

Fuzzy CLIPS

- Fuzzy CLIPS je verzija CLIPS-a (standardne verzije) koja podržava rad sa podacima koji su nepouzđani i neodređeni.
- Veoma je važno pravilno razgraničiti značenje pojmova **nepouzđanost** i **neodređenost**
- **Nepouzđanost** se odnosi na slučaj kada su vrednosti podataka – činjenica (facts) koji se koriste u zaključivanju poznati sa određenim stepenom verovatnoće, tj. kada ne postoji 100% sigurnosti u postojeće vrednosti podataka.
- Mogućnost predstavljanja nepouzđanih podataka i zaključivanja sa takvim podacima je jedan od osnovnih zahteva u radu sa ekspertnim sistemima

Fuzzy CLIPS

- Jedan od prvih ES, pomenuti MYCIN je posedovao mogućnost rada sa nepouzđanim podacima
- Standardni CLIPS nema ugrađene mogućnosti za rad sa nepouzđanim podacima što ne znači da se naknadno ne može isprogramirati sve ono što je potrebno.
- Ideja sa Fuzzy CLIPS-om je da se u same osnove ES ugradi takva standardna mogućnost i da se time bitno olakša rad sa nepouzđanim podacima.
- **Neodređenost** je pojam bitno različit od nepouzđanosti, iako u običnom govoru postoji tendencija nerazlikovanja, mešanja i pogrešne upotrebe ovih pojmova

Fuzzy CLIPS

- Pre svega, engleski termin za neodređenost je **fuzzy**, dok se pouzdanost kaže **certainty** ili **condifence**
- Odatle i Certainty Factors (CF) ili ConFidence u I2+ gde je predviđena mogućnost rada sa CF.
- Neodređenost se odnosi na slučaj lingvističkih – govornih iskaza u običnom govornom jeziku na koje smo već navikli i koji nam ne predstavljaju veći problem za razumevanje i komunikaciju.
- Međutim, ako je potrebno predstaviti takve pojmove egzaktno – matematički ili u sftware-u, onda treba koristiti tzv. Fuzzy logiku.

Fuzzy - neodređenost

- Slede primeri nekoliko rečenica iz običnog života u kojima je prisutna neodređenost.
- Vreme je lepo.
- Drvo je visoko.
- Veliki grad.
- Mlad čovek.
- Auto je brz.
- Nebeska tela su daleko od nas.
- Iz ovih i sličnih rečenica se ne može ništa egzaktno zaključiti, već postoji samo naša maglovita – neodređena predstava o tome

Fuzzy logika

- Da bi se slučajevi neodređenosti mogli egzaktno tretirati, formulisana je tzv. Fuzzy logika.
- Tvorac – utemeljitelj fuzzy logike je Lotfi A. Zadeh



- 1921, Baku Azerbaijan, computer science professor at the University of California, Berkley, USA

Fuzzy logika

- Klasična logika ima svega dve istinitosne vrednosti – tačno / netačno, true / false, 1 / 0,...
- Primenjeno na prethodne neodređene rečenice, to bi nametnulo da neodređenom iskazu **drvo je visoko** možemo pridružiti jednu od dve logičke istinitosne vrednosti, tj. to je ili tačno ili netačno.
- Iz običnog života znamo da to nije tako jednostavno, tj. to nije u duhu govornog jezika.
- Zato je formulisana fuzzy logika koja ima više od dve istinitosne vrednosti.
- Istinitosnih vrednosti može biti proizvoljno mnogo ili beskonačno mnogo, zavisno od reprezentacije

Fuzzy logika

- Ako se posmatra primer starosti ili mladosti kod ljudi, onda se sasvim sigurno starost od jedne godine može smatrati apsolutnom mladošću, dok se starost od 100 godina može smatrati apsolutnom starošću.
- Samo se postavlja pitanje kako tretirati ostalih 98% slučajeva koji su i najčešći, dok su ovo samo ekstremi.
- Na primer, da li je neko mlad ili star sa 25, 35, 45, 55, 65, 75 godina?
- Rešenje je nađeno u razvoju koncepta delimične pripadnosti – grade of membership jednoj ili drugoj istinitosnoj vrednosti.

Fuzzy logika

- Stepem pripadnosti se u fuzzy logici opisuje sa funkcijom pripadnosti - membership function koja je definisana u nekoj oblasti – U - Universe of Discourse.
 - Fuzzy skup A u nekoj oblasti U se u teoriji fuzzy skupova karakteriše sa funkcijom pripadnosti μ_A
- $$\mu_A : U \rightarrow [0,1]$$
- Funkcija pripadnosti $\mu_A(x)$ određuje stepen pripadnosti $[0, 1]$ svakog elementa x oblasti U fuzzy skupu A
 - Neodređeni – fuzzy pojam se može definisati preko fuzzy skupa predstavljenog tabelarno

Tabela fuzzy skupa za pojam mlad - young

Table 1: Fuzzy Term *young*

Age	Grade of Membership
25	1.0
30	0.8
35	0.6
40	0.4
45	0.2
50	0.0

$$\mu_{\text{young}}(25) = 1, \mu_{\text{young}}(30) = 0.8, \dots, \mu_{\text{young}}(50) = 0$$

Grafički prikaz funkcije pripadnosti

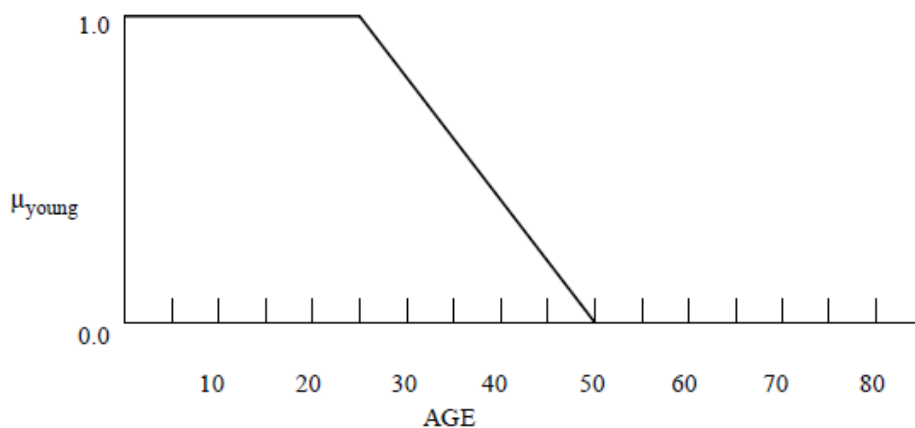


Figure 1: Possibility distribution of *young*

Fuzzy modifikatori

- Modifikovani lingvistički izraz fuzzy pojma mlad kao na pr. **mlad u određenoj meri** ili **veoma mlad** se može predstaviti preko odgovarajućih matematičkih operacija sa fuzzy skupom pri čemu su modifikatori - hedges **u određenoj meri** i **veoma** predstavljeni nekim matematičkim operacijama.
- Modifikator **u određenoj meri** se može predstaviti na pr. kvadratnim korenom funkcije pripadnosti za osnovni fuzzy skup.

Modifikovana fuzzy funkcija pripadnosti

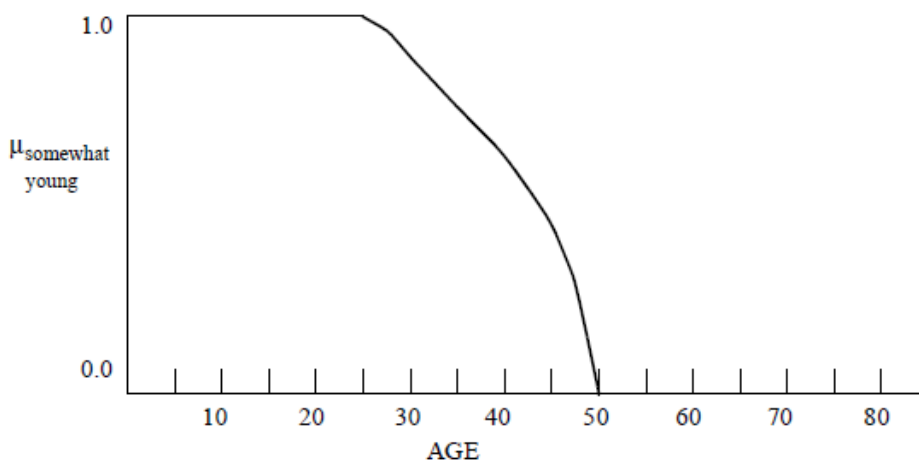


Figure 2: Possibility distribution of *somewhat young*

Fuzzy facts

- (deftemplate age ;definition of fuzzy variable 'age'
- 0 120 years
- ((young (25 1) (50 0)) (old (50 0) (65 1))))
- U konstrukciji deftemplate poznatoj od ranije, naziv **age** predstavlja (fuzzy) lingvističku promenljivu, dok slotovi **young** i **old** predstavljaju (fuzzy) članove – terms
- **0 -120** je oblast dok **years** predstavlja jedinicu
- Svaka lingvistička - fuzzy promenljiva ima pridružen skup fuzzy članova - terms koji se nazivaju primarni članovi i koji predstavljaju vrednosti koje može da ima lingvistička promenljiva

Fuzzy facts i pravila

- (def facts fuzzy-fact (age young) ; a fuzzy fact)
- (defrule one ; a rule that matches and asserts fuzzy facts
- (Speed_error big)
- =>
- (assert (Throttle_change))
-)
- **age**, **Speed_error** i **Throttle_change** su lingvističke promenljive, dok su **young**, **big** i **small** lingvistički članovi, tj. vrednosti koje uzimaju lingvističke promenljive

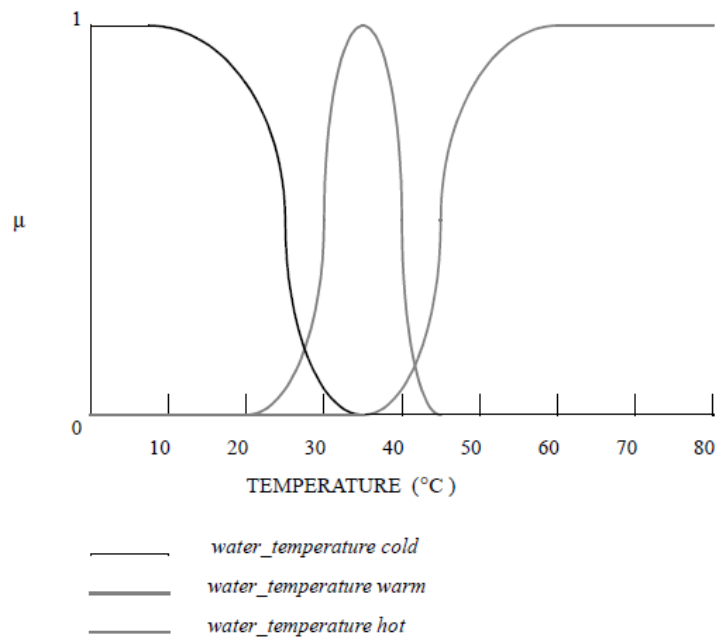


Figure 3: Primary terms of a linguistic variable

Nepouzdanost - uncertainty

- Kod predstavljanja nepouzdanosti činjenica u Fuzzy CLIPS-u se kao i u I2+ koriste CF – Certainty Factors
- (fact) [CF certainty factor]
- Vrednost CF se kreće u granicama od 0 do 1
- (prediction sunny) CF 0.8
- Ako se CF izostavi, onda se pretpostavlja da je CF 1
- (prediction sunny)
- CF takođe može da se koristi i kod pravila.
- **(defrule flight-rule (declare (CF 0.95)) ...)**
- Ako se CF kod pravila izostavi, pretpostavlja se da je 1

CF za pravilo

- (defrule flight-rule
- (declare (CF 0.95)) ;declares certainty factor of the rule
- (animal type bird)
- =>
- (assert (animal can fly))
-)
- Smisao CF za pravilo je definisanje pouzdanosti za pravilo, tj. da je pravilo tačno u 95% slučajeva, ali ne uvek

Kombinovanje Fuzzy i CF

- (deffacts FuzzyAndUncertainFact
- (Speed_error more_or_less zero) CF 0.9)
- (defrule Uncertain_rule
- (declare (CF 0.8))
- (Johns_age young)
- =>
- (assert (John goes to school)))
- **Speed_error** i **Johns_age** su fuzzy promenljive, **zero** i **young** su članovi, dok je **more_or_less** modifikator

