Provenance for Database Transformations

Val Tannen
University of Pennsylvania

Joint work with

J.N. Foster    T.J. Green    G. Karvounarakis    Z. Ives
Cornell        UC Davis      LogicBlox and ICS-FORTH    UPenn
Data Provenance

provenance, n.

The fact of coming from some particular source or quarter; origin, derivation [Oxford English Dictionary]

• Data provenance [BunemanKhannaTan 01]: aims to explain how a particular result (in an experiment, simulation, query, workflow, etc.) was derived.

• Most science today is data-intensive. Scientists, eg., biologists, astronomers, worry about data provenance all the time.
Provenance? Lineage? Pedigree?

• Cf. Peter Buneman:
  
  – Pedigree is for **dogs**
  
  – Lineage is for **kings**
  
  – Provenance is for **art**

• For data, let’s be artistic (artsy?)
Database transformations?

- Queries
- Views
- ETL tools
- Schema mappings (as used in data exchange)
The story of database provenance

• As opposed to workflow provenance, another story. Both waiting to merge (recent progress)!

• Motivated by data integration  [WangMadnick 90, LeeBressanMadnick 98]

• Motivated by data warehousing, “lineage”  [CuiWidomWiener 00, Cui Thesis 01, etc.]

• Motivated by scientific data management, “why- and where-provenance”  [BunemanKhannaTan 01, etc.]

• Excellent accounts of the story in Buneman+ PODS 08 keynote and in Tan+ tutorials, edited collections, and recent journal article
My own journey to the study of provenance

• Working on the integration of genomics databases, since 1992

• At Penn with Peter Buneman and Wang-Chiew Tan, around 1999: “provenance is a form of annotation”.
  (They also studied other forms of annotation, such as time.)
  But I was preoccupied with other things...

• At Penn with Zack Ives, around 2005, I joined his project Orchestra: motivated by data sharing

• Working in phyloinformatics, since 2006, very interesting provenance problems
Teaser

Annotations capture ...

- Provenance
- Uncertainty (conditional tables [ImielinskiLipski 84])
- Trust scores
- Security
- Multiplicity (bag semantics)
This talk is based on the following papers

“Provenance semirings”
[GreenKarvounarakis&T PODS 07]

“Update exchange with mappings and provenance”
[GreenKarvounarakisIves&T VLDB 07]

“Annotated XML: queries and provenance”
[FosterGreen&T PODS 08]

“Containment of conjunctive queries on annotated relations”
[Green ICDT 09]

See also the dissertations of T.J. Green and G. Karvounarakis, University of Pennsylvania 2009.
Rounds

• What’s with the semirings? Annotation propagation

• Housekeeping in the zoo of provenance models

• Beyond tuple annotation

• The fundamental property and its applications

• Queries that annotate

• Datalog
• What’s with the semirings? Annotation propagation
  [GK&T PODS 07, GKI&T VLDB 07]

• Housekeeping in the zoo of provenance models

• Beyond tuple annotation

• The fundamental property and its applications

• Queries that annotate

• Datalog
Propagating annotations through database operations

The annotation $p \cdot r$ means joint use of data annotated by $p$ and data annotated by $r$.
Another way to propagate annotations

The annotation $p + r$ means alternative use of data
Another use of $+$

+ means alternative use of data
An example in positive relational algebra (SPJU)

For selection we multiply with two special annotations, 0 and 1

\[ Q = \sigma_{C=e} \pi_{AC}( \pi_{AC}R \bowtie \pi_{BC}R \cup \pi_{AB}R \bowtie \pi_{BC}R ) \]

\[
\begin{array}{ccc}
A & B & C \\
\hline
a & b & c \\
d & b & e \\
f & g & e \\
\end{array}
\]

\[
\begin{array}{ccc}
A & C \\
\hline
a & c \\
a & e \\
d & c \\
f & e \\
\end{array}
\]

\[
\begin{align*}
p & \cdot p + p \cdot p & \cdot 0 \\
p & \cdot r & \cdot 1 \\
r & \cdot p & \cdot 0 \\
(r \cdot r + r \cdot s + r \cdot r) & \cdot 1 \\
(s \cdot s + s \cdot r + s \cdot s) & \cdot 1 \\
\end{align*}
\]
Summary so far

A space of annotations, $K$

$K$-relations: every tuple annotated with some element from $K$.

