CS294-248 Special Topics in Database Theory
Unit 7: Semirings and K-Relations

Dan Suciu

University of Washington

Outline

- Today: Semirings, K-Relations; positive RA only.

- Thursday: FO over Semirings (guest lecturer Val Tannen)

Semirings

Motivation

Traditional relations: $R(a, b)$ is either true, or false. Boolean.

Many applications require a more nuanced value.

- Bag semantics: $R(a, b)$ occurs 5 times; $R(c, d)$ occurs 0 times.
- Linear algebra: $R[i, j] = -0.5$.
- Security: $R(a, b)$ is secret; $R(c, d)$ is top secret.
- Provenance: $R(a, b)$ was obtained as follows ......

The algebraic concept for all these is a semiring.

Motivation

Traditional relations: $R(a,b)$ is either true, or false. Boolean.

Many applications require a more nuanced value.

- Bag semantics: $R(a,b)$ occurs 5 times; $R(c,d)$ occurs 0 times.
- Linear algebra: $R[i,j] = -0.5$.
- Security: $R(a,b)$ is secret; $R(c,d)$ is top secret.
- Provenance: $R(a,b)$ was obtained as follows ......

The algebraic concept for all these is a semiring.

Motivation

Traditional relations: $R(a, b)$ is either true, or false. Boolean.

Many applications require a more nuanced value.

- Bag semantics: $R(a, b)$ occurs 5 times; $R(c, d)$ occurs 0 times.
- Linear algebra: $R[i, j] = -0.5$.
- Security: $R(a, b)$ is secret; $R(c, d)$ is top secret.
- Provenance: $R(a, b)$ was obtained as follows . . . . . .

The algebraic concept for all these is a semiring.

Motivation

Traditional relations: $R(a, b)$ is either true, or false. Boolean.

Many applications require a more nuanced value.

- Bag semantics: $R(a, b)$ occurs 5 times; $R(c, d)$ occurs 0 times.
- Linear algebra: $R[i, j] = -0.5$.
- Security: $R(a, b)$ is secret; $R(c, d)$ is top secret.
- Provenance: $R(a, b)$ was obtained as follows . . . . . .

The algebraic concept for all these is a semiring.

Motivation

Traditional relations: $R(a, b)$ is either true, or false. Boolean.

Many applications require a more nuanced value.

- Bag semantics: $R(a, b)$ occurs 5 times; $R(c, d)$ occurs 0 times.
- Linear algebra: $R[i, j] = -0.5$.
- Security: $R(a, b)$ is secret; $R(c, d)$ is top secret.
- Provenance: $R(a, b)$ was obtained as follows ......

The algebraic concept for all these is a semiring.

Motivation

Traditional relations: $R(a, b)$ is either true, or false. Boolean.

Many applications require a more nuanced value.

- Bag semantics: $R(a, b)$ occurs 5 times; $R(c, d)$ occurs 0 times.
- Linear algebra: $R[i, j] = -0.5$.
- Security: $R(a, b)$ is secret; $R(c, d)$ is top secret.
- Provenance: $R(a, b)$ was obtained as follows ......

The algebraic concept for all these is a semiring.

Monoids

Definition

A monoid is a tuple $\boldsymbol{M} = (M, \circ, \mathbf{1})$, where:

- $\circ : M \times M \rightarrow M$ is a binary function (operation).
- $\mathbf{1} \in M$ is an element.
- $\circ$ is associative: $(x \circ y) \circ z = x \circ (y \circ z)$.
- $\mathbf{1}$ is a left and right identity: $\mathbf{1} \circ x = x \circ \mathbf{1} = x$.

Monoids

Definition

A monoid is a tuple $\boldsymbol{M} = (M, \circ, \mathbf{1})$, where:

- $\circ : M \times M \to M$ is a binary function (operation).
- $\mathbf{1} \in M$ is an element.
- $\circ$ is associative: $(x \circ y) \circ z = x \circ (y \circ z)$.
- $\mathbf{1}$ is a left and right identity: $\mathbf{1} \circ x = x \circ \mathbf{1} = x$.

The monoid is commutative if $x \circ y = y \circ x$.

Monoids

Definition

A monoid is a tuple $\boldsymbol{M} = (M, \circ, \mathbf{1})$, where:

- $\circ : M \times M \to M$ is a binary function (operation).
- $\mathbf{1} \in M$ is an element.
- $\circ$ is associative: $(x \circ y) \circ z = x \circ (y \circ z)$.
- $\mathbf{1}$ is a left and right identity: $\mathbf{1} \circ x = x \circ \mathbf{1} = x$.

The monoid is commutative if $x \circ y = y \circ x$.

The monoid is a group if $\forall x \in M$, $\exists y \in M$ s.t. $x \circ y = y \circ x = \mathbf{1}$.

Monoids

Definition

A monoid is a tuple $\boldsymbol{M} = (M, \circ, \mathbf{1})$, where:

- $\circ : M \times M \to M$ is a binary function (operation).
- $\mathbf{1} \in M$ is an element.
- $\circ$ is associative: $(x \circ y) \circ z = x \circ (y \circ z)$.
- $\mathbf{1}$ is a left and right identity: $\mathbf{1} \circ x = x \circ \mathbf{1} = x$.

The monoid is commutative if $x \circ y = y \circ x$.

The monoid is a group if $\forall x \in M$, $\exists y \in M$ s.t. $x \circ y = y \circ x = \mathbf{1}$.
prove that y is unique Notation: $y = x^{-1}$.

Examples

Which ones are groups?

