Language Design and Data Provenance

Val Tannen
University of Pennsylvania

Collaborators

T of T award	TJ Green	RelationalAI
	Grigoris Karvounarakis	RelationalAI
G of PODS paper	TJ	
ORCHESTRA	Zack Ives	University of Pennsylvania
	TJ, Grigoris	
Other core papers	Nate Foster	Cornell University
	Yael Amsterdamer	Bar-Ilan University
	Daniel Deutch	Tel Aviv University
	Tova Milo	Tel Aviv University
	Sudeepa Roy	Duke University
	Yuval Moskovitch	Tel Aviv University
Recent work	Erich Grädel	RWTH Aachen
Much gratitude	Peter Buneman	University of Edinburgh

Provenance?

- Provenance is about
  - **trust:** propagate it from inputs to outputs
  - **diagnostics:** faulty outputs come from where?
  - (repairs): fix inputs to fix outputs (reverse provenance analysis).

(Binary) **Trust with Cat Victims**

Sue's notes *

cat	mouse	Yes
cat	rat	Yes

Val's notes *

mouse	gray	No
mouse	red	No
rat	gray	Yes

computation

Zack **

cat	gray	Yes
cat	red	No

prey color

* Sue and Val are noted zoologists.

** Zack is a noted *computational* zoologist

Confidence Scores (non-binary trust)

Sue's notes

cat	mouse	0.9
cat	rat	0.9

Val's notes

mouse	gray	0.6
mouse	red	0.1
rat	gray	0.8

computation

Zack

cat	gray	0.72
cat	red	0.09

$0.72 = \max(0.9 \times 0.8, 0.9 \times 0.6)$

$0.09 = 0.9 \times 0.1$

A Simple Model for Data Pricing

Sue's notes

cat	mouse	$10
cat	rat	$10

Val's notes

mouse	gray	$6
mouse	red	$1
rat	gray	$8

computation

Zack

cat	gray	$16
cat	red	$11

16 = min(10 + 8, 10 + 6)

11 = 10 + 1

Computation? Expressed in a Query Language

Sue's notes

cat	mouse
cat	rat

Val's notes

mouse	gray
mouse	red
rat	gray

computation →

Zack

cat	gray
cat	red

Zack(x,z) :- Sue(x,y) , Val(y,z)

Zack = PROJECT (JOIN (Sue, Val))

Zack = { (u.#pred, v.#color) | u $\in$ Sue , v $\in$ Val , u.#prey=v.#animal }

Do it once and use it repeatedly: provenance

Label (annotate) input items abstractly with **provenance tokens.**

Provenance tracking: propagate **expressions** (involving tokens)
(to annotate intermediate data and, finally, outputs)

Based on **query language design**, track two distinct ways of using data items by computation primitives:

- **jointly** (this alone is basically like keeping a log)
- **alternatively** (doing both is essential; think trust)

Input-output compositional; Modular (in the primitives)

Later, we want to **evaluate** the provenance expressions to obtain
binary trust, confidence scores, data prices, etc.

Algebraic interpretation for RDB

Set X of provenance tokens.

Space of annotations, provenance expressions Prov(X)

Prov(X)-relations:

every tuple is annotated with some element from Prov(X).

Binary operations on Prov(X):

$\cdot$ corresponds to joint use (join, cartesian product),

$+$ corresponds to alternative use (union and projection).

Special annotations:

''Absent'' tuples are annotated with 0.

1 is a ''neutral'' annotation (data we do not track).

K-Relational algebra

Algebraic laws of $(\mathrm{Prov}(X), +, \cdot, 0,1)$? More generally, for annotations from a structure $(K, +, \cdot, 0,1)$?

K-relations. Generalize RA+ to (positive) **K-relational algebra.**

Desired optimization equivalences of K- relational algebra iff

$(K, +, \cdot, 0,1)$ is a **commutative semiring.**

Generalizes SPJU or UCQ or non-rec. Datalog

set semantics $(\mathbb{B}, \vee, \wedge, \bot, \top)$ bag semantics $(\mathbb{N}, +, \cdot, 0, 1)$

c-table-semantics [IL84] $(\mathrm{BoolExp}(X), \vee, \wedge, \bot, \top)$

event table semantics [FR97,Z97] $(\mathcal{P}(\Omega), \cup, \cap, \emptyset, \Omega)$

What is a commutative semiring?

