On Provenance Minimization

Yael Amsterdamer[2] **Daniel Deutch[3]** **Tova Milo[2]** **Val Tannen[1]**

[1] University of Pennsylvania
[2] Tel Aviv University
[3] Ben Gurion University

On Core Provenance

- Provenance Polynomials [Green, Karvounarakis, Tannen '07] represent the computation leading to output tuples of a DB query.
- A query may be computed in different ways.
- Different computations may have different provenance
- We want to find the ***core* provenance** – the part of provenance which is common to all possible query plans.

First Example

- Consider the equivalent queries:

Q_1: $\quad Ans(x) := R(x,y), R(y,x)$

Q_2: $\quad Ans(x) := R(x,y), R(y,x), x \neq y$

$\quad \cup \; Ans(x) := R(x,x)$

Q_3: $\quad Ans(x) := R(x,y), R(y,x), R(x,z), x \neq y, x \neq z$

$\quad \cup \; Ans(x) := R(x,x)$

- We apply the three on relation R:

A	B	Provenance
a	a	s_1
a	b	s_2
b	a	s_3

Computing the Output with Provenance

Query 1

Ans(x) := R(x,y),R(y,x)

Assignment 1: (a) (a,b) (b,a)

$s_2 \cdot s_3$

The input R

A	B	Provenance
a	a	s_1
a	b	s_2
b	a	s_3

The Output

A	Provenance
a	$s_2 \cdot s_3 + s_1 \cdot s_1$

- joint derivation

\+ alternative derivation

Comparing the Outputs

- The output tuple of 3 queries is the same, but the computation is different .
- Thus, the output provenance is different

	A	Provenance
Query 1	a	$s_2 \cdot s_3 + s_1 \cdot s_1$
Query 2	a	$s_2 \cdot s_3 + s_1$
Query 3	a	$s_2 \cdot s_3 \cdot s_2 + s_2 \cdot s_3 \cdot s_3 + s_1$

Why core provenance?

- Captures the "tersest" computation.
- Informative - describes the part of provenance which is inherent to the query.
- It is contained in the provenance of all equivalent queries, thus it is minimal.
  - Compact input to provenance management tools.

Main Goal

- Algorithms for computing, given a query, an equivalent query whose provenance is the core, a provenance minimal (p-minimal) query
  - Is there always such a query? (Spoiler: No!)
  - We study the problem for different classes of queries of increasing expressiveness: CQ, $CQ^{\neq}$, $UCQ^{\neq}$…

Comparing Provenances

- We do this using an order relation which reflects relative “terseness” of provenance polynomials
- **Monomials:** we say that $m \leq m'$ if the multiplicands of m are bag-included in those of m'.
- **Polynomials:** we say that $p \leq p'$ if there is an injective mapping $I: p \rightarrow p'$ s.t. $m \leq I(m)$.
- Example:

$p_1: s_2 \cdot s_3 + s_1 \cdot s_1$

$p_2 \leq p_1$

$p_2: s_2 \cdot s_3 + s_1$

$p_2 \leq p_3$

$p_3: s_2 \cdot s_3 \cdot s_2 + s_2 \cdot s_3 \cdot s_3 + s_1$

However, $p_1 \nleq p_3$

P-Minimality

- $Q \subseteq_P Q'$ iff
  Q, Q' equivalent
  $\forall D \;\; \forall t \in Q(D)=Q'(D)$
  $Prov(t,Q,D) \leq Prov(t,Q',D)$.

- **Problem Statement (p-minimization):**
  Given a class of queries $\mathscr{C}$, and $Q \in \mathscr{C}$, we want to compute a query Q' that is equivalent to Q and is p-minimal,
  i.e. $\forall Q'' \in \mathscr{C}$ equivalent to Q,Q', $Q' \subseteq_P Q''$.

P-Minimality Characterization

- We need to characterize when some query is terser than another.
- We take inspiration from "standard" query minimization, finding an equivalent query with the minimal number of joins.

Standard Query Minimization

- Chandra & Merlin (1977) proved that for every $Q, Q' \in CQ$, there exists a homomorphism $h: Q' \to Q$ iff $Q \subseteq Q'$.
  - A homomorphism is a mapping between relational atoms, respecting the arguments

Standard Query Minimization – Cont.

- Moreover, Q is minimal in the standard sense iff there exists no homomorphism from Q to any of its strict sub-queries.
- Thus, to minimize a query, we look for strict sub-queries that are equivalent to that query.

P-Minimization in CQ

- **Theorem:** Given two equivalent queries Q,Q' in CQ, if there exists a surjective homomorphism $h:Q' \rightarrow Q$ then $Q \subseteq_P Q'$.
- **Theorem:** in CQ, standard minimization is the same as p-minimization
  - The proof uses the two homomorphism theorems.
  - We probably cannot compute efficiently, since standard minimization is known to be DP-Complete (Fagin, Kolaitis and Popa, 2005)
    - **DP**: a pair of an NP and a coNP problems.
  - But also good news - same heuristics and optimization techniques can be used for p-minimization.

Conjunctive Queries with Disequalities ($CQ^{\neq}$)

- So far, we could express equalities by using the same argument .
- The class $CQ^{\neq}$ allows using disequalities ($\neq$).
- For example,

  $$Ans(z) := R(w,a), R(z,v), R(z,w), z \neq a, v \neq w$$

- We want to find the p-minimal equivalent query within $CQ^{\neq}$.

