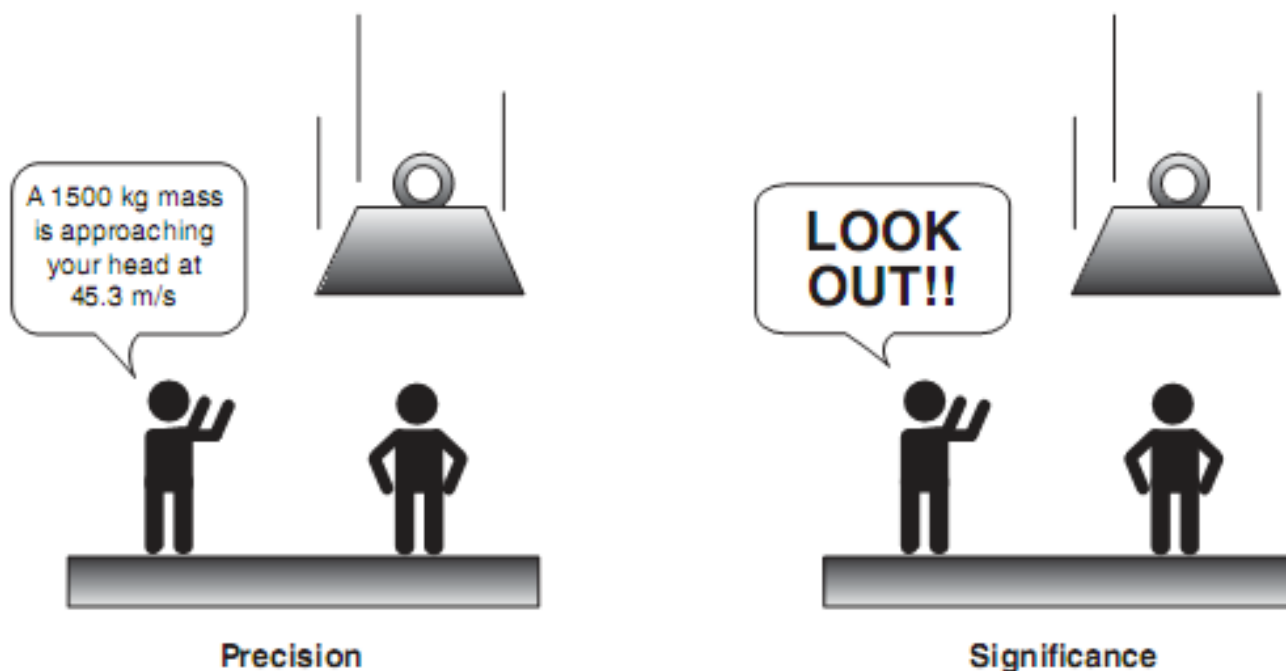


NEIZRAZITA LOGIKA

Uvod

Neizrazita logika omogućuje donošenje odluka na temelju nepreciznih informacija prisutnih u stvarnom svijetu.

Koliko je važno biti precizan kad je i procjena dobra?



Zašto koristiti neizrazitu logiku?

Razumljiva je:

Matematička pozadina je vrlo jednostavna, a proces zaključivanja je intuitivan i razumljiv.

Fleksibilna:

Sustav je jednostavno nadograditi novim svojstvima.

Tolerantna je prema nepreciznim podacima:

Neizrazito zaključivanje prihvaća i uključuje nepreciznost u proces zaključivanja.

Omogućuje modeliranje kontinuirane funkcije proizvoljne složenosti:

Moguće je izgraditi neizraziti sustav koji aproksimira bilo koji skup ulazno-izlaznih podataka.

Zašto koristiti neizrazitu logiku?

Neizraziti sustav zaključivanja moguće je izgraditi na temelju podataka dobivenih od eksperta:

Za razliku od neuronske mreže, proces zaključivanja je transparentan.

Omogućuje kombiniranje s klasičnim sustavima upravljanja:

Primjer: uključivanje klasičnih regulatora ovisno o radnoj točki.

Temeljena je na prirodnom jeziku:

Neizraziti sustav odlučivanja temeljen je na pravilima tipa “*Ako je temperatura visoka, smanji grijanje*”.

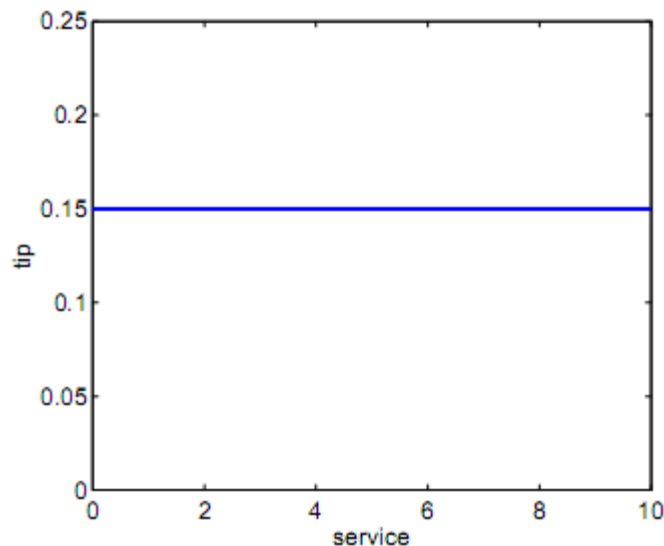
Neizrazitu logiku ne treba koristiti svuda i uvijek, naročito ako postoji jednostavno klasično rješenje.

Primjer:

Izračun napojnice (klasično rješenje)

Koliku napojnicu treba ostaviti u restoranu? (problem je postavljen prema običajima u SAD-u, gdje napojnica u prosjeku iznosi 15% računa)

Pretpostavimo da napojnica iznosi 15% iznosa računa:

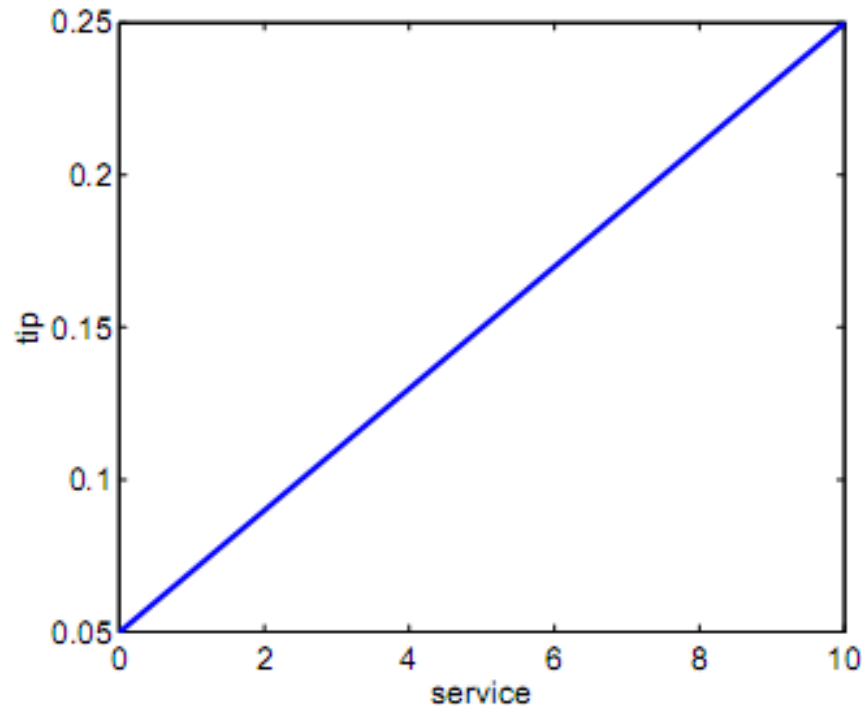


Primjer:

Izračun napojnice (klasično rješenje)

Smisleno je uključiti kvalitetu usluge u izraz:

$$\text{tip} = 0.20/10 * \text{service} + 0.05$$

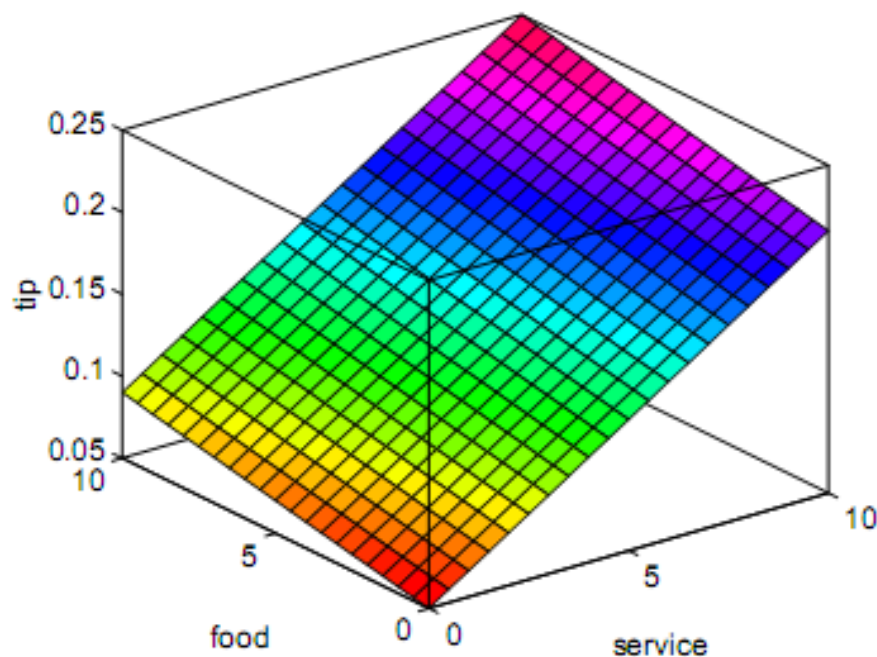


Primjer:

Izračun napojnice (klasično rješenje)

Kvalitetu hrane također treba uzeti u obzir. Neka kvaliteta hrane utječe na napojnicu s 20%, a kvaliteta usluge s 80%.

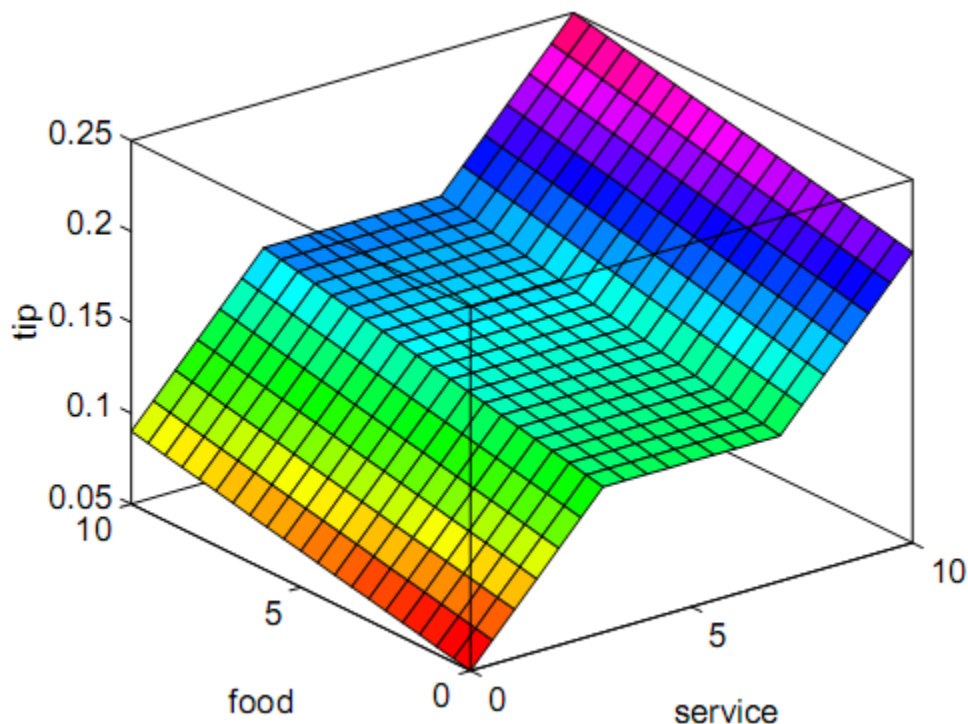
```
servRatio=0.8;  
tip=servRatio*(0.20/10*service+0.05) + ...  
      (1-servRatio)*(0.20/10*food+0.05);
```



Primjer:

Izračun napojnice (klasično rješenje)

Prethodna funkcija je linearna na cijelom području. Realnije bi bilo da u prosjeku dajemo 15% napojnice, a drugačije ako smo posebno zadovoljni ili nezadovoljni.