Binary operations on $K$: $\cdot$ corresponds to joint use (join), and $+$ corresponds to alternative use (union and projection).

We assume $K$ contains special annotations 0 and 1.

“Absent” tuples are annotated with 0!

1 is a “neutral” annotation (no restrictions).

Algebra of annotations? What are the laws of $(K, +, \cdot, 0, 1)$?
Annotated relational algebra

• DBMS query optimizers assume certain equivalences:
  – union is associative, commutative
  – join is associative, commutative, distributes over union
  – projections and selections commute with each other and with union and join (when applicable)
  – Etc., but no \( R \bowtie R = R \cup R = R \) (i.e., no idempotence, to allow for bag semantics)

• Equivalent queries should produce same annotations!

**Proposition.** Above identities hold for queries on \( K \)-relations iff \((K, +, \cdot, 0, 1)\) is a **commutative semiring**
What is a commutative semiring?

An algebraic structure \((K, +, \cdot, 0, 1)\) where:

- \(K\) is the domain
- \(+\) is associative, commutative, with \(0\) identity
- \(\cdot\) is associative, with \(1\) identity
- \(\cdot\) distributes over \(+\)
- \(a \cdot 0 = 0 \cdot a = 0\)

- \(\cdot\) is also **commutative**

Unlike ring, no requirement for inverses to \(+\)
Back to the example

\[
\begin{align*}
R & \quad A & B & C \\
   & a & b & c & p \\
   & d & b & e & r \\
   & f & g & e & s \\
\end{align*}
\]

\[
\begin{align*}
Q & \quad A & C \\
   & a & c & (p \cdot p + p \cdot p) \cdot 0 \\
   & a & e & p \cdot r \cdot 1 \\
   & d & c & r \cdot p \cdot 0 \\
   & d & e & (r \cdot r + r \cdot s + r \cdot r) \cdot 1 \\
   & f & e & (s \cdot s + s \cdot r + s \cdot s) \cdot 1 \\
\end{align*}
\]
Polynomials with coefficients in $\mathbb{N}$ and annotation tokens as indeterminates $p, r, s$ capture a very general form of **provenance**.
Provenance reading of the polynomials

- three different ways to derive $d e$
- two of the ways use only $r$
- but they use it twice
- the third way uses $r$ once and $s$ once
Low-hanging fruit: deletion propagation

We used this in **Orchestra** [VLDB07] for update propagation

We used this in Orchestra [VLDB07] for update propagation

![Diagram showing deletion propagation](image)

Delete d b e from R?

Set \( r = 0 \)!
But are there useful commutative semirings?

<table>
<thead>
<tr>
<th>Semiring</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(\mathbb{B}, \land, \lor, \top, \bot)$</td>
<td>Set semantics</td>
</tr>
<tr>
<td>$(\mathbb{N}, +, \cdot, 0, 1)$</td>
<td>Bag semantics</td>
</tr>
<tr>
<td>$(\mathcal{P}(\Omega), \cup, \cap, \emptyset, \Omega)$</td>
<td>Probabilistic events [FuhrRölleke 97]</td>
</tr>
<tr>
<td>$(\text{BoolExp}(X), \land, \lor, \top, \bot)$</td>
<td>Conditional tables (c-tables) [ImielinskiLipski 84]</td>
</tr>
<tr>
<td>$(\mathbb{R}_+, \min, +, \infty, 0)$</td>
<td>Tropical semiring (cost/distrust score/confidence need)</td>
</tr>
<tr>
<td>$(\mathbb{A}, \min, \max, 0, P)$ where $\mathbb{A} = P &lt; C &lt; S &lt; T &lt; 0$</td>
<td>Access control levels [PODS8]</td>
</tr>
</tbody>
</table>
• What’s with the semirings? Annotation propagation

• **Housekeeping in the zoo of provenance models**
  [GK&T PODS 07, FG&T PODS 08, Green ICDT 09]

• Beyond tuple annotation

• The fundamental property and its applications

• Queries that annotate

• Datalog
Semirings for various models of provenance (1)

Lineage  [CuiWidomWiener 00 etc.]