Fuzzy CLIPS zaključivanje

- Zaključivanje je u fuzzy CLIPS-u složenije nego kod standardnog CLIPS-a
- Ako se ne koriste fuzzy činjenice i CF, zaključivanje je potpuno isto kao i u običnom CLIPS-u
- Kada se koriste fuzzy facts i CF mogu da postoje razni slučajevi zaključivanja u zavisnosti od toga gde se sve nalaze fuzzy promenljive – sa leve strane samo, sa desne ili sa obe strane pravila, zatim od broja fuzzy promenljivih i raznih drugih detalja.
- Postoje tri osnovna tipa pravila prema tome gde se nalaze fuzzy promenljive

Vrste fuzzy pravila

- Tri osnovne vrste fuzzy pravila su:
- CRISP – nema fuzzy promenljivih na levoj strani pravila dok ih na desnoj može biti ali i ne mora
- FUZZY_CRISP – postoje fuzzy promenljive na levoj strani pravila ali ne i na desnoj strani
- FUZZY_FUZZY – postoje fuzzy promenljive i na levoj i na desnoj strani pravila
- CRISP označava činjenice koje nisu fuzzy, tj. nešto što je jasno definisano i što nije fuzzy
- Svaka od navedene tri vrste pravila će se detaljno razmotriti

Prosta pravila tipa – if A then C

- if A then C CFr
- A' CFf
- -----
- C' CFc
- A je uslov pravila
- A' je matching fact u fact database
- C je zaključak pravila
- C' je izračunati zaključak
- CFr je CF pravila
- CFf je CF činjenice
- CFc je CF zaključka

Prosto pravilo tipa CRISP

- Kod pravila tipa CRISP uslov pravila A mora doslovce da bude jednak činjenici A' da bi se pravilo aktiviralo i da bi se činjenica C – zaključak postavio, $C = C'$
- $CFc = CFr * CFf$
- **(defrule crisp-simple-rule**
- **(declare (CF 0.7)) ;crisp rule certainty factor of 0.7**
- **(light_switch off) ;crisp antecedent**
- **=>**
- **(assert (illumination_level dark)); fuzzy consequent) ;**
end of rule definition
- Postavljena činjenica: (light_switch off) CF 0.8
- Zaključak: (illumination_level dark) CF 0.56
- $CFc = 0.7 * 0.8$

Prosto pravilo tipa FUZZY CRISP

- Kod pravila tipa FUZZY CRISP, A' mora biti fuzzy fact sa istom fuzzy promenljivom kao i kod A da bi došlo do pattern matching-a i aktiviranja pravila
- Vrednosti fuzzy promenljivih u A i $A' - F\alpha$ i $F\alpha'$ ne moraju biti jednake ali se moraju preklapati da bi došlo do aktivacije pravila – zadovoljenja uslova
- Fuzzy facts (*temperature high*) i (*pressure high*) imaju različite promenljive - *temperature* i *pressure* i samim tim ne može doći do aktiviranja pravila
- Fuzzy facts (*pressure low*), (*pressure medium*), i (*pressure high*) imaju iste promenljive - *pressure* a neke vrednosti se preklapaju

Preklapanje – matching of fuzzy facts

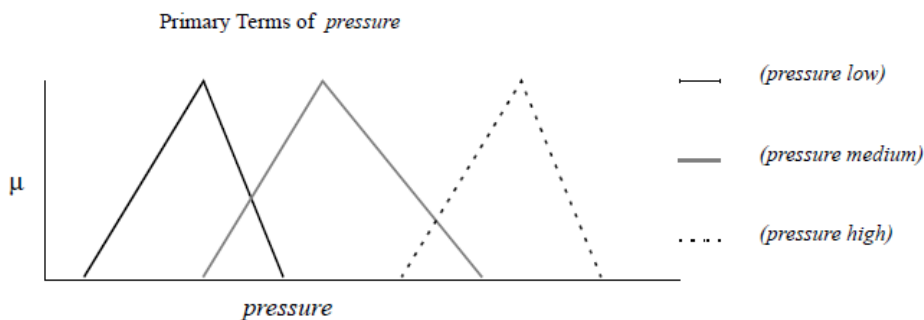


Figure 4: Matching of fuzzy facts

- Zaključak C je jednak sa C' i $CF_c = CF_r * CF_f * S$
- S je mera sličnosti - similarity između $F\alpha$ (fuzzy pattern A) i $F\alpha'$ (fuzzy fact A')

S – similarity - sličnost

- Sličnost se izračunava na osnovu mere mogućnosti – possibility – P i neophodnosti – necessity – N
- $S = P(F\alpha | F\alpha')$ ako je $N(F\alpha | F\alpha') > 0.5$
- $S = (N(F\alpha | F\alpha') + 0.5) * P(F\alpha | F\alpha')$ u svim ostalim slučajevima
- $P(F\alpha | F\alpha') = \max(\min(\mu_{F\alpha}(u), \mu_{F\alpha'}(u))) \forall (u \in U)$
- $N(F\alpha | F\alpha') = 1 - P(\sim F\alpha | F\alpha')$
- $\sim F\alpha$ je komplement of $F\alpha$ definisan sa sledećom funkcijom pripadnosti:
- $\mu_{\sim F\alpha}(u) = 1 - \mu_{F\alpha}(u), \forall (u \in U)$

Primer FUZZY CRISP pravila

- **(defrule simple-fuzzy-crisp-rule**
- **(declare (CF 0.7)) ;rule has a certainty factor of 0.7**
- **(fuzzy-fact fact2) ;fuzzy antecedent**
- **=>**
- **(assert (crisp-fact fact3));crisp consequent)**
- **(fuzzy-fact fact1) CF 0.8** se nalazi u bazi podataka i zadovoljava levu stranu pravila pošto su fuzzy promenljive fuzzy-fact iste i ako se fuzzy skupovi fact1 i fact2 preklapaju

Primer FUZZY CRISP pravila

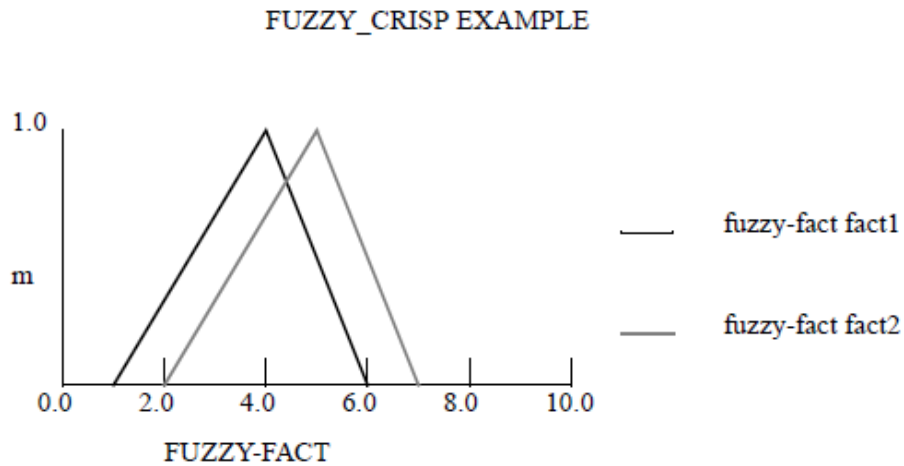


Figure 5: Fact and antecedent fuzzy sets

Izračunavanje neophodnosti

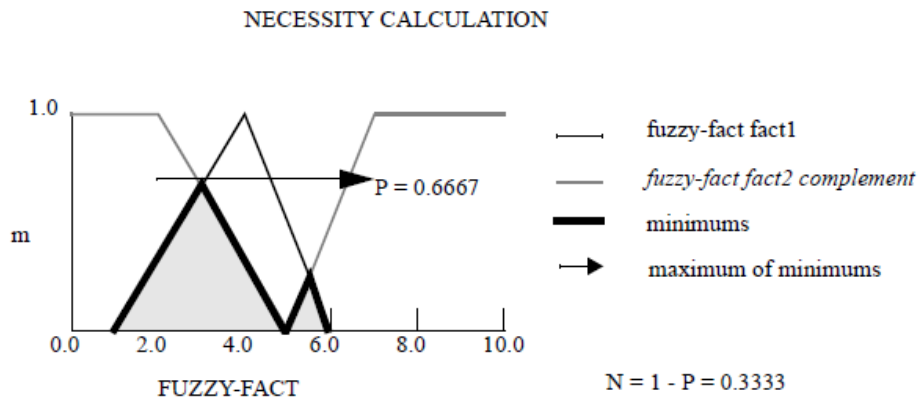
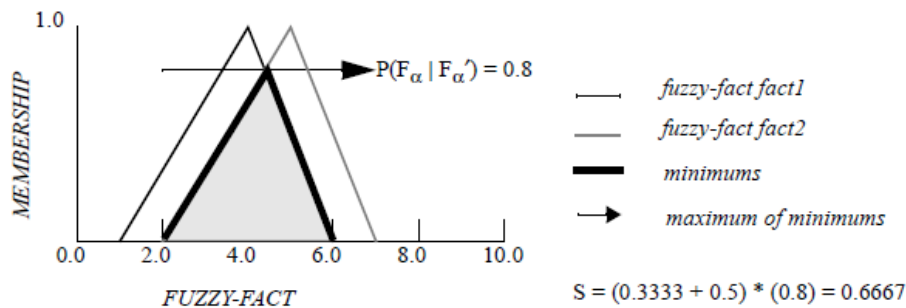


Figure 6: $N(F_\alpha | F'_\alpha)$

Since the necessity is less than 0.5, $S = (N(F_\alpha | F'_\alpha) + 0.5) * P(F_\alpha | F'_\alpha)$ (see Figure 7).

Izračunavanje sličnosti

SIMILARITY CALCULATION



Prosto pravilo FUZZY FUZZY

- Kod FUZZY FUZZY pravila pattern matching je isti kao i kod FUZZY CRISP pravila, dok su uslov i zaključak povezani sa fuzzy relacijom
- $R = F_{\alpha} * F_c$
- F_{α} – fuzzy set u uslovu
- F_c – fuzzy set u zaključku
- Funkcija pripadnosti od R je:
- $\mu R(u, v) = \min(\mu F_{\alpha}(u), \mu F_c(v)), \forall (u, v) \in U \times V$
- Postoje i drugi načini računanja R
- Izračunavanje $F_c' = F_{\alpha}' \circ R$ – fuzzy skup zaključka

Izračunavanje $Fc' = Fa' \circ R$ – fuzzy skup zaključka

- $\mu_{Fc'}(v) = \max_{u \in U} (\min(\mu_{Fa'}(u), \mu_R(u, v)))$
- Što se može pojednostaviti kao:
- $\mu_{Fc'}(v) = \min(z, \mu_{Fc}(v))$
- Gde je z
- $z = \max(\min(\mu_{Fa'}(u), \mu_{Fa}(u)))$
- Faktor pouzdanosti za zaključak se računa kao:
- $CFc = CFr * CFf$

Primer FUZZY FUZZY pravila

- **(defrule fuzzy-fuzzy-rule; both antecedent and consequent are fuzzy objects**
- **(temperature hot)**
- **=>**
- **(assert (temp_change little)))**
- **(temperature warm) ; a fact in the fact database**

Grafički prikaz zaključivanja

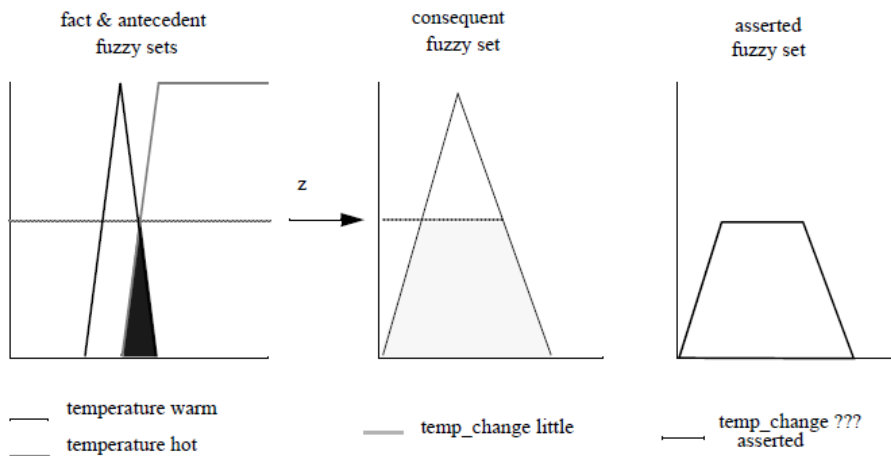


Figure 8: Compositional rule of inference¹ (max-min)

