

## **NEIZRAZITA LOGIKA: KONFIGURIRANJE I KORIŠTENJE NEIZRAZITOG SUSTAVA ZA ODLUČIVANJE O SPORTSKOJ AKTIVNOSTI**

**ZADATAK:** Potrebno je konfigurirati neizraziti sustav pomoću Java biblioteke jFuzzyLogic koji će zaključivati koja sportska aktivnost najviše odgovara korisniku prema zadanim parametrima kao što su željena razina aktivnosti, količina suigrača, željeni rekviziti, idealna temperatura okoline, strahovi od nekih aktivnosti, godišnje doba, količina raspoloživog vremena, itd.

Nakon konfiguracije sustava potrebno je zadati ulazne parametre koji određuju razine ulaznih kategorija, a vraćati rezultat koji opisuje najbolji prijedlog – sportsku aktivnost.

Zadatak je potrebno riješiti prema sljedećim koracima:

1. Unutar razvojnog okruženja Eclipse potrebno je instalirati *plugin* za jFuzzyLogic biblioteku (detaljnije upute nalaze se na sljedećim mrežnim stranicama: <http://jfuzzylogic.sourceforge.net/html/plugin.html>)
2. U Eclipse *workspaceu* potrebno je kreirati novi Java projekt koji će sadržavati implementaciju laboratorijske vježbe. Nakon toga korištenjem opcije „Configure->Add Xtext Nature” potrebno je omogućiti napredno korištenje „FCL” datoteka.
3. U „classpath” projekta potrebno je dodati „JAR” datoteku koju možete preuzeti na mrežnim stranicama <http://jfuzzylogic.sourceforge.net/html/index.html>
4. Unutar Java projekta kreiranog u drugom koraku potrebno je kreirati „fcl” mapu u kojoj je potrebno kreirati „FCL” datoteku proizvoljnog imena. Unutar te datoteke potrebno je kreirati konfiguraciju neizrazitog sustava za odlučivanje o idealnom putovanju.

Na početku je potrebno definirati ulazne varijable, kojih mora biti minimalno pet (a može i više!), te jednu izlaznu varijablu pod nazivom „decision”.

Dan je primjer vezan za donošenje odluke o prikladnosti novog radnog mjesta.

Primjer definiranja ulaznih varijabli može izgledati ovako (u obzir su uzete samo tri ulazne varijable: „interest”, „drive” i „salary”):

```
VAR_INPUT  
interest : REAL;
```

```
drive : REAL;  
salary : REAL;  
END_VAR
```

```
VAR_OUTPUT  
decision : REAL;  
END_VAR
```

5. Unutar „FCL“ datoteke iz drugog koraka potrebno je kreirati parametre za fuzifikaciju svih pet (ili više) ulaznih varijabli.  
Na primjer, primjer fuzifikacije prvih triju varijabli može izgledati ovako (u vlastitoj implementaciji potrebno je koristiti i druge funkcije pripadnosti koje je moguće pronaći na sljedećim mrežnim stranicama:  
<http://jfuzzylogic.sourceforge.net/html/membership.html>):

```
FUZZIFY interest  
  TERM very_low := (0, 1.0) (3, 0.0);  
  TERM low := (2, 0.0) (3, 1.0) (4, 1.0) (5, 0.0);  
  TERM average := (4, 0.0) (5, 1.0) (6, 1.0) (7, 0.0);  
  TERM high := (6, 0.0) (7, 1.0) (8, 1.0) (9, 0.0);  
  TERM very_high := (8, 0.0) (9, 1.0) (10, 1.0) (11, 0.0);  
END_FUZZIFY  
  
FUZZIFY drive  
  TERM very_long := (0, 1.0) (3, 0.0);  
  TERM long := (2, 0.0) (3, 1.0) (4, 1.0) (5, 0.0);  
  TERM average := (4, 0.0) (5, 1.0) (6, 1.0) (7, 0.0);  
  TERM short := (6, 0.0) (7, 1.0) (8, 1.0) (9, 0.0);  
  TERM very_short := (8, 0.0) (9, 1.0) (10, 1.0) (11, 0.0);  
END_FUZZIFY  
  
FUZZIFY salary  
  TERM very_low := (0, 1.0) (3, 0.0);  
  TERM low := (2, 0.0) (3, 1.0) (4, 1.0) (5, 0.0);  
  TERM average := (4, 0.0) (5, 1.0) (6, 1.0) (7, 0.0);  
  TERM high := (6, 0.0) (7, 1.0) (8, 1.0) (9, 0.0);  
  TERM very_high := (8, 0.0) (9, 1.0) (10, 1.0) (11, 0.0);  
END_FUZZIFY
```