Sets of contributing tuples

Semiring:  \((\operatorname{Lin}(X), \cup, \cup^*, \emptyset, \emptyset^*)\)
Semirings for various models of provenance (2)

(Witness, Proof) **why-provenance**
[BunemanKhannaTan 01] & [Buneman+ PODS08]

Sets of witnesses (w. =set of contributing tuples)

**Semiring:** \((\text{Why}(X), \cup, \uplus, \emptyset, \{\emptyset\})\)
Semirings for various models of provenance (3)

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>c</td>
<td>p</td>
</tr>
<tr>
<td>d</td>
<td>b</td>
<td>e</td>
<td>r</td>
</tr>
<tr>
<td>f</td>
<td>g</td>
<td>e</td>
<td>s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>d</td>
<td>e</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Minimal witness **why-provenance**
[BunemanKhannaTan 01]

Sets of minimal witnesses

**Semiring:** \((\text{PosBool}(X), \land, \lor, \top, \bot)\)
Semirings for various models of provenance (4)

\[
\begin{array}{ccc}
R & A & B & C \\
\hline
a & b & c & p \\
d & b & e & r \\
f & g & e & s
\end{array}
\quad \quad \quad
\begin{array}{cc}
Q & A & C \\
\hline
\cdots & \{r\}, \{r\}, \{r, s\}
\end{array}
\]

**Trio lineage**  [Das Sarma+ 08]

Bags of sets of contributing tuples (of witnesses)

**Semiring:** \((\text{Trio}(X), +, \cdot, 0, 1)\) (defined in [Green, ICDT 09])

Notation:

\{ \} set

[ ] bag
Semirings for various models of provenance (5)

<table>
<thead>
<tr>
<th>R</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>c</td>
<td></td>
</tr>
<tr>
<td>d</td>
<td>b</td>
<td>e</td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>g</td>
<td>e</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Q</th>
<th>A</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>⋯</td>
<td>d</td>
<td>e</td>
</tr>
<tr>
<td>⋯</td>
<td></td>
<td>⋯</td>
</tr>
</tbody>
</table>

Sets of bags of contributing tuples

Semiring: \((\mathbb{B}[X], +, \cdot, 0, 1)\)

Polynomials with boolean coefficients \([\text{Green, ICDT 09}]\) \((\mathbb{B}[X]\text{-provenance})\)

Notation:
- \{\} set
- \[[\ ]\] bag
Semirings for various models of provenance (6)

**R**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>c</td>
<td>p</td>
</tr>
<tr>
<td>d</td>
<td>b</td>
<td>e</td>
<td>r</td>
</tr>
<tr>
<td>f</td>
<td>g</td>
<td>e</td>
<td>s</td>
</tr>
</tbody>
</table>

**Q**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d</td>
<td>e</td>
<td>[[r,r], [r,r], [r, s]]</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Provenance polynomials  

[GKT, PODS 07]  

(\(\mathbb{N}[X]\)-provenance)

Bags of bags of contributing tuples

**Semiring:**  

\((\mathbb{N}[X], +, \cdot, 0, 1)\)
A provenance hierarchy

most informative

least informative

Why(X)

Lin(X)  PosBool(X)

Trio(X)

B[X]

N[X]
One semiring to rule them all... (apologies!)

Example: $2x^2y + xy + 5y^2 + z$

```
\[
\begin{align*}
\mathbb{N}[X] & \quad \text{drop coefficients} \\
\mathbb{B}[X] & \quad x^2y + xy + y^2 + z \\
\text{Trio}(X) & \quad \text{drop exponents} \\
\text{Why}(X) & \quad 3xy + 5y + z \\
\text{Lin}(X) & \quad \text{drop both exp. and coeff.} \\
\text{PosBool}(X) & \quad xy + y + z \\
\text{apply absorption} \\
& \quad (ab + b = b) \\
& \quad y + z
\end{align*}
\]
```

A path downward from $K_1$ to $K_2$ indicates that there exists an onto (surjective) semiring homomorphism $h : K_1 \rightarrow K_2$
Using homomorphisms to relate models

Example: $2x^2y + xy + 5y^2 + z$

Homomorphism?

$h(x+y) = h(x) + h(y)$  \hspace{1cm} $h(xy) = h(x)h(y)$  \hspace{1cm} $h(0) = 0$  \hspace{1cm} $h(1) = 1$

Moreover, for these homomorphisms $h(x) = x$
Containment and Equivalence [Green ICDT 09]

Arrow from $K_1$ to $K_2$ indicates $K_1$ containment (equivalence) implies $K_2$ cont. (equiv.)