$(\mathbb{R}, +, 0)$

$(\mathbb{R}, *, 1)$

$(\mathbb{R}^{n\times n}, \cdot, I_n)$: $n \times n$ matrices w/ multiplication

$(S_n, \circ, id_n)$ permutations of n elements w/ composition

$(2^\Omega, \cap, \Omega)$

$(2^\Omega, \cup, \emptyset)$

Semirings

Definition

A semiring is a tuple $\boldsymbol{S} = (S, \oplus, \otimes, \mathbf{0}, \mathbf{1})$ where:

- $(\boldsymbol{S}, \oplus, \mathbf{0})$ is a commutative monoid.
- $(\boldsymbol{S}, \otimes, \mathbf{1})$ is a monoid.
- $\otimes$ distributes over $\oplus$:
$$x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z)$$
$$(y \oplus z) \otimes x = (y \otimes x) \oplus (z \otimes x)$$
- $\mathbf{0}$ is absorbing, also called annihilating: $x \otimes \mathbf{0} = \mathbf{0} \otimes x = \mathbf{0}$

Semirings

Definition

A semiring is a tuple $\boldsymbol{S} = (S, \oplus, \otimes, \mathbf{0}, \mathbf{1})$ where:

- $(\boldsymbol{S}, \oplus, \mathbf{0})$ is a commutative monoid.
- $(\boldsymbol{S}, \otimes, \mathbf{1})$ is a monoid.
- $\otimes$ distributes over $\oplus$:
$$x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z)$$
$$(y \oplus z) \otimes x = (y \otimes x) \oplus (z \otimes x)$$
- $\mathbf{0}$ is absorbing, also called annihilating: $x \otimes \mathbf{0} = \mathbf{0} \otimes x = \mathbf{0}$

$\boldsymbol{S}$ is a commutative semiring if $\otimes$ is commutative.

Semirings

Definition

A semiring is a tuple $\boldsymbol{S} = (S, \oplus, \otimes, \mathbf{0}, \mathbf{1})$ where:

- $(\boldsymbol{S}, \oplus, \mathbf{0})$ is a commutative monoid.
- $(\boldsymbol{S}, \otimes, \mathbf{1})$ is a monoid.
- $\otimes$ distributes over $\oplus$:
$$x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z)$$
$$(y \oplus z) \otimes x = (y \otimes x) \oplus (z \otimes x)$$
- $\mathbf{0}$ is absorbing, also called annihilating: $x \otimes \mathbf{0} = \mathbf{0} \otimes x = \mathbf{0}$

$\boldsymbol{S}$ is a commutative semiring if $\otimes$ is commutative.

A ring is a semiring where $\forall x$ has an additive inverse $-x$.

A field is a commutative ring where $\forall x \neq \mathbf{0}$ has a multiplicative inverse x^{-1}.

Examples

$\mathbb{B} = (\{0,1\}, \vee, \wedge, 0, 1)$ Booleans

$(\mathbb{R}, +, \cdot, 0, 1)$

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

$(\mathbb{R}^{n\times n}, +, \cdot, \mathbf{0}_{n\times n}, \boldsymbol{I}_n)$ Matrices

$\mathbb{T} = ([0,\infty], \min, +, \infty, 0)$
Tropical Semiring

$(2^\Omega, \cup, \cap, \emptyset, \Omega)$
Subsets of Ω

$(\mathbb{R}[x], +, \cdot, 0, 1)$ Polynomials

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

Discussion

- Semirings belong to *Algebra*, with monoids, groups, rings, fields.
- Most semirings of interest to us are not rings, e.g. $\mathbb{B}$ or $\mathbb{N}$.
- We will only consider commutative semirings, $x \otimes y = y \otimes x$.
- We often write $+, \cdot$ instead of $\oplus, \otimes$

  E.g. $x^2y + 3z$ means $x \otimes x \otimes y \oplus (\mathbf{1} \oplus \mathbf{1} \oplus \mathbf{1}) \otimes z$

K-Relations

Overview

A standard relation associates to each tuple a Boolean value: 0 or 1.

A K-relation associates to each tuple a value from a semiring K.

By choosing different semirings, we can support different applications.

K-Relations

Fix an infinite domain Dom and a semiring $\boldsymbol{K} = (K, \oplus, \otimes, \mathbf{0}, \mathbf{1})$.

Definition ([Green et al., 2007])

A K-relation of arity m is a function $R : \text{Dom}^m \to K$ with "finite support": $\texttt{Supp}(R) \stackrel{\text{def}}{=} \{t \in \text{Dom}^m \mid R(t) \neq \mathbf{0}\}$ is finite.

K-Relations

Fix an infinite domain Dom and a semiring $\boldsymbol{K} = (K, \oplus, \otimes, \mathbf{0}, \mathbf{1})$.

Definition ([Green et al., 2007])

A K-relation of arity m is a function $R : \text{Dom}^m \to K$ with "finite support": $\text{Supp}(R) \stackrel{\text{def}}{=} \{t \in \text{Dom}^m \mid R(t) \neq \mathbf{0}\}$ is finite.

A $\mathbb{B}$-relation:

Name	City	
Alice	SF	1
Alice	NYC	0
Bob	Seattle	1

Set semantics:

2 tuples

K-Relations

Fix an infinite domain Dom and a semiring $\boldsymbol{K} = (K, \oplus, \otimes, \mathbf{0}, \mathbf{1})$.

Definition ([Green et al., 2007])

A K-relation of arity m is a function $R : \text{Dom}^m \to K$ with "finite support": $\text{Supp}(R) \stackrel{\text{def}}{=} \{t \in \text{Dom}^m \mid R(t) \neq \mathbf{0}\}$ is finite.

A $\mathbb{B}$-relation:

Name	City	
Alice	SF	1
Alice	NYC	0
Bob	Seattle	1

Set semantics:
2 tuples

A $\mathbb{N}$-relation:

Name	City	
Alice	SF	5
Alice	NYC	0
Bob	Seattle	3

Bag semantics:
8 tuples

K-Relations

Fix an infinite domain Dom and a semiring $\boldsymbol{K} = (K, \oplus, \otimes, \mathbf{0}, \mathbf{1})$.

Definition ([Green et al., 2007])

A K-relation of arity m is a function $R : \text{Dom}^m \to K$ with "finite support": $\text{Supp}(R) \stackrel{\text{def}}{=} \{t \in \text{Dom}^m \mid R(t) \neq \mathbf{0}\}$ is finite.

