Updated 3/15/2023

Topic 2: Complexity of Query Evaluation
Unit 3: Provenance
Lecture 17

Wolfgang Gatterbauer

CS7240 Principles of scalable data management (sp23)

https://northeastern-datalab.github.io/cs7240/sp23/

3/14/2023

Pre-class conversations

- Last class summary
- Project ideas and feedback
- Faculty candidate

- Today:
  - provenance, semirings
- Next class:
  - semirings, more abstract

Outline: T2-3/4: Provenance & Reverse Data Management

- T2-3: Provenance
  - Data Provenance
  - The Semiring Framework for Provenance
  - Algebra: Monoids and Semirings
  - Query-rewrite-insensitive provenance
- T2-4: Reverse Data Management
  - View Deletion Problem
  - Resilience & Causality

Mainly slides by
Val Tannen 2017

Positive relational algebra:

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)

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, access control,
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).

Source: Val Tannen. "The Semiring Framework for Database Provenance", PODS 2017 Test of Time Award talk : https://www.cis.upenn.edu/~val/15MayPODS.pdf
Wolfgang Gatterbauer. Principles of scalable data management: https://northeastern-datalab.github.io/cs7240/

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] $(\mathcal{P}(\Omega), \cup, \cap, \emptyset, \Omega)$

5/15/2017 PODS 2017 10

$\{x, y\}$ $\{x+y, x \cdot y, x \cdot y \cdot y + x, \ldots\}$

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 $+$

Source: Val Tannen. "The Semiring Framework for Database Provenance", PODS 2017 Test of Time Award talk : https://www.cis.upenn.edu/~val/15MayPODS.pdf
Wolfgang Gatterbauer. Principles of scalable data management: https://northeastern-datalab.github.io/cs7240/

Provenance polynomials

$\mathbb{N}[\{x, y\}] = \{xy, x + y, 2xy^2 + x, 2xy^2 + xy + x, \dots\}$

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

Source: Val Tannen. "The Semiring Framework for Database Provenance", PODS 2017 Test of Time Award talk : https://www.cis.upenn.edu/~val/15MayPODS.pdf
Wolfgang Gatterbauer. Principles of scalable data management: https://northeastern-datalab.github.io/cs7240/

Two kinds of semirings in this framework

Provenance semirings, e.g.,

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

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

Application semirings, e.g.,

$(\mathbb{A}, \min, \max, 0, \text{Pub})$ 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

Source: Val Tannen. "The Semiring Framework for Database Provenance", PODS 2017 Test of Time Award talk : https://www.cis.upenn.edu/~val/15MayPODS.pdf

Some application semirings

Example 1: $(\mathbb{B}, \wedge, \vee, \top, \bot)$ *binary trust*

Example 5: $(\mathbb{N}, +, \cdot, 0, 1)$ *multiplicity (number of derivations)*

Example 2: $(\mathbb{A}, \min, \max, 0, \text{Pub})$ *access control*

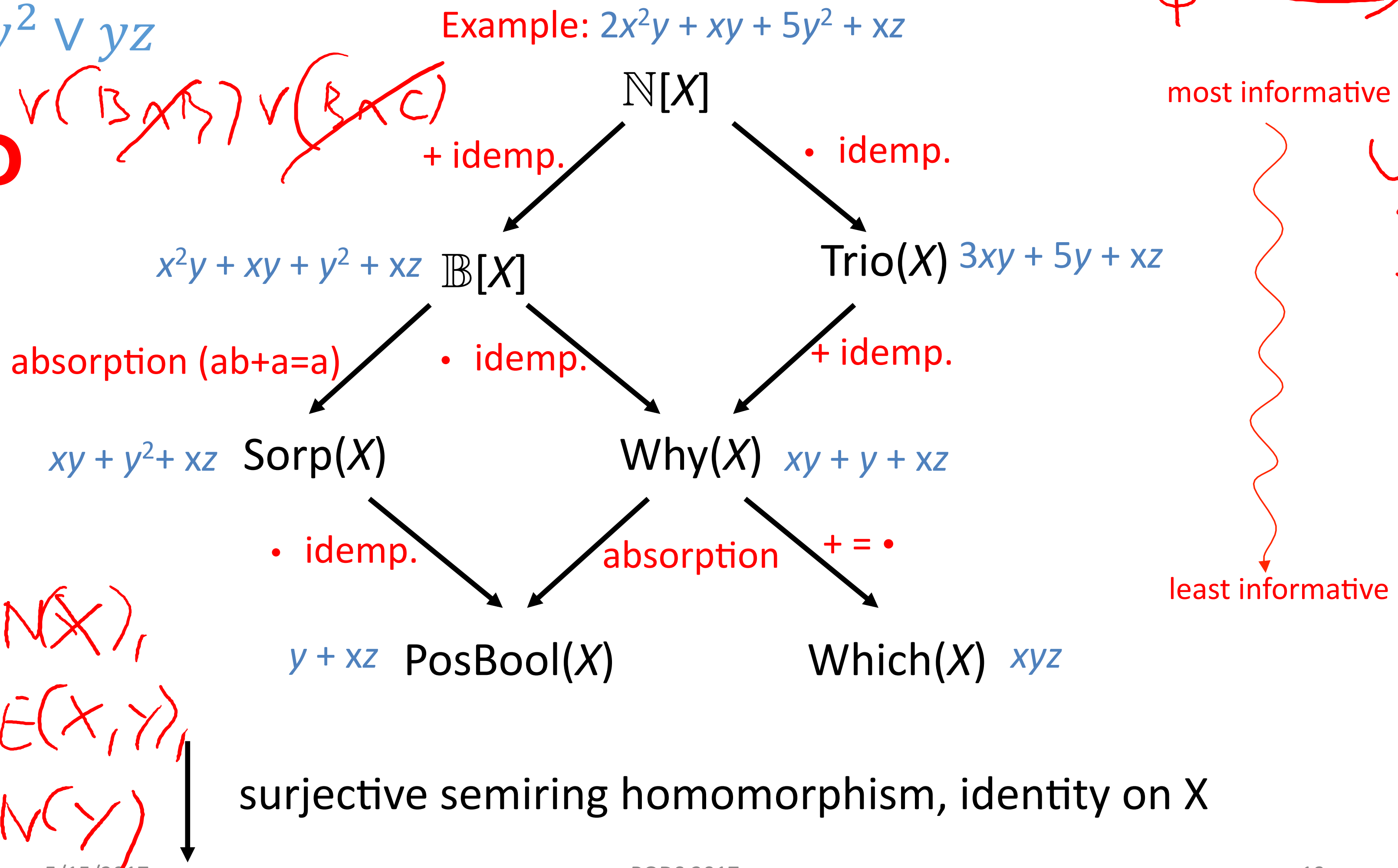
Example 4: $\mathbb{V} = ([0,1], \max, \cdot, 0, 1)$ Viterbi semiring (MPE) *confidence scores*

