Vilnius universitetas
Matematikos ir informatikos fakultetas
Informatikos katedra

Skaitmeninis intelektas ir sprendimų priėmimas (neraiškioji logika)

doc. dr. Olga Kurasova
Olga.Kurasova@mii.vu.lt

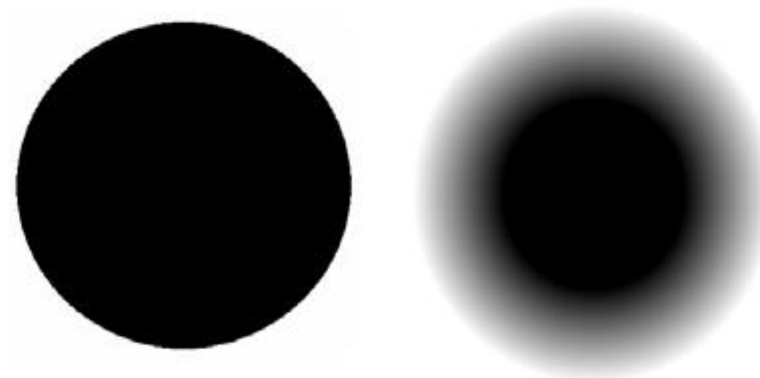
2017

Neraiškioji logika

- **Neraiškioji logika** (*fuzzy logic*) – tai logikos forma, kurioje loginių kintamųjų reikšmės gali būti bet koks realus skaičius tarp 0 ir 1.
- Priešingai nei yra **Būlio logikoje**, (*Boolean logic*, *Crisp logic*), kur loginių kintamųjų reikšmės yra sveikas skaičius 0 arba 1.
- Taigi, neraiškioje logikoje yra **„dalinė“ tiesa**, kas yra itin svarbu **sprendimų priėmime**, kai dažnai sunku pasakyti tik „juoda“ ar „balta“.
- Neraiškiosios logikos pradininkas – **Lofti Zadeh** (1965 metai).
- Ji dar kartais vadinama **miglotąja logika**.

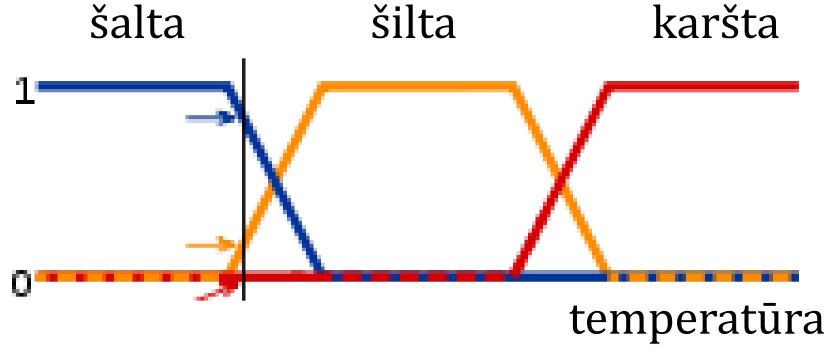
Neraiškioji logika

- Klasikinės logikos ir neraiškiosios logikos grafinis atvaizdavimas



Neraiškioji logika

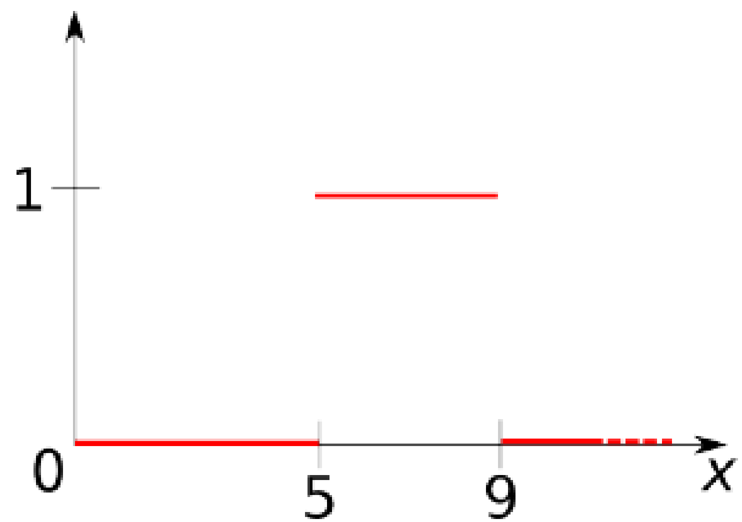
- Žmonės operuoja neraiškiais įvertinamais **kiekvieną dieną**. Pavyzdžiui, paklausus koks šiandien oras, galima sulaukti labai įvairių atsakymų.
- Todėl neraiškiąją logiką būtina perkelti ir į **dirbtinio intelekto** modelius.



Klasikinė logika

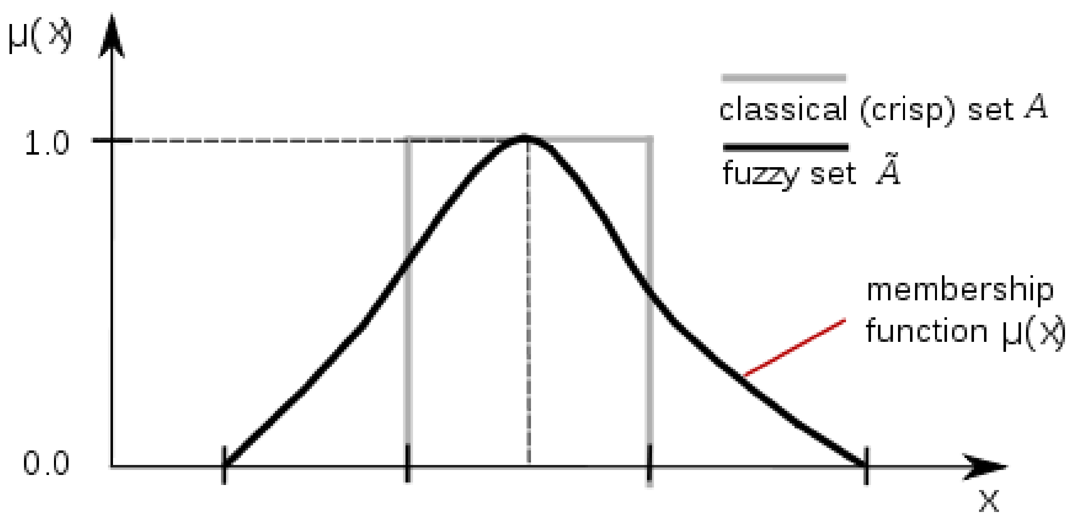
- Tarkime turime aibę X realių skaičių nuo 0 iki 10. Tarkime A = [5 9] yra aibės X poaibis, kurį sudaro realūs skaičiai nuo 5 iki 9.

- **Klasikinėje logikoje** aibė A grafiškai pavaizduota taip:



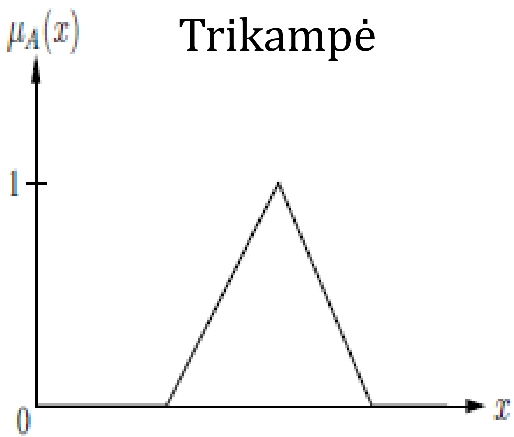
Narystės funkcija

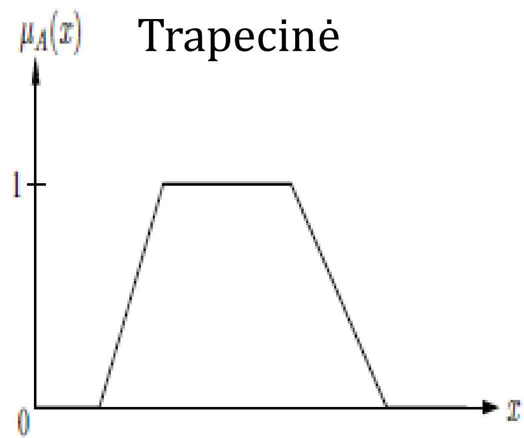
- Aibės X neraiškusis poaibis A yra funkcijos $\mu(x)$, priskiriančios kiekvienam aibės X elementui x **narystės laipsnį**, reikšmės
$$x \in X \mapsto \mu_A(x) \in [0, 1].$$
- Funkcija $\mu(x)$ vadinama **narystės** (*membership*) arba **priklausymo** funkcija.

