Soft Computing

Fuzzy logic is part of soft computing

Contents

- Review of classical logic and reasoning systems
- Fuzzy sets
- Fuzzy logic
- Fuzzy logic systems applications
- Fuzzy Logic Minimization and Synthesis
- Practical Examples
- Approaches to fuzzy logic decomposition
- Our approach to decomposition
- Combining methods and future research

Traditional Logic

- **One of the main aims of logic is to provide rules which can be employed to determine whether a particular argument is correct or not.**
- **The language of logic is based on mathematics and the reasoning process is precise and unambiguous.**

Logical arguments

- Any logical argument consists of statements.
- A statement is a sentence which unambiguously either holds true or holds false.
  - **Example:** Today is Sunday

Predicates

- **Example:** Seven is an even number
  - This example can be written in a mathematical form as follows:
    - $7 \in \{x|\ x \text{ is an even number}\}$
  - or in a more concise way:
    - $7 \in \{x|P(x)\}$
  - where | is read as such that and P(x) stands for `x has property P' and it is known as the predicate.
  - Note that a predicate is not a statement until some particular *x-value* is specified.
  - Once a x value is specified then the predicate becomes a statement whose truth or falsity can be worked out.

For All Quantifier

- For all x and y, x^2-y^2 is the same as (x+y)*(x-y)
  - – This example can be written in a mathematical form as well:
    - $\forall$ x,y ((x,y$\in$R)$\wedge$ (x^2-y^2)=(x+y)*(x-y))
- where the$\forall$ is interpreted as 'for all', $\wedge$ is the logical operator AND, and R represents what is termed as the universe of discourse.

Universe of Discourse

- Using the universe of discourse one assures that a statement is evaluated for relevant values.
  - – The above predicate is then true only for *real numbers.*
- Similarly for the first example the universe of discourse is most likely to be the set of natural numbers rather than buildings, rivers, or anything else.
  - – Hence, using the concept of the universe of discourse any logical paradoxes can not arise.

Existential Quantifier

- Another type of quantifier is the existential quantifier (∃).
- The existential quantifier is interpreted as 'there exists' or 'for some' and describes a statement as being true for at least one element of the set.
- For example, $(\exists x)$ $((river(x) \wedge name(Amazon))$

Connectives and their truth tables

- A number of connectives exist.
  - Their sole purpose is to allow us to join together predicates or statements in order to form more complicated ones.
- Such connectives are NOT (~), AND ($\wedge$), OR ($\vee$).
  - Strictly speaking NOT is not a connective since it only applies to a single predicate or statement.
- In traditional logic the main tools of reasoning are *tautologies*, such as the *modus ponens* $(A\wedge(A\Rightarrow B))\Rightarrow B$ ($\Rightarrow$ means implies).

Truth Tables

		And	Or	Not
A	B	AÙB	AÚB	~ A
True	True	True	True	False
True	False	False	True	False
False	True	False	True	True
False	False	False	False	True

Fuzzy Logic

- **The concept of fuzzy logic was introduced by L.A Zadeh in a 1965 paper.**
- **Whereas Aristotelian concepts have been useful and applicable for many years they present us with certain problems**
  - Cannot express ambiguity
  - Lack of quantifiers
  - Cannot handle exceptions

Traditional Logic Problems

- **Cannot express ambiguity:**
  - Consider the predicate `X is tall'.
  - Providing a specific person we can turn the predicate into a statement.
  - But what is the exact meaning of the word `tall'?
  - What is `tall' to some people is not tall to others.
- **Lack of quantifiers:**
  - Another problem is the lack of being able to express statements such as `Most of the goals came in the first half'.
  - The `most' quantifier cannot be expressed in terms of the universal and/or existential quantifiers.

Traditional Logic Problems

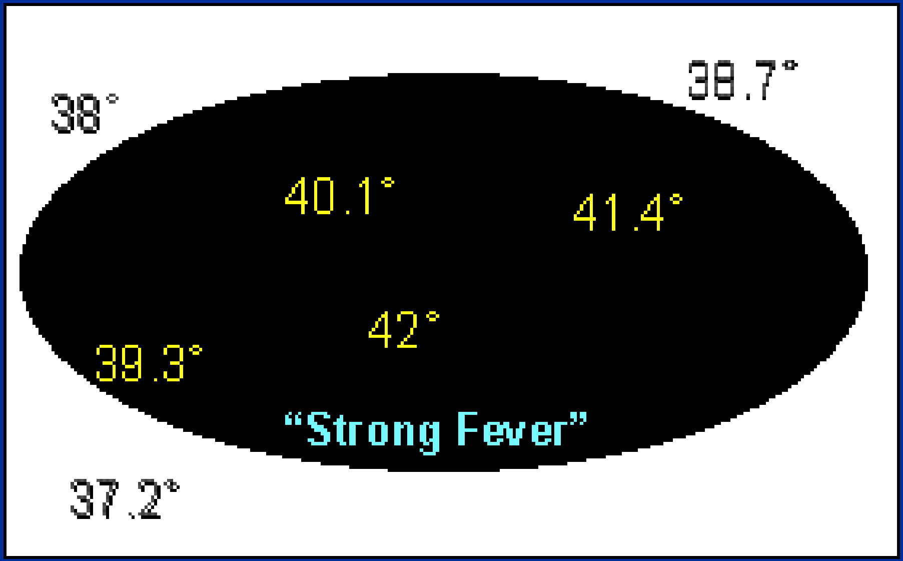
- **Cannot handle exceptions:**
  - Another limitation of traditional predicate logic is expressing things that are sometimes, but not always true.

Fuzzy Sets

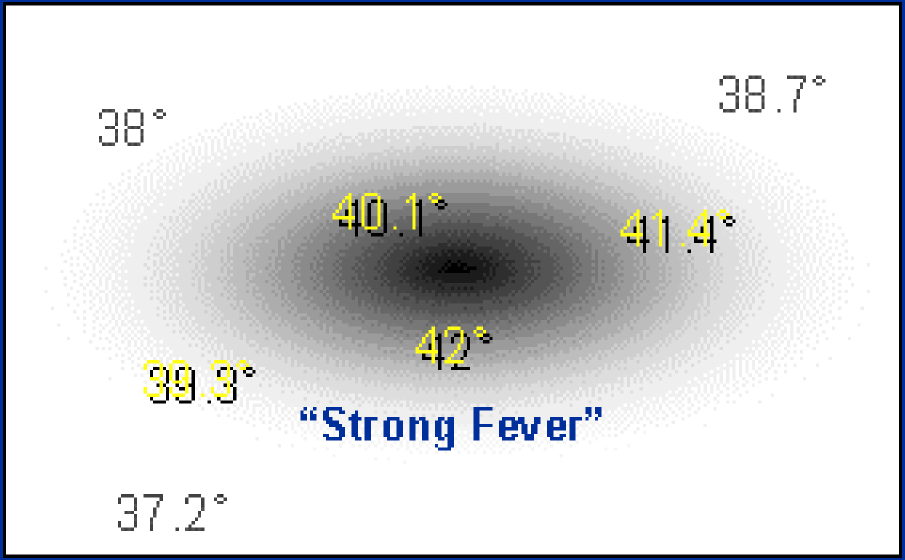
Fuzzy Sets

- **Fuzzy logic is based upon the notion of fuzzy sets.**
  - **Recall from the previous section that an item is an element of a set or not.**
  - **With traditional sets the boundaries are clear cut.**
  - **With fuzzy sets partial membership is allowed.**
  - **Fuzzy logic involves 3 primary processes :**
    - • Fuzzification
    - • Rule evaluation
    - • Defuzzification
  - **With fuzzy logic the *generalised modus ponens* is employed which allows A and B to be characterised by fuzzy sets.**

Fuzzy Set Theory



Fuzzy Set Theory:



Fuzzy Sets

- Definition
- Operations
- Identities
- Transformations

TRADITIONAL vs. FUZZY SETS

- **Traditional sets, influenced from the Aristotelian view of two-valued logic, have only two possible truth values, namely TRUE or FALSE, 1 or 0, yes or no etc.**
- **Something either belongs to a particular set or does not.**
- **The characteristic function or alternatively referred to as the discrimination function is defined below in terms of a functional mapping:**

TRADITIONAL vs. FUZZY SETS

- In fuzzy sets, something may belong partially to a set.
- Therefore it might be very true or somewhat true, 0.2 or 0.9 in numerical terms.
- The membership function using fuzzy sets defined in terms of a functional mapping is as shown below:

TRADITIONAL vs. FUZZY SETS

- Fuzzy logic allows you to violate the *laws of noncontradiction* since an element can be a member of more than one set.
- More set operations are available
- The excluded middle is not applicable, i.e., the intersection of a set with its complement does not necessarily result to an empty set.
- Rule based systems using fuzzy logic in some cases might *increase the amount of computation* required in comparison with systems using classical binary logic.

TRADITIONAL vs. FUZZY SETS

- If fuzzy membership grades are restricted to {0,1} then Boolean sets are recovered.
- For instance, consider the Set Union operator which states that the truth value of two arguments x and y is their maximum:
  - truth(x or y) = max(truth(x), truth(y)).

TRADITIONAL vs. FUZZY SETS

- **If truth grades are either 0 or 1 then following table is found:**
  - **x y truth**
  - **0 0 0**
  - **0 1 1**
  - **1 0 1**
  - **1 1 1**
    - **which is the same truth table as in the Boolean logic.**
- **So, every crisp set is fuzzy, but *not conversely*.**

Definition of Fuzzy Set

- A ***fuzzy set***, defined as A, is a subset of a universe of discourse U, where A is characterized by a membership function uA(x).

- The membership function uA(x) is associated with each point in U and is the "grade of membership" in A.

- The membership function uA(x) is assumed to range in the interval **[0,1],** with value of 0 corresponding to the non-membership, and 1 corresponding to the full membership.

- The ordered pairs form the set **{(u,uA(x))}** to represent the **fuzzy set member** and the **grade of membership.**

Operations

- The fuzzy set operations are defined as follows.
  - Intersection operation of two fuzzy sets uses the symbols: ∩, *, ∧, AND, or min.
  - Union operation of two fuzzy sets uses the symbols: ∪, ∨, +, OR, or max.

- **Equality** of two sets is defined as A = B ↔ **u a(x) = u b(x)** for all x ∈ X.

- **Containment** of two sets is defined as A subset B,

  A ⊆ B ↔ u a(x) ≤ u b(x) for all x ∈ X.

- **Complement** of a set A is defined as A’, where u a’(x) = 1 – u a(x) for all x ∈ X.

- **Intersection** of two sets is defined as A ∩ B

  where u a ∩ b (x) = min{(u a(x),u b(x))} for all x ∈ X.

  Where C ⊆ A, C ⊆ B then C ⊆ A ∩ B.

- **Union** of two sets is defined as A ∪ B where u a ∪ b(x) = max{(u a(x), u b (x))} for all x ∈ X where D ⊇ A, D ⊇ B then D ⊇ A ∪ B.

Operations

- An example of fuzzy operations: X = { 1, 2, 3, 4, 5} and fuzzy sets A and B.
- A = {(3,0.8), (5,1), (2,0.6)} and

  B = {(3,0.7), (4,1), (2,0.5)} **then**
- $A \cap B$ = {(3, 0.7), (2, 0.5)}
- $A \cup B$ = {(3, 0.8), (4, 1), (5, 1), (2, 0.6)}
- A’ = {(1, 1), (2, 0.4), (3, 0.2), (4, 1), (5,0)}

Recap AI and Expert Systems.

Fuzzy logic in this framework

Overview of AI

- **The realization by the Artificial Intelligence community during the 1960's of the weakness of general purpose problem solvers led to the development of expert systems.**
- **Expert systems held the greatest promise for capturing intelligence and have received more attention than any other sub-discipline of Artificial Intelligence.**
  - **The term knowledge-based systems is used interchangeably to avoid the mis-understandings and mis-interpretations of the word 'expert'.**

Expert Systems

- Irrespective of the adjective, each such system is designed to operate in one of a variety of narrow areas.
- The design involves attempts to model and codify the knowledge of human experts.
- One might wonder what makes expert systems different from conventional ones. One might remark that in some sense, any computer program is expert at something.
    - A payroll program incorporates knowledge about accountancy, but it is not included in the expert class.
        - The differences originate from the type of programming language employed.
        - Additionally, expert systems can reason using incomplete data and can generate explanations and justifications, even during execution of their actions.

Components of ES

– ***Knowledge-base module:***

  - this is the essential component of any system.
  - It contains a representation in a variety of forms of knowledge elicited from a human expert

– ***Inference engine module:***

  - the inference engine utilizes the contents of the knowledge base in conjunction with the **data** given by the user in order to achieve a conclusion.

Components of ES

- ***Working memory module:***
  - this is where the user's responses and the system's conclusions for each session are temporarily stored.
- ***Explanation module:***
  - this is an important aspect of an expert system.
  - Answers from a computer are rarely accepted unquestioningly.
  - This is particularly true for responses from an expert system.
  - Any system must be able to explain how it reached its conclusions and why it has not reached a particular result.

Components of ES

- ***Justification module:***
  - using this module the system provides the user with justification(s) of why some piece of information is required.
- ***User interface module:***
  - the user of an expert system asks questions, enters data, examines the reasoning etc.
  - The input-output interface, using menus or restricted language, enables the user to communicate with the system in a simple and uncomplicated way.

Methods of inference

- Much of the power of an expert system comes from the knowledge embedded in it.
- In addition, the way the system infers conclusions is of equal importance.
  - Most expert systems apply forward and/or backward chaining.
    - The mode of chaining describes the way in which the production rules are activated.
    - With forward chaining the user of the expert system asks what conclusions can be made when this data is true.
    - The expert system might or might not ask for further data.
    - With backward chaining the user of the expert system asks what conclusions can be made.
    - The expert system will ask the user for data.

Methods of inference

- To illustrate the two modes consider the following situation.
- As you are driving you notice that behind you is a police car with its lights and siren on.
- So the data is `light is on' and `siren is on'.
  - The expert system will come to a conclusion such as `stop the car' and `someone else to stop the car'.
  - Obviously, the system can not make a hard decision and asks for more data.
    - You suddenly realise that the policeman in the car is waving at you.
    - This third piece of data `policeman is waving at me' suggests to the system that they want you to stop the car rather than someone else.

Methods of inference

- The previous scenario describes the forward chaining of your expert system which in this case happens to be your brain.
- Now, the system starts applying backward chaining.
  - There are numerous conclusions of why the police want you to stop.
  - For instance, 50 miles in a 30-mile zone, not-working brake light, stolen plate number, turning to a one-way street etc.
  - Therefore, your system starts collecting data to support any of the hypothesised reasons.
  - Since you just passed your MOT, know that this is your car, it is not a one-way street the system deduces that you were overspeeding.

Control Strategies

- **This refers to how the expert system comes to a conclusion, i.e., the mode of reasoning describes the way in which the system as a whole is organised.**
  - **For instance, the order of looking at the rules;**
  - **how to use *meta-rules* in order to check for outstanding queries, of a completed goal and the initiation of the evaluation of rules.**
  - **The order of looking at the rules usually is in *lexical order* viz. when scanning rules it will first look at rule 1, and then rule 2 etc.**
    - **When it searches, it inspects each rule to see if the left hand conditions are true.**

Control Strategies

- This is achieved by either reading the working memory or by asking questions or by generating further subgoals.
- In most cases the system continues to the next rule until all rules have inspected.
  - All rules that can execute are placed in a conflict set and one of the rules is selected.
  - The selected rule then executes. This is what is known as the *match, select and execute cycle.*

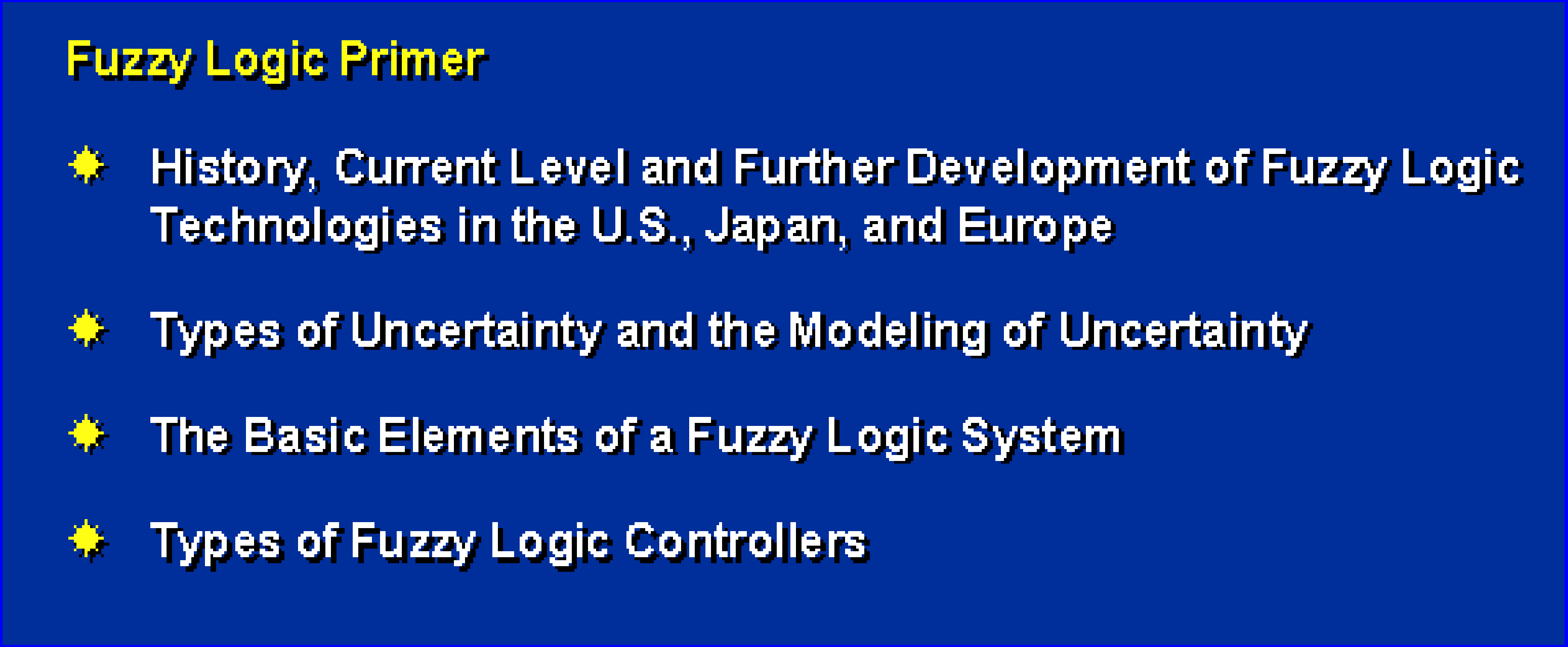
Fuzzy Logic Principles and Learning

- **Fuzzy Logic Principles**
  - Fuzzy control produces actions using a set of fuzzy rules based on fuzzy logic
  - This involves:
    - **fuzzifying:** mapping sensor readings into a set of fuzzy inputs
    - **fuzzy rule base:** a set of IF-THEN rules
    - **fuzzy inference:** maps fuzzy sets onto other fuzzy sets using membership fncts.
    - **defuzzifying:** mapping a set of fuzzy outputs onto a set of crisp output commands

Fuzzy Control

- Fuzzy logic allows for specifying behaviors as fuzzy rules
- Such behaviors can be smoothly blended together (e.g., Flakey robot)
- Fuzzy rules can be **learned**

Industrial Application of Fuzzy Logic Control



Fuzzy Logic Primer

- History, Current Level and Further Development of Fuzzy Logic Technologies in the U.S., Japan, and Europe
- Types of Uncertainty and the Modeling of Uncertainty
- The Basic Elements of a Fuzzy Logic System
- Types of Fuzzy Logic Controllers

History, State of the Art, and Future Development

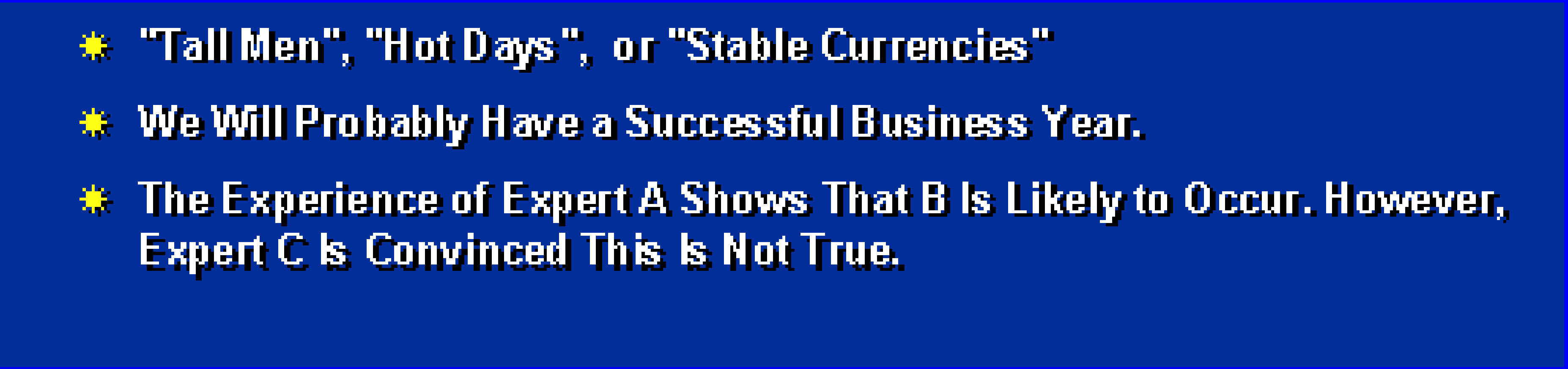
1965	Seminal Paper "Fuzzy Logic" by Prof. Lotfi Zadeh, Faculty in Electrical Engineering, U.C. Berkeley, Sets the Foundation of the "Fuzzy Set Theory"
1970	First Application of Fuzzy Logic in Control Engineering (Europe)
1975	Introduction of Fuzzy Logic in Japan
1980	Empirical Verification of Fuzzy Logic in Europe
1985	Broad Application of Fuzzy Logic in Japan
1990	Broad Application of Fuzzy Logic in Europe
1995	Broad Application of Fuzzy Logic in the U.S.
2000	Fuzzy Logic Becomes a Standard Technology and Is Also Applied in Data and Sensor Signal Analysis. Application of Fuzzy Logic in Business and Finance.

