

Tutorial On Fuzzy Logic

Jan Jantzen <jj@iau.dtu.dk>¹

Abstract

A logic based on the two truth values *True* and *False* is sometimes inadequate when describing human reasoning. Fuzzy logic uses the whole interval between 0 (*False*) and 1 (*True*) to describe human reasoning. As a result, fuzzy logic is being applied in rule based automatic controllers, and this paper is part of a course for control engineers.

Contents

1	Introduction	2
2	Fuzzy Sets	3
2.1	Universe	5
2.2	Membership function	5
2.3	Singletons	7
2.4	Linguistic variables	7
2.5	Tank Level Example	7
3	Operations On Fuzzy Sets	8
3.1	Set operations	8
3.2	Modifiers	9
3.3	Relations Between Sets	11
4	Fuzzy Logic	12
4.1	Connectives	13
4.2	Implication	16
4.3	Inference	17
4.4	Several Rules	18
5	Summary	19

¹ Technical University of Denmark, Department of Automation, Bldg 326, DK-2800 Lyngby, DENMARK. Tech. report no 98-E 868, 19 Aug 1998 (logic).

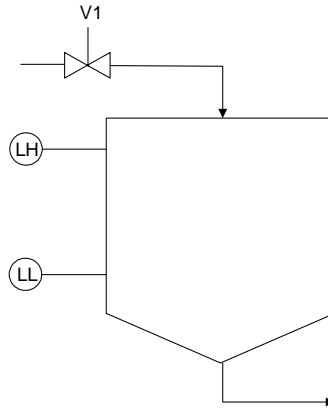


Figure 1: Process diagram of a feed-tank.

1. Introduction

A fuzzy controller, in a cement plant for example, aims to mimic the operator's terms by means of fuzzy logic. To illustrate, consider the tank in Fig. 1, which is for feeding a cement mill such that the feed flow is more or less constant. The simplified design in the figure consists of a tank, two level sensors, and a magnetic valve. The objective is to control the valve V_1 , such that the tank is refilled when the level is as low as LL , and stop the refilling when the level is as high as LH . The sensor LL is 1 when the level is above the mark, and 0 when the level is below; likewise with the sensor LH . The valve opens when V_1 is set to 1, and it closes when V_1 is set to 0. In two-valued (Boolean) logic the controller can be described

$$V_1 = \left\{ \begin{array}{l} 1, \quad \text{if } LL \text{ switches from 1 to 0} \\ 0, \quad \text{if } LH \text{ switches from 0 to 1} \end{array} \right\} \quad (1)$$

An operator, whose responsibility is to open and close the valve, would perhaps describe the control strategy as:

$$\begin{array}{l} \text{If the level is low then open } V_1 \\ \text{If the level is high then close } V_1 \end{array} \quad (2)$$

The former strategy (1) is suitable for a Programmable Logic Controller (PLC) using Boolean logic, and the latter (2) is suitable for a fuzzy controller using fuzzy logic. Our aim here is not to give implementation details of the latter, but to use the example to explain the underlying fuzzy logic.

Lotfi Zadeh, the father of fuzzy logic, claimed that many *sets* in the world that surrounds us are defined by a non-distinct boundary. Indeed, the *set of high mountains*, or, the *set of low level measurements* in Fig 1 are examples of such sets. Zadeh decided to extend two-valued logic, defined by the binary pair $\{0, 1\}$, to the whole continuous interval $[0, 1]$, thereby introducing a gradual transition from falsehood to truth. The original and

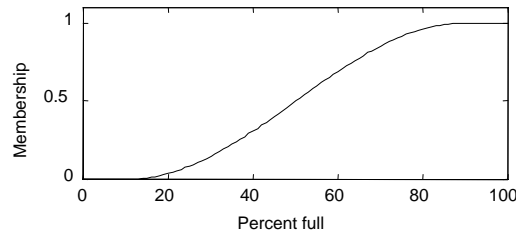


Figure 2: Possible definition of the set *high levels* in the tank in Fig. 1.

pioneering papers on fuzzy sets by Zadeh (*e.g.*, 1965, 1973, 1975) explain the theory of fuzzy sets that result from the extension as well as a fuzzy logic based on the set theory. Primary references can be found conveniently in a book with 18 selected papers by Zadeh (Yager, Ovchinnikov, Tong & Nguyen, 1987). For a thorough introduction to the theory, Zadeh in his article in *IEEE Spectrum* (Zadeh, 1984) recommends the book by Kaufmann (1975). A more recent introduction to fuzzy set theory and its applications is the book by Zimmermann (1993) which is easy to read. Specific questions or definitions can be looked up in the *Systems and Control Encyclopedia* (Singh, 1987; 1990; 1992). The book has a large collection of articles on control concepts in general, and fuzzy control in particular.

Here we will focus on the fuzzy set theory underlying (2), and present the basic definitions and operations. Please be aware that the interpretation of fuzzy set theory in the following is just one of several possible; Zadeh and other authors have suggested alternative definitions. Throughout, letters denoting matrices are in bold upper case, for example **A**; vectors are in bold lower case, for example **x**; scalars are in italics, for example *n*; and operations are in bold, for example **min**.

2. Fuzzy Sets

Fuzzy sets are a further development of the mathematical concept of a set. Sets were first studied formally by the German mathematician Georg Cantor (1845-1918). His theory of sets met much resistance during his lifetime, but nowadays most mathematicians believe it is possible to express most, if not all, of mathematics in the language of set theory. Many researchers are looking at the consequences of 'fuzzifying' set theory, and much mathematical literature is the result. For control engineers, fuzzy logic and fuzzy relations are the most important in order to understand how fuzzy rules work.

Conventional sets A *set* is any collection of objects which can be treated as a whole. Cantor described a set by its members, such that an item from a given universe is either a member or not. The terms *set*, *collection* and *class* are synonyms, just as the terms *item*, *element* and *member*. Almost anything called *a set* in ordinary conversation is an acceptable set in the mathematical sense, cf. the next example.

