Techniques for managing probabilistic data

Dan Suciu
University of Washington

Databases Are Deterministic

- Applications since 1970's required precise semantics
  - Accounting, inventory
- Database tools are deterministic
  - A tuple is an answer or is not
- Underlying theory assumes determinism
  - FO (First Order Logic)

Future of Data Management

We need to cope with uncertainties !

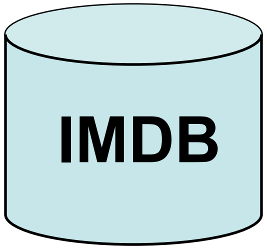
- Represent uncertainties as probabilities
- Extend data management tools to handle probabilistic data

Major paradigm shift affecting both foundations and systems

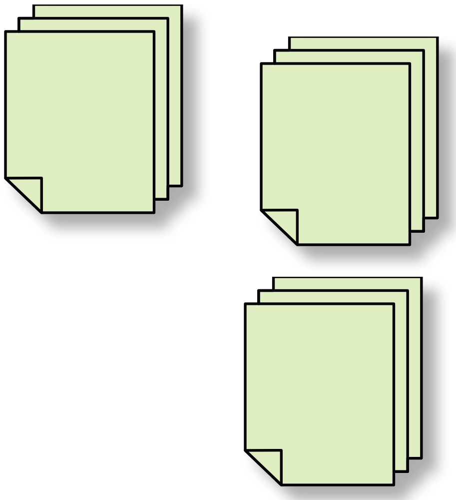
Example: Alice Looks for Movies

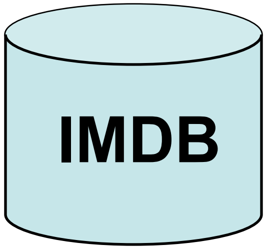
IMDB:

- Lots of data !
- Well maintained and clean
- But no reviews!



On the web there
are lots of reviews…

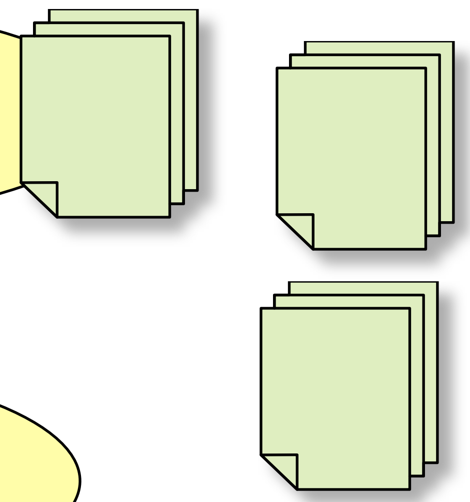


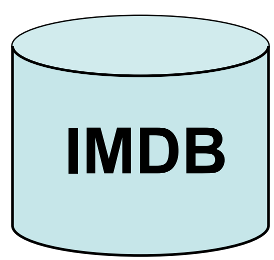


How do I know…
…which movie they talk about?

…if the review is
positive or negative ?

…if I should trust
the reviewer ?





Alice needs:
- fuzzy joins
- information extraction
- sentiment analysis
- social networks

Find actors in Pulp Fiction who appeared in two bad movies five years earlier

Find years when 'Anthony Hopkins' starred in a good movie

IMDB

A **probabilistic database** can help Alice store and query her uncertain data

Application 1: Using Fuzzy Joins

IMDB

titles don’t match

Reviews

Title	Year
Twelve Monkeys	1995
Monkey Love 1997	1997
Monkey Love 1935	1935
Monkey Love Panet	2005

Review	By	Rating
12 Monkeys	Joe	4
Monkey Boy	Jim	2
Monkey Love	Joe	2

Result of a Fuzzy Join

[Arasu'2006]

TitleReviewMatch^p

Title	Review	P
Twelve Monkeys	12 Monkeys	0.7
Monkey Love 1997	12 Monkeys	0.45
Monkey Love 1935	Monkey Love	0.82
Monkey Love 1935	Monkey Boy	0.68
Monkey Love Planet	Monkey Love	0.8

Queries over Fuzzy Joins

IMDB

Title	Year
Twelve Monkeys	1995
Monkey Love 97	1997
Monkey Love 35	1935
Monkey Love PL	2005

TitleReviewMatch^p

Title	Review	P
Twelve Monkeys	12 Monkeys	0.7
Monkey Love 97	12 Monkeys	0.45
Monkey Love 35	Monkey Love	0.82
Monkey Love 35	Monkey Boy	0.68
Monkey Love Planet	Monkey Love	0.8

Reviews

Review	By	Rating
12 Monkeys	Joe	4
Monkey Boy	Jim	2
Monkey Love	Joe	2

Who reviewed movies made in 1935 ?

```
SELECT DISTINCT z.By
FROM IMDB x, TitleReviewMatchp y, Amazon z
WHERE x.title=y.title and x.year=1935 and y.review=z.review
```

Ranked !

Answer:

By	P
Joe	0.73
Fred	0.68
Jim	0.43
. . .	0.12

Find movies reviewed by Jim and Joe

```
SELECT DISTINCT x.Title
FROM IMDB x, TitleReviewMatchp y1, Amazon z1,
             TitleReviewMatchp y2, Amazon z2
WHERE . . .z1.By='Joe' . . . . z2.By='Jim' . . .
```

Answer:

Title	P
Gone with…	0.73
Amadeus	0.68
. . .	0.43

[Gupta&Sarawagi'2006]

Application 2: Information Extraction

...52 A Goregaon West Mumbai ...

Address^p

ID	House-No	Street	City	P
1	52	Goregaon West	Mumbai	0.1
1	52-A	Goregaon West	Mumbai	0.4
1	52	Goregaon	West Mumbai	0.2
1	52-A	Goregaon	West Mumbai	0.2
2	. . . .	. . . .	. . . .	. . . .
2	. . . .			

≈20% of such extractions are correct

Here probabilities are meaningful

Queries

Find people living in ‘West Mumbai’

```
SELECT DISTINCT x.name
FROM Person x, Address^p y
WHERE x.ID = y.ID and y.city = ‘West Mumbai’
```

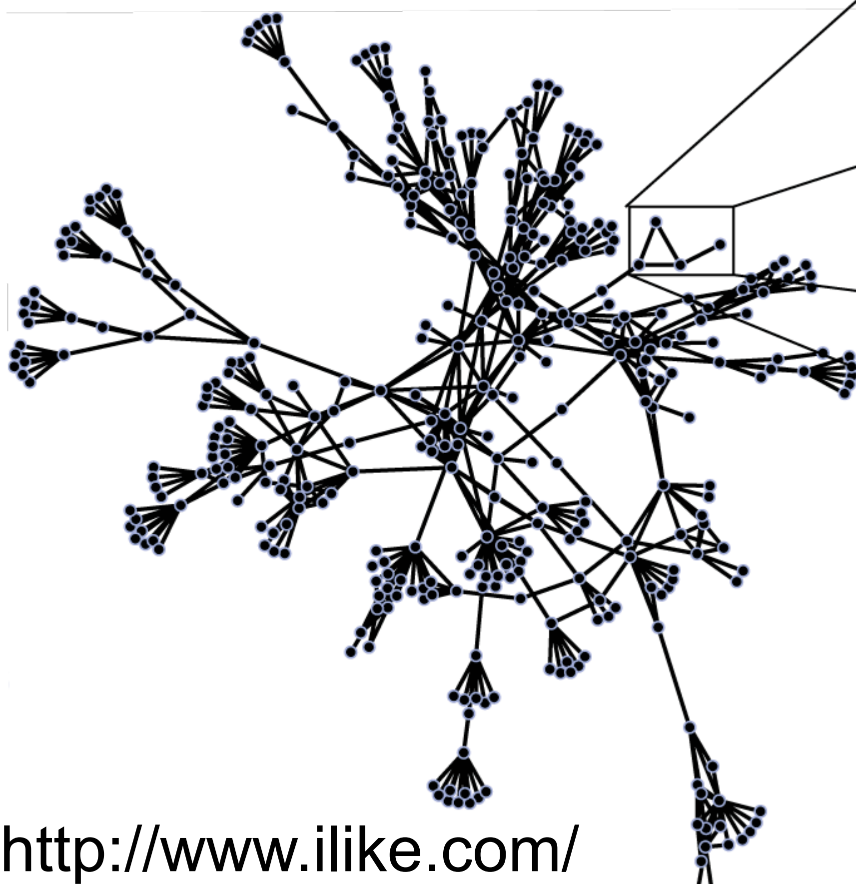
Find people of the same age, living in the same city

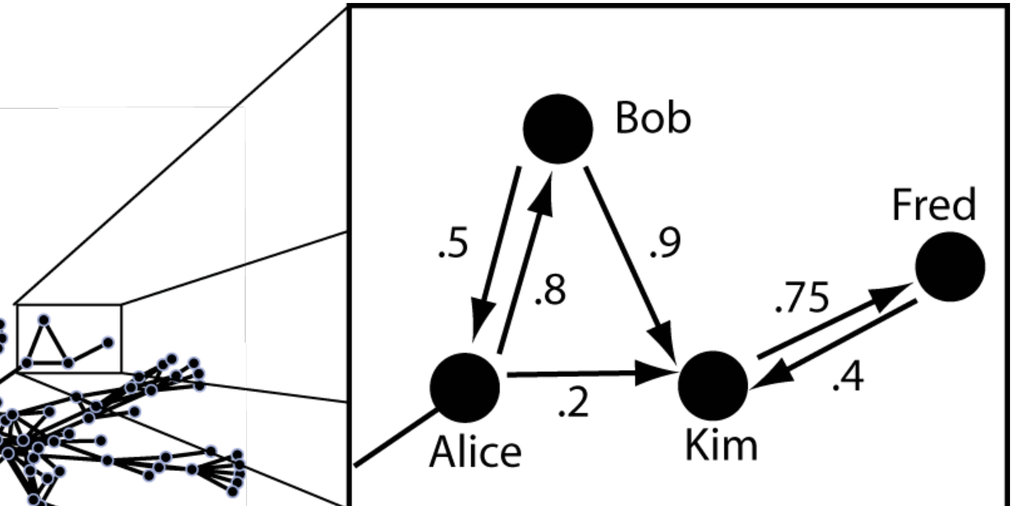
```
SELECT DISTINCT x.name, u.name
FROM Person x, Address^p y, Person u, Address^p v
WHERE x.ID = y.ID and y.city = v.city and u.ID = v.ID
```

Today’s practice is to retain only the most likely extraction; this results in low recall for these queries.
A probabilistic database keeps all extractions: higher recall.

[Adar&Re'2007]

Application 3: Social Networks





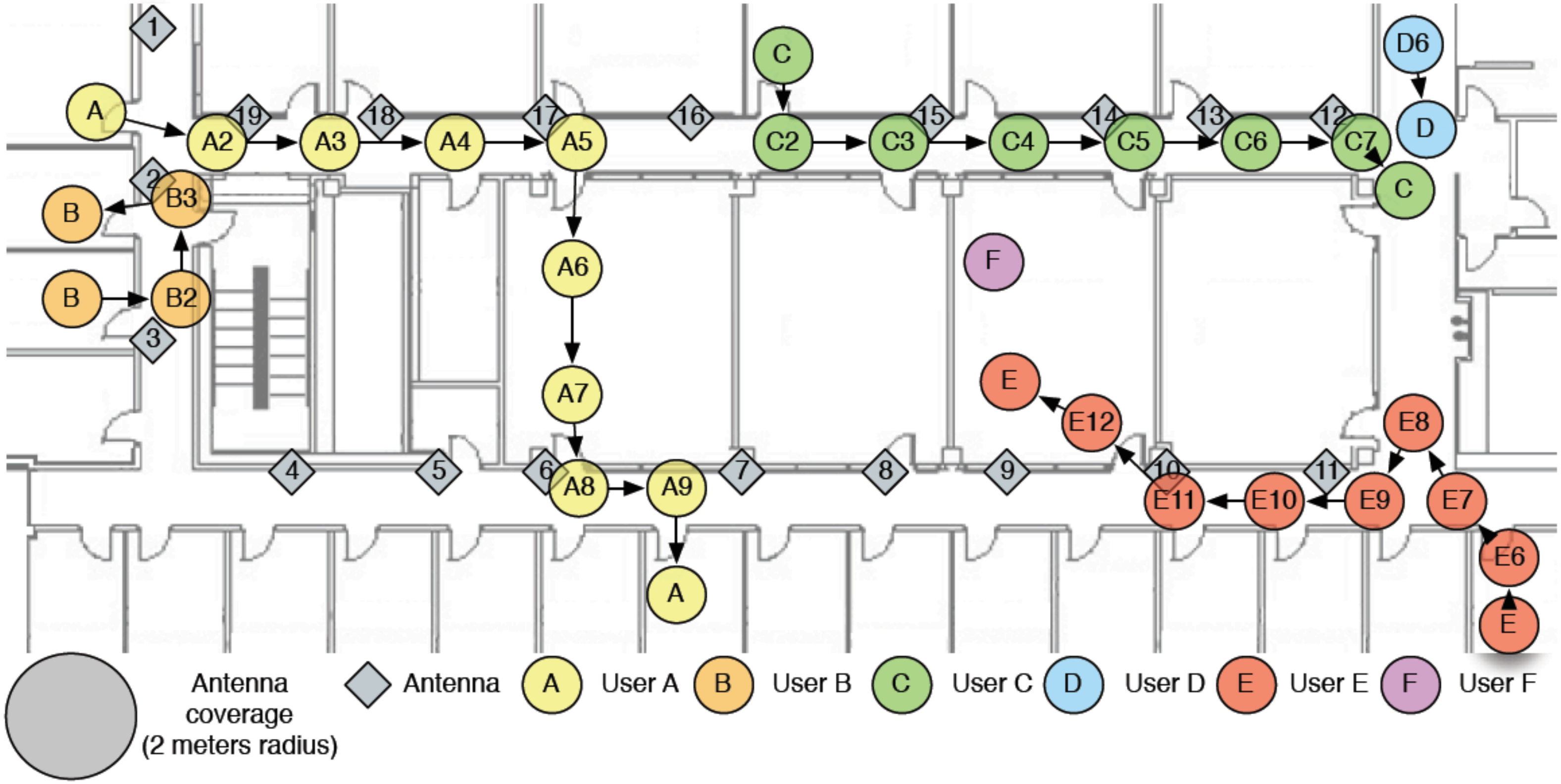
Name1	Name2	P
Alice	Bob	0.5
Alice	Kim	0.2
Bob	Kim	0.9
Bob	Alice	0.5
Kim	Fred	0.75
Fred	Kim	0.4

Name	Age	City
Alice	25	Rome
Fred	21	Venice
Bob	30	Rome
Kim	27	Milan

http://www.ilike.com/

Give 50 free tickets to most influential people in Venice

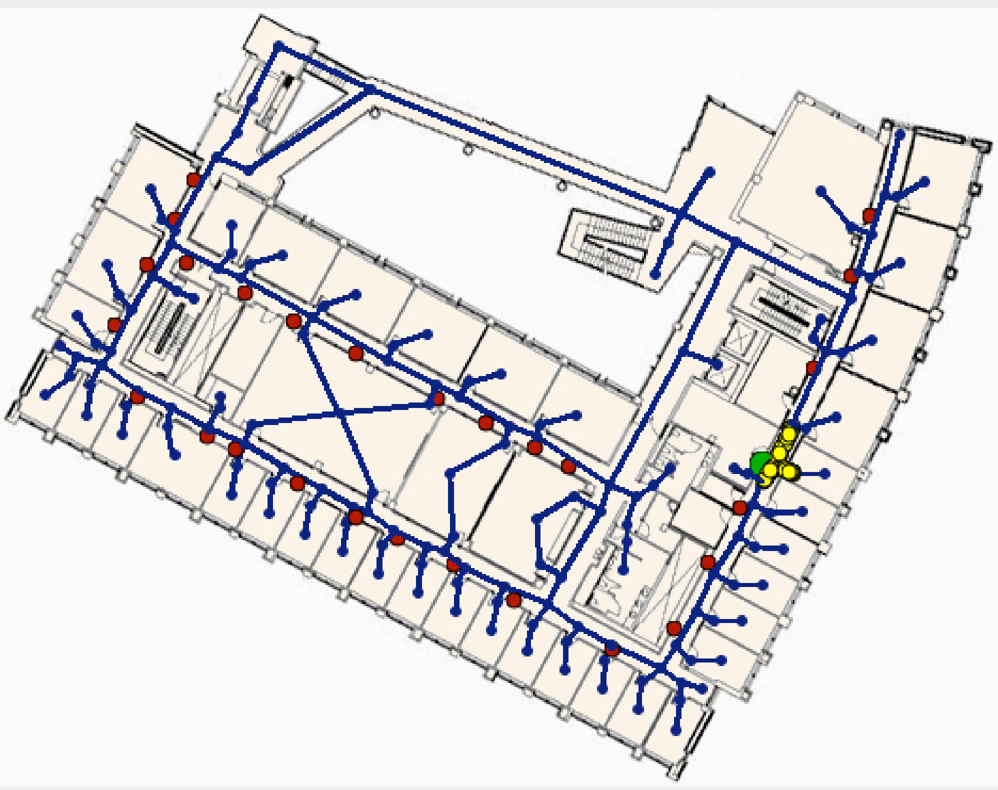
Application 4: RFID Data



RFID Ecosystem at the UW

[Welbourne'2007]

RFID Data



Particle filter with 100 particles

Courtesy of Julie Letchner

Time	Person	Location	P
1	Jim	L54	0.1
		L39	0.4
		L44	0.2
		L10	0.3
2	Jim	L54	0.3
		L12	0.6
		L10	0.1
3	Jim	L12	0.4
		L54	0.6

RFID Data

- Raw data is noisy:
  - SIGHTING(tagID, antennaID, time)

- Derived data = Probabilistic
  - “John is located at L32 at 9:15” prob=0.6
  - “John carried laptop x77 at 11:03” prob=0.8
  - . . .
- Queries
  - “Which people were in Room 478 yesterday ?”