Example 3: $\mathbb{T} = ([0, \infty], \min, +, \infty, 0)$
tropical semiring (shortest paths) *data pricing*

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

Source: Val Tannen. "The Semiring Framework for Database Provenance", PODS 2017 Test of Time Award talk : https://www.cis.upenn.edu/~val/15MayPODS.pdf
Wolfgang Gatterbauer. Principles of scalable data management: https://northeastern-datalab.github.io/cs7240/

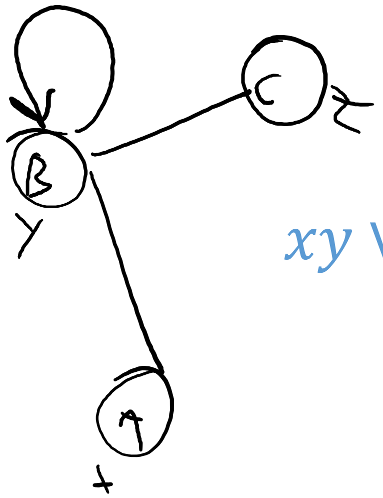
A Hierarchy of Provenance Semirings [G09, DMRT14]

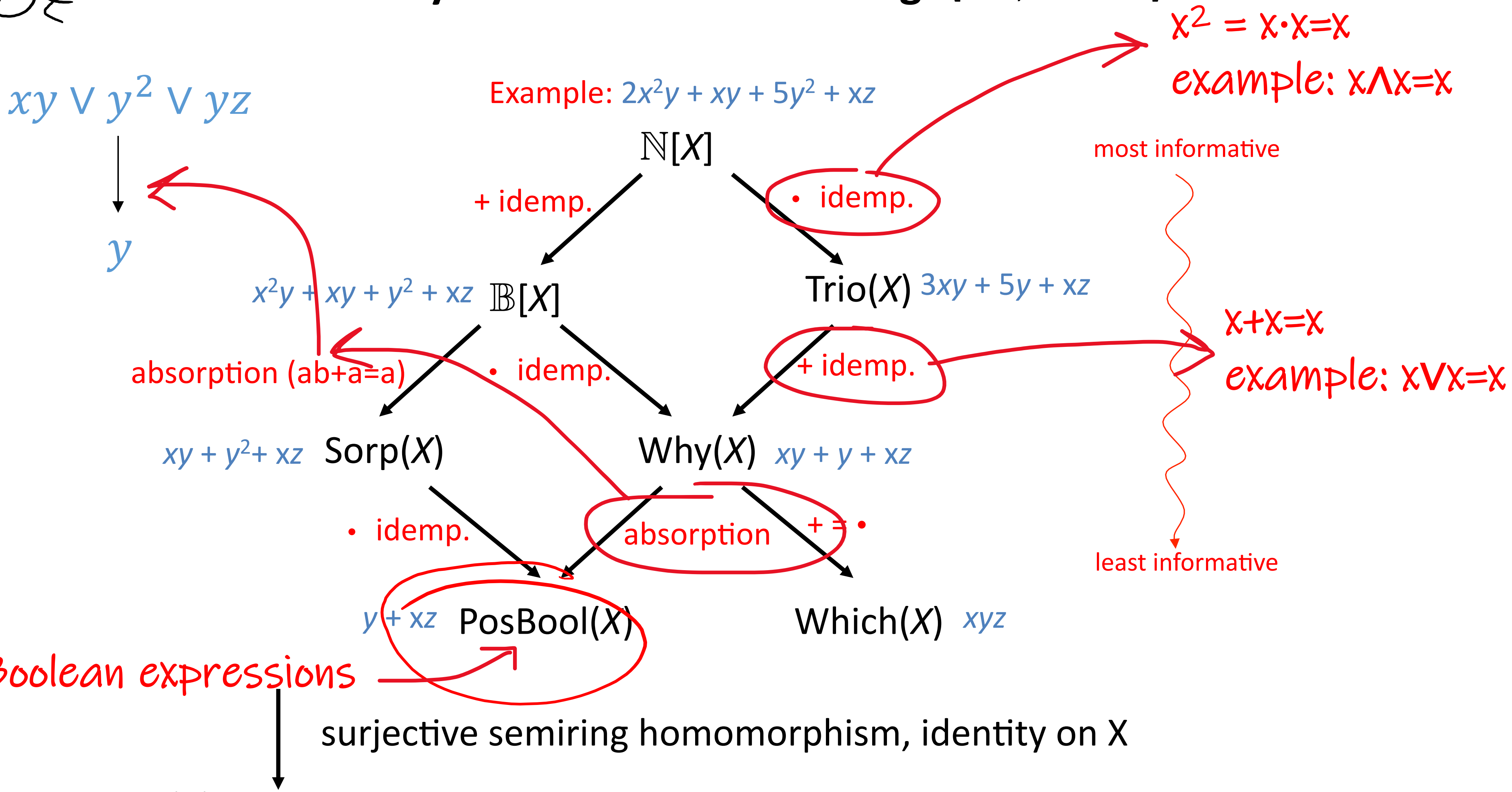


Source: Val Tannen. "The Semiring Framework for Database Provenance", PODS 2017 Test of Time Award talk : https://www.cis.upenn.edu/~val/15MayPODS.pdf