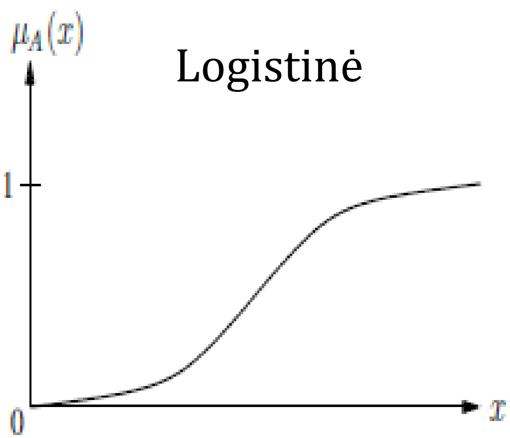


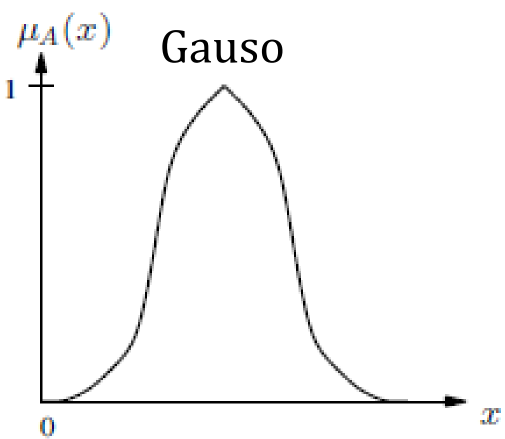
Narystės funkcijos tipai

- Galimos **įvairios narystės funkcijos**:



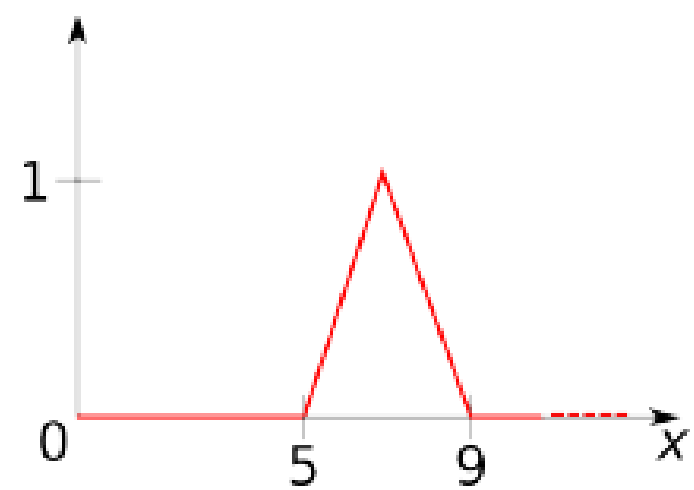






Neraiškiosios aibės pavyzdys

- Pavyzdžiui, tarkime X yra realių skaičių aibė tarp 1 ir 10. Realių skaičių **artimų** 7 neraiškiosios aibės aprašymas gali būti pateiktas tokiu grafiku:

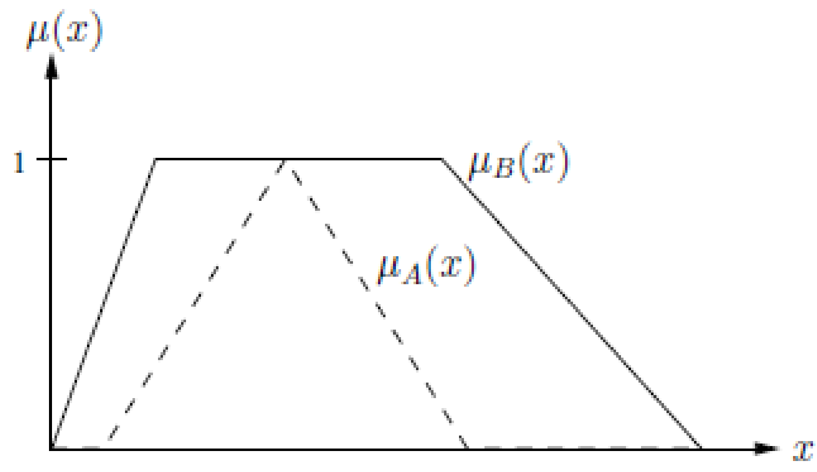


Neraiškiųjų aibių lygybė

- Tegu X yra apibrėžimo sritis arba universalioji aibė, o A ir B yra neraiškiosios aibės, apibrėžtos srityje X.
- Dvi **neraiškiosios aibės** A ir B **yra lygios** tada ir tik tada, kai sutampa jų apibrėžimo sritys ir jų narystės funkcijos yra lygios $\mu_A(x) = \mu_B(x)$ visiems $x \in X$, t. y. $A = B$.

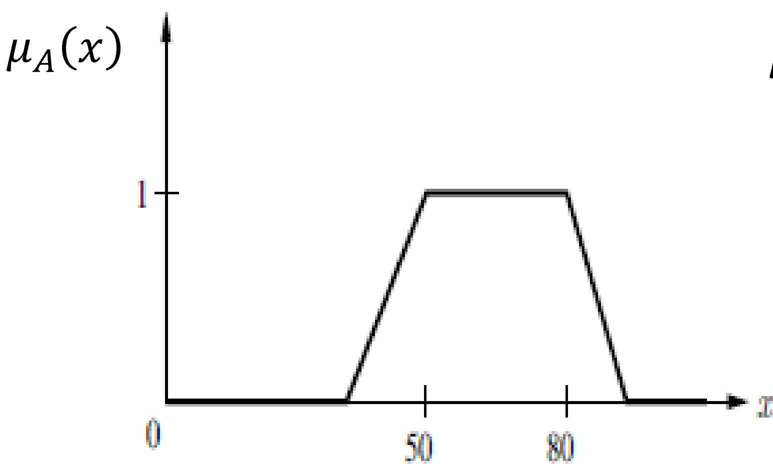
Neraiškiosios aibės poaibis

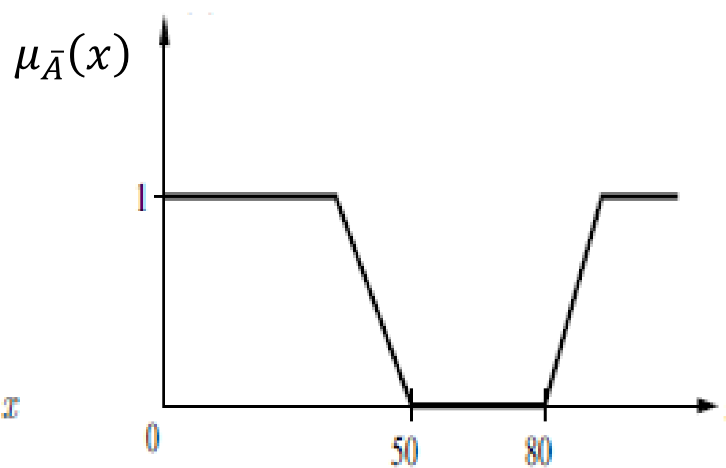
- Neraiškioji aibė A yra neraiškiosios aibės B **poaibis** tada ir tik tada jei $\mu_A(x) \leq \mu_B(x)$ visiems $x \in X$, t. y. $A \subset B$.



Neraiškiosios aibės papildinys

- Neraiškiosios aibės A **papildinys (viršaibis)** sudarytas iš visų aibės A elementų, tačiau narystės laipsniai skiriasi.
- Tegu $\bar{A}$ yra aibės A papildinys. Tuomet visiems $x \in X$: $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$.

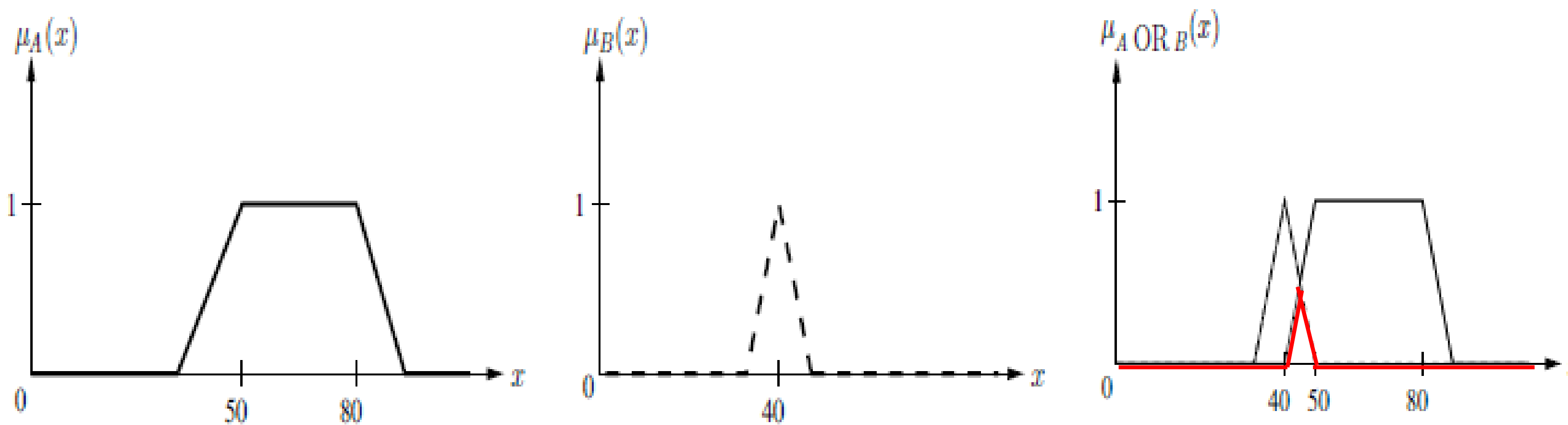




Neraiškiųjų aibių sankirta (AND)