RFID Data = Massive, streaming, **probabilistic**

A Model for Uncertainties

- Data is probabilistic
- Queries formulated in a standard language
- Answers are annotated with probabilities

This tutorial: Managing Probabilistic Data

Long History

Cavallo&Pitarelli:1987

Barbara,Garcia-Molina, Porter:1992

Lakshmanan,Leone,Ross&Subrahmanian:1997

Fuhr&Roellke:1997

Dalvi&S:2004

Widom:2005

Modern Probabilistic DBMS

- Trio at Stanford [Widom et al.]
  - Uncertainty and Lineage ULDB
- MystiQ at the University of Washington [S. et al.]
  - Query evaluation, optimization
- University of Maryland [Getoor, Desphande et al.]
  - Complex probabilistic models, PRMS
- Orion at Purdue University [Prabhakar et al.]
  - Sensor data, continuous random variables
- Data Furnace at Berkeley [Garofalakis, Franklin, Hellerstein]

Focus today: Query Evaluation/Optimization

Has this been solved by AI ?

Input: KB

Fix q
Input: DB

	AI	**Databases**
Deterministic	Theorem prover	Query processing
Probabilistic	Probabilistic inference	[this tutorial]

No: *probabilistic inference* notoriously expensive

Outline

Part 1:

- Motivation
- Data model
- Basic query evaluation

Part 2:

- The dichotomy of query evaluation
- Implementation and optimization
- Six Challenges

[Barbara et al.1992]

What is a Probabilistic Database (PDB) ?

Keys

Non-keys

Probability

HasObject^p

Object	**Time**	Person	P
Laptop77	9:07	John	0.62
		Jim	0.34
Book302	9:18	Mary	0.45
		John	0.33
		Fred	0.11

What does it *mean* ?

Background

Finite probability space = (Ω, P)

$\Omega = \{\omega_1, \ldots, \omega_n\}$ = set of outcomes

$P : \Omega \to [0,1]$

$P(\omega_1) + \ldots + P(\omega_n) = 1$

Event: $E \subseteq \Omega$, $P(E) = \sum_{\omega \in E} P(\omega)$

"*Independent*": $P(E_1 E_2) = P(E_1)\, P(E_2)$

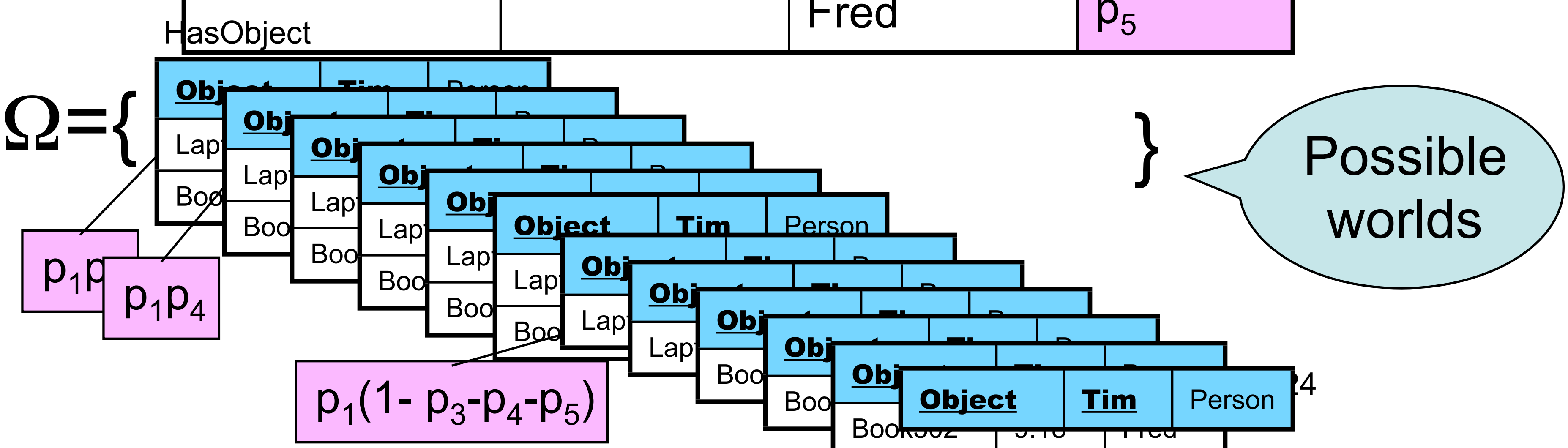
"*Mutual exclusive*" or "*disjoint*": $P(E_1 E_2) = 0$

Possible Worlds Semantics

HasObject^p

Object	**Time**	Person	P
Laptop77	9:07	John	p_1
		Jim	p_2
Book302	9:18	Mary	p_3
		John	p_4
		Fred	p_5

PDB



Representation of a Probabilistic Database

- Impossible to enumerate all worlds !
- Need concise *representation formalism*
- Here we discuss two simple formalisms:
  - Independent tuples
  - Independent/disjoint tuples
- They are *incomplete*
- They become complete by adding *views*

Definition: A tuple-independent table is:

R^p(**A1**, **A2**, …, **Am**, P)

$Meets^p$(**Person1**, **Person2**, **Time**, P)

Person1	**Person2**	**Time**	P
John	Jim	9.12	p_1
Mary	Sue	9:20	p_2
John	Mary	9:20	p_3

Independent tuples

Terminology: Trio calls each such a tuple a *maybe tuple*: it may be in, or it may not be in.

<u>Definition</u>: A tuple-disjoint/independent table is:

$R^p($**<u>A1</u>**, **<u>A2</u>**, …, **<u>Am</u>**, B1, …, Bn, P$)$

HasObject^p(**<u>Object</u>**, **<u>Time</u>**, Person, P)

<u>Object</u>	**<u>Time</u>**	Person	P
Laptop77	9:07	John	p_1
		Jim	p_2
Book302	9:18	Mary	p_3
		John	p_4
		Fred	p_5

Disjoint

Disjoint

Inde-
pen-
dent

<u>Terminology</u>: Disjoint tuples are also called *exclusive*. Trio calls them *x-tuples*.

Two Approaches to Queries

This tutorial

- Standard queries, probabilistic answers
  - Query: “find all movies with rating > 4”
  - Answers: list of tuples with probabilities

- Novel types of queries
  - Query: find all Movie-review matches with probability in [0.3, 0.8]
  - Answer: …

Open research direction (not well studied in literature)

Queries in Datalog Notation

```
SELECT DISTINCT m.year
FROM Movie m, Review r
WHERE m.id = r.mid
    and r.rating > 3
```

SQL

$q(y)$:- $Movie^p(\underline{\mathbf{x}},\underline{\mathbf{y}})$, $Review^p(\underline{\mathbf{x}},\underline{\mathbf{z}})$, $z>3$

Conjunctive query
(datalog)

Semantics 1: Possible Tuples

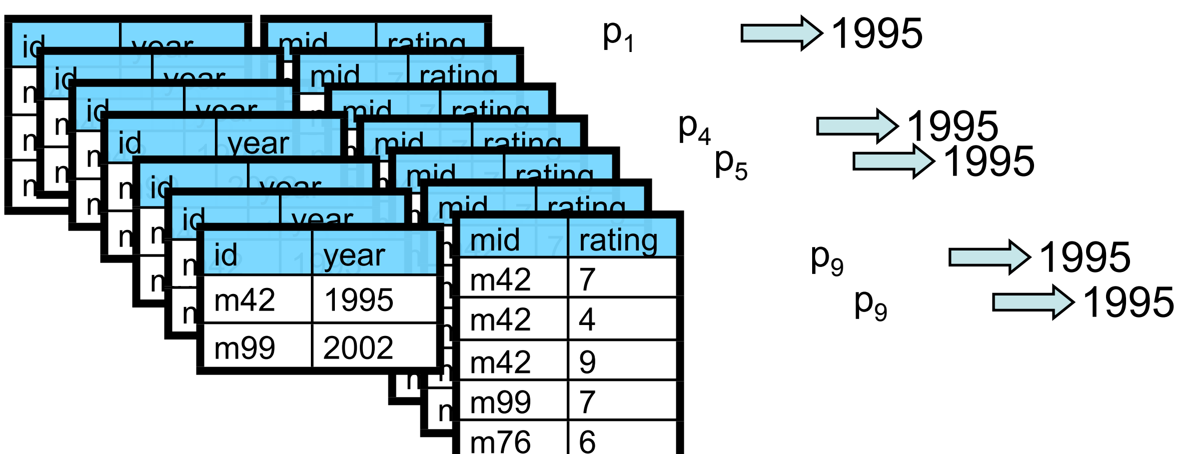
Movie^p

id	year	P
m42	1995	0.6
m99	2002	0.8
m76	2002	0.3

Review^p

mid	rating	P
m42	7	0.5
m42	4	0.3
m42	9	0.9
m99	7	0.6
m99	5	0.2
m76	6	0.3

$q(y) \text{ :- } Movie^p(\underline{\mathbf{x}},\underline{\mathbf{y}}),\ Review^p(\underline{\mathbf{x}},\underline{\mathbf{z}}),\ z>3$



Answer

year	P
1995	$p_1+p_4+p_5+p_8+p_9$
2002	$p_3+p_4+p_7$

Formal Definition

Query q tuple a probability space (Ω, P)

➡ Boolean query $q(a)$

➡ Probabilistic event: $E = \{\omega \mid \omega \models q(a)\}$

<u>Definition</u> $P(q(a)) = P(E) = \sum_{\omega \models q(a)} P(\omega)$

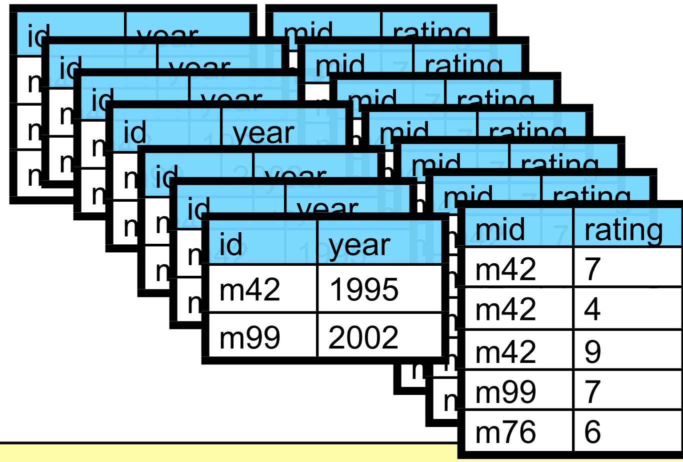
Example $q(y)$:- $Movie^p(\underline{\mathbf{x}},\underline{\mathbf{y}})$, $Review^p(\underline{\mathbf{x}},\underline{\mathbf{z}})$, $z>3$ 1995

$q(1995)$:- $Movie^p(\underline{\mathbf{x}},1995)$, $Review^p(\underline{\mathbf{x}},\underline{\mathbf{z}})$, $z>3$

$P(q(1995))$ = marginal probability of $q(1995)$

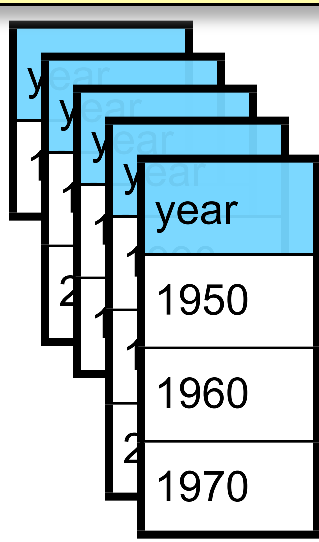
Semantics 2: Possible Answers

Possible worlds



q(y) :- Movie^p(<u>**x**</u>,<u>**y**</u>), Review^p(<u>**x**</u>,<u>**z**</u>), z>3

Possible answers



p_1

p_2

p_3

. . .

Formal Definition

View $\boxed{v}$, Probability space $\boxed{(\Omega, P)}$

➡ New probability space $\boxed{(\Omega', P')}$

Definition $\Omega' = \{\omega' \mid \exists\ \omega \in \Omega,\ v(\omega) = \omega'\}$

$$P'(\omega') = \sum_{\omega\ :\ v(\omega)=\omega'} P(\omega)$$

"Image probability space" [Green&Tannen'06]

Query Semantics

- Possible tuples:
  - Simple, intuitive user interface
  - Query evaluation is probabilistic inference
  - But is not compositional
- Possible answers:
  - Is compositional
  - Open research problems: user interface, query evaluation

Best for expressing user queries

Best for defining views

Complex Models = Simple + Views

Example adapted from [Gupta&Sarawagi'2006]

Address^p

ID	House-No	Street	City	P
1	52	Goregaon West	Mumbai	0.06
1	52-A	Goregaon West	Mumbai	0.15
1	52	Goregaon	West Mumbai	0.12
1	52-A	Goregaon	West Mumbai	0.3
2	. . . . .	. . . . .	. . . . .	. . . . .
2	. . . . .			

Suppose House-no extracted independently from Street and City

$Address^p$

<u>ID</u>	House-No	Street	City	P
1	52	Goregaon West	Mumbai	0.06
1	52-A	Goregaon West	Mumbai	0.15
1	52	Goregaon	West Mumbai	0.12
1	52-A	Goregaon	West Mumbai	0.3
2	. . . .	. . . .	. . . .	. . . .

$AddrH^p$

<u>ID</u>	House-No	P
1	52	0.2
1	52-A	0.5
2	. . . .	. . . .

$AddrSC^p$

<u>ID</u>	Street	City	P
1	Goregaon West	Mumbai	0.3
1	Goregaon	West Mumbai	0.6
2	. . . .	. . . .	. . . .

View: Address(x,y,z,u) :- AddrH(x,y), AddrSC(x,z,u)

Complex Models = Simple + Views

Standard query rewriting:

View: Address(x,y,z,u) :- AddrH(x,y), AddrSC(x,z,u)

User query: q(x) :- Address(x,y,z,'West Mumbai')

↓

Rewritten query

q(x) :- AddrH(x,y), AddrSC(x,z,'West Mumbai')

Complex Models = Simple + Views

- In this simple example the view is already representable as a tuple
disjoint/independent table

- In general views can define more complex probability spaces over possible worlds, that are not disjoint/indepdendent

<u>Theorem</u> [Dalvi&S'2007]
Independent/disjoint tables + conjunctive views = a complete representation system

Discussion of Data Model

Tuple-disjoint/independent tables:

- Simple model, can store in any DBMS

More advanced models:

- Symbolic boolean expressions
- Trio: add lineage
- Probabilistic Relational Models
- Graphical models

Fuhr and Roellke

[Widom05, Das Sarma'06, Benjelloun 06]

[Getoor'2006]

[Sen&Desphande'07]

Outline

Part 1:

- Motivation
- Data model
- Basic query evaluation

Part 2:

- The dichotomy of query evaluation
- Implementation and optimization
- Six Challenges

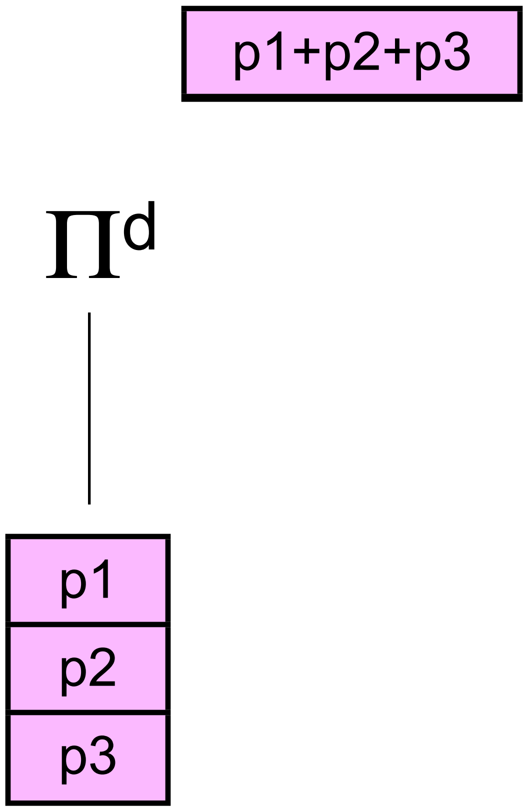
Extensional Operators

Object	Person	Location	P
Laptop77	John	L45	p1
	Jim	L45	p2
	Jim	L66	p3
Book302	Mary	L66	p4
	Mary	L45	p5
	Jim	L66	p6
	John	L45	p7
	Fred	L45	p8

q(z) :- HasObject^p(**Book302**, y, z)

Location	P
L66	p4+p6
L45	p5+p7+p8

Disjoint Project



Extensional Operators

Object	Person	Location	P
Laptop77	John	L45	p1
	Jim	L45	p2
	Jim	L66	p3
Book302	Mary	L66	p4
	Mary	L45	p5
	Jim	L66	p6
	John	L45	p7
	Fred	L45	

$$q(y,z) \text{ :- } HasObject^p(\underline{\mathbf{x}},y,z)$$

Person	Location	P
Jim	L66	1-(1-p3)(1-p6)
John	L45	1-(1-p1)(1-p7)
. . .		

