**Fuzzy logika pagrįstas sprendimas panaudojant intelektinius agentus**

Sukurkite sistemą, kuri susideda iš trijų agentų: **Agentas A,** **Agentas V**, **Agentas SP**.

Įsivaizduokite kad keli draugai sugalvojo eiti miestą pavakarieniauti. Prieš išsirenkant restoraną reikia nupsręsti ar tai brangus, prieinamos kainos ar pigus restoranas. Tam tikslui reikia sukurti sistemą, kuri padės atsakyti į išsikeltą klausimą.

Sistemoje įtrauktas **Agentas A** atlieka apklausą, siekdamas išsiaiškinti kas yra brangu ir nebrangu. **Agentas A** turi trijų patiekalų sąrašą, kurių kainų įvertis jam leidžia klasifikuoti restoranus. Tas sąrašas susideda iš tokių patiekalų/produktų: (1) juodos kavos puodelio kaina, (2) dienos sriubos kaina ir (3) jautienos kepsnio kaina. **Agentas A** komunikuoja su **Agentu V**, kuris pateikia atsakymus į užduotus jam klausimus. **Agentas A** klausia kokia kaina kiekvienam iš patiekalų yra brangu, normalu ir pigu. Pvz., vartotojas gali atsakyti, kad kava brangi >3EUR, normali kaina =2 EUR, pigi >1.5 EUR. Tokius klausimus jis užduoda **Agentui V**, kuris kiekvieną kartą imituoja skirtingus vartotojus su tam tikrais atsakymas. **Agentas A** yra nuolatos veikiantis agentas, kuris cikliškai užduoda tokius klausimus ir kas penktą atsakymą suformuoja/atnaujina fuzzy priklausomybės funkcijas kiekvienam iš įėjimų (kava, sriuba, kepsnys). Kiekvienam įėjimui priskiriamos trys kategorijos, kurios aprašomos priklausomybės funkcijos, tačiau tų funkcijų kintamųjų reikšmės kinta laike, nes su lauku didėja duomenų kiekis. Pvz.:

|  |  |  |  |
| --- | --- | --- | --- |
| Klientas | Kavos kaina | | |
|  | Brangi? | Normali? | Pigi? |
| 1 | >3 | <2 | <1.5 |
| 2 | >4 | <3 | <2 |
| 3 | >2.5 | <1.5 | <1.1 |
| 4 | >2.2 | <1.5 | <1.3 |
| 5 | >3.5 | 2 | <1.8 |
| **Vidurkis** | **3.5** | **2.33** | **2.0** |
| 6 | >3.1 | <2 | <1.3 |
| 7 | **…** | **…** | **….** |

Pagal pateiktą lentelę galima nubrėžti priklausomybės funkcijas (trikampes, trapecines ir pan.), kadangi žinomos vidutinės, minimalios ir maksimalios funkcijų reikšmės.

**Agentas V** gavęs klausimą iš **Agento A,** imituoja skirtingus vartotojus, tačiau kas tam tikrą laiką, **Agentas V** imituoja ir turistą, kuriam pagal pateiktas trijų patiekalų kainas svarbu sužinoti koks tai restoranas, brangus, normalus ar pigus. Šį klausimą su inputų reikšmėmis jis užduoda **Agentui A**.

**Agentų funkcionalumai:**

**Agento A** funkcija yra sudaryti priklausomybių funkcijas visiems trims įėjimas ir jas kas tam tikrą laiką atnaujinti. Šiuos duomenis jis turi perduoti **Agentui SP**, kuris remiantis priklausomybės funkcijomis ir sudarytomis taisyklėmis (jas sudaro Agentas SP) paskaičiuoja atsakymą (output) ir perduoda atgal **Agentui A**. **Agentas V** yra ir vartotojas ir turistas. Jeigu Agentas A užduoda klausimus kas Δt, tai turistas inicijuojamas kas Δt\*s, kur

**Agentas A** nuolatos komunikuoja su **Agentu Vartotojas**.

**Agentas SP** komunikuoja tik su **Agentu A**.

Fuzzy logikos lygmenys:

|  |  |
| --- | --- |
| Kavos kaina, eurais | Kavos kainos Fuzzy lygmuo |
| <2.50 | Cheap |
| 2.00-3.00 | Normal |
| >2.50 | Expensive |

|  |  |
| --- | --- |
| Sriubos kaina, eurais | Sriubos kainos Fuzzy lygmuo |
| <3.50 | Cheap |
| 3.00-4.50 | Normal |
| >4.00 | Expensive |

|  |  |
| --- | --- |
| Kepsnio kaina, eurais | Kepsnio kainos Fuzzy lygmuo |
| <12.00 | Cheap |
| 8.00-15.00 | Normal |
| >13.00 | Expensive |

|  |  |
| --- | --- |
| Restorano įvertinimas, balai | Restorano Fuzzy lygmuo |
| <4 | Cheap |
| 2-8 | Affordable |
| >6 | Luxurious |

Taisyklės:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nr. | Kavos kaina |  | Sriubos kaina |  | Kepsnio kaina | Restoranas |
| 1. | Cheap | *and* | NOT(Cheap) | *and* | Cheap | Cheap |
| 2. | Normal | *and* | NOT(Expensive) | *and* | Cheap | Cheap |
| 3. | NOT(Expensive) | *and* | Cheap | *and* | Normal | Cheap |
| 4. | NOT(Normal) | *and* | Cheap | *and* | Cheap | Cheap |
| 5. | Cheap | *and* | NOT(Expensive) | *and* | Expensive | Affordable |
| 6. | Expensive | *and* | Cheap | *and* | NOT(Cheap) | Affordable |
| 7. | Expensive | *and* | NOT(Cheap) | *and* | Cheap | Affordable |
| 8. | Normal | *and* | Cheap | *and* | Expensive | Affordable |
| 9. | Normal | *and* | Expensive | *and* | NOT(Expensive) | Affordable |
| 10. | NOT(Expensive) | *and* | Normal | *and* | Normal | Affordable |
| 11. | Cheap | *and* | Expensive | *and* | NOT(Cheap) | Luxurious |
| 12. | Expensive | *and* | NOT(Cheap) | *and* | NOT(Cheap) | Luxurious |
| 13. | Normal | *and* | NOT(Cheap) | *and* | Expensive | Luxurious |