

The semijoin algebra

Dirk Leinders

Hasselt University

DBDBD 2007

Outline

Codd theorem for the semijoin algebra

Linear processing

One- and two-pass processing

The relational algebra (RA)

Operations

- ▶ selection $\sigma_{A=B}, \sigma_{A \neq B}, \sigma_{A < B}$
- ▶ projection π_{A_1, \dots, A_k}
- ▶ union $R \cup S$
- ▶ difference $R - S$
- ▶ natural join $R \bowtie S$
- ▶ renaming $\rho_{A \rightarrow B}$

The relational algebra (RA)

Example

Likes(drinker,beer), Serves(bar,beer), Visits(drinker,bar)

Lousy bars:

$$\pi_{\text{bar}}\text{Serves} - \pi_{\text{bar}}(\text{Serves} \bowtie \text{Likes})$$

Theorem (Codd)

RA is equivalent to first-order logic (relational calculus).

The semijoin algebra (SA)

Operations

- ▶ all operations of RA, except for join \bowtie
- ▶ replaced by semijoin \ltimes

$$\begin{aligned} R \ltimes S &:= \{r \in R \mid \exists s \in S : r \text{ and } s \text{ join}\} \\ &= \pi_R(R \bowtie S) \end{aligned}$$

The semijoin algebra (SA)

Example: lousy bars

$$\pi_{\text{bar}} \text{Serves} - \pi_{\text{bar}}(\text{Serves} \bowtie \text{Likes})$$

SA strictly less powerful than RA

Drinkers that visit a bar that serves a beer they like:

$$\pi_{\text{drinker}}(\text{Visits} \bowtie \text{Likes} \bowtie \text{Serves})$$

Expressive power of SA?

$$\begin{array}{ccc} \text{SA} & \subset & \text{RA} \\ \parallel & & \parallel \\ ? & \subset & \text{FO} \end{array}$$

The semijoin algebra (SA)

Example: lousy bars

$$\pi_{\text{bar}} \text{Serves} - \pi_{\text{bar}}(\text{Serves} \bowtie \text{Likes})$$

SA strictly less powerful than RA

Drinkers that visit a bar that serves a beer they like:

$$\pi_{\text{drinker}}(\text{Visits} \bowtie \text{Likes} \bowtie \text{Serves})$$

Expressive power of SA?

$$\begin{array}{ccc} \text{SA} & \subset & \text{RA} \\ \parallel & & \parallel \\ ? & \subset & \text{FO} \end{array}$$

The semijoin algebra (SA)

Example: lousy bars

$$\pi_{\text{bar}} \text{Serves} - \pi_{\text{bar}}(\text{Serves} \bowtie \text{Likes})$$

SA strictly less powerful than RA

Drinkers that visit a bar that serves a beer they like:

$$\pi_{\text{drinker}}(\text{Visits} \bowtie \text{Likes} \bowtie \text{Serves})$$

Expressive power of SA?

$$\begin{array}{ccc} \text{SA} & \subset & \text{RA} \\ \parallel & & \parallel \\ ? & \subset & \text{FO} \end{array}$$

The semijoin algebra (SA)

Example: lousy bars

$$\pi_{\text{bar}} \text{Serves} - \pi_{\text{bar}}(\text{Serves} \bowtie \text{Likes})$$

SA strictly less powerful than RA

Drinkers that visit a bar that serves a beer they like:

$$\pi_{\text{drinker}}(\text{Visits} \bowtie \text{Likes} \bowtie \text{Serves})$$

Expressive power of SA?

$$\begin{array}{ccc} \text{SA} & \subset & \text{RA} \\ \parallel & & \parallel \\ ? & \subset & \text{FO} \end{array}$$

The guarded fragment (GF)

Definition

- ▶ quantifier-free formulas are in GF
- ▶ $\varphi \wedge \psi, \quad \neg\varphi$
- ▶ restriction on quantifiers:

$$\begin{aligned} & \exists \bar{y}(\alpha(\bar{x}, \bar{y}) \wedge \psi(\bar{x}, \bar{y})) \\ & \forall \bar{y}(\alpha(\bar{x}, \bar{y}) \rightarrow \psi(\bar{x}, \bar{y})) \end{aligned}$$

where α is a single relation and all free variables of ψ must occur in α .