Wolfgang Gatterbauer. Principles of scalable data management: https://northeastern-datalab.github.io/cs7240/

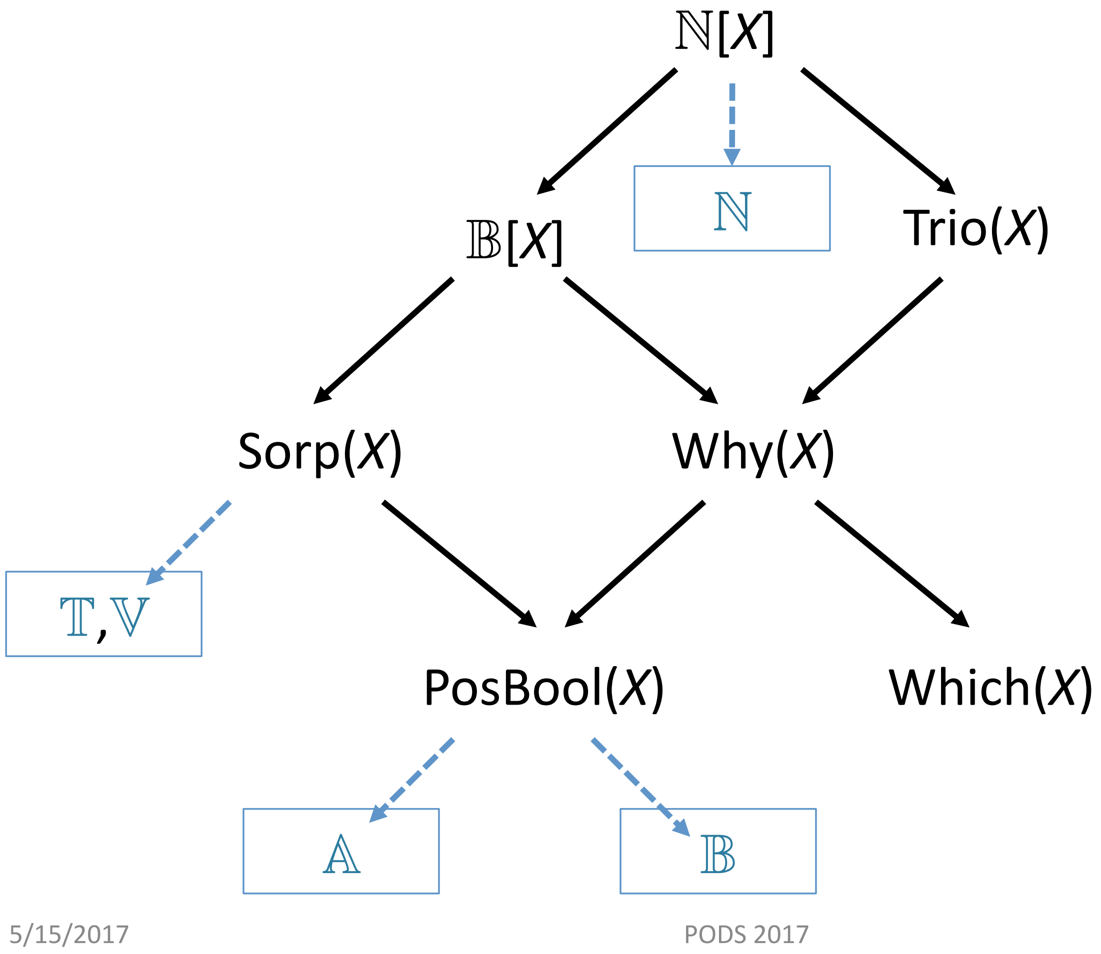
A Hierarchy of Provenance Semirings [G09, DMRT14]





5/15/2017 PODS 2017 19

A Hierarchy of Provenance Semirings [G09, DMRT14]



Source: Val Tannen. "The Semiring Framework for Database Provenance", PODS 2017 Test of Time Award talk : https://www.cis.upenn.edu/~val/15MayPODS.pdf

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]

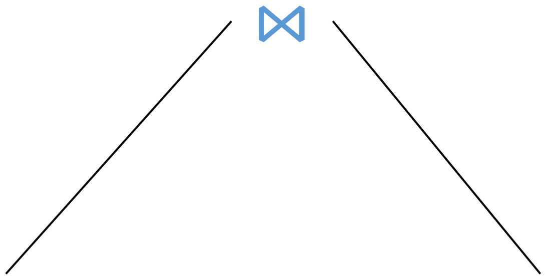
($\mathbb{B}[X]$, $+$, $\cdot$, 0, 1) sets of bags of … Boolean coeff. polynomials [G09]

(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]

Source: Val Tannen. "The Semiring Framework for Database Provenance", PODS 2017 Test of Time Award talk : https://www.cis.upenn.edu/~val/15MayPODS.pdf

Positive relational algebra: Join ⋈



R

A	B
1	1
2	1
2	2

S

B	C
1	1
2	2
2	3

Q=R⋈S

A	B	C

?

