Fuzzy expert systems
Synlogy

Project

Dušan Guller

Department of Applied Informatics
Comenius University
Mlynská dolina, 842 48, Bratislava, Slovakia
guller@fmph.uniba.sk

14.02.2021

begin

Plan of the project:

- Introduction to expert systems (*ES*)
- Basic architecture of rule-based *ES*
- Introduction to fuzzy mathematics
- Fuzzy inference
- Fuzzy expert systems (*FES*) - *Synlogy*
- Conclusions

- Expert systems - computer programs, software:
  - available some of the skills of an expert to nonexperts;
  - emulate the reasoning, thinking patterns of an expert;
  - handle uncertainty, vagueness, incompleteness, impreciseness, and inconsistency.
- Human reasoning - not a static process:
  - data - gathered;
  - hypotheses - advanced, tested, rejected;
  - conclusions - reached.
  - The emulation of reasoning must proceed similarly.
- Human knowledge:
  - declarative, facts stored in memory;
  - procedural, skills in utilising declarative knowledge.

- Approaches to the emulation of expert (human) reasoning:
  - rule-based systems [Anderson, 1993];
  - semantic or associative nets [Quillian, 1968];
  - frames [Minsky, 1975];
  - neural nets [Haykin, 1994].
  - Dominant - the complementary rule-based systems and neural net approaches [Jackson, 1999].
- Important expert systems since the middle 1960's:
  - DENDRAL (1965) - determining molecular structure from mass spectrometer data [Feigenbaum, Buchanan, 1993];
  - MYCIN (1976) - medical diagnosis [Shortliffe, 1976];
  - R1 (1980) - computer configurations [McDermott, 1980];
- *ES* - in fields ranging from space shuttle operations, through hospital intensive-care-unit patient monitoring, geology to financial decision making.

PRODUCTION RULES

- A significant part of human (verbal) reasoning - expressed in production rules [Newell, 1972], [Anderson, 1993]:

  IF the data available meet certain specified conditions, *THEN* take these specified actions.

  - Actions - in a wide context, including drawing conclusions, firm or tentative, defeasible.
  - *IF* part of a rule - the *antecedent*, *THEN* part of a rule - the *consequent*.

IF the car engine will not turn over when attempting to start *THEN* check if the battery is discharged

KNOWLEDGE

- Declarative, factual knowledge - represented by stored data.
  Procedural knowledge - represented by production rules.

- The two primary sources of knowledge:
  - the skills of an expert in the field,
  - available historical data.

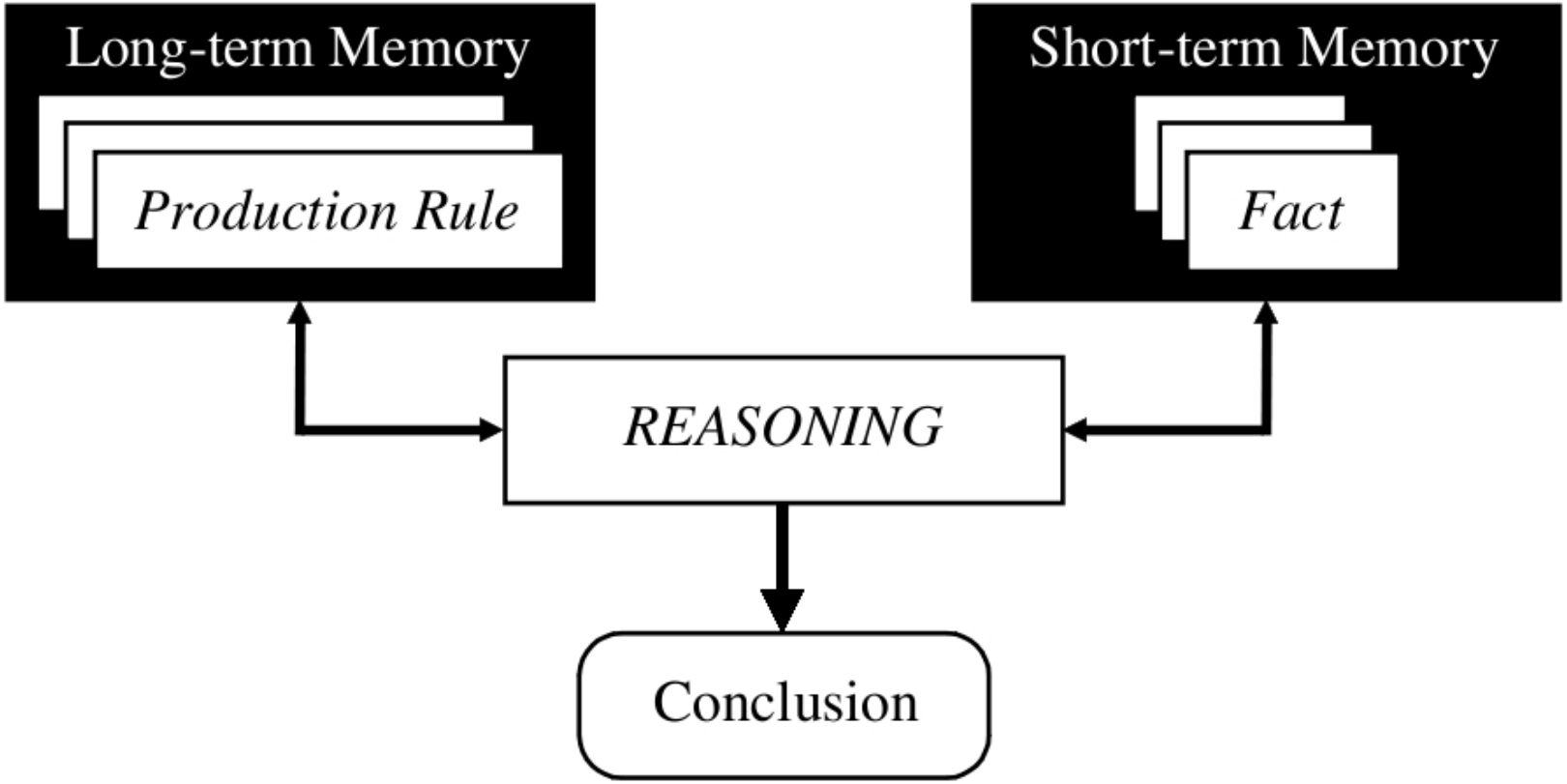
- Rule-based expert systems:
  - much on the skills of an expert in the problem domain,
  - little on historical data.

- Rule-based systems acquire
  - knowledge by adding new facts,
  - skills by adding new rules.
  - Both the learning techniques neccessary.

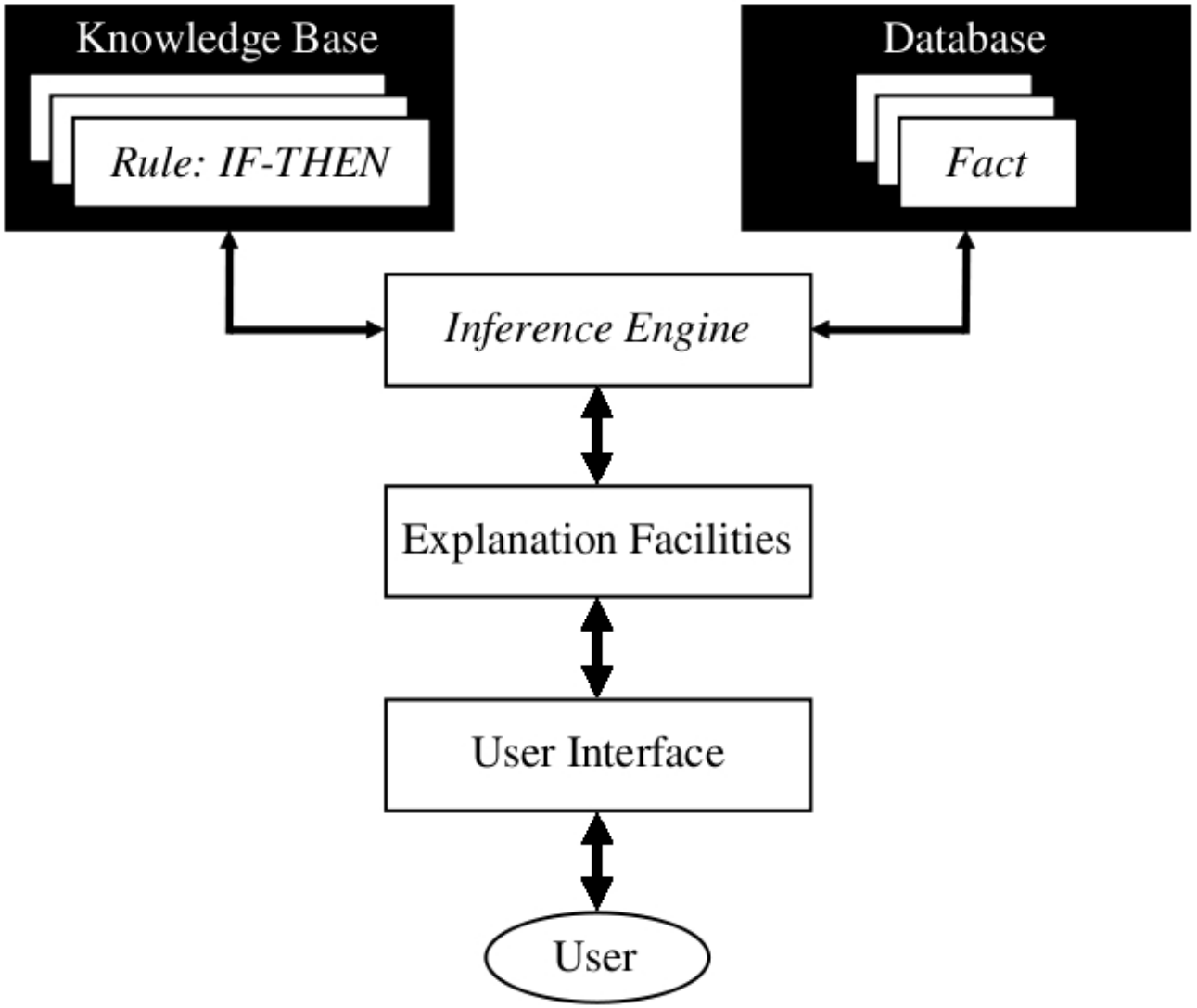
Basic architecture of rule-based *ES* - Plan:

- Production system
- Rule-based expert system
- Facts in *ES*
- Production rules in *ES*
- Inference performed by *ES*
- Forward chaining
- Backward chaining
- Simulation of recursion
- Agenda
- Conflict resolution strategies

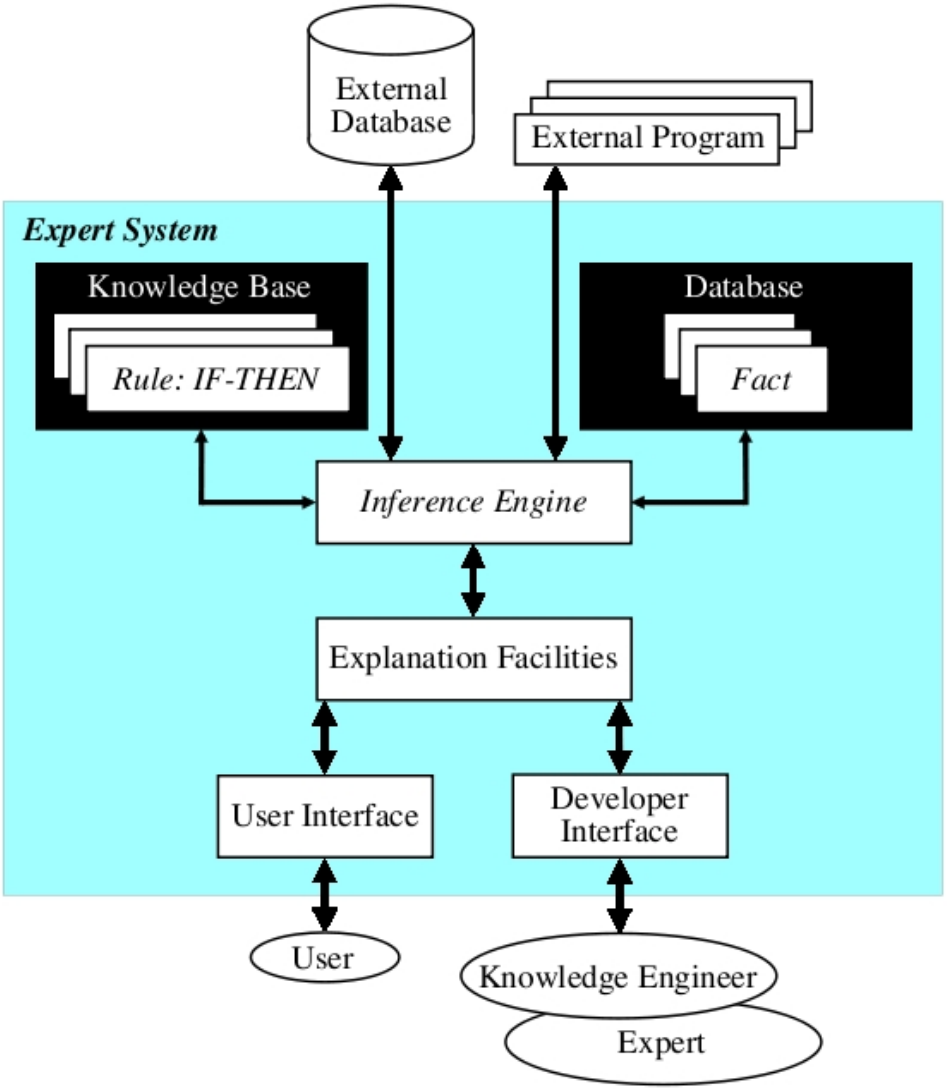
Basic architecture of a production system:



Basic architecture of a rule-based expert system:



Complete architecture of a rule-based expert system:



Basic architecture of rule-based *ES*

Facts in *ES*

- Data types: integer, float, symbol, string, fact index.
- "Ordered" facts (tuples): (the_pump is on)
  (altitude is 10000 feet)
  (grocery-list bread milk eggs)
- "Template" facts:

```
(deftemplate car                (car
 (slot identifier                (identifier car01)
 (slot brand)                    (brand VW)
 (slot model)                    (model Golf)
 (slot engine)                   (engine 1.4_TSI)
 (slot transmission)             (transmission 6speed_manual)
 (multislot accessories))        (accessories Air_conditioner
                                              Dusk_sensor
                                              Hill_hold
                                              Park_assist))
```

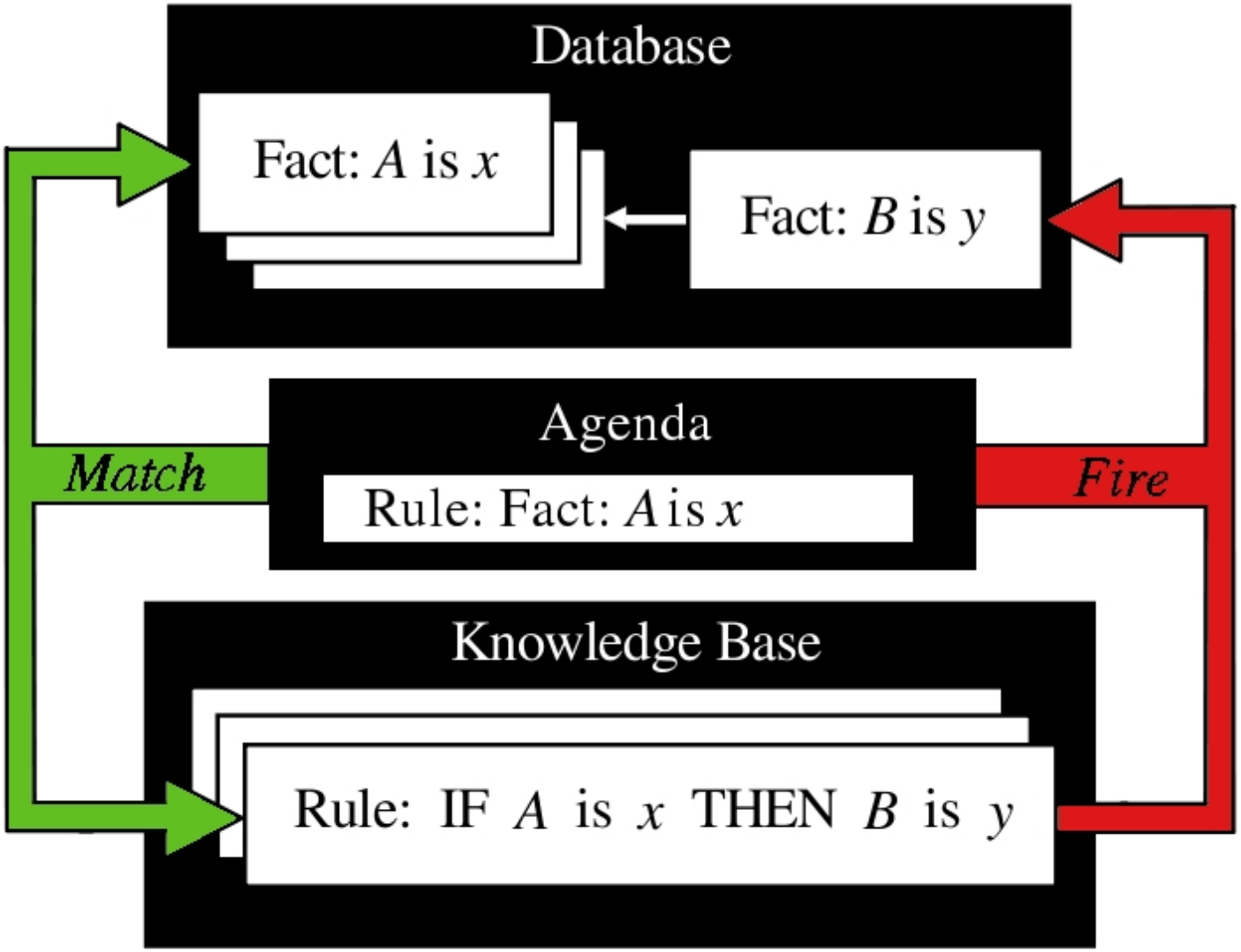
Basic architecture of rule-based *ES*

Production rules in *ES*

- *IF* part (*antecedent*) - built up from conditional elements (*patterns*) using the *and*, *or*, *not* connectives.
- *THEN* part (*consequent*) - a sequence of actions:
  - creation of new data, modifying, deletion of old data and their truth values;
  - input and output of data;
  - instructions for controlling the rule-firing process itself.

IF (speed is fast) *and* (it is raining) *and* *not* (windshield wipers on) *THEN* turn on windshield wipers, reduce speed

An inference cycle:



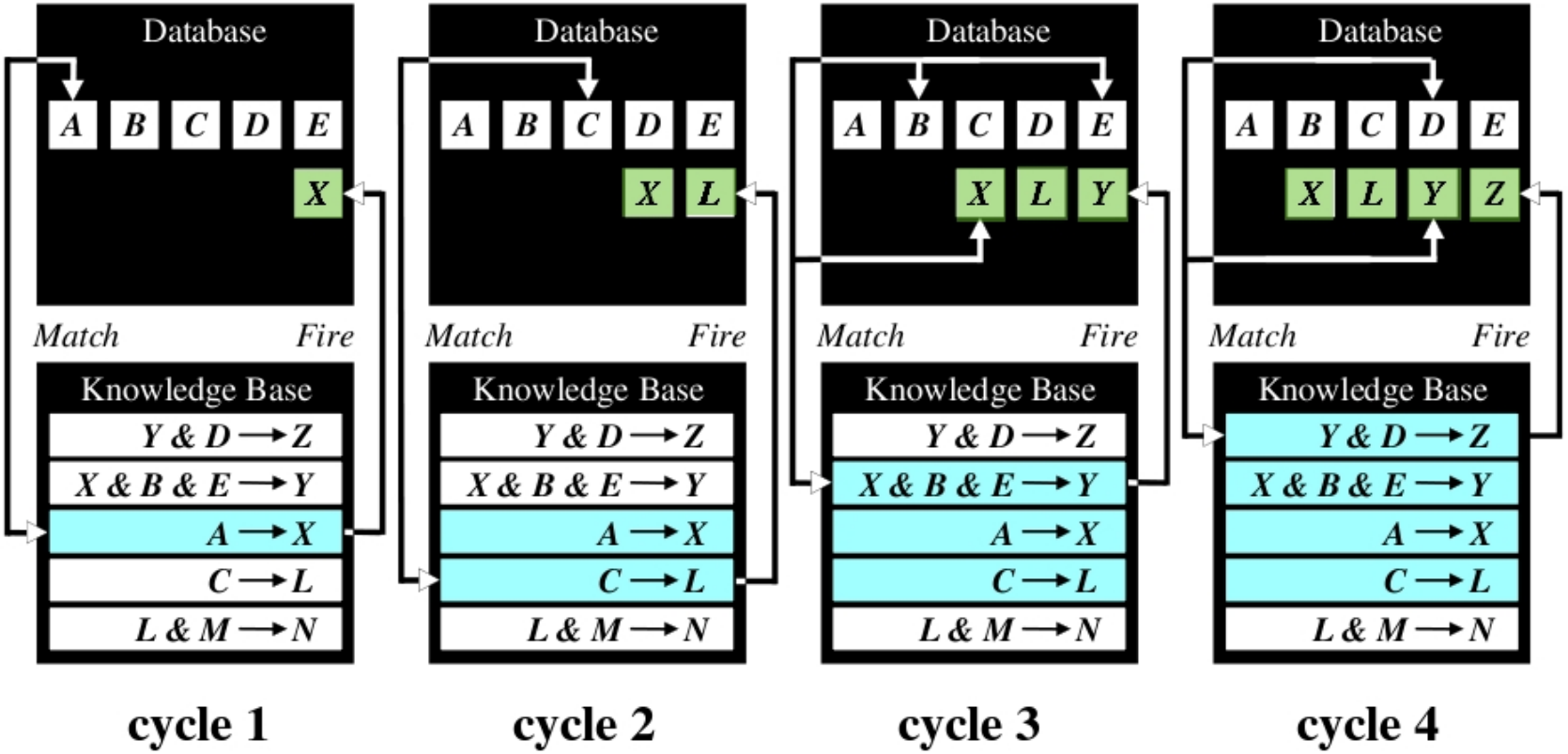
AN INFERENCE CYCLE

1. If there are no rules on the agenda, then execution is halted.
2. The top rule on the agenda is selected for execution.
3. The *consequent* actions of the selected rule are executed.
4. Deactivated rules are removed from the agenda.
5. Activated rules are placed on the agenda.
   Pattern matching - performed using the Rete algorithm [Forgy, 1982].
6. The placement on the agenda is determined by the *salience* of the rule and the current conflict resolution strategy.
7. Dynamic *salience* values for all rules on the agenda are re-evaluated.
8. Repeat the cycle beginning with step (1).

CHARACTERISTICS

- Non-procedural, data-driven.
- A technique for gathering information and then inferring from it whatever can be inferred.
- Many rules may be executed that have nothing to do with the established goal.
- If our goal is to infer only one particular fact, the forward chaining inference technique would not be efficient.

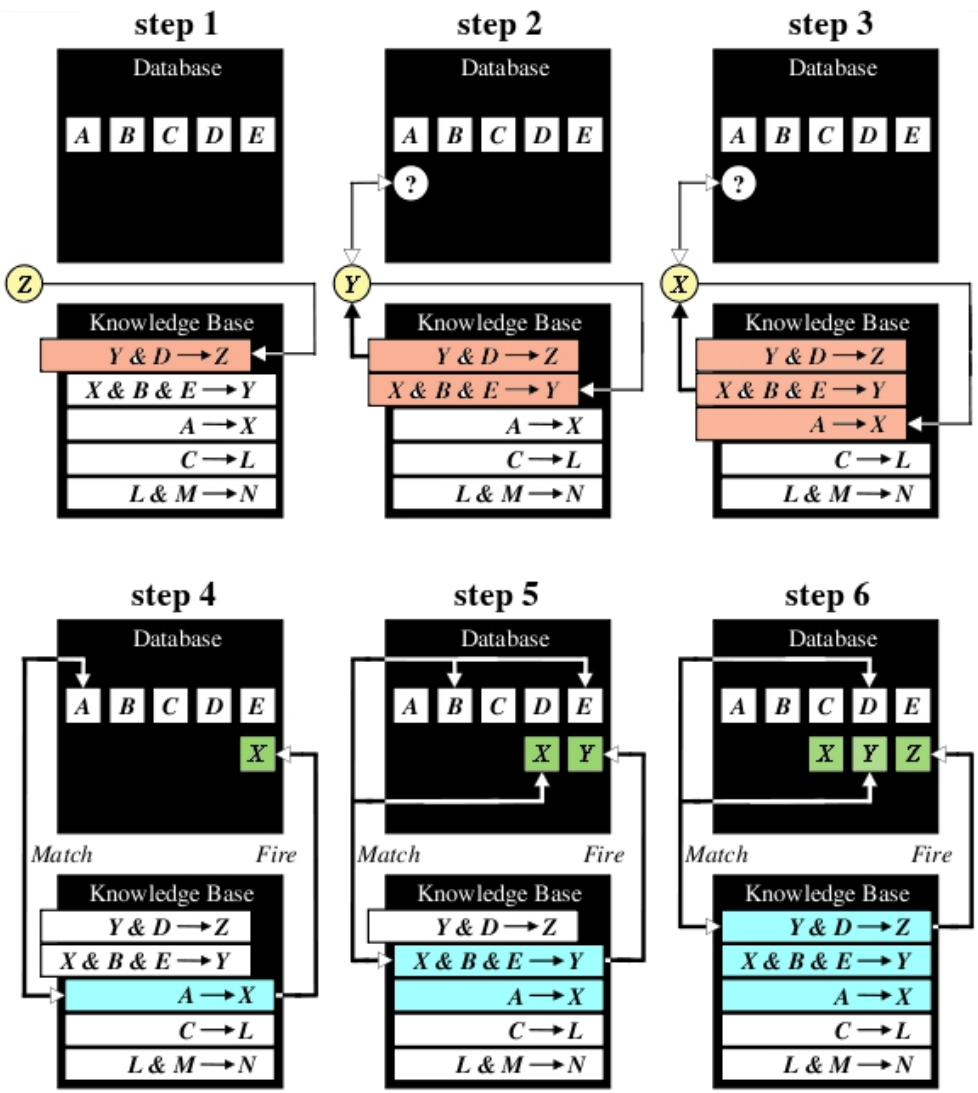
A forward chaining inference:



CHARACTERISTICS

- Non-procedural, goal-driven.
- For a goal (a hypothetical solution) - the inference engine attempts to prove it (find an evidence).
- The knowledge base is searched to find rules that might have the desired goal in their *THEN* parts. They are put on the stack.
- Conditions in their *IF* parts are new sub-goals.
- If such a rule is found that its *IF* part matches data in the database, then the rule is fired and removed from the stack.
- Inference is performed until either the desired goal is proved or the stack is empty.

A backward chaining inference:



USAGE

- Forward chaining - if the expert first needs to gather some information and then tries to infer from it whatever can be inferred.
- Backward chaining - if the expert begins with a hypothetical solution and then attempts to find facts to prove it.
- Most *ES*, *FES* exploit forward chaining.
- Forward chaining much more advanced, especially in *FES*.
- Backward chaining can be simulated by forward chaining and vice-versa.

Hanoi towers - an example of simulation in *CLIPS*

Decomposition rules:

```
(defrule r1
 (hanoi ?N ?A ?B ?C)
 =>
 (assert (run ?N ?N ?A ?B ?C)))

(defrule r2
 (run ?M ?N ?A ?B ?C)
 (test (> ?M 1))
 =>
 (assert (run (- ?M 1) ?N ?A ?C ?B))
 (assert (run (- ?M 1) ?N ?C ?B ?A)))
```

Hanoi towers - an example of simulation in *CLIPS*

Composition rules:

```
(defrule r3
 (run 1 ?N ?A ?B ?C)
 =>
 (assert (sol 1 ?N ?A ?B ?C ?A "->" ?B)))

(defrule r4
 (sol ?M ?N ?A ?C ?B $?Mv1)
 (test (< ?M ?N))
 (sol ?M ?N ?C ?B ?A $?Mv2)
 =>
 (assert (sol (+ ?M 1) ?N ?A ?B ?C $?Mv1 ?A "->" ?B $?Mv2)))

(defrule r5
 (hanoi ?N ?A ?B ?C)
 (sol ?N ?N ?A ?B ?C $?Mv)
 =>
 (assert (hanoi_sol ?N ?A ?B ?C $?Mv)))
```

AGENDA

- The agenda - the list of all rules which have their conditions satisfied and have not yet been executed, fired.
- The agenda - similar to a stack; the top rule on the agenda is the first one to be executed.
- The placement of a newly activated rule:
  - Newly activated rules - above all rules of lower salience and below all rules of higher salience.
  - The conflict resolution strategies are used to determine the placement among rules of equal salience.

- Depth strategy: newly activated rules - placed above all rules of the same salience.

rule-1:	f-1, f-2		rule-2:	f-3, f-1
rule-2:	f-3, f-1	$\Longrightarrow$	rule-4:	f-3, f-2
rule-3:	f-2		rule-3:	f-2
rule-4:	f-3, f-2		rule-1:	f-1, f-2

- Breadth strategy: newly activated rules - placed below all rules of the same salience.

rule-1:	f-1, f-2		rule-1:	f-1, f-2
rule-2:	f-3, f-1	$\Longrightarrow$	rule-3:	f-2
rule-3:	f-2		rule-4:	f-3, f-2
rule-4:	f-3, f-2		rule-2:	f-3, f-1

Basic architecture of rule-based *ES*

Conflict resolution strategies

- The specificity of a rule - the number of comparisons that must be performed on the *IF* part of the rule;
  - a comparison to a constant or previously bound variable +1;
  - a function call as part of a test conditional element +1;
  - the boolean functions *and*, *or*, *not* do not add, but their arguments do.

```
(defrule example
 (item ?x ?y ?x)
 (test (and (numberp ?x) (> ?x (+ 10 ?y)) (< ?x 100)))
   =>)
```

has a specificity of 5.

- Simplicity strategy: newly activated rules - placed above all activations of rules with equal or higher specificity.
- Complexity strategy: newly activated rules - placed above all activations of rules with equal or lower specificity.

Basic architecture of rule-based *ES*

Conflict resolution strategies

- *LEX* strategy:
  - the time tag of a fact - its relative recency with respect to every other fact;
  - the pattern entities associated with a rule activation - sorted in descending order;
  - an activation with more recent pattern entities - placed above activations with less recent pattern entities using the lexicographical order;
  - an activation with more pattern entities than the other activation and the compared time tags - all identical - placed above the other activation;
  - for two activations with the exact same recency, the activation with the higher specificity - placed above the activation with the lower specificity.

- *LEX* strategy:
  - the time tag of *not* CE_1
    - less than the time tag of a pattern entity,
    - greater than the time tag of *not* CE_2 instantiated after *not* CE_1.

rule-6:	f-1, f-4		rule-6:	f-4, f-1
rule-5:	f-1, f-2, f-3		rule-5:	f-3, f-2, f-1
rule-1:	f-2, f-3	$\Longrightarrow$	rule-1:	f-3, f-2
rule-2:	f-3, f-1		rule-2:	f-3, f-1
rule-4:	f-1, f-2		rule-4:	f-2, f-1
rule-3:	f-2		rule-3:	f-2

Basic architecture of rule-based *ES*

Conflict resolution strategies

- *MEA* (Means-end analysis) strategy:
  - an activation with the greater time tag of the fact associated with the first pattern entity than the first pattern entity's time tag of another activation - placed above the other activation;
  - for activations with the same time tag associated with their first pattern entities, the *LEX* strategy used.

rule-6:	f-1, f-4		rule-2:	f-3, f-1
rule-5:	f-1, f-2, f-3		rule-1:	f-2, f-3
rule-1:	f-2, f-3	$\Longrightarrow$	rule-3:	f-2
rule-2:	f-3, f-1		rule-6:	f-1, f-4
rule-4:	f-1, f-2		rule-5:	f-1, f-3, f-2
rule-3:	f-2		rule-4:	f-1, f-2

Introduction to fuzzy mathematics - Plan:

- Fuzzy *and* operator
- Fuzzy *or* operator
- Fuzzy *implication* (*residuum*) operator
- Fuzzy sets
- Fuzzy numbers
- Fuzzy arithmetic
- Comparing fuzzy numbers
- Interval arithmetic
- Linguistic variables
- Hedges

A *t*-norm T is a function $T : [0,1] \times [0,1] \longrightarrow [0,1]$:

$T(a,1) = a$; (1 is a neutral element)

$T(a,b) = T(b,a)$; (commutativity)

if $a_1 \leq a_2$, then $T(a_1,b) \leq T(a_2,b)$; (monotonicity)

$T(a,T(b,c)) = T(T(a,b),c)$; (associativity)

- denoted in infix notation as $*$.

