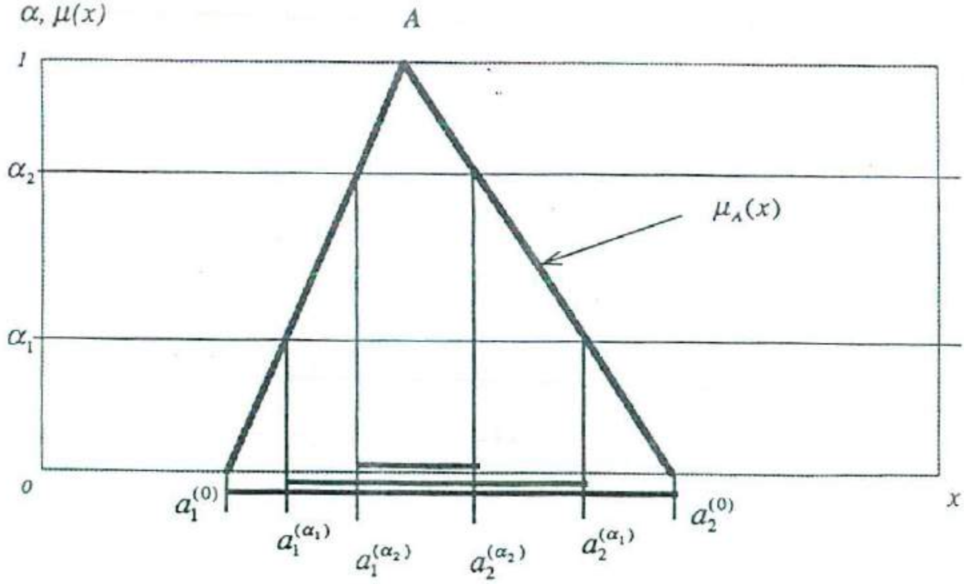


Figure 4.3 Nested intervals (α-cuts) associated with a fuzzy number A.

where the symbol $\subset$ denotes that the interval $[a_1^{(\alpha_2)}, a_2^{(\alpha_2)}]$ is contained within the interval $[a_1^{(\alpha_1)}, a_2^{(\alpha_1)}]$.

We can uniquely describe two fuzzy numbers A and B as two collection of intervals i.e., $[a_1^{(\alpha)}, a_2^{(\alpha)}]$ and $[b_1^{(\alpha)}, b_2^{(\alpha)}]$ respectively. We recall that the α-cuts of A and B (Section 2.6) were defined as the crisp sets

$$A_\alpha = \{x \mid \mu_A(x) \geq \alpha\} \tag{4.2-2}$$

and

$$B_\alpha = \{x \mid \mu_B(x) \geq \alpha\} \tag{4.2-3}$$

The α-cuts in equations (4.2-2) and (4.2-3) are simply intervals on the x axis, and hence for each α we have

$$A_\alpha = \left[a_1^{(\alpha)}, a_2^{(\alpha)}\right] \tag{4.2-4}$$

and

$$B_\alpha = \left[b_1^{(\alpha)}, b_2^{(\alpha)}\right] \tag{4.2-5}$$

Thus the fuzzy numbers A and B can be described (using the resolution principle—see Section 2.7) as collections of intervals, that is,

$$A = \bigvee_{0 \leq \alpha \leq 1} \alpha \cdot A_\alpha = \bigvee_{0 \leq \alpha \leq 1} \alpha \cdot \left[a_1^{(\alpha)}, a_2^{(\alpha)}\right] \tag{4.2-6}$$

and

$$B = \bigvee_{0 \leq \alpha \leq 1} \alpha \cdot B_\alpha = \bigvee_{0 \leq \alpha \leq 1} \alpha \cdot \left[b_1^{(\alpha)}, b_2^{(\alpha)}\right] \tag{4.2-7}$$

To simplify matters, we will not use the rather awkward representation of the two numbers given by equations (4.2-6) and (4.2-7) but will, instead, use equations (4.2-4) and (4.2-5), which we call the *α-cut* or *interval representation* of A and B (with the understanding that the number is the collection of all "slices," all α-cuts as α varies from 0 to 1).

Having two different ways of representing fuzzy numbers, through *membership functions* and through *α-cuts* or *intervals*, gives us the choice of defining arithmetic operations either through the extension principle (i.e., through a fuzzification of arithmetic operations on crisp numbers) or, equivalently, through the operations of interval arithmetic. This last approach is often more practical and straightforward as we will see in several examples.

Let us go next to the definition of *addition, subtraction, multiplication,* and *division with fuzzy numbers*. Although we will define operations for two numbers A and B, they are generally true for more than two numbers. A word of caution: Some of the properties of crisp numbers—for example,

$(7 \div 3) \times 3 = 7$—may not be valid for arithmetic operations involving fuzzy numbers. We will see that usually when fuzzy numbers are involved we have that $(7 \div 3) \times 3$ may not equal 7.

4.3 ADDITION

When *adding* two fuzzy numbers A and B we seek to compute a new fuzzy number $C = A + B$. The new number C is uniquely described when we obtain its membership function, $\mu_C(z) \equiv \mu_{A+B}(z)$, with z being the crisp sum of x and y, the elements of the universe of discourse of A and B. The addition of A and B may be defined in terms of addition of the α-cuts of the two numbers as follows:

$$A + B \equiv \left[a_1^{(\alpha)}, a_2^{(\alpha)}\right] + \left[b_1^{(\alpha)}, b_2^{(\alpha)}\right] \tag{4.3-1}$$

where $[a_1^{(\alpha)}, a_2^{(\alpha)}]$ is the collection of intervals representing the fuzzy number A, and $[b_1^{(\alpha)}, b_2^{(\alpha)}]$ is the collection of intervals representing the fuzzy number B. Intervals are added by adding their corresponding left and right endpoints, and therefore equation (4.3-1) becomes

$$A + B = \left[a_1^{(\alpha)} + b_1^{(\alpha)}, a_2^{(\alpha)} + b_2^{(\alpha)}\right] \tag{4.3-2}$$

Equation (4.3-2) indicates that the new number is also a collection of intervals with endpoints obtained from the endpoints of A and B.

Another way of defining fuzzy addition is through the *extension principle* (Section 2.5). We give here a cursory description of how this is done; more detailed treatments may be found in Dubois and Prade (1980) and in Terano et al. (1992). Suppose we want to add two crisp numbers x and y. The result is another crisp number $z = x + y$. Now, if x and y are variables, obviously their sum may be thought of as a function of x and y; that is,

$$z(x, y) = x + y \tag{4.3-3}$$

Fuzzifying x and y—that is, defining fuzzy sets on x and y—results in a fuzzified function, $z = f(x, y)$. We saw in Section 2.5 how we can use the extension principle to obtain the fuzzy set C on $z = f(x, y)$. Suppose that we have two fuzzy numbers, A and B, defined over x and y (the universe of discourse of real numbers). According to the extension principle, their sum is a fuzzy set on z denoted as C, whose membership function is

$$\mu_C(z) \equiv \bigvee_{z=x+y} \left[\mu_A(x) \wedge \mu_B(y)\right] \tag{4.3-4}$$

Equation (4.3-4) tells us that to compute the grade of membership of a certain crisp number z to the fuzzy number C, we take the maximum of the minima of the grades of membership of all pairs x and y which add up to z. How equation (4.3-4) works will be seen in Example 4.2, where a rather simple tabular way of carrying out the max–min operations will be presented.

Example 4.1 Addition of Discrete Fuzzy Numbers. Let us compute the sum C of two fuzzy numbers $A = \mathit{3}$ and $B = \mathit{7}$ defined as

$$A = \mathit{3} = 0.3/1 + 0.7/2 + 1.0/3 + 0.7/4 + 0.3/5 + 0/6 \quad \text{(E4.1-1)}$$

$$B = \mathit{7} = 0.2/5 + 0.6/6 + 1.0/7 + 0.6/8 + 0.2/9 + 0/10 \quad \text{(E4.1-2)}$$

and seen in Table 4.2. We compute C by adding the α-cuts of A, B in accordance with equation (4.3-2). We see from Table 4.2 that when $\alpha = 0.4$, for example, the 0.4-cuts of A and B are

$$A_{0.4} = \left[a_1^{(0.4)}, a_2^{(0.4)}\right] = [2, 4] \quad \text{(E4.1-3)}$$

and

$$B_{0.4} = \left[b_1^{(0.4)}, b_2^{(0.4)}\right] = [6, 8] \quad \text{(E4.1-4)}$$

The intervals in equations (E4.1-3) and (E4.1-4) are shown as shaded "slices" of cells in Table 4.2. According to equation (4.3-1) the 0.4-cut of C is the sum of the two intervals given by (E4.1-3) and (E4.1-4)—that is,

$$\begin{aligned} C_{0.4} &= \left[a_1^{(0.4)}, a_2^{(0.4)}\right] + \left[b_1^{(0.4)}, b_2^{(0.4)}\right] \\ &= \left[a_1^{(0.4)} + b_1^{(0.4)}, a_2^{(0.4)} + b_2^{(0.4)}\right] \\ &= [2 + 6, 4 + 8] \\ &= [8, 12] \end{aligned} \quad \text{(E4.1-5)}$$

We can obtain the same result from Table 4.2 simply by adding the endpoints of the shaded rows. We repeat this for each α to compute the entire sum. We start from the bottom of the table and go up in a *row-by-row* manner identifying the corresponding intervals of the two numbers and adding them up. The result is the number shown in Table 4.3. The 0.4-cut of C is indicated as a shaded group of cells in the table. As seen from the table, the new fuzzy number reaches unity at crisp number 10 (in the universe of discourse shown at the bottom) and therefore we think of it as a fuzzy number **10**. Thus we see that the sum is $\mathit{7} + \mathit{3} = \mathbf{10}$, as would also be the case with crisp numbers. □

Table 4.2 Fuzzy numbers *3* and *7* in Example 4.1

3:

	0.3	0.7	1	0.7	0.3	0	0	0	0
α=1.0			1						
α=0.9			1						
α=0.8			1						
α=0.7		1	1	1					
α=0.6		1	1	1					
α=0.5		1	1	1					
α=0.4		1	1	1					
α=0.3	1	1	1	1	1				
α=0.2	1	1	1	1	1				
α=0.1	1	1	1	1	1				
α=0.0	1	1	1	1	1	1	1	1	1
	1	2	3	4	5	6	7	8	9

7:

	0	0	0	0	0.2	0.6	1.0	0.6	0.2	0	0
α=1.0							1				
α=0.9							1				
α=0.8							1				
α=0.7							1				
α=0.6						1	1	1			
α=0.5						1	1	1			
α=0.4						1	1	1			
α=0.3						1	1	1			
α=0.2					1	1	1	1	1		
α=0.1					1	1	1	1	1		
α=0.0	1	1	1	1	1	1	1	1	1	1	1
	1	2	3	4	5	6	7	8	9	10	11

Table 4.3 Sum of fuzzy numbers 3 and 7 in Example 4.1

10:

	0	0	0	0	0.	0.2	0.3	0.6	0.7	1.0	0.7	0.6	0.3	0.2	0	0.
α=1.0										1						
α=0.9										1						
α=0.8										1						
α=0.7									1	1	1					
α=0.6								1	1	1	1	1				
α=0.5								1	1	1	1	1				
α=0.4								1	1	1	1	1				
α=0.3							1	1	1	1	1	1	1			
α=0.2						1	1	1	1	1	1	1	1	1		
α=0.1						1	1	1	1	1	1	1	1	1		
α=0.0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Example 4.2 Addition of Fuzzy Numbers Through the Extension Principle. In this example we compute the sum of the two numbers A and B of Example 4.1 using the alternative definition of addition through the extension principle, namely, equation (4.3-4). At first glance, equation (4.3-4) looks somewhat esoteric. We present here a rather simple technique for using it. The same technique may be used with other fuzzy arithmetic operations as well (Kaufmann and Gupta, 1991). Let's repeat equation (4.3-4) here:

$$\mu_{A+B}(z) \equiv \bigvee_{z=x+y} [\mu_A(x) \wedge \mu_B(y)] \qquad \text{(E4.2-1)}$$

A convenient way to compute the sum according to equation (E4.2-1) is to create a table as shown in Table 4.4. We take the *support* of B and make as many columns in the table as there are elements in the support; and similarly we take the *support* of A and make as many rows in the table as there are elements in the support of A. We recall that the support is the part of the universe of discourse that has nonzero membership. A and B can be

Table 4.4 Adding fuzzy numbers through the extension principle

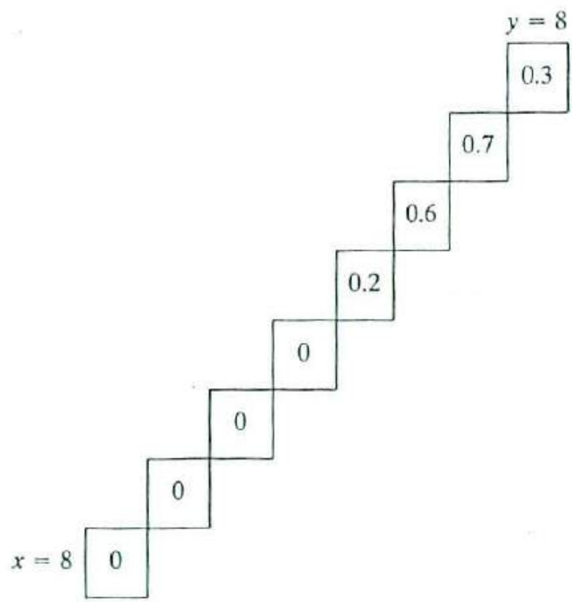
					Support of B					
B / A (Support of A)	y=1	y=2	y=3	y=4	y=5	y=6	y=7	y=8	y=9	y=10
x=1	0.0 / 0.3	0.0 / 0.3	0.0 / 0.3	0.0 / 0.3	0.2 / 0.3	0,6 / 0.3	1.0 / 0.3	0.6 / 0.3	0.2 / 0.3	0.0 / 0.3
x=2	0.0 / 0.7	0.0 / 0.7	0.0 / 0.7	0.0 / 0.7	0.2 / 0:7	0.6 / 0.7	1.0 / 0.7	0.6 / 0.7	0.2 / 0.7	0.0 / 0.7
x=3	0.0 / 1.0	0.0 / 1.0	0.0 / 1.0	0.0 / 1.0	0.2 / 1.0	0.6 / 1.0	1.0 / 1.0	0.6 / 1.0	0.2 / 1.0	0.0 / 1.0
x=4	0.0 / 0.7	0.0 / 0.7	0.0 / 0.7	0.0 / 0.7	0.2 / 0.7	0.6 / 0.7	1.0 / 0.7	0.6 / 0.7	0.2 / 0.7	0.0 / 0.7
x=5	0.0 / 0.3	0.0 / 0.3	0.0 / 0.3	0.0 / 0.3	0.2 / 0.3	0.6 / 0.3	1.0 / 0.3	0.6 / 0.3	0.2 / 0.3	0.0 / 0.3
x=6	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	0.2 / 0.0	0.6 / 0.0	1.0 / 0.0	0.6 / 0.0	0.2 / 0.0	0.0 / 0.0
x=7	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	0.2 / 0.0	0.6 / 0.0	1.0 / 0.0	0.6 / 0.0	0.2 / 0.0	0.0 / 0.0
x=8	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	0.2 / 0.0	0.6 / 0.0	1.0 / 0.0	0.6 / 0.0	0.2 / 0.0	0.0 / 0.0
x=9	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	0.2 / 0.0	0.6 / 0.0	1.0 / 0.0	0.6 / 0.0	0.2 / 0.0	0.0 / 0.0
x=10	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	0.2 / 0.0	0.6 / 0.0	1.0 / 0.0	0.6 / 0.0	0.2 / 0.0	0.0 / 0.0

interchanged in terms or rows and columns, but for the moment let's make columns from the support of A and make rows from the support of A. In every cell of the table we put at the lower left corner the grade of membership of x to A and put in the upper right corner the grade of membership of y to B. Thus we have $\mu_A(x)$ in the lower left corner and $\mu_B(y)$ in the upper right corner as shown in Table 4.4. Now, let's take another look in the equation above. It calls for taking the maximum of pairs of singletons that add up to a certain z. For example, suppose that have $z = 9$. There are three different ways to get $z = 9$: adding $y = 8$ and $x = 1$, adding $y = 7$ and

$x = 2$, adding $y = 6$ and $x = 3$ and so on. Both elements for each addition are found inside a cell. These are the shaded cells shown in Table 4.4. Equation (E4.2-1) says that for $z = 9$ we need to take the maximum of the minima of the three pairs of grades of membership inside the shaded cells. First we find the minimum of the grades of membership inside each cell—that is,

$$\begin{aligned}
\mu_A(1) \wedge \mu_B(8) &= 0.3 \wedge 0.6 = 0.3 \\
\mu_A(2) \wedge \mu_B(7) &= 0.7 \wedge 1.0 = 0.7 \\
\mu_A(3) \wedge \mu_B(6) &= 1.0 \wedge 0.6 = 0.6 \\
\mu_A(4) \wedge \mu_B(5) &= 0.7 \wedge 0.2 = 0.2 \\
\mu_A(5) \wedge \mu_B(4) &= 0.3 \wedge 0 = 0 \\
\mu_A(6) \wedge \mu_B(3) &= 0 \wedge 0 = 0 \\
\mu_A(7) \wedge \mu_B(2) &= 0 \wedge 0 = 0 \\
\mu_A(8) \wedge \mu_B(1) &= 0 \wedge 0 = 0
\end{aligned} \qquad \text{(E4.2-2)}$$

Now if we look only at the shaded part of the table, we can replace the contents of each cell with the minima found in equations (E4.2-2)—that is,



Next, we take the maximum of these numbers, which in this case is 0.7; this is the maximum with respect to $z = 9$ in equation (E4.2-1). At this point we have completed the entire operation on equation (E4.2-1) for $z = 4$—that is,

$$\mu_{A+B}(9) = [(0.3) \vee (0.7) \vee (0.6) \vee (0.2) \vee (0) \vee (0) \vee (0) \vee (0)]$$
$$= 0.7 \tag{E4.2-3}$$

This is the grade of membership of $z = 9$ to the sum $C = A + B$. We repeat this procedure for all other cells to obtain the membership function of C. The result is

$$C = 0/5 + 0.2/6 + 0.3/7 + 0.6/8 + 0.7/9 + 1.0/10$$
$$+ 0.7/11 + 0.6/12 + 0.3/13 + 0.2/14 + 0/15$$

which is the same number as the one we found by the interval approach in Example 4.1—that is, the number shown in Table 4.3. □

4.4 SUBTRACTION

The difference C of two fuzzy numbers A, B may be defined either through interval subtraction utilizing the *α-cut representation* of the two numbers or through the *extension principle*. Using α-cuts we subtract them as follows

$$A - B \equiv \left[a_1^{(\alpha)}, a_2^{(\alpha)}\right] - \left[b_1^{(\alpha)}, b_2^{(\alpha)}\right] \tag{4.4-1}$$

where $[a_1^{(\alpha)}, a_2^{(\alpha)}]$ is the collection of closed intervals representing A, and $[b_1^{(\alpha)}, b_2^{(\alpha)}]$ is the collection of closed intervals representing B. Two intervals are subtracted by subtracting their left and right endpoints, and thus equation (4.4-1) becomes

$$A - B = \left[a_1^{(\alpha)} - b_2^{(\alpha)}, a_2^{(\alpha)} - b_1^{(\alpha)}\right] \tag{4.4-2}$$

The alternative way to define the difference of fuzzy numbers A and B is through the extension principle—that is, by fuzzifying a function $z = x - y$. Fuzzification means that we define fuzzy sets on the universes of discourse where the crisp elements x and y are found. As a result, z gets fuzzified as well; that is, there is a fuzzy set C over the universe of discourse of the z's, which is the result of fuzzifying the function $z = f(x, y) = x - y$. The membership function of $C = A - B$ can be computed from

$$\mu_{A-B}(z) \equiv \bigvee_{z=x-y} [\mu_A(x) \wedge \mu_B(y)] \tag{4.4-3}$$

Equation (4.4-3) gives, of course, the same number C obtained through (4.4-2).

Example 4.3 Subtracting Fuzzy Numbers as Intervals. Let us compute a fuzzy number $C = 7 - 3$, where the fuzzy numbers 7 and 3 are as defined in Table 4.2 (Example 4.1):

$$A = 3 = 0.3/1 + 0.7/2 + 1.0/3 + 0.7/4 + 0.3/5 + 0/6 \quad \text{(E4.3-1)}$$

$$B = 7 = 0.2/5 + 0.6/6 + 1.0/7 + 0.6/8 + 0.2/9 + 0/10 \quad \text{(E4.3-2)}$$

Subtracting the two numbers is the same as interval subtraction at each α. From Table 4.2 we see that when $\alpha = 0.3$, for example, the 0.3-cuts of the two numbers are

$$A_{0.3} = \left[a_1^{(0.3)}, a_2^{(0.3)}\right] = [1, 5] \quad \text{(E4.3-3)}$$

and

$$B_{0.3} = \left[b_1^{(0.3)}, b_2^{(0.3)}\right] = [6, 8] \quad \text{(E4.3-4)}$$

The α-cut of C at $\alpha = 0.3$ is the difference of the α-cuts in by (E4.3-3) and (E4.3-4)

$$\begin{aligned} C_{0.3} &= \left[b_1^{(0.3)}, b_2^{(0.3)}\right] - \left[a_1^{(0.3)}, a_2^{(0.3)}\right] \\ &= \left[b_1^{(0.3)} - a_2^{(0.3)}, b_2^{(0.3)} - a_1^{(0.3)}\right] \\ &= [6 - 5, 8 - 1] \\ &= [1, 7] \end{aligned} \quad \text{(E4.3-5)}$$

shown as a "slice" of shaded cells in Table 4.5. In a similar manner we compute the α-cuts of C at the other levels of α and obtain the fuzzy number

$$\begin{aligned} C = 0.2/0 + 0.3/1 + 0.6/2 + 0.7/3 + 1.0/4 \\ + 0.7/5 + 0.6/6 + 0.3/7 + 0.2/8 \end{aligned}$$

which is also shown in Table 4.5. As may be seen from Table 4.5, C can be considered a fuzzy 4. □

Example 4.4 Subtracting Fuzzy Numbers with Continuous Membership Functions. Consider the two triangular fuzzy numbers A and B shown in Figure 4.4. We want to compute their difference—that is, find a fuzzy number $C = A - B$. When continuous (or piecewise continuous) membership functions are used, we subtract them by parameterizing their membership functions by α and subtracting their α-cuts. The membership functions of

Table 4.5 Difference of fuzzy numbers 7 and 3 in Example 4.3

4:

	0.2	0.3	0.6	0.7	1	0.7	0.6	0.3	0.2	0
$\alpha=1.0$					1					
$\alpha=0.9$					1					
$\alpha=0.8$					1					
$\alpha=0.7$				1	1	1				
$\alpha=0.6$			1	1	1	1	1			
$\alpha=0.5$			1	1	1	1	1			
$\alpha=0.4$			1	1	1	1	1			
$\alpha=0.3$		1	1	1	1	1	1	1		
$\alpha=0.2$	1	1	1	1	1	1	1	1	1	
$\alpha=0.1$	1	1	1	1	1	1	1	1	1	
$\alpha=0.0$	1	1	1	1	1	1	1	1	1	1
	0	1	2	3	4	5	6	7	8	9

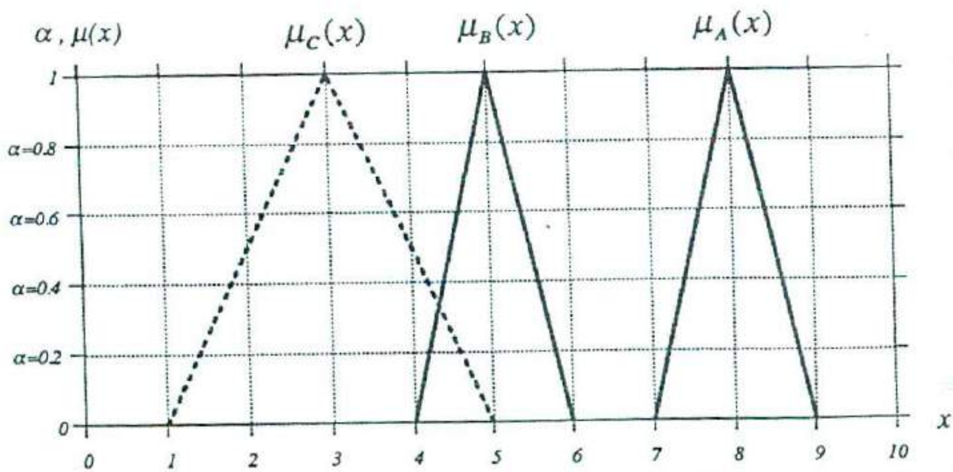


Figure 4.4 Subtracting two fuzzy numbers, $C = A - B$.