Primjer:

Izračun napojnice (klasično rješenje)

```
% Establish constants
lowTip=0.05; averTip=0.15; highTip=0.25;
tipRange=highTip-lowTip;
badService=0; okayService=3;
goodService=7; greatService=10;
serviceRange=greatService-badService;
badFood=0; greatFood=10;
foodRange=greatFood-badFood;

% If service is poor or food is rancid, tip is cheap
if service<okayService,
    tip=(( (averTip-lowTip)/(okayService-badService)) ...
        *service+lowTip)*servRatio + ...
        (1-servRatio)*(tipRange/foodRange*food+lowTip);

% If service is good, tip is average
elseif service<goodService,
    tip=averTip*servRatio + (1-servRatio)* ...
        (tipRange/foodRange*food+lowTip);

% If service is excellent or food is delicious, tip is generous
else,
    tip=(( (highTip-averTip)/ (greatService-goodService))* ...
        (service-goodService)+averTip)*servRatio + ...
        (1-servRatio)*(tipRange/foodRange*food+lowTip);

end
```

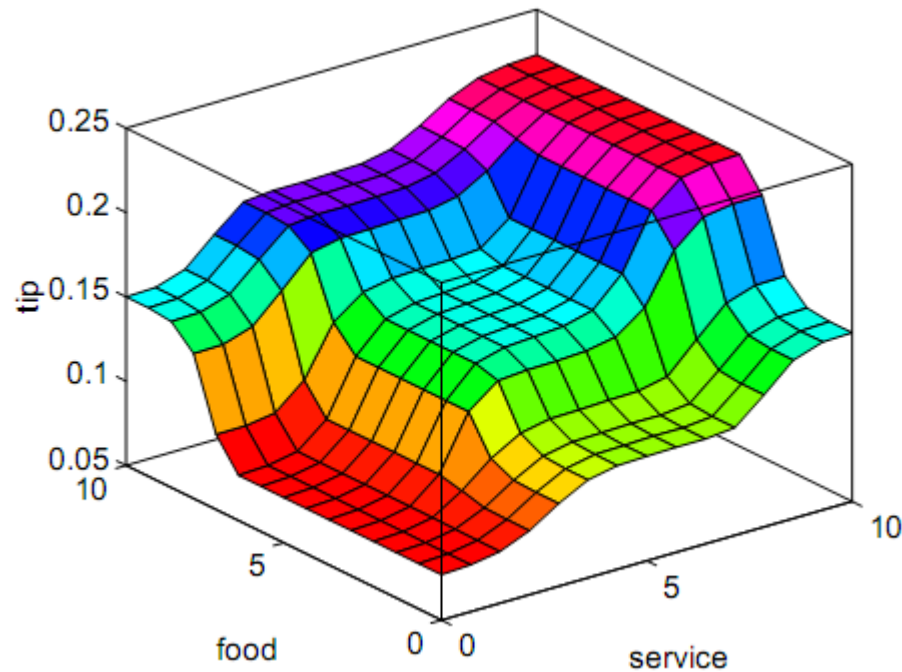
Neizraziti pristup

U neizrazitom sustavu dovoljno je definirati sljedeća pravila:

If service is poor or food is rancid, tip is cheap

If service is good, tip is average

If service is excellent or food is delicious, tip is generous



Neizraziti skup

Kod klasičnih skupova određeni element pripada skupu ili mu uopće ne pripada.

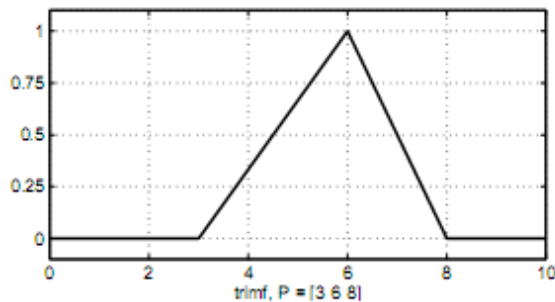
Primjer: Da li je petak dan vikenda?

Kod neizrazitih skupova definiramo pripadnost elementa skupu, koja je broj u intervalu $[0,1]$.

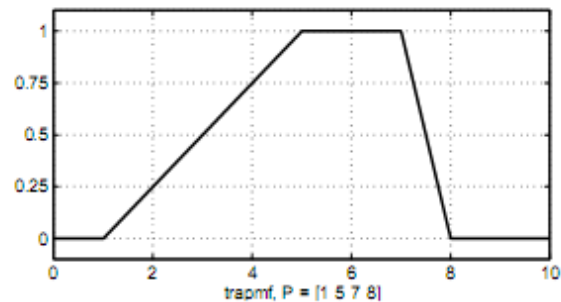
Primjeri: Skup visokih ljudi; Osjet temperature

Funkcija pripadnosti

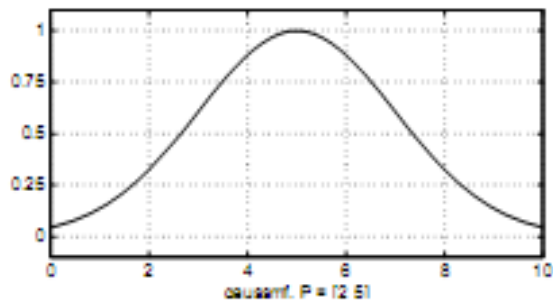
Definira pripadnost elementa skupu.



