CS 595 - Hot topics in database systems: **Data Provenance**

I. Database Provenance

I.1 Provenance Models and Systems

Boris Glavic

September 24, 2012

Outline

1. How-Provenance, Semirings, and Orchestra
   - Introduction
   - Semiring Semantics for Relational Algebra
   - How-Provenance or Provenance Polynomials
   - Relationship to other Provenance Models
   - ORCHESTRA
   - Recap

How-Provenance

Rationale

- In addition to model which tuples influenced a tuple
- . . . model how tuples where combined in the computation
  - Alternative use: need one of the tuples (e.g., union)
  - Conjunctive use: need all tuples together (e.g., join)

Representation

- Formulas over operators and variables
  - Operators define how tuples where combined
  - Variables represent tuples (one variable per tuple)

Approach

Alternative Semantics for the Relational Model

- Tuples are annotated with elements from a semiring
- Define relational algebra operators using the operators of the semiring
- Prove it coincides with set- or bag-semantics for certain semirings

How-Provenance

- Use special semiring that generalizes all semirings
- Elements are symbolic computations

OF TECHNOLOGY

Approach

ORCHESTRA

- **C**ollaborative **D**ata **S**haring **S**ystem
- Independent peers with their own database schema and instance
- Schema mappings between peers schemata
- Peers periodically exchange updates
- Provenance to compute trust in update and deletion propagation

Excursion: Semirings

Commutative Monoids

- $(K, +, 0)$
- A set K
- An operation $K \to K$ (say $+$) with neutral element 0:
  - $(a + b) + c = a + (b + c)$ (associativity)
  - $0 + a = 0 + a = a$ (neutral element)
  - $a + b = b + a$ (associativity)

Example

- $(\mathbb{N}, +, 0)$ - Natural numbers addition
- $(\mathbb{N}, \times, 1)$ - Natural numbers multiplication
- $(\mathbb{B}, \wedge, \mathit{true})$: $\mathbb{B} = \{\mathit{true}, \mathit{false}\}$ - Conjunction over boolean constants
- $(\mathbb{B}, \vee, \mathit{false})$ - Disjunction over boolean constants

Excursion: Semirings

Commutative Semiring

- $(K, +, \times, 0, 1)$
- Set K with operations $+$ and $\times$ (neutral elements 0 and 1)
- $(K, +, 0)$ and $K, \times, 1)$ are commutative monoids
- $a \times (b + c) = (a \times b) + (a \times c)$ (Distributivity)
- $(a + b) \times c = (a \times c) + (b \times c)$ (Distributivity)
- $a \times 0 = 0 \times a = 0$ (multiplication with 0)

Example

- $(\mathbb{N}, +, \times, 0, 1)$ - Natural numbers with addition and multiplication
- $(\mathbb{B}, \vee, \wedge, \mathit{false}, \mathit{true})$ - Conjunctions and disjunctions over boolean constants

Homomorphism

Definition

Homomorphism

- Given two semirings K and K'
- A function from K to K' is a homomorphism h iff:
  - $h(a + b) = h(a) + h(b)$
  - $h(a \times b) = h(a) \times h(b)$
  - $h(0) = 0$
  - $h(1) = 1$
- Homomorphism from K to K' $\Rightarrow$ K is more general then K'

Free Objects

- Given an algebraic structure a free object is one with an homomorphism into all other objects of this type
  - E.g., the free commutative semiring is an structure with homomorphism into all other commutative semirings

Free Objects

- Given an algebraic structure a free object is one with an homomorphism into all other objects of this type
  - E.g., the free commutative semiring is an structure with homomorphism into all other commutative semirings
- $\Rightarrow$The free semiring is the most general semiring

Free Objects

- Given an algebraic structure a free object is one with an homomorphism into all other objects of this type
  - E.g., the free commutative semiring is an structure with homomorphism into all other commutative semirings
- ⇒The free semiring is the most general semiring
- ⇒Only equivalences enforced by the structure being semiring can hold
  - For any additional equivalence: Find semiring where equivalence does not hold ⇒No homomorphism! contradiction

Free Objects

- Given an algebraic structure a free object is one with an homomorphism into all other objects of this type
  - E.g., the free commutative semiring is an structure with homomorphism into all other commutative semirings
- $\Rightarrow$The free semiring is the most general semiring
- $\Rightarrow$Only equivalences enforced by the structure being semiring can hold
  - For any additional equivalence: Find semiring where equivalence does not hold $\Rightarrow$No homomorphism! contradiction
- $\Rightarrow$Elements of free semiring are uninterpreted expressions
  - Placeholders for semiring elements
  - Do not interpret semiring operation

