

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 2, 842 16 Bratislava 4

Zadanie č. 4, Dopredný produkčný systém Dokumentácia

Predmet: Umelá Inteligencia

Autor: Ivan Gulis

Študijný program: Informatika

Semester: 4.

Ak. rok: 2014/2015

Zadanie:

Úlohou je vytvoriť jednoduchý dopredný produkčný systém, s možnosťou kladenia otázok užívateľovi. Produkčný systém na základe odvodzovacieho pravidla *modus ponens* (pravidlo odlúčenia) odvodzuje zo známych faktov a produkčných pravidiel nové fakty.

Systém musí správne pracovať aspoň nad prvou uvedenou (jednoduchou) bázou znalostí, bázu znalostí si musí systém vedieť načítať zo súboru.

Jednoduchá báza znalostí
FAKTY:

(Peter je rodic Jano)
(Peter je rodic Vlado)
(manzelia Peter Eva)
(Vlado je rodic Maria)
(Vlado je rodic Viera)
(muz Peter)
(muz Jano)
(muz Vlado)
(zena Maria)
(zena Viera)
(zena Eva)

PRAVIDLÁ:

DruhyRodic1:

AK ((?X je rodic ?Y)(manzelia ?X ?Z))
POTOM ((pridaj ?Z je rodic ?Y))

DruhyRodic2:

AK ((?X je rodic ?Y)(manzelia ?Z ?X))
POTOM ((pridaj ?Z je rodic ?Y))

Otec:

AK ((?X je rodic ?Y)(muz ?X))
POTOM ((pridaj ?X je otec ?Y))

Matka:

AK ((?X je rodic ?Y)(zena ?X))
POTOM ((pridaj ?X je matka ?Y))

Surodenci:

AK ((?X je rodic ?Y)(?X je rodic ?Z)(\Leftrightarrow ?Y ?Z))
POTOM ((pridaj ?Y a ?Z su surodenci))

Brat:

AK ((?Y a ?Z su surodenci)(muz ?Y))
POTOM ((pridaj ?Y je brat ?Z))

Stryko:

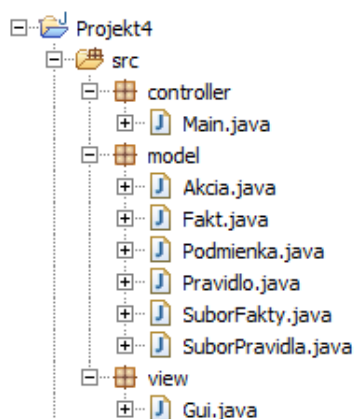
AK ((?Y je brat ?Z)(?Z je rodic ?X))
POTOM ((pridaj ?Y je stryko ?X)(sprava ?X ma stryka))

Test mazania:

AK ((?Y je stryko ?X)(zena ?X))
POTOM ((vymaz zena ?X))

Opis reprezentácie:

Všetky fakty a pravidlá sú uložené v externých súboroch fakty.txt a pravidla.txt. Pri spustení programu sú načítané do listov, reprezentované ako samostatné triedy.



Triedy SuborFakty a SuborPravidla obsahujú spomínané načítané údaje v listoch. Samotný Fakt obsahuje slová v poli stringov. Podobne triedy Podmienka a Akcia majú svoj obsah v poli stringov. Samotné Pravidlo má nasledovné atribúty:

```
private String meno;
private List<Podmienka> podmienky = new ArrayList<Podmienka>();
private List<Akcia> akcie = new ArrayList<Akcia>();
private String[] premenne = new String[26];
```

Meno podmienky, list podmienok, list akcií a mapu premenných, ktorá sa inicializuje na „0“.

V súbore fakty.txt sú fakty postupne zvlášť na riadkoch, bez zbytočných zátvoriek:

```
Peter je rodic Jano
Peter je rodic Vlado
.....
```

V súbore pravidla.txt sú pravidla oddelené mínuskom na novom riadku „-“.

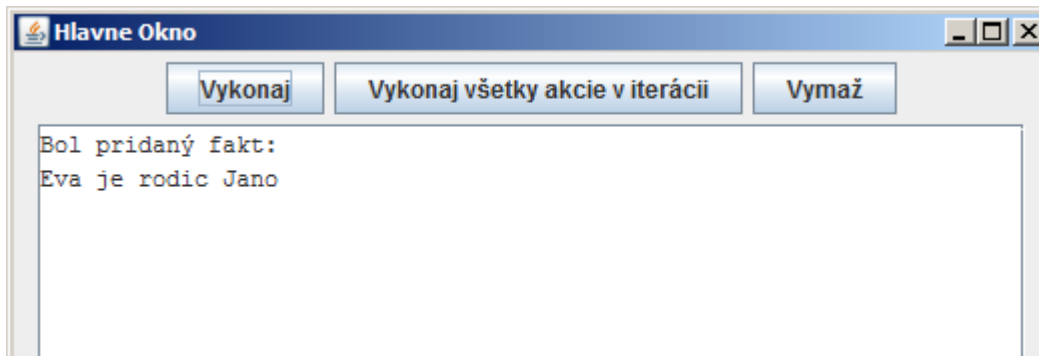
```
DruhyRodica1
2
?X je rodic ?Y
manzelia ?X ?Z
1
pridaj ?Z je rodic ?Y
-
DruhyRodica2
2
?X je rodic ?Y
manzelia ?Z ?X
1
pridaj ?Z je rodic ?Y
-
```

Pravidlá začínajú názvom pravidla.