A $\mathbb{B}$-relation:

Name	City	
Alice	SF	1
Alice	NYC	0
Bob	Seattle	1

Set semantics:
2 tuples

A $\mathbb{N}$-relation:

Name	City	
Alice	SF	5
Alice	NYC	0
Bob	Seattle	3

Bag semantics:
8 tuples

An $\mathbb{R}$-relation:

Name	City	
Alice	SF	-0.5
Alice	NYC	0.1
Bob	Seattle	3.4

A tensor

Query Evaluation

A query Q with inputs $R_1, R_2, \ldots$ returns some output $Q(R_1, R_2, \ldots)$.

What if $R_1, R_2, \ldots$ are K-relations over some fixed semiring K?

We can define the output $Q(R_1, R_2, \ldots)$ when inputs are K-relation.

Basic principle: $\wedge$ becomes $\otimes$ and $\vee$ becomes $\oplus$.

We will do it in two ways: for Positive Relational Algebra, and UCQs

Semantics Using Positive Relational Algebra

We consider only the positive RA: $\bowtie, \sigma, \Pi, \cup$.

Their definition over K-relations is as follows:

Semantics Using Positive Relational Algebra

We consider only the positive RA: $\bowtie, \sigma, \Pi, \cup$.

Their definition over K-relations is as follows:

$(R \bowtie S)(t) \stackrel{\text{def}}{=}$ what?

Semantics Using Positive Relational Algebra

We consider only the positive RA: $\bowtie, \sigma, \Pi, \cup$.

Their definition over K-relations is as follows:

$$(R \bowtie S)(t) \stackrel{\text{def}}{=} R(\pi_{\text{Attr}(R)}(t)) \otimes S(\pi_{\text{Attr}(S)}(t))$$

Semantics Using Positive Relational Algebra

We consider only the positive RA: $\bowtie, \sigma, \Pi, \cup$.

Their definition over K-relations is as follows:

$$(R \bowtie S)(t) \stackrel{\text{def}}{=} R(\pi_{\text{Attr}(R)}(t)) \otimes S(\pi_{\text{Attr}(S)}(t))$$

$$\sigma_p(R)(t) \stackrel{\text{def}}{=} \text{what?}$$

Semantics Using Positive Relational Algebra

We consider only the positive RA: $\bowtie, \sigma, \Pi, \cup$.

Their definition over K-relations is as follows:

$$(R \bowtie S)(t) \stackrel{\text{def}}{=} R(\pi_{\text{Attr}(R)}(t)) \otimes S(\pi_{\text{Attr}(S)}(t))$$

$$\sigma_p(R)(t) \stackrel{\text{def}}{=} \mathbf{1}_{p(t)} \otimes R(t) \qquad \text{where } \mathbf{1}_{p(t)} = \begin{cases} 1 & \text{if } p(t) \text{ is true} \\ \mathbf{0} & \text{otherwise} \end{cases}$$

Semantics Using Positive Relational Algebra

We consider only the positive RA: $\bowtie, \sigma, \Pi, \cup$.

Their definition over K-relations is as follows:

$$(R \bowtie S)(t) \stackrel{\text{def}}{=} R(\pi_{\text{Attr}(R)}(t)) \otimes S(\pi_{\text{Attr}(S)}(t))$$

$$\sigma_p(R)(t) \stackrel{\text{def}}{=} \mathbf{1}_{p(t)} \otimes R(t)$$

where $\mathbf{1}_{p(t)} = \begin{cases} 1 & \text{if } p(t) \text{ is true} \\ \mathbf{0} & \text{otherwise} \end{cases}$

$$\Pi_X(R)(t) \stackrel{\text{def}}{=} \text{what?}$$

Semantics Using Positive Relational Algebra

We consider only the positive RA: $\bowtie, \sigma, \Pi, \cup$.

Their definition over K-relations is as follows:

$$(R \bowtie S)(t) \stackrel{\text{def}}{=} R(\pi_{\text{Attr}(R)}(t)) \otimes S(\pi_{\text{Attr}(S)}(t))$$

$$\sigma_p(R)(t) \stackrel{\text{def}}{=} \mathbf{1}_{p(t)} \otimes R(t)$$

where $\mathbf{1}_{p(t)} = \begin{cases} 1 & \text{if } p(t) \text{ is true} \\ \mathbf{0} & \text{otherwise} \end{cases}$

$$\Pi_X(R)(t) \stackrel{\text{def}}{=} \bigoplus_{t' : \Pi_X(t') = t} R(t')$$

Semantics Using Positive Relational Algebra

We consider only the positive RA: $\bowtie, \sigma, \Pi, \cup$.

Their definition over K-relations is as follows:

$$(R \bowtie S)(t) \stackrel{\text{def}}{=} R(\pi_{\text{Attr}(R)}(t)) \otimes S(\pi_{\text{Attr}(S)}(t))$$

$$\sigma_p(R)(t) \stackrel{\text{def}}{=} \mathbf{1}_{p(t)} \otimes R(t) \qquad \text{where } \mathbf{1}_{p(t)} = \begin{cases} 1 & \text{if } p(t) \text{ is true} \\ \mathbf{0} & \text{otherwise} \end{cases}$$

$$\Pi_X(R)(t) \stackrel{\text{def}}{=} \bigoplus_{t' : \Pi_X(t') = t} R(t')$$

$$(R \cup S)(t) \stackrel{\text{def}}{=} R(t) \oplus S(t)$$

Examples

$$\begin{array}{|c|c|c}\hline A & B & \\\hline a_1 & b_1 & x \\ a_2 & b_1 & y \\ a_2 & b_2 & z \\\hline\end{array} \bowtie \begin{array}{|c|c|c}\hline B & C & \\\hline b_1 & c_1 & u \\ b_2 & c_2 & v \\\hline\end{array} =$$