Uncertainty

Types of Uncertainty and the Modeling of Uncertainty

- **Stochastic Uncertainty:**
  - The Probability of Hitting the Target is 0.8

Lexical Uncertainty:



- "Tall Men", "Hot Days", or "Stable Currencies"
- We Will Probably Have a Successful Business Year.
- The Experience of Expert A Shows That B Is Likely to Occur. However, Expert C Is Convinced This Is Not True.

Methods of inference under uncertainty

- **This is very important to consider when using expert systems since sometimes data is uncertain (i.e., ambiguous, incomplete, noisy etc.).**
- **A number of theories have been devised to deal with uncertainty.**
  - **These include classical probability, Bayesian probability, Shannon theory, Dempster-Shafer theory among others.**
  - **A popular method of dealing with uncertainty uses *certainty factors***

Methods of inference under uncertainty

- The certainty factor indicates the net belief in the conclusion and premises of a rule based on some evidence.
- Certainty factors are hand-crafted by asking potential users questions such as *`How much do you believe that opening valve x will start a flooding'* and *`How much do you disbelieve that opening valve x will start a flooding'*.
- The degree of certainty is the difference between the two responses.

Production Rules

- **Assuming that the *knowledge-base module* contains knowledge represented in the the format of *production rules* the following sections introduce the following:**
  - the concept of a production rule
  - the concept of linguistic variables
  - the fuzzy inference concept
  - the concept of fuzzification and how to accomplish the crisp to fuzzy transformation
  - the concept of defuzzification and how to accomplish the fuzzy to crisp transformation

Knowledge presentation using production rules

- From a philosophical point the concept of knowledge is highly ambiguous and debatable
  - knowledge-base builders treat knowledge from a narrower point of view.
- This way the knowledge is easier to model and understand.
- It remains diverse including:
  - rules,
  - facts,
  - truths,
  - reasons,
  - defaults and
  - heuristics.
- The <u>knowledge engineer</u> needs some technique for capturing what is known about the application.

Knowledge presentation using production rules

- The technique should provide expressive adequacy and notational efficacy.
- Knowledge representation is very much under constant research.
- Several schemes have been suggested in the literature, namely:
  - semantic nets,
  - frames and
  - logic.
- Production rules have also been suggested and are the most popular way of representing knowledge.

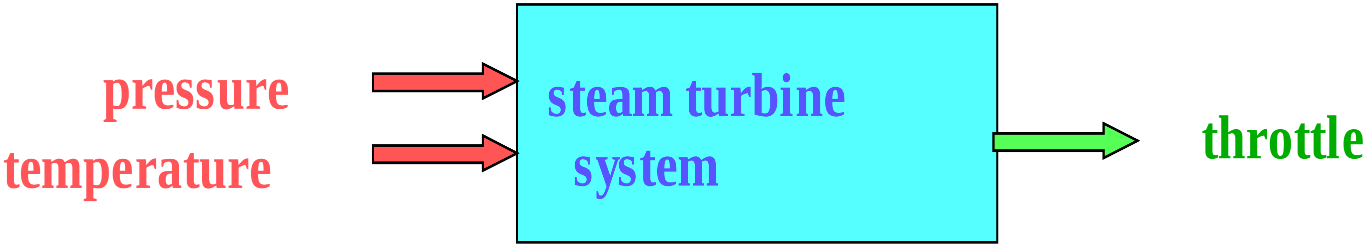
Knowledge presentation using production rules

- Production rules are small chunks of knowledge expressed in the form of *if..then* statements.
  - The left hand side (IF) represents the antecedent or conditional part.
  - The right hand side (THEN) represents the conclusion or action part.
  - A number of rules collectively define a *modularized know-how system.*
  - The principal use of production rules is in the encoding of empirical associations between incoming patterns of data and actions that the system should perform as a consequence.
  - The production rules are either expressed by an expert of the field, or derived using induction.

Fuzzy Logic Control

- Fuzzy controller design consist of turning **intuitions**, and any **other information** about how to control a system, into set of rules.
- These rules can then be applied to the system.
- If the rules **adequately control the system**, the design work is done.
- If the rules are inadequate, the way they fail provides information to change the rules.

Control a Plant



Rules presentation of ...

An Expert system	A fuzzy system
If temperature > 680 and pressure < 25 then throttle is 165	if temperature is hot and pressure is low then throttle is positive medium

Linguistic variables

Linguistic variables

- **Looking at the production rules of either a expert system or a fuzzy expert system one can not see any differences**
  - **except that the fuzzy system is employing linguistic descriptors rather than absolute numerical values.**
  - **However, both parts of fuzzy rules have associated `levels of belief' something lacking in traditional production rules.**
- **Secondly, with traditional production rules even when more than one rule applies <u>only one</u> executes.**
  - **With fuzzy rules <u>all applicable rules contribute</u> in calculating the resulting output.**
    - **All in all, fuzzy expert systems require fewer production rules since <u>fuzzy rules embody more information.</u>**

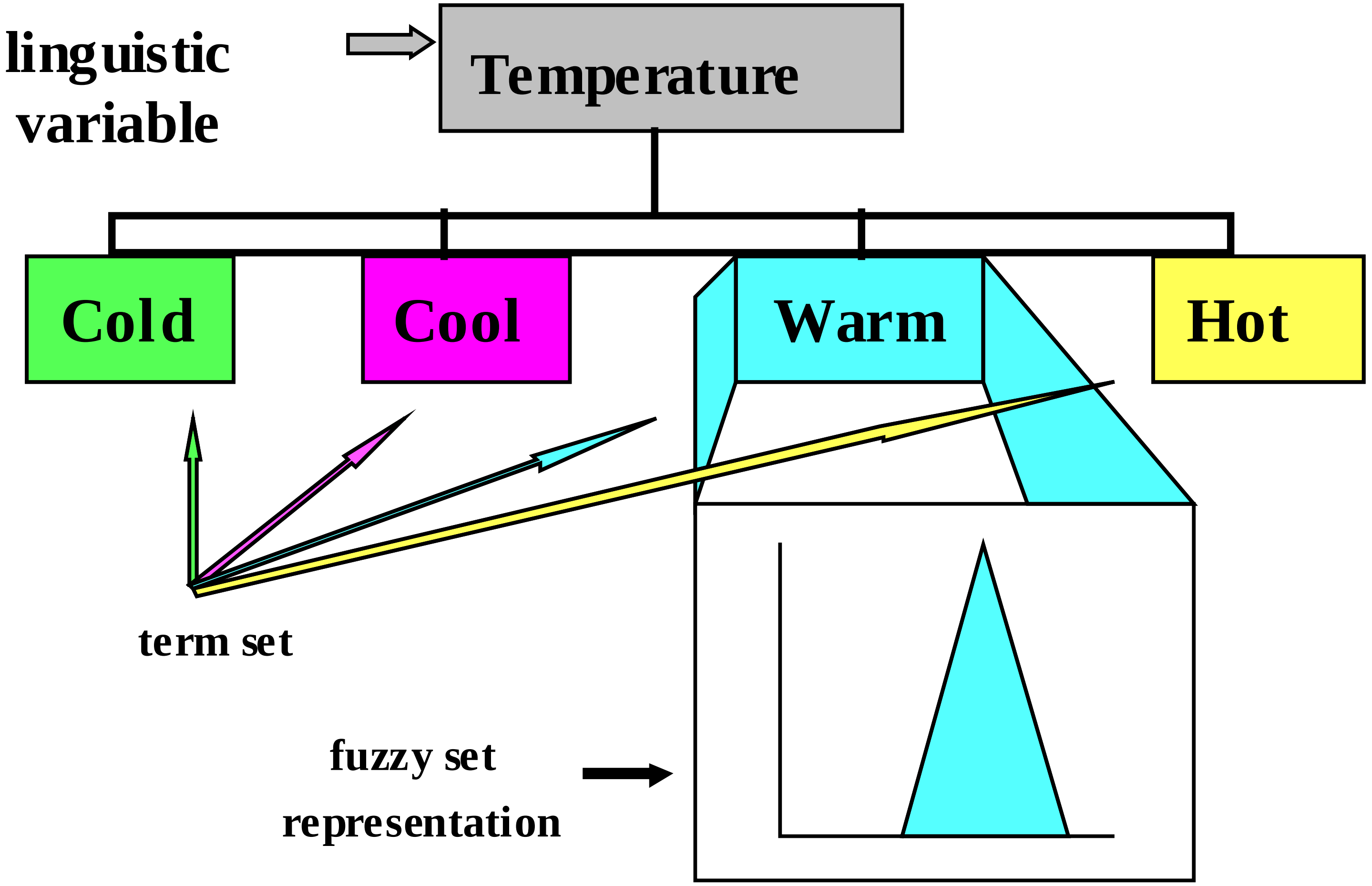
Linguistic variables

- A major reason behind using fuzzy logic is the use of linguistic expressions.
- A linguistic variable consists of:
  - the name of the variable (u),
  - the term set of the variable (T(u)),
  - its universe of discourse (U) in which the fuzzy sets are defined,
  - a syntactic rule for generating the names of values of u, and
  - a semantic rule for associating with each value its meaning.

Linguistic variables

- **For example:**
- **if u is temperature,**
- **then its term set T(temperature) could be:**
  - **T(temperature)={cold, cool, warm, hot} over a universe of discourse U=[0,300].**

Linguistic variables



linguistic variable

Temperature

Cold

Cool

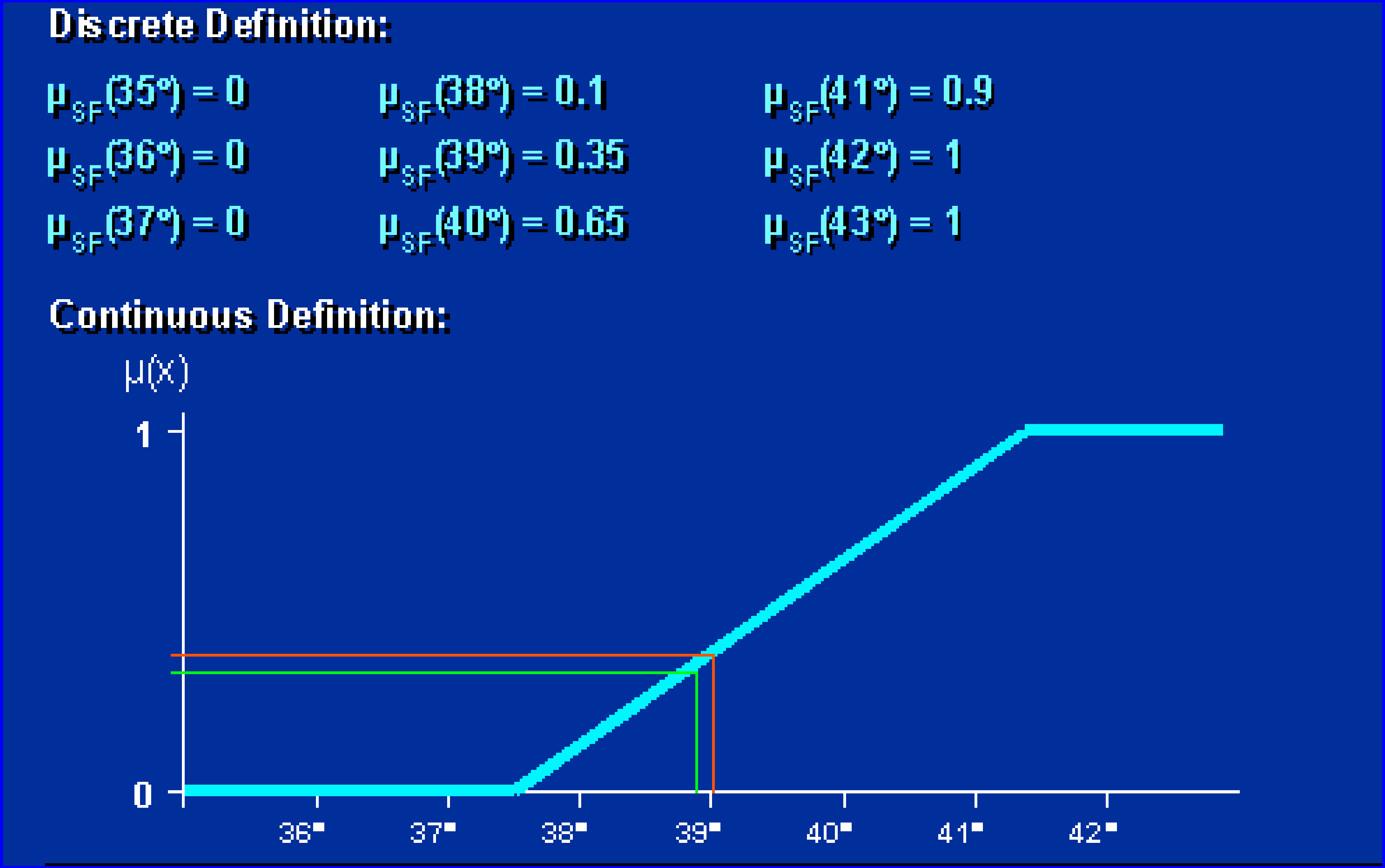
Warm

Hot

term set

fuzzy set representation

Fuzzy Set Definitions



Discrete Definition:

$\mu_{SF}(35°) = 0$

$\mu_{SF}(36°) = 0$

$\mu_{SF}(37°) = 0$

$\mu_{SF}(38°) = 0.1$

$\mu_{SF}(39°) = 0.35$

$\mu_{SF}(40°) = 0.65$

$\mu_{SF}(41°) = 0.9$

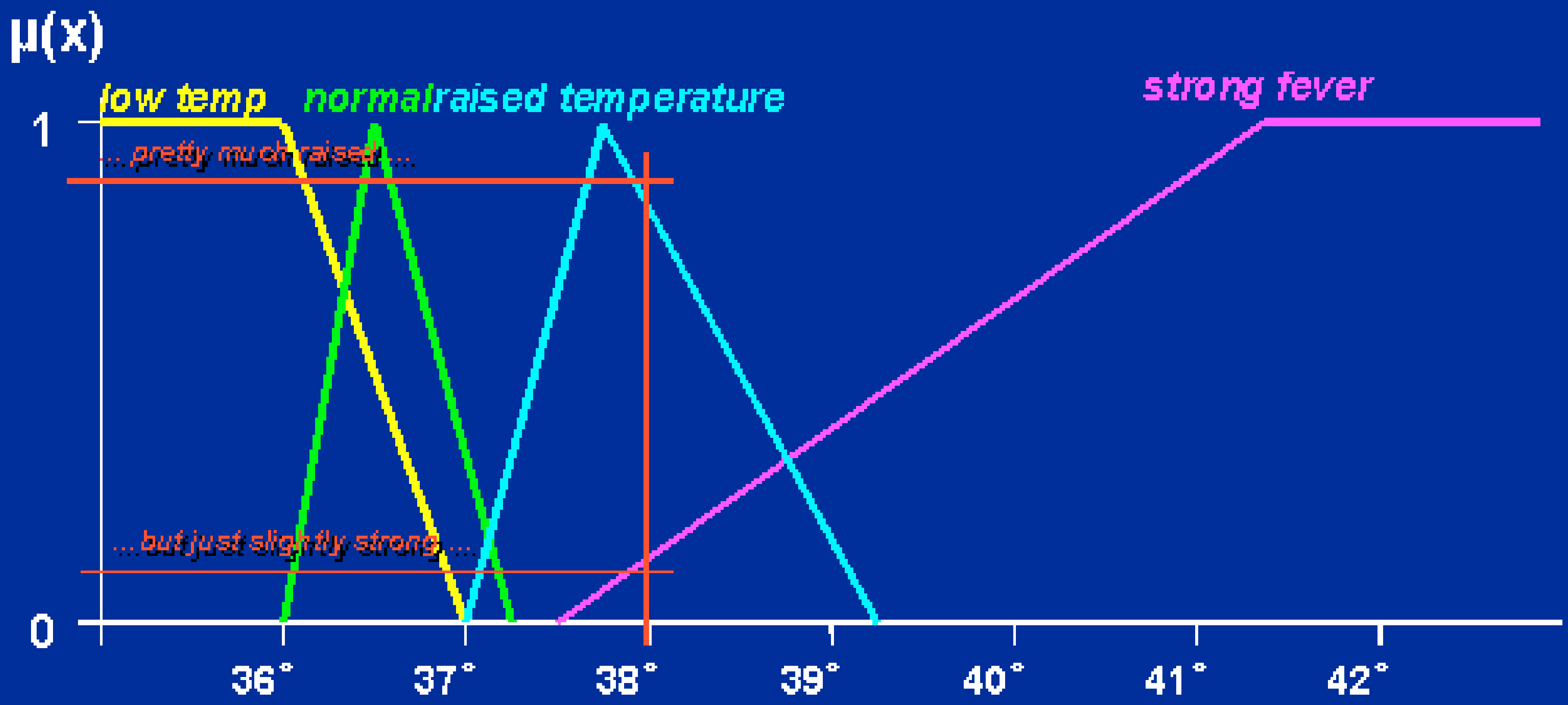
$\mu_{SF}(42°) = 1$

$\mu_{SF}(43°) = 1$

Continuous Definition:

Linguistic Variable

- …. Terms, Degree of Membership, Membership Function, Base Variable…..



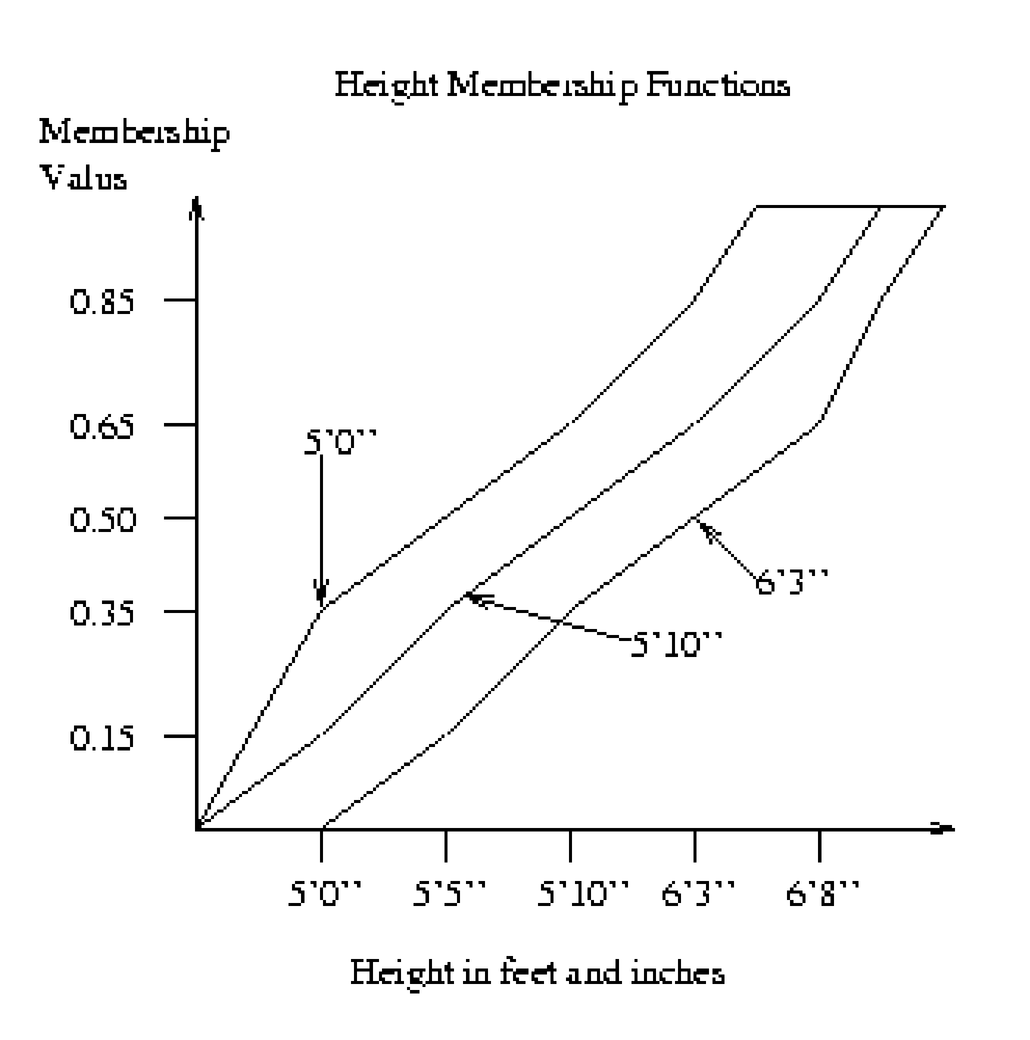
Fuzzy Logic and Functions

Fuzzy Logic

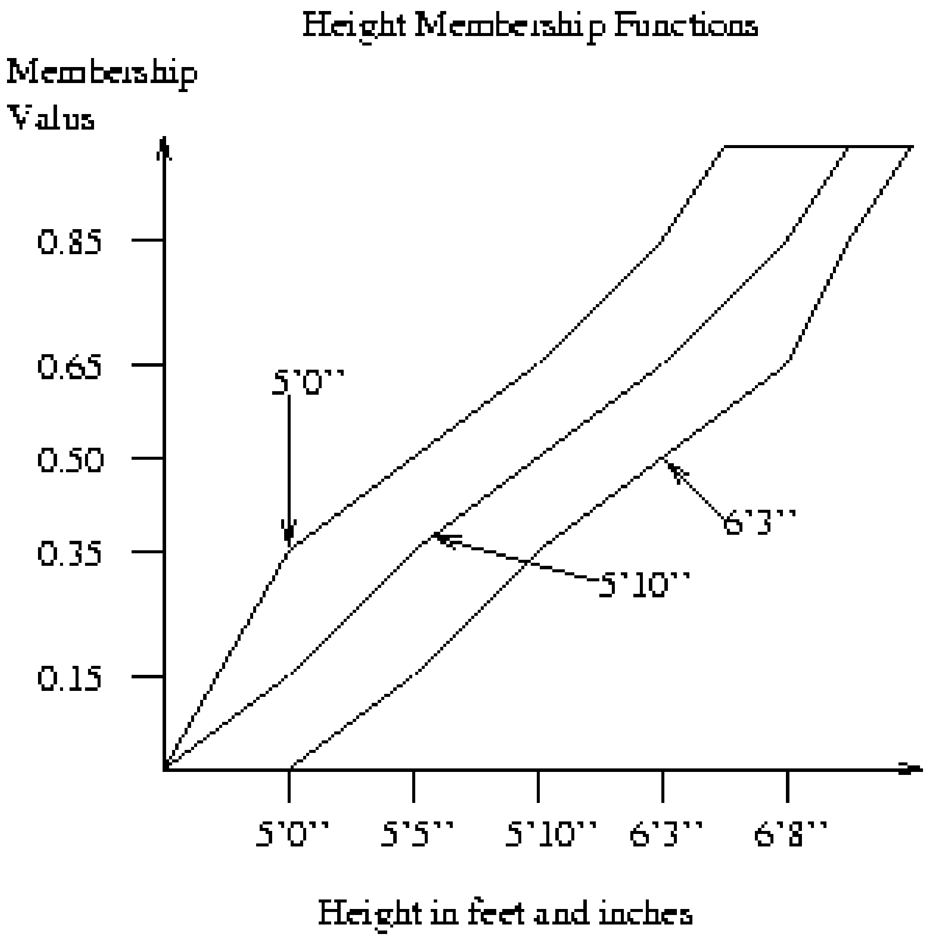
- What is fuzzy logic
  - The definition of fuzzy logic membership function
  - The definition of fuzzy logic rules
  - Ways o f combining fuzzy logic output values
- Fuzzy functions
  - Definition of fuzzy set
  - Operations
  - Identities
  - Transformations
- Differences between Boolean logic and fuzzy logic
  - Validation of fuzzy functions

The Definition of Fuzzy Logic Membership Function

- A person's height membership function graph is shown next with linguistic values of the degree of membership as very tall, tall, average, short and very short being replaced by 0.85, 0.65, 0.50, 0.45 and 0.15.