- Gödel, Zadeh operator: $a *_m b = \min(a,b)$;
- the product, probabilistic operator: $a *_p b = a \cdot b$;
- Łukasiewicz, the bounded difference operator:
$$a *_Ł b = \max(0, a+b-1);$$
- the drastic intersection operator:
$$a *_d b = \begin{cases} a & \text{if } b = 1, \\ b & \text{if } a = 1, \\ 0 & \text{else.} \end{cases}$$

$$*_d \leq *_Ł \leq *_p \leq *_m$$

A *t*-conorm C is a function $C : [0,1] \times [0,1] \longrightarrow [0,1]$:

$C(a,0) = a$; (0 is a neutral element)

$C(a,b) = C(b,a)$; (commutativity)

if $a_1 \leq a_2$, then $C(a_1,b) \leq C(a_2,b)$; (monotonicity)

$C(a,C(b,c)) = C(C(a,b),c)$; (associativity)

- denoted in infix notation as $\oplus$.

- The standard union, Zadeh operator: $a \oplus_m b = \max(a,b)$;
- the algebraic sum, probabilistic operator:
$$a \oplus_p b = a + b - a \cdot b;$$
- Łukasiewicz, the bounded sum operator:
$$a \oplus_Ł b = \min(1, a+b);$$
- the drastic union operator:
$$a \oplus_d b = \begin{cases} a & \text{if } b = 0, \\ b & \text{if } a = 0, \\ 1 & \text{else.} \end{cases}$$

$$\oplus_m \leq \oplus_p \leq \oplus_Ł \leq \oplus_d$$

$$T_m(a_1,\ldots,a_n) = \min(a_1,\ldots,a_n),$$
$$C_m(a_1,\ldots,a_n) = \max(a_1,\ldots,a_n);$$

$$T_Ł(a_1,\ldots,a_n) = \max(0,(\sum_{i=1}^{n} a_i) - n + 1),$$

$$C_Ł(a_1,\ldots,a_n) = \min(1,\sum_{i=1}^{n} a_i);$$

$$T_p(a_1,\ldots,a_n) = a_1 \cdots a_n,$$

$$C_p(a_1,\ldots,a_n) = \sum_{i=1}^{n}(-1)^{i-1} \sum_{\substack{J \subseteq \{1,\ldots,n\},\\ |J| = i}} \prod_{k\in J} a_k;$$

$$T_d(a_1,\ldots,a_n) = \begin{cases} a_i & \textit{if } \exists\, 1 \le i \le n \textit{ and } \forall\, 1 \le j \ne i \le n \;\; a_j = 1, \\ 0 & \textit{else}, \end{cases}$$

$$C_d(a_1,\ldots,a_n) = \begin{cases} a_i & \textit{if } \exists\, 1 \le i \le n \textit{ and } \forall\, 1 \le j \ne i \le n \;\; a_j = 0, \\ 1 & \textit{else}. \end{cases}$$

$$\boldsymbol{L} = ([0,1], \leq, \vee, \wedge, *, \Rightarrow, \overline{}, \oplus, 0, 1)$$

- a complete residuated lattice algebra,

$\leq$ - the linear order,
$\vee$ - the supremum, $a \vee b = \max(a, b)$,
$\vee$ - the infimum, $a \wedge b = \min(a, b)$,
$*$ - a t-norm, $\Rightarrow$ - its residuum, $\oplus$ - its t-conorm,
$\overline{}$ - the negation operator.

1. $a * b \leq c \iff a \leq b \Rightarrow c$, (condition of residuation)
2. $\overline{a} = a \Rightarrow 0$, (condition of negation)
3. $a \oplus b = \overline{\overline{a}^{Ł} * \overline{b}^{Ł}}^{Ł}$, (condition of duality)
4. $a * (b \vee c) = a * b \vee a * c$, (distributivity of $*$ over $\vee$)
5. $a * b \leq a \wedge b$,
6. $a \vee b \leq a \oplus b$.

- Gödel residuum operator:

$$a \Rightarrow_G b = \begin{cases} 1 & \text{if } a \leq b, \\ b & \text{else}; \end{cases}$$

- the product, Goguen residuum operator:

$$a \Rightarrow_p b = \begin{cases} 1 & \text{if } a \leq b, \\ b/a & \text{else}; \end{cases}$$

- Łukasiewicz residuum operator: $a \Rightarrow_Ł b = \max(0, 1 - a + b)$;
- the drastic residuum operator:

$$a \Rightarrow_d b = \begin{cases} b & \text{if } a = 1, \\ 1 & \text{else}; \end{cases}$$

- the crisp, Gaines-Rescher residuum operator:

$$a \Rightarrow_c b = \begin{cases} 1 & \text{if } a \leq b, \\ 0 & \text{else}; \end{cases}$$

- Zadeh residuum operator: $a \Rightarrow_Z b = \max(1 - a, \min(a, b))$.

A fuzzy set F is a function $F : U \longrightarrow [0,1]$ [Zadeh, 1965];
- $U \neq \emptyset$ - the universum;
- a fuzzy set - identified by its membership function;
- $F(x)$ - the grade of membership of x in F.

BASIC OPERATIONS

Let A, B be fuzzy sets.

Intersection:	$A \cap B(x)$	$= A(x) *_m B(x) = \min(A(x), B(x))$;
Union:	$A \cup B(x)$	$= A(x) \oplus_m B(x) = \max(A(x), B(x))$;
Algebraic product:	$A *_p B(x)$	$= A(x) *_p B(x)$;
Algebraic sum:	$A \oplus_p B(x)$	$= A(x) \oplus_p B(x)$;
Bounded product:	$A *_Ł B(x)$	$= A(x) *_Ł B(x)$;
Bounded sum:	$A \oplus_Ł B(x)$	$= A(x) \oplus_Ł B(x)$;
Complement:	$\overline{A}(x)$	$= \overline{A(x)}^{Ł} = 1 - A(x)$.

BASIC NOTATION

Let $A, B : U \longrightarrow [0,1]$ be fuzzy sets and $\alpha \in [0,1]$.

$$A \subseteq B \Longleftrightarrow \forall\, x \in U \;\; A(x) \leq B(x),$$

$$\begin{aligned}
\mathit{support}(A) &= \{x \,|\, A(x) > 0\}, \\
\mathit{kernel}(A) &= \{x \,|\, A(x) = 1\}, \\
A[\alpha] &= \{x \,|\, A(x) \geq \alpha\} \quad \text{- } \alpha\text{-cut of } A, \\
A^{\alpha} &= \{x \,|\, A(x) = \alpha\} \quad \text{- } \alpha\text{-value of } A, \\
\mathit{height}(A) &= \bigvee \{A(x) \,|\, x \in U\}.
\end{aligned}$$

SPECIAL CASES

- Discrete fuzzy sets:

  $D : U \longrightarrow [0,1]$, U - a finite set or only a finite number of $x_1, \ldots, x_n \in U$ such that $D(x_i) > 0$;

$$\left\{ \frac{D(x_1)}{x_1}, \ldots, \frac{D(x_n)}{x_n} \right\};$$

  - numerical - members describe a numeric quantity:

$$\text{Size} = \left\{ \frac{\mu_1}{\text{small}}, \frac{\mu_2}{\text{medium}}, \frac{\mu_3}{\text{large}} \right\};$$

  - non-numerical - members describe a non-numeric quantity:

$$\text{Species} = \left\{ \frac{\mu_1}{\text{hamster}}, \frac{\mu_2}{\text{rabbit}}, \frac{\mu_3}{\text{hare}} \right\}.$$

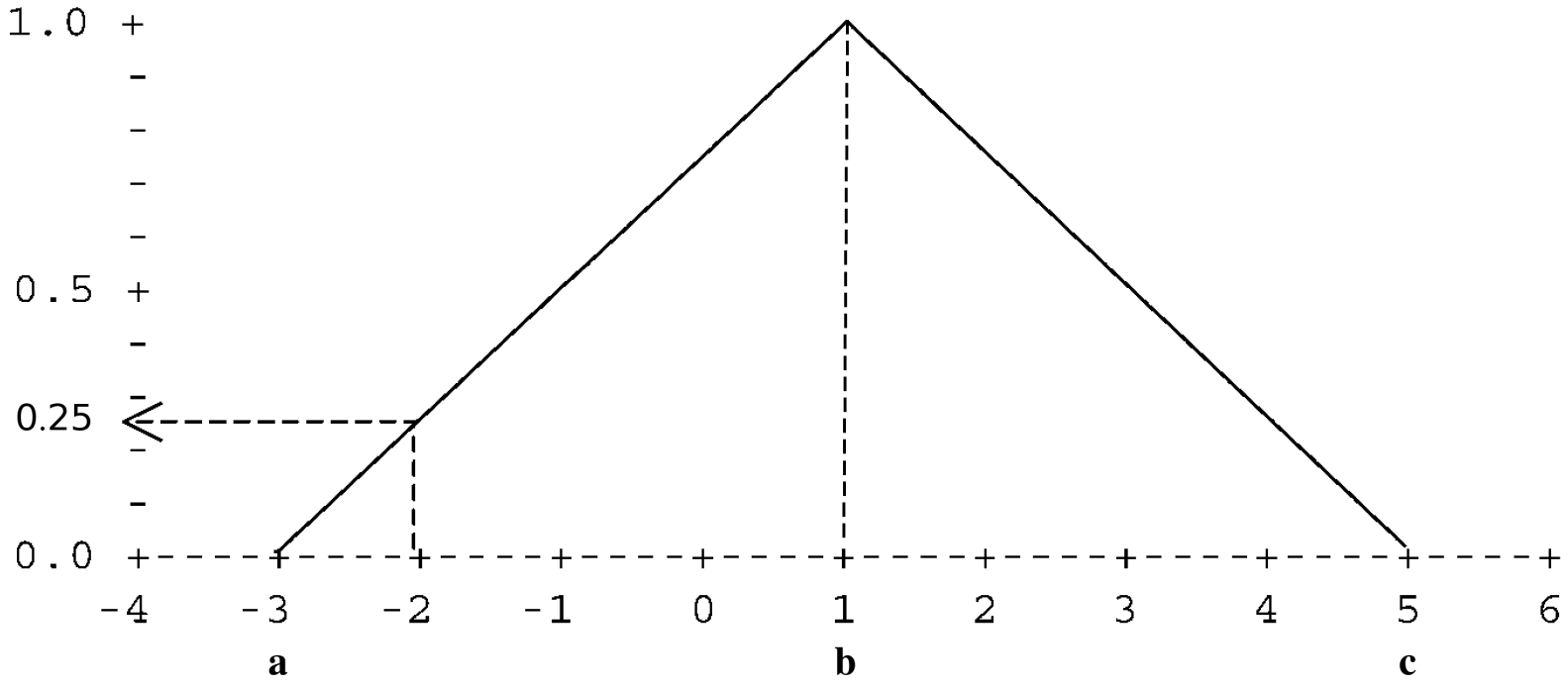
SPECIAL CASES

- Fuzzy numbers:

  $\tilde{n} : \mathbb{R} \longrightarrow [0,1]$, of whose value we are somewhat uncertain;

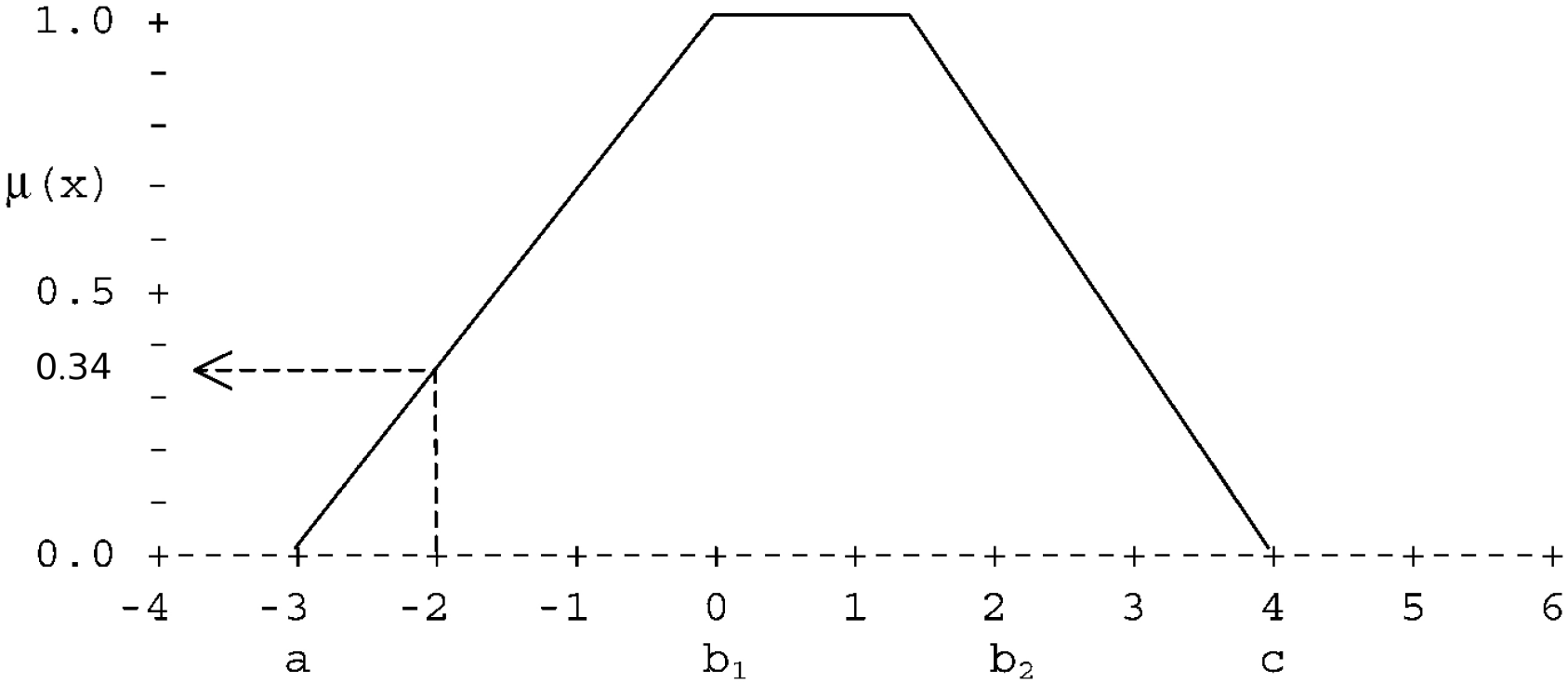
  - $\tilde{n}$ - normal: $\exists\, x\, \tilde{n}(x) = 1$;
    - convex:
      $\forall\, \lambda \in [0,1]\ \tilde{n}(\lambda \cdot x_1 + (1-\lambda) \cdot x_2) \geq \min(\tilde{n}(x_1), \tilde{n}(x_2))$;
    - continuous;
  - shape: - triangular (piecewise linear),
    - *s*-shape (piecewise quadratic),
    - normal (bell-shaped, Gaussian);
  - fuzzy intervals: - trapezoidal shape, a sub-interval of 1,
    - linear, *s*-shape, normal tails,
    - the increasing and decreasing slopes.

A triangular fuzzy number:



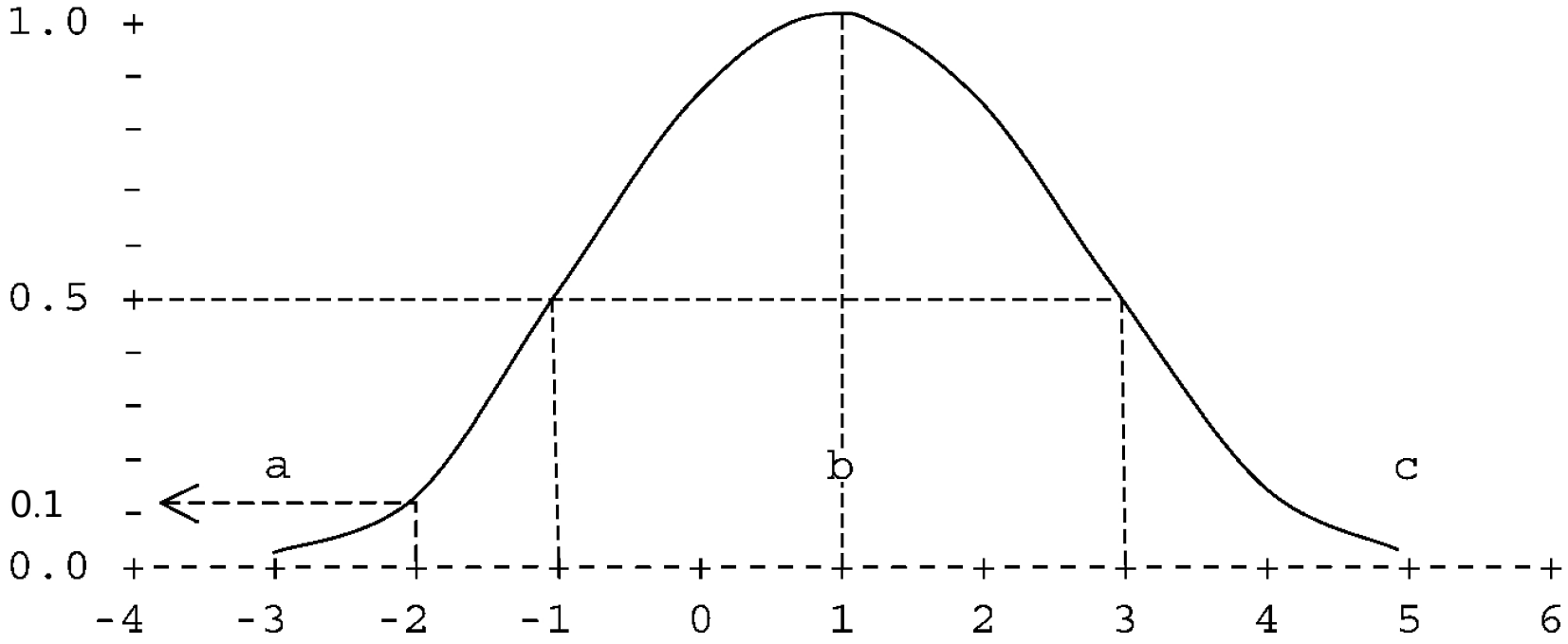
$$\mu(x) = \begin{cases} 0 & \text{if } x \leq a, \\ \frac{x-a}{b-a} & \text{if } a < x \leq b, \\ \frac{c-x}{c-b} & \text{if } b < x \leq c, \\ 0 & \text{if } x > c. \end{cases}$$

A trapezoidal triangular fuzzy number:



$$\mu(x) = \begin{cases} 0 & \text{if } x \leq a, \\ \frac{x-a}{b-a} & \text{if } a < x < b_1, \\ 1 & \text{if } b_1 \leq x \leq b_2, \\ \frac{c-x}{c-b} & \text{if } b_2 < x \leq c, \\ 0 & \text{if } x > c. \end{cases}$$

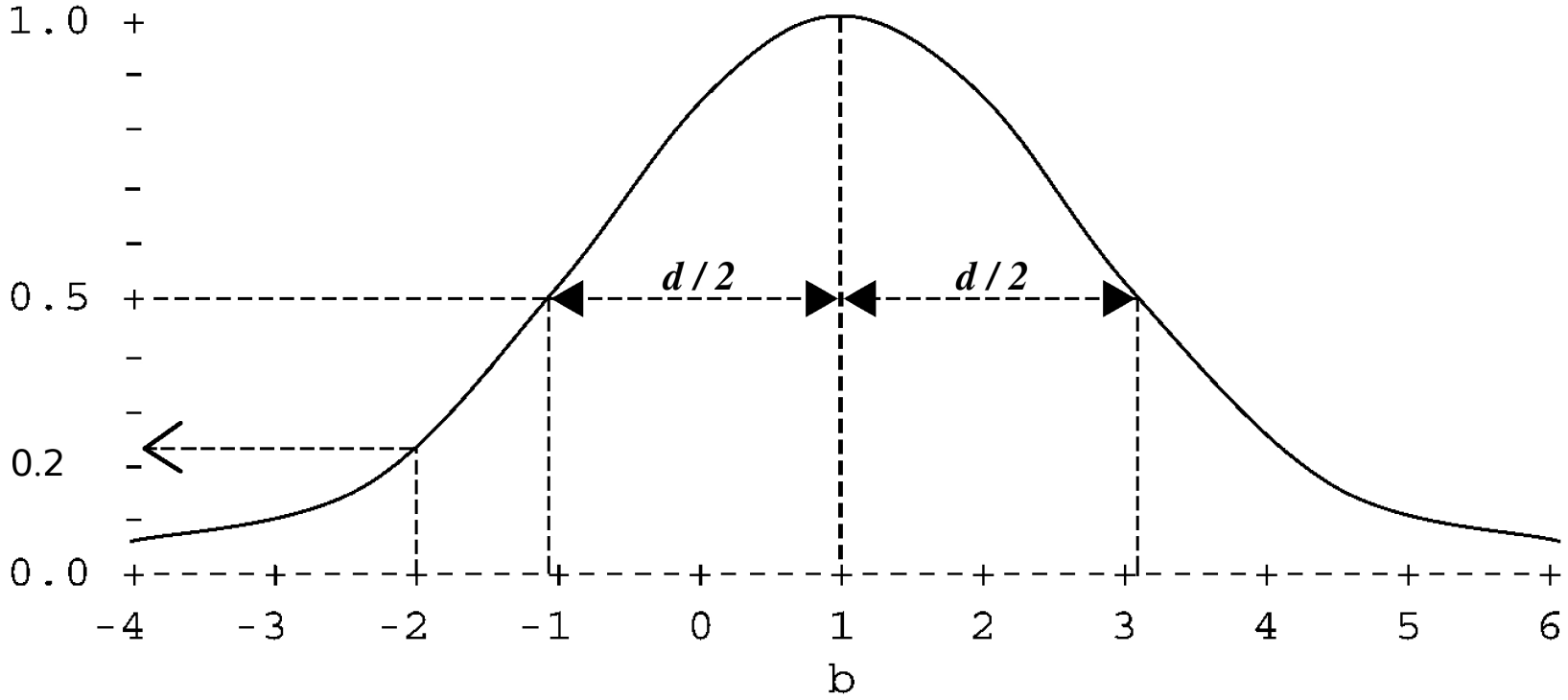
An *s*-shape fuzzy number:



$$\mu(x) = \begin{cases} k_1 \cdot (x-a)^2 & \text{if } a \le x \le \frac{a+b}{2}, \\ 1 - k_1 \cdot (b-x)^2 & \text{if } \frac{a+b}{2} a < x \le b, \\ 1 - k_2 \cdot (x-b)^2 & \text{if } b < x \le \frac{b+c}{2}, \\ k_2 \cdot (c-x)^2 & \text{if } \frac{b+c}{2} < x \le c, \end{cases}$$

k_1, k_2 - from $\mu(x) = 1/2$ for $x = \frac{a+b}{2}$ and $x = \frac{b+c}{2}$.

A normal (bell-shaped, Gaussian) fuzzy number:



$$\mu(x) = \exp^{\left(\ln\frac{1}{2}\right)\left(\frac{x-b}{d/2}\right)^2}.$$

EXTENSION PRINCIPLE [Zadeh, 1975]

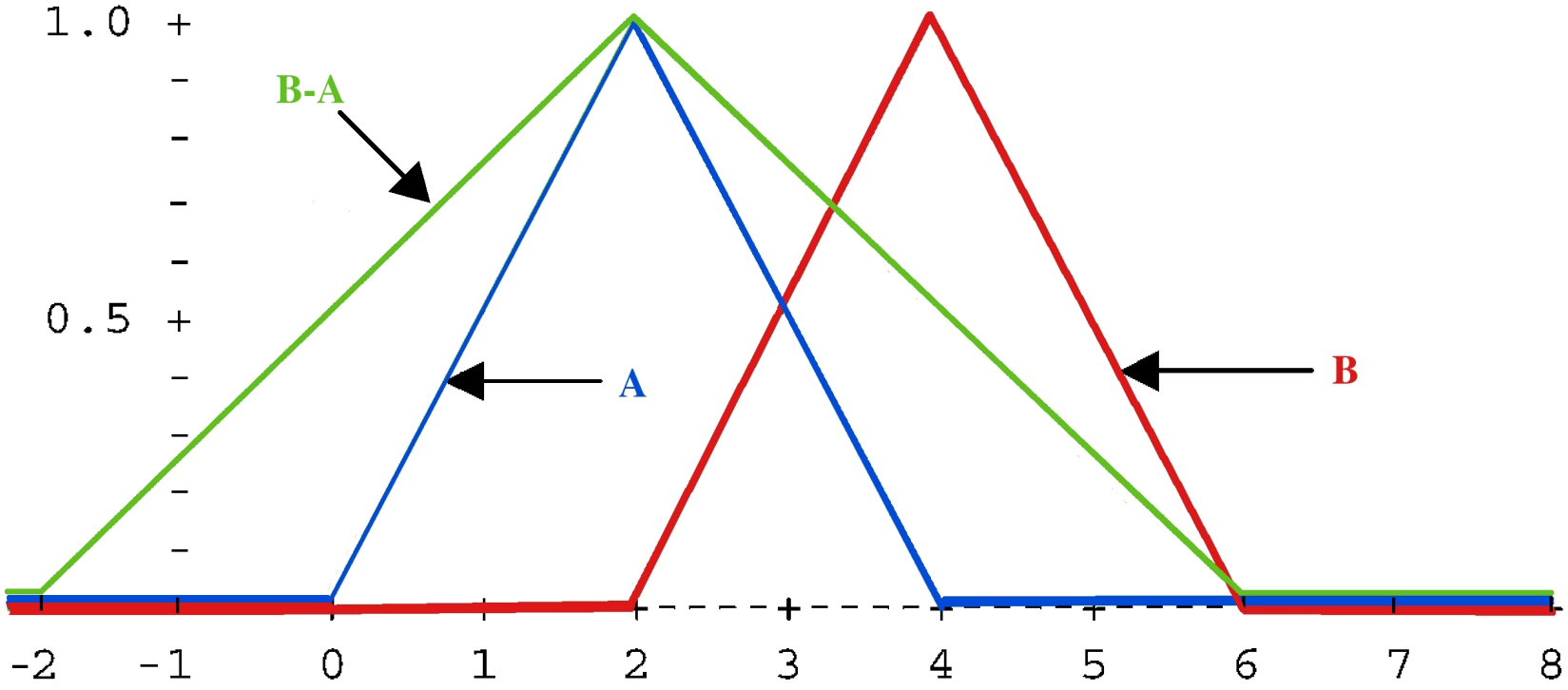
- A general tool used to fuzzify crisp operations $(+, -, \cdot, /)$.

  Let A, B be fuzzy numbers, and $*$ a crisp operation.

$$C(z) = \bigvee_{x\,y} \{\min(A(x), B(y)) \,|\, x * y = z\}.$$

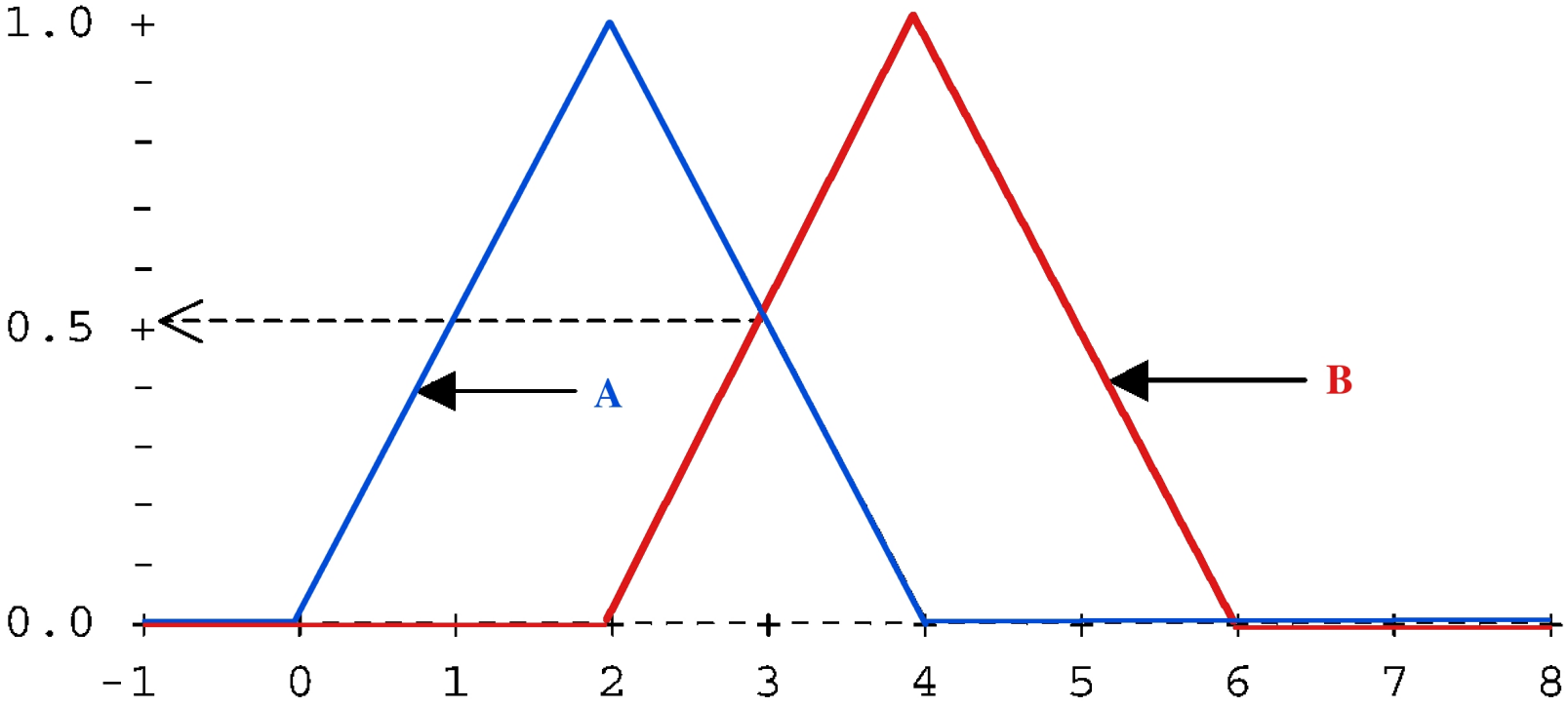
- The procedure is computationally expensive.

