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 [WangMadnic**k** 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**

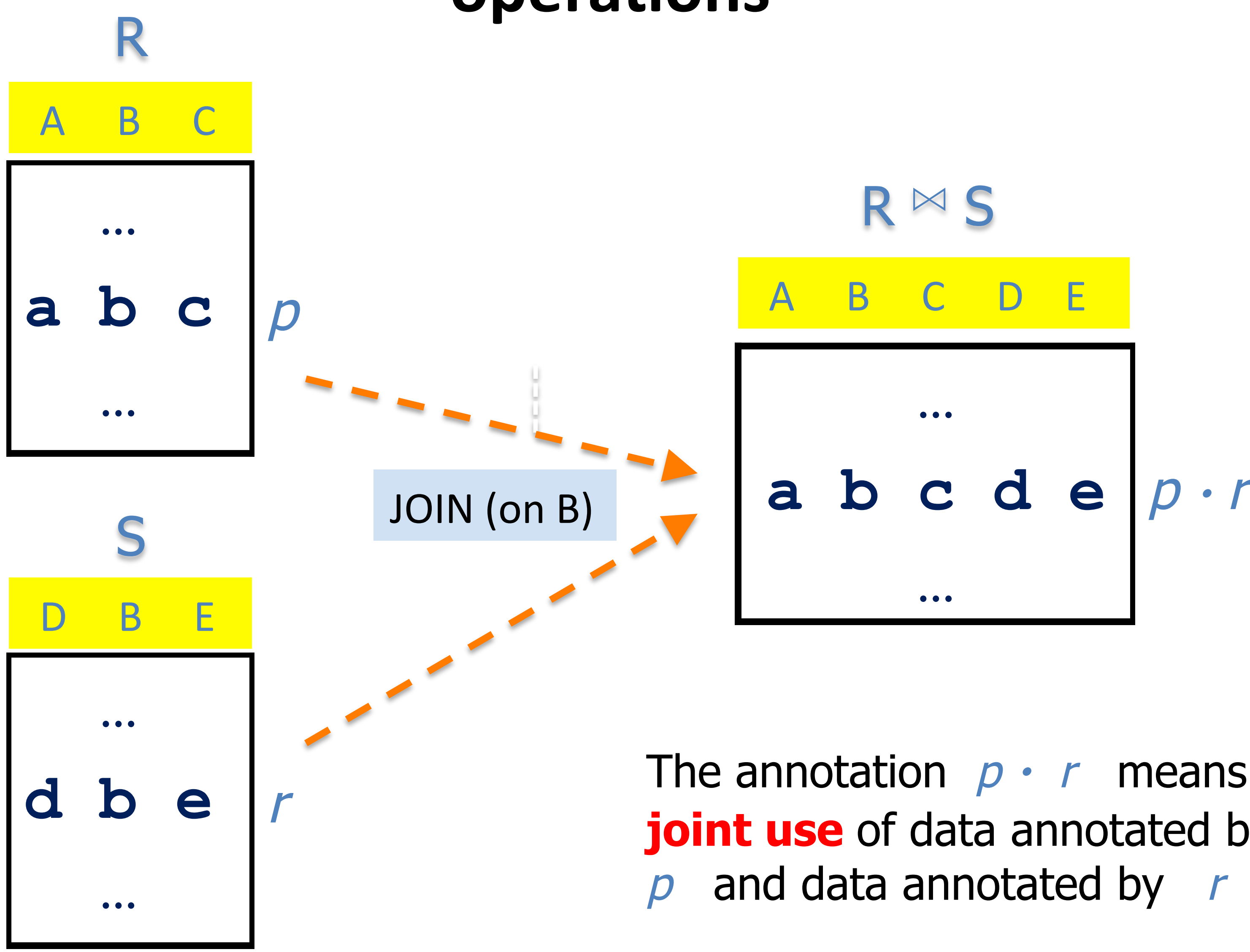
- **Beyond tuple annotation**

- **The fundamental property and its applications**