A, B are

$$\begin{aligned}\mu_A(x) &= 0, & x &\le 7\\ &= x-7, & 7 &\le x \le 8\\ &= -x+9, & 8 &\le x \le 9\\ &= 0, & x &\ge 9\end{aligned} \tag{E4.4-1}$$

and

$$\begin{aligned}\mu_B(x) &= 0, & x &\le 4\\ &= x-4, & 4 &\le x \le 5\\ &= -x+6, & 5 &\le x \le 6\\ &= 0, & x &\ge 6\end{aligned} \tag{E4.4-2}$$

Let us parameterize them by α. To simplify matters, consider the left and right side of each membership function separately. There is one equation for the left side and another for the right side of the membership function of A, and likewise for B. Thus, we have a total of four equations to parameterize. From equations (E4.4-1) we take the part that describes the left side of A, $\mu_A^L(x) = x - 7$, and write it in terms of a. We note that the value of α is the same as the value of the membership function at the left endpoint $a_1^{(\alpha)}$ of an α-cut, and $a_1^{(\alpha)}$ is the value of x at that point. Thus we have for the left side of A,

$$\alpha = a_1^{(\alpha)} - 7 \Rightarrow a_1^{(\alpha)} = \alpha + 7 \tag{E4.4-3}$$

where $a_1^{(\alpha)}$ is the left endpoint of the "slice" of A at level α.

Similarly for the right side of A we parameterize the right endpoint $a_2^{(\alpha)}$ of each α-cut in terms of α as

$$\alpha = -a_2^{(\alpha)} + 9 \Rightarrow a_2^{(\alpha)} = -\alpha + 9 \tag{E4.4-4}$$

Using equations (E4.4-3) and (E4.4-4) the α-cut representation of A is written as

$$A = \left[a_1^{(\alpha)}, a_2^{(\alpha)}\right] = [\alpha + 7, -\alpha + 9] \tag{E4.4-5}$$

The membership function of the number B is parameterized in terms of α in a similar fashion. We express the left endpoint $b_1^{(\alpha)}$ in terms of α by

$$\alpha = b_1^{(\alpha)} - 4 \Rightarrow b_1^{(\alpha)} = \alpha + 4 \tag{E4.4-6}$$

The right endpoint $b_2^{(\alpha)}$ is given as a function of α by

$$\alpha = -b_2^{(\alpha)} + 6 \Rightarrow b_2^{(\alpha)} = -\alpha + 6 \tag{E4.4-7}$$

From equations (E4.4-6) and (E4.4-7) the interval representation of B is

$$B = \left[b_1^{(\alpha)}, b_2^{(\alpha)}\right] = [\alpha + 4, -\alpha + 6] \tag{E4.4-8}$$

From the α-cut representations of A and B (equations (E4.4-5) and (E4.4-8)), we find their difference by subtracting their corresponding intervals at each α, that is,

$$\begin{aligned} C = A - B &= \left[a_1^{(\alpha)} - b_2^{(\alpha)}, a_2^{(\alpha)} - b_1^{(\alpha)}\right] \\ &= [(\alpha + 7) - (-\alpha + 6), (-\alpha + 9) - (\alpha + 4)] \\ &= [2\alpha + 1, -2\alpha + 5] \end{aligned} \tag{E4.4-9}$$

Therefore, C is

$$C = \left[c_1^{(\alpha)}, c_2^{(\alpha)}\right] = [2\alpha + 1, -2\alpha + 5] \tag{E4.4-10}$$

We note that the left and right endpoints of C are functions of α. To express the fuzzy number C in terms of a membership function, we derive equations for the left and right side of C. The left endpoint $c_1^{(\alpha)}$ in equation (E4.4-10) is equal to the value of x when the left-side membership function's value is α. Similarly, the right endpoint $c_2^{(\alpha)}$ is equal to the value of x when the right-side membership function is α. Thus the equation of the left side is obtained by setting $c_1^{(\alpha)} = x$ and recalling that $\alpha = \mu_C^L(x)$, where $\mu_C^L(x)$ is the left-side membership function for C. We have

$$x = 2\mu_C^L(x) + 1 \Rightarrow \mu_C^L(x) = \tfrac{1}{2}(x - 1) \tag{E4.4-11}$$

In a similar manner we obtain an equation for $\mu_C^R(x)$, the right side of the membership function of C, and solve it to obtain the membership function of the right side—that is,

$$x = -2\mu_C^R(x) + 5 \Rightarrow \mu_C^R(x) = -\tfrac{1}{2}(x - 5) \tag{E4.4-12}$$

From equations (E4.4-11) and (E4.4-12) we obtain

$$\begin{aligned} \mu_C(x) &= 0, & x &\le 1 \\ &= \tfrac{1}{2}(x - 1), & 1 &\le x \le 3 \\ &= -\tfrac{1}{2}(x - 5), & 3 &\le x \le 5 \\ &= 0, & x &\ge 5 \end{aligned} \tag{E4.4-13}$$

The number C described by equations (E4.4-11) is shown in Figure 4.4. Note that C has its peak at crisp 3, and therefore it can be considered as a fuzzy number *3* (as expected since, $8 - 5 = 3$). □

4.5 MULTIPLICATION

As in the case of addition and subtraction, fuzzy number multiplication may be defined either as α-cut multiplication or through the extension principle. Using the α-cut representation of two numbers A, B, their product is defined as

$$A \cdot B \equiv \left[a_1^{(\alpha)}, a_2^{(\alpha)}\right] \cdot \left[b_1^{(\alpha)}, b_2^{(\alpha)}\right] \tag{4.5-1}$$

In general, the product of two intervals is a new interval whose left endpoint is the product of the left endpoints of the two intervals and the right endpoint is the product of the right endpoints of the two intervals. Thus, equation (4.5-1) is

$$A \cdot B = \left[a_1^{(\alpha)} \cdot b_1^{(\alpha)}, a_2^{(\alpha)} \cdot b_2^{(\alpha)}\right] \tag{4.5-2}$$

Alternatively, we define the product of A and B through the extension principle by fuzzifying the function $z(x, y) = x \cdot y$. The extension principle tells us that their product is a fuzzy set on z, denoted as $A \cdot B$, whose membership function is

$$\mu_{A \cdot B}(z) \equiv \bigvee_{z = x \cdot y} \left[\mu_A(x) \wedge \mu_B(y)\right] \tag{4.5-3}$$

Of course, equations (4.5-2) and (4.5-3) are equivalent in that they give us the same number $C = A \cdot B$.

A special case of fuzzy multiplication is the product of fuzzy number by crisp number. Let k be a crisp positive real number and A a fuzzy number defined over the universe of discourse of positive real numbers also. We define the product of k with A either as interval multiplication or through the extension principle. Crisp number k may be viewed as an interval also, a trivial interval whose left and right endpoints are the same—that is, $k = [k, k]$. We use equations (4.5-1) and (4.5-2) to obtain the product of k with A as

$$\begin{aligned} k \cdot A &\equiv [k, k] \cdot \left[a_1^{(\alpha)}, a_2^{(\alpha)}\right] \\ &= \left[ka_1^{(\alpha)}, ka_2^{(\alpha)}\right] \end{aligned} \tag{4.5-4}$$

Alternatively, we define the product of fuzzy number A with a crisp number k, $k \cdot A$, through the extension principle. It may be shown using equation

(4.5-3) that the membership function of $k \cdot A$ is

$$\mu_{k \cdot A}(x) = \mu_A\left(\frac{x}{k}\right) \tag{4-5-5}$$

where equations (4.5-4) and (4.5-5) give the same result.

Example 4.5 Multiplication of Two Fuzzy Numbers. Consider the triangular fuzzy numbers $A = 8$ and $B = 2$ defined over the positive real numbers as shown in Figure 4.5 (since both numbers are defined over the same universe of discourse we simply use x to indicate an element of the universe of discourse, instead of x, y, etc.). We want to compute a fuzzy number C which is the product of A and B—that is, $C = A \cdot B$. Let us do this through α-cut multiplication—that is, by parameterizing their membership functions and multiplying their α-cuts in the manner indicated by equation (4.5-2).

First, we write the analytical expressions for the membership functions of A and B:

$$\begin{aligned} \mu_A(x) &= 0, & x &\leq 4 \\ &= \tfrac{1}{4}x - 1, & 4 &\leq x \leq 8 \\ &= -\tfrac{1}{4}x + 3, & 8 &\leq x \leq 12 \\ &= 0, & x &\geq 12 \end{aligned} \tag{E4.5-1}$$

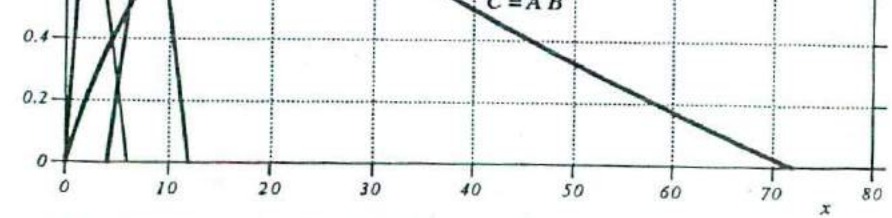


Figure 4.5 The product $C = A \cdot B$ of numbers $A = 8$ and $B = 2$ in Example 4.5.

and

$$\mu_B(x) = 0, \qquad x \le 0$$
$$= \tfrac{1}{2}x, \qquad 0 \le x \le 2$$
$$= -\tfrac{1}{4}x + \tfrac{3}{2}, \qquad 2 \le x \le 6$$
$$= 0 \qquad x \ge 6 \qquad \text{(E4.5-2)}$$

Next, we parameterize the membership functions in equations (E4.5-1) and (E4.5-2) in terms α (a procedure of renaming the left and right side of the membership functions and thus the endpoints of all intervals in terms of α). Let us take the left and right side of each membership function separately and rewrite it in terms of α. It should be noted that a given value of α is the same as the value of the membership function at that level. From equation (E4.5-1) we have that the left and right endpoints of A are

$$\alpha = \tfrac{1}{4}a_1^{(\alpha)} - 1 \Rightarrow a_1^{(\alpha)} = 4(\alpha + 1) \qquad \text{(E4.5-3)}$$

and

$$\alpha = -\tfrac{1}{4}a_2^{(\alpha)} + 3 \Rightarrow a_2^{(\alpha)} = -4(\alpha - 3) \qquad \text{(E4.5-4)}$$

Using equations (E4.5-3) and (E4.5-4) we obtain A as

$$A = \left[a_1^{(\alpha)}, a_2^{(\alpha)}\right] = [4(\alpha + 1), -4(\alpha - 3)] \qquad \text{(E4.5-5)}$$

Similarly, we parameterize the membership function of B and write its left and right endpoints at each α as

$$\alpha = \tfrac{1}{2}b_1^{(\alpha)} \Rightarrow b_1^{(\alpha)} = 2\alpha \qquad \text{(E4.5-6)}$$

and

$$\alpha = -\tfrac{1}{4}b_2^{(\alpha)} + \tfrac{3}{2} \Rightarrow b_2^{(\alpha)} = -4\left(\alpha - \tfrac{3}{2}\right) \qquad \text{(E4.5-7)}$$

Thus, from equations (E4.5-6) and (E4.5-7) the interval representation of B is

$$B = \left[b_1^{(\alpha)}, b_2^{(\alpha)}\right] = \left[2\alpha, -4\left(\alpha - \tfrac{3}{2}\right)\right] \qquad \text{(E4.5-8)}$$

Having the endpoints of A and B in terms of α, we multiply the two numbers using equation (4.5-2) and obtain

$$C = A \cdot B = \left[a_1^{(\alpha)} \cdot b_1^{(\alpha)}, a_2^{(\alpha)} \cdot b_2^{(\alpha)}\right]$$
$$= \left[4(\alpha + 1) \cdot 2\alpha, -4(\alpha - 3) \cdot \left(-4\left(\alpha - \tfrac{3}{2}\right)\right)\right]$$
$$= [8\alpha^2 + 8\alpha, 16\alpha^2 - 72\alpha + 72] \qquad \text{(E4.5-9)}$$

The interval representation of C is

$$C = \left[c_1^{(\alpha)}, c_2^{(\alpha)}\right] = \left[8\alpha^2 + 8\alpha, 16\alpha^2 - 72\alpha + 72\right] \quad \text{(E4.5-10)}$$

where the left and right endpoints in equation (E4.5-10) are functions for α. We can obtain the membership function of C as well. Equation (E4.5-10) provides us with left and right endpoints of each α-cut. The equation for the left-side membership function $\mu_C^L(x)$ is obtained by setting $c_1^{(\alpha)} = x$ and recalling that $\alpha = \mu_C^L(x)$. Thus, we obtain an equation involving $\mu_C^L(x)$, which is

$$8\left(\mu_C^L(x)\right)^2 + 8\mu_C^L(x) - x = 0 \quad \text{(E4.5-11)}$$

Solving quadratic equation (E4.5-11) for $\mu_C^L(x)$, we obtain two solutions and accept only the value of $\mu_C^L(x)$ in $[0, 1]$, ignoring the other one. The result is

$$\mu_C^L(x) = -\tfrac{1}{2} + \tfrac{1}{2}\sqrt{1 + \tfrac{1}{2}x} \quad \text{(E4.5-12)}$$

Similarly we obtain an equation for $\mu_C^R(x)$, the right side of the membership function of C, and solve it, keeping the solution which is within $[0, 1]$. The result is

$$\mu_C^R(x) = \tfrac{1}{2}\left(4.5 - \sqrt{(4.5)^2 - 4\left(4.5 - \tfrac{1}{16}x\right)}\right) \quad \text{(E4.5-13)}$$

The membership function of C is

$$\begin{aligned}
\mu_C(x) &= 0, && x \le 0 \\
&= -\tfrac{1}{2} + \tfrac{1}{2}\sqrt{1 + \tfrac{1}{2}x}, && 0 \le x \le 16 \\
&= \tfrac{1}{2}\left(4.5 - \sqrt{(4.5)^2 - 4\left(4.5 - \tfrac{1}{16}x\right)}\right), && 16 \le x \le 72 \\
&= 0, && x \ge 72 \quad \text{(E4.5-14)}
\end{aligned}$$

as shown in Figure 4.5. It should be noted that C has its peak point at crisp 16 and therefore may be considered a fuzzy number ***16***. It should also be noted that multiplying two fuzzy numbers results in a new number whose shape has been considerably changed, no longer having a triangular membership function with linear sides but in this case parabolic sides. Multiplication in general has the effect of "fattening" the lower part of the membership functions involved. □

4.6 DIVISION

We can find the *quotient* of two fuzzy numbers A and B either through interval division or by the extension principle. In terms of their α-cut representation, we write the *quotient* of the two numbers as

$$A \div B \equiv \left[a_1^{(\alpha)}, a_2^{(\alpha)}\right] \div \left[b_1^{(\alpha)}, b_2^{(\alpha)}\right] \tag{4.6-1}$$

In general, the *quotient* of two intervals is a new interval given by

$$\left[a_1^{(\alpha)}, a_2^{(\alpha)}\right] \div \left[b_1^{(\alpha)}, b_2^{(\alpha)}\right] = \left[\frac{a_1^{(\alpha)}}{b_2^{(\alpha)}}, \frac{a_2^{(\alpha)}}{b_1^{(\alpha)}}\right]$$

Hence, provided that $b_2^{(\alpha)} \neq 0$ and $b_1^{(\alpha)} \neq 0$, the *quotient* of A, B is

$$A \div B = \left[\frac{a_1^{(\alpha)}}{b_2^{(\alpha)}}, \frac{a_2^{(\alpha)}}{b_1^{(\alpha)}}\right] \tag{4.6-2}$$

Alternatively, we find the *quotient* of A and B through the extension principle by fuzzifying the function $z(x, y) = x \div y$, where x and y are crisp elements of the universe of discourse of A and B. The extension principle tells us that $A \div B$ is a fuzzy set with membership function

$$\mu_{A \div B}(z) \equiv \bigvee_{z = x \div y} \left[\mu_A(x) \wedge \mu_B(y)\right] \tag{4.6-3}$$

The results obtained through equations (4.6-4) and (4.6-2) are of course the same. Equation (4.6-3) may be used in the manner shown in Example 4.2. We construct a table such as Table 4.4 and proceed as outlined in the example. A word of caution: Fuzzy number division is not the reverse of multiplication; that is, generally it is not true that $(A \div B) \times C = A$.

Example 4.6 Division of Fuzzy Numbers. Consider the triangular fuzzy numbers $A = 8$ and $B = 2$ used in Example 4.5. Let us find $C = A \div B$ using interval division. The analytical expressions for the membership functions of A and B are given in Example 4.5 [equations (E4.5-1) and (E4.5-2)], and their parameterized interval representation is found in equations (E4.5-5) and (E4.5-8), which for convenience we repeat here:

$$A = \left[a_1^{(\alpha)}, a_2^{(\alpha)}\right] = [4(\alpha + 1), -4(\alpha - 3)] \tag{E4.6-1}$$

$$B = \left[b_1^{(\alpha)}, b_2^{(\alpha)}\right] = \left[2\alpha, -4(\alpha - \tfrac{3}{2})\right] \tag{E4.6-2}$$

Thus their *quotient* $C = A \div B$ is obtained using equation (4.6-2):

$$C = A \div B = \left[\frac{a_1^{(\alpha)}}{b_2^{(\alpha)}}, \frac{a_2^{(\alpha)}}{b_1^{(\alpha)}}\right]$$

$$= \left[\frac{4(\alpha+1)}{\left(-4\left(\alpha-\frac{3}{2}\right)\right)}, \frac{-4(\alpha-3)}{2\alpha}\right] \qquad \text{(E4.6-3)}$$

The α-cut representation of C is

$$C = \left[c_1^{(\alpha)}, c_2^{(\alpha)}\right] = \left[-\frac{(\alpha+1)}{\left(\alpha-\frac{3}{2}\right)}, -\frac{2(\alpha-3)}{\alpha}\right] \qquad \text{(E4.6-4)}$$

where the left and right endpoints are functions of α. We may also express C in terms of a membership function by deriving equations for the left and right sides of the membership function as we did in Example 4.5. Equation (E4.6-4) gives us the endpoints of the interval of each α-cut. The equation of the left side is obtained by setting $c_1^{(\alpha)} = x$ and recalling that $\alpha = \mu_C^L(x)$, where, $\mu_C^L(x)$ is the left side membership function for C. The result is

$$\mu_C^L(x) = \frac{\frac{3}{2}x - 1}{x + 1} \qquad \text{(E4.6-5)}$$

Similarly we obtain an equation for $\mu_C^R(x)$, the right side of the membership function of C, and solve it to obtain

$$\mu_C^R(x) = \frac{6}{x + 2} \qquad \text{(E4.6-6)}$$

The quotient is shown in Figure 4.6, and the analytical description of the

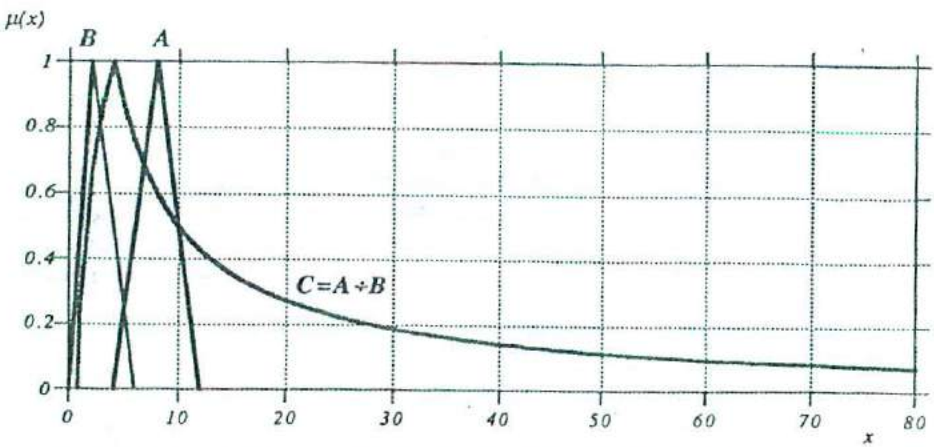


Figure 4.6 The quotient $C = A \div B$ of the fuzzy numbers $A = 8$ and $B = 2$ in Example 4.6.

membership function of C is

$$\begin{aligned}\mu_C(x) &= 0, & x \le 0\\ &= \frac{\frac{3}{2}x - 1}{x+1}, & 0 \le x \le 4\\ &= \frac{6}{x+2}, & 4 \le x \le 72\\ &= 0, & x \ge 72\end{aligned} \qquad \text{(E4.6-7)}$$

It should be noted from equation (E4.6-7) that the quotient is a new fuzzy number that no longer has a triangular shape with linear sides. As may be seen from the figure, the fuzzy number C only asymptotically reaches zero and hence we may consider the use of a *level fuzzy set* (Chapter 2) in order to limit and exclude trivially small grades of membership—for example, less than 0.2. □

4.7 MINIMUM AND MAXIMUM

The minimum and maximum of two fuzzy numbers A, B result in finding the smallest and the biggest one, respectively, and may be defined either through their interval representation or by the extension principle. In interval arithmetic the minimum of two intervals is a new interval whose left endpoint is the minimum of the left endpoints of the original intervals and whose right endpoint is the minimum of the right endpoints of the two intervals. Thus the minimum of A, B is a new number, $A \wedge B$, given by

$$\begin{aligned}A \wedge B &\equiv \left[a_1^{(\alpha)}, a_2^{(\alpha)}\right] \wedge \left[b_1^{(\alpha)}, b_2^{(\alpha)}\right]\\ &= \left[a_1^{(\alpha)} \wedge b_1^{(\alpha)}, a_2^{(\alpha)} \wedge b_2^{(\alpha)}\right]\end{aligned} \qquad \text{(4 7-1)}$$

Alternatively, the minimum of two fuzzy numbers may be obtained through the extension principle. The membership function of $A \wedge B$ is

$$\mu_{A \wedge B}(z) \equiv \bigvee_{z = x \wedge y} \left[\mu_A(x) \wedge \mu_B(y)\right] \qquad \text{(4.7-2)}$$

In an analogous manner we define the maximum of two fuzzy numbers A and B, recalling that in interval arithmetic the maximum of two intervals is a new interval whose left endpoint is the maximum of the left endpoints of the original intervals and whose right endpoint is the maximum of the right endpoints of the two intervals. Thus the maximum $A \vee B$ is given by

$$\begin{aligned}A \vee B &\equiv \left[a_1^{(\alpha)}, a_2^{(\alpha)}\right] \vee \left[b_1^{(\alpha)}, b_2^{(\alpha)}\right]\\ &= \left[\alpha_1^{(\alpha)} \vee b_1^{(\alpha)}, a_2^{(\alpha)} \vee b_2^{(\alpha)}\right]\end{aligned} \qquad \text{(4.7-3)}$$

Alternatively, by the extension principle the membership function of the

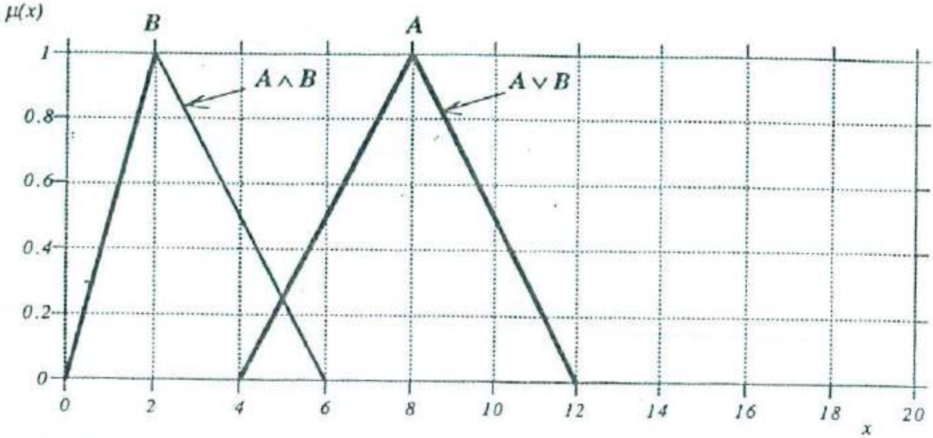


Figure 4.7 The minimum and maximum of the two numbers $A = 8$ and $B = 2$ used in Example 4.6.

maximum of the two numbers A and B is

$$\mu_{A \vee B}(z) = \bigvee_{z = x \vee y} [\mu_A(x) \wedge \mu_B(y)] \tag{4.7-4}$$

It should be noted that the *maximum* and *minimum* of two fuzzy numbers are different than the maximum and minimum of membership functions used in connection with the *union* and *intersection* of two fuzzy sets. Let us illustrate this by finding the minimum of the numbers $A = 8$ and $B = 2$ used in Examples 4.5 and 4.6 and redrawn in Figure 4.7. Equations (4.7-1) or (4.7-2) do not give us the little wedge between A and B, which is the *intersection* of A and B. They will simply give us the number $B = 2$ itself, which is the smallest of the two fuzzy numbers. Similarly the largest of the numbers is found by using the maximum operation of either equation (4.7-3) or (4.7-4), which is simply the number $A = 8$, as shown in Figure 4.7. For more intricately overlapping membership functions the maximum or minimum may not simply be a number with the membership function of either A or B, but may have a totally new shape (Kaufmann and Gupta, 1991).

