The Semijoin Algebra

Jan Van den Bussche
Hasselt University

joint work with Dirk Leinders, Maarten Marx, Jerzy Tyszkiewicz

The relational algebra, RA

Projection $\pi_{A,B,C}$

- allow repetitions: $\pi_{A,A/B}$

Selection $\sigma_{A=B}$, $\sigma_{A<B}$

Renaming ρ_R

Union, intersection, difference

Equijoin $R \underset{\substack{R.A=S.B \\ R.C=S.D}}{\bowtie} S$

- special cases: natural join, cartesian product

RA expressions

Build up expressions for complex queries

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

Losers:

$$\pi_{V.d}(V) - \pi_{V.d}\sigma_{V.d=L.d}(V \bowtie S \bowtie L)$$

Codd's theorem: RA is equivalent to first-order logic (relational calculus)

The semijoin algebra, SA

Equi-semijoin:

$$
\begin{aligned}
R \underset{\theta}{\ltimes} S &:= \{t \in R \mid \exists s \in S : \theta(t, s) \text{ is true}\} \\
&= \pi_R(R \underset{\theta}{\bowtie} S)
\end{aligned}
$$

with θ a conjunction of equalities

SA is RA where we replace $\bowtie$ by $\ltimes$

Visitors of lousy bars:

$$V \ltimes (\pi_{bar}(S) - \pi_{bar}(S \ltimes L))$$

The guarded fragment of first-order logic, GF

[Andréka, van Benthem, Németi]

Quantifiers are restricted to the following form:

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

- α atomic formula (single relation)

- all free variables of ψ must occur in α

Visitors of lousy bars:

$$\{d, ba \mid V(d, ba) \wedge \neg \exists be(S(ba, be) \wedge \exists d\, L(d, be))\}$$

Originally introduced in the context of modal, algebraic logic

Codd theorem for SA

$\boxed{\text{SA is equivalent to GF}}$

From SA to GF:

$$V \ltimes (\pi_{bar}(S) - \pi_{bar}(S \ltimes L))$$
$$V \ltimes (\pi_{bar}(S) - \pi_{bar}\{ba, be \mid S(ba, be) \wedge \exists d\, L(d, be)\})$$
$$V \ltimes (\pi_{bar}(S) - \{ba \mid \exists be(S(ba, be) \wedge \exists d\, L(d, be))\})$$
$$V \ltimes (\{ba \mid \exists be\, S(ba, be)\} - \{ba \mid \exists be(S(ba, be) \wedge \exists d\, L(d, be))\})$$
$$V \ltimes (\{ba \mid \exists be\, S(ba, be) \wedge \neg\exists be(S(ba, be) \wedge \exists d\, L(d, be))\})$$
$$\{d, ba \mid V(d, ba) \wedge \exists be\, S(ba, be) \wedge \neg\exists be(S(ba, be) \wedge \exists d\, L(d, be))\})$$

From GF to SA:

$$\{d, ba \mid V(d, ba) \wedge \neg\exists be(S(ba, be) \wedge \exists d\, L(d, be))\}$$

$$\{d, ba \mid V(d, ba) \wedge \neg\exists be(S(ba, be) \wedge be \in \pi_{be}(L))\}$$

$$\{d, ba \mid V(d, ba) \wedge \neg\exists be((ba, be) \in S \ltimes \pi_{be}(L))\}$$

$$\{d, ba \mid V(d, ba) \wedge \neg(ba \in \pi_{ba}(S \ltimes \pi_{be}(L))\}$$

$$\{d, ba \mid V(d, ba) \wedge ba \in (\pi_{ba}(S) - \pi_{ba}(S \ltimes \pi_{be}(L))\}$$

$$V \ltimes (\pi_{ba}(S) - \pi_{ba}(S \ltimes \pi_{be}(L))$$

Consequences of SA = GF

Equivalance extends to fixpoint logic: μSA $= \mu$GF

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

$$\mathsf{LFP}\, X.\, (R \ltimes T) \cup (R \underset{R.B=X.A}{\ltimes} X)$$

SA has the finite model property

Our translation SA $\rightarrow$ GF is exponential; still:

- Satisfiability of SA-expressions is decidable (complete for EXPTIME)

Polynomial translation SA $\rightarrow$ GF?