Grafički prikaz zaključivanja 2

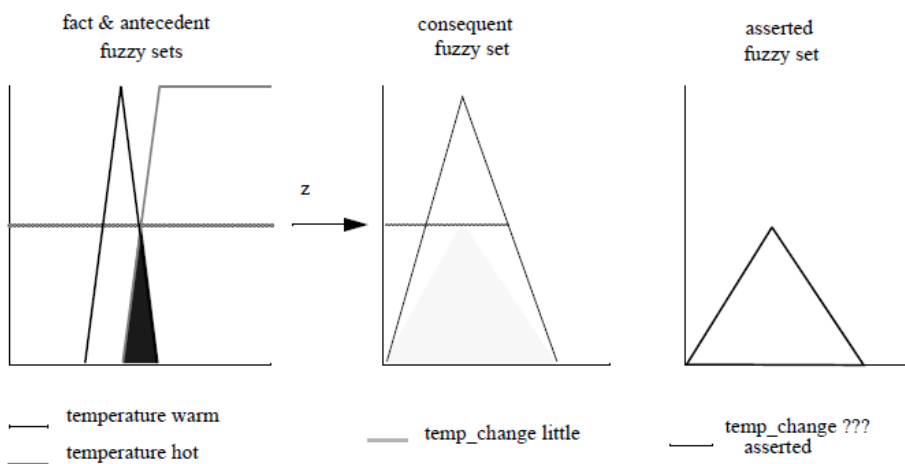


Figure 9: Compositional rule of inference (max-product)

Složena pravila, višestruki zaključci

- Ako se sa desne strane nalazi više činjenica, onda se takvo pravilo može ekvivalentno posmatrati kao više pravila sa istom levom stranom i sa po jednim zaključkom sa desne strane složenog pravila
- Složeno pravilo tipa:
 - if Antecedents then C1 and C2 and ... and Cn
- Je ekvivalentno sa:
 - if Antecedents then C1
 - if Antecedents then C2
 - ...
 - if Antecedents then Cn

Složeno pravilo više uslova

- Opšti oblik pravila:
 - if A1 and A2 then C CFr
- A1' CFf1
- A2' CFf2
- -----
- C' CFc

$$F_c' = F_{c1}' \cap F_{c2}'$$

- if A1 then C – Fc1'
- if A2 then C – Fc2'

Grafički prikaz zaključivanja složenog pravila

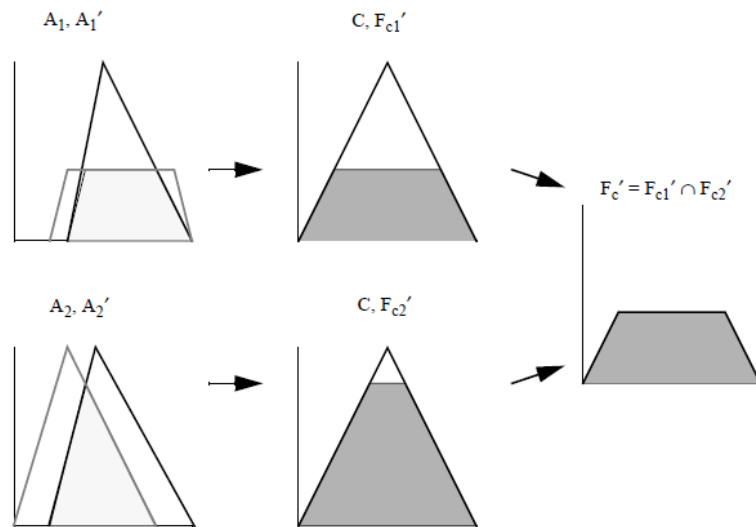


Figure 10: Compositional rule for multiple antecedents

CF složenih pravila

- $CF_c = \min(CF_{f1'}, CF_{f2'}) * CF_r$
- $CF_{f1'}$ je CF prostog pravila if A1 then C kada je činjenica A1'
- $CF_{f2'}$ je CF prostog pravila if A2 then C kada je činjenica A2'
- Postupak je direktno primenljiv na slučaj više od dva antecedenta

$$F_c' = F_{c1'} \cap F_{c2'} \dots \cap F_{cn'}$$

$$CF_c = \min(CF_{f1'}, CF_{f2'}, \dots, CF_{fn'}) * CF_r$$

Globalni doprinos činjenicama

- Kod običnog CLIPS-a, jednom potvrđena činjenica znači da ponovno potvrđivanje nema efekta osim ako nije dozvoljeno dupliranje činjenica.
- Međutim, kod fuzzy činjenica i činjenica sa CF, ponovno potvrđivanje može da da dodatni doprinos u smislu kombinovanja postojeće fuzzy vrednosti sa novom vrednošću, kao i uzimanje većeg od više CF

$$F_g = F_f \cup F_c'$$

- F_g – nova fuzzy vrednost, F_f postojeća fuzzy vrednost i F_c' fuzzy vrednost koja se potvrđuje
- $CF_g = \text{maximum}(CF_f, CF_c')$

Globalni doprinos, grafički prikaz

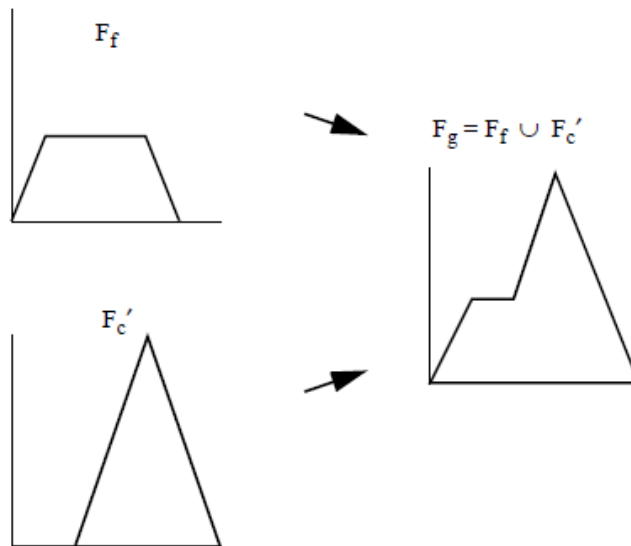


Figure 11: Union of fuzzy sets - global contribution

CF threshold - prag

- $CF_{rule} * \min(CF_1, \dots, CF_n)$
- (defrule below-threshold-rule
- (declare (CF 0.5)) ;rule certainty factor of 0.5
- (fuzzy-fact antecedent-fact) ;fuzzy antecedent
- =>
- (assert (crisp-fact c1)) ;crisp consequent
- (assert (another-fuzzy-fact c2)) ;fuzzy consequent)
- (fuzzy-fact fact-list-fact) CF 0.6
- $CF = 0.5 * 0.6 = 0.3$ za pravilo
- $CF = 0.5 * 0.6 * S$ ($S = 0.8$)
- =>
- (assert (crisp-fact c1)); calculated CF = 0.24
- (assert (another-fuzzy-fact c2)); calculated CF = 0.3

CF threshold - prag

- (defrule complex
- (declare (CF 0.9))
- (crisp1)
- (fuzzy1 very few)
- =>
- (assert (crisp2)
- (fuzzy2 hot)))
- (crisp1) CF 0.8 ($S=0.6$)
- (fuzzy1 few) CF 0.7
- CF of (crisp2) = $0.9 * \min(0.8, 0.7*0.6) = 0.378$
- CF of (fuzzy1 few) = $0.9 * \min(0.8, 0.7) = 0.63$

CF kod assert

- (assert (some-consequent) CF 0.8)
- (defrule assert-cf-rule
- (declare (CF 0.8)); rule CF is 0.8
- (fact 1)
- =>
- (assert (c1))
- (assert (c2) CF 0.7); assert c2 with CF 0.7
- (assert (c3))
- (assert (c4)))
- (fact 1) CF 0.9
- (assert (c1)); CF = $0.8 * 0.9 = 0.72$
- (assert (c2) CF 0.7); CF = $0.8 * 0.9 * 0.7 = 0.504$
- (assert (c3)); CF = $0.8 * 0.9 = 0.72$
- (assert (c4)); CF = $0.8 * 0.9 = 0.72$

Defuzzification – Centre of gravity

$$x' = \frac{\int_{(x \in U)} (x \cdot f(x)) dx}{\int_{(x \in U)} f(x) dx}$$

$$x' = \frac{\sum_{i=1}^n x_i' \cdot A_i}{\sum_{i=1}^n A_i}$$

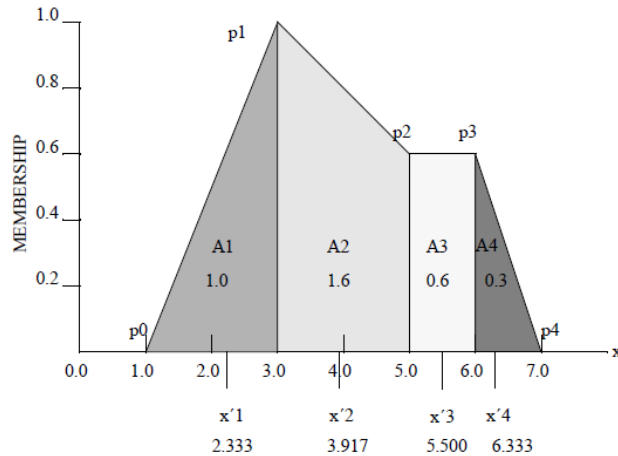


Figure 12: Example of COG defuzzification

For each shaded subsection in Figure 12, the area and centre of gravity is calculated according to the shape identified (i.e., triangle, rectangle or trapezoid). The centre of gravity of the whole set is then determined:

$$x' = \frac{2.333 \cdot 1.0 + 3.917 \cdot 1.6 + 5.5 \cdot 0.6 + 6.333 \cdot 0.3}{1.0 + 1.6 + 0.6 + 0.3} = 3.943$$

Defuzzification –mean of maxima

If the maximum y value is reached at more than one point, then the average of all the x'' is taken. More formally:

$$x' = \sum_{j=1}^J \frac{x_j''}{J}$$

where x_j'' are the x-coordinates of all the maxima, and J is the total number of maxima (see Figure 13).

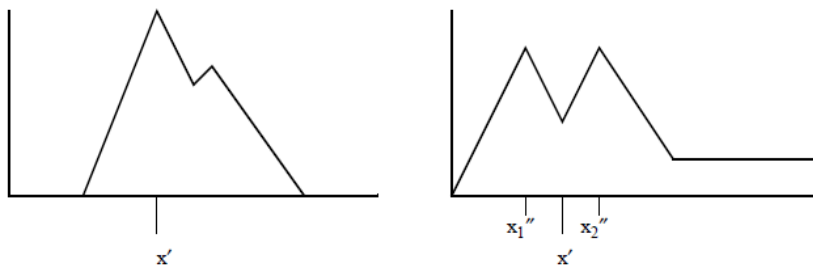


Figure 13: Examples of MOM defuzzification

Defuzzyfication –mean of maxima ambiguity - višeznačnost

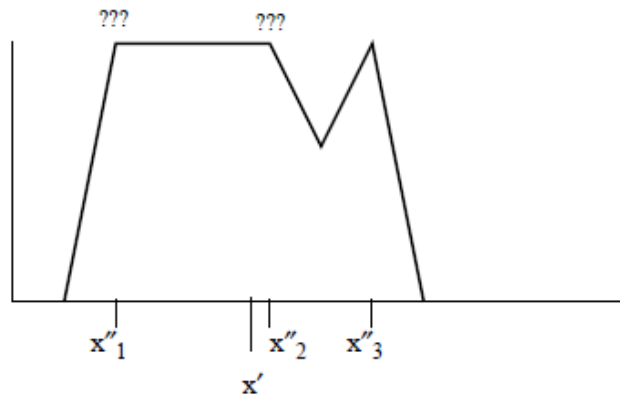


Figure 14: MOM example - Ambiguity

Fuzzy CLIPS Extensions fuzzy promenljive - deftemplate

- Neki elementi – konstrukcije standardnog CLIPS-a se koriste kao modifikovani za definisanje koncepata u Fuzzy CLIPS-u.
- Fuzzy promenljive se definišu koristeći modifikovanu konstrukciju deftemplate
- (deftemplate <name> [“<comments>”]
- <from> <to> [<unit>] ; universe of discourse
- (
 - t1
 - . ; list of primary terms
 - tn
-)
-)
- <name> je ime fuzzy promenljive

Fuzzy promenljive - deftemplate

- **(deftemplate water_flow**
- **0 100 liters/sec**
- ...
-)
- water_flow je ime fuzzy promenljive definisane preko defteplate konstrukcije
- Sadržaj fuzzy promenljive se definiše preko primary terms – primarnih članova
- Primary term
- (<name> <description of fuzzy set>)