All implications not marked $\iff$ are strict
• What’s with the semirings? Annotation propagation

• Housekeeping in the zoo of provenance models

• **Beyond tuple annotation**  [FG&T PODS 08]

• The fundamental property and its applications

• Queries that annotate

• Datalog
Relation, attribute and field annotation (1)

\[
R^u
\]

\[
\begin{array}{ccc}
A^x & B^y & C^1 \\
\vdots & & \vdots \\
a^1 & b^1 & c^1 \\
\vdots & & \vdots \\
\end{array}
\]

\[
\pi_{AC}( \pi_{AB} R \bowtie ( \pi_{BC} R \cup S ))
\]

\[
\begin{array}{ccc}
A^1 & C^1 \\
\vdots & & \vdots \\
a^1 & c^1 \\
\vdots & & \vdots \\
\end{array}
\]

\[
u^2 p^2 x y^2 + uvpmxyz
\]

Neutral annotation 1 used when we don’t bother to track data.
Relation, attribute and field annotation (2)

\[ \pi_{AC}(\pi_{AB}R \bowtie (\pi_{BC}R \cup S)) \]

\[ u^2p^2xy^2 + uvpmxyz \]

We omit 1 for convenience. From now on 1 is the default.

Here, we track both relations, attributes A and B in \( R \), and the first field of \( m \).
Same value, different annotations (where-provenance)

\[ \sigma_{C=e} \pi_{AC}(\pi_{AB} R \bowtie \pi_{BC} R) \]

Track both e’s
Different field annotations produce different tuples

What happens when we add a projection on C?

\[ \pi_C \sigma_{C=e} \pi_{AC} ( \pi_{AB} R \bowtie \pi_{BC} R ) \]
When we don’t care to track so many details

Add a homomorphism $h(w) = h(z) = u$. (Add to language.)

$$h \left( \pi_C \sigma_{C=e} \pi_{AC} \left( \pi_{AB} R \bowtie \pi_{BC} R \right) \right)$$
Why vs. where

A provenance token on a field is treated like any other token. In the semiring framework the why-where distinction is blurred.
• What’s with the semirings? Annotation propagation

• Housekeeping in the zoo of provenance models

• Beyond tuple annotation

• **The fundamental property and its applications**
  [GK&T PODS 07, FG&T PODS 08, Green&T EDBTworkshop 06]

• **Queries that annotate**

• **Datalog**
Fundamental property

For every query $q$ and every homomorphism of commutative semirings $h : K_1 \rightarrow K_2$ the following "commutes":

Doesn’t always work, eg. difference.
Most important source of homomorphisms

If \( K \) is a commutative semiring, then any function on tokens, \( f : X \to K \) extends uniquely to a homomorphism \( h : \mathbb{N}[X] \to K \).

(“Extends means that \( h \) coincides with \( f \) on tokens.)

Think of \( h(pr+r^2+s^2) \) as evaluating \( pr+r^2+s^2 \) in \( K \). Examples are coming up.
An application of the fundamental property: 

**Compositionality**

Input to A: tokens $X = \{p, r, s\}$; Output of A provenance in $\mathbb{N}[X]$

Input to B: tokens $Y = \{m, n\}$; Output of B provenance in $\mathbb{N}[Y]$

Say that for data $A \rightarrow B$ \quad $p + rs = m$, \quad $prs + 2s^2 = n$

This gives $f : Y \rightarrow \mathbb{N}[X]$ which extends to $h : \mathbb{N}[Y] \rightarrow \mathbb{N}[X]$

Say that one output of B has provenance $m^2 + 2n$

Then, as an output of A composed w/ B it has provenance

$$h(m^2 + 2n) = p^2 + 4prs + r^2s^2 + 4s^2$$
More applications of the fundamental property

• Renaming provenance tokens

• Deletion: mapping some tokens to 0 (seen earlier)

• Hiding detail, increasing abstraction:
  – mapping provenance tokens, many to few (seen earlier)
  – stop tracking tokens by mapping them to 1 (neutral)
Another application: all through provenance

Because it is the free commutative semiring.

Systems (like Orchestra) can compute and maintain only polynomial provenance, which is then evaluated, as needed, to provide:

– trust scores (see next)

– access control levels (see next)

– more frugal provenance like Trio($X$), $\mathbb{B}[X]$, etc.

– and even multiplicity

Because it is the free commutative semiring.