Subtraction of fuzzy numbers:



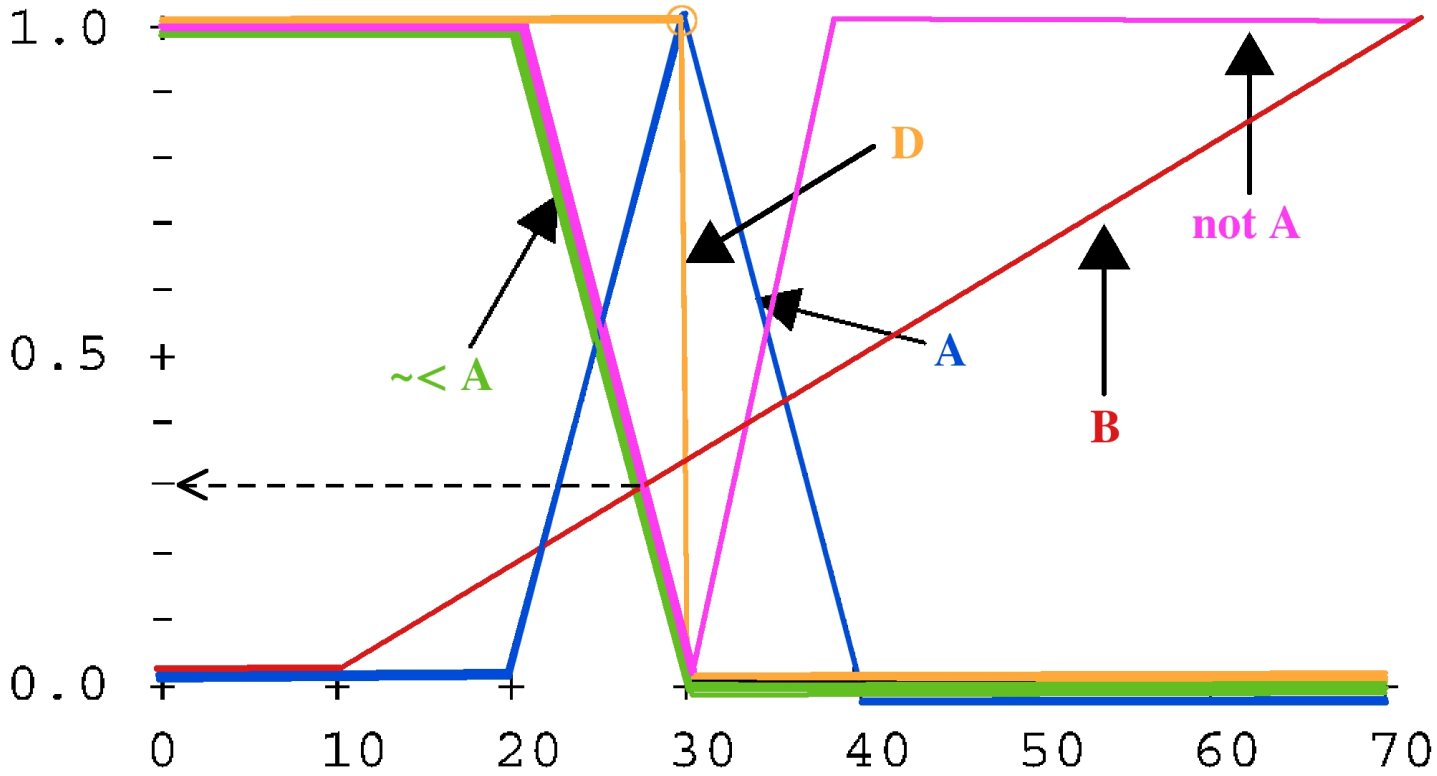
$$B - A(z) = \bigvee_{x\,y} \{\min(A(x), B(y)) \,|\, y - x = z\}.$$

Approximate equality $\sim=$:



$$A \sim= B = \bigvee_x \{\min(A(x), B(x))\}.$$

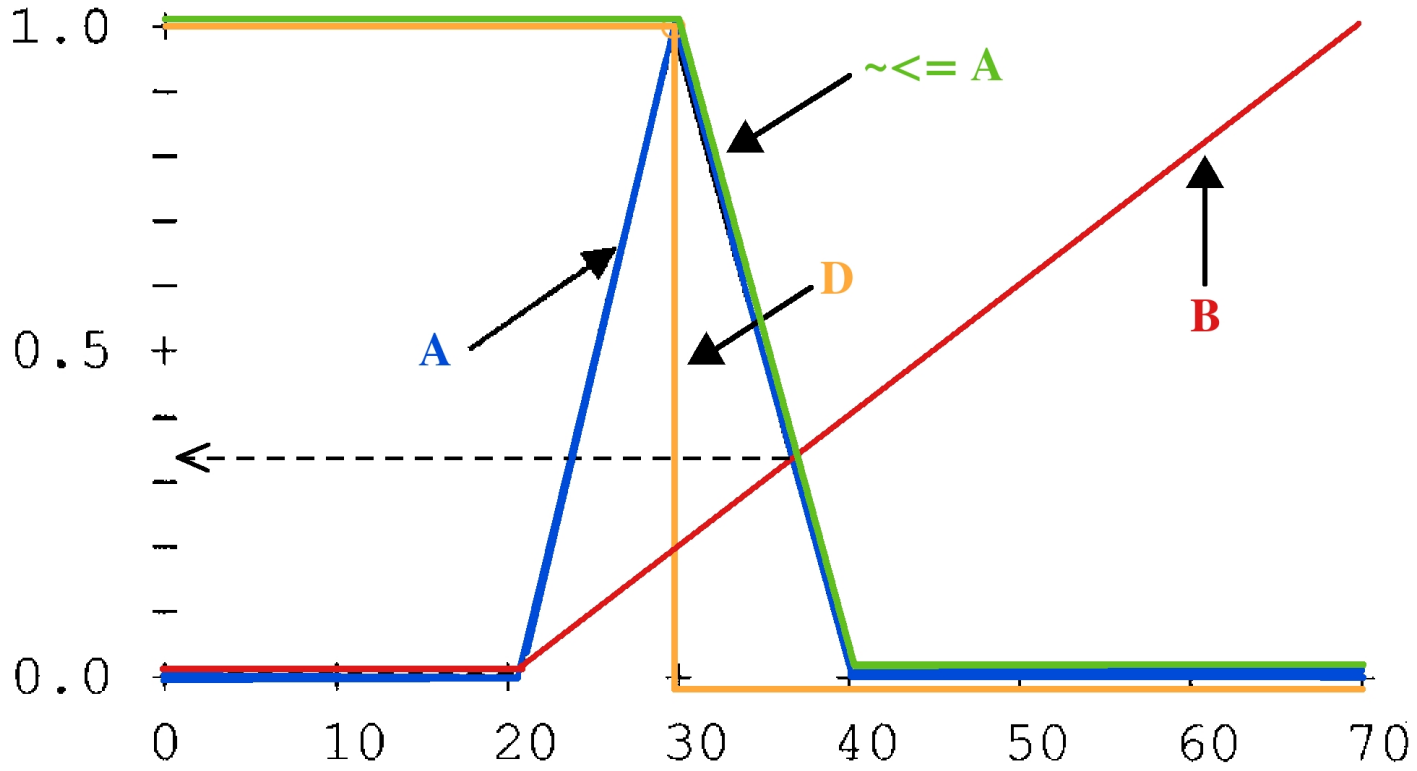
Approximate less $\sim<$:



$$B \sim< A = \bigvee_x \{\min(\overline{A} \cap D(x), B(x))\},$$

$$D(z) = \begin{cases} 1 \ \textit{if } z < \min_v \{v \mid A(v) = 1\}, \\ 0 \ \textit{else.} \end{cases}$$

Approximate less or equal $\sim<=$:



$$B \sim<= A = \bigvee_x \{\min(A \cup D(x), B(x))\},$$

$$D(z) = \begin{cases} 1 \ \textit{if } z < \min_v \{v \mid A(v) = 1\}, \\ 0 \ \textit{else.} \end{cases}$$

ALTERNATE METHOD [Klir, Yuan, 1995]

Let A be a fuzzy number and $kernel(A) = [a, b]$.

$$\delta = \frac{a+b}{2},$$

$$L(x) = \begin{cases} 1 \ if \ x \leq \delta, \\ 0 \ else, \end{cases} \qquad R(x) = \begin{cases} 1 \ if \ x \geq \delta, \\ 0 \ else, \end{cases}$$

$$\sim< A = \overline{A} \cap L,$$

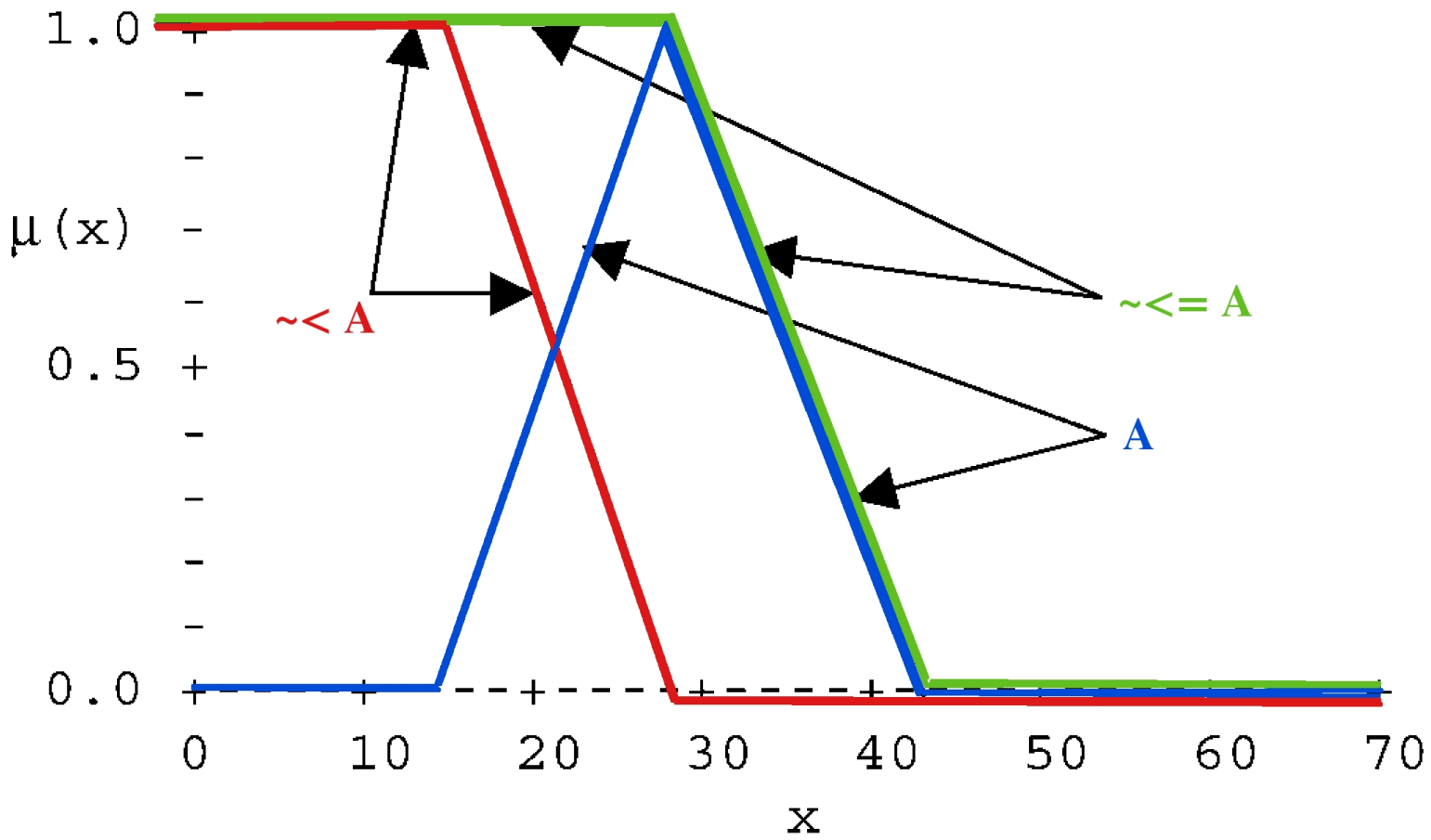
$$\sim<= A = \sim< A \oplus_{Ł} L,$$

$$\sim> A = \overline{A} \cap R,$$

$$\sim>= A = \sim> A \oplus_{Ł} R.$$

ALTERNATE METHOD

$\sim< A$ and $\sim<= A$:



BASE OF A FUZZY NUMBER

Let A be a fuzzy number and $\alpha \in (0, 1]$.

- The base of a fuzzy number:

$$\begin{aligned} &\text{triangular:} && A[0] = [a, b], \\ &\text{trapezoidal:} && A[0] = [a, c], \\ &s\text{-shape:} && A[0] = [a, c], \\ &\text{normal:} && A[0] = [a - 3\delta, a + 3\delta]. \end{aligned}$$

- $A[\alpha] = [a_1(\alpha), a_2(\alpha)]$ - a closed, bounded, interval.

INTERVAL EXTENSION

Let $[a, b]$, $[c, d]$ be closed, bounded, intervals;
$*$ denote $+$, $-$, $\cdot$, $/$ on $\mathbb{R}$.

- The interval extension:

$$
\begin{aligned}
[a, b] * [c, d] &= \{x * y \mid x \in [a, b], y \in [c, d]\}, \\
[a, b] + [c, d] &= [a + c, b + d], \\
[a, b] - [c, d] &= [a - d, b - c], \\
[a, b] \cdot [c, d] &= [\min\{a \cdot c, a \cdot d, b \cdot c, b \cdot d\}, \\
&\qquad \max\{a \cdot c, a \cdot d, b \cdot c, b \cdot d\}], \\
[a, b] / [c, d] &= [a, b] \cdot [\frac{1}{d}, \frac{1}{c}], \quad 0 \notin [c, d], \\
A * B[\alpha] &= A[\alpha] * B[\alpha] = [a_1(\alpha), a_2(\alpha)] * [b_1(\alpha), b_2(\alpha)].
\end{aligned}
$$

Both the methods - the same results for $+$, $-$, $\cdot$, $/$; not e.g. for $\frac{A+B}{A}$.

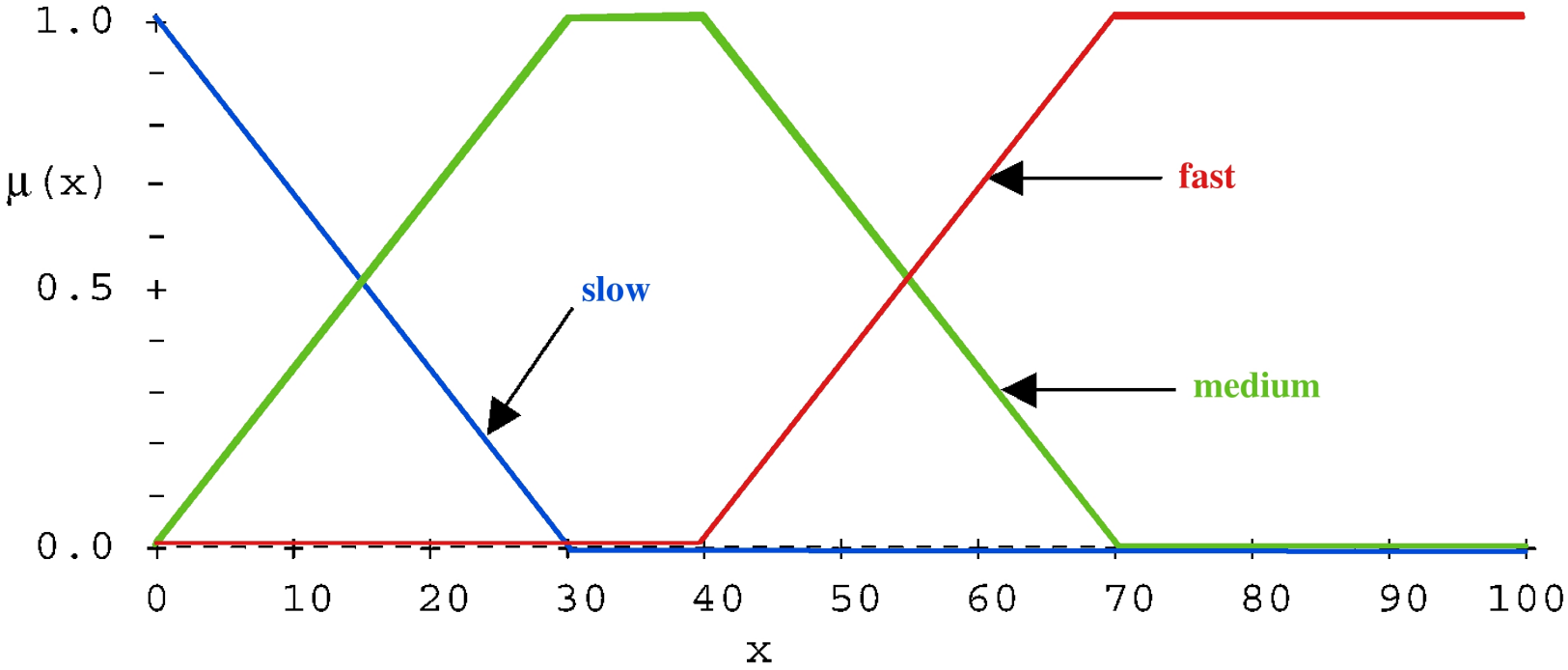
LINGUISTIC VARIABLE - A MODIFIED VERSION

[Zadeh, 1974], [Zadeh, 1975]

- The linguistic type of a linguistic variable - determined by
  - a finite set T of linguistic terms (slow, medium,. . .);
  - the universum U - $\mathbb{R}$ (an interval);
  - for each linguistic term $t \in T$ - a fuzzy set (fuzzy number) $F_t : U \longrightarrow [0, 1]$.
- A value of a linguistic variable is a numerical discrete fuzzy set $D : T \longrightarrow [0, 1]$;

$$\text{Speed} = \left\{ \frac{\mu_1}{\text{slow}}, \frac{\mu_2}{\text{medium}}, \frac{\mu_3}{\text{fast}} \right\}.$$

Linguistic terms of a linguistic variable Speed and their fuzzy sets:



$$\text{Speed} = \left\{ \frac{0.4}{\text{slow}}, \frac{0.6}{\text{medium}}, \frac{0.1}{\text{fast}} \right\}.$$

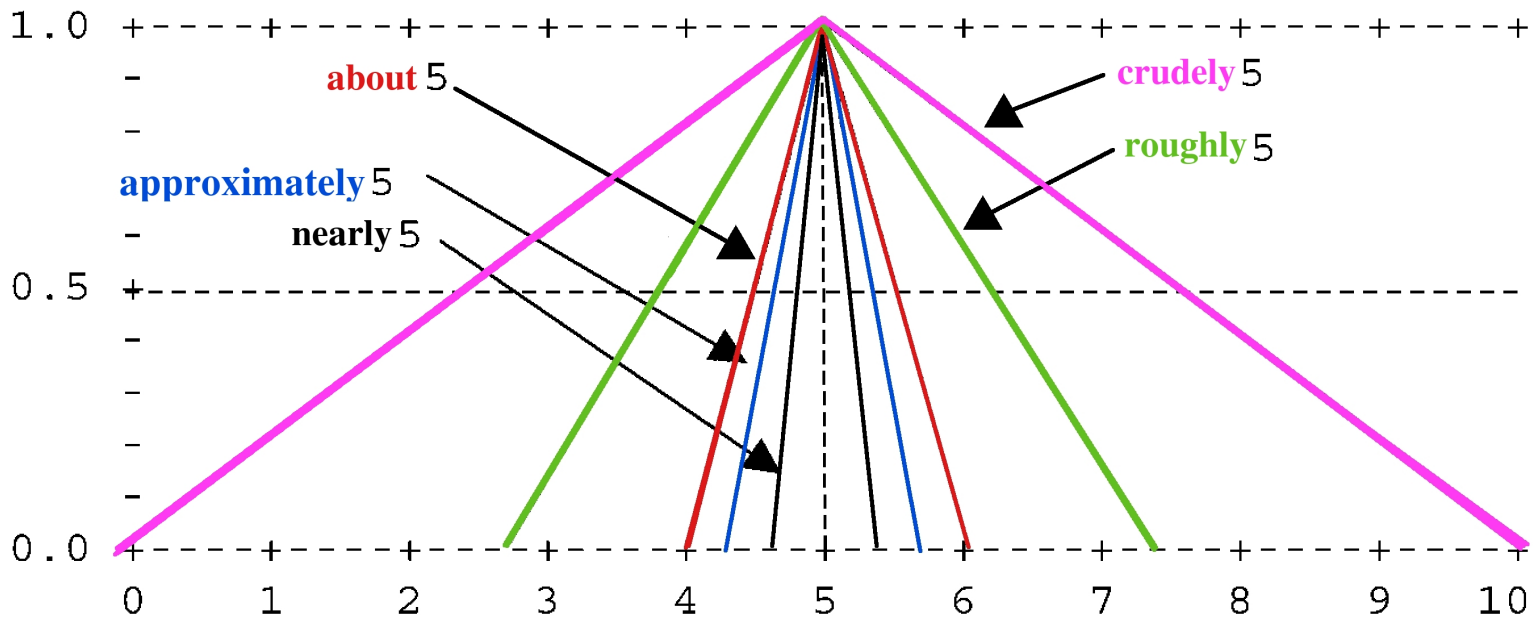
HEDGES [Zadeh, 1972], [Cox, 1999]

Hedges - modifiers, adjectives, or adverbs, which change truth values;
- about, nearly, roughly, extremely;
- several types of hedges.

Dispersion hedges for triangular fuzzy numbers:

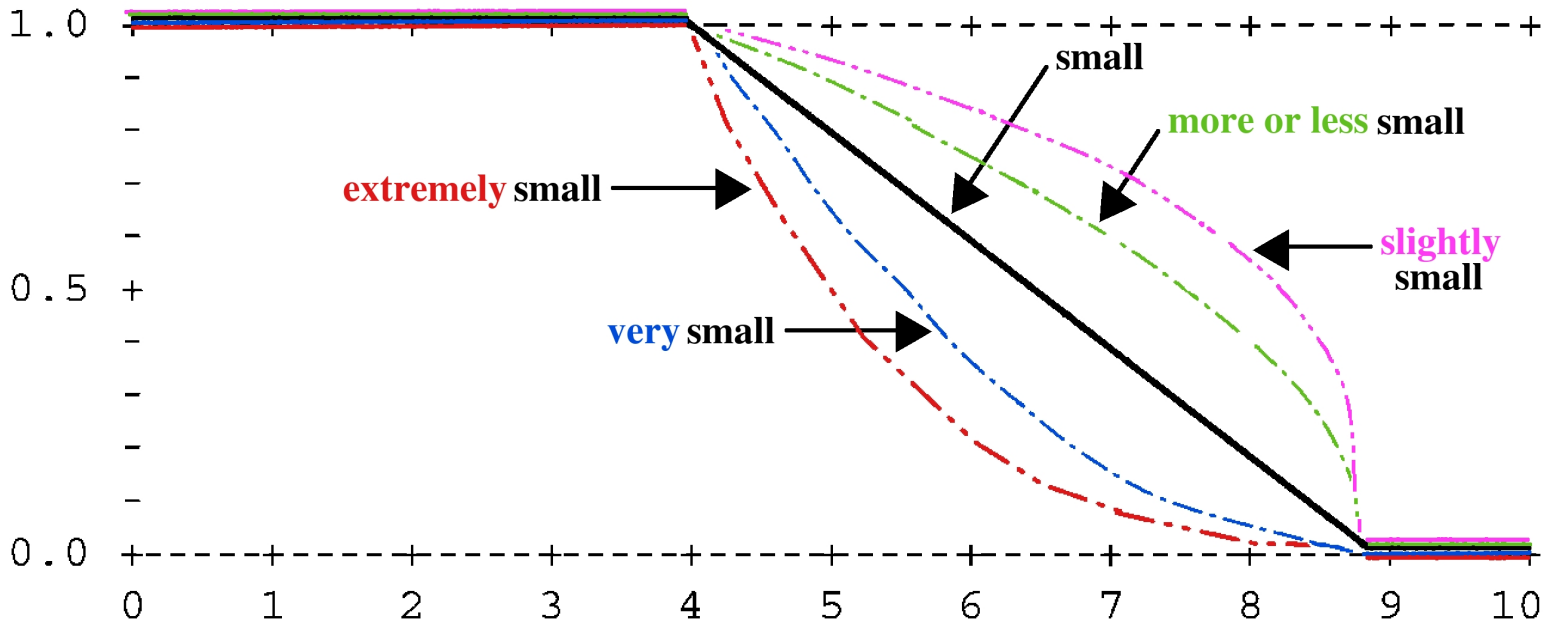
Hedge	Spread: $[z \cdot (1 - \frac{x}{100}), z \cdot (1 + \frac{x}{100})]$ of the central value z at a membership of 0.5
nearly	5%
approximately	10%
about	15%
roughly	25%
crudely	50%

Triangular fuzzy numbers of 5 created by various dispersion hedges:



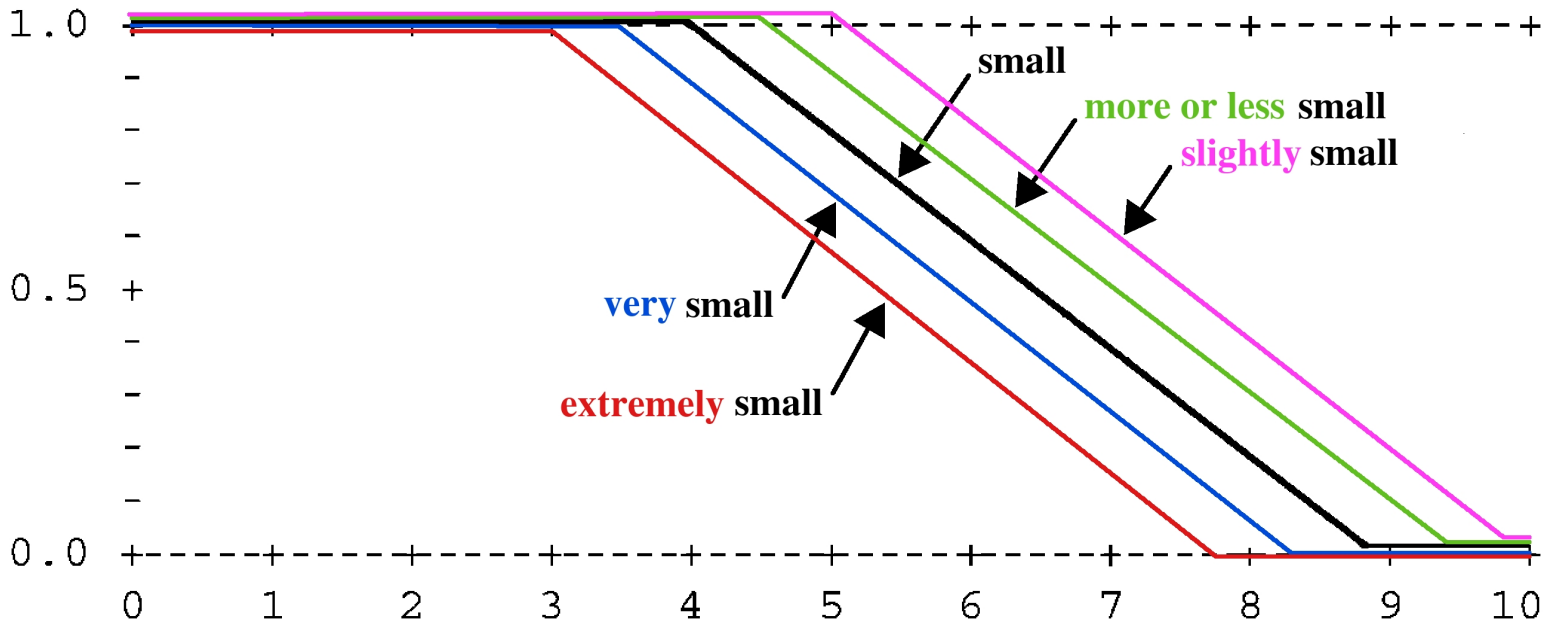
Powering hedges:

Hedge	Power
slightly	$F(x)^{\frac{1}{3}}$
more or less	$F(x)^{\frac{1}{2}}$
very	$F(x)^{2}$
extremely	$F(x)^{3}$



Shifting hedges for decreasing fuzzy sets:

Hedge	Shift
slightly	$F(x-1)$
more or less	$F(x-\frac{1}{2})$
very	$F(x+\frac{1}{2})$
extremely	$F(x+1)$



Fuzzy inference - Plan:

- Fuzzification
- Defuzzification
- Fuzzy rule (Mamdani type)
- Fuzzy rule (Sugeno-Takagi type)
- Fuzzy inference systems (*FIS*)

BASICS

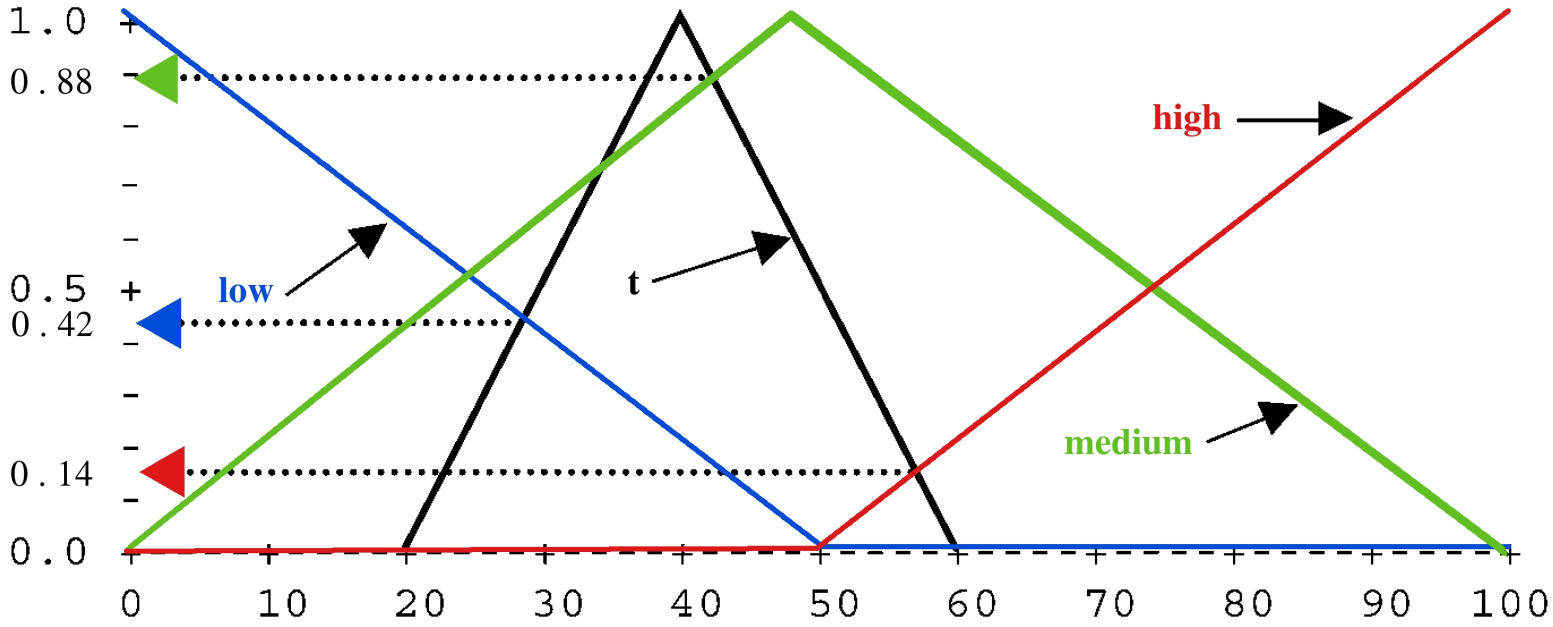
Fuzzification means

1. to find the fuzzy version of a crisp concept;
2. to find the degrees of membership of linguistic terms of a linguistic variable for an input (scalar, fuzzy) number.

- A fuzzy number $\mathbf{t}$ with the universum $[0, 100]$ (°C) of temperature;
- a linguistic variable Temperature with linguistic terms low, medium, high on $[0, 100]$.
- Fuzzification - the degree of membership of each linguistic term:

$$\mu(\text{low}) = \mathbf{t} \sim= \text{low},$$
$$\mu(\text{medium}) = \mathbf{t} \sim= \text{medium},$$
$$\mu(\text{high}) = \mathbf{t} \sim= \text{high}.$$

Fuzzification of $\mathbf{t}$ into the linguistic variable Temperature:



$$\text{Temperature} = \left\{ \frac{0.42}{\text{low}}, \frac{0.88}{\text{medium}}, \frac{0.14}{\text{high}} \right\}.$$

OVERVIEW

Defuzzification is the conversion of a fuzzy set to a crisp set.

- Defuzzification to a point:
  - maxima methods,
  - distribution methods,
  - area methods.
- Defuzzification to a set:
  - averaging procedures,
  - average α-cuts.
- An example of defuzzification methods.
- Recovering the crisp original value.

Defuzzification to a point

Maxima methods:

- The computation (selection) of an element on the basis of the maxima (local, global) of a fuzzy set as the defuzzification value - simplicity.

- *RCOM* - a random choice of maximum;
- *FOM* - the first of maxima;
- *LOM* - the last of maxima;
- *MiOM* - the middle of maxima;
- *MOM* - the mean of maxima:

$$MOM(F) = \frac{\sum_{i=1}^{n} x\_max_i}{n};$$

- *WAOM* - the weighted average of maxima:

$$WAOM(F) = \frac{\sum_{i=1}^{n} x\_loc\_max_i \cdot F(x\_loc\_max_i)}{\sum_{i=1}^{n} F(x\_loc\_max_i)}.$$

Defuzzification to a point

Distribution methods:

- The conversion of a fuzzy set F into a probability distribution, the computation of the expected value.