Example: lousy bars

$\{\text{bar} \mid \neg \exists \text{beer}(\text{Serves}(\text{bar}, \text{beer}) \wedge \exists \text{drinker Likes}(\text{drinker}, \text{beer}))\}$

Codd theorem

Theorem

$GF \equiv SA$

Consequences

- ▶ SA has the finite model property
- ▶ satisfiability of SA expressions is EXPTIME-complete

Fixed point extension

$\mu SA \equiv \mu GF$

Example: reachability in μSA

Database relations $R(A, B)$ and $S(A)$,
relation variable $X(A)$:

$$\text{LFP } X. S \cup \rho_{B \rightarrow A} \pi_B (R \ltimes_{R.A=X.A} X)$$

Application 1

Prove that query is not expressible in SA

- ▶ \Rightarrow prove that query is not expressible in GF
- ▶ Ex. drinkers that visit a bar that serves a beer they like
- ▶ GF is invariant under **guarded bisimilarity**, \simeq_g

Guarded bisimilarity

Definition

Databases A and B , same schema, tuple \bar{a} in A , tuple \bar{b} in B
 $(A, \bar{a}) \simeq_g (B, \bar{b})$ if player II can keep up forever in the following game:

1. initial game position is (A, \bar{a}) and (B, \bar{b})
2. player I chooses a tuple in one of the databases, say A
3. player II responds in other database $\Rightarrow (A, \bar{a}')$ and (B, \bar{b}')
 - ▶ \bar{a}' and \bar{b}' must satisfy precisely same relations, predicates
 - ▶ if \bar{a} and \bar{a}' agree in i th position, then \bar{b} and \bar{b}' must too
4. if player II cannot respond correctly he loses;
otherwise repeat from new position (A, \bar{a}') and (B, \bar{b}') .

Application 1

Prove that query is not expressible in SA

- ▶ \Rightarrow prove that query is not expressible in GF
- ▶ Ex. drinkers that visit a bar that serves a beer they like
- ▶ GF is invariant under **guarded bisimilarity**, \simeq_g

Example

A

Visits(alex, pareto bar)
Serves(pareto bar, westmalle)
Likes(alex, westmalle)

B

Visits(alex, pareto bar)
Visits(bart, qwerty bar)
Serves(pareto bar, westmalle)
Serves(qwerty bar, westvleteren)
Likes(alex, westvleteren)
Likes(bart, westmalle)

Application 2: Linear query processing

Linear RA expression

= on every database, every intermediate result has linear size

linear: $(\sigma R \cup \pi S) - T$

not linear: $R \cap (S \bowtie T)$

linear: $R(A, B) \bowtie \pi_B(S) = R \bowtie S$

Queries expressible in SA are always linear

Other linear queries? NO.

Theorem

Every query expressible by a linear RA expression is already expressible by an SA expression

Relational division

$$R(A, B) \div S(C) := \{a \mid \forall b \in S : (a, b) \in R\}$$

RA expressions for division are inefficient:

$$R \div S = \pi_A R - \pi_A((\pi_A R \bowtie \rho_{C \rightarrow B} S) - R)$$

A			B	
R	S		R	S
1 7	7		1 7	7
1 8	8		1 8	8
2 7			2 8	9
2 8			2 9	
			3 7	
			3 9	

⇒ There is no linear RA expression for division.

Set equality/containment join

Division is a restricted kind of *set join*.

Set join

- ▶ Let $P(X, Y)$ be a predicate about sets.
For relations $R(A, B)$ and $S(C, D)$:

$$R \bowtie_P^{\text{set}} S := \{a, c \mid P(\{b : R(a, b)\}, \{d : S(c, d)\})\}$$

- ▶ set-containment join: $\bowtie_{X \subseteq Y}^{\text{set}}$
- ▶ set-equality join: $\bowtie_{X=Y}^{\text{set}}$
- ▶ standard equijoin: $\bowtie_{X \cap Y \neq \emptyset}^{\text{set}}$

\Rightarrow There is no linear RA expression for “Is the set equality/containment join nonempty?”

Outline

Codd theorem for the semijoin algebra

Linear processing

One- and two-pass processing

One-pass and two-pass processing

one-pass

- ▶ selection
- ▶ projection
- ▶ union (no duplicate elimination)

two-pass

- ▶ difference
- ▶ semijoin

Finite Cursor Machines (FCM)

- ▶ work on relations: lists of tuples
- ▶ fixed number of cursors on each relation
- ▶ cursors are 1-way
- ▶ fixed number of registers, store bitstrings
- ▶ finite state control
- ▶ built-in bitstring functions on data elements & bitstrings

Sorting is needed

Theorem

$R - S$ can not be computed by an $o(n)$ -FCM.

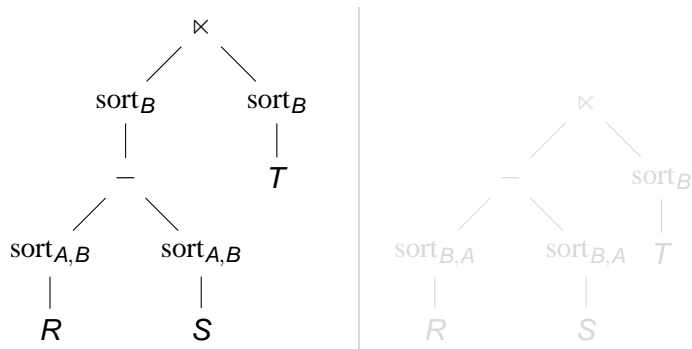
$R \times S$ can not be computed by an $o(n)$ -FCM.

Theorem

Every SA query can be computed by a query plan composed of FCMs and sorting operations

Intermediate sorting

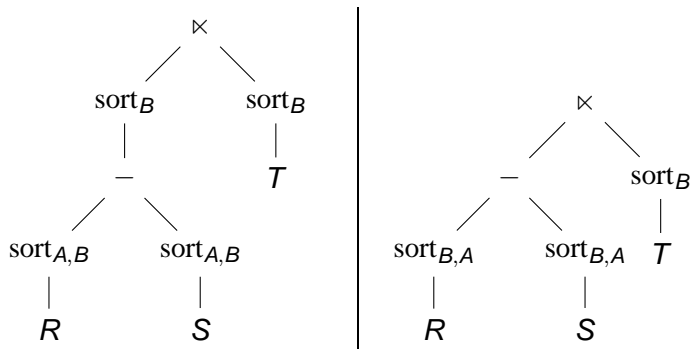
Example $(R(A, B) - S(A, B)) \times T(B, C)$



\Rightarrow Can intermediate sorting always be avoided?

Intermediate sorting

Example $(R(A, B) - S(A, B)) \times T(B, C)$



\Rightarrow Can intermediate sorting always be avoided?

Intermediate sorting is needed

Theorem

- ▶ $RST\text{-query} = R(A) \times (S(A, B) \times T(B))$
- ▶ *emptiness test of RST-query is **not** computable by an $o(n)$ -FCM on sorted inputs (ascending and descending orders)!*

Conclusion

Codd theorem

SA equals GF, satisfiability EXPTIME-complete

? θ -semijoins

Linear processing




SA equals linear RA, division and set containment/equality joins are not linear

? which set joins are linear

One- and two-pass query processing

Two-pass processing, re-sorting really needed

References

-  D. Leinders, M. Marx, J. Tyszkiewicz, and J. Van den Bussche.
The semijoin algebra and the guarded fragment.
Journal of Logic, Language and Information,
14(3):331–343, 2005.
-  D. Leinders and J. Van den Bussche.
On the complexity of division and set joins in the relational algebra.
PODS, pages 76–83, 2005.
-  M. Grohe, Y. Gurevich, D. Leinders, N. Schweikardt, J. Tyszkiewicz, and J. Van den Bussche.
Database query processing using finite cursor machines.
ICDT, pages 284–298. Springer, 2007.