

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 16 Bratislava 4

Umelá inteligencia - Dopredný produkčný systém

Róbert Dudáš

Študijný program: Informatika

Ročník: 2, Krúžok: streda 8:00

Predmet: Umelá inteligencia

Cvičiaci: Ing. Ivan Kapustík

Ak. rok: 2013/14

Riešený problém – Dopredný produkčný systém

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. Ak systém nemá dostatok vstupných údajov, môže klať užívateľovi otázky.

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 prvou uvedenou (jednoduchou) bázou znalostí, bázu znalostí si musí systém vedieť načítať zo súboru. V tom prípade je tiež potrebné napísať do dokumentácie, ako by bolo možné systém rozšíriť. Je možné si vytvoriť aj vlastné bázy znalostí.

Stručný opis riešenia a podstatných častí

Reprezentácia pravidiel

Každé pravidlo si reprezentujem ako objekt *Pravidlo*. Táto trieda obsahuje meno pravidla, zoznam podmienok (*ArrayList*), zoznam špeciálnych podmienok a zoznam akcií.

Reprezentácia faktov

Každý fakt načítaný zo súboru si ukladám do zoznamu fakty (*ArrayList*) a počas behu programu sa doňho pridávajú nové fakty alebo vymazávajú.

Reprezentácia inštancií pravidiel

Inštanciu pravidla si ukladám do triedy, ktorá sa nazýva *Instanciapravidla*. Táto trieda má zoznam akcií, ktoré sa vykonajú ak táto inštancia pravidla bude vybratá zo všetkých inštancií pravidiel. Vyberám prvú inštanciu prvého pravidla ak je vykonateľná, ktorej akcie majú nejaký zmysel. Táto trieda tiež obsahuje metódu *vytvorenieinstancie()*, ktorá má tri vstupné parametre ,objekt *Pravidlo*, nájdený fakt a jednu z podmienok pravidla. Metóda naviaže z faktu a podmienky nové premenné, viem že na mieste kde v podmienke je slovo začína ? tak na tom istom mieste vo fakte na túto premennu

môžem naviazať je hodnotu, ak sa dajú naviazať tak vráti stav true a tento objekt je pridaný do zoznamu možných aplikovateľných podmienok. V prípade že existujúca premenná s nemá rovnakú hodnotu tak metóda nenaviazá nič a vráti false. V dátovej štruktúre *HashMap* si ukladám premenné a hodnoty na ktoré sú naviazané. Ak prejdem cez všetky podmienky pravidla a aj cez špeciálne podmienky viem že pravidlo je aplikovateľné a pridám ho do zoznamu aplikovateľných inštancií pravidiel.

Trieda *Prodsystem*

Táto trieda je hlavná trieda. Ukladám si v nej načítané fakty zo súboru, zoznam všetkých pravidiel načítaných zo súboru a zoznam vytvorených inštancií pravidiel a aplikovateľných inštancií pravidiel. Nachádzajú sa tu metódy na čítanie súboru. Na vyhľadanie faktov pre podmienky v pravidlách používam *regulárne výrazy*, na hľadanie zhody využívam metódu Javy find(). Ak nájde zhodu pridám fakt do zoznamu aplikovateľných faktov pre danú podmienku. Program je vo while cyklu , a cyklus ide pokiaľ nenájde všetky nové vykonateľné akcie. Algoritmus začína od pravidla, ktoré je ako prvé zadané v súbore a následne na to prechádza cez všetky podmienky v pravidle. Špeciálne podmienky vyhodnocujem až vtedy, keď som vytvoril všetky možné inštancie pravidiel pre dané pravidlo. V podstate je algoritmus ten istý ako v zadaní.

Zhodnotenie

Program bol testovaný nad jednoduchou bázou znalostí zo zadania a na tejto báze znalostí fungoval správne. Program vie načítať bázu znalostí zo súboru. Systém nie je implementovaný tak, že by kládol používateľovi otázky. Program nemá grafické používateľské prostredie.

Možné rozšírenie

Program bol testovaný len nad jednoduchou bázou znalostí zo zadania. Dal by sa však rozšíriť o ďalšiu bázu, funkciu eval ktorá by fungovala pre faktoriál.

Testovanie

Program som otestoval tak ako je to napísané v zadaní. Tu sa nachádza ukážkový výstup programu :

```
Pracovna pamat:
Peter je rodic Vlado
Peter je rodic Jano
manzelia Peter Eva
Vlado je rodic Maria
Vlado je rodic Viera
muz Peter
muz Jano
muz Vlado
zena Maria
zena Viera
zena Eva
Eva je rodic Vlado
Spravy:
Pracovna pamat:
Peter je rodic Vlado
Peter je rodic Jano
manzelia Peter Eva
Vlado je rodic Maria
Vlado je rodic Viera
muz Peter
muz Jano
muz Vlado
zena Maria
zena Viera
zena Eva
Eva je rodic Vlado
Eva je rodic Jano
```

Zhodnotenie

Keďže bol program implementovaný v jazyku *Java*, tak by sa dal veľmi jednoducho prerobiť do iného objektovo orientovaného jazyka. Použil som dátové štruktúry (*ArrayList*, *HashMap*), ktoré dostupné aj v iných programovacích jazykoch.

Použitá literatúra

Pavol Návrat a kol. – Umelá inteligencia

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/indexOf