- *COG* - the center of gravity:

$$COG(F) = \frac{\sum_{i=1}^{n} loc\_center_i \cdot loc\_area_i}{\sum_{i=1}^{n} loc\_area_i}, \qquad COG(F) = \frac{\sum_{i=1}^{n} x_i \cdot F(x_i)}{\sum_{i=1}^{n} F(x_i)};$$

- *BADD* - the basic defuzzification distribution:

$$BADD(F,\gamma) = \frac{\sum_{i=1}^{n} loc\_center_i \cdot loc\_area_i^{\gamma}}{\sum_{i=1}^{n} loc\_area_i^{\gamma}}, \ \gamma \in [0,\infty) \text{ - the confidence;}$$

- *GLSD* - the generalized level set defuzzification:

$$GLSD(F,\gamma) = \frac{\sum_{i=1}^{n} \mu_i \cdot |F[\alpha_i]| \cdot \gamma^i}{\sum_{i=1}^{n} |F[\alpha_i]| \cdot \gamma^i}, \ \gamma \in (0,\infty), \ \mu_i \text{ - the average of } F[\alpha_i].$$

Defuzzification to a point

Area methods:

- *COA* - the center of area:

$$COA(F) = z \text{ such that } \left| \sum_{x=\min U}^{z} F(x) - \sum_{x=z}^{\max U} F(x) \right| \text{ - least.}$$

Defuzzification to a set

Averaging procedures:

- The arithmetic mean of two sets:

$$\frac{1}{2}(A+B) = \{\tfrac{1}{2}(x+y) \,|\, x \in A, y \in B\};$$

- A' - an averaging procedure:

$$A'(F,\alpha) = \frac{1}{2}(F[\alpha] + F[1-\alpha]);$$

- M - an averaging procedure:

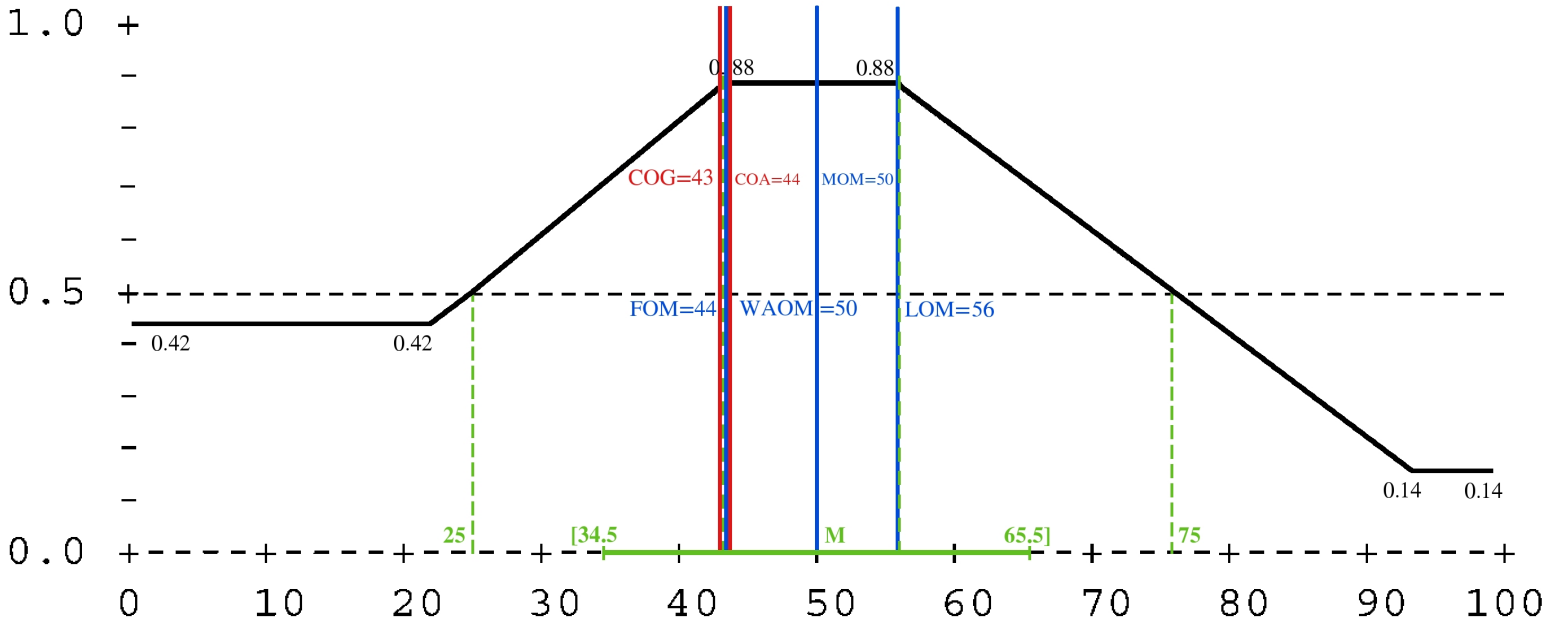
$$M(F) = \frac{1}{2}(F[1] + F[\tfrac{1}{2}]).$$

Average α-cuts:

- A - the average α-cut:

$$A(F) = [\int_{[0,1]} \min F[\alpha]\,\mathrm{d}\alpha, \int_{[0,1]} \max F[\alpha]\,\mathrm{d}\alpha].$$

An example of defuzzification methods:



$$\text{Temperature} = \left\{ \frac{0.42}{\text{low}}, \frac{0.88}{\text{medium}}, \frac{0.14}{\text{high}} \right\}.$$

Defuzzification recovering the crisp original value

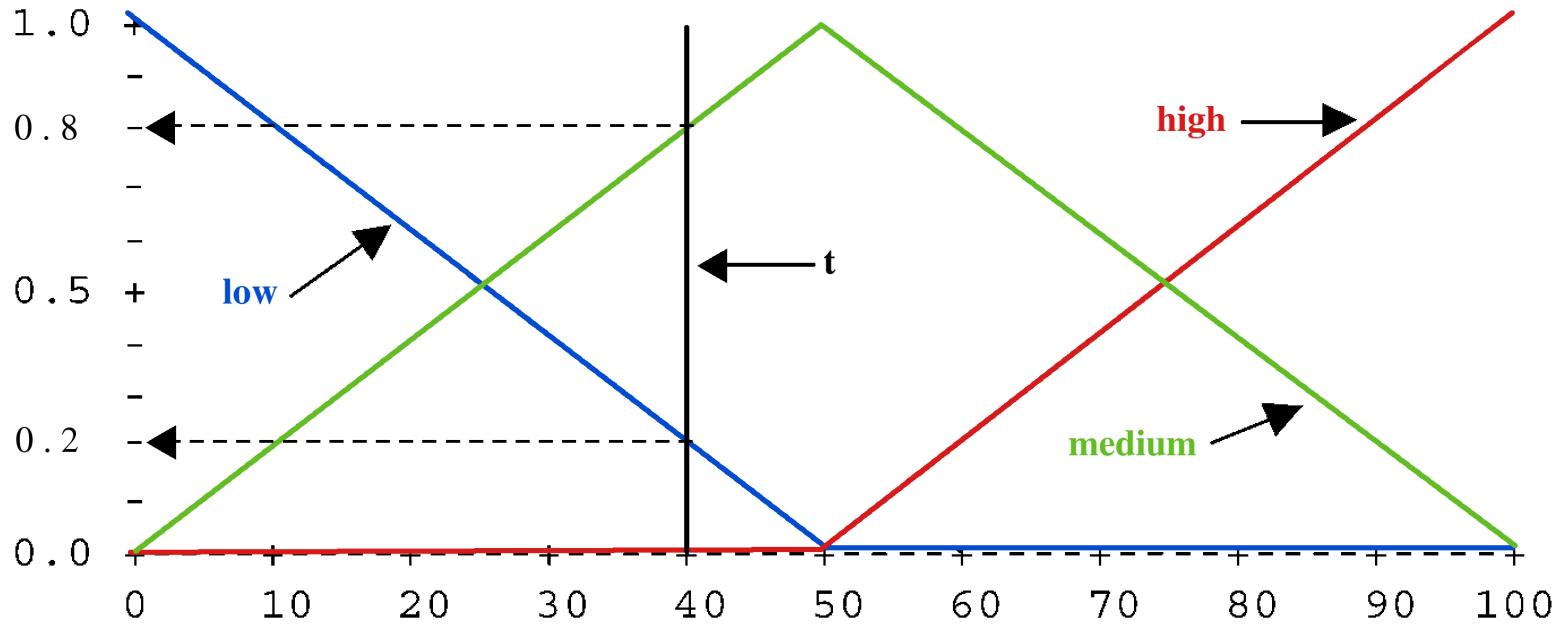
- A value $y \in \mathbb{R}$ is fuzzified;
- a vector $\boldsymbol{v}$ of membership degrees of all used linguistic terms - known;
- y - recovered from $\boldsymbol{v}$.
- Suitable for fuzzy reasoning.

- *HeM* - the height method:

$$HeM(\boldsymbol{v}) = \frac{\sum_{i=1}^{n} v_i \cdot B_i}{\sum_{i=1}^{n} v_i},$$

  - f_i - a triangular fuzzy set with the barycenter B_i;
  - $\boldsymbol{v} = (v_1, \ldots, v_n)$ - a vector of membership degrees of y in f_i;
  - for every v_i there is some y_i such that $v_i = f_i(y_i)$.

Defuzzification using the height method:



$$\text{low}(\mathbf{t}) = 0.2 \qquad B_1 = 0$$
$$\text{medium}(\mathbf{t}) = 0.8 \qquad B_2 = 50$$
$$\text{high}(\mathbf{t}) = 0 \qquad B_3 = 100$$
$$\boldsymbol{v} = (0.2, 0.8, 0)$$

$$HeM(\boldsymbol{v}) = \frac{\sum_{i=1}^{3} v_i \cdot B_i}{\sum_{i=1}^{3} v_i} = \frac{0.2 \cdot 0 + 0.8 \cdot 50 + 0 \cdot 100}{0.2 + 0.8 + 0} = 40 = \mathbf{t}.$$

- Definition of a fuzzy rule (Mamdani type)
- Fuzzy relations
- Compositional rule of inference (*CRI*)
- *CRI*'s using t-norms
- Mamdani-Assilian method
- A simplified scheme of *CRI*
- Larsen method
- Drastic product method
- Bounded product method
- Tsukamoto method

DEFINITION

Let A_i be fuzzy sets on X_i, B on Y.
A fuzzy rule of the Mamdani type is of the form

$$IF\ a_1\ is\ A_1\ and \cdots and\ a_m\ is\ A_m\ THEN\ b\ is\ B$$

a_i, b - fuzzy variables, having fuzzy sets as values.

Let A_j^i be fuzzy sets on X_j^i, B^i on Y^i.
A fuzzy rule base $\mathbb{RB}$ is a set of fuzzy rules:

$$IF\ a_1\ is\ A_1^1\ and \cdots and\ a_m\ is\ A_m^1\ THEN\ b\ is\ B^1$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$IF\ a_1\ is\ A_1^n\ and \cdots and\ a_m\ is\ A_m^n\ THEN\ b\ is\ B^n$$

IF Distance *is* SHORT *and* Speed *is* HIGH *THEN* Braking *is* HARD

BASICS

Let X, Y, Z be sets.
A fuzzy relation R is a function $R : X \times Y \longrightarrow [0,1]$;

- a special case of fuzzy set, with the universum $X \times Y$;
- given analytically or as matrices if the sets involved - finite.

Let R be a fuzzy relation on $X \times Y$ and S on $Y \times Z$.
The composition $R \circ S : X \times Z \longrightarrow [0,1]$:

$$R \circ S(x,z) = \bigvee_y \{R(x,y) * S(y,z)\};$$

- for $X = \{x_i \,|\, 1 \leq i \leq m\}$, $Y = \{y_j \,|\, 1 \leq j \leq n\}$,
  $Z = \{z_k \,|\, 1 \leq k \leq p\}$, being finite non-empty sets,

$$R \circ S(x_i, z_k) = \bigoplus_{y_j} \{R(x_i, y_j) * S(y_j, z_k)\}.$$

COMPOSITION OF A FUZZY SET AND A RELATION

Let A be a fuzzy set on X and R a fuzzy relation on $X \times Y$.
For $y \in Y$ - a fuzzy set $R[y] : X \longrightarrow [0,1] : R[y](x) = R(x,y)$.

The composition $A \circ_* R : Y \longrightarrow [0,1]$:

$$A \circ_* R(y) = \bigvee_x \{A(x) * R(x,y)\} = \bigvee_x \{A * R[y](x)\};$$

- for $X = \{x_i \,|\, 1 \leq i \leq m\}$, $Y = \{y_j \,|\, 1 \leq j \leq n\}$, being finite non-empty sets,

$$A \circ_* R(y_j) = \bigoplus_{x_i} \{A(x_i) * R(x_i, y_j)\} = \bigoplus_{x_i} \{A * R[y_j](x_i)\}.$$

MATRICES

- the multiplication of the matrices of membership degrees:

$$\begin{bmatrix} R \circ S(x_1, z_1) \cdots R \circ S(x_1, z_p) \\ \vdots \quad \cdots \quad \vdots \\ R \circ S(x_m, z_1) \cdots R \circ S(x_m, z_p) \end{bmatrix} = \begin{bmatrix} R(x_1, y_1) \cdots R(x_1, y_n) \\ \vdots \quad \cdots \quad \vdots \\ R(x_m, y_1) \cdots R(x_m, y_n) \end{bmatrix} \circ \begin{bmatrix} S(y_1, z_1) \cdots S(y_1, z_p) \\ \vdots \quad \cdots \quad \vdots \\ S(y_n, z_1) \cdots S(y_n, z_p) \end{bmatrix}$$

$$m \times p \qquad\qquad m \times n \qquad\qquad n \times p$$

For $* = \min = *_m$, $\oplus = \max = \oplus_m$:

$$\begin{bmatrix} 0.6 & 0.4 \\ 0.9 & 0.5 \end{bmatrix} = \begin{bmatrix} 0.2 & 0.4 & 0.6 \\ 0.3 & 0.6 & 0.9 \end{bmatrix} \circ \begin{bmatrix} 0.5 & 1 \\ 0.7 & 0.5 \\ 1 & 0 \end{bmatrix}$$

$$B \quad = \quad A \quad \circ_{*_m} \quad R$$

$$\begin{bmatrix} 0.6 & 0.4 \end{bmatrix} = \begin{bmatrix} 0.2 & 0.4 & 0.6 \end{bmatrix} \circ_{*_m} \begin{bmatrix} 0.5 & 1 \\ 0.7 & 0.5 \\ 1 & 0 \end{bmatrix}$$

- A on $X = \{x_1, x_2, x_3\}$, B on $Y = \{y_1, y_2\}$, R on $X \times Y$.

DEFINITION

- The classical (crisp) modus ponens: $\dfrac{A,\ IF\ A\ THEN\ B}{B}$.

Let $*$ be a t-norm, $\Rightarrow$ a residuum operator.
Let A, A' be fuzzy sets on X, B on Y.
- The generalized (fuzzy) modus ponens: [Zadeh, 1973]

$$\frac{x = A',\ IF\ x = A\ THEN\ y = B}{y = B'}.$$

A compositional rule of inference:

$$B'(y) = A' \circ_* [A \Rightarrow B](y) = \bigvee_x \{A'(x) * [A \Rightarrow B](x, y)\}$$
$$= \bigvee_x \{A'(x) * (A(x) \Rightarrow B(y))\},$$
$$[A \Rightarrow B](x, y) = A(x) \Rightarrow B(y) \text{ on } X \times Y.$$

For $* = *_Ł$, $\Rightarrow = \Rightarrow_Ł$:

$a *_Ł b = \max(0, a + b - 1)$,

$[A \Rightarrow B](x_i, y_j) = A(x_i) \Rightarrow_Ł B(y_j) = \min(1, 1 - A(x_i) + B(y_j))$.

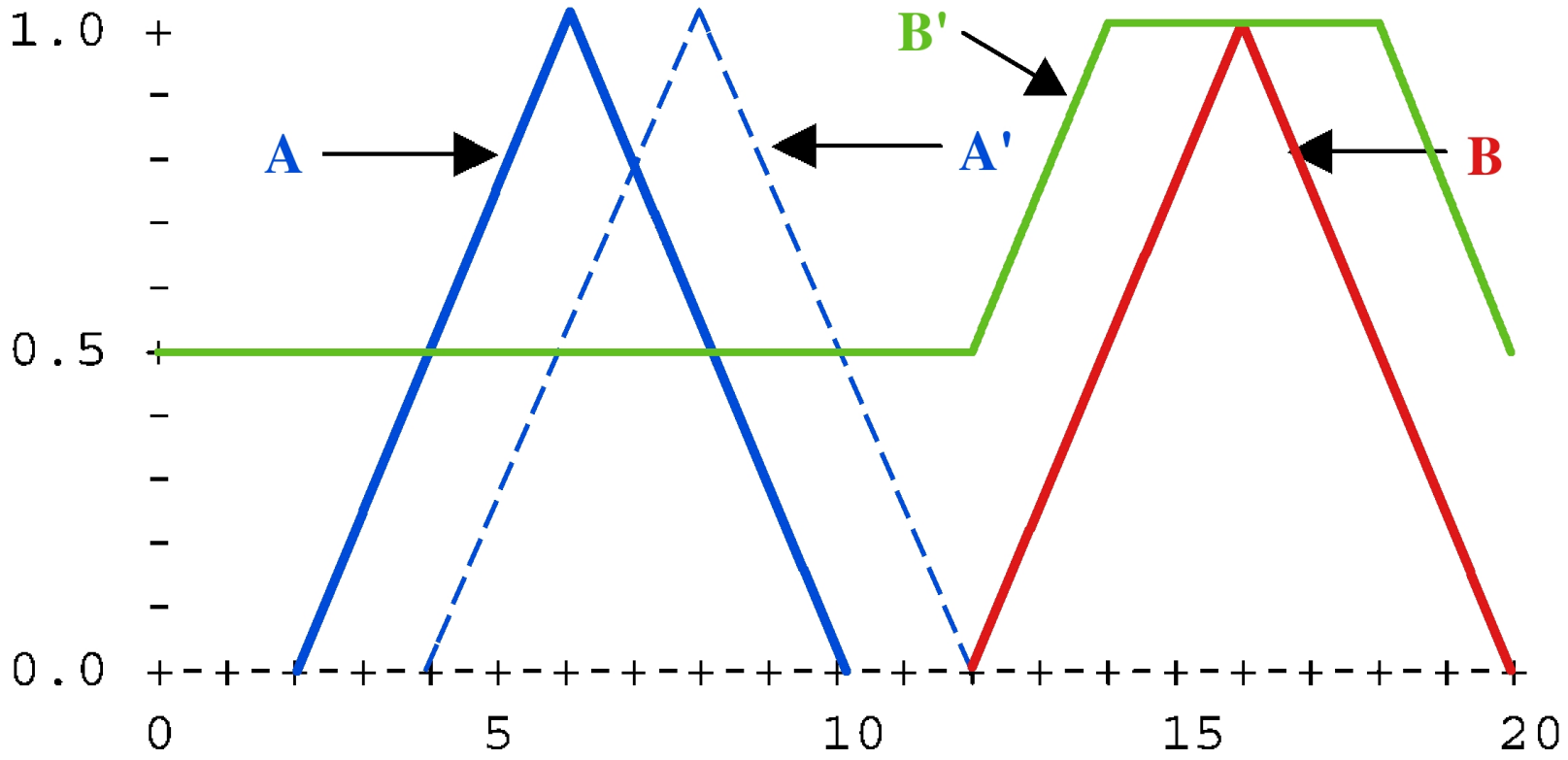
Let $A = \left\{ \frac{0.8}{x_1}, \frac{0.7}{x_2}, \frac{1}{x_3} \right\}$, $B = \left\{ \frac{0.5}{y_1}, \frac{1}{y_2}, \frac{0.6}{y_3} \right\}$, $A' = \left\{ \frac{0.5}{x_1}, \frac{0.2}{x_2}, \frac{0.3}{x_3} \right\}$.

$$[A \Rightarrow B] = \begin{bmatrix} 0.7 & 1 & 0.8 \\ 0.8 & 1 & 0.9 \\ 0.5 & 1 & 0.6 \end{bmatrix}$$

$$B' = A' \circ_{*_Ł} [A \Rightarrow B]$$

$$\begin{bmatrix} 0.2 & 0.5 & 0.3 \end{bmatrix} = \begin{bmatrix} 0.5 & 0.2 & 0.3 \end{bmatrix} \circ_{*_Ł} \begin{bmatrix} 0.7 & 1 & 0.8 \\ 0.8 & 1 & 0.9 \\ 0.5 & 1 & 0.6 \end{bmatrix}$$

For $* = *_m$, $\Rightarrow = \Rightarrow_G$:



DEFINITION

Let $*$ be a t-norm. Let A, A' be fuzzy sets on X, B on Y.
A compositional rule of inference using a t-norm:

$$B'(y) = A' \circ_{*_m} [A * B](y) = \bigvee_x \min\Big(A'(x), [A * B](x, y)\Big)$$
$$= \bigvee_x \min\Big(A'(x), (A(x) * B(y))\Big),$$

$$[A * B](x, y) = A(x) * B(y) \text{ on } X \times Y.$$

MAMDANI-ASSILIAN METHOD [Mamdani, Assilian, 1975]

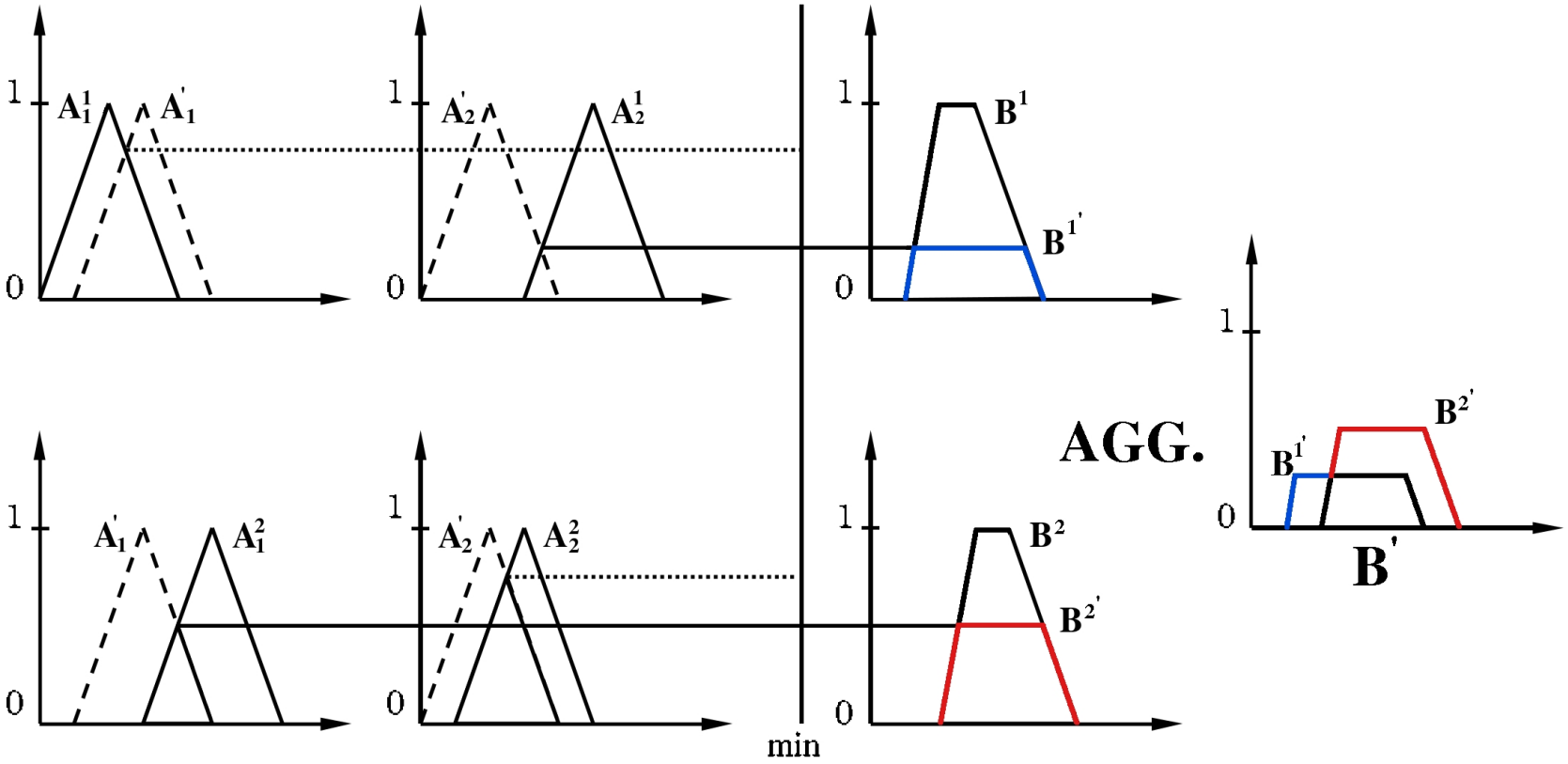
Let $\mathbb{RB}$ be a rule base.

$$B^{i\prime}(y) = \min_{j=1}^{m} \bigvee_{x} \min\Big(A'_j(x), A^i_j(x), B^i(y)\Big)$$

$$= \min\Bigg(\overbrace{\underbrace{\min_{j=1}^{m} \bigvee_{x} \min\Big(A'_j(x), A^i_j(x)\Big)}_{\alpha^i_j \text{ - the matching degree}}}^{\alpha^i \text{ - the firing strength}}, B^i(y)\Bigg),$$

$$B' = \bigcup_{i=1}^{n} B^{i\prime} \text{ - the aggregation.}$$

An example of Mamdani-Assilian method:



IF a_1 *is* A_1^1 *and* a_2 *is* A_2^1 *THEN* b *is* B^1

IF a_1 *is* A_1^2 *and* a_2 *is* A_2^2 *THEN* b *is* B^2

DEFINITION

Let $*$ be a t-norm. Let $\mathbb{RB}$ be a rule base.

A simplified compositional rule of inference using a t-norm:

$$B^{i\prime}(y) = \overbrace{\left(\underbrace{\min_{j=1}^{m} \bigvee_{x} \min\left(A'_j(x), A^i_j(x)\right)}_{\alpha^i_j \text{ - the matching degree}}\right)}^{\alpha^i \text{ - the firing strength}} * B^i(y),$$

$$B' = \bigcup_{i=1}^{n} B^{i\prime} \text{ - the aggregation.}$$

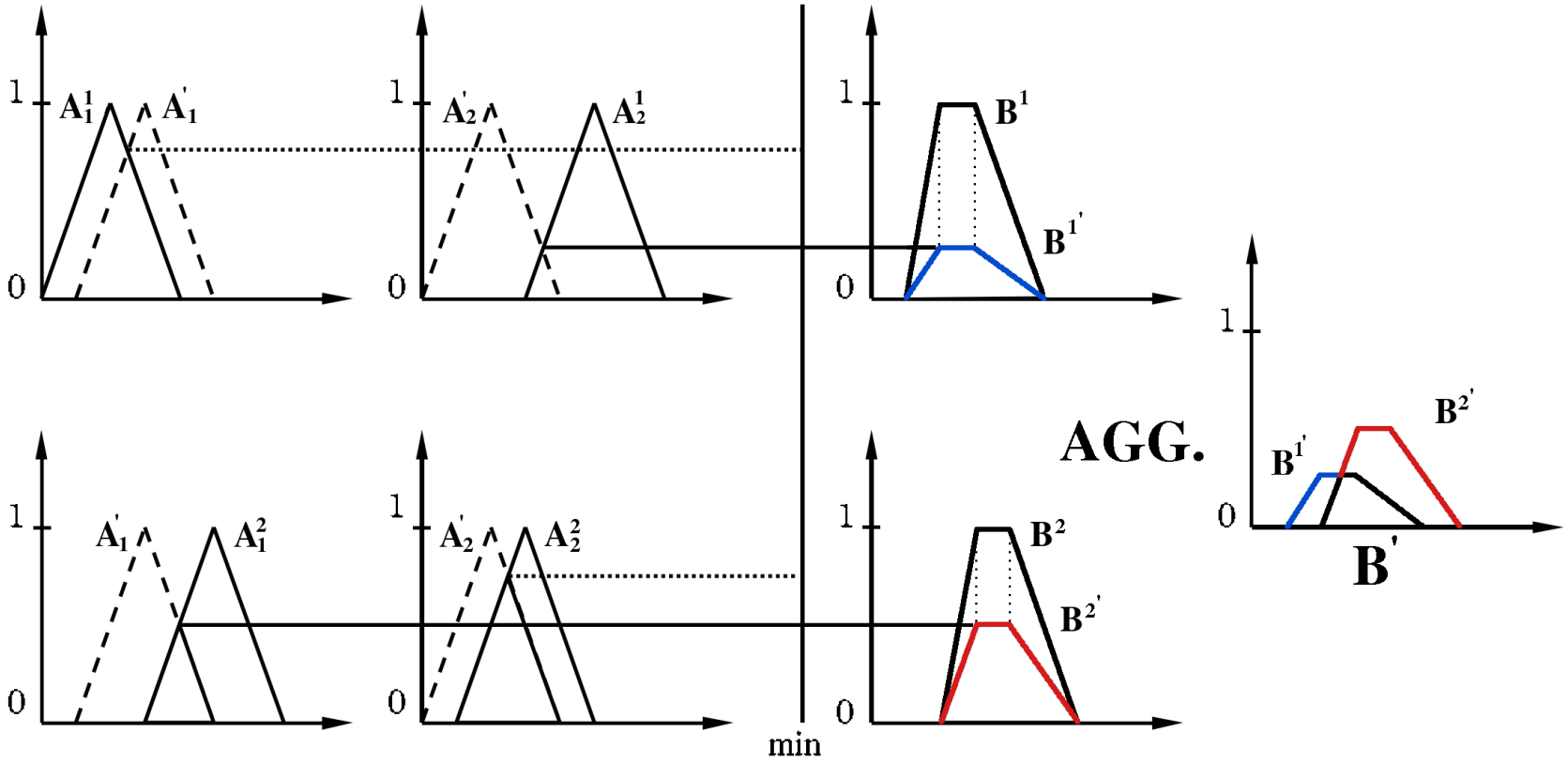
LARSEN METHOD [Larsen, 1980]

Let $\mathbb{RB}$ be a rule base.

$$B^{i\prime}(y) = \left(\min_{j=1}^{m} \bigvee_{x} \min\left(A'_j(x), A^i_j(x) \right) \right) \cdot B^i(y),$$

$$B' = \bigcup_{i=1}^{n} B^{i\prime} \text{ - the aggregation.}$$

An example of Larsen method:



IF a_1 *is* A_1^1 *and* a_2 *is* A_2^1 *THEN* b *is* B^1

IF a_1 *is* A_1^2 *and* a_2 *is* A_2^2 *THEN* b *is* B^2

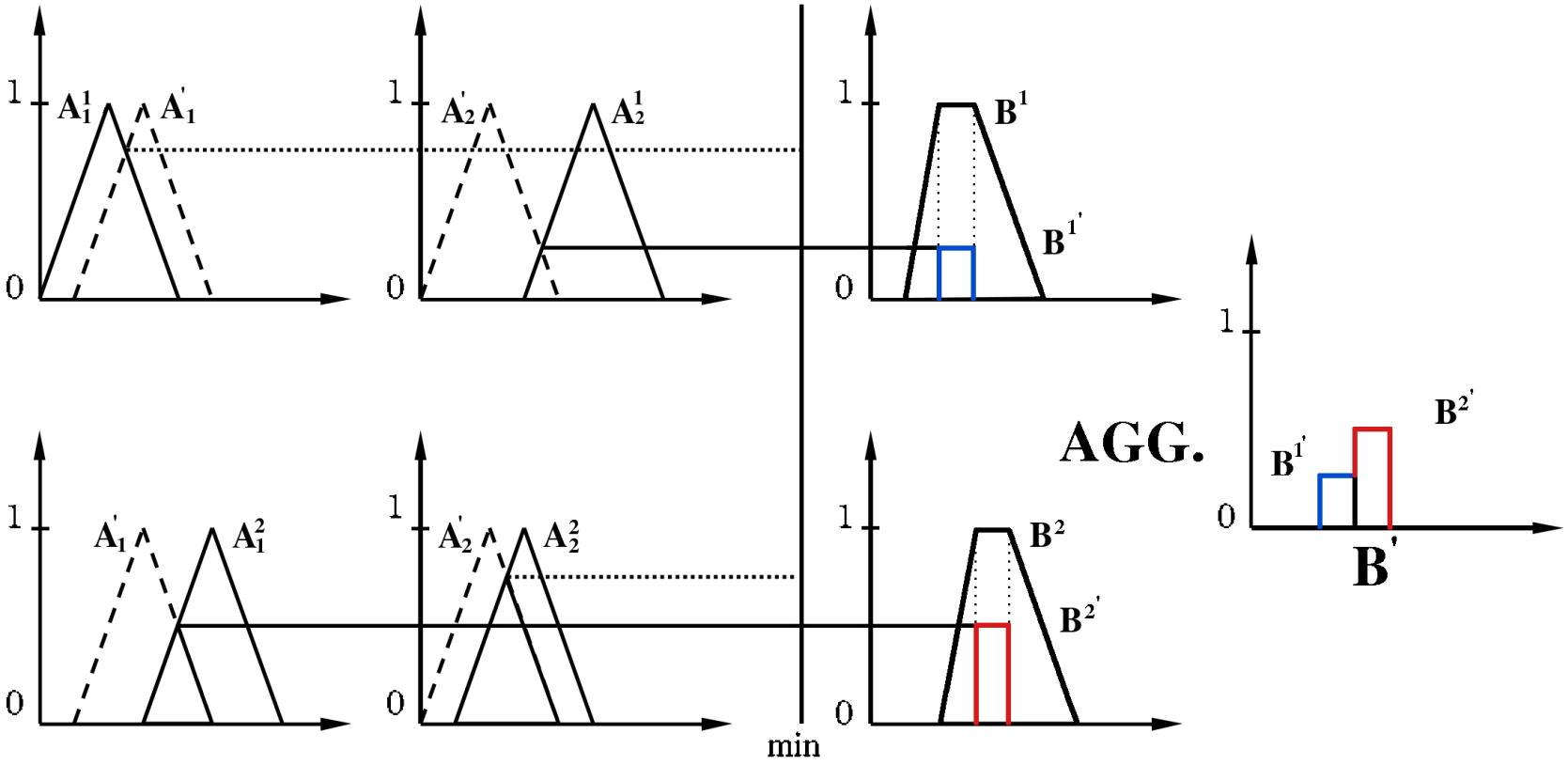
DRASTIC PRODUCT METHOD [Larsen, 1980]

Let $\mathbb{RB}$ be a rule base.

$$B^{i'}(y) = \left(\min_{j=1}^{m} \bigvee_{x} \min\left(A'_j(x), A^i_j(x) \right) \right) *_d B^i(y),$$

$$B' = \bigcup_{i=1}^{n} B^{i'} \text{ - the aggregation.}$$

An example of the drastic product method:



IF a_1 *is* A_1^1 *and* a_2 *is* A_2^1 *THEN* b *is* B^1

IF a_1 *is* A_1^2 *and* a_2 *is* A_2^2 *THEN* b *is* B^2

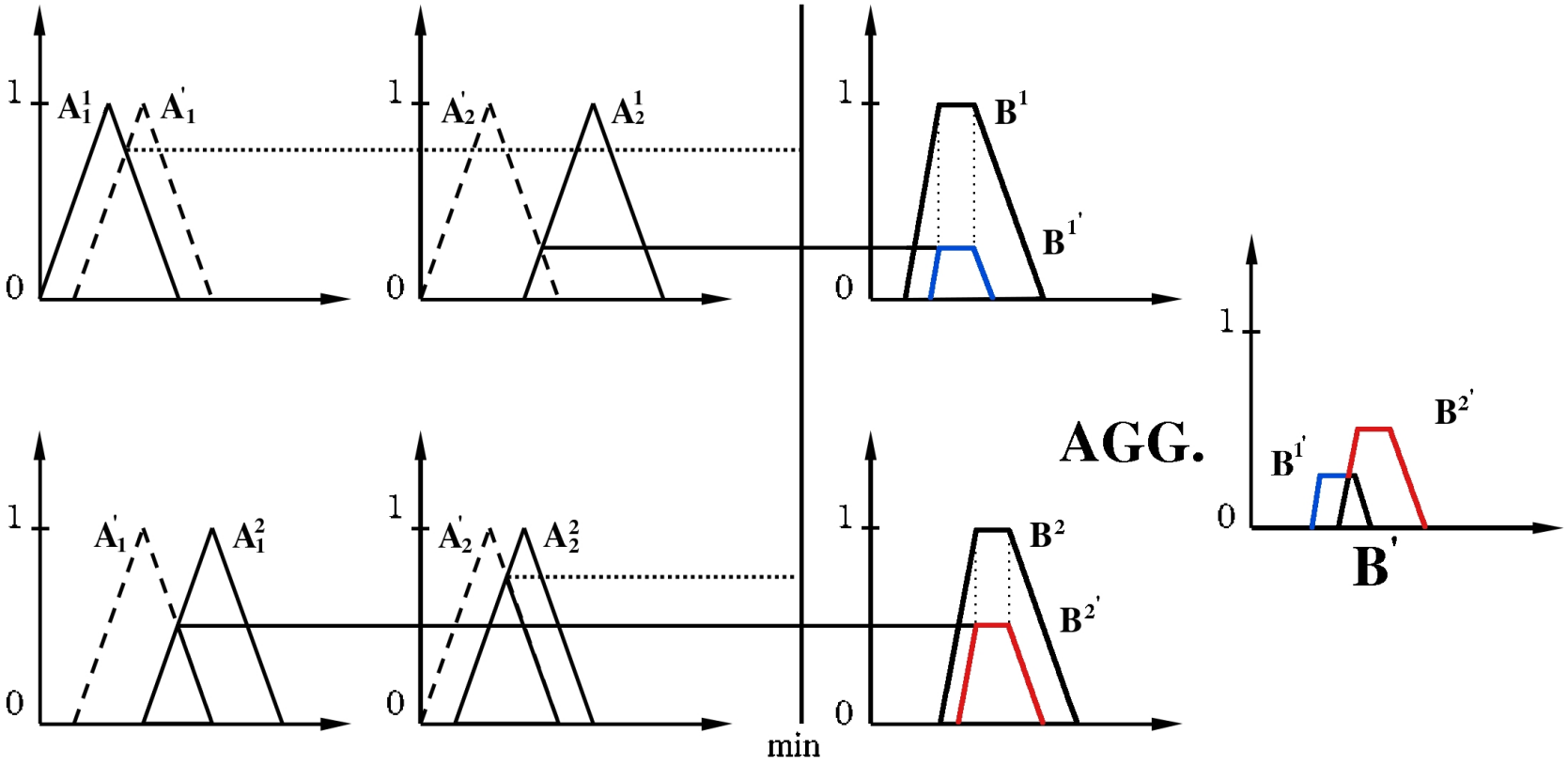
BOUNDED PRODUCT METHOD [Larsen, 1980]

Let $\mathbb{RB}$ be a rule base.

$$B^{i'}(y) = \left(\min_{j=1}^{m} \bigvee_{x} \min\left(A'_j(x), A^i_j(x) \right) \right) *_{Ł} B^i(y),$$

$$B' = \bigcup_{i=1}^{n} B^{i'} \text{ - the aggregation.}$$

An example of the bounded product method:



IF a_1 *is* A_1^1 *and* a_2 *is* A_2^1 *THEN* b *is* B^1

IF a_1 *is* A_1^2 *and* a_2 *is* A_2^2 *THEN* b *is* B^2

TSUKAMOTO METHOD [Larsen, 1980]

Let $\mathbb{RB}$ be a rule base. Let B^i are monotonic.

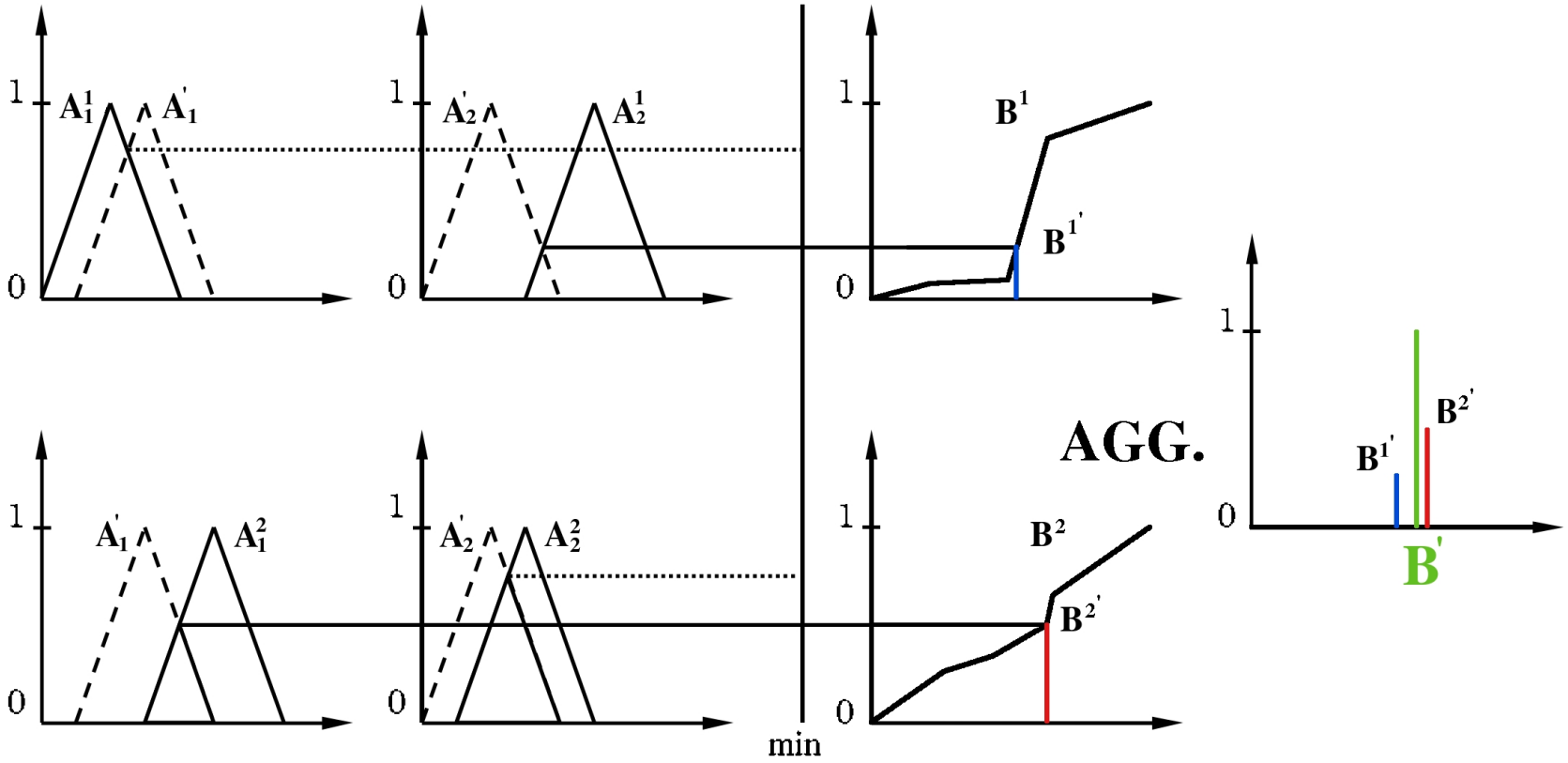
$$\alpha^i = \min_{j=1}^{m} \underbrace{\bigvee_x \min\Big(A'_j(x), A^i_j(x)\Big)}_{\alpha^i_j \text{ - the matching degree}} \quad \text{- the firing strength,}$$

$$\alpha = \sum_{i=1}^{n} \alpha^i,$$

$$B^{i\prime}(y) = \begin{cases} \alpha^i & \textit{if } y = B^{i^{-1}}(\alpha^i), \\ 0 & \textit{else}, \end{cases}$$

$$B'(y) = \begin{cases} 1 & \textit{if } y = \dfrac{\sum_{i=1}^{n} \alpha^i \cdot B^{i^{-1}}(\alpha^i)}{\alpha}, \\ 0 & \textit{else} \end{cases} \quad \text{- the aggregation.}$$

An example of Tsukamoto method:



IF a_1 *is* A_1^1 *and* a_2 *is* A_2^1 *THEN* b *is* B^1

IF a_1 *is* A_1^2 *and* a_2 *is* A_2^2 *THEN* b *is* B^2

DEFINITION

Let A_i be fuzzy sets on $\mathbb{R}$, $q_i \in \mathbb{R}$.
A fuzzy rule of the Sugeno-Takagi type is of the form

$$IF\ \tilde{a}_1\ is\ A_1\ and \cdots and\ \tilde{a}_m\ is\ A_m\ THEN\ z = q_0 + \sum_{j=1}^{m} q_j \cdot a_j$$

a_i - input crisp variables; z - an output crisp variable;
$\tilde{a}_i$ - the corresponding fuzzy variables to a_i.

DEFINITION

Let A_j^i be fuzzy sets on $\mathbb{R}$, $q_j^i \in \mathbb{R}$.
A fuzzy rule base $\mathbb{RB}$ is a set of fuzzy rules:

$$\textit{IF } \tilde{a}_1 \textit{ is } A_1^1 \textit{ and} \cdots \textit{and } \tilde{a}_m \textit{ is } A_m^1 \textit{ THEN } z = q_0^1 + \sum_{j=1}^{m} q_j^1 \cdot a_j$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$\textit{IF } \tilde{a}_1 \textit{ is } A_1^n \textit{ and} \cdots \textit{and } \tilde{a}_m \textit{ is } A_m^n \textit{ THEN } z = q_0^n + \sum_{j=1}^{m} q_j^n \cdot a_j$$

$$\textit{IF } \widetilde{\text{Distance}} \textit{ is } \text{SHORT} \textit{ and } \widetilde{\text{Speed}} \textit{ is } \text{HIGH}$$
$$\textit{THEN } \text{Braking} = 100 - 2 \cdot \text{Distance} + 5 \cdot \text{Speed}$$

SUGENO-TAKAGI-KANG METHOD [Takagi, Sugeno, 1985]

Let $\mathbb{RB}$ be a rule base.

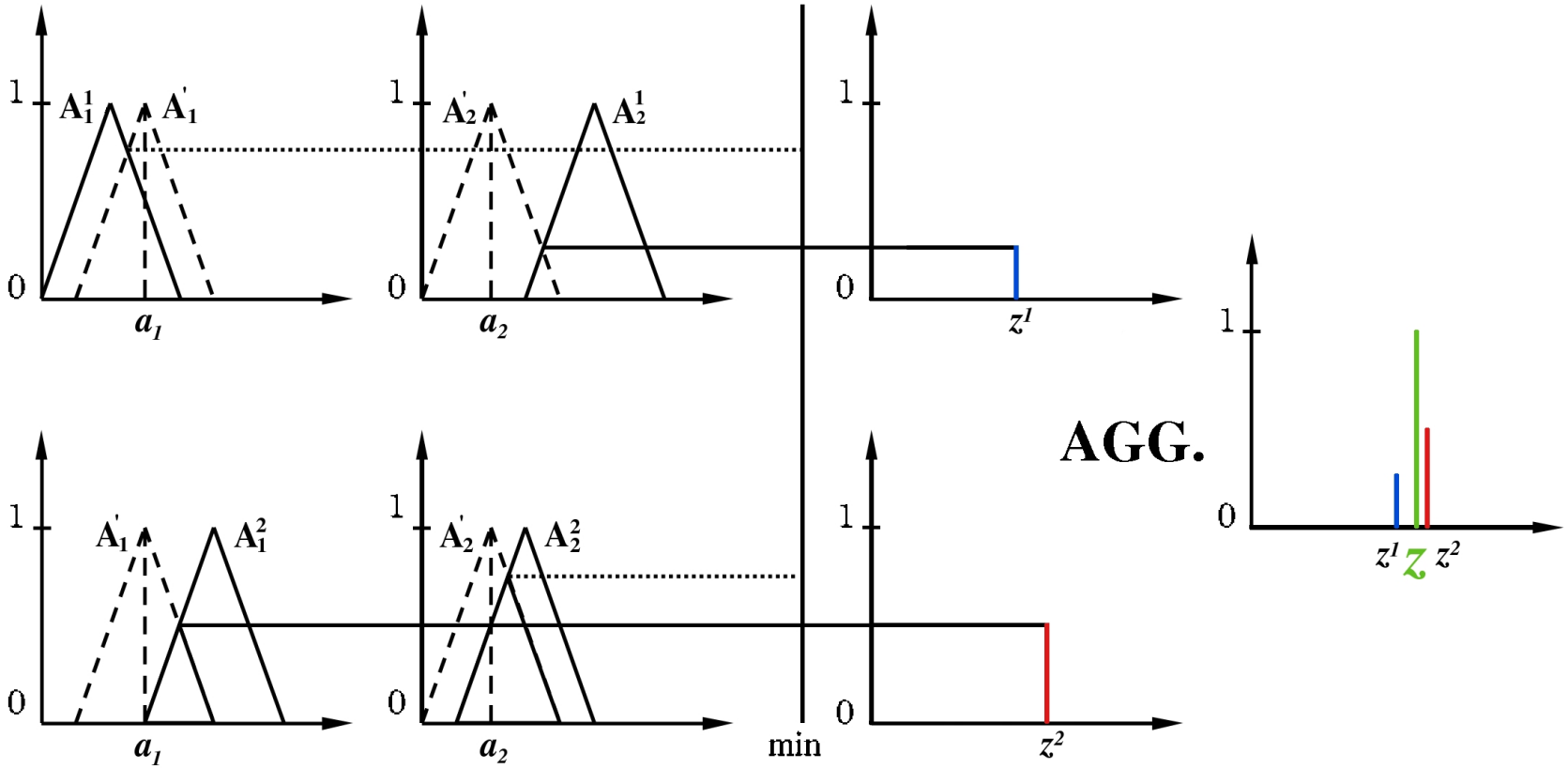
$$\alpha^i = \min_{j=1}^{m} \underbrace{\bigvee_x \min\Big(A'_j(x), A^i_j(x)\Big)}_{\alpha^i_j \text{ - the matching degree}} \quad \text{- the firing strength,}$$

$$\alpha = \sum_{i=1}^{n} \alpha^i,$$

$$z^i = q^i_0 + \sum_{j=1}^{m} q^i_j \cdot a_j,$$

$$z = \frac{\sum_{i=1}^{n} \alpha^i \cdot z^i}{\alpha} \quad \text{- the aggregation.}$$

An example of Sugeno-Takagi-Kang method:

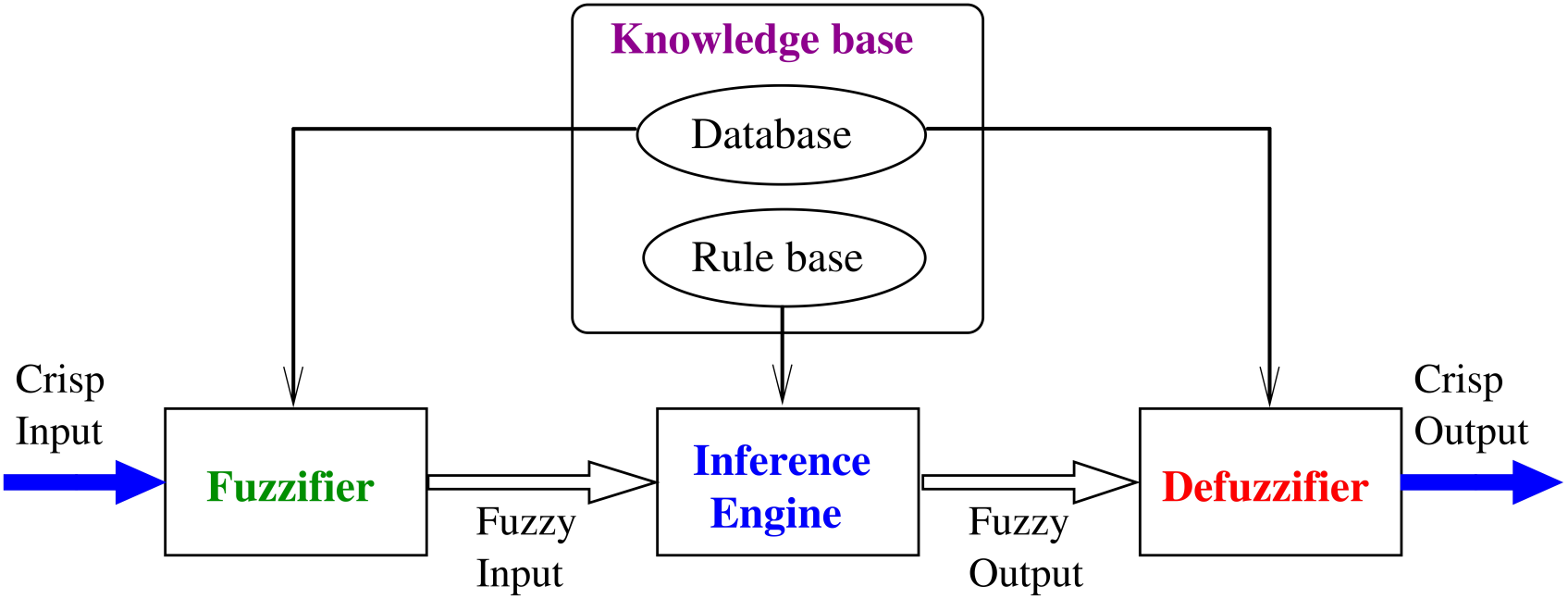


$$IF\ \tilde{a}_1\ is\ A_1^1\ and\ \tilde{a}_2\ is\ A_2^1\ THEN\ z^1 = q_0^1 + q_1^1 \cdot a_1 + q_2^1 \cdot a_2$$

$$IF\ \tilde{a}_1\ is\ A_1^2\ and\ \tilde{a}_2\ is\ A_2^2\ THEN\ z^2 = q_0^2 + q_1^2 \cdot a_1 + q_2^2 \cdot a_2$$

- Basic architecture of a *FIS*
- Design of a *FIS*
- Applications of *FIS*
- A simple fuzzy controller
  - Input/output variables and fuzzy sets
  - Rule base
  - Aggregation and defuzzification
  - Control surface for $* = \min$
  - Control surface for $* = \cdot$

Basic architecture of a *FIS*:



DESIGN OF A *FIS*

1. The definition of the problem.
2. The definition of input/output crisp/fuzzy variables.
3. The definition of fuzzy sets.
4. The definition of fuzzy rules.
5. Building (coding) a fuzzy expert system in some expert system tool (*FuzzyCLIPS*, *FuzzyShell*, *FuzzyJess*, *OPS5*, *FRIL*, *FLOPS*).
6. Testing the system.
7. Tuning the rules and fuzzy sets.

APPLICATIONS OF *FIS*

- Control engineering.
- Modelling of complex systems.
- Pattern recognition.
- Operation research.
- Decision analysis support systems.

A SIMPLE FUZZY CONTROLLER

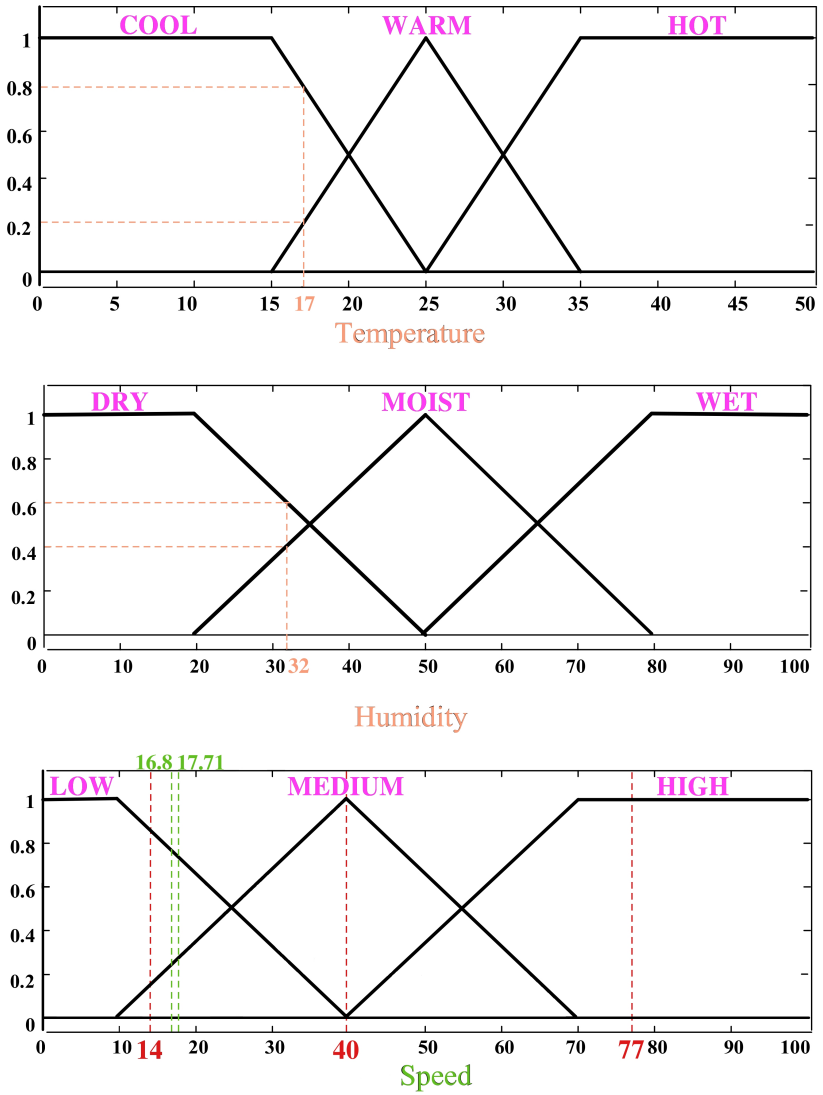
1. Design a motor speed controller for an air conditioner.
2. Input crisp variables and fuzzy sets:

   - Temperature - COOL, WARM, HOT (°C);
   - Humidity - DRY, MOIST, WET (%).

3. An output crisp variable and fuzzy sets:

   - Speed - LOW, MEDIUM, HIGH (%).

Input/output crisp variables and fuzzy sets:



A SIMPLE FUZZY CONTROLLER

(4) The rule base $\mathbb{RB}$:

IF Temperature *is* COOL *and* Humidity *is* DRY *THEN* Speed *is* LOW
IF Temperature *is* COOL *and* Humidity *is* MOIST *THEN* Speed *is* LOW
IF Temperature *is* COOL *and* Humidity *is* WET *THEN* Speed *is* MEDIUM
IF Temperature *is* WARM *and* Humidity *is* DRY *THEN* Speed *is* LOW
IF Temperature *is* WARM *and* Humidity *is* MOIST *THEN* Speed *is* MEDIUM
IF Temperature *is* WARM *and* Humidity *is* WET *THEN* Speed *is* HIGH
IF Temperature *is* HOT *and* Humidity *is* DRY *THEN* Speed *is* MEDIUM
IF Temperature *is* HOT *and* Humidity *is* MOIST *THEN* Speed *is* HIGH
IF Temperature *is* HOT *and* Humidity *is* WET *THEN* Speed *is* HIGH

A SIMPLE FUZZY CONTROLLER

(5) Aggregation and defuzzification by *WAF*:

Let $\mathbb{RB}$ consist of n rules of the form

$$IF\ x\ is\ A_i\ and\ y\ is\ B_i\ THEN\ z\ is\ C_i.$$

Let $\mu_i = A_i(x) * B_i(y)$ - the firing strength; $*$ is either $\min$ or $\cdot$.
Let c_i be the center of gravity of C_i.

The crisp output z:

$$\mu = \sum_{i=1}^{n} \mu_i,$$

$$z = \frac{\sum_{i=1}^{n} \mu_i \cdot c_i}{\mu}.$$

A SIMPLE FUZZY CONTROLLER

(6) Aggregation and defuzzification by *WAF* (an example):

Let Temperature $= 17°\mathrm{C}$ (x), Humidity $= 32\%$ (y).

$\mathrm{COOL}(17) = 0.8$, $\mathrm{WARM}(17) = 0.2$, $\mathrm{HOT}(17) = 0$.

$\mathrm{DRY}(32) = 0.6$, $\mathrm{MOIST}(32) = 0.4$, $\mathrm{WET}(32) = 0$.

$c_{\mathrm{LOW}} = 14$, $c_{\mathrm{MEDIUM}} = 40$, $c_{\mathrm{HIGH}} = 77$.

Speed $= ??\%$ (z).

