Equivalence of RA and RC

Dr Paolo Guagliardo

dbs-lecturer@ed.ac.uk

THE UNIVERSITY of EDINBURGH
informatics

Fall 2018

Algebra $\equiv$ Safe calculus

Fundamental theorem of database theory:
Relational algebra and Safe relational calculus **equally expressive**

- For every query in safe relational calculus
  there exists an equivalent query in relational algebra

- For every query in relational algebra
  there exists an equivalent query in safe relational calculus

From algebra to calculus

Translate each RA expression E into a FOL formula φ

Assumption: the attributes of a relation are **ordered**
(R over A, B, C means the 1st column is A, the 2nd is B, the 3rd is C)

Environment η
Maps each attribute A in the schema to a variable x_A

From algebra to calculus

Base relation

$$R \text{ over } A_1, \ldots, A_n \text{ is translated to } R\big(\eta(A_1), \ldots, \eta(A_n)\big)$$

Example
If R is a base relation over A, B

$$\eta = \{\ A \mapsto x_A,\ B \mapsto x_B,\ \ldots\ \}$$

then R is translated to $R(x_A, x_B)$

From algebra to calculus

Renaming $\rho_{A \to B}(E)$

1. Translate E to φ
2. If there is no mapping for B in η, add $\{B \mapsto x_B\}$
3. Replace every occurrence of $\eta(B)$ in φ with a **fresh** variable
4. Replace every (free) occurrence of $\eta(A)$ in φ by $\eta(B)$

Example

If R is a base relation over A, B
then $\rho_{A \to B}\big(\rho_{B \to C}(R)\big)$ is translated to $R(x_B, x_C)$

From algebra to calculus

Projection

$\pi_\alpha(E)$ is translated to $\exists X\ \varphi$

where

- φ is the translation of E
- $X = \textbf{free}(\varphi) - \eta(\alpha)$
  (attributes that are **not** projected become quantified)

Example

If R is a base relation over A, B
then $\pi_A(R)$ is translated into $\exists x_B\ R(x_A, x_B)$

From algebra to calculus

Selection

$$\sigma_\theta(E) \text{ is translated to } \varphi \wedge \eta(\theta)$$

where

- φ is the translation of E
- $\eta(\theta)$ is obtained from θ by replacing each attribute A by $\eta(A)$

Example

If R is a base relation over A, B
then $\sigma_{A=B}(R)$ is translated into $R(x_A, x_B) \wedge x_A = x_B$

From algebra to calculus

Cartesian Product, Union, Difference

Product	$E_1 \times E_2$	is translated to	$\varphi_1 \wedge \varphi_2$
Union	$E_1 \cup E_2$	is translated to	$\varphi_1 \vee \varphi_2$
Difference	$E_1 - E_2$	is translated to	$\varphi_1 \wedge \neg\varphi_2$

where

- φ_1 is the translation of E_1
- φ_2 is the translation of E_2

Example

$$\begin{aligned} \text{Customer} &: \text{CustID, Name} \\ \text{Account} &: \text{Number, CustID} \end{aligned}$$

Environment $\eta = \{\ \text{CustID} \mapsto x_1,\ \text{Name} \mapsto x_2,\ \text{Number} \mapsto x_3\ \}$

How do we translate Customer $\bowtie$ Account ? Blackboard time!

$$\exists x_4\ \text{Customer}(x_1, x_2) \land \text{Account}(x_3, x_4) \land x_1 = x_4$$

Active domain in relational algebra

For R over attributes $A_1, \ldots, A_n$

$$\textbf{Adom}(R) = \rho_{A_1 \to A}\big(\pi_{A_1}(R)\big) \cup \cdots \cup \rho_{A_n \to A}\big(\pi_{A_n}(R)\big)$$

$\textbf{Adom}(D) = \bigcup_{R \in D} \textbf{Adom}(R)$

$\textbf{Adom}_A$ is the relation $\textbf{Adom}(D)$ over attribute A

From calculus to algebra

Translate each FOL formula φ into an RA expression E

Assumptions (without loss of generality)