- Neraiškiųjų aibių A ir B **sankirta** gali būti apibrėžta taip:

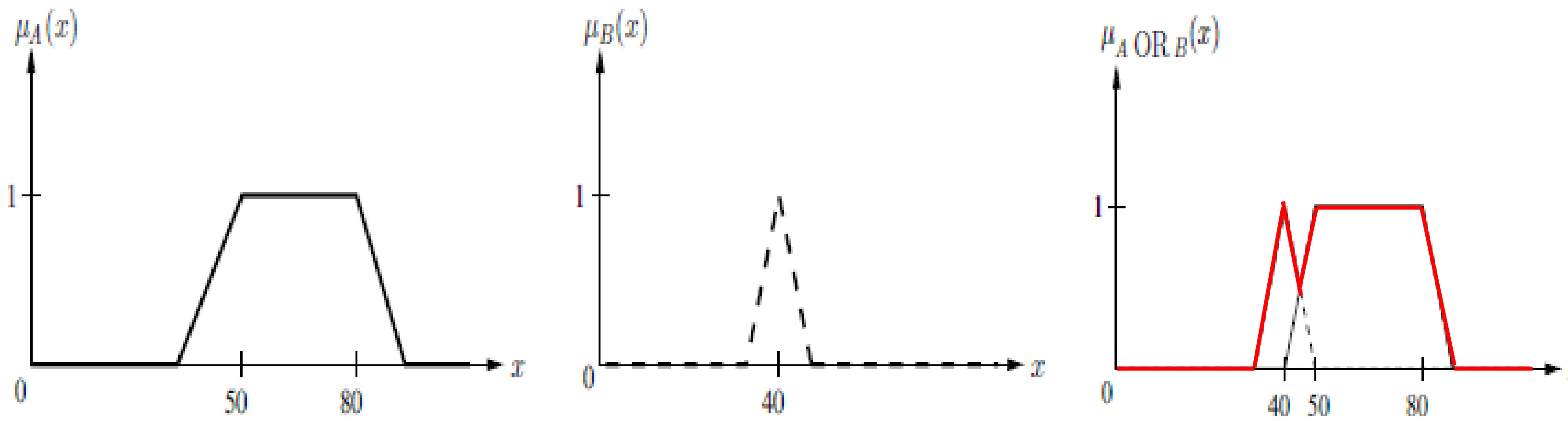
$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}, \forall x \in X$$



Neraiškiųjų aibių sąjunga (OR)

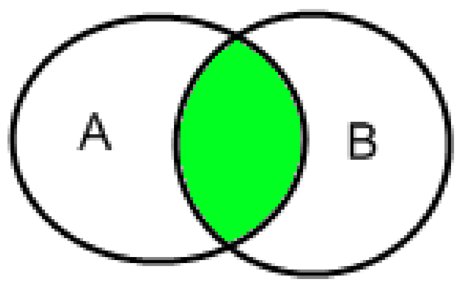
- Neraiškiųjų aibių A ir B **sąjunga** gali būti apibrėžta taip:

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

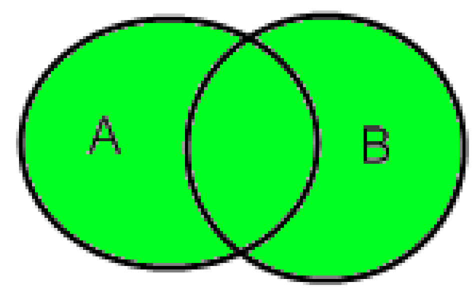


Klasikinių aibių operacijos

Sankirta $A \cap B$



Sąjunga $A \cup B$



Neraiškiųjų aibių pavyzdžiai

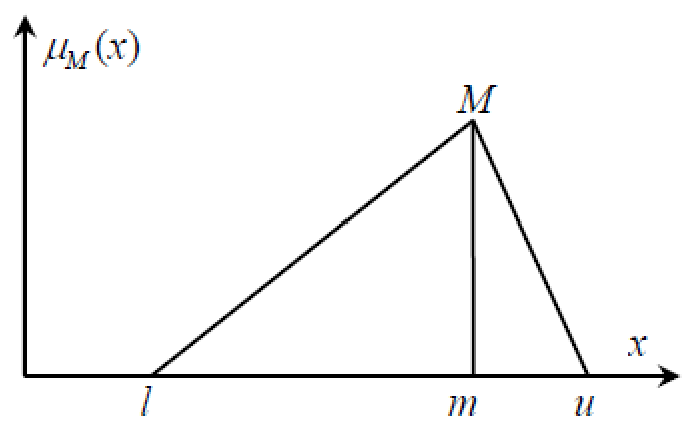
- Tarkime turime tris **neraiškiąsias aibes** *aukštas*, *geras_sportininkas* ir *geras_krepšininkas* ir
- $\mu_{aukštas}(Petras) = 0{,}8$ ir
  $\mu_{geras\_sportininkas}(Petras) = 0{,}9$
- $\mu_{aukštas}(Tomas) = 0{,}9$ ir
  $\mu_{geras\_sportininkas}(Tomas) = 0{,}6$
- $\mu_{geras\_krepšininkas}(Petras) = \min\{0{,}8, 0{,}9\}=0{,}8$ ir
  $\mu_{geras\_krepšininkas}(Tomas) = \min\{0{,}9, 0{,}6\} = 0{,}6$
- Petras **geresnis krepšininkas** už Tomą.

Neraiškieji skaičiai

- Naudojant trikampę narystės funkciją, **neraiškieji skaičiai** užrašomi triada

$$M = (l, m, u), l \leq m \leq u$$

- Čia m – tikėtiniausia reikšmė, l – apatinė riba, u – viršutinė riba.
- Taip apibrėžti skaičiai vadinami **trikampiais** neraiškiais skaičiais.



Neraiškiųjų skaičių sudėtis

Jei $l_1 > 0$ ir $l_1 > 0$, tai dviejų trikampių neraiškiųjų skaičių $M_1 = (l_1, m_1, u_1)$ ir $M_2 = (l_2, m_2, u_2)$ **suma**, **sandauga**, **daugyba iš skaliaro** ir **inversija** apibrėžiama taip:

- $M_1 \oplus M_2 = (l_1 + l_2, m_1 + m_2, u_1 + u_2)$,
- $M_1 \otimes M_2 = (l_1 l_2, m_1 m_2, u_1 u_2)$,
- $(\lambda, \lambda, \lambda) \otimes (l_1, m_1, u_1) = (\lambda l_1, \lambda m_1, \lambda u_1)$,
  $\lambda > 0, \lambda \in R$
- $(l_1, m_1, u_1)^{-1} = \left(\frac{1}{u_1}, \frac{1}{m_1}, \frac{1}{l_1}\right)$

Lingvistiniai kintamieji

- Įprastoje matematikos kintamieji gali įgyti skaitines reikšmes, neraiškiosios logikos taikymuose dažnai naudojamos neskaitinės kintamųjų reikšmės, palengvinančios taisyklių išraiškas. Tokie kintamieji vadinami **lingvistiniais kintamaisiais**.
- Pavyzdžiui, terminai „aukštas“ arba „žemas“, „toli“ arba „arti“ gali būti **lingvistiniai terminai**, apibūdinantys tokius dydžius, kaip slėgis ir atstumas.
- Pasitelkiant neraiškiąją aibių teoriją, lingvistinę, kasdieninės kalbos **informaciją galima perteikti kompiuteriui**, o tai leidžia lengviau ir suprantamai formuoti įvairias valdymo ar kitas užduotis.