Examples

$$\begin{array}{|c|c|}\hline A & B \\ \hline a_1 & b_1 \\ a_2 & b_1 \\ a_2 & b_2 \\ \hline \end{array}\begin{array}{c} x \\ y \\ z \end{array} \bowtie \begin{array}{|c|c|}\hline B & C \\ \hline b_1 & c_1 \\ b_2 & c_2 \\ \hline \end{array}\begin{array}{c} u \\ v \end{array} = \begin{array}{|c|c|c|}\hline A & B & C \\ \hline a_1 & b_1 & c_1 \\ a_2 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ \hline \end{array}$$

Examples

$$\begin{array}{|c|c|}\hline A & B \\ \hline a_1 & b_1 \\ a_2 & b_1 \\ a_2 & b_2 \\ \hline \end{array}\begin{array}{c} x \\ y \\ z \end{array} \bowtie \begin{array}{|c|c|}\hline B & C \\ \hline b_1 & c_1 \\ b_2 & c_2 \\ \hline \end{array}\begin{array}{c} u \\ v \end{array} = \begin{array}{|c|c|c|}\hline A & B & C \\ \hline a_1 & b_1 & c_1 \\ a_2 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ \hline \end{array}\begin{array}{c} xu \\ \\ \\ \end{array}$$

Examples

$$\begin{array}{|c|c|c}\hline A & B & \\ \hline a_1 & b_1 & x \\ a_2 & b_1 & y \\ a_2 & b_2 & z \\ \hline \end{array} \bowtie \begin{array}{|c|c|c}\hline B & C & \\ \hline b_1 & c_1 & u \\ b_2 & c_2 & v \\ \hline \end{array} = \begin{array}{|c|c|c|c}\hline A & B & C & \\ \hline a_1 & b_1 & c_1 & xu \\ a_2 & b_1 & c_1 & yu \\ a_2 & b_2 & c_2 & zv \\ \hline \end{array}$$

Examples

$$\begin{array}{|c|c|}\hline A & B \\ \hline a_1 & b_1 \\ a_2 & b_1 \\ a_2 & b_2 \\ \hline \end{array}\begin{array}{c} x \\ y \\ z \end{array} \bowtie \begin{array}{|c|c|}\hline B & C \\ \hline b_1 & c_1 \\ b_2 & c_2 \\ \hline \end{array}\begin{array}{c} u \\ v \end{array} = \begin{array}{|c|c|c|}\hline A & B & C \\ \hline a_1 & b_1 & c_1 \\ a_2 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ \hline \end{array}\begin{array}{c} xu \\ yu \\ zv \end{array}$$

$$\sigma_{A=a_2}\left(\begin{array}{|c|c|}\hline A & B \\ \hline a_1 & b_1 \\ a_2 & b_1 \\ a_2 & b_2 \\ a_3 & b_1 \\ \hline \end{array}\begin{array}{c} x \\ y \\ z \\ u \end{array}\right) = \begin{array}{|c|c|}\hline A & B \\ \hline a_1 & b_1 \\ a_2 & b_1 \\ a_2 & b_2 \\ a_3 & b_1 \\ \hline \end{array}$$

Examples

$$\begin{array}{|c|c|}\hline A & B \\ \hline a_1 & b_1 \\ a_2 & b_1 \\ a_2 & b_2 \\ \hline \end{array}\begin{array}{c} \\ x \\ y \\ z \end{array} \bowtie \begin{array}{|c|c|}\hline B & C \\ \hline b_1 & c_1 \\ b_2 & c_2 \\ \hline \end{array}\begin{array}{c} \\ u \\ v \end{array} = \begin{array}{|c|c|c|}\hline A & B & C \\ \hline a_1 & b_1 & c_1 \\ a_2 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ \hline \end{array}\begin{array}{c} \\ xu \\ yu \\ zv \end{array}$$

$$\sigma_{A=a_2}\left(\begin{array}{|c|c|}\hline A & B \\ \hline a_1 & b_1 \\ a_2 & b_1 \\ a_2 & b_2 \\ a_3 & b_1 \\ \hline \end{array}\begin{array}{c} \\ x \\ y \\ z \\ u \end{array}\right) = \begin{array}{|c|c|}\hline A & B \\ \hline a_1 & b_1 \\ a_2 & b_1 \\ a_2 & b_2 \\ a_3 & b_1 \\ \hline \end{array}\begin{array}{c} \\ x \cdot 0 \\ y \cdot 1 \\ z \cdot 1 \\ u \cdot 0 \end{array}$$

Examples

$$\begin{array}{|c|c|}\hline A & B \\ \hline a_1 & b_1 \\ a_2 & b_1 \\ a_2 & b_2 \\ \hline \end{array} \begin{array}{c} x \\ y \\ z \end{array} \bowtie \begin{array}{|c|c|}\hline B & C \\ \hline b_1 & c_1 \\ b_2 & c_2 \\ \hline \end{array} \begin{array}{c} u \\ v \end{array} = \begin{array}{|c|c|c|}\hline A & B & C \\ \hline a_1 & b_1 & c_1 \\ a_2 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ \hline \end{array} \begin{array}{c} xu \\ yu \\ zv \end{array}$$

$$\sigma_{A=a_2}\left(\begin{array}{|c|c|}\hline A & B \\ \hline a_1 & b_1 \\ a_2 & b_1 \\ a_2 & b_2 \\ a_3 & b_1 \\ \hline \end{array} \begin{array}{c} x \\ y \\ z \\ u \end{array} \right) = \begin{array}{|c|c|}\hline A & B \\ \hline a_1 & b_1 \\ a_2 & b_1 \\ a_2 & b_2 \\ a_3 & b_1 \\ \hline \end{array} \begin{array}{c} x \cdot 0 \\ y \cdot 1 \\ z \cdot 1 \\ u \cdot 0 \end{array}$$

$$\Pi_A\left(\begin{array}{|c|c|}\hline A & B \\ \hline a_1 & b_1 \\ a_2 & b_1 \\ a_2 & b_2 \\ a_2 & b_3 \\ \hline \end{array} \begin{array}{c} x \\ y \\ z \\ u \end{array} \right) = \begin{array}{|c|}\hline A \\ \hline a_1 \\ a_2 \\ \hline \end{array}$$

