Relačná algebra pre výpočet rekurzívnych dotazov

Matúš Hedera
školiteľ: doc. Dr. Tomáš Plachetka

KI FMFI UK v BA

Jún 2022

Cieľ práce

- navrhnúť relačnú algebru rozšírenú o operátory rekurzie
- minimalizovať počet operátorov
- minimalizovať parametre operátorov
- dať do súvisu rôzne abstrakcie (logika, sémantika, výpočet)

Jazyk logiky

- definujeme zúženie logiky 1. rádu tak, aby zodpovedalo jazyku Datalog
- z jednoduchých—atomických formúl vznikajú zložené formuly
- ak A je jednoduchá formula, tak $\neg A$ je formula
- ak B, C sú formuly, tak $B \wedge C$ je formula
- implikácia sa používa výhradne na definíciu predikátu

Dotaz

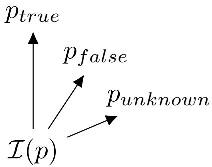
$$\left.\left.\begin{array}{c}\forall X_1,\ldots,\forall X_k(B_1 \Rightarrow A_1)\\ \wedge \\ \forall X_1,\ldots,\forall X_l(B_2 \Rightarrow A_2)\\ \wedge \\ \vdots \\ \wedge \\ \forall X_1,\ldots,\forall X_m(B_n \Rightarrow A_n)\end{array}\right\}\text{teória}\atop \left.\begin{array}{c}\wedge \\ A\end{array}\right\}\text{dotaz}\right\}\text{program}$$

- X_i sú všetky premenné i-tej implikácie, B_i sú konjunktívne formuly, A_i a A sú atomické formuly
- program zodpovedá Datalogovému programu
  $A_1 \leftarrow B_1.$
  $A_2 \leftarrow B_2.$
  $\ldots$
  $A_n \leftarrow B_n.$
  $?A$

- atomická formula A môže mat' premenné $X_1, \ldots, X_r$
- r-tica konštánt $[k_1, \ldots, k_r]$ je vo výsledku, ak $A[X_1|k_1, \ldots, X_r|k_r]$ je *true*
- dotaz je otázka na *true* logické objekty
- otázky na *false* a *unknown* nie sú (syntakticky) povolené

Trojhodnotová logika

- logický objekt $p(k_1, \ldots, k_n)$ pozostáva z predikátu p a konštánt $k_1, \ldots, k_n$
- každý logický objekt nadobúda jednu z hodnôt: $true, false, unknown$
- interpretácia $\mathcal{I}$ priradí každému predikátu p 3 relácie.



- sémantika je interpretácia, od ktorej požadujeme existenciu, jednoznačnosť, stabilitu, úplnosť (samozrejme požadujeme, aby sémantika bola modelom)
- well-founded sémantika (minimálny stabilný 3-hodnotový model) má všetky požadované vlastnosti

Prečo nestačia 2 hodnoty

- v klasickej 2-hodnotovej sémantike sa považuje za pravdivé (len) to, čo je pravdivé vo všetkých modeloch danej teórie. To nepostačuje, lebo prienik všetkých modelov nemusí byť modelom danej teórie. Napríklad, pre:

$$
\begin{aligned}
&p(1).\\
&p(2) \leftarrow \neg p(3).\\
&p(3) \leftarrow \neg p(2).
\end{aligned}
$$

  $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3$ sú všetky modely teórie, $\mathcal{M}_1 \cap \mathcal{M}_2 \cap \mathcal{M}_3$ nemusí byť modelom

	$p(1)$	$p(2)$	$p(3)$
$\mathcal{M}_1$	T	T	F
$\mathcal{M}_2$	T	F	T
$\mathcal{M}_3$	T	T	T
$\mathcal{M}_1 \cap \mathcal{M}_2 \cap \mathcal{M}_3$	$\boldsymbol{T}$	$\boldsymbol{F}$	$\boldsymbol{F}$

Výpočet rekurzívnych Datalogových programov

súčasný stav:

- relácia p_{true} je počítaná "intuitívne", od čoho sa odvíja sémantika
- zastavenie nie je garantované

```
Každému predikátu priraď prázdnu reláciu;
do
    // Urob 1 krok iterácie
    Aplikuj raz každé pravidlo programu;
    Výsledok priraď do príslušných relácií;
while Nejaká relácia sa zmenila;
```

---

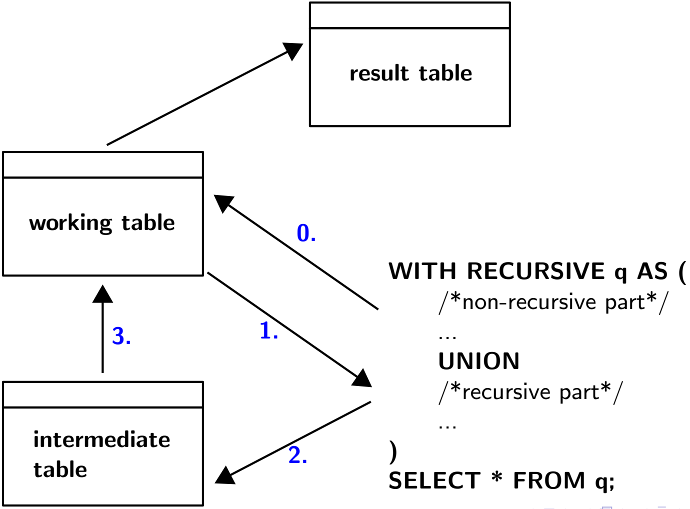
naše zlepšenie:

- relácia p_{true} je počítaná podľa vopred dohodnutej well-founded sémantiky
- relácie p_{false} a $p_{unknown}$ sa nepočítajú a neukladajú

```
Každému predikátu priraď prázdnu reláciu;
do
    do
        // Urob 1 krok iterácie
        Aplikuj raz každé pravidlo programu;
        Výsledok priraď do príslušných relácií;
    while Nejaká relácia sa zmenila;
while NOT ukončené;
```

Rekurzia v SQL

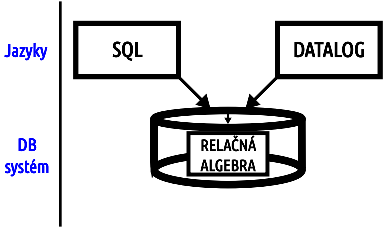
- súčasný stav: SQL systémy ukladajú medzivýsledky rekurzie do tabuliek
- naše zlepšenie: každá tabuľka je nahradená iterátorom, t.j. v danej chvíli sa počíta len 1 výstupný riadok



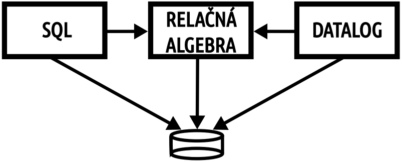
Relačná algebra, motivácia

- nezávislosť výpočtu dotazov od procedurálnych jazykov
- výpočet má štruktúru (algebraické operátory)
- kontrola nad výpočtom (optimalizácie algebraických výrazov)

súčasný stav:



naše zlepšenie:



Rekurzia v relačnej algebre

- všetky dvojice vrcholov spojené cestou v grafe r:

$$p(X, Y) \leftarrow r(X, Y).$$
$$p(X, Y) \leftarrow r(X, Z), p(Z, Y).$$
$$?p(X, Y)$$

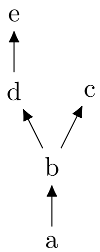
- výpočet rekurzie ako cyklus:

$p_{true} = \emptyset;$
do
  $p'_{true} = p_{true};$
  $p = r \cup \pi_{X,Y}(r \bowtie p);$
while $p'_{true} \neq p_{true};$

- výpočet rekurzie ako operátor:

$$\Psi(r \cup \pi_{X,Y}(r \bowtie p))$$

graf relácie r:



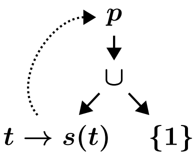
	X	Y
1. iterácia	a	b
	b	c
	c	d
	d	e
2. iterácia	a	c
	a	d
	b	e
3. iterácia	a	e

Funkčné symboly

- $p_{true} = \{1,\ s(1),\ s(s(1)),\ s(s(s(1))),\ \dots\}$
- rekurzívny operátor je smerník na algebraický výraz
- výpočet je popísaný aj v jazyku logiky (Ψ a Γ^*)

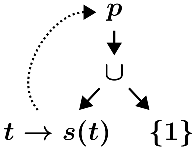
- SQL neumožňuje výpočet s funkčnými symbolmi
- výpočet $?p(X)$ je nekonečný, ale napríklad výpočet $?p(s(s(1)))$ skončí v konečnom čase

$$p(1).$$
$$p(s(X)) \leftarrow p(X).$$

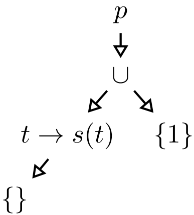


Príklad iteratívneho výpočtu

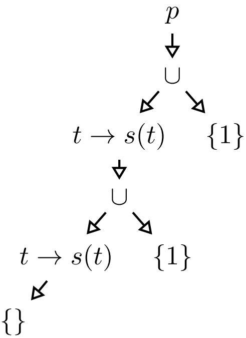
"šablóna" rekurzívneho výpočtu:



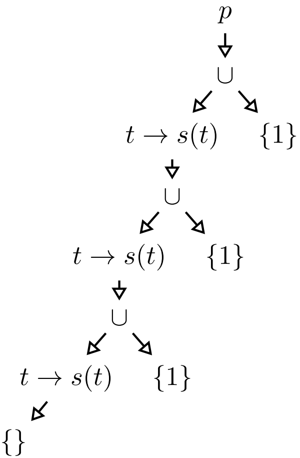
strom pre hĺbku 1:



strom pre hĺbku 2:



strom pre hĺbku 3:



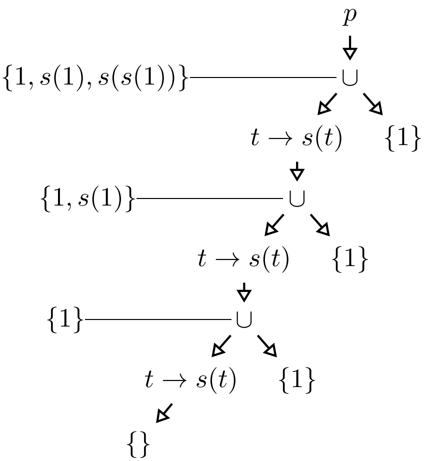
Príklad iteratívneho výpočtu

$p(1).$

$p(s(X)) \leftarrow p(X).$

$?p(s(s(1)))$

strom pre hĺbku 3:



Zhrnutie výsledkov

- rekurziu sme popísali deklaratívne (v jazyku logiky) aj imperatívne (relačná algebra, Java)
- v porovnaní so súčasnými systémami sa náš výpočet zastaví pre väčšiu triedu dotazov (garantované pre všetky dotazy bez funkčných symbolov), výsledok sa počíta podľa well-founded sémantiky
- minimalizovali sme počet algebraických operátorov a ich parametre
- z relačnej algebry sme eliminovali príkazy externého programovacieho jazyka (cykly, priradenia, ...)
- pamäťová zložitosť výpočtu dotazu nezávisí od veľkostí vstupných relácií
- vďaka integrácii s predošlými projektami vieme počítať aj dotazy s funkčnými symbolmi