Foundations of Query Languages

Dr. Fang Wei

Lehrstuhl für Datenbanken und Informationssysteme
Universität Freiburg

SS 2011

Relational Query Languages

Formal languages with syntax and semantics.

- Syntax: algebraic or logical formalism or specific query language (SQL). Uses DB schema vocabulary.
- Semantics: $M[Q]$ a mapping that transforms a database (instance) D into a database (instance) $D' = M[Q](D)$.

We always disregard queries that are dependent on the particular representation of domain values. We thus restrict our attention to *Generic queries*: Queries that Produce isomorphic results on isomorphic databases.

Expressive Power

Expressive power of a query language L:

$$E(L) = \{M[Q] | Q \in L\}$$

This way we can compare query languages, e.g.: if $E(L) \subset E(L')$ then L' is strictly more expressive than L.
One often writes $L = L'$ instead of $E(L) = E(L')$.

Relational Algebra

- σ Selection *
- π Projection *
- $\times$ Cross product *
- $\bowtie$ Join
- ρ Rename *
- $-$ Difference *
- $\cup$ Union *
- $\cap$ Intersection

*: Primitive operations, all others can be obtained from these.

First-order Formulas (Relational Calculus)

- Quantifiers:
  $\forall$, $\exists$
- Boolean connectives
  $\wedge$, $\vee$, $\neg$
- Parentheses
  (,)
- Atoms
  $R(t_1, \ldots, t_n)$

Example

Schema

- Author (AID integer, name string, age integer)
- Paper (PID string, title string, yearinteger)
- Write (AID integer, PID integer)

Instance

- {(h142, Knuth, 70), (h123, Ullman, 60), ...}
- {(h181140pods, Query containment, 1998), ...}
- {(h123, 181140pods), (h142, 193214algo), ...}

Example

PID's of the papers that Knuth is NOT writing.

$$\pi_{PID} Paper - \pi_{PID}(Write \bowtie \sigma_{name='Knuth'} Author)$$

$$\{PID \mid \exists title, year.Paper(PID, title, year) \wedge \\ \neg \exists age, AID.(Write(AID, PID) \wedge Author(AID, 'Knuth', age))\}$$

Rules:

$$KnuthPapers(P) \leftarrow Write(A, P), Author(A, 'Knuth', age).$$

$$NoKnuthPapers(P) \leftarrow Paper(P, T, Y), \neg KnuthPapers(P).$$

Example

PID's of the papers that Knuth is NOT writing.

$$\pi_{PID} Paper - \pi_{PID}(Write \bowtie \sigma_{name='Knuth'} Author)$$

$$\{PID | \exists title, year.Paper(PID, title, year) \wedge \neg \exists age, AID.(Write(AID, PID) \wedge Author(AID, 'Knuth', age))\}$$

Rules:

$$KnuthPapers(P) \leftarrow Write(A, P), Author(A, 'Knuth', age).$$

$$NoKnuthPapers(P) \leftarrow Paper(P, T, Y), \neg KnuthPapers(P).$$

Example

PID's of the papers that Knuth is NOT writing.

$$\pi_{PID} Paper - \pi_{PID}(Write \bowtie \sigma_{name='Knuth'} Author)$$

$$\{PID \mid \exists title, year.Paper(PID, title, year) \land \\ \neg\exists age, AID.(Write(AID, PID) \land Author(AID, 'Knuth', age))\}$$

Rules:

$$KnuthPapers(P) \leftarrow Write(A, P), Author(A, 'Knuth', age).$$

$$NoKnuthPapers(P) \leftarrow Paper(P, T, Y), \neg KnuthPapers(P).$$

Example

PID's of the papers that Knuth is NOT writing.

$$\pi_{PID} Paper - \pi_{PID}(Write \bowtie \sigma_{name='Knuth'} Author)$$

$$\{PID | \exists title, year. Paper(PID, title, year) \wedge \\ \neg \exists age, AID.(Write(AID, PID) \wedge Author(AID, 'Knuth', age))\}$$

Rules:

$$KnuthPapers(P) \leftarrow Write(A, P), Author(A, 'Knuth', age).$$

$$NoKnuthPapers(P) \leftarrow Paper(P, T, Y), \neg KnuthPapers(P).$$

Example

AID's of authors who wrote exactly one paper

$$S2 = \rho_{Write}(AID1, PID1) \bowtie_{AID1=AID2, PID1 \neq PID2} \rho_{Write}(AID2, PID2)$$

$$S = \pi_{AID} Write - \pi_{AID1} S2$$

$$\{AID \mid \exists PID.(Write(AID, PID) \wedge \neg \exists PID2.(Write(AID, PID2) \wedge PID \neq PID2))\}$$

Rules:

$$MoreThanOne(A) \leftarrow Write(A, P1), Write(A, P2), P1 \neq P2.$$

$$OnlyOne(A) \leftarrow Write(A, P), \neg MoreThanOne(A).$$

Example

AID's of authors who wrote exactly one paper

$$S2 = \rho_{Write}(AID1, PID1) \bowtie_{AID1=AID2, PID1 \neq PID2} \rho_{Write}(AID2, PID2)$$

$$S = \pi_{AID} Write - \pi_{AID1} S2$$

$$\{AID | \exists PID.(Write(AID, PID) \wedge \neg \exists PID2.(Write(AID, PID2) \wedge PID \neq PID2))\}$$

