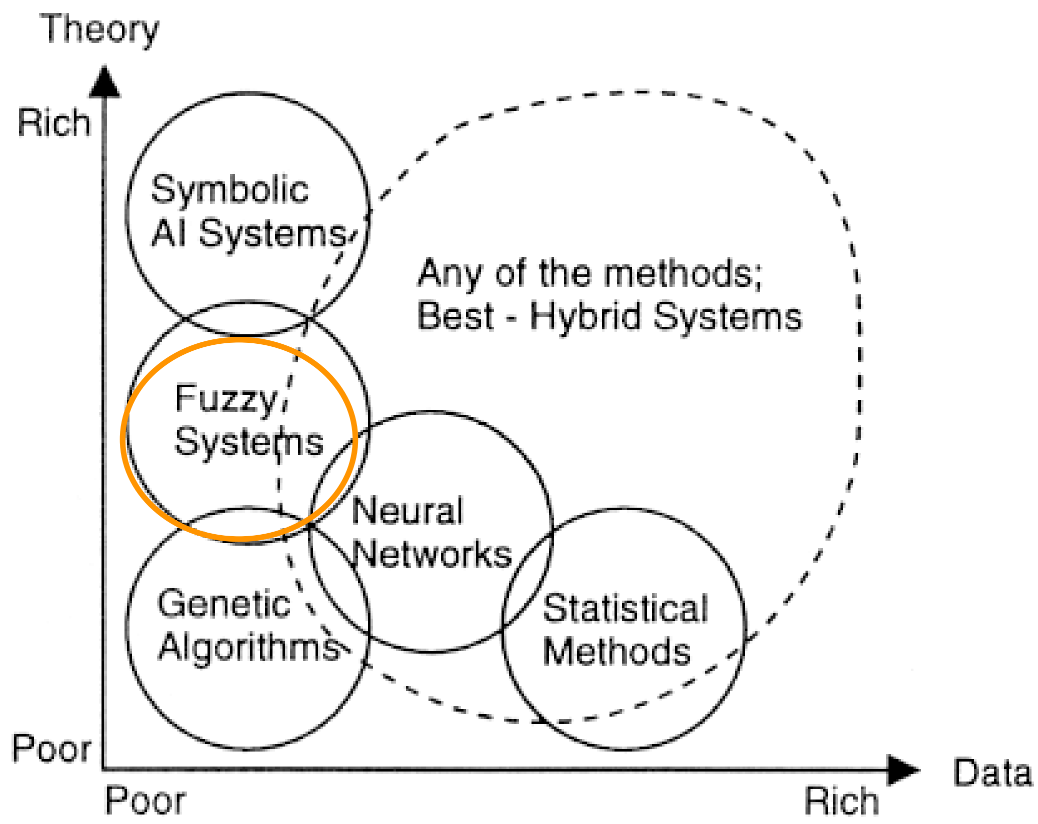
What is all the *Fuzz* about?

Fuzzy Systems: Introduction

CPSC 533

Christian Jacob
Dept. of Computer Science
Dept. of Biochemistry & Molecular Biology
University of Calgary

Fuzzy Systems in Knowledge Engineering



Fuzzy Systems

1. **Motivation**
2. Fuzzy Sets
3. Fuzzy Numbers
4. Fuzzy Sets and Fuzzy Rules
5. Extracting Fuzzy Models from Data
6. Examples of Fuzzy Systems

What does Fuzzy Logic mean?

- Fuzzy logic was introduced by Lotfi Zadeh (UC Berkeley) in 1965.
- Fuzzy logic is based on fuzzy set theory, an extension of classical set theory.
- Fuzzy logic attempts to formalize "approximate" knowledge and reasoning.
- Fuzzy logic did not attract any attention until the 1980s (fuzzy controller applications).

Fuzzy is Just Human

- Humans primarily use fuzzy terms: *large, small, fast, slow, warm, cold, …*

- We say:

  "If the weather is nice and I have a little time, I will probably go for a hike along the Bow."

- We don't say:

  "If the temperature is above 24 degrees and the cloud cover is less than 10%, and I have 3 hours time, I will go for a hike with a probability of 0.47."

Fuzzy is Economical

- Zadeh: "*Make use of the leeway of fuzziness.*"
- Fuzziness as a principle of economics:
  - Precision is expensive.
  - Only apply as much precision to a problem as necessary.
  - Example: Backing into a parking space
    - How long would it take if we had to park a car with a precision of ±2 mm?

Fuzzy Systems

1. Motivation
2. **Fuzzy Sets**
3. Fuzzy Numbers
4. Fuzzy Sets and Fuzzy Rules
5. Extracting Fuzzy Models from Data
6. Examples of Fuzzy Systems

Basics of Fuzzy Sets

- Example: the set of "young people"

$$young = \{x \in P \mid age(x) \leq 20\}$$

- We can define a characteristic function for this set:

$$\mu_{young}(x) = \begin{cases} 1 & : \quad age(x) \leq 20 \\ 0 & : \quad 20 < age(x) \end{cases}$$

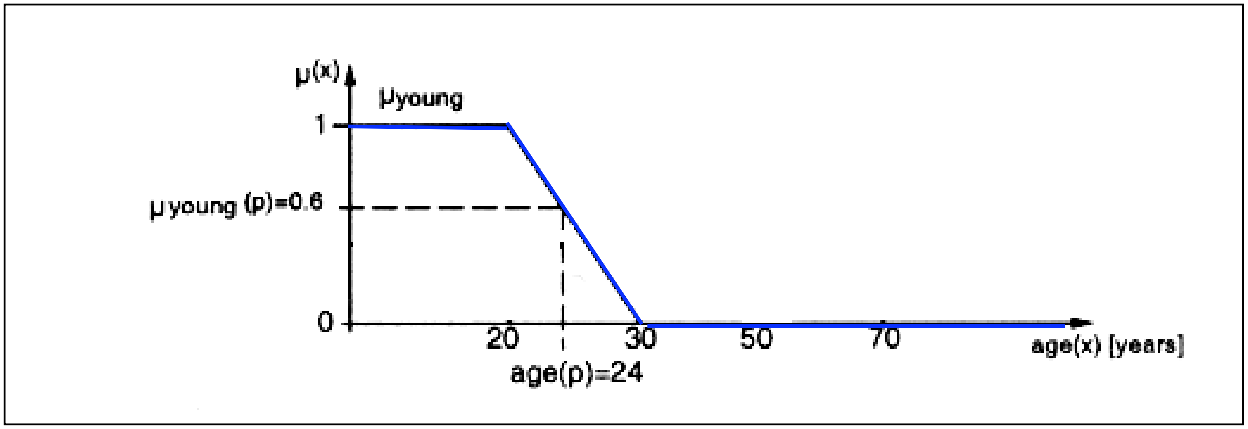
Basics of Fuzzy Sets

- Fuzzy set theory offers a variable notion of *membership*:
  - A person of age 25 could still belong to the set of young people, but only to a degree of less than on, say 0.9.

$$\mu_{young}(x) = \begin{cases} 1 & : \quad age(x) \leq 20 \\ 1 - \frac{age(x) - 20}{10} & : \quad 20 < age(x) \leq 30 \\ 0 & : \quad 30 < age(x) \end{cases}$$

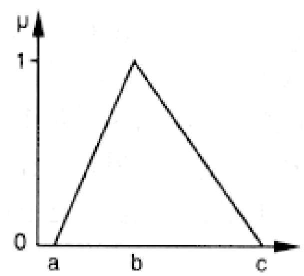
  - Now the set of *young* contains people with ages between 20 and 30, with a linearly decreasing degree of membership.

Fuzzy Membership Function

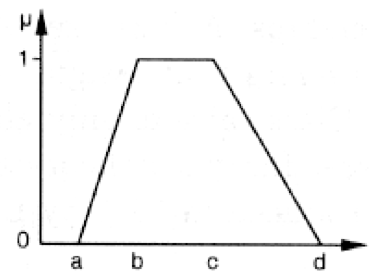


$$\mu_{young}(x) = \begin{cases} 1 & : \quad age(x) \leq 20 \\ 1 - \frac{age(x)-20}{10} & : \quad 20 < age(x) \leq 30 \\ 0 & : \quad 30 < age(x) \end{cases}$$

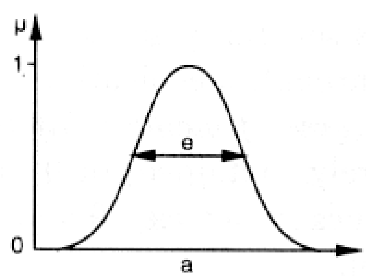
Shapes for Membership Fcts.



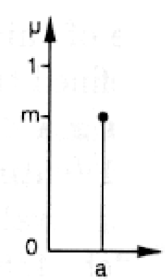
Triangle: [a,b,c]



Trapezoid: [a,b,c,d]



Gaussian: [a,□]



Singleton: [a,m]

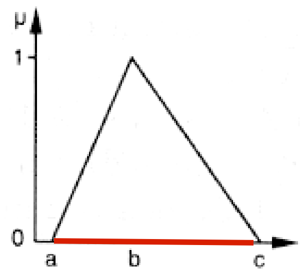
Parameters of FMFs

- **Support**: $s_A := \{ x : \quad _A(x) > 0 \}$
  - The area where the membership function is positive.
- **Core**: $c_A := \{ x : \quad _A(x) = 1 \}$
  - The area for which elements have a maximum degree of membership to the fuzzy set A.

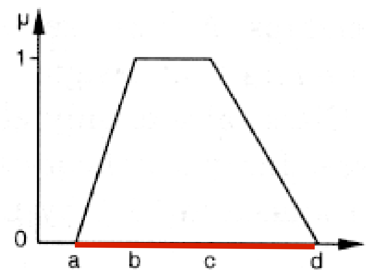
Parameters of FMFs

- **α-Cut**: $\alpha_A := \{ x : \;\; _A(x) = \alpha \}$
  - The cut through the membership function of A at height a.
- **Height**: $h_A := \max_x \{ \;\; _A(x) \}$
  - The maximum value of the membership function of A.

