HADAS
HETEROGENEOUS AUTONOMOUS DISTRIBUTED DATA SERVICES

Provenance for database transformations
or
an algebraic view of annotated data

Val Tannen et al : "Provenance Semirings"
[GreenKarvounarakis&T PODS 07]

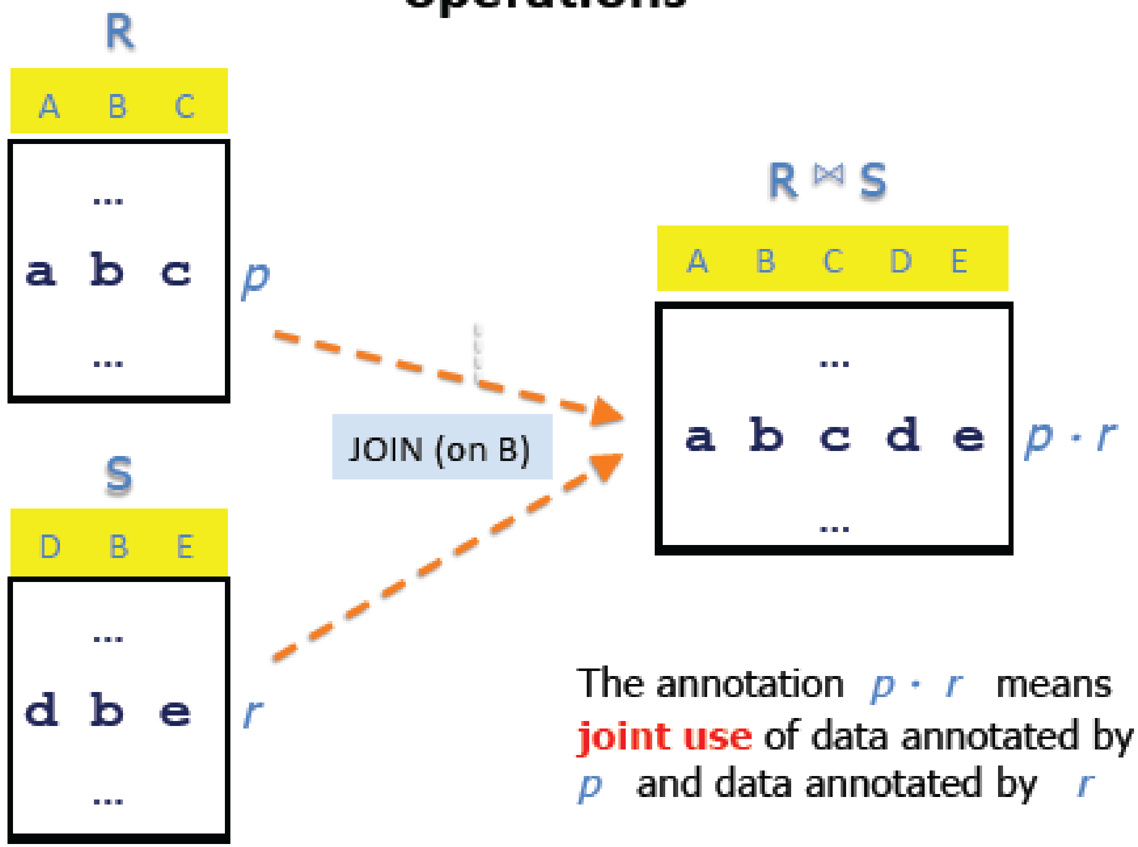
LIG

Motivation

- **Annotations capture:**
  - PROVENANCE
  - UNCERTAINTY
  - TRUST
  - SECURITY
  - MULTIPLICITY
- **Semirings bring:**
  - Operations for annotation propagation in a uniform view and with good properties