Example 1 (sets) *The following are well defined lists or collections of objects, and therefore entitled to be called sets:*

- (a) *The set of non-negative integers less than 4. This is a finite set with four members: 0, 1, 2, and 3.*
- (b) *The set of live dinosaurs in the basement of the British Museum. This set has no members, and is called an empty set.*
- (c) *The set of measurements greater than 10 volts. Even though this set is infinite, it is possible to determine whether a given measurement is a member or not.*

A set can be specified by its members, they characterize a set completely. The list of members $A = \{0, 1, 2, 3\}$ specifies a finite set. Nobody can list all elements of an *infinite* set, we must instead state some property which characterizes the elements in the set, for instance the predicate $x > 10$. That set is defined by the elements of the universe of discourse which make the predicate true. So there are two ways to describe a set: explicitly in a list or implicitly with a predicate.

Fuzzy sets Following Zadeh many sets have more than an *either-or* criterion for membership. Take for example the set of *young people*. A one year old baby will clearly be a member of the set, and a 100 years old person will not be a member of this set, but what about people at the age of 20, 30, or 40 years? Another example is a weather report regarding high temperatures, strong winds, or nice days. In other cases a criterion appears nonfuzzy, but is perceived as fuzzy: a speed limit of 60 kilometres per hour, a check-out time at 12 noon in a hotel, a 50 years old man. Zadeh proposed a *grade of membership*, such that the transition from membership to non-membership is gradual rather than abrupt.

The grade of membership for all its members thus describes a fuzzy set. An item's grade of membership is normally a real number between 0 and 1, often denoted by the Greek letter μ . The higher the number, the higher the membership (Fig. 2). Zadeh regards Cantor's set as a special case where elements have full membership, *i.e.*, $\mu = 1$. He nevertheless called Cantor's sets *nonfuzzy*; today the term *crisp* set is used, which avoids that little dilemma.

Notice that Zadeh does not give a formal basis for how to determine the grade of membership. The membership for a 50 year old in the set *young* depends on one's own view. The grade of membership is a precise, but subjective measure that depends on the context.

A fuzzy membership function is different from a statistical probability distribution. This is illustrated next in the so-called egg-eating example.

Example 2 (Probability vs possibility) *(Zadeh in Zimmermann, 1991) Consider the statement "Hans ate X eggs for breakfast", where $X \in U = \{1, 2, \dots, 8\}$. We may associate a probability distribution p by observing Hans eating breakfast for 100 days,*

$$\begin{aligned} U &= [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8] \\ p &= [.1 \ .8 \ .1 \ 0 \ 0 \ 0 \ 0 \ 0] \end{aligned}$$

A fuzzy set expressing the grade of ease with which Hans can eat X eggs may be the following so-called possibility distribution π ,

$$\begin{aligned} U &= [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8] \\ \pi &= [1 \ 1 \ 1 \ 1 \ .8 \ .6 \ .4 \ .2] \end{aligned}$$

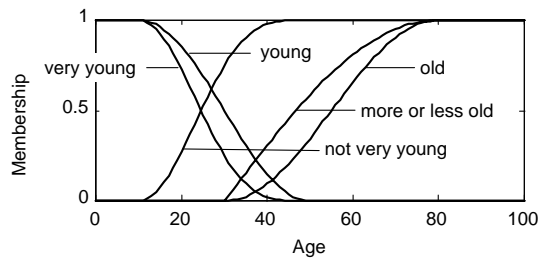


Figure 3: The sets *more or less old*, *very young*, and *not very young* are derived from *young* and *old*.

Where the possibility for $X = 3$ is 1, the probability is only 0.1.

The example shows, that a possible event does not imply that it is probable. However, if it is probable it must also be possible. You might view a fuzzy membership function as your personal distribution, in contrast with a statistical distribution based on observations.

2.1 Universe

Elements of a fuzzy set are taken from a *universe of discourse*, or *universe* for short. The universe contains all elements that can come into consideration. Even the universe depends on the context, as the next example shows.

Example 3 (universe) (a) The set of young people could have all human beings in the world as its universe. Alternatively it could be the numbers between 0 and 100; these would then represent age (Fig. 3).

(b) The set $x \gg 10$ (x much greater than 10) could have as a universe all positive measurements.

An application of the universe is to suppress faulty measurement data, for example negative values for the level in our tank example.

In case we are dealing with a non-numerical quantity, for instance *taste*, which cannot be measured against a numerical scale, we cannot use a numerical universe. The elements are then said to be taken from a *psychological continuum*; an example of such a universe could be $\{\text{bitter, sweet, sour, salt, hot, ...}\}$.

2.2 Membership function

Every element in the universe of discourse is a member of the fuzzy set to some grade, maybe even zero. The set of elements that have a non-zero membership is called the *support* of the fuzzy set. The function that ties a number to each element x of the universe is called the *membership function* $\mu(x)$.

Continuous And Discrete Representations There are two alternative ways to represent a membership function in a computer: continuous or discrete. In the continuous form the membership function is a mathematical function, possibly a program. A membership function is for example bell-shaped (also called a π -curve), s-shaped (called an s -curve), a reverse s -curve (called z -curve), triangular, or trapezoidal. There is an example of an s -curve in Fig. 2. In the discrete form the membership function and the universe are discrete points in a list (vector). Sometimes it can be more convenient with a sampled (discrete) representation.

As a very crude rule of thumb, the continuous form is more CPU intensive, but less storage demanding than the discrete form.

Example 4 (continuous) A cosine function can be used to generate a variety of membership functions. The s -curve can be implemented as

$$s(x_l, x_r, x) = \begin{cases} 0 & , x < x_l \\ \frac{1}{2} + \frac{1}{2} \cos\left(\frac{x-x_r}{x_r-x_l}\pi\right) & , x_l \leq x \leq x_r \\ 1 & , x > x_r \end{cases} \quad (3)$$

where x_l is the left breakpoint, and x_r is the right breakpoint. The z -curve is just a reflection,

$$z(x_l, x_r, x) = \begin{cases} 1 & , x < x_l \\ \frac{1}{2} + \frac{1}{2} \cos\left(\frac{x-x_l}{x_r-x_l}\pi\right) & , x_l \leq x \leq x_r \\ 0 & , x > x_r \end{cases} \quad (4)$$

Then the π -curve can be implemented as a combination of the s -curve and the z -curve, such that the peak is flat over the interval $[x_2, x_3]$,

$$\pi(x_1, x_2, x_3, x_4, x) = \min(s(x_1, x_2, x), z(x_3, x_4, x)) \quad (5)$$

Figure 2 was drawn using $\pi(10, 90, 100, 100, x)$.