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

$$\left.\begin{array}{l} \bullet\ K \text{ is the domain} \\ \bullet\ + \text{ is associative, commutative, with } 0 \text{ identity} \\ \bullet\ \cdot \text{ is associative, with } 1 \text{ identity} \\ \bullet\ \cdot \text{ distributes over } + \\ \bullet\ a \cdot 0 = 0 \cdot a = 0 \end{array}\right\} \textbf{semiring}$$

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

Unlike ring, no requirement for inverses to $+$

Provenance: abstract semiring annotation

Sue's notes

cat	mouse	p
cat	rat	q

Val's notes

mouse	gray	r
mouse	red	s
rat	gray	t

Zack(x,z):-
Sue(x,y),Val(y,z)

Provenance polynomials
$(\mathbb{N}[X], +, \cdot, 0, 1)$ semiring

Zack

cat	gray	$p \cdot r+q \cdot t$
cat	red	$p \cdot s$

Keep $X=\{\, p,q,r,s,t \,\}$ *abstract.*
Diagnostic for wrong answers;
Deletion propagation.
E.g., $r=s=0$

Provenance propagation through language operations

Sue

cat	mouse	*p*
cat	rat	*q*

Val

mouse	gray	*r*
mouse	red	*s*
rat	gray	*t*

JOIN

cat	mouse	gray	$p \cdot r$
cat	mouse	red	$p \cdot s$
cat	rat	gray	$q \cdot t$

PROJECT

cat	gray	$p \cdot r + q \cdot t$
cat	red	$p \cdot s$

Provenance polynomials

$(\mathbb{N}[X], +, \cdot, 0, 1)$ is the commutative semiring **freely generated** by X
(universality property involving homomorphisms)

Provenance polynomials are **PTIME**-computable (data complexity).
(query complexity depends on language and representation)
ORCHESTRA provenance (graph representation) about **30%** overhead

Monomials correspond to **logical derivations** (proof trees in non-rec. Datalog)

Provenance reading of polynomails:

output tuple has provenance $2r^2 + rs$

three derivations of the tuple
- two of them use r, twice,
- the third uses r and s, once each

Specialize provenance for confidence scores

Sue's notes

cat	mouse	0.9
cat	rat	0.9

Val's notes

mouse	gray	0.6
mouse	red	0.1
rat	gray	0.8

Zack(x,z):-
Sue(x,y),Val(y,z)

Zack

cat	gray	0.72
cat	red	0.09

$\mathbb{V} = ([0,1], \max, \cdot, 0, 1)$ the Viterbi semiring

$f\colon X \to [0,1]$ $f(p)=f(q)=0.9$ $f(r)=0.6$ $f(s)=0.1$ $f(t)=0.8$

$eval(f)\colon \mathbb{N}[X] \to \mathbb{V}$ $eval(f)(pr+qt)=0.72$ $eval(f)(ps)=0.09$

Some application semirings

$(\mathbb{B}, \wedge, \vee, \top, \bot)$ *binary trust*

$(\mathbb{N}, +, \cdot, 0, 1)$ *multiplicity (number of derivations)*

$(\mathbb{A}, \min, \max, 0, \mathrm{Pub})$ *access control*

$\mathbb{V} = ([0,1], \max, \cdot, 0, 1)$ Viterbi semiring (MPE) *confidence scores*

$\mathbb{T} = ([0, \infty], \min, +, \infty, 0)$
tropical semiring (shortest paths) *data pricing*

$\mathbb{F} = ([0,1], \max, \min, 0, 1)$ "fuzzy logic" semiring

Two kinds of semirings in this framework

Provenance semirings, e.g.,

$(\mathbb{N}[X], +, \cdot, 0, 1)$ provenance polynomials [GKT07]