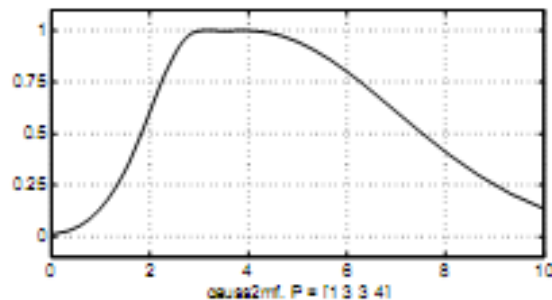
trimf



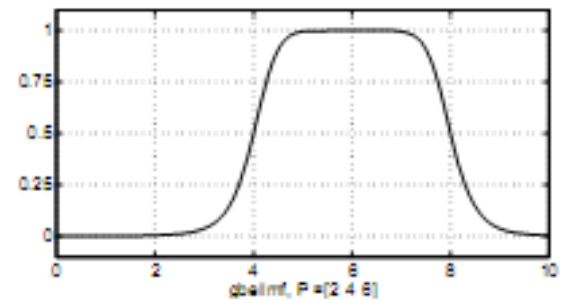
trapmf



gaussmf

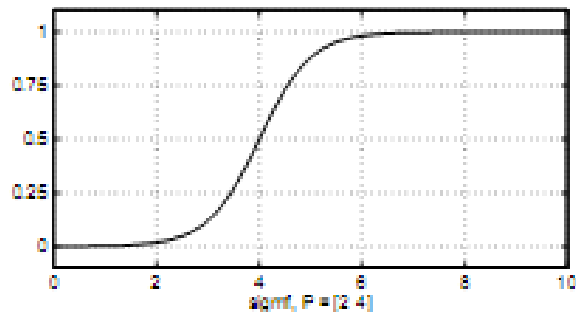


gauss2mf

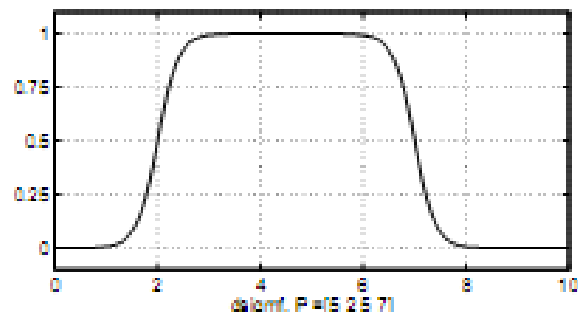


gbellmf

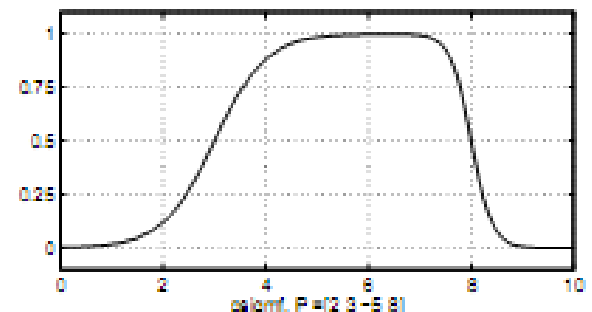
Funkcija pripadnosti



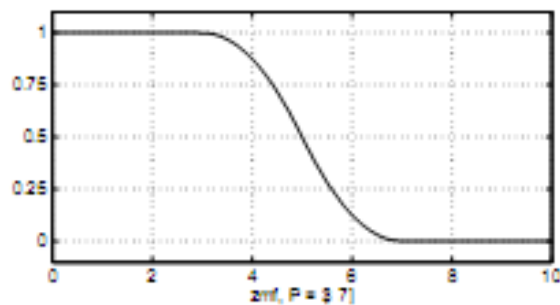
sigmf



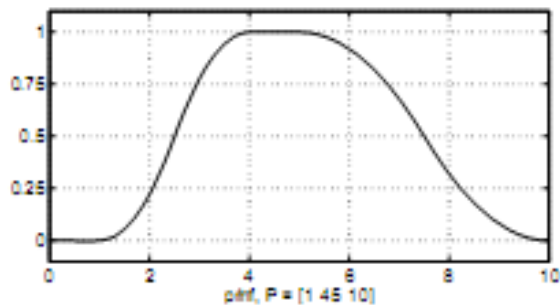
dsigmoid



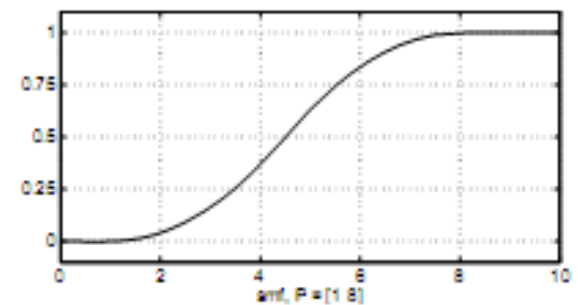
psigmoid



zmf



pimf

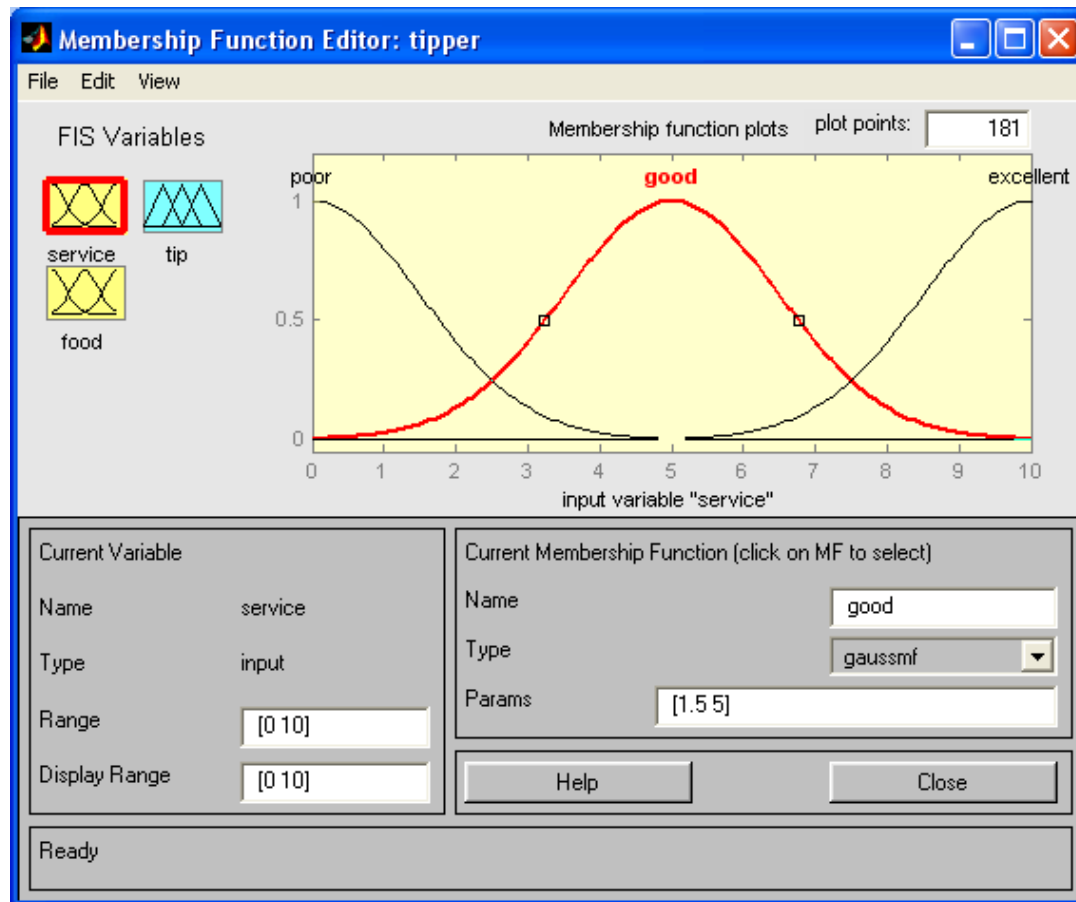


smf

Prikazane su najčešće korištene funkcije pripadnosti, a moguće je definirati i vlastitu. **Jedini uvjet je da poprima vrijednosti u intervalu $[0,1]$.**

Funkcija pripadnosti

Područja ulaznih i izlaznih varijabli definiraju se funkcijama pripadnosti:



Logički operatori

Standardni logički operatori I, ILI i NE prošireni su na cijeli interval $[0,1]$, time da na granicama prelaze u klasične Boole-ove operatore.