Example 5 (discrete) To get a discrete representation equivalent to Fig. 2, assume the universe \mathbf{u} is represented by a number of samples, say,

$$\mathbf{u} = [0 \quad 20 \quad 40 \quad 60 \quad 80 \quad 100]$$

Insertion results in the corresponding list of membership values:

$$\begin{aligned} \pi(10, 90, 100, 100, \mathbf{u}_1) &= 0 \\ \pi(10, 90, 100, 100, \mathbf{u}_2) &= 0.04 \\ \pi(10, 90, 100, 100, \mathbf{u}_3) &= 0.31 \\ \pi(10, 90, 100, 100, \mathbf{u}_4) &= 0.69 \\ \pi(10, 90, 100, 100, \mathbf{u}_5) &= 0.96 \\ \pi(10, 90, 100, 100, \mathbf{u}_6) &= 1 \end{aligned}$$

or, for short,

$$\pi(10, 90, 100, 100, \mathbf{u}) = [0 \quad 0.04 \quad 0.31 \quad 0.69 \quad 0.96 \quad 1]$$

Normalisation A fuzzy set is *normalised* if its largest membership value equals 1. You normalise by dividing each membership value by the largest membership in the set, $\mathbf{a}/\max(\mathbf{a})$.

2.3 Singletons

Strictly speaking, a fuzzy set A is a collection of ordered pairs

$$A = \{(x, \mu(x))\} \quad (6)$$

Item x belongs to the universe and $\mu(x)$ is its grade of membership in A . A single pair $(x, \mu(x))$ is called a fuzzy *singleton*; thus the whole set can be viewed as the union of its constituent singletons. It is often convenient to think of a set A just as a vector

$$\mathbf{a} = (\mu(x_1), \mu(x_2), \dots, \mu(x_n))$$

It is understood then, that each position i ($1, 2, \dots, n$) corresponds to a point in the universe of n points.

2.4 Linguistic variables

Just like an algebraic variable takes numbers as values, a *linguistic variable* takes words or sentences as values (Zadeh in Zimmermann, 1991). The set of values that it can take is called its *term set*. Each value in the term set is a *fuzzy variable* defined over a *base variable*. The base variable defines the universe of discourse for all the fuzzy variables in the term set. In short, the hierarchy is as follows: linguistic variable \rightarrow fuzzy variable \rightarrow base variable.

Example 6 (term set) Let x be a linguistic variable with the label “Age”. Terms of this linguistic variable, which are fuzzy sets, could be “old”, “young”, “very old” from the term set

$$T = \{Old, VeryOld, NotSoOld, MoreOrLessYoung, \\ QuiteYoung, VeryYoung\}$$

Each term is a fuzzy variable defined on the base variable, which might be the scale from 0 to 100 years.

Primary terms A *primary term* is a term or a set that must be defined a priori, for example *Young* and *Old* in Fig. 3, whereas the sets *VeryYoung* and *NotYoung* are modified sets.

2.5 Tank Level Example

We have now come a little closer to the representation of a fuzzy control rule. In the premise

If the level is low,

clearly *low* is a fuzzy variable, a value of the linguistic variable *level*. The term *low* can be represented in the computer as a vector **low**. It is defined on a universe, which is the range

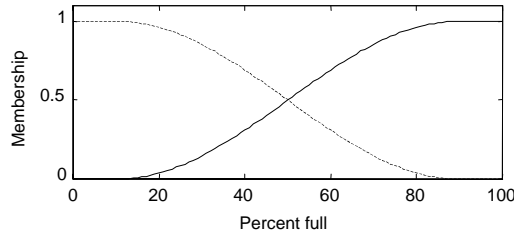


Figure 4: Two terms defining *high* tank levels (solid) and *low* tank levels (dashed).

of the expected values of *level*, i.e., the interval $[0, 100]$ percent full. The measurement *level* is a scalar, and the statement *level is low* looks up the membership value $\mathbf{low}(i)$, where *level* is rounded to the nearest element in the universe to find the appropriate index i . The outcome is a number $\mu = [0, 1]$ telling how well the premise is fulfilled. Figure 4 suggests a possible definition of the term set $\{low, high\}$ for the tank level problem.

3. Operations On Fuzzy Sets

The membership function is obviously a crucial component of a fuzzy set. It is therefore natural to define operations on fuzzy sets by means of their membership functions.

3.1 Set operations

In fact a fuzzy *set operation* creates a new set from one or several given sets (Fig. 5). For example, given the sets A and B the intersection is a new fuzzy set with its own membership function.

Definition 1 (set operations) *Let A and B be fuzzy sets on a mutual universe.*

(a) *The intersection of A and B is*

$$A \cap B \equiv \mathbf{a \min b}$$

*The operation **min** is an item-by-item minimum comparison between corresponding items in \mathbf{a} and \mathbf{b} .*

(b) *The union of A and B is*

$$A \cup B \equiv \mathbf{a \max b}$$

*where **max** is an item-by-item maximum operation.*

(c) *The complement of A is*

$$\overline{A} \equiv 1 - \mathbf{a}$$

where each membership value in \mathbf{a} is subtracted from 1.

A fuzzy set X is a *fuzzy subset* of the set Y , written $X \subseteq Y$, if its membership function is less than or equal to the membership function of Y . In Fig. 5 we have $(A \cap B) \subseteq (A \cup B)$.

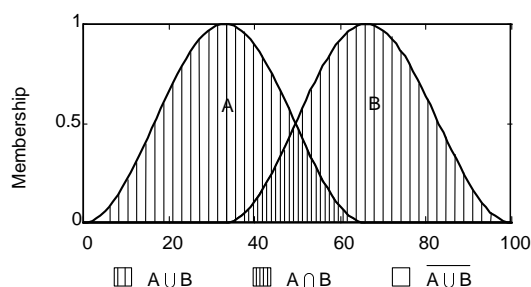


Figure 5: The three primitive set operations.

