

Zobecněný model preferencí nad relačním modelem dat

Radim Nedbal

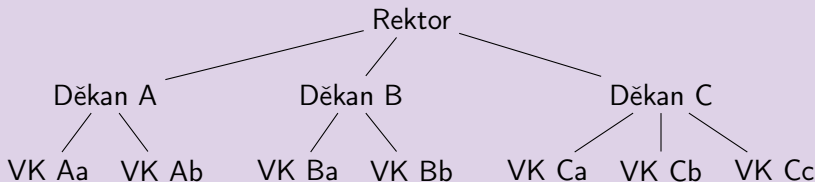
Ústav informatiky AV ČR, Praha
&
České vysoké učení technické v Praze,
Fakulta jaderná a fyzikálně inženýrská

Šumava, 6.-7. října 2006

- 1 Obsah
- 2 Model preferencí nad relačním modelem dat
 - Motivační příklad
 - První krok – definice informace reprezentovatelné uspořádáním
 - Druhý krok – datový model pro částečné uspořádání
 - Třetí krok – fyzický datový model
- 3 Dotaz nad modelem preferencí
 - Příklad dotazu s preferencemi
 - Zpracování dotazu
- 4 Závěr
 - Shrnutí
 - Práce do budoucna

Databázový dotaz

- Vstup
- DB zaměstnanců university.
 - Informace o zaměstnanecké hierarchii:



Výstup Skupina zaměstnanců university odpovídající

- striktnímu požadavku
S min. 3-členná
- optimalizačním požadavkům, tj. preferencím:
P₁ nikdo není nikoho nadřízeným;
P₂ výše postavení zaměstnanci.

Cílem je dát odpověď na otázky:

- Jak (logicky) definovat preference?
- Jak (logicky) začlenit informaci reprezentovatelnou částečným uspořádáním do RMD a do dotazovacího jazyka?
- Jaká (fyzická) datová struktura je vhodná pro implementaci?

Jazyk preferencí (Kießling 2002, Chomicki 2003)

Induktivní definice

- Základní preference P
- Restrikce preference: $P_1 := P^{\subseteq}$.
- Duální preference: $P_2 := P^{\partial}$.
- Složená preference
 - Pareto akumulace: $P_3 := P_1 \otimes P_2$.
 - Prioritní akumulace: $P_4 := P_1 \& P_2$.

$\text{POS}(\text{ATT}, \text{SET})$ udává množinu preferovaných hodnot

$P := \text{POS}(\text{BARVA}, \{\text{černá}, \text{šedá}\}) \dots P = (\text{Dom}(\text{BARVA}), \sqsubseteq_P)$

Restrikce P na množinu pěti barev

$P_1 := P^{\subseteq} = (\{\text{červená}, \text{modrá}, \text{zelená}, \text{žlutá}, \text{černá}\}, \sqsubseteq_{P^{\subseteq}})$

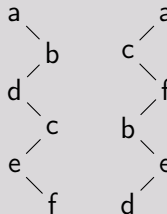
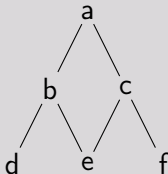
Logický datový model pro částečné uspořádání

Definice realizeru

Realizer $\mathcal{R}$ částečného uspořádání $\sqsubseteq_R$ na R je množina lineárních uspořádání $<_i, \dots$ na R splňujících:

$$a \parallel_{\sqsubseteq_R} b \rightarrow \exists <_i, <_j (a <_i b \wedge b <_j a)$$

Př. realizeru reprezentujícího částečné uspořádání



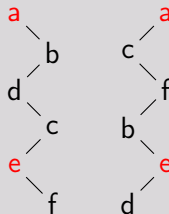
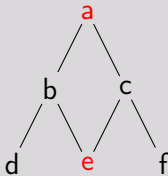
Logický datový model pro částečné uspořádání

Definice realizeru

Realizer $\mathcal{R}$ částečného uspořádání $\sqsubseteq_R$ na R je množina lineárních uspořádání $<_i, \dots$ na R splňujících:

$$a \parallel_{\sqsubseteq_R} b \rightarrow \exists <_i, <_j (a <_i b \wedge b <_j a)$$

Př. realizeru reprezentujícího částečné uspořádání



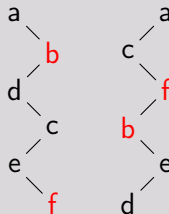
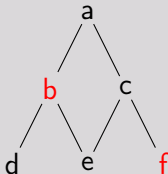
Logický datový model pro částečné uspořádání

Definice realizeru

Realizer $\mathcal{R}$ částečného uspořádání $\sqsubseteq_R$ na R je množina lineárních uspořádání $<_i, \dots$ na R splňujících:

$$a \parallel_{\sqsubseteq_R} b \rightarrow \exists <_i, <_j (a <_i b \wedge b <_j a)$$

Př. realizeru reprezentujícího částečné uspořádání



Sloupcově orientovaný datový model (Sasha, EDBT 1999)