Primary term

- (<name> <description of fuzzy set>)
- <name> je ime primarnog člana
- <description of fuzzy set> je funkcija pripadnosti
- Funkcija pripadnosti može biti zadata na tri načina:
- Preko singleton-a, standardnih funkcija i lingvističkih izraza
- Lingvistički izraz koristi prethodno definisane primarne članove
- <description of fuzzy set> ::= <singletons> | <standard> | <linguistic-expr>

Predstavljanje preko **singleton-a**

- $\mu_A(x)$ je funkcija pripadnosti koja određuje stepen pripadnosti elementa $x \in U$ (Universe of discourse - domen)
- Par $(\mu_A(x), x)$ se naziva singleton i može se još predstaviti kao $\mu_A(x)/x$ ili kratko $\mu(x)/x$
- Fuzzy set – skup A u domenu U (Universe of discourse) se definiše kao:

$$A = \int_{x \in U} \mu_A(x)/x$$

- Simbol integrala označava uniju singleton-a

Predstavljanje preko **singleton-a**

- Ako domen U sadrži konačan skup elemenata, onda se fuzzy skup A predstavlja preko singleton-a kao:

$$A = \sum_{i=1}^n \mu(x_i)/x_i = \mu_1/x_1 + \mu_2/x_2 + \dots + \mu_n/x_n$$

- Kod Fuzzy CLIPS-a domen U je uvek odsečak realne brojne ose a nikada konačan diskretni skup elemenata.
- Singleton je par $(x_i, \mu(x_i))$ a Fuzzy skup A se predstavlja nizom singleton-a
- $\langle \text{singletons} \rangle ::= (x_1 \mu_1) (x_2 \mu_2) \dots (x_n \mu_n)$

Predstavljanje preko **singleton**-a

- $\langle \text{singletons} \rangle ::= (x_1 \mu_1) (x_2 \mu_2) \dots (x_n \mu_n)$
- $x_i \leq x_{i+1}$ for $i = 1, 2, \dots, n-1$
- $x_i \in U$
- μ_i je broj koji označava stepen pripadnosti elementa x fuzzy skupu A
- Funkcija pripadnosti elemenata fuzzy skupa A se predstavlja kao uređeni skup tačaka koje su povezane pravim linijama
- Vrednosti pripadnosti u tačkama x_i su date sa μ_i , dok se vrednosti pripadnosti između tačaka izračunavaju linearnom interpolacijom

Linearna interpolacija za singleton-e

$$\mu(x) = \mu(x_1), \quad x \leq x_1$$

$$\mu(x) = \mu(x_i) + \frac{\mu(x_{i+1}) - \mu(x_i)}{x_{i+1} - x_i}(x - x_i), \quad x_i < x \leq x_{i+1}$$

$$\mu(x) = \mu(x_n), \quad x_n < x$$

- Funkcija pripadnosti za elemente $x \leq x_1$ jednaka je μ_1 , dok je za elemente $x \geq x_n$ jednaka μ_n .
- Za vrednosti $x_1 \leq x \leq x_n$ koristi se gornja interpolaciona formula koja je već ugrađena u Fuzzy CLIPS

Primer funkcije pripadnosti zadate sa singleton-ima

- $U = \{x \mid 0 \leq x \leq 9\}$
- fuzzy set pod nazivom “few” je definisan skupom tačaka koji određuje odsečke $\mu(x)$
- $\mu(0) = 0, \mu(1) = 0, \mu(2) = 0.3, \mu(3) = 0.9,$
- $\mu(4) = 1, \mu(5) = 0.8, \mu(6) = 0.5, \mu(7) = 0,$
- $\mu(8) = 0, \mu(9) = 0$
- Ovom skupu tačaka odgovara niz singleton-a:
- $(1 \ 0) (2 \ 0.3) (3 \ 0.9) (4 \ 1) (5 \ 0.8) (6 \ 0.5) (7 \ 0)$
- Odgovarajući grafički prikaz je dat na sledećoj slici

Grafički prikaz fuzzy skupa “few”

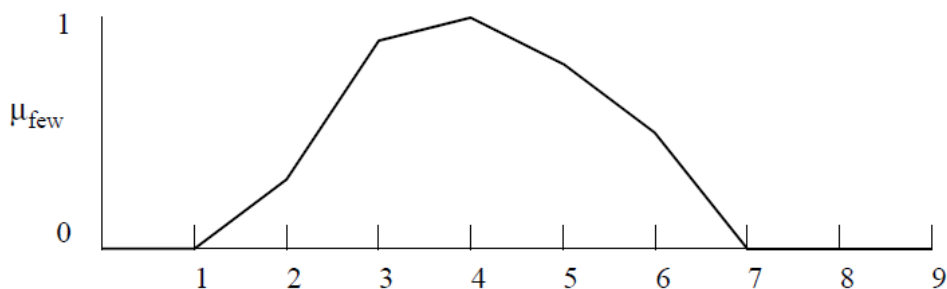


Figure 15: Fuzzy Set of group few

Fuzzy CLIPS sintaksa za fuzzy promenljivu “group” i primarni član “few” zadat nizom singleton-a

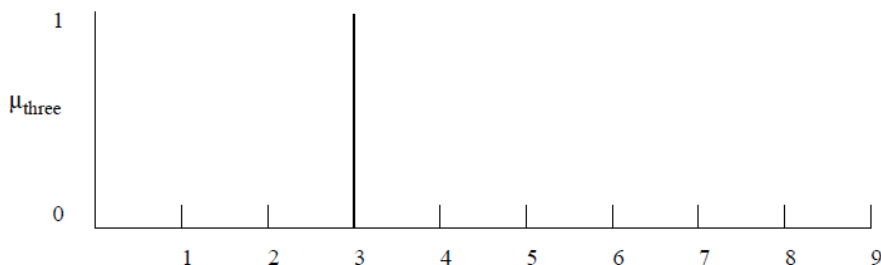
```
(deftemplate group          ;a linguistic variable declaration
  0 9                      ;universe of discourse limits (no units)
  (                        ;start of primary term declarations
    ; a primary term few described in singleton notation
    (few (1 0) (2 0.3) (3 0.9) (4 1) (5 0.8) (6 0.5) (7 0))
  )                        ;end of primary term declarations
)                          ;end of fuzzy deftemplate
```

Predstavljanje tačne – crisp vrednosti 3 kao fuzzy skup preko niza singleton-a

The singleton set described as

```
(three (3 0) (3 1) (3 0))
```

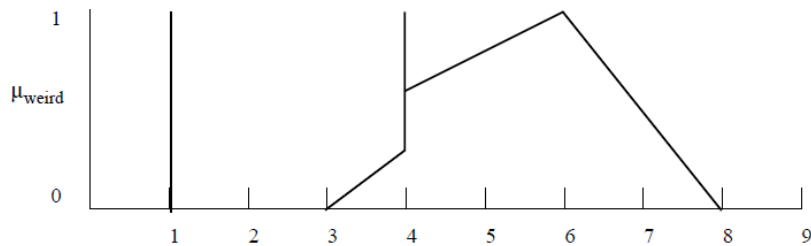
might represent the crisp concept 3 as a fuzzy set. It is shown graphically as



Another more complex (and probably unrealistic set might be defined with the following set of singleton values

```
(weird      (1 0) (1 1) (1 0) (3 0) (4 .25)
             (4 1) (4 .4) (4 .5) (6 1) (8 0) )
```

The graph follows. Note that in this case the point (4 .4) is discarded.



Standardna funkcija S

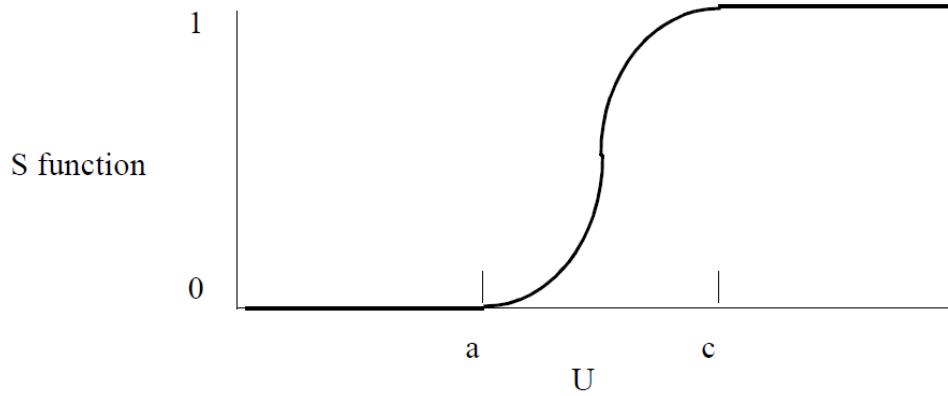
$$S(u, a, c) = 0, \quad u \leq a, u \in U$$

$$S(u, a, c) = 2 \left(\frac{u-a}{c-a} \right)^2, \quad a < u \leq \frac{a+c}{2}$$

$$S(u, a, c) = 1 - 2 \left(\frac{c-u}{c-a} \right)^2, \quad \frac{a+c}{2} < u \leq c$$

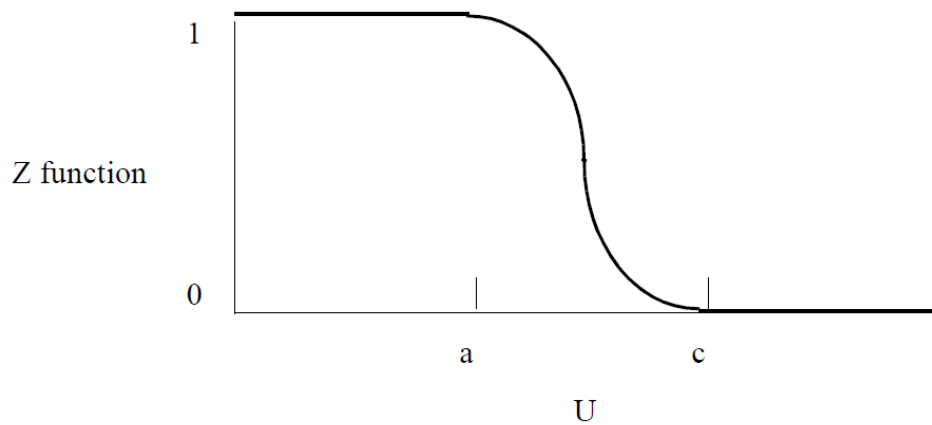
$$S(u, a, c) = 1, \quad c < u$$

Grafički prikaz S funkcije



Z funkcija

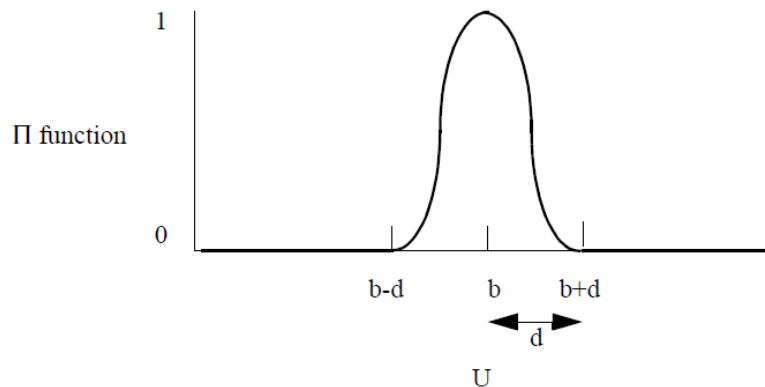
$$Z(u, a, c) = 1 - S(u, a, c)$$



Π funkcija

$$\Pi(u, d, b) = S(u, b - d, b), \quad u \leq b$$

$$\Pi(u, d, b) = Z(u, b, b + d), \quad b < u$$



Sintaksa za standardne funkcije

- $\langle \text{standard} \rangle ::= (S \ a \ c) \mid (s \ a \ c) \mid (Z \ a \ c) \mid (z \ a \ c) \mid (PI \ d \ b) \mid (pi \ d \ b)$
- **(deftemplate Tx “output water temperature”**
- **5 65 Celsius**
- **((cold (z 10 26)) ;standard set representation**
- **(OK (PI 2 36))**
- **(hot (s 37 60))**
- **)**
- **)**

