Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 2

842 16 Bratislava 4

Umelá Inteligencia – zadanie

# Zadanie č. 4

Vypracoval: Tomáš Liščák   
Krúžok: streda 10:00  
Akademický rok: 2014/2015, letný semester

**Riešený problém a reprezentácia znalostí:**

Cieľom programu je vytvoriť produkčný systém, ktorý rozšíri danú množinu faktov odvodením nových faktov pomocou množiny pravidiel. Pravidlá a fakty si program načíta zo súboru. Moja implementácia tohto zadania je načítanie faktov vo forme vzťah, ktorý je text celého faktu a za ním argumenty, ktoré sú v texte faktu reprezentované symbolom #. Táto forma je vyjadrená v triede podmienka. Pravidlá v sebe obsahujú množinu podmienok , ako ich ľavú stranu a množinu akcii ako ich pravú stranu a sú načítavané rovnakým spôsobom ako fakty. Akcia obsahuje názov úkonu akcie a podmienku ktorá sa ňou vykoná.

**Použitý algoritmus:**

Najskôr sa načíta báza znalostí zo súboru. Potom sa v slučke vykonáva postupnosť krokov, počnúc zavolania rekurzívnej funkcie na každé pravidlo. Tá skúša splniť jednu podmienku pre každý fakt, a ak sa hodnoty daného faktu dajú zlúčiť s premennými podmienky pravidla, tak sa zavolá táto funkcia na ďalšiu podmienku pravidla. Ak sa testuje už posledná podmienka a tiež existuje fakt, ktorý pre ňu platí. Pravidlo sa odovzdá funkcii, ktorá mapuje jej pravú stranu, teda akcie, na príslušné hodnoty premenných identifikovaných v testovaní jeho podmienok. Táto identifikácia je uložená v objekte mapovania, ktorý obsahuje listy premenných a im prislúchajúcich hodnôt. Tento objekt sa vytvorí každým novým úspešným mapovaním faktu na podmienku. Pri ďalších testovaniach podmienok už sa prihliada na predchádzajúce mapovanie podmienok. Takýmto spôsobom sa otestuje každá možná kombinácia aplikovania faktov na podmienky pravidla. Keď sa pravidlo odovzdá spolu s jeho mapovaním do funkcie, ktorá ma mapovať jeho pravú stranu, tak sa ešte kontrolujú špeciálne podmienky pravidlá a to sú čí sa nejaká hodnota nerovná ďalšej hodnote argumentu podmienky. Po mapovaní akcii sa vytvorí inštancia, ktorá v sebe obsahuje názov pravidla a list akcii pravidla.

Následne sa vytvorené inštancie filtrujú. Inštancie sú pridané do aplikovateľných inštancii ak obsahujú akciu pridania, ktorá nepridáva už známy fakt, alebo akciu mazania, ktorá maže existujúci fakt.

Po odfiltrovaní sa testuje koniec základnej slučky programu. Pokiaľ ešte existujú aplikovateľné inštancie, tak program pokračuje a vykoná prvú inštanciu prvého pravidla. Pokiaľ neexistujú tak sa overí či existuje pravidlo, na ktoré sme sa ešte nepýtali a ktoré nemá vytvorenú inštanciu, ktorá bola už odfiltrovaná. Tak sa zabezpečí, že pravidlo má aspoň jednu podmienku nesplnenú. Následne sa prejdú všetky fakty a zistí sa, či sa dá aplikovať nejaký fakt na aspoň jednu podmienku daného pravidla. Ak áno, tak sa postupne všetky podmienky pravidla testujú a pre tie, ktoré sa nenašli žiadne fakty, ktoré sa dajú mapovať, sa program spýta na dané fakty používateľa. Pokiaľ sa zmapovali všetky premenné danej podmienky tak sa program opýta len či daný fakt existuje a pokiaľ sa nezmapovali, tak sa vyzve k dosadeniu premenných a následne sa zostaví fakt, na ktorého existenciu sa opýta. Po pridaní faktu do znalostí sa znova začnú mapovať pravidlá a program pokračuje odtiaľ.

**Opis testovania:**

Program zvláda prvú jednoduchú bázu znalostí. Nechal som si vypisovať všetky vytvárané inštancie v každom kroku slučky a zhodovali sa zo vzorovým riešením v každom kroku. Snažil som sa experimentovať s vlastnými bázami spočiatku v programe a program fungoval.

**Možnosti rozšírenia:**

Program by bolo možné rozšíriť o otázky ako akcie pravidla, alebo spustenie ďalších pravidiel v akcii pravidla. Taktiež by sa dal rozšíriť o vyhodnocovanie pravej strany pravidla funkciou eval, alebo testovanie konkrétnych hodnôt , alebo intervalu hodnôt v podmienkach pravidla. Túto prídavnú funkcionalitu som neimplementoval z časového hľadiska.