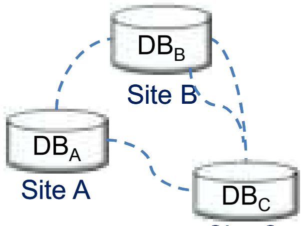
Foundations and Applications of Data Provenance

Grigoris Karvounarakis

BigProv'13 Workshop

Data sharing in the era of Big Data (3V's)



- Data sharing among large number of diverse data sources (high **variety**)
  - Sites can have different schemas or even data models, and viewpoints on "truth"
- Sites contribute and import (map) large **volumes** of data
- Need to handle frequent updates to local and imported data and mappings efficiently (high **velocity**)
- Big Data Analytics: quality of results only as high as that of input data, need to determine what to trust

1

"Where Did this Data Come from and how?"

- A common set of questions:
  - Which sources did the data originate from?
  - What operations were used to create and propagate the data?
  - How can we assess trust, data quality etc based on this information?
- *Data provenance* captures the relationships between items in data instances created through declarative queries or views
  - Different from workflow provenance (e.g., [OPM], [PROV-O]) which captures procedural code and usually treats operations as black boxes due to their complexity

2

LOGICBLOX

Main topics of this talk

- What data provenance models are there? What data models and query language operators can they capture? How do they compare to each other?
- How can data provenance support assessment of various dimensions of data quality and help in dealing with the 3V's? What systems and applications take advantage of this?
- What are the benefits of data provenance in Big Data settings, what are the challenges introduced by the 3V's, and how can we deal with them?

3

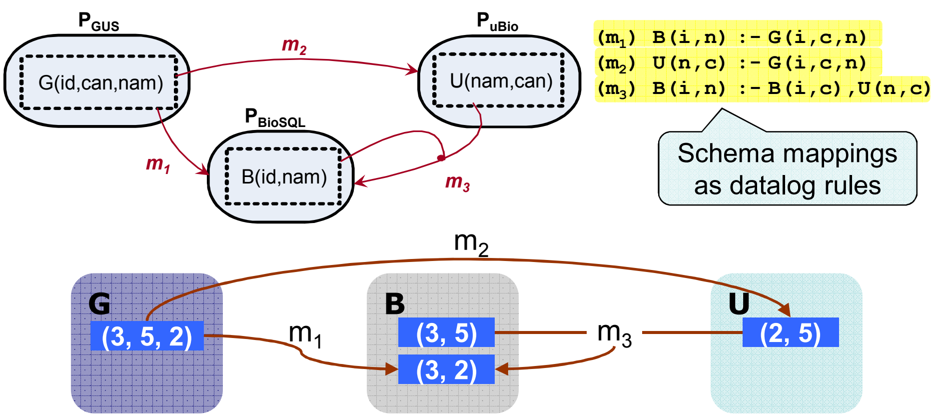
LOGICBLOX

Outline

- Data provenance models for positive relational algebra queries
- Applications of data provenance
- Extensions to the theoretical framework
- Benefits and challenges of data provenance on Big Data

