

Slovenská technická univerzita
Fakulta informatiky a informačných technológií

Umelá inteligencia

Zadanie č. 4 – Produkčný systém

Dávid Šinko, FIIT STU, 3. ročník, ak. rok 2017/2018

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 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.

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 faktov a produkčných pravidiel

Fakt zodpovedá výroku - ak ho systém obsahuje, považuje ho za pravdivý; ak ho neobsahuje, systém ho považuje za nepravdivý alebo nechá rozhodnúť používateľa. Fakty neobsahujú premenné!

Fakty sa na začiatku riešenia nachádzajú v báze faktov, a sú reprezentované napríklad zoznamom

```
(toto je fakt)
```

Produkčné pravidlá sú uložené v báze pravidiel. Jednoduchý produkčný systém nesmie modifikovať pravidlá počas svojej činnosti.

Pravidlo má tri časti:

- meno
- podmienky
- akcie

Meno je identifikátorom pravidla a používa sa na sledovanie priebehu inferencie – je vhodné, aby systém mal jednoduchý aj rozšírený (debug, verbal) režim.

Podmienka je konjunkciou elementárnych podmienok, to znamená, že podmienka pravidla je splnená, ak sú splnené všetky elementárne podmienky a nenastal konflikt v žiadnej premennej.

Elementárna podmienka je buď vzorom faktu (môže obsahovať premenné)

```
(?co je fakt)
```

alebo je to špeciálna podmienka

```
(<> ?a ?b)
```

ktorá sa rozpozná na základe tvaru.

Elementárna podmienka je splnená, ak sa zhoduje s niektorým faktom. Ak je to špeciálna podmienka, tak musí byť splnený ňou definovaný vzťah. Horeuvedená špeciálna podmienka je splnená, ak sa hodnota *a* nezhoduje s hodnotou *b*.

Akcie produkčného pravidla predstavujú zoznam akcií, ktoré sa aktivujú, ak je pravidlo určené na vykonanie. Pravidlá môžu obsahovať tri základné typy akcií:

- PRIDAJ vzor
- VYMAZ vzor
- SPRAVA text

Príklady:

BRAT:

AK ((?R je rodic ?X) (?R je rodic ?Y) (muz ?Y) (<> ?X ?Y))

POTOM ((PRIDAJ ?Y je brat ?X) (SPRAVA Viem ze ?X ma brata, vola sa ?Y))

Popis programu

Program sa vykonáva akoby v dvoch fázach. Najprv sa spracúva textový súbor, ktorý obsahuje pravidlá. Každému názvu pravidla prislúcha jedna alebo viac podmienok a jedna alebo viac akcií. K danému pravidlu sa analyzujú podmienky, ak ich je viac, tak sa uložia po jednom a vo vyhovujúcom formáte bez zátvoriek a nepotrebných znakov. Tak isto to je aj s akciami. Z akcie potom vieme určiť, o akú akciu sa jedná. Na analyzovanie a získavanie výrazov v potrebnom tvare používam regulárne výrazy, v C# takzvané Regex funkcie.

V druhej fáze programu prechádzam faktami a aplikujem ich na pravidlách. O to sa stará cyklus, ktorý sa opakuje dovtedy, kým nastala medzi faktami nejaká zmena. To znamená, ak bol nejaký fakt pridaný alebo vymazaný. Nastavujem si to príslušnou premennou *changed*. V tomto cykle je vykonávaných aj niekoľko ďalších na dosiahnutie výsledku. Prechádzam všetkými faktami, ktoré sú zaznamenané v HashSet-e *fakty*. Následne porovnávam podmienku z pravidiel s faktami postupne jeden po druhom a zisťujem, či sa zhodujú. Ak sa našla zhoda, vykoná sa akcia podľa toho akú dané pravidlo ponúka. Pri akcii, ktorá modifikuje pravidlá sa hodnota premennej *changed* zmení na *true*, čo znamená, že došlo k zmene. Inak keď sa vypisuje správa, tak sa iba vypíše a nič ďalej sa nedeje.

Program pokračuje až kým neskončia všetky cykly a aj ten hlavný keď sa nič nezmenilo vo faktoch. Nakoniec sa všetky fakty vypíšu do konzoly a program skončí.

Zhodnotenie

Program som testoval na faktoch a pravidlách zo vzorového programu v javasripte Rodinné vzťahy a Fiaty. Program sa vykonáva v jednom cykle, čo neumožňuje krokovanie jednotlivých akcií. Avšak program je ľahko rozšíriteľný o ďalšie typy pravidiel a taktiež malou zmenou by bolo umožnené krokovanie. Program neponúka grafické rozhranie a celý sa vykonáva v konzole, kde nakoniec aj vypíše výsledok. Keďže sa program vykonáva v jednom cykle, pri vymazávaní mám pomocné výpisy, že došlo k vymazaniu nejakého faktu. A nakoniec sa to dá zistiť aj porovnaním vstupného textového súboru obsahujúceho fakty s konečným výpisom v konzole.