Examples

$$\begin{array}{|c|c|}\hline A & B \\ \hline a_1 & b_1 \\ a_2 & b_1 \\ a_2 & b_2 \\ \hline \end{array}\begin{array}{c} x \\ y \\ z \end{array} \bowtie \begin{array}{|c|c|}\hline B & C \\ \hline b_1 & c_1 \\ b_2 & c_2 \\ \hline \end{array}\begin{array}{c} u \\ v \end{array} = \begin{array}{|c|c|c|}\hline A & B & C \\ \hline a_1 & b_1 & c_1 \\ a_2 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ \hline \end{array}\begin{array}{c} xu \\ yu \\ zv \end{array}$$

$$\sigma_{A=a_2}\left(\begin{array}{|c|c|}\hline A & B \\ \hline a_1 & b_1 \\ a_2 & b_1 \\ a_2 & b_2 \\ a_3 & b_1 \\ \hline \end{array}\begin{array}{c} x \\ y \\ z \\ u \end{array}\right) = \begin{array}{|c|c|}\hline A & B \\ \hline a_1 & b_1 \\ a_2 & b_1 \\ a_2 & b_2 \\ a_3 & b_1 \\ \hline \end{array}\begin{array}{c} x \cdot 0 \\ y \cdot 1 \\ z \cdot 1 \\ u \cdot 0 \end{array}$$

$$\Pi_A\left(\begin{array}{|c|c|}\hline A & B \\ \hline a_1 & b_1 \\ a_2 & b_1 \\ a_2 & b_2 \\ a_2 & b_3 \\ \hline \end{array}\begin{array}{c} x \\ y \\ z \\ u \end{array}\right) = \begin{array}{|c|}\hline A \\ \hline a_1 \\ a_2 \\ \hline \end{array} \qquad x$$

Examples

$$\begin{array}{|c|c|}\hline A & B \\ \hline a_1 & b_1 \\ a_2 & b_1 \\ a_2 & b_2 \\ \hline \end{array}\begin{array}{c} \\ x \\ y \\ z \end{array} \bowtie \begin{array}{|c|c|}\hline B & C \\ \hline b_1 & c_1 \\ b_2 & c_2 \\ \hline \end{array}\begin{array}{c} \\ u \\ v \end{array} = \begin{array}{|c|c|c|}\hline A & B & C \\ \hline a_1 & b_1 & c_1 \\ a_2 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ \hline \end{array}\begin{array}{c} \\ xu \\ yu \\ zv \end{array}$$

$$\sigma_{A=a_2}\left(\begin{array}{|c|c|}\hline A & B \\ \hline a_1 & b_1 \\ a_2 & b_1 \\ a_2 & b_2 \\ a_3 & b_1 \\ \hline \end{array}\begin{array}{c} \\ x \\ y \\ z \\ u \end{array}\right) = \begin{array}{|c|c|}\hline A & B \\ \hline a_1 & b_1 \\ a_2 & b_1 \\ a_2 & b_2 \\ a_3 & b_1 \\ \hline \end{array}\begin{array}{c} \\ x \cdot 0 \\ y \cdot 1 \\ z \cdot 1 \\ u \cdot 0 \end{array}$$

$$\Pi_A\left(\begin{array}{|c|c|}\hline A & B \\ \hline a_1 & b_1 \\ a_2 & b_1 \\ a_2 & b_2 \\ a_2 & b_3 \\ \hline \end{array}\begin{array}{c} \\ x \\ y \\ z \\ u \end{array}\right) = \begin{array}{|c|}\hline A \\ \hline a_1 \\ a_2 \\ \hline \end{array}\begin{array}{c} \\ x \\ y+z+u \end{array}$$

Special Cases

- Suppose the semiring is that of Booleans $\mathbb{B}$. What does the positive relational algebra compute?

- Suppose the semiring is that of natural numbers $\mathbb{N}$. What does the positive relational algebra compute?

Special Cases

- Suppose the semiring is that of Booleans $\mathbb{B}$. What does the positive relational algebra compute? Standard set semantics

- Suppose the semiring is that of natural numbers $\mathbb{N}$. What does the positive relational algebra compute?

Special Cases

- Suppose the semiring is that of Booleans $\mathbb{B}$. What does the positive relational algebra compute? Standard set semantics

An *idempotent* semiring is one where $x \oplus x = x$

- Suppose the semiring is that of natural numbers $\mathbb{N}$. What does the positive relational algebra compute?

Special Cases

- Suppose the semiring is that of Booleans $\mathbb{B}$. What does the positive relational algebra compute? Standard set semantics

  An *idempotent* semiring is one where $x \oplus x = x$

- Suppose the semiring is that of natural numbers $\mathbb{N}$. What does the positive relational algebra compute? Bag semantics

  Notice that $\mathbb{N}$ is not idempotent

Semantics Using UCQs

Recall that a Conjunctive Query (CQ) is:

$$\boxed{Q(\boldsymbol{X}) = \exists \boldsymbol{Y} \left(R_1(\boldsymbol{Z}_1) \wedge R_2(\boldsymbol{Z}_2) \wedge \cdots\right)}$$

Semantics Using UCQs

Recall that a Conjunctive Query (CQ) is:

$$\boxed{Q(\boldsymbol{X}) = \exists \boldsymbol{Y}\, (R_1(\boldsymbol{Z}_1) \wedge R_2(\boldsymbol{Z}_2) \wedge \cdots)}$$

Let $\boldsymbol{V} \stackrel{\text{def}}{=} \text{Vars}(Q)$ (all variables).

If $R_1, R_2, \ldots$ are K-relations, then the semantics of Q is defined as:

Semantics Using UCQs

Recall that a Conjunctive Query (CQ) is:

$$\boxed{Q(\boldsymbol{X}) = \exists \boldsymbol{Y}\,(R_1(\boldsymbol{Z}_1) \wedge R_2(\boldsymbol{Z}_2) \wedge \cdots)}$$

Let $\boldsymbol{V} \stackrel{\text{def}}{=} \text{Vars}(Q)$ (all variables).