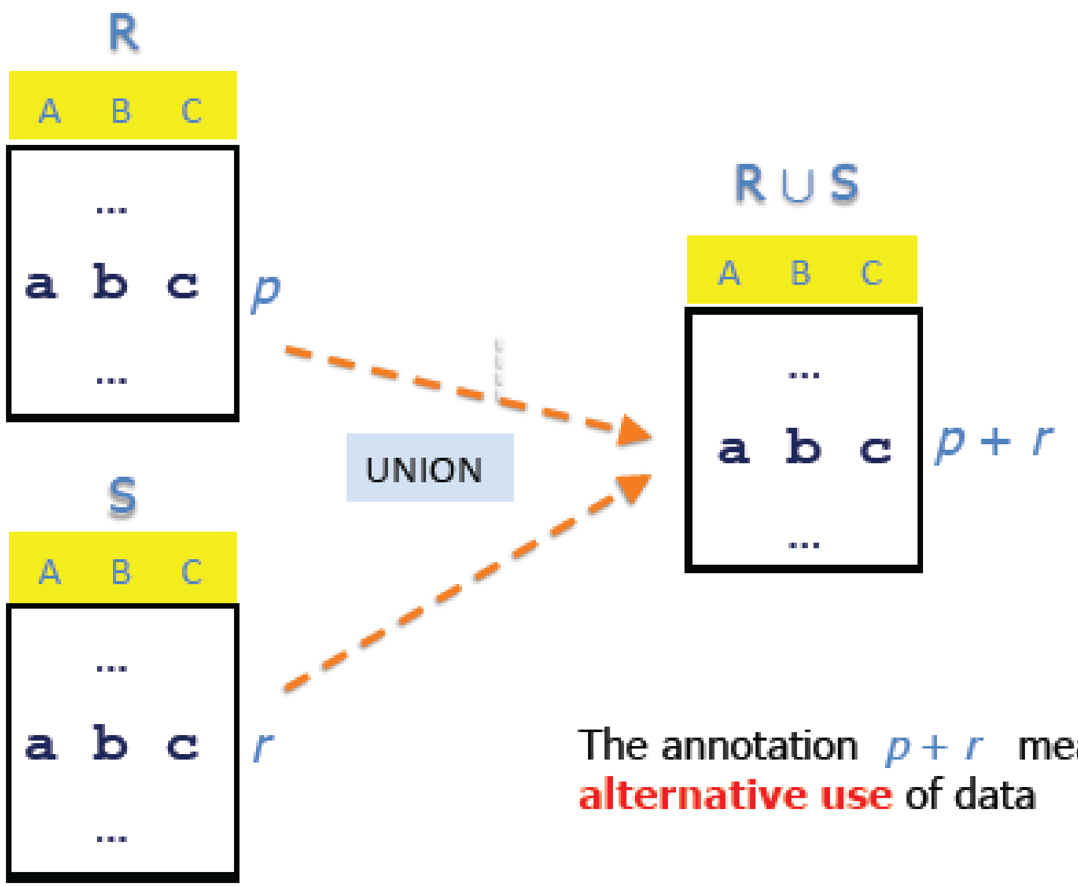
HADAS
HETEROGENEOUS AUTONOMOUS DISTRIBUTED DATA SERVICES

Propagating annotations through database operations



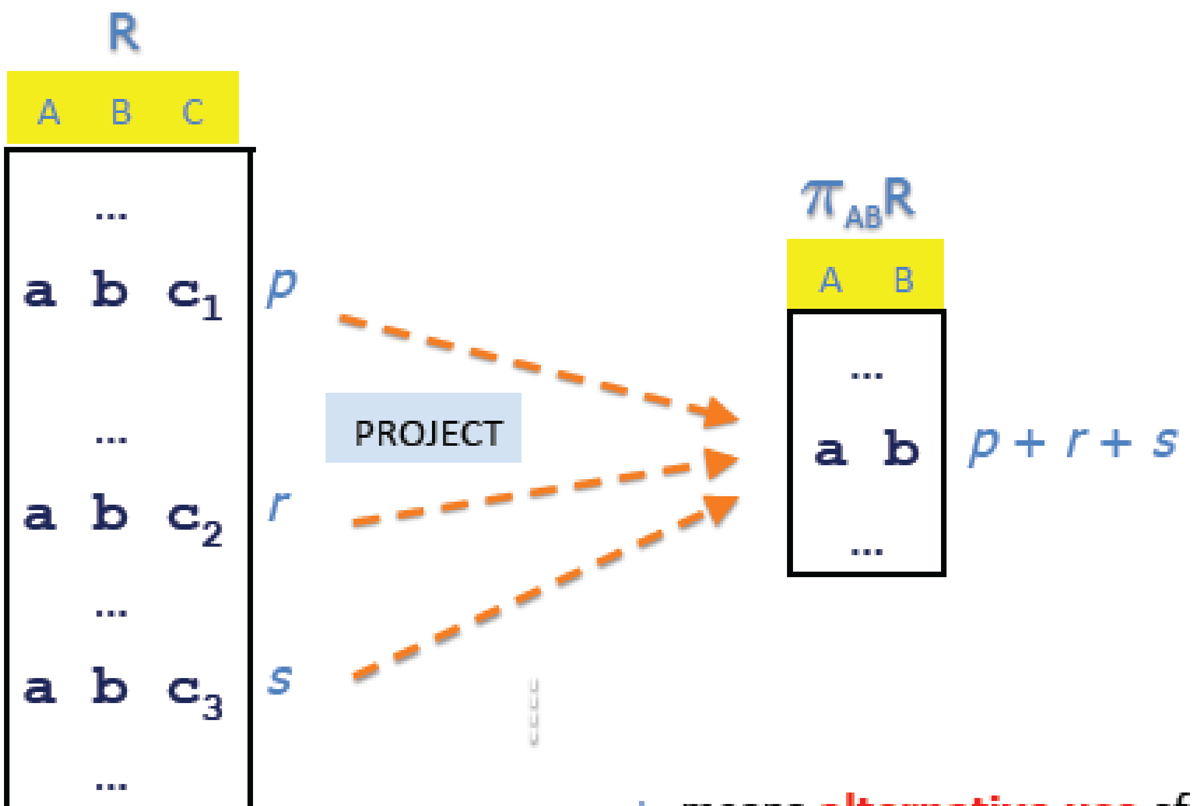
The annotation $p \cdot r$ means **joint use** of data annotated by p and data annotated by r

Another way to propagate annotations



The annotation $p + r$ means **alternative use** of data

Another use of +



+ means **alternative use** of data

An example in positive relational algebra (SPJU)

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

R

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

An example in positive relational algebra (SPJU)

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

R

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

A	C	
a	c	$(p \cdot p + p \cdot p) \cdot 0$
a	e	$p \cdot r \cdot 1$
d	c	$r \cdot p \cdot 0$
d	e	$(r \cdot r + r \cdot s + r \cdot r) \cdot 1$
f	e	$(s \cdot s + s \cdot r + s \cdot s) \cdot 1$

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

Formalization

A space of annotations, K

K-**relations**: every tuple annotated with some element from K.

Binary operations on K: $\cdot$ corresponds to joint use (join), and $+$ corresponds to alternative use (union and projection).

We assume K contains special annotations 0 and 1.

"Absent" tuples are annotated with 0!

1 is a "neutral" annotation (no restrictions).

Algebra of annotations? What are the **laws** of $(K, +, \cdot, 0, 1)$?

Required laws on K

- **Equivalent queries should produce the same annotations**
  - Union is associative and commutative
  - Join is associative, commutative and distributes over union
  - Projection and selection commute with each other and with union and join (when applicable)
- **Theorem:**
  - Above identities holds for queries on K-relations iff (K, +, ., 1,0) is a commutative semiring

What is a commutative semiring?