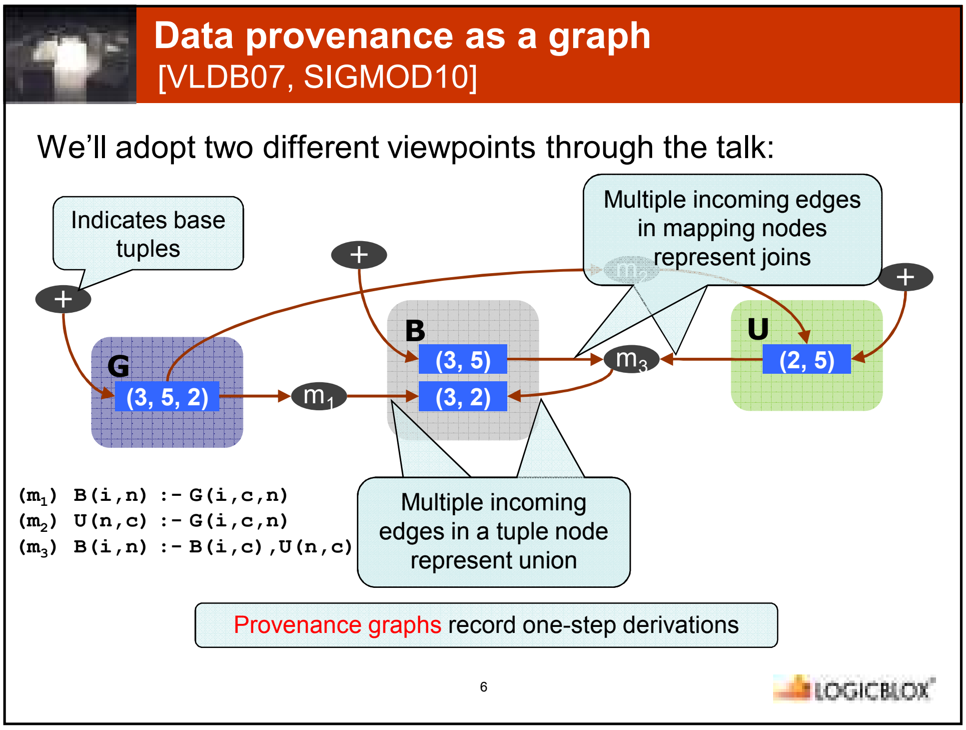
LOGICBLOX

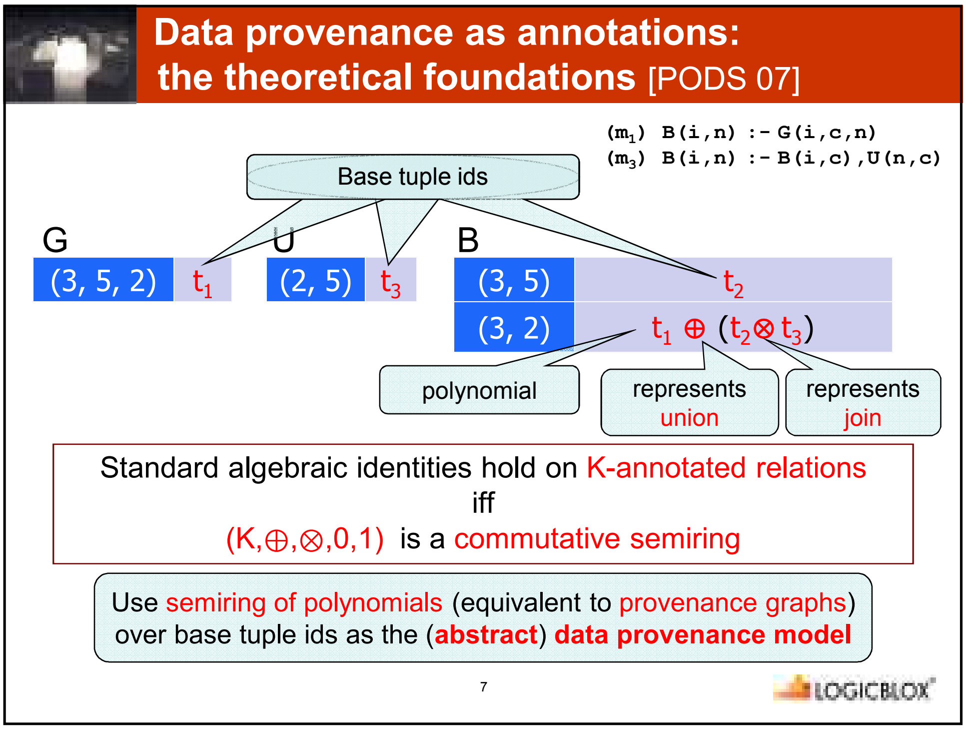
An Example Data Sharing Scenario: Collaborative Data Sharing in ORCHESTRA [VLDB07]



How do we record provenance for the operations prescribed by these mappings (join, union)?