If $R_1, R_2, \ldots$ are K-relations, then the semantics of Q is defined as:

$$\boxed{Q(\boldsymbol{x}) \stackrel{\text{def}}{=} \bigoplus_{\boldsymbol{v} \in \text{Dom}^{\boldsymbol{V}} : \pi_{\boldsymbol{X}}(\boldsymbol{v}) = \boldsymbol{x}} R_1(\pi_{\boldsymbol{Z}_1}(\boldsymbol{v})) \otimes R_2(\pi_{\boldsymbol{Z}_2}(\boldsymbol{v})) \otimes \cdots}$$

Semantics Using UCQs

Recall that a Conjunctive Query (CQ) is:

$$\boxed{Q(\boldsymbol{X}) = \exists \boldsymbol{Y}\,(R_1(\boldsymbol{Z}_1) \wedge R_2(\boldsymbol{Z}_2) \wedge \cdots)}$$

Let $\boldsymbol{V} \stackrel{\text{def}}{=} \text{Vars}(Q)$ (all variables).

If $R_1, R_2, \ldots$ are K-relations, then the semantics of Q is defined as:

$$\boxed{Q(\boldsymbol{x}) \stackrel{\text{def}}{=} \bigoplus_{\boldsymbol{v} \in \text{Dom}^{\boldsymbol{V}} : \pi_{\boldsymbol{X}}(\boldsymbol{v}) = \boldsymbol{x}} R_1(\pi_{\boldsymbol{Z}_1}(\boldsymbol{v})) \otimes R_2(\pi_{\boldsymbol{Z}_2}(\boldsymbol{v})) \otimes \cdots}$$

The semantics of an UCQ $\boxed{Q(\boldsymbol{X}) = Q_1(\boldsymbol{X}) \cup Q_2(\boldsymbol{X}) \cup \cdots}$

is: $\boxed{Q(t) \stackrel{\text{def}}{=} Q_1(t) \oplus Q_2(t) \oplus \cdots}$

Short Comment

The semantics over K-relations is simple!

$\boxed{\text{Replace } \vee, \wedge \text{ with } \oplus, \otimes}$

Sparse Tensors

$\mathbb{R}$-relations are logically equivalent to sparse tensors.

Sparse Tensors

$\mathbb{R}$-relations are logically equivalent to sparse tensors.

A sparse matrix:

$$M = \begin{pmatrix} 9 & 0 & 0 \\ 0 & 0 & 7 \\ 1.1 & -5 & 0 \end{pmatrix}$$

Representation as an $\mathbb{R}$-relation:

X	Y	
1	1	9
2	3	7
3	1	1.1
3	2	-5

Einstein Summations and CQs

An Einstein summation is the same as a CQ interpreted over $\mathbb{R}$-relations.

Einstein Summations and CQs

An Einstein summation is the same as a CQ interpreted over $\mathbb{R}$-relations.

CQ:

$$\boxed{Q(X, Z) = \exists Y(A(X, Y) \wedge B(Y, Z))}$$

Einstein Summations and CQs

An Einstein summation is the same as a CQ interpreted over $\mathbb{R}$-relations.

CQ:

$$\boxed{Q(X, Z) = \exists Y (A(X, Y) \wedge B(Y, Z))}$$

Einstein Summation:

$$\boxed{Q[i, k] = \sum_j A[i, j] \cdot B[j, k]}$$

Einsums[1]

Einsums "drop the quantifiers": $Q(X,Z) = A(X,Y) \wedge B(Y,Z)$.

Transpose: $B[i,j] = A[i,j]$

Summation: $S = A[i,j]$

Row sum: $R[i] = A[i,j]$

Dot product: $P = A[i] * B[i]$

Outer product $T[i,j] = A[i] * B[j]$

Batch matrix multiplication: $C[i,k,m] = A[i,j,m] * B[j,k,m]$

[1] `https://rockt.github.io/2018/04/30/einsum`

Einsums[1]

Einsums "drop the quantifiers": $Q(X, Z) = A(X, Y) \wedge B(Y, Z)$.

Transpose: $B[i, j] = A[i, j]$

Summation: $S = A[i, j]$

Row sum: $R[i] = A[i, j]$

Dot product: $P = A[i] * B[i]$

Outer product $T[i, j] = A[i] * B[j]$

Batch matrix multiplication: $C[i, k, m] = A[i, j, m] * B[j, k, m]$

[1] `https://rockt.github.io/2018/04/30/einsum`

Einsums[1]

Einsums "drop the quantifiers": $Q(X, Z) = A(X, Y) \wedge B(Y, Z)$.

Transpose: $B[i, j] = A[i, j]$

Summation: $S = A[i, j]$

Row sum: $R[i] = A[i, j]$

Dot product: $P = A[i] * B[i]$

Outer product $T[i, j] = A[i] * B[j]$

Batch matrix multiplication: $C[i, k, m] = A[i, j, m] * B[j, k, m]$

[1] https://rockt.github.io/2018/04/30/einsum

Einsums[1]

Einsums "drop the quantifiers": $Q(X, Z) = A(X, Y) \wedge B(Y, Z)$.

Transpose: $B[i, j] = A[i, j]$

Summation: $S = A[i, j]$

Row sum: $R[i] = A[i, j]$

Dot product: $P = A[i] * B[i]$

Outer product $T[i, j] = A[i] * B[j]$

Batch matrix multiplication: $C[i, k, m] = A[i, j, m] * B[j, k, m]$

[1] `https://rockt.github.io/2018/04/30/einsum`

Einsums[1]

Einsums "drop the quantifiers": $Q(X, Z) = A(X, Y) \wedge B(Y, Z)$.