Doesn’t work if prov. is Trio($X$) Works with $\mathbb{B}[X]$.

Works even with prov. in PosBool($X$)

Doesn’t work if prov. is $\mathbb{B}[X]$. With Trio($X$) only set-bag semantics.
Application with (dis)trust scores (1)

Semiring is \( K = (\mathbb{R}_+^\infty, \min, +, \infty, 0) \)

Tokens are \( X = \{ p, r, s \} \)

Assignment function is \( f : X \rightarrow K \) where we suppose \( p \) is completely trusted \( f(p) = 0 \), \( r \) is less trusted \( f(r) = 1.5 \), and \( s \) is untrusted \( f(s) = \infty \)

The homomorphism \( h \) that extends \( f \) computes like this:
\[
h(2r^2 + rs) = h(r \cdot r + r \cdot r + r \cdot s) =
\]
\[
= \min(f(r) + f(r), f(r) + f(r), f(r) + f(s)) =
\]
\[
= \min(1.5 + 1.5, 1.5 + 1.5, 1.5 + \infty) = 3.0
\]
Application with (dis)trust scores (2)

\((\mathbb{R}_+ \infty, \text{min}, +, \infty, 0)\)

The fundamental property

\[
\begin{array}{cc}
\text{a} & \text{b} \\
\text{d} & \text{b} \\
\text{f} & \text{g} \\
\end{array}
\]

\[
\begin{array}{c}
p \\
r \\
s \\
\end{array}
\]

\(f(p) = 0, \quad f(r) = 1.5, \quad f(s) = \infty\)

\[
\begin{array}{cc}
\text{a} & \text{b} \\
\text{d} & \text{b} \\
\text{f} & \text{g} \\
\end{array}
\]

\[
\begin{array}{c}
0 \\
1.5 \\
\end{array}
\]

“Accept tuples with score \(\leq 2.5\)”

\[
\begin{array}{cccc}
a & c & 2p^2 \\
a & e & pr \\
d & c & pr \\
d & e & 2r^2 + rs \\
f & e & 2s^2 + rs \\
\end{array}
\]

eval with homomorphisms \(h\), the extension of \(f\)

\[
\begin{array}{cccc}
a & c & 0 \\
a & e & 1.5 \\
d & c & 1.5 \\
d & e & 3.0 \\
\end{array}
\]

deleted
Application to access control

\((A, \text{min}, \text{max}, 0, P)\)  where \(A = P < C < S < T < 0\)

Suppose \(p\) is public, \(r\) is secret, \(s\) is top secret

```
\[
\begin{array}{ccc}
\text{a} & \text{b} & \text{c} & \text{p} \\
\text{d} & \text{b} & \text{e} & \text{r} \\
\text{f} & \text{g} & \text{e} & \text{s}
\end{array}
\]
```

with \(p = P, r = S, s = T\)

```
\[
\begin{array}{ccc}
\text{a} & \text{c} & 2p^2 \\
\text{a} & \text{e} & pr \\
\text{d} & \text{c} & pr \\
\text{d} & \text{e} & 2r^2 + rs \\
\text{f} & \text{e} & 2s^2 + rs
\end{array}
\]
```

Fundamental property implies that applying the clearance to the database or to the query answer yields the same result. (But only the second is actually feasible!)

```
\[
\begin{array}{ccc}
\text{a} & \text{c} & P \\
\text{a} & \text{e} & S \\
\text{d} & \text{c} & S \\
\text{d} & \text{e} & S \\
\text{f} & \text{e} & T
\end{array}
\]
```

"User with secret clearance"
Another application: uncertainty (1)

- **Possible worlds** model:
  - incomplete $K$-database = a set of $K$-instances
  - probabilistic $K$-database = a distribution on the set of all $K$-instances

- Unwieldy size! Want representation systems, like the (boolean) $c$-tables [ImielinskiLipski 84]: tables annotated with elements from the semiring $\text{BoolExp}(X)$.