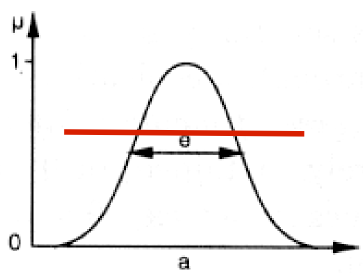
Support: $s_A := \{ x : \quad _A(x) > 0 \}$



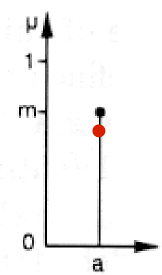
Triangle: [a,b,c]



Trapezoid: [a,b,c,d]

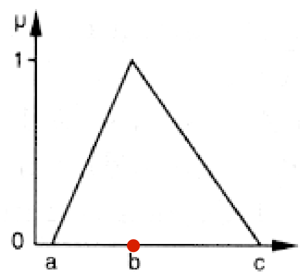


Gaussian: [a,□]

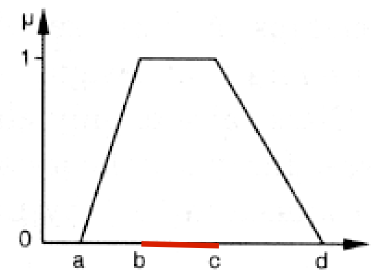


Singleton: [a,m]

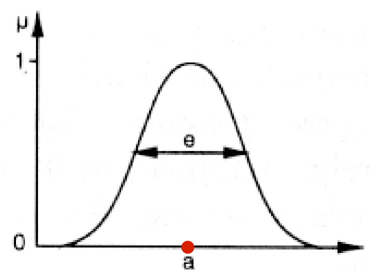
Core: $c_A := \{ x : \quad _A(x) = 1 \}$



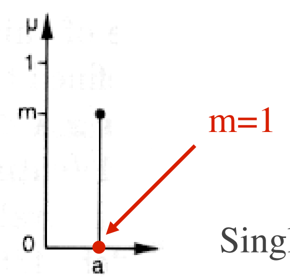
Triangle: [a,b,c]



Trapezoid: [a,b,c,d]

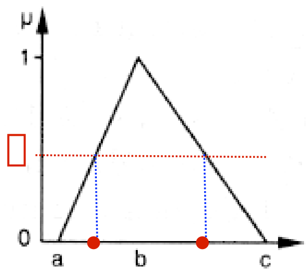


Gaussian: [a,□]

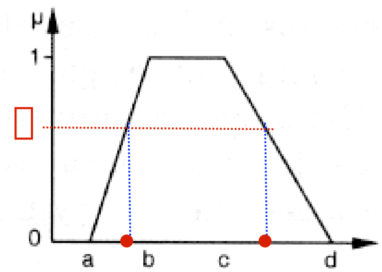


Singleton: [a,m]

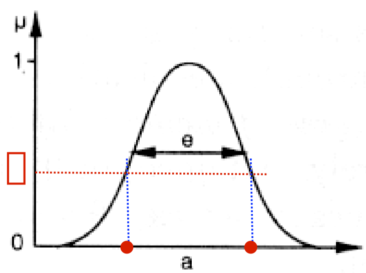
α-Cut: $\alpha_A := \{ x : \quad _A(x) = \alpha \}$



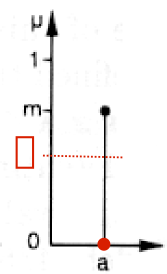
Triangle: [a,b,c]



Trapezoid: [a,b,c,d]

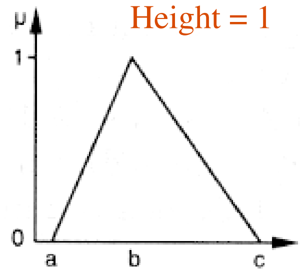


Gaussian: [a,▯]



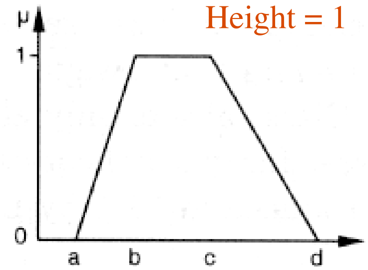
Singleton: [a,m]

Height: $h_A := \max_x \{ \quad _A(x) \}$



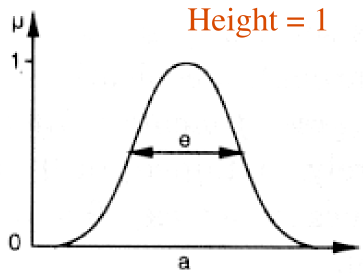
Height = 1

Triangle: [a,b,c]



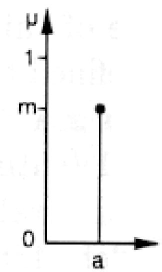
Height = 1

Trapezoid: [a,b,c,d]



Height = 1

Gaussian: [a,□]



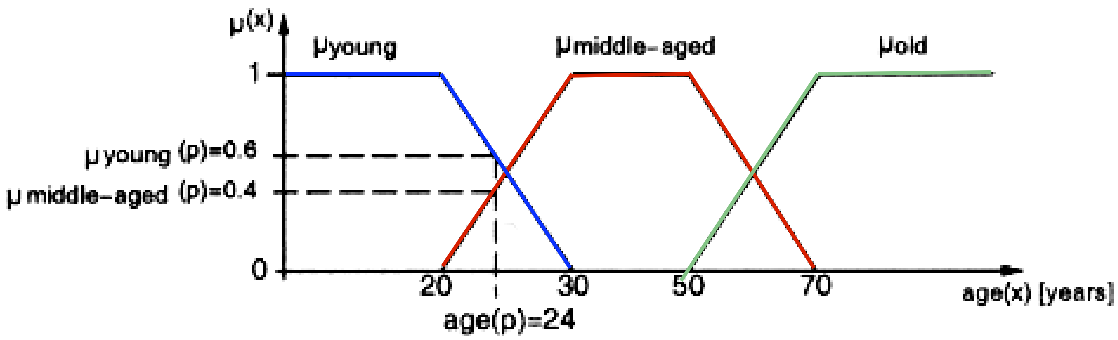
Height = m

Singleton: [a,m]

Linguistic Variables

- The covering of a variable domain with several fuzzy sets, together with a corresponding semantics, defines a *linguistic variable*.

- Example: linguistic variable *age*

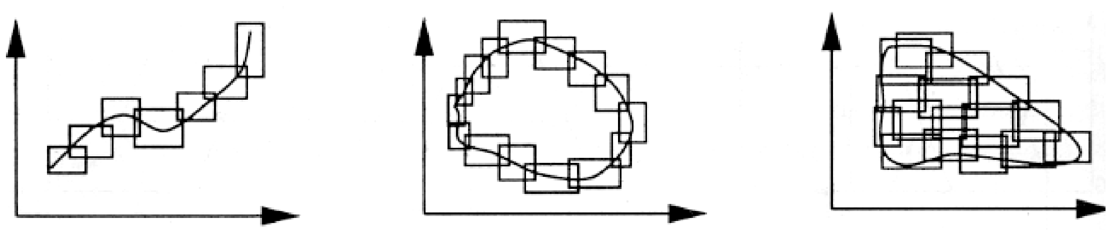


Granulation

- Using fuzzy sets, we can incorporate the fact that no sharp boundaries between 'groups', such as *young*, *middle-aged*, and *old*, exist.
- The corresponding membership functions overlap in certain areas, forming non-crisp (fuzzy) boundaries.
- This compositional way of defining fuzzy sets over a domain of a variable is called *granulation*.

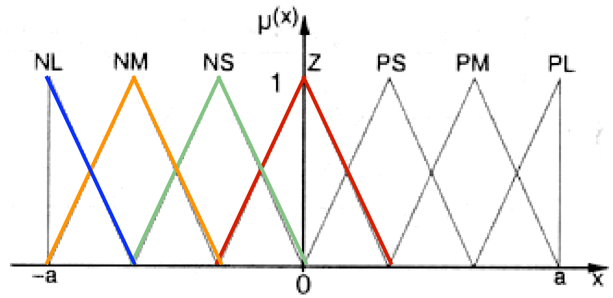
Fuzzy Granules

- Granulation results in a grouping of objects into imprecise clusters of *fuzzy granules*.
- The objects forming a granule are drawn together by similarity.
- This can be seen as a form of fuzzy data compression.



Finding Fuzzy Granules

- If expert knowledge on a domain is not available, an automatic granulation is used.



- Standard granulation using an odd number of membership functions:
    - NL: negative large, NM: negative medium, NS: negative small, Z: zero, ...

Fuzzy Systems

1. Motivation
2. Fuzzy Sets
3. **Fuzzy Numbers**
4. Fuzzy Sets and Fuzzy Rules
5. Extracting Fuzzy Models from Data
6. Examples of Fuzzy Systems

Fuzzy vs. Crisp Numbers

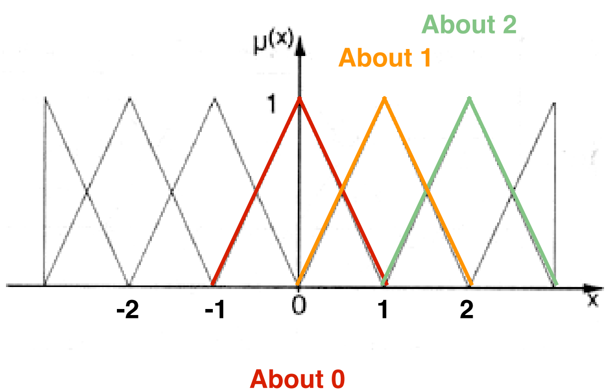
- Real-world measurements are always imprecise.
- Usually, such imprecise measurements are modeled through
  - a crisp number *x*, denoting the most typical value,
  - together with an interval, describing the amount of imprecision.
- In a linguistic sense: "about *x*"

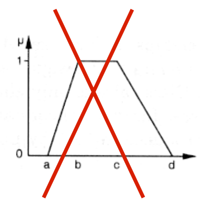
Fuzzy Numbers as Fuzzy Sets

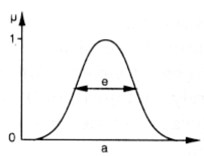
- Fuzzy numbers are a special type of fuzzy sets with specific membership functions:
  - $_{A}$ must be *normalized* ($c_A \neq \emptyset$).
  - $_{A}$ must be *singular*. There is precisely one point which lies inside the core, modeling the typical value (= *modal value*) of the fuzzy number.
  - $_{A}$ must be monotonically increasing left of the core and monotonically decreasing on the right (only one peak!).

Fuzzy Number Example

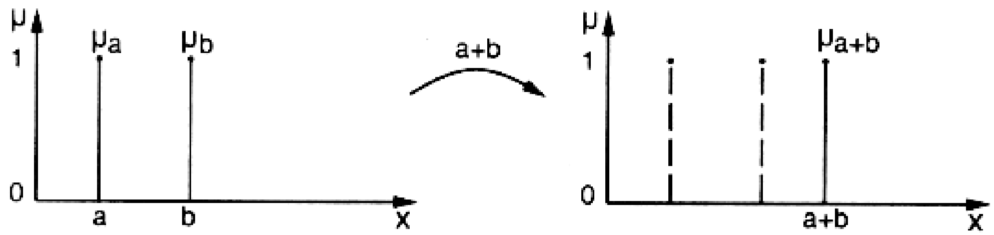
- Typically, triangular membership functions are chosen for fuzzy numbers.







Addition: Crisp Numbers



$$\mu_{a+b}(x) = \begin{cases} 1 & : \quad if\ \exists\, y,z \in R : \mu_a(y) = 1 \wedge \mu_b(z) = 1 \wedge y+z = x \\ 0 & : \quad else \end{cases}$$

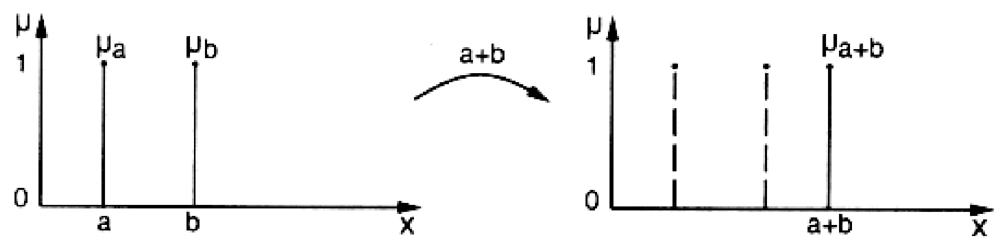
Operations on Fuzzy Numbers

- We extend classical operators (addition, multiplication, etc.) to their fuzzy counterparts, such that we can also handle intermediate degrees of membership.
- For an arbitrary binary operator $\otimes$:

$$\mu_{A\otimes B}(x) = max_{y,z\in}\ \{min\{\mu_A(y), \mu_B(z)\} \mid y \otimes z = x\}$$

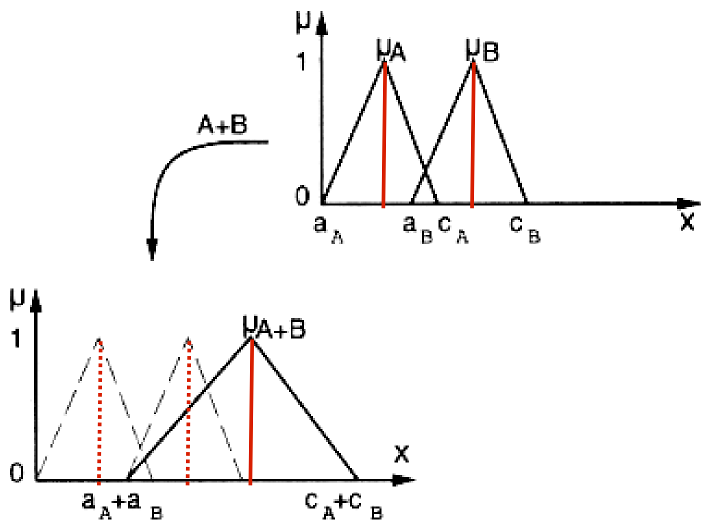
Fuzzy Addition: Crisp Numbers

- We check whether the fuzzy addition is consistent with ‘normal’ addition on crisp numbers.