LOGICBLOX


Data provenance as a graph
[VLDB07, SIGMOD10]
We'll adopt two different viewpoints through the talk:
Indicates base tuples
Multiple incoming edges in mapping nodes represent joins
G
(3, 5, 2)
m1
B
(3, 5)
(3, 2)
m3
U
(2, 5)
(m1) B(i,n) :- G(i,c,n)
(m2) U(n,c) :- G(i,c,n)
(m3) B(i,n) :- B(i,c),U(n,c)
Multiple incoming edges in a tuple node represent union
Provenance graphs record one-step derivations
6
LOGICBLOX


Data provenance as annotations: the theoretical foundations [PODS 07]
(m1) B(i,n) :- G(i,c,n)
(m3) B(i,n) :- B(i,c),U(n,c)
Base tuple ids
G
(3, 5, 2) t1
U
(2, 5) t3
B
(3, 5) t2
(3, 2) t1 ⊕ (t2⊗t3)
polynomial
represents union
represents join
Standard algebraic identities hold on K-annotated relations
iff
(K,⊕,⊗,0,1) is a commutative semiring
Use semiring of polynomials (equivalent to provenance graphs) over base tuple ids as the (abstract) data provenance model
7
LOGICBLOX

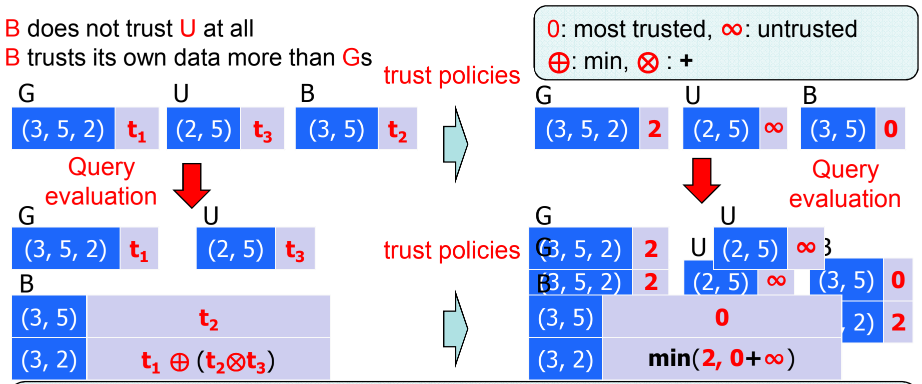
Semirings unify commonly-used database semantics involving annotations

Standard database models	
set semantics	
bag semantics	
Trust, security	
boolean trust, derivability	[VLDB07]
ranked trust	[SIGMOD10]
confidentiality	[Foster+08]
Uncertainty, incompleteness	
incomplete DBs	[Imielinski+84]
probabilistic DBs	[Fuhr+97]
ranks, scores	[Talukdar+08]

8

Example: computing ranked (dis)trust annotations through provenance

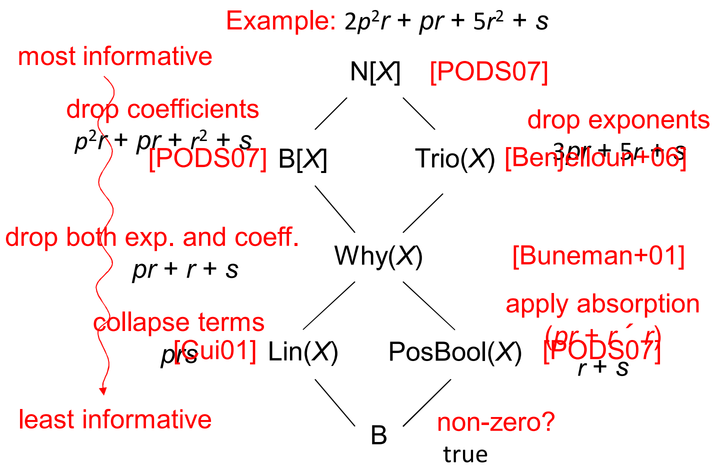
Provenance polynomials abstract calculations in **all** commutative semirings



- Record data provenance (i.e., abstract annotations) during query evaluation
- Evaluate different trust policies or various other annotations through provenance later, often on small subset of query results, without recomputation
  - Especially important for Big Data, due to high volume and velocity

9

Hierarchy of relational abstract provenance models [Green11]



A path downward from K_1 to K_2 indicates that we can compute K_2 from K_1

LOGICBLOX

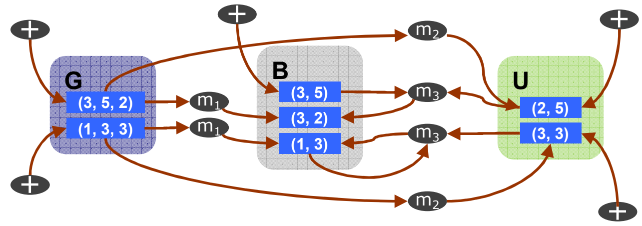
Outline

- Data provenance models for positive relational algebra queries
- Applications of data provenance
  - Provenance querying and annotation computations
  - Uses in research prototypes and commercial systems
- Extensions to the theoretical framework
- Benefits and challenges of data provenance on Big Data

