Database Theory

VU 181.140, SS 2011

3. Codd's Theorem

Reinhard Pichler

Institut für Informationssysteme
Arbeitsbereich DBAI
Technische Universität Wien

29 March, 2011

Outline

What is Codd's Theorem?

- Domain Independence: Some queries are "unsafe" and must be avoided.
- Range Restriction: A syntactic condition for domain independence.
- Codd's Theorem: relational algebra and range-restricted calculus are equally expressive.
- We prove the theorem: show how to translate back and forth between the two languages.
  - use of relativization: bounding the ranges of quantifiers using finite domains.
- Strengthening of Codd's Theorem: the following languages are equally expressive:
  - domain independent calculus,
  - relational algebra,
  - non-recursive Datalog with negation, and
  - calculus under the active domain semantics.

Domain Independence

- Domain Independence: Given a query and a database, the query must evaluate to the same result on the database no matter what the domain is assumed to be.
- Idea: exclude "unsafe" queries, i.e., in particular, queries that may yield an infinite answer.
- Let $Q_B(\mathcal{A})$ denote the result of evaluating query Q on database $\mathcal{A}$ assuming domain B.

Definition (domain independence)

A query Q is *domain independent* iff there do **not** exist

- a database instance $\mathcal{A}$ and
- two sets B, C that contain all constants that appear in $\mathcal{A}$ or in Q (also known as the active domain),

such that $Q_B(\mathcal{A}) \neq Q_C(\mathcal{A})$.

Queries Violating Domain Independence

Example (Unsafe Queries)

- $\{x \mid \neg R(x)\}$
  - $R = \emptyset, 1 \in B, 1 \notin C$: $1 \in Q_B(R), 1 \notin Q_C(R)$.
- $\{x \mid R(x) \vee R(y)\}$
- $\{y \mid R(x)\}$
- $\{x \mid R(x) \vee \neg R(x)\}$

Remark. Over infinite domains, these queries may yield an infinite result.

Undecidability of Domain Independence

- We would like to require domain independence in queries.
- Domain independence is an undecidable property for FO queries.
- Usual solution: syntactic restrictions, e.g. range-restricted queries. Sufficient condition for domain independence.
- We shall use the following theorem without a proof:

Theorem (A)

For every domain-independent FO query, there is an equivalent range-restricted FO query.

- See Abiteboul, Hull, Vianu, "Foundations of Databases", Chapter 5.
- We show here that RR FO and rel. algebra have the same expressive power. For Theorem A, it suffices to show that also rel. algebra and domain-independent FO have the same expressive power.
- From rel. algebra to domain independent calculus: Lemma 5.3.11.
- From domain independent calculus via active domain semantics to rel. algebra: Lemma 5.3.12.

Range-restricted Queries

Preprocessing. A formula is turned into safe-range normal form (SRNF) by the following steps:

1. Variable substitution: no distinct pair of quantifiers may employ the same variable and no variable may occur both bound and free.
   Example: $(\exists x\ \varphi(x)) \vee (\exists x\ \psi(x)) \vdash (\exists x\ \varphi(x)) \vee (\exists x'\ \psi(x'))$.
2. Remove universal quantifiers: $\forall x\ \varphi \vdash \neg \exists x\ \neg \varphi$.
3. Remove implications: $\varphi \Rightarrow \psi \vdash \neg \varphi \vee \psi$.
4. Remove double negation: $\neg\neg\varphi \vdash \varphi$.
5. Flatten and/or, e.g.: $(\varphi \wedge \psi) \wedge \pi \vdash \varphi \wedge \psi \wedge \pi$.

Range-restriction Check

Definition

Given: Input formula π in SRNF.

$$
\begin{aligned}
rr(x = a) &:= \{x\} \\
rr(R(t_1, \ldots, t_n)) &:= Vars(\{t_1, \ldots, t_n\}) \\
rr(\varphi \wedge \psi) &:= rr(\varphi) \cup rr(\psi) \\
rr(\varphi \vee \psi) &:= rr(\varphi) \cap rr(\psi) \\
rr(\varphi \wedge x = y) &:= \begin{cases} rr(\varphi) & \ldots \quad \{x, y\} \cap rr(\varphi) = \emptyset \\ rr(\varphi) \cup \{x, y\} & \ldots \quad \text{otherwise} \end{cases} \\
rr(\neg\varphi) &:= \emptyset \\
rr(\exists x\, \varphi) &:= \begin{cases} rr(\varphi) - \{x\} & \ldots \quad x \in rr(\varphi) \\ \text{fail} & \ldots \quad \text{otherwise} \end{cases}
\end{aligned}
$$

If free(π) $\subseteq rr(\pi)$, then π is range-restricted (RR).