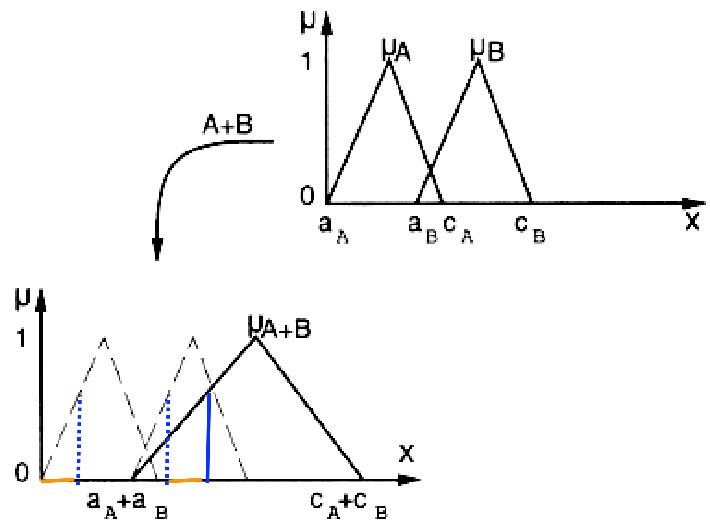
$$\mu_{A \otimes B}(x) = max_{y,z \in} \; \{min\{\mu_A(y), \mu_B(z)\} \mid y \otimes z = x\}$$

Fuzzy Addition



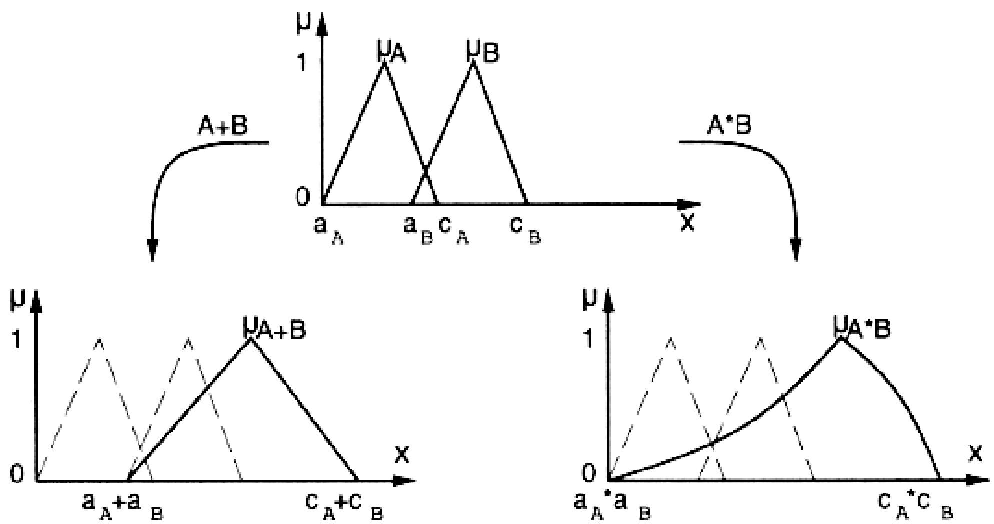
$$\mu_{A \otimes B}(x) = max_{y,z \in} \ \{min\{\mu_A(y), \mu_B(z)\} \mid y \otimes z = x\}$$

Fuzzy Addition



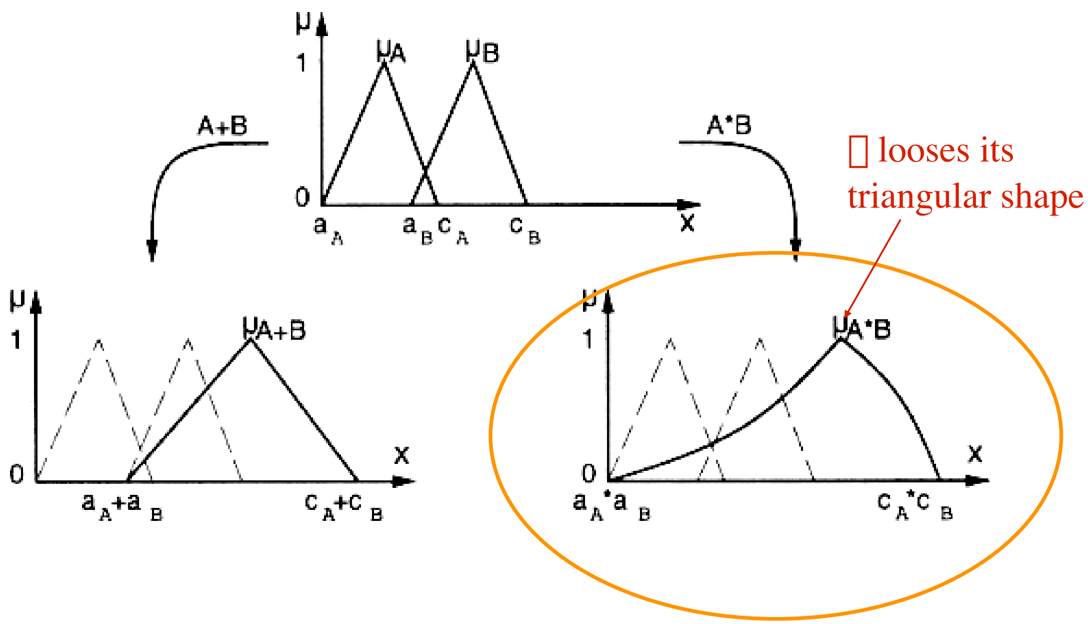
$$\mu_{A \otimes B}(x) = max_{y,z \in} \ \{min\{\mu_A(y), \mu_B(z)\} \mid y \otimes z = x\}$$

Fuzzy Addition & Multiplication



$$\mu_{A \otimes B}(x) = max_{y,z \in} \ \{min\{\mu_A(y), \mu_B(z)\} \mid y \otimes z = x\}$$

Fuzzy Addition & Multiplication



$$\mu_{A \otimes B}(x) = max_{y,z \in} \ \{min\{\mu_A(y), \mu_B(z)\} \mid y \otimes z = x\}$$

Fuzzy Number Operations

- For practical purposes, we can calculate the result of applying a <u>monotonical</u> operation $\otimes$ on fuzzy numbers as follows:
  - Subdivide $_A(x)$ and $_B(x)$ into monotonically increasing and decreasing parts.
  - Perform the operation $\otimes$ jointly on the increasing (decreasing) parts of numbers A and B.
  - Plateaus can be dealt with in a single computation step.

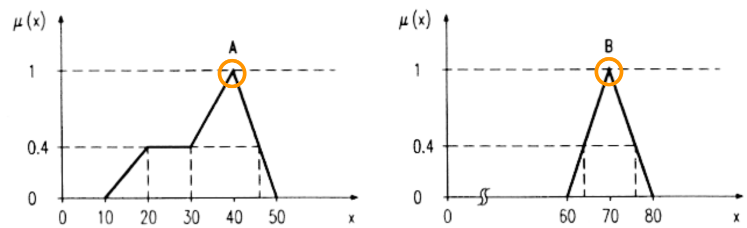
Fuzzy Number Operations

- Let A and B be fuzzy numbers, and $\otimes$ a strongly monotonical operation.
- Let $[a_1, a_2]$ and $[b_1, b_2]$ be the intervals in which ${}_A(x)$ and ${}_B(x)$ are monotonically increasing (decreasing).
- If there exist subintervals $[\alpha_1, \alpha_2] \subseteq [a_1, a_2]$ and $[\beta_1, \beta_2] \subseteq [b_1, b_2]$, such that
  - $\forall x_A \in [\alpha_1, \alpha_2], \forall x_B \in [\beta_1, \beta_2]: \mu_A(x_A) = \mu_B(x_B) = \lambda$

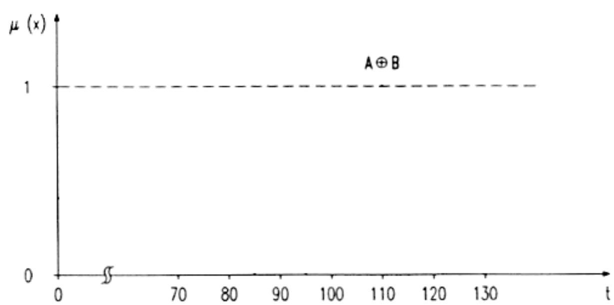
then

  - $\forall t \in [\alpha_1 \otimes \beta_1, \alpha_2 \otimes \beta_2]: \mu_{A \otimes B}(t) = \lambda$.

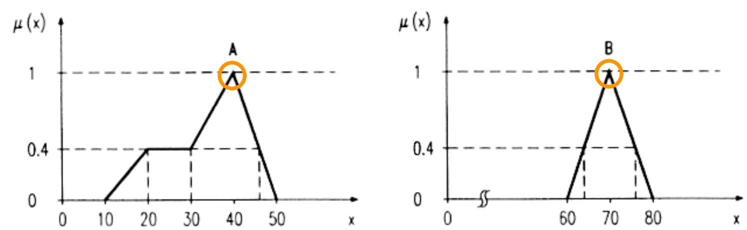
Fuzzy Addition



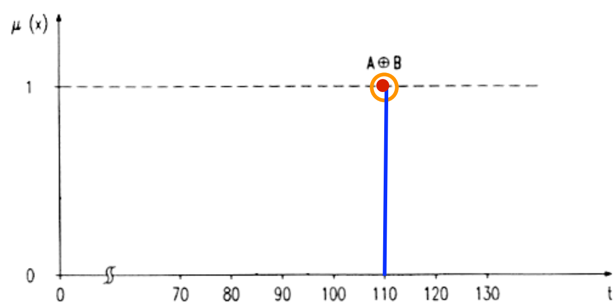
$\mu_{A+B}(t) = 1.0 \quad t = \ldots$



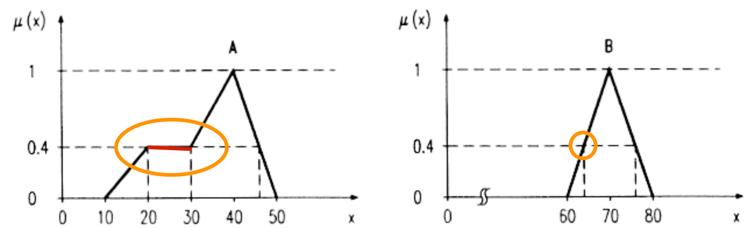
Fuzzy Addition



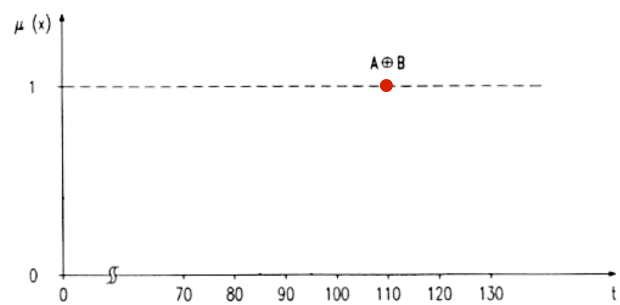
$\mu_{A+B}(t) = 1.0 \quad t = 40 + 70 = 110$