Guarded bisimilarity

GF is invariant under *guarded bisimilarity*, $\simeq_g$

Databases A and B, same schema, tuple $\bar{a}$ in A, tuple $\bar{b}$ in B

Def. $(A, \bar{a}) \simeq_g (B, \bar{b})$ 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 $(B, \bar{b}')$

- $\overline{a}'$ and $\overline{b}'$ must satisfy precisely same relations, predicates
- if $\overline{a}$ and $\overline{a}'$ agree in ith position, then $\overline{b}$ and $\overline{b}'$ must too

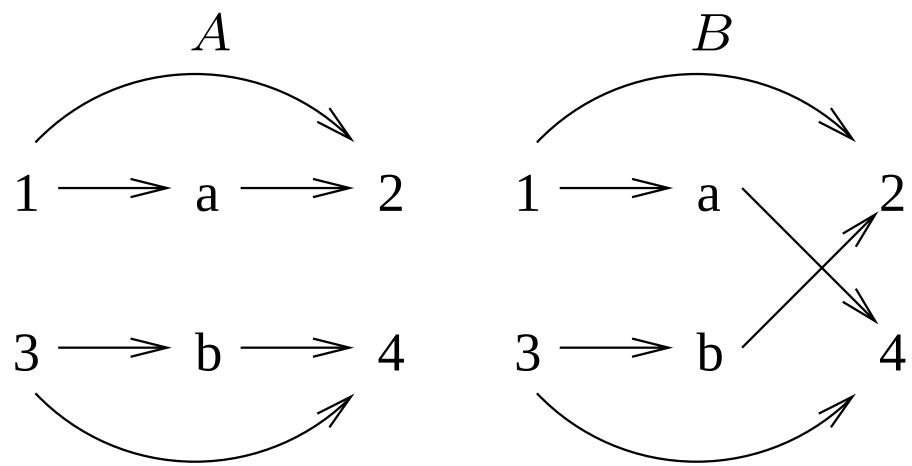
4. if player II cannot respond correctly he looses;
   otherwise repeat from new position $(A, \overline{a}')$ and $(B, \overline{b}')$.

Invariance property

If $(A, \bar{a}) \simeq_g (B, \bar{b})$ then for all SA-expressions E: $$\bar{a} \in E(A) \quad \Leftrightarrow \quad \bar{b} \in E(B)$$

Use to prove SA-inexpressibility of certain queries

Ex. single relation R:



$\Rightarrow$ "R is transitive" not SA-expressible

Division

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

RA-expressible, but not SA:

A

R

1	7
1	8
2	7
2	8

S

7
8

B

R

1	7
1	8
2	8
2	9
3	7
3	9

S

7
8
9

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: $R \underset{R.A=S.B}{\bowtie} \pi_B(S) = R \underset{R.A=S.B}{\ltimes} S$

$\boxed{\text{Every query expressible by a linear RA expression is already expressible by a SA expression}}$

Note that SA-expressions are always linear

Proof idea

For $E_1 \bowtie_\theta E_2$ to be linear, every joining tuple pair $(\bar{a}, \bar{b})$ must satisfy $\forall i \exists j : a_i = b_j$ or vice versa

- if not, we could "blow up" the database by duplicating the "free" values in $\bar{a}$ and $\bar{b}$

- blown up database is guarded bisimilar

- since E_1 and E_2 can be assumed linear by induction, they will output the duplicate tuples $\Rightarrow$ quadratic join size

Such joins can be expressed in SA

Corollaries

Every RA-expression is either linear, or has a subexpression that has quadratic output size

Every RA-expression either produces quadratic intermediate results, or is equivalent to an SA-expression

Set joins

We now know that division is not expressible in linear RA

Division is a restricted kind of *set join*

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

$$R \underset{P}{\bowtie}^{\mathsf{set}} S := \{a, c \mid P(\{b : R(a,b)\}, \{d : S(c,d)\})\}$$

subset join: $\bowtie^{\mathsf{set}}_{X \subseteq Y}$

set-equality join: $\bowtie^{\mathsf{set}}_{X=Y}$

standard equijoin: $\bowtie^{\mathsf{set}}_{X \cap Y \neq \emptyset}$

$\boxed{\text{If the emptiness query for } \bowtie^{\mathsf{set}}_{P} \text{ can be expressed in linear RA, then } P \text{ must be monotone}}$

Sidenote: Grouping and aggregation

It is well known that division can be linearly expressed using counting:

$$R(A,B) \div S(C) =$$
$$\pi_A(\gamma_{A,\mathsf{count}(B)}(R \ltimes_{B=C} S) \underset{\mathsf{count}(B)=\mathsf{count}(C)}{\ltimes} \gamma_{\mathsf{count}(C)}(S))$$

Theta-equijoins

$\ltimes_\theta$ with θ *more* than just conjunction of equalities?

Ex. "$R(A,B) \models A \to B$":

$$R \underset{\substack{R.A=S.A \\ R.B\neq S.B}}{\ltimes} \rho_S(R)$$

Satisfiability of SA$^{\neq}$ is undecidable

Do our linearity results extend to SA$^{\neq}$? and to SA$^{<}$?

Ω-guarded bisimilarity

Ω, signature of predicates that can be used in θ

SA$^{=}$: if $\bar{a}$ and $\bar{a}'$ agree in ith position, then $\bar{b}$ and $\bar{b}'$ must too

SA$^{\Omega}$: $(\bar{a}, \bar{a}')$ and $(\bar{b}, \bar{b}')$ must satisfy precisely the same predicates from Ω

Conclusion

SA = GF but SA^{Ω} is more powerful

SA = linear RA

Division, set joins not linear RA

- theoretical explanation why these queries are hard on the query processor

Many open problems remain