

Umelá inteligencia ZADANIE 4

Richard Križan

ÚVOD

Úlohou bolo vytvoriť dopredný produkčný systém, ktorý používa modus ponens pravidiel. Pomocou týchto pravidiel odvodzuje nové fakty definované ako akcia. Nové vygenerované fakty sú závislé už na známych faktoch, ktoré sú aplikované na pravidlá. Systém môže používať rôzne akcie k podmienkam v pravidlách napr. pridať, zmazať fakt alebo vypísať správu o fakte.

RIEŠENIE PROBLÉMU

Pri riešení problému na reprezentáciu dát, sme využívali reálny objektový pohľad na ľudí a vzťahy medzi nimi.

REPREZENTÁCIA FAKTOV

Znalosti a fakty jednotlivých osôb sú reprezentované formou objektu *Osoba*. Tento objekt predstavuje samotnú osobu, ktorá je špecifického druhu s vlastným menom a jeho vzťahmi k ostatným ľuďom. Taktiež obsahuje aj jednotlivé fakty o osobe. Báza znalosti je reprezentovaná množinu objektov triedy typu ľudia.

```
private String nameOfPerson;  
private HashMap<String, ArrayList<Person>> relationships  
private HashMap<String, Boolean> facts
```

1. Listing : *privátne atribúty triedy Osoba*

Príklad, po načítaní faktov z *listing 1*. sa v už existujúcom objekte pridá vzťah do hash mapy. Kľúčom „*je rodič*“ kam sa pod tento kľúč môže uložiť do poľa viacej osôb, s kým tento vzťah udržiava daná osoba. Pri načítaní samotného faktu o osobe sa pridá fakt len do poľa reťazcov.

(Peter je rodic Jano)

(muz Peter)

2. Listing : príklad spracovaného vzťahu a faktu

REPREZENTÁCIA PRAVIDIEL

Pravidlá sú reprezentované vlastnou triedou *Rule*, kde sa pravidlo spracuje, pridá sa mu názov. Hlavným atribútom triedy je agregácia inštancie triedy *Condition* pre ďalšie spracovanie podmienky a porovnaní s vzťahmi a faktami osôb. Taktiež táto trieda na základe definovaných vstupných premenných osôb do pravidla, dokáže skonštruovať požadované akcie pridania alebo odobratia faktu.

SPRACOVANIE PRAVIDIEL

Vybratie potrebnej časti podmienky z pravidla, tzn. rozdelenie podmienok po častiach udržiavané v nasledovnom poli vnorenej triedy *Entry*.

Pre celé pravidlo sa spracujú aj potrebné atribúty na dosadenie potrebných osôb medzi vzťahmi.

Príklad po načítaní „?X“ sa pridá znak „X“ ako kľúč do hash mapy k hodnote ktorá sa zväčšuje podľa pridania nového znaku do hash mapy. Takto vieme ľahko určiť, čo je potrebné dosadiť do podmienky podľa indexu.

```
ArrayList<Entry> aVarCondList = null;
public HashMap<Character, Integer> vars = null;

public class Entry{
    String relations;
    ArrayList<Integer> varsValues;
    String action;
}
```

3. Listing : príklad poľa vnorenej triedy Entry v triede Condition

ALGORITMUS

Po spracovaní údajov, jeden s nasledujúcich krokov je vytvorenie *n-tic* všetkých možností osôb. Tieto polia prezentujú možné kombinácie vzťahov a faktov. Vstupujú do procesu vykonávania krokov, cez tieto kombinácie, aplikované na aktuálne pravidlo. Testovanie a porovnanie zhôd, pre pravidlo sa testuje pomocou dosadenia premenných (index z kombinácie) podľa vnorenej triedy *Entry*. Ďalej sa vykoná akcia na základe správnosti pravidla. Pravidlo bolo pravdivé tak, zo správnej kombinácie opäť vyberáme indexy osôb na pozície pre doplnenie požadovanej akcie, podľa vnorenej triedy *Entry*.

TESTOVANIE

Program bol testovaný na vstupoch, priložených k programu v zložke */vstupy*. Testovacie vstupy obsahujú rodinné príklady, prebraté z ukázkového vstupu[1], ktoré sú upravené a pridané ako ďalšie vstupy, rozšírené o fakty aj pravidlá v rodinnom príklade.

ZHODNOTENIE

Produkčný systém bol implementovaný v programovacom jazyku java v IDE IntelliJ. Hlavnou výhodou systému je, že jeho triedy osôb vytvárajú až takmer strom rodokmeňov, ktorý sa dá reprezentovať grafom. Fakt a vzťah človeka je teda reprezentovaný podrobnejšie a hlbšie ako len typom *String*. Zložitosť jedného kroku aplikovania všetkých faktov na pravidlo, závisí od počtu potrebných premenných v pravidle a počtu osôb.