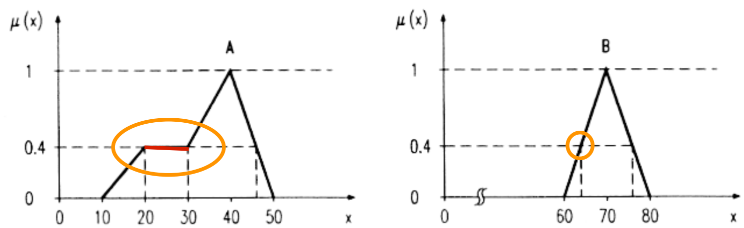
Fuzzy Addition



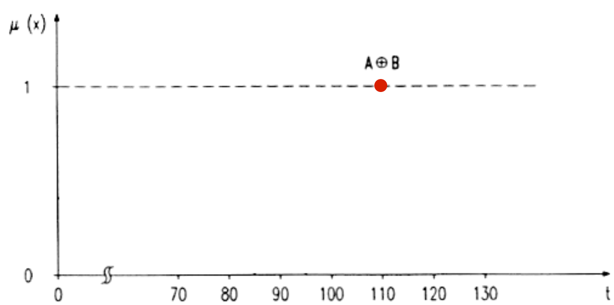
$\square_{A + B}(t) = 0.4 \quad \square \; t \; \square \; [\ldots, \ldots]$



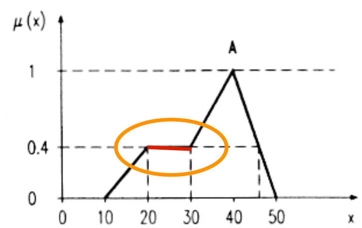
Fuzzy Addition

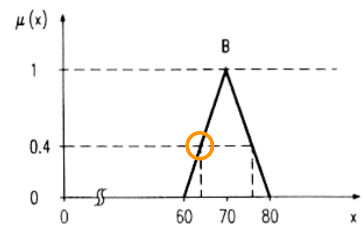


$\mu_{A+B}(t) = 0.4 \quad \Leftrightarrow \quad t \in [20+64, 30+64]$

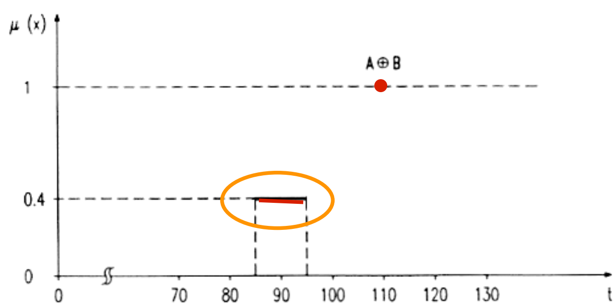


Fuzzy Addition

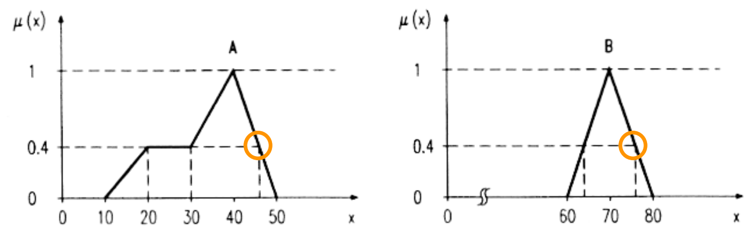




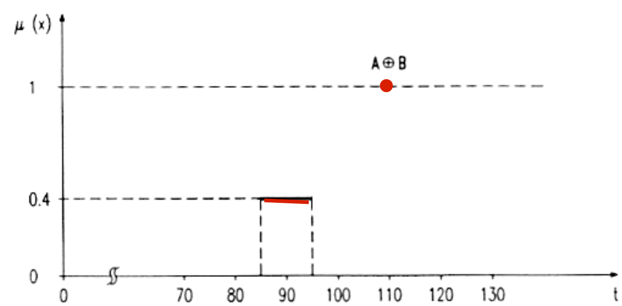
$$\mu_{A+B}(t) = 0.4 \Leftrightarrow t \in [20+64, 30+64] = [84, 94]$$



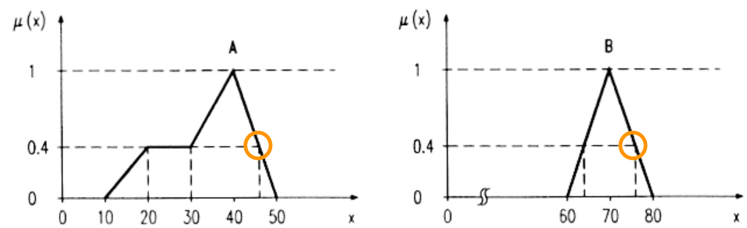
Fuzzy Addition



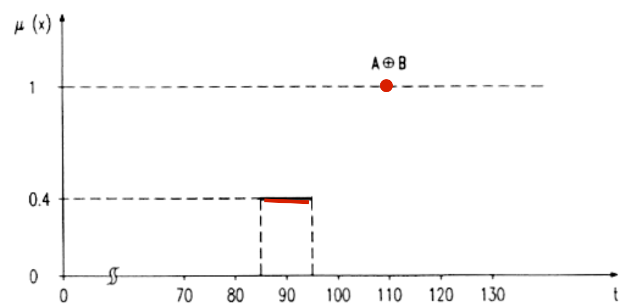
$\mu_{A+B}(t) = 0.4 \quad t = \ldots$



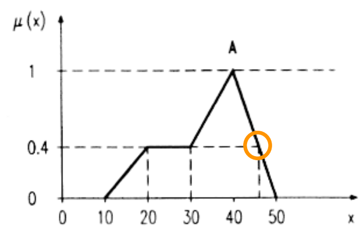
Fuzzy Addition

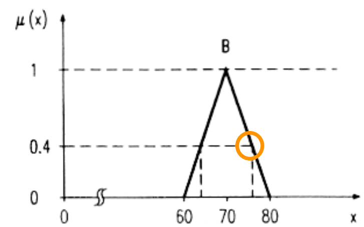


$\mu_{A+B}(t) = 0.4 \quad t = 46 + 76 = 122$

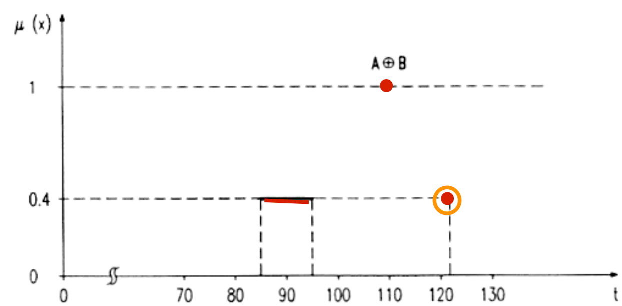


Fuzzy Addition

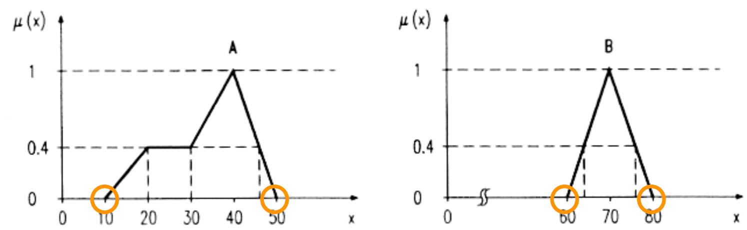




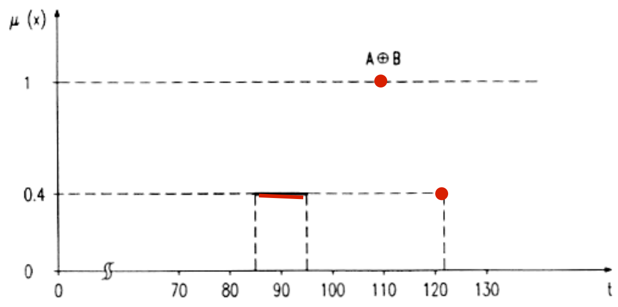
$\mu_{A+B}(t) = 0.4 \quad t = 46 + 76 = 122$



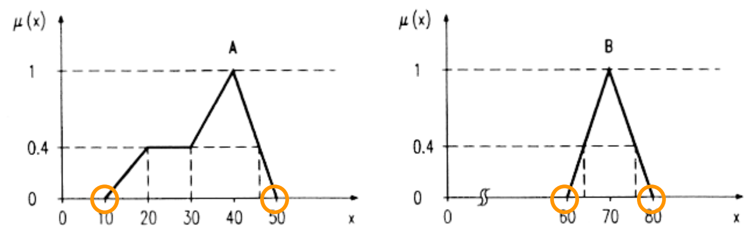
Fuzzy Addition



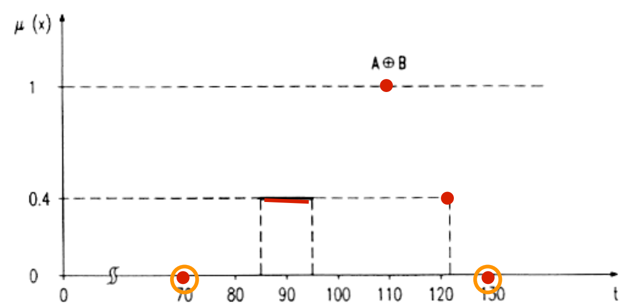
Support: $s_{A\,+\,B}$ = [..., ...]



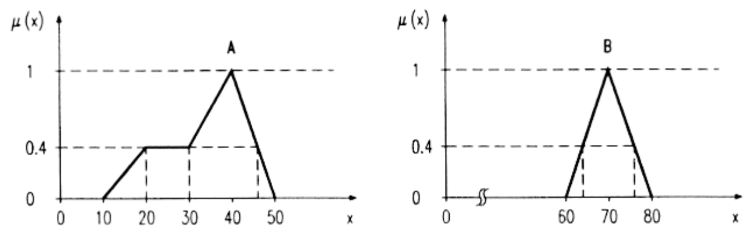
Fuzzy Addition



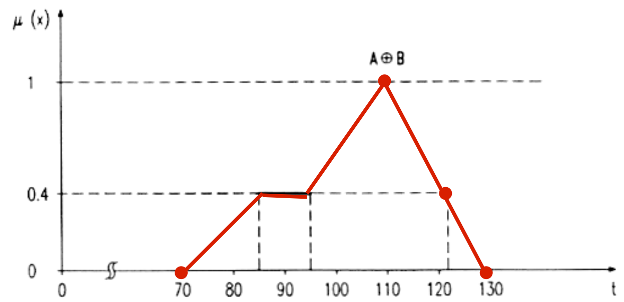
Support: s_{A+B} = [10+60, 50+80] = [70, 130]



Fuzzy Addition

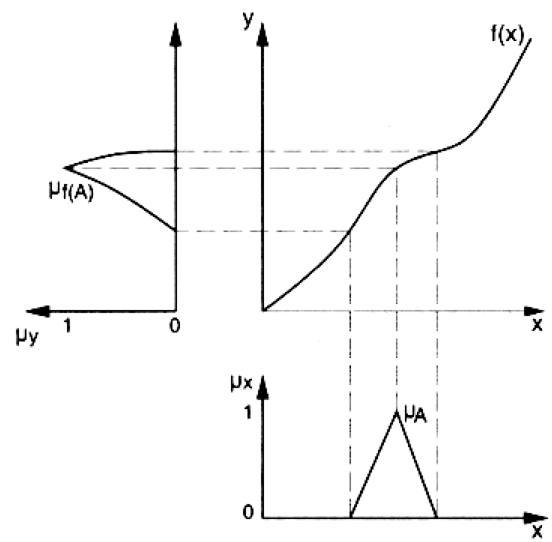


… and the rest by linear interpolation:



Done!

Monotonic Function



$$\mu_{f(A)}(y) = max\{\mu_A(x) \mid \forall x : f(x) = y\}$$

Fuzzy Systems

Operations on Fuzzy Sets

- Let A and B be fuzzy sets.
  - **Intersection** (conjunction):

    $$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}$$

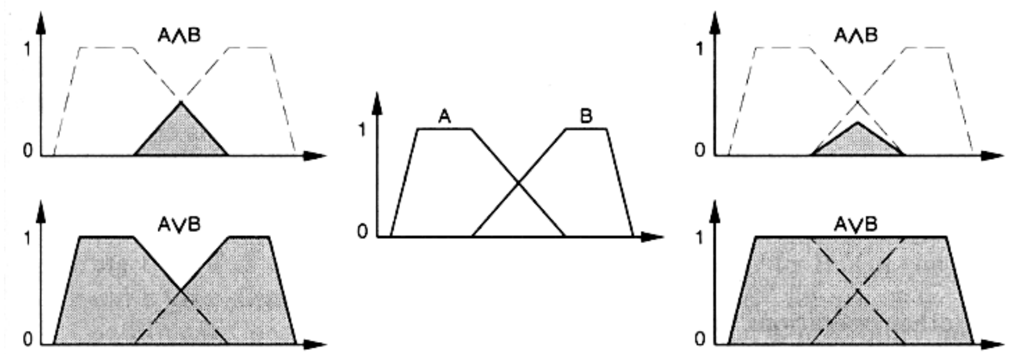
  - **Union** (disjunction):

    $$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}$$

  - **Complement**:

    $$\mu_{\neg A}(x) = 1 - \mu_A(x)$$

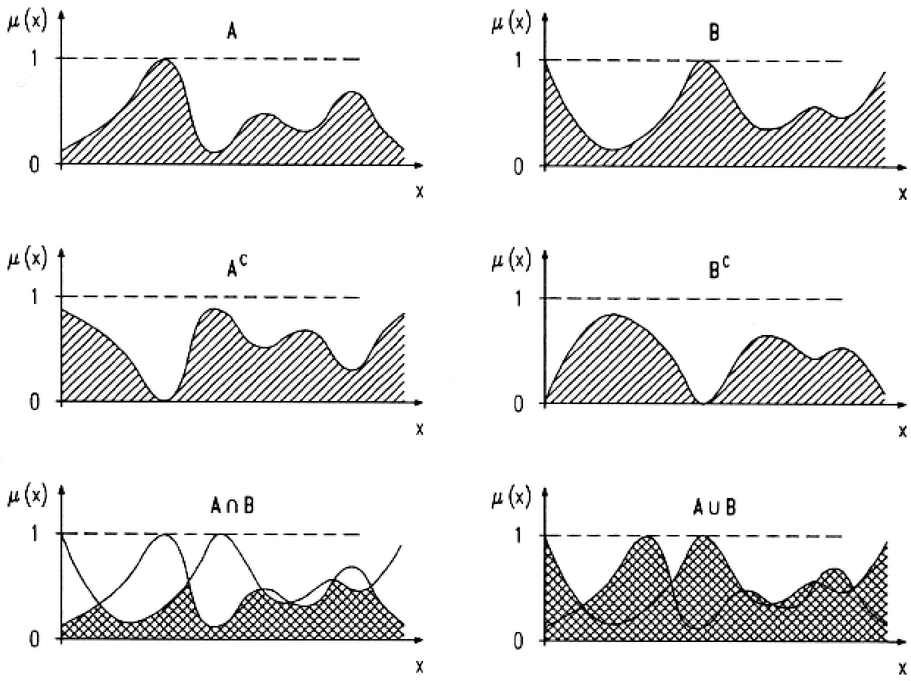
Fuzzy Union & Intersection



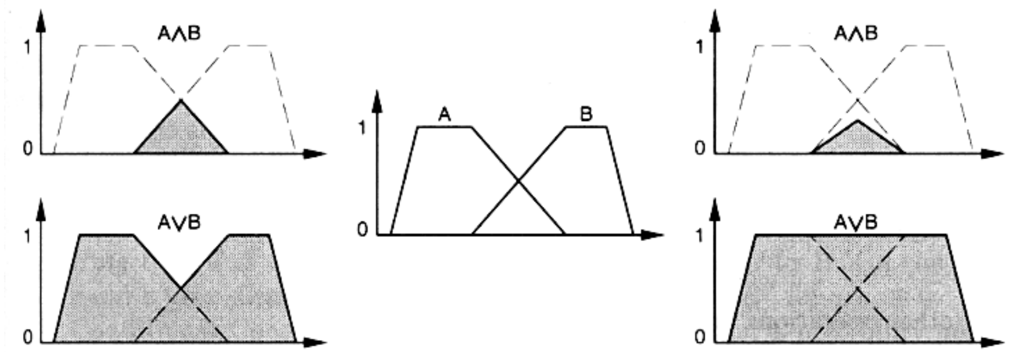
$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}$$

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}$$

Fuzzy Union & Intersection



Union & Intersection: Alternatives



$$\mu_{A\cap B}(x) = min\{\mu_A(x), \mu_B(x)\}$$

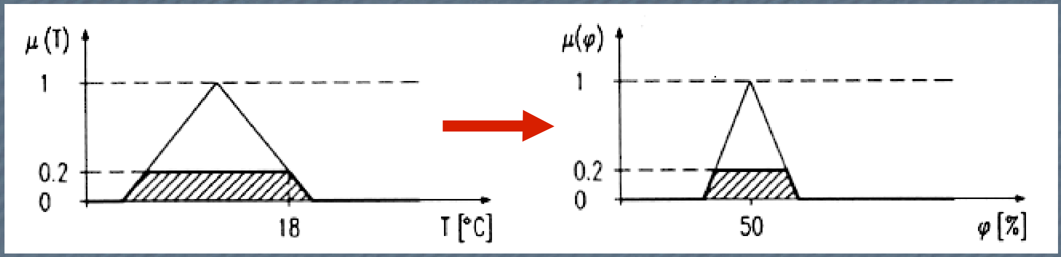
$$\mu_{A\cap B}(x) = \mu_A(x) \cdot \mu_B(x)$$

$$\mu_{A\cup B}(x) = max\{\mu_A(x), \mu_B(x)\}$$

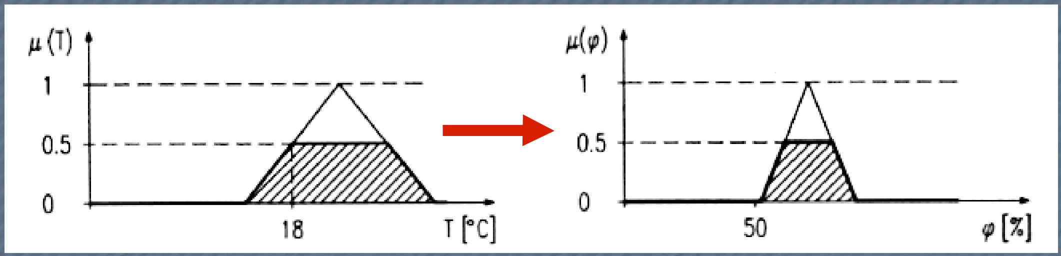
$$\mu_{A\cup B}(x) = min\{\mu_A(x) + \mu_B(x), 1\}$$

Fuzzy Rules

IF *temperature* = *low* THEN *cooling valve* = *half open.*



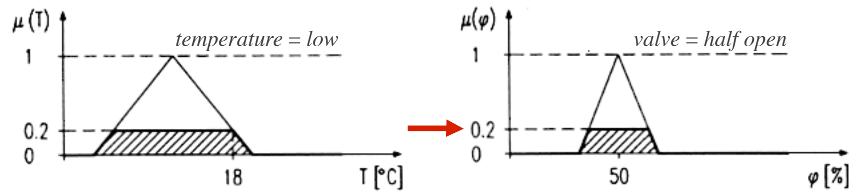
IF *temperature* = *medium* THEN *cooling valve* = *almost open.*



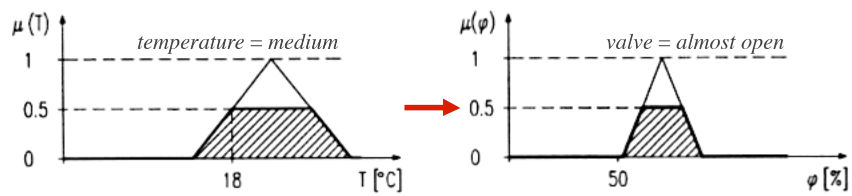
Max-Min Inference

Max-Min Inference

Approximate Reasoning



IF *temperature* = *low* THEN *cooling valve* = *half open*



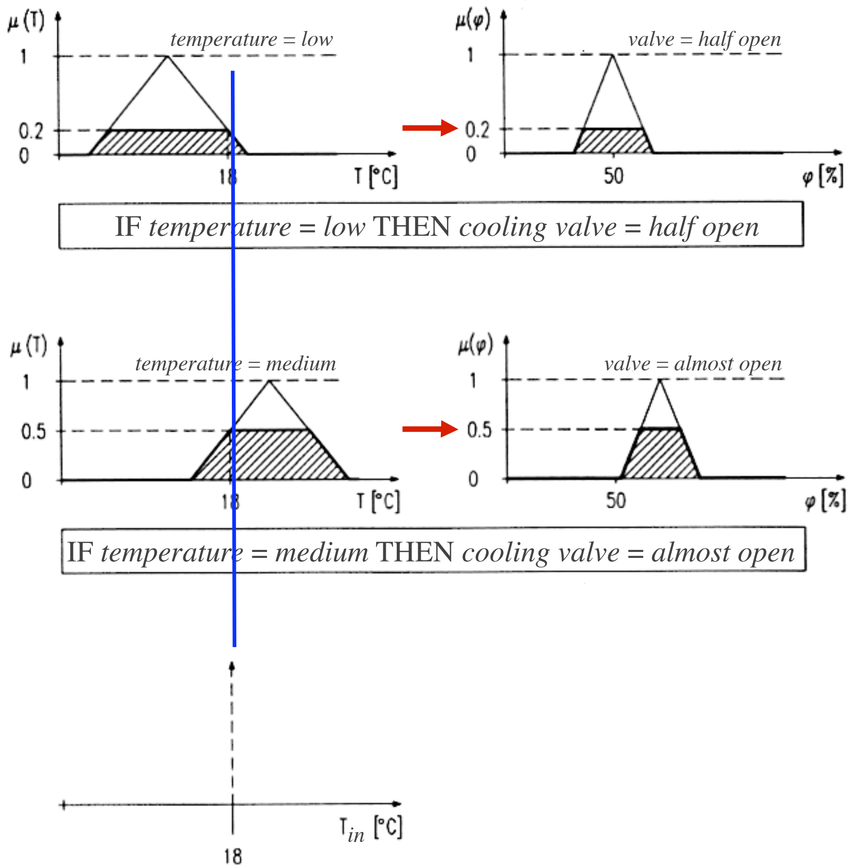
IF *temperature* = *medium* THEN *cooling valve* = *almost open*

