

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4

Zadanie č. 4

Róbert Jačko

Študijný program: Informatika
Ročník: 2.
Predmet: Umelá inteligencia
Ak. rok: 2019/2020

Zadanie

Dopredný produkčný systém

Definovanie problému

Úlohou je vytvoriť jednoduchý dopredný produkčný systém, s prípadnými rozšíreniami, napríklad o kladenie otázok používateľovi alebo vyhodnocovanie matematických výrazov.

Produkčný systém patrí medzi znalostné systémy, teda medzi systémy, ktoré so svojimi údajmi narábajú ako so znalosťami. Znalosti vyjadrujú nielen informácie o nejakom objekte, ale aj súvislosti medzi objektami, vlastnosti zvolených problémov a spôsoby hľadania ich riešenia. Znalostný systém je teda v najjednoduchšom prípade dvojica – program, ktorý dokáže všeobecne manipulovať so znalosťami a báza znalostí, ktorá opisuje problém a vzťahy, ktoré tam platia. Znalosti majú definovanú nejakú štruktúru a spôsob narábania s touto štruktúrou – to sa nazýva formalizmus reprezentácie znalostí. Program vie pracovať s týmto formalizmom, ale nesmie byť závislý od toho, aké konkrétne znalosti spracováva, inak by to už nebol systém, kde riešenie úlohy je dané použitými údajmi.

Produkčný systém na základe odvodzovacieho pravidla *modus ponens* (pravidlo odlúčenia) odvodzuje zo známych faktov a produkčných pravidiel nové fakty. Ak systém nemá dostatok vstupných údajov, môže klásť používateľovi otázky.

Produkčný systém ako program nepozná konkrétne pravidlá ani fakty! Pozná len formalizmus, v tomto prípade štruktúru pravidiel a faktov a spôsob ich spracovania. Pozná akcie (pridaj, vymaž, ...), ktoré sa môžu vykonávať, lebo tie patria do opisu formalizmu.

Táto úloha sa v tomto tvare nemôže riešiť v jazyku PROLOG, pretože PROLOG už má vstavaný mechanizmus na odvodzovanie znalostí a výsledný program by neriešil úlohu, len vhodne načítal znalosti. V prípade riešenia úlohy v jazyku PROLOG je zadanie rozšírené o pravidlá *M z N*, dynamickú prioritu a niekoľko ďalších špeciálnych podmienok na porovnávanie. Pravidlo *M z N* je splnené, ak je splnených aspoň *M* elementárnych podmienok z jeho celkového počtu *N*. Priorita určuje, ktoré pravidlo bude vykonané, ak ich je splnených viac a závisí od statickej – vopred definovanej priority a dynamickej časti, ktorá v našom prípade uvádza pomer počtu splnených podmienok ku všetkým podmienkam. Príklad báz poznatkov, nad ktorými by takýto systém mal dať korektnú odpoveď, je tu.

K funkčnému programu je potrebné pripojiť aj dokumentáciu s opisom konkrétneho riešenia (reprezentácia znalostí, algoritmus, špecifické vlastnosti) a zhodnotením činnosti vytvoreného systému. Systém musí správne pracovať aspoň nad jednoduchou bázou znalostí (ekvivalentnou s prvou uvedenou), bázu znalostí si musí systém vedieť načítať zo súboru. Je vhodné si vytvoriť aj vlastné bázy znalostí a odovzdať spolu so zdrojovým kódom.

Reprezentácia znalostí

Fakty

Fakty sú načítavané buď zo súboru, alebo z používateľského rozhrania. Každý fakt musí byť oddelený novým riadkom. V programe sú potom reprezentované ako list stringov.

```
Vlado je rodič Viera
muž Peter
žena Maria
faktorial 5
```

Hodnoty, ktoré budú dosádzané za premenné (?X, ?Y...) začínajú veľkým písmenom alebo to sú čísla. Z predchádzajúcich faktov by tieto hodnoty teda boli: Vlado, Viera, Peter, Maria, 5. V programe sú reprezentované ako list.

Pravidlá

Pravidlá sú tiež načítavané buď zo súboru, alebo z používateľského rozhrania. Každé pravidlo pozostáva z troch riadkov: meno pravidla, podmienky pravidla, akcie pravidla. Najmä kvôli prehľadnosti sú dve pravidlá od seba oddelené prázdny riadkom (nie len novým riadkom ako u faktov). V programe je potom pravidlo reprezentované ako trieda Rule a spolu teda tvoria list tohto typu.

Class Rule

atributy: **name(string)** – meno pravidla, ktoré program nájde za reťazcom “Meno: “
conditions(list(string)) – podmienky pravidla, ktoré program nájde za reťazcom “AK: “ a rozdelí ich do listu na základe oddeľovača “,”
actions(list(string)) – akcie pravidla, ktoré program nájde za reťazcom “POTOM “ a rozdelí ich do listu na základe oddeľovača “,”

```
Meno: Otec
AK    ?X je rodič ?Y,muž ?X
POTOM pridaj ?X je otec ?Y,sprava ?X je otec ?Y

Meno: Matka
AK    ?X je rodič ?Y,žena ?X
POTOM pridaj ?X je matka ?Y
```

Pred každou **premennou**, za ktorú chceme dosádzať **hodnoty** z faktov, musí byť znak “?”.