- **Queries that annotate**

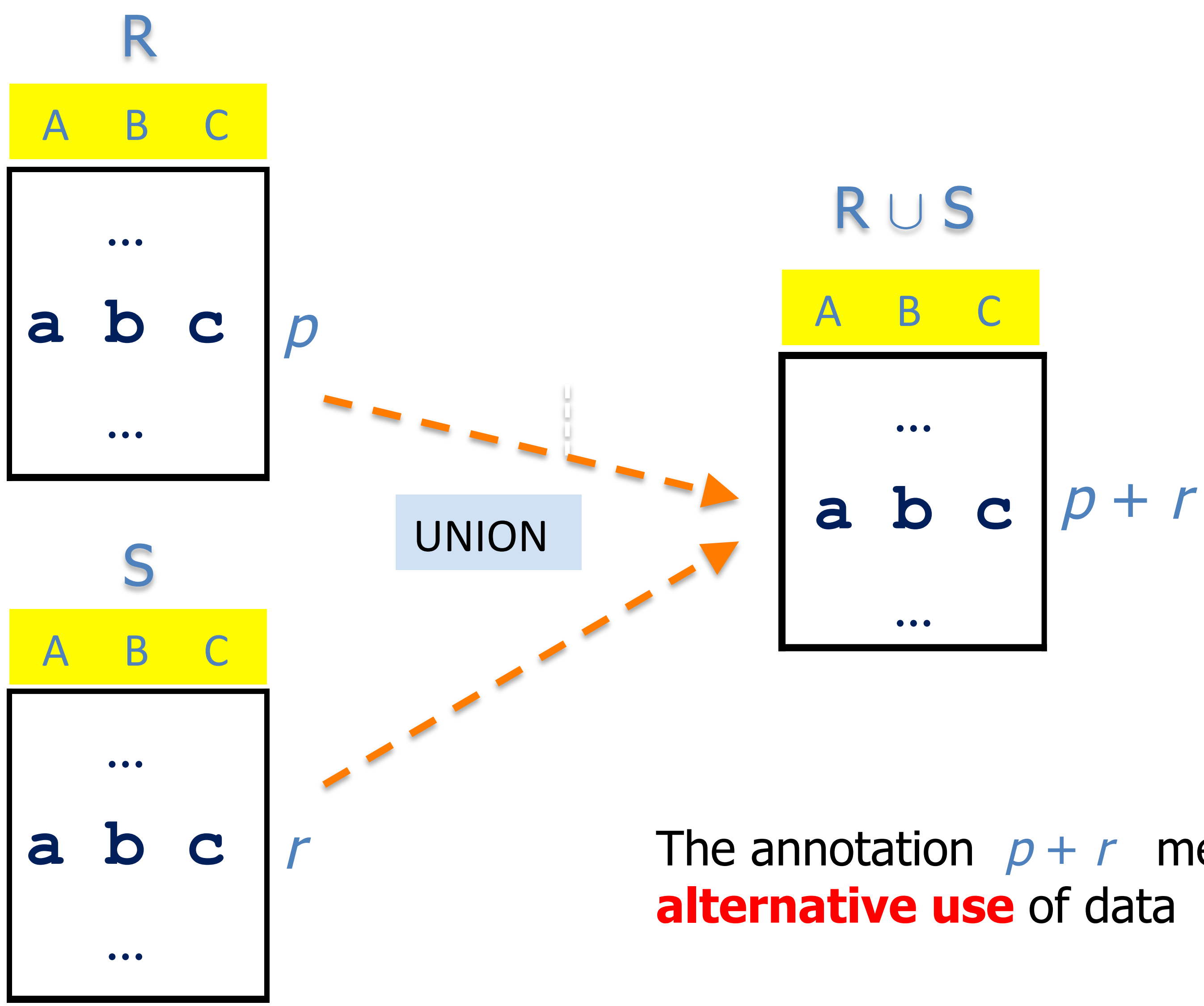
- **Datalog**

Propagating annotations through database operations



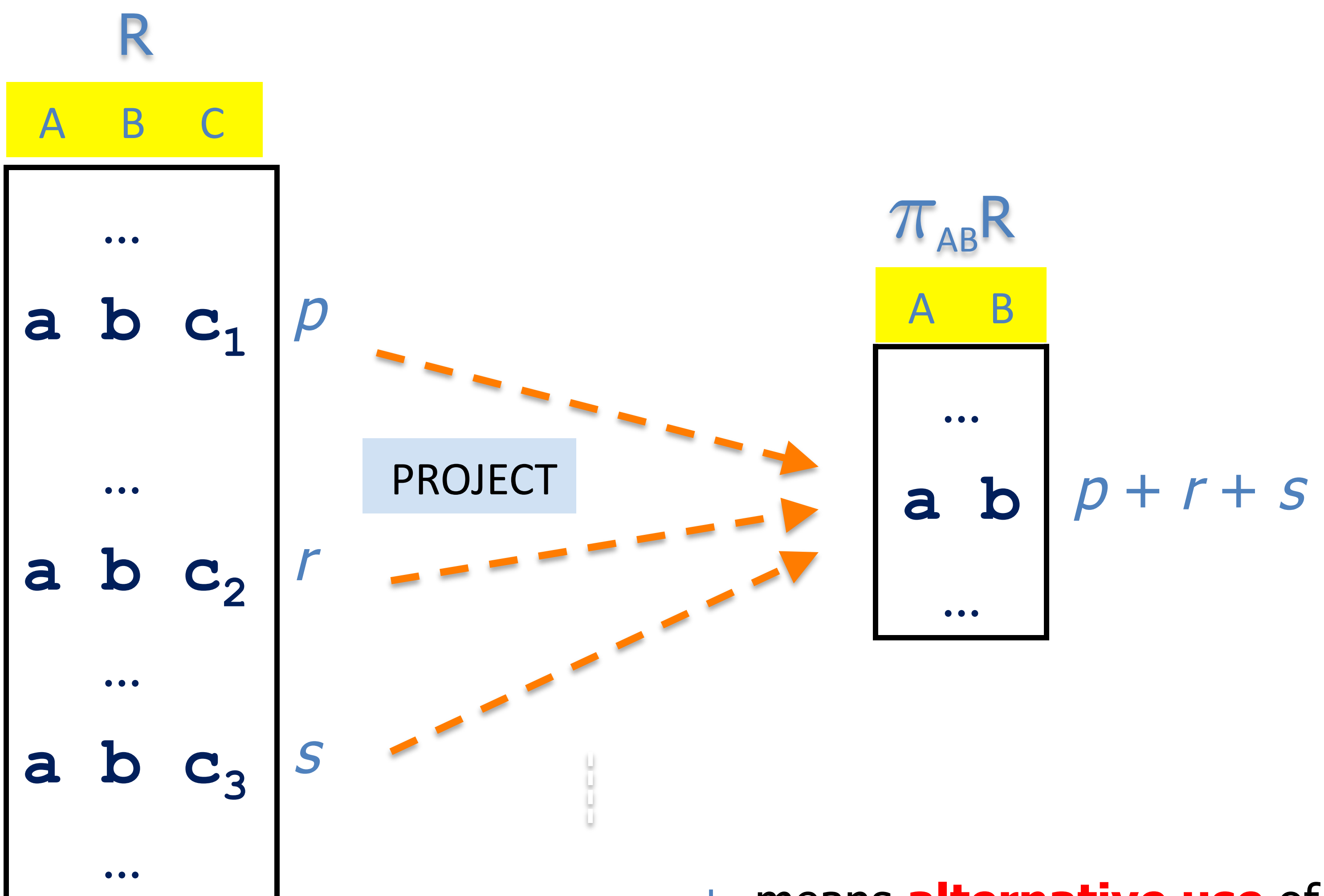
The annotation $p \cdot r$ means **joint use** of data annotated by p and data annotated by r

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)

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

R

A	B	C	
a	b	c	p
d	b	e	r
f	g	e	s

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	$(s \cdot s + s \cdot r + s \cdot s) \cdot 1$

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

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 $+$
- $a \cdot 0 = 0 \cdot a = 0$

semiring

- $\cdot$ is also **commutative**

Unlike ring, no requirement for inverses to $+$

Back to the example

R

A	B	C	
a	b	c	p
d	b	e	r
f	g	e	s

Q

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	$(s \cdot s + s \cdot r + s \cdot s) \cdot 1$

Using the laws: polynomials

R

A	B	C	
a	b	c	p
d	b	e	r
f	g	e	s

Q

A	C	
a	e	pr
d	e	$2r^2 + rs$
f	e	$rs + 2s^2$

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

R

A	B	C	
a	b	c	p
d	b	e	r
f	g	e	s

Q

A	C	
a	e	pr
d	e	$2r^2 + rs$
f	e	$rs + 2s^2$

- 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

R

A	B	C	
a	b	c	p
d	b	e	r
f	g	e	s

Q

A	C	
a	e	pr
d	e	$2r^2 + rs$
f	e	$rs + 2s^2$

Q

A	C	
a	e	0
d	e	0
f	e	$2s^2$

Q

A	C	
f	e	$2s^2$

Delete `d b e` from R ?

Set $r = 0$!

But are there useful commutative semirings?

($\mathbb{B}$, $\wedge$, $\vee$, $\top$, $\bot$)	**Set semantics**
($\mathbb{N}$, +, $\cdot$, 0, 1)	**Bag semantics**
($\mathcal{P}(\Omega)$, $\cup$, $\cap$, $\emptyset$, Ω)	Probabilistic events [FuhrRölleke 97]
(BoolExp(X), $\wedge$, $\vee$, $\top$, $\bot$)	Conditional tables (c-tables) [ImielinskiLipski 84]
($\mathbb{R}_+^\infty$, min, +, ∞, 0)	Tropical semiring (cost/distrust score/confidence need)
($\mathbb{A}$, min, max, 0, P) where $\mathbb{A}$ = P < C < S < T < 0	Access control levels [PODS8]

top secret

public

- 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 (probably not enough time…)**

Semirings for various models of provenance (1)

R

A	B	C	
a	b	c	p
d	b	e	r
f	g	e	s

Q

A	C	
…		
d	e	$\{r, s\}$
…		

Lineage [CuiWidomWiener 00 etc.]

Sets of contributing tuples

Semiring: $(\mathrm{Lin}(X), \cup, \cup^*, \emptyset, \emptyset^*)$

Semirings for various models of provenance (2)

R

A	B	C	
a	b	c	p
d	b	e	r
f	g	e	s

Q

A	C	
…		
d	e	$\{\{r\}, \{r, s\}\}$
…		

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

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

Semiring: $(Why(X), \cup, \Cup, \emptyset, \{\emptyset\})$

Semirings for various models of provenance (3)

R

A	B	C	
a	b	c	*p*
d	b	e	*r*
f	g	e	*s*

Q

A	C	
…		
d	e	$\{\{r\}\}$
…		

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

Sets of minimal witnesses

Semiring: (PosBool(X), $\wedge$, $\vee$, $\top$, $\bot$)

Semirings for various models of provenance (4)

R

A	B	C	
a	b	c	p
d	b	e	r
f	g	e	s

Q

A	C	
…		
d	e	$[\{r\}, \{r\}, \{r, s\}]$
…		

Notation:

{ } set

[ ] bag

Trio lineage [Das Sarma+ 08]

Bags of sets of contributing tuples (of witnesses)

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

Semirings for various models of provenance (5)