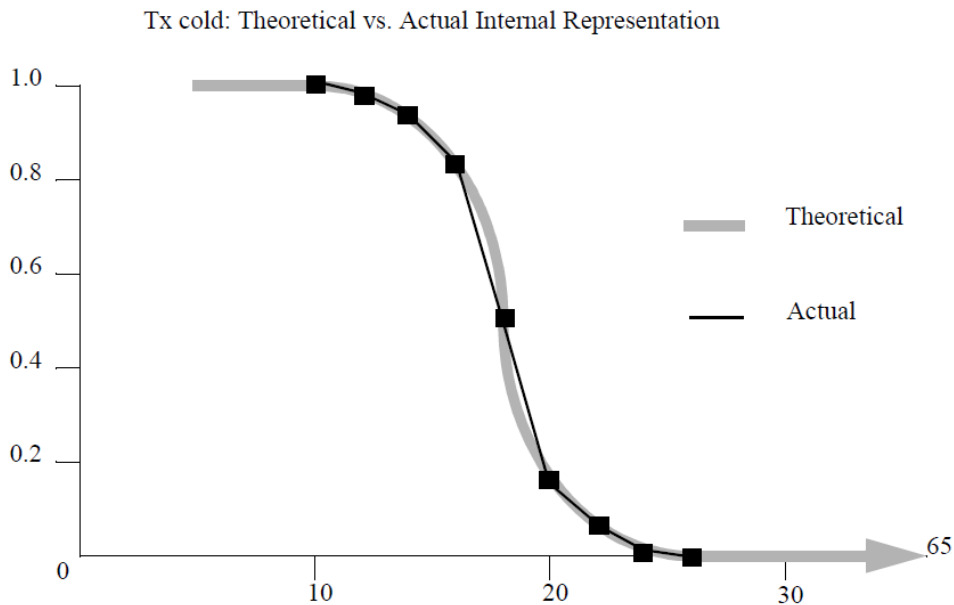


Figure 17: Approximation of standard functions

```
Fuzzy fact(fuzzy-slot> ::= (slot <slotname> (type FUZZY-VALUE
                                <fuzzy-deftemplate-name>)
                                s i fuzzy slots
```

- ;; assume that the fuzzy deftemplates fz-height and
- ;; fz-weight have already been defined
- (deftemplate person
- (slot name (type SYMBOL))
- (slot height (type FUZZY-VALUE fz-height))
- (slot weight (type FUZZY-VALUE fz-weight))
-)
- (defrule big-person
- (person (name ?n)
- (weight heavy)
- (height tall))
- =>
- (printout t ?n " is a big person" crlf)
-)

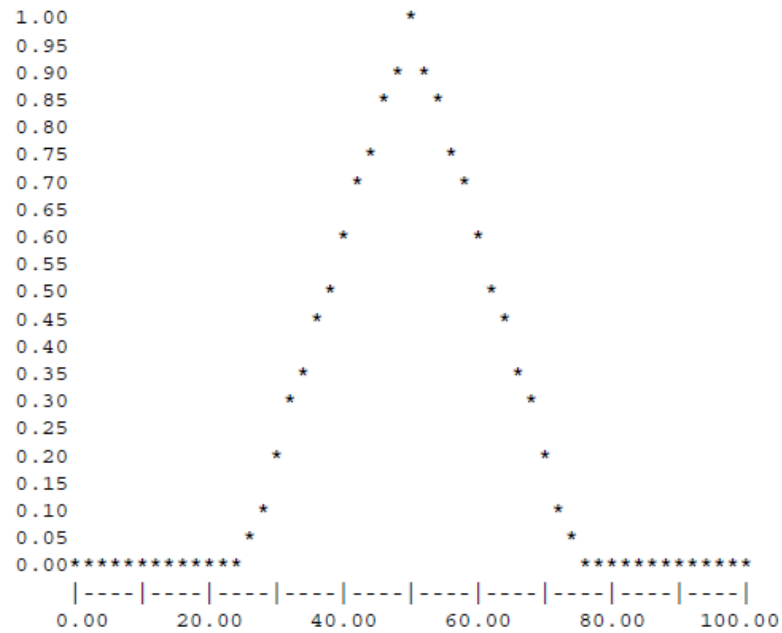
Modifiers (Hedges) and Linguistic Expressions

• Modifier Name	Modifier Description
• not	$1-y$
• very	y^{**2}
• somewhat	$y^{**0.333}$
• more-or-less	$y^{**0.5}$
• extremely	y^{**3}
• above	(see [12])
• below	(see [12])

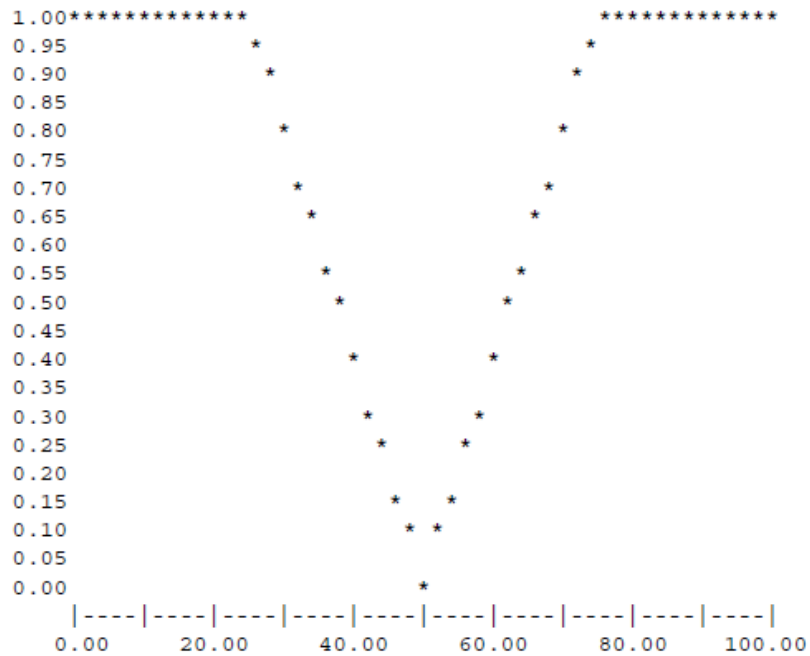
Modifiers (Hedges) and Linguistic Expressions

• Modifier Name	Modifier Description
• Intensify	$2 (y^{**2})$ if y in $[0, 0.5]$
• ----- -----	$1 - 2(1-y)^{**2}$ if y in $(0.5, 1]$
• plus	$y^{**1.25}$
• norm	normalizes the fuzzy set so that the maximum value of the set is scaled to be 1.0 ($y = y*1.0/\text{max-value}$)
• slightly	intensify (norm (plus A AND not very A))

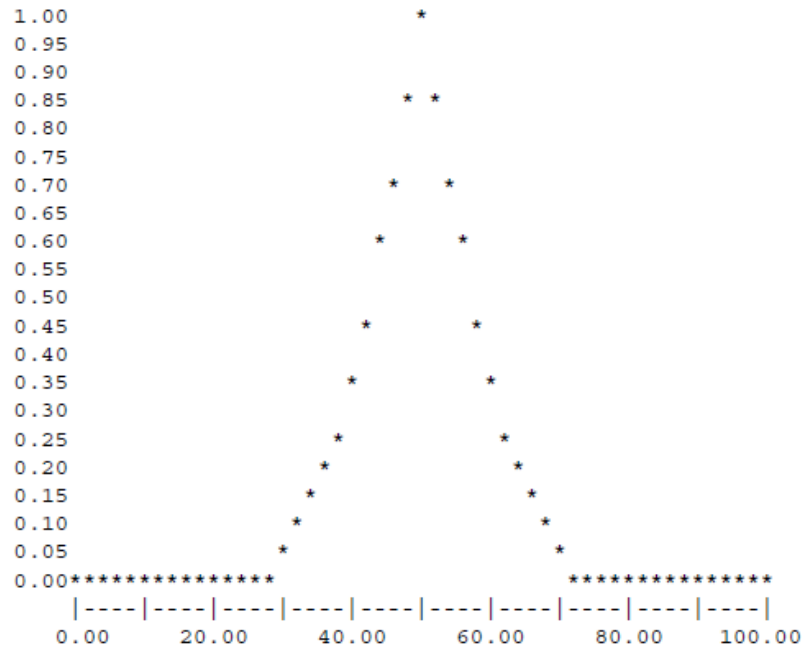
Fuzzy Value: base-fv
Linguistic Value: base (*)



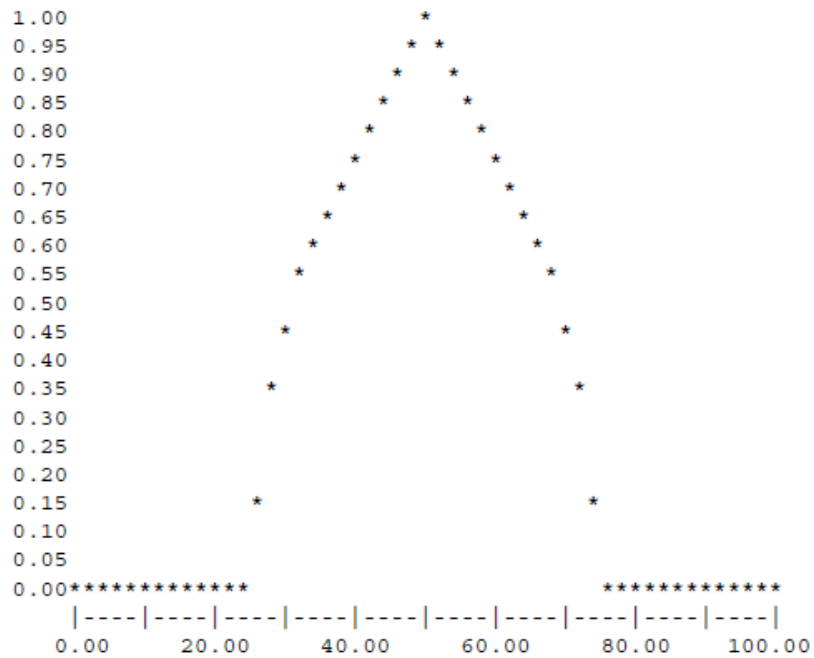
Fuzzy Value: base-fv Linguistic Value: not base (*)



Fuzzy Value: base-fv Linguistic Value: very base (*)



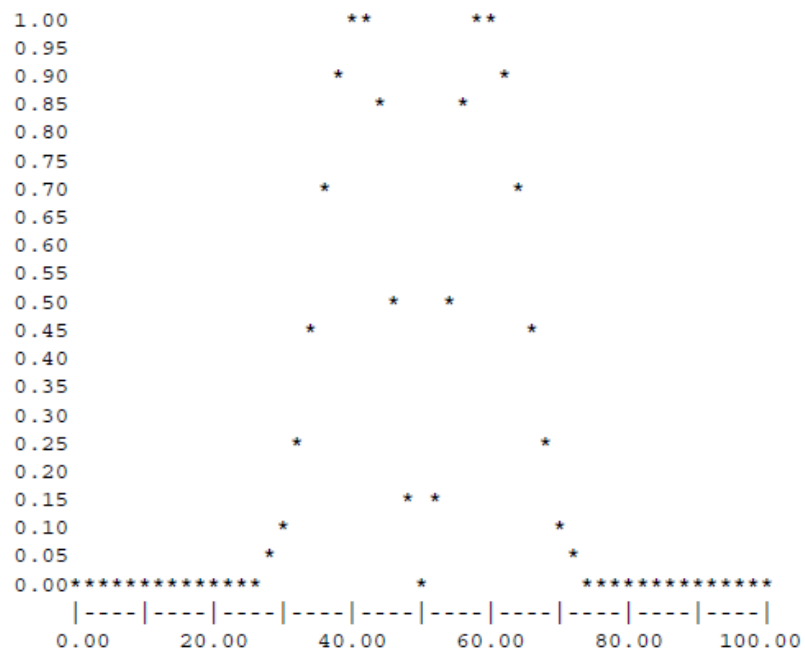
Fuzzy Value: base-fv Linguistic Value: more-or-less base (*)



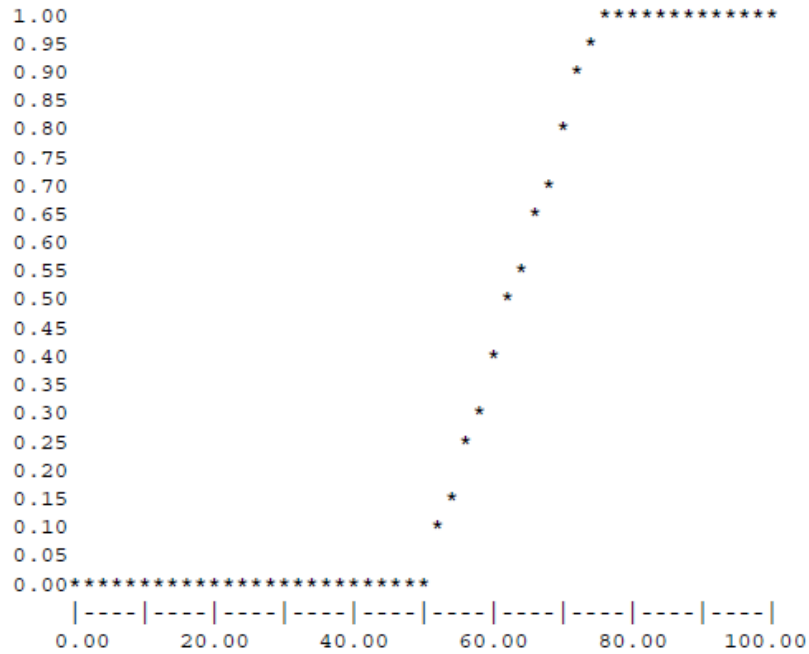
Fuzzy Value: base-fv Linguistic Value: somewhat base (*)