- In traditional logic, statements can be either true or false, and sets can either contain an element or not.
- These logic values and set memberships are typically represented with number 1 and 0.
- Fuzzy logic generalizes traditional logic by allowing statements to be somewhat true, partially true, etc.
- Likewise, sets can have full members, partial members, and so on.
- For example, a person whose height is 5’ 9” might be assigned a membership of 0.6 in the fuzzy set “tall people”.
- The statement “Joe is tall” is 60% true of Joe is 5’9”.
- Fuzzy logic is a set of “if--then” statements based on combining fuzzy sets. (Beale & Demuth..Fuzzy Systems Toolbox.)



Fuzzy Sets, Statements, and Rules

- A crisp set is simply a collection of objects taken from the universe of objects.
- Fuzzy refers to linguistic uncertainty, like the word "tall".
- Fuzzy sets allow objects to have membership in more than one set:
  - e.g. 6' 0" has grade 70% in the set "tall" and grade 40% in the set "medium".
- A fuzzy statement describes the grade of a fuzzy variable with an expression:
  - e.g. Pick a real number greater than 3 and less than 8.

The Definition of Fuzzy Logic Rules

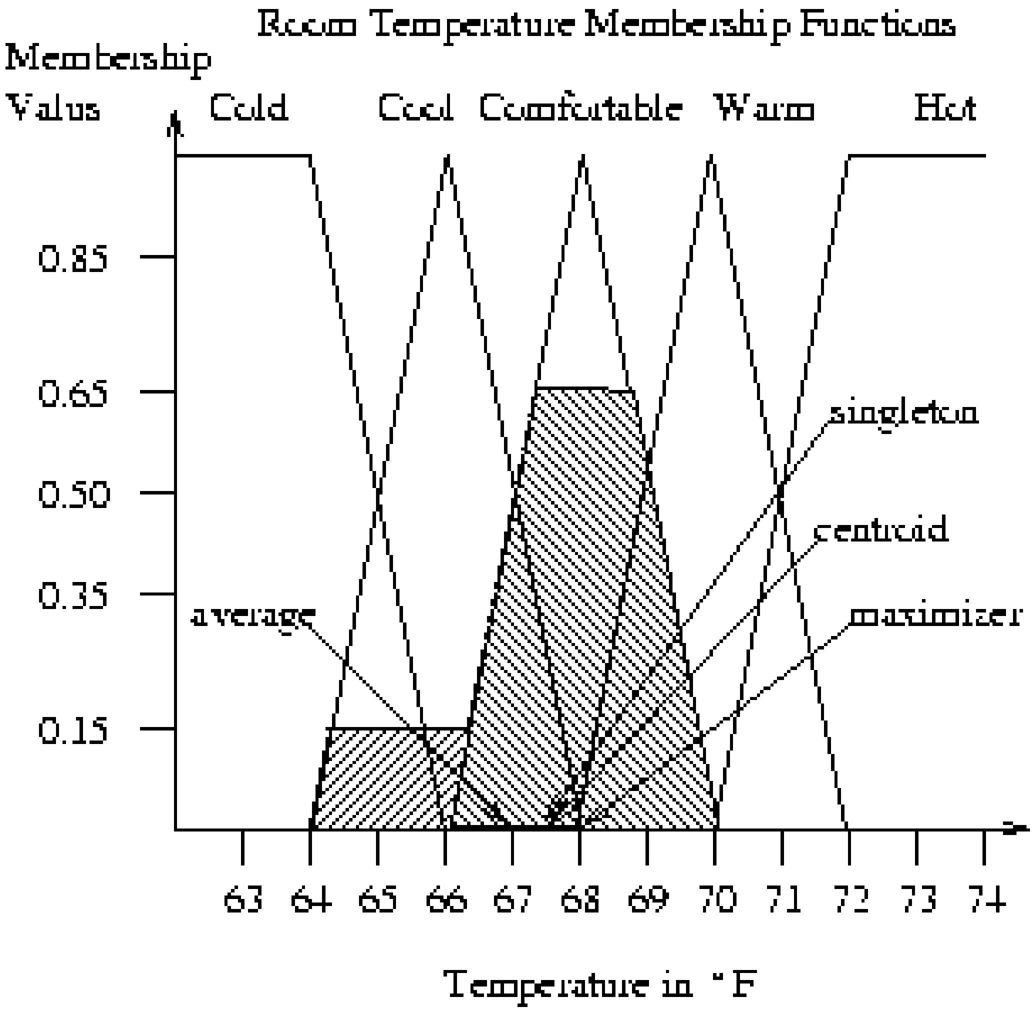
- A fuzzy logic system uses fuzzy logic rules, as in an expert system where there are many *if-then* rules.
  - A fuzzy logic rule uses membership functions as variables.
- A **fuzzy logic rule** is defined as an *if* variable(s) and *then* output fuzzy variable(s).
- Fuzzy logic variables are **connected together** like binary equations with the variables separated with operators of AND, OR, and NOT.

Ways of Combining Fuzzy Logic Output Values

There are four different techniques to combine the fuzzy logic rules output values.

They are:

- Maximizer
- Average
- Centroid
- Singleton



Identities

- The form of identities used in fuzzy variables are the same as elements in fuzzy sets.
- The definition of an element in a fuzzy set, {(x,u a(x))}, is the same as a fuzzy variable x and this form will be used in the remainder of the paper.
- Fuzzy functions are made up of fuzzy variables.

The identities for fuzzy algebra are:

Idempotency: $X + X = X$, $X * X = X$

Commutativity: $X + Y = Y + X$, $X * Y = Y * X$

Associativity: $(X + Y) + Z = X + (Y + Z)$,

$(X * Y) * Z = X * (Y * Z)$

Absorption: $X + (X * Y) = X$, $X * (X + Y) = X$

Distributivity: $X + (Y * Z) = (X + Y) * (X + Z)$,

$X * (Y + Z) = (X * Y) + (X * Z)$

Complement: $X'' = X$

DeMorgan's Laws: $(X + Y)' = X' * Y'$, $(X * Y)' = X' + Y'$

Transformations

Some transformations of fuzzy sets with examples follow:

$x'b + xb = (x + x')b \neq b$

$xb + xx'b = xb(1 + x') = xb$

$x'b + xx'b = x'b(1 + x) = x'b$

$a + xa = a(1 + x) = a$

$a + x'a = a(1 + x') = a$

$a + xx'a = a$

$a + 0 = a$

$x + 0 = x$

$x * 0 = 0$

$x + 1 = 1$

$x * 1 = x$

Examples:

$a + xa + x'b + xx'b = a(1 + x) + x'b(1 + x) = a + x'b$

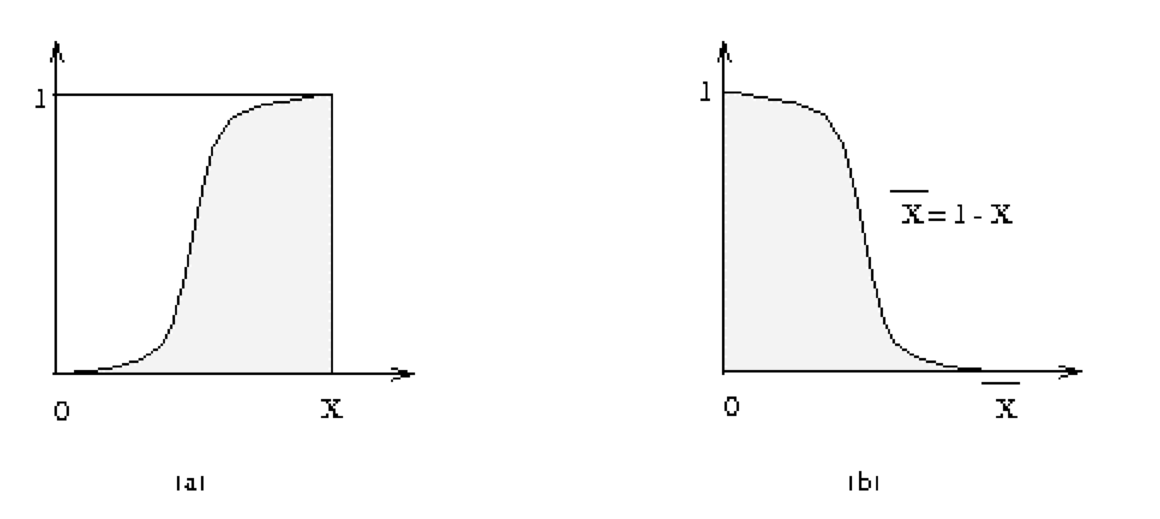
$a + xa + x'a + xx'a = a(1 + x + x' + xx') = a$

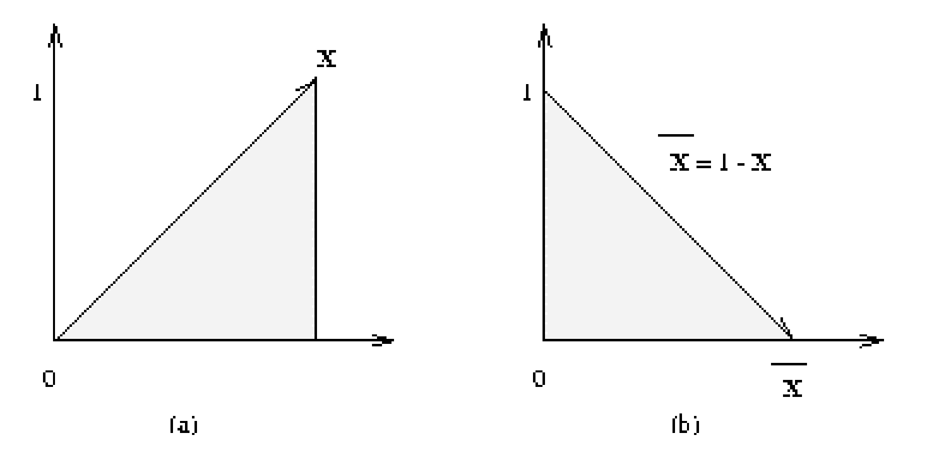
Differences Between Boolean Logic and Fuzzy Logic

Boolean logic the value of a variable and its inverse are always disjoint (X * X’ = 0) and (X + X’ = 1) because the values are either zero or one.

Fuzzy logic membership functions can be either disjoint or non-disjoint.

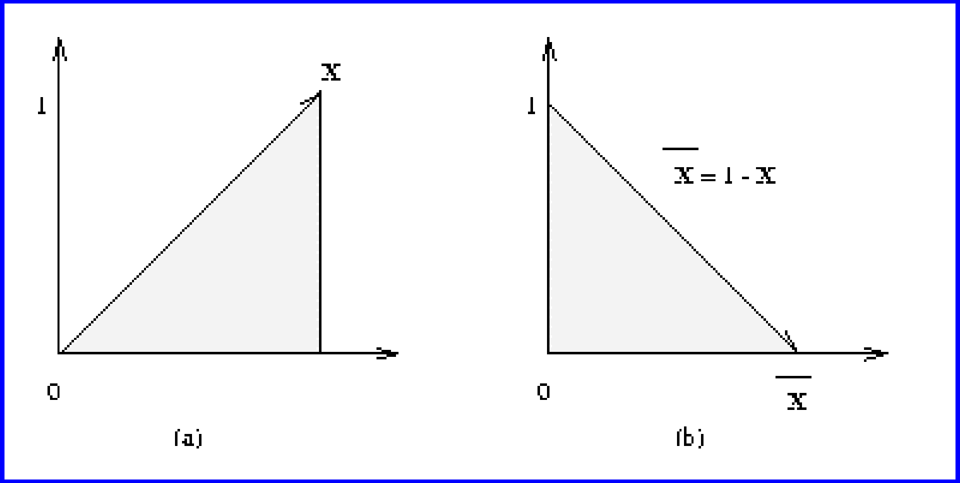
Example of a fuzzy non-linear and linear membership function X is shown (a) with its inverse membership function shown in (b).



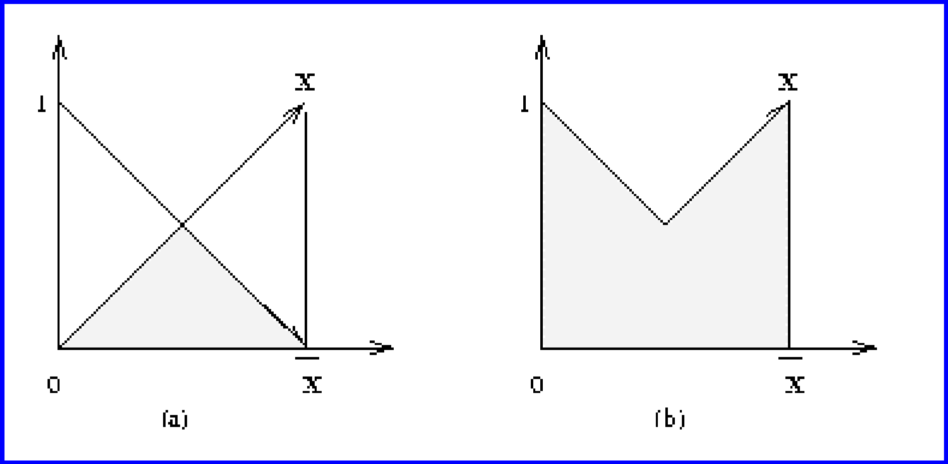


Fuzzy Intersection and Union

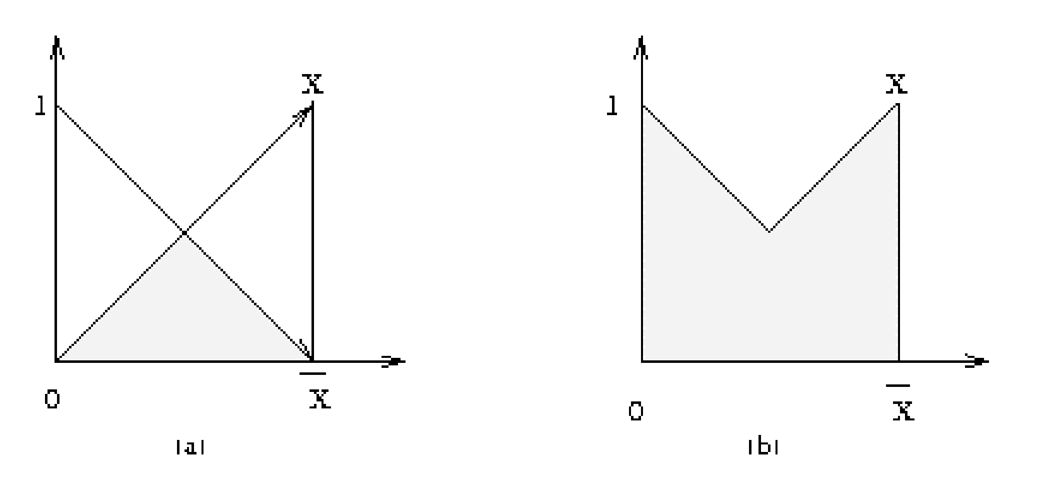
- From the membership functions shown in the top in (a), and complement X’ (b) the intersection of fuzzy variable X and its complement X’ is shown bottom in (a).



- From the membership functions shown in the top in (a), and complement X’ (b) the union of fuzzy variable X and its complement X’ is shown bottom in (b).



Validation of Fuzzy Functions



- Two fuzzy functions are valid iff the function outputs are ≥ 0.5 under all possible assignments.
- This is like doing EXOR of two binary functions shown in (b) which is the same as union.
- Two fuzzy functions are inconsistent iff the function output is ≤ 0.5 under all possible assignments. Thus, if the output of the two fuzzy functions is < 0.5 then the two fuzzy functions are inconsistent.
- This is like exnor of two binary functions of shown in (a) which is the same as intersection.

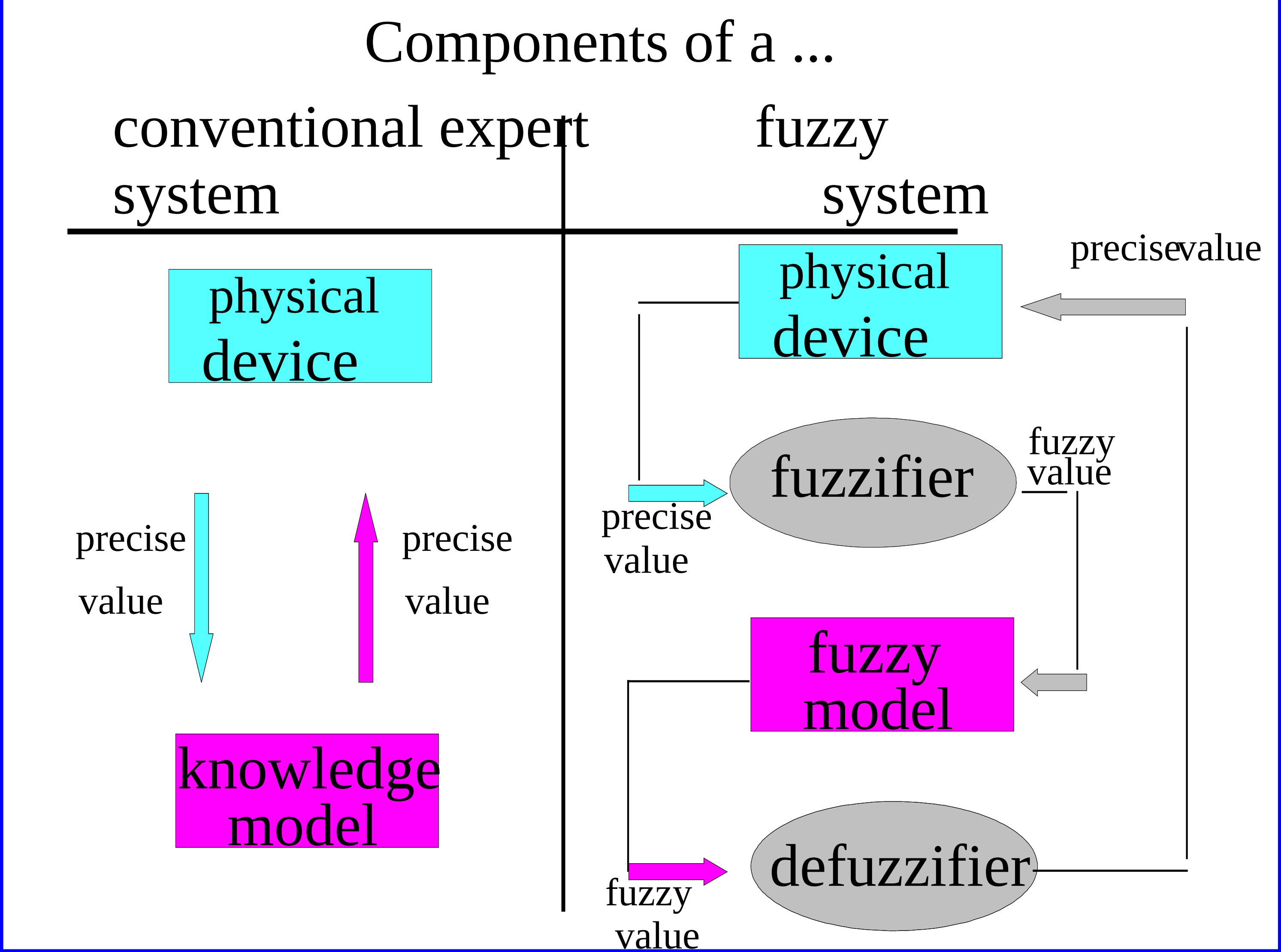
FUZZY LOGIC SYSTEMS APPLICATIONS

- Sections of a Fuzzy Logic Control System
- Steps in Designing a Fuzzy Logic Control System
- Design of a Fuzzy Logic Control System
  - – Input membership function
  - – Fuzzy logic rules table
  - – Output membership function

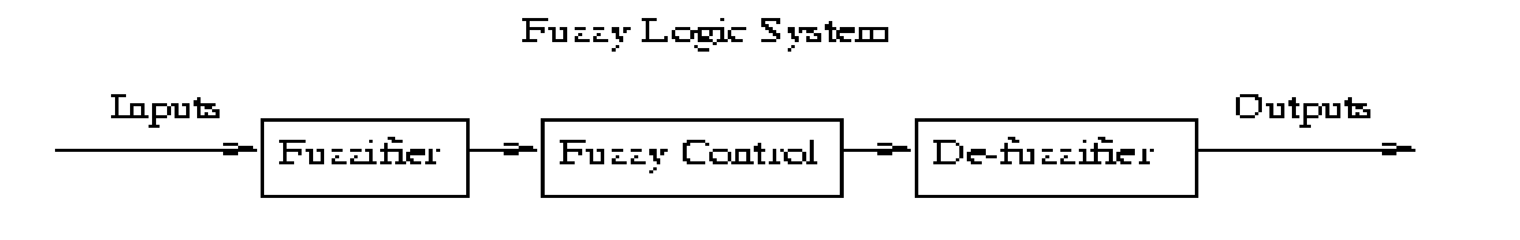
Components of Fuzzy system

- The components of a conventional expert system and a fuzzy system are the same.
- Fuzzy systems though contain **`fuzzifiers',**
  - which convert crisp numbers into fuzzy numbers,
- and **`defuzzifiers',**
  - which convert fuzzy numbers into crisp numbers.