R

A	B	C	
a	b	c	p
d	b	e	r
f	g	e	s

Q

A	C	
…		
d	e	$\{[r,r], [r, s]\}$
…		

Notation:
{ } set
[ ] bag

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

Sets of bags of contributing tuples

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

Semirings for various models of provenance (6)

R

A	B	C	
a	b	c	p
d	b	e	r
f	g	e	s

Q

A	C	
…		
d	e	$[[r,r], [r,r], [r, s]]$
…		

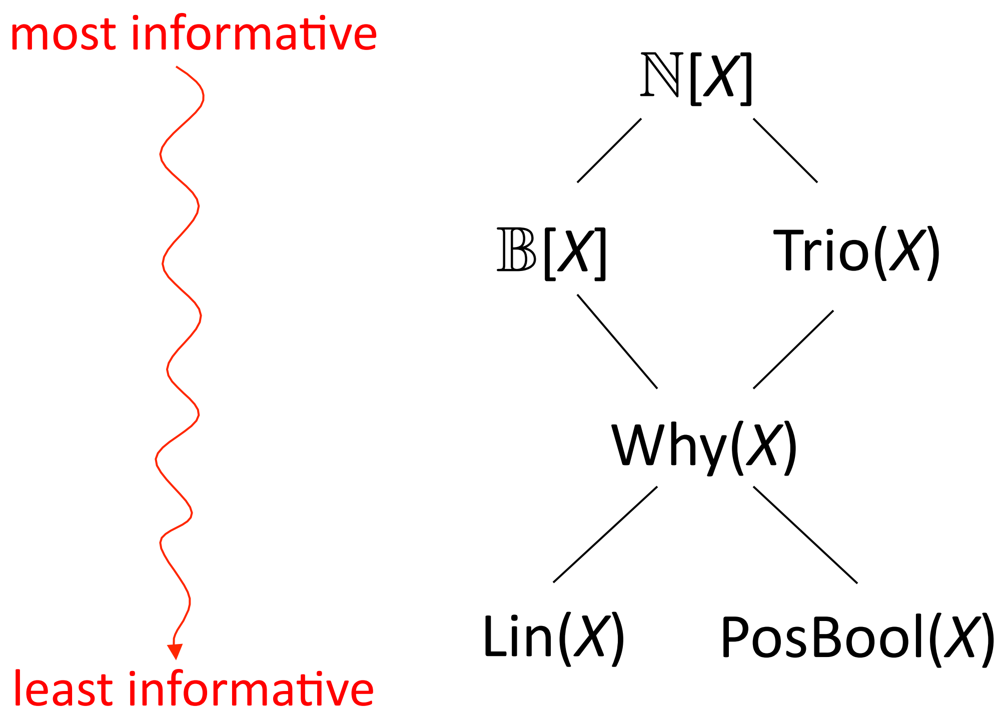
Provenance polynomials [GKT, PODS 07]