REFERENCES

Dubois, D., and Prade, H., *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, Boston, 1980.

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

Kaufmann, A., and Gupta, M. M., *Introduction to Fuzzy Arithmetic*, Van Nostrand Reinhold, New York, 1991.

Klir, G. J., and Folger, T. A., *Fuzzy Sets, Uncertainty, and Information*, Prentice-Hall, Englewood Cliffs, NJ, 1988.

Pal, S. K., and Majumder, D. K. D, *Fuzzy Mathematical Approach to Pattern Recognition*, John Wiley & Sons, New York, 1986.

Sanchez, E., Non Standard Fuzzy Arithmetic, in *Between Mind and Computer-Fuzzy Science and Engineering*, P-Z. Wang, and K-F. Loe, eds. World Scientific, Singapore, 1993, pp. 271–282.

Terano, T., Asai, K., and Sugeno, M., *Fuzzy Systems Theory and Its Applications*, Academic Press, Boston, 1992.

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

PROBLEMS

1. The fuzzy numbers A and B are given by

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

Subtract B from A to give fuzzy number C. Draw a sketch of C.

2. Multiply fuzzy numbers A and B of Problem 1. Draw a sketch of C.

3. Divide fuzzy number A by fuzzy number B where the fuzzy numbers are defined in Problem 1. Draw a sketch of C.

4. Modify Example 4.2 to subtract the two fuzzy numbers using the extension principle.

5. Consider the fuzzy numbers A and B described by the membership functions:

$$\begin{aligned}
\mu_A(x) &= 0, & x &\le 8,\\
&= \frac{1}{10}x - \frac{8}{10}, & 8 &\le x \le 18,\\
&= -\frac{1}{14}x + \frac{32}{14}, & 18 &\le x \le 32,\\
&= 0, & x &> 32,\\
\mu_B(x) &= 0, & x &\le -3,\\
&= \frac{1}{9}x - \frac{1}{3}, & -3 &\le x \le 6,\\
&= -\frac{1}{18}x + \frac{4}{3}, & 6 &\le x \le 24,\\
&= 0, & x &> 24
\end{aligned}$$

Compute:

(a) $A\ (+)\ B$,

(b) $A\ (-)\ B$,

(c) $A\ (\div)\ B$.

6. Repeat the computations in Problem 5 for the fuzzy numbers A and B given below, and using C state and show the distributivity property (with respect to addition and multiplication)

$$A = 0.6/1 + 0.8/2 + 1.0/3 + 0.6/4$$

$$B = 0.5/0 + 0.7/1 + 0.9/2 + 1.0/3 + 0.4/4$$

$$C = 0.7/1 + 0.8/2 + 1.0/3 + 0.3/4$$

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*

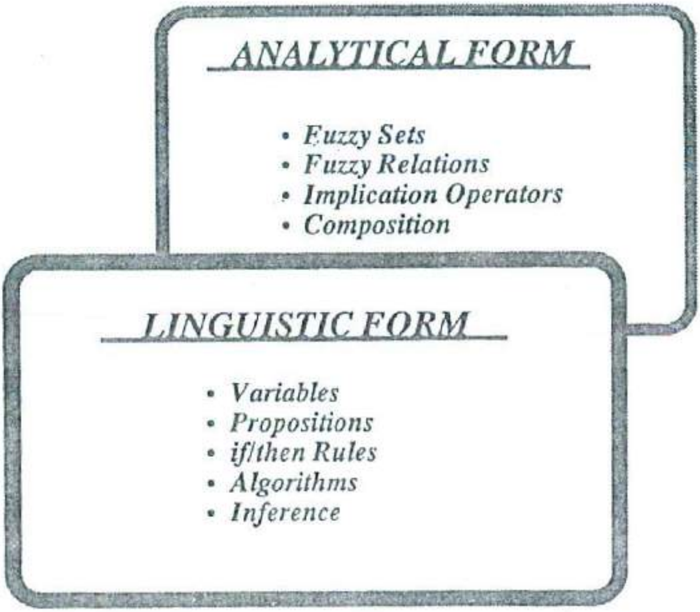


Figure 5.1 Fuzzy linguistic descriptions possess a linguistic form as well as a background analytical form involving fuzzy set operations.

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

$$\begin{array}{l} (a_1, b_1) \\ (a_2, b_2) \\ \dots \\ (a_i, b_i) \\ \dots \\ (a_n, b_n) \end{array} \qquad (5.1\text{-}1)$$

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:

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

[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*.

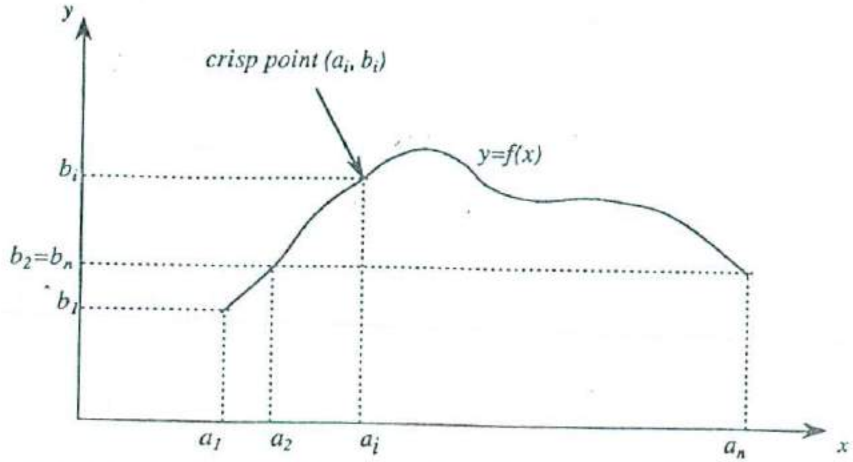


Figure 5.2 The function $y = f(x)$ may be thought of as a collection of crisp points (a_i, b_i), and each point may also be articulated as a crisp *if / then* rule.

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 circled area around (a_i, b_i) which may be thought of as an *area-cum-point*, an

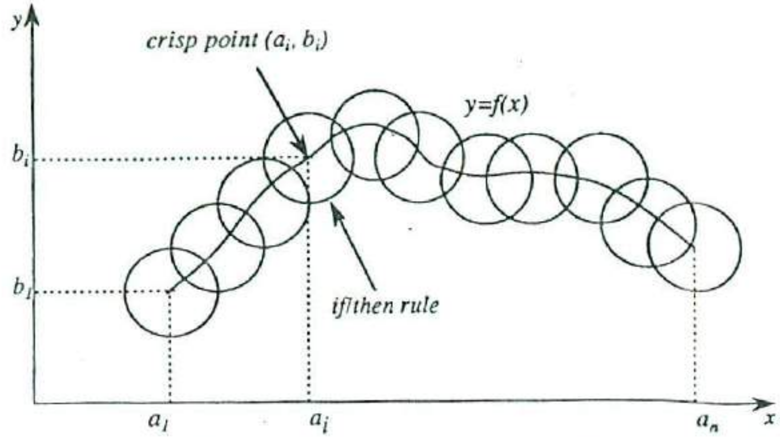


Figure 5.3 Building a linguistic description of the function $y = f(x)$.

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,

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

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 $R_a(x, y)$ 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*

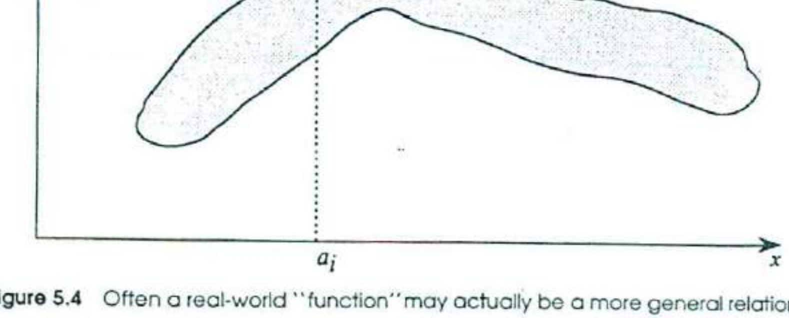


Figure 5.4 Often a real-world "function" may actually be a more general relation.

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

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

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

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

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*,

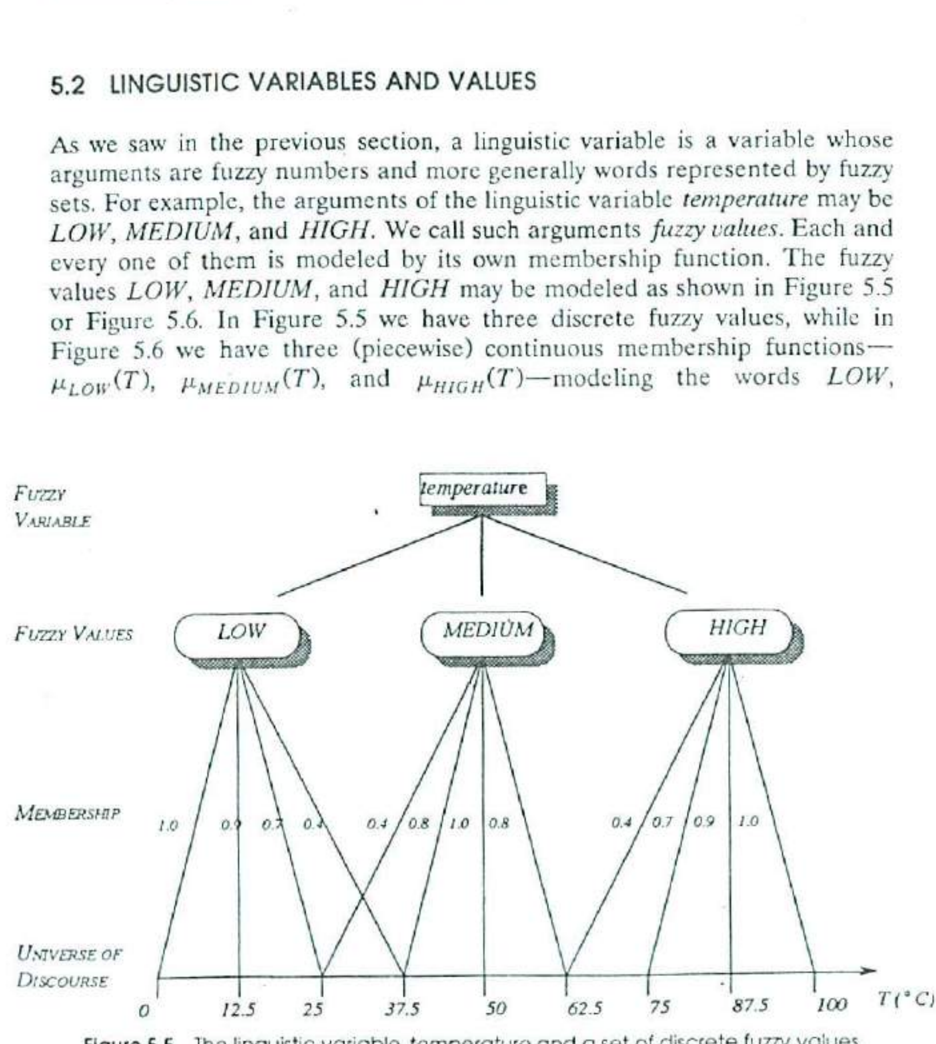


Figure 5.5 The linguistic variable *temperature* and a set of discrete fuzzy values.

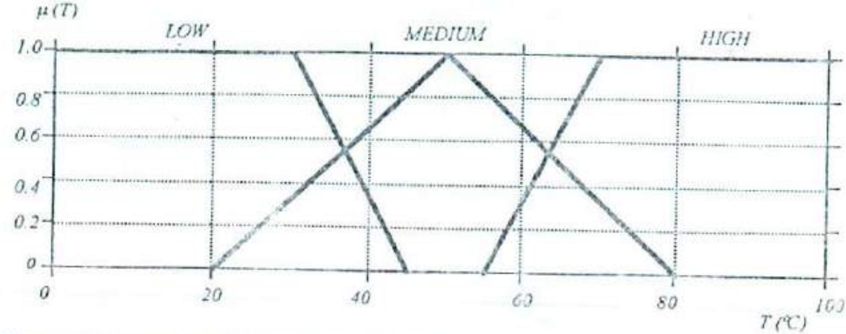


Figure 5.6 Membership functions $\mu(T)$ used for describing the primary values, *LOW, MEDIUM*, and *HIGH*, of the linguistic variable *temperature*.

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

[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 pr*[illegible], or the *term set*. We will mostly use the name *fuzzy value* or simply *value*. Linguistic varia[illegible] also called *fuzzy variables*.

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.

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,

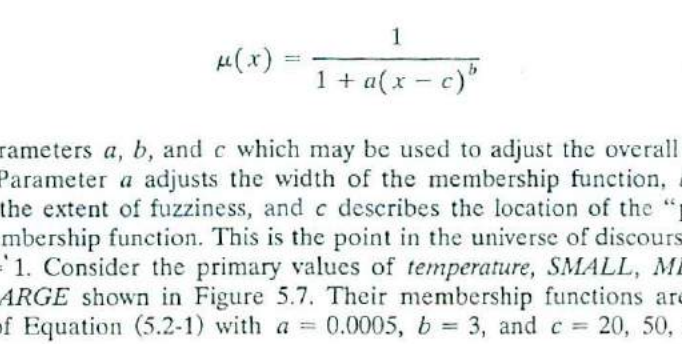


Figure 5.7 Adjustable membership functions for modeling primary values.

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:

$$S(x; \alpha, \beta, \gamma) = 0 \qquad \text{for } x \leq \alpha$$

$$S(x; \alpha, \beta, \gamma) = 2\left(\frac{x - \alpha}{\gamma - \alpha}\right)^2 \qquad \text{for } \alpha \leq x \leq \beta$$

$$S(x; \alpha, \beta, \gamma) = 1 - 2\left(\frac{x - \gamma}{\gamma - \alpha}\right)^2 \qquad \text{for } \beta \leq x \leq \gamma \qquad (5.2\text{-}3)$$

$$S(x; \alpha, \beta, \gamma) = 1 \qquad \text{for } x \geq \gamma$$

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

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

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

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

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

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

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.

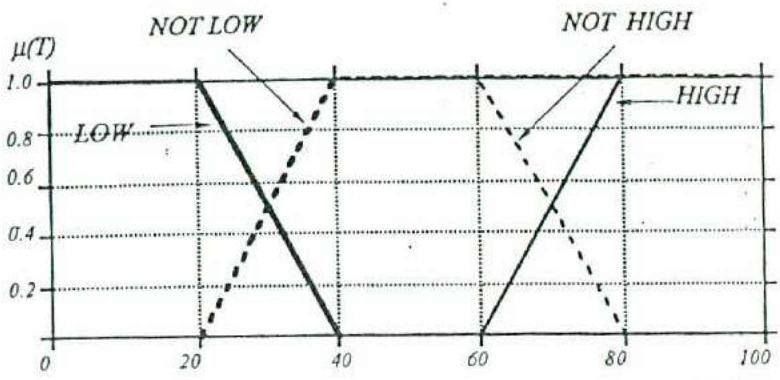
The membership function of a compound value produced by negating another value is the complement of the membership function of the original value—that is,

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

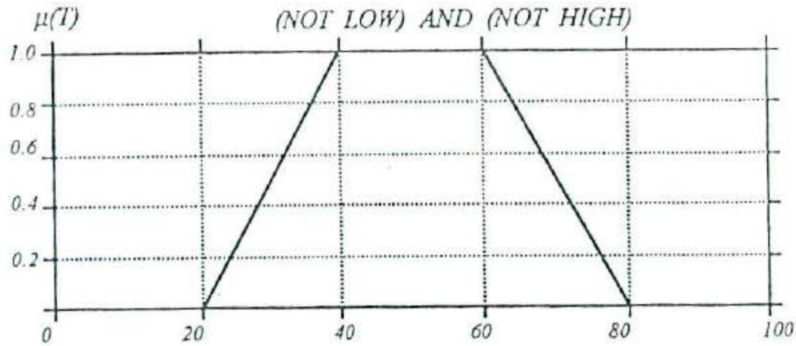
The semantics of the modifier *NOT* are fairly straightforward, and it may be used very much as negation is used in natural language—for example, "*temperature* is (*NOT HIGH*)."

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:

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



(a)



(b)

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	$\mu_{DIL(A)}(x) \equiv [\mu_A(x)]^{1/2}$
INTENSIFIED 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,

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

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

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

The Standard Sequence Implication Operator

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

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

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

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

Gödelian Implication Operator

The *Gödelian* implication operator is defined as

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

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 $\mu_B(y)=1$ $\mu_B(x)$, if $\mu_A(y)=1$ 0, if $\mu_A(y)<1, \mu_B(y)<1$
ϕ_s, *Standard Sequence*	1, if $\mu_A(x) \le \mu_B(y)$ 0, if $\mu_A(x) > \mu_B(y)$
ϕ_Δ, *Gougen*	1, if $\mu_A(x) \le \mu_B(y)$ $\frac{\mu_B(y)}{\mu_A(x)}$, if $\mu_A(x) > \mu_B(y)$
ϕ_g, *Gödelian*	1, if $\mu_A(x) \le \mu_B(y)$ $\mu_B(y)$, if $\mu_A(x) > \mu_B(y)$

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

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

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)

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

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,

$$\begin{array}{ll} \textit{if } x \textit{ is } A & \textit{then } y \textit{ is } B \\ & y \textit{ is } B' \\ \hline x \textit{ is } A' & \end{array} \tag{5.4-4}$$

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)

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

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

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

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':

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

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

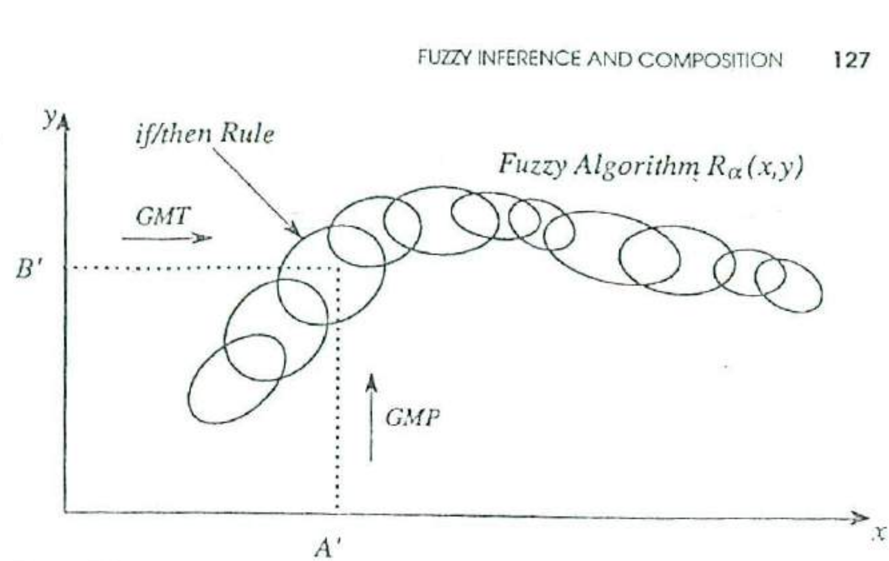


Figure 5.9 GMP and GMT are procedures for evaluating fuzzy linguistic descriptions.

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:

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

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

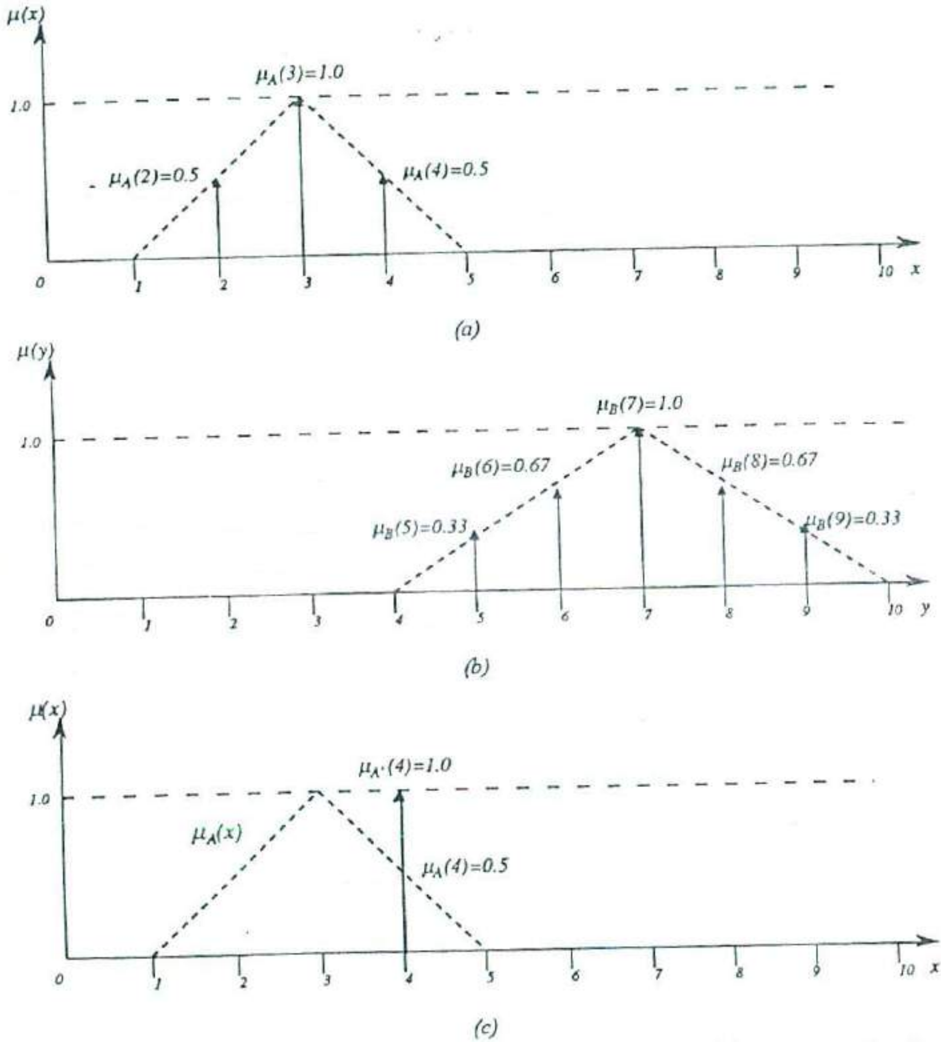


Figure 5.10 (*a*) The membership function of the antecedent *A*. (*b*) The fuzzy value *B* of the consequent. (c) A fuzzy *A'* that approximately matches the antecedent in Example 5.1.

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 [illegible] min implication operator. Having discrete fuzzy values we use discrete fuzzy relations as well. From Table 5.2 we see that the membership function of the implication relation is given by

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

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

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

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

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. □

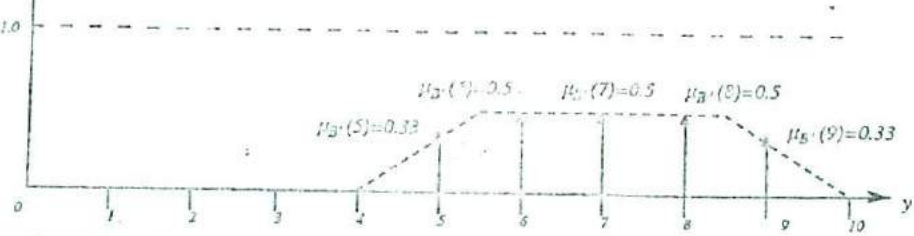


Figure 5.11 The fuzzy set B' produced by evaluating the linguistic description of Example 5.1.

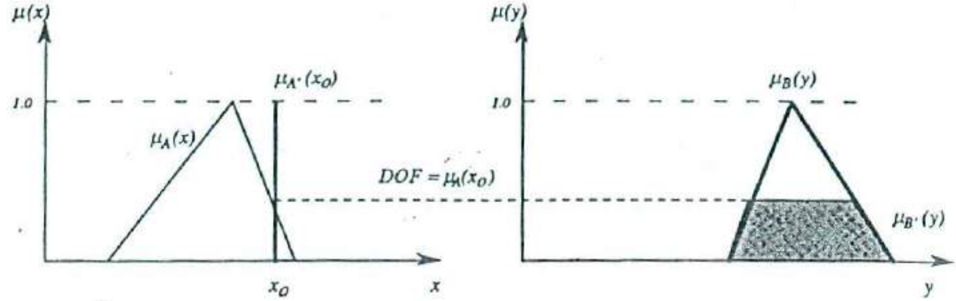


Figure 5.12 When the Mamdani min implication operator is used to model an implication relation, GMP clips the membership function of the consequent by the DOF of the rule.