LOGICBLOX

Storing provenance in relations
(Orchestra [VLDB07, SIGMOD10] – LogicBlox [Datalog12])

- Similar to storing graphs in edge relations, but here mapping nodes have multiple "input" and "output" tuples: hyperedges
- Use tuple values (keys) as ids for base tuples



(m_1) B(i,n) :- G(i,c,n) (m_2) U(n,c) :- G(i,c,n) (m_3) B(i,n) :- B(i,c), U(n,c)

P_1(B.i, B.n, G.i, G.c, G.n)
(3, 2, 3, 5, 2)
(1, 3, 1, 3, 3)

P_2(i,c,n)
(3, 5, 2)
(1, 3, 3)

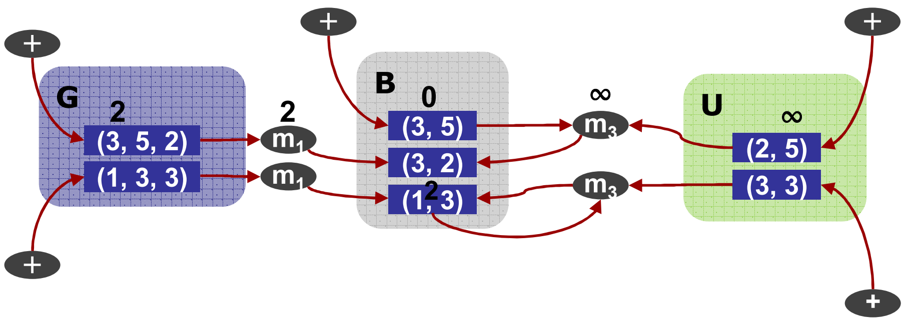
P_3(i,c,n)
(3, 5, 2)
(1, 3, 3)

12

LOGICBLOX

Example of provenance query for ranked (dis)trust computation [SIGMOD10]

Find derivations of B(3,2) from base data



B does not trust U at all

B trusts its own data more than G's

What is the trust rank of B(3,2)?

13

LOGICBLOX

ProQL syntax for ranked (dis)trust assessment [SIGMOD10]

```
EVALUATE RANK OF {
     FOR [B $x] <-+ []
     WHERE $x.id = 3 AND $x.nam = 2
     INCLUDE PATH [$x] <-+ []
     RETURN $x
}
ASSIGNING EACH leaf_node $y {
     CASE $y in G : SET 2
     CASE $y in U : SET inf
     DEFAULT : SET 0
}
ASSIGNING EACH mapping $p($z) {
     CASE $p = m2 : SET 2*$z
     DEFAULT : SET $z
}
```

DERIVABILITY
TRUST
LINEAGE
CONFIDENTIALITY
PROBABILITY

14

Provenance enables incremental algorithms for handling updates to data and views

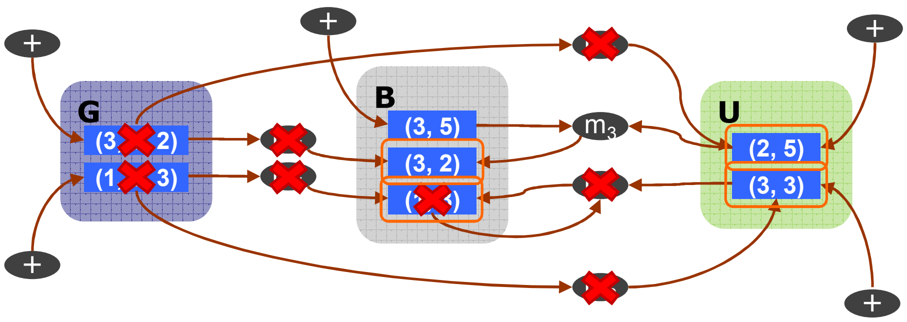
- Updates to source data (incremental view maintenance)
  - Past approaches (DRed [Gupta+93]) over-delete and recompute
  - Use data provenance to determine incrementally if derived tuples should be deleted without recomputation [V