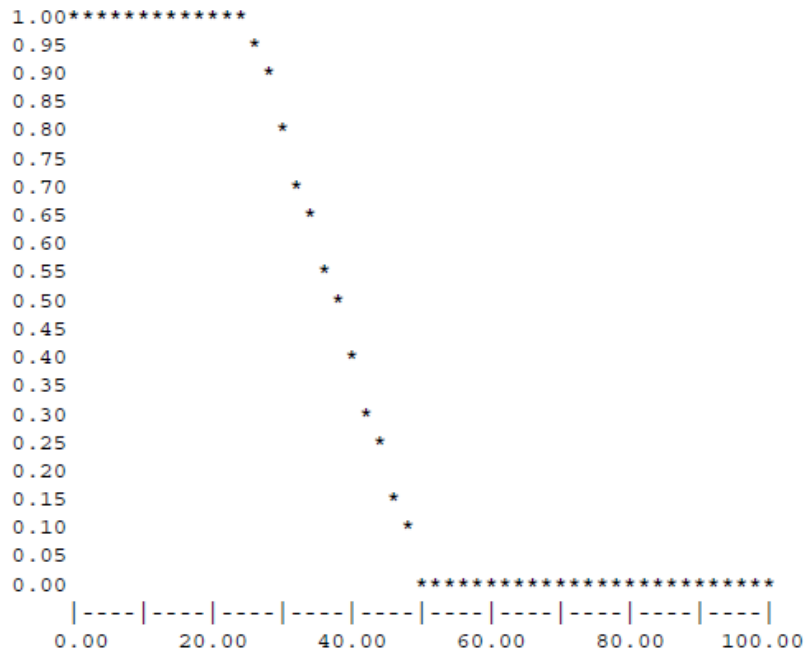
Fuzzy Value: base-fv Linguistic Value: slightly base (*)



Fuzzy Value: base-fv Linguistic Value: above base (*)



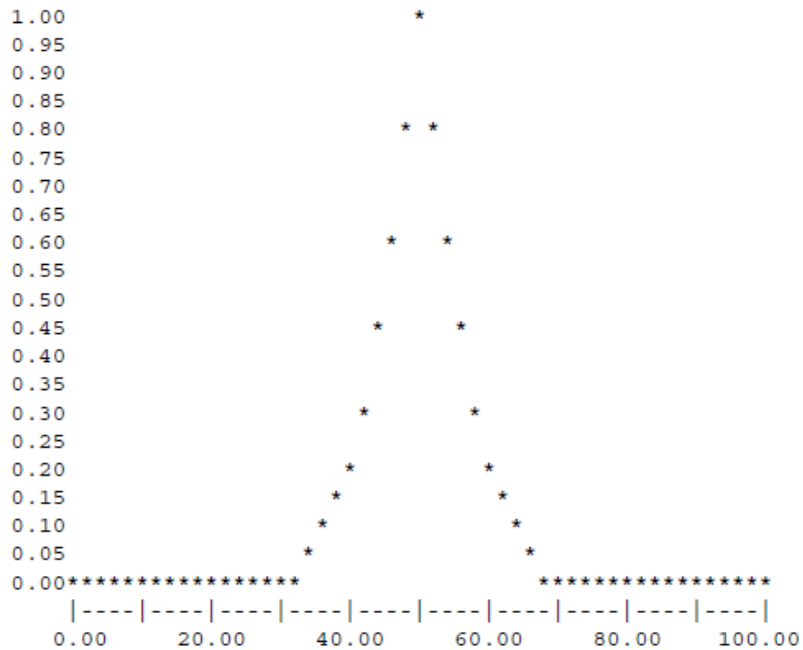
Fuzzy Value: base-fv Linguistic Value: below base (*)



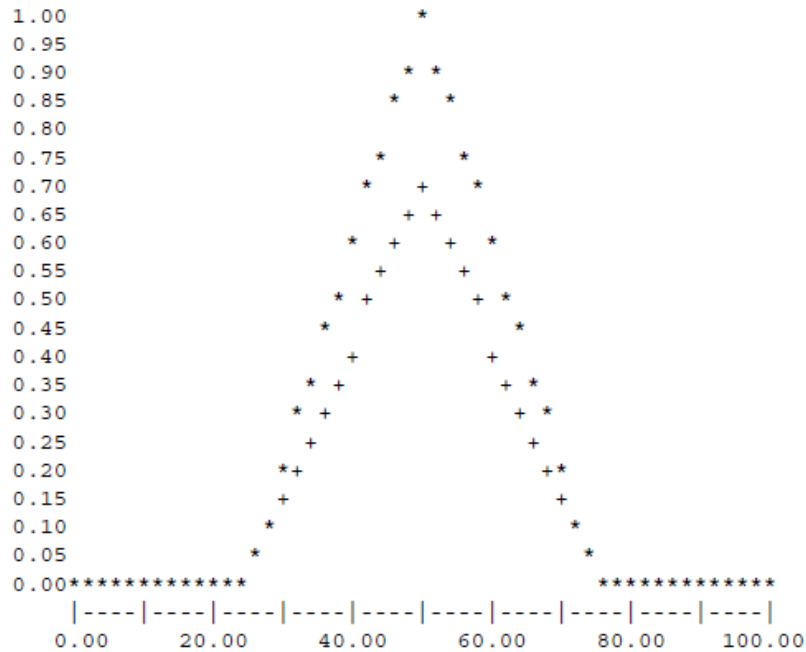
Fuzzy Value: base-fv Linguistic Value: intensify base (*)



Fuzzy Value: base-fv Linguistic Value: extremely base (*)



Fuzzy Value: base-fv2Linguistic Value: base (+), norm base (*)



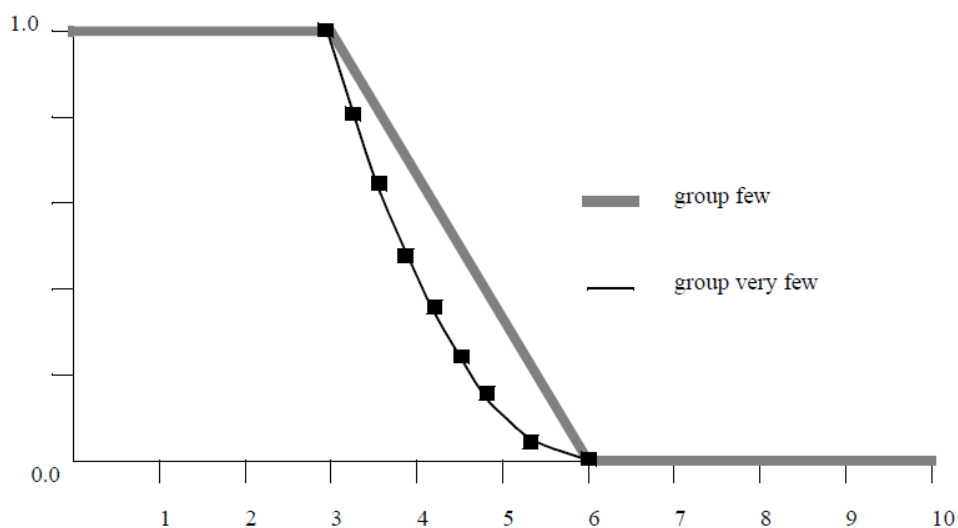
Predefinisani modifikatori

- Predefinisani modifikatori su uvek dostupni za korišćenje u fuzzy deftemplate konstrukcijama, fuzzy pattern matching kod pravila, za fuzzy slot assert i za fuzzy modify
- (deftemplate temp
- 0 100 C
- ((cold (Z 20 40))
- (hot (S 60 80))
- (freezing extremely cold)
-)
-)
- (defrule temp-rule
- (temp not hot and not cold)
- =>
- (printout t "It's such a pleasant day" crlf)
-)

User Defined Modifiers – korisnički modifikatori

- `(add-fuzzy-modifier modname modfunction)`
- `(add-fuzzy-modifier my-somewhat sqrt)`
- `(deffunction most-extremely-fcn (?x))`
- `(** ?x 5)`
- `)`
- `(add-fuzzy-modifier most-extremely most-extremely-fcn)`
- `(deftemplate temp`
- `0 100 C`
- `((low (10 1) (50 0))`
- `(high (50 0) (90 1))`
- `(sort-of-high my-somewhat high)`
- `(incredibly-low most-extremely low)`
- `)`
- `)`

Few – nekoliko, very few – manje od nekoliko



Linguistic Expressions

- Lingvistički izrazi omogućavaju adekvatno matematičko karakterisanje – modelovanje raznih vrsta neodređenih iskaza iz realnog sveta korišćenjem primarnih članova, modifikatora i logičkih operatora and i or.
- temperature very hot or very cold
- height below tall and above short
- $\langle \text{LExp} \rangle ::= \langle \text{LTerm} \rangle \mid \langle \text{LExp} \rangle \text{ OR } \langle \text{LTerm} \rangle$
- $\langle \text{LTerm} \rangle ::= \langle \text{modExp} \rangle \mid \langle \text{LTerm} \rangle \text{ AND } \langle \text{modExp} \rangle$
- $\langle \text{modExp} \rangle ::= \text{MODIFIER } \langle \text{modExp} \rangle \mid \langle \text{element} \rangle$
- $\langle \text{element} \rangle ::= \text{PRIMARY-TERM} \mid [\langle \text{LExp} \rangle]$

Linguistic Expressions

- MODIFIER je valid modifier (not, very, etc.)
- PRIMARY-TERM je term - član defined in a fuzzy deftemplate
- Logički operator AND ima viši prioritet nad operatorom OR
- Sledeći izrazi su ekvivalentni:
 - A or B and C or D
 - A or [B and C] or D

fuzzy sets *temp hot .*, *temp cold +*, and
*temp not [hot or cold] **.

Fuzzy Value: temp
 Linguistic Value: hot (.), cold (+), not [hot OR cold] (*)



Using Fuzzy Variables in LHS Patterns

- A fuzzy LHS pattern is of the form
- (fuzzy-variable-name <linguistic-expr>)
- or
- (fuzzy-variable-name ?)
- or
- (fuzzy-variable-name ?<var-name>)
- or
- (fuzzy-variable-name ?<var-name> & <linguistic-expr>)
- or
- (template-name <slot-description>+)

Using Fuzzy Variables in LHS Patterns

- + indicates that there are one or more of the <slot-description> entries, at least
- one of these is a <fuzzy-slot-description>, and a <fuzzy-slot-description> is
- (fuzzy-slot-name <linguistic-expr>)
- or
- (fuzzy-slot-name ?)
- or
- (fuzzy-slot-name ?<var-name>)
- or
- (fuzzy-slot-name ?<var-name> & <linguistic-expr>)

Using Fuzzy Variables in LHS Patterns

- The <linguistic-expr> is a fuzzy set specified by a combination of primary terms, modifiers, and the logical operators
- NOT and OR are logical operators and fuzzy-variable-name is the name of any fuzzy deftemplate.
- A template-name is the name of any non-fuzzy deftemplate. A fuzzy-slot-name is the name of a slot declared to have type FUZZY-VALUE

- **(deftemplate group ;declaration of fuzzy variable group**
- **0 20 members**
- **((few (3 1) (6 0)) ;primary term few**
- **(many (4 0) (6 1)) ;primary term many**
- **)**
- **)**

- **(defrule simple-LHS**
- **(group few) ;a simple fuzzy LHS pattern**
- **...**
- **)**

LE u LHS pravila

- **(defrule more-complex-lhs**
- **?f <- (group very few or very many)**
- **=>**
- **(printout t "We are at the extreme limits of the number of" (get-u-units ?f) " in our club" crlf)**
- **)**

- **get-u-units je funkcija koja daje jedinicu za fuzzy fact sa adresom ?f koja zadovoljava LHS pravila**

