Reconcilable Differences

Todd J. Green Zachary G. Ives Val Tannen
University of Pennsylvania

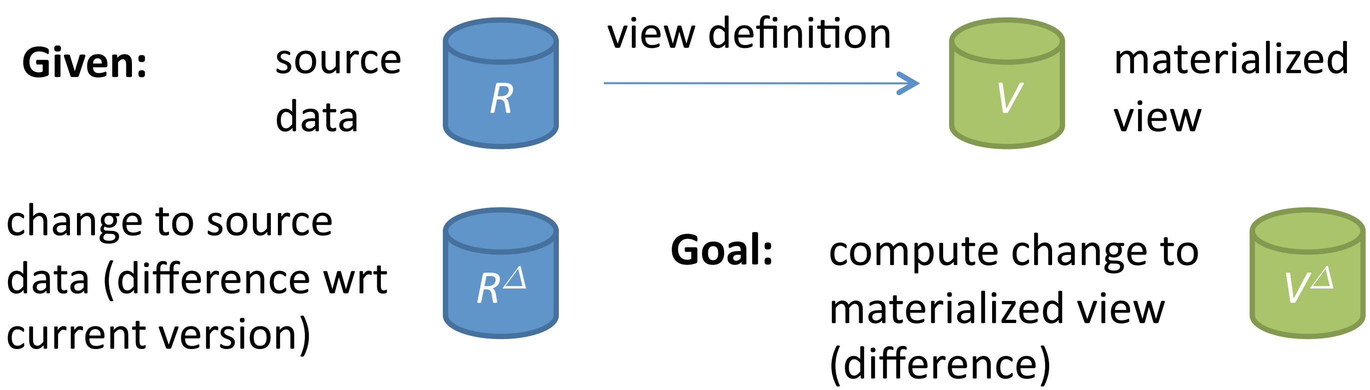
March 24, 2009
@ ICDT 09, Saint Petersburg

Change is a Constant in Data Management

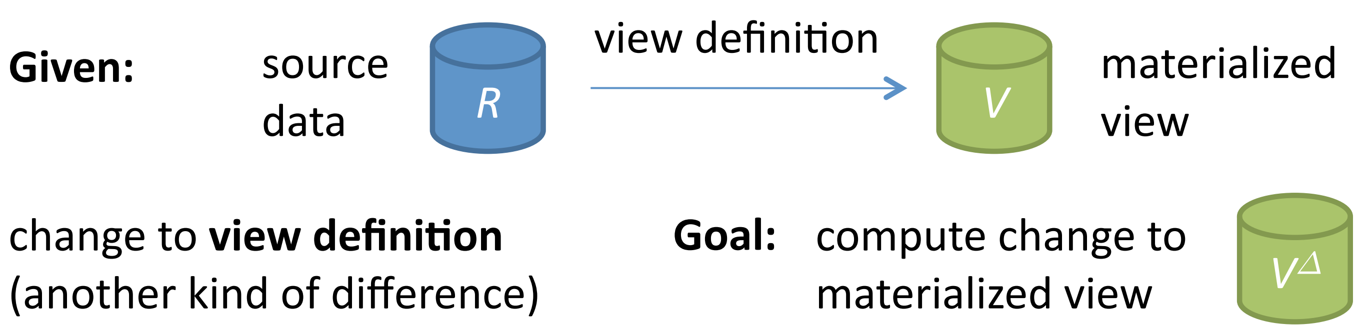
- Databases are highly **dynamic**; many kinds of **changes** need to be propagated efficiently:
  - To **data** ("view maintenance")
  - To **view definitions** ("view adaptation")
  - Others, such as schema evolution, etc.
- Data exchange and collaborative data sharing systems (e.g., ORCHESTRA [Ives+ 05]) exacerbate this need:
  - Large numbers of materialized views
  - Frequent updates to data, schemas, view definitions

Change Propagation: a Problem of Computing Differences

View maintenance



View adaptation



Challenges in Change Propagation

- **View maintenance**: studied since at least the mid-eighties [Blakeley+ 86], but existing solutions quite narrow and limited
  - Various known methods to compute changes "incrementally", e.g., **count algorithm** [Gupta+ 93]
  - How do we **optimize** this process? What is space of **all** update plans?
- **View adaptation**: less attention, but renewed importance in context of data exchange/collaborative data sharing systems
  - Previous approaches: limited to case-based methods for simple changes [Gupta+ 01]
  - **Complex** changes? Again, space of **all** update plans?
- Key challenge: compute changes using database queries!