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

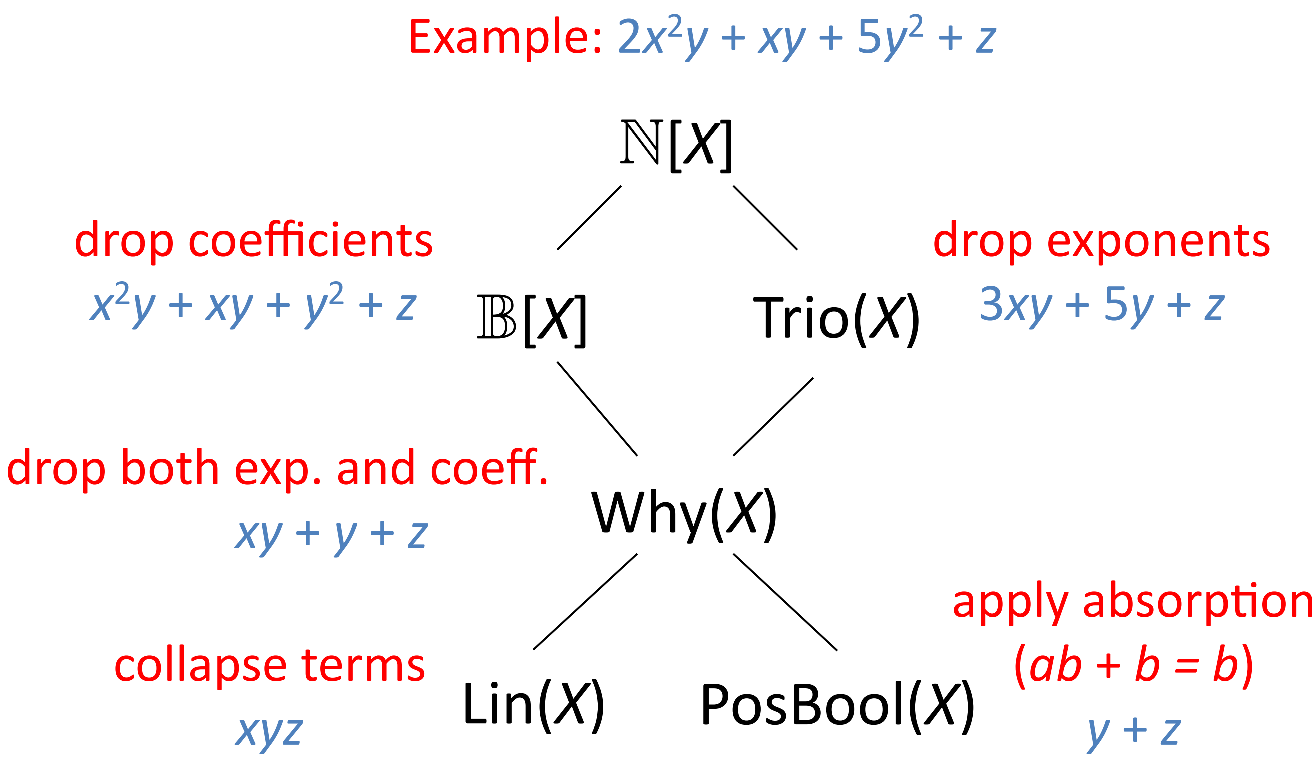
Bags of bags of contributing tuples

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

A provenance hierarchy



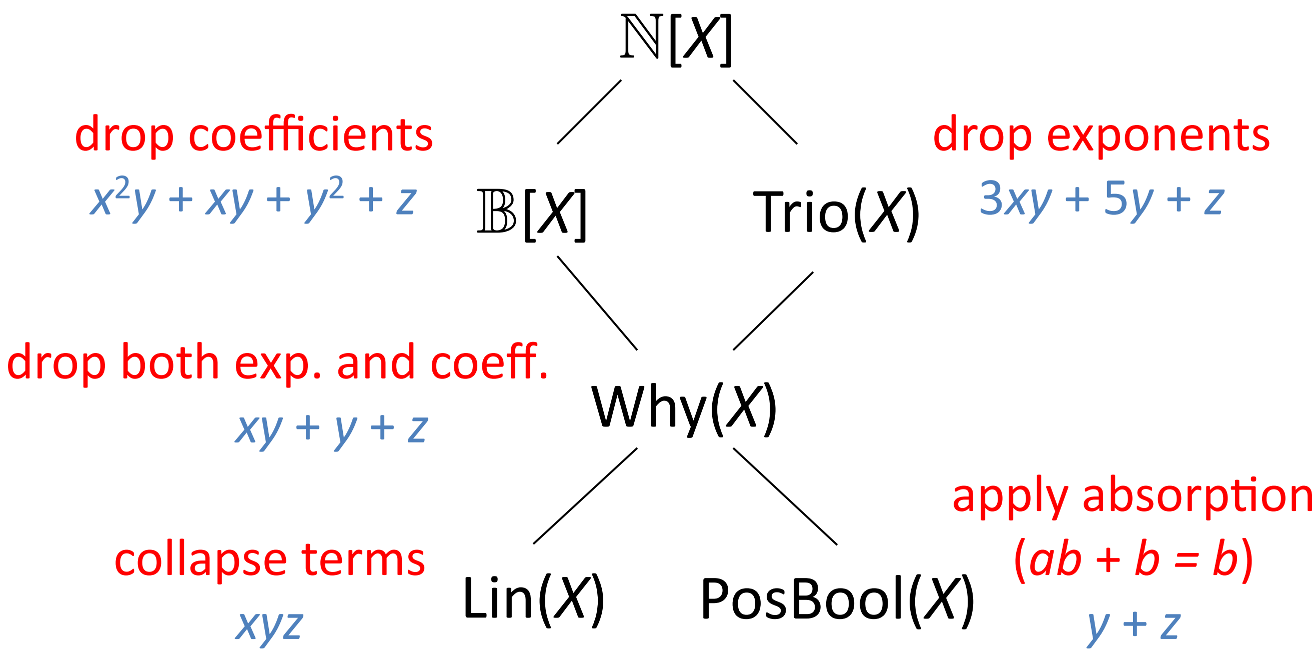
One semiring to rule them all… (apologies!)



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?

$h(x+y) = h(x)+h(y)$ $\quad h(xy)=h(x)h(y)$ $\quad h(0)=0$ $\quad h(1)=1$

Moreover, for these homomorphisms $h(x)= x$

Containment and Equivalence [Green ICDT 09]

$$\begin{tikzcd}
\mathbb{B}[X] \arrow[r, leftrightarrow] \arrow[d] & \mathbb{N}[X] \\
\mathrm{Why}(X) \arrow[r, leftrightarrow] \arrow[d] & \mathrm{Trio}(X) \\
\mathbb{N} \arrow[d] & \\
\mathrm{Lin}(X) \arrow[d] & \\
\mathrm{PosBool}(X) \arrow[r, leftrightarrow] & \mathbb{B}
\end{tikzcd}$$

SPJ containment

$$\begin{tikzcd}
\mathbb{B}[X] \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] & \mathbb{N}[X] \arrow[d, leftrightarrow] \\
\mathrm{Why}(X) \arrow[r, leftrightarrow] \arrow[d] & \mathrm{Trio}(X) \arrow[d, leftrightarrow] \\
\mathrm{Lin}(X) \arrow[d] & \mathbb{N} \\
\mathrm{PosBool}(X) \arrow[r, leftrightarrow] & \mathbb{B}
\end{tikzcd}$$