- So why not $\text{Trio}(X), \mathbb{B}[X], \text{Why}(X), \mathbb{N}[X], \text{PosBool}(X)$? (provided we stick to positive queries, SPJU, no D) $\mathbb{N}[X]$ always works. For the others it depends on $K$. 
Another application: uncertainty (2)

- $\mathbb{N}[X]$ always works. For incomplete databases:
  - Take as representation table $T$ a $\mathbb{N}[X]$-relation
  - For each assignment function $f : X \rightarrow K$
    - extend to a homomorphism $h : \mathbb{N}[X] \rightarrow K$
    - use $h$ to eval. into $K$ the polynomials annotating $T$
    - thus obtaining a possible world, a $K$-relation

- The fact that this works properly (it is a strong representation system) follows from the fundamental property!
Another application: uncertainty (3)

• For probabilistic databases follow Green’s idea of pc-tables [Green&T 06]

• Again representation tables are $\mathbb{N}[X]$-relations

• Treat variables in $X$ as independent. For each variable assume a prob. distribution on the values in $K$ it can take.

• This gives a probability distribution on assignment functions $f : X \rightarrow K$, therefore on the possible worlds.
\( N[X] \) always works, for the others it depends on \( K \)

**Diagram:**
- **\( N[X] \)**
- **\( B[X] \)**
- **\( Trio(X) \)**
- **\( \mathbb{R}_{+ \infty} \)**
- **\( Why(X) \)**
- **\( PosBool(X) \)**
- **\( B \)**
- **\( A \)**

- **Tuples have cost/score**
- **Bag instances**
- **Set instances**
- **Limited to assignments \( x \to 0 \) or \( 1 \)**
- **A downward path from provenance \( P[X] \) to a blue leaf \( L \)** means that any assignment \( X \to L \) extends to a homomorphism \( P[X] \to L \)

That’s all we need!
• What’s with the semirings? Annotation propagation

• Housekeeping in the zoo of provenance models

• Beyond tuple annotation

• The fundamental property and its applications

• **Queries that annotate**  [FG&T PODS 08]

• **Datalog**
Queries that annotate

• In Orchestra [GKI&T VLDB07] we annotate schema mappings with provenance “unary operations”.

• In [FGT PODS 08] we introduced into the query language an operation of “scalar multiplication”.

• The “scalar” $k$ is from $K$ and the “vector” $S$ is a $K$-relation (or $K$-set). For $kS$ each annotation in $S$ is multiplied by $k$.

• We have also seen how useful homomorphisms are.

• This suggests an operation $hS$ where the homomorphism $h$ is applied to each annotation in $S$. 
Extending query languages to manipulate annotation/provenance (AnnotatedSQL?)

```sql
SELECT RenameH ( r.Name AS Name, s.Project AS Project )
FROM r IN db1 Employee, s IN db2 Project
WHERE ...
DEFINE RenameH AS ( db1 -> PersonnelDB, db2 -> BillingDB )
```

The tuples in the query answer have provenances in terms of the tokens PersonnelDB and BillingDB as well as tokens from the annotations of the tuples in Employee and Project.
• What’s with the semirings? Annotation propagation

• Housekeeping in the zoo of provenance models

• Beyond tuple annotation

• The fundamental property and its applications

• Queries that annotate

• **Datalog**  [GK&T PODS 07]
**K-Datalog?**

n-ary \( K \)-relations: functions \( R : U \rightarrow K \) \( R \) in \( K^U \)

where \( U \) is the set of all \( n \)-tuples over some domain, such that

\[
\text{supp}(R) = \{ t \mid R(t) \neq 0 \} \quad \text{is finite}
\]

The immediate consequence operator of a program \( P \) (incorporates edb) in \( K \)-relation semantics

\[
T_P : K^U \rightarrow K^U
\]

For what semirings \( K \) does \( T_P \) have a fixpoint?

Recall that \( T_P \) computes annotations that are defined by polynomials.
ω-continuous semirings

Natural preorder: \( x \leq y \) iff there exists \( z \) s.t. \( x + z = y \)

Naturally ordered semiring: when \( \leq \) is an order relation (all semirings seen here are naturally ordered)

ω-completeness: when \( x_0 \leq x_1 \leq \ldots \leq x_n \leq \ldots \) have l.u.b.'s

ω-continuity when + and \( \cdot \) preserve those l.u.b.'s
Least fixpoints and formal power series

Over \( \omega \)-continuous semirings functions defined by polynomials have least fixpoints (usual definition) hence:

\[
\text{fix}(P) = \bigcup_{k \geq 0} T^k_P(0)
\]

Most of the semirings that interest us are already \( \omega \)-continuous.

\((\mathbb{N}, +, \cdot, 0, 1)\) is not, but its “completion” \((\mathbb{N}^\infty = \mathbb{N} \cup \{\infty\}, +, \cdot, 0, 1)\) is.