Opis algoritmu

1. Na základe výberu používateľa sa načítajú informácie zo súboru alebo z používateľského prostredia. Fakty sa uložia do listu stringov **facts** a pravidlá sa uložia do listu typu Rule **rules**.
2. Program prejde list **facts** a vyhľadá v ňom čísla alebo podreťazce začínajúce veľkým písmenom. Tie unikátne uloží do listu **values**.
3. Z listu pravidiel sa postupne vyberie **záznam**.
4. Program prejde list **conditions** a vyhľadá v ňom všetky podreťazce začínajúce znakom "?". Tie unikátne uloží do listu **variables**.
5. Z hodnôt z listu **values** sa vytvoria variácie k-tej triedy s opakovaním, k = dĺžka listu **variables**, a tieto variácie sa uložia do listu **combination_values**.
6. Postupne sa prechádza list **combination_values** a hodnoty sa dosádzajú do všetkých podmienok namiesto **premenných**. Akcie pravidla, ktorého všetky podmienky s dosadenými hodnotami sa nachádzajú vo faktoch alebo sa vyhodnotia ako pravdivé sa pridajú do listu **applicable_instances_raw**.
7. Akcie z listu **applicable_instances_raw**, ktoré pridávajú už existujúce fakty, mažu neexistujúce fakty alebo iba vypisujú správy sa odstránia a vytvorí sa tak list **applicable_instances**.
8. Ak je list **applicable_instances** prázdny tak program skončí s úspechom.
9. Vykonajú sa akcie, ktoré su v liste **applicable_instances** na prvom mieste.
10. Prejdenie na krok 2.

Testovanie

TEST1

Pravidlá

Meno: vymen

AK NSD ?a ?b,< ?a ?b

POTOM vymaz NSD ?a ?b,pridaj NSD ?b ?a

Meno: nula

AK NSD ?a 0

POTOM vymaz NSD ?a 0,pridaj NSD ?a a 0 neexistuje,sprava najvacsi spolocny delitel ?a a 0 neexistuje

Meno: start

AK NSD ?a ?b

POTOM vymaz NSD ?a ?b,pridaj medzivypocet ?a ?b ?b { ?a % ?b }

Meno: nsd1

AK medzivypocet ?x ?y ?a 0

POTOM vymaz medzivypocet ?x ?y ?a 0,pridaj NSD ?x a ?y = ?a,sprava najvacsi spolocny delitel ?x a ?y je ?a

Meno: nsd2

AK medzivypocet ?x ?y ?a ?b

POTOM vymaz medzivypocet ?x ?y ?a ?b,pridaj medzivypocet ?x ?y ?b { ?a % ?b }

Fakty

NSD 12 8456

Výstup

SPRAVY: najvacsi spolocny delitel 8456 a 12 je 4

NOVE FAKTY: NSD 8456 a 12 = 4

TEST2

Pravidlá

Meno: Druhyrodic1

AK ?X je rodic ?Y,manzelia ?X ?Z

POTOM pridaj ?Z je rodic ?Y

Meno: Druhyrodic2

AK ?X je rodic ?Y,manzelia ?Z ?X

POTOM pridaj ?Z je rodic ?Y

Meno: Otec

AK ?X je rodic ?Y,muz ?X

POTOM pridaj ?X je otec ?Y

Meno: Matka

AK ?X je rodic ?Y,zena ?X

POTOM pridaj ?X je matka ?Y

Meno: Surodenci

AK ?X je rodic ?Y,?X je rodic ?Z,<> ?Y ?Z

POTOM pridaj ?Y a ?Z su surodenci

Meno: Brat

AK ?Y a ?Z su surodenci,muz ?Y

POTOM pridaj ?Y je brat ?Z

Meno: Stryko

AK ?Y je brat ?Z,?Z je rodic ?X

POTOM pridaj ?Y je stryko ?X,sprava ?X ma stryka

Fakty

Marek je rodic Robo

manzelia Marek Eva

muz Jano

muz Robo

zena Eva

Eva je rodic Jozo

Jano a Eva su surodenci

Výstup

SPRAVY: Robo ma stryka

Jozo ma stryka

NOVE FAKTY: Eva je rodic Robo

Marek je rodic Jozo

Eva je matka Robo

Eva je matka Jozo

Robo a Jozo su surodenci

Jozo a Robo su surodenci

Robo je brat Jozo

Jano je brat Eva

Jano je stryko Robo

Jano je stryko Jozo

Zhodnotenie

Program vie pracovať iba s jednoduchými faktami bez logických operátorov. Podmienok v pravidlách môže byť aj viac, ale zreťazia sa len logickým operátorom AND. Vie vyhodnocovať podmienky >, <, <> a namiesto { matematicky_vyraz } doplní výsledok tohto výrazu. Keďže pri dosádzaní za premenné (?X, ?Y...) sa z dosádzaných hodnôt vytvoria permutácie k-tej triedy s opakovaním (k = počet unikátnych premenných) a každá z týchto variácií sa skúša dosádzať za premenné, pri narastajúcom počte unikátnych hodnôt a premenných bude zložitosť programu narastať.

Zložitosť dosádzania za premenné:

$$O(n) = n^k,$$

n = počet unikátnych hodnôt
k = počet unikátnych premenných