Range-restriction Examples

Example (in SRNF)

$$\overbrace{\exists z : \overbrace{\overbrace{P(x,y,z)}^{rr(\cdot)=\{x,y,z\}} \vee \overbrace{(\overbrace{R(x,y)}^{rr(\cdot)=\{x,y\}} \wedge \overbrace{(\overbrace{(\overbrace{S(z)}^{rr(\cdot)=\{z\}} \wedge \overbrace{\neg T(x,z)}^{rr(\cdot)=\emptyset})}^{rr(\cdot)=\{z\}} \vee T(y,z))}^{rr(\cdot)=\{z\}})}^{rr(\cdot)=\{x,y,z\}}}^{rr(\cdot)=\{x,y,z\}}}^{rr(\cdot)=\{x,y\}}$$

$rr(*) = \text{free}(*) = \{x, y\} \Rightarrow$ range-restricted!

Main Theorem

Theorem (Codd)

A query is definable in relational algebra if and only if it is definable in the range-restricted domain calculus.

Proof (details on the following slides):

1. From the Algebra to RR FO: almost by definition.
2. From RR FO to the Algebra:
    1. Put into SRNF.
    2. Put into Relational-Algebra NF (RANF):
        - RANF: Each subformula is range-restricted. (Exception: In a subformula $\pi = \varphi_1 \wedge \cdots \wedge \varphi_k \wedge \neg\psi$, π has to be RR and the φ_i and ψ have to be in RANF, but $\neg\psi$ does not have to be RR).
    3. Translate the RANF formula into relational algebra. This can be done inductively from the leaves to the root of the parse tree of the formula.

Query languages of this expressive power are called relationally complete.

From the Algebra to RR FO

$$\begin{aligned}
R &:= \{\vec{x} \mid R(\vec{x})\} \\
\sigma_{x=y}(\{\vec{x} \mid \varphi(\vec{x})\}) &:= \{\vec{x} \mid \varphi(\vec{x}) \wedge x = y\} \\
\pi_{\vec{y}}(\{\vec{x} \mid \varphi(\vec{x})\}) &:= \{\vec{y} \mid \exists \vec{z}\, \varphi(\vec{x})\} \quad (\vec{x} = \vec{y}\vec{z}) \\
\{\vec{x} \mid \varphi(\vec{x})\} \times \{\vec{y} \mid \psi(\vec{y})\} &:= \{\vec{x}\vec{y} \mid \varphi(\vec{x}) \wedge \psi(\vec{y})\} \\
\{\vec{x} \mid \varphi(\vec{x})\} \cup \{\vec{x} \mid \psi(\vec{x})\} &:= \{\vec{x} \mid \varphi(\vec{x}) \vee \psi(\vec{x})\} \\
\{\vec{x} \mid \varphi(\vec{x})\} - \{\vec{x} \mid \psi(\vec{x})\} &:= \{\vec{x} \mid \varphi(\vec{x}) \wedge \neg\psi(\vec{x})\}
\end{aligned}$$

From SRNF to RANF: Active Domain

- Active domain contains precisely the union of all the atomic/field values occurring anywhere in the database (or the query).
- A unary active domain relation can be assumed: It is definable in relational algebra.

Example

For schema $R(A, B, C), S(B, D), T(E)$, the active domain relation is

$$\pi_A R \cup \pi_B R \cup \pi_C R \cup \pi_B S \cup \pi_D S \cup T.$$

- We assume a single domain here; in the real-world case with values of different types (e.g. strings, integers), we can compute an active domain relation for each type.

From SRNF to RANF

Given a formula in SRNF.

- The formula is in RANF if each subformula is range-restricted.
- Possible obstacles: subformulae of the form $\varphi \lor \psi$ or $\neg\varphi$.
- Only these remove possibly relevant variables from rr.
- Solution: relativize using the active domain relation D:

$$\varphi \lor \psi \quad \vdash \quad \underbrace{(\varphi \land \bigwedge_{x \in (\mathrm{free}(\psi) - \mathrm{free}(\varphi))} D(x)) \lor (\psi \land \bigwedge_{x \in (\mathrm{free}(\varphi) - \mathrm{free}(\psi))} D(x))}_{rr(*) = \mathrm{free}(*) = \mathrm{free}(\varphi) \cup \mathrm{free}(\psi)}$$

$$\neg\varphi \quad \vdash \quad (\neg\varphi \land \bigwedge_{x \in \mathrm{free}(\varphi)} D(x))$$

- Shorter (=better) RANF queries can be achieved by rewriting the input formula using equivalences we already know.

From RANF to Relational Algebra

RANF formulae can be translated to relational algebra using the following rules:

$$
\begin{array}{rcl}
Alg(\varphi \wedge \psi) & := & Alg(\varphi) \bowtie Alg(\psi) \\
Alg(\varphi \wedge \neg\psi) & := & Alg(\varphi) - Alg(\psi) \\
 & \ldots & \varphi \text{ and } \psi \text{ have the same schema} \\
Alg(\exists y\ \varphi(\vec{x}, y)) & := & \pi_{\vec{x}} Alg(\varphi(\vec{x}, y)) \\
Alg(\varphi \wedge x \vartheta t) & := & \sigma_{x \vartheta t} Alg(\varphi) \\
 & \ldots & \vartheta \text{ is either } = \text{ or } \neq \text{ and } t \text{ is a term.} \\
Alg(R(x_1, \ldots, x_k)) & := & \rho_{A_1 \ldots A_k \to x_1 \ldots x_k} R \\
 & \ldots & \text{relation } R \text{ has schema } R(A_1, \ldots, A_k)
\end{array}
$$

From RR Queries to the Algebra

Example

Let D be the active domain relation with schema $D(D)$ and let R have schema $R(A, B)$. The formula

$$\exists x \left(D(x) \land \exists y \left(D(y) \land \neg R(x, y)\right)\right)$$

corresponds to the RANF formula

$$\exists x \, \exists y \left((D(x) \land D(y)) \land \neg R(x, y)\right).$$

An equivalent relational algebra expression looks as follows.

$$\begin{aligned} & Alg(\exists x \, \exists y \, ((D(x) \land D(y)) \land \neg R(x, y))) \\ \vdash \quad & \pi_{\emptyset}\left(Alg((D(x) \land D(y)) - Alg(R(x, y))\right) \\ \vdash \quad & \pi_{\emptyset}(\rho_x D \bowtie \rho_y D - \rho_{xy} R) \end{aligned}$$

From RR Queries to the Algebra

Example

In SRNF but not in RANF (i.e., not locally range-restricted):

$$\{(x,y) \mid \exists z : P(x,y,z) \vee (R(x,y) \wedge (\overbrace{(S(z) \wedge \neg T(x,z))}^{rr(*)=\{z\}} \vee T(y,z)))\}$$

We transform this formula using the rewrite rule

$$\varphi \wedge (\psi_1 \vee \psi_2) \vdash (\varphi \wedge \psi_1) \vee (\varphi \wedge \psi_2)$$

into RANF:

$$\{(x,y) \mid \exists z : P(x,y,z) \vee (R(x,y) \wedge S(z) \wedge \neg T(x,z)) \vee \\ (R(x,y) \wedge T(y,z))\}$$

From RR Queries to the Algebra

Example (in RANF)

$$\{(x,y) \mid \exists z : \overbrace{P(x,y,z)}^{e_1=\rho_{xyz} P} \lor$$

$$\overbrace{\overbrace{(\overbrace{R(x,y)}^{\rho_{xy} R} \land \overbrace{S(z)}^{\rho_z S}}^{(\cdot)\bowtie(\cdot)} \land \neg \overbrace{T(x,z)}^{e_{21}=\rho_{xz} T})}^{e_2=(\cdot)-((\rho_y D)\bowtie(\cdot))} \lor \overbrace{(\overbrace{R(x,y)}^{\rho_{xy} R} \land \overbrace{T(y,z)}^{\rho_{yz} T})}^{e_3=(\cdot)\bowtie(\cdot)}\}$$

Equivalent relational algebra expression: $\pi_{xy}(e_1 \cup e_2 \cup e_3)$.

From RR Queries to the Algebra

Example

"Select all professors (P) who only give lectures (L) in the field of computer science (C)." The schemata are $P(P), L(P,C), C(C)$ and the active domain is given by a relation D with schema $D(D)$.

$$\{x \mid P(x) \wedge \forall y : L(x,y) \rightarrow C(y)\}$$

$$\overset{\text{to SRNF}}{\vdash} \quad \{x \mid P(x) \wedge \neg\exists y : L(x,y) \wedge \neg C(y)\}$$

$$\overset{\text{to RANF}}{\vdash} \quad \{x \mid \underbrace{P(x) \wedge \neg\, \underbrace{\exists y : \underbrace{L(x,y) \wedge (\underbrace{D(y) \wedge \neg C(y)}_{(\rho_C D)-C})}_{L \bowtie ((\rho_C D)-C)}}_{\pi_P(L \bowtie ((\rho_C D)-C))}}_{P-\pi_P(L \bowtie ((\rho_C D)-C))}\}$$

From RR Queries to the Algebra

Example

The schemata are $L(P, C), C(C)$ and the domain is given by $D(D)$.

$$\begin{aligned}
\{\langle x,y\rangle \mid L(x,y) \wedge \neg C(y)\} &\equiv \{\langle x,y\rangle \mid \underbrace{L(x,y) \wedge (D(y) \wedge \neg C(y))}_{L \bowtie ((\rho_C D) - C)}\} \\
&\equiv \{\langle x,y\rangle \mid \underbrace{L(x,y) \wedge \neg(D(x) \wedge C(y))}_{L - ((\rho_C D) \times C)}\}
\end{aligned}$$

D	D
	1
	2
	3
	4

L	P	C
	1	2
	3	4

C	C
	2

$(\rho_C D) - C$	C
	1
	3
	4

$(\rho_C D) \times C$	P	C
	1	2
	2	2
	3	2
	4	2

$L \bowtie (\rho_C D) - C$	P	C
	3	4

From RR Queries to the Algebra

Example

Schema $R(AB)$, $S(C)$, $T(AC)$; active domain $D(D)$.

$$
\begin{array}{rl}
 & \{\langle x, y, z\rangle \mid R(x, y) \wedge (S(z) \wedge \neg T(x, z))\} \\
\vdash & \{\langle x, y, z\rangle \mid \underbrace{R(x, y) \wedge (\underbrace{\underbrace{(D(x) \wedge S(z))}_{D \times S} \wedge \neg T(x, z)}_{(D \times S) - T})}_{R \bowtie ((D \times S) - T)}\}
\end{array}
$$

From RR Queries to the Algebra

Example

Schema $R(AB)$, $S(AC)$, $T(BC)$; active domain $D(D)$.

$$\begin{aligned} & \{\langle x,y,z\rangle \mid R(x,y) \wedge (S(x,z) \vee T(y,z))\} \\ \vdash\quad & \{\langle x,y,z\rangle \mid \underbrace{\underbrace{(R(x,y) \wedge S(x,z))}_{R \bowtie S} \vee \underbrace{(R(x,y) \wedge T(y,z))}_{R \bowtie T}}_{(R \bowtie S) \cup (R \bowtie T)}\} \end{aligned}$$

$$\begin{aligned} \text{or}\quad & \{\langle x,y,z\rangle \mid R(x,y) \wedge (S(x,z) \vee T(y,z))\} \\ \vdash\quad & \{\langle x,y,z\rangle \mid \underbrace{R(x,y) \wedge ((S(x,z) \wedge D(y)) \vee (T(y,z) \wedge D(x)))}_{R \bowtie ((S \times \rho_B D) \cup (T \times D))}\} \end{aligned}$$

This is correct because
$R(x,y) \wedge (\varphi \vee \psi) \equiv R(x,y) \wedge D(x) \wedge D(y) \wedge (\varphi \vee \psi) \equiv$
$R(x,y) \wedge ((D(x) \wedge D(y) \wedge \varphi) \vee (D(x) \wedge D(y) \wedge \psi))$.

Strengthening of Codd's Theorem

Theorem

Domain independent calculus and relational algebra are equally expressive.

- A direct consequence of Theorem (A) and Codd's Theorem.
- In fact, "Codd's Theorem" usually refers to the above formulation.
- By the above theorem and a transtion via relational algebra the following holds:

Theorem

Domain independent calculus and aggregation-free SQL queries are equally expressive.

Strengthening of Codd's Theorem (2)

- Nonrecursive Datalog with negation (nr-$Datalog^{\neg}$) prohibits cycles in the dependency graph $DEP(P)$ of a program P.
  - nonrecursiveness means that negation is trivially stratified!

Theorem

Domain independent domain calculus and non-recursive Datalog with negation are equally expressive.

- Easiest way to see: provide a translation between nr-$Datalog^{\neg}$ and relational algebra; then apply the theorem from the previous slide.
- See Abiteboul, Hull, Vianu, "Foundations of Databases", Chapter 5.

Strengthening of Codd's Theorem (3)

- Active domain semantics for domain calculus: quantified variables range over the active domain, i.e. over values occurring in the database (or the query).

Example

- Assume a database D with $R = \{\langle 1, 1\rangle\}$ and $dom(R) = \{1, 2\}$.
- Consider the Boolean query $q = \{\langle\rangle \mid \forall x, y.R(x, y).\}$:
  - q is false in D under the standard semantics, but
  - q is true in D under the active domain semantics.

Theorem

Domain independent domain calculus and domain calculus under the active domain semantics are equally expressive.

- See Abiteboul, Hull, Vianu, "Foundations of Databases", Chapter 5.

Learning objectives

Understanding:

- the notion of domain independence,
- the active domain,
- the notion of range restricted queries,
- the formulation and the proof of Codd's Theorem,
- the strengthening of Codd's Theorem.