SPJ equivalence

$$\begin{tikzcd}
\mathbb{B}[X] \arrow[d] & \mathbb{N}[X] \arrow[l] \arrow[d] \\
\mathrm{Why}(X) \arrow[d] & \mathrm{Trio}(X) \arrow[l] \arrow[d] \\
\mathrm{Lin}(X) \arrow[d] & \mathbb{N} \arrow[l] \\
\mathrm{PosBool}(X) \arrow[r, leftrightarrow] & \mathbb{B}
\end{tikzcd}$$

SPJU containment

$$\begin{tikzcd}
\mathbb{N} \arrow[r, leftrightarrow] \arrow[d] & \mathbb{N}[X] \arrow[d, leftrightarrow] \\
\mathbb{B}[X] \arrow[d] & \mathrm{Trio}(X) \\
\mathrm{Why}(X) \arrow[d] & \\
\mathrm{Lin}(X) \arrow[d] & \\
\mathrm{PosBool}(X) \arrow[r, leftrightarrow] & \mathbb{B}
\end{tikzcd}$$

SPJU equivalence

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

All implications not marked $\leftrightarrow$ 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

A^x	B^y	C^1	
…			
a^1	b^1	c^1	p
…			

S^v

B^1	C^1	
…		
b^z	c^1	m
…		

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

A^1	C^1	
…		
a^1	c^1	$u^2p^2xy^2 + uvpmxyz$
…		

Neutral annotation 1 used when we don't bother to track data.

Relation, attribute and field annotation (2)

R^u

A^x	B^y	C	
…			
a	**b**	**c**	p
…			

S^v

B	C	
…		
b^z	**c**	m
…		

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

A	C	
…		
a	**c**	$u^2p^2xy^2 + uvpmxyz$
…		

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)

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

R

A	B	C	
a	b	c	*p*
d	b	e^{z}	*r*
f	g	e^{w}	*s*

Track both **e**'s

A	C	
a	e^{z}	*pr*
d	e^{z}	r^2
f	e^{w}	s^2

Different field annotations produce different tuples

What happens when we add a projection on C ?

R

A	B	C	
a	b	c	p
d	b	e^z	r
f	g	e^w	s

$$\pi_C \, \sigma_{C=e} \pi_{AC}(\pi_{AB} R \bowtie \pi_{BC} R)$$

C	
e^z	$pr + r^2$
e^w	s^2

When we don't care to track so many details

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

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

R

A	B	C	
a	b	c	p
d	b	e^z	r
f	g	e^w	s

C	
$\cancel{e^z}$	$pr + r^2$
$\cancel{e^w}$	s^2

C	
e^u	$pr + r^2 + s^2$

Why vs. where

R^u

A^x	B^y	C	
…			
a	b	c	p
…			

S^v

B	C	
…		
b^z	c	m
…		

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

A	C	
…		
a	c	$u^2p^2xy^2 + uvpmxyz$
…		

where-provenance

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**

Doesn't always work, eg. difference.

Fundamental property

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

$$\begin{tikzcd} K_1\text{-data} \arrow[r, "h"] \arrow[d, "q"'] & K_2\text{-data} \arrow[d, "q"] \\ K_1\text{-data} \arrow[r, "h"] & K_2\text{-data} \end{tikzcd}$$

Most important source of homomorphisms

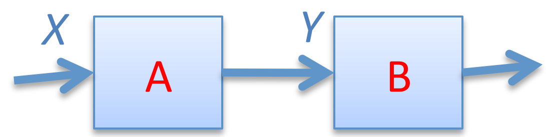
If K is a commutative semiring, then any function on tokens, $f : X \to K$ extends uniquely to a homomorphism $h : \mathbb{N}[X] \to K$.

It's the free commutative semiring generated by *X*.
The "free" one rules them all!

("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 $A \to B$ $p+rs = m,$ $prs+2s^2 = n$

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

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)

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

- access control levels (see next)

  Works even with prov. in PosBool(X)

- more frugal provenance like Trio(X), $\mathbb{B}[X]$, etc.
- and even multiplicity

  Doesn't work if prov. is $\mathbb{B}[X]$. With Trio(X) only set-bag semantics

Application with (dis)trust scores (1)

Semiring is $K = (\mathbb{R}_+^\infty, \min, +, \infty, 0)$ (a.k.a. insurance costs)

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:

$h\,(2r^2 + rs\,) = h\,(\,r{\cdot}r + r{\cdot}r + r{\cdot}s\,) =$

addition of reals

$= \min(\,f(r) + f(r),\ f(r) + f(r),\ f(r) + f(s)\,) =$

"addition" of provenance polynomials

$= \min(\,1.5 + 1.5,\ 1.5 + 1.5,\ 1.5 + \infty\,) = 3.0$

Application with (dis)trust scores (2)

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

The fundamental property

deleted

a	b	c	p
d	b	e	r
f	g	e	s

$\xrightarrow{q}$

a	c	$2p^2$
a	e	pr
d	c	pr
d	e	$2r^2 + rs$
f	e	$2s^2 + rs$

deleted

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

$\downarrow$

$\downarrow$ eval with homomorphisms h , the extension of f

a	b	c	0
d	b	e	1.5

$\xrightarrow{q}$

a	c	0
a	e	1.5
d	c	1.5
~~d~~	~~e~~	~~3.0~~

"Accept tuples with score $\leq$ 2.5"

