IDS Lecture 9: Active Domain & Translating Relational Algebra/Calculus

Active Domain

Active Domain (Adom(R) all constansts occuring in the database (a set of all values).

- Calulating queries with Adom makes for *safe* relational calculus
- queries are finite as there are finitly many elements

Evaluation of Quantifiers under active domain

Assume **Adom**$(D) = \{1,2,3\}$

$$
\begin{aligned}
&D, v \models \exists x R(x,y) \land S(x) \\
&\quad\iff \\
&D, v \models (R(1,y) \land S(1)) \lor (R(2,y) \land S(2)) \lor (R(3,y) \land S(4))
\end{aligned}
$$

Relational Algebra (RA) $\equiv$ Safe Relational Calculus (RC)

RA and RC are syntactically different but semantically *equally expressive*.

- important as your database engine needs to be able translate your what to a how.

Relational Algebra to Relational Calculus

Translate each RA expression E into a FOL formula φ.

Environment (η) *Injective Map* from attributes to values.

- map convention to be used in class $\eta(A) = x_A$

Base Relation

R over $A_1, ..., A_n$ is translated to $R(\eta(A_1), ..., \eta(A_n))$

Example: If R is a base realtion over A, B

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

Renaming

$$\rho_{\text{OLD}\to\text{NEW}}(E)$$

Process *Rename($\rho_{OLD\to NEW}(E)$) $\to$ RC:*

1. Translate E to φ.

2. If there is no mapping for NEW in η add $\{\text{NEW} \mapsto x_{\text{new}}\}$.
3. Replace every occurrence of *eta*(NEW) in φ with a *fresh* variable.
4. Replace every (free) occurrence of $\eta(\text{OLD})$ in φ by $\eta(\text{NEW})$.

Example: If R is a base relation over A, B then translate the following (RA) to relational calculus, $\rho_{A\to B}(\rho_{B\to C}(R))$.

1. Translate inner bracket $\rho_{B\to C}(R))$

    1. Translate inner bracket R over A, B gives

    $$R(x_A, x_B)$$

    2. No mapping for C (NEW) so adding C to map,

    $$M = \{A \mapsto x_A, B \mapsto x_B, C \mapsto x_C\}$$

    3. no occurrence of x_C (η(NEW)) in $R(x_A, x_B)$ so this step does nothing.
    4. Replacing x_B with x_C gives

    $$R(x_A, x_C)$$

2. Mapping for B so does nothing.
3. No instance of x_B so this step does nothing.
4. Replacing x_A with x_B

$$R(x_B, x_C)$$

Hence,

$$\rho_{A\to B}(\rho_{B\to C}(R)) \iff R(x_B, x_C)$$

Projection

$$\pi(E) \text{ is translated to } \exists X \varphi$$

where

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

Example: For a base relation R over A, B, translate $\pi_A(R)$.

$$\exists x_B R(a_A, x_B)$$

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: For relation R over A, B, translate $\sigma_{A=B \vee B=21}(R)$.

$$R(x_A, x_B) \wedge (x_A = x_B \vee x_B = 21)$$

Product, Union, Difference

Product $E_1 \times E_2$ is translated to $\varphi \wedge \varphi$
Union $E_1 \cup E_2$ is translated to $\varphi \vee \varphi$
Difference $E_1 \times E_2$ is translated to $\varphi \wedge \neg\varphi$

Example: For Relations,

- Customer (C) : CustID, Name
- Account (A) : Number, CustID
- Environment : $\eta = \{\text{CustID} \mapsto x_1, \text{Name} \mapsto x_2, \text{Number} \mapsto x_1\}$

Translate the following, Customer $\bowtie$ Account.

Solution: Expressing the join in primitive operations

$$\pi_{CustID,Name,Number}(\sigma_{CustID=CustID'}(C \times \rho_{CustID \to CustID'}(A)))$$

Translating to RC

1. Customer: $C \Rightarrow C(x_1, x_x)$
2. Account: $A \Rightarrow A(x_3, x_1)$
3. Renaming: $\rho_{CustID \to CustID'}(2) \Rightarrow A(x_3, x_1')$
4. Innermost product: $1 \times 3 \Rightarrow C(x_1, x_2) \wedge A(x_3, x_1)$
5. Selection: $\sigma_{CustID=CustID'}(4) \Rightarrow C(x_1, x_2) \wedge A(x_3, x_1) \wedge x_1 = x_4$
6. Projection: $\pi_{CustID,Name,Number}(5) \Rightarrow$

$$\exists x_4\, C(x_1, x_2) \wedge A(x_3, x_1) \wedge x_1 = x_4$$