- (deftemplate height
- 0 8 feet
- ((short (Z 3 4.5))
- (medium (pi 0.8 5))
- (tall (S 5.5 6))
-)
-)
- (deftemplate person
- (slot name)
- (slot ht (type FUZZY-VALUE height))
-)

Složeno pravilo

- (defrule quite-complex-lhs
- (person (name ?n) (ht ?h & very tall))
- =>
- (printout t ?n " is very tall, with a height of about "
- (maximum-defuzzify ?h) " " (get-u-units ?h) crlf)
-)
- maximum-defuzzify je funkcija koja vraća crisp vrednost koja se određuje prema maximumu

Using Fuzzy Variables in Deffacts Constructs

- (deffacts <deffacts-name> [<comment>]
- <RHS-pattern>*
-)
- <RHS-pattern> ::= <ordered-RHS-pattern> |
- <template-RHS-pattern> |
- <fuzzy-template-RHS-pattern>
- <ordered-RHS-pattern> ::= (<symbol> <RHS-field>+)
- [CF <certainty factor> | <certainty factor expression>]

Using Fuzzy Variables in Deffacts Constructs

- <template-RHS-pattern> ::= (deftemplate-name> <RHS-slot>*) [CF <certainty factor> | <certainty factor expression>]
- <fuzzy-template-RHS-pattern> ::=
- (<fuzzy-template-name> <description of fuzzy set>
- [CF <certainty factor> | <certainty factor expression>]

- (deffacts groupA “some fuzzy facts”
- (my_group (1 0) (5 1) (7 0)) ;singleton description
- (your_group (z 4 8)) ;standard description
- (their_group (s (+ 1 1) 4))
- (person (name ralph) (ht tall))
-)
- (deffacts groupB “some fuzzy facts with certainty factors”
- (this_group (1 0) (5 1) (7 0)) CF 0.35
- (that_group (pi 2 (+ 3 4))) CF (+ .2 .3)
-)

Using Fuzzy Variables in Assert Statements

- (assert
- (<crisp fact> | fuzzy-variable-name <*description of fuzzy set*> / template-name <slot-description>+)
- [CF <*certainty factor*> | <*certainty factor expression*>]
-)
- (fuzzy-slot-name <*description of fuzzy set*>)
- <*description of fuzzy set*> is
- <linguistic-expr> | <standard> | <singletons>

Using Fuzzy Variables in Assert Statements

- `(assert (group few))`
- `(assert (group (1 0) (5 1) (7 0)))`
- `(assert (group NOT [very few OR many]))`
- `(assert (group (z 4 8)))`
- `(assert (person (name john) (ht extremely tall)))`
- `(assert (person (name dan) (ht (pi 0 5.6))))`
- `(assert (temp (24 0) (25 1) (26 0)))`

Using Fuzzy Variables in Assert Statements

- `(defrule assert-rule-1`
- `(zmin ?minval) ;variable ?minval must be numeric`
- `(zmax ?maxval) ;variable ?maxval must be numeric`
- `=>`
- `(assert`
- `(group (z ?minval ?maxval))`
- `) ;fuzzy set description with variables`
- `)`

Using Fuzzy Variables in Assert Statements

- (defrule assert-rule-2 ;asserts standard set with functions
 - (zmin ?minval)
 - (zmax ?maxval)
 - =>
 - (assert (group (z ?minval (+ ?maxval 2))))
 -)
-
- (defrule assert-rule-3; asserts singleton set with functions
 - (x1 ?x1val)
 - (x2 ?x2val)
 - =>
 - (assert
 - (group (?x1val 0) (?x2val 1) ((+ ?x2val 1) 0.6)
 - ((sqr ?x2val) 0))
 -)
 -)

Using Fuzzy Variables in Assert Statements

- `(assert-string "(group (z 4 8))")`
- `(assert-string "(group (s 4 (+ 5 3)))")`
- `(assert-string "(person (name bob) (ht medium))")`
- `(assert (somefact) CF 0.8) ;with CF of 0.8`
- `(assert (group few) CF (+ 0.2 0.4)) ;with CF of 0.6`

Using Fuzzy Variables in Assert Statements

- `(defrule assert-rule-4 ;illustrates various CF assertions`
- `(certainty-factor ?cf) ;where ?cf is between 0 and 1`
- `?f <- (somefact)`
- `=>`
- `(assert (fact1) CF ?cf)`
- `(assert (fact3) CF (* 0.8 ?cf))`
- `(assert (fact5) CF (get-cf ?f))`
- `;get-cf is a function discussed in Section 5.9`
- `)`

Defuzzification

- Defuzzification je postupak dobijanja crisp vrednosti od fuzzy set-a
- Postoje dva osnovna načina za defuzzification – COG i MOM
- (moment-defuzzify ?<fact-var> | integer | <fuzzy-value>) ; COG algorithm
- (maximum-defuzzify ?<fact-var> | integer | <fuzzy-value>) ; MOM algorithm
- fact-1 (temperature warm)
- (defrule defuzzification-1
- ?f <- (temperature ?) ;? used to assure match of the
- ;temperature fuzzy fact
- =>
- (bind ?t (maximum-defuzzify ?f)) ;get the value
- (printout t "Temperature is " ?t crlf) ;print 32.5
-)
- (defrule defuzzification-2
- (temperature ?fv) ;?fv used to hold the fuzzy value
- ;of the matching fuzzy fact
- =>
- (printout t "Temperature is " (maximum-defuzzify ?fv) crlf)
- \

Certainty Factors of Rules

- (defrule some-rule
- (declare (CF *<certainty factor>*))
-
- =>
-
-)

- (deffacts initial-facts
- (fact1) CF 0.8 ;fact with crisp CF of 0.8
-)

Certainty Factors of Rules

- (defrule some-other-rule
- (declare (CF 0.7)) ;a rule with CF of 0.7
- (fact1)
- =>
- (printout t "Hello!")
-)

FuzzyCLIPS Commands and Functions

- Funkcije za Accessing the Universe of Discourse
- get-u
- get-u-from
- get-u-to
- get-u-units

Command: get-u

- Syntax:
- (get-u ?<fact-var>) or
- (get-u <integer>) or
- (get-u <fuzzy-template-name>) or
- (get-u <fuzzy-value>)
- ?<fact-var> is a fact variable
- <integer> is the number of a fact on the fact list (constant or expression)
- <fuzzy-template-name> is a symbol that represents the name of a fuzzy deftemplate
- <fuzzy-value> is an element of type FUZZY-VALUE

Command: get-u

- (get-u ?t) ;; ?t is bound to the temp fuzzy fact
- (get-u 2) ;; 2 is a fact index
- (get-u temp) ;; temp is the name of the fuzzy deftemplate
- (defrule test
- (temp ?fv) ;; ?fv is a fuzzy value
- =>
- (printout t (get-u ?fv))
-)

Command: get-u-from

- Syntax: (get-u-from ?<fact-var>) or
- (get-u-from <integer>) or
- (get-u-from <fuzzy-template-name>) or
- (get-u-from <fuzzy-value>)
- Purpose: Returns the lower limit of the universe of discourse in floating point format.
- **Command: get-u-to**
- **Command: get-u-units**

- **Command: get-fs**
- Syntax: (get-fs ?<fact-var>) or
- (get-fs <integer>) or
- (get-fs <fuzzy-value>)
- Purpose: Returns the entire fuzzy set in singleton representation, in string format.
- **Command: get-fs-length**
- Syntax: (get-fs-length ?<fact-var>) or
- (get-fs-length <integer>) or
- (get-fs-length <fuzzy-value>)
- Purpose: Returns the number of pairs in a fuzzy set description as an integer.
- **Command: get-fs-x**
- Syntax: (get-fs-x ?<fact-var> <i>) or
- (get-fs-x <integer> <i>) or
- (get-fs-x <fuzzy-value> <i>)
- where <i> is an integer, variable, or function expression.
- **Command: get-fs-y**
- Syntax: (get-fs-y ?<fact-var> <i>) or
- (get-fs-y <integer> <i>) or
- (get-fs-y <fuzzy-value> <i>)

- **Command: get-fs-lv**
- Syntax: (get-fs-lv ?<fact-var>) or (get-fs-lv <integer>) or (get-fs-lv <fuzzy-value>)
- Purpose: Returns the linguistic value associated with the fuzzy set
- **if (temp very hot) is asserted and the variable ?fuzzyfact is assigned to this fact then the function call**
- **(get-fs-lv ?fuzzyfact)**
- **would return the string "very hot"**

- **Command: get-fs-value**
- Syntax: (get-fs-value ?<fact-var> <number>) or (get-fs-value <integer> <number>) or (get-fs-value <fuzzy-value> <number>)
- Purpose: Returns the value of the fuzzy set at the specified x value (<number>). The <number> is a value that must lie between the lower and upper limits of the universe of discourse for the fuzzy set.
- **(OK (30 0) (60 1) (90 0))**
- **(get-fs-value ?fact 50.0) 0.6666667**

Accessing the Certainty Factor (get-cf)

- **Command:** `get-cf`
- Syntax: `(get-cf ?<fact-var>)`
- `(get-cf <integer>)`
- Purpose: Returns the certainty factor of a fact as a floating point number.

Accessing the Threshold Certainty Factor (threshold, get-threshold)

- **Command:** `set-threshold1`
- Syntax: `(set-threshold <NUMBER>)`
- Purpose: Sets threshold certainty factor to the value of `<NUMBER>`.
- `<NUMBER>` must evaluate to a floating value between 0.0 and 1.0.
- By default the threshold value is 0.0.
- **Command:** `get-threshold`
- Syntax: `(get-threshold)`
- Purpose: Returns the floating point value of the threshold certainty factor if threshold capability is ON. If it is OFF, then a value of 0.0 is returned.

Setting the Rule CF Evaluation Behaviour (set-CF-evaluation, get-CF-evaluation)

- **Command: set-CF-evaluation**
- Syntax: (set-CF-evaluation <value>)
- Purpose: Sets the behavior for evaluating the CF of rules to <value>. of **when-defined (default) or when-activated**
- **Command: get-CF-evaluation**
- Syntax: (get-CF-evaluation)
- Purpose: Returns the current setting of the behavior for evaluating the CF of rules. Return **when-defined (default) or when-activated**

Controlling the Fuzzy Set Display Precision (set-fuzzy-display-precision, get-fuzzy-display-precision)

- **Command: set-fuzzy-display-precision**
- Syntax: (set-fuzzy-display-precision <integer>) 2 –16
- **(set-fuzzy-display-precision 16)**
- **(facts)**
- **f-0 (speed_error more_or_less large_positive) CF 1.00**
- **((0.0 0.0) (0.1 0.3162277660168379)**
- **(0.2 0.4472135954999579)**
- **Command: get-fuzzy-display-precision**
- Syntax: (get-fuzzy-display-precision)

Controlling the Fuzzy Inference Method (set-fuzzy-inference-type, get-fuzzy-inference-type)

- **Command: set-fuzzy-inference-type**
- Syntax: (set-fuzzy-inference-type <inf-type>)

- *max-min or max-prod*

- **Command: get-fuzzy-inference-type**
- Syntax: (get-fuzzy-inference-type)

Setting the Fuzzy Pattern Matching Threshold

- **Command: set-alpha-value**
- Syntax: (set-alpha-value <alpha-val>)
- **Command: get-alpha-value**
- Syntax: (get-alpha-value)
- Alpha-value je min vrednost preseka dva fuzzy skupa – činjenice i LHS vrednosti promenljive
- Po default-u je 0, što znači da bilo koji presek dovodi do aktivacije

- **Command: fuzzyvaluep**
- Syntax: (fuzzyvaluep <arg>) true / false

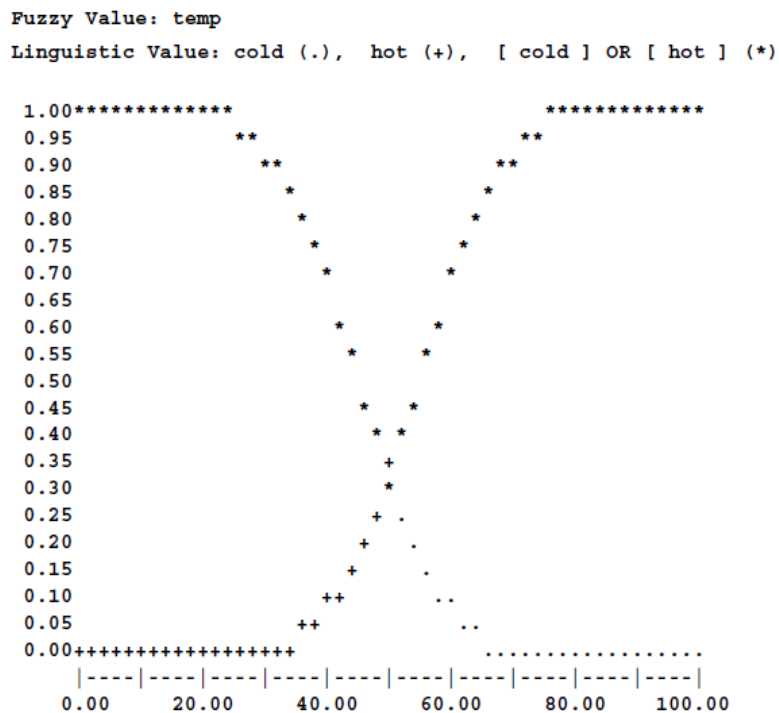
- **Command: create-fuzzy-value**
- Syntax: (create-fuzzy-value <fuzzy-deftemplate-name> <description of fuzzy set>)