For provenance, the completion of \( \mathbb{N}[X] \) is not \( \mathbb{N}^\infty[X] \). Instead of (finite) polynomials we need (possibly infinite) formal power series. They form a semiring, \( \mathbb{N}^\infty[[X]] \).
Proof semantics

By considering all (possibly infinitely many) proof trees $\tau$ and the annotations of the tuples on their leaves:

$$\text{proof}(P)(t) = \sum_{\tau \text{ yields } t} \left( \prod_{t' \text{ leaf}(\tau)} R(t') \right)$$

We have $\text{proof}(P) = \text{fix}(P)$

There is also an equivalent “least model” semantics [G09]

Also, $\text{supp}(\text{fix}(P))$ is finite, and equals the (usual) $\mathcal{B}$-relations semantics (set semantics).
An equivalent perspective

\[
\begin{align*}
T(X, Y) &::= E(X, Y) \\
T(X, Y) &::= T(X, Z), T(Z, Y)
\end{align*}
\]

Polynomials are the provenance of the immediate consequence operator.

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>m</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>c</td>
<td>n</td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>p</td>
</tr>
<tr>
<td>b</td>
<td>d</td>
<td>r</td>
</tr>
<tr>
<td>d</td>
<td>d</td>
<td>s</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
x &= m + yz \\
y &= n \\
z &= p \\
u &= r + uv \\
v &= s + v^2 \\
w &= xu + wv \\
t &= zu + tv
\end{align*}
\]

Solve!
Solving in the power series semiring

\[ x = m + np \]
\[ y = n \]
\[ z = p \]
\[ v = s + s^2 + 2s^3 + 5s^4 + 14s^5 + \ldots \]
\[ u = r v^* \]
\[ w = r(m+np)(v^*)^2 \]
\[ t = pr(v^*)^2 \]

where

\[ v^* \triangleq 1 + v + v^2 + v^3 + \ldots \]

In general the coefficients are from \( \mathbb{N}^\infty \)
Decidability results

- Given $t \in q(I)$, it is **decidable** whether the provenance of $t$ is a proper (infinite) power series. (Generalizing a result in [Mumick Shmueli 93] about bag semantics for Datalog)

- Given $t \in q(I)$, and a monomial $\mu$, the coefficient of $\mu$ in the power series that is the provenance of $t$ is **computable** (including when it is $\infty$).
• From CFG ambiguity, we know that testing whether all coefficients are $\leq 1$ is undecidable.

• However, testing whether all coefficients are $\neq \infty$ is decidable.
Extensions and sequels (1)

• Implementation in ORCHESTRA
  [GKI&T VLDB 07, KarvounarakisIves WebDB 08]
  – Schema mappings are Datalog with Skolem functions, weakly acyclic recursion
  – Provenance polynomials are represented as a graph with two kinds of nodes, tuples and mappings. More economical: sharing common subexpressions

• Provenance information is data too! Provenance query language on the Orchestra graph provenance representation; also allows evaluation is particular semirings: trust, security, etc.
  [KarvounarakisIves&T SIGMOD 10]
Extensions and sequels (2)

• Complex value data, Nested Relational Calculus, trees, unordered XML and XQuery [FG&T PODS 08].

• Comprehensive study of SPJ (conjunctive queries) and SPJU (non-recursive Datalog) containment and equivalence under annotated relations semantics [Green ICDT 09]

• Relations annotated with integers (positive and negative), semantics and reformulation with views for the full relational algebra [GI&T ICDT 09]
A tiny bit of related work

• Formal languages [ChomskiSchützenberger63]

• CSP (Bistarelli et al.)

• Debugging schema mappings [ChiticariuTan06]

• “Closed” semirings used in Datalog optimization (Consens&Mendelzon)

• Lots more related work on data provenance, bag semantics, NLP, programming languages, etc.
Conclusions and Further Work

General and versatile framework.
Dare I call it “semiring-annotated databases”? Many apparent applications.
We clarified the hazy picture of multiple models for database provenance.
Essential component of the data sharing system Orchestra.

- Dealing with **negation** (progress: [Geerts&Poggi 08, GI&T ICDT 09])
- Dealing with **aggregates** (progress: [T ProvWorkshop 08])
- Dealing with **order** (speculations...)

03/24/10
EDBT Keynote, Lausanne
Thank you!