Relational Calculus and QBE

Chapter 8 (skim)

The Relational Calculus

- Like the Relational Algebra...
  - An abstract DB language
  - No aggregate operators
  - R.A. and R.C. can accomplish the same results
    - "relationally complete"
  - "Queries" describe sets \( \{ \} \) with desired properties
  - More declarative than RA: no operations!
  - Two flavors: "tuple" and "domain"

Tuple Relational Calculus

- Set elements are tuples from relations
- 'Find employees named John'
  \[ \{ t \mid \text{EMPLOYEE}(t) \text{ and } t.FNAME = "John" \} \]
  - Read as: "The set of all t such that t is an element (member) of the relation EMPLOYEE, and the NAME attribute of t has the value John."
- QUEL (Ingres) and to some extent SQL are based on tuple R.C. model

Join Example

- "Find names of employees who work for the Research department"
  \[ \{ e.FNAME, e.LNAME \mid \text{EMPLOYEE}(e) \text{ and } ( ( \exists d ) ( \text{DEPARTMENT}(d) \text{ and } e.DNO = d.DNUMBER \text{ and } d.DNAME = "Research" ) ) ) \} \]

Relational Calculus vs. Predicate Calculus

- Both flavors of R.C. are extensions of the predicate calculus
- Can give formal definitions of terms, formulas, scoping and binding of variables, etc.
  - can build a model theory with TRUE, FALSE
- P.C. quantifiers like \( \exists, \forall \) and operators like \( \neg \) (not) can be used
  - Care must be taken with \( \forall \) and \( \neg \)

Domain Relational Calculus

- The variables represent individual domain values (rather than whole tuples)
"Find employees whose pay is over $25,000"
  \[ \{ f,n \mid ( \exists p ) \text{EMPLOYEE}(f,m,n,s,b,a,x,p,u,d) \text{ and } p > 25000 \} \]
- This is already a shorthand: should say f exists and belongs to the domain "string", etc. etc. for each variable.
Domain R.C. Join Example

• "Find employees of the Research dept. whose pay is over $25,000"

(fn,ln | 
(∃ p) (∃ d1) (EMPLOYEE (fn,m,ln,s,b,a,x,p,a,d1) and p > 25000) and 
(∃ d2) (∃ dn) DEPARTMENT (dn,d2,mg,ms) and d1= d2 and dn="Research")

Many of the variables are just placeholders.
In Prolog you could use "anonymous variables" (_).

Not so abstract, really...

• "Query by Example" uses this framework
• User see 2-dimensional layout of schema
• Can select which tables are involved, which columns (variables) appear in output (i.e., what vars are to the left of |), what conditions apply between columns, etc. etc.
• This is the predominant paradigm in direct end-user query systems (e.g. Microsoft Access).