

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 2, 842 16 Bratislava 4

Zadanie 4

Produkčný systém

Roman Slonskyi

Študijný program: Informatika-4

Ročník: 3

Krúžok: Ut 16:00

Predmet: Umelá inteligencia

Vedúci projektu: Ing. Jakub Mačina

Ak. rok: 2018/2019

Opis zadania

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.

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.

Opis inferencie

Inferenčný cyklus:

1. Získaj zoznam známych faktov, choď na krok 2.
2. Vytvor zoznam všetkých aplikovateľných inštancií pravidiel, choď na krok 3.
3. Odfiltruj všetky také pravidlá, ktoré by nezmenili fakty v pracovnej pamäti. Choď na krok 4.
4. Ak neexistuje žiadna aplikovateľná inštancia pravidla na aplikovanie, koniec, inak choď na krok 5.
5. Stratégiou riešenia konfliktov vyber najvhodnejšie pravidlo a jeho najvhodnejšiu inštanciu. Choď na krok 6.
6. Vykonaj vybranú inštanciu pravidla. Choď na krok 2.

Reprezentácia faktov a produkčných pravidiel

Fakt je reťazec, ktorý by mal mať tvar ľubovoľného pravidla z bázy znalostí. Na konci faktu by mal byť znak konca riadku. Napríklad: *toto je 42 fakt*

Pravidlo by malo pozostávať z dvoch častí: podmienka a akcia.

Na začiatku podmienky je slovo IF a za tým slovom by mal nasledovať reťazec, ktorý je vzorom faktu. Podmienka môže v sebe obsahovať premenné. Premenné začínajú otáznikom a za otáznik musí nasledovať jedno písmeno.

Špeciálnym typom premennej je tzv. wildcard. V tejto premennej môže obsahovať ľubolné množstvo slov alebo čísel. Vyzerá ako tri bodky uzatvorené hranatými zátvorkami ([...]).

Za podmienkou môže ísť špeciálna podmienka, ktorá je oddelená znakom čiarky.

Napríklad, `<> { ?x == ?y }`

Takisto podmienka môže mať matematickú funkciu s operáciami `+`, `-`, `*`, `/`, `>>`, `<<`, `&`, `%`, `==`, `!=`. Príklad využitia matematickej funkcie: `{ ?x + 1 + ?y }`

Druhá časť pravidla začína slovom THEN a za tým sú akcie.

Akcie, ktoré program pozná:

ADD – pridá nový fakt do súboru *facts.txt*.

DELETE – vymaže určitý fakt zo súboru *facts.txt*.

Akcie môžu mať tvar: `DELETE(fact vypocet ?x ?y)`

Ak treba urobiť niekoľko akcií naraz, jednotlivé akcie sa oddeľujú čiarkou.

Každé pravidlo môže mať svoju prioritu vykonania. Povolené priority sú 1, 2, ... 255.

Priorita sa určuje pred podmienkou a má nasledujúci tvar: `[3]`

Defaultná hodnota je 255 v prípade, že priorita nie je určená.

Implementácia

Zadanie bolo implementované v programovacom jazyku Python 3 v programovacom prostredí Microsoft Code.

User interface

Aplikácia má konzolové rozhranie a v nej sú definované takéto príkazy:

help – vypíše zoznam príkazov.

add fact - pridá nový fakt do súboru facts.txt.

step – uskutoční jeden krok inferencie.

till end – uskutoční celú inferenciu.

Zhodnotenie

Podarilo sa mi vytvoriť jednoduchý produkčný systém podľa špecifikácie zadania.

Nevýhodou mojho riešenia je v tom, že v klauzula podmienky môže pozostávať iba z dvoch prvkov.

Výhodou je to, že pravidlo môžu obsahovať špeciálnu podmienku (wildcard).