Independent Project

1-(1-p1)(1-p2)(1-p3)

Π^i

p1
p2
p3

$q(y)$:- $Movie^p(\underline{\mathbf{x}},\underline{\mathbf{y}})$, $Review^p(\underline{\mathbf{x}},\underline{\mathbf{z}})$, $z>3$

A Taste of Query Evaluation

Movie

id	year	P
m42	1995	p1
m99	2002	p2
m76	2002	p3

Review

mid	rating	P
m42	7	q1
m42	4	q2
m42	9	q3
m99	7	q4
m99	5	q5
m76	6	q6

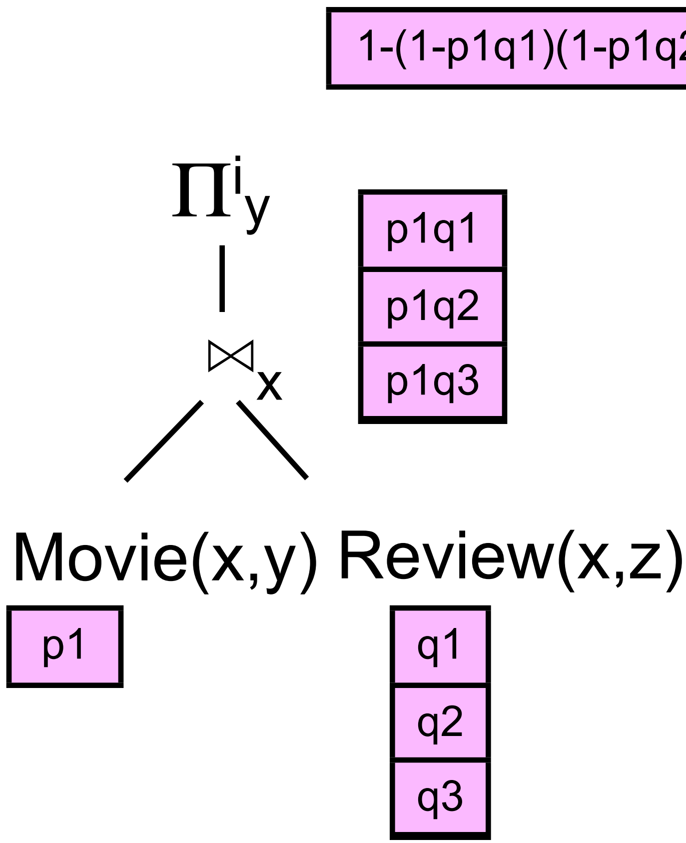
Answer

year	P
1995	p1 × (1 - (1 - q1)×(1 - q2)×(1 - q3))
2002	1 - (1 - p2 × (1 - (1 - q4)×(1 - q5))) × (1 - p3 × q6)

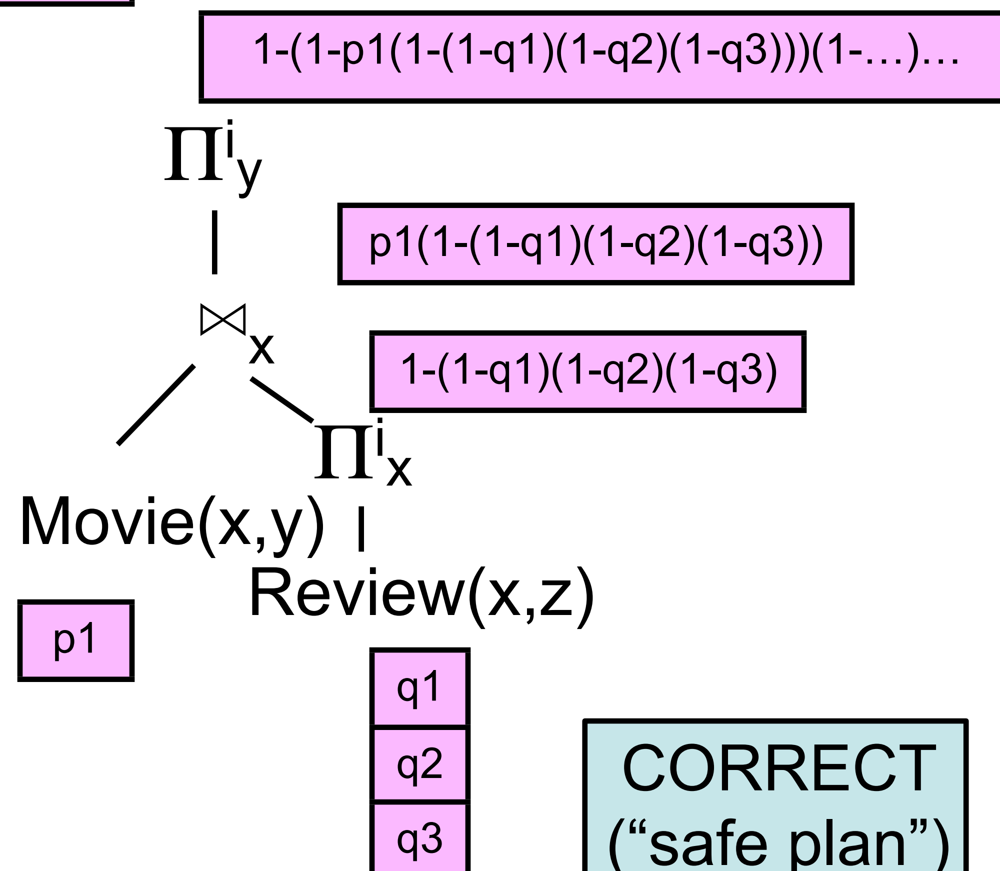
$q(y) \text{ :- } Movie^p(\underline{\mathbf{x}},\underline{\mathbf{y}}), Review^p(\underline{\mathbf{x}},\underline{\mathbf{z}})$

q(1995)

Answer depends on query plan !



INCORRECT



CORRECT
(“safe plan”)

Safe Plans are Efficient

- Very efficient: run almost as fast as regular queries
- Require only simple modifications of the relational operators
- Or can be translated back into SQL and sent to any RDBMS

Can we always generate a safe plan ?

A Hard Query

R^p

A	B	P
a	x1	p1
a	x2	p2

S

B	C
x1	y1
x1	y2
x2	y1

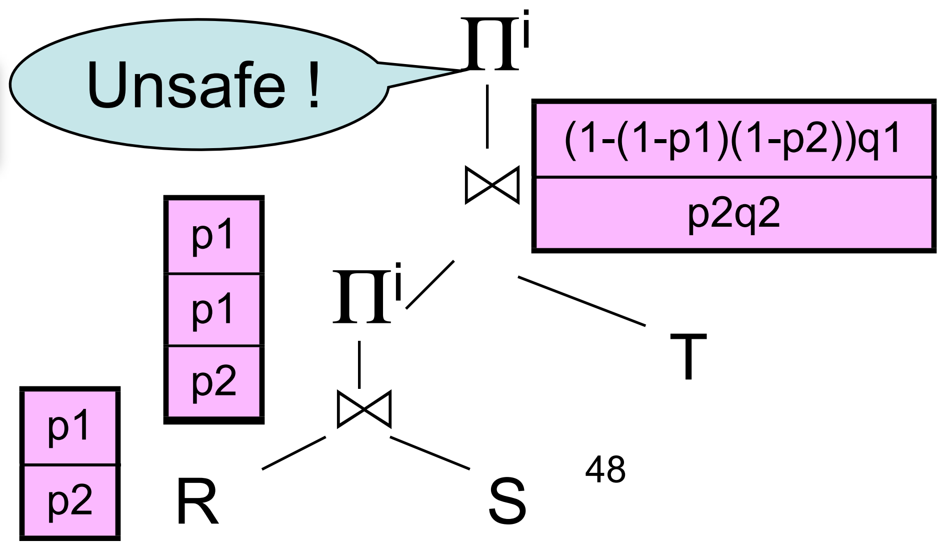
T^p

C	D	P
y1	c	q1
y2	c	q2

h(u,v) :- R^p(**u**,**x**),S(**x**,**y**),T^p(**y**,**v**)

h(a,c)

There is no safe plan !



Independent Queries

Let q1, q2 be two boolean queries

<u>Definition</u> q1, q2 are “independent” if $P(q1, q2) = P(q1)\,P(q2)$

Also: $P(q1 \vee q2) = 1 - (1 - P(q1))(1 - P(q2))$

Quiz: which are independent ?

q1	q2	Indep.?
Movie^p(**m41**,**y**)	Review^p(**m41**, **z**)	
Movie^p(**m42**,**y**),Review^p(**m42**,**z**)	Movie^p(**m77**,**y**),Review^p(**m77**,**z**)	
Movie^p(**m42**,**y**),Review^p(**m42**,**z**)	Movie^p(**m42**, **1995**)	
Movie^p(**m42**,**y**),Review^p(**m42**,**7**)	Movie^p(**m42**,**y**),Review^p(**m42**,**4**)	
R^p(**x**,**y**,**z**,**z**,**u**), R^p(**x**,**x**,**x**,**y**,**y**)	R^p(**a**,**a**,**b**,**b**,**c**)	

Answers

q1	q2	Indep.?
Movie^p(**m41**,**y**)	Review^p(**m41**, **z**)	YES
Movie^p(**m42**,**y**),Review^p(**m42**,**z**)	Movie^p(**m77**,**y**),Review^p(**m77**,**z**)	YES
Movie^p(**m42**,**y**),Review^p(**m42**,**z**)	Movie^p(**m42**, **1995**)	NO
Movie^p(**m42**,**y**),Review^p(**m42**,**7**)	Movie^p(**m42**,**y**),Review^p(**m42**,**4**)	NO
R^p(**x**,**y**,**z**,**z**,**u**), R^p(**x**,**x**,**x**,**y**,**y**)	R^p(**a**,**a**,**b**,**b**,**c**)	YES

Prop If no two subgoals unify then q1,q2 are independent

Note: *necessary* but not *sufficient* condition

Theorem Independece is Π^p_2 complete [Miklau&S'04]

Reducible to query containment [Machanavajjhala&Gehrke'06]

Disjoint Queries

Let q1, q2 be two boolean queries

<u>Definition</u> q1, q2 are “disjoint” if P(q1, q2) = 0

Iff q1, q2 depend on two disjoint tuples t1, t2

Quiz: which are disjoint ?

q1	q2	?
$HasObject^p$(**<u>'book'</u>**, **<u>'9'</u>**, 'Mary', x)	$HasObject^p$(**<u>'book'</u>**, **<u>'9'</u>**, 'Jim', x)	
$HasObject^p$(**<u>'book'</u>**, **<u>t</u>**, 'Mary', x)	$HasObject^p$(**<u>'book'</u>**, **<u>t</u>**, 'Jim', x)	
$HasObject^p$(**<u>'book'</u>**, **<u>'9'</u>**, u, x)	$HasObject^p$(**<u>'book'</u>**, **<u>'9'</u>**, v, x)	

Answers

q1	q2	?
$HasObject^p$(‘**<u>book</u>**’, ‘**<u>9</u>**’, ‘Mary’, x)	$HasObject^p$(‘**<u>book</u>**’, ‘**<u>9</u>**’, ‘Jim’, x)	Y
$HasObject^p$(‘**<u>book</u>**’, **<u>t</u>**, ‘Mary’, x)	$HasObject^p$(‘**<u>book</u>**’, **<u>t</u>**, ‘Jim’, x)	N
$HasObject^p$(‘**<u>book</u>**’, ‘**<u>9</u>**’, u, x)	$HasObject^p$(‘**<u>book</u>**’, ‘**<u>9</u>**’, v, x)	N

<u>Proposition</u> q1, q2 are “disjoint” if they contain subgoals g1, g2:

- Have the same values for the key attributes
- these values are constants
- have at least one different constant in the non-key attributes

Definition of Safe Operators

$q1(x)q2(x)$

$\bowtie$

$q1(x)$ $q2(x)$

"safe" if ∀a, q1(a), q2(a) are independent

$q(x)$

$\sigma_{x=a}$

$q(x)$

Always "safe"

q

Π^i

$q(x)$

"safe" if ∀a, b, q(a), q(b) are independent

q

Π^d

$q(x)$

"safe" if ∀a, b, q(a), q(b) are disjoint

$q(y^c)$:- $Movie^p(\underline{\mathbf{x}},y^c)$, $Review^p(\underline{\mathbf{x}},\underline{\mathbf{z}})$

y^c “is a constant”

Example 1

q1 :- Movie(x,y^c), Review(x,z)

Π^i_y

Unsafe

Because these are dependent:
q1(m42,7)=Movie(m42,y^c),Review(m42,7)
q1(m42,4)=Movie(m42,y^c),Review(m42,4)

q1(x,z) :- Movie(x,y^c), Review(x,z)

$\bowtie_x$

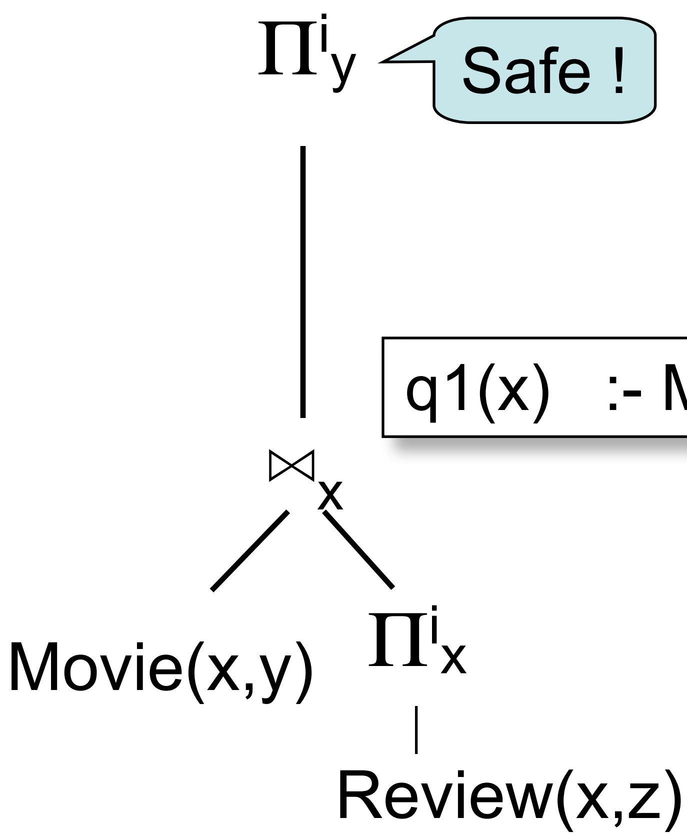
Movie(x,y) Review(x,z)

$q(y^c)$:- $Movie^p(\underline{\mathbf{x}},y^c)$, $Review^p(\underline{\mathbf{x}},\underline{\mathbf{z}})$

y^c "is a constant"

Example 2

q1 :- Movie(x,y^c), Review(x,z)



Now these are independent !
q1(m42) = Movie(m42,y^c), Review(m42,z)
q1(m77) = Movie(m77,y^c), Review(m77,z)

q1(x) :- Movie(x,y^c), Review(x,z)

[Valiant'79]

Complexity Class #P

<u>Definition</u> #P is the class of functions f(x) for which there exists a PTIME non-deterministic Turing machine M s.t. f(x) = number of accepting computations of M on input x

Examples:

SAT = "given formula Φ, is Φ satisfiable ?"
= NP-complete

#SAT = "given formula Φ, count # of satisfying assignments"
= #P-complete

[Valiant'79] [Provan&Ball'83]

All You Need to Know About #P

Class	Example	SAT	#SAT
3CNF	$(X \vee Y \vee Z) \wedge (\neg X \vee U \vee W) \ldots$	NP	#P
2CNF	$(X \vee Y) \wedge (\neg X \vee U) \ldots$	PTIME	#P
Positive, partitioned 2CNF	$(X1 \vee Y1) \wedge (X1 \vee Y4) \wedge (X2 \vee Y1) \wedge (X3 \vee Y1) \ldots$	PTIME	#P
Positive, partitioned 2DNF	$(X1 \wedge Y1) \vee (X1 \wedge Y4) \vee (X2 \wedge Y1) \vee (X3 \wedge Y1) \ldots$	PTIME	#P

Here NP, #P means "NP-complete, #P-complete"

See also [Graedel et al. 98]

#P-Hard Queries

hd1 :- $R^p(\underline{\mathbf{x}}), S(\underline{\mathbf{x}}, \underline{\mathbf{y}}), T^p(\underline{\mathbf{y}})$

Theorem The query hd1 is #P-hard

Proof: Reduction from partitioned, positive 2DNF

E.g. Φ = x1 y1 V x2 y1 V x1 y2 V x3 y2 reduces to

R^p

A	P
x1	0.5
x2	0.5
x3	0.5

S

A	B
x1	y1
x2	y1
x1	y2
x3	y2

T^p

B	P
y1	0.5
y2	0.5

$\#\Phi = P(hd1) * 2^n$

#P-Hard Queries

- #P-hard queries do not have safe plans
- Do not have *any* PTIME algorithm
  - Unless P = NP
- Can be evaluated using probabilistic inference
  - Exponential time exact algorithms or
  - PTIME approximations, e.g. Luby&Karp
- In our experience with MystiQ, unsafe queries are 2 orders of magnitude slower than safe queries, and that only after optimizations

Lessons

What do users want ?

- *Arbitrary* queries, not just safe queries
  - Safe query ➔ very fast
  - Unsafe query ➔ begs for optimizations

What should the system do ?

- Aggressively check if a query is safe
- If not, aggressively search safe subqueries

Key problem: identifying the safe queries

Dichotomy Property

REP = a representation formalism (Independent or independent/disjoint)

LANG = a query language.