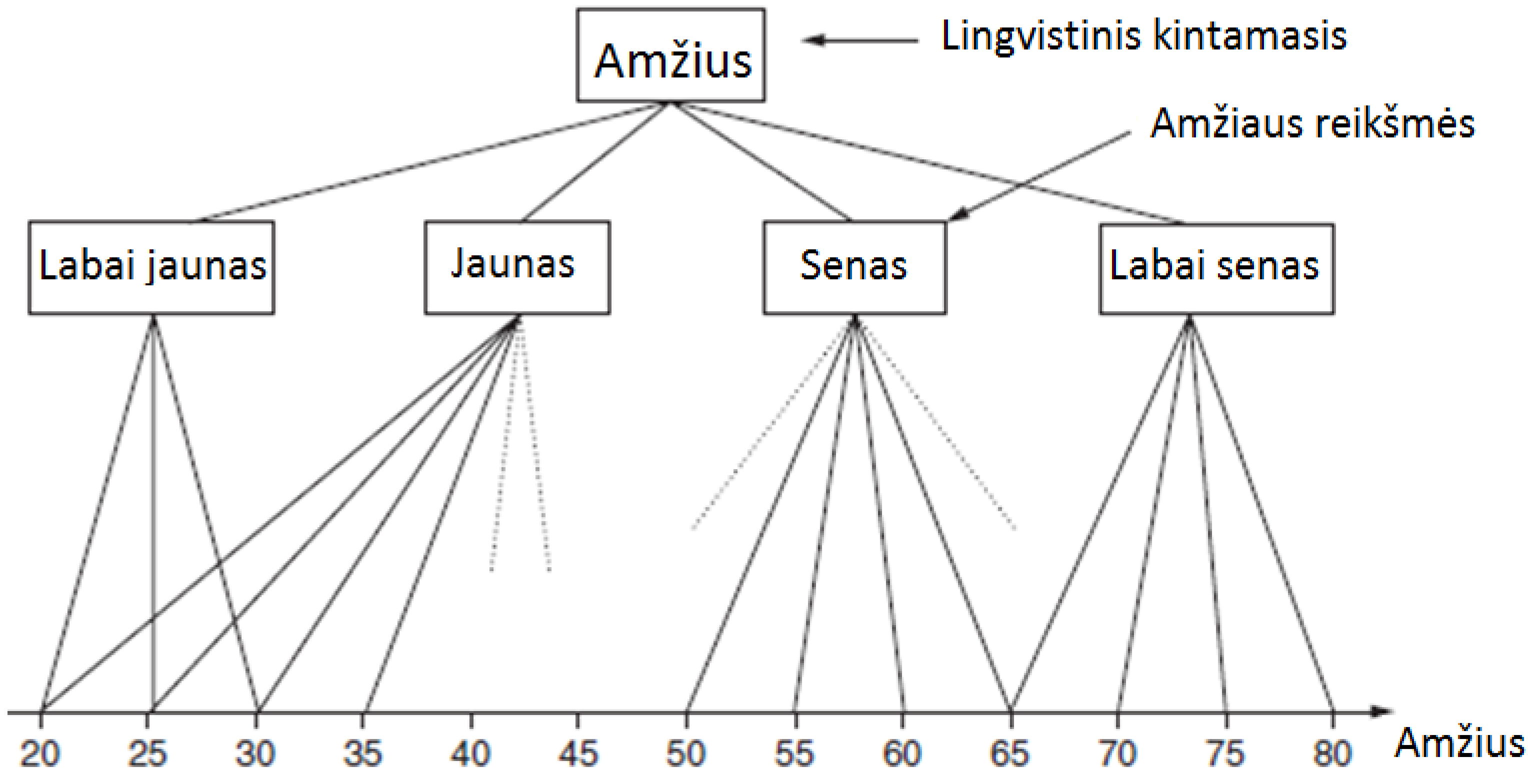
Lingvistiniai kintamieji

Lingvistiniai kintamieji gali būti skirstomi į tokias kategorijas:

- **Kiekybiniai** kintamieji, pvz., visi, dauguma, daug, niekas ir kt.
- **Dažnumo** kintamieji, pvz., kartais, dažnai, visada, retkarčiais ir kt.
- **Tikėtinumo** kintamieji, pvz., galima, žinoma, tikėtina ir kt.

Lingvistinio kintamojo pavyzdys

- **Lingvistinis kintamasis** – amžius



Neraiškiosios taisyklės

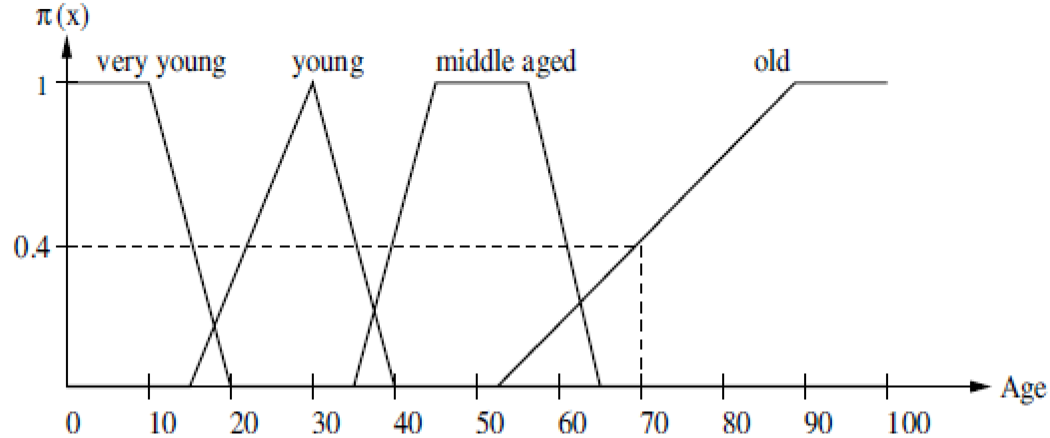
- Bendroji neraiškiųjų taisyklių forma:

  If sąlyga(-os) **then** pasekmė(-ės)

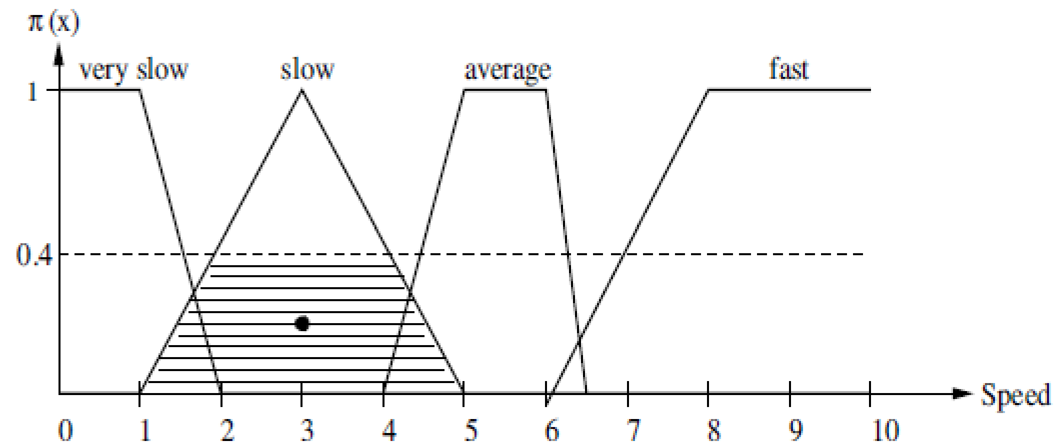
- Neraiškiosiose taisyklėse sąlygą ir pasekmę sudaro lingvistiniai kintamieji.
- Pavyzdžiui,
- **If** *amžius* yra *senas* **then** *greitis* yra *mažas*

Neraiškiųjų taisyklių interpretavimas

Koks greitis, jei amžius = 70?

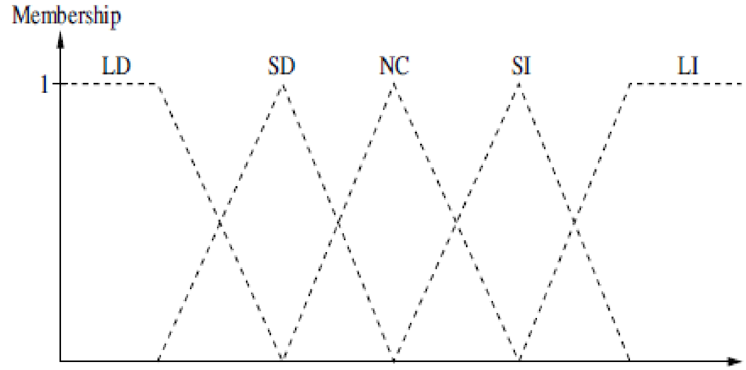


Atsakymas: greitis = 3

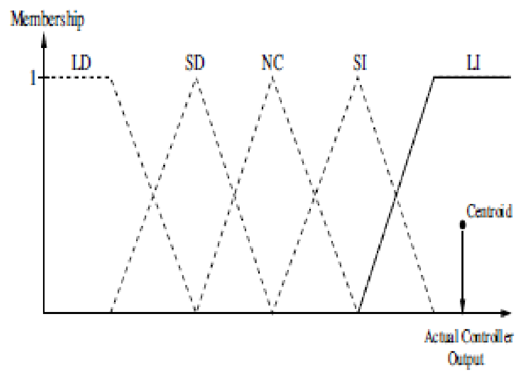


Kiekybinis rezultatas

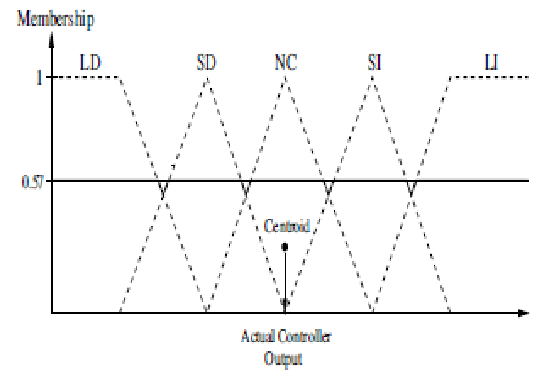
- Operuojant neraiškiąją logika ir neraiškiosiomis taisyklėmis, realiuose uždaviniuose, pvz., valdymo, sprendimų priėmimo, dažniausiai reikia kiekybinio atsakymo, t. y. būtina **eliminuoti neraiškumą** (*defuzzification*).
- Yra taikomi įvairūs **neraiškumo eliminavimo** metodai: Max-min metodas, vidurkinimo metodas, kvadratinės sumos metodas, gravitacijos centro metodas.