$(\mathrm{Why}(X), \cup, \Cup, \emptyset, \{\emptyset\})$ witness why-provenance [BKT01]

Application semirings, e.g.,

$(\mathbb{A}, \min, \max, 0, \mathrm{Pub})$ access control [FGT08]

$\mathbb{V} = ([0,1], \max, \cdot, 0, 1)$ Viterbi semiring (MPE) [GKIT07]

Provenance specialization relies on

- Provenance semirings are freely generated by provenance tokens
- Query commutation with semiring homomorphisms

Query commutation with homomorphisms

query in $\mathcal{QL}$ homomorphism $h : K_1 \to K_2$

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

$\mathcal{QL}$ = RA+, Datalog [GKT07]

and extensions [FGT08, GP10, ADT11a, T13, DMT15, GUKFC16, T17]

K-Nested Relational Calculus

K-sets. Every element of the set is annotated with some $k \in K$.

where $(K, +, \cdot, 0,1)$ is a **commutative semiring.**

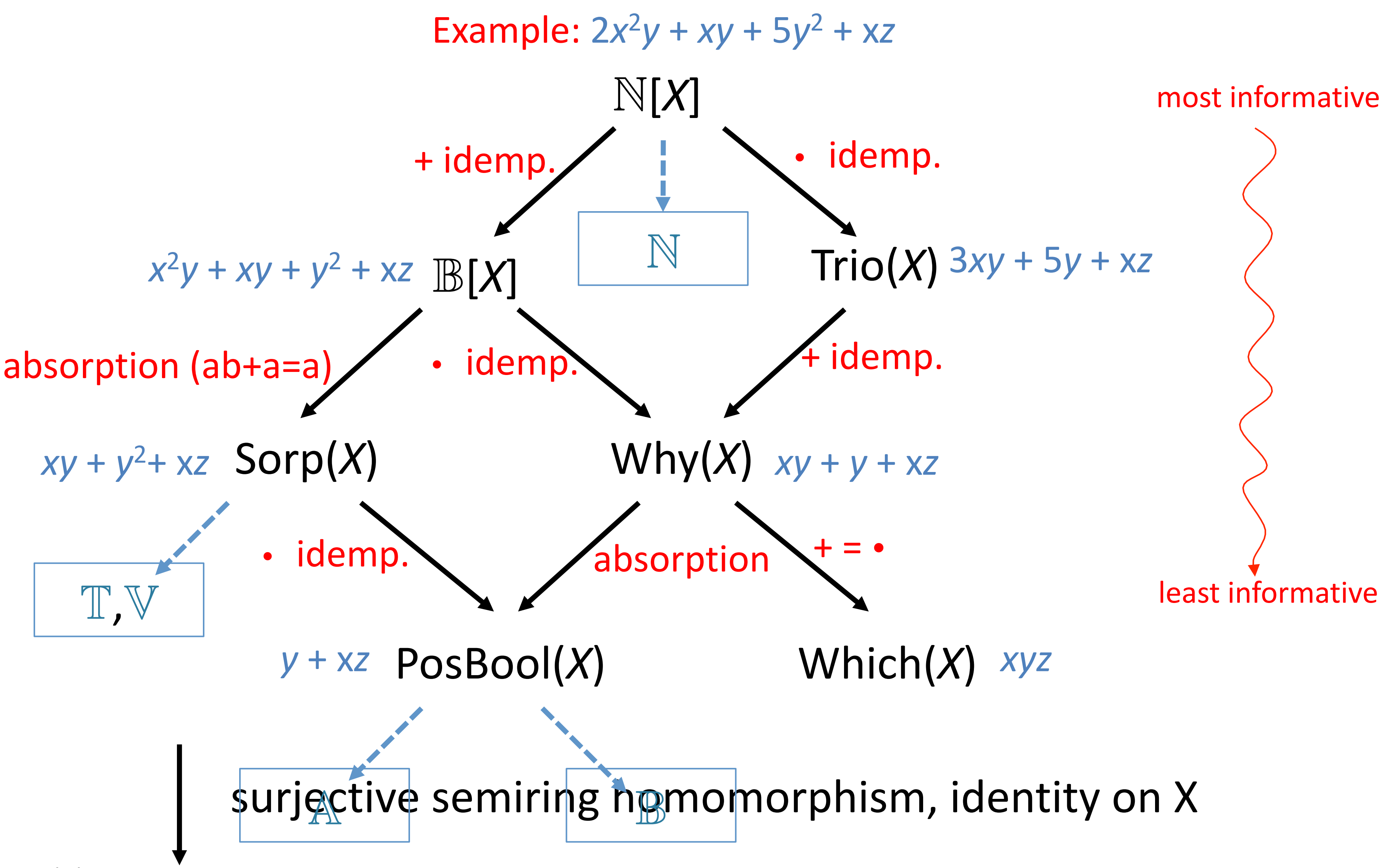
Map f on S $\{ f(x) \mid x \in S \}$

If x is annotated by k then the annotation of $f(x)$ is multiplied by k .

K-sets also form a commutative semiring. This gives annotations for

“FlatMap” g on S $\cup \{ g(x) \mid x \in S \}$

A Hierarchy of Provenance Semirings [G09, DMRT14]



A menagerie of provenance semirings

(Which(X), $\cup$, $\cup^*$, $\emptyset$, $\emptyset^*$) sets of contributing tuples "Lineage" (1) [CWW00]

(Why(X), $\cup$, $\Cup$, $\emptyset$, $\{\emptyset\}$) sets of sets of … Witness why-provenance [BKT01]

(PosBool(X), $\wedge$, $\vee$, $\top$, $\bot$) minimal sets of sets of… Minimal witness why-provenance [BKT01] also "Lineage" (2) used in probabilistic dbs [SORK11]

(Trio(X), $+$, $\cdot$, 0, 1) bags of sets of … "Lineage" (3) [BDHT08,G09]