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

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

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

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

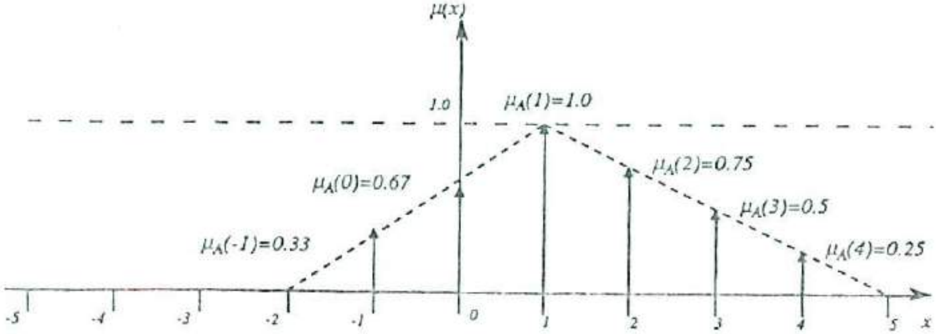
$$= 0.50/(-4) + 1.0/(-3) + 0.67/(-2) + 0.33/(-1) \quad \text{(E5.2-2)}$$

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

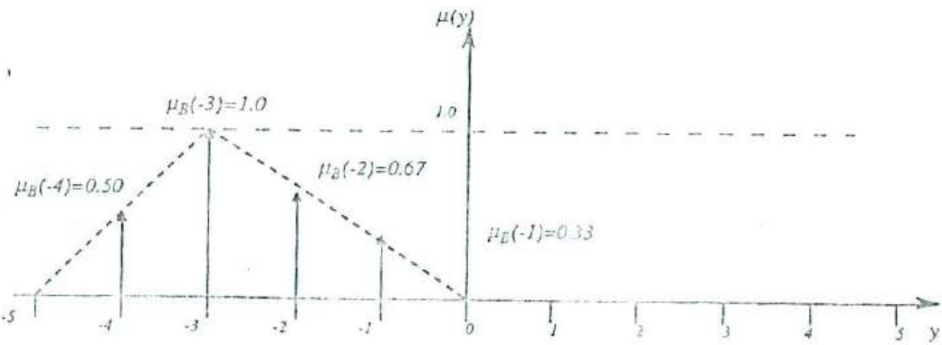
Using equations (5.4-2) and (5.4-3) for GMP we can compute the membership function of value B'. First, however, we have to obtain the membership function of the implication relation, $\mu(x, y)$, using the Larsen product fuzzy implication operator ϕ_p (see Table 5.2). The implication relation has the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

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

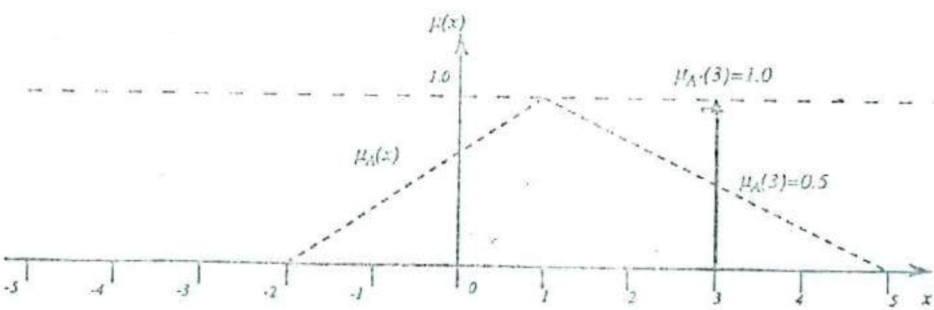
and plugging in numbers from equations (E5.2-1) to (E5.2-3) we obtain the



(a)



(b)



(c)

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.

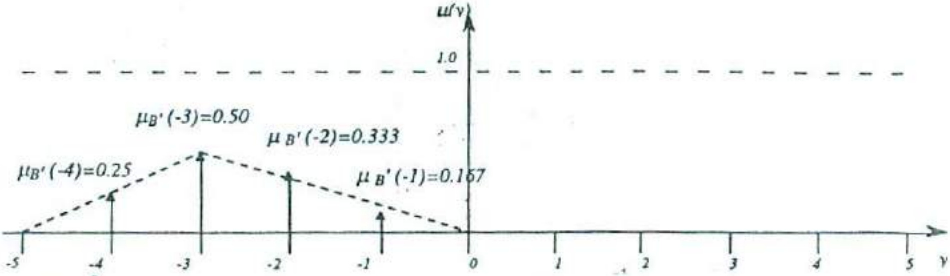


Figure 5.14 The fuzzy set B' produced by evaluating the linguistic description in Example 5.2.

implication relation

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

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 (×). Thus we have

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

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

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

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

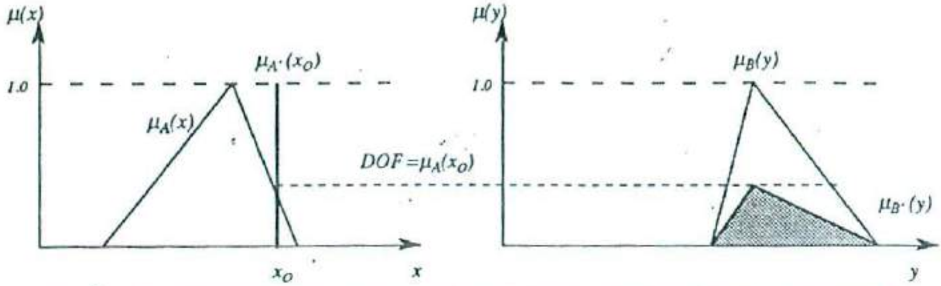


Figure 5.15 When GMP is used to evaluate a rule whose implication relation is modeled by the Larsen product, the membership function of the consequent is scaled by the DOF.

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

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

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*

$$R_a(x, y) = \int_{(x,y)} \mu_a(x, y)/(x, y) \tag{5.5-2}$$

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

(5.5-1) through GMP formally stated as

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

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

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

The membership function of B' is

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

Then inverse problem is solved through GMT, stated as

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

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

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

with the membership function of A' given by

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

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

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

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

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

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

if $\quad {}_{11}$ *AND* x_2 *is* A_{21} *AND* ⋯ *AND* x_{1m} *is* A_{m1} *then y is* B_1 *ELSE*

if $\quad {}_{12}$ *AND* x_2 *is* A_{22} *AND* ⋯ *AND* x_m *is* A_{m2} *then y is* B_2 *ELSE*

…

if $\quad A_{1n}$ *AND* x_2 *is* A_{2n} *AND* ⋯ *AND* x_m *is* A_{mn} *then y is* B_n

(5.5-14)

The f zy algorithm of (5.5-14) is analytically described by an algorithmic relatio of the form

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

and when discrete fuzzy sets are used we obtain

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

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,

if x_1 *is* A_{11} *AND* x_2 *is* A_{21} *AND* ⋯ *AND* x_{1m} *is* A_{m1} *then y is* B_1 *ELSE*

if x_1 *is* A_{12} *AND* x_2 *is* A_{22} *AND* ⋯ *AND* x_m *is* A_{m2} *then y is* B_2 *ELSE*

…

if x_1 *is* A_{1n} *AND* x_2 *is* A_{2n} *AND* ⋯ *AND* x_m *is* A_{mn} *then y is* B_n

x_1 *is* A'_1 $\quad$ x_2 *is* A'_2 $\quad$ … $\quad$ x_m *is* A'_m

---

y is B'

(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.

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

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.

6

FUZZY CONTROL

6.1 INTRODUCTION

Fuzzy control primarily refers to the control of processes through fuzzy linguistic descriptions. Since 1974, when E. H. Mamdani and S. Assilian (Mamdani, 1974) demonstrated that fuzzy *if/then* rules could regulate a model steam engine, a great number of fuzzy control applications have been successfully deployed. The list is very long and growing and includes cement kilns, subway trains, unmanned helicopters, autonomous mobile robots, process heat exchangers, and blast furnaces (Mamdani, 1977; Ostergaard, 1982; Yasunobu and Miyamoto, 1985; King and Karonis, 1988).[1] In the 1970s and early 1980s most applications were minicomputer-based, often found in the process industry in areas where automatic control was rather difficult to realize and hence left in the hands of human operators. More recently, with the advent of fuzzy microprocessors, a growing number of fuzzy control applications have emerged in consumer electronics and home appliances such as hand-held cameras, vacuum cleaners, air conditioners, and washing machines (Hirota, 1993; Yamakawa, 1989; Schwartz, 1992; Terano et al., 1992).

In this chapter we begin by reviewing conventional process control in order to establish the relevant context and proceed to fuzzy control, a subject we view primarily as an application of fuzzy linguistic descriptions (Chapter 5). Of course, the appropriate choice of controller in engineering applications

[1] There are a number of excellent books available on fuzzy control. The interested reader may want to consult, for example, Driankov et al., 1993; Pedrycz, 1993; Harris et al., 1993; Yager and Filev, 1994; and Wang, 1994.

is made not as much by a commitment to a particular methodology or technology as by careful examination of the needs and features of a given application. In fact, some of the most successful applications of fuzzy control have been in conjunction with conventional controllers such as the *proportional integral derivative* (PID) controller (Lee, 1990a, b). In fuzzy control we are concerned with two broad questions: How can we implement a control strategy as a fuzzy linguistic description? and What are the crucial factors involved in fuzzy algorithmic synthesis and analysis? Although fuzzy linguistic descriptions are a subject of wider interest than the replacement or enhancement of PID controllers, their application to control serves to illustrate some of the basic ideas we encountered in earlier chapters.

Consider the simple process system shown in Figure 6.1. Here, a tank is filled with liquid flowing from a pipe at the top (inlet flow). Liquid leaves the tank through a pipe at the bottom (outlet flow). The upper pipe is fitted with control valve A, used to adjust inlet flow, and the bottom pipe with valve B is assumed to remain at a preset position. A *controller* maintains the liquid in the tank at the desired level. By *process* here we mean the tank, the liquid, the pipes, and the valves. The term *process control system* refers to the

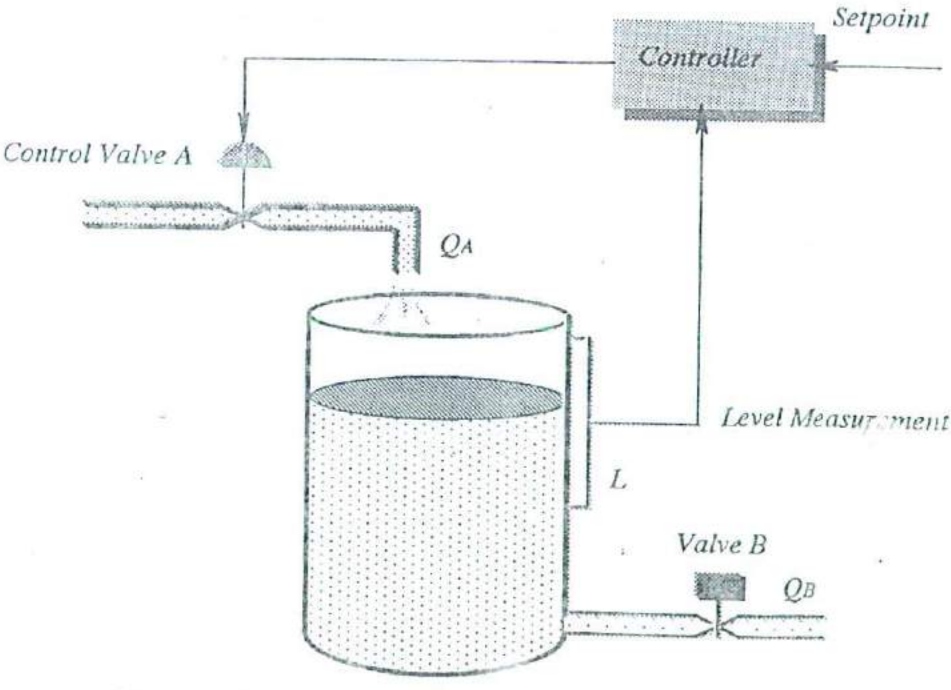


Figure 6.1 A process system with level control through control valve A.

process plus the controller and any required components for measurement and actuation.

The purpose of any process control system is to regulate some *dynamic variable* or *variables* of the process. In the liquid level process control system shown in Figure 6.1, the dynamic variable is the liquid level L, a process parameter that depends on other parameters and thus suffers changes from many different inputs. We select one of these other parameters to be our *controlling parameter*—in this case control valve A, the adjustment of which leads to control of flow rate, Q_A. Liquid level depends on flow rates via control valve A and valve B, ambient temperature T_A (not shown), liquid temperature T_l (also not shown), and the physical condition of valves A and B. This dependence may be described by a process relation of the form

$$L = f(Q_A, Q_B, T_a, T_l) \qquad (6.1\text{-}1)$$

where Q_A is the flow rate through control valve A, Q_B is the flow rate through valve B, T_a is the ambient temperature, and T_l is the liquid temperature. In many cases the relationship of equation (6.1-1) is not analytically known and actually may not be a function (a *many-to-one mapping*) but instead a more general relation (a *many-to-many mapping*) as we discussed in Chapter 5.

The input to the controller is usually not L itself but instead the error e between a measured indication of L, denoted as y, and a *setpoint* or *reference* value r representing the desired value of the dynamic variable. The controller's *output* or *manipulated variable* is denoted by u and is a signal representing action to be taken when the measured value of the dynamic variable y deviates from reference r. Thus, the output of the controller u serves as input to the process. The error $e = r - y$ is actually *smoothed* and *scaled* before input to the controller. Smoothing is performed in sampled systems in order to avoid the instantaneous changes during sampling that misled the general direction of change for the variable. Such a smoothing function may be defined recursively as $(e_k) \equiv 0.9e_{k-1} + 0.1e_k$, where e_k is the error value at time $t = k$. Scaling is required in order to transform instrument values to a predetermined interval or transform them to a range of numbers that correspond to natural magnitudes.

The most common controller in the process industry is the *PID controller*, where the control relation associated with equation (6.1-1) takes the form

$$u(t) = K_P e(t) + K_P K_I \int_{t=0}^{t} e(t)\, dt + K_P K_D \frac{de(t)}{dt} + u(0) \qquad (6.1\text{-}2)$$

where K_P is the *controller gain* representing a proportionality constant between error and controller output (dimensionless), K_I is the *reset constant* relating the rate to the error in units of $[\%/(\% - \text{sec})]$, K_D is the *rate constant* (or *derivative gain constant*) in units of $[(\% - \text{sec})/\%]$, and $u(0)$ is the controller output at $t = 0$ (when a deviation from setpoint starts).

The first term in equation (6.1-2) is called the *proportional term*, and if it was the only term in the equation it would represent a mode of control where the output of the controller $u(t)$ is changed in proportion to the error $e(t)$, which is the percent deviation from the setpoint. The second term is called the *integral term* and represents a mode of control where the present controller output depends on the history of errors from when observations started at $t = 0$. The amount of corrective action due to integral mode is directly proportional to the length of time that the error has existed. The reset constant K_I expresses the scaling between error and controller output. A large value of K_I means that a small error produces a large rate of change of u and vice versa. If this term alone was used in equation (6.1-2), in addition to the constant $u(0)$, then we would have a mode of control called *integral mode*. The third term in equation (6.1-2) represents the *derivative mode of control*. This mode provides that the controller output depends on the rate of change of error. Derivative mode tends to minimize oscillation of the system and prevent overshooting. Since derivative control is based solely on the rate of change of error, the controlled variable can stabilize at a value different from r, a condition termed "offset." In pure derivative mode the output depends upon the rate at which the error is changed and not on the value of the error. Integral control is used to address situations when permanent offset or slow returns to desired values cannot be tolerated. The combination of these three modes is called *proportional integral derivative* or (PID) control. PID is a powerful composite mode of control that has been used for virtually any linear process condition.

The process of adjusting the coefficients of each mode of control in equation (6.1-2) is called *tuning*. There are several methods for determining the optimum value of these gains such as *frequency response methods* and the *Ziegler–Nichols method* (Johnson, 1977). Fuzzy and neural approaches with adaptive characteristics have also been used for PID tuning and more generally for emulating and enhancing PID controllers (Matia et al., 1992; Maeda and Murakami, 1992; Shoureshi and Rahmani, 1992; He et al., 1993).

Example 6.1 PID Level Control. Consider the process control system shown in Figure 6.1. Suppose that we control the liquid level in the tank by adjusting control valve A (inlet flow) through a PID controller. The output of the controller $u(t)$ is based on the error $e(t)$—that is, the difference between a reference value r and the measured value of level y. The output of the controller is given by equation (6.1-2) as

$$u(t) = K_P e(t) + K_P K_I \int_{t=0}^{t} e(t)\,dt + K_P K_D \frac{de(t)}{dt} + u(0) \quad \text{(E6.1-1)}$$

with the following values for the various gains and initial controller output: $K_P = -1.3$, $K_I = 0.5[\%/(\% - \text{min})]$, $K_D = 1.9[(\% - \text{min})/\%]$, and $u(0) = 50\%$.

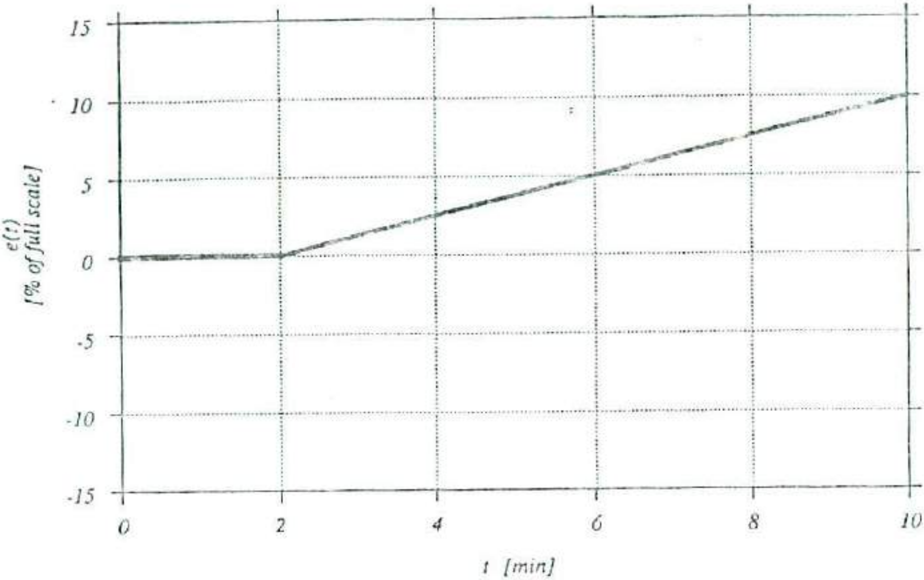


Figure 6.2 Error introduced to the PID level controller of Example 6.1.

Suppose that the error shown in Figure 6.2 is introduced to the system at $t = 2$ min. Such an error may be due to any change in process parameters—for example, an unforeseen change in the position of valve B—since valve B is not under control. The equation of error as function of time is

$$e(t) = 1.25t - 2.5 \tag{E6.1-2}$$

Using equation (E6.1-2) in equation (E6.1-1) the output of the controller after $t = 2$ is given by

$$u(t) = -1.3[1.25t - 2.5] - 1.3(0.5\ \text{min}^{-1})\int_{t=2}^{t}[1.25t - 2.5]\,dt$$

$$- 1.3(1.9\ \text{min})\frac{d}{dt}[1.25t - 2.5] + 50\% \tag{E6.1-3}$$

The first term in equation (6.1-3) represents the *proportional mode* of the controller, the second term the *integral mode*, and the third term the *derivative mode*. Let us call them $u_P(t)$, $u_I(t)$, and $u_I(t)$, respectively. Figure 6.3 shows the response due to each mode and the total response of the controller $u(t)$, which is the sum of the three terms plus the initial output of the controller, in this case 50%. Looking at Figure 6.3 we note that at the

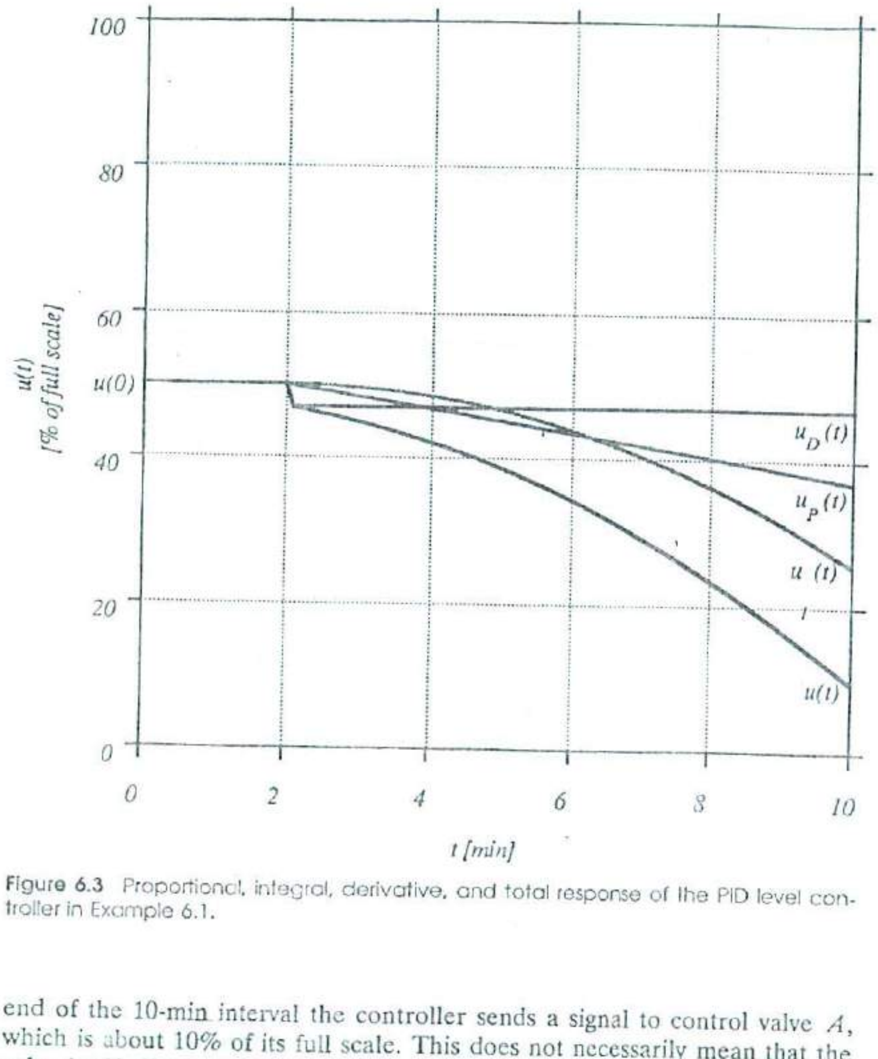


Figure 6.3 Proportional, integral, derivative, and total response of the PID level controller in Example 6.1.

end of the 10-min interval the controller sends a signal to control valve A, which is about 10% of its full scale. This does not necessarily mean that the valve itself allows at that time 10% of full flow to the tank. Different valves have different characteristics, often nonlinear. The relation between a dynamic variable and its transduced equivalent, although desired to be linear for many transducers, is almost always actually nonlinear. As an example, suppose that control valve A is an *equal-percentage* valve. In such valves a given percent change in the valve's stem position (which is what actually the controller controls) produces an equivalent change in flow, hence the name *equal-percentage*. Generally this type of valve does not shut off flow completely in its limit of stem travel. Let Q denote the flow rate through the valve (in m^3/sec), Q_{min} the minimum flow when the stem of the valve is at the lowest limit of its travel, and Q_{max} the flow rate when the valve is fully open. The ratio $R = (Q_{max}/Q_{min})$ is called the *rangeability* of the valve; it is a parameter specific to a given valve. The actual flow at any given time varies nonlinearly with rangeability and is often given by an exponential expression of the form

$$Q = Q_{min} R^{u/u_{max}} \tag{E6.1-4}$$

where u/u_{max} is the ratio of the actual to the maximum control signal sent to the actuator (actually the valve stem position at any given time divided by the maximum position of valve stem). Suppose that control valve A has rangeability $R = 30$. Thus when the control signal is 10% of full range—that is, $(u/u_{max}) = 0.1$—the flow rate according to equation (E6.1-4) becomes

$$Q = Q_{min}(30)^{0.1} = 1.4Q_{min} \tag{E6.1-5}$$

We can see from equation (E6.1-5) that at the end of the 10-min interval the controller output is 10% of its maximum value even though the flow through control valve A is 1.4 times the minimum flow through the valve. Actuators in general have such nonlinear characteristics, and furthermore their characteristics change due to aging or other environmental factors. Some of the difficulties in the field utilization of control algorithms, such as the PID level controller here, arise from the collective impact of such changes. Their extent and nature may not be fully known when the controller is designed, tested, and initially deployed. In the course of time the control engineer has to make various judgments about the overall performance of the process control system and, in collaboration with operations and maintenance personnel, intervene to retune gains, repair or replace equipment, revise procedures for operation, and so on. An objective of linguistic control is to make this entire process somewhat easier. It may therefore be seen as irony in the choice of words, but indeed a benefit of *fuzzy* control is introducing even more *clarity* to the development, evaluation, and maintenance of control systems. □

6.2 FUZZY LINGUISTIC CONTROLLERS

The core of a fuzzy controller is a linguistic description prescribing appropriate action for a given state. As we saw in Chapter 5, fuzzy linguistic descriptions involve associations of fuzzy variables and procedures for inferencing. Whereas in a conventional PID controller what is modeled is the physical system or process being controlled, in fuzzy controllers the aim is to incorporate expert human knowledge in the control algorithm. In this sense, a fuzzy controller may be viewed as a real-time expert system—that is, a model of the thinking processes an expert might go through in the course of manipulating the process.

