Relational Calculus

Dr Paolo Guagliardo

THE UNIVERSITY *of* EDINBURGH
informatics

Fall 2020 (v20.1.0)

First-order logic

$$
\begin{aligned}
\text{term } t &:= x && \text{(variable)} \\
&\mid c && \text{(constant)}
\end{aligned}
$$

$$
\begin{aligned}
\text{formula } \varphi &:= P(t_1, \ldots, t_n) \\
&\mid t_1 \,\textbf{op}\, t_2 \quad \text{with } \textbf{op} \in \{=, \neq, >, <, \geqslant, \leqslant\} \\
&\mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \neg\varphi \mid \varphi_1 \rightarrow \varphi_2 \\
&\mid \exists x\ \varphi \mid \forall x\ \varphi \qquad \text{if } x \in \textbf{free}(\varphi)
\end{aligned}
$$

$\textbf{free}(\varphi) = \{$ variables that are not in the scope of any quantifier $\}$

Notation

We write $\exists x_1 \exists x_2 \cdots \exists x_n\, \varphi$ as $\exists x_1, \ldots, x_n\, \varphi$

We assume quantifiers bind till the end of the line:

Example

$\exists x\ R(x) \wedge S(x)$ stands for $\exists x \big(R(x) \wedge S(x)\big)$

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

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

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

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*

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

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

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

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

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

Interpretations

First-order structure $\mathcal{I}$

- Δ non-empty domain of objects (universe)
- $\cdot^{\mathcal{I}}$ gives meaning to constant/relation symbols

$$c^{\mathcal{I}} \in \Delta$$
$$R^{\mathcal{I}} \subseteq \Delta^{n}$$

Standard Name Assumption (SNA)

Every constant is intepreted as itself: $c^{\mathcal{I}} = c$

Answers to queries

- Fix an underlying domain Δ under SNA
  $\Longrightarrow$ first-order structures are just databases

Recall: an **assignment** ν maps variables to objects in Δ

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

$$Q(D) = \{\ \nu(\bar{x}) \mid \nu \colon \textbf{free}(\varphi) \to \Delta \text{ such that } D, \nu \models \varphi\ \}$$

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

Safety

A query is **safe** if it gives a finite answer on **all** databases and this answer does not depend on the universe Δ

Examples of unsafe queries:

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

Question: Are Boolean queries safe?

Bad news

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

Active domain

$\textbf{Adom}(R) = \{$ all constants occuring in R $\}$

Example

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

The active domain of a database D is

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

Active domain semantics

Evaluate queries within **Adom**(D) $\implies$ **safe relational calculus**

$$Q(D) = \{\ \nu(\bar{x}) \mid \nu\colon \textbf{free}(\varphi) \to \textbf{Adom}(D) \text{ s.t. } D, \nu \models \varphi\ \}$$

For each $\nu\colon$ **free**$(\varphi) \to$ **Adom**(D) (there are finitely many)
output $\nu(\bar{x})$ whenever $D, \nu \models \varphi$

For a safe query Q, we have that **Adom**$\big(Q(D)\big) \subseteq$ **Adom**(D)

Evaluation of quantifiers

under active domain semantics

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

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

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

Evaluation of quantifiers: Examples

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

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

$$\iff$$

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

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

$$\iff$$

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