Free Objects

Example

- $(a + b) \times c$ is an element
- $k_1 = (a + b)$ and $k_2 = (c \times d)$: $k_1 + k_2 = (a + b) + (c \times d)$

ILLINOIS INSTITUTE OF TECHNOLOGY

Outline

Semiring Annotated Relations

K-Relations

- *U-tuple*: tuples over set of attributes U
  - $U - Tup$ = set of all U-tuples
- Semiring K
- A *K-relation R* over a set of attributes U is
  - function $U - Tup \rightarrow K$
  - $support(R) = \{t \mid R(t) \neq 0\}$ is finite

Notation

- $K = \mathbb{N}$

R

	a
1	1
3	2

Interpretations of Semirings

Semiring Interpretations

- $(\mathbb{N}, +, \times, 0, 1)$: Tuples annotated with integers
  $\Rightarrow$Bag-semantics

Example

R

	a
1	1
3	2

Interpretations of Semirings

Semiring Interpretations

- $(\mathbb{B}, \vee, \wedge, \mathit{false}, \mathit{true})$: $\mathbb{B} = \{\mathit{false}, \mathit{true}\}$: Tuples with true/false annotations ⇒Set-semantics

Example

R

	a
true	1
true	2

Interpretations of Semirings

Semiring Interpretations

- $(PosBool(X), \vee, \wedge, false, true)$: $PosBool(X)$ = set of variables: Tuples annotated with boolean expressions $\Rightarrow$c-tables (probabilistic databases)

Example

R

	a
$x_1 \vee (x_2 \wedge x_3)$	1
x_4	2

ILLINOIS INSTITUTE OF TECHNOLOGY

Relational Algebra for K-relations

Rationale

- Express relational algebra operators as semiring operations
- Sanity checks:
  - For $K = \mathbb{B} \Rightarrow$ same results (equivalences) as set-semantics
  - For $K = \mathbb{N} \Rightarrow$ same results (equivalences) as bag-semantics

ILLINOIS INSTITUTE OF TECHNOLOGY

Operator Definitions

Selection

- $(\sigma_C(R))(t) = R(t) \times C(t)$
- Selection predicate C is function $U - Tup \rightarrow \{0, 1\}$
  - Recall $a \times 0 = 0$ and $a \times 1 = a$

Projection

- $(\pi_A(R))(t) = \sum_{t=t'.A} R(t')$
  - $A \subseteq U$

Union

- $(R_1 \cup R_2)(t) = R_1(t) + R_2(t)$

Operator Definitions

Natural Join

- $(R_1 \bowtie R_2)(t) = R_1(t_1) \times R_2(t_2)$
  - $t_1 = t.U_1$
  - $t_2 = t.U_2$

Renaming

- $(\rho_\beta(R))(t) = R(t \circ \beta)$
  - $\beta : U \rightarrow U'$ attribute renaming

ILLINOIS INSTITUTE OF TECHNOLOGY

Evaluation Example

Example

- Semiring is $\mathbb{N}$
- $q = \sigma_{a=1}(\pi_a(R))$
- $q(t) = \sum_{t'.a=t} R(t') \times (a=1)(t)$

R

	a	b
2	1	3
3	1	4
2	2	4

$(2+3) \times 1 = 5$
$2 \times 0 = 0$

Q

a
1
2

Equivalence Examples

- Union:
  - Associative: $R \cup (S \cup T) = (R \cup S) \cup T$
  - Commutative: $R \cup S = S \cup R$
  - Identity $\emptyset$: $R \cup \emptyset = R$
- Join
  - Associative: $R \bowtie (S \bowtie T) = (R \bowtie S) \bowtie T$
  - Commutative: $R \bowtie S = S \bowtie R$
- Selection
  - $\sigma_{false}(R) = \emptyset$
  - $\sigma_{true}(R) = R$

ILLINOIS INSTITUTE OF TECHNOLOGY

Homomorphisms in Query Evaluation

Homomorphisms commute with Query Evaluation

- $Q(h(I)) = h(Q(I))$
- $\Rightarrow$We can apply h either before or after evaluating the query without affecting the result

Example

- Homomorphism from $\mathbb{N}$ (bag-semantics) to $\mathbb{B}$ (set-semantics): $h(n) = true$ except $h(0) = false$
- E.g., $\sigma_{a>1}(R)$