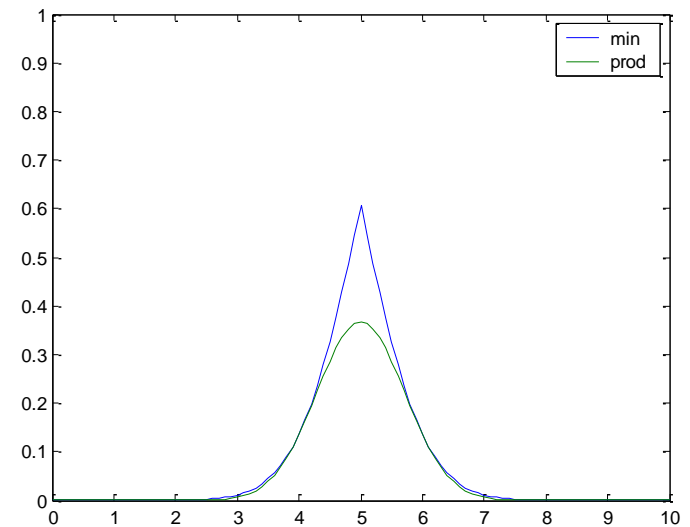
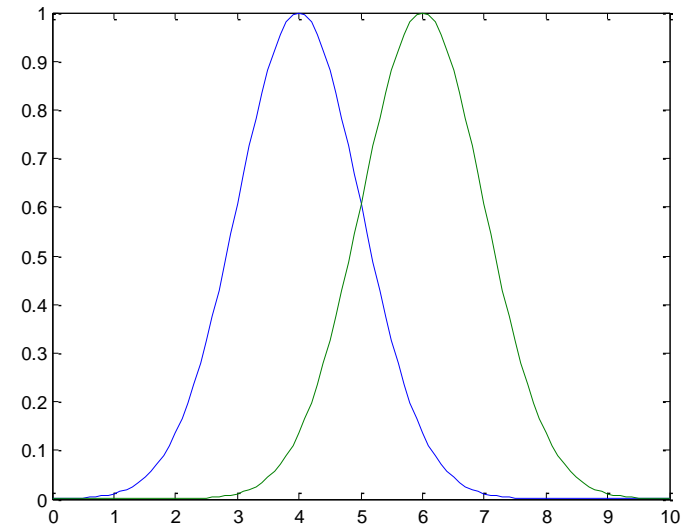
NE operator definiran je kao:

$$y = 1 - a$$

Logički operatori

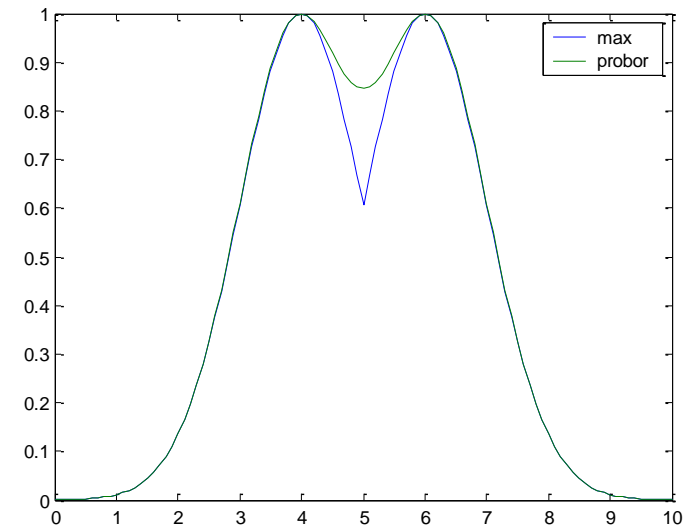
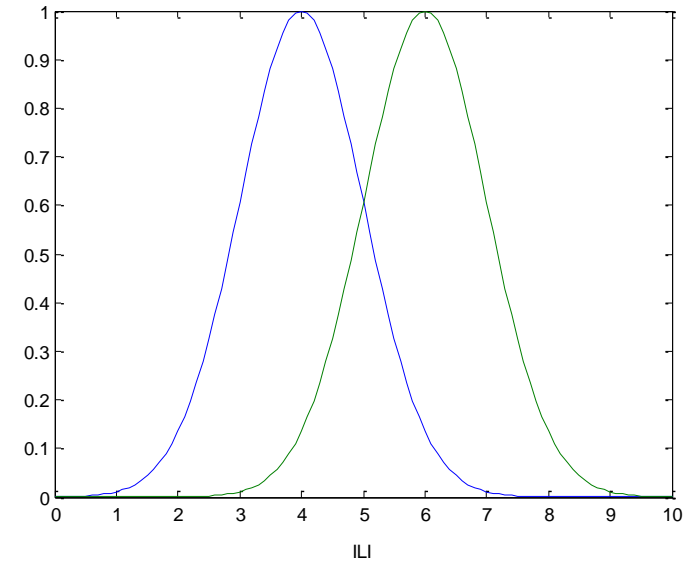
○ Za I operator koriste se funkcije:

- $y = \min(a,b)$
- $y = \text{prod}(a,b) = a * b$



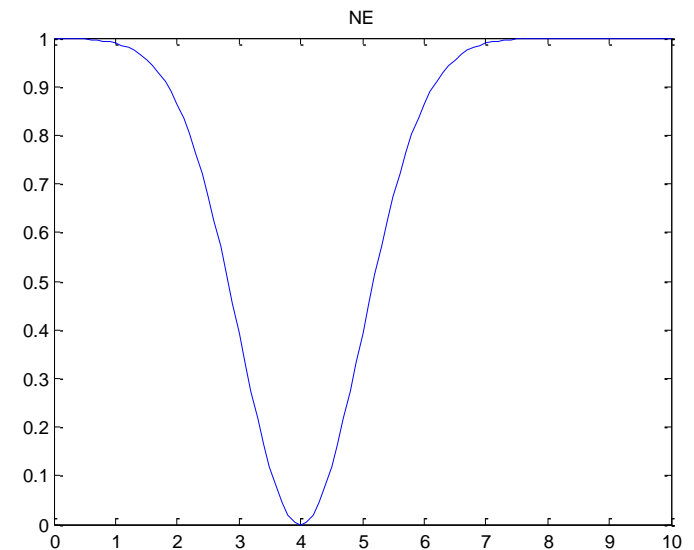
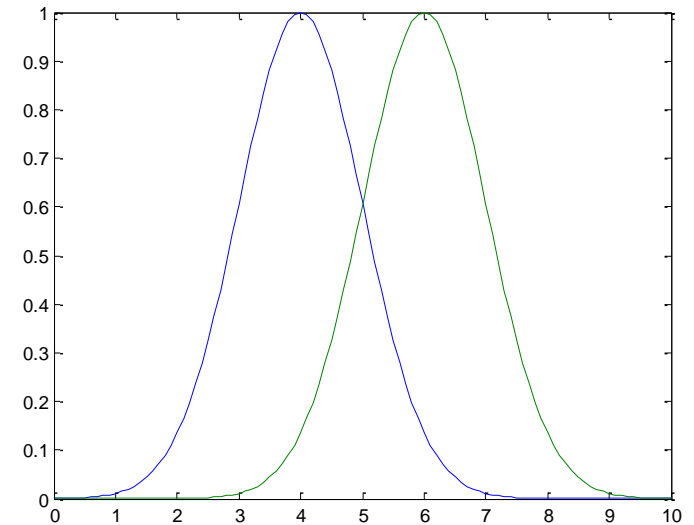
Logički operatori

- Za ILI operator koriste se funkcije:
 - $y = \max(a,b)$
 - $y = \text{probor}(a,b) = a + b - a * b$



Logički operatori

- NE operator definiran je kao:
 - $y = 1 - a$



Pravila

Pravila povezuju područja vrijednosti ulaznih i izlaznih varijabli.