Max-Min Inference

Max-Min Inference

Approximate Reasoning

1. Input of crisp value

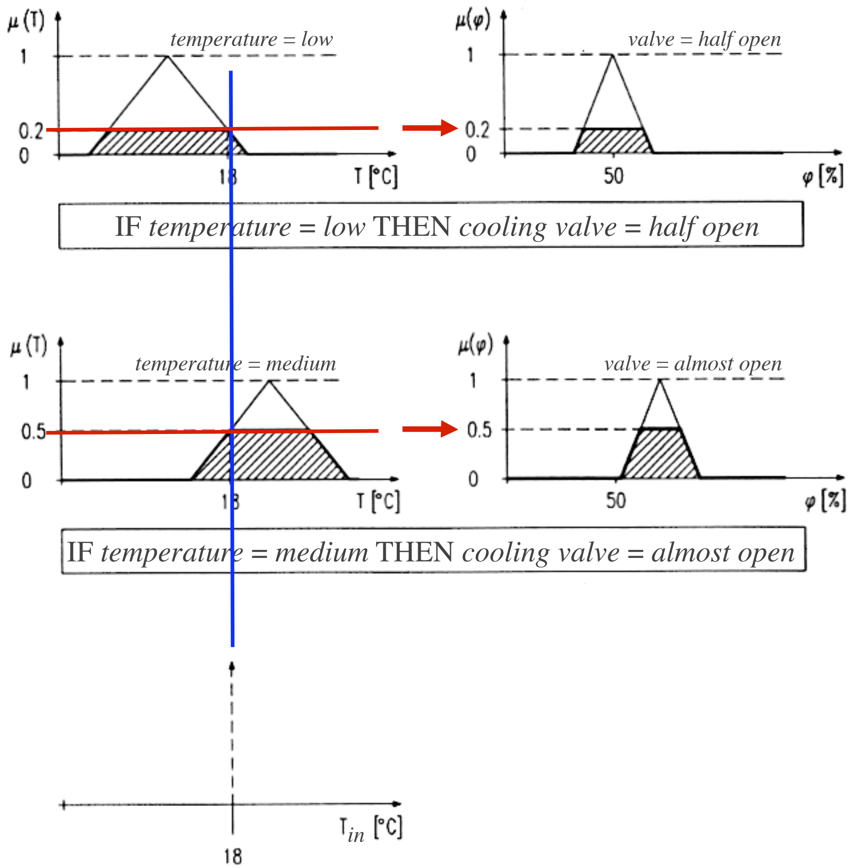


Max-Min Inference

Max-Min Inference

Approximate Reasoning

1. Input of crisp value
2. Fuzzification

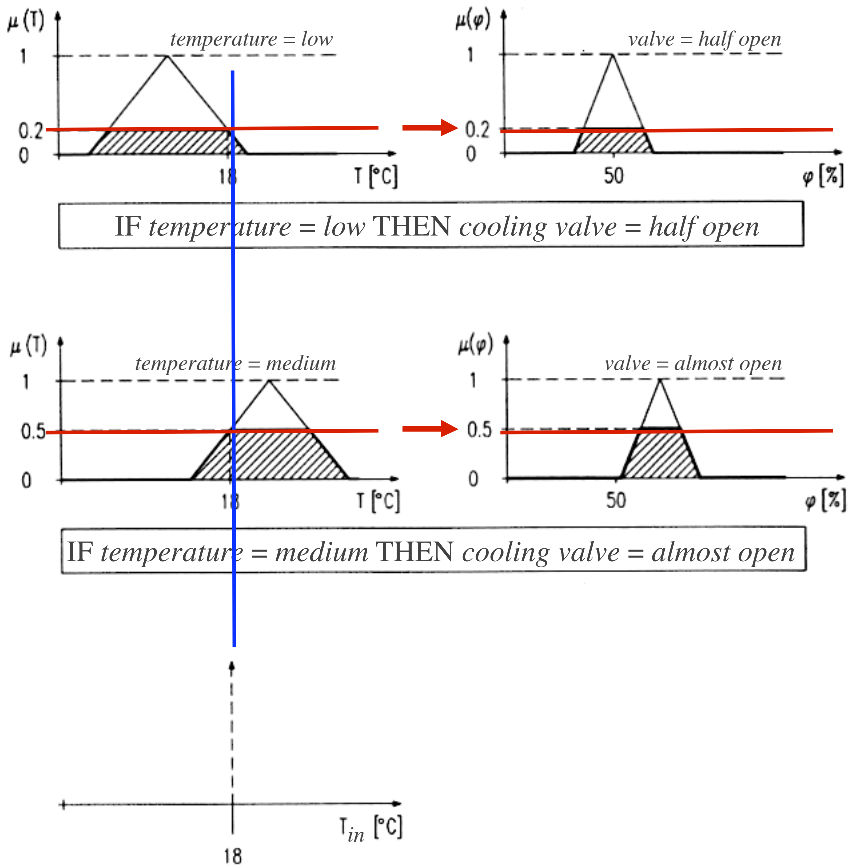


Max-Min Inference

Max-Min Inference

Approximate Reasoning

1. Input of crisp value
2. Fuzzification
3. Inference

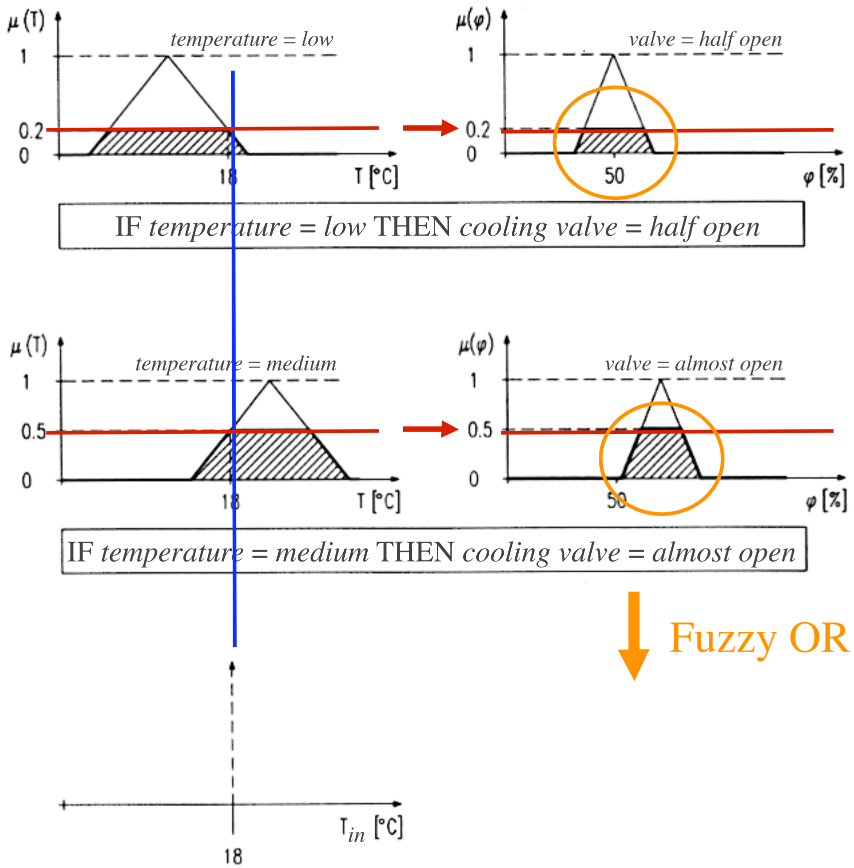


Max-Min Inference

Max-Min Inference

Approximate Reasoning

1. Input of crisp value
2. Fuzzification
3. Inference
4. Output set

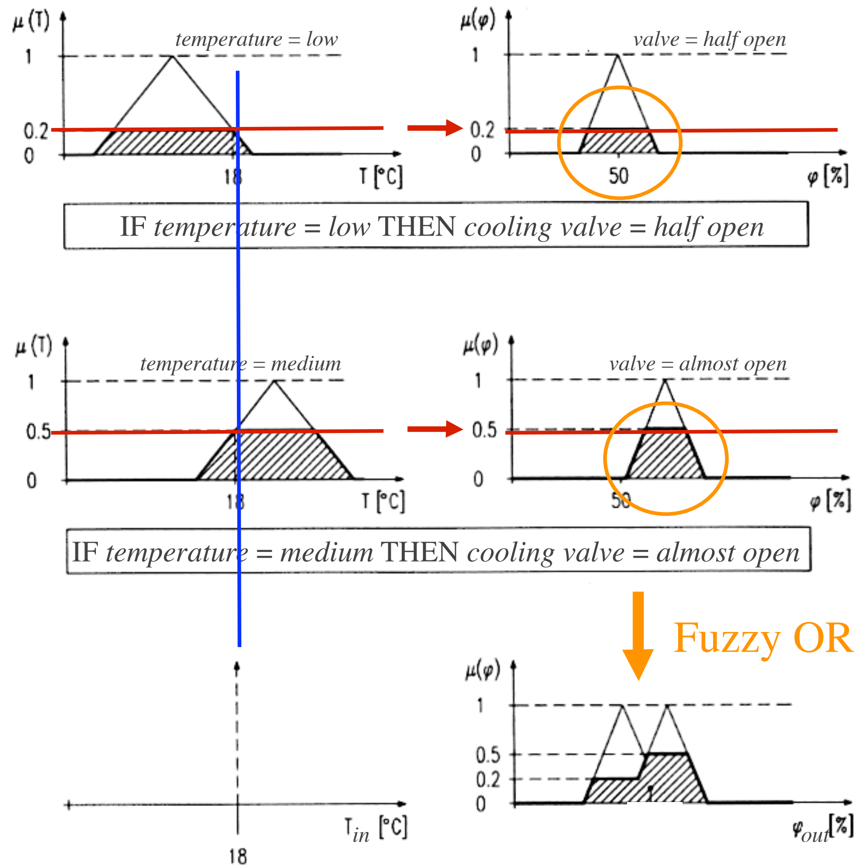


Max-Min Inference

Max-Min Inference

Approximate Reasoning

1. Input of crisp value
2. Fuzzification
3. Inference
4. Output set

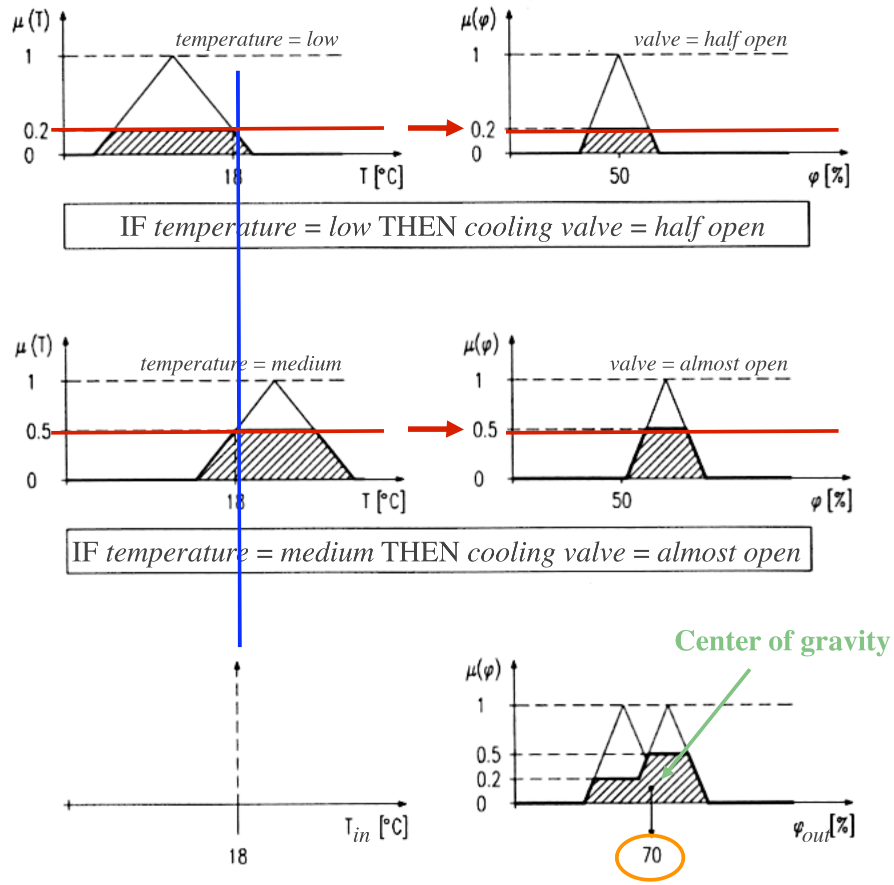


Max-Min Inference

Max-Min Inference

Approximate Reasoning

1. Input of crisp value
2. Fuzzification
3. Inference
4. Output set
5. Defuzzification

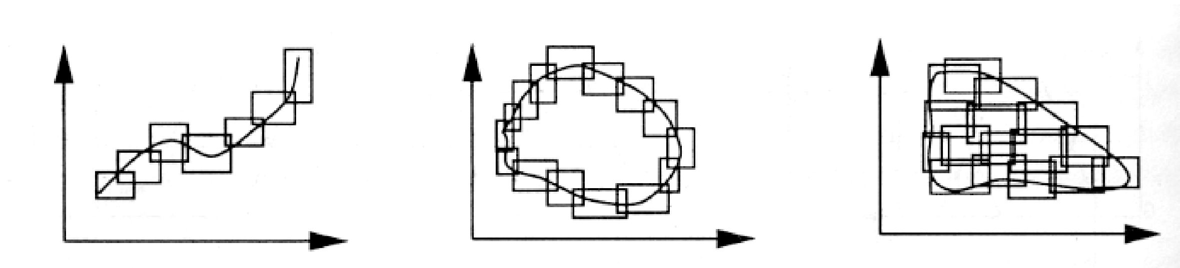


Fuzzy Systems

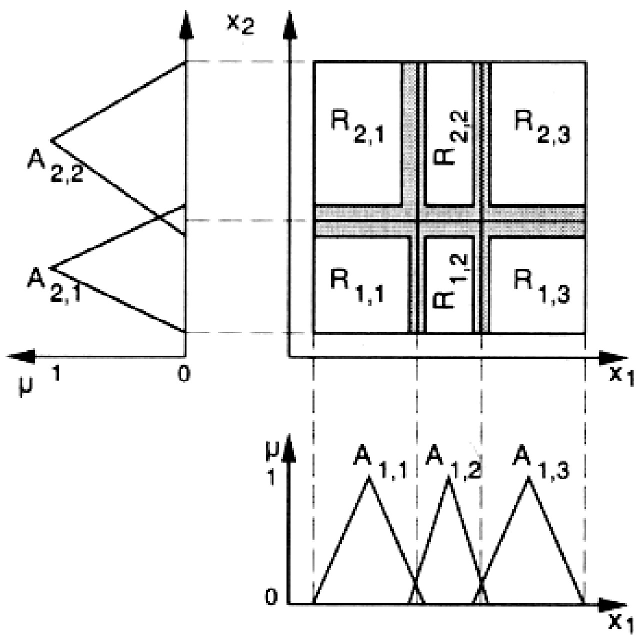
1. Motivation
2. Fuzzy Sets
3. Fuzzy Numbers
4. Fuzzy Sets and Fuzzy Rules
5. **Extracting Fuzzy Models from Data**
6. Examples of Fuzzy Systems

Extracting Fuzzy Models: Graphs

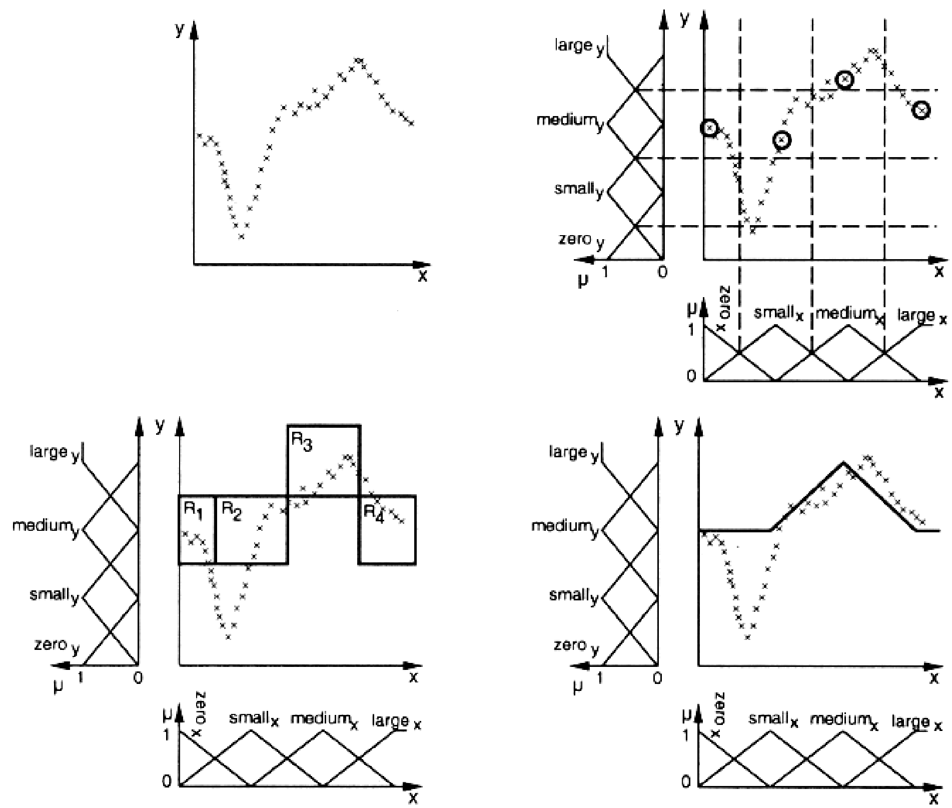
- Approximate representation of functions, contours, and relations



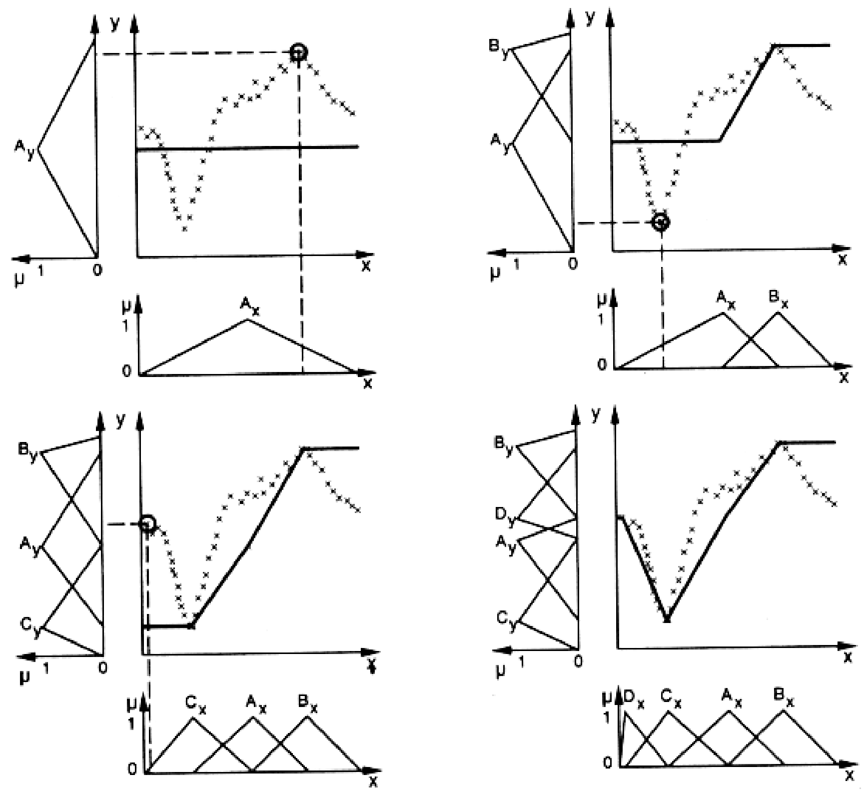
Global Granulation of Input Space



Fixed-Grid Rules Extraction



Adaptive-Grid Rules Extraction



Fuzzy Systems

1. Motivation
2. Fuzzy Sets
3. Fuzzy Numbers
4. Fuzzy Sets and Fuzzy Rules
5. Extracting Fuzzy Models from Data
6. **Examples of Fuzzy Systems**

Logic System

- A *logic system* consists of four parts:
  - Alphabet: a set of basic symbols from which more complex sentences are made.
  - Syntax: a set of rules or operators for constructing expressions (sentences).
  - Semantics: for defining the meaning of the sentences