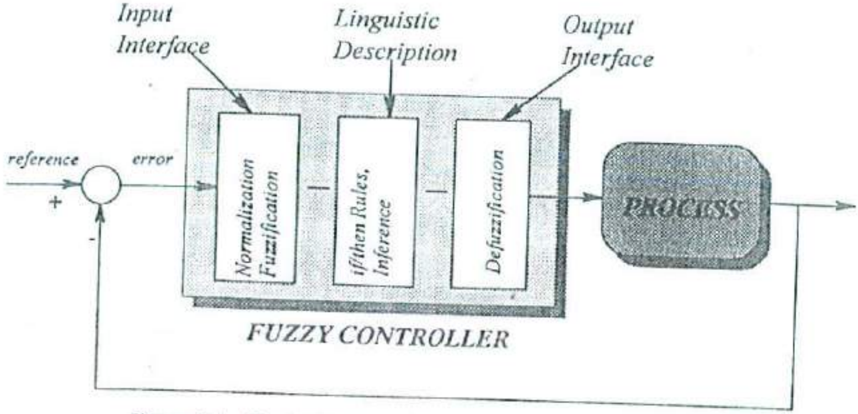


Figure 6.4 Block diagram of fuzzy process control system.

The basic structure of a fuzzy controller is outlined in Figure 6.4. The fact that measuring devices give crisp measurements and that actuators require crisp inputs calls for two additional considerations when linguistic descriptions are employed for control purposes: *fuzzifying* the input of the controller and *defuzzifying* its output. *Fuzzification* can be achieved through a fuzzifier kernel as we saw in Section 2.3, and *defuzzification* can be achieved through special procedures that select a crisp value representative of the fuzzy output (see Section 6.3). Many controllers, however, use directly crisp inputs. Figure 6.4 shows that in addition to a set of *if/then* rules,[2] a fuzzy controller has an *input interface* and an *output interface* handling fuzzification and defuzzification as well as various signal manipulations such as *normalization*, *scaling*, *smoothing*, and *quantization*. Scaling maps the range of values of the controlled variables into predefined universes of discourse, and quantization procedures assist in the mapping when discrete membership functions are used (Larkin, 1985; Efstathiou, 1987; Yager and Filev, 1994).

Fuzzy controllers operate in discrete time intervals. The rules are evaluated at regular intervals in the same way as in conventional digital control, with several rules being executed together (in *parallel*) within the same time interval. This parallel feature makes it possible to develop highly dispersed fuzzy algorithms as we will see later on. We use the subscript k to indicate a specific moment in time—that is, when $t = t_k$. The choice of sampling interval depends on the process being controlled and is usually selected so that at least several significant control actions are made during the process settling time (King and Mamdani, 1977).

Let us look at typical *input* or *left-hand side* (LHS) and *output* or *right-hand side* (RHS) fuzzy variables used in the knowledge base of fuzzy

[2]The set of *if/then* rules is also referred to as the controller's *knowledge base*.

controllers. Many fuzzy controllers use *error*, *change of error*, and *sum of errors* in the LHS,[3] based on measured process variables and setpoint values, and any process variable that can be manipulated directly in the RHS.

Input Variables

The most common LHS variable in fuzzy control is the *error*, or e. It is usually defined on the universe of discourse of crisp error e, which is the deviation of some measured variable y from a setpoint or reference r. At any time $t = k$ crisp error is defined as

$$e(k) \equiv r - y(k) \qquad (6.2\text{-}1)$$

The change in error, Δe or Δ*error*, between two successive time steps is also commonly used as an LHS variable. It is defined on the universe of discourse of crisp changes in error. At time $t = k$ the crisp change in error is the difference between present error and error in the previous time step $t = k - 1$, namely,

$$\Delta e(k) \equiv e(k) - e(k-1) \qquad (6.2\text{-}2)$$

Fuzzy variables can also be defined for the *rate of change in error* $\Delta^2 e(k) \equiv \Delta e(k) - \Delta e(k-1)$, and so on. The *sum of errors* $\bar{e}(k)$ may be used as an LHS fuzzy variable also. It takes into account the integrated effect of all past errors and is defined as

$$\bar{e}(k) \equiv \sum_{i=1}^{k} e_i \qquad (6.2\text{-}3)$$

In some cases, actual state variables may be used (instead of error, etc.) depending on the availability of parameter and structure estimation knowledge. It is even possible to use variables not directly measurable, such as *performance* or *reliability*, provided that they can be estimated in a timely and reliable manner (Tsoukalas, 1991).

Output Variables

RHS variables may be any directly manipulated variable. An RHS fuzzy variable u can be defined on the universe of discourse of a crisp manipulated variable. Actually the change in output Δu is more often used as the RHS variable. Δu indicates the extent of change of the control variable u at time $t = k$—that is, the *change in action*. Hence, if the defuzzified output at

[3]Using these variables, one can write *if/then* rules emulating PID modes of control.

time k is $\Delta u^*(k)$, the overall crisp output of the controller will be

$$u(k) = u(k-1) + \Delta u^*(k) \tag{6.2-4}$$

Using Δu is preferable, since it requires a smaller number of data points in the output universe of discourse in order for the controller to operate with reasonable accuracy.

if / then Rules and Inference

Often, but not always, LHS and RHS variables are scaled to the same universe of discourse and possess fuzzy values that have the same form. Scaling to a common universe of discourse with a common set of values for all variables may offer considerable savings in memory and speed as far as the computer implementation of a fuzzy algorithm is concerned. In addition, it may be helpful in analyzing the behavior of the controller itself, as we will see later in this chapter. With the advent of fuzzy microprocessors and fuzzy development shells, it is no longer necessary for a user to do scaling because it is done by the system automatically. Nonetheless, scaling helps to simplify algorithmic development and investigate factors involved in synthesis and analysis. As an example, consider the fuzzy values for the variables *error*, Δ*error*, and Δu shown in Figure 6.5 in connection with a fuzzy controller that emulates the derivative mode of a conventional controller (Sugeno, 1985;

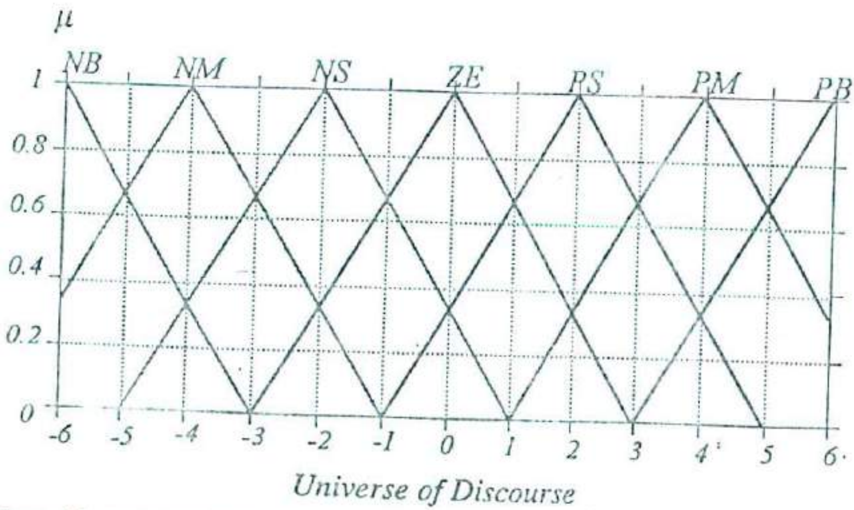


Figure 6.5 Common fuzzy values for the *error*, Δ *error*, and Δu variables, scaled to the same universe of discourse.

Mizumoto, 1988). The common fuzzy values are as follows:

$$
\begin{aligned}
&NB \equiv \textit{negative big}, && PS \equiv \textit{positive small}\\
&NM \equiv \textit{negative medium}, && PM \equiv \textit{positive medium}\\
&NS \equiv \textit{negative small}, && PB \equiv \textit{positive big}\\
&ZE \equiv \textit{zero} &&
\end{aligned}
$$

All variables share the same universe of discourse ranging between -6 and $+6$ as shown in Figure 6.5. In computer implementations, fuzzy values are usually quantized and stored in memory in the form of a look-up table as shown in Table 6.1. In this case the fuzzy values are stored in a 7×13 table, with every row in the table representing a quantized fuzzy value. The fuzzy algorithm of a controller that emulates a derivative mode is comprised of the following *if/then* rules:

$$
\begin{aligned}
&R_1: && \textit{if error is NB AND } \Delta \textit{error is ZE then } \Delta u \textit{ is PB} && ELSE\\
&R_2: && \textit{if error is NM AND } \Delta \textit{error is ZE then } \Delta u \textit{ is PM} && ELSE\\
&R_3: && \textit{if error is NS AND } \Delta \textit{error is ZE then } \Delta u \textit{ is PS} && ELSE\\
&R_4: && \textit{if error is ZE AND } \Delta \textit{error is ZE then } \Delta u \textit{ is ZE} && ELSE\\
&R_5: && \textit{if error is PS AND } \Delta \textit{error is ZE then } \Delta u \textit{ is NS} && ELSE\\
&R_6: && \textit{if error is PM AND } \Delta \textit{error is ZE then } \Delta u \textit{ is NM} && ELSE\\
&R_7: && \textit{if error is PB AND } \Delta \textit{error is ZE then } \Delta u \textit{ is NB} && ELSE \qquad (6.2\text{-}5)\\
&R_8: && \textit{if error is ZE AND } \Delta \textit{error is NB then } \Delta u \textit{ is PB} && ELSE\\
&R_9: && \textit{if error is ZE AND } \Delta \textit{error is NM then } \Delta u \textit{ is PM} && ELSE\\
&R_{10}: && \textit{if error is ZE AND } \Delta \textit{error is NS then } \Delta u \textit{ is PS} && ELSE\\
&R_{11}: && \textit{if error is ZE AND } \Delta \textit{error is PS then } \Delta u \textit{ is NS} && ELSE\\
&R_{12}: && \textit{if error is ZE AND } \Delta \textit{error is PM then } \Delta u \textit{ is NM} && ELSE\\
&R_{13}: && \textit{if error is ZE AND } \Delta \textit{error is PB then } \Delta u \textit{ is NB} &&
\end{aligned}
$$

When two LHS and one RHS variables are used as in (6.2-5), the algorithm can be visualized in the form of a table as shown in Table 6.2. Such an arrangement is sometimes called a "fuzzy associative memory (FAM) matrix." Blank items in the table indicate that there is no rule present for the particular combination of LHS variables. Obviously for algorithms with more than two LHS variables a tabular representation requires additional dimensions.

Table 6.1 Table of fuzzy values

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
NB	1	0.67	0.33	0	0	0	0	0	0	0	0	0	0
NM	0.33	0.67	1	0.67	0.33	0	0	0	0	0	0	0	0
NS	0	0	0.33	0.67	1	0.67	0.33	0	0	0	0	0	0
ZE	0	0	0	0	0.33	0.67	1	0.67	0.33	0	0	0	0
PS	0	0	0	0	0	0	0.33	0.67	1	0.67	0.33	0	0
PM	0	0	0	0	0	0	0	0	0.33	0.67	1	0.67	0.33
PB	0	0	0	0	0	0	0	0	0	0	0.33	0.67	1

Table 6.2 A fuzzy algorithm in tabular form

Δerror / *error*	*NB*	*NM*	*NS*	*ZE*	*PS*	*PM*	*PB*
NB				*PB*			
NM				*PM*			
NS				*PS*			
ZE	*PB*	*PM*	*PS*	*ZE*	*NS*	*NM*	*NB*
PS				*NS*			
PM				*NM*			
PB				*NB*			

Fuzzy control algorithms are evaluated using *generalized modus ponens* (GMP). We recall from Chapter 5 that GMP is a data-driven inferencing procedure that analytically involves the composition of fuzzy relations, usually *max–min composition*. We also saw that max–min composition under a given implication operator affects the RHS in a specific manner—for example, by *clipping* (when *Mamdani min*, ϕ_c, is used) or *scaling* (when *Larsen product*, ϕ_p, is used). In general, GMP can be thought of as a transformation of the RHS by a degree commensurate with the degree of fulfillment (DOF) of the rule and in a manner dictated by the implication operator chosen (see Examples 5.1 and 5.2). In this chapter, instead of explicitly using composition operations, we will mostly focus on such transformations as is often done, for the sake of convenience, in many fuzzy control applications. As far as the entire algorithm is concerned, the connective *ELSE* is analytically modeled as either *OR* ($\vee$) or *AND* ($\wedge$), again depending on the implication operator used for the individual *if/then* rules. For example, when the Mamdani min implication is used, the connective *ELSE* is interpreted as *OR* (see Table 5.5).

Fuzzy controller inputs are usually crisp numbers. Fuzzy inputs may also be considered in the case of uncertain or noisy measurements and crisp numbers may be fuzzified (see Section 2.3). Consider the situation shown in Figure 6.6 involving rules R_3, R_4, and R_{11} of (6.2-5). When at time $t = k$ crisp error e' and crisp change in error $\Delta e'$ as shown in Figure 6.6 are given to these rules we say that the rules have "fired," provided that their DOF is not zero. For example, in rule R_3 the crisp error e' shown has a 0.8 degree of membership to *NS* while the crisp change in error $\Delta e'$ has a 0.6 degree of membership to *ZE*. Thus the degree of fulfillment of rule R_3 at this particular time is

$$\mathrm{DOF}_3 = \mu_{NS}(e') \wedge \mu_{ZE}(\Delta e') = 0.8 \wedge 0.6 = 0.6 \qquad (6.2\text{-}6)$$

Provided that we interpreted the LHS connective *AND* as min ($\wedge$) [a common alternative is *product* ($\cdot$)], the RHS value *PS* will be transformed in accordance with DOF_3 in equation (6.2-6). The nature of the transformation depends on the implication used as we saw in Chapter 5. When Mamdani min is used the transformation amounts to clipping *PS* at the height of DOF_3 as shown in Figure 6.6. Thus R_3 contributes $\mu_{PS'}(\Delta u)$, the shaded part of the RHS value, to the total fuzzy output. Similarly rules R_4 and R_{11} have degrees of fulfillment

$$\mathrm{DOF}_4 = \mu_{ZE}(e') \wedge \mu_{ZE}(\Delta e') = 0.4 \wedge 0.6 = 0.4 \qquad (6.2\text{-}7)$$

$$\mathrm{DOF}_{11} = \mu_{ZE}(e') \wedge \mu_{PS}(\Delta e') = 0.4 \wedge 1.0 = 0.4 \qquad (6.2\text{-}8)$$

and they contribute $\mu_{ZE'}(\Delta u)$ and $\mu_{NS'}(\Delta u)$, shown as shaded parts of the RHS values in Figure 6.6. The rest of the rules of algorithm (6.2-5) do not fire, that is, they contribute a zero output. The total fuzzy output is the *union* of the three outputs since we interpret the connective *ELSE* in (6.2-5) as *OR* ($\vee$)—that is,

$$\mu_{OUT}(\Delta u) = \mu_{PS'}(\Delta u) \vee \mu_{ZE'}(\Delta u) \vee \mu_{NS'}(\Delta u) \qquad (6.2\text{-}9)$$

$\mu_{OUT}(\Delta u)$ is shown at the lower part of Figure 6.6. At this point we need to defuzzify $\mu_{OUT}(\Delta u)$ and obtain a crisp value Δu_k^* representative of $\mu_{OUT}(\Delta u)$ to be used as input to the process. In the next section we will look at different methods for defuzzification.

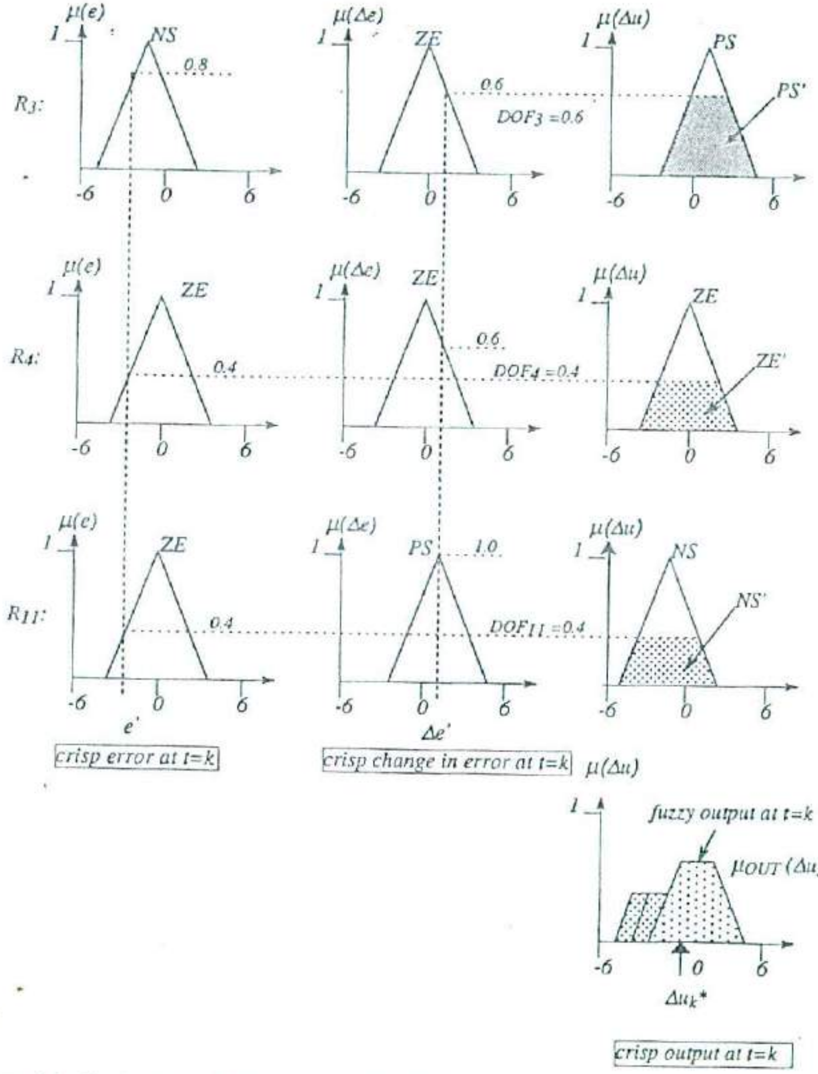


Figure 6.6 Evaluation of three control rules at time $t = k$ using Mamdani min implication and min interpretation of *AND* (DOF).

If *Larsen product* is used as the fuzzy implication operator for the individual rules of (6.2-5), the membership function of the RHS value is *scaled* by the degree of fulfillment of each rule as shown in Figure 6.7 (see Example 5.2). Since the connective *ELSE* is interpreted as *OR* ($\vee$) when Larsen product implication is used (see Table 5.5), the total output $\mu_{OUT}(\Delta u)$ is also the *union* of the three individual outputs. Out of that we need to select a representative crisp value as input to the process. We note in Figure 6.7 that $\mu_{OUT}(\Delta u)$ looks quite different from the total fuzzy output obtained using Mamdani min shown in Figure 6.6. Other fuzzy implication operators (see Table 5.2) would produce different transformations in the shape of the RHS fuzzy value and, hence, a different $\mu_{OUT}(\Delta u)$.

Interpretations of *AND* other than min ($\wedge$) may be used in the *AND* connective found in the LHS of the rules, hence obtaining different degrees of fulfillment. Arithmetic product has been used (particularly in conjunction with max-product implication) and more generally *T-norms* (Zimmermann, 1985; Fuller and Zimmermann, 1992).[4] Using arithmetic product the degree of fulfillment for the rules of (6.2-5) that fire would be evaluated in the manner shown in Figure 6.8. The degree of fulfillment for R_3, R_4, and R_{11} are

$$\mathrm{DOF}_3 = \mu_{NS}(e') \cdot \mu_{ZE}(\Delta e') = 0.8 \cdot 0.6 = 0.48$$

$$\mathrm{DOF}_4 = \mu_{ZE}(e') \cdot \mu_{ZE}(\Delta e') = 0.4 \cdot 0.6 = 0.24$$

$$\mathrm{DOF}_{11} = \mu_{ZE}(e') \cdot \mu_{PS}(\Delta e') = 0.4 \cdot 1.0 = 0.4 \qquad (6.2\text{-}10)$$

Comparing equations (6.2-10) with equations (6.2-6)–(6.2-8) we see that generally the two different interpretations of *AND* lead to different results under the same fuzzy implication operators as we can also see by comparing Figures 6.7 and 6.8.

After we defuzzify $\mu_{OUT}(\Delta u)$ by one of the methods discussed in the next section, we obtain a crisp value Δu_k^*, which in the case of the algorithm of (6.2-5) would be an integer between -6 and $+6$. Values greater than the extremes of the universe of discourse are set to the extreme values, in this case -6 or $+6$. This value is then multiplied by a scaling factor that maps it into the appropriate range of the manipulated variable before using it to actuate a device (Larkin, 1985).

Since so much of actual process control knowledge has historically been obtained through PID controllers, it is often convenient to emulate various modes and combinations of the PID controller by fuzzy rules. Thus a fuzzy controller emulating a conventional PD mode of control controller would

[4]See the Appendix for an introduction to *T norms* and their co-norms, called *S norms*.

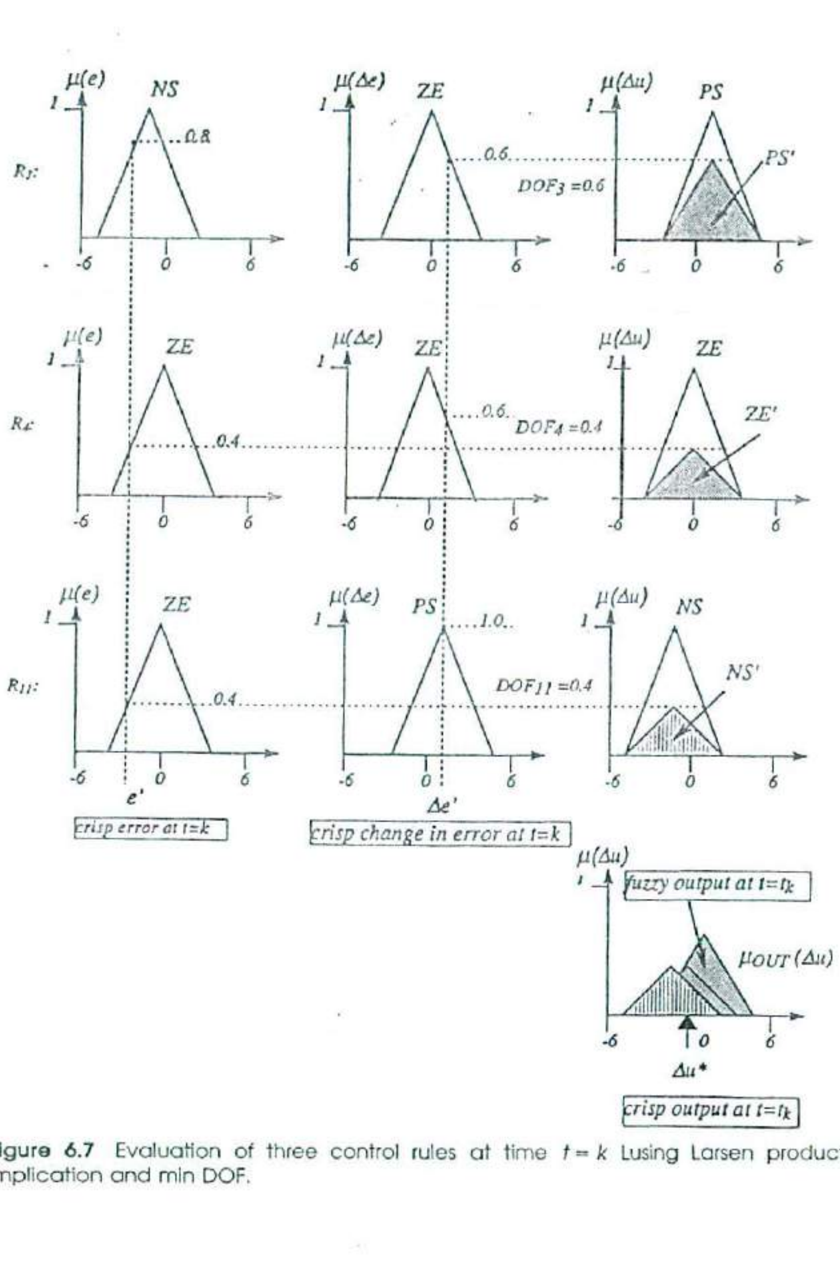


Figure 6.7 Evaluation of three control rules at time $t = k$ Lusing Larsen product implication and min DOF.