R

	a
3	1
2	2

R

	a
true	1
true	2

Outline

Provenance Polynomials

Rationale

- Use semiring annotations to model provenance
- Annotate a query result tuple with the semiring expression that was used to compute it
- $\Rightarrow$need free semiring

Provenance Polynomials Semiring

- $(\mathbb{N}[I], +, \times, 0, 1)$
- $\mathbb{N}[I]$ - Polynomials with natural number exponents
  - Variables: One per tuple in I
- Convention: annotate each instance tuple with a variable named after its tuple *id*

Provenance Polynomials Example

Example

$$q = \pi_a(R)$$

R

	a	b
t_1	1	2
t_2	1	3

Q

	a
$t_1 + t_2$	1

ILLINOIS INSTITUTE OF TECHNOLOGY

Provenance Polynomials Example II

Example

$$q = \pi_{Name}(E \bowtie \sigma_{Dep=CS}(P) \bowtie A)$$

Employee

	Id	Name
e_1	1	Peter
e_2	2	Gertrud
e_2	3	Michael

Assigned

	PName	Id
a_1	Server	1
a_2	Server	2
a_3	Webpage	2
a_4	Fire CS	3

Project

	PName	Dep
p_1	Server	CS
p_2	Webpage	CS
p_3	Fire CS	HR

Provenance Polynomials Example II

Example

$$q = \pi_{Name}(E \bowtie \sigma_{Dep=CS}(P) \bowtie A)$$

$$(q)(t) = \sum_{u.A=t} E(u.E) \times P(u.P) \times (Dep = CS)(u.P) \times A(u.P)$$

Q

	Name
$e_1 \times a_1 \times p_1$	Peter
$(e_2 \times a_2 \times p_1) + (e_2 \times a_3 \times p_2)$	Gertrud

Employee

	Id	Name
e_1	1	Peter
e_2	2	Gertrud
e_2	3	Michael

Assigned

	PName	Id
a_1	Server	1
a_2	Server	2
a_3	Webpage	2
a_4	Fire CS	3

Project

	PName	Dep
p_1	Server	CS
p_2	Webpage	CS
p_3	Fire CS	HR

The Fundamental Property

- The semiring of provenance polynomials is the free commutative semiring
- $\Rightarrow$there exists a homomorphism from $\mathbb{N}[I]$ into any commutative semiring
- $Eval_K : \mathbb{N}[I] \rightarrow K$ is this unique homomorphism defined as
  - Replace each tuple variable t with the element of K assigned to the tuple represented by t
  - Interpret the abstract operations from $\mathbb{N}[I]$ as operations from K

ILLINOIS INSTITUTE OF TECHNOLOGY

Example Application of the Fundamental Property

Example

$$q = \pi_a(R)$$

R

	a	b
t_1	1	2
t_2	1	3

Q

	a
$t_1 + t_2$	1

Example Application of the Fundamental Property

Example

$$q = \pi_a(R)$$

Interpretation in $\mathbb{N}$

R

	a	b
2	1	2
1	1	3

Q

	a
$t_1 + t_2$	1

ILLINOIS INSTITUTE OF TECHNOLOGY

Example Application of the Fundamental Property

Example

$$q = \pi_a(R)$$

Interpretation in $\mathbb{N}$

R

	a	b
2	1	2
1	1	3

Q

	a
$2 + 1 = 3$	1

ILLINOIS INSTITUTE OF TECHNOLOGY

The "How" Part

Interpretation of $+$ and $\times$

- $+$: Alternative use of tuples
  - Operators: Union, Projection
  - Check set-semantics: only one tuples is need $\Rightarrow \vee$ as $+$ operation
  - Check bag-semantics: multiplicities are additive $\Rightarrow$natural number addition as $+$
- $\times$: Conjunctive use of tuples
  - Operations: Join
  - Check set-semantics: both tuples are needed $\Rightarrow \wedge$ as $\times$ operation
  - Check bag-semantics: multiplicities of matching tuples are multiplied $\Rightarrow$natural number multiplication as $\times$

Insensitivity to Query Rewrite

Bag-semantics

- Modelling relational algebra as commutative semiring operations
  - Possible, because same equivalences
- $\mathbb{N}[I]$ is free commutative semiring
- $\Rightarrow$Equivalences for $\mathbb{N}[I]$ and bag-semantics are the same!
- $\Rightarrow\mathbb{N}[I]$ is insensitive