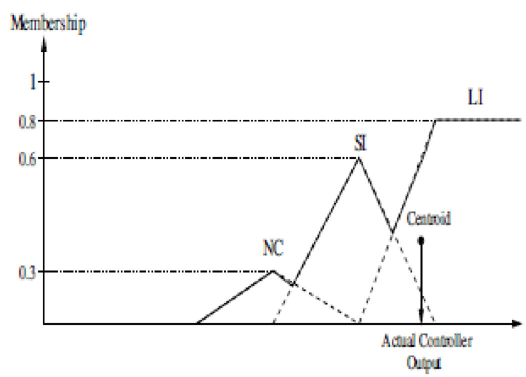
(a) Output Membership Functions



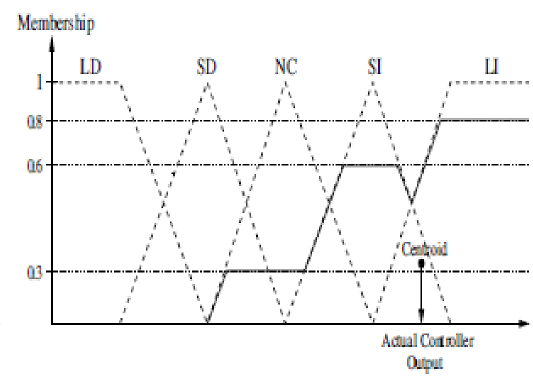
(b) Max-Min Method



(c) Averaging Method



(d) Root-Sum-Square Method



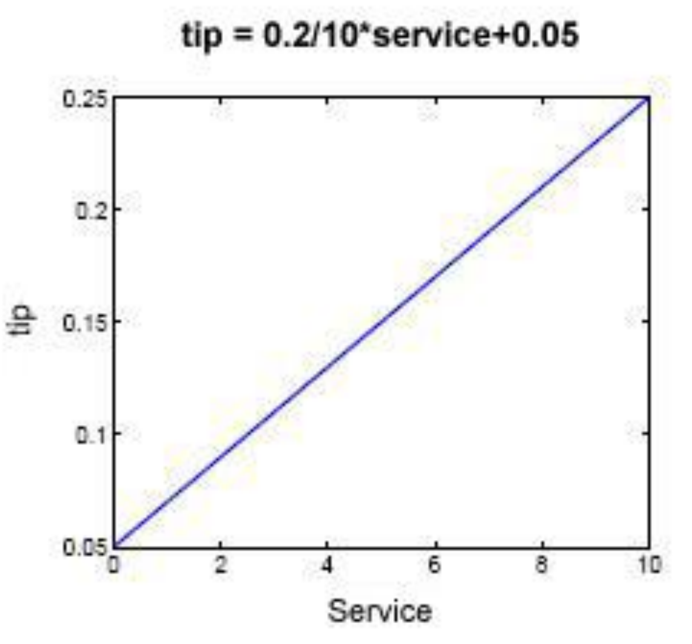
(e) Clipped Center of Gravity Method

Pavyzdys

- Tarkime restorano paslaugų kokybė vertinama balais nuo 0 iki 10.
- **Klausimas – kiek arbatpinigių verta palikti?**

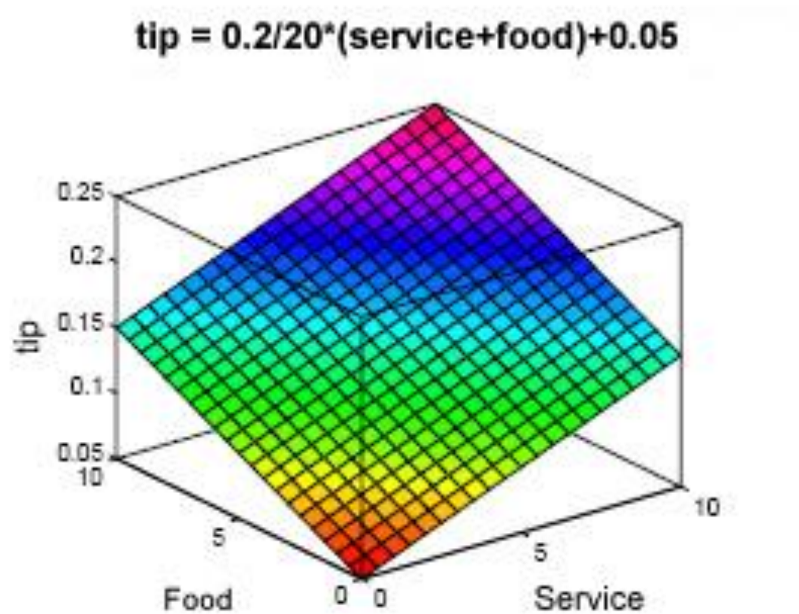
Klasikinis būdas

- **Arbatpinigiai**:
  - 5 % – jei paslaugos yra prastos,
  - 25 % – jei paslaugos puikios.



Klasikinis būdas

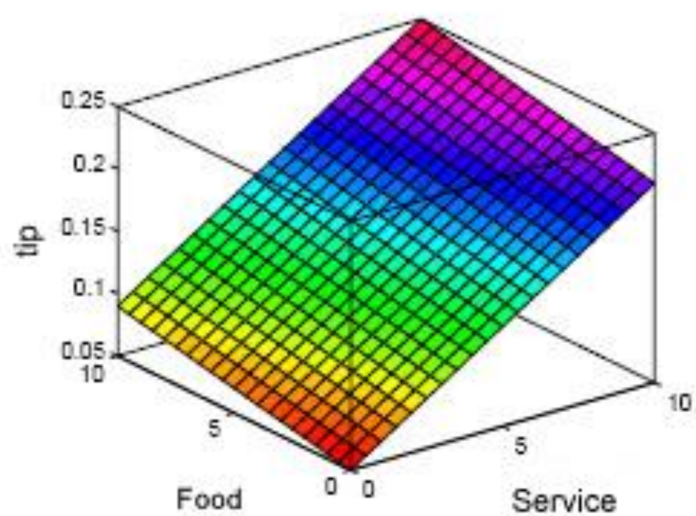
- Tarkime arbatpinigiai priklauso ne tik nuo paslaugų (aptarnavimo) kokybės, bet ir nuo **maisto kokybės**, kuri taip pat vertinamas nuo 0 iki 10 balų.



Klasikinis būdas

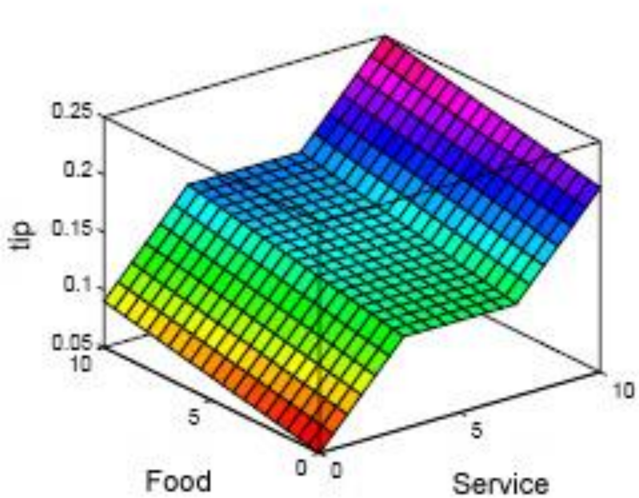
- Gali būti, kad **paslaugų kokybė** yra svarbesnė nei **maisto kokybė** ir atvirkščiai.
- Tarkime **paslaugų kokybė** daro įtaką 80 %, o **maisto** – 20 %.

```
servRatio=0.8;
tip=servRatio*(0.2/10*service+0.05)+
    + (1-servRatio)*(0.2/10*food+0.05);
```



Klasikinis būdas

- Tarkime, bendruoju atveju mes norime duoti 15 % arbatpinigių. Juos padidinsime arba sumažinsime tik tuomet, jei maistas bus **ypatingai geras** ar **blogas**.




```
servRatio=0.8;

if service<3,
   tip=((0.1/3)*service+0.05)*servRatio+
         + (1-servRatio)*(0.2/10*food+0.05);
elseif service<7,
   tip=(0.15)*servRatio+
         + (1-servRatio)*(0.2/10*food+0.05);
else
   tip=((0.1/3)*(service-7)+0.15)*servRatio+
         + (1-servRatio)*(0.2/10*food+0.05);

end
```

Klasikinis būdas

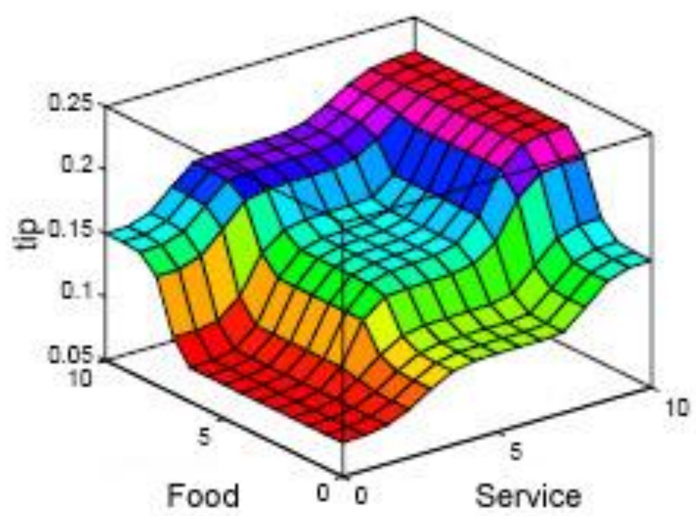
Trūkumai:

- Sunku modifikuoti, atsiradus kitoms sąlygoms.
- Teks perrašyti taisykles atsiradus naujiems kintamiesiems, pvz., norint vertinti restorano dizainą ar švarą.