6. Unutar „FCL“ datoteke je potrebno definirati i izraze za defuzifikaciju rezultata, koji mogu izgledati npr. ovako (u vlastitoj implementaciji je potrebno koristiti i drugačije metode za defuzifikaciju, te ih prikazati i navesti u izvješčaju):

```
DEFUZZIFY decision  
  TERM very_bad_job := (0,0) (5,1) (10,0);  
  TERM bad_job := (10,0) (15,1) (20,0);
```

```
TERM average_job := (20,0) (25,1) (30,0);
TERM good_job := (30,0) (35,1) (40,0);
TERM very_good_job := (40,0) (45,1) (50,0);
TERM excellent_job := (50,0) (55,1) (60,0);
METHOD : COG;
DEFAULT := 0;
END_DEFUZZIFY
```

7. Unutar „FCL“ datoteke potrebno je definirati pravila koja su vezana uz korištene varijable i izraze definirane unutar datoteke, a omogućuju donošenje zaključaka vezanih za to koji je sport najbolji. Primjer nekoliko pravila je naveden u nastavku (u vlastitoj implementaciji potrebno je koristiti znatno više pravila zbog većeg broja ulaznih varijabli i ponuđenih sportova):

```
RULEBLOCK job_rules

AND : MIN;
OR : MAX;
ACT : MIN;
ACCU : MAX;

RULE 1 : IF interest IS very_low OR drive IS very_long OR salary IS
very_low THEN decision IS very_bad_job;

RULE 2 : IF interest IS low AND drive IS long AND salary IS average
THEN decision IS bad_job;

RULE 3 : IF interest IS average AND drive IS average AND salary IS
average THEN decision IS average_job;

RULE 4 : IF interest IS high AND drive IS short AND salary IS high
THEN decision IS good_job;

RULE 5 : IF interest IS very_high AND drive IS short AND salary IS
high THEN decision IS very_good_job;

RULE 6 : IF interest IS very_high AND drive IS very_short AND salary
IS very_high THEN decision IS excellent_job;

END_RULEBLOCK
```

8. Implementirati Java klasu koja će koristiti definiranu „FCL“ datoteku, ispisivati rezultate zaključivanja i prikazati grafove funkcija pripadnosti ulaznih varijabli, izlaznih varijabli, te samog rezultata zaključivanja. Primjer Java implementacije iz našeg primjera za tri ulazne varijable izgleda ovako:

```
String fileName = "fcl/job.fcl";
FIS fis = FIS.Load(fileName, true);

if (fis == null) {
    System.err.println("Can't load file: '" + fileName + "'");
    return;
}

fis.chart();

fis.setVariable("interest", 10);
fis.setVariable("drive", 10);
fis.setVariable("salary", 10);

fis.evaluate();

fis.getVariable("decision").chartDefuzzifier(true);

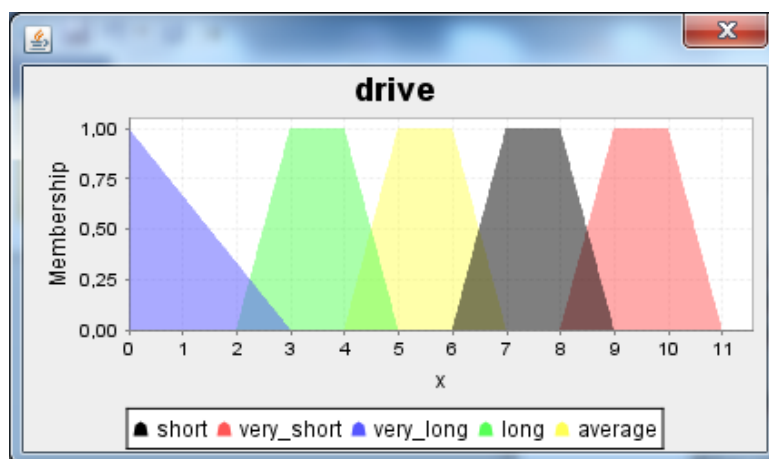
System.out.println(fis);

System.out.println("REZULTAT:");
System.out.println(fis.getVariable("decision").getValue());

for (Rule r : fis.getFunctionBlock("job_chooser")
    .getFuzzyRuleBlock("job_rules").getRules()) {
    System.out.println(r);
}
```