Application to access control

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

Suppose p is public, r is secret, s is top secret

a	b	c	p
d	b	e	r
f	g	e	s

$\xrightarrow{q}$

a	c	$2p^2$
a	e	pr
d	c	pr
d	e	$2r^2 + rs$
f	e	$2s^2 + rs$

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!)

with p = P, r = S, s = T

a	b	c	P
d	b	e	S
f	g	e	T

$\xrightarrow{q}$

eval with p = P, r = S, s = T
(using min for "+", max for "·")

a	c	P
a	e	S
d	c	S
d	e	S
~~f~~	~~e~~	~~T~~

"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 BoolExp(X).

- So why not Trio(X), $\mathbb{B}[X]$, Why(X), $\mathbb{N}[X]$, PosBool(X)?
  For SPJU (no D) $\mathbb{N}[X]$ always works.
  For the others it depends on K.

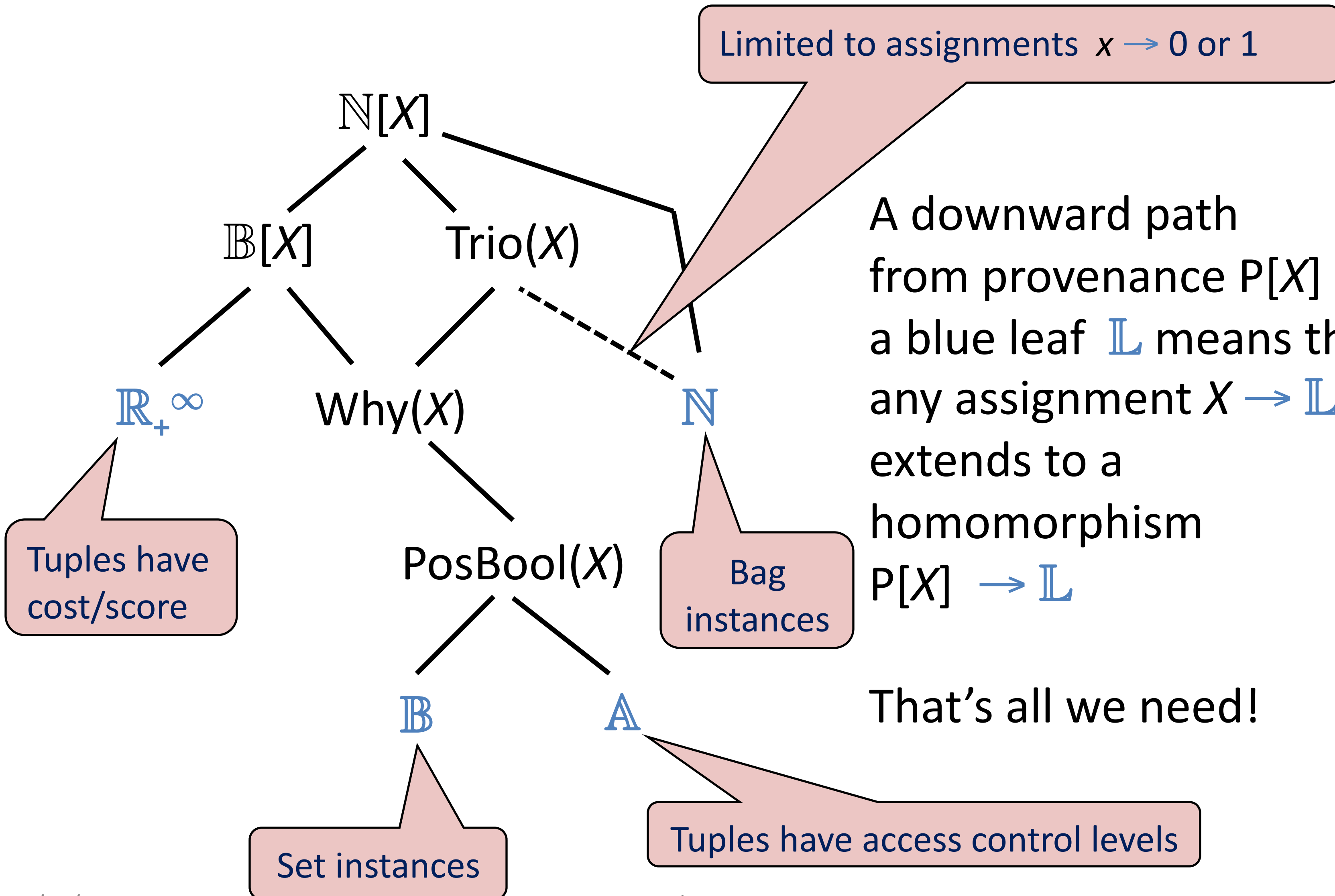
Another application: uncertainty (2)

- $\mathbb{N}[X]$ always works. For incomplete databases:
- Take as representation table T a $\mathbb{N}[X]$-relation
- For **each** assignment function $f : X \to K$
  - extend to a homomorphism $h : \mathbb{N}[X] \to K$
  - 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. (Can be generalized, eg., to Bayesian networks.)
- This gives a probability distribution on assignment functions $f : X \to K$, therefore on the possible worlds.