Transpose: $B[i, j] = A[i, j]$

Summation: $S = A[i, j]$

Row sum: $R[i] = A[i, j]$

Dot product: $P = A[i] * B[i]$

Outer product $T[i, j] = A[i] * B[j]$

Batch matrix multiplication: $C[i, k, m] = A[i, j, m] * B[j, k, m]$

[1] https://rockt.github.io/2018/04/30/einsum

Einsums[1]

Einsums "drop the quantifiers": $Q(X, Z) = A(X, Y) \wedge B(Y, Z)$.

Transpose: $B[i, j] = A[i, j]$

Summation: $S = A[i, j]$

Row sum: $R[i] = A[i, j]$

Dot product: $P = A[i] * B[i]$

Outer product $T[i, j] = A[i] * B[j]$

Batch matrix multiplication: $C[i, k, m] = A[i, j, m] * B[j, k, m]$

[1] https://rockt.github.io/2018/04/30/einsum

Einsums[1]

Einsums "drop the quantifiers": $Q(X,Z) = A(X,Y) \wedge B(Y,Z)$.

Transpose: $B[i,j] = A[i,j]$

Summation: $S = A[i,j]$

Row sum: $R[i] = A[i,j]$

Dot product: $P = A[i] * B[i]$

Outer product $T[i,j] = A[i] * B[j]$

Batch matrix multiplication: $C[i,k,m] = A[i,j,m] * B[j,k,m]$

[1] `https://rockt.github.io/2018/04/30/einsum`

Access Control

- Discretionary Access Control: read/write/etc permissions for each user/resource pair.

- Mandatory Access Control: clearance levels. Secret, Top Secret, etc.

Mandatory Access Control

The access control semiring: $\boxed{(\mathbb{A}, \min, \max, 0, P)}$

$\mathbb{A} = \{\text{Public} < \text{Confidential} < \text{Secret} < \text{Top-secret} < 0\}$ 0 "No Such Thing"

Pics

PID	
p1	S
p2	T

Occ

PID	DID	
p1	d1	P
p2	d1	P
p2	d2	P

Docs

DID	
d1	C
d2	0

$$\boxed{Q(p) = \text{Pics}(p) \wedge \text{Occ}(p, d) \wedge \text{Docs}(d)}$$

Mandatory Access Control

The access control semiring: $\boxed{(\mathbb{A}, \min, \max, 0, P)}$

$\mathbb{A} = \{\text{Public} < \text{Confidential} < \text{Secret} < \text{Top-secret} < 0\}$ 0 "No Such Thing"

Pics

PID	
p1	S
p2	T

Occ

PID	DID	
p1	d1	P
p2	d1	P
p2	d2	P

Docs

DID	
d1	C
d2	0

Answer

PID
p1
p2

$$\boxed{Q(p) = \text{Pics}(p) \wedge \text{Occ}(p, d) \wedge \text{Docs}(d)}$$

What are the annotations of the output tuples?

Mandatory Access Control

The access control semiring: $\boxed{(\mathbb{A}, \min, \max, 0, P)}$

$\mathbb{A} = \{\text{Public} < \text{Confidential} < \text{Secret} < \text{Top-secret} < 0\}$ 0 "No Such Thing"

Pics

PID	
p1	S
p2	T

Occ

PID	DID	
p1	d1	P
p2	d1	P
p2	d2	P

Docs

DID	
d1	C
d2	0

Answer

PID	
p1	S
p2	T

$$\boxed{Q(p) = \text{Pics}(p) \wedge \text{Occ}(p, d) \wedge \text{Docs}(d)}$$

What are the annotations of the output tuples?

Discussion

- K-Relations: powerful abstraction that allows us to apply concepts from the relational model to other domains
- Einsum notation popular in ML: numpy, TensorFlow, pytorch
  Note slight variation in syntax. (Read the manual!)
- The original motivation of K-relations in [Green et al., 2007] was to model *provenance*. Will discuss next.

Provenance Polynomials

Overview

Run a query over the input data. Look at one output tuple t.

Where does t come from?

Provenance, or lineage, aims to define some formalism to answer this question.

Many variants were proposed in the literature before K-relations, with an unclear winner.

K-relations proved to be able to capture them all, in an elegant framework.

Provenance Polynomials

Fix a standard database instance $\boldsymbol{D} = (R_1^D, R_2^D, \ldots)$.

Annotate each tuple with a distinct tag $x_1, x_2, \ldots$; abstract tagging.

Consider the semiring of polynomials $\mathbb{N}[\boldsymbol{x}] = \mathbb{N}[x_1, x_2, \ldots]$

Each relation R_i^D becomes an $\mathbb{N}[\boldsymbol{x}]$-relation.

Compute the query Q over the these $\mathbb{N}[\boldsymbol{x}]$-relations.

Output tuples annotated with polynomials: provenance polynomials.

Example

From [Green et al., 2007]

A	B	C	
a	b	c	x
d	b	e	y
f	g	e	z

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

$Q(A, C) =$
$\exists A_1 B_1 C_1 (R(A, B_1, C_1) \wedge R(A_1, B_1, C))$
$\vee \exists A_1 B_1 B_2 (R(A, B_1, C) \wedge R(A_1, B_2, C))$

Example

From [Green et al., 2007]

A	B	C	
a	b	c	x
d	b	e	y
f	g	e	z

A	C	
a	c	$2x^2$
a	e	xy
d	c	xy
d	e	$2y^2 + yz$
f	e	$2z^2 + yz$

$Q(A, C) =$
$\exists A_1 B_1 C_1 (R(A, B_1, C_1) \wedge R(A_1, B_1, C))$
$\vee \exists A_1 B_1 B_2 (R(A, B_1, C) \wedge R(A_1, B_2, C))$

Example

From [Green et al., 2007]

A	B	C	
a	b	c	x
d	b	e	y
f	g	e	z

$Q(A, C) =$
$\exists A_1 B_1 C_1 (R(A, B_1, C_1) \wedge R(A_1, B_1, C))$
$\vee \exists A_1 B_1 B_2 (R(A, B_1, C) \wedge R(A_1, B_2, C))$

A	C	
a	c	$2x^2$
a	e	xy
d	c	xy
d	e	$2y^2 + yz$
f	e	$2z^2 + yz$

Interpretation:

- (a, e) is derived from x and y.