Primjer ispisa rezultata izvođenja programa može izgledati ovako:

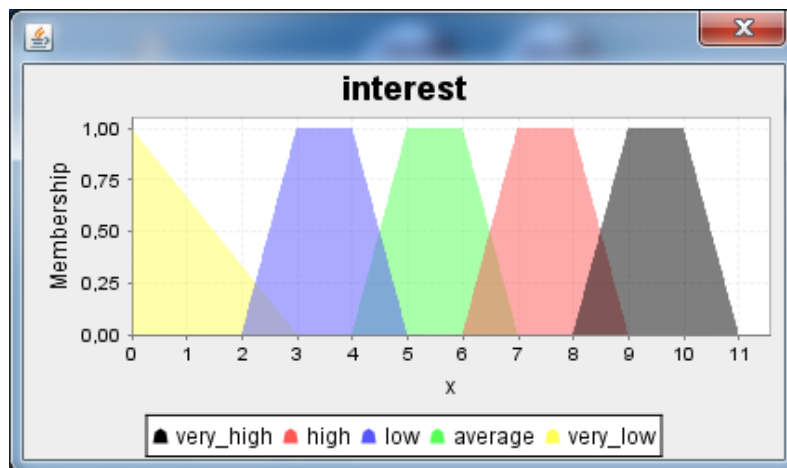
Graf za ulaznu varijablu „drive“:



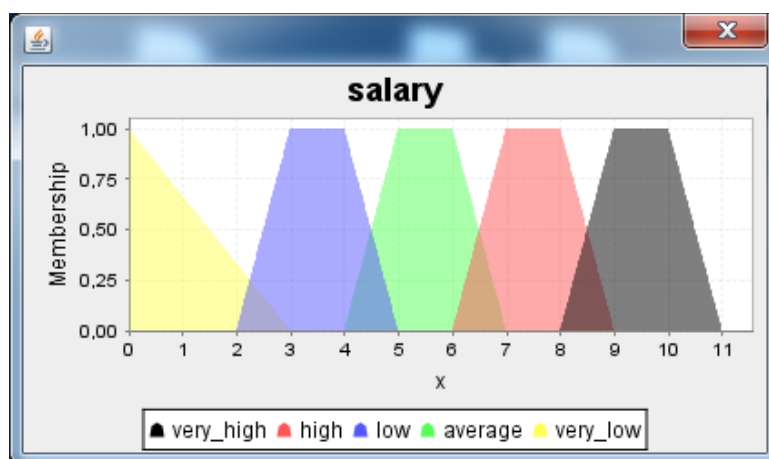
Šesta laboratorijska vježba  
 Nekonvencionalni računalni postupci

---

Graf za ulaznu varijablu „interest“:

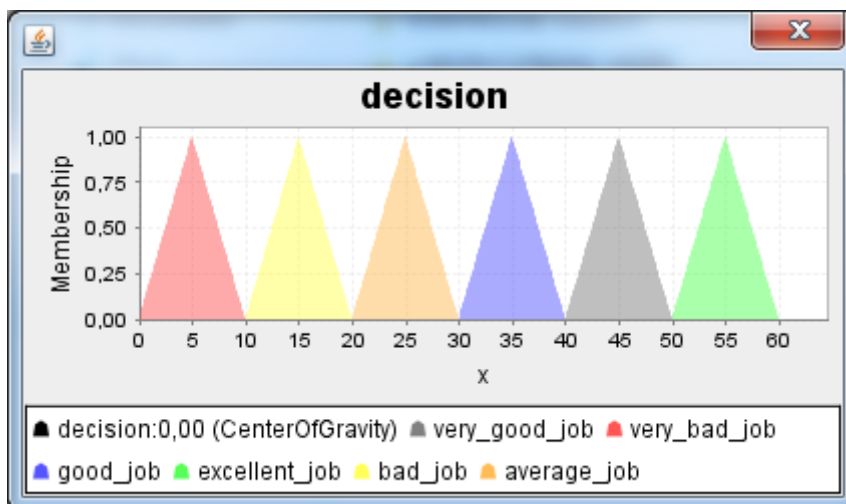


Graf za ulaznu varijablu „salary“:

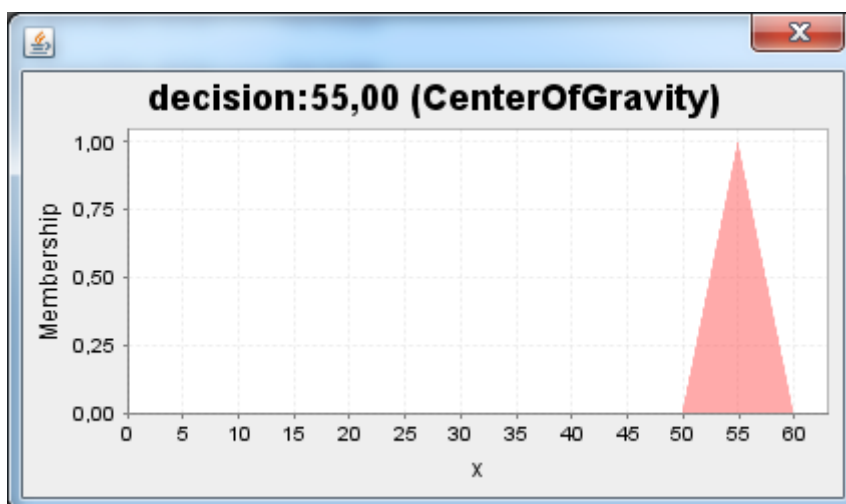


Graf za izlaznu varijablu „decision“:

Šesta laboratorijska vježba  
Nekonvencionalni računalni postupci



Graf za prikaz rezultata zaključivanja prema zadanim ulaznim vrijednostima:



U konzoli se ispisuje sljedeće:

```
FUNCTION_BLOCK job_chooser
VAR_INPUT
    drive : REAL;
    interest : REAL;
    salary : REAL;
END_VAR
VAR_OUTPUT
    decision : REAL;
END_VAR
FUZZIFY drive
```

