Introduction to Relational Databases

- Bachelor CS, Lille 1 University
- Nov. 9th, 2011 (lecture 10/12)
- Today's lecturer: A. Bonifati
- Topic: Formal query languages (from SQL a step back to Tuple-oriented Relational Calculus - TRC)

Relational calculus

- **It is part of a family of formal languages**
- **Two major languages**
  - TRC, Tuple Relational Calculus
  - DRC, Domain Relational Calculus
- **There exist two versions of TRC:**
  - with range-restricted tuples
  - with arbitrary tuples
- **We will study TRC with arbitrary tuples**

TRC is declarative

- It lets express *what* we do want in the result but not *how* to obtain it
- As such, it is quite different from relational algebra
- Declarativeness is a typical feature of relational languages. It holds for TRC and SQL!

TRC: formal definition

- **Standard form: { t | p(t) }**
  p(t) is a formula, built with atoms
- **Definition of atoms:**
  $t \in R$
  expr comp expr
- comp **is comparison operator**: =, <>, >, >=, <, <=
- expr is an expression using constants and t[A]
- t[A] is a restriction on the attribute A of a tuple t
- Example: $\{t \mid t \in R\}$

TRC: formal definition

Rules to construct a valid formula:

- an atom is a formula
- if p is a formula, ¬p and (p) are also valid formulas (negation)
- if p1 and p2 are valid formulas, then $p1 \wedge p2$, $p1 \vee p2$, $p1 \Rightarrow p2$ are valid formulas (conjunction, disjunction, implication)
- if p is a formula in which s is a variable, then $\exists\, s \in R\ (p(s))$, $\forall\, s \in R\ (p(s))$ are valid formulas (existential, universal quantification

Properties of the TRC

- De Morgan's Law
  $p1 \wedge p2 \equiv \neg(\neg p1 \vee \neg p2)$
- Correspondence betweenquantifiers
  $\forall\, t \in R\ (p(t)) \equiv \neg \exists\, t \in R\ (\neg p(t))$
- Definition of the implication
  $p1 \Rightarrow p2 \equiv \neg p1 \vee p2$

Normal forms

- The above three laws imply that it is possible to write all kinds of expressions with/out implication:
  - Only one quantifier
  - Only one binary operator
- The most known normal form (similar to SQL) uses the existential quantifier and conjunction

TRC: examples

- Names of the students who were graded A in "mathematics"

$$\{\, t \mid \exists\, t1 \in STUDENT,\ \exists\, t2 \in EXAM,\ \exists\, t3 \in CLASS\ (t[Name]=t1[Name] \wedge t1[Sid]=t2[Sid] \wedge t2[Cid]=t3[Cid] \wedge t2[Grade]=A \wedge t3[Title]=\text{'mathematics'}) \,\}$$

TRC: examples

- Ids of the students who have passed the exam for "mathematics" but not for "databases"

{ t | ∃ t1 ∈ EXAM, ∃ t2 ∈ CLASS
(t[Sid]=t1[Sid] ∧
t1[Cid]=t2[Cid] ∧
t2[Title]='mathematics') ∧
¬ (∃ t3 ∈ EXAM, ∃ t4 ∈ CLASS
(t[Sid]=t3[Sid] ∧
t3[Cid]=t4[Cid] ∧
t4[Title]='databases'))) }

Correctness

- We avoid to use 'unsafe' formulas:
  { t | t ∉ R } returns an infinite result!
- Only formulas that are independent of the domain are correct
  - the solution does not depend of the domain of the attributes, but solely on the database instances

RA is expressible with TRC

It is sufficient to show that the RA fundamental operators can be expressed in TRC:

- Selection, $\sigma_{A=1}$ R:
  { t | ∃ t ∈ R (t[A]=1) }
- Projection, Π_{AC} R:
  { t | ∃ t1 ∈ R (t[A,C]=t1[A,C]) }

RA is expressible with TRC

- Cartesian Product, R(A,B,C) × S(D,E,F):
  { t | ∃ t1 ∈ R, ∃ t2 ∈ S
  (t[A,B,C]=t1[A,B,C] ∧
  t[D,E,F]=t2[D,E,F]) }
- Join, R(A,C) $\bowtie_{A=B}$ S(B,D):
  { t | ∃ t1 ∈ R, ∃ t2 ∈ S
  (t[A,C] = t1[A,C] ∧
  t[B,D] = t2[B,D] ∧
  t[A] = t[B]) }

RA is expressible with TRC

- **Union, R ∪ S:**

  $\{ t \mid \exists t1 \in R, \exists t2 \in S \; (t = t1 \lor t = t2) \}$

- **Difference, R − S:**

  $\{ t \mid \exists t \in R \; (t \notin S) \}$

Also TRC is expressible with the RA

- **However, the proof is ways more complex**
- **It needs to exclude the unsafe and domain-dependent expressions**
- **Under this hypothesis, TRC and AR have the same expressive power**

Exercise 1

- Given the following database schema:
  - MOVIE (Mid, Title, Year, Type, Director)
  - INTERPRETATION (Mid, Actor, Role, Starring)
  - ARTIST (Name, DateOfBirth, PlaceOfBirth)
- Return (in TRC) the artists that have played in a movie or have directed a movie in 2011.

Exercise 2

- Given the following database schema:
  - PARTNERS (Project, Name)
  - PROJECT (Number, Title, Budget, StartDate, EndDate, PercBudget)
  - ACTIVITY (NProg, Person, Type, NrHour)
  - COORDINATOR (NProg, Person)
  - PERSON (Name, CostPerHour, MainActivity)
- Return (in TRC) the persons who have coordinated projects or performed activities, but have never coordinated a project in which they have performed activities.