REP, LANG have the **DICHOTOMY PROPERTY** if $\forall$ q $\in$ LANG

(1) The complexity of q is PTIME, or

(2) The complexity of q is #P-hard

LANG: CQ = conjunctive queries

CQ^1 = conjunctive queries without self-joins

Theorems The dichotomy property holds for:

1. CQ^1 and independent dbs.
2. CQ^1 and disjoint/independent dbs.
3. CQ and independent dbs.

Summary So Far

- Lots of applications need probabilistic data
- Tuple disjoint/independent data model
  - Sufficient for many applications
  - Can be made complete through views
  - Ideal for studying query evaluation
- Query evaluation
  - Some (many ?) queries are inherently hard
  - Main optimization tool: safe queries

Outline

Part 1:

- Motivation
- Data model
- Basic query evaluation

Part 2:

- The dichotomy of query evaluation
- Implementation and optimization
- Six Challenges

Dichotomy Property

REP = a representation formalism (Independent or independent/disjoint)

LANG = a query language.

REP, LANG have the **DICHOTOMY PROPERTY** if $\forall$ q $\in$ LANG

(1) The complexity of q is PTIME, or

(2) The complexity of q is #P-hard

LANG: CQ = conjunctive queries

CQ^1 = conjunctive queries without self-joins

Theorems The dichotomy property holds for:

1. CQ^1 and independent dbs.
2. CQ^1 and disjoint/independent dbs.
3. CQ and independent dbs.

PTIME Queries | #P-Hard Queries

R(**x**, **y**), S(**x**, **z**)

R(**x**, y), S(**y**), T(**'a'**, y)

R(**x**), S(**x**, **y**), T(**y**), U(**u**, y), W(**'a'**, u)

. . .

hd1 = R(**x**), S(**x**, **y**), T(**y**)

hd2 = R(**x**,y), S(**y**)

hd3 = R(**x**,y), S(x,**y**)

. . .

Will discuss next how to decide their complexity and how evaluate PTIME queries

Hierarchical Queries

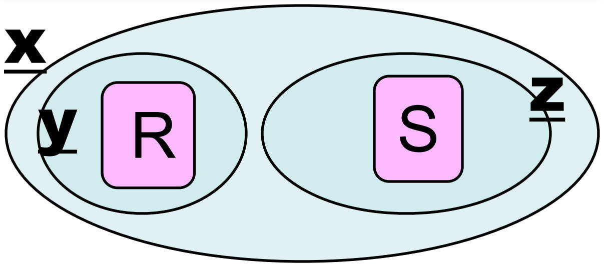
sg(x) = set of subgoals containing the variable x in a key position

Definition A query q is *hierarchical* if forall x, y:

$$sg(x) \supseteq sg(y) \quad \text{or} \quad sg(x) \subseteq sg(y) \quad \text{or} \quad sg(x) \cap sg(y) = \emptyset$$

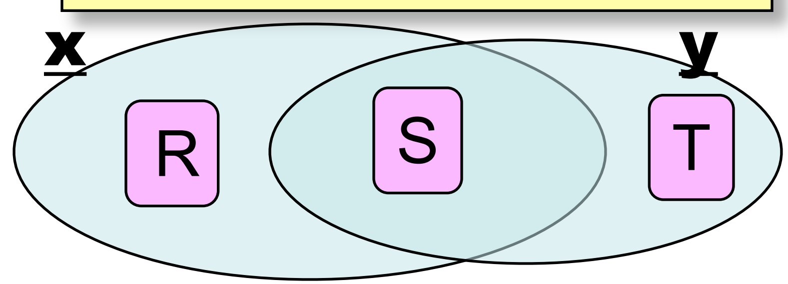
Hierarchical

$q = R(\underline{\mathbf{x}}, \underline{\mathbf{y}}), S(\underline{\mathbf{x}}, \underline{\mathbf{z}})$



Non-hierarchical

$h1 = R(\underline{\mathbf{x}}), S(\underline{\mathbf{x}}, \underline{\mathbf{y}}), T(\underline{\mathbf{y}})$



Case 1: CQ^1 + Independent

- Dichotomy established in [Dalvi&S'2004]

- CQ^1 (conjunctive queries, no self-joins):
  - $R(\underline{\mathbf{x}},\underline{\mathbf{y}}), S(\underline{\mathbf{y}},\underline{\mathbf{z}})$ OK
  - $R(\underline{\mathbf{x}},\underline{\mathbf{y}}), R(\underline{\mathbf{y}},\underline{\mathbf{z}})$ Not OK
- Independent tuples only:
  - $R(\underline{\mathbf{x}},\underline{\mathbf{y}})$ OK
  - $S(\underline{\mathbf{y}},z)$ Not OK

[Dalvi&S'2004]

CQ^1 + Independent

Theorem Forall $q \in CQ^1$:

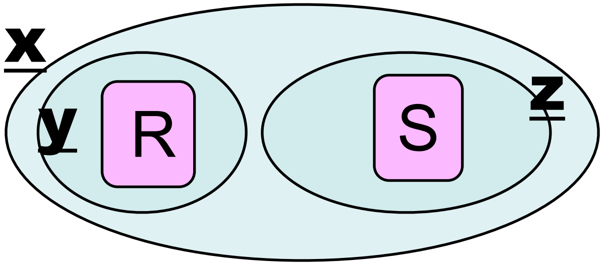
- q is hierarchical, has a safe plan, and is in PTIME, OR
- q is not hierarchical and is #P-hard

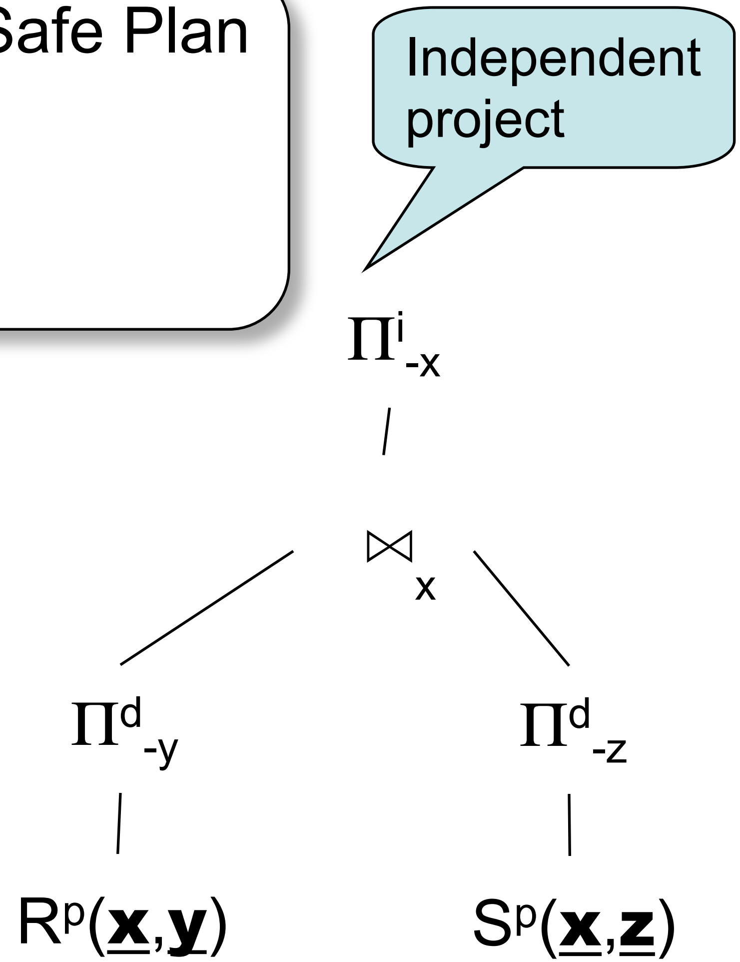
The PTIME Queries

Algorithm: convert a Hierarchy to a Safe Plan

1. Root variable u ➔ Π^{i}_{-u}
2. Connected components ➔ Join
3. Single subgoal ➔ Leaf node

q = R(**x**, **y**), S(**x**, **z**)





P(q) =
$1 - (1-p_1(1-(1-q_1)(1-q_2))) * (1-p_2(1-(1-q_3)(1-q_4)(1-q_5)))$

q =

R(<u>**x**</u>, <u>**y**</u>), S(<u>**x**</u>, <u>**z**</u>)

Π_{-x}

$\bowtie_{x}$

Π_{-y} Π_{-z}

R^p(<u>**x**</u>,<u>**y**</u>) S^p(<u>**x**</u>,<u>**z**</u>)

A	P
a_1	$p_1(1-(1-q_1)(1-q_2))$
a_2	$p_2(1-(1-q_3)(1-q_4)(1-q_5))$

A	P
a_1	$1-(1-q_1)(1-q_2)$
a_2	$1-(1-q_3)(1-q_4)(1-q_5)$

<u>**A**</u>	<u>**C**</u>	P
a_1	c_1	q_1
a_1	c_2	q_2
a_2	c_3	q_3
a_2	c_4	q_4
a_2	c_5	q_5

<u>**A**</u>	<u>**B**</u>	P
a_1	b_1	p_1
a_2	b_2	p_2

[D&S'2004]

The #P-Hard Queries

Are precisely the non-hierarchical queries. Example:

hd1 :- R(<u>**x**</u>), S(<u>**x, y**</u>), T(<u>**y**</u>)

More general:

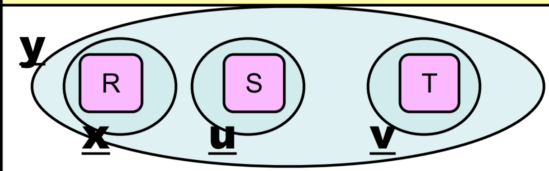
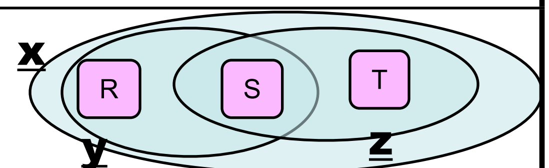
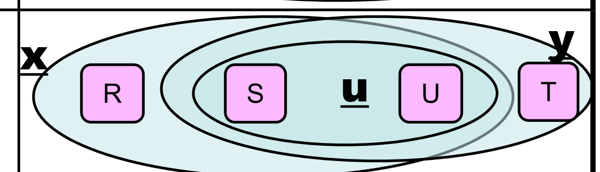
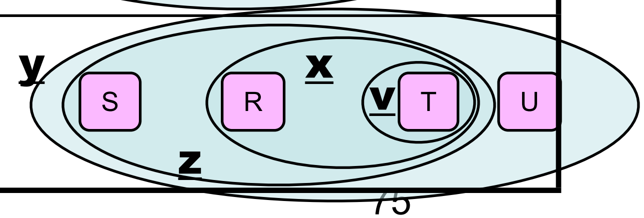
q :- …, R(<u>**x**</u>, …), S(<u>**x, y**</u>, …), T(<u>**y**</u>, …) , …

<u>**Theorem**</u> Testing if q is PTIME or #P-hard is in AC^0

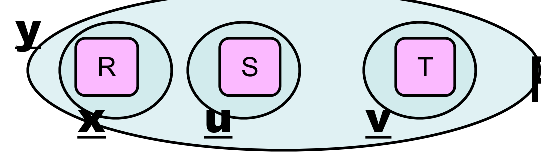
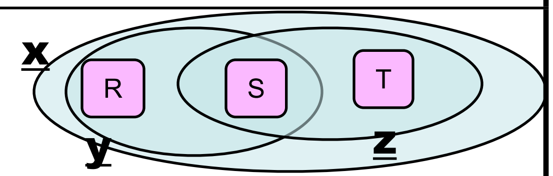
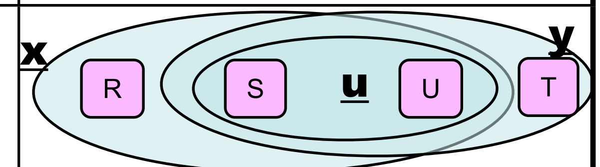
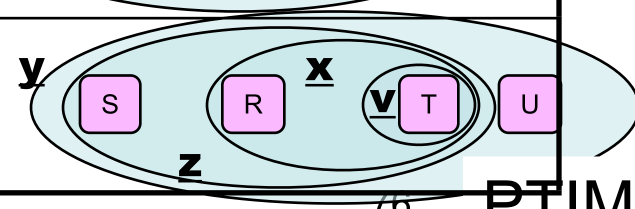
Quiz: What is their complexity ?

q	PTIME or #P ?
R($\underline{\mathbf{x}}$,$\underline{\mathbf{y}}$),S($\underline{\mathbf{y}}$,$\underline{\mathbf{a}}$,$\underline{\mathbf{u}}$),T($\underline{\mathbf{y}}$,$\underline{\mathbf{y}}$,$\underline{\mathbf{v}}$)	
R($\underline{\mathbf{x}}$,$\underline{\mathbf{y}}$), S($\underline{\mathbf{x}}$,$\underline{\mathbf{y}}$,$\underline{\mathbf{z}}$), T($\underline{\mathbf{x}}$,$\underline{\mathbf{z}}$)	
R($\underline{\mathbf{x}}$,$\underline{\mathbf{a}}$),S($\underline{\mathbf{y}}$,$\underline{\mathbf{u}}$,$\underline{\mathbf{x}}$),T($\underline{\mathbf{u}}$,$\underline{\mathbf{y}}$),U($\underline{\mathbf{x}}$,$\underline{\mathbf{y}}$)	
R($\underline{\mathbf{x}}$,$\underline{\mathbf{y}}$,$\underline{\mathbf{z}}$),S($\underline{\mathbf{z}}$,$\underline{\mathbf{u}}$,$\underline{\mathbf{y}}$),T($\underline{\mathbf{y}}$,$\underline{\mathbf{v}}$,$\underline{\mathbf{z}}$,$\underline{\mathbf{x}}$),U($\underline{\mathbf{y}}$)	

Hint…

q	PTIME or #P ?
R(**x**,**y**),S(**y**,**a**,**u**),T(**y**,**y**,**v**)	
R(**x**,**y**), S(**x**,**y**,**z**), T(**x**,**z**)	
R(**x**,**a**),S(**y**,**u**,**x**),T(**u**,**y**),U(**x**,**y**)	
R(**x**,**y**,**z**),S(**z**,**u**,**y**),T(**y**,**v**,**z**,**x**),U(**y**)	

…Answer

q	PTIME or #P ?
R(**x**,**y**),S(**y**,**a**,**u**),T(**y**,**y**,**v**)	 PTIME
R(**x**,**y**), S(**x**,**y**,**z**), T(**x**,**z**)	 #P
R(**x**,**a**),S(**y**,**u**,**x**),T(**u**,**y**),U(**x**,**y**)	 #P
R(**x**,**y**,**z**),S(**z**,**u**,**y**),T(**y**,**v**,**z**,**x**),U(**y**)	 PTIME

Case 2: CQ^1+Disjoint/independent

- Dichotomy: in [Dalvi et al.'06,Dalvi&S'07]
- Some safe plans also in [Andritsos'2006]

- CQ^1 (conjunctive queries, no self-joins)
- Independent/independent tables are OK

Theorem Forall $q \in CQ^1$

- q has a safe plan and is in PTIME, OR
- q is #P-hard

The PTIME Queries

<u>Algorithm</u>: find a Safe Plan

1. Root variable u ➔ Π^{i}_{-u}
2. Variable u occurs in a subgoal with constant keys ➔ Π^{D}_{-u}
3. Connected components ➔ Join

- Single subgoal ➔ Leaf node

$q(y)$:- $R(\underline{\mathbf{x}},y,z)$

Π_{-x}^{i}

|

Π_{-z}^{D}

|

$R(\underline{\mathbf{x}},y,z)$

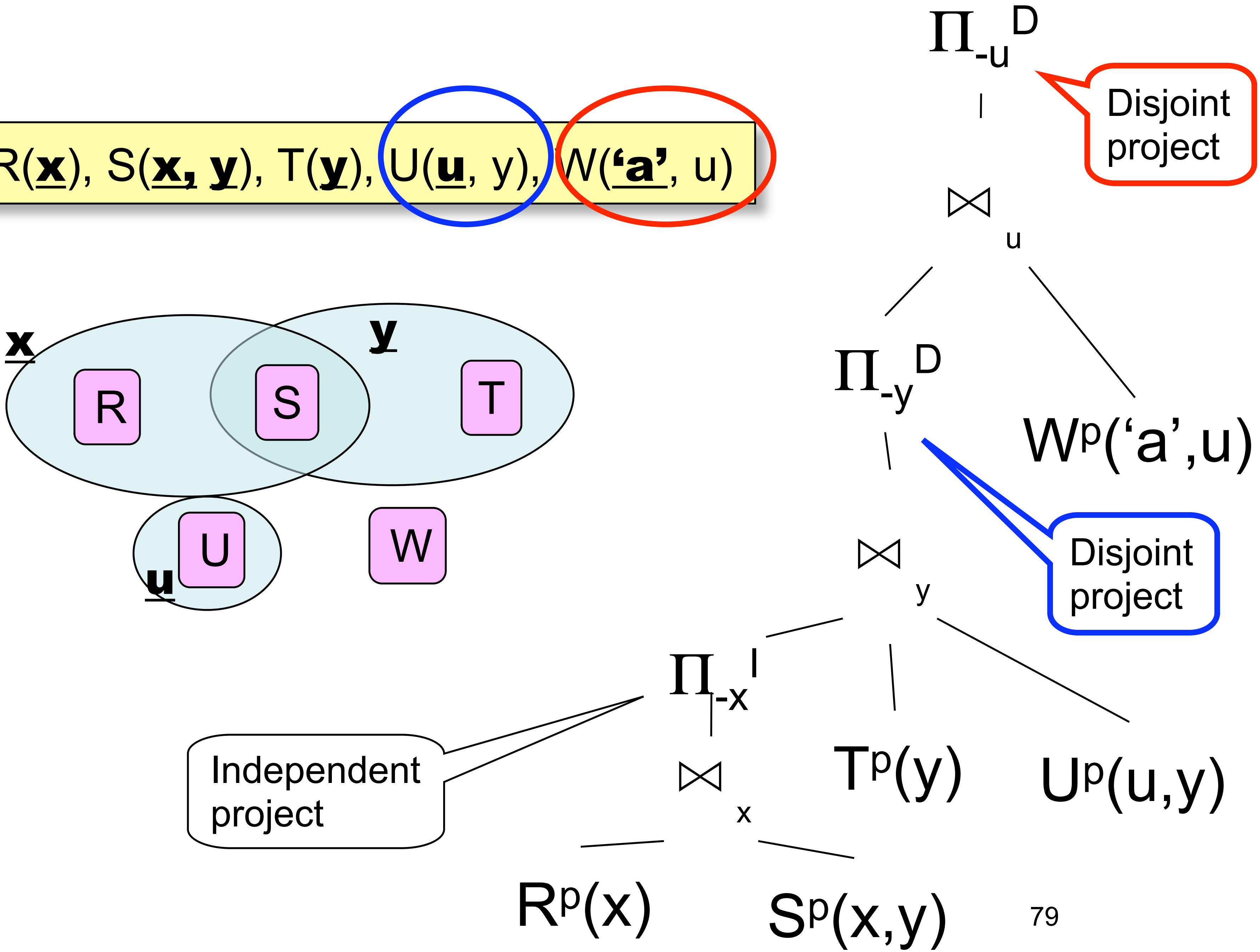
$q1(x^c,y^c)$:-$R(\underline{\mathbf{x}}^c,y^c,z)$

y	P
b	1-(1-p1-p2)(1-p3-p4)

$\underline{\mathbf{x}}$	y	P
a1	b	p1+p2
a2	b	p3+p4

$\underline{\mathbf{x}}$	y	z	P
a1	b	c1	p1
	b	c2	p2
a2	b	c1	p3
	b	c2	p4

R(**x**), S(**x, y**), T(**y**), U(**u**, y), W(**'a'**, u)



[Dalvi&S'2007]

The #P-Hard Queries

hd1 = R(<u>**x**</u>), S(<u>**x**</u>, <u>**y**</u>), T(<u>**y**</u>)

hd2 = R(<u>**x**</u>,y), S(<u>**y**</u>)

hd3 = R(<u>**x**</u>,y), S(x,<u>**y**</u>)

There are variations on hd2, hd3 (see paper)

In general, a query is #P-hard if it can be "rewritten" to hd1, hd2, hd3 or one of their "variations".

