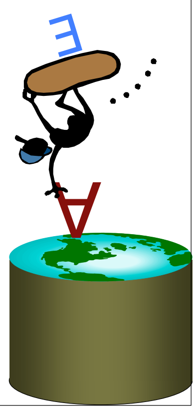
Relational Calculus

R&G, Chapter 4



> We will occasionally use this arrow notation unless there is danger of no confusion.
>
> Ronald Graham
> *Elements of Ramsey Theory*

Relational Calculus

- ***Query*** **has the form:** $\{T \mid p(T)\}$
  - $p(T)$ is a *formula* containing T
- ***Answer*** **= tuples T for which $p(T)$ = *true*.**

Formulae

- ***Atomic formulae:***
  - $R \in Relation$
  - $R.a \ op \ S.b$
  - $R.a \ op \ constant$
    - … *op* is one of $<, >, =, \leq, \geq, \neq$
- **A *formula* can be:**
  - an atomic formula
  - $\neg p, p \wedge q, p \vee q, p \Rightarrow q$
  - $\exists R(p(R))$
  - $\forall R(p(R))$

Free and Bound Variables

- **Quantifiers: $\exists$ and $\forall$**
- **Use of $\exists X$ or $\forall X$ *binds* X.**
  - A variable that is not bound is free.
- **Recall our definition of a query:**
  - $\{T \mid p(T)\}$
- **Important restriction:**
  - T must be the *only* free variable in $p(T)$.
  - all other variables must be bound using a quantifier.

Simple Queries

- **Find all sailors with rating above 7**

$$\{S \mid S \in Sailors \ \wedge \ S.rating > 7\}$$

- **Find names and ages of sailors with rating above 7.**

$$\{S \mid \exists S1 \in Sailors(S1.rating > 7 \wedge S.sname = S1.sname \wedge S.age = S1.age)\}$$

  - Note: S is a variable of 2 fields (i.e. S is a projection of *Sailors)*

Joins

Find sailors rated > 7 who've reserved boat #103

$$\{ S \mid S \in Sailors \wedge S.rating > 7 \wedge \exists R(R \in Reserves \wedge R.sid = S.sid \wedge R.bid = 103) \}$$

Joins (continued)

Find sailors rated > 7 who've reserved a red boat

$\{ S \mid S \in Sailors \wedge S.rating > 7 \wedge$
$\quad \exists R(R \in Reserves \wedge R.sid = S.sid$
$\quad\quad \wedge \exists B(B \in Boats \wedge B.bid = R.bid$
$\quad\quad\quad \wedge B.color = \text{'red'})) \}$

- **This may look cumbersome, but it's not so different from SQL!**

Universal Quantification

Find sailors who've reserved all boats

$\{ S \mid S \in Sailors \wedge$
$\quad \forall B \in Boats\ (\exists R \in Reserves$
$\quad\quad (S.sid = R.sid$
$\quad\quad \wedge B.bid = R.bid)) \}$

A trickier example…

Find sailors who've reserved all Red boats

$\{ S \mid S \in Sailors \wedge$
$\quad \forall B \in Boats\ (B.color = \textbf{'red'} \Rightarrow$
$\quad\quad \exists R(R \in Reserves \wedge S.sid = R.sid$
$\quad\quad\quad \wedge B.bid = R.bid)) \}$

Alternatively…

$\{ S \mid S \in Sailors \wedge$
$\quad \forall B \in Boats\ (B.color \neq \textbf{'red'} \vee$
$\quad\quad \exists R(R \in Reserves \wedge S.sid = R.sid$
$\quad\quad\quad \wedge B.bid = R.bid)) \}$

$a \Rightarrow b$ is the same as $\neg a \vee b$

a \ b	T	F
T	T	F
F	T	T

A Remark: Unsafe Queries

- **∃ syntactically correct calculus queries that have an infinite number of answers! *Unsafe* queries.**
  - e.g., $\{ S \mid \neg \{ S \in Sailors \} \}$
  - Solution???? Don't do that!

Expressive Power

- **Expressive Power (Theorem due to Codd):**
  - Every query that can be expressed in relational algebra can be expressed as a safe query in relational calculus; the converse is also true.
- ***Relational Completeness*:**

  Query language (e.g., SQL) can express every query that is expressible in relational algebra/calculus.

  (actually, SQL is more powerful, as we will see…)

Summary

- **Formal query languages — simple and powerful.**
  - *Relational algebra* is operational
    - used as internal representation for query evaluation plans.
  - *Relational calculus* is "declarative"
    - query = "what you want", not "how to compute it"
  - *Same expressive power*

    --> *relational completeness.*
- **Several ways of expressing a given query**
  - a *query optimizer* should choose the most efficient version.