This Time It's Different…

- **Lemma:** There exist $Q_1 \equiv Q_2$, two DBs D, D', s.t. $P((),Q_2,D) < P((),Q_1,D)$ but $P((),Q_1,D') < P((),Q_2,D')$.
- **Lemma:** There exists no other query equivalent to Q_1, Q_2 with less provenance on D, D'.
- Thus they have no p-minimal equivalent.
- How come?
  - The homomorphism theorems fail in $CQ^{\neq}$.
- We will see later that a p-minimal equivalent can be found in a larger query class which allows union.

Standard minimization in $CQ^{\neq}$?

- A standard minimal equivalent query always exists – the equivalent query with least joins…
- Since the homomorphism thm. does not hold, Klug (1988) gives a different way to find the minimal query.
- One open question posed by Klug:
  Is the minimal query unique? (as in CQ)
- By a construction given to us by Georg Gottlob: No!
  - Q_1 and Q_2 are minimal in the standard sense and equivalent, but not equal (isomorphic)…

Complete Conjunctive Queries (cCQ $^{\neq}$)

- Last class of conjunctive queries to consider[1].
- Consists of queries where there are explicit disequalities stated between each pair of distinct arguments.
- For example:

$$Ans(z) := R(w,a), R(z,w), z \neq a, w \neq a, z \neq w$$

[1] This class is very important for the algorithm of UCQ $^{\neq}$ query minimization, which is not detailed in this presentation.

Our results for cCQ$^{\neq}$

- **Good news:** The homomorphism theorem holds for cCQ$^{\neq}$ $\Rightarrow$ again, in cCQ$^{\neq}$ standard minimality and p-minimality are the same.
- **More good news:** unlike CQ, the p-minimal equivalent can be computed in cCQ$^{\neq}$ in PTIME.
  - **Lemma:** a query in cCQ$^{\neq}$ is (p-)minimal iff it does not contain duplicated relational atoms
    ans(x) := R(x),R(y),S(x,y),R(x),x≠y
  - The duplicated atoms can be easily found and removed in PTIME.

Conjunctive Queries - Summary

CQ	$CQ^{\neq}$	$cCQ^{\neq}$
Queries w.o. disequalities. • Standard minimization = p-minimization. • A (p-)minimal equivalent always exists. • The decision problem is DP-Complete.	General conjunctive queries w. disequalities • Standard minimization ≠ p-minimization • For some queries there exists no p-minimal equivalent	all the distinct arguments are disequated • Same as CQ, but the p-minimal equivalent can be computed in PTIME

Motivation for Using Unions

- Q_1: Ans(x) := R(x,y),R(y,x) is p-minimal in CQ.
  - Proof: There is no homomorphism from Q_1 to any of its sub-queries.
- We have also seen
  Q_2: Ans(x) := R(x,y),R(y,x),x≠y
  $\cup$ Ans(x) := R(x,x)
- We gave an example of a DB where the provenance of Q_2 is actually terser:
  $P((a), Q_2, D) = s_2 \cdot s_3 + s_1 < s_2 \cdot s_3 + s_1 \cdot s_1 = P((a), Q_1, D)$
- In fact $Q_2 \subset_P Q_1$!
- This means we can do better using unions…

Unions of Conjunctive Queries (UCQ$^{\neq}$)

- Captures SPJ**U** queries.
- Queries of the form $Q=Q_1 \cup Q_2 \cup \ldots \cup Q_n$, where
  - $Q_1,Q_2,\ldots,Q_n \in CQ^{\neq}$ are called the adjuncts of Q.
- The provenance of an output tuple t is

$$Prov(t,Q,D) = Prov(t,Q_1,D)+\ldots+Prov(t,Q_n,D)$$

- E.g. **Q_2**:

$$Ans(x) := R(x,y),R(y,x),x\neq y$$
$$\cup\ Ans(x) := R(x,x)$$

  - $P((a), Q, D) = s_2 \cdot s_3 + s_1$

A	B	Provenance
a	a	s_1
a	b	s_2
b	a	s_3

Good News about $UCQ^{\neq}$

- **Theorem:** For every query Q in $UCQ^{\neq}$ there exists a p-minimal equivalent in $UCQ^{\neq}$.
- We have an (exponential time) algorithm for computing it.
- In particular, since $CQ^{\neq} \subseteq UCQ^{\neq}$, this means we can find a p-minimal equivalent to every $CQ^{\neq}$ query in $UCQ^{\neq}$.
- For some p-minimal queries in CQ, an equivalent query with terser provenance can be found outside CQ.

Related Work

- **Management of provenance information**
  - Specific provenance management techniques, e.g. why provenance, Trio provenance, Provenance semirings
  - Provenance management tools
- **Standard query minimization**
  - Conjunctive queries (Chandra & Merlin 1977)
  - Unions (Sagiv and Yannakakis 1980)
  - CQ w. inequalities (Klug 1988)
  - Many others
- **Data Exchange** – core of universal solutions

Acknowledgement

We are grateful to Georg Gottlob for providing us with a counter-example for non-unique minimal query for $CQ^{\neq}$, which helped in proving the non-existence of a p-minimal query for this class.

Future Work

- Find restricted cases with lower complexity bounds.
- P-minimization in other classes of queries (e.g. general **in**equalities <, ≤,…, aggregation queries).
- Study the connection to core in data exchange.
- Optimizations
  - Employing existing heuristics and optimization techniques (of standard minimization) for p-minimization .

Conclusion

In this work we have studied:

- Core provenance information, which is common to all equivalent queries.
- When a query that realizes the core provenance exists and how to compute it, in different query classes:
  - Conjunctive queries: CQ, $CQ^{\neq}$, $cCQ^{\neq}$.
  - Unions thereof: $UCQ^{\neq}$.
- Direct computation of core provenance polynomials from provenance information.

Thank You!

Q&A