$\mathbb{N}[X]$ always works, for the others it depends on K



A downward path from provenance P[X] to a blue leaf $\mathbb{L}$ means that any assignment $X \rightarrow \mathbb{L}$ extends to a homomorphism P[X] $\rightarrow \mathbb{L}$

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, G PhDThesis 09]
- **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”.
- This was further developed by Green [G PhDThesis 09].
- The “scalar” k is from K and the “vector” S is a K-relation (or K-set). For $k\ S$ each annotation in S is multiplied by k.
- We have also seen how useful homomorphisms are.
- This suggests an operation $h\ S$ where the homomorphism h is applied to each annotation in S.

Extending query languages to manipulate annotation/ provenance (AnnotatedSQL?)

SELECT RenameH (r.Name AS Name, s.Project AS Project)

FROM r IN db1 Employee, s IN db2 Project

WHERE …

homomorphism

provenance token
(useful in integration)

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.

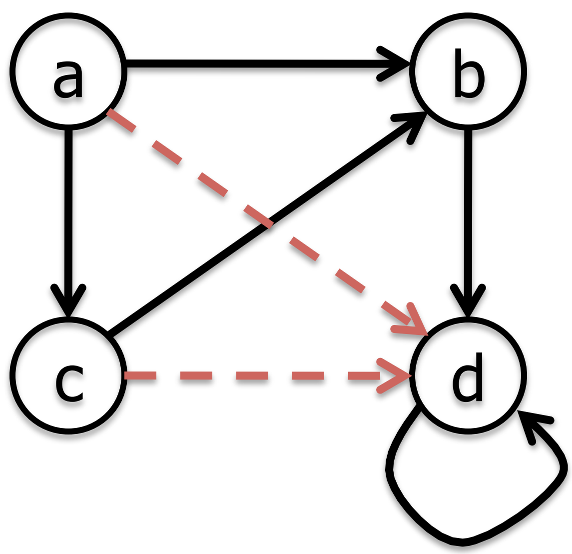
- What's with the semirings? Annotation propagation
- Housekeeping in the zoo of provenance models
- Beyond tuple annotation
- The fundamental property and its applications
- Queries that annotate
- **Datalog** [GK&T PODS 07]

A Datalog program

E

a	b	m
a	c	n
c	b	p
b	d	r
d	d	s

T(X, Y) :- E(X, Y)
T(X, Y) :- T(X, Z), T(Z, Y)



T

a	b	?
a	c	?
c	b	?
b	d	?
d	d	?
a	d	?
c	d	?

K-Datalog?

n-ary K-relations: functions $R : U \to K$ R in K^U

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

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

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

$$T_{\mathbf{P}} : K^U \to K^U$$

For what semirings K does $T_{\mathbf{P}}$ have a fixpoint?

Recall that $T_{\mathbf{P}}$ 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 ω-continuous semirings functions defined by polynomials have least fixpoints (usual definition) hence:

$$\mathrm{fix}(\mathbf{P}) = \mathrm{lub}_{k\geq 0}\, T_{\mathbf{P}}^{k}(0)$$

Most of the semirings that interest us are already ω-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 τ and the annotations of the tuples on their leaves:

$$\mathrm{proof}(\mathbf{P})(t) = \sum_{\tau \text{ yields } t} \Big(\prod_{t' \text{ leaf}(\tau)} R(t') \Big)$$

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

There is also an equivalent "least model" semantics [G09]

Also, $\mathrm{supp}(\mathrm{fix}(\mathbf{P}))$ is finite, and equals the (usual) $\mathbb{B}$-relations semantics (set semantics).

An equivalent perspective

a	b	m
a	c	n
c	b	p
b	d	r
d	d	s

T(X, Y) :- E(X, Y)
T(X, Y) :- T(X, Z), T(Z, Y)

Polynomials are the provenance of the immediate consequence operator

T

a	b	x
a	c	y
c	b	z
b	d	u
d	d	v
a	d	w
c	d	t

$$x = m + yz$$
$$y = n$$
$$z = p$$
$$u = r + uv$$
$$v = s + v^2$$
$$w = xu + wv$$
$$t = zu + tv$$

Solve!

Solving in the power series semiring

$\mathbf{x} = m + np$

$\mathbf{y} = n$

$\mathbf{z} = p$

$\mathbf{v} = s + s^2 + 2s^3 + 5s^4 + 14s^5 + \ldots$

$\mathbf{u} = r\,\mathbf{v}^*$

$\mathbf{w} = r(m+np)(\mathbf{v}^*)^2$

$\mathbf{t} = pr(\mathbf{v}^*)^2$

Coefficients have the form $\frac{2k!}{k!(k+1)!}$

where

$\mathbf{v}^* \triangleq 1 + \mathbf{v} + \mathbf{v}^2 + \mathbf{v}^3 + \ldots$

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 μ, the coefficient of μ in the power series that is the provenance of t is computable (including when it is ∞).

- From CFG ambiguity, we know that testing whether all coefficients are ≤ 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…)

Thank you!