Example 7 (buy a house) (Zimmermann, 1993): A four-person family wants to buy a house. An indication of how comfortable they want to be is the number of bedrooms in the house. But they also want a large house. Let $\mathbf{u} = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$ be the set of available houses described by their number of bedrooms. Then the fuzzy set \mathbf{c} (for Comfortable) may be described as

$$\mathbf{c} = [0.2 \quad 0.5 \quad 0.8 \quad 1 \quad 0.7 \quad 0.3 \quad 0 \quad 0 \quad 0 \quad 0]$$

Let \mathbf{l} be the fuzzy set Large defined as

$$\mathbf{l} = [0 \quad 0 \quad 0.2 \quad 0.4 \quad 0.6 \quad 0.8 \quad 1 \quad 1 \quad 1 \quad 1]$$

The intersection of Comfortable and Large is then

$$\mathbf{c} \cap \mathbf{l} = [0 \quad 0 \quad 0.2 \quad 0.4 \quad 0.6 \quad 0.3 \quad 0 \quad 0 \quad 0 \quad 0]$$

To interpret this, five bedrooms is optimal, but only satisfactory to the grade 0.6. The second best solution is four bedrooms.

The union of Comfortable and Large is

$$\mathbf{c} \cup \mathbf{l} = [0.2 \quad 0.5 \quad 0.8 \quad 1 \quad 0.7 \quad 0.8 \quad 1 \quad 1 \quad 1 \quad 1]$$

Here four bedrooms is fully satisfactory (1) because it is comfortable, and 7-10 bedrooms also, because that would mean a large house. The complement of Large is

$$\bar{\mathbf{l}} = [1 \quad 1 \quad 0.8 \quad 0.6 \quad 0.4 \quad 0.2 \quad 0 \quad 0 \quad 0 \quad 0]$$

The operations \cup and \cap associate, commute and more (Table 1). These properties are important, because they help to predict the outcome of long sentences.

Other definitions of the primitive operations are possible, but using **max** and **min** is most common.

3.2 Modifiers

A linguistic *modifier*, is an operation that modifies the meaning of a term. For example, in the sentence “very close to 0”, the word *very* modifies *Close to 0* which is a fuzzy set. A modifier is thus an operation on a fuzzy set. Examples of other modifiers are *a little*, *more*

Property	Name
$A \cup B = B \cup A$	<i>Commutative</i>
$A \cap B = B \cap A$	<i>Commutative</i>
$(A \cup B) \cup C = A \cup (B \cup C)$	<i>Associative</i>
$(A \cap B) \cap C = A \cap (B \cap C)$	<i>Associative</i>
$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	<i>Distributive</i>
$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	<i>Distributive</i>
$\overline{A \cap B} = \overline{A} \cup \overline{B}$	<i>DeMorgan</i>
$\overline{A \cup B} = \overline{A} \cap \overline{B}$	<i>DeMorgan</i>
$(A \cap B) \cup A = A$	<i>Absorption</i>
$(A \cup B) \cap A = A$	<i>Absorption</i>
$A \cup A = A$	<i>Idempotency</i>
$A \cap A = A$	<i>Idempotency</i>
$A \cup \overline{A} \neq 1$	<i>Exclusion not satisfied</i>
$A \cap \overline{A} \neq 0$	<i>Exclusion not satisfied</i>

Table 1: Properties of the primitive operations

or less, possibly, and definitely.

Even though it is difficult precisely to say what effect the modifier *very* has, it does have an intensifying effect. The modifier *more or less*, or *morl* for short, has the opposite effect. They are often approximated by the operations

$$\mathbf{very\ a} \equiv \mathbf{a}^2, \mathbf{morl\ a} \equiv \mathbf{a}^{\frac{1}{2}}$$

The power function applies to each vector element of \mathbf{a} in turn. Here we have limited ourselves to squaring and square root, but any power function may be used. On discrete form, we might have a universe $\mathbf{u} = (0, 20, 40, 60, 80)$. Given the set

$$\mathbf{young} = [1 \quad 0.6 \quad 0.1 \quad 0 \quad 0]$$

then we can derive the discrete membership function for the set *very young* by squaring all elements,

$$\mathbf{young}^2 = [1 \quad 0.36 \quad 0.01 \quad 0 \quad 0]$$

The set *very very young* is by induction,

$$\mathbf{young}^4 = [1 \quad 0.13 \quad 0 \quad 0 \quad 0]$$

The derived sets inherit the universe of the primary set. The plots in Fig. 3 were generated using these definitions. Some examples of other modifiers are

$$\mathbf{extremely\ a} = \mathbf{a}^3$$

$$\mathbf{slightly\ a} = \mathbf{a}^{\frac{1}{3}}$$

$$\mathbf{somewhat\ a} = \mathbf{moreorless\ a\ and\ not\ slightly\ a}$$

A whole family of modifiers is generated by \mathbf{a}^p where p is any power between zero and infinity. With $p = \infty$ the modifier could be named **exactly**, because it would suppress all memberships lower than 1.0.

3.3 Relations Between Sets

In any fuzzy controller, relationships among objects play a fundamental role. Some relations concern elements within the same universe: one measurement is larger than another, one event occurred earlier than another, one element resembles another, etc. Other relations concern elements from disjoint universes: the measurement is large and its rate of change is positive, the x -coordinate is large and the y -coordinate is small, for example. These examples are relationships between two objects, but in principle we can have relationships which hold for any number of objects.

Formally, a *binary relation* or simply a *relation* R from a set A to a set B assigns to each ordered pair $(a, b) \in A \times B$ exactly one of the following statements: (i) "a is related to b", or (ii) "a is not related to b". The Cartesian product $A \times B$ is the set of all possible combinations of the items of A and B . A *fuzzy relation* from a set A to a set B is a fuzzy subset of the Cartesian product $U \times V$ between their respective universes U and V .

Assume for example that Donald Duck's nephew Huey resembles Dewey to the grade 0.8, and Huey resembles Louie to the grade 0.9. We have therefore a relation between to subsets of the nephews in the family. This is conveniently represented in a matrix (with one row),

$$\mathbf{R}_1 = \begin{array}{c} \text{Huey} \end{array} \begin{array}{cc} \text{Dewey} & \text{Louie} \\ \boxed{0.8} & \boxed{0.9} \end{array}$$

Composition In order to show how two relations can be combined let us assume another relation between Dewey and Louie on the one side, and Donald Duck on the other,

$$\mathbf{R}_2 = \begin{array}{c} \text{Dewey} \\ \text{Louie} \end{array} \begin{array}{c} \text{Donald} \\ \boxed{0.5} \\ \boxed{0.6} \end{array}$$

It is tempting to try and find out how much Huey resembles Donald by combining the information in the two matrices:

- (i) Huey resembles (0.8) Dewey, and Dewey resembles (0.5) Donald, or
- (ii) Huey resembles (0.9) Louie, and Louie resembles (0.6) Donald.

Statement (i) contains a chain of relationships, and it seems reasonable to combine them with an **intersection** operation. With our definition, this corresponds to choosing the weakest membership value for the (transitive) Huey-Donald relationship, *i.e.*, 0.5. Similarly with statement (ii). Performing the operation along each chain in (i) and (ii), we get

- (iii) Huey resembles (0.5) Donald, or
- (iv) Huey resembles (0.6) Donald.

Both (iii) and (iv) seem equally valid, so it seems reasonable to apply the **union** operation. With our definition, this corresponds to choosing the strongest relation, *i.e.*, the maximum membership value. The final result is

(v) Huey resembles (0.6) Donald

The general rule when combining or *composing* fuzzy relations, is to pick the minimum fuzzy value in a 'series connection' and the maximum value in a 'parallel connection'. It is convenient to do this with an *inner product*.

The inner product is similar to an ordinary matrix (dot) product, except multiplication is replaced by **intersection** (\cap) summation by **union** (\cup). Suppose \mathbf{R} is an $m \times p$ and \mathbf{S} is a $p \times n$ matrix. Then the inner \cup, \cap product is an $m \times n$ matrix $\mathbf{T} = (t_{ij})$ whose ij -entry is obtained by combining the i th row of \mathbf{R} with the j th column of \mathbf{S} , such that

$$t_{ij} = (r_{i1} \cap s_{1j}) \cup (r_{i2} \cap s_{2j}) \cup \dots \cup (r_{ip} \cap s_{pj}) = \bigcup_{k=1}^p r_{ik} \cap s_{kj} \quad (7)$$

As a notation for the generalised inner product, we shall use $\mathbf{f.g}$, where \mathbf{f} and \mathbf{g} are any functions that take two arguments, in this case \cup and \cap . With our definitions of the set operations, composition reduces to what is called *max-min composition* in the literature (Zadeh in Zimmermann, 1991).

If \mathbf{R} is a relation from \mathbf{a} to \mathbf{b} and \mathbf{S} is a relation from \mathbf{b} to \mathbf{c} , then the composition of \mathbf{R} and \mathbf{S} is a relation from \mathbf{a} to \mathbf{c} (transitive law).

Example 8 (inner product) For the tables \mathbf{R}_1 and \mathbf{R}_2 above we get

$$\mathbf{R}_1 \cup . \cap \mathbf{R}_2 = \begin{bmatrix} 0.8 & 0.9 \end{bmatrix} \cup . \cap \begin{bmatrix} 0.5 \\ 0.6 \end{bmatrix} = \bigcup \begin{bmatrix} 0.5 & 0.6 \end{bmatrix} = 0.6$$

which agrees with the previous result.

The **max-min** composition is distributive with respect to **union**,

$$(\mathbf{R} \cup \mathbf{T}) \cup . \cap \mathbf{S} = (\mathbf{R} \cup . \cap \mathbf{S}) \cup (\mathbf{T} \cup . \cap \mathbf{S}),$$

but not with respect to **intersection**. Sometimes the **min** operation in *max-min composition* is substituted by $*$ for multiplication; then it is called *max-star composition*.

4. Fuzzy Logic

Logic started as the study of language in arguments and persuasion, and it may be used to judge the correctness of a chain of reasoning, in a mathematical proof for example. In *two-valued logic* a proposition is either *true* or *false*, but not both. The "truth" or "falsity" which is assigned to a statement is its *truth-value*. In *fuzzy logic* a proposition may be true or false or have an intermediate truth-value, such as *maybe true*. The sentence *the level is high* is an example of such a proposition in a fuzzy controller. It may be convenient to restrict the possible truth values to a discrete domain, say $\{0, 0.5, 1\}$ for *false*, *maybe true*, and *true*; in that case we are dealing with *multi-valued* logic. In practice a finer subdivision of the unit interval may be more appropriate.

4.1 Connectives

In daily conversation and mathematics, sentences are connected with the words *and*, *or*, *if-then* (or *implies*), and *if and only if*. These are called *connectives*. A sentence which is modified by the word "not" is called the *negation* of the original sentence. The word "and" is used to join two sentences to form the *conjunction* of the two sentences. Similarly a sentence formed by connecting two sentences with the word "or" is called the *disjunction* of the two sentences. From two sentences we may construct one of the form "If ... then ..."; this is called a *conditional* sentence. The sentence following "If" is the *antecedent*, and the sentence following "then" is the *consequent*. Other idioms which we shall regard as having the same meaning as "If p , then q " (where p and q are sentences) are " p implies q ", " p only if q ", " q if p ", etc. The words "if and only if" are used to obtain from two sentences a *biconditional* sentence.

By introducing letters and special symbols, the connective structure can be displayed in an effective manner. Our choice of symbols is as follows

\neg	for "not"
\wedge	for "and"
\vee	for "or"
\Rightarrow	for "if-then"
\Leftrightarrow	for "if and only if"

The next example illustrates how the symbolic forms can provide a quick overview.

Example 9 (baseball) Consider the sentence,

If either the Pirates or the Cubs lose and the Giants win, then the Dodgers will be out of first place, and I will lose a bet.

It is a conditional, so it may be symbolised in the form $r \Rightarrow s$. The antecedent is composed from the three sentences p ("The Pirates lose"), c ("The Cubs lose"), and g ("The Giants win"). The consequent is the conjunction of d ("The Dodgers will be out of first place") and b ("I will lose a bet"). The original sentence may be symbolised by $((p \vee c) \wedge g) \Rightarrow (d \wedge b)$.