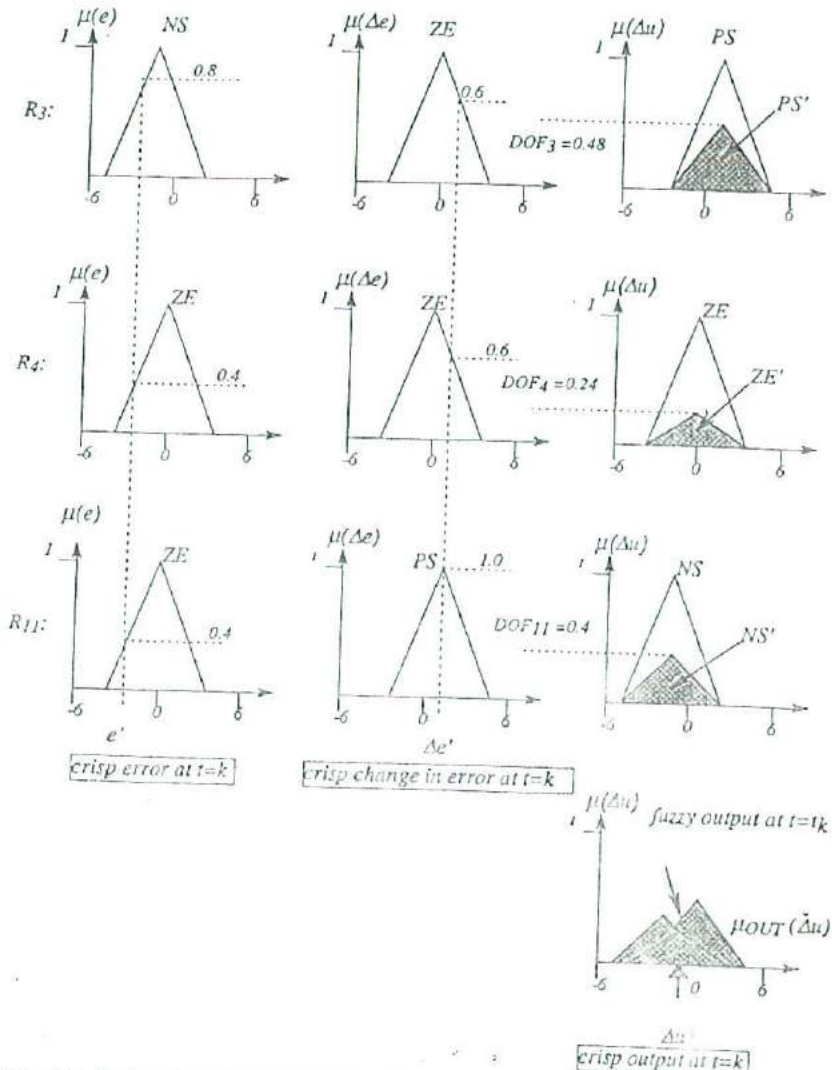


Figure 6.8 Evaluation of three control rules at time $t = k$ using Larsen product implication and product DOF.

consist of rules having the form

$$\textit{if} \quad e \textit{ is } A \quad AND \quad \Delta e \textit{ is } B \quad \textit{then } u \textit{ is } C \qquad (6.2\text{-}11)$$

where e is the error and Δe is the change in error. A PI-like fuzzy controller would have rules of the form

$$\textit{if} \quad e \textit{ is } A \quad AND \quad \Delta e \textit{ is } B \quad \textit{then } \Delta u \textit{ is } C \qquad (6.2\text{-}12)$$

while a P-like controller would have rules

$$\textit{if} \quad e \textit{ is } A \quad \textit{then} \quad u \textit{ is } C \qquad (6.2\text{-}13)$$

The rule form of a PID-like fuzzy controller is

$$\textit{if} \quad e \textit{ is } A \quad AND \quad \Delta e \textit{ is } B \quad AND \quad \bar{e} \textit{ is } C \quad \textit{then} \quad u \textit{ is } D \qquad (6.2\text{-}14)$$

where $\bar{e}$ is the sum of errors.

Although we have formulated fuzzy algorithms in terms of rules involving fuzzy values on their RHS (such rules are referred to as *Mamdani rules*), there are advantages to consider crisp or special shape membership functions as well. Several fuzzy controllers use rules where the output variable is given in terms of a functional relation of the inputs. This is known as the *Sugeno* or *TSK*[5] *form* of fuzzy rules. Such rules are typically written as

$$\textit{if} \quad x_1 \textit{ is } A_1 \quad AND \quad x_2 \textit{ is } A_2 \ldots \textit{then} \quad u = f(x_1, \ldots, x_n) \qquad (6.2\text{-}15)$$

where f is a function of the inputs $x_1, \ldots, x_n$. When $f(x_1, \ldots, x_n)$ is a constant, rules of the form (6.1-15) constitute a *zero-order Sugeno controller*. When $f(x_1, \ldots, x_n)$ is a first-order polynomial we have what is called a *first-order Sugeno controller*. For example, we may describe a PI controller of (6.2-12) by rules of the form

$$\textit{if} \quad e \textit{ is } LARGE \quad AND \quad \Delta e \textit{ is } MEDIUM \quad \textit{then } u = 2e + 3\Delta e \qquad (6.2\text{-}16)$$

An interesting application of Sugeno rules is when a PID controller is put directly in the RHS of (6.2-15). The result is a fuzzy "supervisor" changing the parameters of a PID controller [see Tzafestas and Papanikolopoulos (1990)]. Sugeno fuzzy models are well suited for modeling nonlinear systems by interpolating multiple linear models and are also well suited to mathematical analysis and lend themselves to adaptive techniques, whereas Mamdani rules are more intuitive and better suited to human. [See Jang and Sun (1995) and Jang and Gulley (1995) for a review of different controllers.]

[5]After the *Takagi*, *Sugeno*, and *Kang* who first proposed it in 1985; also referred to as the *Sugeno fuzzy model*.

An alternative to either Sugeno or Mamdani rules is to consider rules whose consequents employ monotonical membership functions. This form of rules is known as the *Tsukamoto fuzzy model* (Tsukamoto, 1979). In Tsukamoto rules the inferred output of each rule is a crisp value equal to the rule's *degree of fulfillment*, with the overall output being taken as the weighted average of all outputs (a crisp value).

Fuzzy algorithms such as (6.2-5) are inherently parallel in the sense that individual *if/then* rules are fired independent of each other, with a specific input being processed by several rules each contributing to a collective result, namely, $\mu_{OUT}(\Delta u)$. Actual process systems, however, may have many inputs and outputs, and hence they are referred to as *many-input-many-output* (MIMO) systems. The question then arises of how relevant are the rather simple *if/then* rules we have seen thus far, such as the algorithm of (6.2-5), to the control of such systems and what happens to parallelism at a higher level of system complexity.

Generally the control strategies of complicated process systems may be organized in such a manner that relatively simple *if/then* rules are used (Terano, 1992; Yager, 1994). This is achieved by partitioning the knowledge base of the controller into rule clusters. In each cluster there are *if/then* rules that may have several LHS variables but only one RHS variable. Suppose that we have p input variables $x_1, x_2, \ldots, x_p$ and r manipulated variables $u_1, u_2, \ldots, u_r$. The algorithmic development generally proceeds from some general and maybe complicated *if/then* rules that form the *a priori* knowledge prescribing what has to be done under a set of hypothetical situations. Often, but not always, it is possible to reduce these initial rules to simpler rules with one control variable in the RHS. Rules that have the same RHS variable are collected together to form a rule cluster. In the end we have one cluster of rules whose RHS is used to manipulate variable u_1, another for variable u_2, and so on. Thus, a complicated process control system may be decomposed into a number of *many-input-single-output* controllers. Such rule clusters may be executed independently, hence maintaining the overall parallel characteristics of fuzzy systems. Of course, more elaborate architectures can be devised that may include *metarules*. The developers of fuzzy control algorithms exercise considerable creativity in setting up special variables and rules for the interaction of these clusters. In principle, however, rule clusters can be noninteractive, in which case they can be executed in parallel, achieving considerable speed and computational efficiency.

6.3 DEFUZZIFICATION METHODS

After the input to the controller has been processed by the control algorithm the result is a fuzzy output $\mu_{OUT}(u)$. Selecting a crisp number u^* representative of $\mu_{OUT}(u)$ is a process known as *defuzzification*. Over the years several defuzzification techniques have been suggested (Terano et al., 1992; Pedrycz, 1993; Yager and Filev, 1994). The choice of defuzzification method may have a significant impact on the speed and accuracy of a fuzzy controller.[6] The most frequently used ones are the *centroid* or *center of area* (COA), the *center of sums* (COS), and *mean of maxima* (MOM).

Center of Area (COA) Defuzzification

In COA defuzzification[7] the crisp value u^* is taken to be the geometrical center of the output fuzzy value $\mu_{OUT}(u)$, where $\mu_{OUT}(u)$ is formed by taking the union of all the contributions of rules whose DOF > 0.[8] The center is the point which splits the area under the $\mu_{OUT}(u)$ curve in two equal parts. Let us assume we have a discretized universe of discourse. The defuzzified output is defined as

$$u^* = \frac{\sum_{i=1}^{N} u_i \, \mu_{OUT}(u_i)}{\sum_{i=1}^{N} \mu_{OUT}(u_i)} \tag{6.3-1}$$

where the summation (integration) is carried over (discrete) values of the universe of discourse u_i sampled at N points. COA is a well known and often used defuzzification method. Some potential drawbacks of COA are that it favors "central" values in the universe of discourse and that, due to its complexity, it may lead to rather slow inference cycles. COA defuzzification takes into account the area of the resultant membership function $\mu_{OUT}(u)$ as a whole. If the areas of two or more contributing rules overlap, equation (6.3-1) does not take into account the overlapping area only once [since we take the union to form $\mu_{OUT}(u)$, the resultant membership function].

When $\mu_{OUT}(u) = 0$ we simply set the crisp output to a pre-agreed value (in order to avoid dividing by zero), typically $u^* = 0$. The crisp output value may also be computed in terms of the DOF of each contributing rule as

$$u^* = \frac{\sum_{k=1}^{n} \mathrm{DOF}_k \cdot M_k}{\sum_{k=1}^{n} \mathrm{DOF}_k \cdot B'_k} \tag{6.3-2}$$

where B'_k is the contribution due to the firing of rule k,[9] M_k is the moment of B'_k, and DOF_k is the *degree of fulfillment* of the kth rule ($k = 1, \ldots, n$).

[6] Certain defuzzification methods may introduce nonlinearities and discontinuities in the control hypersurface [see Jager (1995)].

[7] Also known as *center of gravity* defuzzification, a name more appropriate for multidimensional fuzzy output.

[8] For convenience we use the control signal u as the output variable. The control signal change Δu or any other output variable may be used as well.

[9] The subscript k is used to indicate the kth rule and should not be confused with the letter k used earlier to indicate the $t = k$ time step.

We recall that the moment of B'_k is the product of B'_k and the distance of its center of gravity from the μ axis (the moment about zero).

Center of Sums (COS) Defuzzification

To address the problems associated with COA and take into account the overlapping areas of multiple rules more than once, a variant of COA called *center of sums* (COS) is used. As shown in Figure 6.9, COS builds the resultant membership function by taking the sum (not just the union) of output from each contributing rule. Hence overlapping areas are counted more than once. COS is actually the most commonly used defuzzification method. It can be implemented easily and leads to rather fast inference cycles. It is given by

$$u^* = \frac{\sum_{i=1}^{N} u_i \cdot \sum_{k=1}^{n} \mu_{B'_k}(u_i)}{\sum_{i=1}^{N} \sum_{k=1}^{n} \mu_{B'_k}(u_i)} \tag{6.3-3}$$

where $\mu_{B'_k}(u_i)$ is the membership function (at point u_i of the universe of discourse) resulting from the firing of the kth rule.

Mean of Maxima (MOM) Defuzzification

One simple way to defuzzify the output is to take the crisp value with the highest degree of membership in $\mu_{OUT}(u)$. Oftentimes, however, there may be more than one element in the universe of discourse having the maximum value, as may be seen in the $\mu_{OUT}(u)$ of Figure 6.6. In such cases we can randomly select one of them or, even better, take the mean value of the maxima. Suppose that we have M such maxima in a discrete universe of discourse. The crisp output can be obtained by

$$u^* = \sum_{m=1}^{M} \frac{u_m}{M} \tag{6.3-4}$$

where u_m is the mth element in the universe of discourse where the membership function of $\mu_{OUT}(u)$ is at the maximum value, and M is the total number of such elements.

MOM defuzzification is faster than COA, and furthermore it allows the controller to reach values near the edges of the universe of discourse. A disadvantage of this method, however, is that it does not consider the overall shape of the fuzzy output $\mu_{OUT}(u)$. On the other hand, with COA the extreme values of the universe of discourse cannot be reached—for example, near ± 6 in Figure 6.5. Both methods have been used in control applications; several variants of them exist, such as the *indexed center of gravity method*, where a threshold level is used to eliminate elements with degrees of membership lower than a threshold in the computation of $\mu_{OUT}(\Delta u)$

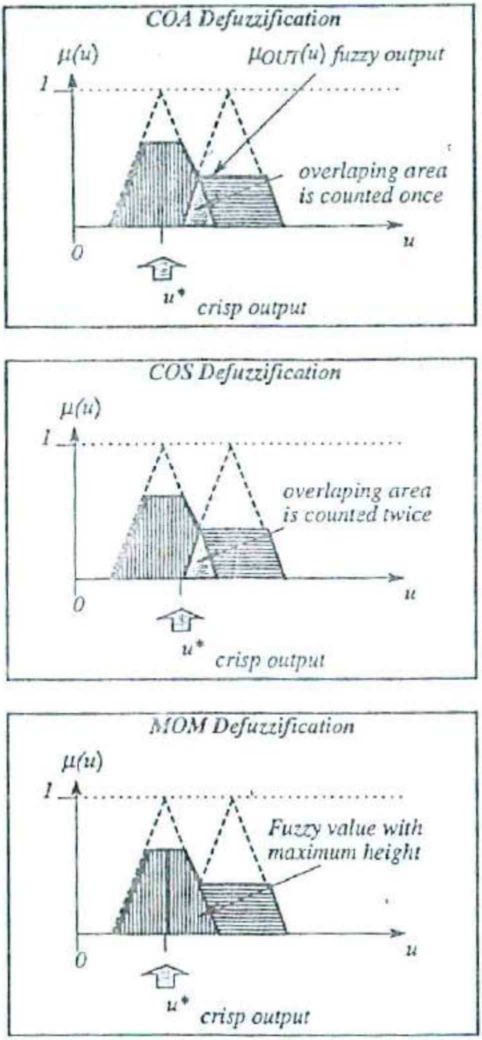
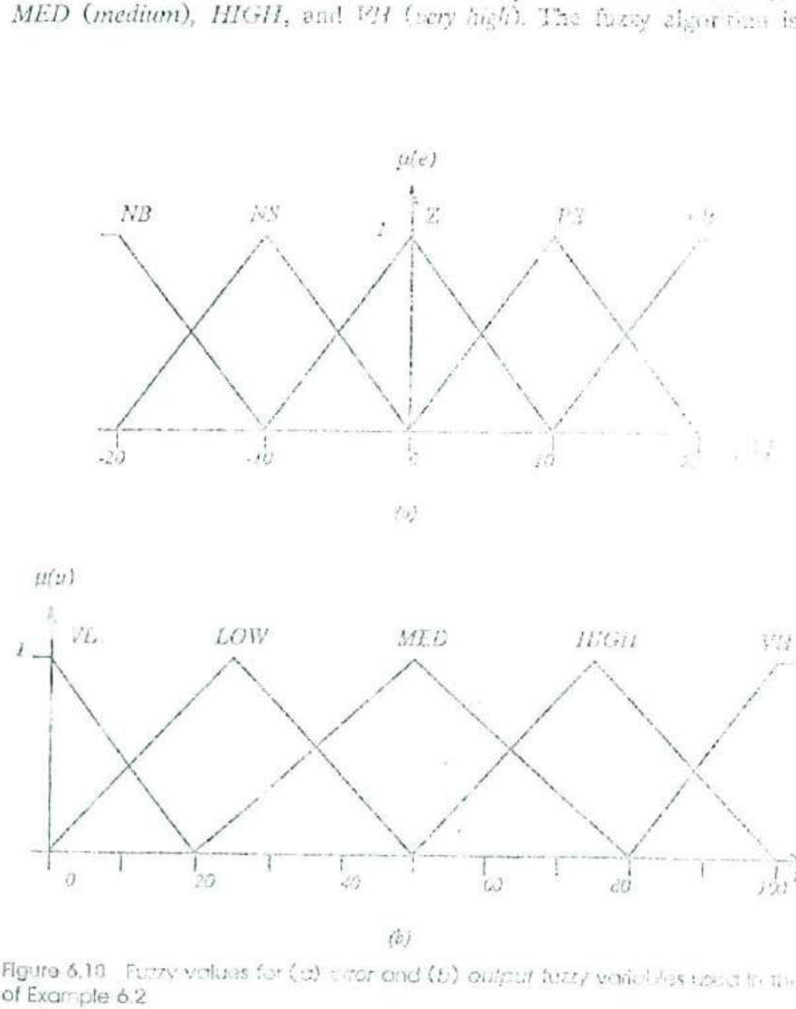


Figure 6.9 Three different defuzzification methods: center of area (COA, center of sums (COS), and mean of maxima (MOM).

(Pedrycz, 1993). It is also possible to employ defuzzification methods in an adaptive manner (Yager and Filev, 1993, 1994).

Example 6.2 A Simple Fuzzy Controller for Level Control. Consider the process system shown in Figure 6.1 (and controlled by a PID controller in

Example 6.1). We want to develop a fuzzy controller to control the flow of liquid in the tank and maintain its level at a reference value. The simplest possible fuzzy controller would have only one LHS variable and one RHS variable. We define fuzzy variables *error* and *output* for the LHS and RHS, respectively, as shown in Figure 6.10. It should be noted that the simplicity of the problem does not call for scaling our variables to a common universe of discourse. The universe of discourse for *error* is made of crisp percent error values from −20% to 20%. The fuzzy values of the variable *error* are: *NB* (*negative big*), *NS* (*negative small*), *Z* (*zero*), *PS* (*positive small*), and *PB* (*positive big*). The universe of discourse for *output* is the crisp values of output ranging from 0% to 100%, with fuzzy values: *VL* (*very low*), *LOW*, *MED* (*medium*), *HIGH*, and *VH* (*very high*). The fuzzy algorithm is com-



(a)

(b)

Figure 6.10 Fuzzy values for (a) error and (b) output fuzzy variables used in the rules of Example 6.2

prised of the following rules:

$$
\begin{array}{llll}
R_1: & \textit{if error is NB} & \textit{then output is VH} & ELSE \\
R_2: & \textit{if error is NS} & \textit{then output is HIGH} & ELSE \\
R_3: & \textit{if error is Z} & \textit{then output is MED} & ELSE \\
R_4: & \textit{if error is PS} & \textit{then output is LOW} & ELSE \\
R_5: & \textit{if error is PB} & \textit{then output is VL} &
\end{array}
\qquad \text{(E6.2-1)}
$$

We use Mamdani min for fuzzy implication and hence interpret the connective *ELSE* in (E6.2-1) as *OR* (see Table 5.5). Suppose that at time $t = 0$ min the output of our controller was at 50% of its full range and 2 min later we introduce the error shown in Figure 6.2. What would be the output according to the algorithm of (E6.2-1)? Let us look at what happens at $t = 3$ min. From Figure 6.2 we see that the input to the algorithm at this time is a crisp error $e_k = 1.25\%$, which belongs to the Z value of *error* to a degree of 0.87 and to PS to a degree of 0.12. The degree of membership to other fuzzy values is zero, as can be seen in Figure 6.10. Thus, rules R_3 and R_4 of algorithm (E6.2-1) will fire, since they are the only rules involving the Z and PS values. The situation is shown in Figure 6.11, where we see a schematic (geometrical) rendition of the evaluation of the control algorithm under GMP at time $t =$ 3 min. The *degree of fulfillment* of R_3 is $\mathrm{DOF}_3 = 0.87$ and for R_4 we have $\mathrm{DOF}_4 = 0.12$. All other rules have DOF = 0.0. Using Mamdani min implication the result of evaluating rules R_3 and R_4 under GMP is to *clip* the RHS values of rules R_3 and R_4—that is, *MED* and *LOW*—at $\mu_{LOW'}(u) = 0.87$ and $\mu_{MED}(u) = 0.12$, respectively. In other words, *MED* is clipped at 0.12. Out of all rules, only R_3 and R_4 contribute at $t = 3$ min, and their contributions are $\mu_{LOW'}(u)$ and $\mu_{MED'}(u)$ as shown in Figure 6.11. Since we interpret the connective *ELSE* as *OR*, the total fuzzy output of the entire algorithm at $t = 3$ min is the union of these two values—that is,

$$\mu_{OUT}(u) = \mu_{LOW'}(u) \vee \mu_{MED'}(u) \qquad \text{(E6.2-2)}$$

$\mu_{OUT}(u)$ is shown in the lower part of Figure 6.11. We use the COA defuzzification—that is, equation (6.3-2)—to defuzzify $\mu_{OUT}(u)$. The result is $u_3^* = 47\%$. If MOM is used, the average value of the maximum values of $\mu_{OUT}(u)$ is at the middle of the plateau of $\mu'_{MED}(u)$—that is, at about 50%. Hence, the two methods give somewhat different results. In MOM the contribution of $\mu'_{LOW}(u)$ is totally ignored since only the values where $\mu_{OUT}(u)$ is at a maximum are taken into account. The above procedure is repeated in subsequent time steps. The defuzzified output of the controller using COA is shown in Figure 6.12. Comparing with the three different modes of PID control shown in Figure 6.3, we see that our fuzzy controller

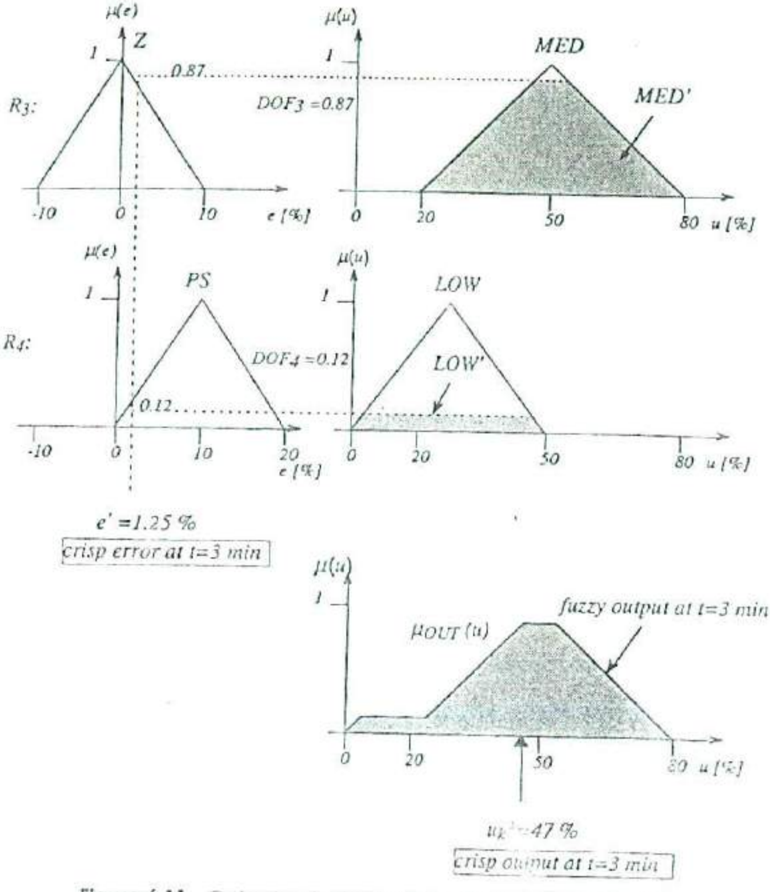


Figure 6.11 Output calculation at $t = 3$ min in Example 6.2.

behaves like a proportional controller, since the form (not the magnitude) of the output is similar to $u_p(t)$. □

Example 6.3 A Two-Input Fuzzy Controller for Level Control. Consider the process system shown in Figure 6.1, addressed by a PID controller in Example 6.1 and a simple fuzzy controller in Example 6.2. Let us now develop a fuzzy controller with two LHS and one RHS variables. We use *error* and change in error, Δ*error*, in the LHS of the rules and use *output* in the RHS. The fuzzy values of these variables are shown in Figure 6.13. Δ*error* e[illegible] the direction of change in error; that is, *increasing* is described by the f[illegible]e *P* (*positive*), *decreasing* is described by *N* (*negative*), and *no* *change* is described by *ZE* (*zero*). We use Mamdani min implication and COA defuzzification. The fuzzy algorithm is:

R_1: *if error is NB AND* Δ*error is N then output is HIGH ELSE*

R_2: *if error is NB AND* Δ*error is ZE then output is VH ELSE*

R_3: *if error is NB AND* Δ*error is P then output is VH ELSE*

R_4: *if error is NS AND* Δ*error is N then output is HIGH ELSE*

R_5: *if error is NS AND* Δ*error is ZE then output is HIGH ELSE*

R_6: *if error is NS AND* Δ*error is P then output is MED ELSE*

R_7: *if error is Z AND* Δ*error is N then output is MED ELSE*

R_8: *if error is Z AND* Δ*error is ZE then output is MED ELSE*

R_9: *if error is Z AND* Δ*error is P then output is MED ELSE*

R_{10}: *if error is PS AND* Δ*error is N then output is MED ELSE*

R_{11}: *if error is PS AND* Δ*error is ZE then output is LOW ELSE*

R_{12}: *if error is PS AND* Δ*error is P then output is LOW ELSE*

R_{13}: *if error is PB AND* Δ*error is N then output is LOW ELSE*

R_{14}: *if error is PB AND* Δ*error is ZE then output is VL ELSE*

R_{15}: *if error is PB AND* Δ*error is P then output is VL*

(E6.3-1)

We recall that with Mamdani min the connective *ELSE* in (E6.3-1) is interpreted as *OR* and therefore the total fuzzy output will be the *union* of individual rule contributions (see Table 5.5). It is customary in the control literature to refer to the fuzzy relations (E6.3-1) as *control surfaces* (or *hypersurfaces*). Figure 6.14 is a graphical representation of the control surface indicating hypersurface dependence on the rules. In Figure 6.14*a*, no rules exist in our algorithm; hence the control hypersurface is a flat plane at $u = 0$. If the control algorithm in (E6.3-1) was comprised only of the two rules R_1 and R_2 (the rest did not exist), then the control hypersurface would look like what is shown in Figure 6.14*b*. If only the first eight rules of the algorithm are present, the control hypersurface looks like Figure 6.14*c*, while if the first 13 rules are present, the control hypersurface would look like Figure 6.14*d*. Finally, if all 15 rules are present, the control hypersurface looks like

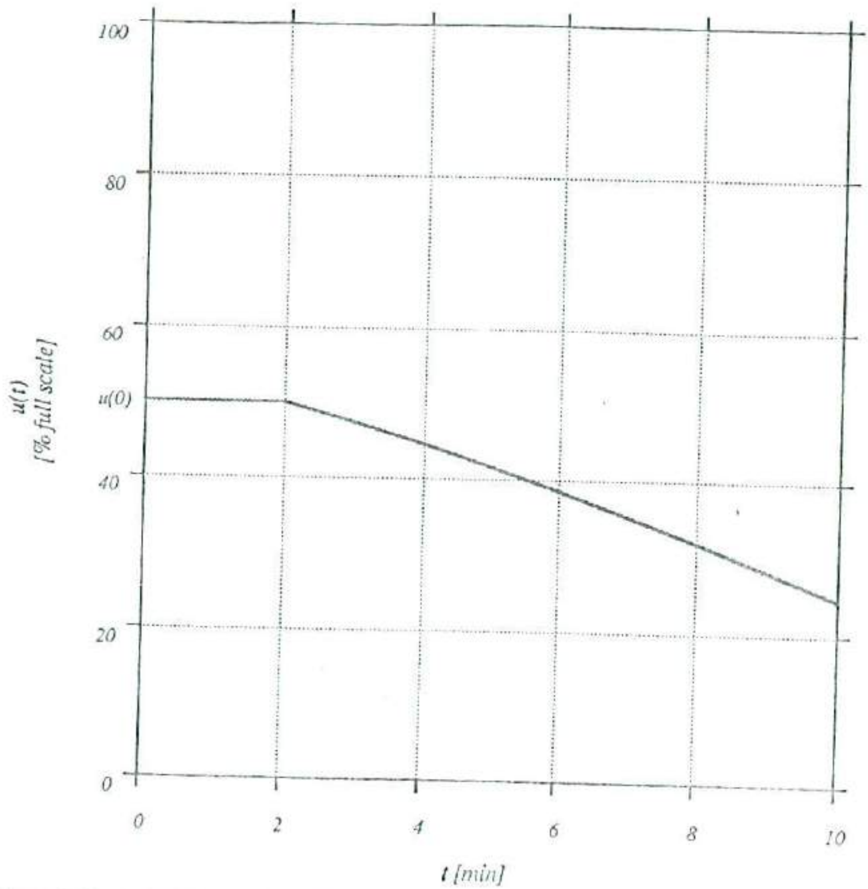
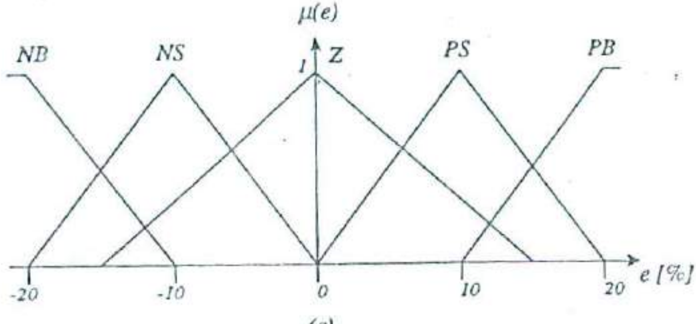


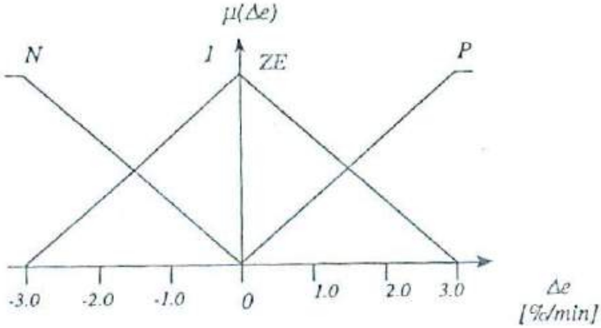
Figure 6.12 Defuzzified output of fuzzy level controller with one input and one output.

Figure 6.14*e*. Plotting the control hypersurface helps to visualize the manner in which a fuzzy controller covers the control space. Unfortunately it is not convenient to use when more than three variables are present.

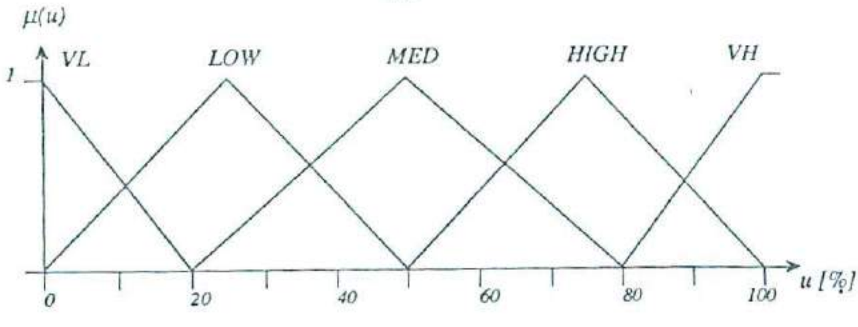
At $t = 2$ min we introduce the error shown in Figure 6.2 (same as in Examples 6.1 and 6.2). Figure 6.15 shows a schematic representation of the fuzzy inference or *generalized modus ponens* (GMP) at $t = 5$ min. Crisp inputs $e' = 3.75\%$ and $\Delta e' = 1.25\%$ are presented to the algorithm (E6.3-1) at this time. Crisp error $e' = 3.75\%$ belongs to fuzzy value Z to degree of 0.75 and to fuzzy value PS to a degree of 0.4. Similarly, crisp change-in-error $\Delta e' = 1.25\%$ belongs to fuzzy value ZE to a degree of 0.6 and to fuzzy value P to a degree of 0.4. Hence the only rules that will have DOF greater than



(a)



(b)



(c)

Figure 6.13 Fuzzy values for (*a*) *error*, (*b*) Δ*error*, and (*c*) *output* fuzzy variables used in Example 6.3.

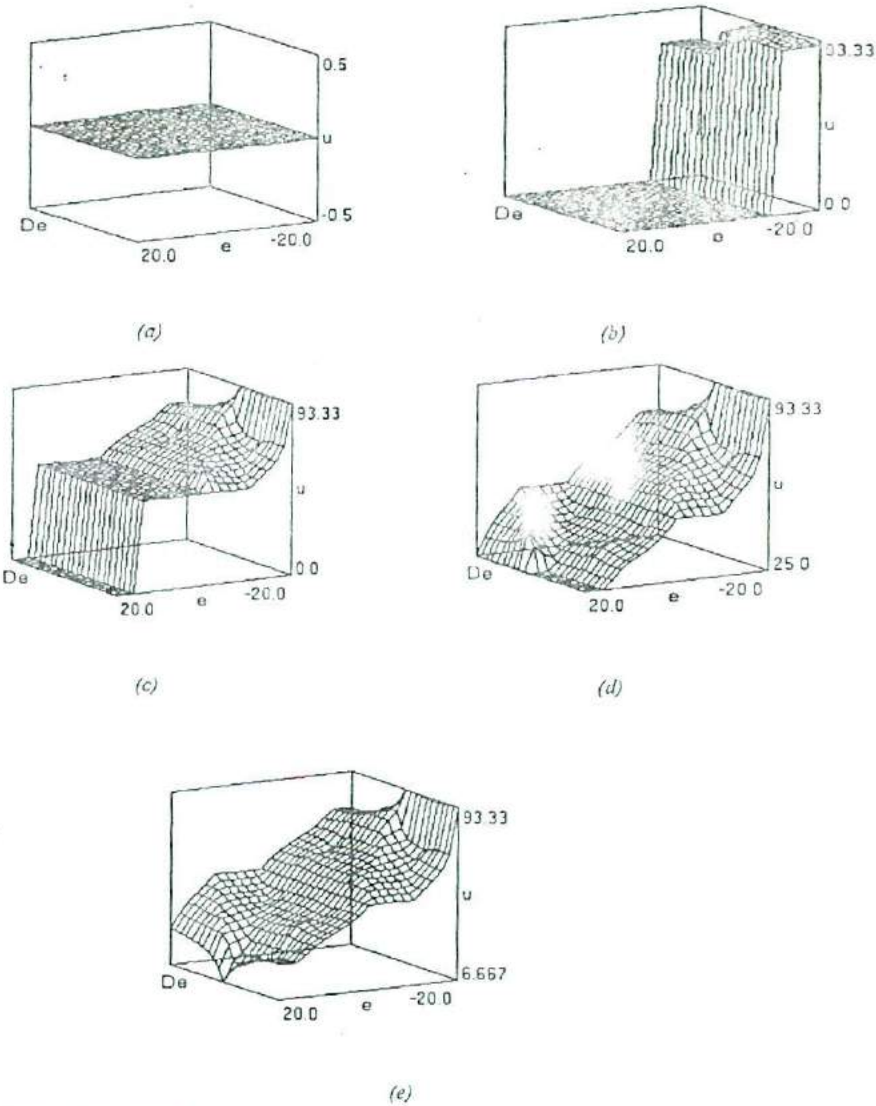


Figure 6.14 (*a*) The control hypersurface when there are no rules present, (*b*) with rules R_1 and R_2 only, (*c*) with rules R_1 through R_5, (*d*) with rules R_1 through R_{13}, and (*e*) with all 15 rules.

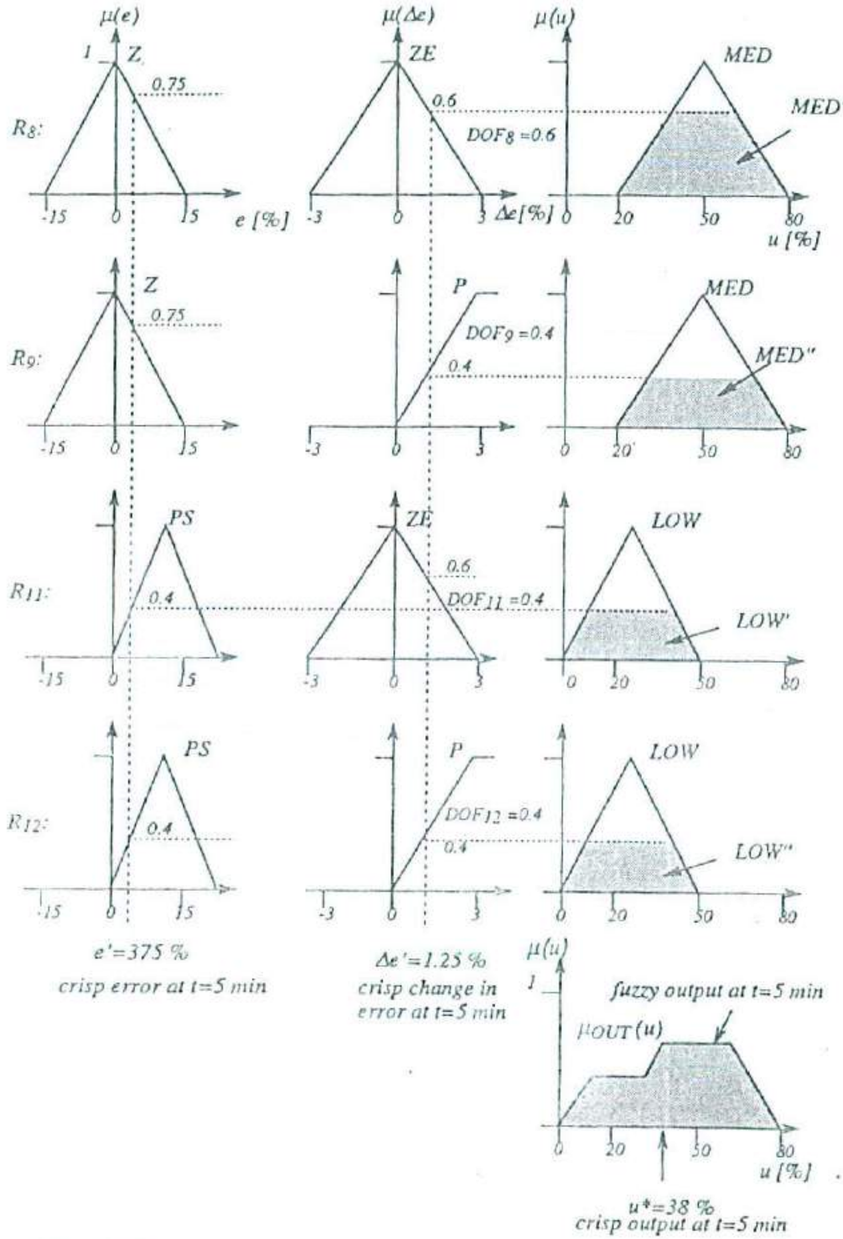


Figure 6.15 Evaluating the fuzzy algorithm of Example 6.3 at time $t = 5$ min.

zero in (E6.3-1)—that is, the rules that fire—will be R_8, R_9, R_{11}, and R_{12}. Using the min form of DOF (i.e., the min ($\wedge$) interpretation of *AND*), each rule contributes the shaded part of the RHS value shown in Figure 6.14. We recall that GMP with Mamdani min implication clips the RHS at the height of DOF, as shown in Figure 6.15. Rule R_8 contributes *MED'*, R_9 contributes *MED"*, R_{11} contributes *LOW'*, and R_{12} contributes *LOW"*. The fuzzy output $\mu_{out}(u)$ is the union (max) of these four contributions (shaded parts); that is,

$$\mu_{out}(u) = \mu_{MED'}(u) \vee \mu_{MED''}(u) \vee \mu_{LOW'}(u) \vee \mu_{LOW''}(u) \quad \text{(E6.3-2)}$$

$\mu_{out}(u)$ is shown at the lower part of Figure 6.15. Using COA defuzzification, we obtain the crisp output $u^* = 38\%$. The procedure is repeated for other time steps. The crisp output of the controller for the duration of the problem is shown in Figure 6.16. Comparing with Figure 6.13, we see that introducing

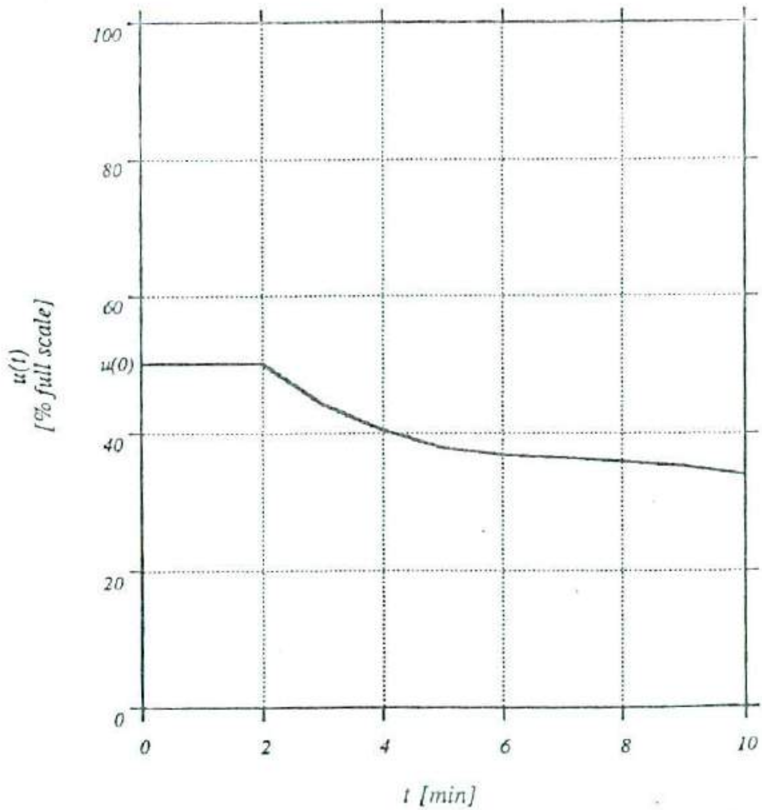


Figure 6.16 Defuzzified output of fuzzy controller with two inputs (*error* and Δ*error*) under ramp input.

one more variable, namely the change in error $\Delta error$, makes a significant difference in the control action. To obtain a desired response by our controller, we can now modify the shape and position of constituent membership functions or the rules, the implications used, and so on. A number of factors contribute to different outputs, such as the knowledge encoded in (E6.3-1), the fuzzy values used, the interpretation of *AND* (affecting DOF), the implication operator, and the defuzzification method used. The role and significance of these factors will be examined in the next section. □

6.4 ISSUES INVOLVED IN DESIGNING FUZZY CONTROLLERS

Although there are automatic ways of identifying the rules and membership functions involved in a fuzzy controller,[10] in many ways the development of a good fuzzy controller reflects the maturity of knowledge about a process. The choice of fuzzy variables and values and the rules themselves are intimately related to the knowledge a developer has about the entire process control system. The knowledge can be extracted by interviewing skilled operators or analyzing records of system responses to prototypes of input sequences (Dubois and Prade, 1980; Bernard, 1988). In addition, important decisions need to be made about the algorithm itself, such as what kind of implication to use, the appropriate defuzzification method, and implementation-related issues such as how to store the fuzzy relation of the algorithm, how to quantize membership functions, and so on. A difficult issue in fuzzy control arises in connection with determining the stability characteristics of the system. Stability itself can be thought of as a fuzzy variable and can be included in a description, with various degrees of stability (not just *stable* or *unstable*) being considered. Generally though, stability questions are hard to answer exclusively within fuzzy linguistic descriptions (Kiszka et al., 1985; Jianqin and Laijiu, 1993).

Once an algorithm has been developed, its quality can be assessed by examining the shape of the fuzzy output. Consider the situation shown in Figure 6.17 (King and Mamdani, 1977). Here we have three different general shapes for the membership function of the fuzzy output at some particular time step. They reveal three different instances of algorithmic quality. In situation A, a well-peaked fuzzy output indicates presence of strong firing rules. In situation B, the output points to two different and opposite areas of the universe of discourse, and hence we identify the presence of some contradictory rules or groups of rules, at the same time suggesting an output toward -3 and toward $+3$. An algorithm that points its output in opposite directions at the same time needs some further refinement to remove this kind of contradiction. In situation C we have the presence of an *unsatisfactory* set of rules since there is no representative output. In general, low

[10]Often referred to as *structure* and *parameter identification*.

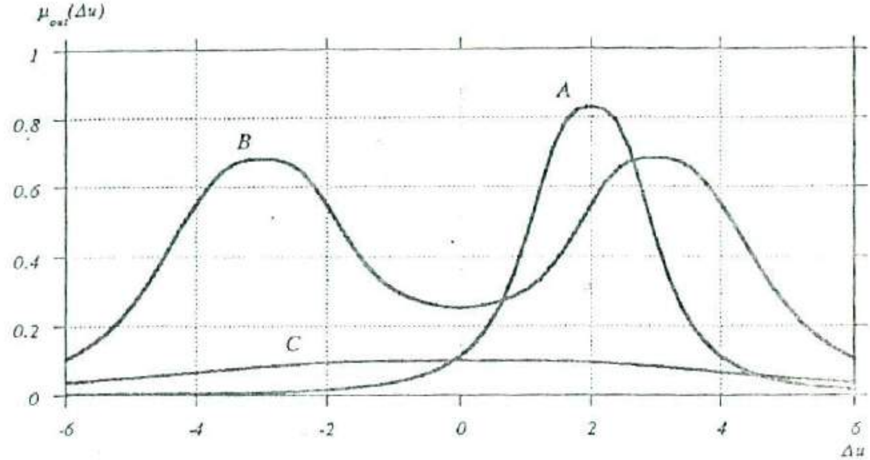


Figure 6.17 Three different cases of fuzzy output indicative of algorithmic quality: (*A*) *dominant* rule, (*B*) *contradictory* rules, and (*C*) *no satisfactory* rule.

plateaus like what is shown in situation C indicate that the knowledge encoded in the algorithm is incomplete and that additional rules are needed.

Let us now turn our attention to the various factors involved in the development of a fuzzy algorithm other than the quality of the encoded knowledge. Fuzzy algorithms are linguistic descriptions of the desirable behavior of a system. As such they have an analytical form involving fuzzy variables, relations, implication operators, and inferencing procedures. In order to examine the factors involved in algorithmic synthesis and analysis, let us look at the general analytical description of a fuzzy algorithm. Suppose that we have a control algorithm with linguistic form:

$$\begin{array}{lllllll} \textit{if} & x \text{ is } A_1 & AND & y \text{ is } B_1 & \textit{then } u \text{ is } C_1 & ELSE \\ \textit{if} & x \text{ is } A_2 & AND & y \text{ is } B_2 & \textit{then } u \text{ is } C_2 & ELSE \\ \dots \\ \textit{if} & x \text{ is } A_j & AND & y \text{ is } B_j & \textit{then } u \text{ is } C_j & ELSE \\ \dots \\ \textit{if} & x \text{ is } A_n & AND & y \text{ is } B_n & \textit{then } u \text{ is } C_n \end{array} \tag{6.4-1}$$

At time $t = k$, crisp inputs x' and y' are given to algorithm (6.4-1), and through GMP (see Chapters 3 and 5) we determine the output membership function. Analytically the operation of inferring a fuzzy output at any given time step may be written as

$$\begin{aligned}\mu_{C'}(u) = {} & \phi(\mathrm{DOF}_1(k), \mu_{C_1}(u)) \\ & \vee\ \phi(\mathrm{DOF}_2(k), \mu_{C_2}(u)) \\ & \cdots \\ & \vee\ \phi(\mathrm{DOF}_j(k), \mu_{C_j}(u)) \\ & \cdots \\ & \vee\ \phi(\mathrm{DOF}_n(k), \mu_{C_n}(u))\end{aligned} \tag{6.4-2}$$

for implication operators ϕ where the connective *ELSE* is interpreted as *union* (see Table 5.5). For implication operators interpreted as *intersection* we change ($\vee$) in equation (6.4-2) to ($\wedge$). Equation (6.4-2) tells us that the collective output of the controller depends on aggregating the outputs of individual rules with the output of each rule depending, in turn, on the degree of fulfillment plus the consequent membership function of the rule.

The degree of fulfillment of the jth rule DOF_j in (6.4-2) depends on the interpretation of the connective *AND* (generally thought of as a T norm (see Appendix)). If *AND* is analytically described as min ($\wedge$), the *degree of fulfillment* at timestep k is

$$\mathrm{DOF}_j(k) = \mu_{A_j}(x') \wedge \mu_{B_j}(y') \tag{6.4-3}$$

where x' and y' are the measured input values at a given time k. If *AND* is analytically described as *product* ($\cdot$), the DOF is

$$\mathrm{DOF}_j(k) = \mu_{A_j}(x') \cdot \mu_{B_j}(y') \tag{6.4-4}$$

It should be noted that the DOF is a function of time as different input values activate the rules to different degrees at different times. Equation (6.4-2) gives the fuzzy output (before defuzzification) of the controller in a general form and helps us to identify choices the developer needs to make such as the appropriate fuzzy implication operator ϕ and the associated interpretation of the connective *ELSE*, the form of DOF, and the defuzzification method.