ILLINOIS INSTITUTE OF TECHNOLOGY

Insensitivity to Query Rewrite

Set-semantics

- $\mathbb{N}[I]$ no longer insensitive
  - E.g., $R \not\equiv R \bowtie R$
- $\mathbb{B}[I]$: polynomials with boolean coefficients and exponents has same equivalences as set semantics
- There exists an homomorphism from $\mathbb{N}[I]$ to $\mathbb{B}[I]$ ($\mathbb{N}[I]$ is free object!)
- $\Rightarrow$apply equivalences of $\mathbb{B}[I]$ to $\mathbb{N}[I]$ then insensitive for set-semantics

ILLINOIS INSTITUTE OF TECHNOLOGY

How-provenance

Notation

- We write $\mathbb{N}[I](q, t)$ for
- $(q)(t)$ evaluated in $\mathbb{N}[I]$
- also use this for other semirings K

Beyond Positive Relational Algebra

Set Difference

- Need additional operator $-$
- $\Rightarrow$from semiring to structures $(S, +, \times, -, 1, 0)$
  - Different equivalences hold!
- Provenance use (more complex) free object for such structures

Aggregation

- Annotate attribute values with combinations of
  - tuple semiring provenance
  - annotation for values for computations on values (representing aggregation)

Outline

1. How-Provenance, Semirings, and Orchestra
   - Introduction
   - Semiring Semantics for Relational Algebra
   - How-Provenance or Provenance Polynomials
   - Relationship to other Provenance Models
   - ORCHESTRA
   - Recap

Relationship of Provenance Polynomials and other Provenance Models

Rationale

- How is the provenance polynomials model related to other provenance models?
- Can we find semirings that models, e.g., Why-Provenance?

ILLINOIS INSTITUTE OF TECHNOLOGY

Why-Provenance

Semiring

- $K_{Why} = (\mathcal{P}(\mathcal{P}(I)), \cup, \Cup, \emptyset, \{\emptyset\})$
- $\mathcal{P}$ = powerset
- $\Rightarrow \mathcal{P}(\mathcal{P}(I))$ is all sets containing subsets of the instance I
- $\Rightarrow$all potential sets of witnesses
- $+$ is normal set union
- $\times$ is $S_1 \Cup S_2 = \{(a \cup b) \mid a \in S_1 \wedge b \in S_2\}$
  - $\Rightarrow$pairwise union
  - $\Rightarrow$combining witnesses

ILLINOIS INSTITUTE OF TECHNOLOGY

Why-Provenance

Example

$$q = \pi_{Name}(E \bowtie \sigma_{Dep=CS}(P) \bowtie A)$$

$$(q)(t) = \sum_{u.A=t} E(u.E) \times P(u.P) \times (Dep = CS)(u.P) \times A(u.P)$$

Q

	Name
$\{\{e_1, a_1, p_1\}\}$	Peter
$\{\{e_2, a_2, p_1\}, \{e_2, a_3, p_2\}\}$	Gertrud

Employee

	Id	Name
$\{\{e_1\}\}$	1	Peter
$\{\{e_2\}\}$	2	Gertrud
$\{\{e_2\}\}$	3	Michael

Assigned

	PName	Id
$\{\{a_1\}\}$	Server	1
$\{\{a_2\}\}$	Server	2
$\{\{a_3\}\}$	Webpage	2
$\{\{a_4\}\}$	Fire CS	3

Project

	PName	Dep
$\{\{p_1\}\}$	Server	CS
$\{\{p_2\}\}$	Webpage	CS
$\{\{p_3\}\}$	Fire CS	HR

Insensitive Why-Provenance

Semiring

- $K_{IWhy} = (min(\mathcal{P}(\mathcal{P}(I))), \cup_{min}, \Cup_{min}, \emptyset, \{\emptyset\})$
- $\Rightarrow \mathcal{P}(\mathcal{P}(I))$ is all sets containing subsets of the instance I
- $min(S) = \{a \mid a \in S \wedge \nexists b \in S : b \subseteq a\}$
- $S_1 \cup_{min} S_2 = min(S_1 \cup S_2)$
- $S_1 \Cup_{min} S_2 = min(S_1 \Cup S_2)$
- $\Rightarrow$Same operations, compute minimal elements