The possible truth-values of a statement can be summarised in a *truth-table*. Take for example the truth-table for the two-valued proposition $p \vee q$. The usual form (below, left) lists all possible combinations of truth-values, i.e., the Cartesian product, of the arguments p and q in the two leftmost columns. Alternatively the truth-table can be rearranged into a two-dimensional array, a so-called Cayley table (below, right).

p	q	$p \vee q$	\vee	0	1	q
0	0	0	0	0	1	
0	1	1	1	1	1	
1	0	1		1	1	
1	1	1	p			

The vertical axis carries the possible values of the first argument p , and the horizontal axis

the possible values of the second argument q . At the intersection of row i and column j is the truth value of the expression $p_i \vee q_j$. The truth-values on the axes of the Cayley table can be omitted since, in the two-valued case, these are always 0 and 1, and in that order. Truth-tables for binary connectives are thus given by two-by-two matrices, where it is understood that the first argument is associated with the vertical axis and the second with the horizontal axis. A total of 16 such two-by-two tables can be constructed, and each has been associated with a connective.

It is possible to evaluate, in principle at least, a logic statement by an exhaustive test of all combinations of truth-values of the variables, cf. the so-called *array based logic* (Franksen, 1979). The next example illustrates an application of array logic.

Example 10 (array logic) *In the baseball example, we had $((p \vee c) \wedge g) \Rightarrow (d \wedge b)$. The sentence contains five variables, and each variable can take only two truth-values. This implies $2^5 = 32$ possible combinations. Only 23 are legal, however, in the sense that the sentence is valid (true) for these combinations, and $32 - 23 = 9$ cases are illegal, that is, the sentence is false for those particular combinations. Assuming that we are interested only in the legal combinations for which I win the bet ($b = 0$), then the following table results*

p	c	g	d	b
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
1	0	0	0	0
1	0	0	1	0
1	1	0	0	0
1	1	0	1	0

There are thus 10 winning outcomes out of 32 possible.

We can make similar truth-tables in fuzzy logic. If we for example start out by defining negation and disjunction, then we can derive other truth-tables from that. Let us assume that *negation* is defined as the set theoretic complement, i.e. $\text{not } p \equiv 1 - p$, and that *disjunction* is equivalent to set theoretic union, i.e., $p \vee q \equiv p \mathbf{max} q$. Then we can find

truth-tables for *or*, *nor*, *nand*, and *and*

<p>Or $p \vee q$</p> <table border="1" style="margin: auto; border-collapse: collapse;"> <tr><td>0</td><td>0.5</td><td>1</td></tr> <tr><td>0.5</td><td>0.5</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	0	0.5	1	0.5	0.5	1	1	1	1	<p>Nor $\neg(p \vee q)$</p> <table border="1" style="margin: auto; border-collapse: collapse;"> <tr><td>1</td><td>0.5</td><td>0</td></tr> <tr><td>0.5</td><td>0.5</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table>	1	0.5	0	0.5	0.5	0	0	0	0
0	0.5	1																	
0.5	0.5	1																	
1	1	1																	
1	0.5	0																	
0.5	0.5	0																	
0	0	0																	
<p>Nand $(\neg p) \vee (\neg q)$</p> <table border="1" style="margin: auto; border-collapse: collapse;"> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0.5</td><td>0.5</td></tr> <tr><td>1</td><td>0.5</td><td>0</td></tr> </table>	1	1	1	1	0.5	0.5	1	0.5	0	<p>And $\neg((\neg p) \vee (\neg q))$</p> <table border="1" style="margin: auto; border-collapse: collapse;"> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0.5</td><td>0.5</td></tr> <tr><td>0</td><td>0.5</td><td>1</td></tr> </table>	0	0	0	0	0.5	0.5	0	0.5	1
1	1	1																	
1	0.5	0.5																	
1	0.5	0																	
0	0	0																	
0	0.5	0.5																	
0	0.5	1																	

(8)

The two rightmost tables are negations of the left hand tables, and the bottom tables are reflections along the anti-diagonal (orthogonal to the main diagonal) of the top tables. It is comforting to realise that even though the truth-table for "and" is derived from the table for "or", the table for "and" can also be generated using the **min** operation, in agreement with the definition for set intersection.

The *implication* operator, however, has always troubled the fuzzy theoretic community. If we define it in the usual way, i.e., $p \Rightarrow q \equiv \neg p \vee q$, then we get a truth-table which is counter-intuitive and unsuitable, because several logical laws fail to hold.

Many researchers have tried to come up with other definitions; Kiszka, Kochanska & Sliwinska (1985) list 72 alternatives to choose from. One other choice is the so-called *Gödel implication* which is better in the sense that more "good old" (read: two-valued) logical relationships become valid (Jantzen, 1995). Three examples are $(p \wedge q) \Rightarrow p$ (simplification), $[p \wedge (p \Rightarrow q)] \Rightarrow q$ (modus ponens), and $[(p \Rightarrow q) \wedge (q \Rightarrow r)] \Rightarrow (p \Rightarrow r)$ (hypothetical syllogism). Gödel implication can be written

$$p \Rightarrow q \equiv (p \leq q) \vee q \tag{9}$$

The truth-table for equivalence (\Leftrightarrow) is determined from implication and conjunction, once it is agreed that $p \Leftrightarrow q$ is the same as $(p \Rightarrow q) \wedge (q \Rightarrow p)$.

<p>Implication $(p \leq q) \vee q$</p> <table border="1" style="margin: auto; border-collapse: collapse;"> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0.5</td><td>1</td></tr> </table>	1	1	1	0	1	1	0	0.5	1	<p>Equivalence $(p \Rightarrow q) \wedge (q \Rightarrow p)$</p> <table border="1" style="margin: auto; border-collapse: collapse;"> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0.5</td></tr> <tr><td>0</td><td>0.5</td><td>1</td></tr> </table>	1	0	0	0	1	0.5	0	0.5	1
1	1	1																	
0	1	1																	
0	0.5	1																	
1	0	0																	
0	1	0.5																	
0	0.5	1																	

(10)

Fuzzy array logic can be applied to theorem proving, as the next example will show.