Contributions

- A novel, unified approach to view maintenance, view adaptation that allows the incorporation of optimization strategies:
  - Representing changes and data together: **$\mathbb{Z}$-relations**
  - View maintenance, view adaptation as special cases of a more general problem: **rewriting queries using views** (on $\mathbb{Z}$-relations)
- A sound and complete algorithm for rewriting relational algebra (*RA*) queries (with difference!) using *RA* views on $\mathbb{Z}$-relations
  - Enabled by the surprising decidability of $\mathbb{Z}$-equivalence of *RA* queries
- Maintaining/adapting views under bag or set semantics via excursion through $\mathbb{Z}$-semantics

Representing Changes as Data: $\mathbb{Z}$-Relations

- Can think of changes to data as a kind of **annotated relation**

inserted tuple	+
deleted tuple	–

- **$\mathbb{Z}$-relation**: a relation where each tuple is associated with a (positive or negative) **count**

R^{Δ}

a	b	2
c	d	–3

  – Positive counts indicate (multiple) insertions; negative counts, (multiple) deletions

  – Uniform representation for both **data** and **changes** to data

  – Update application = union (a query!)

$$R' = R \cup R^{\Delta}$$

Relational Algebra (*RA*) on $\mathbb{Z}$-Relations

join ($\bowtie$) **multiplies** counts

union ($\cup$), **projection** (π) **add** counts

selection (σ) **multiplies** counts by 0 or 1

difference (–) **subtracts** counts

Same as for bag semantics, except difference can lead to negative annotations (unlike "proper subtraction" in bag semantics where negative counts are truncated to 0)

Incremental View Maintenance: An Application of $\mathbb{Z}$-Relations

Source relation:

R

a	b	1
c	b	1
b	c	1
b	a	1

R^{Δ}

b	**a**	**-1**
c	**d**	**+1**

deletion (b, a, -1)

insertion (c, d, +1)

Materialized view (with duplicates):

$V(x,y) :– R(x,z), R(z,y)$

a	a	1
a	c	1
b	b	2
c	c	1

2 copies of (b,b)

V^{Δ}

b	**b**	**-1**
b	**d**	**+1**

delete 1 copy of (b,b)

insert 1 copy of (b,d)

Delta rules [Gupta+ 93] **for *V* with $\mathbb{Z}$-relations semantics:**

$V^{\Delta}(x,y) :– R(x,z), R^{\Delta}(z,y)$

$V^{\Delta}(x,y) :– R^{\Delta}(x,z), R'(z,y)$

Delta Rules: a Special Case of Rewriting Queries Using Views on $\mathbb{Z}$-Relations

Query (to compute diff.):

$V^{\Delta}(x,y) :- R'(x,z), R'(z,y)$
$- V^{\Delta}(x,y) :- R(x,z), R(z,y)$

Materialized views:

$V(x,y) :- R(x,z), R(z,y)$

$R'(x,y) :- R(x,y)$
$R'(x,y) :- R^{\Delta}(x,y)$

rewrite V^{Δ} using the materialized views

... OTHER PLANS...?

Delta rules rewriting:

$V^{\Delta}(x,y) :- R(x,z), R^{\Delta}(z,y)$
$V^{\Delta}(x,y) :- R^{\Delta}(x,z), R'(z,y)$

Another delta rules rewriting:

$V^{\Delta}(x,y) :- R^{\Delta}(x,z), R(z,y)$
$V^{\Delta}(x,y) :- R'(x,z), R^{\Delta}(z,y)$

View Adaptation: Another Application of Rewriting Queries Using Views

Old view definition:

V(x,y) :– R(x,z), R(z,y)
V(x,y) :– R(x,z), R(y,z)

New view definition:

V'(x,y) :– R(x,z), R(z,y)

reformulate using materialized view V

… AGAIN, OTHER PLANS…?

A plan to "adapt" V into V':

V'(x,y) :– V(x,y)
– V'(x,y) :– R(x,z), R(y,z)

Bag Semantics, Set Semantics via $\mathbb{Z}$-Semantics

- Even if we can solve the problems for $\mathbb{Z}$-relations, what does this tell us about the answers we actually need: for bag semantics or set semantics?
- For **positive** *RA* (RA^+) queries/views on bags
  - $\mathbb{Z}$-semantics and bag semantics agree
  - Further, eliminate duplicates to get set semantics
  - Still works if rewriting is actually in *RA* (introduces difference)!
- Also works for *RA* queries/views with restricted use of difference
  - Still covers, e.g., the incremental view maintenance case

$\mathbb{Z}$-Equivalence Coincides with Bag-Equivalence for Positive *RA* (*RA*$^+$)

Lemma. For *RA*$^+$ queries *Q*, *Q'* we have $Q \equiv_{\mathbb{Z}} Q'$ (equivalent on $\mathbb{Z}$-relations) iff $Q \equiv_{\mathbb{N}} Q'$ (equivalent on bag relations)

Corollary. Checking $\mathbb{Z}$-equivalence for *RA*$^+$: convert to **unions of conjunctive queries** (UCQs), check if **isomorphic**

- CQs $Q \equiv_{\mathbb{N}} Q'$ iff $Q \cong Q'$ [Lovász 67, Chaudhuri&Vardi 93]
- UCQs $Q \equiv_{\mathbb{N}} Q'$ iff $Q \cong Q'$ [Cohen+ 99]

Complexity of above: graph-isomorphism complete for UCQs; for *RA*$^+$ (exponentially more concise than UCQs), don't know!