Components of Fuzzy system



Sections of a Fuzzy Logic Control System



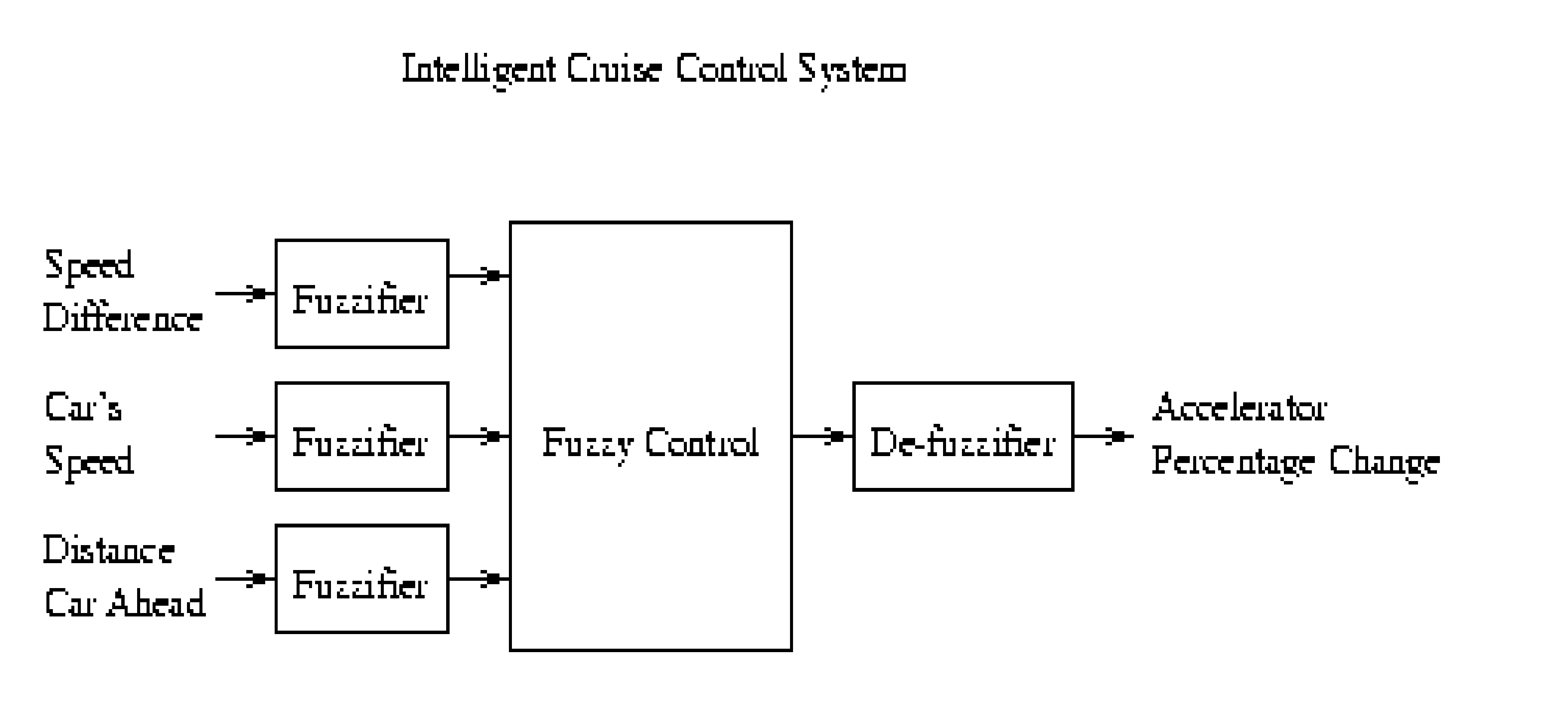
- **Fuzzifier section**
  - System inputs with range of values
  - Mapped to membership function can be non-linear in correspondence
- **Fuzzy control section**
  - Input fuzzy values
  - Through fuzzy rules
  - Produce the fuzzy output values
- **De-fuzzifier section**
  - Fuzzy output values are combined together in values needed for output

Steps in Designing a Fuzzy Logic Control System

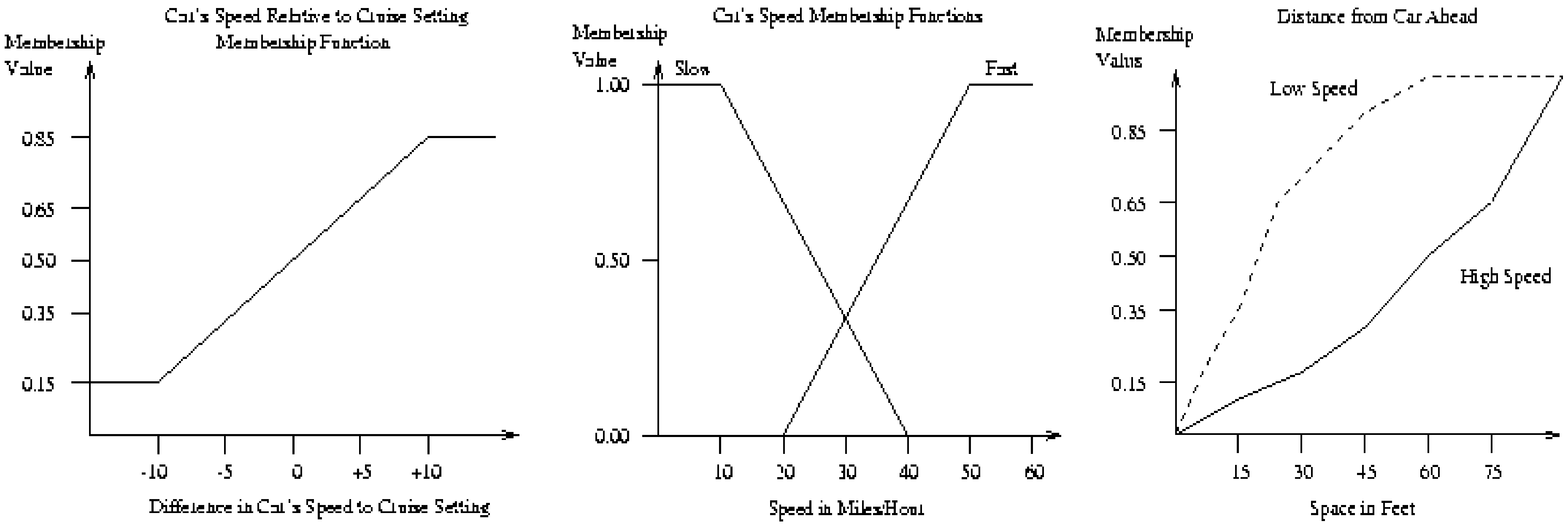
- Identify the system input variables, their ranges, and membership functions.
- Identify the output variables, their ranges, and membership functions.
- Identify the rules that describe the relations of the inputs to the outputs.
- Determine the de-fuzzifier method of combining fuzzy rules into system outputs.

Design of a Fuzzy Logic Control System

The block diagram of the intelligent cruise control system.



Input membership functions



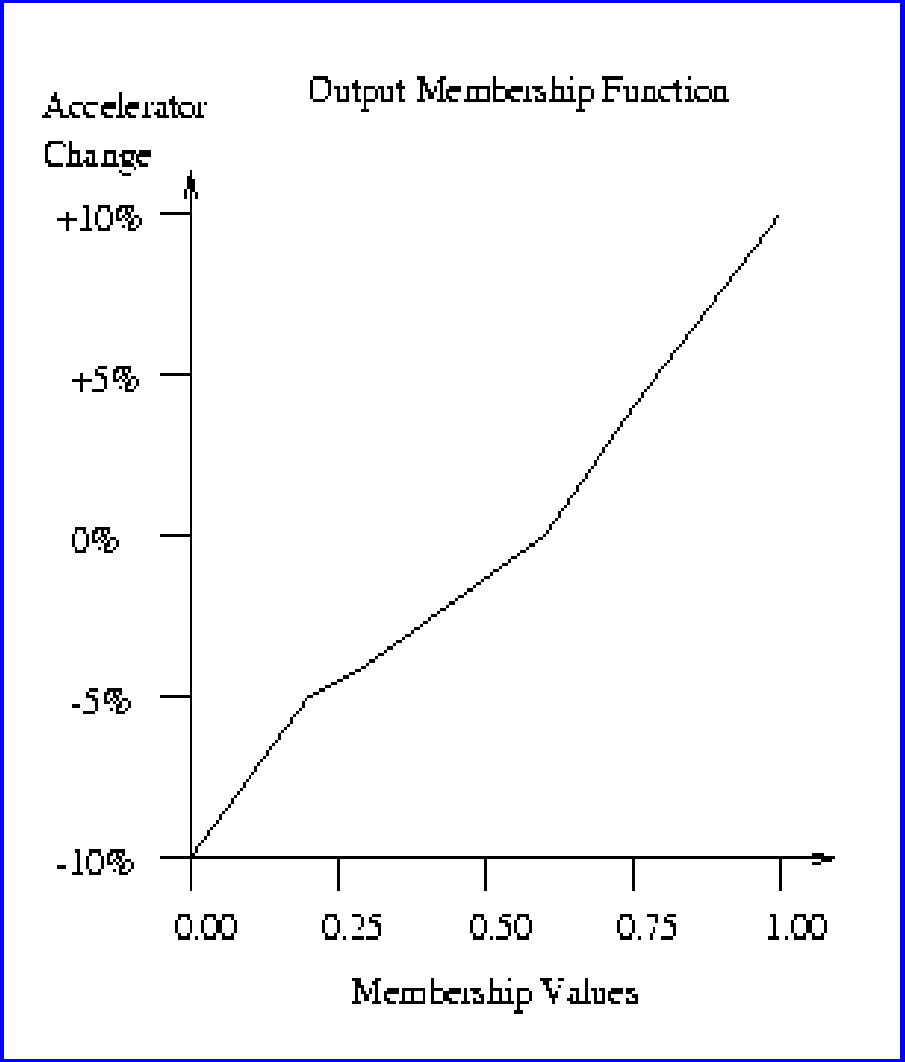
The three input membership functions

Fuzzy Logic Rules Table

cruise speed	slow / distance from car ahead					fast / distance from car ahead				
	15	30	45	60	75	15	30	45	60	75
+10	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15
+5	0.15	0.15	0.35	0.35	0.35	0.15	0.15	0.15	0.35	0.35
0	0.15	0.50	0.50	0.50	0.50	0.15	0.15	0.35	0.50	0.50
-5	0.35	0.65	0.65	0.85	0.85	0.15	0.35	0.50	0.65	0.85
-10	0.35	0.65	0.85	0.85	0.85	0.15	0.50	0.65	0.85	0.85

Output membership function

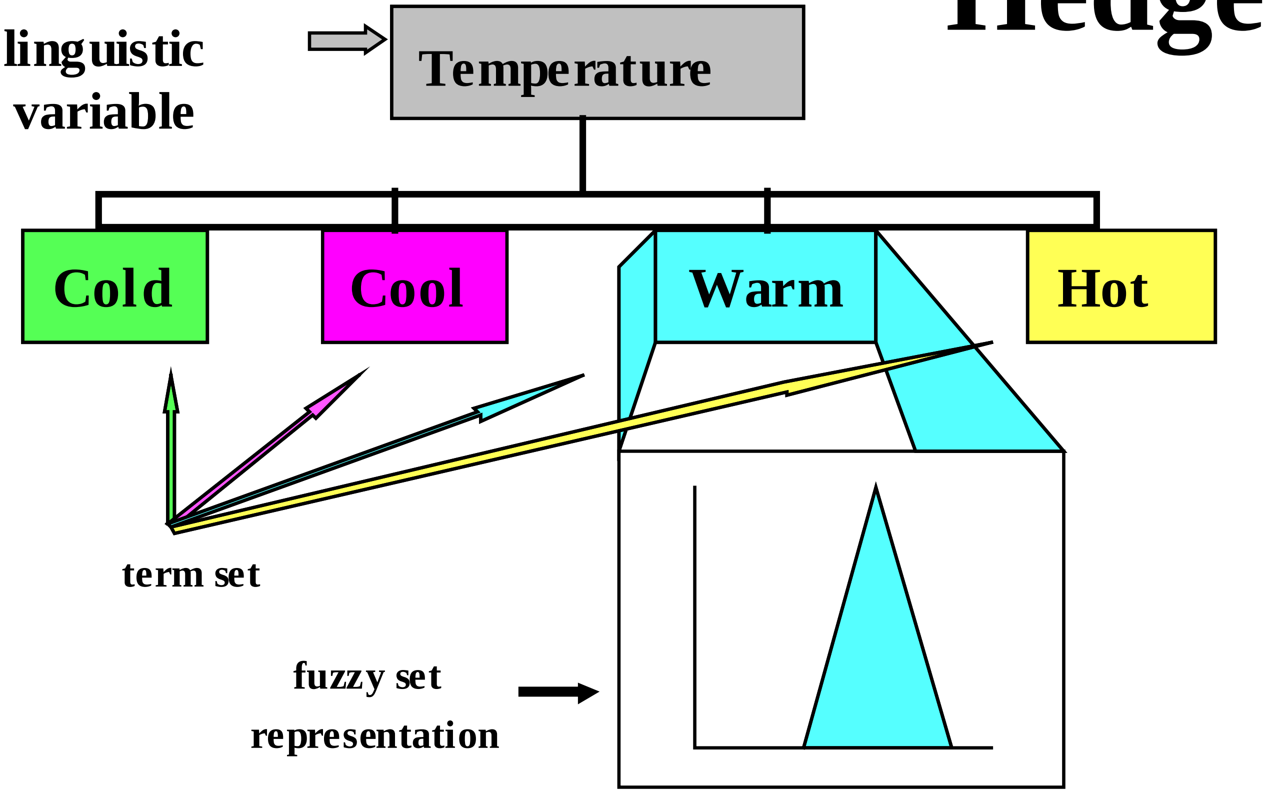
The output is the ***accelerator percentage change*** needed to keep the car a **safe distance** behind the car ahead and to keep the car at cruising speed.



Hedges

- The above diagram shows the relationship between linguistic variables, term sets and fuzzy representations.
  - Cold, cool, warm and hot are the linguistic values of the linguistic variable temperature.
  - In general a value of a linguistic variable is a composite term $u = u_1, u_2, \ldots, u_n$ where each u_n is an *atomic* term.

Hedges



Hedges

- The above diagram shows the relationship between linguistic variables, term sets and fuzzy representations.
  - Cold, cool, warm and hot are the linguistic values of the linguistic variable temperature.
  - In general a value of a linguistic variable is a composite term $u = u_1, u_2,...,u_n$ where each u_n is an *atomic* term.
  - From one atomic term by employing *hedges* we can create more terms.
    - Hedges such as very, most, rather, slightly, more or less etc.
    - Therefore, the purpose of the hedge is to create a larger set of values for a linguistic variable from a small collection of primary atomic terms.

Hedges

- This is achieved using the processes of:
  - normalisation,
  - intensifier,
  - concentration, and
  - dilation.
- For example, using concentration very u is defined by :

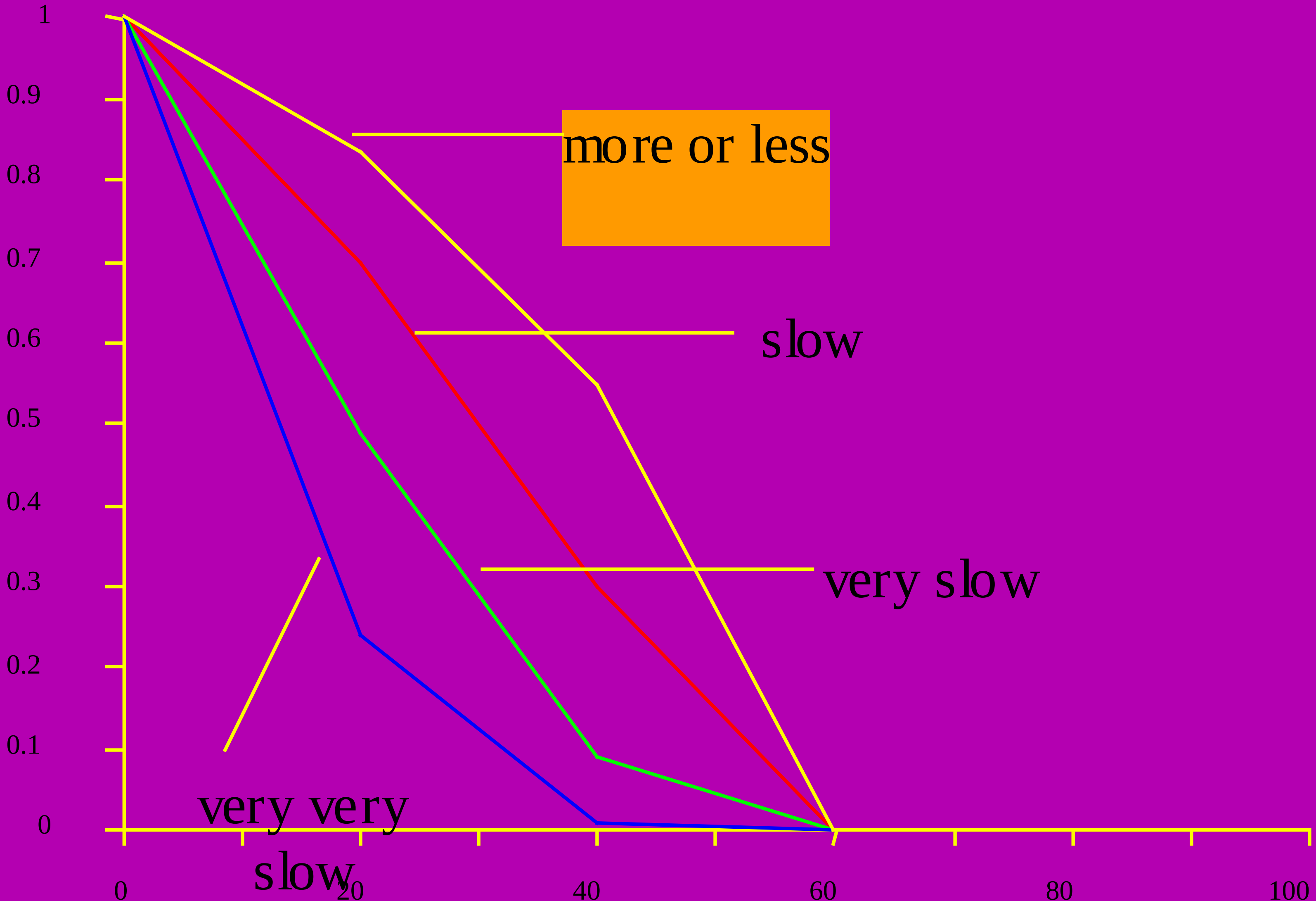
  very u = u^2 and

  very very u = u^4.