Positive relational algebra: Join ⋈

⋈

R

A	B
1	1
2	1
2	2

S

B	C
1	1
2	2
2	3

Q=R⋈S

A	B	C
1	1	1
2	1	1
2	2	2
2	2	3

Positive relational algebra: Join ⋈

⋈

R

A	B	
1	1	r_1
2	1	r_2
2	2	r_3

S

B	C	
1	1	s_1
2	2	s_2
2	3	s_3

Q=R⋈S

A	B	C
1	1	1
2	1	1
2	2	2
2	2	3

?

Positive relational algebra: Join ⋈

The annotation "r · s" means <u>joint use</u> of data annotated by r and data annotated by s

⋈

R

A	B	
1	1	r_1
2	1	r_2
2	2	r_3

S

B	C	
1	1	s_1
2	2	s_2
2	3	s_3

$Q = R \bowtie S$

A	B	C	
1	1	1	$r_1 \cdot s_1$
2	1	1	$r_2 \cdot s_1$
2	2	2	$r_3 \cdot s_2$
2	2	3	$r_3 \cdot s_3$

Positive relational algebra: Projection π

$\pi_{\text{-B}}$

R

A	B	
1	1	r_1
1	2	r_2
2	1	r_3
2	2	r_4
2	3	r_5

$Q=\pi_{\text{-B}}R=\pi_{A}R$

A
?

Positive relational algebra: Projection π

π_{-B}

R

A	B	
1	1	r_1
1	2	r_2
2	1	r_3
2	2	r_4
2	3	r_5

$Q=\pi_{-B}R=\pi_{A}R$

A
1
2

?

Positive relational algebra: Projection π

The annotation "r + s" means <u>alternative use</u> of data

π_{-B}

R

A	B	
1	1	r_1
1	2	r_2
2	1	r_3
2	2	r_4
2	3	r_5

$Q=\pi_{-B}R=\pi_{A}R$

A	
1	$r_1 + r_2$
2	$r_3 + r_4 + r_5$

$r_3 \vee r_4 \vee r_5$

Positive relational algebra: Union ∪

∪

R

A	B	
1	1	r_1
2	1	r_2

S

A	B	
2	1	s_1
2	2	s_2

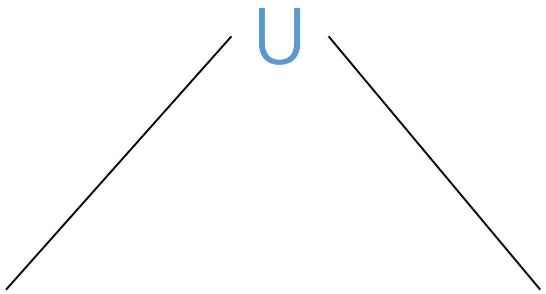
Q=R∪S

A	B

?

Positive relational algebra: Union ∪

The annotation "r + s" means <u>alternative use</u> of data



$\{(2,1),(2,1)\} = (2,1) \mapsto 2$

R

A	B	
1	1	r_1
2	1	r_2

S

A	B	
2	1	s_1
2	2	s_2

Q=R∪S

A	B	
1	1	r_1
2	1	$r_2 + s_1$
2	2	s_2

$R \cup S = \pi_{AB}(R \cup_{BAG} S)$

Positive relational algebra: Selection σ

$\sigma_{A=1}$

R

A	B	
1	1	r_1
1	2	r_2
2	1	r_3
2	2	r_4
2	3	r_5

$Q=\sigma_{A=1}R$

A	B

?

Positive relational algebra: Selection σ

Two options for filtering:

1. Remove the tuples filtered out.

$\sigma_{A=1}$

R

A	B	
1	1	r_1
1	2	r_2
2	1	r_3
2	2	r_4
2	3	r_5

$Q = \sigma_{A=1} R$

A	B	
1	1	r_1
1	2	r_2

Positive relational algebra: Selection σ

$\sigma_{A=1}$

R

A	B	
1	1	r_1
1	2	r_2
2	1	r_3
2	2	r_4
2	3	r_5

Two options for filtering:

1. Remove the tuples filtered out.
2. Or keep them around …

$Q=\sigma_{A=1}R$

A	B	
1	1	$r_1 \cdot 1$
1	2	$r_2 \cdot 1$
2	1	$r_3 \cdot 0$
2	2	$r_4 \cdot 0$
2	3	$r_5 \cdot 0$

Boolean Query Provenance

Q :- R(x,y), S(y,z)

Calculate the provenance, operator-by-operator, with two algebraically equivalent query plans:

R

A	B	
1	1	r_1
2	2	r_2
3	2	r_3

S

B	C	
1	1	s_1
1	2	s_2
2	3	s_3

Query plan 1: $\pi_{\text{-A,B,C}}(R \bowtie S)$

?

Query plan 2: $\pi_{\text{-B}}(\pi_{\text{-A}}(R) \bowtie \pi_{\text{-C}}(S))$

?

Boolean Query Provenance

Q :- $R(x,y), S(y,z)$

R

A	B	
1	1	r_1
2	2	r_2
3	2	r_3

S

B	C	
1	1	s_1
1	2	s_2
2	3	s_3

Calculate the provenance, operator-by-operator, with two algebraically equivalent query plans:

Query plan 1: $\pi_{\text{-A,B,C}}(R \bowtie S)$

$R \bowtie S$

?

$\pi_{\text{-A,B,C}}(\ldots)$

?

Query plan 2: $\pi_{\text{-B}}(\pi_{\text{-A}}(R) \bowtie \pi_{\text{-C}}(S))$

?

Boolean Query Provenance


Q :- R(x,y), S(y,z)

Calculate the provenance, operator-by-operator, with two algebraically equivalent query plans:

R

A	B	
1	1	r_1
2	2	r_2
3	2	r_3

S

B	C	
1	1	s_1
1	2	s_2
2	3	s_3

Query plan 1: $\pi_{\text{-A,B,C}}(R \bowtie S)$

$R \bowtie S$

A	B	C	
1	1	1	$r_1 \cdot s_1$
1	1	2	$r_1 \cdot s_2$
2	2	3	$r_2 \cdot s_3$
3	2	3	$r_3 \cdot s_3$

$\pi_{\text{-A,B,C}}(\ldots)$

?