Neraiškusis būdas

Užrašome taisykles:

1. **If** *service* is *poor* or *food* is *bad* **then** *tip* is *cheap*
2. **If** *service* is *good* **then** *tip* is *average*
3. **If** *service* is *excellent* or *food* is *delicious* **then** *tip* is *generous*



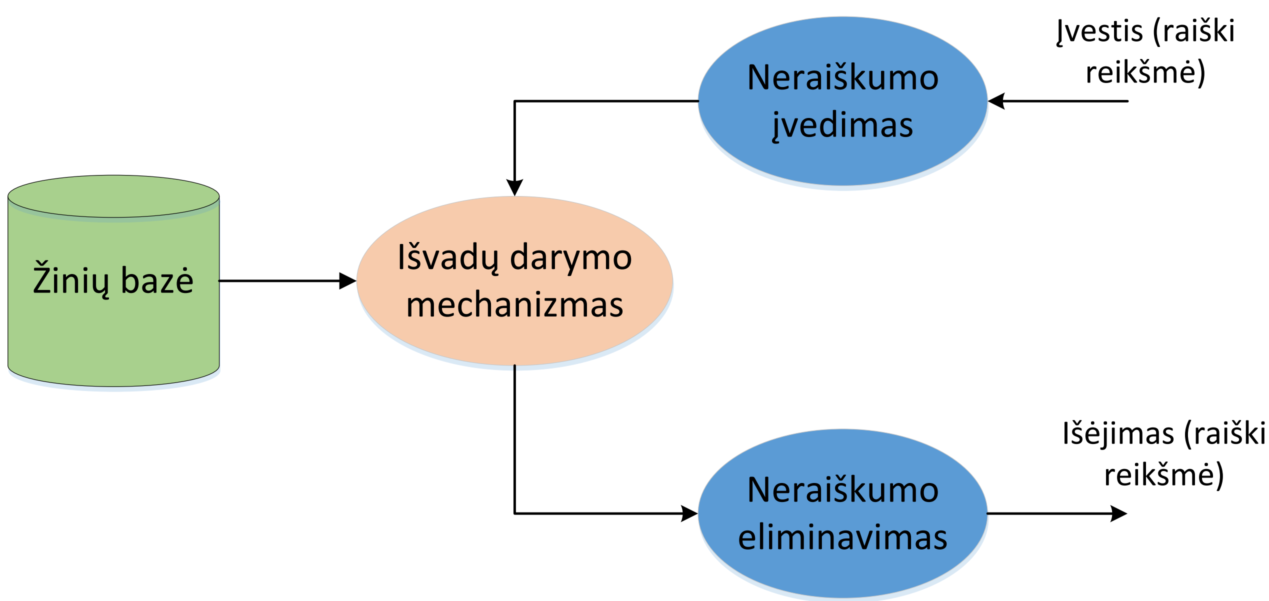
Neraiškusis būdas

Privalumai:

- Taisyklės panašios į žmogaus mąstymą
- Atsiradus naujiems kintamiesiems ar sąlygoms, reiks tik pakoreguoti esamas ar parašyti naujas taisykles.
- Nereikia perrašyti algoritmo.

Neraiškiųjų išvadų darymo sistema

- **Neraiškiųjų išvadų darymo sistema** (Fuzzy Inference System, FIS) – tai būdas atvaizduoti įėjimo aibę į išėjimų aibę naudojant neraiškiąją logiką.
- FIS dažnai naudojama **sprendimams priimti**.



Neraiškiųjų išvadų darymo sistema

- **Neraiškumo įvedimo** (*fuzzification*) modulis transformuoja raiškias reikšmes (skaičius) į neraiškiąsias aibes.
- **Žinių bazė** (*knowledge base*) saugo ekspertų pateiktas IF-THEN taisykles.
- **Išvadų darymo mechanizmas** (*inference engine*) simuliuoja žmogaus sprendimų priėmimo procesą darant neraiškias išvadas pagal įėjimus ir IF-THEN taisykles.
- **Neraiškumo eliminavimo** (*defuzzification*) modulis transformuoja išvadų darymo mechanizmo gautą neraiškiąją aibę, gautą į raiškias reikšmes (skaičius).

Neraiškiųjų išvadų darymo sistemos

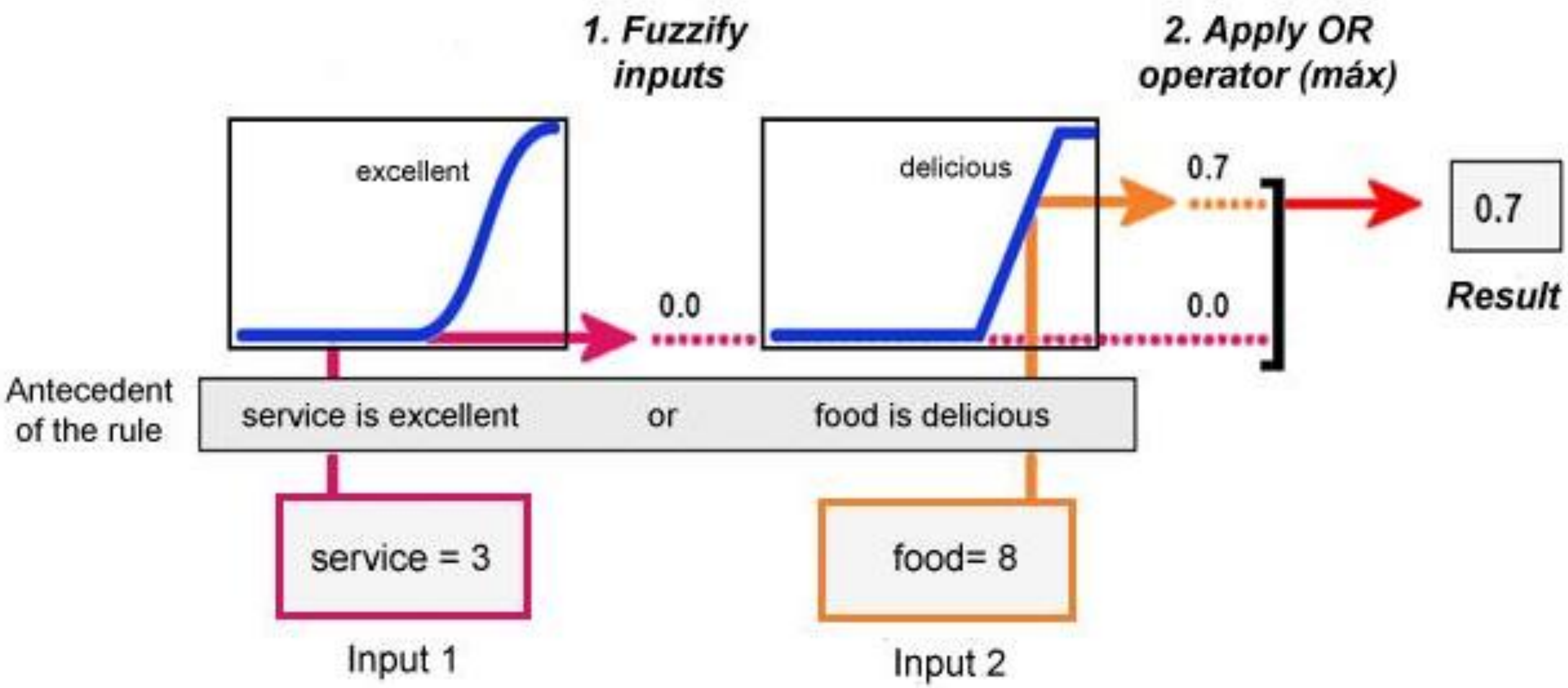
- Populiariausi neraiškiųjų išvadų darymo sistemų **tipai**:
  - Mamdani,
  - Takagi-Sugeno.