Hedges

- Let us assume the following definition for linguistic variable slow:
- u= 1.0/0 + 0.7/20 + 0.3/40 + 0.0/60 + 0.0/80 + 0.0/100.
- Then,
  - Very slow = u^2 = 1.0/0 + 0.49/20 + 0.09/40 + 0.0/60 + 0.0/80 + 0.0/100
  - Very Very slow = u^4 = 1.0/0 + 0.24/0 + 0.008/40 + 0.0/60 + 0.0/80 + 0.0/100
  - More or less slow = $u^{0.5}$ = 1.0/0 + 0.837/20 + 0.548/40 + 0.0/60 + 0.0/80 + 0.0/100

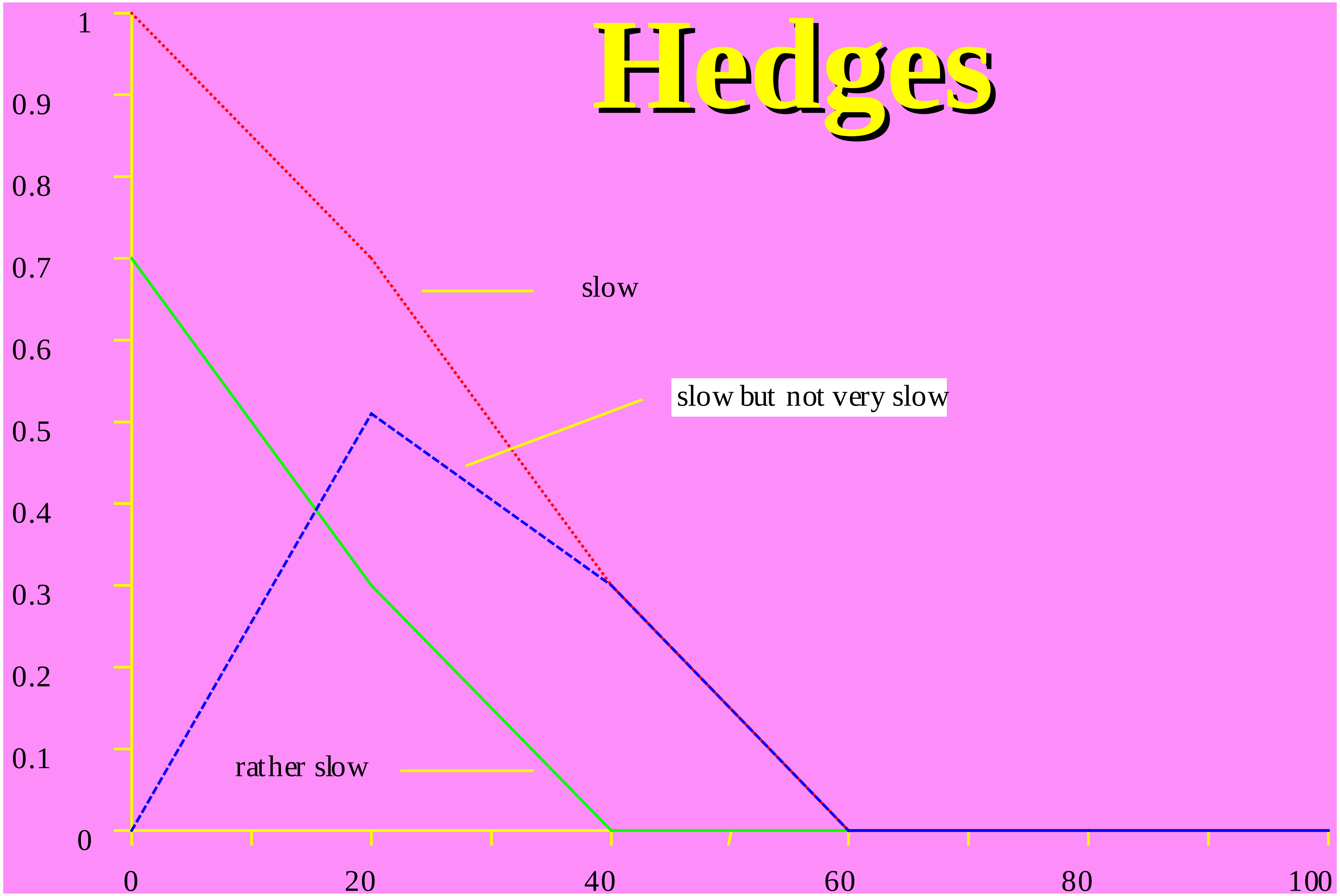
Hedges


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
20
40
60
80
100
more or less
slow
very slow
very very
slow

Hedges

- The hedge ***rather*** is a linguistic modifier that moves each membership by an appropriate amount C.
  - Setting C to unity we get.
  - Rather slow = 0.7/0 + 0.3/20 + 0.0/40 + 0.0/60 + 0.0/80
- The ***slow but not very slow*** is a modification which is using the connective ***but,*** which in turn is an intersection operator.
- The membership function in its discrete form was found as follows:
  - slow = 1.0/0 + 0.7/20 + 0.3/40 + 0.0/60 + 0.0/80 + 0.0/100
  - very slow = 1.0/0 + 0.49/20 + 0.09/40 + 0.0/60 + 0.0/80 + 0.0/100
  - not very slow = 0.0/0 + 0.51/20 + 0.91/40 + 1.0/60 + 1.0/80 + 1.0/100
  - slow but not very slow = min(slow, not very slow) = 0.0/0 + 0.51/20 + 0.3/40 + 0.0/60 + 0.0/80 + 0.0/100

Hedges


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
20
40
60
80
100
slow
slow but not very slow
rather slow

Hedges

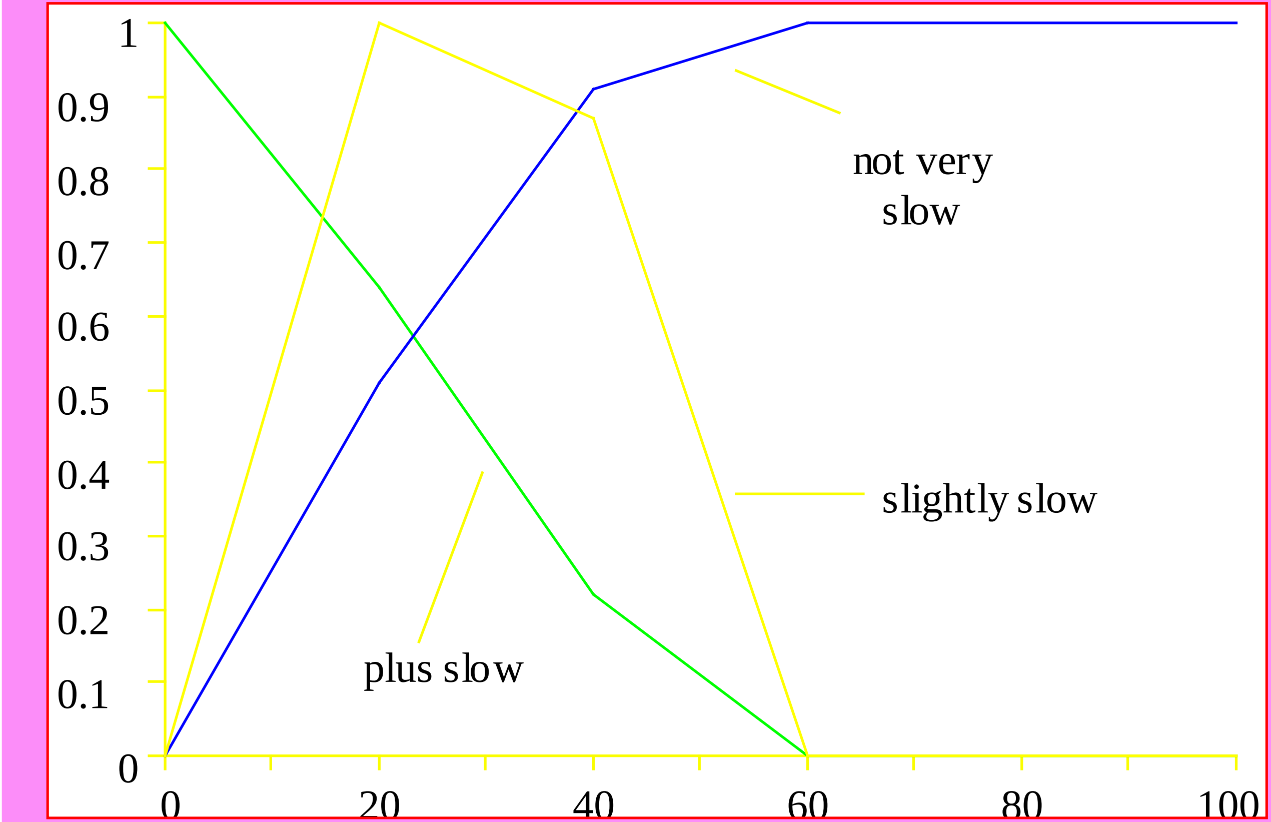
- The *slightly* hedge is the fuzzy set operator for intersection acting on the fuzzy sets *Plus* slow and *Not (Very* slow).
- Slightly slow = INT(NORM(PLUS slow and NOT VERY slow) where Plus slow is slow to the power of 1.25, *and* is the intersection operator.
  - slow = 1.0/0 + 0.7/20 + 0.3/40 + 0.0/60 + 0.0/80 + 0.0/100
  - plus slow = 1.0/0 + 0.64/20 + 0.222/40 + 0.0/60 + 0.0/80 + 0.0/100
  - not very slow = 0.0/0 + 0.51/20 + 0.91/40 + 1.0/60 + 1.0/80 + 1.0/100
  - plus slow and not very slow = min(plus slow, not very slow) = 0.0/0 + 0.51/20 + 0.222/40 + 0.0/60 + 0.0/80 + 0.0/100

Hedges

- norm (plus slow and not very slow) = (plus slow and not very slow/max) = 0.0/0 + 1.0/20 + 0.435/40 + 0.0/60 + 0.0/80 + 0.0/100

  - slightly slow = int (norm) = 0.0/0 + 1.0/20 + 0.87/40 + 0.0/60 + 0.0/80 +0.0/100.

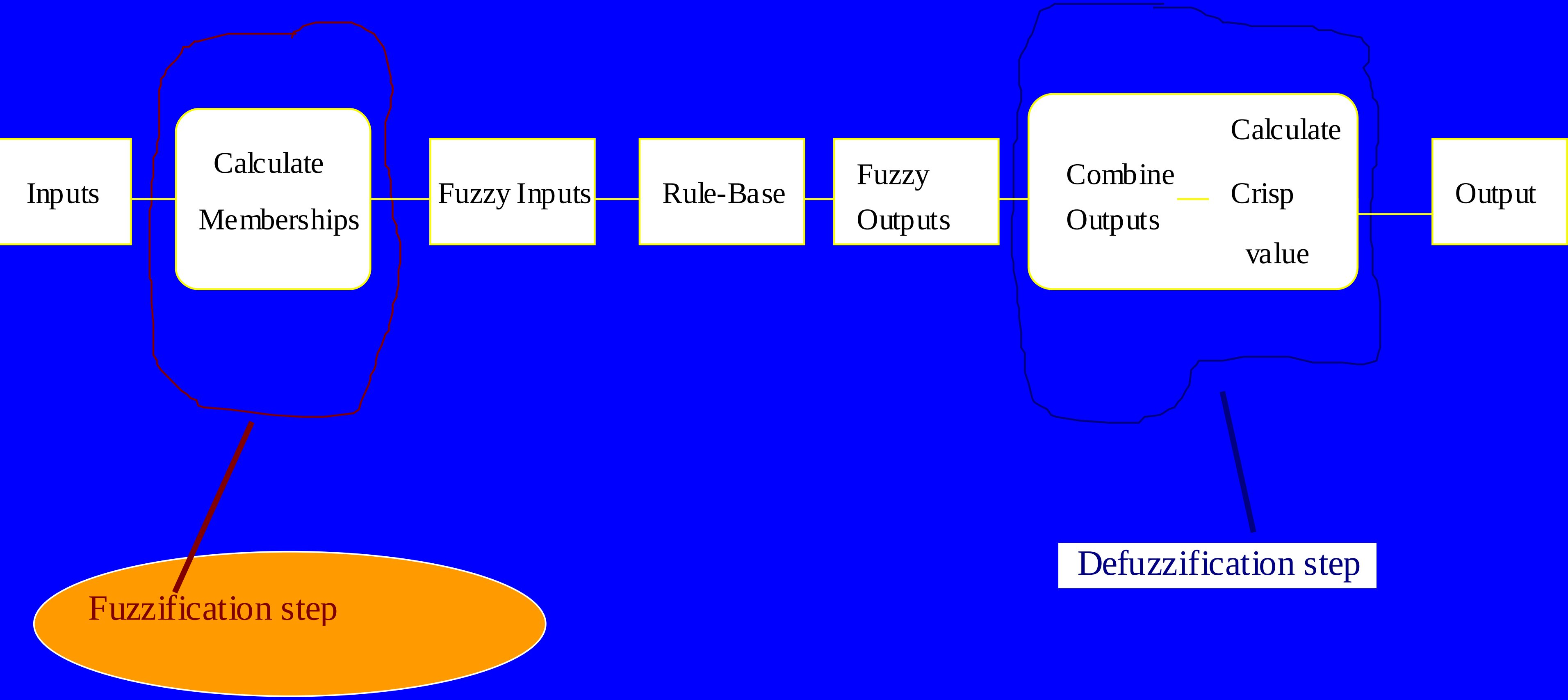
Hedges


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
20
40
60
80
100
not very slow
slightly slow
plus slow

Hedges

- **Now we are in a better position to understand the meaning of the syntactic and semantic rule.**
  - **A syntactic rule defines, in a recursive fashion, more term sets by using a hedge.**
  - **For instance T(slow)={slow, very slow, very very slow,...}.**
  - **The semantic rule defines the meaning of terms such as *very slow* which can be defined as *very slow* = *(slow)*^{2}.**
  - **One is obviously allowed either to generate new hedges or to modify the meaning of existing ones**

Fuzzy data processing



Inputs

Calculate Memberships

Fuzzy Inputs

Rule-Base

Fuzzy Outputs

Combine Outputs

Calculate Crisp value

Output

Fuzzification step

Defuzzification step

Fuzzification

- The function of the fuzzification component is to convert crisp numbers to equivalent fuzzy sets.
- Please notice that the inputs might require some pre-processing in order to fit the range of the fuzzy system.

Fuzzy Inference

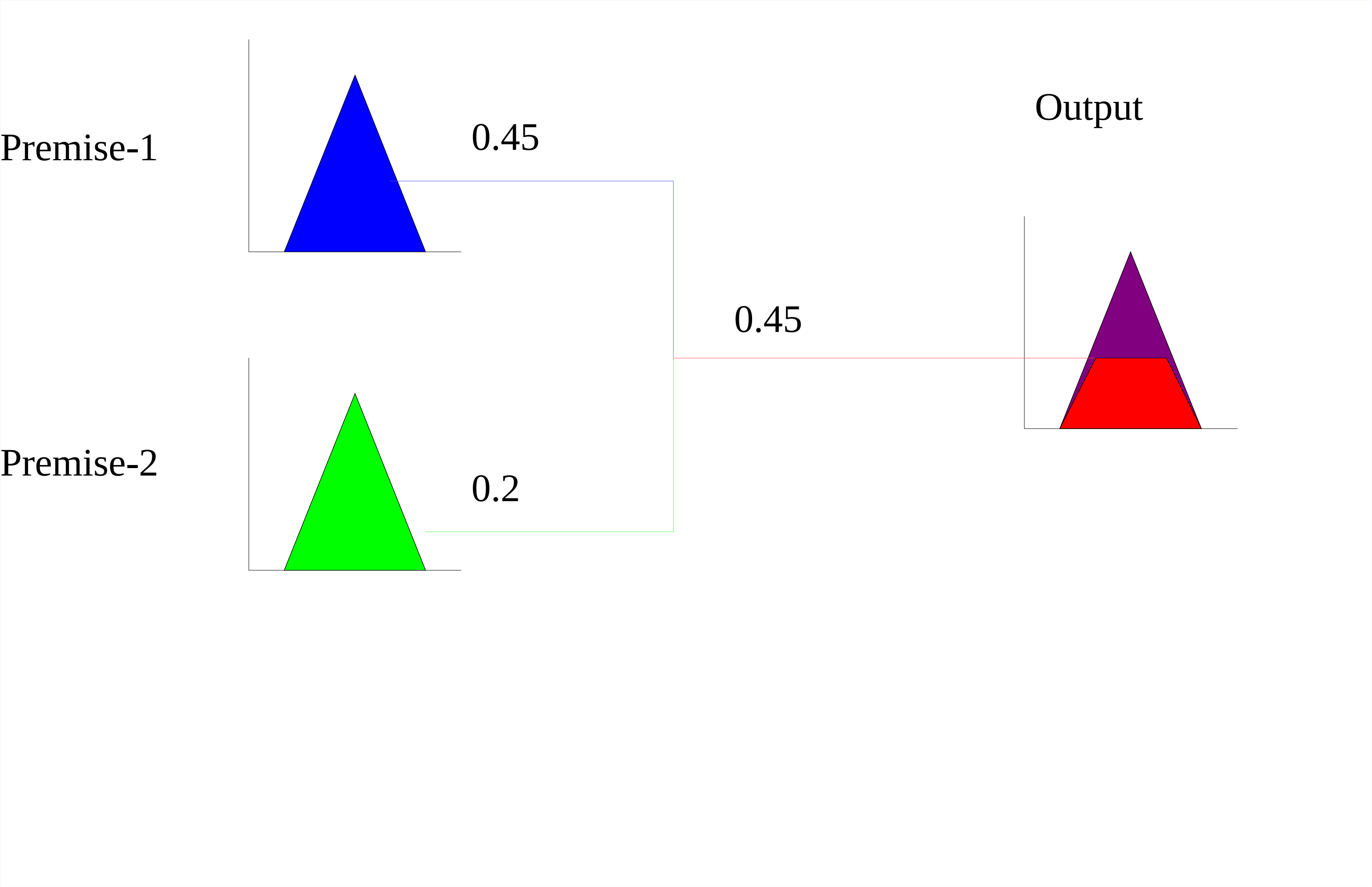
- At the end of the fuzzification step the working memory module contains the values of the fuzzified input.
  - Each production rule is examined and all the rules that have their premises satisfied `fire'.
  - Hence, the only rules which do not fire are those that at least one of their premises has a membership degree of zero.
  - In the case that more than rule fires, this is common and desirable, the system generates a single fuzzy output.
    - This is achieved by combining all fuzzy outputs.
  - The single fuzzy output is then passed to the defuzzification module which generates a crisp value.
    - Then the system is ready to start the entire process all over again.

Calculating the output of a rule

Calculating the output of a rule

- **Rules have <u>promises</u> which are usually combined together by the connective *and*.**
- **The output is calculated by taking the degree of membership of the lesser of all premises as the value of the combination and truncating the output fuzzy set at that level.**

Calculating the output of a rule



Premise-1

0.45

Output

0.45

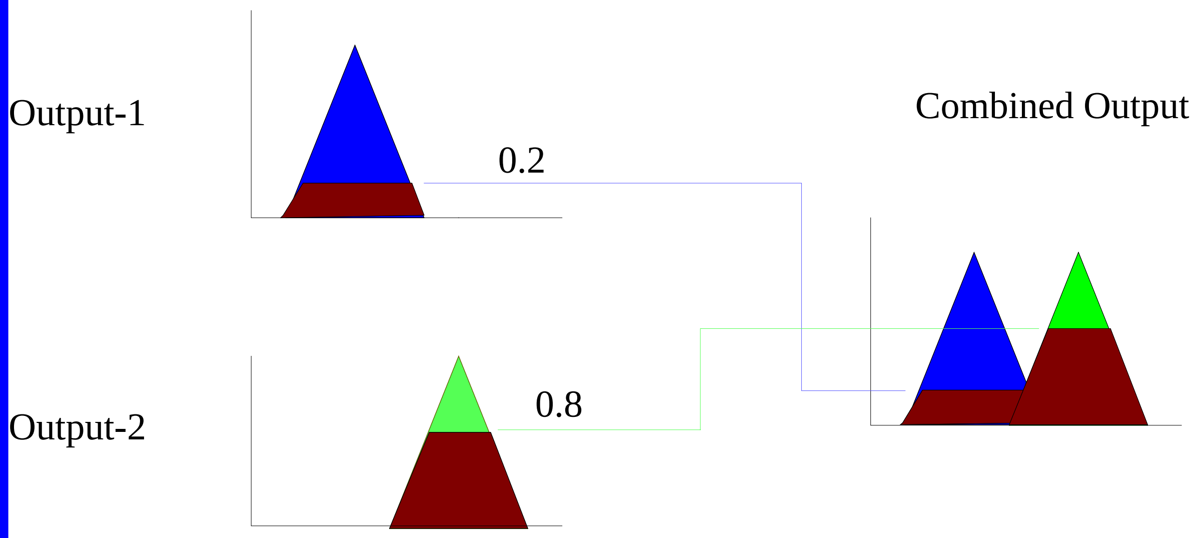
Premise-2

0.2

Combining rule outputs

- When all rules have been evaluated a single fuzzy set is calculated by combining all outputs.
- The combination involves the connective *or*.
- The single output is calculated by taking the **maximum** of their respective output fuzzy sets grades at each point along the horizontal axis.