Query plan 2: $\pi_{\text{-B}}(\pi_{\text{-A}}(R) \bowtie \pi_{\text{-C}}(S))$

?

Boolean Query Provenance


Q :- R(x,y), S(y,z)

Calculate the provenance, operator-by-operator, with two algebraically equivalent query plans:

R

A	B	
1	1	r_1
2	2	r_2
3	2	r_3

S

B	C	
1	1	s_1
1	2	s_2
2	3	s_3

Query plan 1: $\pi_{\text{-A,B,C}}(R \bowtie S)$

$R \bowtie S$

A	B	C	
1	1	1	$r_1 \cdot s_1$
1	1	2	$r_1 \cdot s_2$
2	2	3	$r_2 \cdot s_3$
3	2	3	$r_3 \cdot s_3$

$\pi_{\text{-A,B,C}}(\ldots)$

$$r_1 \cdot s_1 + r_1 \cdot s_2 + r_2 \cdot s_3 + r_3 \cdot s_3$$

Query plan 2: $\pi_{\text{-B}}(\pi_{\text{-A}}(R) \bowtie \pi_{\text{-C}}(S))$

?

Boolean Query Provenance

Q :- R(x,y), S(y,z)

Calculate the provenance, operator-by-operator, with two algebraically equivalent query plans:

R

A	B	
1	1	r_1
2	2	r_2
3	2	r_3

S

B	C	
1	1	s_1
1	2	s_2
2	3	s_3

Query plan 1: $\pi_{\text{-}A,B,C}(R \bowtie S)$

$R \bowtie S$

A	B	C	
1	1	1	$r_1 \cdot s_1$
1	1	2	$r_1 \cdot s_2$
2	2	3	$r_2 \cdot s_3$
3	2	3	$r_3 \cdot s_3$

$\pi_{\text{-}A,B,C}(\ldots)$

$r_1 \cdot s_1 + r_1 \cdot s_2 + r_2 \cdot s_3 + r_3 \cdot s_3$

Query plan 2: $\pi_{\text{-}B}(\pi_{\text{-}A}(R) \bowtie \pi_{\text{-}C}(S))$

$\pi_{\text{-}A}(R)$

?

$\pi_{\text{-}C}(S)$

?

$\pi_{\text{-}B}(R' \bowtie S')$

?

Boolean Query Provenance

Q :- R(x,y), S(y,z)

Calculate the provenance, operator-by-operator, with two algebraically equivalent query plans:

R

A	B	
1	1	r_1
2	2	r_2
3	2	r_3

S

B	C	
1	1	s_1
1	2	s_2
2	3	s_3

Query plan 1: $\pi_{\text{-}A,B,C}(R \bowtie S)$

$R \bowtie S$

A	B	C	
1	1	1	$r_1 \cdot s_1$
1	1	2	$r_1 \cdot s_2$
2	2	3	$r_2 \cdot s_3$
3	2	3	$r_3 \cdot s_3$

$\pi_{\text{-}A,B,C}(\ldots)$

$r_1 \cdot s_1 + r_1 \cdot s_2 + r_2 \cdot s_3 + r_3 \cdot s_3$

Query plan 2: $\pi_{\text{-}B}(\pi_{\text{-}A}(R) \bowtie \pi_{\text{-}C}(S))$

$\pi_{\text{-}A}(R)$

B	
1	r_1
2	$r_2 + r_3$

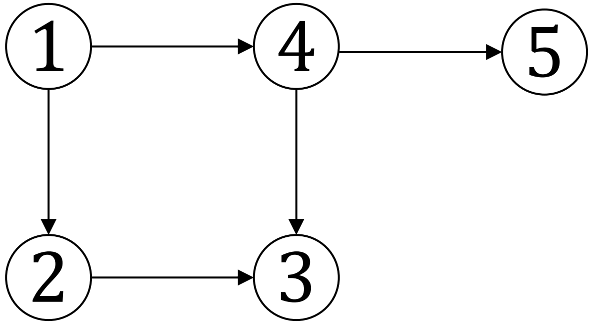
$\pi_{\text{-}C}(S)$

B	
1	$s_1 + s_2$
2	s_3

$\pi_{\text{-}B}(R' \bowtie S')$

$r_1 \cdot (s_1 + s_2) + (r_2 + r_3) \cdot s_3$

Back to our Example: now with Semiring notation



Now assume we use semiring notation.
Idea: keep the tuple identifiers abstract.
Use provenance polynomials ($\mathbb{N}[X]$, +, ·, 0, 1)

E

1	2
2	3
1	4
4	3
4	5

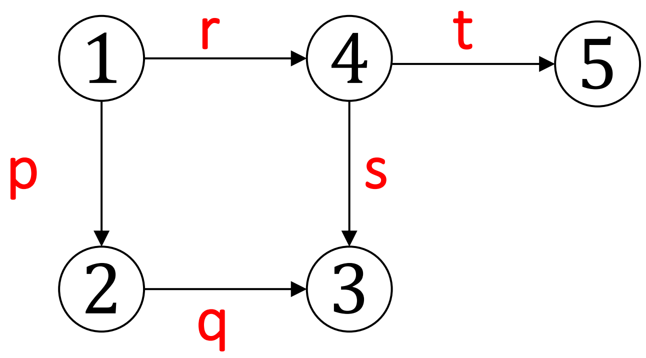
Q(z) :- E(1,y), E(y,z) →

Q

3
5

Q: Points reachable in 2 hops, starting at node "1"

Back to our Example: now with Semiring notation



Now assume we use semiring notation.
Idea: keep the tuple identifiers abstract.
Use provenance polynomials (ℕ[X], +, ·, 0, 1)