Arrable

ID	JM.		FUNKCE		H.
101	Red		Ředitel		$\mathcal{S}_h$
102	Dbb		Nám. A		
103	Nb		Nám. B		
104	Vaa		Ved. Aa		
105	Vba		Ved. Ba		
106	Vbb		Ved. Bb		
107	Daa		Asist. Aa1		
108	Aaa2		Asist. Aa2		
109	Aba1		Asist. Ba1		
110	Abb1		Asist. Bb1		

Definice arrable

Arrable je soubor pojmenovaných konečných polí $A_1, \dots, A_l$ stejné délky. Prvky pole A_k jsou neuspořádané n -tice.

k -tý řádek arrable

$$R^*[k] = (A_1[k], \dots, A_n[k]) .$$

Databázový dotaz

- Vstup**
- DB zaměstnanců university 'R' a firmy 'S' (reprezentované pomocí dvou arrable).
 - Informace o zaměstnanecké hierarchii (reprezentované realizery $\mathcal{R}_h$ a $\mathcal{S}_h$).

- Výstup** Skupina zaměstnanců university 'R' nebo firmy 'S' odpovídající
- striktnímu požadavku
 S min. 3-členná
 - optimalizačním požadavkům, tj. preferencím:
 P_1 nikdo není nikoho nadřazeným ani v 'R' ani v 'S';
 P_2 výše postavení zaměstnanci.

Vstup DB dotazu – arrable a preference

Arrable univerzity – UNIV

ID	JM.	FUNKCE	H.
1	Rrr	Rektor	$\mathcal{R}_h$
2	Daa	Děkan A	
3	Dbb	Děkan B	
4	Dcc	Děkan C	
5	Kaa	VK Aa	
6	Kab	VK Ab	
7	Kba	VK Ba	
8	Kbb	VK Bb	
9	Kca	VK Ca	
10	Kcb	VK Cb	
11	Kcc	VK Cc	

Jazyk preferencí

$P_1 := \text{NEG}(\text{UNIV}, \text{NEG-set})$,

$P_2 := \text{PF}(S, T)$,

$P := P_1 \ \& \ P_2$.

Definice termů

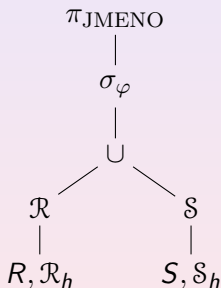
NEG-set =

$\{R' \subseteq R \mid \exists r_1, r_2 \in R' (r_1 \sqsubseteq_R r_2)\}$,

$\text{PF}(S, T) =$

$\forall t \in T (t \in S \vee \exists s \in S (t \sqsubseteq_R s))$.

Prováděcí strom dotazu



- 1 Konstrukce (bublínkový algoritmus) realizérů $\mathcal{R}$ a $\mathcal{S}$ – daná preferencemi vyjádřenými prostřednictvím preferenčního termu P .
- 2 Sjednocení $\mathcal{T} = \mathcal{R} \cup \mathcal{S}$ – algoritmus sjednocení realizérů.
- 3 Selektce $\sigma_{\varphi}(\mathcal{T})$ – algoritmus selektce na realizéru. Selektční podmínka φ využívá informaci reprezentovanou pomocí $\mathcal{R}_h$ a $\mathcal{S}_h$.
- 4 Projekce na atribut JMÉNO – při použití realizéru a arrable je triviální operací.

Shrnutí

Navržený model:

- 1 umožňuje pomocí realizérů reprezentovat **informaci obsaženou v částečném uspořádání**,
- 2 zohledňuje **uživatelské preference**, které mohou souviset i s informací reprezentovatelnou částečným uspořádáním.
- 3 V RA na realizerech platí stejné ekvivalence jako v klasické RA, např:

$$\sigma_{\varphi_1 \vee \varphi_2}(\mathcal{R}) \equiv \sigma_{\varphi_1}(\mathcal{R}) \cup \sigma_{\varphi_2}(\mathcal{R}) ,$$

$$\sigma_{\neg \varphi}(\mathcal{R}) \equiv \mathcal{R} - \sigma_{\varphi}(\mathcal{R}) ,$$

$$\mathcal{R} \cap \mathcal{S} \equiv \mathcal{R} - (\mathcal{R} - \mathcal{S}) ,$$

$$\mathcal{R} \bowtie \mathcal{S} \equiv \pi_{A \cup B}(\sigma_{\varphi}(\mathcal{R} \times \mathcal{S})) .$$

Lze tedy **převzít** velkou část pravidel pro klasickou algebraickou **optimalizaci** DB dotazů!

Práce do budoucna

- ❶ Provéřit další možnosti **optimalizace**:
 - Kdy je výhodnější 'tlačit' preference dotazu stromem dotazu směrem nahoru a kdy směrem dolů?
 - Která klasická optimalizační pravidla lze uplatnit a za jakých podmínek?
 - Jak využít iterátoru? Je možná *postupná* konstrukce realizery reflektující postupné přidávání preferenční informace? (Jaké by byly důsledky v relační algebře?)
- ❷ **Efektivnost** na velkých datech.