19 Database Foundation: *Relational Calculus*

Summer Term 2010

Robert Elsässer

Albert-Ludwigs-Universität Freiburg

UNI FREIBURG

Relational Calculus

Syntax

Formulas (*R-Formulas*) are built out of constants, variables, relation names, junctors $\neg, \wedge, \vee$, quantors $\forall, \exists$ und auxiliary symbols '(', ')', ','.

- Let R be a relation name of arity k. Let $a_1, \ldots, a_k$ be constants or variables.

  $R(a_1, \ldots, a_k)$ is an (atomic) R-formula.

- Let *selection condition* α be given as $X\theta Y$, or $X\theta a$, or $a\theta X$, where X, Y variables, a a constant and $\theta \in \{=, \neq, \leq, <, \geq, >\}$ a comparison operator.

  α is an (atomic) R-formula.

- Let F be a R-formula.

  $\neg F$ is a R-Formula.

- Let F be a R-formula containing a variable X, however not containing an expression of the form $\exists X$, resp. $\forall X$. X is called *free* in F and *bound* in F otherwise.

  $\exists X \; F$ is called a ($\exists$-*quantified*) R-formula.
  $\forall X \; F$ is called a ($\forall$-*quantified*) R-formula.

  F is the *scope* of the $\exists$-, resp. $\forall$-quantifier.
- Let F and G be R-formulas and let $\mathcal{V}_F$, resp. $\mathcal{V}_G$ the set of variables contained in F, resp. G, where variables occuring in $\mathcal{V}_F \cap \mathcal{V}_G$ are free in F and free in G.

  The *conjunction* $(F \wedge G)$ is a R-formula.
  The *disjunction* $(F \vee G)$ is a R-formula.

- A relational calculus query Q over a database schema $\mathcal{R}$ is given as

$$\{(a_1, \ldots, a_n) \mid F\},$$

  where F a R-formula over $\mathcal{R}$ and $a_1, \ldots, a_n$ variables and constants.
- The set of variables among the a_i equals the set of free variables in F.
- To state a format of the result we can write

$$\{(a_1 : A_1, \ldots, a_n : A_n) \mid F\}.$$

Semantic: atomic queries

- Let the set of variables of F be $\mathcal{V}_F$. A *variable assignement* ν of F is a function over $\mathcal{V}_F$:

$$\nu : \mathcal{V}_F \rightarrow dom.$$

- We extend ν by identity for constants; for any constant a there holds $\nu(a) = a$.
- Consider a query over schema $R(A_1, \ldots, A_n)$ given as

$$Q = \{(a_1 \ldots, a_n) \mid R(a_1, \ldots, a_n)\}.$$

  Let r be an instance of R and let $F = R(a_1, \ldots, a_n)$. The *answer* to Q w.r.t. r, $Q(r)$, is defined as:

$$Q(r) = \{(\nu(a_1), \ldots, \nu(a_n)) \mid \quad \begin{array}{r} \nu \textit{ a variable assignement to } \mathcal{V}_F \\ \textit{such that } (\nu(a_1), \ldots, \nu(a_n)) \in r\} \end{array}$$

Examples

Examples

Consider schemata $R(A, B)$ and $S(B, C)$ with instances r, s.

- $\pi[A]\sigma[B = 5]R \equiv$
- $\pi[A]R \equiv$
- $\sigma[A = B]R \equiv$
- $R \bowtie S \equiv$
- $R \cup \delta[B \rightarrow A, C \rightarrow B]S \equiv$
- $R - \delta[B \rightarrow A, C \rightarrow B]S \equiv$
- Let $T(B)$ be a relation scheme and let $t = \pi[B]s$. $R \div T \equiv$

Semantic

- Let $Q = \{(a_1, \ldots, a_n) \mid F\}$ be a query, where $a_1, \ldots, a_n$ variables and constants.
  The *answer* to Q w.r.t. instance $\mathcal{I}$, $Q(\mathcal{I})$ is as follows:

$$Q(\mathcal{I}) = \{(\nu(a_1), \ldots, \nu(a_n)) \mid \quad \textit{ν a variable assignement $\mathcal{V}_F$ such that} \\ F \text{ true under } \nu \text{ w.r.t. } \mathcal{I}\}.$$

Example

Consider schemata $R(A, B), S(C, D)$ with instances r, s. Let $Q = \{(X : A, Y : B, V : C, W : D) \mid R(X, Y) \wedge S(V, W) \wedge Y > V\}$ a query.

$$r = \begin{array}{cc} A & B \\ \hline 1 & 2 \\ 2 & 2 \\ 2 & 1 \end{array} \qquad s = \begin{array}{cc} C & D \\ \hline 1 & 1 \\ 1 & 2 \\ 3 & 1 \end{array} \qquad \overset{Q}{\Longrightarrow}$$

Domain independence

- Let $Q := \{(a_1, \ldots, a_n) \mid F\}$. Let $\mathcal{I}$ an instance to $\mathcal{R}$ and *adom* the set which contains all constants in Q and all constants mentioned in $\mathcal{I}$. *adom* is called *active* domain Q; *adom* is finite.
- Q, resp. F, are called *domain independent*, if for any set $D \supset adom$ there holds:

$$Q(\mathcal{I}, adom) = Q(\mathcal{I}, D).$$

Example

Example: queries, which are not domain independent

- $R(A)$ a schema, Q a query given as

$$\{X \mid \neg R(X)\},$$

  where $\mathcal{I}(R) = \{1\}$.
- $R(A, B)$ and $S(B, C)$ schemata. Q a query given as

$$\{(X, Z) \mid \exists Y (R(X, Y) \vee S(Y, Z))\},$$

  where $\mathcal{I}(R) = \{(1, 1)\}$, resp. $\mathcal{I}(S) = \emptyset$.

Safety

If R-formula F is *safe*, then F domain independent.

- F does not contain $\forall$.
- If $F_1 \vee F_2$ subformulas of F, then F_1 and F_2 have to contain the same free variables.
- A subformula G of F is called *maximally conjunctiv*, if F does not contain a subformula of the form $H \wedge G$ or $G \wedge H$.

  Let $F_1 \wedge \cdots \wedge F_m, m \geq 1$, be a maximally conjunctive subformula of F. All free variables X have to be *bounded* in the following sense $(1 \leq j \leq m)$:

  - If X is free in F_j, where F_j neither a comparison nor negated, then X bounded.
  - If F_j of the form $X = a$ or $a = X$ and a a constant, then X bounded.
  - If F_j of the form $X = Y$ or $Y = X$ and Y bounded, then X bounded.

Examples

- $\{(X, Y) \mid X = Y \vee R(X, Y)\}$ is not safe.
- $\{(X, Y) \mid X = Y \wedge R(X, Y)\}$ is safe.
- $\{(X, Y, Z) \mid R(X, Y, Z) \wedge \neg(S(X, Y) \vee T(Y, Z))\}$ is not safe, however is safe when written equivalently as

$$R(X, Y, Z) \wedge \neg S(X, Y) \wedge \neg T(Y, Z)$$

- R-Formula (division!) $\{X \mid \forall Y (S(Y) \Rightarrow R(X, Y))\}$ is equivalent to

$$\{X \mid \neg \exists Y (S(Y) \wedge \neg R(X, Y))\}$$

Both formulas are not safe, however the equivalent writing

$$\{X \mid \neg \exists Y (S(Y) \wedge \neg R(X, Y) \wedge \exists Z\ R(X, Z)) \ \wedge \ \exists U \exists V\ R(U, V) \ \wedge \ X = U\}$$

is safe.