- No universal quantifiers, implications, double negations
- No distinct pair of quantifiers binds the same variable
- No variable occurs both free and bound
- No variable is repeated within a predicate
- No constants in predicates
- No atoms of the form x **op** x or c_1 **op** c_2

Environment η

Maps each variable x to an attribute A_x

From calculus to algebra

Let R be over attributes $A_1, \ldots, A_n$

Predicate

$R(x_1, \ldots, x_n)$ is translated to $\rho_{A_1 \to \eta(x_1), \ldots, A_n \to \eta(x_n)}(R)$

Example

For R over attributes A, B, C,
$R(x, y, z)$ is translated into $\rho_{A \to A_x,\, B \to A_y,\, C \to A_z}(R)$

From calculus to algebra

Existential quantification

$$\exists x\ \varphi \text{ is translated } to\, \pi_{\eta(X-\{x\})}(E)$$

where

- E is the translation of φ
- $X = \textbf{free}(\varphi)$

Example

For φ with free variables x, y, z and translation E,
$\exists y\ \varphi$ is translated to $\pi_{A_x, A_z}(E)$

From calculus to algebra

Comparisons

$$x \textbf{ op } y \text{ is translated to } \sigma_{\eta(x)\, \textbf{op}\, \eta(y)}\big(\textbf{Adom}_{\eta(x)} \times \textbf{Adom}_{\eta(y)}\big)$$

$$x \textbf{ op } c \text{ is translated to } \sigma_{\eta(x)\, \textbf{op}\, c}\big(\textbf{Adom}_{\eta(x)}\big)$$

Example

$x = y$ is translated to $\sigma_{A_x = A_y}\big(\textbf{Adom}_{A_x} \times \textbf{Adom}_{A_y}\big)$

$x > 1$ is translated to $\sigma_{A_x > 1}\big(\textbf{Adom}_{A_x}\big)$

From calculus to algebra

Negation

$$\neg\varphi \text{ is translated into } \Big(\bigtimes_{x \in \textbf{free}(\varphi)} \textbf{Adom}_{\eta(x)} \Big) - E$$

where E is the translation of φ

Example

For φ with free variables x, y and translation E
$\neg\varphi$ is translated to $\textbf{Adom}_{A_x} \times \textbf{Adom}_{A_y} - E$

From calculus to algebra

Disjunction: $\varphi_1 \vee \varphi_2$ is translated to

$$E_1 \times \Big(\bigtimes_{x \in X_2 - X_1} \textbf{Adom}_{\eta(x)} \Big) \cup E_2 \times \Big(\bigtimes_{x \in X_1 - X_2} \textbf{Adom}_{\eta(x)} \Big)$$

where, for $i \in \{1, 2\}$,

- E_i is the translation of φ_i
- $X_i = \textbf{free}(\varphi_i)$

Conjunction: same as disjunction, but use $\cap$ instead of $\cup$

Example

$$\begin{aligned} \text{Customer} &: \text{CustID, Name} \\ \text{Account} &: \text{Number, CustID} \end{aligned}$$

Translate $\exists x_4\ \text{Customer}(x_1, x_2) \wedge \text{Account}(x_3, x_4) \wedge x_1 = x_4$

Environment $\eta = \{\ x_1 \mapsto A,\ x_2 \mapsto B,\ x_3 \mapsto C, x_4 \mapsto D\ \}$

$$\begin{aligned} \pi_{A,B,C}\Big(&\big(E_1 \times \textbf{Adom}_C \times \textbf{Adom}_D\big) \cap \\ &\big(\textbf{Adom}_A \times \textbf{Adom}_B \times E_2\big) \cap \\ &\big(\sigma_{A=D}(\textbf{Adom}_A \times \textbf{Adom}_D) \times \textbf{Adom}_B \times \textbf{Adom}_C\big)\Big) \end{aligned}$$

where

- $E_1 = \rho_{\text{CustID}\to A,\ \text{Name}\to B}(\text{Customer})$
- $E_2 = \rho_{\text{Number}\to C,\ \text{CustID}\to D}(\text{Account})$