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

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

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

Unlike ring, no requirement for inverses to $+$

Using the laws: polynomials

R

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

Q

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

Polynomials with coefficients in $\mathbb{N}$ and
annotation tokens as indeterminates p, r, s
capture a very general form of **provenance**

Provenance reading of the polynomials

R

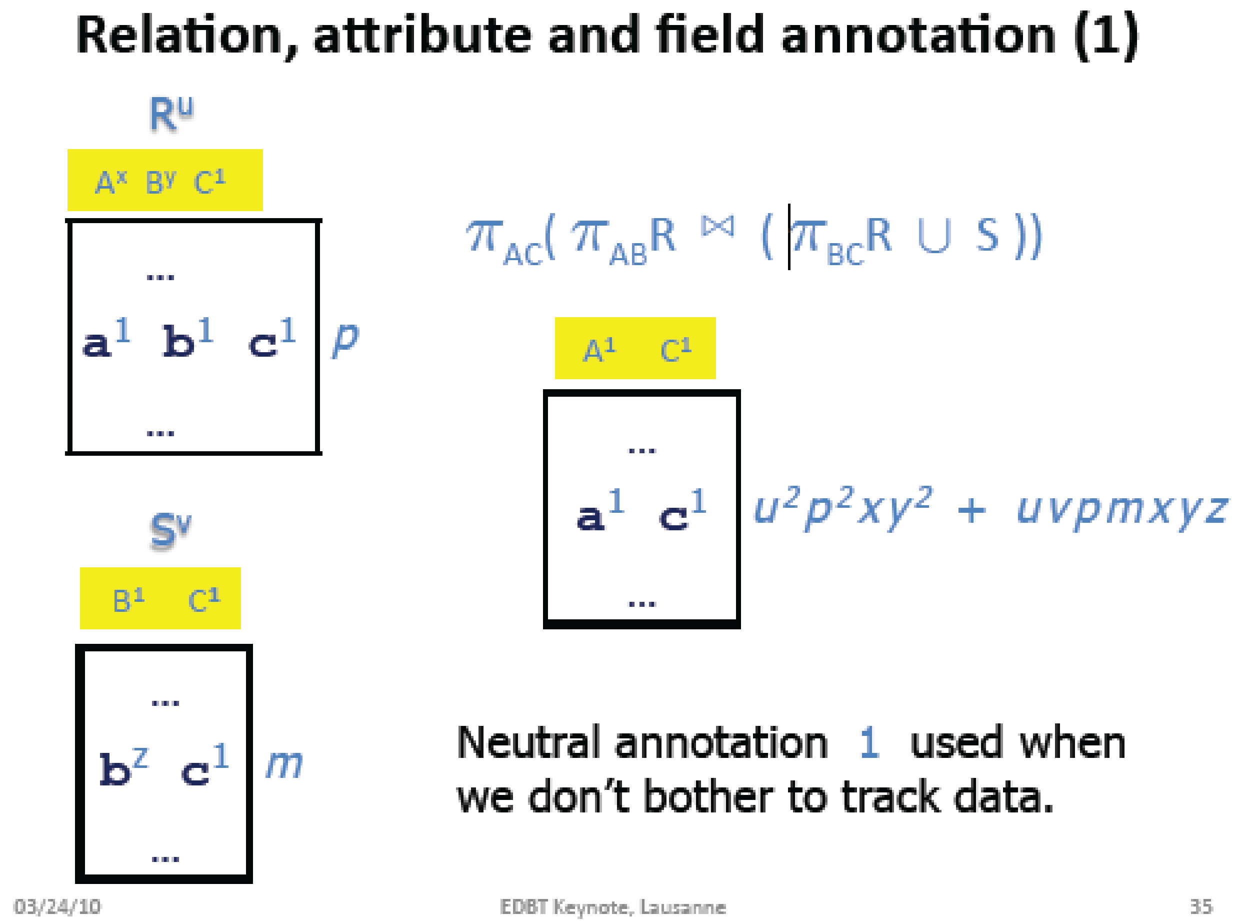
A	B	C	
a	b	c	p
d	b	e	r
f	g	e	s

Q

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

- three different ways to derive **d e**
- two of the ways use only r
- but they use it twice
- the third way uses r once and s once