Input/Output

A SIMPLE FUZZY CONTROLLER

(6) Aggregation and defuzzification by *WAF* (an example):

For $* = \min$,

$$\mu = \sum_{i=1}^{9} \mu_i = \min(0.8, 0.6) + \min(0.8, 0.4) + \min(0.8, 0) + \min(0.2, 0.6) + \min(0.2, 0.4) + \min(0.2, 0) + \min(0, 0.6) + \min(0, 0.4) + \min(0, 0)$$

$$= 0.6 + 0.4 + 0 + 0.2 + 0.2 + 0 + 0 + 0 + 0$$

$$= 1.4,$$

$$\text{Speed} = \frac{\sum_{i=1}^{9} \mu_i \cdot c_i}{\mu} = \frac{0.6 \cdot 14 + 0.4 \cdot 14 + 0 \cdot 40 + 0.2 \cdot 14 + 0.2 \cdot 40 + 0 \cdot 77 + 0 \cdot 40 + 0 \cdot 77 + 0 \cdot 77}{1.4} = \frac{24.8}{1.4}$$

$$= 17.714285714\%.$$

Input/Output

A SIMPLE FUZZY CONTROLLER

(6) Aggregation and defuzzification by *WAF* (an example):

For $* = \cdot$,

$$\mu = \sum_{i=1}^{9} \mu_i = 0.8 \cdot 0.6 + 0.8 \cdot 0.4 + 0.8 \cdot 0 + 0.2 \cdot 0.6 + 0.2 \cdot 0.4 + 0.2 \cdot 0 + 0 \cdot 0.6 + 0 \cdot 0.4 + 0 \cdot 0$$

$$= 0.48 + 0.32 + 0 + 0.12 + 0.08 + 0 + 0 + 0 + 0$$

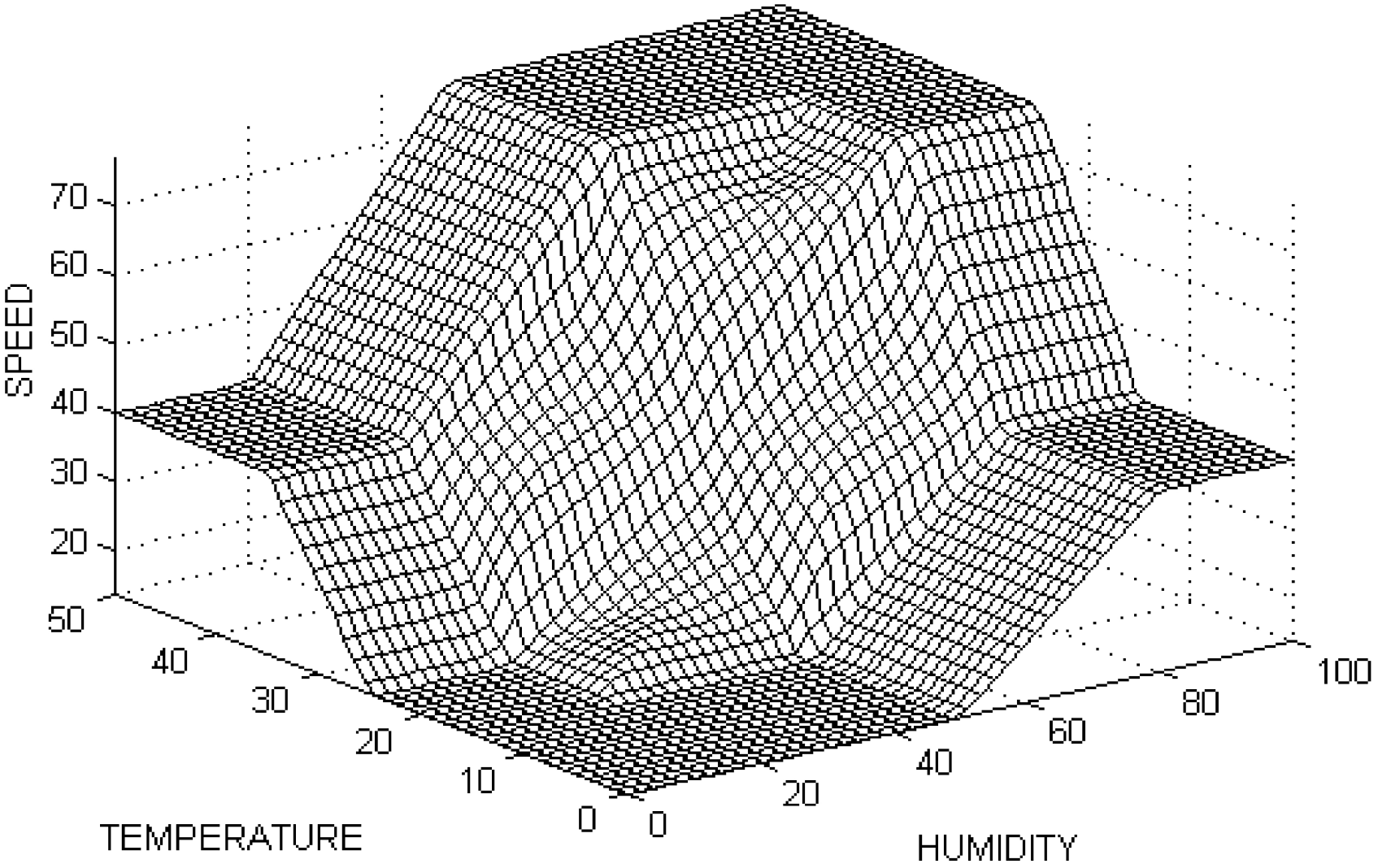
$$= 1,$$

$$\text{Speed} = \frac{\sum_{i=1}^{9} \mu_i \cdot c_i}{\mu} = \frac{0.48 \cdot 14 + 0.32 \cdot 14 + 0 \cdot 40 + 0.12 \cdot 14 + 0.08 \cdot 40 + 0 \cdot 77 + 0 \cdot 40 + 0 \cdot 77 + 0 \cdot 77}{1} = \frac{16.8}{1}$$

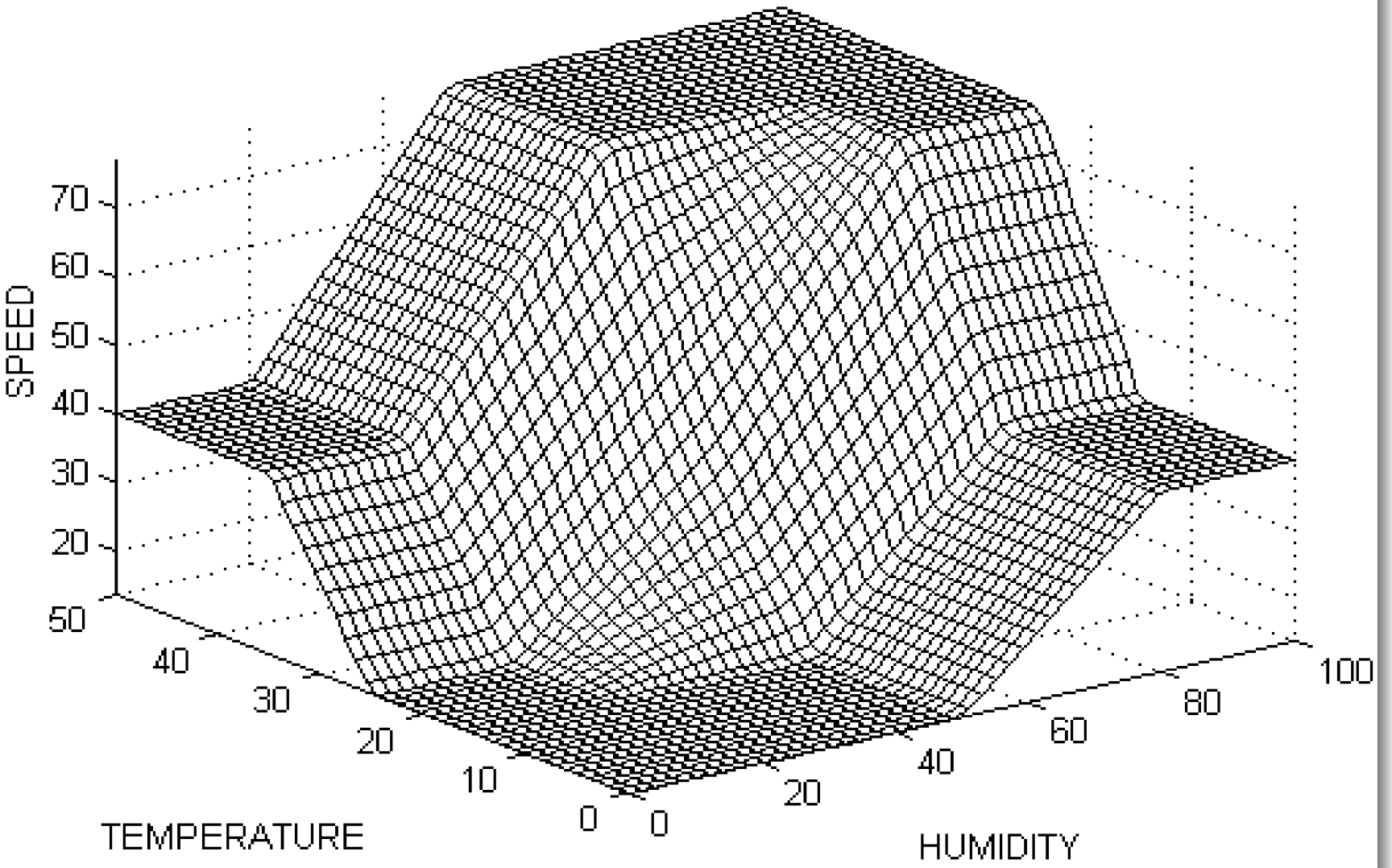
$$= 16.8\%.$$

Input/Output

Control surface for $* = \min$:



Control surface for $* = \cdot$:



Project *Synlogy*

Fuzzy expert systems - *Synlogy* - Plan:

- A bilattice approach
- Data types in *FES*
- Antecedents of fuzzy rules in *FES*
- Fuzzy rules in *FES*
- Consequents of fuzzy rules in *FES*
- Template facts
- Parallel computation in blocks
- Hardware tuning of *PC* - an example of *Synlogy*

BILATTICES [Fitting, 1988], [Ginsberg, 1988]

$$\mathscr{R} = ([0,1] \times [0,1], \leq_k, \vee_k, \wedge_k, 0|0, 1|1, \leq_t, \vee_t, \wedge_t, 0|1, 1|0),$$

$$a|b \equiv (a, b),$$

$$a|b \leq_{\mathrm{k}} c|d \Leftrightarrow a \leq c, b \leq d, \qquad \text{(knowledge order)}$$

$$a|b \vee_{\mathrm{k}} c|d = \max(a, c)|\max(b, d), \qquad \text{(knowledge supremum)}$$

$$a|b \wedge_{\mathrm{k}} c|d = \min(a, c)|\min(b, d), \qquad \text{(knowledge infimum)}$$

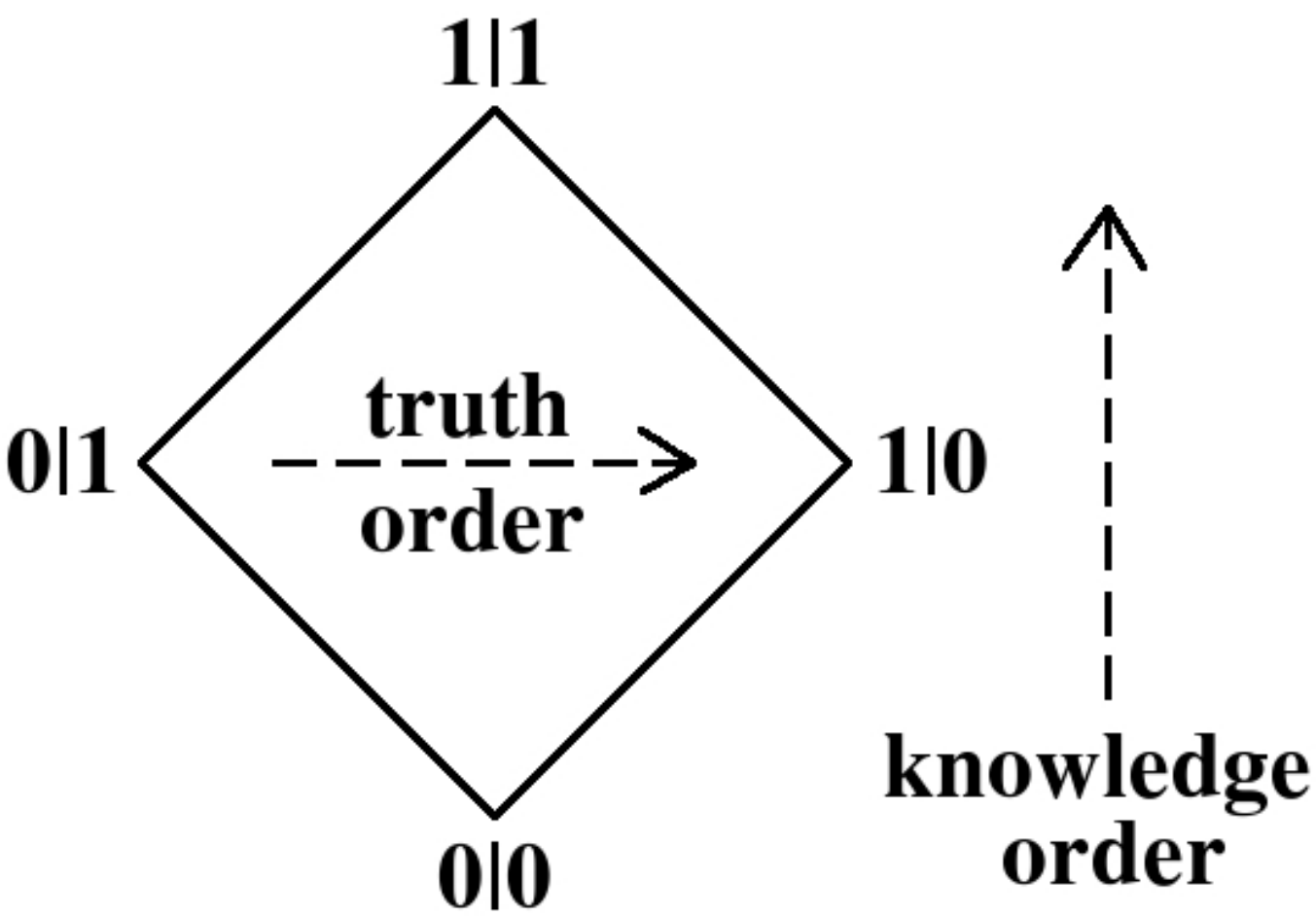
$$a|b \leq_{\mathrm{t}} c|d \Leftrightarrow a \leq c, b \geq d, \qquad \text{(truth order)}$$

$$a|b \vee_{\mathrm{t}} c|d = \max(a, c)|\min(b, d), \qquad \text{(truth supremum)}$$

$$a|b \wedge_{\mathrm{t}} c|d = \min(a, c)|\max(b, d). \qquad \text{(truth infimum)}$$

- The data type: $truth\_value \equiv [0,1]\ float \mid [0,1]\ float$;
- the constructor: $a|b$;
- the selectors: $for(a|b) = a$, $against(a|b) = b$.

Bilattice $\mathscr{R}$:



OTHER BASIC OPERATORS

$$a|b * c|d = (a * c)|(b * d), \qquad \text{(knowledge } t\text{-norm)}$$
$$a|b \Diamond c|d = (a * c)|(b \oplus d), \qquad \text{(truth } t\text{-norm)}$$
$$a|b \oplus c|d = (a \oplus c)|(b \oplus d), \qquad \text{(knowledge } t\text{-conorm)}$$
$$a|b \Box c|d = (a \oplus c)|(b * d), \qquad \text{(truth } t\text{-conorm)}$$
$$\neg a|b = b|a, \qquad \text{(negation)}$$
$$\leftrightharpoons a|b = (1 - b)|(1 - a). \qquad \text{(conflation, possibility)}$$

- The certainty factor $a|b$ - attached to a datum;
  a - an extent for the datum, b - an extent against the datum;
  accessed by $datum.cf$.
- An expression exp of $truth\_value$ is well-formed
  from atoms: $\underset{float}{exp_1}|\underset{float}{exp_2}$, $datum.cf$
  using the operators: $\vee_k$, $\wedge_k$, $\vee_t$, $\wedge_t$, $*$, $\Diamond$, $\oplus$, $\Box$, $\neg$, $\leftrightharpoons$.

PARACONSISTENCY

- $a|b$ - consistent iff $a + b \leq 1$;
- $a|b$ - exact iff $a + b = 1$;
- $a|b$ - inconsistent iff $a + b > 1$.

- $a|b$ - consistent iff $a|b \leq_{\mathrm{k}} \leftrightharpoons a|b$;
- $a|b$ - exact iff $a|b = \leftrightharpoons a|b$;
- $a|b$ - inconsistent iff $a|b >_{\mathrm{k}} \leftrightharpoons a|b$.

BASIC PROPERTIES

$$a|b \leq_k c|d \Leftrightarrow \neg a|b \leq_k \neg c|d,$$
$$a|b \leq_t c|d \Leftrightarrow \neg a|b \geq_t \neg c|d,$$
$$\neg\neg a|b = a|b,$$
$$a|b \leq_k c|d \Leftrightarrow \leftrightharpoons a|b \geq_k \leftrightharpoons c|d,$$
$$a|b \leq_t c|d \Leftrightarrow \leftrightharpoons a|b \leq_t \leftrightharpoons c|d,$$
$$\leftrightharpoons\leftrightharpoons a|b = a|b,$$
$$\neg\leftrightharpoons a|b = \leftrightharpoons\neg a|b = (1-a)|(1-b), \qquad \text{(default negation)}$$
$$\neg(a|b * c|d) = \neg a|b * \neg c|d,$$
$$\neg(a|b \oplus c|d) = \neg a|b \oplus \neg c|d,$$
$$\neg(a|b \Diamond c|d) = \neg a|b \Box \neg c|d,$$
$$\neg(a|b \Box c|d) = \neg a|b \Diamond \neg c|d,$$
$$\leftrightharpoons(a|b \vee_k c|d) = \leftrightharpoons a|b \wedge_k \leftrightharpoons c|d,$$
$$\leftrightharpoons(a|b \wedge_k c|d) = \leftrightharpoons a|b \vee_k \leftrightharpoons c|d,$$
$$\leftrightharpoons(a|b \vee_t c|d) = \leftrightharpoons a|b \vee_t \leftrightharpoons c|d,$$
$$\leftrightharpoons(a|b \wedge_t c|d) = \leftrightharpoons a|b \wedge_t \leftrightharpoons c|d.$$

BASIC DATA TYPES AND THEIR TRUTH VALUES

- Scalars: integers, floats, chars, strings;
  - $literal.cf = 1|0$.

$declare \;\; x \;\; integer$
$x := 5 \;\; x.cf := 0.8|0.1$
$7.cf = 1|0$.

- Fuzzy numbers:
  - $fnum.cf = 1|0$;
  - approximate comparison and arithmetic operators.

$declare \;\; x \;\; triang\_fnum$
$x := create\_triang\_fnum(5, 0.5)$; $\tilde{5}$ with a dispersion of 0.5
$x.cf = 1|0$.

BASIC DATA TYPES AND THEIR TRUTH VALUES

- Membership functions:
  - $memf.cf = 1|0$;
  - basic fuzzy set operators.

```
defmemftype rotation shape : polygon domain : 0 9000
declare memf rotation
memf := create_polygon_memf(0 0 1000 0.8 5000 1 9000 0)
memf.cf = 1|0.
```

- Discrete fuzzy sets:
  - a value - a vector of the membership degrees for|against of the members of a discrete fuzzy set,
  - accessed by $name.member.cf = name(member)$.

```
defdisfstype Species members : hamster rabbit hare
declare species1 Species
species1 :=
    create_disfs(hamster 0.3|0.4 rabbit 0.6|0.2 hare 0.8|0.2)
species1.rabbit.cf = 0.6|0.2.
```

BASIC DATA TYPES AND THEIR TRUTH VALUES

- Numerical discrete fuzzy sets:
  - membership functions - attached to the members, accessed by *name*.*member*;
  - linguistic terms - members of a numerical discrete fuzzy set;
  - a value of a linguistic variable - a vector of the membership degrees for|against of the linguistic terms, accessed by *name*.*member*.*cf* = *name*(*member*).

```
defmemftype H_Length shape : polygon domain : 0 250
defnumfstype Hum_Length members : small medium tall
                                memftype : H_Length
Hum_Length.small := create_polygon_memf(0 1 160 0 250 0)
Hum_Length.medium := create_polygon_memf(0 0 170 1 250 0)
Hum_Length.tall := create_polygon_memf(0 0 170 0 250 1)
declare length_1 Hum_Length
length_1 := create_numfs(small 0.2|0.5 medium 0.6|0.2 tall 0|0.9)
length_1.medium.cf = 0.6|0.2  length_1.medium = (0 0 170 1 250 0).
```

Antecedents of fuzzy rules in *FES*

Basic conditional elements (*CE*)

- *assigned*(x)
  variable
- exp_1 | exp_2 $datum.cf$
  float *float*
- exp_1 *crisp_comparison* exp_2
  $scalar_1$ $scalar_2$
- exp_1 *comparison* exp_2
  $fnumtype_1$ $fnumtype_2$
- exp_1 *comparison* exp_2
  $memftype_1$ $memftype_2$
- x *is member*
  fstype
- x *is member*
  numfstype
- exp_1 *is* exp_2
  float *memftype*
- exp exp_1 *crisp_comparison* exp_2
  truth_value *truth_value* *truth_value*

BASIC CONDITIONAL ELEMENTS (*CE*)

Let P be a proposition. $\nu(P)$ - the truth value $a|b$ of P.
Let exp be an expression. We shall identify exp with its value.

- $P: \quad assigned(\underset{variable}{x})$

  - x - a variable:

$$\nu(P) = \begin{cases} x.cf \text{ if } x \text{ has assigned a value,} \\ 0|0 \quad else \end{cases}$$

$declare \;\; x \; y \;\; integer \qquad x := 35 \;\; x.cf := 0.8|0.1$

$declare \;\; fnum_1 \;\; fnum_2 \;\; triang\_fnum$

$fnum_1 := create\_triang\_fnum(5, 0.5)$

$P_1: \quad assigned(x) \quad P_2: \quad assigned(y)$

$P_3: \quad assigned(fnum_1) \quad P_4: \quad assigned(fnum_2)$

$\nu(P_1) = 0.8|0.1 \;\; \nu(P_2) = 0|0 \;\; \nu(P_3) = 1|0 \;\; \nu(P_4) = 0|0.$

BASIC CONDITIONAL ELEMENTS (*CE*)

Let *exp* be an expression.

The certainty factor $\xi(exp)$ of *exp*:

$$\xi(exp) = \begin{cases} x.cf & \text{if } exp \equiv x \text{ - a variable of } scalar, \\ 1|0 & \text{if } exp \text{ - a literal}, \\ 1|0 & \text{if } exp \text{ - of a fuzzy number or membership function type}, \\ 1|0 & \text{if } exp \equiv datum.cf, \\ 1|0 & \text{if } exp \equiv assigned(x), \\ \overset{n}{\underset{1}{\Diamond}} \xi(exp_i) & \text{if } exp \equiv \gamma(exp_1, \ldots, exp_n),\ n \geq 1, \\ & \gamma \not\equiv assigned \text{ - an operator.} \end{cases}$$

BASIC CONDITIONAL ELEMENTS (*CE*)

- $P_1 : \quad \underset{float}{exp_1} | \underset{float}{exp_2} \qquad\qquad P_2 : \quad datum.cf$

  - exp_i - an expression of *float*:

$$\nu(P_1) = \xi(exp_1) \lozenge exp_1 | exp_2 \lozenge \xi(exp_2),$$
$$\nu(P_2) = datum.cf.$$

BASIC CONDITIONAL ELEMENTS (*CE*)

$declare \;\; x \;\; y \;\; float$

$x := 0.5 \;\; x.cf := 0.8|0.1 \;\; y := 0.4 \;\; y.cf := 0.7|0.2$

$species_1 :=$

$\quad create\_disfs(hamster\ 0.3|0.6\ rabbit\ 0.6|0.2\ hare\ 0.8|0.2)$

$length_1 := create\_numfs(small\ 0.2|0.8\ medium\ 0.7|0.3\ tall\ 0|0)$

$P_1 : \quad x|y$

$P_2 : \quad x.cf$

$P_3 : \quad species_1.rabbit.cf$

$P_4 : \quad length_1.medium.cf$

$\nu(P_1) = x.cf \lozenge x|y \lozenge y.cf = 0.8|0.1 \lozenge 0.5|0.4 \lozenge 0.7|0.2$

$\nu(P_2) = x.cf = 0.8|0.1$

$\nu(P_3) = species_1.rabbit.cf = 0.6|0.2$

$\nu(P_4) = length_1.medium.cf = 0.7|0.3$

for $* = \min, \oplus = \max, \lozenge = \lozenge_m : \quad \nu(P_1) = 0.5|0.4.$

BASIC CONDITIONAL ELEMENTS (*CE*)

- P : $\underset{scalar_1}{exp_1}$ *crisp_comparison* $\underset{scalar_2}{exp_2}$

  - exp_i - an expression of $scalar_i$;
  - *crisp_comparison* - $=, <, >, \ldots$:

$$\nu(P) = \begin{cases} \xi(exp_1) \Diamond \xi(exp_2) & \text{if } exp_1 \text{ crisp\_comparison } exp_2, \\ \neg(\xi(exp_1) \Diamond \xi(exp_2)) & \text{else.} \end{cases}$$

BASIC CONDITIONAL ELEMENTS (*CE*)

declare $x\ y\ z$ *integer*

$x := 35 \quad x.cf := 0.8|0.1 \quad y := 35 \quad y.cf := 0.6|0.2$

$z := 33 \quad z.cf := 0.7|0.1$

$P_1: \quad x = y$

$P_2: \quad x < z$

$P_3: \quad x + y > y + z$

$$\nu(P_1) = 0.8|0.1 \lozenge 0.6|0.2$$

$$\nu(P_2) = \neg(0.8|0.1 \lozenge 0.7|0.1)$$

$$\nu(P_3) = (\underbrace{0.8|0.1 \lozenge 0.6|0.2}_{\xi(x+y)}) \lozenge (\underbrace{0.6|0.2 \lozenge 0.7|0.1}_{\xi(y+z)})$$

for $* = \cdot, \oplus = \oplus_p, \lozenge = \lozenge_p$: $\quad \nu(P_1) = 0.48|0.28$

$$\nu(P_2) = 0.19|0.56$$

$$\nu(P_3) = 0.2016|0.4816.$$

BASIC CONDITIONAL ELEMENTS (*CE*)

- $P: \quad \underset{fnumtype_1}{exp_1} \quad comparison \quad \underset{fnumtype_2}{exp_2}$

  - exp_i - an expression of $fnumtype_i$;
  - $comparison$ - $\sim=$, $\sim<$, $\sim<=$:

$$\nu(P) = cmp|(1 - cmp),$$

$$cmp = fnum_1 \; comparison \; fnum_2$$

$declare \;\; x \; y \; z \;\; triang\_fnum$

$x := create\_triang\_fnum(5, 2)$

$y := create\_triang\_fnum(7, 2)$

$z := create\_triang\_fnum(10, 2)$

$P_1: \;\; x \sim= y \;\; P_2: \;\; x \sim= z \;\; P_3: \;\; x \sim< y \;\; P_4: \;\; z \sim<= y$

$\nu(P_1) = 0.5|0.5 \;\; \nu(P_2) = 0|1 \;\; \nu(P_3) = 1|0 \;\; \nu(P_4) = 0.25|0.75.$

BASIC CONDITIONAL ELEMENTS (*CE*)

- $P: \quad \underset{memftype_1}{exp_1} \quad comparison \quad \underset{memftype_2}{exp_2}$

  - exp_i - an expression of $memftype_i$;
  - $comparison$ - $\subset$, $\subseteq$, is :

$$\nu(P) = cmp|(1 - cmp),$$
$$cmp = exp_1 \; comparison \; exp_2,$$
$$exp_1 \; is \; exp_2 = height(exp_1 \cap exp_2)$$

declare $memf_1$ $memf_2$ *rotation*

$memf_1 := create\_polygon\_memf(0\ 0\ 1000\ 0\ 5000\ 1\ 9000\ 1)$

$memf_2 := create\_polygon\_memf(0\ 1\ 500\ 1\ 2500\ 0\ 9000\ 0)$

$P_1: \quad memf_1 \; is \; memf_2 \quad P_2: \quad memf_1 \subset memf_2$

$\nu(P_1) = 0.25|0.75 \quad \nu(P_2) = 0|1.$

BASIC CONDITIONAL ELEMENTS (*CE*)

- $P: \quad \underset{fstype}{x} \quad is\ member$

  - x - a variable of *fstype*;
  - *member* - a member of a (numerical) discrete fuzzy set of *fstype*:

$$\nu(P) = x.member.cf$$

declare $species_1$ *Species*

$species_1 :=$
$\quad create\_disfs(hamster\ 0.3|0.6\ rabbit\ 0.6|0.2\ hare\ 0.8|0.1)$

declare $length_1$ *H_Length*

$length_1 := create\_numfs(small\ 0.2|0.7\ medium\ 0.6|0.2\ tall\ 0|1)$

$P_1: \quad species_1\ is\ hamster \quad P_2: \quad length_1\ is\ small$

$\nu(P_1) = 0.3|0.6 \quad \nu(P_2) = 0.2|0.7.$

BASIC CONDITIONAL ELEMENTS (*CE*)

- $P: \quad \underset{numfstype}{x} \quad is\ member$

  - x - a variable of *numfstype*;
  - *member* - a member of a numerical discrete fuzzy set of *numfstype*:

$$\nu(P) = cmp|(1 - cmp),$$

$$cmp = (x.member.cf * x.member)\ is\ numfstype.member,$$

$$a|b * A(u) = \begin{cases} \frac{a * A(u) * \gamma + b * (1 - A(u)) * \delta}{a+b} & if\ a + b > 0, \\ \frac{1}{2} & else, \end{cases}$$

A - a fuzzy set, $\gamma, \delta \in [0, 1]$.

BASIC CONDITIONAL ELEMENTS (*CE*)

$Hum\_Length.small := create\_polygon\_memf(0\ 1\ 160\ 0\ 250\ 0)$

$declare\ \ length_1\ \ Hum\_Length$

$length_1 := create\_numfs(small\ 0.2|0.7\ \ medium\ 0.6|0.2\ \ tall\ 0|1)$

$length_1.small := create\_polygon\_memf(0\ 1\ 160\ 0.3\ 250\ 0)$

$P_1 :\ \ length_1\ is\ small \quad P_2 :\ \ length_1\ is\ small$

$\nu(P_1) = 0.2|0.7$

$$\begin{aligned} cmp &= (length_1.small.cf * length_1.small)\ is\ Hum\_Length.small \\ &= (0.2|0.7 * length_1.small)\ is\ Hum\_Length.small \end{aligned}$$

$\nu(P_2) = cmp|(1 - cmp)$

for $* = \cdot, \gamma = \delta = 1 :\quad 0.2|0.7 \cdot length_1.small = (0\ \frac{2}{9}\ 160\ \frac{5.5}{9}\ 250\ \frac{7}{9})$

$$\begin{aligned} cmp &= (0\ \tfrac{2}{9}\ 160\ \tfrac{5.5}{9}\ 250\ \tfrac{7}{9})\ is\ (0\ 1\ 160\ 0\ 250\ 0) \\ &= 1 - \tfrac{7}{12.5} = 1 - 0.56 = 0.44 \end{aligned}$$

$\nu(P_2) = 0.44|0.56.$

BASIC CONDITIONAL ELEMENTS (*CE*)

- $P: \quad \underset{float}{exp_1} \; is \; \underset{memftype}{exp_2}$

  - exp_1 - an expression of *float*;
  - exp_2 - an expression of *memftype*:

$$\nu(P) = \xi(exp_1) \Diamond exp_2(exp_1)|(1 - exp_2(exp_1))$$

declare $x \; y \; z$ *float*
$x := 500 \;\; x.cf := 0.8|0.1 \;\; y := 6 \;\; y.cf := 0.3|0.5 \;\; z := 120 \;\; z.cf := 0.6|0.1$
declare *memf* *rotation*
$memf := create\_polygon\_memf(0\ 0\ 1000\ 0.8\ 5000\ 1\ 9000\ 0)$
declare *fnum* *triang_fnum* $\quad fnum := create\_triang\_fnum(5, 2)$
$Hum\_Length.small := create\_polygon\_memf(0\ 1\ 160\ 0\ 250\ 0)$

$P_1: \quad x \; is \; memf \quad P_2: \quad y \; is \; fnum \quad P_3: \quad z \; is \; Hum\_Length.small$

$\nu(P_1) = 0.8|0.1 \Diamond 0.4|0.6 \quad \nu(P_2) = 0.3|0.5 \Diamond 0.5|0.5$
$\nu(P_3) = 0.6|0.1 \Diamond 0.25|0.75$

for $\Diamond = \Diamond_p: \quad \nu(P_1) = 0.32|0.64 \quad \nu(P_2) = 0.15|0.75 \quad \nu(P_3) = 0.15|0.775.$

BASIC CONDITIONAL ELEMENTS (*CE*)

- P^* : $\underset{float}{exp_1}$ *is* $p\_modifier$ $\underset{memftype}{exp_2}$

  P^+ : $\underset{float}{exp_1}$ *is* $s\_modifier$ $\underset{memftype}{exp_2}$

  - exp_1 - an expression of *float*;
  - exp_2 - an expression of *memftype*:

$$\nu(P^*) = \xi(exp_1) \lozenge exp_2(exp_1)^p | (1 - exp_2(exp_1)^p),$$
$$\nu(P^+) = \xi(exp_1) \lozenge exp_2(exp_1 + s) | (1 - exp_2(exp_1 + s)).$$

BASIC CONDITIONAL ELEMENTS (*CE*)

$x := 500 \quad x.cf := 0.8|0.1 \quad y := 120 \quad y.cf := 0.6|0.3$

$memf := create\_polygon\_memf(0\ 0\ 1000\ 0.8\ 5000\ 1\ 9000\ 0)$

$Hum\_Length.small := create\_polygon\_memf(0\ 1\ 160\ 0\ 250\ 0)$

$P_1: \quad x\ is\ very^*\ memf$

$P_2: \quad y\ is\ slightly^*\ Hum\_Length.small$

$P_3: \quad x\ is\ very^+\ memf$

$P_4: \quad y\ is\ slightly^+\ Hum\_Length.small$

$$\nu(P_1) = 0.8|0.1 \lozenge 0.4^2|(1-0.4^2)$$

$$\nu(P_2) = 0.6|0.3 \lozenge 0.25^{\frac{1}{3}}|(1-0.25^{\frac{1}{3}})$$

$$\begin{aligned}\nu(P_3) &= 0.8|0.1 \lozenge memf(500-100)|(1-memf(500-100)) \\ &= 0.8|0.1 \lozenge 0.32|(1-0.32)\end{aligned}$$

$$\begin{aligned}\mu &= Hum\_Length.small(120-20) \\ \nu(P_4) &= 0.6|0.3 \lozenge \mu|(1-\mu) = 0.6|0.3 \lozenge 0.375|0.625\end{aligned}$$

for $\lozenge = \lozenge_p$: $\quad \nu(P_1) = 0.128|0.856 \quad \nu(P_2) \doteq 0.378|0.5590$

$\nu(P_3) = 0.256|0.712 \quad \nu(P_4) = 0.225|0.7375.$

BASIC CONDITIONAL ELEMENTS (*CE*)

- $P: \quad \underset{float}{exp_1} \; is \; d\_modifier \; \underset{float}{exp_2}$

  - exp_i - an expression of *float*:

$$\mu = create\_fnum(exp_2, d)(exp_1),$$
$$\nu(P) = \xi(exp_1) \Diamond \mu | (1-\mu) \Diamond \xi(exp_2)$$

$x := 97.5 \quad x.cf := 0.8|0.1 \quad y := 100 \quad y.cf := 0.7|0.2$

$P_1: \quad x \; is \; nearly \; y \quad P_2: \quad x \; is \; approximately \; y$

$$\mu_1 = create\_triang\_fnum(y, 5)(x) = 0.5|0.5$$
$$\nu(P_1) = 0.8|0.1 \Diamond \mu_1|(1-\mu_1) \Diamond 0.7|0.2 = 0.8|0.1 \Diamond 0.5|0.5 \Diamond 0.7|0.2$$

$$\mu_2 = create\_triang\_fnum(y, 10)(x) = 0.75|0.25$$
$$\nu(P_2) = 0.8|0.1 \Diamond \mu_2|(1-\mu_2) \Diamond 0.7|0.2 = 0.8|0.1 \Diamond 0.75|0.25 \Diamond 0.7|0.2$$

for $\Diamond = \Diamond_p$: $\quad \nu(P_1) = 0.28|0.64 \quad \nu(P_2) = 0.42|0.46.$

BASIC CONDITIONAL ELEMENTS (*CE*)

- P_1 : $\underset{truth\_value}{exp}$

  P_2 : $\underset{truth\_value}{exp_1}$ $crisp\_comparison$ $\underset{truth\_value}{exp_2}$

  - exp, exp_i - an expression of $truth\_value$;
  - $crisp\_comparison$ - $=$, $\leq_k$, $<_k$, $\leq_t$, $<_t$:

$$\nu(P_1) = exp \lozenge \xi(exp),$$

$$\nu(P_2) = \begin{cases} \xi(exp_1) \lozenge \xi(exp_2) & \text{if } exp_1 \; crisp\_comparison \; exp_2, \\ \neg(\xi(exp_1) \lozenge \xi(exp_2)) & \text{else.} \end{cases}$$

BASIC CONDITIONAL ELEMENTS (*CE*)

$x := 0.5 \quad x.cf := 0.9|0.1 \quad y := 0.4 \quad y.cf := 0.8|0.2$

$species_1 :=$
$\quad create\_disfs(hamster\ 0.3|0.6\ rabbit\ 0.6|0.2\ hare\ 0.8|0.2)$

$length_1 := create\_numfs(small\ 0.2|0.8\ medium\ 0.6|0.2\ tall\ 0|0)$

$P_1: \quad species_1.hamster.cf \leq_k x|y$

$P_2: \quad species_1.rabbit.cf >_t 0.5|0.3$

$P_3: \quad x|y >_k \leftrightharpoons x|y$

$P_4: \quad species_1.rabbit.cf = length_1.tall.cf$

$\nu(P_1) = \neg(0.9|0.1 \lozenge 0.8|0.2); \quad 0.3|0.6 \not\leq_k 0.5|0.4$

$\nu(P_2) = 1|0 \lozenge 1|0 = 1|0; \quad 0.6|0.2 >_t 0.5|0.3$

$\nu(P_3) = \neg((0.9|0.1 \lozenge 0.8|0.2) \lozenge (0.9|0.1 \lozenge 0.8|0.2)); 0.5|0.4 \not>_k 0.6|0.5$

$\nu(P_4) = \neg(1|0 \lozenge 1|0) = 0|1; \quad 0.6|0.2 \neq 0|0$

for $\lozenge = \lozenge_p: \quad \nu(P_1) = 0.28|0.72 \quad \nu(P_3) = 0.4816|0.5184.$

ANTECEDENT OF A FUZZY RULE

- Basic connectives and their interpretation:

and	$*$	*t_and*	$\Diamond$	*not*	$\neg$
or	$\oplus$	*t_or*	$\Box$	*pos*	$\leftrightharpoons$

- A literal l is either a CE or $cn_1\ CE$ or $cn_1\ cn_2\ CE$ where $cn_i \in \{not, pos\}$.
- A clause Cl is an expression well-formed from literals using *or*, *t_or*.
- An antecedent Ant is an expression well-formed from clauses using *and*, *t_and*.
- $\nu(l)$, $\nu(Cl)$, $\nu(Ant)$ - the truth value of l, Cl, Ant
  - defined on the structure of the respective formula.