In the design of fuzzy systems it is important to adequately cover the state space of the problem. Generally the development of a rule set that is both complete and correct is one of the most difficult problems in fuzzy control. Although various approaches have been suggested for learning a control algorithm on-line and adapting it to changing process conditions (Graham and Newell, 1988; Cox, 1993), this is still a rather heuristic process, and a good understanding of the various factors influencing the output of the controller is very helpful in its development and evaluation. Generally, which rules and to what extent will contribute toward an output at any given time depends primarily on the *form of the degree of fulfillment* (min or product), the *defuzzification method*, and the *implication operator*.

Let us consider the jth rule of (6.4-1) where triangular membership functions are used as shown in Figure 6.18. We assume min ($\wedge$) form for DOF as in equation (6.4-3). We also assume common quantized universe of discourse for all variables, of the type shown in Table 6.1. In Figure 6.18 we see the part of the Cartesian product of LHS variables covered by the jth rule. The $x \times y$ plane is the *state space* of our system. The state space covered by the jth rule is a square of six units edge, centered at (x_{jc}, y_{jc}) as shown in the figure. At time $t = k$, crisp inputs (x', y') are given to the rule. Let us first see what happens when the point (x', y') is located within the innermost square centered at (x_{jc}, y_{jc}) that has an edge of 2 units as shown in Figure 6.18. In such a case the degree of fulfillment DOF_j of the jth rule will be the same regardless of the exact location of point (x', y') so long as it remains within this particular square, since we have that

$$\begin{aligned}
\mu_{A_j}(x_{jc}+1) \wedge \mu_{B_j}(y_{jc}+1) &= 0.67 \wedge 0.67 = 0.67 \\
\mu_{A_j}(x_{jc}) \wedge \mu_{B_j}(y_{jc}-1) &= 0.67 \wedge 0.67 = 0.67 \\
\mu_{A_j}(x_{jc}) \wedge \mu_{B_j}(y_{jc}+1) &= 0.67 \wedge 0.67 = 0.67 \\
\mu_{A_j}(x_{jc}-1) \wedge \mu_{B_j}(y_{jc}-1) &= 0.67 \wedge 0.67 = 0.67 \\
\mu_{A_j}(x_{jc}-1) \wedge \mu_{B_j}(y_{jc}) &= 0.67 \wedge 0.67 = 0.67 \\
\mu_{A_j}(x_{jc}-1) \wedge \mu_{B_j}(y_{jc}+1) &= 0.67 \wedge 0.67 = 0.67
\end{aligned} \qquad (6.4\text{-}5)$$

Thus the DOF is 0.67 everywhere within this innermost square. Similar considerations lead us to the conclusion that if the point (x', y') falls within a square of edge 4, the DOF is 0.33 everywhere, whereas if it falls outside, the DOF is 0. Thus the distribution of the DOFs of a rule centered at (x_{jc}, y_{jc}) is as shown in Table 6.3.

When at time $t = k$ the crisp inputs (x', y') are given to the controller, the DOF of individual rules depends linearly on the distance between the input and the center (or peak) of the rule (x_{jc}, y_{jc}). Obviously the number of rules that will influence and contribute to the collective fuzzy output at any given time are only those within a distance d from input (x', y') (see Figure 6.18). Thus in a control algorithm, only the part of state space a distance d from a crisp input needs to be considered for rules that may be "fired." The rest have DOF = 0. The distance d is taken to be half of the support of a fuzzy value (considering the support to be where the membership function is not trivial). As shown in Figure 6.18, the edge of a square with (x_{jc}, x_{jc}) at its

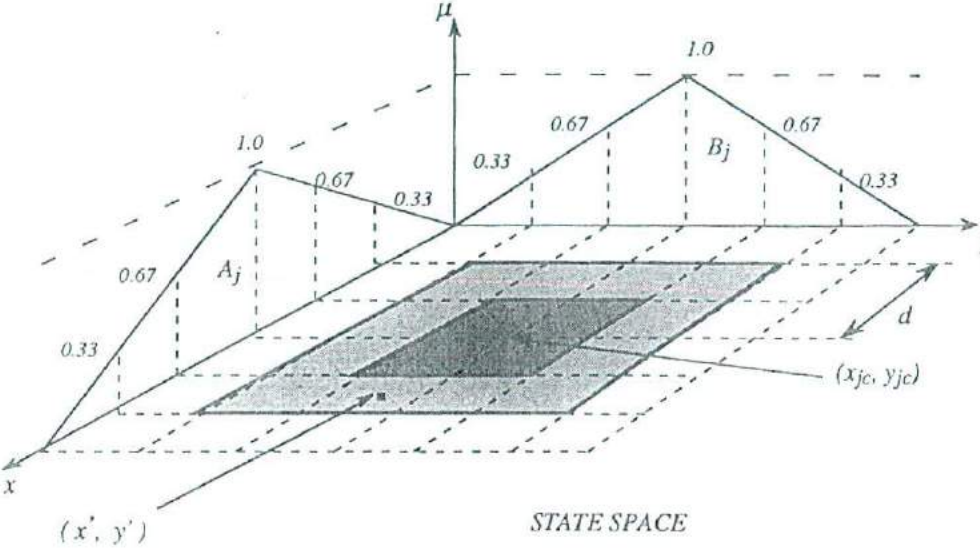


Figure 6.18 Region of influence of a rule in state space.

Table 6.3 Distribution of DOFs around rule center (rule with min DOF)

0	0	0	0	0	0	0
0	0.33	0.33	0.33	0.33	0.33	0
0	0.33	0.67	0.67	0.67	0.33	0
0	0.33	0.67	1.0	0.67	0.33	0
0	0.33	0.67	0.67	0.67	0.33	0
0	0.33	0.33	0.33	0.33	0.33	0
0	0	0	0	0	0	0

center is $2d$. We assumed of course that the two fuzzy values A_j and B_j have supports of the same length. When the support sets do not have the same length, then instead of a square we have a parallelogram and the distance of (x', y') from (x_{jc}, x_{jc}) will vary directionally as we move in different locations of state space.

The number of rules contributing to the fuzzy output depends also on the defuzzification method. When MOM is used, only the rules that are very close to the input (x', y') contribute maximum values to the output and therefore they are the only ones that need to be taken into account. We recall that with MOM, only the maximum values of the various contributions to the fuzzy output are used; and hence, only rules with high DOF and therefore small distance from (x', y') will influence the output. When COA is used, all the rules within a distance d from (x', y') need to be taken into account. Of course their contribution is in proportion to their distance from the input. Those which are the closest have the highest degree of fulfillment and therefore contribute more than those far away. Nonetheless, all rules within a distance d from (x', y') need to be taken into account.

On the other hand, if product is used in the DOF—that is, $\text{DOF}_j = \mu_{A_j}(x_k) \cdot \mu_{B_j}(y_k)$—the distribution of the DOF for input values in the vicinity of the jth rule would be as shown in Table 6.4. We see that DOF is varying with distance from (x_{jc}, y_{jc}) in a nonlinear manner. Again, if COA defuzzification is used, all the rules within a distance d need to be taken into account. When MOM is used, the rule peaking at (x_{jc}, y_{jc}) will have less influence than the earlier situation when the degree of fulfillment was defined through min. In the present case, it will influence the rule in a directional manner. For this reason with product DOF, COA defuzzification is more appropriate. Sometimes we may have very low DOFs, and therefore a cutoff number ought to be used to limit the number of rules that need to be considered. Thus we may choose $\text{DOF}_j = \alpha$ and ignore rules below whose DOF is less than α.

When continuous instead of discrete fuzzy values are used, their membership functions can be defined by various functions such as *S-shaped* and *Π-shaped* functions (see Sections 2.6.3, 2.6.4, and 2.6.5). Similar considerations hold for such cases as for discretized membership functions. With MOM, tremendous accuracy is not required since only the relative size of the membership values influences the final result and not the precise magnitude. Thus in order to take advantage of the fact that we have more precise membership values with continuous membership functions, it is best to utilize the COA (or COS) method of defuzzification.

We turn our attention now to the influence of the shape of membership functions describing the antecedent and consequent fuzzy values. The support of the fuzzy values of the antecedents (e.g., $2d$) determines the area of influence of every rule and hence plays a crucial role in the calculation of the control value. Generally the *shape* of the membership functions of LHS values has a substantial impact on the computation of the control action at any given time since it affects the DOF of each rule. Obviously the shape of

Table 6.4 Distribution of DOFs around rule center (rule with product DOF)

0	0	0	0	0	0	0
0	0.09	0.21	0.33	0.21	0.09	0
0	0.21	0.49	0.67	0.49	0.21	0
0	0.33	0.67	1.0	0.67	0.33	0
0	0.21	0.49	0.67	0.49	0.21	0
0	0.09	0.21	0.33	0.21	0.09	0
0	0	0	0	0	0	0

the RHS membership function affects directly the contribution of the rule to the overall fuzzy output.

When MOM is used, the exact shape of the LHS membership functions does not play a major role provided that it is in the general shape of a "hill, and symmetric with respect to a normal point." MOM defuzzification effectively distinguishes the rules with the highest priority (highest DOF) that is, the rules closest to the input (x', y'). Thus with MOM, the DOF suggests the distance from (x', y') and therefore the absolute values of the membership functions are not crucial, just their magnitude in relation to the membership functions of other rules. Similarly the exact shape of RHS membership functions does not play a crucial role in the calculation of the crisp output. When the support is not symmetric, the peaks in the membership function of the antecedents move relative to the support set and thus offer different DOF for the same inputs to the controller and different nonsymmetric shapes of "hills."

When COA defuzzification is used, the exact shape of the membership functions of antecedent as well as consequent plays an important role, even when symmetric membership functions are used. This happens because COA defuzzification takes into account the area under the curve of the total fuzzy output at any given time. This area is influenced directly by the shape of the consequent membership functions of the contributing rules, and indirectly through the DOF (the shape of the consequent membership functions of the contributing rules). The above-mentioned influence on the crisp controller output is emphasized (accentuated) even more if nonsymmetric membership functions are used, as well as if different membership functions are used for the different variables.

Let us now examine the influence of fuzzy implication operator ϕ on the computations of the controller output at a time $t = k$. We consider a hypothetical case where only one rule exists in the vicinity of (x', y'); that is, only one rule fires. Suppose that we use Mamdani min implication operator ϕ_c and fuzzy sets defined through symmetric triangular functions of the form shown in Table 6.1 (also Figure 6.5). Figure 6.19 (top) shows what happens to the RHS value for different degrees of fulfillment of a rule. The consequent membership function remains the same for DOF = 1 and is gradually clipped, finally becoming zero when DOF = 0. The defuzzified output is the same with either COA or MOM methods. The situation is similar when Larsen product implication operator ϕ_p is used as can be seen in Figure 6.18.

On the other hand, if the *Boolean* implication operator ϕ_b is used, a "plateau" is created that grows, as DOF is getting smaller, until it covers the entire universe of discourse when DOF = 0. While this is exactly the opposite of what happens when the Mamdani min implication operator is used, it is counterbalanced by interpreting *ELSE* as intersection (min) when a number of rules are connected in order to compute the total fuzzy output of the controller. In fact, this is the reason for using min for the connective *ELSE* with this implication operator (see Table 5.5). It should be noted from Figure 6.19 that COA and MOM defuzzification may give different crisp outputs when Boolean implication is used.

If the *arithmetic* implication operator ϕ_a is used, a "plateau" is also formed as it happens with Boolean implication. The peak of the function is *clipped* as with Mamdani min implication. When DOF = 0, the "plateau" covers the entire universe of discourse. Again it should be noted that COA and MOM defuzzification may give different crisp outputs.

As can be seen in Figure 6.19, when Mamdani min and Larsen product implications are used, both COA and MOM defuzzification methods give similar results. In Boolean and arithmetic implications, on the other hand, the two defuzzification methods will give rather different results due to the developing plateaus. When plateaus appear, MOM defuzzification is better because COA considers the peak of the rule together with the developing plateau, and hence it shifts the final crisp output away from the location that is suggested by the peak of the rule. This is undesirable since "plateaus" do not contain useful information. They can be interpreted as a fuzzy value "*unknown*." In Figure 6.19 we note that when DOF = 0, Mamdani min and Larsen product implications give "nothing" as the output of the controller.

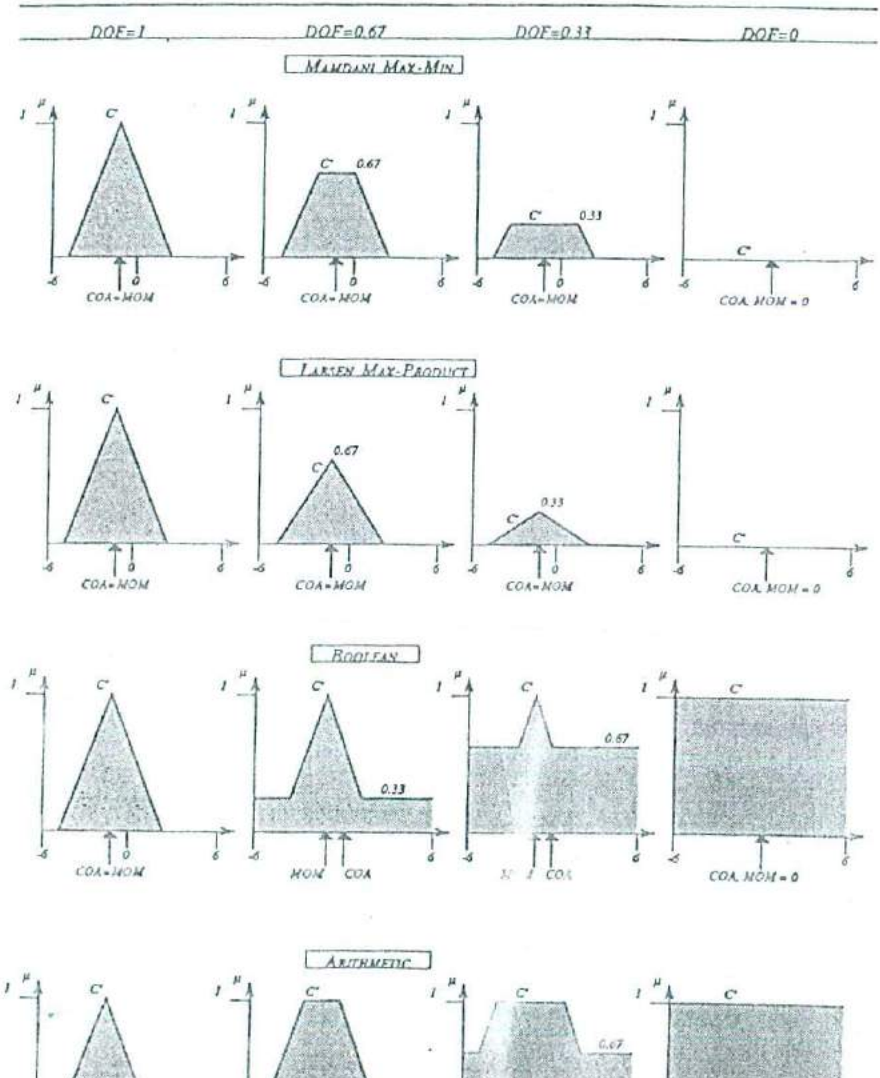


Figure 6.19 Influence on output of various implications under different degrees of fulfillment.

On the other hand, using the Boolean and arithmetic implications produces *unknown* as output. In all cases we can set the output to crisp zero.

In addition, with Mamdani min and Larsen product implications we may effectively use either method of defuzzification since the total fuzzy output contains contributions from all (or many) rules. Generally, if a rule or several rules are an equal distance away from the point (x', y'), we have the same results with either defuzzification method. With Boolean and arithmetic implications we do not need to use COA defuzzification (which is actually computationally more demanding) since in many cases the total fuzzy output does not effectively represent the contribution of the individual rules (due to the "plateau" or "flattening" effect). In general, it is preferable (but not required) to use MOM in conjunction with Boolean and arithmetic implications.

REFERENCES

Bernard, J. A., Use of a Rule-Based System for Process Control, *IEEE Control Systems Magazine*, pp. 3–13, October 1988.

Cox, E., Adaptive Fuzzy Systems, *IEEE Spectrum*, pp. 27–31, February 1993.

Driankov, D., Hellendoorn, H., and Reinfrank, M., *An Introduction to Fuzzy Control*, Springer-Verlag, Berlin, 1993.

Dubois, D., and Prade, H., *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, Boston, 1980.

Efstathiou, J., Rule-Based Process Control Using Fuzzy Logic, in *Approximate Reasoning in Intelligent Systems, Decision and Control*, E. Sanchez and L. A. Zadeh, eds., Pergamon Press, Oxford, 1987, pp. 145–158.

Fuller, R., and Zimmermann, H.-J., On Computation of the Compositional Rule of Inference under Triangular Norms, *Fuzzy Sets and Systems*, Vol. 51, pp. 267–275, 1992.

Graham, B. P., and Newell, R. B., Fuzzy Identification and Control of a Liquid Level Rig, *Fuzzy Sets and Systems*, Vol. 26, pp. 255–273, 1988.

Harris, C. J., Moore, C. G., and Brown, M., *Intelligent Control—Aspects of Fuzzy Logic and Neural Nets*, World Scientific, Singapore, 1993.

He, S-Z., Tan, S., and Xu, F-L., Fuzzy Self-Tuning of PID Controllers, *Fuzzy Sets and Systems*, Vol. 56, pp. 37–46, 1993.

Hirota, K., *Industrial Applications of Fuzzy Control*, Springer-Verlag, Tokyo, 1993.

Jager, R., *Fuzzy Logic in Control*, Unpublished Ph.D. Dissertation, University of Delft, The Netherlands, 1995.

Jang, J-S., and Gulley, N., *Fuzzy Logic Toolbox User's Guide*, Mathworks, 1995.

Jang, J-S., and Sun, C-T., Neuro-Fuzzy Modeling and Control, *Proceedings of the IEEE*, Vol. 83, No. 3, pp. 378–406, 1995.

Jiangin, C., and Laijiu, C., Study on Stability of Fuzzy Closed-Loop Control Systems, *Fuzzy Sets and Systems*, Vol. 57, pp. 159–168, 1993.

Johnson, C. D., *Process Control Instrumentation Technology*, John Wiley & Sons, New York, 1977.

King, P. J., and Mamdani, E. H., The Application of Fuzzy Control Systems to Industrial Processes, in *Fuzzy Automata and Decision Processes*, M. M. Gupta, G. N. Saridis, and B. R. Gaines, eds., North-Holland, New York, 1977, pp. 321–330.

King, R. E., and Karonis, F. C., Multi-Level Expert Control of a Large-Scale Industrial Process, in *Fuzzy Computing Theory, Hardware, and Applications*, M. M. Gupta and T. Yamakawa, eds., Elsevier/North-Holland, Amsterdam, 1988, pp. 323–340.

Kiszka, J. B., Gupta, M. M., and Nikiforuk, P. N., Some Properties of Expert Control Systems, in *Approximate Reasoning in Expert Systems*, M. M. Gupta, A. Kandel, W. Bandler, and J. B. Kiszka, eds., Elsevier/North-Holland, Amsterdam, 1985, pp. 283–306.

Larkin, L. I., A Fuzzy Logic Controller for Aircraft Flight Control, in *Industrial Applications of Fuzzy Control*, M. Sugeno, ed., North-Holland, New York, 1985, pp. 87–103.

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, March/April 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.

Maeda, M., and Murakami, S., A Self-Tuning Fuzzy Controller, *Fuzzy Sets and Systems*, Vol. 51, pp. 29–40, 1992.

Mamdani, E. H., Application of Fuzzy Algorithms for Control of Simple Dynamic Plants, *Proceedings of IEE*, Vol. 121, No. 12, pp. 1585–1588, 1974.

Mamdani, E. H., Advances in the Linguistic Synthesis of Fuzzy Controllers, *International Journal of Man-Machine Studies*, Vol. 8, pp. 669–678, 1976.

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.

Matia, F., Jimenez, A., Galan, R., and Sanz, R., Fuzzy Controllers: Lifting the Linear–Nonlinear Frontier, *Fuzzy Sets and Systems*, Vol. 52, pp. 113–128, 1992.

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

Ostergaard, J. J., and Holmblad, L. P., Control of Cement Kiln by Fuzzy Logic, in *Fuzzy Information and Decision Processes*, M. M. Gupta and E. Sanchez, eds., North-Holland, Amsterdam, 1982, pp. 389–399.

Pedrycz, W., *Fuzzy Control and Fuzzy Systems*, second (extended) edition, John Wiley & Sons, New York, 1993.

Schwartz, D. G., Fuzzy Logic Flowers in Japan, *IEEE Spectrum*, pp. 32–35, July 1992.

Shoureshi, R., and Rahmani, K., Derivation and Application of an Expert Fuzzy Optimal Control System, *Fuzzy Sets and Systems*, Vol. 49, pp. 93–101, 1992.

Sugeno, M., An Introductory Survey of Fuzzy Control, *Information Sciences*, Vol. 36, pp. 59–83, 1985.

Terano, T., Asai, K., and Sugeno, M., *Fuzzy Systems Theory and Its Applications*, Academic Press, Boston, 1992.

Tsukamoto, Y., An Approach to Fuzzy Reasoning Method, *Advances in Fuzzy Set Theory and Applications*, M. M. Gupta, R. K. Ragade, and R. R. Yager, eds., North-Holland, Amsterdam, 1979, pp. 134–149.

Tzafestas, S. G., and Papanikolopoulos, N. P., Incremental Fuzzy Expert PID Control, *IEEE Transactions on Industrial Electronics*, Vol. 37, No. 5, pp. 365–371, 1990.

Wang, Li-Xin, *Adaptive Fuzzy Systems and Control*, Prentice-Hall, Englewood Cliffs, NJ, 1994.

Yager, R. R., and Filev, D. P., SLIDE: A Simple Adaptive Defuzzification Method, *IEEE Transactions on Fuzzy Systems*, Vol. 1, No. 1, pp. 69–78, 1993.

Yager, R. R., and Filev, D. P., *Essentials of Fuzzy Modeling and Control*, John Wiley & Sons, New York, 1994.

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., Stabilization of an Inverted Pendulum by a High-Speed Fuzzy Logic Controller Hardware System, *Fuzzy Sets and Systems*, Vol. 32, pp. 161–180, 1989.

Yasunobu, S., and Miyamoto, S., Automatic Train Operation System by Predictive Fuzzy Control, in *Industrial Applications of Fuzzy Control*, M. Sugeno, ed., North-Holland, Amsterdam, 1985, pp. 1–18.

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

PROBLEMS

1. A fuzzy control system used inputs of error e and change in error Δe to control an output variable u. Their fuzzy membership functions have the following characteristics:

Variable	Range		Description of membership function
e Error (%)	-20 to $+20$	N (negative)	Straight line from 1 at -20% to 0 at 0%
		Z (zero)	Straight line from 0 at -20% to 1 at 0% and another straight line from 1 at 0% to 0 at $+20\%$
		P (positive)	Straight line from 0 at 0% to 1 at $+20\%$
Δe Change in error (%/min)	-10 to $+10$	N (negative)	Straight line from 1 at -10%/min to 0 at $+10\%$/min
		P (positive)	Straight line from 0 at -10%/min to 1 at $+10\%$/min
u Output (%)	-25 to $+25$	N (negative)	Straight line from 1 at -25% to 0 at 0%
		Z (zero)	Straight line from 0 at -25% to 1 at 0% and another straight line from 1 at 0% to 0 at $+25\%$
		P (positive)	Straight line from 0 at 0% to 1 at $+25\%$

The fuzzy algorithm is given below. Determine the output u for $e' = +16\%$ and $\Delta e' = -2\%$/min using the Mamdani min implication operator and max–min composition (as well as min interpretation for *AND* in the degree of fulfillment). Use the Center of area method to defuzzify the answer. (Sketch the various membership functions involved and show how you obtained your solution.)

FUZZY ALGORITHM

R_1	*if e is N*	*AND*	Δe *is N*	*then u is P*	*ELSE*
R_2	*if e is N*	*AND*	Δe *is P*	*then u is P*	*ELSE*
R_3	*if e is Z*	*AND*	Δe *is N*	*then u is Z*	*ELSE*
R_4	*if e is Z*	*AND*	Δe *is P*	*then u is Z*	*ELSE*
R_5	*if e is P*	*AND*	Δe *is N*	*then u is N*	*ELSE*
R_6	*if e is P*	*AND*	Δe *is P*	*then u is N*	

2. Repeat Problem 1 using the Larsen product implication, max–min composition, and product for the degree of fulfillment.

3. In Problem 1, the error starts at a value of +16% at time 0 and decreases at a rate of 2%/min for 4 minutes. Determine the output u at times $t = 0$, 1, 2, 3, and 4.

4. Analyze the fuzzy controller given in Problem 1 using the criteria given in Section 4. Are there contradictions within the rule set? Is there a dominant rule? Are the rules covering the state space in a satisfactory manner?

5. Using MATLAB, draw the control hypersurface for the fuzzy controller given in Problem 1. Simulate the controller for the range of all possible inputs and answer the questions posed in Problem 4.

6. Show what the different interpretations for *ELSE* could be for the fuzzy controller of Problem 1 and the implication operators given in Table 5.2.

II

NEURAL NETWORKS: CONCEPTS AND FUNDAMENTALS