Insensitive Why-Provenance

Example

$$q = \pi_{Name}(E \bowtie \sigma_{Dep=CS}(P) \bowtie A)$$

$$(q)(t) = \sum_{u.A=t} E(u.E) \times P(u.P) \times (Dep = CS)(u.P) \times A(u.P)$$

Q

Provenance	Name
$\{\{e_1, a_1, p_1\}\}$	Peter
$\{\{e_2, a_2, p_1\}, \{e_2, a_3, p_2\}\}$	Gertrud

Employee

	Id	Name
$\{\{e_1\}\}$	1	Peter
$\{\{e_2\}\}$	2	Gertrud
$\{\{e_2\}\}$	3	Michael

Assigned

	PName	Id
$\{\{a_1\}\}$	Server	1
$\{\{a_2\}\}$	Server	2
$\{\{a_3\}\}$	Webpage	2
$\{\{a_4\}\}$	Fire CS	3

Project

	PName	Dep
$\{\{p_1\}\}$	Server	CS
$\{\{p_2\}\}$	Webpage	CS
$\{\{p_3\}\}$	Fire CS	HR

Insensitive Why-Provenance

Semiring

- $K_{IWhy} = (min(\mathcal{P}(\mathcal{P}(I))), \cup_{min}, \uplus_{min}, \emptyset, \{\emptyset\})$
- $\Rightarrow \mathcal{P}(\mathcal{P}(I))$ is all sets containing subsets of the instance I
- $min(S) = \{a \mid a \in S \wedge \nexists b \in S : b \subseteq a\}$
- $S_1 \cup_{min} S_2 = min(S_1 \cup S_2)$
- $S_1 \uplus_{min} S_2 = min(S_1 \uplus S_2)$
- $\Rightarrow$Same operations, compute minimal elements

ILLINOIS INSTITUTE OF TECHNOLOGY

Insensitive Why-Provenance

Example

$$q = \pi_{Name}(E \bowtie \sigma_{Dep=CS}(P) \bowtie A)$$

$$(q)(t) = \sum_{u.A=t} E(u.E) \times P(u.P) \times (Dep = CS)(u.P) \times A(u.P)$$

Q

	Name
$\{\{e_1, a_1, p_1\}\}$	Peter
$\{\{e_2, a_2, p_1\}, \{e_2, a_3, p_2\}\}$	Gertrud

Employee

	Id	Name
$\{\{e_1\}\}$	1	Peter
$\{\{e_2\}\}$	2	Gertrud
$\{\{e_2\}\}$	3	Michael

Assigned

	PName	Id
$\{\{a_1\}\}$	Server	1
$\{\{a_2\}\}$	Server	2
$\{\{a_3\}\}$	Webpage	2
$\{\{a_4\}\}$	Fire CS	3

Project

	PName	Dep
$\{\{p_1\}\}$	Server	CS
$\{\{p_2\}\}$	Webpage	CS
$\{\{p_3\}\}$	Fire CS	HR

Lineage

Different Model

- The inventors of provenance polynomials consider a slightly different Lineage model
- Provenance is a set of tuples instead of a list of sets of tuples

Semiring

- $K_{Lin} = (\mathcal{P}(I), \cup_{\perp}, \cup^{*}_{\perp}, \perp, \emptyset)$
- $\Rightarrow \mathcal{P}(I)$ is all subsets of the instance I
- $\perp$ is a not defined element
- $\cup_{\perp}$ and $\cup^{*}_{\perp}$ are union with different behaviour on $\perp$
- $\perp \cup_{\perp} S = S \cup_{\perp} \perp = S$
- $\perp \cup^{*}_{\perp} S = S \cup^{*}_{\perp} \perp = \emptyset$

Lineage

Example

$$q = \pi_{Name}(E \bowtie \sigma_{Dep=CS}(P) \bowtie A)$$

$$(q)(t) = \sum_{u.A=t} E(u.E) \times P(u.P) \times (Dep = CS)(u.P) \times A(u.P)$$

Q

	Name
$\{e_1, a_1, p_1\}$	Peter
$\{e_2, a_2, a_3, p_1, p_2\}$	Gertrud

Employee

	Id	Name
$\{e_1\}$	1	Peter
$\{e_2\}$	2	Gertrud
$\{e_2\}$	3	Michael

Assigned

	PName	Id
$\{a_1\}$	Server	1
$\{a_2\}$	Server	2
$\{a_3\}$	Webpage	2
$\{a_4\}$	Fire CS	3

Project

	PName	Dep
$\{p_1\}$	Server	CS
$\{p_2\}$	Webpage	CS
$\{p_3\}$	Fire CS	HR

Lineage

"Real" Lineage

- Can we also model the list of sets of tuples lineage as a semiring?