<u>**Theorem**</u> Testing if q is PTIME or #P-hard is PTIME complete

[Dalvi&S'2007b]

Case 3: Any conjunctive query, independent tables

Let q be hierarchical

- $x \supseteq y$ denotes: x is above y in the hierarchy
- $x \equiv y$ denotes: $x \supseteq y$ and $x \subseteq y$

<u>Definition</u> An inversion is a chain of unifications:
$x \supset y$ with $u_1 \equiv v_1$ with … with $u_n \equiv v_n$ with $x' \subset y'$

<u>Theorem</u> Forall $q \in$ CQ:

- If q is non-hierarchical, or has an inversion* then it is #P-hard
- Otherwise it is in PTIME

*without "eraser": see paper.

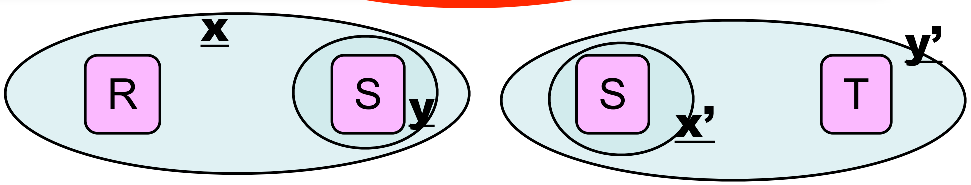
[Dalvi&S'2007b]

The #P-hard Queries

Hierarchical queries with "inversions":

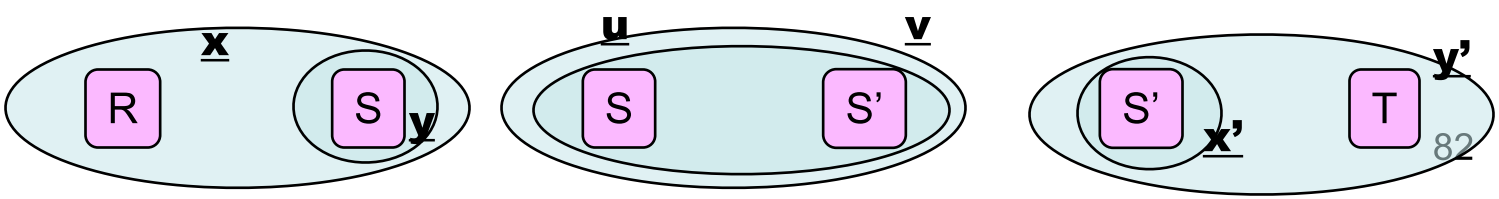
hi1 = R(<u>**x**</u>), S(<u>**x**</u>,<u>**y**</u>), S(<u>**x'**</u>,<u>**y'**</u>), T(<u>**y'**</u>)

$x \supset y$ unifies with $x' \subset y'$



hi2 = R(<u>**x**</u>), S(<u>**x**</u>,<u>**y**</u>), S(<u>**u**</u>,<u>**v**</u>), S'(<u>**u**</u>,<u>**v**</u>),S'(<u>**x'**</u>,<u>**y'**</u>), T(<u>**y'**</u>)

$x \supset y$ unifies with $u \equiv v$, which unifies with $x' \subset y'$



The #P-hard Queries

A query with a long inversion:

$$\begin{aligned} hi_k = R(\underline{\mathbf{x}}), S_0(\underline{\mathbf{x}},\underline{\mathbf{y}}), & \\ & S_0(\underline{\mathbf{u}}_1,\underline{\mathbf{v}}_1),\ S_1(\underline{\mathbf{u}}_1,\underline{\mathbf{v}}_1) \\ & \qquad S_1(\underline{\mathbf{u}}_2,\underline{\mathbf{v}}_2), S_2(\underline{\mathbf{u}}_2,\underline{\mathbf{v}}_2), \ldots \\ & \qquad\qquad S_k(\underline{\mathbf{x}'},\underline{\mathbf{y}'}), T(\underline{\mathbf{y'}}) \end{aligned}$$

The #P-hard Queries

Sometimes inversions are exposed only after making a copy of the query

$q = R(\underline{\mathbf{x}},\underline{\mathbf{y}}), R(\underline{\mathbf{y}},\underline{\mathbf{z}})$

$R(x,y), R(y,z)$
$\quad R(x',y'), R(y',z')$

The PTIME Queries

Find movies with high reviews from Joe and Jim:

q(x) :- Movie(x,y),Match(x,r), Review(r,Joe,s), s > 4
Match(x,r'), Review(r',Jim,s'),s'>4

Unify, but no inversion

Don't unify

Note: the query is hierarchical because x is a "constant"

[Dalvi&S'2007b]

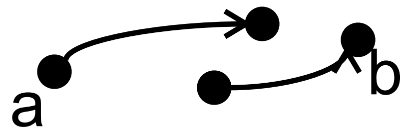
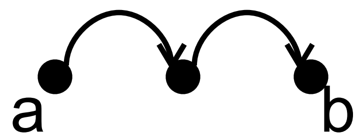
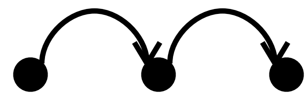
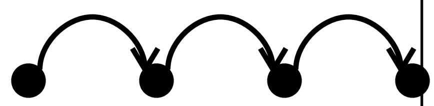
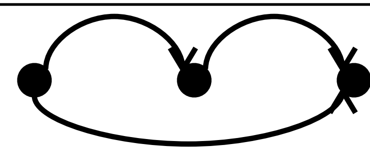
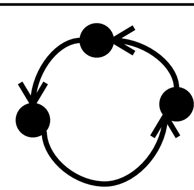
The PTIME Queries

Note: no "safe plans" are known ! PTIME algorithm for an inversion-free query is given in terms of expressions, not plans. Example:

q :- R(**a**,**x**), R(**y**,**b**)

p(q) =
$p(R(a,b))+(1-p(R(a,b))(1-(1-\prod_{y \in Dom, y \neq a}(1-p(R(y,b))))(1-\prod_{x \in Dom, x \neq b}(1-p(R(a,x))))$

Open Problem: what are the natural operators that allow us to compute inversion-free queries in a database engine ?

Query		Com-plexity	Why
R(a,x), R(y,b)		PTIME	
R(a,x), R(x,b)		PTIME	
R(x,y), R(y,z)		#P	Inversion
R(x,y),R(y,z),R(z,u)		#P	Non-hierarchical
R(x,y),R(y,z),R(z,x)		#P	Non-hierarchical
R(x,y),R(y,z),R(x,z)		#P	Non-hierarchical

History

- [Graedel, Gurevitch, Hirsch'98]
  - L(x,y),R(x,z),S(y),S(z) is #P-hard
    This is non-hierarchical, with a self-join
- [Dalvi&S'2004]
  - R(x),S(x,y),T(y) is #P-hard
    This is non-hierarchical, w/o self-joins
  - Without self-joins: non-hierarchical = #P-hard, and hierarchical = PTIME
- [Dalvi&S'2007]
  - <u>*All*</u> non-hierarchical queries are #P-hard

Summary on the Dichotomy

WHY WE CARE:
Safe queries = most powerful optimization we have

What we know:

- Three dichotomies, of increasing complexity
- Dichotomy for aggregates in HAVING

[Re&S.2007]

What is open

- CQ + independent/disjoint
- Extensions to $\leq$, $\geq$, $\neq$
- Extensions to unions of conjunctive queries

Outline

Part 1:

- Motivation
- Data model
- Basic query evaluation

Part 2:

- The dichotomy of query evaluation
- Implementation and optimization
- Six Challenges

Implementation and Optimization

Topics:

- General probabilistic inference
- Optimization 1: Safe-subplans
- Optimization 2: Top K
- Performance of MystiQ

General Query Evaluation

- Query q + database DB
  ➔ boolean expression Φ_q^{DB}

- Run any probabilistic inference algorithm on Φ_q^{DB}

This approach is taken in Trio

Background: Probability of Boolean Expressions

Given:

$$\Phi = X_1X_2 \lor X_1X_3 \lor X_2X_3$$

$P(X_1)= p_1$, $P(X_2)= p_2$, $P(X_3)= p_3$

Compute $P(\Phi)$

$\Omega =$

X_1	X_2	X_3	P	Φ
0	0	0		0
0	0	1		0
0	1	0		0
0	1	1	$(1-p_1)p_2p_3$	1
1	0	0		0
1	0	1	$p_1(1-p_2)p_3$	1
1	1	0	$p_1p_2(1-p_3)$	1
1	1	1	$p_1p_2p_3$	1

$$\Pr(\Phi)=(1-p_1)p_2p_3 + p_1(1-p_2)p_3 + p_1p_2(1-p_3) + p_1p_2p_3$$

#P-complete [Valiant:1979]

Query q + Database PDB ➔ Φ

q= $R(\underline{\mathbf{x}}, \underline{\mathbf{y}}), S(\underline{\mathbf{x}}, \underline{\mathbf{z}})$

PDB=

R^p

A	**B**	P	
a_1	b_1	p_1	X_1
a_2	b_2	p_2	X_2

S^p

A	**C**	P	
a_1	c_1	q_1	Y_1
a_1	c_2	q_2	Y_2
a_2	c_3	q_3	Y_3
a_2	c_4	q_4	Y_4
a_2	c_5	q_5	Y_5

$$\Phi = X_1Y_1 \vee X_1Y_2 \vee X_2Y_3 \vee X_2Y_4 \vee X_2Y_5$$

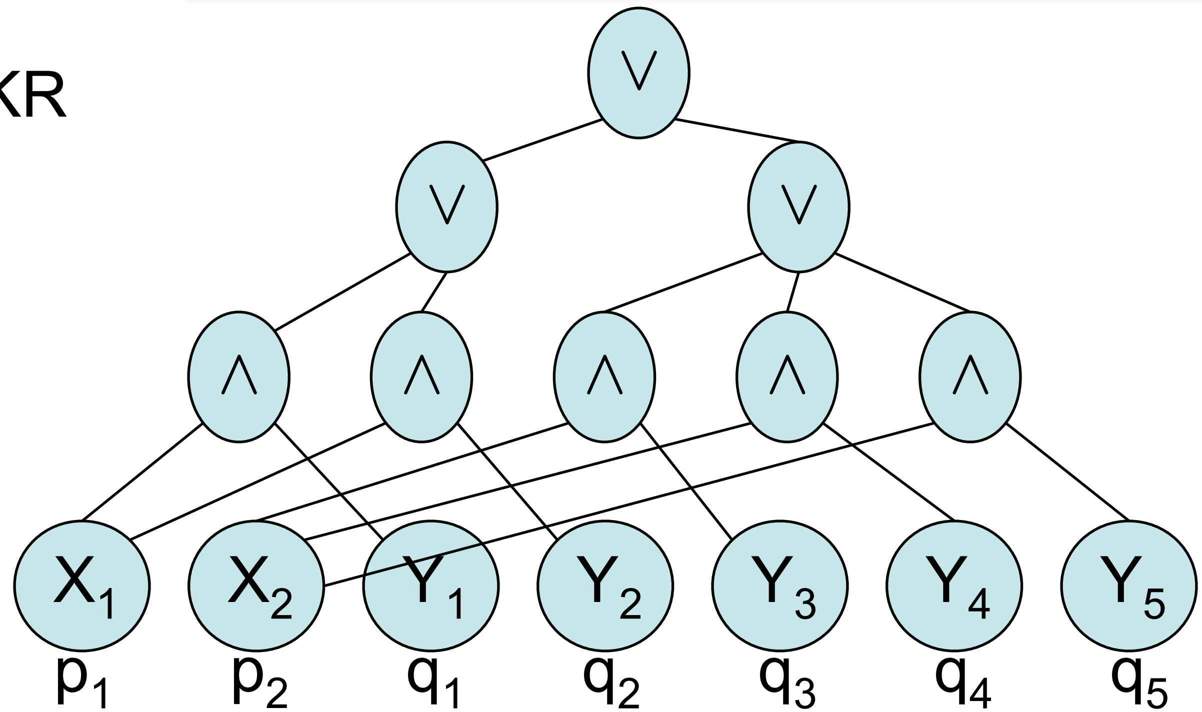
Probabilistic Networks

Nodes = random variables

Edges = dependence

$R(\underline{\mathbf{x}}, \underline{\mathbf{y}}), S(\underline{\mathbf{x}}, \underline{\mathbf{z}})$

$\Phi = X_1Y_1 \vee X_1Y_2 \vee X_2Y_3 \vee X_2Y_4 \vee X_2Y_5$



Studied intensively in KR

Typical networks:

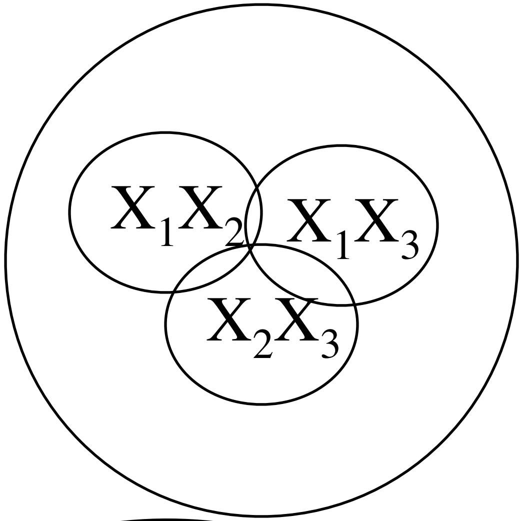
- Bayesian networks
- Markov networks
- <u>*Boolean expressions*</u>

Inference Algorithms for Boolean Expressions

- Randomized:
  - Naïve Monte Carlo
  - Luby and Karp
- Deterministic
  - Algorithmic guarantees: [Trevisan'04], [Luby&Velickovic'91]
  - Inference algorithms in AI: variable elimination, junction trees,…
  - Tractable cases: bounded-width trees [Zabiyaka&Darwiche'06]

Naive Monte Carlo Simulation

$$E \;=\; X_1X_2 \vee X_1X_3 \vee X_2X_3$$



Cnt $\leftarrow$ 0
repeat N times
    randomly choose $X_1, X_2, X_3 \in \{0,1\}$
    if $E(X_1, X_2, X_3) = 1$
        then Cnt = Cnt+1
P = Cnt/N
return P /* $\simeq$ Pr(E) */

May be big
(in theory)

Theorem (0-1 estimator) If $N \geq (1/\Pr(E)) \times (4\ln(2/\delta)/\varepsilon^2)$
then $\Pr[\, | P/\Pr(E) - 1 | > \varepsilon \,] < \delta$

[Karp&Luby:1983]

[Graedel,Gurevitch,Hirsch:1998]

Improved Monte Carlo Simulation

$E = C_1 \vee C_2 \vee \ldots \vee C_m$

Cnt $\leftarrow$ 0; S $\leftarrow$ Pr(C_1) + … + Pr(C_m);

repeat N times

  randomly choose i $\in$ {1,2,…, m}, with prob. Pr(C_i) / S

  randomly choose X_1, …, X_n $\in$ {0,1} s.t. C_i = 1

  if C_1=0 and C_2=0 and … and C_{i-1} = 0

    then Cnt = Cnt+1

P = Cnt/N * S / 2^n

return P /* $\simeq$ Pr(E) */

Now it's in PTIME

<u>Theorem</u>. If $N \geq (1/m) \times (4\ln(2/\delta)/\varepsilon^2)$ then:

$$\Pr[\,|\,P/\Pr(E) - 1\,| > \varepsilon\,] < \delta$$

[Re,Dalvi&S'2007]

An Example

$q(x,u)$:- $R^p(\underline{\mathbf{x}},\underline{\mathbf{y}})$, $S^p(\underline{\mathbf{y}},\underline{\mathbf{z}})$, $T^p(\underline{\mathbf{z}},u)$

R^p

<u>A</u>	**<u>B</u>**	P
a1	b1	p1
	b2	p2
a2	b1	p3

S^p

<u>B</u>	**<u>C</u>**	P
b1	c1	q1
b2	c1	q2
	c2	q3
	c3	q4

T^p

<u>C</u>	**<u>D</u>**	P
c1	d1	r1
	d2	r2
c2	d1	r3
	d2	r4
	d3	r5

<u>Step 1:</u> evaluate this query *on the representation* to get the data

$qTemp(x,y,p,y,z,q,z,u, r)$:- $R(x,y,p)$, $S(y,z,q)$, $T(z,u,r)$

R^p

A	**B**	P
a1	b1	p1
a1	b2	p2
a2	b1	p3

S^p

B	**C**	P
b1	c1	q1
b2	c1	q2
b2	c2	q3
b2	c3	q4

T^p

C	**D**	P
c1	d1	r1
	d2	r2
c2	d1	r3
	d2	r4
	d3	r5

qTemp(x,y,p,y,z,q,z,u, r) :- R(x,y,p), S(y,z,q), T(z,u,r)

Temp

A	B	P	B	C	P	C	D	P
a1	b1	p1	b1	c1	q1	c1	d1	r1
a1	b2	p2	b2	c2	q3	c2	d1	r3
a2	b1		. .					
. .	. .		. .					