Rules:

$$MoreThanOne(A) \leftarrow Write(A, P1), Write(A, P2), P1 \neq P2.$$

$$OnlyOne(A) \leftarrow Write(A, P), \neg MoreThanOne(A).$$

Example

AID's of authors who wrote exactly one paper

$$S2 = \rho_{Write}(AID1, PID1) \bowtie_{AID1=AID2, PID1 \neq PID2} \rho_{Write}(AID2, PID2)$$

$$S = \pi_{AID} Write - \pi_{AID1} S2$$

$$\{AID | \exists PID.(Write(AID, PID) \wedge \neg \exists PID2.(Write(AID, PID2) \wedge PID \neq PID2))\}$$

Rules:

$$MoreThanOne(A) \leftarrow Write(A, P1), Write(A, P2), P1 \neq P2.$$

$$OnlyOne(A) \leftarrow Write(A, P), \neg MoreThanOne(A).$$

Example

AID's of authors who wrote exactly one paper

$$S2 = \rho_{Write}(AID1, PID1) \bowtie_{AID1=AID2, PID1 \neq PID2} \rho_{Write}(AID2, PID2)$$

$$S = \pi_{AID} Write - \pi_{AID1} S2$$

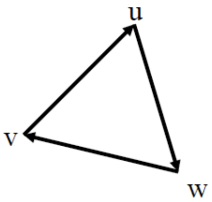
$$\{AID \mid \exists PID.(Write(AID, PID) \wedge \neg \exists PID2.(Write(AID, PID2) \wedge PID \neq PID2))\}$$

Rules:

$$MoreThanOne(A) \leftarrow Write(A, P1), Write(A, P2), P1 \neq P2.$$

$$OnlyOne(A) \leftarrow Write(A, P), \neg MoreThanOne(A).$$

Example: True or False?



E	from	to
	v	u
	u	w
	w	v

$$\forall x \exists y \, E(x, y)$$

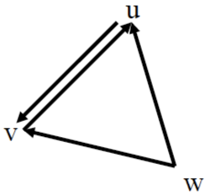
$$\forall x \exists y \exists z (E(z, x) \wedge E(x, y))$$

$$\exists x \forall y \exists z (E(z, x) \wedge E(x, y))$$

$$\forall x \exists y \exists z (\neg(y = z) \wedge E(x, y) \wedge E(x, z))$$

The above are Boolean queries

Example: True or False?



E	from	to
	u	v
	v	u
	w	u
	w	v

$$\forall x \exists y\, E(x,y)$$

$$\forall x \exists y \exists z (E(z,y) \wedge E(x,y))$$

$$\exists x \forall y \exists z (E(z,y) \wedge E(x,y))$$

$$\forall x \exists y \exists z (\neg(y = z) \wedge E(x,y) \wedge E(x,z))$$

$$\forall x \exists y (E(x,y) \wedge \exists z (E(x,z) \wedge \neg(y = z)))$$

RA to FO

$R(A, B) S(C, D)$

- $\sigma_{A=3} R$
  $\{(x, y) | R(x, y) \wedge x = 3\}$
  rule: $R'(x, y) \leftarrow R(x, y), x = 3$
- $\pi_A R$
  $\{x | \exists y. R(x, y)\}$
- $R \bowtie B$
  $R'(x, y, z) \leftarrow R(x, y), S(y, z)$
- union? difference?

FO to RA

$$S(x, y) \leftarrow \exists z(R(x, y) \wedge (P(y, z) \vee Q(z, x)))$$

Unsafe FO Queries

Sometimes FO queries can express queries which are 'unsafe'.

$$\{x \mid \forall y R(x, y)\}$$

- $R = \{(1,1)\}$, dom= $\{1,2\}$, $x = \emptyset$
- $R = \{(1,1)\}$, dom= $\{1\}$, $x = \{1\}$

What query returns depends on the domain, not on the database!

Unsafe FO Queries

Another 'unsafe' query.

$$\{x | \neg R(x)\}$$

- $R = \{(1)\}$, dom$= \{1, 2\}$, $x = \{2\}$
- $R = \{(1)\}$, dom$= \{1\}$, $x = \emptyset$

What query returns depends on the domain, not on the database!

Active Domain

- The active domain interpretation restricts quantified variables of the query to range over the active domain of the query and the input
- Given the database and the query, the domain is determined using the active domain semantics

$$R = \{(1,1)\}\ \text{dom}(R)=\{1,2\}$$
$$\textbf{adom} = \{1\}\ \text{not}\ \{1,2\}!$$
$$x = \{1\}$$

Active Domain

- The active domain interpretation is a viable solution. Under this interpretation, every query in FO is expressible in RA, and thus FO and RA have the same expressive Power: RA=FO (See AHV, Section 5).
- The same is true for safe, i.e., domain independent queries. But it is unfortunately undecidable whether a query is safe. (We will come back to this later)
- It is even true for a subset of the safe queries, the range restricted queries (see AHV)