- **(create-fuzzy-value temp cold)**
- **(create-fuzzy-value temp very hot or very cold)**
- **(create-fuzzy-value temp (pi 10 20))**
- **(create-fuzzy-value temp (s ?x (+ ?x 10)))**
- **(create-fuzzy-value temp (10 1) (20 0))**

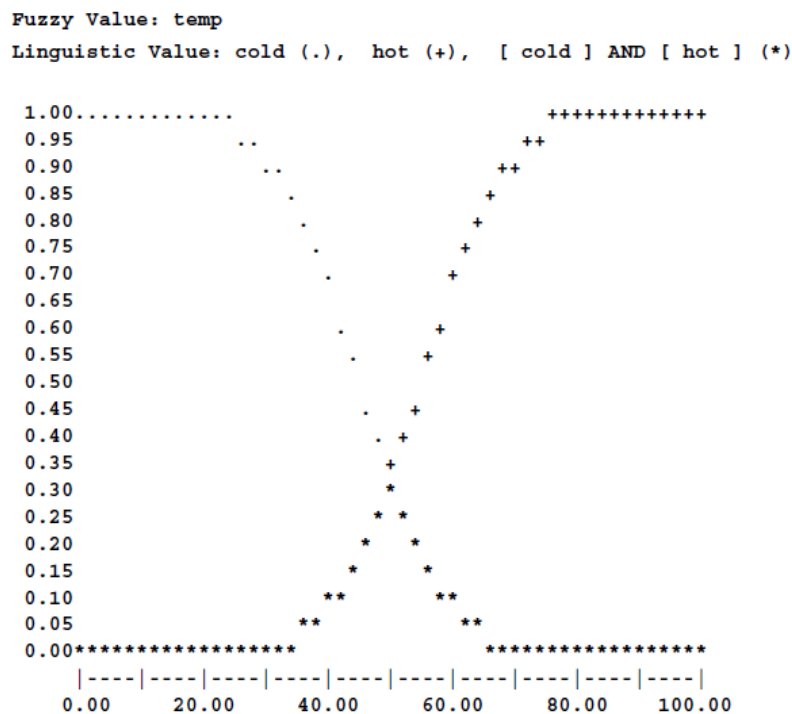
- **Command: fuzzy-union**
- Syntax: (fuzzy-union <fuzzy-value> <fuzzy-value>)
- **(deftemplate temp**
- **0 100 C**
- **((cold (z 20 70))**
- **(hot (s 30 80))**
- **)**
- **)**
- **(fuzzy-union (create-fuzzy-value temp cold)**
- **(create-fuzzy-value temp hot))**
- **cold or hot**

Crtanje fuzzy vrednosti

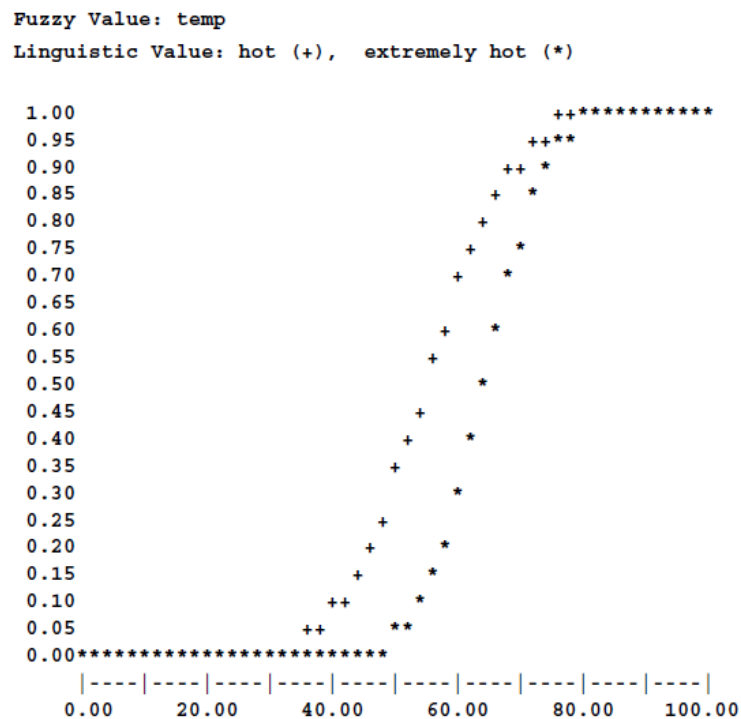
- (plot-fuzzy-value t ".+*" nil nil
- (create-fuzzy-value temp cold)
- (create-fuzzy-value temp hot)
- (fuzzy-union (create-fuzzy-value temp cold)
- (create-fuzzy-value temp hot))
-)



- **Command: fuzzy-intersection**
- Syntax: (fuzzy-intersection <fuzzy-value> <fuzzy-value>)
- (fuzzy-intersection (create-fuzzy-value temp cold)
- (create-fuzzy-value temp hot))
- cold and hot
- (plot-fuzzy-value t ".+*" nil nil
- (create-fuzzy-value temp cold)
- (create-fuzzy-value temp hot)
- (fuzzy-intersection (create-fuzzy-value temp cold)
- (create-fuzzy-value temp hot))



- **Command: fuzzy-modify**
- Syntax: (fuzzy-modify <fuzzy-value> <modifier>)
- (plot-fuzzy-value t "+"* nil nil
- (create-fuzzy-value temp hot)
- (fuzzy-modify (create-fuzzy-value temp hot) extremely))

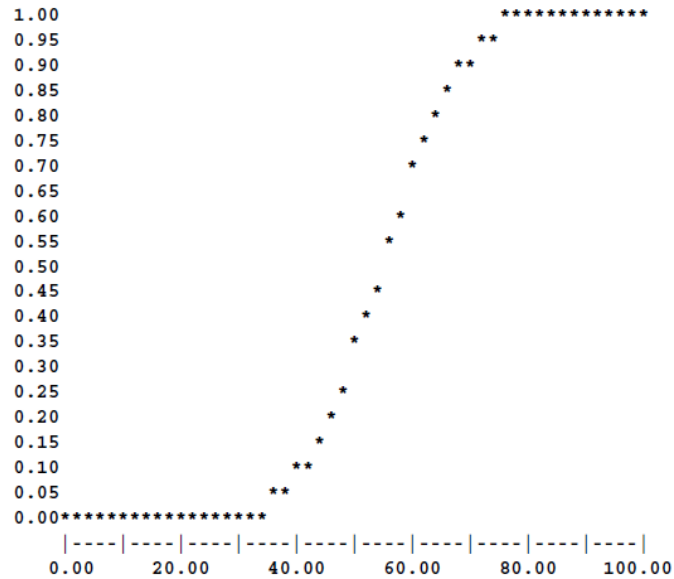


- **Command: get-fuzzy-slot**
- Syntax: (get-fuzzy-slot ?<fact-var> [<slot-name>])
- (get-fuzzy-slot <integer> [<slot-name>])
- **(defrule test1-get-fuzzy-slot**
- **?f <- (temp hot)**
- **=>**
- **(plot-fuzzy-value t * nil nil (get-fuzzy-slot ?f))**
- **)**
- **(defrule test2-get-fuzzy-slot**
- **?f <- (system (name sysA) (t-outflow hot))**
- **=>**
- **(plot-fuzzy-value t * nil nil (get-fuzzy-slot ?f t-outflow))**
- **)**

Plotting a Fuzzy Value (plot-fuzzy-value)

- **Command: plot-fuzzy-value**
- Syntax: (plot-fuzzy-value <logicalName> <plot-chars> <low-limit> <high-limit> <fuzzy-value>+)
- **(deftemplate temp**
- **0 100 C**
- **((cold (z 20 70))**
- **(hot (s 30 80))**
- **)**
- **)**
- **(plot-fuzzy-value t * nil nil (create-fuzzy-value temp hot))**

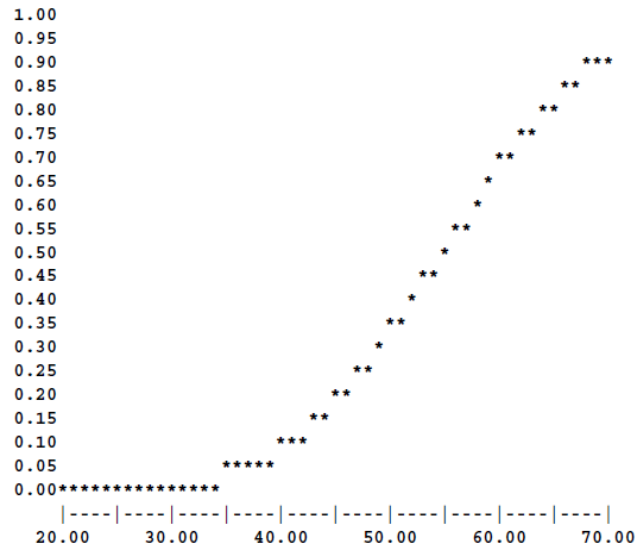
Fuzzy Value: temp
Linguistic Value: hot (*)



Universe of Discourse: From 0.00 to 100.00

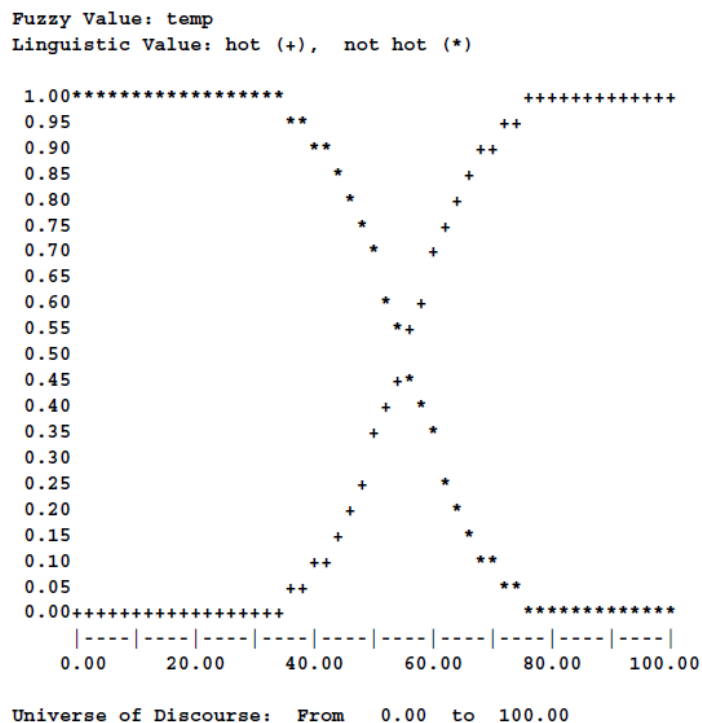
(plot-fuzzy-value t * 20 70 (create-fuzzy-value temp hot))

Fuzzy Value: temp
Linguistic Value: hot (*)



Universe of Discourse: From 0.00 to 100.00

- (deftemplate system
- (slot name)
- (slot t-outflow (type FUZZY-VALUE temp))
-)
- (assert (system (name sysA) (t-outflow not hot)))
- <Fact-1>
- (plot-fuzzy-value t "+"* nil nil
- (create-fuzzy-value temp hot)
- (get-fuzzy-slot 1 t-outflow)
-)



- **Command: is-defuzzify-value-valid**
- Syntax: (is-defuzzify-value-valid)
- **(defrule defuzzify-temperature**
- **?f <- (temperature ?)**
- **=>**
- **(bind ?temperature-value (moment-defuzzify ?f))**
- **(if (is-defuzzify-value-valid)**
- **then**
- **... do something with the defuzzified value**
- **else**
- **... perhaps use the maximum-defuzzify function**
- **))**