Mamdani metodas

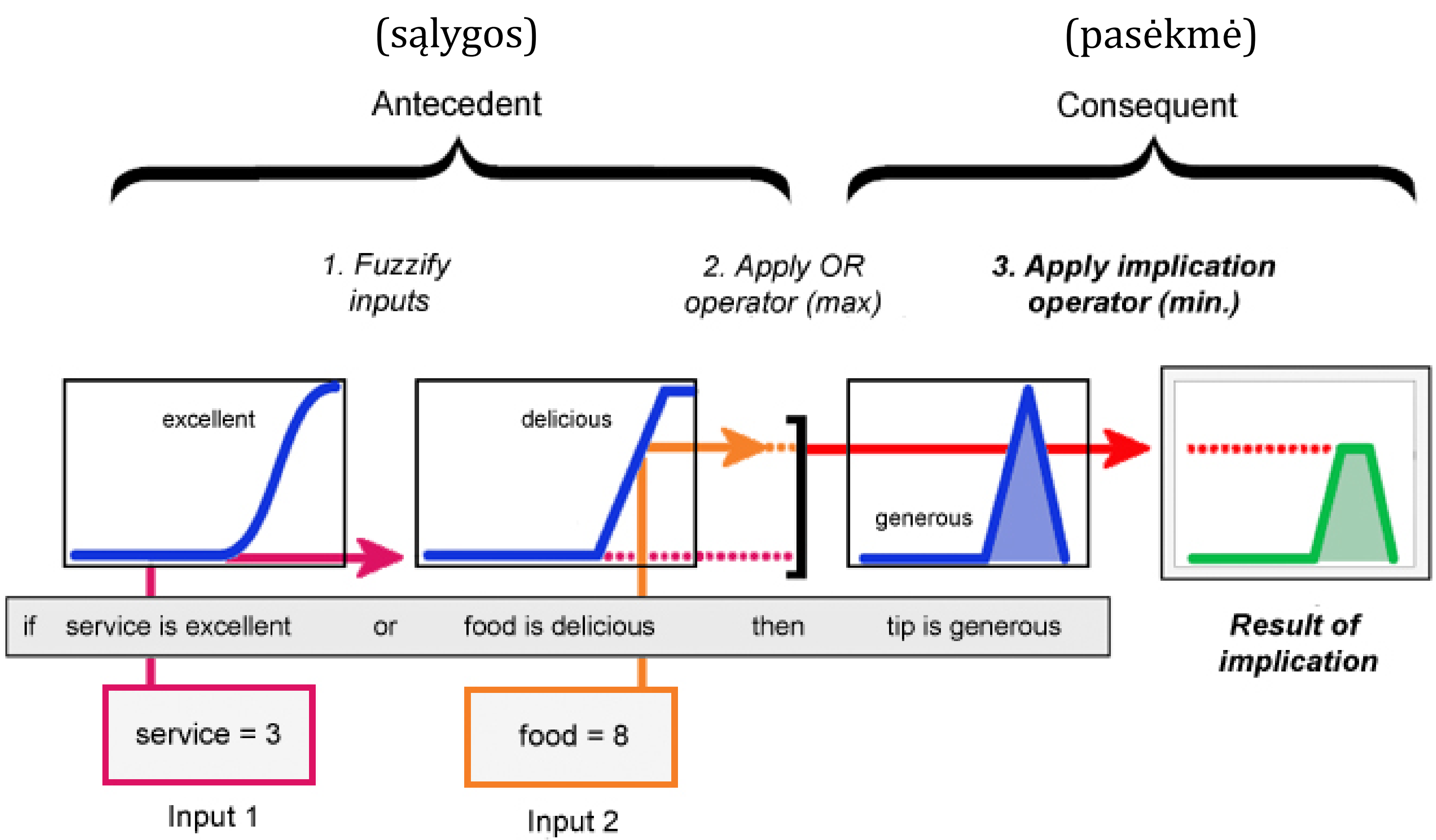
Pagrindiniai žingsniai:

1. Įvertinamos kiekvienos **taisyklės sąlygos**.
2. Gaunamos kiekvienos **taisyklės išvados**.
3. **Apjungiamos** išvados.
4. Atliekamas **neraiškumo eliminavimas**.
5. Daromos **išvados**.

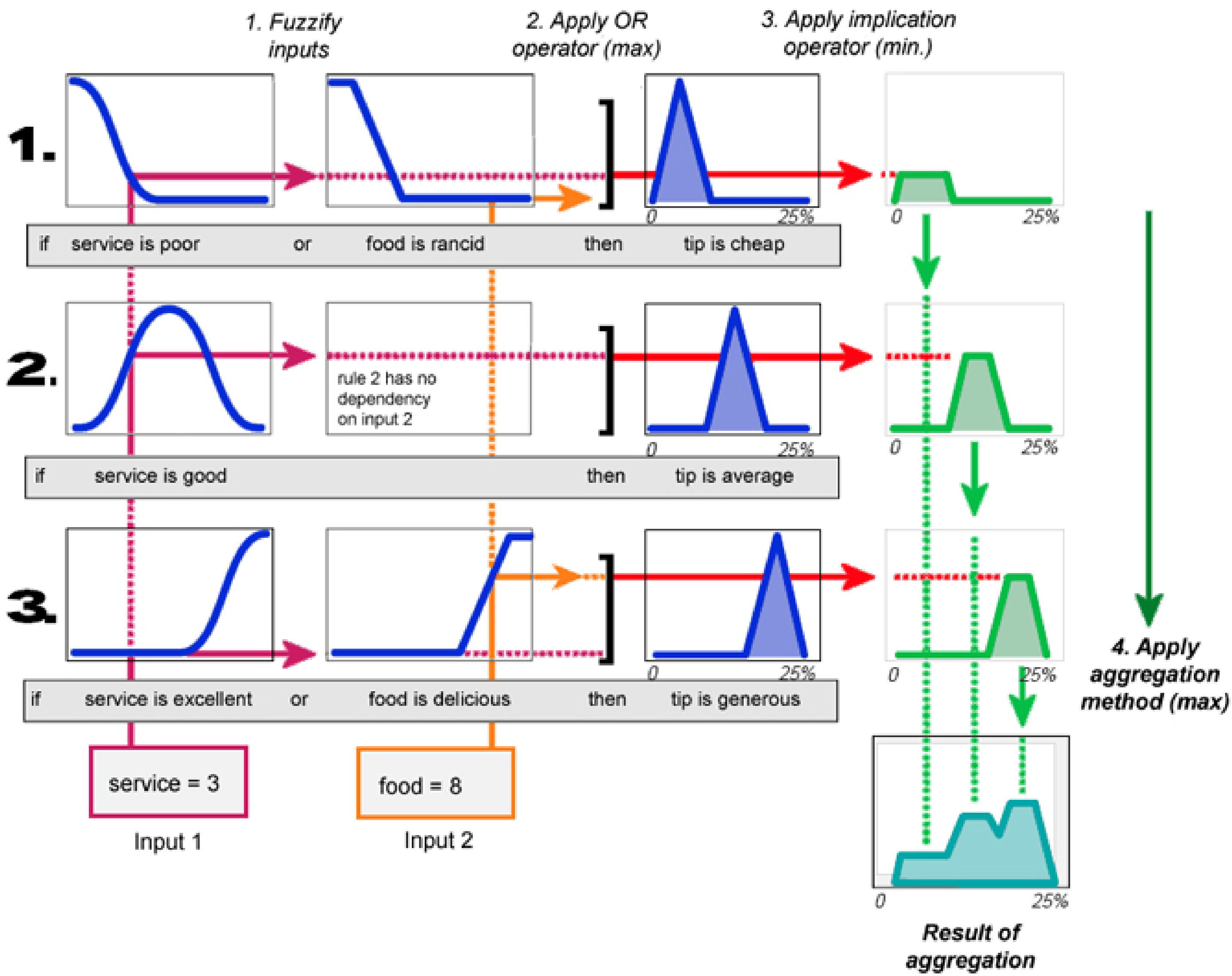
Pavyzdys: kiekvienos taisyklės sąlygų įvertinimas