Step 2: group Temp by the head variables in q

$q(x,u)$:- $R^p(\underline{\mathbf{x}},\underline{\mathbf{y}})$, $S^p(\underline{\mathbf{y}},\underline{\mathbf{z}})$, $T^p(\underline{\mathbf{z}},u)$

	A	B	P	B	C	P	C	D	P
	a1	b1	p1	b1	c1	q1	c1	d1	r1
q(a1,d1)	a1	b2	p2	b2	c2	q3	c2	d1	r3
	. . .								
	a1	b1	p1	b1	c1	q1	c1	d2	r2
q(a1,d2)	a1	b1		. .				d2	
	. .	. .		. .					
	. . .								

Step 3: each group is a DNF formula; run Monte Carlo

q(a1,d1)

A	B	P	B	C	P	C	D	P
a1	b1	p1	b1	c1	q1	c1	d1	r1
a1	b2	p2	b2	c2	q3	c2	d1	r3
. . .								
a1	. . .						d2	

$\Phi_{a1,d1} = X_{11}Y_{11}Z_{11} \vee X_{12}Y_{22}Z_{21} \vee \ldots$ ➔ $P(\Phi_{a1,d1}) = s1$

$\Phi_{a1,d2} = X_{11}Y_{11}Z_{12} \vee \ldots$ ➔ $P(\Phi_{a1,d2}) = s2$

. . . . . .

Where $X_{11} = R(a1,b1)$ $X_{12} = R(a1,b2)$ $Y_{11} = S(b1,c1)$ etc

Step 4: collect all results, return top k

Tem
p

A	B	P	B	C	P	C	D	P
a1	b1	p1	b1	c1	q1	c1	d1	r1
…								
a1	b1	p1	b1	c1	q1	c1	d2	r2
…	…		…					

➔

Answer to q(x,u)

A	D	P
a1	d1	s1
a1	d2	s2
…		

Remark:
- The DBMS executes only the query qTemp: only selections and joins are done in the engine
- The probabilistic inference is done in the middleware

Summary on Monte Carlo

General method for evaluating $P(q)$, $\forall\, q \in CQ$

- Naïve MC: $N = O(1/P(q))$ steps
- Luby&Karp: $N = O(m)$ steps

Lessons from MystiQ: no big difference

- Typically: $P(q) \approx 0.1$ or higher
- Typically: $m \approx 5 - 10$ or higher

Typical number of steps: $N \approx 100{,}000$: this is for *one single* tuple in the answer !

Optimization 1: Safe Subqueries

Main idea:

2. Find subqueries of q that are
   - Safe
   - "Representable"

The "representability" problem is discussed in [Re&S.2007]

4. Evaluate the subqueries using safe plans

6. Rewrite q to q_{opt} by using the subqueries, then evaluate q_{opt} using Monte Carlo

Example

We illustrate with a boolean query (for simplicity):

$$q \text{ :- } R^p(\underline{\mathbf{x}},y),\ S^p(\underline{\mathbf{y}},z),\ T^p(\underline{\mathbf{y}},\underline{\mathbf{z}},\underline{\mathbf{u}})$$

1. Find the following subquery:

$$sq(y) \text{ :- } S^p(\underline{\mathbf{y}},z),\ T^p(\underline{\mathbf{y}},\underline{\mathbf{z}},\underline{\mathbf{u}})$$

- sq is safe: $sq = \Pi^d_y(S \bowtie T)$
- sq(b) is independent from sq(b'), whenever $b \neq b'$

2. Compute sq(y) on the representation using the safe plan:

```
SELECT S.B, sum(S.P*T.P) as P
FROM S,T
WHERE S.C=T.C
GROUP BY S.B
```

→ SQ^p

B	P
b1	t1
b2	t2
. .	

3. Rewrite q to q_{opt}:

q_{opt} :- $R^p(\underline{\mathbf{x}},y), SQ^p(\underline{\mathbf{y}})$

Continue as before:

- Send this to the engine:
- Run Monte Carlo on result

$qTemp_{opt}(x,p,y,q)$:- $R(x,y,p),sq(y,q)$

What's improved:

- Some of the probabilistic inference pushed in RDBMS
- Monte Carlo runs on a smaller DNF

[Re'2007]

Optimization 2: Top-K Ranking

Main idea:

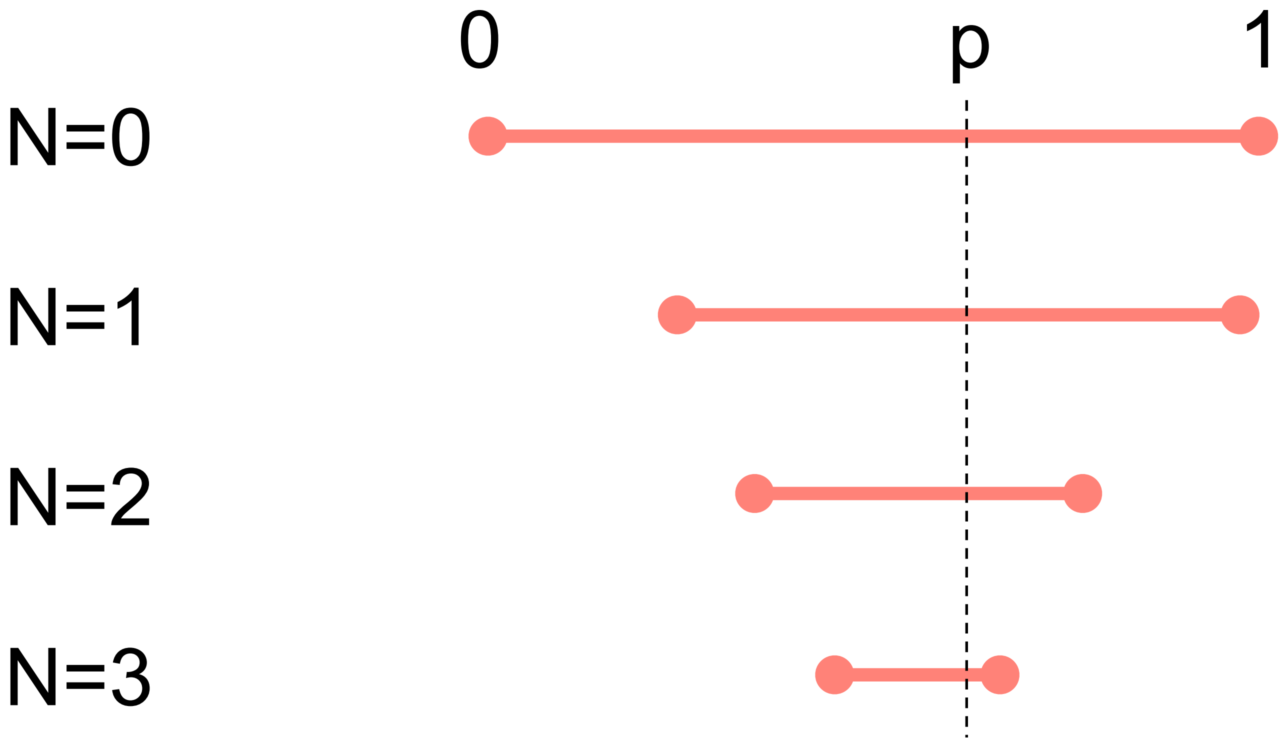
- Number of potential answers is huge
  - 100s or 1000s
- Users want to see only the top-k
  - Typical: top 10, or top 20

Catch 22:

- Run the expensive Monte Carlo *only* on top k
- But to discover the top-k we need to run MC !

Interleave Monte Carlo steps with ranking

Modeling Monte Carlo Simulation



$q(x,u) \text{ :- } R^p(\underline{\mathbf{x}},\underline{\mathbf{y}}), S^p(\underline{\mathbf{y}},\underline{\mathbf{z}}), T^p(\underline{\mathbf{z}},u)$

Current Approximation

A	D	P
a1	d1	0.2 – 0.7
a2	d2	0.6 – 0.8
a3	d3	0 – 1.0
a1000	d1000	0.3 – 0.9

Final, ranked Answer

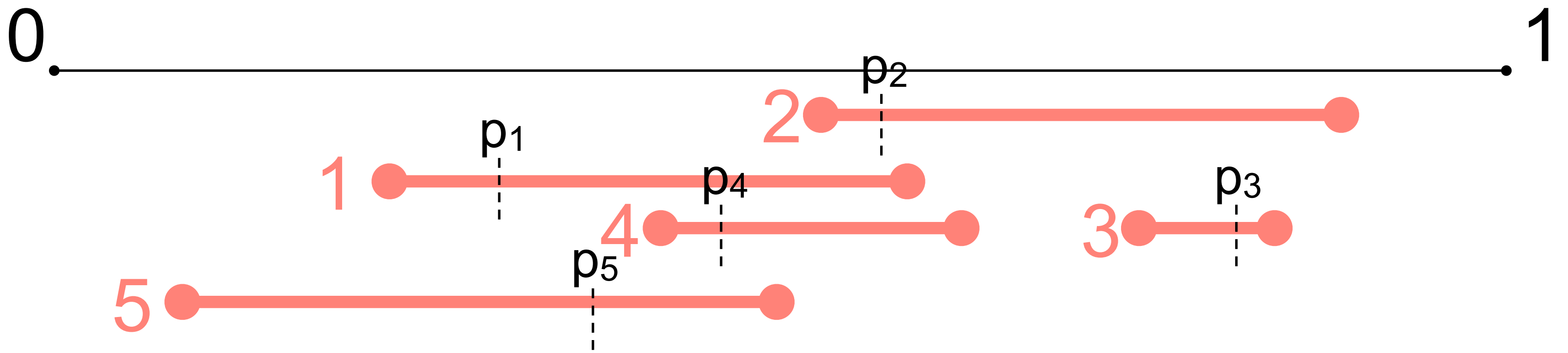
Top-k

Bottom n-k

A	D	P
a49	d49	0.99
a22	d22	0.90
a87	b87	0.85
a522	b522	0.01

Last Quiz: which one should we simulate next ?

We have n objects
How to find the top k ?

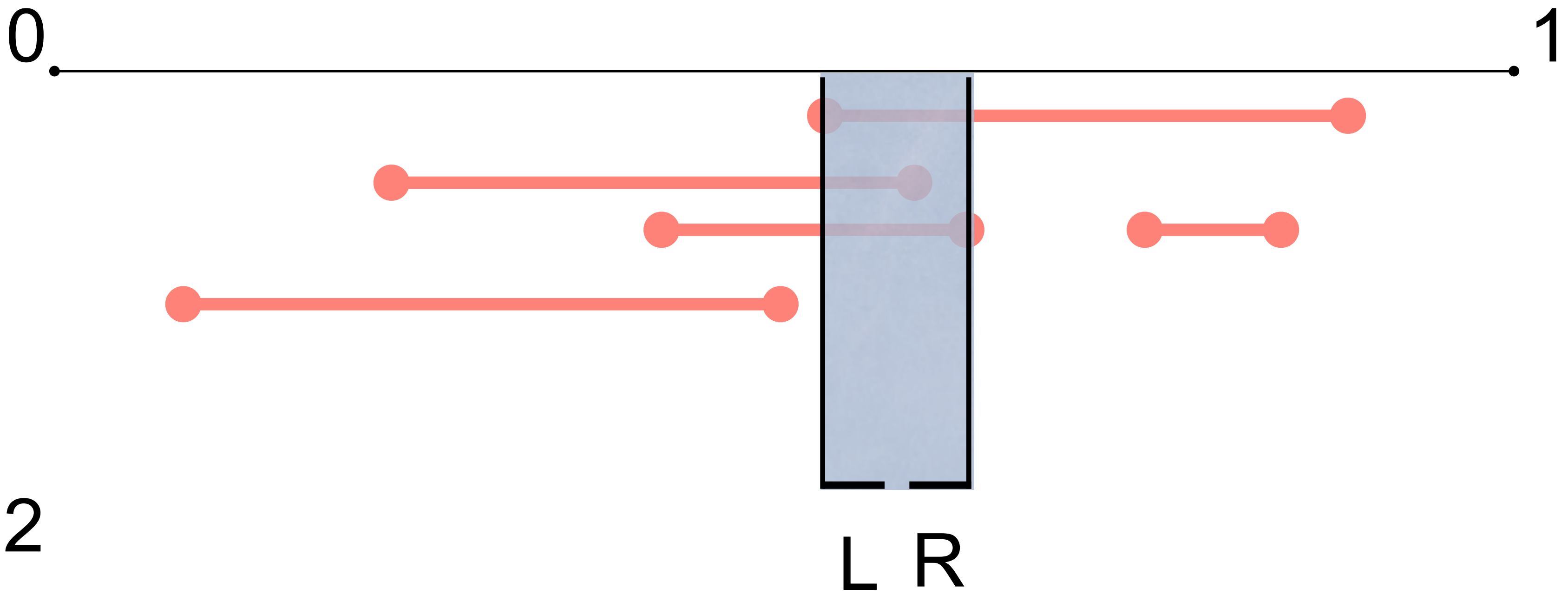


Example: looking for top k=2;

Which one simulate next ?

Multisimulation

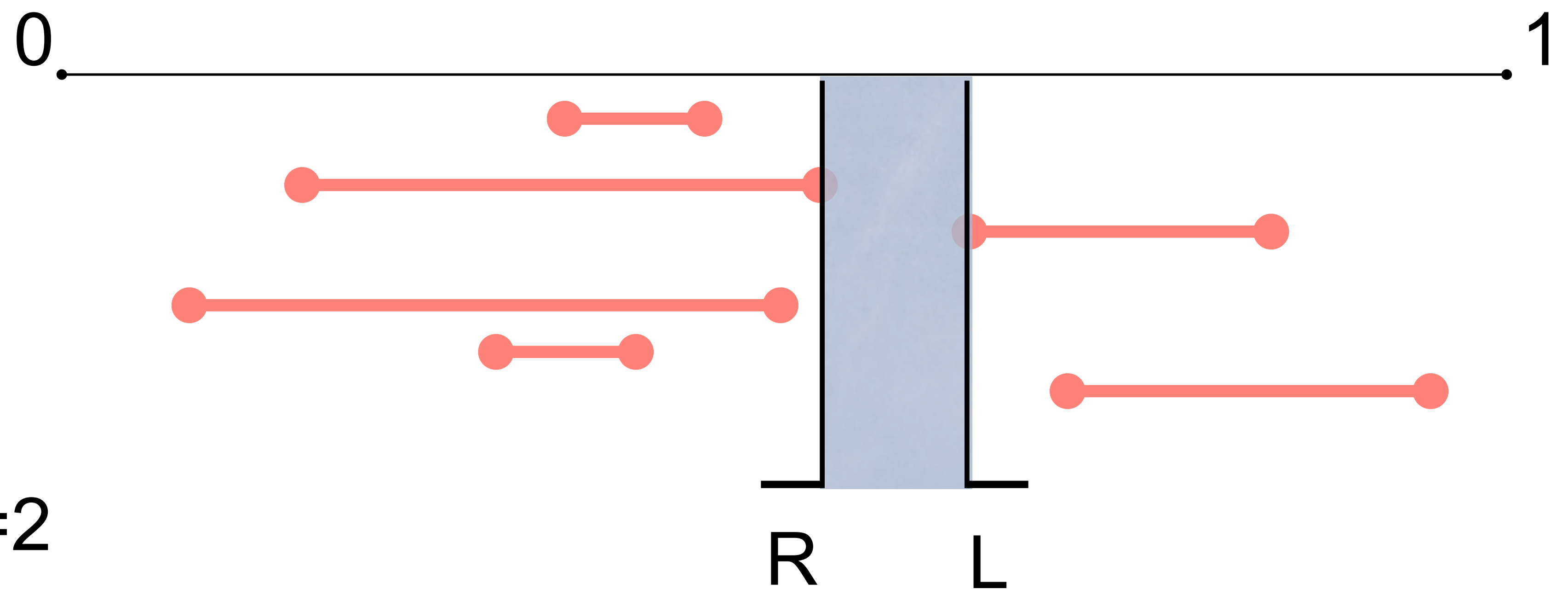
Critical region:
(k'th left, k+1'th right)