$\mathbb{Z}$-Equivalence is Decidable for *RA*

Key idea. Every *RA* query *Q* can be (effectively) rewritten as a single difference $A - B$ where *A* and *B* are positive

- **Not true under set or bag semantics!**

Corollary. $\mathbb{Z}$-equivalence of *RA* queries is decidable

Proof. $A - B \equiv_{\mathbb{Z}} C - D$ where *A*, *B*, *C*, *D* are positive

$\Leftrightarrow$ $A \cup D \equiv_{\mathbb{Z}} B \cup C$

$\Leftrightarrow$ $A \cup D \equiv_{\mathbb{N}} B \cup C$ which is decidable [Cohen+ 99]

- **Same problem undecidable for set, bag semantics!**

Alternative representation of relational algebra queries justified by above: **differences of UCQs**

Rewriting Queries Using Views with $\mathbb{Z}$-Relations

Given: query Q and set $\mathcal{V}$ of materialized views, expressed as differences of UCQs

Goal: enumerate **all** $\mathbb{Z}$-equivalent rewritings of Q (w.r.t. $\mathcal{V}$)

Approach: term rewrite system with two rewrite rules

unfolding	replace view predicate with its definition
cancellation	e.g., $(A \cup B) - (A \cup C)$ becomes $B - C$

By repeatedly applying rewrite rules – both **forwards** and **backwards** (**folding** and **augmentation**) – we reach all (and only) $\mathbb{Z}$-equivalent rewritings

An Infinite Space of Rewritings

- There are only finitely many **positive** (nontrivial) rewritings of *RA* query Q using *RA* views $\mathcal{V}$
- With difference, can always rewrite ad infinitum by adding terms that "cancel"
- But even without this:

Let *RS* denote **relational composition** of R with S, i.e.,

$RS(x,y) :- R(x,z), R(z,y)$

Let $\mathcal{V}$ contain single view

$V = R \cup R^3$

repeated relational composition

Now consider

$$
\begin{aligned}
Q &= R^2 \\
&\equiv_{\mathbb{Z}} VR - R^4 \qquad \text{(equiv. is w.r.t. } \mathcal{V}\text{)} \\
&\equiv_{\mathbb{Z}} VR - VR^3 \cup R^6 \\
&\equiv_{\mathbb{Z}} VR - VR^3 \cup VR^5 - R^8 \\
&\equiv_{\mathbb{Z}} \ldots
\end{aligned}
$$

none of these have "cancelling" terms!

How Do We Bound the Space of Rewritings? Use Cost Models!

Can make some reasonable **cost model** assumptions:

– cost($A \cup B$) ≥ cost(A) + cost(B)