Very important due to high volume and velocity

- Updates to derived data (view update)
  - Past approaches ([Dayal+82]) statically check and reject views that may cause side effects on some inputs
  - Use data provenance to determine at runtime if propagating specific deletions to source tuples will actually cause side effects [WebDB07]
- Updates to views (view adaptation)
  - Can be cast as applications of rewriting queries using materialized views, and data provenance can enable more efficient rewritings [Green+11, Green+12]

15

Provenance for incremental deletion propagation along unidirectional mappings [VLDB07]



- **Step 1**: Use provenance to find derived tuples which should also be deleted
- **Step 2**: Use provenance to also test other affected tuples for derivability, and delete any not derivable
- **Step 3**: Repeat until fixpoint

16

LOGICBLOX

Program analysis and debugging in LogicBlox [Datalog+12]

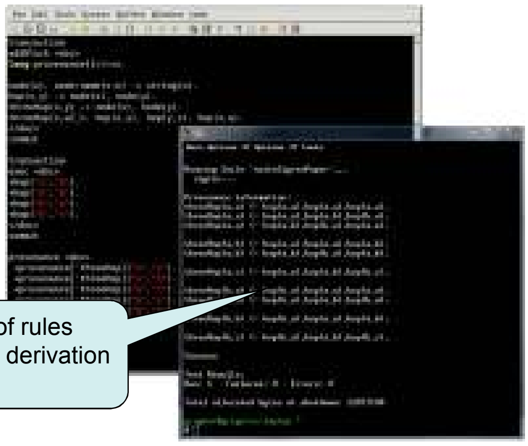
- Static (BloxAnalysis)
  - Represent Datalog programs using relational predicates
  - Use Datalog to query and analyze Datalog programs
- Examples of BloxAnalysis queries:
  - "get all predicates whose names matches foo and all rules in which those predicates appear in the head"
  - "find all predicates that are ``reachable'' from a certain predicate through rules in the program"
- Basis for performing more complex reasoning about Datalog programs in Datalog: dead code detection, clone detection …

17

LOGICBLOX

Analysis and debugging of declarative programs in LogicBlox

- Dynamic/runtime:
  - Record data provenance during program evaluation,
  - Use Datalog programs to explore and query resulting provenance graph.



Instantiations of rules involved in the derivation of each fact

- [Rugaber+13] describe how this functionality could be exposed in an Interactive Development Environment (IDE) for Datalog for program debugging

18

Provenance for debugging in other systems

- GPad [Koehler+12] Declarative debugging for Datalog
  - Implemented on top of LogicBlox, taking advantage of BloxAnalysis and provenance recording capabilities
  - Uses Datalog and Statelog to represent and query provenance (firing) graphs, e.g., to compute stage of program evaluation at which each fact was derived
- SPIDER [Chiticariu+06] uses a form of data provenance (routes) for debugging schema mappings
  - Compute a single derivation for an output tuple, or enumerate all derivations (when there are finitely many)

19

Outline

- Data provenance models for positive relational algebra queries
- Applications of data provenance
- Extensions to the theoretical framework
  - Provenance models for other query languages/operators
  - Other related theoretical results
- Benefits and challenges of data provenance on Big Data

20

LOGICBLOX

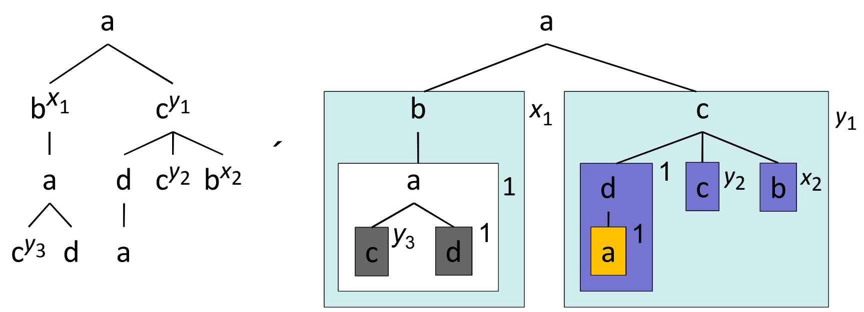
Relational difference

- M-semirings [Geerts+10] extend semirings with a monus operator to capture relational difference
- Unfortunately there is no suitable abstract structure that can be used as provenance model for m-semirings e.g., to compute various annotations
- [Amsterdamer+11c] identified further difficulties due to the fact that relational difference satisfies two sets of incompatible equivalences in the set and bag semantics

21

LOGICBLOX

Provenance for unordered XML [Foster+08]



K-UXQuery: Based on Xquery, contains FOR loops and *//*

Main result: Provenance of K-UXQueries over unordered XML can still be captured by provenance polynomials, and annotations can be computed through it.