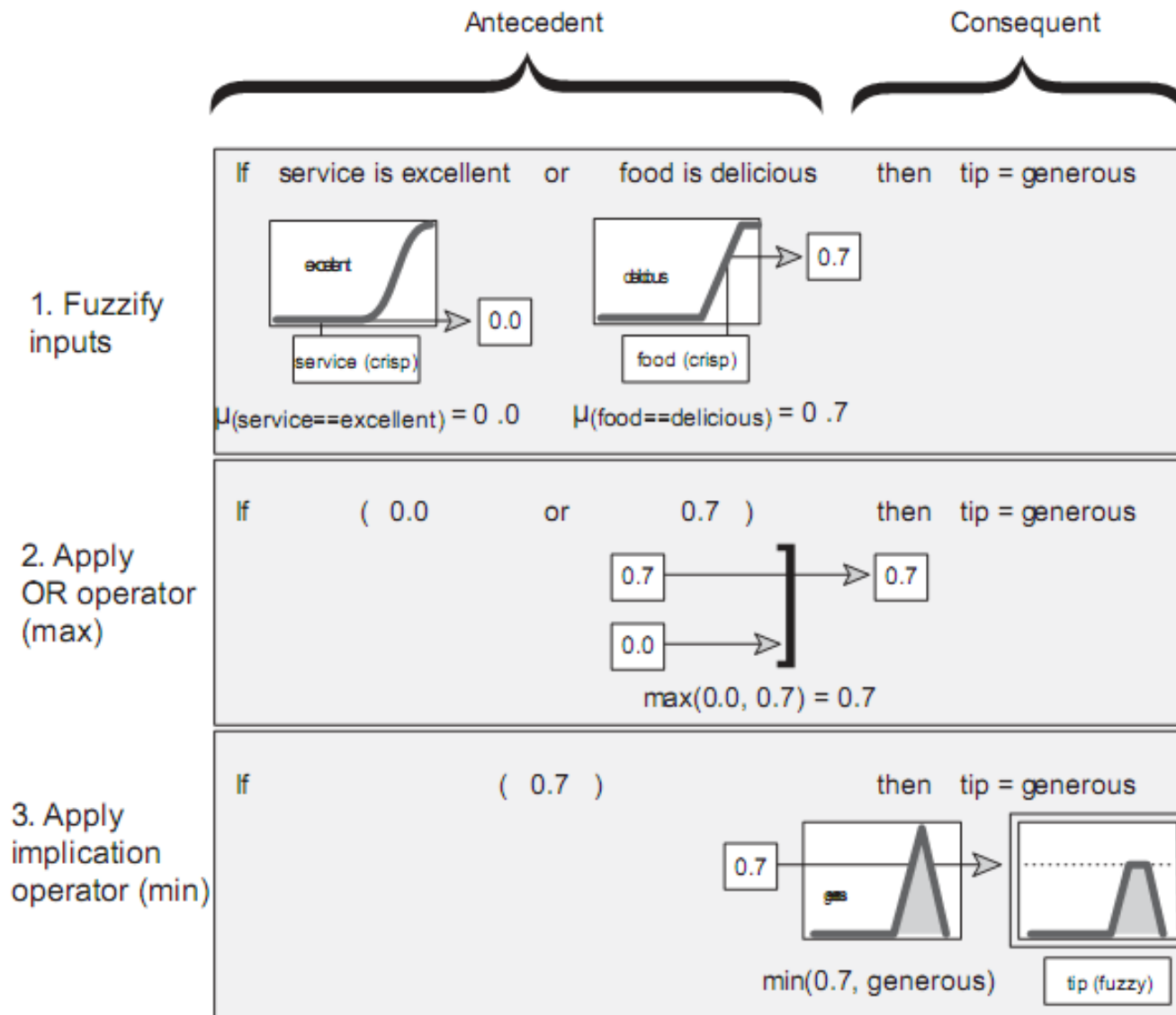
If service is excellent or food is delicious then tip = generous

Za svako pravilo određuje se pripadnost ulazne varijable odgovarajućem skupu (npr. “*service is excellent*”). **Taj postupak naziva se fuzifikacija ulaza.**

Zatim se primjenjuje pravilo i izlaz se prenosi na odgovarajuću izlaznu funkciju pripadnosti (primjena neizrazitih operatora).

Implikacijom se određuje utjecaj pravila na izlaznu varijablu.

Pravila



Združivanje (agregacija izlaza)

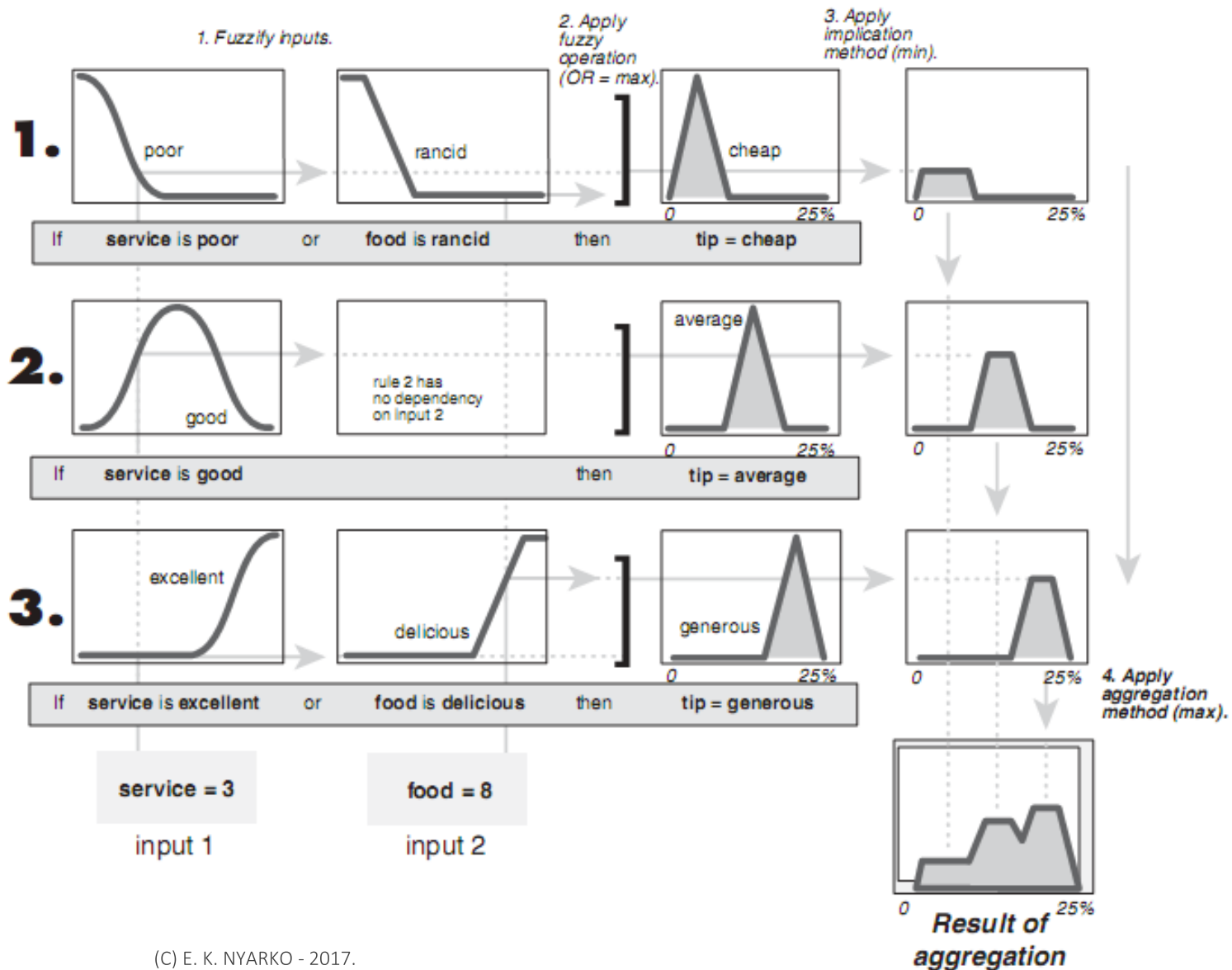
Izlaz neizrazitog sustava određen je svim pravilima, tako da je potrebno združiti rezultate svih pravila u po jedan izlazni neizraziti skup za svaku izlaznu varijablu.