Šesta laboratorijska vježba  
Nekonvencionalni računalni postupci

---

```

    TERM average := (4.0, 0.0) (5.0, 1.0) (6.0, 1.0) (7.0, 0.0) ;
    TERM long := (2.0, 0.0) (3.0, 1.0) (4.0, 1.0) (5.0, 0.0) ;
    TERM short := (6.0, 0.0) (7.0, 1.0) (8.0, 1.0) (9.0, 0.0) ;
    TERM very_long := (0.0, 1.0) (3.0, 0.0) ;
    TERM very_short := (8.0, 0.0) (9.0, 1.0) (10.0, 1.0) (11.0,
0.0) ;
END_FUZZIFY

FUZZIFY interest
    TERM average := (4.0, 0.0) (5.0, 1.0) (6.0, 1.0) (7.0, 0.0) ;
    TERM high := (6.0, 0.0) (7.0, 1.0) (8.0, 1.0) (9.0, 0.0) ;
    TERM low := (2.0, 0.0) (3.0, 1.0) (4.0, 1.0) (5.0, 0.0) ;
    TERM very_high := (8.0, 0.0) (9.0, 1.0) (10.0, 1.0) (11.0,
0.0) ;
    TERM very_low := (0.0, 1.0) (3.0, 0.0) ;
END_FUZZIFY

FUZZIFY salary
    TERM average := (4.0, 0.0) (5.0, 1.0) (6.0, 1.0) (7.0, 0.0) ;
    TERM high := (6.0, 0.0) (7.0, 1.0) (8.0, 1.0) (9.0, 0.0) ;
    TERM low := (2.0, 0.0) (3.0, 1.0) (4.0, 1.0) (5.0, 0.0) ;
    TERM very_high := (8.0, 0.0) (9.0, 1.0) (10.0, 1.0) (11.0,
0.0) ;
    TERM very_low := (0.0, 1.0) (3.0, 0.0) ;
END_FUZZIFY

DEFUZZIFY decision
    TERM average_job := (20.0, 0.0) (25.0, 1.0) (30.0, 0.0) ;
    TERM bad_job := (10.0, 0.0) (15.0, 1.0) (20.0, 0.0) ;
    TERM excellent_job := (50.0, 0.0) (55.0, 1.0) (60.0, 0.0) ;
    TERM good_job := (30.0, 0.0) (35.0, 1.0) (40.0, 0.0) ;
    TERM very_bad_job := (0.0, 0.0) (5.0, 1.0) (10.0, 0.0) ;
    TERM very_good_job := (40.0, 0.0) (45.0, 1.0) (50.0, 0.0) ;
    METHOD : COG;
    DEFAULT := 0.0;
    RANGE := (0.0 .. 60.0);
END_DEFUZZIFY

RULEBLOCK job_rules
    ACT : MIN;
    ACCU : MAX;
    AND : MIN;
    RULE 1 : IF ((interest IS very_low) OR (drive IS very_long))
OR (salary IS very_low) THEN decision IS very_bad_job;
    RULE 2 : IF ((interest IS low) AND (drive IS long)) AND
(salary IS average) THEN decision IS bad_job;

```

```
RULE 3 : IF ((interest IS average) AND (drive IS average)) AND  
(salary IS average) THEN decision IS average_job;  
RULE 4 : IF ((interest IS high) AND (drive IS short)) AND  
(salary IS high) THEN decision IS good_job;  
RULE 5 : IF ((interest IS very_high) AND (drive IS short)) AND  
(salary IS high) THEN decision IS very_good_job;  
RULE 6 : IF ((interest IS very_high) AND (drive IS  
very_short)) AND (salary IS very_high) THEN decision IS  
excellent_job;  
END_RULEBLOCK
```

```
END_FUNCTION_BLOCK
```

REZULTAT:

54.99992031872522

```
1 (0.0) if ((interest IS very_low) OR (drive IS very_long)) OR  
(salary IS very_low) then decision IS very_bad_job [weight: 1.0]  
2 (0.0) if ((interest IS low) AND (drive IS long)) AND (salary IS  
average) then decision IS bad_job [weight: 1.0]  
3 (0.0) if ((interest IS average) AND (drive IS average)) AND  
(salary IS average) then decision IS average_job [weight: 1.0]  
4 (0.0) if ((interest IS high) AND (drive IS short)) AND (salary  
IS high) then decision IS good_job [weight: 1.0]  
5 (0.0) if ((interest IS very_high) AND (drive IS short)) AND  
(salary IS high) then decision IS very_good_job [weight: 1.0]  
6 (1.0) if ((interest IS very_high) AND (drive IS very_short))  
AND (salary IS very_high) then decision IS excellent_job [weight:  
1.0]
```

NAPOMENE:

- A) Za dodatne informacije koristiti dokument „iec.pdf“ koji se nalazi na službenim stranicama kolegija, te koji sadrži detaljnije informacije o samoj IEC specifikaciji koja se koristiti unutar „FCL“ datoteke, kao i ostalim mogućim opcijama koje postoje unutar „jFuzzyLogic“ biblioteke.
- B) Dodatne informacije o samoj „jFuzzyLogic“ biblioteci moguće je pronaći i na sljedećim mrežnim stranicama:  
<http://jfuzzylogic.sourceforge.net/html/manual.html>
- C) Za dodatne bodove, konfigurirajte sustav koji pronalazi idealnu zajedničku aktivnost **za dvoje** ljudi s različitim preferencijama.