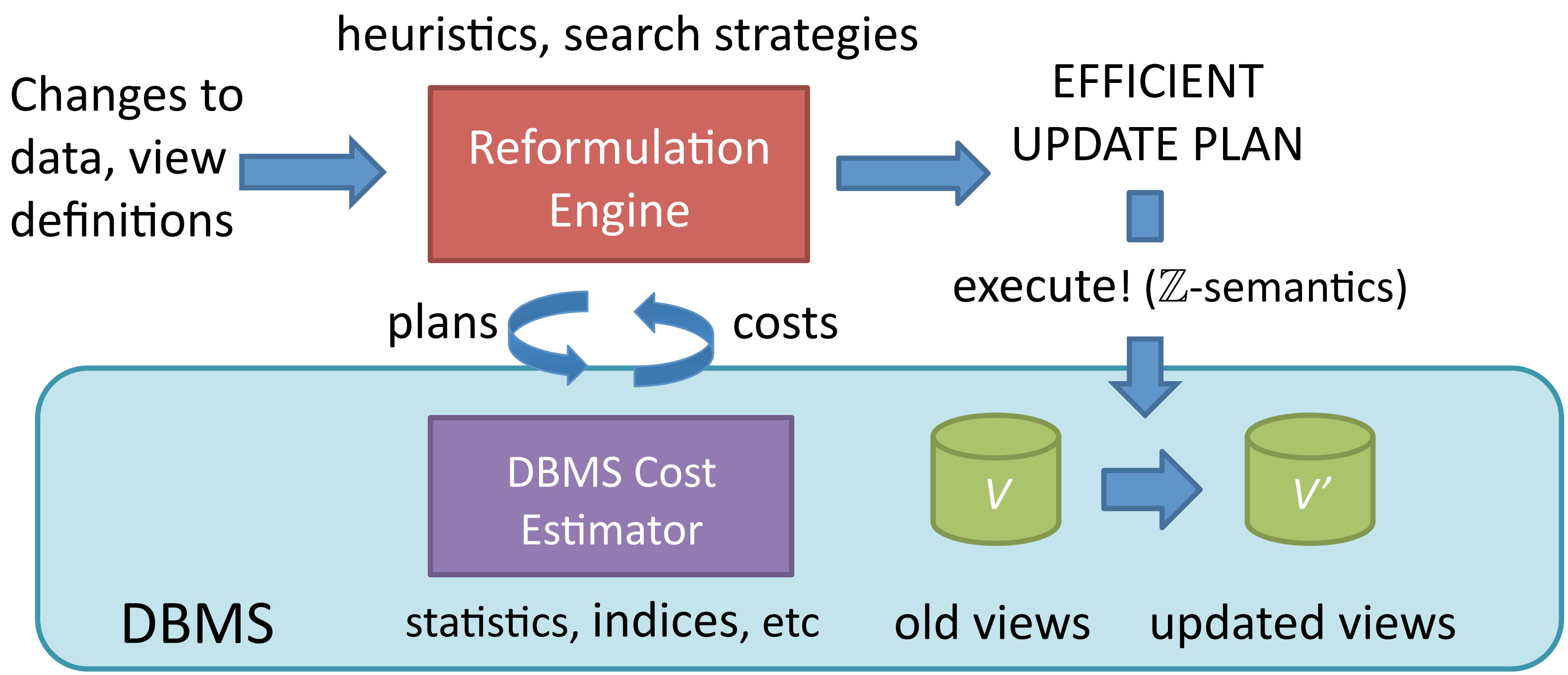
– cost($A \bowtie B$) ≥ cost(A) + cost(B) + card($A \bowtie B$)

– etc.

Theorem. Under above assumptions, can find minimal-cost reformulation of *RA* query *Q* using *RA* views $\mathcal{V}$ in a bounded number of steps

Blueprint for a Practical Implementation

Approach: pair reformulation algorithm with DBMS cost estimator, cost-based search strategies



Main challenge: find effective heuristics, strategies to guide search through (finite but huge) space; find good (not optimal) plan quickly

Highlights of Other Results

- $\mathbb{Z}$-equivalence remains decidable for *RA* with built-in predicates (<, ≤, >, ≥, ≠) over dense linear order
  - Basic idea: can **linearize** (cf., e.g., [Cohen+ 99]) queries, then test for isomorphism

    e.g., *Q*(x,y) :- *R*(x,y), x ≠ y → *Q*(x,y) :- *R*(x,y), x < y ; *Q*(x,y) :- *R*(x,y), y < x

- Full characterization of class of *RA* queries where $\mathbb{Z}$-semantics and bag semantics agree on all bag instances, hence where $\mathbb{Z}$-semantics can be used for evaluation
  - Bad news: undecidable class
  - Good news: covers incremental maintenance of positive views (where difference is used only for changes to sources)

Related Work

- **Incremental view maintenance** [Blakeley+ 86], [Gupta+ 93], ...
  - "deltas" [Gupta+ 93]: an early form of our $\mathbb{Z}$-relations
- **Answering queries using views** [Levy+ 95], [Chaudhuri+ 95], [Afrati&Pavlaki 06], ...
- **Bag-containment/bag-equivalence of CQs/UCQs** [Lovász 67], [Chaudhuri&Vardi 93], [Ioannidis&Ramakrishnan 95], [Cohen+ 99], [Jayram+ 06]
- **Containment/equivalence with provenance annotations** [Tan 03], [Green ICDT 09]
- **View adaptation** [Mohania&Dong 96], [Gupta+ 01]
- **Mapping evolution** [Velegrakis+ 03]

Conclusion

- Change propagation for *RA* views can be **optimized**, via **rewriting queries using views** and **$\mathbb{Z}$-relations**
  - Sound and **complete** rewriting algorithm
- Wider impact: techniques also work for **provenance-annotated $\mathbb{Z}$[*X*]-relations**, cf. [Green+ 07], [Geerts&Poggi 08]
- Open problems: exact complexity of checking...
  - $\mathbb{Z}$-equivalence of *RA* queries? (in PSPACE, GI-hard)
  - Bag-equivalence of RA^+ queries? (also in PSPACE, GI-hard)
  - Above problems, for queries with built-in predicates?