Lineage

"Real" Lineage

- Can we also model the list of sets of tuples lineage as a semiring?
- NO!:
  - Assume existence of semiring K_{RLin} that models lineage
  - Equivalent queries $q = R \cup S$ and $q' = S \cup R$
  - Assume tuple t is in the result of q/q' and was derived from r_1 and s_1
  - Lineage: $Lin(q,t) =< \{r_1\}, \{s_1\} > \neq < \{s_1\}, \{r_1\} >= Lin(q', t)$
  - Evaluation in K_{RLin}: $(q)(t) = r_1 + s_1 = s_1 + r_1 = (q')(t)$
  - $\Rightarrow$no assumptions except that K_{RLin} is semiring
  - $\Rightarrow K_{RLin}$ cannot exists

Perm Influence Contribution Semantics

Semiring

- Cannot exists for the same reason as Lineage
  - Assume existence of semiring K_{PI} that models PI-CS
  - Equivalent queries $q = R \cup S$ and $q' = S \cup R$
  - Assume tuple t is in the result of q/q' and was derived from r_1 and s_1
  - Lineage: $\mathcal{PI}(q,t) = \{< r_1, s_1 >\} \neq \{< s_1, r_1 >\} = \mathcal{PI}(q',t)$
  - Evaluation in K_{PI}: $(q)(t) = r_1 + s_1 = s_1 + r_1 = (q')(t)$
  - $\Rightarrow K_{PI}$ cannot exists

ILLINOIS INSTITUTE OF TECHNOLOGY

Perm Influence Contribution Semantics

Discussion

- Lineage and PI-CS consider the order of leaves in the algebra tree
- However, equivalent queries can have different orders
- If we abstract from the order, is the result expressible in the semiring model?
- **Rationale**: Define mapping H from $\mathcal{PI}$ to $\mathbb{N}[I]$ that gets rid of the order

ILLINOIS INSTITUTE OF TECHNOLOGY

Perm Influence Contribution Semantics

From $\mathcal{PI}$ to $\mathbb{N}[I]$

- Witness-lists are basically $\times$
- The set of witness-lists is basically $+$

$$H(\mathcal{PI}(q,t)) = \sum_{w \in \mathcal{PI}(q,t)} \prod_{i \in \{1,\ldots,n\}} w'[i]$$

$$w'[i] = \begin{cases} w[i] & \text{if } w[i] \neq \perp \\ 1 & \text{otherwise} \end{cases}$$

Perm Influence Contribution Semantics

Example

$$q = \pi_a(R) \cup (\pi_a(R \bowtie S))$$

$$\mathcal{PI}(q, t_1) = \{< r_1, \perp >, < r_1, s_1 >\}$$

$$H(\mathcal{PI}(q, t_1)) = \sum_{w \in \mathcal{PI}(q,t_1)} \prod_{i \in \{1,\ldots,n\}} w'[i]$$

$$= r_1 \times 1 + r_1 \times s_1 = r_1 + r_1 \times s_1$$

$$= \mathbb{N}[I](q, t_1)$$

S

	b
r_1	1
r_2	2

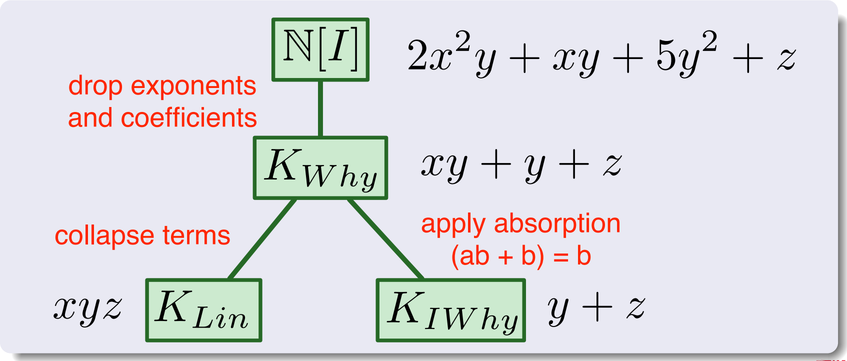
S

	a
s_1	1

Q

	a
t_1	1
t_2	2

Relationships between Provenance Semirings



ILLINOIS INSTITUTE OF TECHNOLOGY

Outline

ORCHESTRA

Overview

- **C**ollaborative **D**ata **S**haring **S**ystem
- Network of peers
- Each peer has independent schema and instance
- Peers update their instances without restrictions
- Schema mappings define relationships between schemata
  - Can be partial
- Periodically peers trigger exchange of updates based on mappings

