

Relational Calculus

Dr Paolo Guagliardo



THE UNIVERSITY *of* EDINBURGH
informatics

v26.0

Changelog

v26.0 Initial version

First-order logic

term $t := x$ (variable)
 $| c$ (constant)

formula $\varphi := P(t_1, \dots, t_n)$
 $| t_1 \text{ **op** } t_2$ with **op** $\in \{=, \neq, >, <, \geq, \leq\}$
 $| \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \neg \varphi \mid \varphi_1 \rightarrow \varphi_2$
 $| \exists x \varphi \mid \forall x \varphi$ if $x \in \text{free}(\varphi)$

free(φ) = variables that are not in the scope of any quantifier

Notation

We write $\exists x_1 \exists x_2 \dots \exists x_n \varphi$ as $\exists x_1, \dots, x_n \varphi$
and $\forall x_1 \forall x_2 \dots \forall x_n \varphi$ as $\forall x_1, \dots, x_n \varphi$

We assume **quantifiers bind till the end** of the formula:

Example

$\exists x R(x) \wedge S(x)$ stands for $\exists x (R(x) \wedge S(x))$
not for $(\exists x R(x)) \wedge S(x)$

Relational calculus

A **relational calculus query** is an expression of the form $\{\bar{x} \mid \varphi\}$ where the set of variables in $\bar{x}$ is **free**(φ)

Examples

- ▶ $Q = \{x, y \mid \exists z R(x, z) \wedge S(z, y)\}$
- ▶ $Q = \{y, x \mid \exists z R(x, z) \wedge S(z, y)\}$
- ▶ $Q = \{x, x \mid \forall y R(x, y)\}$

Queries without free variables are called **Boolean queries**

Examples

- ▶ $Q = \{() \mid \forall x R(x, x)\}$
- ▶ $Q = \{() \mid \forall x \exists y R(x, y)\}$

Data model

Relations (tables) are **sets** of **tuples** of the same length

Schema

- ▶ Set of **relation names**
- ▶ **Arity** (i.e., number of columns) of each relation name
Note that columns are ordered but have no names

Instance

- ▶ Each relation name (from the schema) of arity k is associated with a k -ary relation
(i.e., a set of tuples that are all of length k)

Examples

Customer : ID, Name, Age

Account : Number, Branch, CustID

Q_1 : Name of customers younger than 33 or older than 50

$$\{ y \mid \exists x, z \text{ Customer}(x, y, z) \wedge (z < 33 \vee z > 50) \}$$

Q_2 : Name and age of customers having an account in London

$$\{ y, z \mid \exists x \text{ Customer}(x, y, z) \wedge \exists w \text{ Account}(w, \text{'London'}, x) \}$$

Q_3 : ID of customers who have an account in **every** branch

$$\{ x \mid \exists y, z \text{ Customer}(x, y, z) \wedge (\forall u, w, v \text{ Account}(u, w, v) \rightarrow \exists u' \text{ Account}(u', w, x)) \}$$

Answers to queries

A **database instance** can be seen as a **semantic function** in FOL
(mapping relation names to relations of appropriate arity)

Recall An **assignment** ν maps variables to constants
(and each constant to itself)

Notation

For $\bar{x} = (x_1, x_2, \dots, x_n)$, $\nu(\bar{x}) = (\nu(x_1), \nu(x_2), \dots, \nu(x_n))$

The **answer** to a query $Q = \{\bar{x} \mid \varphi\}$ on a database D is

$$Q(D) = \{ \nu(\bar{x}) \mid \nu \text{ is an assignment for which } D, \nu \models \varphi \}$$

The answer to a **Boolean query** is either $\{()\}$ (**true**) or $\emptyset$ (**false**)

Safety

A query is **safe** if it gives a **finite answer** on **all** databases and this answer does **not depend** on the underlying set of constants

Examples of unsafe queries:

- ▶ $\{x \mid \neg R(x)\}$
- ▶ $\{x, y \mid R(x) \vee R(y)\}$
- ▶ $\{x, y \mid x = y\}$

Question: Are Boolean queries safe?

Bad news

Whether a relational calculus query is safe is **undecidable**

Active domain

$\mathbf{adom}(R) = \{ \text{all constants occurring in } R \}$

Example

$$\mathbf{adom} \left(\begin{array}{c|cc} R & A & B \\ \hline & a_1 & b_1 \\ & a_1 & b_2 \end{array} \right) = \{a_1, b_1, b_2\}$$

The active domain of a **database** D is

$$\mathbf{adom}(D) = \bigcup_{R \in D} \mathbf{adom}(R)$$

Active domain semantics

Evaluate queries within $\mathbf{adom}(D) \Rightarrow$ **safe relational calculus**

$$Q(D) = \{ \nu(\bar{x}) \mid \nu \text{ is an assignment of constants in } \mathbf{adom}(D) \\ \text{for which } D, \nu \models \varphi \}$$

Interpretation of formulas under (D, ν)

$D, \nu \models P(t_1, \dots, t_n)$	$\iff$	$(\nu(t_1), \dots, \nu(t_n)) \in P^D$
$D, \nu \models \neg \varphi$	$\iff$	$D, \nu \not\models \varphi$
$D, \nu \models \varphi \wedge \psi$	$\iff$	$D, \nu \models \varphi$ and $D, \nu \models \psi$
$D, \nu \models \varphi \vee \psi$	$\iff$	$D, \nu \models \varphi$ or $D, \nu \models \psi$
$D, \nu \models \varphi \rightarrow \psi$	$\iff$	if $D, \nu \models \varphi$ then $D, \nu \models \psi$
$D, \nu \models \forall x \varphi$	$\iff$	for every $c \in \mathbf{adom}(D)$: $D, \nu[x/c] \models \varphi$
$D, \nu \models \exists x \varphi$	$\iff$	there is $c \in \mathbf{adom}(D)$ s.t. $D, \nu[x/c] \models \varphi$

Evaluation of quantifiers

under active domain semantics

$$D, \nu \models \exists x \varphi \iff D, \nu \models \bigvee_{a \in \mathbf{adom}(D)} \varphi[x/a]$$

$$D, \nu \models \forall x \varphi \iff D, \nu \models \bigwedge_{a \in \mathbf{adom}(D)} \varphi[x/a]$$

where $\varphi[x/a]$ denotes the formula obtained from φ by replacing all **free** occurrences of x with a

Evaluation of quantifiers: Examples

Assume $\mathbf{adom}(D) = \{1, 2, 3\}$

$$D, \nu \models \exists x R(x, y) \wedge S(x)$$

$$\iff$$

$$D, \nu \models (R(1, y) \wedge S(1)) \vee (R(2, y) \wedge S(2)) \vee (R(3, y) \wedge S(3))$$

$$D, \nu \models \forall x S(x) \rightarrow R(x, y)$$

$$\iff$$

$$D, \nu \models (S(1) \rightarrow R(1, y)) \wedge (S(2) \rightarrow R(2, y)) \wedge (S(3) \rightarrow R(3, y))$$