⊕ ⊗

E

1	2	p
2	3	q
1	4	r
4	3	s
4	5	t

X = {p, q, …, t}

Q(z) :- E(1,y), E(y,z)

Q: Points reachable in 2 hops, starting at node "1"

Q

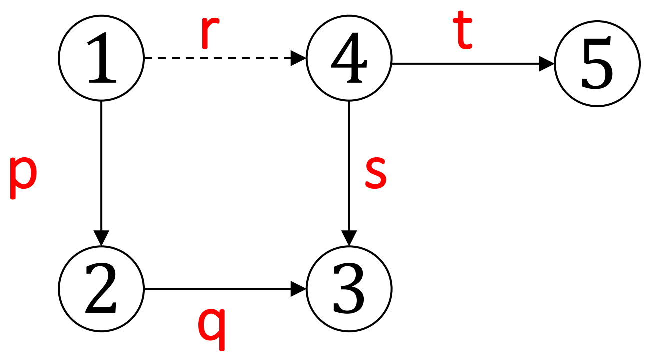
3	r·s+p·q
5	r·t

∈ ℕ(X)

ℕ[X]=(ℕ[X], +, ·, 0, 1): Provenance polynomials

Example variant 1

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



Now assume only certain edges are available (available yes/no or true/false). Which of the points remain reachable?

E

1	2	p = 1
2	3	q = 1
~~1~~	~~4~~	~~r = 0~~
4	3	s = 1
4	5	t = 1

Q(z) :- E(1,y), E(y,z)

Q

3	$r \cdot s + p \cdot q = 1$
5	$r \cdot t = 0$

$(0 \wedge 1) \vee (1 \wedge 1) = 1$

$(0 \wedge 1) = 0$

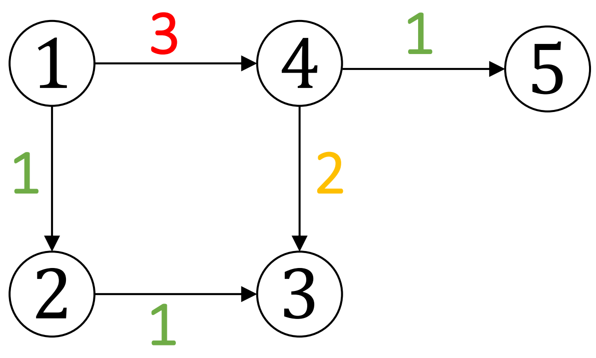
Q: Points reachable in 2 hops, starting at node "1"

{0,1}

$\mathbb{B}$=($\mathbb{B}$, ∨, ∧ , 0, 1): Boolean algebra

Example variant 2

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



Now assume passing along an edge needs a certain security clearance (1<2<3). What clearance do you need for reaching each point?

infix ⇝ prefix

min[max[3,2], max[1,1]] = 1

E

1	2
2	3
1	4
4	3
4	5

$p = 1$
$q = 1$
$r = 3$
$s = 2$
$t = 1$

Q(z) :- E(1,y), E(y,z)

Q: Points reachable in 2 hops, starting at node "1"

Q

3
5

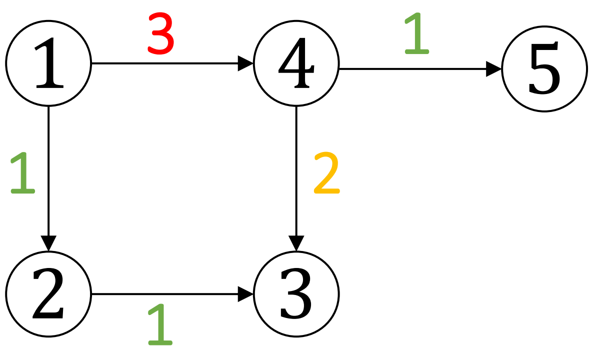
$r \cdot s + p \cdot q = 1$
$r \cdot t = 3$

max[3,1] = 3

({1,2,3,∞}, min, max, ∞,1)

Example variant 3

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



Now assume each edge has a weight.

What is the shortest path to reach each point?

E

1	2	$p = 1$
2	3	$q = 1$
1	4	$r = 3$
4	3	$s = 2$
4	5	$t = 1$

Q(z) :- E(1,y), E(y,z)

Q: Points reachable in 2 hops, starting at node "1"

Q

3	$r \cdot s + p \cdot q = 2$
5	$r \cdot t = 4$

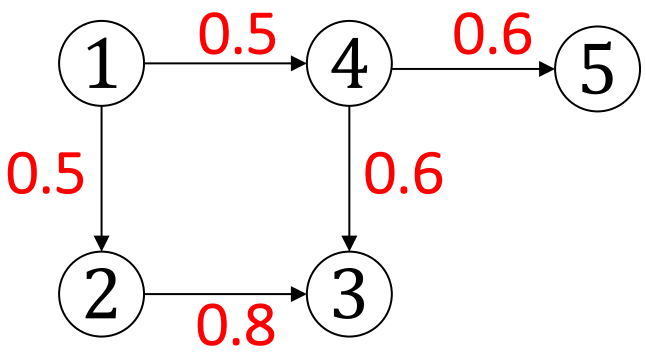
min[3+2,1+1] = 2

3+1 = 4

$\mathbb{T}=(\mathbb{R}_+^\infty, \min, +, \infty, 0)$: Tropical semiring

Example variant 4

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



Now assume each edge has a confidence (probability of being available).
What is the probability of the most likely path?