BASIC FUZZY RULE STRUCTURE

A fuzzy rule is of the form

$$IF\ antecedent\ THEN_{th}^{cf}\ consequent$$

- *antecedent* - an antecedent;
- $cf = a|b$ - the certainty (neccessity) factor of the fuzzy rule;
- $pos = \leftrightharpoons a|b$ - the possibility of the fuzzy rule;
- $th = c|d$ - the threshold of the fuzzy rule;
- *consequent* - the consequent of a fuzzy rule - a sequence of actions: *BCA* and other program instructions (control, I/O,...);
- $\nu(antecedent)$ - the matching degree of the fuzzy rule;
- $fs = \nu(antecedent) \Diamond cf$ - the firing strength of the fuzzy rule.
- If $fs \geq_t th$ - the fuzzy rule fires.

$$IF\ distance\ is\ short\ and\ speed\ is\ high\ THEN_{0.6|0.4}^{0.8|0.1}\ braking := hard$$

Consequents of fuzzy rules in *FES*

Basic consequent actions (*BCA*)

- $\underset{variable}{x} := \underset{scalar}{exp_1}$ $\quad \underset{variable}{y}.cf := \underset{truth\_value}{exp_2}$
- $\underset{variable}{x} := \underset{memftype}{exp}$
- $\underset{fstype}{x} := create\_disfs(.), create\_numfs(.)$

  $\underset{fstype}{y}$ is $member_1$

  $\underset{numfstype}{z}$ is $member_2$
- $\underset{numfstype}{x} := fuzzify(\underset{float}{u})$

  $\underset{numfstype}{y} := fuzzify(\underset{fnumtype}{v})$

  $\underset{float}{z} := defuzzify(\underset{numfstype}{w}, method)$

BASIC CONSEQUENT ACTIONS (*BCA*)

- Crisp assignments:

$$A_1 : \underset{variable}{x} := \underset{scalar}{exp_1} \qquad A_2 : \underset{variable}{y}.cf := \underset{truth\_value}{exp_2}$$

- x - a variable of *scalar*;
- y - a variable of *truth_value*;
- exp_1 - an expression of *scalar*;
- exp_2 - an expression of *truth_value*:

$$fs : \; IF \; antecedent \; THEN_{th}^{cf} \; A_1, A_2$$

$$A_1 : \; x = exp_1,$$
$$x.cf = \xi(exp_1),$$
$$A_2 : \; y.cf = exp_2 \lozenge \xi(exp_2).$$

BASIC CONSEQUENT ACTIONS (*BCA*)

declare x y w z *integer*

$x := 5 \quad x.cf := 0.8|0.1 \quad y := 4 \quad y.cf := 0.7|0.2$

$A_1: \quad w := x + y$

$A_2: \quad z.cf := \frac{x}{10}|\frac{y}{10}$

$A_1: \quad w = 5 + 4 = 9 \quad w.cf = (0.8|0.1 \lozenge 0.7|0.2)$

$A_2: \quad z.cf = \frac{5}{10}|\frac{4}{10} \lozenge (0.8|0.1 \lozenge 0.7|0.2)$

for $\lozenge = \lozenge_p$:

$w.cf = 0.56|0.28$

$z.cf = 0.5|0.4 \lozenge 0.56|0.28 = 0.28|0.568.$

BASIC CONSEQUENT ACTIONS (*BCA*)

- Fuzzy assignments:

$$F: \quad \underset{variable}{x} := \underset{memftype}{exp}$$

- x - a variable of *memftype*;
- exp - an expression of *memftype*:

$$fs: \; IF \; antecedent \; THEN_{th}^{cf} \; F_1$$

$$F: \; x = exp,$$
$$x.cf = 1|0.$$

BASIC CONSEQUENT ACTIONS (*BCA*)

defmemftype *rotation* *shape* : *polygon* *domain* : 0 9000

defnumfstype *Motor_Speed* *members* : *slow* *medium* *fast*
memftype : *rotation*

declare $memf_1$ $memf_2$ $memf_3$ *rotation*

declare $speed_1$ *Motor_Speed*

$memf_1 := create\_polygon\_memf(0\ 0\ 2000\ 0\ 4000\ 1\ 9000\ 1)$

$memf_2 := create\_polygon\_memf(0\ 0\ 2500\ 0\ 3500\ 1\ 9000\ 1)$

F_1 : $memf_3 := memf_1 \cup memf_2$

F_2 : $speed_1.fast := very\ memf_3$

F_1 : $memf_3 = (0\ 0\ 2500\ 0\ 3000\ 0.5\ 3500\ 1\ 9000\ 1)$
$memf_3.cf = 1|0$

F_2 : $speed_1.fast = (0\ 0\ 2500\ 0\ 3000\ 0.25\ 3500\ 1\ 9000\ 1)$
$speed_1.fast.cf = 1|0.$

BASIC CONSEQUENT ACTIONS (*BCA*)

- Fuzzy assignments:

$$F_1: \quad \underset{fstype}{x} := create\_disfs(.), create\_numfs(.)$$

$$F_2: \quad \underset{fstype}{y} \; is\ member_1$$

$$F_3: \quad \underset{numfstype}{z} \; is\ member_2$$

- x, y - variables of *fstype*;
- z - a variable of *numfstype*;
- $member_1$ - a member of a (numerical) discrete fuzzy set of *fstype*;
- $member_2$ - a member of a numerical discrete fuzzy set of *numfstype*.

BASIC CONSEQUENT ACTIONS (*BCA*)

$$fs: \; IF \; antecedent \; THEN_{th}^{cf} \; F_1, F_2, F_3$$

$$F_1: \; x = create\_disfs(.), create\_numfs(.),$$
$$x.cf = 1|0,$$

$$F_2: \; y.member_1.cf = fs,$$
$$y.cf = 1|0,$$

$$F_3: \; z.member_2 = \nu(antecedent) * numfstype.member_2,$$
$$z.member_2.cf = cf,$$
$$z.cf = 1|0.$$

BASIC CONSEQUENT ACTIONS (*BCA*)

$$\textit{declare}\ \ \textit{species}_1\ \textit{species}_2\ \ \textit{Species}$$

$$\textit{declare}\ \ \textit{length}_1\ \textit{length}_2\ \ \textit{Hum\_Length}$$

$$\textit{Hum\_Length}.\textit{small} := \textit{create\_polygon\_memf}(0\ 1\ 160\ 0.5\ 250\ 0)$$

$$F_1: \quad \textit{species}_1\ \textit{species}_2 := \textit{create\_disfs}(\textit{hamster}\ 0.3|0.6\ \textit{rabbit}\ 0.6|0.2\ \textit{hare}\ 0.8|0.1)$$

$$F_2: \quad \textit{length}_1\ \textit{length}_2 := \textit{create\_numfs}(\textit{small}\ 0.2|0.7\ \textit{medium}\ 0.6|0.2\ \textit{tall}\ 0|0.9)$$

$$F_3: \quad \textit{species}_2\ \textit{is hamster}$$

$$F_4: \quad \textit{length}_2\ \textit{is small}$$

$$F_1: \quad \textit{species}_1\ \textit{species}_2 = (\textit{hamster}\ 0.3|0.6\ \textit{rabbit}\ 0.6|0.2\ \textit{hare}\ 0.8|0.1)$$

$$F_1: \quad \textit{species}_1.cf = 1|0\ \ \textit{species}_2.cf = 1|0$$

$$F_2: \quad \textit{length}_1\ \textit{length}_2 = (\textit{small}\ 0.2|0.7\ \textit{medium}\ 0.6|0.2\ \textit{tall}\ 0|0.9)$$

$$F_2: \quad \textit{length}_1.cf = 1|0\ \ \textit{length}_2.cf = 1|0.$$

BASIC CONSEQUENT ACTIONS (*BCA*)

$$F_3: \quad species_2.hamster.cf = fs = \nu(antecedent) \lozenge cf$$
$$F_3: \quad species_2.cf = 1|0$$

$$F_4: \quad length_2.small = \nu(antecedent) * Hum\_Length.small$$
$$F_4: \quad length_2.small.cf = cf$$
$$F_4: \quad length_2.cf = 1|0$$

for $\nu(antecedent) = 0.7|0.2, cf = 0.9|0.1, * = \cdot, \gamma = \delta = 1, \lozenge = \lozenge_p$:

$$F_3: \quad species_2.hamster.cf = 0.7|0.2 \lozenge 0.9|0.1 = 0.63|0.28$$

$$F_4: \quad length_2.small = 0.7|0.2 \cdot (0\ 1\ 160\ \tfrac{1}{2}\ 250\ 0)$$
$$= (0\ \tfrac{7}{9}\ 160\ \tfrac{1}{2}\ 250\ \tfrac{2}{9})$$
$$F_4: \quad length_2.small.cf = 0.9|0.1.$$

BASIC CONSEQUENT ACTIONS (*BCA*)

- Fuzzification, defuzzification:

$$F_1: \quad \underset{numfstype}{x} := fuzzify(\underset{float}{u})$$

$$F_2: \quad \underset{numfstype}{y} := fuzzify(\underset{fnumtype}{v})$$

$$A: \quad \underset{float}{z} := defuzzify(\underset{numfstype}{w}, method)$$

- x, y, w - variables of *numfstype*;
- u, z - variables of *float*;
- v - a variable of *fnumtype*;
- *method* - a method of defuzzification (*COG*, *MOM*,...).

BASIC CONSEQUENT ACTIONS (*BCA*)

$$fs: \; IF \; antecedent \; THEN^{cf}_{th} \; F_1, F_2, A$$

$$F_1: \; x = fuzzify(u) \diamond u.cf,$$

$$[A \diamond a|b](m) = A(m) \diamond a|b; \; A \text{ - a discrete fuzzy set},$$

$$x.cf = 1|0,$$

$$F_2: \; y = fuzzify(v),$$

$$y.cf = 1|0,$$

$$F_3: \; z = defuzzify(w); \; \text{the defuzzification of} \bigcup_{m \text{ in } w} w.m.cf * w.m,$$

$$z.cf = fs.$$

BASIC CONSEQUENT ACTIONS (*BCA*)

defmemftype *Celsius* *shape* : *polygon* *domain* : 0 100

defnumfstype *Temperature* *members* : *low* *medium* *high*
memftype : *Celsius*

declare $temp_1$ $temp_2$ *Temperature*

Temperature.*low* := *create_polygon_memf* (0 1 50 0 100 0)

Temperature.*medium* := *create_polygon_memf* (0 0 50 1 100 0)

Temperature.*high* := *create_polygon_memf* (0 0 50 1 100 1).

BASIC CONSEQUENT ACTIONS (*BCA*)

- Fuzzification:

$declare \;\; x \;\; y \;\; float$

$x := 40 \;\; x.cf := 0.7|0.2$

$declare \;\; fnum \;\; triang\_fnum$

$fnum := create\_triang\_fnum(40, 20)$

$F_1: \quad temp_1 := fuzzify(x)$

$F_2: \quad temp_2 := fuzzify(fnum)$

$F_1: \quad temp_1 = (low \; 0.2|0.8 \; medium \; 0.8|0.2 \; high \; 0|1) \Diamond 0.7|0.2$

picture

$F_2: \quad temp_2 = (low \; 0.42|0.58 \; medium \; 0.88|0.12 \; high \; 0.14|0.86)$

picture

for $\Diamond = \Diamond_p$:

$F_1: \quad temp_1 = (low \; 0.14|0.84 \; medium \; 0.56|0.36 \; high \; 0|1).$

BASIC CONSEQUENT ACTIONS (*BCA*)

- Defuzzification:

declare z_1 z_2 *float*

A_1 : $z_1 := defuzzify(temp_2, COG)$

A_2 : $z_2 := defuzzify(temp_2, MOM)$

for $fs = 0.9|0.1, * = \min, \gamma = 1, \delta = 0$:

$$\bigcup_{m \in \{low, medium, high\}} temp_2.m.cf * temp_2.m =$$

$\min(0.42|0.58, (0\ 1\ 50\ 0\ 100\ 0)) \cup \min(0.88|0.12, (0\ 0\ 50\ 1\ 100\ 0)) \cup$

$\min(0.14|0.86, (0\ 0\ 50\ 1\ 100\ 1)) =$

$(0\ 0.42\ 29\ 0.42\ 50\ 0\ 100\ 0) \cup (0\ 0\ 44\ 0.88\ 56\ 0.88\ 100\ 0) \cup (0\ 0\ 7\ 0.14\ 100\ 0.14) =$

$(0\ 0.42\ 21\ 0.42\ 44\ 0.88\ 56\ 0.88\ 93\ 0.14\ 100\ 0.14)$

A_1 : $z_1 = 43$ $z_1.cf = 0.9|0.1$ picture

A_2 : $z_2 = 50$ $z_2.cf = 0.9|0.1$. picture

DEFINITION

deftemplate	*name_of_fact_template*	
slot	$name_1$	$scalar_1$
multislot	$name_2$	$scalar_2$
slot	$name_3$	$truth\_value_3$
multislot	$name_4$	$truth\_value_4$
slot	$name_5$	$fnumtype_5$
slot	$name_6$	$memftype_6$
slot	$name_7$	$disfstype_7$
slot	$name_8$	$numfstype_8$

- *deftemplate* - the definition of a fact template;
- *slot* - a single field,
- *multislot* - a sequence of fields of $scalar_i$, $truth\_value_i$;
- *name_of_fact_template*.$name_i$.*cf* - the certainty factor $a|b$ of the *slot* *name_of_fact_template*.$name_i$ of $scalar_i$.

ASSERTION AND RETRACTION

assert	*name_of_fact_template*	
$name_1$	exp_1	(an expression of $scalar_1$)
$name_2$	seq_2	(a sequence of expressions of $scalar_2$)
$name_3$	exp_3	(an expression of $truth\_value_3$)
$name_4$	seq_4	(a sequence of expressions of $truth\_value_4$)
$name_5$	exp_5	(an expression of $fnumtype_5$)
$name_6$	exp_6	(an expression of $memftype_6$)
$name_7$	x_7	(a variable of $disfstype_7$)
$name_8$	x_8	(a variable of $numfstype_8$)

- *assert* - including an individual fact of a defined fact template to the fact list with the unique fact number *fn* (pointer);
- x_i - a single field/value variable of $scalar_i/truth\_value_i$;
- $\$x_i$ - a sequence of fields/values variable of $scalar_i/truth\_value_i$;
- for a slot $name\_of\_fact\_template.name_i$ of $scalar_i$ and an expression exp_i of $scalar_i$, $name\_of\_fact\_template.name_i.cf = \xi(exp_i)$.
- *retract*(*fn*) - retracting the individual fact with *fn*.

BASIC PATTERN ELEMENTS (*PE*) FOR MATCHING

$\mathrm{I} \leftarrow / not \quad name\_of\_fact\_template$

$name_1$	exp_1	(an expression of $scalar_1$)
$name_2$	seq_2	(a sequence of expressions of $scalar_2$)
$name_3$	exp_3	(an expression of $truth\_value_3$)
$name_4$	seq_4	(a sequence of expressions of $truth\_value_4$)
$name_5$	exp_5	(an expression of $fnumtype_5$)
$name_6$	exp_6	(an expression of $memftype_6$)
$name_7$	x_7	(a variable of $disfstype_7$)
$name_8$	x_8	(a variable of $numfstype_8$)

- I - *fn* of a matched fact / *not* - all the variables in *PE* instantiated;
- for a single value variable x_i of $scalar_i$ and
  a slot $name\_of\_fact\_template.name_i$ of $scalar_i$,
  $x_i.cf = name\_of\_fact\_template.name_i.cf$.
- The access to (multi)slots, their certainty factors - by $\mathrm{I}.name_i(.cf)$.
- $retract(\mathrm{I})$ - retracting the individual fact with $fn = \mathrm{I}$.

DEFINITION AND ASSERTION

```
deftemplate car
        slot  type      string
        slot  engine    float
        slot  age       age_of_car
        slot  condition condition_of_car

fn = 1: assert car
                type      golf                        1|0
                engine    1.4                       0.9|0.1
                age       (new 0.9|0.1 medium 0.2|0.6 old 0|1)
                condition (good 0.7|0.1 medium 0.5|0.2 bad 0.1|0.8)

fn = 2: assert car
                type      passat                      1|0
                engine    2.0                       0.8|0.2
                age       (new 0.6|0.3 medium 0.5|0.3 old 0|1)
                condition (good 0.5|0.1 medium 0.5|0.3 bad 0.1|0.9)
```

PATTERN MATCHING AND RETRACTION

$declare\ \ x\ y\ \ float$

$declare\ \ age_1\ age_2\ \ age\_of\_car$

$declare\ \ cond_1\ cond_2\ \ condition\_of\_car$

$\mathrm{I} \leftarrow car$

$type$	$golf$	
$engine$	x	$(x = 1.4)$
age	age_1	$(age_1 = (new\ 0.9\|0.1\ medium\ 0.2\|0.6\ old\ 0\|1))$
$condition$	$cond_1$	$(cond_1 = (good\ 0.7\|0.1\ medium\ 0.5\|0.2\ bad\ 0.1\|0.8))$

$\mathrm{I} = 1\ \ x.cf = 0.9|0.1\ \ \mathrm{I}.type = golf\ \ \mathrm{I}.type.cf = 1|0\ \ retract(\mathrm{I})$

$\mathrm{J} \leftarrow car$

$type$	$passat$	
$engine$	y	$(y = 2.0)$
age	age_2	$(age_2 = (new\ 0.6\|0.3\ medium\ 0.5\|0.3\ old\ 0\|1))$
$condition$	$cond_2$	$(cond_2 = (good\ 0.5\|0.1\ medium\ 0.5\|0.3\ bad\ 0.1\|0.9))$

$\mathrm{J} = 2\ \ y.cf = 0.8|0.2\ \ \mathrm{J}.type = passat\ \ \mathrm{J}.type.cf = 1|0\ \ retract(\mathrm{J}).$

BASIC BLOCK STRUCTURE

$$
\begin{array}{l}
\textit{block} \;\; \textit{name\_of\_block} \;\; \textit{priority} \;\; P \\
\quad \textit{definitions of types} \\
\quad \textit{declarations of variables} \\
\quad \textit{preamble instructions} \; (\textit{variable assignments}) \\
\qquad\qquad\qquad\qquad \vdots \\
\quad \textit{IF} \;\; \textit{patern}_i : \textit{antecedent}_i \;\; \textit{THEN}^{cf_i}_{th_i} \;\; \textit{consequent}_i : \textit{agg} \; \bar{x}_i : \textit{assertion}_i \\
\qquad\qquad\qquad\qquad \vdots \\
\quad \textit{ending instructions} \; (\textit{control}, \; I/O, \; \textit{focus}, \ldots)
\end{array}
$$

- $P \in [-10000, 10000]$;
- $\textit{patern}_i$ - a sequence of *PE*'s;
- $\textit{consequent}_i$ may contain *retract*'s of old facts;
- $\textit{assertion}_i$ - a sequence of *assert*'s of new facts;
- $\bar{x}_i$ - the aggregated variables in $\textit{assertion}_i$.

- *FES* - a collection of blocks.

AN INFERENCE CYCLE

(1) The execution of the *preamble instructions*.

(2) Matching the rule *paterns* againts the fact list; one rule may have several matches.

(3) The evaluation of the rule *antecedents* for successful matches.

(4) The execution of the rule *consequents* with respect to the evaluated *antecedents* with possible retraction of old facts from the fact list.

(5) The aggregation of selected block variables.

(6) The assertion of new facts with respect to aggregated block variables.

(7) The execution of the *ending instructions*.

An example of parallel computation.

AN INFERENCE CYCLE

- *Vars* - the set of all the variables in the block.
- *R* - the set of all the rules in the block.

Let $r \in R$.

- $pattern\_vars(r)$ - the set of all the variables in the $patern(r)$;
- $antecedent\_vars(r)$ ..... in the $antecedent(r)$;
- $consequent\_vars(r)$ ..... in the $consequent(r)$;
- $assert\_vars(r)$ ..... in the $assertion(r)$;
- $agg\_vars(r) = \bar{x}(r) \subseteq assert\_vars(r)$ where $agg\ \bar{x}(r)$;

  $*\_vars(r) \subseteq Vars$.

- $\sigma : dom(\sigma) \longrightarrow \bigcup_{t \text{ is a type}} t$ - a variable assignment,

  $dom(\sigma) \subseteq Vars$ - the domain of σ;
  if $x \in dom(\sigma)$ is of *scalar*, then $x.cf \in dom(\sigma)$.

AN INFERENCE CYCLE

1. The output of the *preamble instructions*: σ_p.

2. The output of matching: $\sigma_1, \ldots, \sigma_n$, $R_1, \ldots, R_n$, $\sigma_i \supseteq \sigma_p$, $dom(\sigma_i) = dom(\sigma_p) \cup \bigcup_{r \in R_i} pattern\_vars(r)$, $\emptyset \neq R_i \subseteq R$;

   σ_i - an input variable assignment;
   R_i - the set of all the matched rules by σ_i against the fact list;

   σ_i, R_i - both maximal; there are no proper super sets of them satisfying some match.

3. The output of the evaluation of *antecedents*: $\forall 1 \leq i \leq n$, $r \in R_i$, $\nu(antecedent(r))$ with respect to σ_i; $dom(\sigma_i) \supseteq antecedent\_vars(r)$.

AN INFERENCE CYCLE

4. The output of the evaluation of *consequents*:
   $\forall 1 \leq i \leq n,\ r \in R_i^A = \{r \mid r \in R_i, fs_r \geq_t th_r\}$ wrt. σ_i,
   $R_i^A \subseteq R_i$ - the subset of all activated rules;

   $$\rho_r^i,\ dom(\rho_r^i) = dom(\sigma_i) \cup consequent\_vars(r) \supseteq assert\_vars(r)$$

   - an output variable assignment with respect to σ_i, $\nu(antecedent(r))$, cf_r, fs_r.

AN INFERENCE CYCLE

5. The output of aggregation:
$\forall 1 \leq i \leq n,\ r \in R_i^A,$

$\xi_r^i,\ dom(\xi_r^i) = assert\_vars(r) \supseteq agg\_vars(r)$

- an aggregated output variable assignment with respect to $agg\_vars(r)$ defined as follows:

- $fixed\_vars(r) = assert\_vars(r) - agg\_vars(r)$.

Let $x \in agg\_vars(r) \subseteq assert\_vars(r) = dom(\xi_r^i) \subseteq dom(\rho_r^i)$.

- $aggregation(\rho_r^i, x) = \{\rho_s^j \mid 1 \leq j \leq n, s \in R_j^A,$
$fixed\_vars(r) \cup \{x\} \subseteq dom(\rho_s^j),$
$\rho_r^i|_{fixed\_vars(r)} = \rho_s^j|_{fixed\_vars(r)}\}$.

AN INFERENCE CYCLE

5. For $x \in fixed\_vars(r)$,

$$\xi_r^i(x) = \rho_r^i(x);$$

for $x \in agg\_vars(r)$ of *scalar*,

$$\xi_r^i(x) = \rho_r^i(x),$$

$$\xi_r^i(x.cf) = \Box\{\rho_s^j(x.cf) \mid \rho_s^j \in aggregation(\rho_r^i, x), \rho_s^j(x) = \rho_r^i(x)\};$$

for $x \in agg\_vars(r)$ of *truth_value*,

$$\xi_r^i(x) = \Box\{\rho_s^j(x) \mid \rho_s^j \in aggregation(\rho_r^i, x)\};$$

for $x \in agg\_vars(r)$ of *memftype*,

$$\xi_r^i(x) = \oplus\{\rho_s^j(x) \mid \rho_s^j \in aggregation(\rho_r^i, x)\};$$

for $x \in agg\_vars(r)$ of *disfstype*,

$$\xi_r^i(x.m.cf) = \Box\{\rho_s^j(x.m.cf) \mid \rho_s^j \in aggregation(\rho_r^i, x)\}, \ m \text{ in } x;$$

for $x \in agg\_vars(r)$ of *numfstype*,

$$\xi_r^i(x.m) = \oplus\{\rho_s^j(x.m) \mid \rho_s^j \in aggregation(\rho_r^i, x)\}, \ m \text{ in } x,$$

$$\xi_r^i(x.m.cf) = \Box\{\rho_s^j(x.m.cf) \mid \rho_s^j \in aggregation(\rho_r^i, x)\}, \ m \text{ in } x.$$

AN INFERENCE CYCLE

6. The output of assertion:
   $\forall 1 \leq i \leq n$, $r \in R_i^A$, new facts asserted by *assertion*(r) with respect to ξ_r^i; $dom(\xi_r^i) = assert\_vars(r)$.
7. The output of the *ending instructions*:
   e.g. control, I/O, focus actions.

AN INFERENCE CYCLE

```
defmemftype H_Age shape : polygon domain : 0 100

defnumfstype Hum_Age members : young medium old
                      memftype : H_age

Hum_Age.young := create_polygon_memf(0 1 50 0 100 0)
Hum_Age.medium := create_polygon_memf(0 0 50 1 100 0)
Hum_Age.old := create_polygon_memf(0 0 50 1 100 1)

deftemplate person
        slot name     string
        slot siblings integer
        slot age      Hum_Age
```

AN INFERENCE CYCLE

block b_1 *priority* 0

declare x *string* s *integer* a *Hum_Age*

r_1 *IF* *person* : 1|0
name x
siblings s
age a

THEN$^{0.9|0.1}_{0.3|0.5}$ *agg* s, a : *assert* *conc_of_person*
name x
siblings s
age a

AN INFERENCE CYCLE

fn = 1:	*assert*	*person*	
	name	*John*	1|0
	siblings	1	0.3|0.1
	age	(*young* 0.3|0.4 *medium* 0.4|0.3 *old* 0.2|0.1)	
fn = 2:	*assert*	*person*	
	name	*John*	1|0
	siblings	3	0.4|0.2
	age	(*young* 0.5|0.4 *medium* 0.3|0.3 *old* 0.2|0.1)	
fn = 3:	*assert*	*person*	
	name	*John*	1|0
	siblings	3	0.6|0.3
	age	(*young* 0.6|0.2 *medium* 0.5|0.3 *old* 0.2|0.1)	
fn = 4:	*assert*	*person*	
	name	*Mary*	1|0
	siblings	3	0.5|0.5
	age	(*young* 0.2|0.4 *medium* 0.7|0.2 *old* 0.2|0.4)	

AN INFERENCE CYCLE

1. The output of the *preamble instructions*: $\sigma_p = \emptyset$.
2. The output of matching:

$$\sigma_1 = \{x = John, x.cf = 1|0, s = 1, s.cf = 0.3|0.1, a = (young\ 0.3|0.4\ medium\ 0.4|0.3\ old\ 0.2|0.1)\} \cup \{a.m = Hum\_Age.m \mid m \text{ in } a\};$$

$$\sigma_2 = \{x = John, x.cf = 1|0, s = 3, s.cf = 0.4|0.2, a = (young\ 0.5|0.4\ medium\ 0.3|0.3\ old\ 0.2|0.1)\} \cup \{a.m = Hum\_Age.m \mid m \text{ in } a\};$$

$$\sigma_3 = \{x = John, x.cf = 1|0, s = 3, s.cf = 0.6|0.3, a = (young\ 0.6|0.2\ medium\ 0.5|0.3\ old\ 0.2|0.1)\} \cup \{a.m = Hum\_Age.m \mid m \text{ in } a\};$$

$$\sigma_4 = \{x = Mary, x.cf = 1|0, s = 3, s.cf = 0.5|0.5, a = (young\ 0.2|0.4\ medium\ 0.7|0.2\ old\ 0.2|0.4)\} \cup \{a.m = Hum\_Age.m \mid m \text{ in } a\};$$

$$R_1 = R_2 = R_3 = R_4 = \{r_1\}.$$

AN INFERENCE CYCLE

3. The output of the evaluation of *antecedents*:
$\nu(antecedent(r_1)) = 1|0$ with respect to σ_1, σ_2, σ_3, σ_4.

4. The output of the evaluation of *consequents*:
for $i = 1, \ldots, 4$, $\rho^i_{r_1} = \sigma_i$, $R^A_i = R_i = \{r_1\}$;
$fs_{r_1} = \nu(antecedent(r_1)) \lozenge cf_{r_1} = 1|0 \lozenge 0.9|0.1 = 0.9|0.1$
$\geq_t 0.3|0.5 = th_{r_1}$.