Combining rule outputs



Output-1

0.2

Combined Output

Output-2

0.8

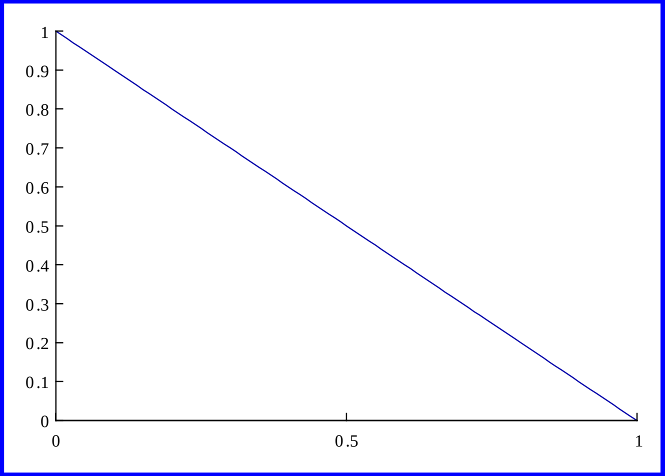
Defuzzification

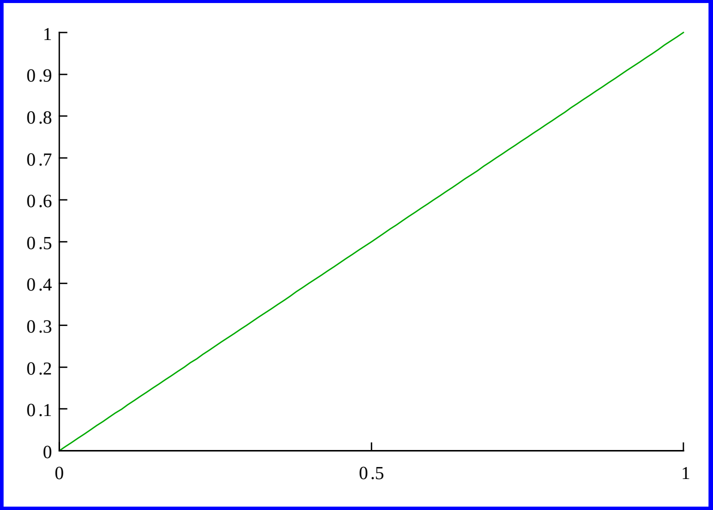
- The output of the combined operation is defuzzified before being broadcast to the external world.
- This implies the conversion of a fuzzy set to a crisp number.
- There are several techniques of defuzzification.

A SIMPLE EXAMPLE

- Assumptions
  - Let X,Y and Z be the linguistic variables.
  - Let the membership functions be low and high.
  - Let the membership functions be the same for all linguistic variables.
  - Define the membership functions as:
    - low(linguistic) = 1 - t
    - high(linguistic) = t
      - where t is a value in the interval [0,1].

A Simple Example (Membership Functions)





LOW

HIGH

A Simple Example (Rules)

- The rule base contains the following four rules:
  - if X is low and Y is low then Z is high (rule-1)
  - if X is low and Y is high then Z is low (rule-2)
  - if X is high and Y is low then Z is low (rule-3)
  - If X is high and Y is high then Z is high (rule-4)

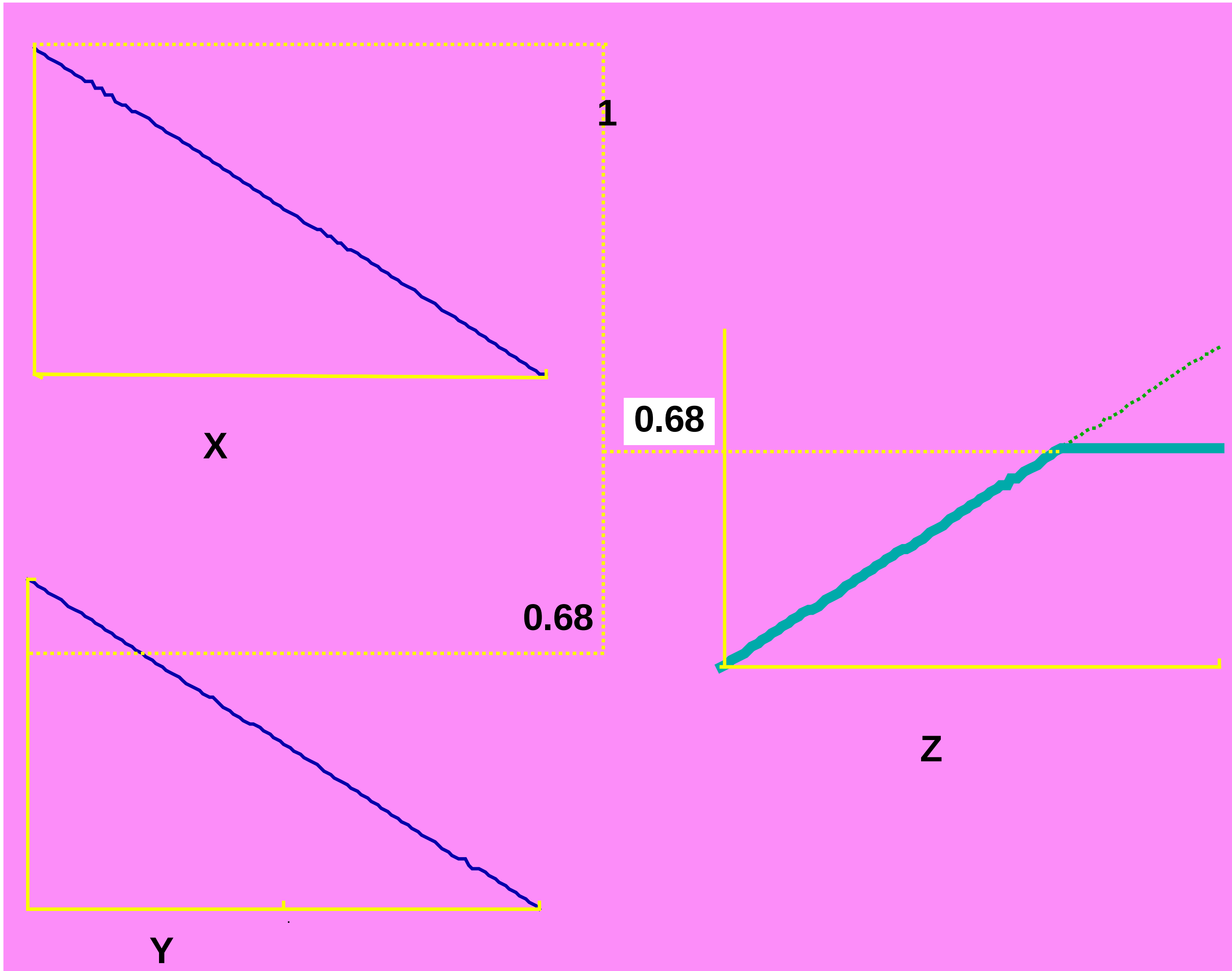
A Simple Example (Calculations)

- Next assume that the inputs are 0 and 0.32 for the linguistic variables X and Y respectively.
  - The problem then is to find if Z is high or low and its crisp value.
- The next steps are followed:
  - **Step-1:** Find the membership grade for the premises of each rule. We have 8 premises but due to replication only 4 are needed.
    - Low(X) = 1-t = 1-0 = 1
    - Low(Y) = 1-t = 1-0.32 = 0.68
    - High(X) = t = 0
    - High(Y) = t = 0.32

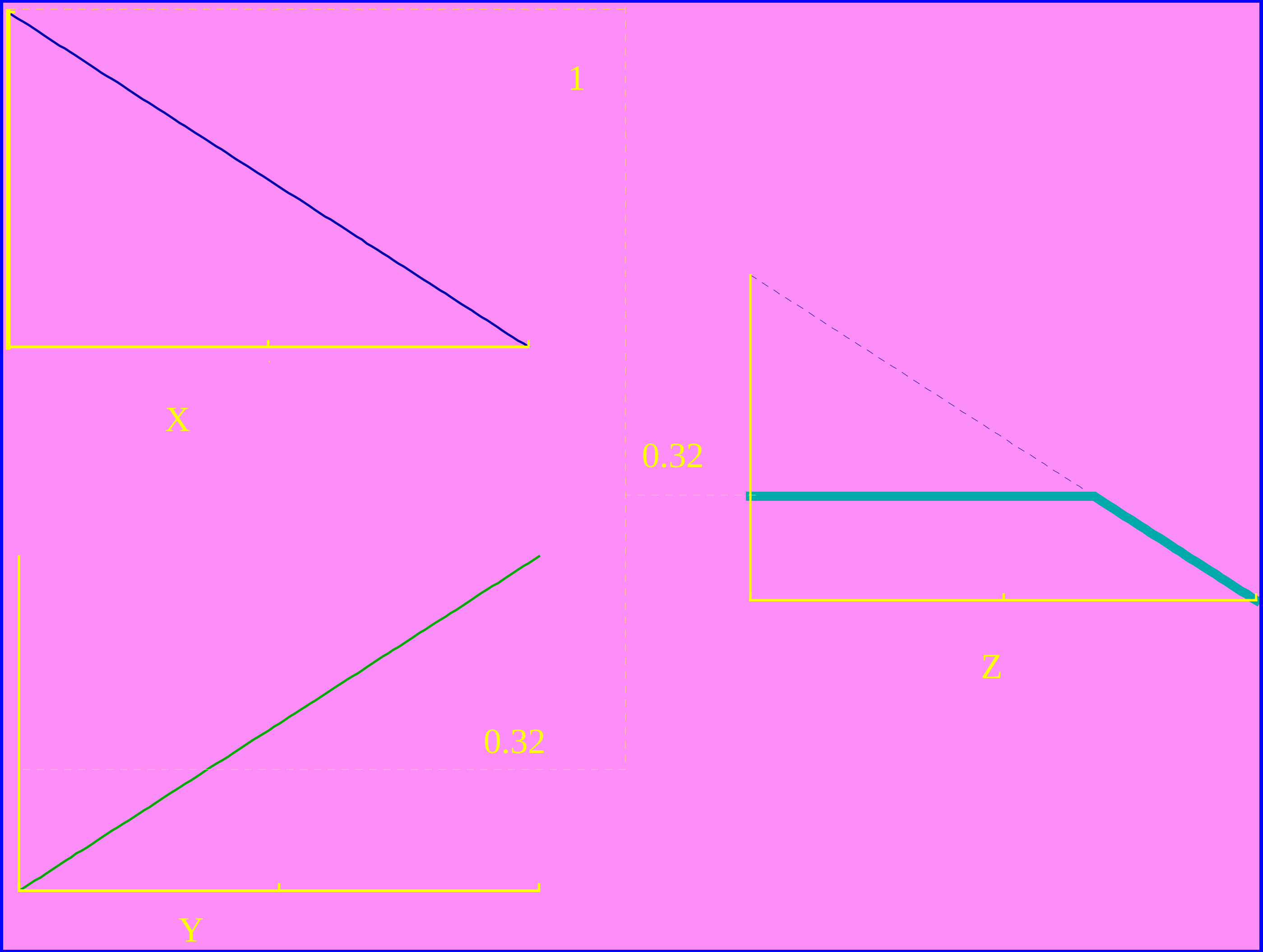
A Simple Example (Calculations)

- Step-2: Identify the rules that can fire. In our case only the first two rules can fire since the last two rules have a premise with a zero degree of membership.
- Step-3: For each rule that can fire find the strength of the firing.
  - Rule-1 = min(1,0.68) = 0.68
  - Rule-2 = min(1,0.32) = 0.32
  - Note that *min* is used because the premises are connected with a logical AND.

A Simple Example



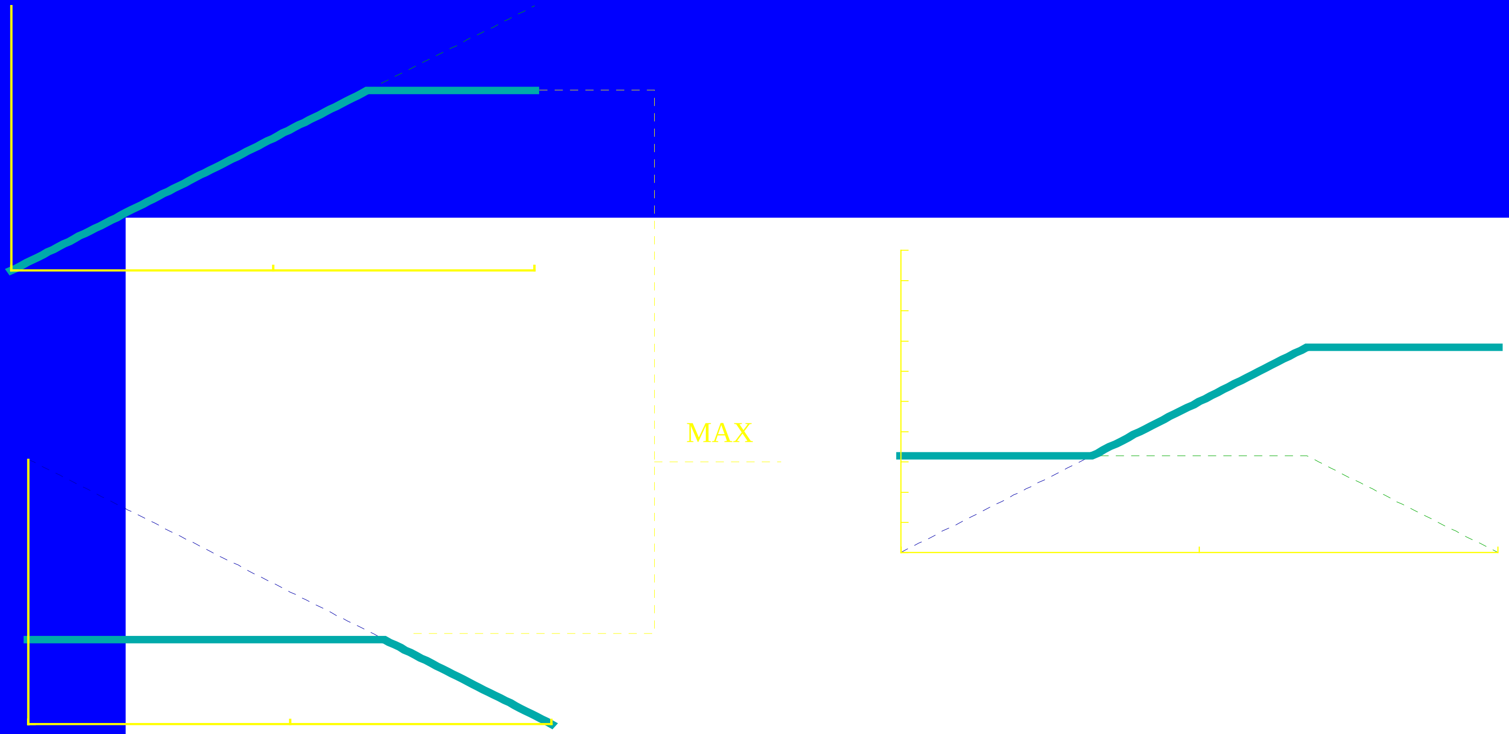
A Simple Example



A Simple Example (Calculations)

- Step-4: All the fuzzy outputs are combined together to form a single fuzzy subset.
- One way is to take the pointwise maximum value over all fuzzy outputs.
- In our case we have two fuzzy outputs, so for each point we take the maximum value.

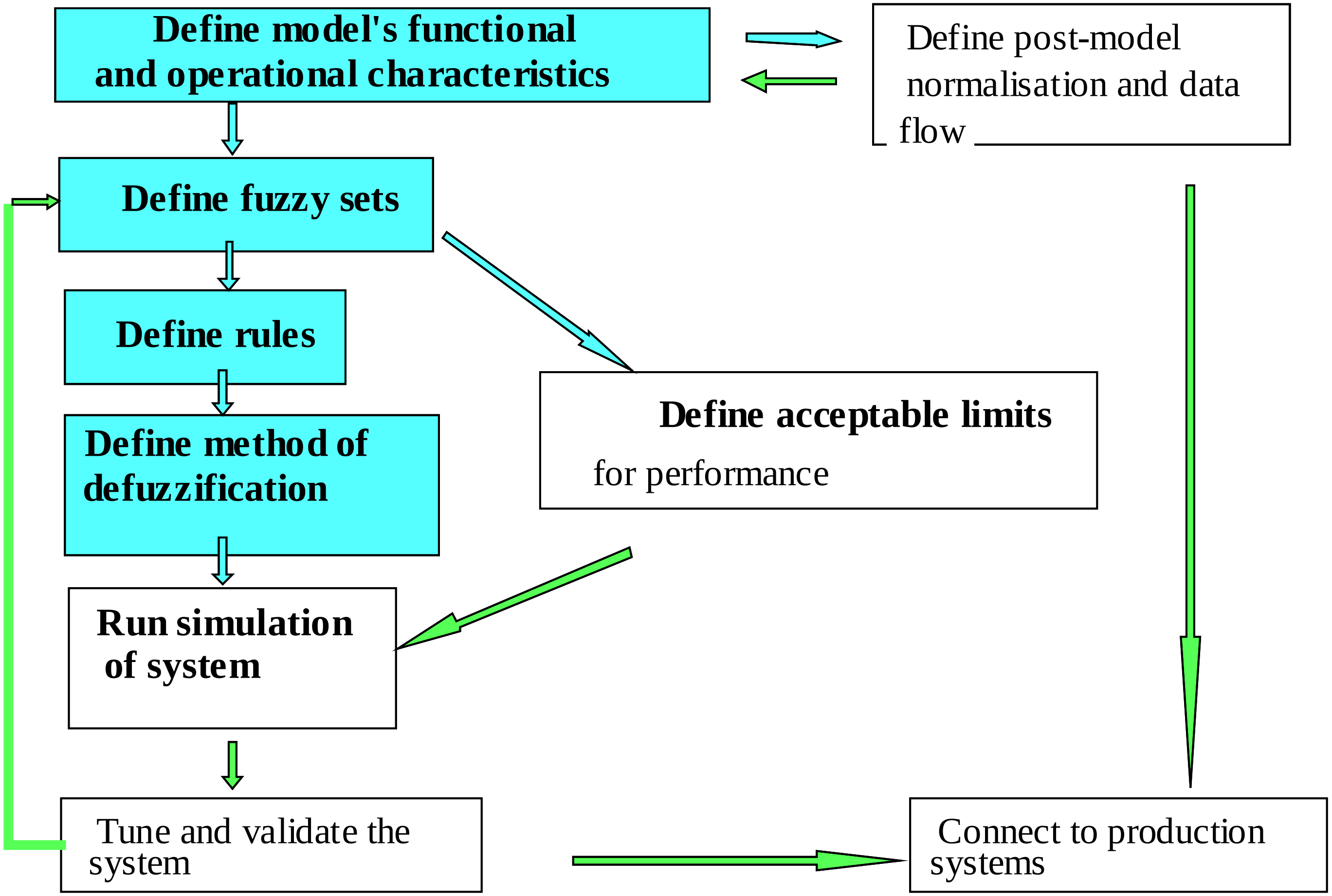
A Simple Example



A Simple Example (Calculations)

- Step-5: The fuzzy combined output needs to be converted to a single crisp value.
- Using one method which takes the Average-of-Maxima. With this method one finds the maximum peak of the fuzzy combined output - in our case this is 0.68.
- Then one collects all the t values for which the maximum value occurs - in our case there are 42 cases.
- Finally, the crisp value is the average of such variables - in our case 0.84.

Development Cycle


Define model's functional and operational characteristics
Define post-model normalisation and data flow
Define fuzzy sets
Define rules
Define acceptable limits for performance
Define method of defuzzification
Run simulation of system
Tune and validate the system
Connect to production systems

Development Cycle

Development Cycle

- Step 1: **The goal is to establish the characteristics of the system, and also to define the specific operating properties of the proposed fuzzy model.**
  - **Traditional systems analysis and knowledge engineering techniques can be employed at this stage. The designer must identify the relevant and appropriate inputs to the system; the basic transformations, if any, that are performed on the inputs,; and what output is expected from the system.**
  - **The designer must also decide if the fuzzy system is a subsystem of a `global' system and if so to define where the fuzzy subsystem fits into the `global architecture', or if it is the `global' system itself.**
  - **The numerical ranges of inputs and outputs must also be specified. This step applies to the development of all systems. Be prepared to throw away original versions at this stage.**

Development Cycle

- Step 2: **The goal is to decompose each control variable (i.e., inputs and outputs) into fuzzy sets and to give unique names to them.**
  - **It has been reported in the literature that the number of labels associated with a control variable should generally be odd and between five and nine.**
  - **Also, in order to obtain a smooth transition from a state to another each label should overlap somewhat with its neighbours. Overlapping of 10 to 50 percent is advised.**
  - **Finally, the density of the fuzzy sets should be highest around the optimal control point of the system and should thin out as the distance from that point increases.**

Development Cycle

- If possible begin with an exhaustive list of production rules and deal with redudant, impossible and implausible rules and/or conditions later.
  - When the number of rules increases dramatically, may be several rule-bases might be constructed.
  - Each rule-base to deal with a particular situation/condition of the system to be modelled.