Operator združivanja mora biti komutativan, tj. izlaz ne smije ovisiti o redosljedu evaluacije pravila.

Najčešće se koriste operatori *max*, *probor* i *sum*.

Postupak združivanja prikazan je sljedećom slikom (operator združivanja je *max*):

Združivanje (agregacija izlaza)



Defuzifikacija

Izlaz neizrazitog sustava je neizraziti skup. U većini primjena neizrazitih sustava (automatska regulacija, ekspertni sustavi) izlaz treba biti jedna numerička vrijednost.

Defuzifikacijom se značajke izlaznog neizrazitog skupa predstavljaju jednom numeričkom vrijednošću.

Najčešće se koriste sljedeće metode defuzifikacije (Mamdani neizraziti sustav):

- Centroid – centar mase

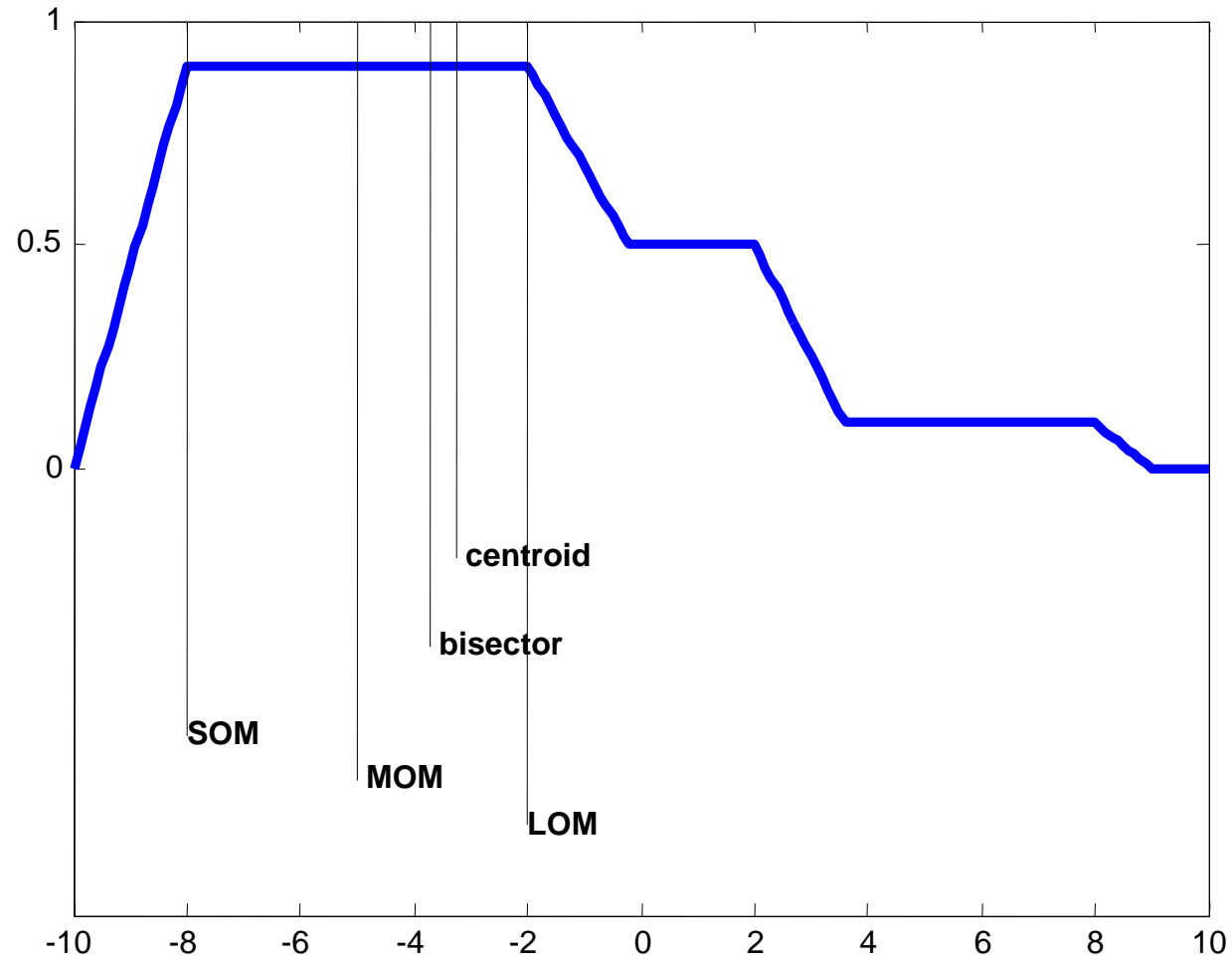
- Bisector – centar površine

- SOM – najmanja vrijednost za koju izlazna funkcija ima maksimum

- MOM – srednja vrijednost – || –

- LOM – najveća vrijednost – || –

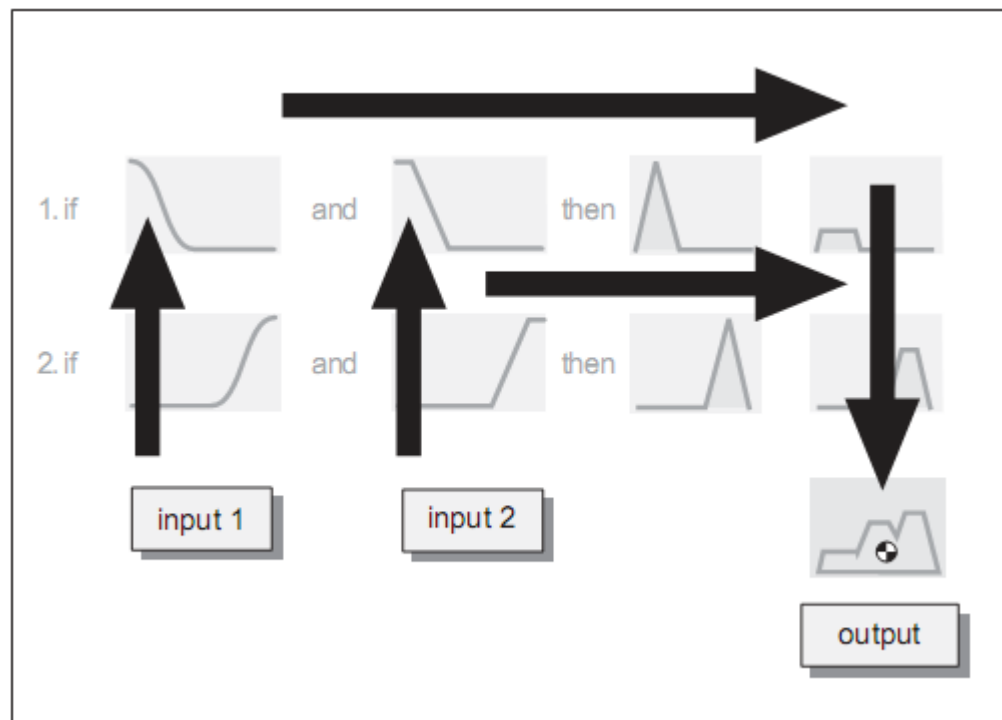
Defuzifikacija



Neizrazito zaključivanje

Neizrazito zaključivanje provodi se u sljedećim koracima:

1. Fuzzifikacija ulaza
2. Primjena neizrazitih operatora (pretpostavka pravila)
3. Implikacija (zaključak pravila)
4. Agregacija izlaza
5. Defuzifikacija



Sugeno neizraziti sustav

Prethodno je opisan Mamdani neizraziti sustav.