ILLINOIS INSTITUTE OF TECHNOLOGY

Schema mappings

- Schema mapping: Logical constraints that define the relationship between two schemata
- Different schema may store the same information in different structure
- Schema mappings model these structures in the schema relate
- With some extra mechanism can be use to translate data from one schema into the other

Example

- **Schema** S_1: `Person(Name, AddrId)`, `Address(Id, City, Street)`
- **Schema** S_2: `LivesAt(Name, City)`

Update Exchange

- Each peer updates its instance as he pleases
- A log of update operations is kept
- Peers can trigger an update exchange

Update Exchange

- Determine updates since last exchange
- Translate updates from peers according to schema mappings
- Eagerly compute provenance during update exchange

ILLINOIS INSTITUTE OF TECHNOLOGY

Provenance in ORCHESTRA

- Use $\mathbb{N}[I]$
- Add functions $m_1, \ldots, m_n$ to represent mappings
- E.g., $m_1(x + yz) + m_2(u)$ means that tuple was derived by
  - applying mapping m_1 to x, y, z
  - applying mapping m_2 to u

ILLINOIS INSTITUTE OF TECHNOLOGY

Provenance Use in ORCHESTRA

Trust

- Instead of applying all update: only apply “trusted” updates
- Peers decide on a per mapping/peer basis whether they trust data.
  - Use Trust semiring: $(R^{\text{inf}}, min, +, \text{inf}, 0)$
  - Evaluate provence in the trust semiring using the trust value for peers and mappings

Provenance Use in ORCHESTRA

Deletion Propagation

- Deletion in semiring model $\Rightarrow$annotating with 0 element of semiring
- We have provenance for query result
- Assume set D of tuples got deleted
- Set every occurrence of D in the provenance of some tuple t to 0
- Compute whether t is still derivable
- Here even without index on provenance useful, because repeating whole update exchange is unfeasible

OF TECHNOLOGY

Outline

Recap

Semiring Semantics for the Relational Model

- Alternative semantics for relational algebra
- Given a semiring $(K, +, \times, 0, 1)$
  - K-relations are functions from tuples of an arity U to semiring elements
  - Operators take functions (relations) as input and produce an output function (relation)
- Using different semirings we get standard semantics or extensions of the relational model
  - $(\mathbb{B}, \vee, \wedge, \mathit{false}, \mathit{true})$: Set semantics
  - $(\mathbb{N}, +, \times, 0, 1)$: Bag semantics
  - $(\mathit{PosBool}(X), \vee, \wedge, \mathit{false}, \mathit{true})$: c-tables

Recap

How-Provenance (Provenance semiring)

- **Rationale**: Provenance for t is expression that represent the semiring computation that lead to creation of tuple t.
- **Representation**: Polynomial over tuple variables (= element of Provenance Semiring)
- **Syntactic Definition**:
  - For USPJ queries + extensions for A and D
- **The Fundamental Property**: Given an query result in $\mathbb{N}[I]$, we can compute the query result for any semiring K from that
- **Relation to other Provenance Types**:
  - Semirings that model other provenance models
  - Why-Provenance: $(\mathcal{P}(\mathcal{P}(I)), \cup, \Cup, \emptyset, \{\emptyset\})$
  - IWhy-Provenance: $(min(\mathcal{P}(I)), \cup_{min}, \times_{min}, \emptyset, \{\emptyset\})$
  - Lineage*: $(\mathcal{P}(I) \cup \{\bot\}, +, \times, \bot, \emptyset)$

Recap

ORCHESTRA

- Peer-to-Peer update exchange system
- Schema mappings between peers
- Updates are exchanged between periodically based on mappings
- Provenance used for
  - Trust
  - Deletion propagation