  - Inference rules: for constructing semantically equivalent but syntactically different sentences

Predicate Logic

- The following types of symbols are allowed in *predicate logic*:
  - Terms
  - Predicates
  - Connectives
  - Quantifiers

Predicate Logic

- Terms:
  - Constant symbols: symbols, expressions, or entities which do not change during excecution (e.g., *true* / *false*)
  - Variable symbols: represent entities that can change during excecution
  - Function symbols: represent functions which process input values on a predefined list of parameters and obtain resulting values

Predicate Logic

- Predicates:
  - Predicate symbols: represent true/false-type relations between objects. Objects are represented by constant symbols.

Predicate Logic

- Connectives:
  - Conjunction
  - Disjunction
  - Negation
  - Implication
  - Equivalence
  - ... (same as for propositional calculus)

Predicate Logic

- Quantifiers:
  - valid for variable symbols
  - Existential quantifier: "There exists at least one value for x from its domain."
  - Universal quantifier: "For all x in its domain."

First-Order Logic

- First-order logic allows quantified variables to refer to objects, but <u>not</u> to predicates or functions.
- For applying an *inference* to a set of predicate expressions, the system has to process <u>matches of expressions</u>.
- The process of matching is called *unification*.

PROLOG
Programming in Logic

- In PROLOG, a quantifier-free *Horn-clausal* notation is adopted.
- $\forall x, \mathit{Human}(x) \Rightarrow \mathit{Mortal}(x)$
  - `mortal(X) :- human(X).`
- *Human*(Socrates)
  - `human(Socrates).`

PROLOG: Horn Clauses

- A rule
  - $(A_1, A_2, \ldots, A_n) \Rightarrow B$
- as a Horn clause has the following form:
  - `B :- A1, A2, ..., An.`
- The goal B is true if all the sub-goals A_1, A_2, ..., A_n are true.
- A_1, A_2, ..., A_n are predicate expressions.