Susreće se i naziv Takagi-Sugeno-Kang sustav.

Razlika je u izlaznim funkcijama pripadnosti, koje su kod Sugeno sustava polinomi (u pravilu nultog ili prvog reda), tako da pravilo ima sljedeći oblik:

If Input1==x and Input2==y, then Output is $z=ax+by+c$

Za Sugeno sustav nultog reda vrijedi $a=b=0$

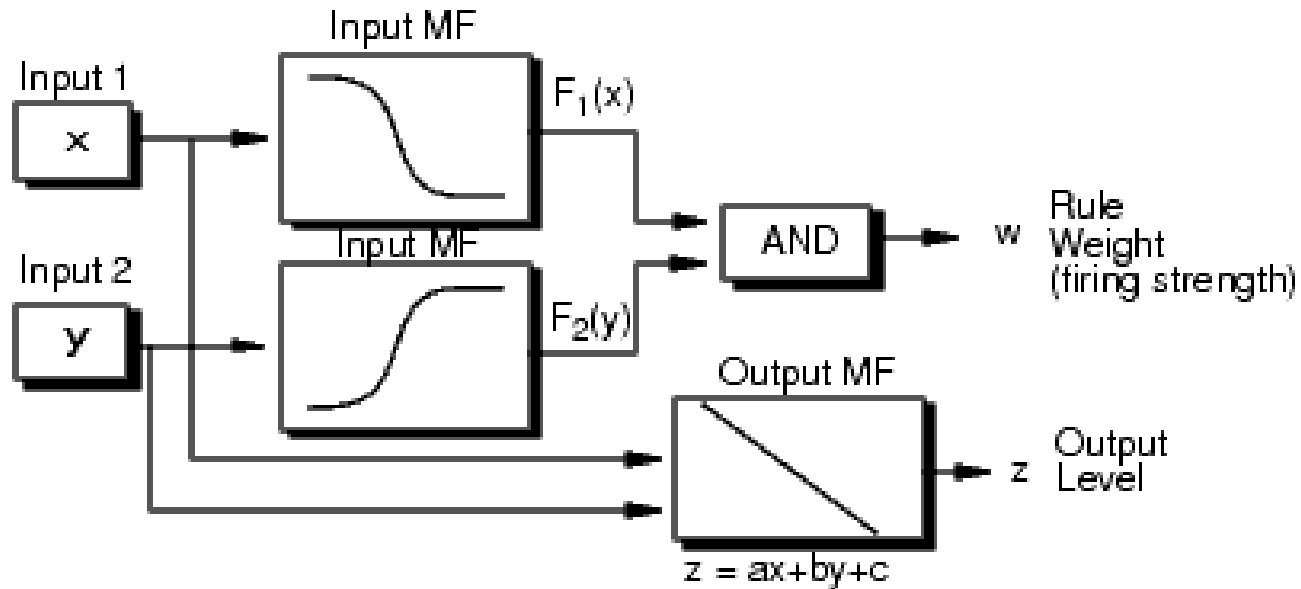
Vrijednost izlazne varijable je uvijek broj, a ne neizraziti skup i izračunava se kao težinska srednja vrijednost izlaza pravila:

$$I_{\text{zlaz}} = \frac{\sum_{i=1}^N w_i z_i}{\sum_{i=1}^N w_i}$$

w_i - izlaz i -tog pravila

z_i - vrijednost i -te funkcije pripadnosti

Sugeno neizraziti sustav



Sugeno neizraziti sustav

Kod Sugeno sustava prvog reda izlazi pravila određuju položaj i visinu izlazne vrijednosti, a kod sustava nultog reda samo visinu (položaj je konstantan).

Sugeno sustav ima nekoliko prednosti pred Mamdani sustavom:

- Računski je manje zahtjevan (defuzifikacija),

- Prikladan je za sustave automatskog upravljanja (npr. interpolacija izlaza više PID kontrolera),

- Omogućuje primjenu optimizacijskih i adaptacijskih tehnika (genetski algoritmi, neuronske mreže i dr.),

- Garantira neprekidnost izlazne plohe,

- Jednostavniji je za analizu.

Sugeno neizraziti sustav