AN INFERENCE CYCLE

5. The output of the aggregation $agg\ s, a$:
for $* = \cdot$, $\oplus = \oplus_p$, $\Box = \Box_p$:

$$\xi^1_{r_1} = \{x = John, x.cf = 1|0, s = 1, s.cf = 0.3|0.1, a = (young\ 0.86|0.032\ medium\ 0.79|0.027\ old\ 0.488|0.001)\} \cup \{a.m = Hum\_Age.m \mid m \text{ in } a\};$$

$$\xi^2_{r_1} = \{x = John, x.cf = 1|0, s = 3, s.cf = 0.76|0.06, a = (young\ 0.86|0.032\ medium\ 0.79|0.027\ old\ 0.488|0.001)\} \cup \{a.m = Hum\_Age.m \mid m \text{ in } a\};$$

$$\xi^3_{r_1} = \{x = John, x.cf = 1|0, s = 3, s.cf = 0.76|0.06, a = (young\ 0.86|0.032\ medium\ 0.79|0.027\ old\ 0.488|0.001)\} \cup \{a.m = Hum\_Age.m \mid m \text{ in } a\};$$

$$\xi^4_{r_1} = \{x = Mary, x.cf = 1|0, s = 3, s.cf = 0.5|0.5, a = (young\ 0.2|0.4\ medium\ 0.7|0.2\ old\ 0.2|0.4)\} \cup \{a.m = Hum\_Age.m \mid m \text{ in } a\}.$$

AN INFERENCE CYCLE

6. The output of assertion:
   new facts asserted by *assertion*(r_1) with respect to $\xi^1_{r_1}$, $\xi^2_{r_1} = \xi^3_{r_1}$, $\xi^4_{r_1}$:

$fn = 5$: *assert* *conc_of_person*
name *John* 1|0
siblings 1 0.3|0.1
age (*young* 0.86|0.032 *medium* 0.79|0.027 *old* 0.488|0.001)
age.m *Hum_Age.m*, *m* in *Hum_Age*

$fn = 6$: *assert* *conc_of_person*
name *John* 1|0
siblings 3 0.76|0.06
age (*young* 0.86|0.032 *medium* 0.79|0.027 *old* 0.488|0.001)
age.m *Hum_Age.m*, *m* in *Hum_Age*

$fn = 7$: *assert* *conc_of_person*
name *Mary* 1|0
siblings 3 0.5|0.5
age (*young* 0.2|0.4 *medium* 0.7|0.2 *old* 0.2|0.4)
age.m *Hum_Age.m*, *m* in *Hum_Age*

Hardware tuning of *PC* - an example of *Synlogy*

- Introduction
- Data types
- Templates
- Fuzzy rules
- A computation
- Summary

INTRODUCTION

- Design a *FES* for tuning the hardware configuration of a *PC*.
- Evaluative criteria:
  - Power of a configured *PC* - subjective (*low*, *medium*, *high*),
  - Temperature of the *CPU* - objective $(0 - 100^{\circ}\mathrm{C})$ (measured e.g. by *BIOS*).
- Configuration factors:
  - *RAM* - the size of the operational memory $(0 - 32\mathrm{GB})$,
  - *GAM* - the size of the memory of the graphic card $(0 - 4\mathrm{GB})$,
  - *FCPU* - the frequency of the *CPU* $(0 - 4\mathrm{GHz})$.

FUZZIFICATION

- Evaluative criteria:
  - *PC_Power* - a discrete fuzzy set with members: *low*, *medium*, *high*.
  - *CPU_Temperature* - a numerical discrete fuzzy set with members: *low*, *medium*, *high*.
- Configuration factors:
  - *Memory_Size* - a numerical discrete fuzzy set with members: *small*, *medium*, *large*.
  - *Graphic_Size* - a numerical discrete fuzzy set with members: *small*, *medium*, *large*.
  - *CPU_Frequency* - a numerical discrete fuzzy set with members: *low*, *medium*, *high*.

TEMPLATE FACTS

- *PC_configuration_crisp*

variant	$(X\ Y)$	
RAM	2	1|0
GAM	0.256	1|0
FCPU	1.2	1|0

- *PC_configuration*

variant	$(X\ Y)$
RAM	(*small* 0.875|0.125 *medium* 0.125|0.875 *large* 0|1)
GAM	(*small* 0.872|0.128 *medium* 0.128|0.872 *large* 0|1)
FCPU	(*low* 0.4|0.6 *medium* 0.6|0.4 *high* 0|1)

- *PC_test*

variant	$(X\ Y)$
power	(*low* 0.9|0 *medium* 0.2|0.8 *high* 0|1)
TCPU	25 1|0

HEURISTIC FUZZY RULES

r_1: *IF* *power is low t_and RAM is small* *THEN*$^{0.9|0.1}_{0.3|0.3}$ *RAM is large*

r_2: *IF* *GAM is small t_and possible RAM is medium*
THEN$^{0.7|0.2}_{0.5|0.5}$ *GAM is large*

r_3: *IF* *TCPU is low t_and not RAM is small*
THEN$^{0.6|0.3}_{0.5|0.5}$ *FCPU is high*, *aggregate FCPU*

r_4: *IF* *possible not TCPU is high*
THEN$^{0.75|0.25}_{0.3|0.3}$ *FCPU is medium*, *aggregate FCPU*

r_5: *IF* *PC_configuration_crisp*
THEN$^{1|0}_{1|0}$ *fuzzify PC_configuration_crisp*

r_6: *IF* *PC_configuration* *THEN*$^{1|0}_{1|0}$ *defuzzify PC_configuration*

defdisfstype *PC_Power* *members* : *low medium high*

defmemftype *M_Size* *shape* : *polygon* *domain* : 0 32 (GB)

defnumfstype *Memory_Size* *members* : *small medium large*
memftype : *M_Size*

Memory_Size.*small* := *create_polygon_memf* (0 1 16 0 32 0)
Memory_Size.*medium* := *create_polygon_memf* (0 0 16 1 32 0)
Memory_Size.*large* := *create_polygon_memf* (0 0 16 0 32 1)

defmemftype *G_Size* *shape* : *polygon* *domain* : 0 4 (GB)

defnumfstype *Graphic_Size* *members* : *small medium large*
memftype : *G_Size*

Graphic_Size.*small* := *create_polygon_memf* (0 1 2 0 4 0)
Graphic_Size.*medium* := *create_polygon_memf* (0 0 2 1 4 0)
Graphic_Size.*large* := *create_polygon_memf* (0 0 2 0 4 1)

defmemftype *C_Temperature* *shape* : *polygon* *domain* : 0 100 (°C)

defnumfstype *CPU_Temperature* *members* : *low* *medium* *high*
memftype : *C_Temperature*

CPU_Temperature.*low* := *create_polygon_memf* (0 1 50 0 100 0)
CPU_Temperature.*medium* := *create_polygon_memf* (0 0 50 1 100 0)
CPU_Temperature.*high* := *create_polygon_memf* (0 0 50 0 100 1)

defmemftype *C_Frequency* *shape* : *polygon* *domain* : 0 4 (GHz)

defnumfstype *CPU_Frequency* *members* : *low* *medium* *high*
memftype : *C_Frequency*

CPU_Frequency.*low* := *create_polygon_memf* (0 1 2 0 4 0)
CPU_Frequency.*medium* := *create_polygon_memf* (0 0 2 1 4 0)
CPU_Frequency.*high* := *create_polygon_memf* (0 0 2 0 4 1)

```
deftemplate  PC_test
  multislot  variant  char
       slot  power    PC_Power
       slot  TCPU     float

deftemplate  PC_configuration
  multislot  variant  char
       slot  RAM      Memory_Size
       slot  GAM      Graphic_Size
       slot  FCPU     CPU_Frequency

deftemplate  PC_configuration_crisp
  multislot  variant  char
       slot  RAM      float
       slot  GAM      float
       slot  FCPU     float
```

block b_1 *priority* 0

declare $v *char* p *PC_Power* t *float* r *Memory_Size*
g *Graphic_Size* f *CPU_Frequency*

r_1 *IF* *PC_test* *PC_configuration* : *p is low t_and r is small*

variant	$v	*variant*	$v
power	p	*RAM*	r
TCPU	t	*GAM*	g
		FCPU	f

$THEN^{0.9|0.1}_{0.3|0.3}$ r *is large*, $r.small.cf := \neg r.large.cf$,

$r.medium.cf := \frac{for(r.small.cf)+for(r.large.cf)}{2} \Big| 0$:

assert *PC_configuration*

variant	$v A
RAM	r
GAM	g
FCPU	f

block b_1 *priority* 0

r_2 *IF* *PC_test* *PC_configuration* : *g is small t_and*

variant	$v	*variant*	$v	*pos* *r is medium*
power	*p*	*RAM*	*r*	
TCPU	*t*	*GAM*	*g*	
		FCPU	*f*	

$$THEN^{0.7|0.2}_{0.5|0.5}\ g\ is\ large,\ g.small.cf := \neg g.large.cf,$$

$$g.medium.cf := \frac{for(g.small.cf)+for(g.large.cf)}{2}\Big|0 :$$

assert *PC_configuration*

variant	$v B
RAM	*r*
GAM	*g*
FCPU	*f*

block b_1 *priority* 0

r_3 *IF* *PC_test* *PC_configuration* : *t is CPU_Temperature.low*

variant	*$v*	*variant*	*$v*	*t_and not r is small*
power	*p*	*RAM*	*r*	
TCPU	*t*	*GAM*	*g*	
		FCPU	*f*	

$THEN^{0.6|0.3}_{0.5|0.5}$ *f is high*, *f.low.cf* := 0|1, *f.medium.cf* := 0|1 :

agg f : *assert* *PC_configuration*

variant	*$v C*
RAM	*r*
GAM	*g*
FCPU	*f*

block b_1 *priority* 0

r_4 *IF* *PC_test* *PC_configuration* : *pos not* *t is* *CPU_Temperature.high*

variant	$v	*variant*	$v
power	*p*	*RAM*	*r*
TCPU	*t*	*GAM*	*g*
		FCPU	*f*

$$THEN^{0.75|0.25}_{0.3|0.3}\ f\ is\ medium,$$

$$f.high.cf := \frac{for(\nu(antecedent))+for(f.high.cf)}{2}\Bigg| \frac{against(\nu(antecedent))+against(f.high.cf)}{2},$$

$$f.low.cf := \neg f.high.cf\ :$$

agg *f* : *assert* *PC_configuration*

variant	$v *C*
RAM	*r*
GAM	*g*
FCPU	*f*

block b_2 *priority* -1

declare *cr* *cg* *cf* *float* *$v* *char* *r* *Memory_Size*
g *Graphic_Size* *f* *CPU_Frequency*

r_5 *IF* *PC_configuration_crisp* : $1|0$ *THEN*$^{1|0}_{1|0}$ *assert* *PC_configuration*

variant	()	*variant*	()
RAM	*cr*	*RAM*	*fuzzify*(*cr*)
GAM	*cg*	*GAM*	*fuzzify*(*cg*)
FCPU	*cf*	*FCPU*	*fuzzify*(*cf*)

r_6 *IF* *PC_configuration* : $\$v \neq ()$ *THEN*$^{1|0}_{1|0}$ *assert*

variant	*$v*	*PC_configuration_crisp*	
RAM	*r*	*variant*	*$v*
GAM	*g*	*RAM*	*defuzzify*(*r*, *MOM*)
FCPU	*f*	*GAM*	*defuzzify*(*g*, *MOM*)
		FCPU	*defuzzify*(*f*, *MOM*)

For $* = \min, \oplus = \max, \Diamond = \Diamond_m, \Box = \Box_m, \gamma = \delta = 1$:

0 : *PC_configuration_crisp* : *user*

variant	()	
RAM	2	1|0
GAM	0.256	1|0
FCPU	1.2	1|0

Matching r_5, 0:

1 : *PC_configuration* : r_5, 0

variant	()
RAM	(*small* 0.875|0.125 *medium* 0.125|0.875 *large* 0|1)
GAM	(*small* 0.872|0.128 *medium* 0.128|0.872 *large* 0|1)
FCPU	(*low* 0.4|0.6 *medium* 0.6|0.4 *high* 0|1)

2 : *PC_test* : *user*

variant	()
power	(*low* 0.9|0 *medium* 0.2|0.8 *high* 0|1)
TCPU	25 1|0

Matching $r_1, 2, 1$:

$$\nu(antecedent(r_1)) = p.low.cf \lozenge_m r.small.cf = 0.9|0 \lozenge_m 0.875|0.125 = 0.875|0.125,$$

$$fs_{r_1} = \nu(antecedent(r_1)) \lozenge_m cf_{r_1} = 0.875|0.125 \lozenge_m 0.9|0.1 = 0.875|0.125 \geq_t 0.3|0.3,$$

$$r.large.cf = 0.875|0.125,\ r.small.cf = \neg 0.875|0.125 = 0.125|0.875,$$
$$r.medium.cf = \frac{0.125+0.875}{2}\Big|0 = 0.5|0;$$

$3:\ PC\_configuration \qquad : r_1, 2, 1$

variant	(A)
RAM	$(small\ 0.125\|0.875\ medium\ 0.5\|0\ large\ 0.875\|0.125)$
GAM	$(small\ 0.872\|0.128\ medium\ 0.128\|0.872\ large\ 0\|1)$
FCPU	$(low\ 0.4\|0.6\ medium\ 0.6\|0.4\ high\ 0\|1)$

Matching $r_2, 2, 1$:

$$\nu(antecedent(r_2)) = g.small.cf \lozenge_m \leftrightharpoons r.medium.cf = 0.872|0.128 \lozenge_m \leftrightharpoons 0.125|0.875 = 0.125|0.875,$$

$$fs_{r_2} = \nu(antecedent(r_2)) \lozenge_m cf_{r_2} = 0.125|0.875 \lozenge_m 0.7|0.2 = 0.125|0.875 \not\geq_t 0.5|0.5.$$

Matching $r_3, 2, 1$:

$$\nu(antecedent(r_3)) = \nu(t_{=25}\ is\ CPU\_Temperature.low)\ \Diamond_m\ \neg r.small.cf$$
$$= 0.5|0.5\ \Diamond_m\ \neg 0.875|0.125 = 0.125|0.875,$$

$$fs_{r_3} = \nu(antecedent(r_3))\ \Diamond_m\ cf_{r_3} = 0.125|0.875\ \Diamond_m\ 0.6|0.3$$
$$= 0.125|0.875 \not\geq_t 0.5|0.5;$$

Matching $r_4, 2, 1$:

$$\nu(antecedent(r_4)) = \leftrightharpoons \neg\nu(t_{=25}\ is\ CPU\_Temperature.high) = \leftrightharpoons \neg 0|1 = 1|0,$$

$$fs_{r_4} = \nu(antecedent(r_4))\ \Diamond_m\ cf_{r_4} = 1|0\ \Diamond_m\ 0.75|0.25 = 0.75|0.25 \geq_t 0.3|0.3,$$

$$f.medium.cf = 0.75|0.25,\ f.high.cf = \frac{1+0}{2}\Big|\frac{0+1}{2} = 0.5|0.5,$$
$$f.low.cf = \neg 0.5|0.5 = 0.5|0.5;$$

$4:\ PC\_configuration \qquad : r_4, 2, 1$

variant	(C)
RAM	$(small\ 0.875\|0.125\ medium\ 0.125\|0.875\ large\ 0\|1)$
GAM	$(small\ 0.872\|0.128\ medium\ 0.128\|0.872\ large\ 0\|1)$
FCPU	$(low\ 0.5\|0.5\ medium\ 0.75\|0.25\ high\ 0.5\|0.5)$

Matching $r_6, 3$:
$\nu(antecedent(r_6)) = \nu((A) \neq ()) = 1|0,$

$fs_{r_6} = \nu(antecedent(r_6)) \lozenge_m cf_{r_6} = 1|0 \lozenge_m 1|0 = 1|0 \geq_t 1|0,$

$\$v = (A),$

$r = (small\ 0.125|0.875\ medium\ 0.5|0\ large\ 0.875|0.125),$
$defuzzify(r, MOM) = \frac{16+32}{2} = 24,\ \max_r = 0.875,$

$g = (small\ 0.872|0.128\ medium\ 0.128|0.872\ large\ 0|1),$
$defuzzify(g, MOM) = \frac{0+2}{2} = 1,\ \max_g = 1,$

$f = (low\ 0.4|0.6\ medium\ 0.6|0.4\ high\ 0|1),$
$defuzzify(f, MOM) = \frac{0+2}{2} = 1,\ \max_f = 1;$

5 : *PC_configuration_crisp* : $r_6, 3$

variant	(A)	
RAM	24	1|0
GAM	1	1|0
FCPU	1	1|0

Matching $r_6, 4$:
$\nu(antecedent(r_6)) = \nu((A) \neq ()) = 1|0,$

$fs_{r_6} = \nu(antecedent(r_6)) \Diamond_m cf_{r_6} = 1|0 \Diamond_m 1|0 = 1|0 \geq_t 1|0,$

$\$v = (C),$

$r = (small\ 0.875|0.125\ medium\ 0.125|0.875\ large\ 0|1),$
$defuzzify(r, MOM) = \frac{0+16}{2} = 8,\ \max_r = 1,$

$g = (small\ 0.872|0.128\ medium\ 0.128|0.872\ large\ 0|1),$
$defuzzify(g, MOM) = \frac{0+2}{2} = 1,\ \max_g = 1,$

$f = (low\ 0.5|0.5\ medium\ 0.75|0.25\ high\ 0.5|0.5),$
$defuzzify(f, MOM) = \frac{2}{1} = 2,\ \max_f = 0.75;$

$6:\ PC\_configuration\_crisp\ : r_6, 4$

variant	(C)	
RAM	8	$1\|0$
GAM	1	$1\|0$
FCPU	2	$1\|0$

7 : *PC_test* : *user*
variant (*A*)
power (*low* 0.5|0.5 *medium* 0.8|0.2 *high* 0|1)
TCPU 25 1|0

8 : *PC_test* : *user*
variant (*C*)
power (*low* 0.6|0.4 *medium* 0.4|0.6 *high* 0|1)
TCPU 45 1|0

Matching $r_1, 7, 3$:

$$\nu(antecedent(r_1)) = p.low.cf \lozenge_m r.small.cf = 0.5|0.5 \lozenge_m 0.125|0.875 = 0.125|0.875,$$

$$fs_{r_1} = \nu(antecedent(r_1)) \lozenge_m cf_{r_1} = 0.125|0.875 \lozenge_m 0.9|0.1 = 0.125|0.875 \not\geq_t 0.3|0.3;$$

Matching $r_2, 7, 3$:

$$\nu(antecedent(r_2)) = g.small.cf \lozenge_m \leftrightharpoons r.medium.cf = 0.872|0.128 \lozenge_m \leftrightharpoons 0.5|0 = 0.872|0.5,$$

$$fs_{r_2} = \nu(antecedent(r_2)) \lozenge_m cf_{r_2} = 0.872|0.5 \lozenge_m 0.7|0.2 = 0.7|0.5 \geq_t 0.5|0.5,$$

$$g.large.cf = 0.7|0.5,\ g.small.cf = \neg 0.7|0.5 = 0.5|0.7,$$
$$g.medium.cf = \frac{0.5+0.7}{2}\Big|0 = 0.6|0;$$

9 : *PC_configuration* : $r_2, 7, 3$

variant	$(A\ B)$
RAM	(*small* 0.125|0.875 *medium* 0.5|0 *large* 0.875|0.125)
GAM	(*small* 0.5|0.7 *medium* 0.6|0 *large* 0.7|0.5)
FCPU	(*low* 0.4|0.6 *medium* 0.6|0.4 *high* 0|1)

Matching $r_3, 7, 3$:

$$\nu(antecedent(r_3)) = \nu(t_{=25}\ is\ CPU\_Temperature.low)\ \Diamond_m\ \neg r.small.cf$$
$$= 0.5|0.5\ \Diamond_m\ \neg 0.125|0.875 = 0.5|0.5,$$

$$fs_{r_3} = \nu(antecedent(r_3))\ \Diamond_m\ cf_{r_3} = 0.5|0.5\ \Diamond_m\ 0.6|0.3 = 0.5|0.5 \geq_t 0.5|0.5,$$

$$f.high.cf = 0.5|0.5,\ f.low.cf = 0|1,\ f.medium.cf = 0|1;$$

Matching $r_4, 7, 3$:

$$\nu(antecedent(r_4)) = \leftrightharpoons \neg\nu(t_{=25}\ is\ CPU\_Temperature.high) = \leftrightharpoons \neg 0|1 = 1|0,$$

$$fs_{r_4} = \nu(antecedent(r_4))\ \Diamond_m\ cf_{r_4} = 1|0\ \Diamond_m\ 0.75|0.25 = 0.75|0.25 \geq_t 0.3|0.3,$$

$$f.medium.cf = 0.75|0.25,\ f.high.cf = \frac{1+0}{2}\Big|\frac{0+1}{2} = 0.5|0.5,$$
$$f.low.cf = \neg 0.5|0.5 = 0.5|0.5;$$

$$agg\ f:\ f.low.cf = 0|1\ \Box_m\ 0.5|0.5 = 0.5|0.5,\ f.high.cf = 0.5|0.5\ \Box_m\ 0.5|0.5 = 0.5|0.5,$$
$$f.medium.cf = 0|1\ \Box_m\ 0.75|0.25 = 0.75|0.25,$$

$$10:\ PC\_configuration \qquad : r_3, r_4, 7, 3$$
$$\begin{array}{ll} variant & (A\ C) \\ RAM & (small\ 0.125|0.875\ medium\ 0.5|0\ large\ 0.875|0.125) \\ GAM & (small\ 0.872|0.128\ medium\ 0.128|0.872\ large\ 0|1) \\ FCPU & (low\ 0.5|0.5\ medium\ 0.75|0.25\ high\ 0.5|0.5) \end{array}$$

Matching $r_1, 8, 4$:

$$\nu(antecedent(r_1)) = p.low.cf \lozenge_m r.small.cf = 0.6|0.4 \lozenge_m 0.875|0.125 = 0.6|0.4,$$

$$fs_{r_1} = \nu(antecedent(r_1)) \lozenge_m cf_{r_1} = 0.6|0.4 \lozenge_m 0.9|0.1 = 0.6|0.4 \not\geq_t 0.3|0.3;$$

Matching $r_2, 8, 4$:

$$\nu(antecedent(r_2)) = g.small.cf \lozenge_m \leftrightharpoons r.medium.cf = 0.872|0.128 \lozenge_m \leftrightharpoons 0.125|0.875 = 0.125|0.875,$$

$$fs_{r_2} = \nu(antecedent(r_2)) \lozenge_m cf_{r_2} = 0.125|0.875 \lozenge_m 0.7|0.2 = 0.125|0.875 \not\geq_t 0.5|0.5.$$

Matching $r_3, 8, 4$:

$$\nu(antecedent(r_3)) = \nu(t_{=45}\ is\ CPU\_Temperature.low)\ \Diamond_m\ \neg r.small.cf$$
$$= 0.1|0.9\ \Diamond_m\ \neg 0.875|0.125 = 0.1|0.9,$$

$$fs_{r_3} = \nu(antecedent(r_3))\ \Diamond_m\ cf_{r_3} = 0.1|0.9\ \Diamond_m\ 0.6|0.3 = 0.1|0.9 \not\geq_t 0.5|0.5;$$

Matching $r_4, 8, 4$:

$$\nu(antecedent(r_4)) = \leftrightharpoons \neg\nu(t_{=45}\ is\ CPU\_Temperature.high) = \leftrightharpoons \neg 0|1 = 1|0,$$

$$fs_{r_4} = \nu(antecedent(r_4))\ \Diamond_m\ cf_{r_4} = 1|0\ \Diamond_m\ 0.75|0.25 = 0.75|0.25 \geq_t 0.3|0.3,$$

$$f.medium.cf = 0.75|0.25,\ f.high.cf = \frac{1+0.5}{2}\Big|\frac{0+0.5}{2} = 0.75|0.25,$$
$$f.low.cf = \neg 0.75|0.25 = 0.25|0.75;$$

$11:\ PC\_configuration \qquad : r_4, 8, 4$

variant	$(C\ C)$
RAM	$(small\ 0.875\|0.125\ medium\ 0.125\|0.875\ large\ 0\|1)$
GAM	$(small\ 0.872\|0.128\ medium\ 0.128\|0.872\ large\ 0\|1)$
FCPU	$(low\ 0.25\|0.75\ medium\ 0.75\|0.25\ high\ 0.75\|0.25)$

Matching $r_6, 9$:
$\nu(antecedent(r_6)) = \nu((A) \neq ()) = 1|0,$

$fs_{r_6} = \nu(antecedent(r_6)) \Diamond_{\mathrm{m}} cf_{r_6} = 1|0 \Diamond_{\mathrm{m}} 1|0 = 1|0 \geq_{\mathrm{t}} 1|0,$

$\$v = (A\,B),$

$r = (small\ 0.125|0.875\ medium\ 0.5|0\ large\ 0.875|0.125),$
$defuzzify(r, MOM) = \frac{16+32}{2} = 24,\ \max_r = 0.875,$

$g = (small\ 0.5|0.7\ medium\ 0.6|0\ large\ 0.7|0.5),$
$defuzzify(g, MOM) = \frac{2+4}{2} = 3,\ \max_g = 0.7,$

$f = (low\ 0.4|0.6\ medium\ 0.6|0.4\ high\ 0|1),$
$defuzzify(f, MOM) = \frac{0+2}{2} = 1,\ \max_f = 1;$

12 : *PC_configuration_crisp* : $r_6, 9$

variant	$(A\,B)$	
RAM	24	1|0
GAM	3	1|0
FCPU	1	1|0

Matching r_6, 10:
$\nu(antecedent(r_6)) = \nu((A) \neq ()) = 1|0$,

$fs_{r_6} = \nu(antecedent(r_6)) \Diamond_m cf_{r_6} = 1|0 \Diamond_m 1|0 = 1|0 \geq_t 1|0$,

$\$v = (A\ C)$,

$r = (small\ 0.125|0.875\ medium\ 0.5|0\ large\ 0.875|0.125)$,
$defuzzify(r, MOM) = \frac{16+32}{2} = 24$, $\max_r = 0.875$,

$g = (small\ 0.872|0.128\ medium\ 0.128|0.872\ large\ 0|1)$,
$defuzzify(g, MOM) = \frac{0+2}{2} = 1$, $\max_g = 1$,

$f = (low\ 0.5|0.5\ medium\ 0.75|0.25\ high\ 0.5|0.5)$,
$defuzzify(f, MOM) = \frac{2}{1} = 2$, $\max_f = 0.75$;

13 : *PC_configuration_crisp* : r_6, 10

variant	$(A\ C)$	
RAM	24	1|0
GAM	1	1|0
FCPU	2	1|0

Matching $r_6, 11$:
$\nu(antecedent(r_6)) = \nu((A) \neq ()) = 1|0,$

$fs_{r_6} = \nu(antecedent(r_6)) \Diamond_m cf_{r_6} = 1|0 \Diamond_m 1|0 = 1|0 \geq_t 1|0,$

$\$v = (C\ C),$

$r = (small\ 0.875|0.125\ medium\ 0.125|0.875\ large\ 0|1),$
$defuzzify(r, MOM) = \frac{0+16}{2} = 8,\ \max_r = 1,$

$g = (small\ 0.872|0.128\ medium\ 0.128|0.872\ large\ 0|1),$
$defuzzify(g, MOM) = \frac{0+2}{2} = 1,\ \max_g = 1,$

$f = (low\ 0.25|0.75\ medium\ 0.75|0.25\ high\ 0.75|0.25),$
$defuzzify(f, MOM) = \frac{2+4}{2} = 3,\ \max_f = 0.75;$

$14:\ PC\_configuration\_crisp\ : r_6, 11$

variant	$(C\ C)$	
RAM	8	$1\|0$
GAM	1	$1\|0$
FCPU	3	$1\|0$

Hardware tuning of *PC*: a computation - the summary

0 : *PC_configuration_crisp*

variant	()	
RAM	2	1|0
GAM	0.256	1|0
FCPU	1.2	1|0

2 : *PC_test*

variant	()
power	(*low* 0.9|0 *medium* 0.2|0.8 *high* 0|1)
TCPU	25 1|0

5 : *PC_configuration_crisp*

variant	(*A*)	
RAM	24	1|0
GAM	1	1|0
FCPU	1	1|0

7 : *PC_test*

variant	(*A*)
power	(*low* 0.5|0.5 *medium* 0.8|0.2 *high* 0|1)
TCPU	25 1|0

12 : *PC_configuration_crisp*

variant	(*A B*)	
RAM	24	1|0
GAM	3	1|0
FCPU	1	1|0

13 : *PC_configuration_crisp*

variant	(*A C*)	
RAM	24	1|0
GAM	1	1|0
FCPU	2	1|0

6 : *PC_configuration_crisp*
variant (*C*)
RAM 8 1|0
GAM 1 1|0
FCPU 2 1|0

14 : *PC_configuration_crisp*
variant (*C C*)
RAM 8 1|0
GAM 1 1|0
FCPU 3 1|0

8 : *PC_test*
variant (*C*)
power (*low* 0.6|0.4 *medium* 0.4|0.6 *high* 0|1)
TCPU 45 1|0

Conclusions

- *FES* - *Synlogy* - the next generation of *ES*:
  - the data-driven programming paradigm,
  - multi-step fuzzy inference,
  - parallel firing rules (in blocks),
  - combining uncertainty, vagueness, impreciseness,
  - mathematical foundations.
- Future research:
  - the development of formal semantics for *FES*,
  - the paraconsistent (e.g. bilattice) approach,
  - dealing with inconsistency in a reasonable manner,
  - an efficient implementation of large-scale pattern matching - the Fuzzy rete algorithm.

Bibliography I

Anderson, J. R.
Rules of the Mind.
Lawrence Erlbaum. 1993.

Cox, E.
The Fuzzy Systems Handbook: A Practitioner's Guide to Building, Using and Maintaining Fuzzy Systems.
Morgan Kaufmann. 1999.

Feigenbaum, E. A. and Buchanan, B. G.
DENDRAL and Meta-DENDRAL: roots of knowledge systems and expert systems applications.
Artificial Intelligence, 59:233-240. 1993.

Fitting, M. C.
Logic programming on a topological bilattice.
Fundamenta Informaticae, 11:209-218. 1988.

Bibliography II

Forgy, Ch.
A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem.
Artificial Intelligence, 19:17-37. 1982.

Ginsberg, M.
Multivalued logics: a uniform approach to reasoning in artificial intelligence.
Computational Intelligence, 4:265-316. 1988.

Haykin, S.
Neural Networks: A Comprehensive Foundation.
Prentice-Hall. 1994.

Jackson, P.
Introduction to Expert Systems.
Addison-Wesley. 1999.

Bibliography III

Klir, G. J. and Yuan, B.
Fuzzy Sets and Fuzzy Logic: Theory and Applications.
Prentice-Hall. 1995.

Larsen, P. M.
Industrial applications of fuzzy logic control.
International Journal of Man-Machine Studies, 12:3-10. 1980.

Mamdani, E. H. and Assilian, S.
An experiment in linguistic synthesis with a fuzzy logic controller.
International Journal of Man-Machine Studies, 7:1-13. 1975.

Mamdani, E. H.
Advances in the linguistic synthesis of fuzzy controller.
International Journal of Man-Machine Studies, 8(6):669-678. 1976.

Bibliography IV

McDermott, J.
R1: an expert in the computing system domain.
Proceedings of the National Conference on Artificial Intelligence, 269-271. 1980.

Minsky, M.
A framework for representing knowledge.
In The Psychology of Computer Vision, Winston, P. H. ed. McGraw-Hill. 1975.

Newell, A. and Simon, H. A.
Human Problem Solving.
Prentice-Hall. 1972.

Quillian, M. R.
Semantic Memory.
In Semantic Information Processing, Minsky, M. ed., 227-270. *MIT Press*. 1968.

Bibliography V

Shortliffe, E. H.
Computer-based medical consultation: MYCIN.
Elsevier Science. 1976.

Takagi, T. and Sugeno, M.
Fuzzy identification of systems and its applications to modeling and control.
IEEE Transactions on Systems, Man and Cybernetics, Vol. 15:116-132. 1985.

Tsukamoto, Y.
An approach to fuzzy reasoning model.
In Advances in fuzzy set theory and applications. North-Holland. 1979.

Vojtáš, P.
Fuzzy logic programming.
Fuzzy sets and systems, 124(3):361-370, 2001.

Bibliography VI

Zadeh, L. A.
Fuzzy sets.
Information and Control, 8:338-353. 1965.

Zadeh, L. A.
A fuzzy-set-theoretic interpretation of linguistic hedges.
Journal of Cybernetics, 2:4-34. 1972.

Zadeh, L. A.
Outline of a new approach to the analysis of complex systems and decision processes.
IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-3(1):28-44. 1973.

Zadeh, L. A.
Fuzzy logic and its application to approximate reasoning.
In Information Processing 74, Proceedings of IFIP Congress 1974, (3):591-594. 1974.

Bibliography VII

Zadeh, L. A.
On the analysis of large-scale systems.
In Fuzzy Sets and their Application to Cognitive and Decision Processes.
Academic Press. 1974.

Zadeh, L. A.
The concept of a linguistic variable and its application to approximate reasoning, I-III.
Information Sciences, 8:199-251, 301-357, 9:43-80. 1975, 1976.