($\mathbb{B}[X]$, $+$, $\cdot$, 0, 1) sets of bags of … Boolean coeff. polynomials [G09]

(Sorp(X), $+$, $\cdot$, 0, 1) minimal sets of bags of … absorptive polynomials [DMRT14]

($\mathbb{N}[X]$, $+$, $\cdot$, 0, 1) bags of bags of… universal provenance polynomials [GKT07]

Further aspects of the framework

Extension to tree data (Nested Relational Calculus, structural recursion on trees, unordered XQuery) [FGT08]

Study of CQ/UCQ on provenance-annotated relations [G09]

Extension to aggregates (poly-size overhead) [ADT11a]

Poly-size provenance for Datalog (circuits; PosBool(X), Sorp(X)…) [DMRT14]

Extension to data-dependent finite state processes [DMT15]

Connections to semiring monad [FGT08, T13]
to semimodules [ADT11a]
to tensor products [ADT11a, DMT15]

Provenance for aggregation

D	S	
a	20	*x*
a	10	*y*
b	15	*q*
b	10	*r*
b	25	*s*

SUM S

GROUP BY D

D	S-agg	
a	20+10	?
b	15+10+25	?

Desiderata

1. Compatibility with set/bag semantics
2. Fundamental property (commutation with homomorphisms)
3. Poly-size overhead! $1+2+4+\ldots+2^{n-1}$ => 2^n results

Solution inspired by (semi) linear algebra

(K-Rel, $\cup$, $\emptyset$) is a K-semimodule with the singletons as basis.

Relations are the result of $\cup$-aggregation!

What if ($\mathbb{R}$, +, 0) were a Prov(X)-semimodule?

D	S	
a	20	x
a	10	y
b	15	q
b	10	r
b	25	s

D	S-agg	
a	$x\,20 + y\,10$	?
b	$q\,15 + r\,10 + s\,25$	?

($\mathbb{R}$, +, 0) is not a Prov(X)-semimodule, but…

Tensor product construction

Embed a commutative monoid M (for sum, max or min) into a K-semimodule $K \otimes M$ (new values!)

Consistency: embedding should be faithful.