22

Provenance for RDF

- RDF inference rules [Flouris+09, Udrea+10,Buneman+11,Zimmerman+12]
  - based on similar algebraic structures (idempotent semirings)
- SPARQL queries: main challenge from non-monotone OPTIONAL operator (akin to relational left-outer join)
  - Provenance polynomials can still capture provenance of positive SPARQL queries [IntComp11]
  - OPTIONAL can be encoded through relational difference [Damasio+12] (caveat: problems capturing provenance of relational difference)
  - Semirings with an embedded boolean algebra [ICDT13] can be used to construct a suitable data provenance model for SPARQL queries that can be used e.g., to compute various annotations

23

Other related theoretical results

- Recursion [PODS07]: Results for positive relational algebra can be extended to Datalog for omega-continuous semirings
  - Provenance games [Zinn+13]: Novel unifying framework for dealing with recursion, negation and "why-not" provenance
- Aggregate operators [Amsterdamer+11a]
- Query containment [Green11,Kostylev+12]
- Minimization [Amsterdamer+11b]
- Factorization [Olteanu+12]

24

LOGICBLOX

Outline

- Data provenance models for positive relational algebra queries
- Applications of data provenance
- Extensions to the theoretical framework
- Benefits and challenges of data provenance on Big Data

25

LOGICBLOX

Benefits of data provenance for Big Data (1/2)

- Data provenance is crucial for trusted Big Data Analytics due to 3Vs
  - Can be used to assess quality of data imported from large number of diverse sources, possibly using different data models and query languages (relational, XML, RDF) (high **variety**)
    - combinations of data and workflow provenance [Acar+10, Amsterdamer+12] may help in also dealing with unstructured data
  - Enables quality assessment at any time, even if sources have changed or are unavailable (due to high **velocity**)
  - Facilitates more efficient provenance querying and annotation computations for small subsets of data, by avoiding recomputation (infeasible, due to high **volume** and **velocity**)

26

LOGICBLOX

Benefits of data provenance for Big Data (2/2)

  - Can be used for assessment of various dimensions of data quality, based of different users' beliefs, again preventing the need for recomputation
- Data provenance enables efficient incremental algorithms for handling updates to data and views in data sharing systems
  - Such updates are frequent, due to high **velocity**
  - Avoid redundant computations, enable more flexible propagation and more efficient rewritings
- Supports analysis and debugging of declarative programs
  - Especially useful because large numbers of views/mappings are generated automatically by tools

27

LOGICBLOX

Challenges and research directions

- Preliminary experiments in [VLDB07,SIGMOD10] have shown feasibility for medium-size data sharing settings, investigate performance and scalability in Big Data settings

Some potential ideas/directions

- Optimizations for provenance querying and annotation computations on Big provenance graphs
  - Indexing (some preliminary work in [SIGMOD10])
  - Leverage distributed techniques (e.g., MapReduce)
  - Partial provenance evaluation (e.g., may only care if provenance evaluates to "non-zero" or is over some threshold, not exact value)

28

LOGICBLOX

Challenges and research directions

- Techniques to optimize provenance storage overhead, and study trade-off with query performance
  - Storage scheme of [SIGMOD10] is a step in this direction
  - Theoretical results on provenance minimization and factorization
  - Explore compression methods or distributed/cloud-based storage
- Tradeoffs between more expressive provenance models and cost of storage/querying
  - Use provenance models that are as informative as necessary
  - Consider partial provenance information e.g., only involving subset of data or query operators, or ignoring some sources

29

LOGICBLOX

Acknowledgments

- Work in this area done in collaboration with (and some material in a few slides borrowed from):
  - Val Tannen (University of Pennsylvania)
  - Zack Ives (University of Pennsylvania)
  - Todd J. Green (LogicBlox)
  - Floris Geerts (University of Antwerp)
  - Vassilis Christophides (University of Crete & ICS-FORTH)
  - Irini Fundulaki (ICS-FORTH)

30

Based on work described in:

[PODS07] *T.J. Green, G. Karvounarakis, V. Tannen*: Provenance semirings. PODS 2007

[VLDB07] *T.J. Green, G. Karvounarakis, Z.G. Ives, V. Tannen*: Update Exchange with Mappings and Provenance. VLDB 2007

[WebDB08] *G. Karvounarakis and Z.G. Ives*. Bidirectional mappings for data and update exchange. In WebDB, 2008.