ILLINOIS INSTITUTE OF TECHNOLOGY

Provenance Model Comparison

Property	Why	Lin	PI-CS	Where	How
Representation	Set of Set of Tuples	List of Set of Tuples	Set/Bag of List of Tuples	Sets of Attribute Value Positions	Values of provenance semiring
Granularity	Tuple	Tuple	Tuple	Attribute Value	Tuple
Language Support	USPJ	ASPJ-Set	ASPJ-Set + Nested subqueries	U-SPJ	A*SPJ-UD*
Semantics	Set	Set + Bag*	Bag	Set	Set + Bag
Variants	Wit, Why, IWhy	Set/Bag	Influence + Copy	SPJ + Insensitive + Insensitive Union	semirings
Definition	Decl. - Synt. - Decl./Synt.	Decl. + Synt.	Decl. + Synt.	Synt.	Synt.
Design Principles	Sufficiency - No false positives	Sufficiency + No false negatives + no false positives	Sufficiency + No false negatives + No false positives	Copying	Equivalent to query evaluation
Systems	-	WHIPS	Perm	DBNotes	ORCHESTRA
Insensitivity	Yes - No - Yes	No	No	No - Yes - Yes	Yes

Literature I

Yael Amsterdamer, Daniel Deutch, and Val Tannen.
On the limitations of provenance for queries with difference.
In Tapp '11: 3rd usenix workshop on the theory and practice of provenance, 2011.

Y. Amsterdamer, D. Deutch, T. Milo, and V. Tannen.
On provenance minimization.
In PODS '11, 2011.

Y. Amsterdamer, D. Deutch, and V. Tannen.
Provenance for Aggregate Queries.
Arxiv preprint arXiv:1101.1110, 2011.

T.J. Green, G. Karvounarakis, and Z.G.I.V. Tannen.
Provenance in ORCHESTRA.
, 2010.

Nicholas E. Taylor and Zachary G. Ives.
Reliable storage and querying for collaborative data sharing systems.
In ICDE '10, 2010.

G. Karvounarakis, Z.G. Ives, and V. Tannen.
Querying data provenance.
In Proceedings of the 2010 international conference on management of data, 951–962, ACM, , 2010.

ILLINOIS INSTITUTE OF TECHNOLOGY

Literature II

F. Geerts and A. Poggi.
On database query languages for K-relations.
Journal of Applied Logic, 8(2):173–185, 2010.

Todd J. Green.
Containment of Conjunctive Queries on Annotated Relations.
In ICDT '09: Proceedings of the 16th International Conference on Database Theory, 296–309, 2009.

Todd J. Green.
Collaborative data sharing with mappings and provenance.
PhD thesis, University of Pennsylvania, 2009.

Todd J. Green, Zachary G. Ives, and Val Tannen.
Reconcilable differences.
In ICDT '09: Proceedings of the 16th International Conference on Database Theory, 212–224, Saint Petersburg, Russia, March 2009. , .

Zachary G. Ives, Todd J. Green, Grigoris Karvounarakis, Nicholas E. Taylor, Val Tannen, Partha Pratim Talukdar, Marie Jacob, and Fernando Pereira.
The ORCHESTRA Collaborative Data Sharing System.
SIGMOD Record, 37(2):26–32, 2008.

J. Nathan Foster, Todd J. Green, and Val Tannen.
Annotated XML: Queries and Provenance.
In PODS '08: Proceedings of the 27th Symposium on Principles of Database Systems, 2008.

ILLINOIS INSTITUTE OF TECHNOLOGY

Literature III

Todd J. Green, Gregory Karvounarakis, and Val Tannen.
Provenance Semirings.
In PODS '07: Proceedings of the 26th Symposium on Principles of Database Systems, 31–40, 2007.

Todd J. Green, Grigoris Karvounarakis, Nicholas E. Taylor, Olivier Biton, Zachary G. Ives, and Val Tannen.
ORCHESTRA: Facilitating Collaborative Data Sharing.
In SIGMOD '07: Proceedings of the 33th SIGMOD International Conference on Management of Data, 2007.

Todd J. Green, Grigoris Karvounarakis, Zachary G. Ives, and Val Tannen.
Update Exchange with Mappings and Provenance.
In VLDB '07: Proceedings of the 33th International Conference on Very Large Data Bases, 675–686, 2007.

Floris Geerts and Jan Van den Bussche.
Relational Completeness of Query Languages for Annotated Databases.
Lecture Notes in Computer Science, 4797:127, 2007.

Zachary G. Ives, Nitin Khandelwal, Aneesh Kapur, and Murat Cakir.
ORCHESTRA: Rapid, Collaborative Sharing of Dynamic Data.
In CIDR '05: Proceedings of the 2th Conference on Innovative Data Systems Research, 2005.

ILLINOIS INSTITUTE OF TECHNOLOGY