D	S-agg	
a	$x \otimes 20 + y \otimes 10$	$x + y$
b	$q \otimes 15 + r \otimes 10 + s \otimes 25$	$q + r + s$

Negative information; non-monotone operations (difference)

Boolean expressions [IL84]. Limited.

Add a binary operation corresponding to difference

m-semirings (common gen. of set and bag difference) [GP10]

spm-semirings (OPTIONAL in SPARQL) [GUKFC16]

Encode difference by aggregation [ADT11a]

Different equational theories, different algebraic optimizations [ADT11b]

Still not clear how to track **negative information**.

useful: non-answers (why not?), insertion propagation.

Logical model checking ("*provenance of … truth?*")

negation as duality (NNFs), logical games

ongoing work with Grädel [T16, T17]

Current targets

ANALYTICS COMPUTATIONS

"Fine-grained provenance for linear algebra operators"
Yan, T., **Ives** TaPP 16

DISTRIBUTED SYSTEMS/NETWORK PROVENANCE

"Time-aware provenance for distributed systems",
Zhou, Ding, **Haeberlen, Ives, Loo** TaPP 11

"Diagnosing missing events in distributed systems with negative provenance",
Wu, Zhao, **Haeberlen,** Zhou, **Loo** SIGCOMM 14

STATIC ANALYSIS OF SOFTWARE

"On abstraction refinement for program analyses in Datalog"
Zhang, Mangal, Grigore, **Naik** PLDI 14

Framework references (I)

[GKT07]
"Provenance semirings" Green, Karvounarakis, Tannen PODS 07.

[GKIT07]
"Update exchange with mappings and provenance" Green, Karvounarakis, Ives, Tannen VLDB 07.

[FGT08]
"Annotated XML: queries and provenance" Foster, Green, Tannen PODS 08.

[G09]
"Containment of conjunctive queries on annotated relations" Green ICDT 09.

[GP10]
"On database query languages for K-relations", Geerts, Poggi J Appl. Logic 2010.

Framework references (II)

[ADT11a]
"Provenance for aggregate queries", Amsterdamer, Deutch, Tannen PODS 11.

[ADT11b]
"On the limitations of provenance for queries with difference",
Amsterdamer, Deutch, Tannen TaPP 11

[T13]
"Provenance propagation in complex queries"
Tannen Buneman Festschrift 2013

[DMRT14]
"Circuits for Datalog provenance", Deutch, Milo, Roy, T. ICDT 14.

[DMT15]
"Provenance-based analysis of data-centric processes"
Deutch, Moskovitch, Tannen VLDB J. 2015

Framework references (III)

[GUKFC16]
"Algebraic structures for capturing the provenance of SPARQL queries"
Geerts, Unger, Karvounarakis, Fundulaki, Christophides JACM 2016

[T16]
"About the provenance of truth" Tannen Simons Inst. Website 16
https://simons.berkeley.edu/talks/val-tannen-2016-12-09

[T17]
"Provenance analysis for FOL model checking" Tannen SIGLOG News 2017

[GT17a]
"The semiring framework for database provenance", Green, Tannen PODS 2017.

[GT17b]
"Semiring provenance for first-order model checking", Grädel, Tannen
CoRR abs/1712.01980 (2017)

Other references

[IL84]
"Incomplete information in relational databases" Imieliński, Lipski JACM 1984

[FR97]
"A probabilistic relational algebra" Fuhr, Röllecke TOIS 1997

[Z97]
"Query evaluation in probabilistic relational databases" Zimányi DDS 1997

[CWW00]
"Tracing the lineage of view data in a warehousing environment" Cui, Widom, Wiener TODS 2000

[BKT01]
"Why and where: a characterization of data provenance" Buneman, Khanna, Tan ICDT 2001

[BDHTW08]
"Databases with uncertainty and lineage" Benjelloun, Das Sarma, Halevy, Theobald, Widom VLDB J. 2008

[SORK11]
"Probabilistic databases" Suciu, Olteanu, Ré, Koch SLDM 2011

Thank you!