k=2

Multisimulation Algorithm

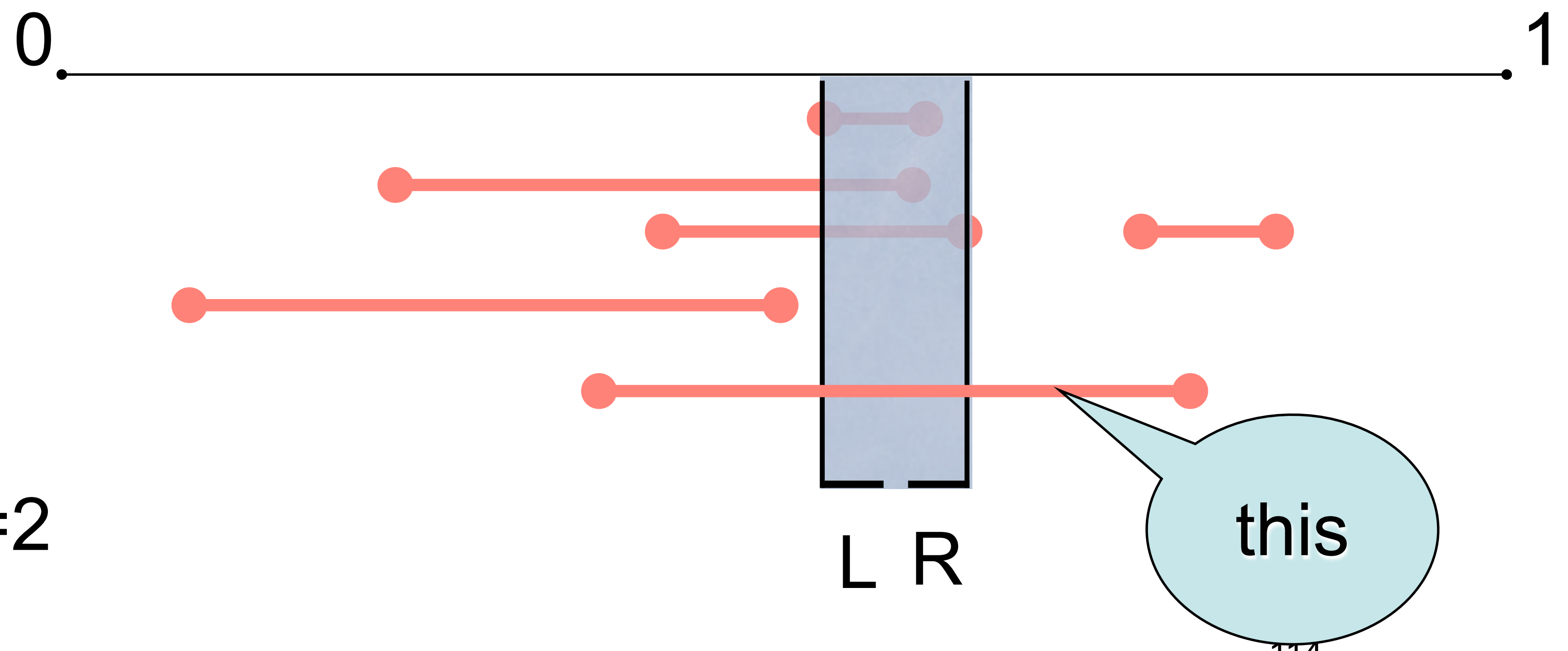
End: when critical region is “empty”



k=2

Multisimulation Algorithm

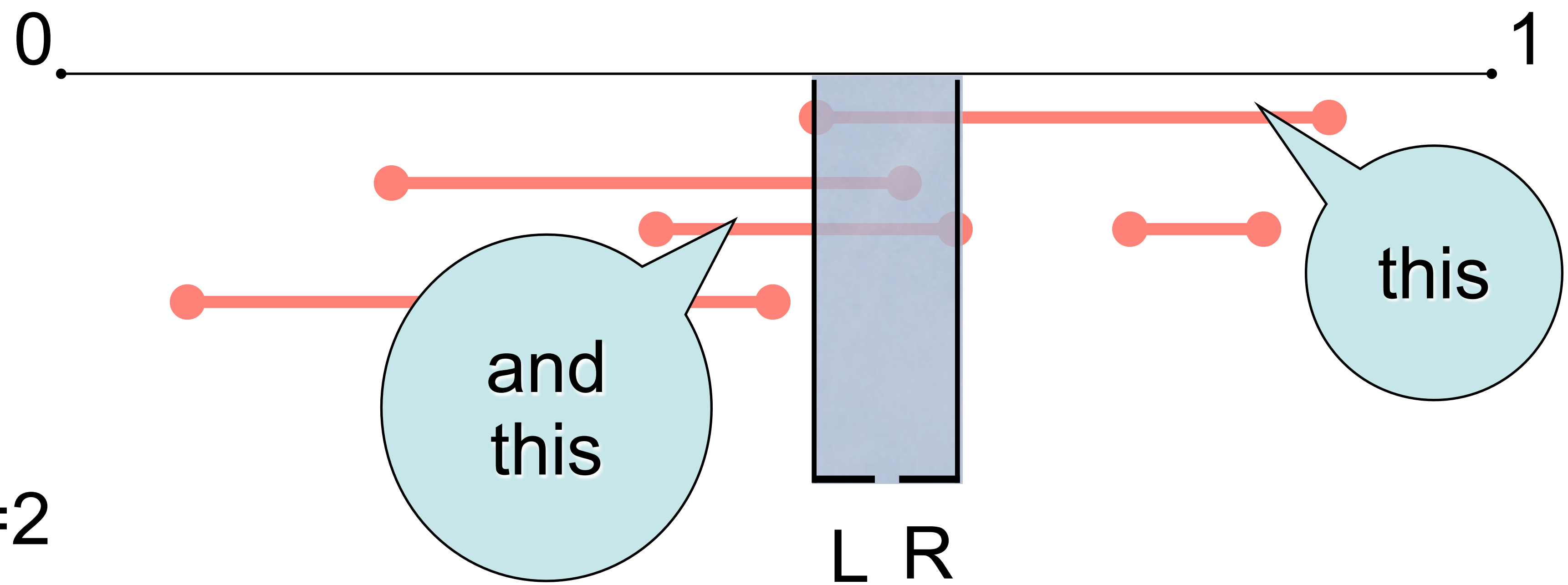
Case 1: pick a “double crosser”
and simulate it



k=2

Multisimulation Algorithm

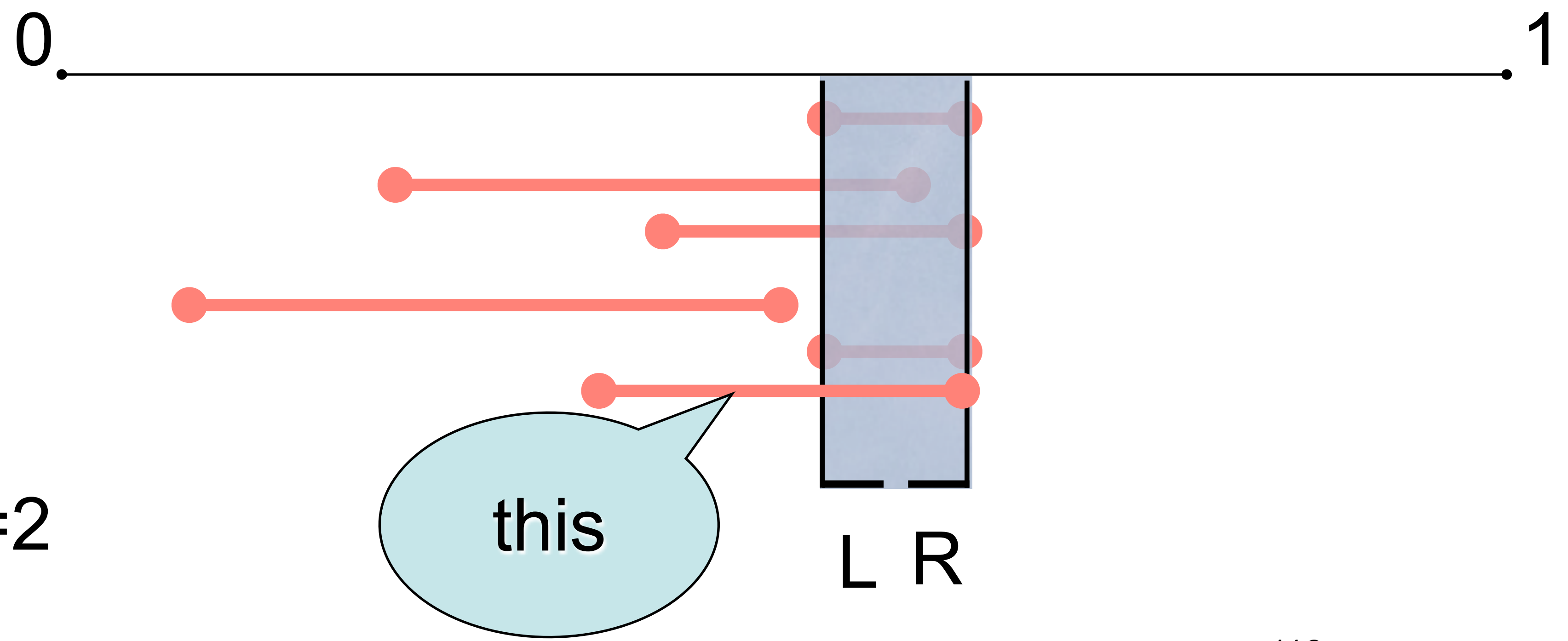
Case 2: pick both a “left” AND a “right” crosser



k=2

Multisimulation Algorithm

Case 3: pick a “max crosser” and simulate it



k=2

[Re'2007]

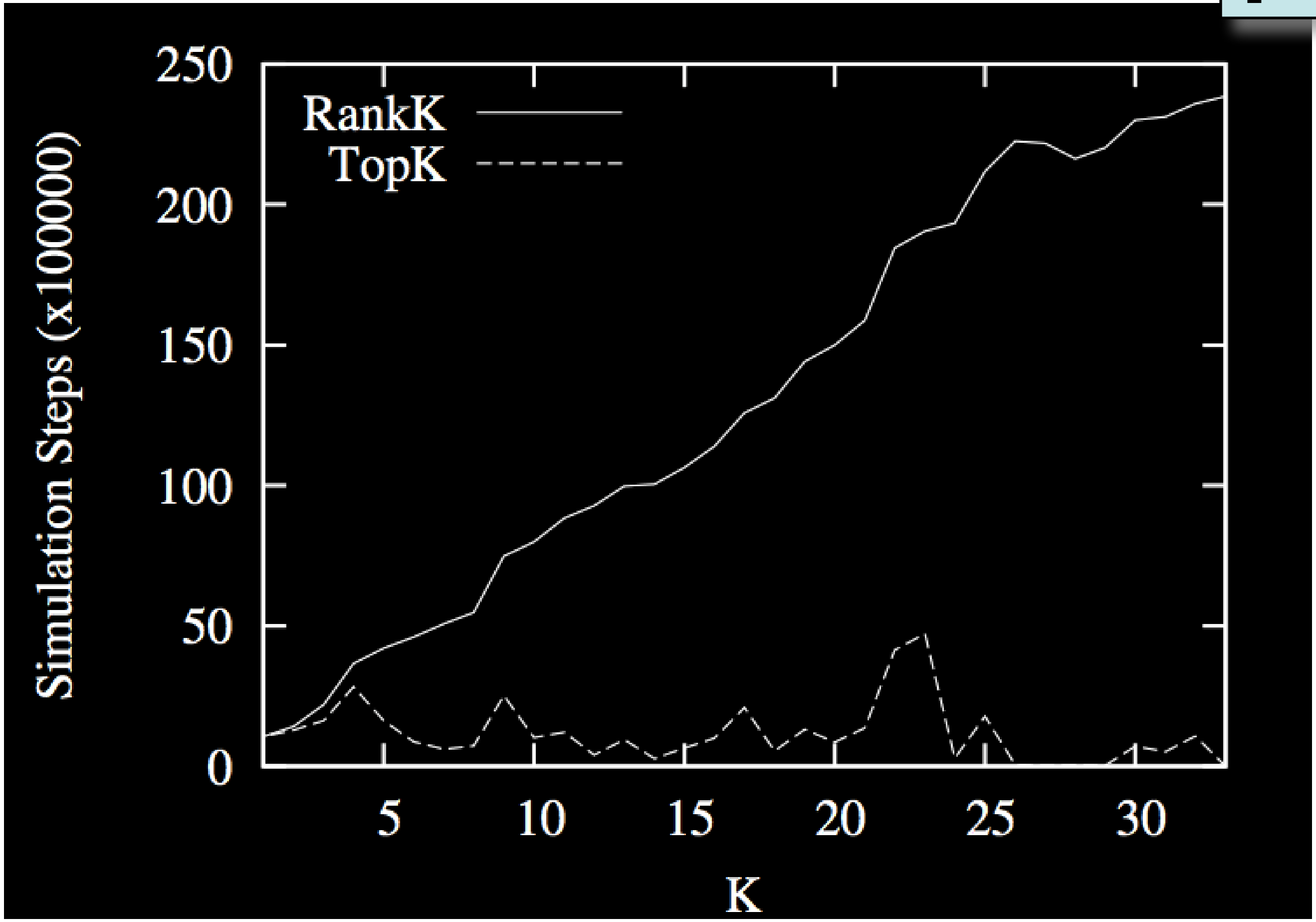
Multisimulation Algorithm

<u>Theorem</u> (1) It runs in < 2 Optimal # steps
(2) no other deterministic algorithm does better

Performance of MystiQ

10 million probabilistic tuples; DB2

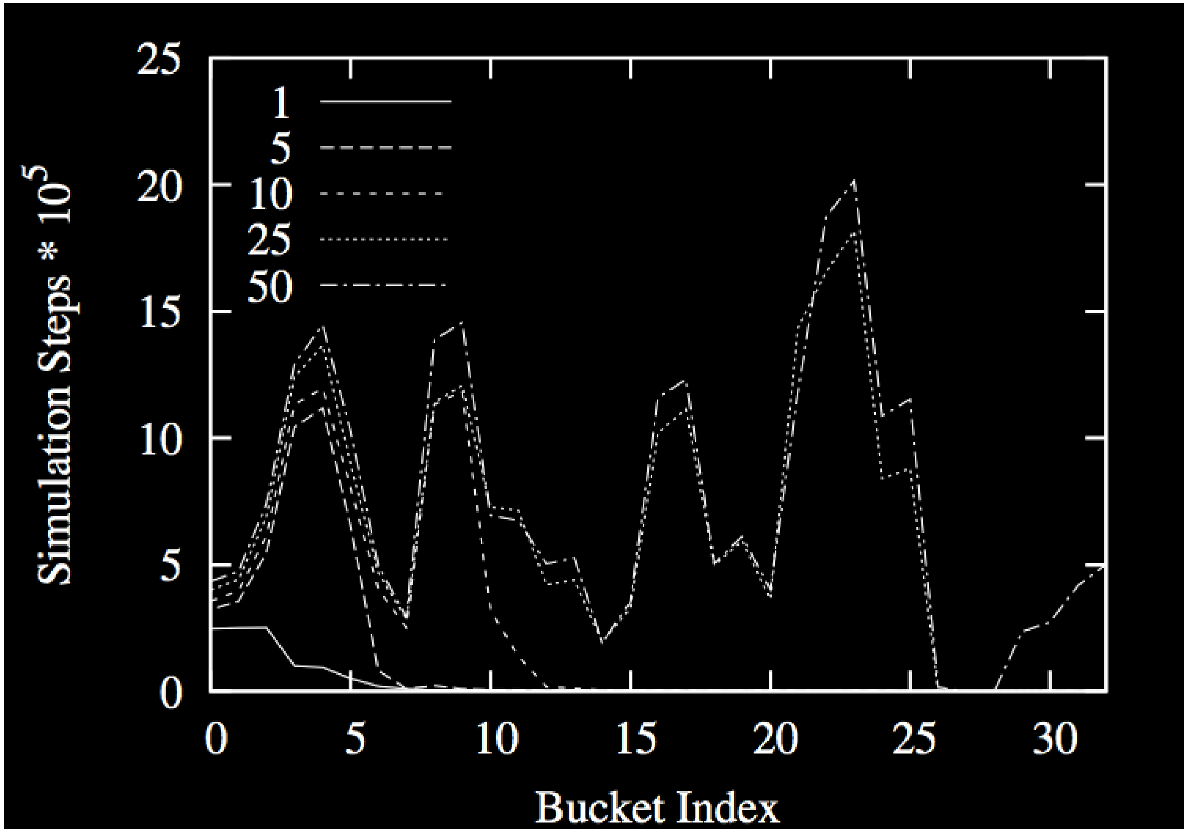
[Re'2007]



Finiding top k = O(1); finding and sorting top k = O(k)