E

1	2	$p = 0.5$
2	3	$q = 0.8$
1	4	$r = 0.5$
4	3	$s = 0.6$
4	5	$t = 0.6$

Q(z) :- E(1,y), E(y,z)

Q: Points reachable in 2 hops, starting at node "1"

Q

3	$r \cdot s + p \cdot q$	$= 0.4$
5	$r \cdot t$	$= 0.3$

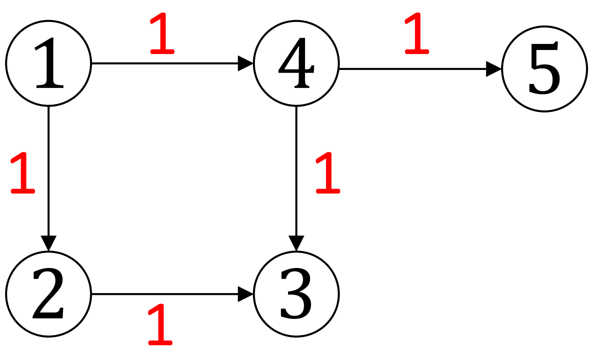
$\max[0.5 \cdot 0.6, 0.5 \cdot 0.8] = 0.4$

$0.5 \cdot 0.6 = 0.3$

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

Example variant 5

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



Finally assume we want to calculate the number of paths to a node. We start by annotating the tuples in the database with their duplicity (which is 1 to start with)

E

1	2	$p = 1$
2	3	$q = 1$
1	4	$r = 1$
4	3	$s = 1$
4	5	$t = 1$

Q(z) :- E(1,y), E(y,z)

Q: Points reachable in 2 hops, starting at node "1"

Q

3	$r \cdot s + p \cdot q = 2$	$1 \cdot 1 + 1 \cdot 1 = 2$
5	$r \cdot t = 1$	$1 \cdot 1 = 1$

$(\mathbb{N}, +, \cdot, 0, 1)$: Counting derivations / bag semantics

A more complex example with exponents


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

R

A	B	C
a	b	c
d	b	e
f	g	e

$\pi_{AB}R \bowtie \pi_{BC}R$

A	B	C

?

$\pi_{AC}R \bowtie \pi_{BC}R$

A	B	C

?

$Q_1 \cup Q_2$

A	B	C

?

Q

A	C

?

Example from Section 2 of Green, Karvounarakis, Val Tannen. "Provenance Semirings", PODS 2007. https://doi.org/10.1145/1265530.1265535

A more complex example with exponents

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

R

A	B	C	
a	b	c	X
d	b	e	Y
f	g	e	Z

$\pi_{AB}R \bowtie \pi_{BC}R$

A	B	C
a	b	c
a	b	e
d	b	c
d	b	e
f	g	e

?

$\pi_{AC}R \bowtie \pi_{BC}R$

A	B	C
a	b	c
d	b	e
d	g	e
f	b	e
f	g	e

?

$Q_1 \cup Q_2$

A	B	C
a	b	c
a	b	e
d	b	c
d	b	e
d	g	e
f	b	e
f	g	e

?

Q

A	C
a	c
a	e
d	c
d	e
f	e

?

Example from Section 2 of Green, Karvounarakis, Val Tannen. "Provenance Semirings", PODS 2007. https://doi.org/10.1145/1265530.1265535

A more complex example with exponents

$$Q(R) = \pi_{AC}(\pi_{AB}R \bowtie \pi_{BC}R \;\cup\; \pi_{AC}R \bowtie \pi_{BC}R)$$

R

A	B	C	
a	b	c	X
d	b	e	Y
f	g	e	Z

$\pi_{AB}R \bowtie \pi_{BC}R$

A	B	C	
a	b	c	X^2
a	b	e	XY
d	b	c	XY
d	b	e	Y^2
f	g	e	Z^2

$\pi_{AC}R \bowtie \pi_{BC}R$

A	B	C
a	b	c
d	b	e
d	g	e
f	b	e
f	g	e

?

$Q_1 \cup Q_2$

A	B	C
a	b	c
a	b	e
d	b	c
d	b	e
d	g	e
f	b	e
f	g	e

?

Q

A	C
a	c
a	e
d	c
d	e
f	e

?

Example from Section 2 of Green, Karvounarakis, Val Tannen. "Provenance Semirings", PODS 2007. https://doi.org/10.1145/1265530.1265535

A more complex example with exponents

$$Q(R) = \pi_{AC}(\underbrace{\pi_{AB}R \bowtie \pi_{BC}R}\ \cup\ \underbrace{\pi_{AC}R \bowtie \pi_{BC}R})$$

R

A	B	C	
a	b	c	X
d	b	e	Y
f	g	e	Z

$\pi_{AB}R \bowtie \pi_{BC}R$

A	B	C	
a	b	c	X^2
a	b	e	XY
d	b	c	XY
d	b	e	Y^2
f	g	e	Z^2

$\pi_{AC}R \bowtie \pi_{BC}R$

A	B	C	
a	b	c	X^2
d	b	e	Y^2
d	g	e	YZ
f	b	e	YZ
f	g	e	Z^2

$Q_1 \cup Q_2$

A	B	C
a	b	c
a	b	e
d	b	c
d	b	e
d	g	e
f	b	e
f	g	e

?

Q

A	C
a	c
a	e
d	c
d	e
f	e

?

Example from Section 2 of Green, Karvounarakis, Val Tannen. "Provenance Semirings", PODS 2007. https://doi.org/10.1145/1265530.1265535

A more complex example with exponents

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

R

A	B	C	
a	b	c	X
d	b	e	Y
f	g	e	Z

$\pi_{AB}R \bowtie \pi_{BC}R$

A	B	C	
a	b	c	X^2
a	b	e	XY
d	b	c	XY
d	b	e	Y^2
f	g	e	Z^2

$\pi_{AC}R \bowtie \pi_{BC}R$

A	B	C	
a	b	c	X^2
d	b	e	Y^2
d	g	e	YZ
f	b	e	YZ
f	g	e	Z^2

$Q_1 \cup Q_2$

A	B	C	
a	b	c	$2X^2$
a	b	e	XY
d	b	c	XY
d	b	e	$2Y^2$
d	g	e	YZ
f	b	e	YZ
f	g	e	$2Z^2$

Q

A	C
a	c
a	e
d	c
d	e
f	e

?