Example 11 (fuzzy modus ponens) *It is possible to prove a law by an exhaustive search of all combinations of truth-values of the variables in fuzzy logic, provided the domain of truth values is discrete and limited. Take for example modus ponens*

$$[p \wedge (p \Rightarrow q)] \Rightarrow q \tag{11}$$

The sentence contains two variables, and let us assume that each variable can take, say,

three truth-values. This implies $3^2 = 9$ possible combinations,

p	q	$p \Rightarrow q$	$[p \wedge (p \Rightarrow q)]$	$[p \wedge (p \Rightarrow q)] \Rightarrow q$
0	0	1	0	1
0	0.5	1	0	1
0	1	1	0	1
0.5	0	0	0	1
0.5	0.5	1	0.5	1
0.5	1	1	0.5	1
1	0	0	0	1
1	0.5	0.5	0.5	1
1	1	1	1	1

Since the right column is all ones, the modus ponens (11) is valid, even for fuzzy logic. The scope of the validity is limited to the chosen truth domain $(0, 0.5, 1)$; this could be extended, however, and the test performed again in case a higher resolution is required.

Example 12 (fuzzy baseball) We will modify the baseball example (example 9) to see what difference fuzzy logic makes. The sentence contains five variables, but in the fuzzy case each variable can take, say, three truth-values. This implies $3^5 = 243$ possible combinations; 148 of these are legal in the sense that the sentence is true (truth-value 1) for these combinations. There are other cases where there is a 0.5 possibility of winning the bet depending on the possibilities of wins and losses of the Dodgers, etc. If we are interested again in the combinations for which there is some possibility that I win the bet, i.e., $b \in \{0, 0.5\}$ then there are 88 possible combinations. Instead of listing all of them, we will just show one for illustration,

$$(p, c, g, d, b) = [0 \quad 0.5 \quad 0.5 \quad 1 \quad 0.5] .$$

The example shows that fuzzy logic provides more solutions and it requires more computational effort than in the case of two-valued logic. This is the price to pay for having intermediate truth-values describe uncertainty.

Originally, Zadeh interpreted a truth-value in fuzzy logic, for instance *Very true*, as a fuzzy set (Zadeh, 1988). Thus Zadeh based fuzzy (linguistic) logic on treating *Truth* as a linguistic variable that takes words or sentences as values rather than numbers (Zadeh, 1975). Please be aware that our approach differs, being built on scalar truth-values rather than vector truth-values.

4.2 Implication

The rule *If the level is low, then open V_1* is called an *implication*, because the value of *level* implies the value of V_1 in the controller. It is uncommon, however, to use the Gödel implication (9) in fuzzy controllers. Another implication, called *Mamdani implication*, is often used.

Definition 2 (Mamdani implication) (Mamdani, 1977) Let \mathbf{a} and \mathbf{b} be two fuzzy sets,

not necessarily on the same universe. The Mamdani implication is defined

$$\mathbf{a} \Rightarrow \mathbf{b} \equiv \mathbf{a} \circ .\mathbf{min} \mathbf{b} \quad (12)$$

where $\circ .\mathbf{min}$ is the outer product, applying **min** to each element of the cartesian product of **a** and **b**.

Let **a** be represented by a column vector and **b** by a row vector, then their *outer min product* may be found as a 'multiplication table',

$$\begin{array}{c} \circ.\mathbf{min} \\ \begin{array}{|c|} \hline a_1 \\ \hline \dots \\ \hline a_n \\ \hline \end{array} \end{array} \begin{array}{|c|c|c|c|} \hline b_1 & b_2 & \dots & b_m \\ \hline a_1 \wedge b_1 & a_1 \wedge b_2 & \dots & a_1 \wedge b_m \\ \hline \dots & \dots & \dots & \dots \\ \hline a_n \wedge b_1 & a_n \wedge b_2 & \dots & a_n \wedge b_m \\ \hline \end{array} \quad (13)$$

Example 13 (outer product) Take the implication If the level is low, then open V_1 , with **low** and **open** defined as,

$$\mathbf{low} = (1, 0.75, 0.5, 0.25, 0)$$

$$\mathbf{open} = (0, 0.5, 1)$$

The implication is then represented by the scheme

$$\begin{array}{c} \circ.\mathbf{min} \\ \begin{array}{|c|c|c|} \hline 0 & 0.5 & 1 \\ \hline \end{array} V_1 \end{array}$$

1	0	0.5	1
0.75	0	0.5	0.75
0.5	0	0.5	0.5
0.25	0	0.25	0.25
0	0	0	0

level

This is a very important way to construct an implication table from a rule.

The outer **min** product (Mamdani, 1977) as well as the outer product with **min** replaced by * for multiplication (Holmblad & Østergaard, 1982), is the basis for most fuzzy controllers; therefore the following chapters will use that. However, Zadeh and other researchers have proposed many other theoretical definitions (e.g., Zadeh, 1973; Wénstøp, 1980; Mizumoto, Fukami & Tanaka, 1979; Fukami, Mizumoto & Tanaka, 1980; see also the survey by Lee, 1990).

4.3 Inference

In order to draw conclusions from a rule base we need a mechanism that can produce an output from a collection of *if-then* rules. This is done using the *compositional rule of inference* (CROI). The verb *to infer* means to conclude from evidence, deduce, or to have as a logical consequence – do not confuse 'inference' with 'interference'. To understand the concept, it is useful to think of a function $y = f(x)$, where f is a given function, x is the independent variable, and y the result; a value y_0 is inferred from x_0 given f .

The famous rule of inference *modus ponens*,

$$a \wedge (a \Rightarrow b) \Rightarrow b \quad (14)$$

can be stated as follows: If it is known that a statement $a \Rightarrow b$ is true, and also that a is true, then we can infer that b is true. Fuzzy logic generalises this into *generalised modus ponens* (GMP):

$$a' \wedge (a \Rightarrow b) \Rightarrow b' \quad (15)$$

Notice that fuzzy logic allows a' and b' to be slightly different in some sense from a and b , for example after applying modifiers. The GMP is closely related to forward chaining, i.e., reasoning in a forward direction in a rule base containing chains of rules. This is particularly useful in the fuzzy controller. The GMP inference is based on the compositional rule of inference.