10 million probabilistic tuples; DB2

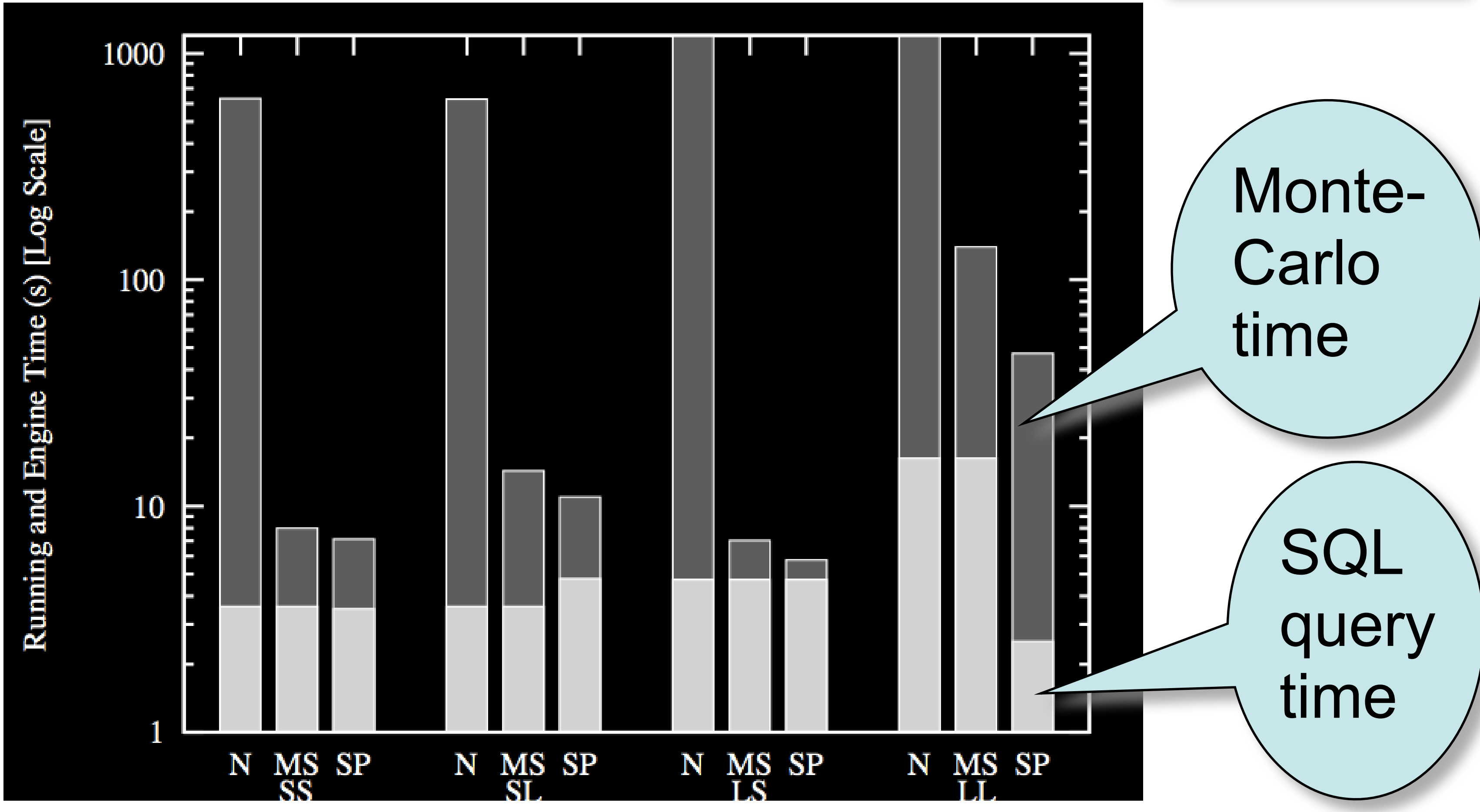
[Re'2007]



Simulation steps are concentrated in the top ≈ k buckets

10 million probabilistic tuples; DB2

[Re'2007]



N =naïve (simulate all),
MS = top-k multisimulation,
SP = adds safe-plan optimization

Times in Seconds
(logarithmic scale !)

Summary of Implementation and Systems

- General-purpose inference algorithms
  - Several available, but slooooow !!
  - Run outside the RDBMS
- Optimization 1: push some of the probability inference in the engine through "safe plans"
- Optimization 2: exploit the fact that uses want top-k answers only

Outline

Part 1:

- Motivation
- Data model
- Basic query evaluation

Part 2:

- The dichotomy of query evaluation
- Implementation and optimization
- Six Challenges

[Re'2007,Re'2007b]

1. Query Optimization

Even a #P-hard query often has subqueries that are in PTIME. Needed:

- Combine safe plans + probabilistic inference
- "Interesting indepence/disjointness"
- Model a probabilistic engine as black-box

CHALLENGE 1: Integrate a black-box probabilistic inference in a query processor.

2. Probabilistic Inference

Open the box ! Logical to physical

Examine specific algorithms from KR:

- Variable elimination
- Junction trees
- Bounded treewidth

[Sen&Deshpande'2007]

[Bravo&Ramakrishnan'2007]

CHALLENGE 2: (1) Study the space of optimization alternatives. (2) Estimate the cost of specific probabilistic inference algorithms.

3. Open Theory Problems

- Self-joins are much harder to study
  – Solved only for independent tuples

[D&S'2007]

- Extend to richer query language
  – Unions, predicates (< , ≤, ≠), aggregates
- Do hardness results still hold for Pr = 1/2 ?

CHALLENGE 3: Complete the analysis of the query complexity over probabilistic databases

4. Complex Probabilistic Model

- Independent and disjoint tuples are insufficient for real applications
- Capturing complex correlations:
  - Lineage
  - Graphical models

[Das Sarma'06,Benjelloum'06]

[Getoor'06,Sen&Deshpande'07]

CHALLENGE 4: Explore the connection between complex models and views

[Verma&Pearl'1990]

[Shen'06, Andritsos'06, Richardson'06,Chaudhuri'07]

5. Constraints

Needed to clean uncertainties in the data

- Hard constraints:
  - Semantics = conditional probability
- Soft constraints:
  - What is the semantics ?

Lots of prior work, but still little understood

CHALLENGE 5: Study the impact of hard/soft constraints on query evaluation

[Evfimievski'03,Miklau&S'04,DMS'05]

6. Information Leakage

A view V should not leak information about a secret S

$P(S) \approx P(S \mid V)$

- Issues: Which prior P ? What is ≈ ?

Probability Logic:

- $U \rightarrow V$ means $P(V \mid U) \approx 1$

[Pearl'88, Adams'98]

CHALLENGE 6: Define a probability logic for reasoning about information leakage

Conclusions

- Prohibitive cost of cleaning data
- Represent uncertainties explicitly
- Need new approaches to data management

A call to arms:
The management of probabilistic data

Bibliography

[Ada98] Ernest Adams. A Primer of Probability Logic. CSLI Publications, Stanford, California, 1998.

[AFM06] P. Andritsos, A. Fuxman, and R. J. Miller. Clean answers over dirty databases. In ICDE, 2006.

[AGK06] A. Arasu, V. Ganti, and R. Kaushik. Efficient exact set-similarity joins. In VLDB, pages 918–929, 2006.

[AKO07a] L. Antova, C. Koch, and D. Olteanu. 10^(10^6) worlds and beyond: Efficient representation and processing of incomplete information. In ICDE, 2007.

[AKO07b] L. Antova, C. Koch, and D. Olteanu. World-set decompositions: Expressiveness and efficient algorithms. In ICDT, pages 194–208, 2007.

Bibliography

[AS06] S. Abiteboul and P. Senellart. Querying and updating probabilistic information in XML. In EDBT, pages 1059–1068, 2006.

[BDJ+06] D. Burdick, P. Deshpande, T. S. Jayram, R. Ramakrishnan, and S. Vaithyanathan. Efficient allocation algorithms for olap over imprecise data. In VLDB, pages 391–402, 2006.

[BDSHW06] O. Benjelloun, A. Das Sarma, A. Halevy, and J. Widom. ULDBs: Databases with uncertainty and lineage. In VLDB, pages 953–964, 2006.

[BGHK96] F. Bacchus, A. Grove, J. Halpern, and D. Koller. From statistical knowledge bases to degrees of belief. Artificial Intelligence, 87(1- 2):75–143, 1996.

Bibliography

[BGMP92] D. Barbara, H. Garcia-Molina, and D. Porter. The management ofprobabilistic data. IEEE Trans. Knowl. Data Eng., 4(5):487–502, 1992.

[BZ06] G. Borriello and F. Zhao. World-Wide Sensor Web: 2006 UWMSR Summer Institute Semiahmoo Resort, Blaine, WA, 2006. www.cs.washington.edu/mssi/2006/schedule.html.

[CDLS99] R. Cowell, P. Dawid, S. Lauritzen, and D. Spiegelhalter, editors. Probabilistic Networks and Expert Systems. Springer, 1999.

[Coo90] G. Cooper. Computational complexity of probabilistic inference using bayesian belief networks (research note). Artificial Intelligence, 42:393–405, 1990.

Bibliography

[CPWL06] T. Choudhury, M. Philipose, D. Wyatt, and J. Lester. Towards activity databases: Using sensors and statistical models to summarize people's lives. IEEE Data Eng. Bull, 29(1):49–58, March 2006.

[CRF03] W. Cohen, P. Ravikumar, and S. Fienberg. A comparison of string distance metrics for name-matching tasks. In IIWeb, pages 73–78, 2003.

[Dal07] Nilesh Dalvi. Query evaluation on a database given by a random graph. In ICDT, pages 149–163, 2007. 20

[Dar03] Adnan Darwiche. A differential approach to inference in bayesian networks. Journal of the ACM, 50(3):280–305, 2003.

[DGM+04] A. Deshpande, C. Guestrin, S. Madden, J. M. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In VLDB,

Bibliography

[DGM+05] A. Deshpande, C. Guestrin, S. Madden, J. M. Hellerstein, and W. Hong. Using probabilistic models for data management in acquisitional environments. In CIDR, pages 317–328, 2005.

[DGR01] A. Deshpande, M. Garofalakis, and R. Rastogi. Independence is good: Dependency-based histogram synopses for high-dimensional data. In SIGMOD, pages 199–210, 2001.

[DL93] P. Dagum and M. Luby. Approximating probabilistic inference in bayesian belief networks is NP-hard. Artificial Intelligence, 60:141–153, 1993.

[DMS05] N. Dalvi, G. Miklau, and D. Suciu. Asymptotic conditional probabilities for conjunctive queries. In ICDT, 2005.

[dR95] Michel de Rougemont. The reliability of queries. In PODS, pages 286–291, 1995.

Bibliography

[DRC+06] A. Doan, R. Ramakrishnan, F. Chen, P. DeRose, Y. Lee, R. McCann, M. Sayyadian, and W. Shen. Community information management. IEEE Data Engineering Bulletin, Special Issue on Probabilistic Data Management, 29(1):64–72, March 2006.

[DRS06] N. Dalvi, Chris Re, and D. Suciu. Query evaluation on probabilistic databases. IEEE Data Engineering Bulletin, 29(1):25–31, 2006.

[DS04] N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. In VLDB, Toronto, Canada, 2004.

[DS05] N. Dalvi and D. Suciu. Answering queries from statistics and probabilistic views. In VLDB, 2005.

[DS07a] N. Dalvi and D. Suciu. The dichotomy of conjunctive queries on probabilistic structures. In PODS, pages 293–302, 2007.

Bibliography

[DS07b] N. Dalvi and D. Suciu. Management of probabilistic data: Foundations and challenges. In PODS, pages 1–12, Beijing, China, 2007. (invited talk).

[DSBHW06] A. Das Sarma, O. Benjelloun, A. Halevy, and J. Widom. Working models for uncertain data. In ICDE, 2006.

[ea07] M. Balazinska et al. Data management in the world-wide sensor web. IEEE Pervasive Computing, 2007. To appear.

[FHM05] M. Franklin, A. Halevy, and D. Maier. From databases to dataspaces: a new abstraction for information management. SIGMOD Record, 34(4):27–33, 2005.

Bibliography

[FR97] Norbert Fuhr and Thomas Roelleke. A probabilistic relational algebra for the integration of information retrieval and database systems. ACM Trans. Inf. Syst., 15(1):32–66, 1997.

[FS69] Ivan Felligi and Alan Sunter. A theory for record linkage. Journal of the American Statistical Society, 64:1183–1210, 1969.

[Get06] Lise Getoor. An introduction to probabilistic graphical models for relational data. IEEE Data Engineering Bulletin, Special Issue on Probabilistic Data Management, 29(1):32–40, March 2006.

[GGH98] E. Gr¨adel, Y. Gurevich, and C. Hirsch. The complexity of query reliability. In PODS, pages 227–234, 1998.

Bibliography

[GHR95] R. Greenlaw, J. Hoover, and W. Ruzzo. Limits to Parallel Computation. P-Completeness Theory. Oxford University Press, New York, Oxford, 1995.

[GS06a] Minos Garofalakis and Dan Suciu. Special issue on probabilistic data management. IEEE Data Engineering Bulletin, pages 1–72, 2006.

[GS06b] R. Gupta and S. Sarawagi. Creating probabilistic databases from information extraction models. In VLDB, pages 965–976, 2006.

[GT06] T. Green and V. Tannen. Models for incomplete and probabilistic information. IEEE Data Engineering Bulletin, 29(1):17–24, March 2006.

[Hal06] J. Halpern. From statistical knowledge bases to degrees of belief: an overview. In PODS, pages 110–113, 2006. 22

Bibliography

[Hec02] D. Heckerman. Tutorial on graphical models, June 2002.

[HFM06] A. Halevy, M. Franklin, and D. Maier. Principles of dataspace systems. In PODS, pages 1–9, 2006.

[HGS03] E. Hung, L. Getoor, and V.S. Subrahmanian. PXML: A probabilistic semistructured data model and algebra. In ICDE, 2003.

[HRO06] A. Halevy, A. Rajaraman, and J. Ordille. Data integration: The teenage years. In VLDB, pages 9–16, 2006.

[IMH+04] I.F. Ilyas, V. Markl, P.J. Haas, P. Brown, and A. Aboulnaga. Cords: Automatic discovery of correlations and soft functional dependencies. In SIGMOD, pages 647–658, 2004.

Bibliography

[JGF06] S. Jeffery, M. Garofalakis, and M. Franklin. Adaptive cleaning for RFID data streams. In VLDB, pages 163–174, 2006.

[JKR+06] T.S. Jayram, R. Krishnamurthy, S. Raghavan, S. Vaithyanathan, and H. Zhu. Avatar information extraction system. IEEE Data Engineering Bulletin, 29(1):40–48, 2006.

[JKV07] T.S. Jayram, S. Kale, and E. Vee. Efficient aggregation algorithms for probabilistic data. In SODA, 2007.

[KBS06] N. Khoussainova, M. Balazinska, and D. Suciu. Towards correcting input data errors probabilistically using integrity constraints. In MobiDB, pages 43–50, 2006.

[KL83] R. Karp and M. Luby. Monte-Carlo algorithms for enumeration and reliability problems. In Proceedings of the annual ACM symposium on Theory of computing, 1983.

Bibliography

[Kol] D. Koller. Representation, reasoning, learning. Computers and Thought 2001 Award talk.

[Kol05] P. Kolaitis. Schema mappings, data exchange, and metadata management. In PODS, pages 61–75, 2005.

[LCK+05] J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford. A hybrid discriminative/generative approach for modeling human activities. In IJCAI, pages 766–772, 2005.

[LLRS97] L. Lakshmanan, N. Leone, R. Ross, and V.S. Subrahmanian. Probview: A flexible probabilistic database system. ACM Trans. Database Syst., 22(3), 1997.

[MCD+07] J. Madhavan, S. Cohen, X. Dong, A. Halevy, S. Jeffery, D. Ko, and C. Yu. Web-scale data integration: You can afford to pay as you go. In CIDR, pages 342–350, 2007.

Bibliography

[MS04] G. Miklau and D. Suciu. A formal analysis of information disclosure in data exchange. In SIGMOD, 2004.

[PB83] J. S. Provan and M. O. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. SIAM J. Comput., 12(4):777–788, 1983.

[Pea88] Judea Pearl. Probabilistic reasoning in intelligent systems. Morgan Kaufmann, 1988.

[RD07a] C. Re and D.Suciu. Efficient evaluation of having queries on a probabilistic database. In Proceedings of DBPL, 2007.

[RD07b] C. Re and D.Suciu. Materialized views in probabilistic databases for information exchange and query optimization. In Proceedings of VLDB, 2007.

Bibliography

[RDS07] C. Re, N. Dalvi, and D. Suciu. Efficient Top-k query evaluation on probabilistic data. In ICDE, 2007.

[RSG05] R. Ross, V.S. Subrahmanian, and J. Grant. Aggregate operators in probabilistic databases. JACM, 52(1), 2005.

[Sar] Sunita Sarawagi. Automation in information extraction and data integration. Tutorial presented at VLDB'2002.

[SD07] Prithviraj Sen and Amol Deshpande. Representing and querying correlated tuples in probabilistic databases. In ICDE, 2007.

[SLD05] W. Shen, X. Li, and A. Doan. Constraint-based entity matching. In AAAI, pages 862–867, 2005.

Bibliography

[Val79] L. Valiant. The complexity of enumeration and reliability problems. SIAM J. Comput., 8:410–421, 1979.

[vKdKA05] M. van Keulen, A. de Keijzer, and W. Alink. A probabilistic XML approach to data integration. In ICDE, pages 459–470, 2005.

[VP90] T. Verma and J. Pearl. Causal networks: Semantics and expressiveness. Uncertainty in Artificial Intelligence, 4:69–76, 1990.

[Win99] William Winkler. The state of record linkage and current research problems. Technical report, Statistical Research Division, U.S. Bureau of the Census, 1999.

[ZD06] Y. Zabiyaka and A. Darwiche. Functional treewidth: Bounding complexity in the presence of functional dependencies. In SAT, pages