[SIGMOD10] *G. Karvounarakis, Z.G. Ives, V. Tannen*: Querying data provenance. SIGMOD 2010

[IntComp11] *Y. Theoharis, I. Fundulaki, G. Karvounarakis, V. Christophides*: On Provenance of Queries on Semantic Web Data. IEEE Internet Computing 15(1): 31-39 (2011)

[Datalog12] *T.J. Green, M. Aref, G. Karvounarakis*: LogicBlox, Platform and Language: A Tutorial. Datalog 2012: 1-8

[ICDT13] *F. Geerts, G. Karvounarakis, V. Christophides and I. Fundulaki*: Algebraic Structures for Capturing the Provenance of SPARQL Queries, ICDT 2013

31

References

[Acar+10] *U. A. Acar, P. Buneman, J. Cheney, N. Kwasnikowska, S. Vansummeren, and J. van den Bussche*. A graph model for data and workflow provenance. In Workshop on the Theory and Practice of Provenance, 2010.

[Amsterdamer+11a] Y. Amsterdamer, D. Deutch, and V. Tannen. Provenance for Aggregate Queries. In PODS 2011.

[Amsterdamer+11b] *Y. Amsterdamer, D. Deutch, T. Milo, and V. Tannen*. On provenance minimization. In PODS, 2011.

[Amsterdamer+11c] *Y. Amsterdamer, D. Deutch, and V. Tannen*. On the limitations of provenance for queries with difference. In TaPP, 2011.

[Amsterdamer+12] *Y. Amsterdamer, S. B. Davidson, D. Deutch, T. Milo,J. Stoyanovich, and V. Tannen*. Putting lipstick on pig: Enabling database-style workflow provenance. In VLDB, 2012.

[Benjelloun+06] *O. Benjelloun, A. Das Sarma, A. Y. Halevy, J. Widom*: ULDBs: Databases with Uncertainty and Lineage. VLDB 2006

[Buneman+01] *P. Buneman, S. Khanna, W.-C. Tan*: Why and Where: A Characterization of Data Provenance. ICDT 2001

[Buneman+11] *P. Buneman and E. V. Kostylev*. Annotation algebras for RDFS. In SWPM, 2011.

LOGICBLOX

References

[Chiticariu+06] *L. Chiticariu, W.-C. Tan*: Debugging Schema Mappings with routes. VLDB 2006

[Cui+00] *Y. Cui, J. Widom, and J. L. Wiener:* Tracing the lineage of view data in a warehousing environment. ACM TODS, 25(2), 2000

[Damasio+12] *C. Damasio, A. Analyti, and G. Antoniou*. Provenance for SPARQL queries. In ISWC, 2012.

[Dayal+82] U. Dayal, and P.A. Bernstein. On the correct translation of update operations on relational views. ACM TODS 7(3), 1982

[Dividino+09] *R. Dividino, S. Sizov, S. Staab, and B. Schueler*. Querying for provenance, trust, uncertainty and other meta knowledge in RDF. J. Web Semantics, 7(3), 2009.

[Flouris+09] *G. Flouris, I. Fundulaki, P. Pediaditis, Y. Theoharis, and V. Christophides*. Coloring RDF Triples to Capture Provenance. In ISWC 2009.

[Foster+08] *J. N. Foster, T. J. Green, V. Tannen*: Annotated XML: queries and provenance. PODS 2008

LOGICBLOX

References

[Fuhr+97] *N. Fuhr and T. Rolleke*. A probabilistic relational algebra for the integration of information retrieval and database systems. TOIS 14(1),1997.

[Geerts+10] *F. Geerts and A. Poggi*. On database query languages for K-relations. J. Applied Logic, 8(2), 2010.