BEYOND TUPLE ANNOTATIONS



Homomorphisms

- **Useful to evaluate the polynomials**
  - by mapping X to K and extending it to an homomorphism from N[X] to K :
    - h(p1 +p2) = h(p1) + h(p2) h(p1.p2)= h(p1).h(p2)
    - h(1) = 1 h(0) = 0
  - to obtain:
    - trust scores
    - multiplicity
    - uncertainty values
    - access control levels
- **Useful to hide detail and increase abstraction**
  - by mapping provenance tokens, many to few
  - Stop tracking tokens by mapping them to 1 (neutral)

Doesn't always work, eg. difference.

Fundamental property

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

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

$$q(h(R)) = h(q(R))$$

(Direct) application to multiplicity: bag semantics

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

$\xrightarrow{q}$

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

h (with p = 2, r=5, s =1)

A B C	
$a\ b\ c$	2
$d\ b\ e$	5
$f\ g\ e$	1

eval with p = 2, r=5, s =1

A C	
$a\ c$	$2 \cdot 2 + 2 \cdot 2 = 8$
$a\ e$	$2 \cdot 5 = 10$
$d\ c$	$2 \cdot 5 = 10$
$d\ e$	$5 \cdot 5 + 5 \cdot 5 + 5 \cdot 1 = 55$
$f\ e$	$1 \cdot 1 + 1 \cdot 1 + 5 \cdot 1 = 7$

Application to c-tables: boolean conditions used as annotations for modeling incomplete databases

$$(\mathsf{BoolExp}(X), \wedge, \vee, \top, \bot)$$

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

q

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

h (with p = b1, r=b2, s =b3)

eval
with p = b1, r=b2, s =b3
(using $\vee$ for ‘+’ and $\wedge$ for ‘. ’)

A B C	
$a\ b\ c$	b_1
$d\ b\ e$	b_2
$f\ g\ e$	b_3

q

A C	
$a\ c$	$(b_1 \wedge b_1) \vee (b_1 \wedge b_1)$
$a\ e$	$b_1 \wedge b_2$
$d\ c$	$b_1 \wedge b_2$
$d\ e$	$(b_2 \wedge b_2) \vee (b_2 \wedge b_2) \vee (b_2 \wedge b_3)$
$f\ e$	$(b_3 \wedge b_3) \vee (b_3 \wedge b_3) \vee (b_2 \wedge b_3)$

A C	
$a\ c$	b_1
$a\ e$	$b_1 \wedge b_2$
$d\ c$	$b_1 \wedge b_2$
$d\ e$	b_2
$f\ e$	b_3

Application to event tables: probabilistic databases

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

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

q

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

h (with p = X, r=Y, s = Z)

eval
with p = X, r=Y, s =Z
(using $\cup$ for ‘+’ and $\cap$ for ‘. ’)

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

q

A C	
$a\ c$	X
$a\ e$	$X \cap Y$
$d\ c$	$X \cap Y$
$d\ e$	Y
$f\ e$	Z

E	Pr
X	0.6
Y	0.5
Z	0.1

Application to access control

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

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

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

$\xrightarrow{q}$

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

Fundamental property implies that applying the clearance to the database or to the query answer yields the same result. (But only the second is actually feasible!)

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

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

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

$\xrightarrow{q}$

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

"User with secret clearance"

Conclusion

- Provenance is the basis for :
  - UNCERTAINTY
  - TRUST
  - SECURITY
  - MULTIPLICITY
- The semiring (N[X], +, ., 0, 1) is the algebraic foundation for computing provenance over queries
- Homomorphisms are the tools for evaluating provenance in different settings

HADAS
HETEROGENEOUS AUTONOMOUS DISTRIBUTED DATA SERVICES