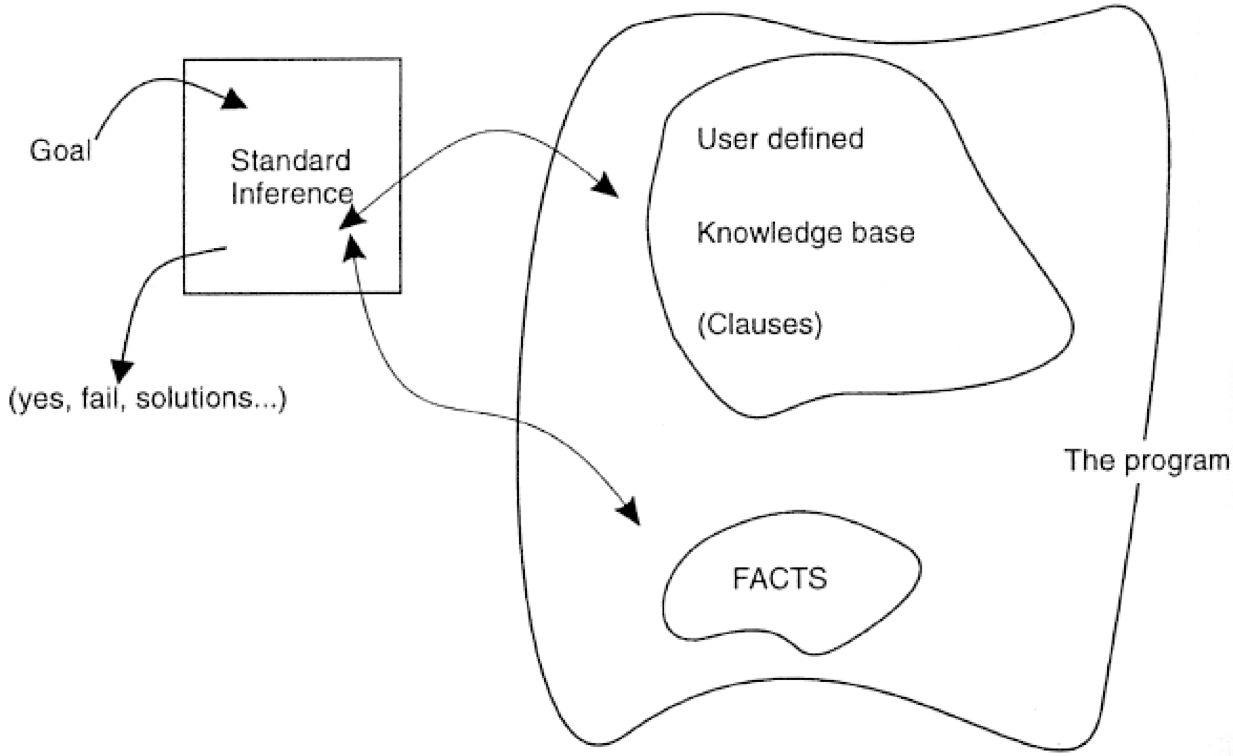
PROLOG: Facts

- A fact is represented as a literal clause with the right side being the constant *true*.
- Therefore, a clause representing a fact is always true.
- Example:
  - `human(Socrates) :- true.`
  - `human(Socrates).`

Knowledge Representation

- In PROLOG knowledge is represented as a set of clauses with
  - *premises* (right side of the clause) and
  - one *conclusion* (left side).
- AND is denoted by the character ",".
- OR is denoted by the character ";".
- NOT is denoted by "¬".

PROLOG Architecture



Family Example: *Facts*

- `father(John, Mary).`
- `father(Jack, Andy).`
- `father(Jack, Tom).`
- `mother(Helen, Jack).`
- `mother(Mary, Jilly).`
- `grandfather(Barry, Jim).`

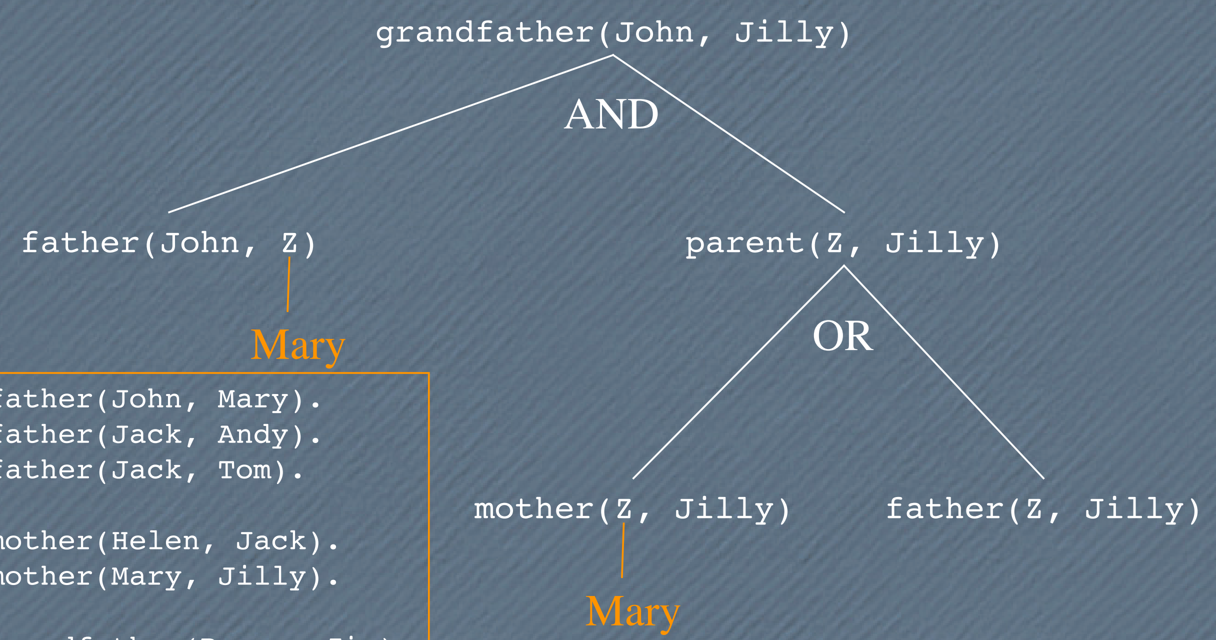
Family Example: *Rules*

```
grandfather(X,Y)
  :- father(X,Z), parent(Z,Y).
grandmother(X,Y)
  :- mother(X,Z), parent(Z,Y).
parent(X,Y) :- mother(X,Y).
parent(X,Y) :- father(X,Y).
```

PROLOG Sample Dialogue

```
Goal?- grandfather(John, Jilly).
  yes
Goal?- grandmother(Helen, X).
  2 solutions
  X = Andy; X = Tom
Goal?- grandmother(X,Y), X = Y.
  fail
```

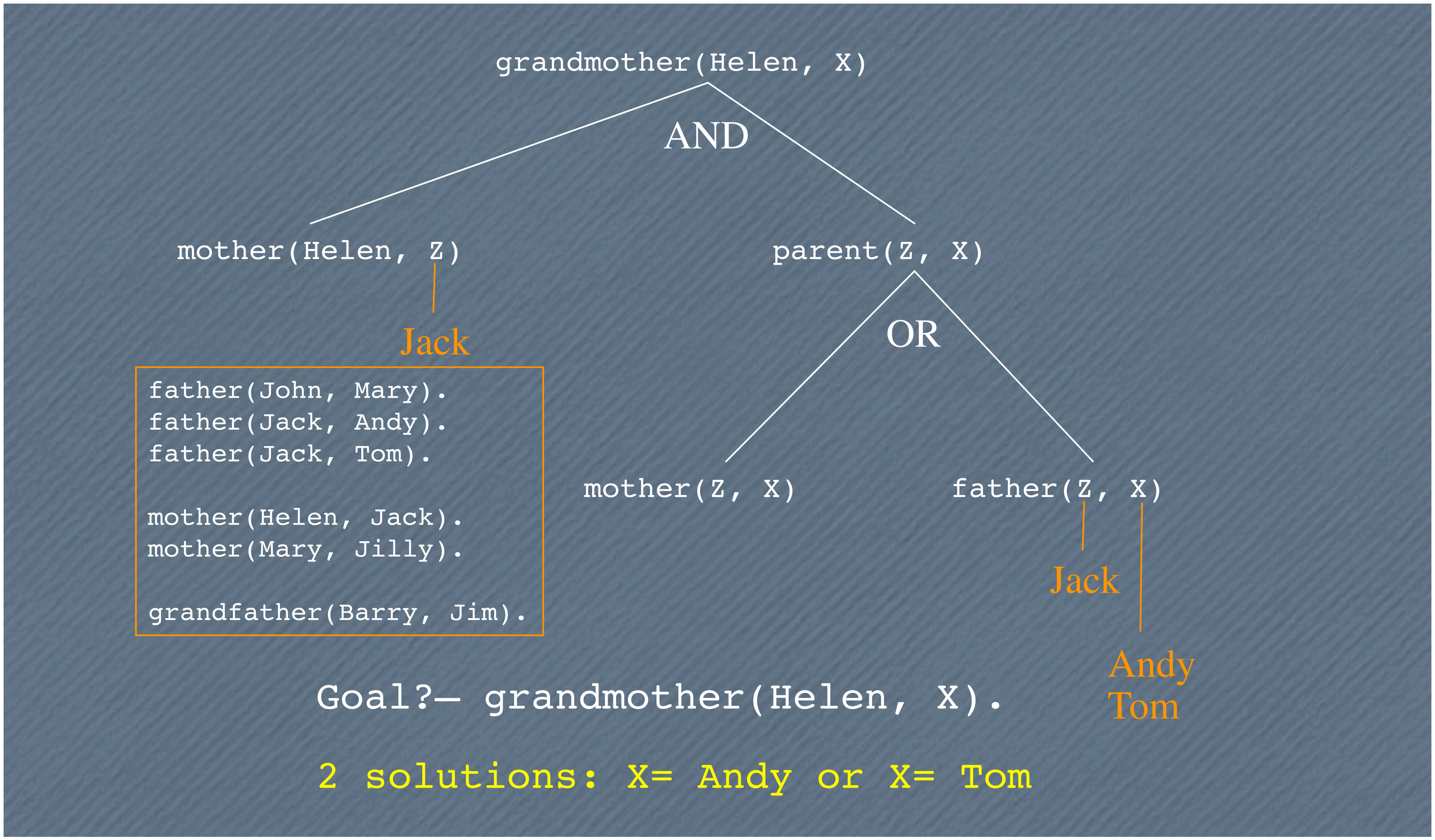
PROLOG Sample Inference




```
father(John, Mary).
father(Jack, Andy).
father(Jack, Tom).

mother(Helen, Jack).
mother(Mary, Jilly).

grandfather(Barry, Jim).
```

```
Goal?– grandfather(John, Jilly).

yes
```

PROLOG Sample Inference



References

- Berthold, M., and Hand, D. J. (2003). *Intelligent Data Analysis*. Berlin, Springer.
- Bothe, H. (1995). *Fuzzy Logic — Einführung in die Theorie und Anwendungen*. Berlin, Springer.
- Kasabov, N. (1996). *Foundations of Neural Networks, Fuzzy Systems, amd Knowledge Engineering*. Cambridge, MA, MIT Press.