Development Cycle

- – Step 3: **The goal is to obtain the production rules that tie the input values to the output values.**
- – **Since each production rule `declares' a small chunk of knowledge, the order in the knowledge base is unimportant.**
- – **Nevertheless, in order to maintain the knowledge base one should group the production rules by their premise variable. How many rules is obviously dependent on the application and is related to the number of control variables.**

Development Cycle

- – Step 4: **The goal is to decide on the way that is going to be used in order to convert a an output fuzzy set into a crisp solution variable.**
- – **As already indicated in another section there are many ways to perform the conversion but by and large, process control applications use the centroid technique.**
- – **The rest of the steps are similar to any modelling exercise. For instance, at the end of the fuzzy system construction the process of simulation commences. The model is compared against known test cases and the results are validated. When the results are not as desired changes are made until the desired performance is achieved.**

Popular Membership Functions

Popular Membership Functions

- **These are used in order to return the degree of membership of a numerical value for a particular set.**
  - **Fuzzy membership functions can have different shapes, depending on someone's experience or even preference.**
  - **Here we review some of the membership functions used in order to capture the modeler's sense of fuzzy numbers.**
  - **Membership functions can be drawn using** Subjective evaluation and elicitation (**Experts specify at the end of an elicitation phase the appropriate membership functions) or** Ad-hoc forms (**One can draw from a set of given different curves.**

Popular Membership Functions

- This simplifies the problem, for example to choosing just the central value and the slope on either side) or Converted frequencies (Information from a frequency histogram can be used as the basis to construct a membership function or by Learning and adaptation.
- For example, let us consdier the fuzzy membership function of the linguistic variable Tall. The following function can be one presentation:

Popular Membership Functions

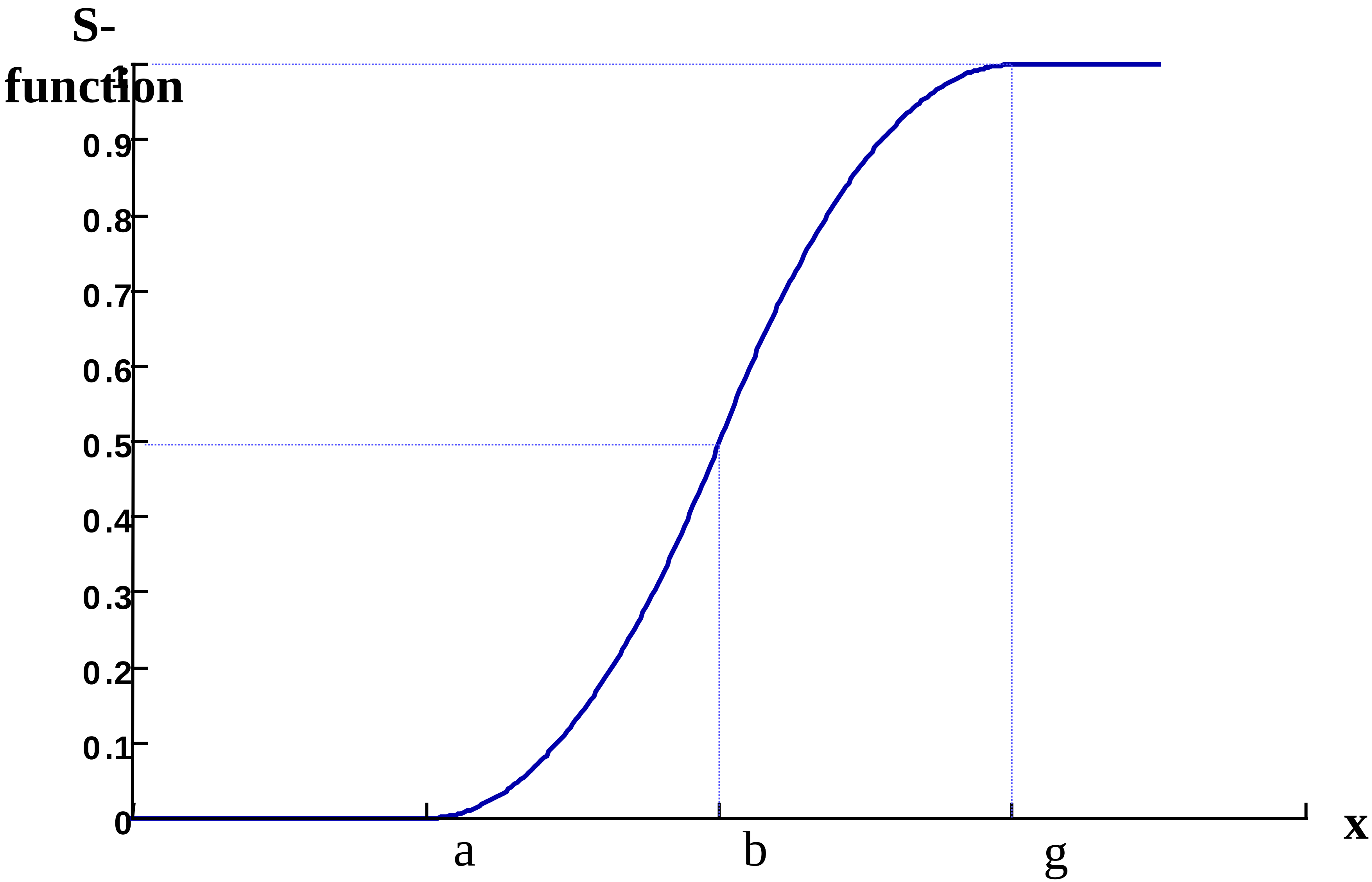
$$\text{Tall}(x) = \begin{cases} 0 \text{ if } \text{height}(x) < 5 \text{ feet} \\ \dfrac{\text{height}(x) - 5}{2} \text{ if } 5 \text{ feet} \le \text{height}(x) \le 7 \text{ feet} \\ 1 \text{ if } \text{height}(x) > 7 \text{ feet} \end{cases}$$

Popular Membership Functions

- Given the above definition the membership grade for an expression like `Dimitris is TAll' can be evaluated. Assuming a height of 6' 11'' the membership grade is 0.54
- Other popular shapes used are triangles and trapezoidals.

The S-Function

$$S(x;\alpha,\beta,\gamma)=\begin{cases} 0\, for\, x \le \alpha \\ 2\left(\dfrac{x-\alpha}{\gamma-\alpha}\right)^2 \; for \alpha \langle x \le \beta \\ 1-2\left(\dfrac{x-\alpha}{\gamma-\alpha}\right)^2 \; for\, \beta \langle x \le \gamma \\ 1 for\, x \rangle \gamma \end{cases}$$

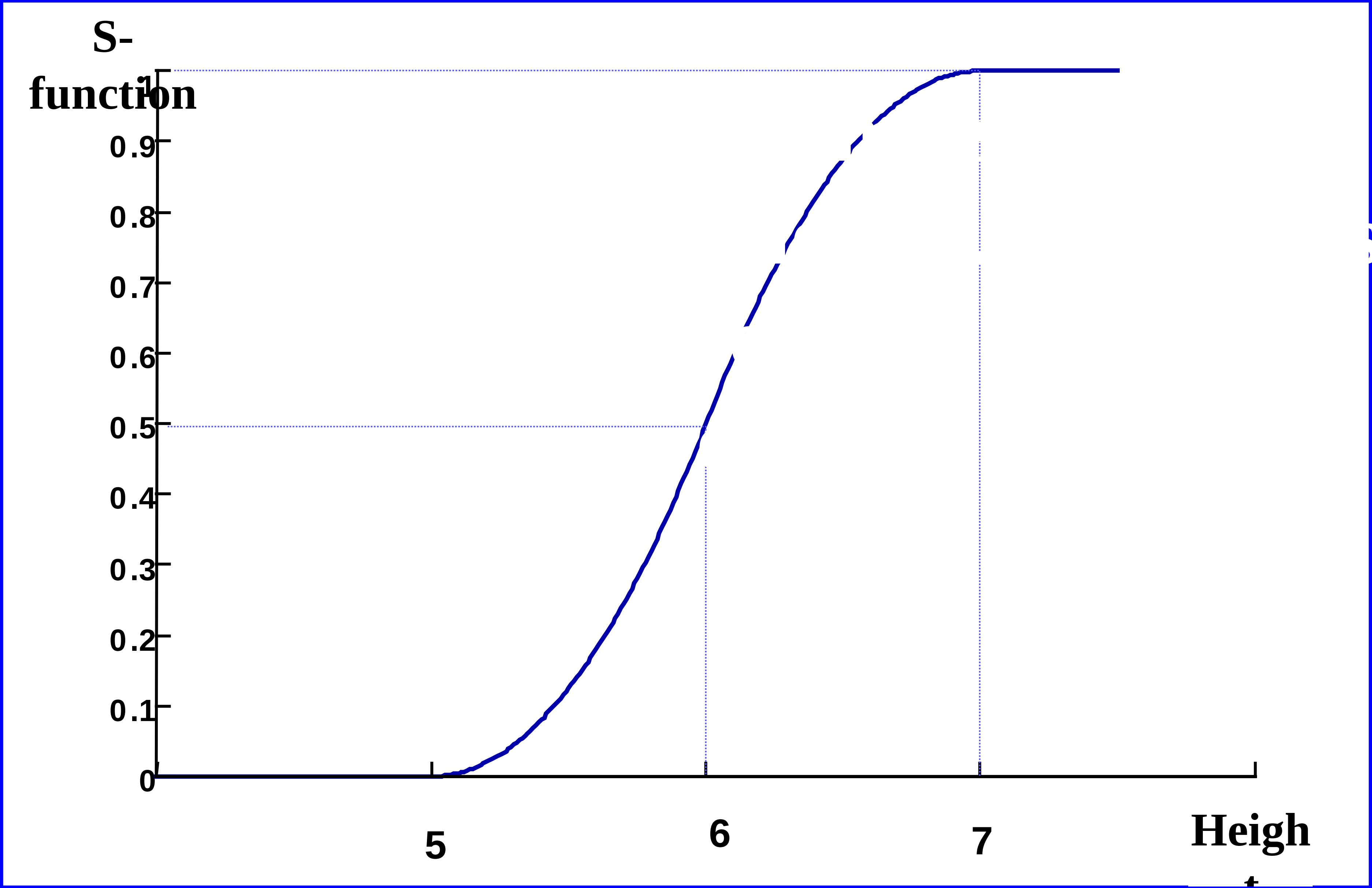

S-
function
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
x
a
b
g

The S-Function

- As one can see the S-function is flat at a value of 0 for $x \leq$ and at 1 for $x \geq$. In between and the S-function is a quadratic function of x.
  - To illustrate the S-function we shall use the fuzzy proposition *Dimitris is tall*.
  - Assuming that Dimitris is an adult and that the universe of discource is normal people (i.e., excluding the extremes of basketball players etc.) then we may assume that anyone less than 5 feet is not tall (i.e., =5) and anyone more than 7 feet is tall (i.e., =7). Hence, =6. Anyone between 5 and 7 feet has a membership function which increases monotonically with his height.

S-Function

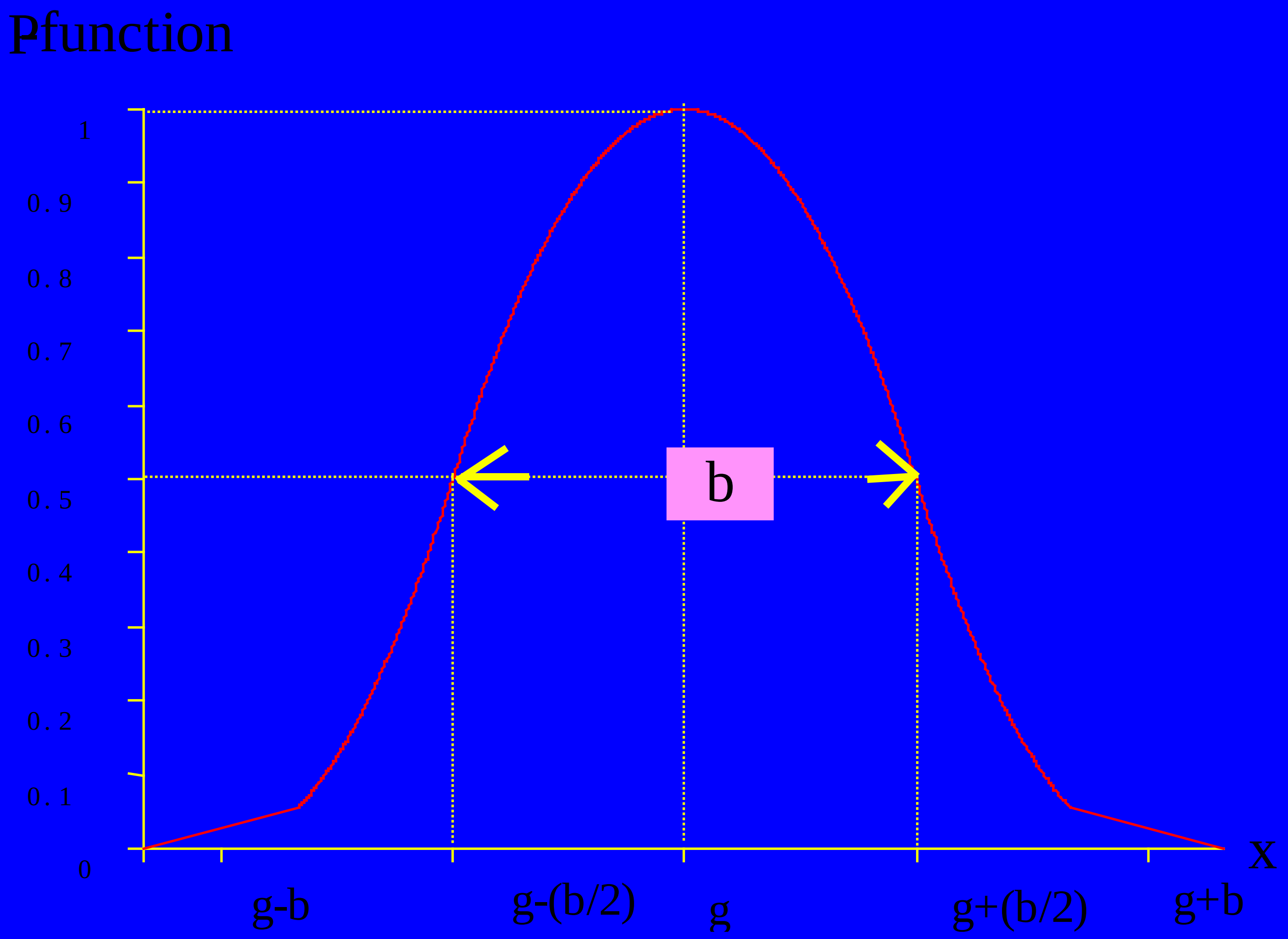
$$S(x;5,6,7)=\begin{cases} 0\, for\, x\le 5 \\ \left(\frac{x-5}{2}\right)^2 \; for 5\langle x\le 6 \\ 1-\left(\frac{x-7}{2}\right)^2 \; for 6\langle x\le 7 \\ 1 for x\rangle 7 \end{cases}$$


S-
function
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
5
6
7
Heigh
t

P-Function

$$\Pi(x;\beta,\gamma)=\begin{cases} S(x;\gamma-\beta,\gamma-\frac{\beta}{2},\gamma)\text{ for x}\le\gamma \\ 1-S(x;\gamma,\gamma+\frac{\beta}{2},\gamma+\beta)\text{ for x}\ge\gamma \end{cases}$$

P-Function

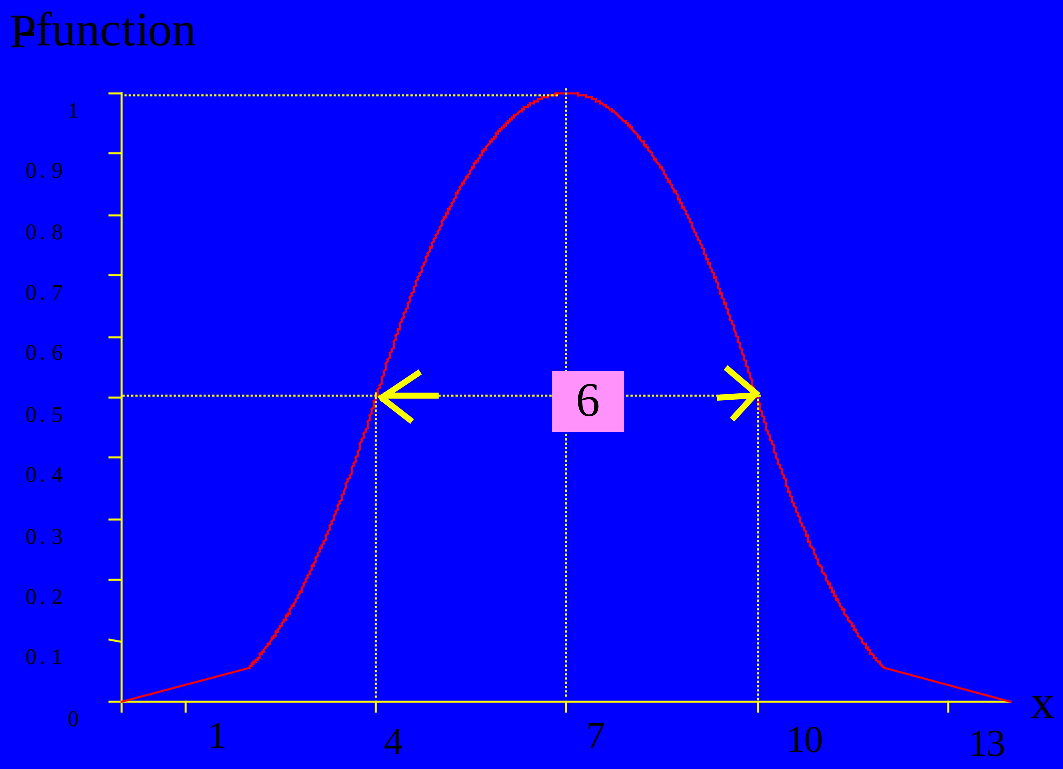


P-Function

- **The -function goes to zero at γ β, and the 0.5 point is at γ (β/2). Notice that the β parameter represents the bandwidth of the 0.5 points.**

$$\Pi(x;6,7)=\begin{cases} S(x;1,4,7) \text{ for x} \le 7 \\ 1-S(x;7,10,13) \text{ for x} \ge 7 \end{cases}$$

P-Function

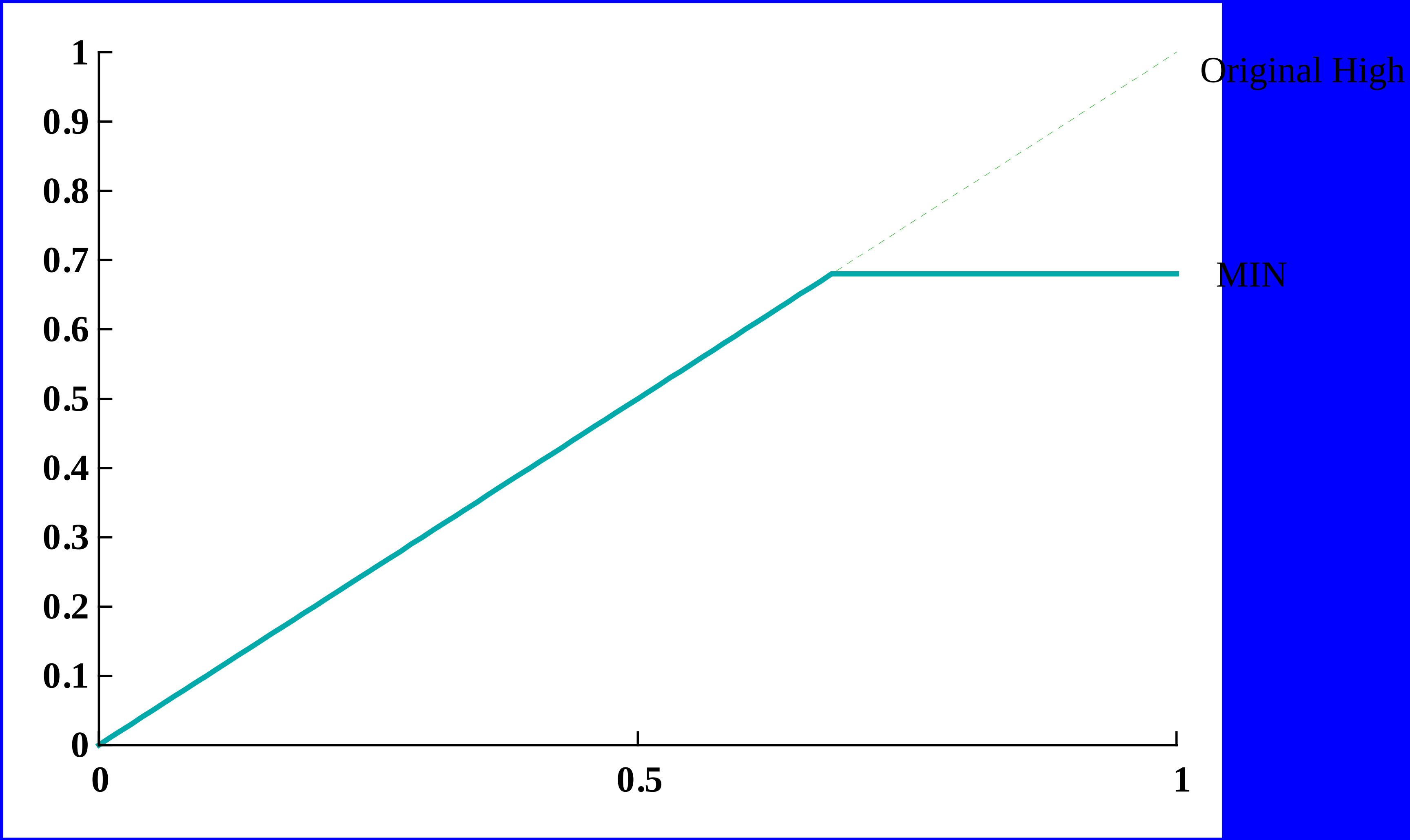


Popular Fuzzy Inference Methods

Popular Fuzzy Inference Methods

- Under the fuzzy inference process the grade of each premise is found and applied to the conclusion part of each rule.
  - MIN or PRODUCT are two popular inference methods.
  - With MIN inferencing the output of the conclusion part is clipped off at a height equal to the rule's degree of firing. This was used in the simple example before and the MIN inference for the first rule is re-shown below (Recall that the rule's degree of firing was 0.68):