Čísla na samostatných riadkoch reprezentujú počet podmienok a počet akcií pravidla.

Akcie začínajú slovom „pridaj“, „vymaz“, „sprava“ (podľa typu akcie). Akcia „sprava“ vypíše správu v tvare „-----XXXXX-----“ aby bola odlišiteľná od iných výpisov.

Rozhranie:



Jednoduché Gui. Tlačidlom Vykonaj sa spustí 1 krok algoritmu, prvé vykonateľné zmysluplné akcie inštancie pravidla. Druhým tlačidlom je možné nechať vykonať všetky akcie danej iterácie až do konca. Akcie popridávajú, prípadne pomažú fakty, a pri ďalšom kliknutí tlačidla sa ráta s už upravenou bázou.

Opis riešenia:

Algoritmus obsahuje nasledovné kroky:

1. Načítanie faktov a pravidiel do listov tried SuborFakty a SuborPravidla, kde sa postupne vytvoria objekty faktov a pravidiel.
2. Pre vytváranie zoznamu aplikovateľných inštancií pravidiel je púšťané postupne každé pravidlo do funkcie „rekurzia“, kde sa vždy vytvorí nový objekt pravidla s prekopírovanými údajmi, a postupne sa kontroluje zhoda prvej podmienky s každým faktom. Keď sa nájde slovo začínajúce „?“ , znamená to, že našiel premennú. Ďalší znak je veľké písmeno označujúce konkrétnu premennú. Každé pravidlo má ako atribút 26miestne pole ako mapu premenných. Index sa získava cez „charAt(1) - 'A'“. Ak sa tam nachádza „0“, tak sa do mapy najskôr zapíše nová premenná z faktu. Následne sa do pravidla prepíše premenná za hodnotu z mapy. Postupne sa paralelne porovnávajú slová podmienky a faktu. Ak sa celý fakt zhoduje, zavolá sa rekurgia na nasledujúcu podmienku. Ak už neexistuje nasledujúca podmienka, celé pravidlo sa uloží do zoznamu aplikovateľných inštancií.
3. Odfiltrujú sa všetky nepotrebné inštancie. Funkcie pridaj() a zmaz() vracajú „0“ pri neúspechu, alebo celý pridávaný alebo mazaný fakt pri úspechu. Ak sú funkcie spustené s parametrom „false“, nič sa v skutočnosti nepridáva ani nemaže, len sa testuje, či je to možné. Pri filtrovaní sa v cykle pre každú inštanciu skúšajú ich akcie (podľa prvého slova akcie). Ak žiadna z akcií nevráti string iný než „0“, inštancia je zmazaná. Pri akcii so správou zostáva hodnota „0“.

4. Spustí sa cyklus pre všetky inštancie. Návrátové hodnoty sa ukladajú do premennej, ak boli akcie pravidla úspešne vykonané, cyklus sa breakne. Preto budú vykonané akcie vždy prvej inštancie pravidla. Ak je táto funkcia spustená s parametrom true, cyklus pokračuje pre všetky inštancie.
5. Ak sú pri stlačení tlačidla vyfiltrované všetky inštancie, čiže už nie je čo vykonávať, cyklus už ďalej neprebehne.

Obmedzenia a možnosti rozšírenia:

Program pracuje len nad základnou bázou znalostí. Hlavné obmedzenie je na strane reprezentácie pravidiel. Externý súbor je syntakticky zložitý. Každé pravidlo obsahuje číslom určený počet podmienok a akcií, a všetky pravidlá sú oddelené práve jedným riadkom so znakom „-“ (alebo iným jedným znakom, ale nie prázdny riadkom!). Každá premenná v pravidle musí mať 1. znak otáznik, nasledovaný práve jedným veľkým písmenom, takže premenných môže byť maximálne 26.

Akcie sú obmedzené len na 3 základné typy: „pridaj, vymaz, sprava“. Iné akcie ako „otazka“ alebo „aktivuj“ nie je možné použiť.

Hlavným rozšírením by mohlo byť pridanie spomenutých typov akcií. V prípade otázky by v určitom kroku, kde neexistuje už žiadne vykonateľné pravidlo, bola používateľovi položená otázka na rozhodujúci fakt.

Prípadne aj akciu „aktivuj“, ktorá aktivuje konkrétne pravidlo.

Ďalším rozšírením by boli viacznakové premenné, kde by sa podstatne rozšíril ich počet, a tiež podmienené fakty (premenné typu ?5alebo3).

Iným rozšírením by mohla byť možnosť výpočtu matematických výrazov (napr. faktoriál) kde by ako fakt bola počiatočná pozícia.

Zhodnotenie:

Program je napísaný v jazyku Java, vytvorený v prostredí Eclipse. Funguje správne nad základnou bázou znalostí a využíva externé vstupné súbory, cez ktoré je možné fakty aj pravidlá manuálne pridávať (v správnom syntaxe).

Príklad výstupu:

