

Idea prekladu dotazu z datalógu do SQL

Ročníkový projekt

1. Úvod

Dotaz napísaný v datalógu užívateľom dostaneme ako obsah z textového okna a teda ako string. Ako prvé si ho budeme chcieť uložiť do nejakej štrukturovanej formy, s ktorou sa nám bude neskôr dobre pracovať. Či má dotaz správny formát (zatiaľ len na úrovni čiarok, medzier, veľkých, malých písmen) overíme parserom, ktorý nám bude vyhovujúce poradie charov postupne ukladať ako objekty. Postupne bude vytvárať objekty:

- * Atribút (bude obsahovať len názov premennej),
- * Predikát (bude obsahovať názov, zoznam vlastných atribútov, zoznam pravidiel, ktoré ho tvoria (oddelené na pole predikátov a pole podmienok))
- * Podmienka (obsahuje atribúty ktoré su v podmienke, prípadnú konštantu a operáciu (<, >, =)).

Okrem toho si bude predikát a podmienka pamätať či je negovaná, alebo nie. Ak bol teda dotaz napísaný v správnom formáte, návratovou hodnotou funkcie bude zoznam predikátov v poradí v akom boli definované, v inom prípade vráti funkcia null.

Parser overil vonkajší formát, ale neoveril či su splnené podmienky na logickej úrovni a preto si nato vytvoríme triedu Verification, ktorou oddelíme logiku overovania od triedy Predikát. Potrebujeme overiť či predikáty spĺňajú nasledovné:

- * všetky atribúty novodefinovaného predikátu sú ohraničené
- * všetky premenné z negácie sa predtým nachádzali aj v pozitívnom kontexte
- * v negovanom pravidle sa nenachádzajú singleton premenné
- * predikát nie je rekurzívny
- * každá premenná sa nachádza v celom predikáte aspoň dvakrát (inak vyhodí upozornenie, na nepoužitú premennú)

2. Preklad do SQL

Ako posledné nám ostáva samotný preklad do SQL. Vytvoríme triedu Table, ktorá nám bude predstavovať jednotlivú tabuľku uloženú v databáze. Bude obsahovať názov tabuľky a pole atribútov v správnom poradí (index 0 v poli predstavuje 1. stĺpec tabuľky). Ďalej vytvoríme triedu Databáza (pomocou návrhového vzoru Singleton), v ktorej budú uložené všetky tabuľky databázy, vrátane tých pomocných. Pre jednoduchosť si môžeme vytvoriť 3 stringy (select, from, where), ktoré na konci spojíme do konečného dotazu.

2.1 Časť FROM

Najjednoduchšia bude časť FROM, kde postupne len pridáme názvy tabuliek s pomenovaniami: ak sme mali pravidlá v poli v nasledovnom poradí *navstivil(, P1, K)*, *navstivil(I, P2, K)*, *vypil(I, A,)*, do nášho stringu pridáme *navstivil navstivil_1, navstivil_2, vypil vypil_3*
=> pomenovanie bude teda názov tabulky + (poradie pravidla v poli + 1)

2.2 Časť SELECT

Pre SELECT budeme postupne hľadať pre každú premennú (predstavujúcu atribút) nového predikátu pravidlo, ktoré ako prvé obsahuje danú premennú

=> teda pre P vo $vyplDost(P, M) :- navstivil(I, P, _), vypl(I, _, M), M \geq 20$. by to bol navstivil a pre M na rovnakom príklade vyplil.

Pomocou poradia nájdeného pravidla v poli ľahko odvodíme názov a pomocou inštancie triedy Table pre konkrétnu tabuľku sa ľahko dostaneme aj k názvu atribútu, v konkrétnom stĺpčeku. Pre prehľadnosť premenujeme jednotlivé stĺpce dotazu ktorý tvoríme na “stlpec_i”, kde i je poradie stĺpca (číslované od 0).

2.3 Časť WHERE

Potrebujeme zistiť, ktoré riadky tabuliek sa zhodujú v jednotlivých atribútoch na základe rovnakých použitých premenných. Budeme prechádzať postupne všetky premenné a hľadať ich výskyt v ostatných pravidlách. Rovnosť zapíšeme rovnakým princípom, ako sme vybrali atribúty pri selecte a pospájame stringom “ AND “. Doplníme aj vhodne naformátované podmienky predikátu, ktoré sme mali uložené pomocou inštancií triedy Podmienka (hľadáme prvý výskyt použitých premenných ako pri selecte, operáciu vhodne prepíšeme ak je potrebné).

2.4 Spracovanie NEGATION

Posledné zostáva vyriešiť negované pravidlá. Pre prehľadnosť môžeme vytvoriť nový string NEGATION. Začneme prechádzať všetky pravidlá predikátu a pokiaľ narazíme na negované, doplníme do NEGATION “NOT EXISTS (select * from “. Do sekcie from následne uložíme dvakrát názov predikátu (oddelené medzerou), ktorý tvorí dané negované pravidlo, “_”, číslo ktoré nasleduje v našom pomenovaní a “ where “. Použijeme algoritmus opísaný v časti WHERE iba s rozdielom, že budeme prechádzať pravidlá len po naše negované, ktoré aktuálne spracovávame. Rovnako doplníme prípadné podmienky predikátu (instancie triedy Podmienka), tvoriaceho negované pravidlo a ukončíme “) ”. Jednotlivé negácie spájame pomocou “ AND “.

2.5 Finalizácia SQL

Vytvoríme výsledný string FIN, ktorý bude obsahovať kompletný SQL dotaz pre tvorbu pomocnej tabuľky. Najskôr do FIN pridáme “CREATE TEMPORARY TABLE ”, názov predikátu, “ AS ”. Pridáme stringy SELECT, FROM, WHERE, NEGATION (v tomto poradí) a ukončíme dotaz “;”. Do inštancie triedy Databáza pridáme novú inštanciu triedy Tabulka, ktorá bude obsahovať názov predikátu (pre ktorý sme tvorili pomocnú tabuľku) a pole atribútov (v tomto prípade to bude pole stringov, ktoré bude obsahovať stlpec_i, kde i bude jeho poradie v poli a $i < \text{počet atribútov daného predikátu}$).