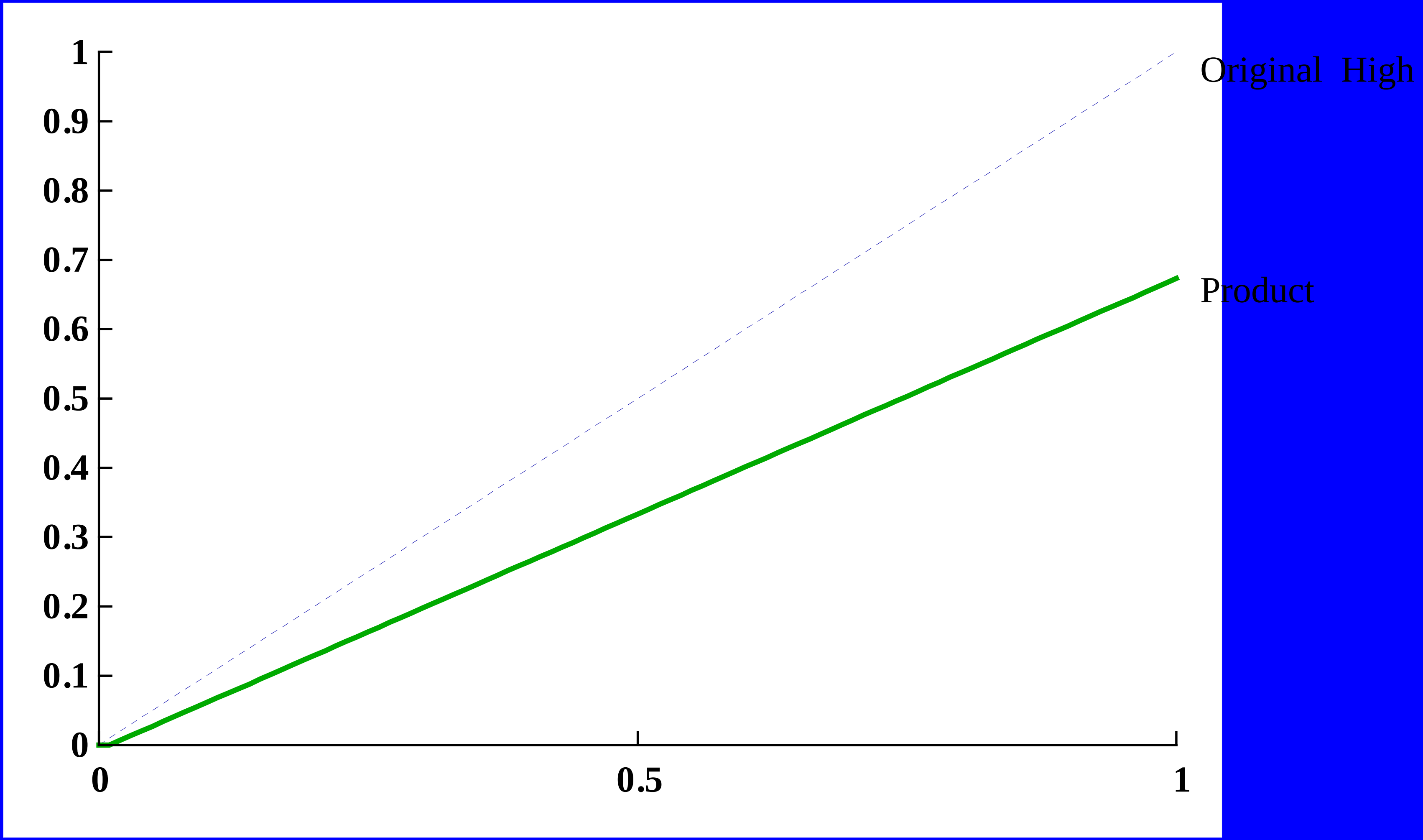
MIN



PRODUCT

- With PRODUCT inferencing the output membership function is scaled by the rule's degree of firing.
- Using the first rule of the simple example and the PRODUCT inference we get the following

Product

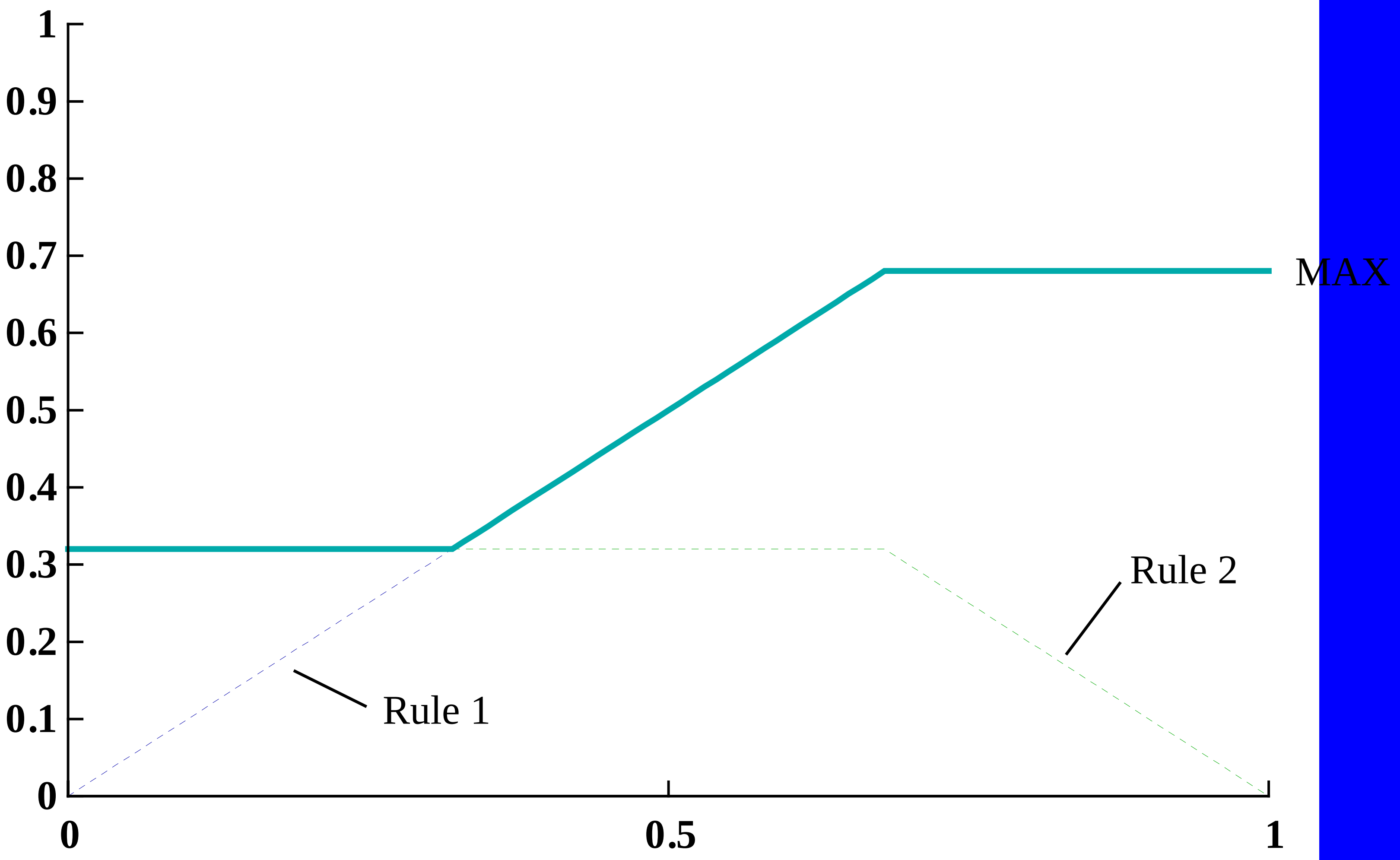


Popular methods for combining outputs

Popular methods for combining outputs

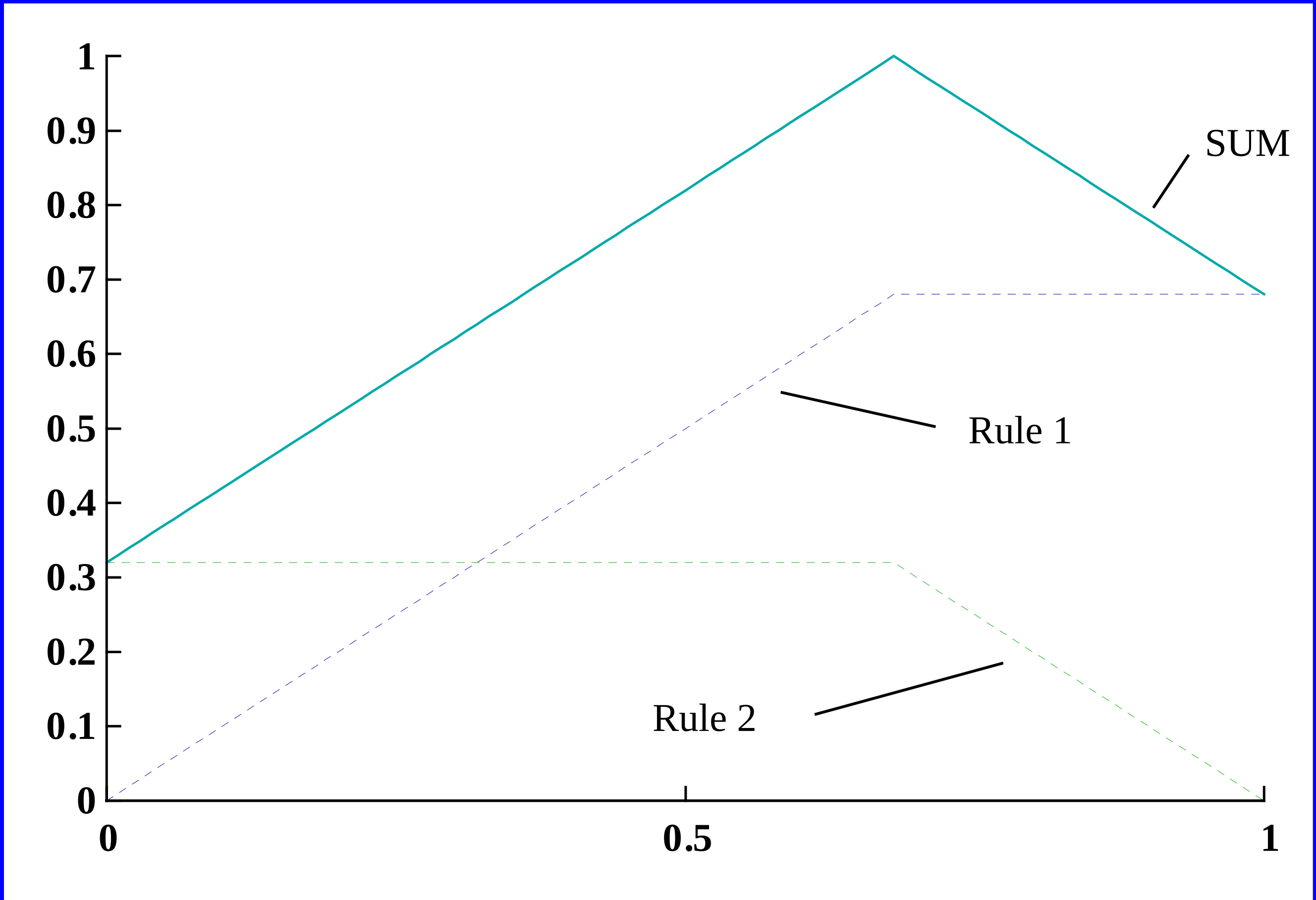
- Under the combining outputs process all outputs are ***combined together*** to produce a single fuzzy output.
  - MAX and SUM are two popular techniques for combining outputs.
  - With MAX the pointwise maximum for all fuzzy outputs is taken.
  - For the first rule of the simple example we used the MAX techniques which resulted in the following diagram:

Max



SUM

- With SUM the pointwise sum for all fuzzy outputs is taken.
- Again for the first rule of the simple example we get:



Popular defuzzification methods

- Under the defuzzification process a fuzzy output is converted to a crisp number.
  - Two of the more popular techniques are the MAXIMUM and the CENTROID.
  - With MAXIMUM one selects the maximum value of the fuzzy output as the crisp value.
  - There are several variations to the MAXIMUM theme.
    - One such variation is the AVERAGE-OF-MAXIMA which was used in the simple example and gave us a crisp value of 0.84.

- With the CENTROID method, the center of gravity of the fuzzy output gives the crisp value.
- Using PRODUCT inferencing and the SUM combination the CENTROID results in a crisp value of 0.56.

Example of calculation of Maximum

$$\frac{\sum_{i=1}^{n} \text{fuzzy point} * \text{strength of point}}{\sum_{i=1}^{n} \text{fuzzy point}}$$

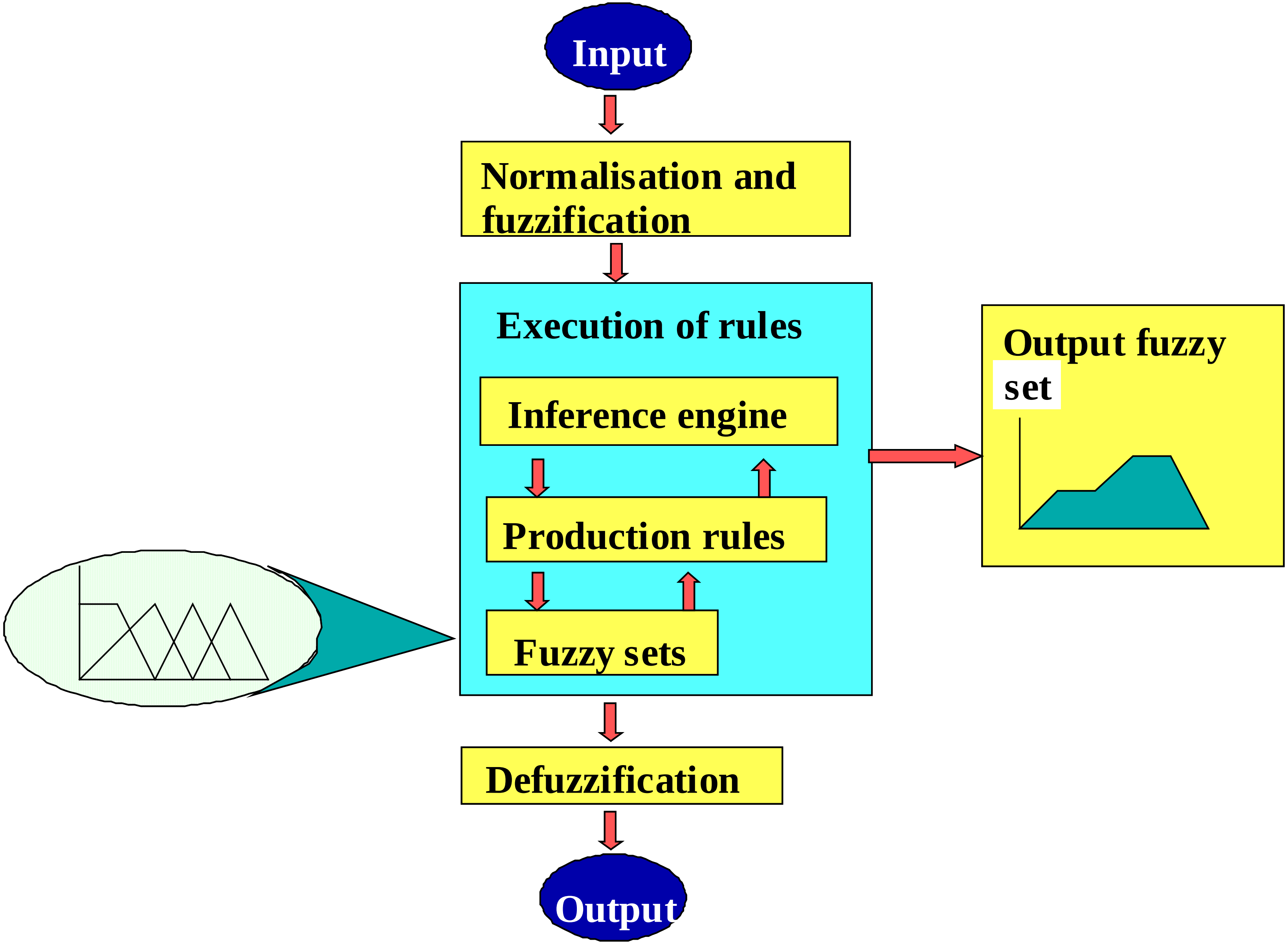
where *n* is the total number of fuzzy points (strips).

STATIC, ADAPTIVE, SELF-ORGANISING SYSTEMS

STATIC, ADAPTIVE, SELF-ORGANISING SYSTEMS

- Up to now we have seen that fuzzy logic supports the task of designing a control system.
  - What we examined in previous sections is known as the `static' fuzzy system.
  - In such a system, conventional techniques are employed in order to elicit or induce production rules.
  - Such a system, receives inputs which are then normalised and converted to fuzzy representations, the knowledge base is executed, an output fuzzy set is generated which is then *converted to a crisp value*.

static' fuzzy system.



Input

Normalisation and fuzzification

Execution of rules

Inference engine

Production rules

Fuzzy sets

Output fuzzy set

Defuzzification

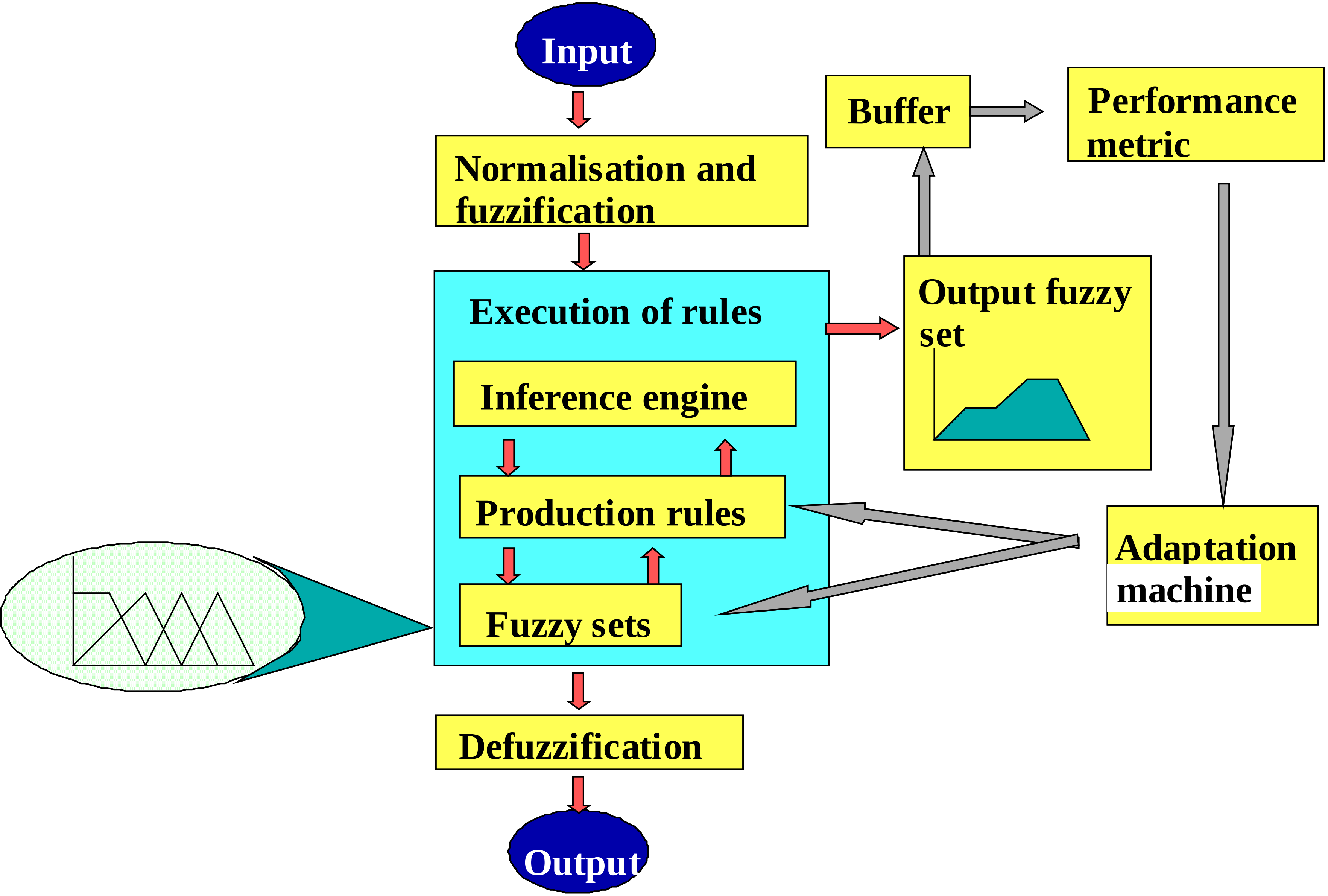
Output

STATIC, ADAPTIVE, SELF-ORGANISING SYSTEMS

- The rules do not change, except if modified by the hand of the designer, for all the knowledge base lifetime.
- Static systems are fine for applications in which the environment is known and predictable.
  - But because they can not adapt to gradual changes in their environments they can lead to disaster when the assumptions upon which they are built are violated.

- An adaptive system adjusts to time or process conditions.
- This means that such a system modifies:
  - the charactersitics of the rules,
  - the topology of the fuzzy sets, and
  - the method of defuzzification
  - among others.

An adaptive system



STATIC, ADAPTIVE, SELF-ORGANISING SYSTEMS

- A performance metric, often another expert system or an algorithm, compares the current and the stored array of past solutions and the comparison result is passed to an adaptation machine.
- The adaptation machine, often another expert system, decides what changes to make in the underlying fuzzy model.
  - For instance:
    - the contribution weights connected with each rule can be modified or
    - the membership functions re-drawn.

Summary

- Fuzzy logic captures intuitive, human expressions.
- Fuzzy sets, statements, and rules are the basis of control.
- The technique is extremely powerful, and appears in mills at a growing rate.
- It is combined with other methods and is the base of soft computing
- Used much in Intelligent Robotics

Sources

- **Paul and Mildred Burkey**
- **Weilin Pan**
- **Xuekun Kou**
- **Kevin Morris Marler**
- Dr Dimitris Tsaptsinos

Kingston University, Mathematics

http://www.kingston.ac.uk/~ma_s435

ma_s435@kingston.ac.uk