Example

From [Green et al., 2007]

A	B	C	
a	b	c	x
d	b	e	y
f	g	e	z

$Q(A, C) =$
$\exists A_1 B_1 C_1 (R(A, B_1, C_1) \wedge R(A_1, B_1, C))$
$\vee \exists A_1 B_1 B_2 (R(A, B_1, C) \wedge R(A_1, B_2, C))$

A	C	
a	c	$2x^2$
a	e	xy
d	c	xy
d	e	$2y^2 + yz$
f	e	$2z^2 + yz$

Interpretation:

- (a, e) is derived from x and y.
- (a, c) is derived in two ways: using x twice, and using x twice.
- (d, e) is derived ...

Other Notions of Provenance

Many variations on the following themes:

- Do we distinguish between conjunction and disjunction?
  Do $R \cup R$ and $R \cap R$ have the same provenance?
- Do we require idempotence?
  Does $R \cup R$ have the same provenance as $R \cup R \cup R$?
- Do we require multiplicative idempotence?
  Does $R \cap R$ have the same provenance as R?

More informative

$$
\begin{tikzcd}
 & & \begin{array}{c}\mathbb{N}[X] \\ 2x^2y + xy + 5y^2 + xz\end{array} \arrow[dl, "\downarrow\text{coeff.}"', "\text{idemp.}"] \arrow[dr, "\downarrow\text{exp.}"] & \\
 & \begin{array}{c}\mathbb{B}[X] \\ x^2y + xy + y^2 + xz\end{array} \arrow[dl, "\text{absorb}"] \arrow[dr, "\downarrow\text{exp.}"] & & \begin{array}{c}\mathit{Trio}(X) \\ 3xy + 5y + xz\end{array} \arrow[dl, "\downarrow\text{coeff.}"', "\text{idemp.}"] \\
\begin{array}{c}\mathit{Sorp}(X) \\ xy + y^2 + xz\end{array} \arrow[dr, "\downarrow\text{exp.}"] & & \begin{array}{c}\mathit{Why}(X) \\ xy + y + xz\end{array} \arrow[dl, "\text{absorb}"] \arrow[dr, "+ = \cdot"] & \\
 & \begin{array}{c}\mathit{PosBool}(X) \\ y + xz\end{array} & & \begin{array}{c}\mathit{Which}(X) \\ xyz\end{array}
\end{tikzcd}
$$

Less informative

Discussion

- Fine-grained provenance: complete information on how a tuple was produced.
  - Provenance polynomials are fine-grained

- Coarse-grained provenance: data science pipelines
  - What input files where used? What versions? When were they collected?
  - What tools were used in the pipeline? What version? What (hyper-)parameter settings?
  - When was the pipeline executed? On what OS, what configuration?

Optimization Rules

Review: The Algebraic Laws of Relational Algebra

There is no finite axiomatization of the Relational Algebra why?

But there is a finite axiomatization of Positive Relational Algebra why?

Examples:

$$(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$$
$$(R \cup S) \bowtie T = R \bowtie T \cup S \bowtie T$$
$$\sigma_p(R \bowtie S) = \sigma_p(R) \bowtie S$$
$$\ldots$$

What are the Algebraic Laws over K-relations?

Homomorphisms

A homomorphism $f : (S, \oplus, \otimes, \mathbf{0}, \mathbf{1}) \to (K, +, \cdot, 0, 1)$ is a function $f : S \to K$ such that:

$$f(\mathbf{0}) = 0 \qquad f(\mathbf{1}) = 1$$

$$f(x \oplus y) = f(x) + f(y) \qquad f(x \otimes y) = f(x) \cdot f(y)$$

Universality Property

Theorem

Fix a set $\boldsymbol{x} = \{x_1, x_2, \ldots\}$*. The semiring* $(\mathbb{N}[\boldsymbol{x}], +, \cdot, 0, 1)$ *is the freely generated commutative semiring.*

$$\begin{tikzcd} X \arrow[r] \arrow[dr, "\forall f"'] & \mathbb{N}[X] \arrow[d, "\exists! f^* \ (\text{hom})"] \\ & K \end{tikzcd}$$

Applications to Query Optimization

Corollary

Consider an identity in semirings $E_1 = E_2$. The following are equivalent:

1. $E_1 = E_2$ *holds in* $(\mathbb{N}, +, \cdot, 0, 1)$.
2. $E_1 = E_2$ *holds in* $(\mathbb{N}[\boldsymbol{x}], +, \cdot, 0, 1)$.
3. $E_1 = E_2$ *holds in all commutative semirings.*

Proof (in class) Item 1 $\Rightarrow$ Item 2 $\Rightarrow$ Item 3 $\Rightarrow$ Item 1

Example:
$(x + y)(x + z)(y + z) = xy(x + y) + xz(x + z) + yz(y + z) + 2xyz$

Applications for Query Optimization

Consider an identity $E_1 = E_2$ in the Positive Relational Algebra $(\bowtie, \sigma, \Pi, \cup)$.

The following are equivalent:

- $E_1 = E_2$ holds under bag semantics.
- $E_1 = E_2$ holds for all K-relations, i.e. for any semiring K.

Example $R \bowtie (S \cup T) = (R \bowtie S) \cup (R \bowtie T)$.

What about set semantics? Do we have more identities? Fewer identities? Give examples!

Discussion

- Semirings and K-relations significantly expand the scope of the relational data model to a rich set of applications.

- Cost-based query optimizers designed for SQL could, in theory, be deployed in several other domains. E.g. sparse tensor processing.

Green, T. J., Karvounarakis, G., and Tannen, V. (2007).
Provenance semirings.
In Libkin, L., editor, *Proceedings of the Twenty-Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 11-13, 2007, Beijing, China*, pages 31–40. ACM.