Example 14 (GMP) Given the relation $R = \text{low} \circ \text{min open}$ from the previous example, and an input vector

$$\text{level} = (0.75, 1, 0.75, 0.5, 0.25),$$

then

$$\mathbf{v}_1 = \text{level} \vee . \wedge \mathbf{R} \quad (16)$$

$$= \begin{bmatrix} 0.75 & 1 & 0.75 & 0.5 & 0.25 \end{bmatrix} \vee . \wedge \begin{array}{|c|c|c|} \hline 0 & 0.5 & 1 \\ \hline 0 & 0.5 & 0.75 \\ \hline 0 & 0.5 & 0.5 \\ \hline 0 & 0.25 & 0.25 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \quad (17)$$

$$= \begin{bmatrix} 0 & 0.5 & 0.75 \end{bmatrix} \quad (18)$$

Obviously, the input **level** is a fuzzy set representing a level somewhat higher than **low**. The result after inference is a vector \mathbf{v}_1 slightly less than "open". Incidentally, if we try putting **level** = **low**, we would expect to get a vector \mathbf{v}_1 equal to **open** after composition with \mathbf{R} . This is indeed so, but the confirmation is left as an exercise for the student.

4.4 Several Rules

A rule base usually contains several rules, how do we combine them? Returning to the simple rule base

$$\text{If the level is low then open } V_1 \quad (19)$$

$$\text{If the level is high then close } V_1$$

We implicitly assume a logical **or** between rules, such that the rule base is read as $R_1 \vee R_2$, where R_1 is the first, and R_2 the second rule. The rules are equivalent to implication matrices \mathbf{R}_1 and \mathbf{R}_2 , therefore the total rule base is the logical **or** of the two tables, item by item. In general terms, we have

$$\mathbf{R} = \bigvee \mathbf{R}_i$$

Inference can then be performed on \mathbf{R} .

In case there are n inputs, that is, if each *if*-side contains n variables, the relation matrix \mathbf{R} generalises to an $n + 1$ dimensional array. Let \mathbf{e}_i ($i = 1, \dots, n$) be the inputs, then inference is carried out by a generalised composition,

$$\mathbf{u} = (\mathbf{e}_1 \times \mathbf{e}_2 \times \dots \times \mathbf{e}_n) \vee . \wedge \mathbf{R}$$

Inference is still the usual composition operation; we just have to keep track of the dimensions.

5. Summary

We have achieved a method of representing and executing a rule *If the level is low then open V_1* in a computer program. In summary:

1. Define fuzzy sets **low** and **open** corresponding to a low level and an open valve; these can be defined on different universes.
2. Represent the implication as a relation \mathbf{R} by means of the outer product, $\mathbf{R} = \mathbf{low} \circ \mathbf{min} \mathbf{open}$. The result is a matrix.
3. Perform the inference with an actual measurement. In the most general case this measurement is a fuzzy set, say, the vector **level**. The control action \mathbf{v}_1 is obtained by means of the compositional rule of inference, $\mathbf{v}_1 = \mathbf{level} \vee . \wedge \mathbf{R}$.

Fuzzy controllers are implemented in a more specialised way, but they were originally developed from the concepts and definitions presented above, especially inference and implication.

References

- Franksen, O. I. (1978). Group representation of finite polyvalent logic, in A. Niemi (ed.), *Proceedings 7th Triennial World Congress*, International federation of automatic control, IFAC, Pergamon, Helsinki.
- Fukami, S., Mizumoto, M. and Tanaka, K. (1980). Some considerations of fuzzy conditional inference, *Fuzzy Sets and Systems* **4**: 243–273.
- Holmblad, L. P. and Østergaard, J.-J. (1982). Control of a cement kiln by fuzzy logic, in Gupta and Sanchez (eds), *Fuzzy Information and Decision Processes*, North-Holland, Amsterdam, pp. 389–399. (Reprint in: FLS Review No 67, FLS Automation A/S, Høffdingsvej 77, DK-2500 Valby, Copenhagen, Denmark).
- Jantzen, J. (1995). Array approach to fuzzy logic, *Fuzzy Sets and Systems* **70**: 359–370.
- Kaufmann, A. (1975). *Introduction to the theory of fuzzy sets*, Academic Press, New York.
- Kiszka, J. B., Kochanska, M. E. and Sliwinska, D. S. (1985). The influence of some fuzzy implication operators on the accuracy of a fuzzy model, *Fuzzy Sets and Systems* **15**: (Part1) 111–128; (Part 2) 223 – 240.
- Lee, C. C. (1990). Fuzzy logic in control systems: Fuzzy logic controller, *IEEE Trans. Systems, Man & Cybernetics* **20**(2): 404–435.

- Mamdani, E. H. (1977). Application of fuzzy logic to approximate reasoning using linguistic synthesis, *IEEE Transactions on Computers* **C-26**(12): 1182–1191.
- Mizumoto, M., Fukami, S. and Tanaka, K. (1979). Some methods of fuzzy reasoning, in Gupta, Ragade and Yager (eds), *Advances in Fuzzy Set Theory Applications*, North-Holland, New York.
- Singh, M. G. (ed.) (1987). *Systems and Control Encyclopedia: Theory, Technology, Applications*, Pergamon, Oxford.
- Singh, M. G. (ed.) (1990). *Systems and Control Encyclopedia: Theory, Technology, Applications*, Vol. Supplementary volume 1, Pergamon, Oxford.
- Singh, M. G. (ed.) (1992). *Systems and Control Encyclopedia: Theory, Technology, Applications*, Vol. Supplementary volume 2, Pergamon, Oxford.
- Wenstøp, F. (1980). Quantitative analysis with linguistic values, *Fuzzy Sets and Systems* **4**(2): 99–115.
- Yager, R. R., Ovchinnikov, R., Tong, M. and Nguyen, H. T. (eds) (1987). *Fuzzy sets and applications - selected papers by L. A. Zadeh*, Wiley and Sons, New York etc.
- Zadeh, L. A. (1965). Fuzzy sets, *Inf. and control* **8**: 338–353.
- Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Trans. Systems, Man & Cybernetics* **1**: 28–44.
- Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning, *Information Sciences* **8**: 43–80.
- Zadeh, L. A. (1984). Making computers think like people, *IEEE Spectrum* pp. 26–32.
- Zadeh, L. A. (1988). Fuzzy logic, *IEEE Computer* **21**(4): 83–93.
- Zimmermann, H.-J. (1993). *Fuzzy set theory - and its applications*, second edn, Kluwer, Boston. (1. ed. 1991).