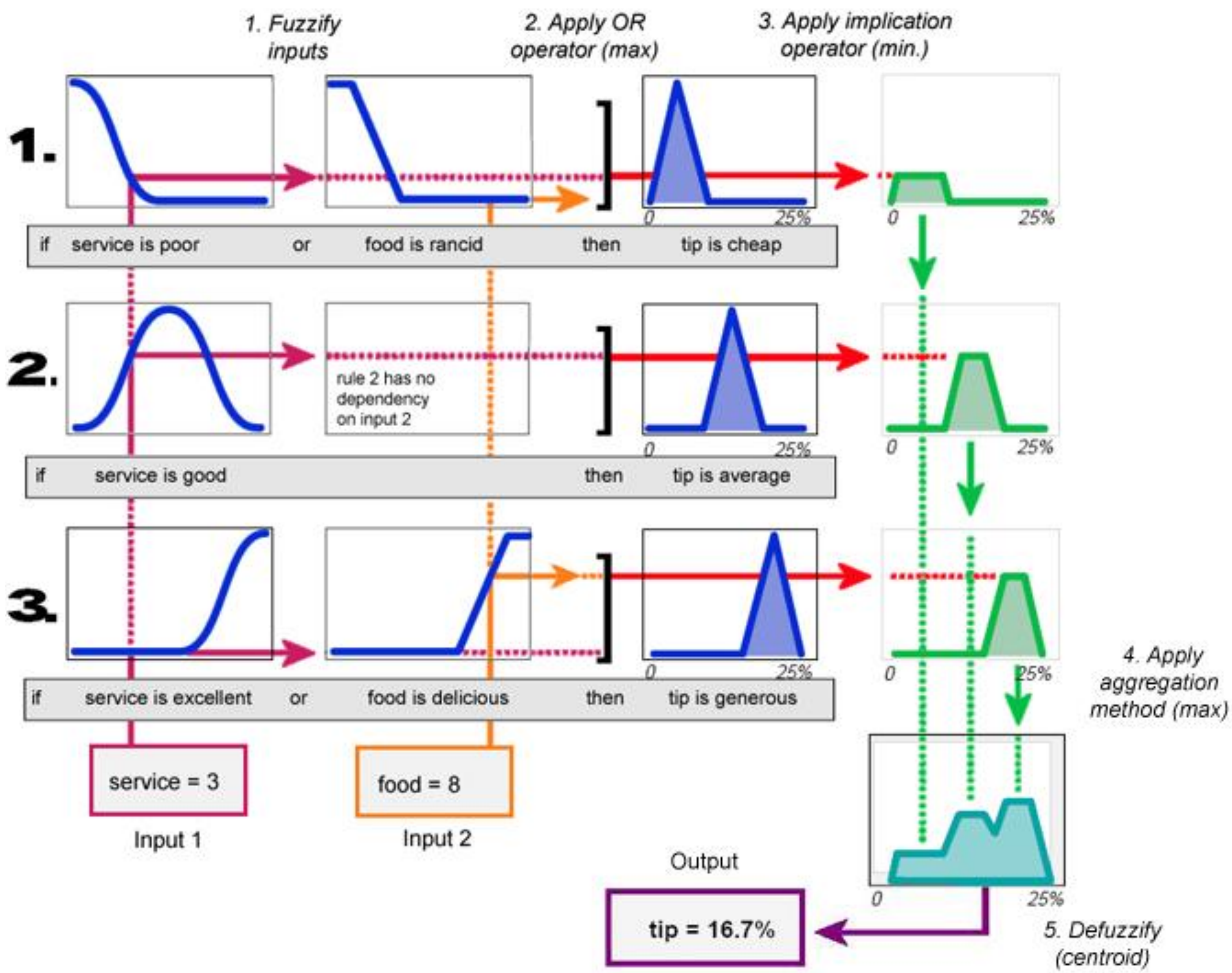
Pavyzdys: kiekvienos taisyklės išvadų darymas



Pavyzdys: išvadų apjungimas



Pavyzdys: neraiškumo eliminavimas, išvadų darymas



Takagi-Sugeno metodas

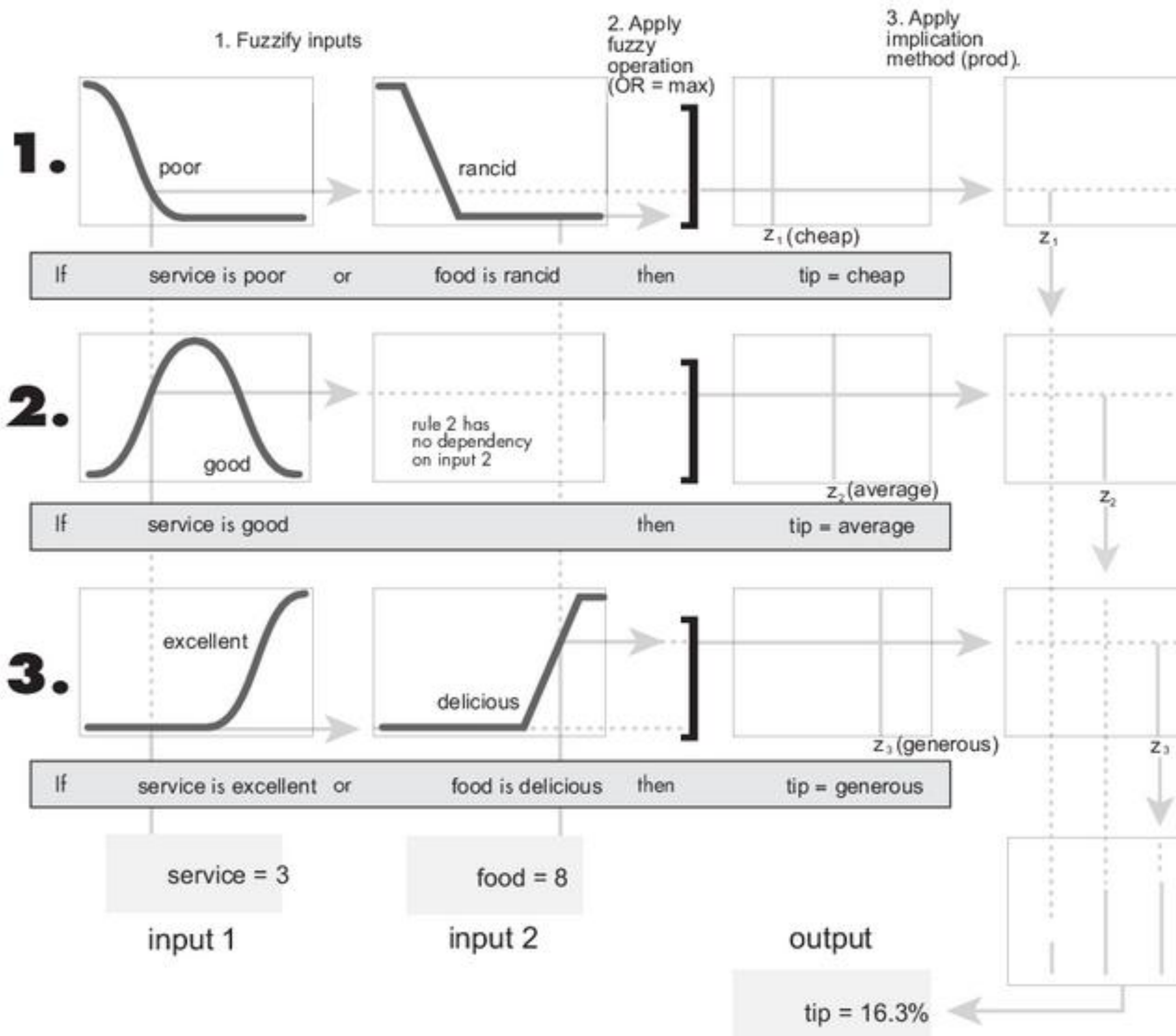
- **Taisyklės** formuojamos taip:

  IF x is A AND y is B THEN z is $f\,(x, y)$

  čia x, y ir z yra lingvistiniai kintamieji, A ir B yra neraiškiosios aibės, $\boldsymbol{f\,(x, y)}$ – matematinė funkcija.

- Takagi-Sugeno FIS naudoja **svertinius vidurkius** skaičiuojant raiškias išėjimų reikšmes, tuo tarpu Mamdani FIS tam naudoja **neraiškumo eliminavimą** (*defuzzification*).

Takagi-Sugeno metodo taikymo pavyzdys



Neraiškieji valdikliai valdymo uždaviniams

Neraiškioji logika dažnai taikoma **valdymo uždaviniams** spręsti.

Įvairūs **pavyzdžiai**:

- Stabdžių antiblokavimo sistema (ABS).
- Šildymo (šaldymo) sistemų valdikliai.
- Skalbimo mašinos valdikliai.
- Robotų valdymas.
- Kiti.

Pavyzdys: šviesoforų valdiklis

- **Tikslas** – minimizuoti automobilių laukimo laiką prie šviesoforo raudono signalo, taip pat automobilių eilę.
- Įėjimo kintamieji:
  - **Atvykimas**: automobilių, pravažiuojančių esant žaliam signalui, kiekis: **labai mažai**, **keletas**, **daug**, **labai daug**.
  - **Eilė**: automobilių eilės ilgis esant raudonam signalui: **labai trumpa**, **trumpa**, **vidutinė**, **ilga**.
- Išėjimo kintamasis:
  - **Laikas**: Žalio signalo laikas: **trumpas**, **vidutinis**, **ilgas**.

Pavyzdys: šildymo valdiklis

- Šildymo valdiklis nuolat vertina pro jį pratekančio vandens temperatūrą siekiant išlaikyti žmogaus nustatytą kambario temperatūrą.
- Įėjimo kintamieji:
  - **Temperatūrų skirtumai** tarp esamos kambario temperatūros ir temperatūros, nustatytos žmogaus. Galimos reikšmės: **žemesnė**, **tiksli**, **aukštesnė**.
  - **Dabartinis laikas**. Galimos reikšmės: **rytas**, **pietūs**, **vakaras**, naktis.
- Išėjimo kintamasis:
  - **Vandens temperatūra**: Galimos reikšmės: **šaltas**, **šiltas**, **karštas**, **labai karštas**.