[Gartner12] http://www.gartner.com/it-glossary/big-data/

[Glavic+09] *B. Glavic, G. Alonso*: Perm: Processing Provenance and Data on the Same Data Model through Query Rewriting. ICDE 2009

[Green+12] *T. J. Green and Z. G. Ives*. Recomputing materialized instances after changes to mappings and data. In ICDE, 2012.

[Green+11a] *T. J. Green, Z. G. Ives, and V. Tannen*. Reconcilable differences. Theory of Computing Systems, 49(2), 2011.

[Green11] *T. J. Green*. Containment of Conjunctive Queries on Annotated Relations. Theory of Computing Systems, 49(2), 2011.

[Gupta+93] *A. Gupta, I.S. Mumick, and V.S. Subrahmanian*. Maintaining views incrementally, SIGMOD 1993

34

LOGICBLOX

References

[Imielinski+84] *T. Imielinski and W. Lipski*. Incomplete information in relational databases. JACM, 31(4), 1984.

[Koehler+12] *S. Koehler, B. Ludaescher and Y. Smaragdakis*: Declarative Datalog Debugging for Mere Mortals, Datalog 2.0 2012

[Koch10] *C. Koch*: Incremental query evaluation in a ring of databases. PODS 2010

[Kostylev+12] *E. V. Kostylev, J. L. Reutter, and A. Z. Salamon*. Classification of annotation semirings over query containment. In PODS, 2012.

[Li+13] *C.Li, D.Li, G.Miklau and D.Suciu*. A Theory of Pricing Private Data, ICDT 2013

[Olteanu+12] *D. Olteanu and J. Zavodny*. Factorised representations of query results: Size bounds and readability. In ICDT, 2012.

[OPM] Open Provenance Model. openprovenance.org

[PROV-O] PROV-O: The PROV Ontology. http://www.w3.org/TR/prov-o/

[Rugaber+13] *S. Rugaber, Z. Hemel, K. Stirewalt.* Live Logic Programming

Workshop on Live Programming, in conjunction with ICSE 2013

35

LOGICBLOX

References

[Talukdar+08] *P. P. Talukdar, M. Jacob, M. S. Mehmood, K. Crammer, Z. G. Ives, F. Pereira, and S. Guha*. Learning to create data-integrating queries. In VLDB, 2008.

[Udrea+10] *O. Udrea, D. R. Recupero, and V. S. Subrahmanian*. Annotated RDF. ACM Trans. Comput. Log., 11(2), 2010.

[Zinn+13] *D. Zinn, S. Koehler, B. Ludaescher.* Provenance Games. Work in progress

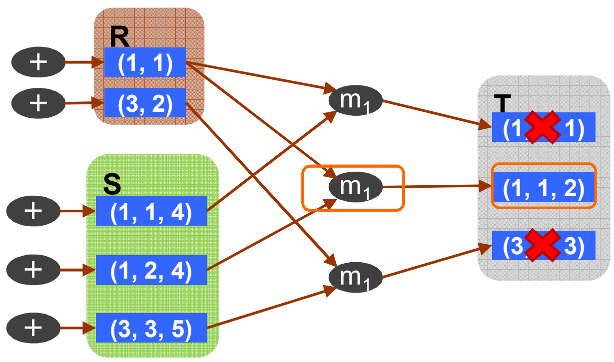
[Zimmerman+12] *A. Zimmermann, N. Lopes, A. Polleres, and U. Straccia*. A general framework for representing, reasoning and querying with annotated semantic web data. J. Web Semantics, 11, 2012.

36

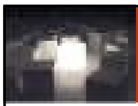
Using provenance to avoid side effects in bidirectional update exchange [WebDB08]

- Akin to view update

m_1: `*R(x,y),S(x,z) :- T(x,y,z)`



37



Recomputing materialized instances after changes to mappings and data [Green+12]

- Based on Z-relations [Green+11], where updates are represented as annotations
  - Update application can be expressed as regular query: R' = R ∪ R^δ

Source relation:

R

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

R^δ

b	**a**	**-1**
c	**d**	**+2**

deletion

insertion

- View maintenance and view adaptation can then be cast as applications of rewriting queries using materialized views
  - Uses provenance to "separate" disjuncts of a union, or "recover" values projected away and enable new (and possibly more efficient) rewritings
  - DBToaster [Koch+10] uses a similar approach for incremental view maintenance and query evaluation