Example from Section 2 of Green, Karvounarakis, Val Tannen. "Provenance Semirings", PODS 2007. https://doi.org/10.1145/1265530.1265535

A more complex example with exponents

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

R

A	B	C	
a	b	c	X
d	b	e	Y
f	g	e	Z

$\pi_{AB}R \bowtie \pi_{BC}R$

A	B	C	
a	b	c	X^2
a	b	e	XY
d	b	c	XY
d	b	e	Y^2
f	g	e	Z^2

$\pi_{AC}R \bowtie \pi_{BC}R$

A	B	C	
a	b	c	X^2
d	b	e	Y^2
d	g	e	YZ
f	b	e	YZ
f	g	e	Z^2

$Q_1 \cup Q_2$

A	B	C	
a	b	c	$2X^2$
a	b	e	XY
d	b	c	XY
d	b	e	$2Y^2$
d	g	e	YZ
f	b	e	YZ
f	g	e	$2Z^2$

Q

A	C	
a	c	$2X^2$
a	e	XY
d	c	XY
d	e	$2Y^2+YZ$
f	e	$YZ+2Z^2$

A more complex example with exponents


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

R

A	B	C	
a	b	c	X =2
d	b	e	Y =5
f	g	e	Z =1

$\pi_{AB}R \bowtie \pi_{BC}R$

A	B	C	
a	b	c	X^2
a	b	e	XY
d	b	c	XY
d	b	e	Y^2
f	g	e	Z^2

$\pi_{AC}R \bowtie \pi_{BC}R$

A	B	C	
a	b	c	X^2
d	b	e	Y^2
d	g	e	YZ
f	b	e	YZ
f	g	e	Z^2

$Q_1 \cup Q_2$

A	B	C	
a	b	c	$2X^2$
a	b	e	XY
d	b	c	XY
d	b	e	$2Y^2$
d	g	e	YZ
f	b	e	YZ
f	g	e	$2Z^2$

Q

A	C	
a	c	$2X^2$
a	e	XY
d	c	XY
d	e	$2Y^2+YZ$
f	e	$YZ+2Z^2$

Let's assume bag semantics and duplicities in the input. How many output tuples do we get?

?

$(\mathbb{N}, +, \cdot, 0, 1)$: Counting derivations / bag semantics

Example from Section 2 of Green, Karvounarakis, Val Tannen. "Provenance Semirings", PODS 2007. https://doi.org/10.1145/1265530.1265535

A more complex example with exponents


$$Q(R) = \pi_{AC}(\underbrace{\pi_{AB}R \bowtie \pi_{BC}R}\ \cup\ \underbrace{\pi_{AC}R \bowtie \pi_{BC}R})$$

R

A	B	C	
a	b	c	X =2
d	b	e	Y =5
f	g	e	Z =1

$\pi_{AB}R \bowtie \pi_{BC}R$

A	B	C	
a	b	c	X^2
a	b	e	XY
d	b	c	XY
d	b	e	Y^2
f	g	e	Z^2

$\pi_{AC}R \bowtie \pi_{BC}R$

A	B	C	
a	b	c	X^2
d	b	e	Y^2
d	g	e	YZ
f	b	e	YZ
f	g	e	Z^2

$Q_1 \cup Q_2$

A	B	C	
a	b	c	$2X^2$
a	b	e	XY
d	b	c	XY
d	b	e	$2Y^2$
d	g	e	YZ
f	b	e	YZ
f	g	e	$2Z^2$

Q

A	C		
a	c	$2X^2$	=8
a	e	XY	=10
d	c	XY	=10
d	e	$2Y^2$+YZ	=55
f	e	YZ+$2Z^2$	=7

Let's assume bag semantics and duplicities in the input. How many output tuples do we get?

$(\mathbb{N}, +, \cdot, 0, 1)$: Counting derivations / bag semantics

Example from Section 2 of Green, Karvounarakis, Val Tannen. "Provenance Semirings", PODS 2007. https://doi.org/10.1145/1265530.1265535

A more complex example with exponents

617

$$Q(R) = \pi_{AC}(\underbrace{\pi_{AB}R \bowtie \pi_{BC}R}_{\pi_{R.A,R.B,R2.C}(R \bowtie_{R.B=R2.B} \rho_{R \rightarrow R2} R)} \cup \underbrace{\pi_{AC}R \bowtie \pi_{BC}R}_{\pi_{R.A,R2.B,R.C}(R \bowtie_{R.C=R2.C} \rho_{R \rightarrow R2} R)})$$

R

A	B	C	
a	b	c	$X = 2$
d	b	e	$Y = 5$
f	g	e	$Z = 1$

```
SELECT A, C, COUNT(*)
FROM (
   SELECT R.A, R.B, R2.C
   FROM R, R R2
   WHERE R.B = R2.B
   UNION ALL
   SELECT R.A, R2.B, R.C
   FROM R, R R2
   WHERE R.C = R2.C) X
GROUP BY A, C
ORDER BY A, C
```

Q

A	C		
a	c	$2X^2$	$=8$
a	e	XY	$=10$
d	c	XY	$=10$
d	e	$2Y^2+YZ$	$=55$
f	e	$YZ+2Z^2$	$=7$

	a character varying	c character varying	count bigint
1	a	c	8
2	a	e	10
3	d	c	10
4	d	e	55
5	f	e	7

SQL example available at: https://github.com/northeastern-datalab/cs3200-activities/tree/master/sql

Example from Section 2 of Green, Karvounarakis, Val Tannen. "Provenance Semirings", PODS 2007. https://doi.org/10.1145/1265530.1265535