

Management of Probabilistic Data: Foundations and Challenges

Nilesh Dalvi and Dan Suciu
Univerisity 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

Uncertainties Everywhere

- In the schema mappings:
 - Data spaces
 - *Pay as you go* data integration
- In the data mapping
 - Life science data integration
 - Object reconciliation, fuzzy joins
- In the data itself
 - Data “by the masses”
 - Information Extraction
 - RFID data, sensor data

[Halevy'2007]

[Philippi&Kohler'2006]

[Arasu'06]

[Gupta&Sarawagi'2006]

[Welbourne'2007]

Example 1

Data Integration in Life Sciences

- U2 integrates several biological databases

Example: find functional annotations of ABCD1

User types: "Gene ABCD1"
U2 finds 80 "related" proteins
Ranks them by *uncertainty score*
Correct 9 functions are among top 11

EntrezProtein,
Pfam,
TIGRFAM,
NCBI Blast,
EntrezGene

Need to represent uncertainties explicitly

Example 2

Information Extraction

...52 A Goregaon West Mumbai ...

[Gupta&Sarawagi'2006]

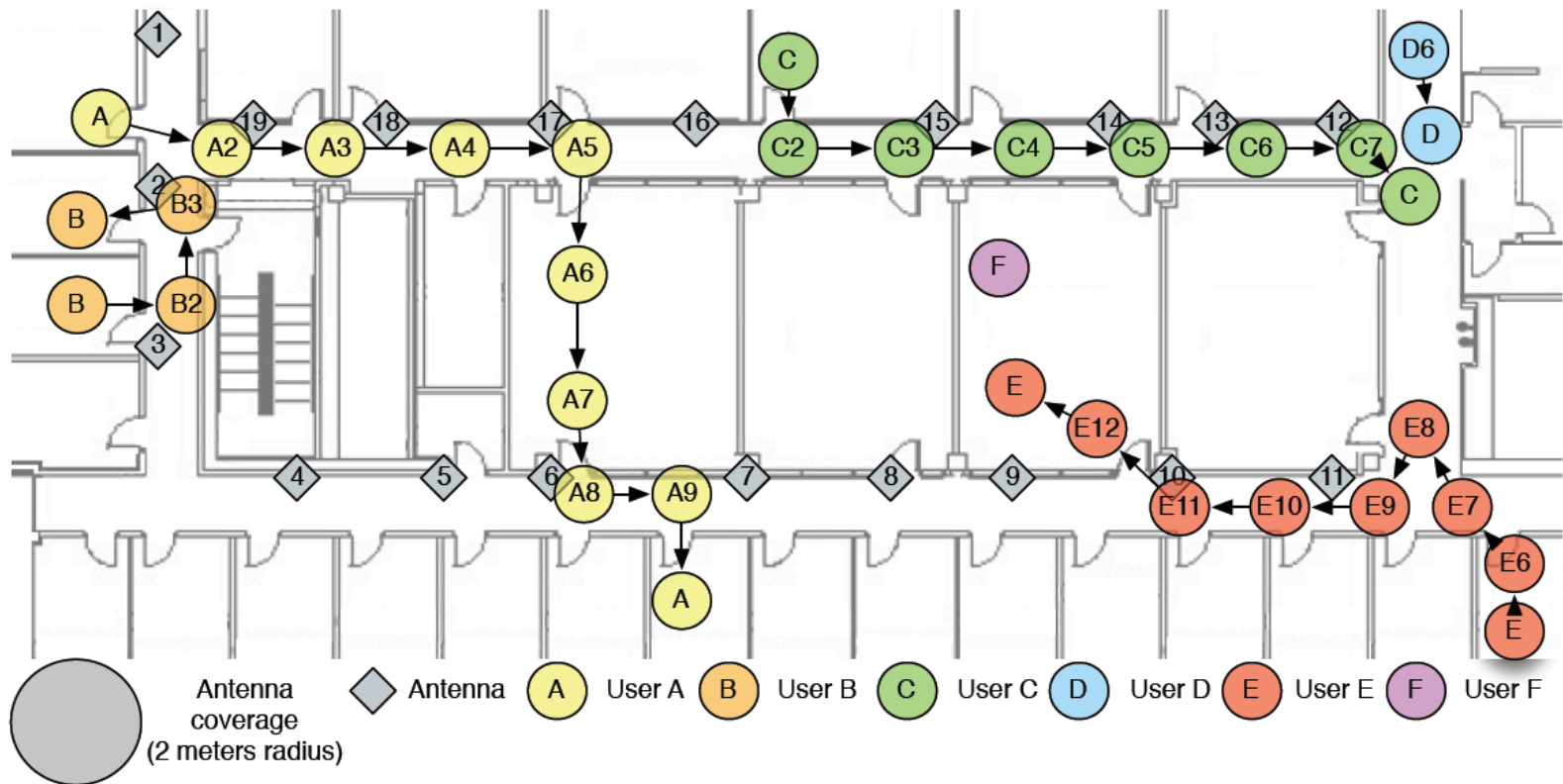
<u>ID</u>	House-No	Street	City	P
1	52	Goregaon West	Mumbai	0.1
1	52-A	Goregaon West	Mumbai	0.4
1	52	Goregaon	West Mumbai	0.2
1	52-A	Goregaon	West Mumbai	0.2
2
2			

≈20% of such extractions are correct

Here probabilities are meaningful

Example 3

RFID Ecosystem at UW



[Welbourne'2007]

- RFID data = noisy
 - SIGHTING(tagID, antennaID, time)
- Derived data = Probabilistic
 - “John entered Room 524 at 9:15” prob=0.6
 - “John carried laptop x77 at 11:03” prob=0.8
 - . . .
- Queries
 - “Which people were in Room 478 yesterday ?”

Massive amounts of probabilistic data from RFIDs, sensors

A Model for Uncertainties

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

This talk: Probabilistic Databases

Probabilistic databases: Long History

Cavallo&Pitarelli:1987

Barbara,Garcia-Molina, Porter:1992

Lakshmanan,Leone,Ross&Subrahmanian:1997

Fuhr&Roelke:1997

Dalvi&S:2004

Widom:2005

Focus today: the Query Evaluation Problem

Has this been solved by AI ?

Input: KB

Fix q
Input: DB

AI

Databases

Deterministic	Theorem prover	Query processing
Probabilistic	Probabilistic inference	[this talk]

Outline

- Data model
- Query evaluation
- Challenges

[Barbara et al.1992]

What is a Probabilistic Database (PDB) ?

HasObject^p

<u>Object</u>	<u>Time</u>	Person	P
Laptop77	9:07	John	0.62
		Jim	0.34
Book302	9:18	Mary	0.45
		John	0.33
		Fred	0.11

[Barbara et al.1992]

What is a Probabilistic Database (PDB) ?

HasObject^p

Keys

Non-keys

Probability

<u>Object</u>	<u>Time</u>	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* ? ¹³

Background

Finite probability space = (Ω, P)

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

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

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

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

<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

PDB

$\Omega = \{$

$\}$ Possible worlds

Possible Worlds Semantics

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

$\Omega = \{$

Object	Time	Person
Laptop77	9:07	John
Book302	9:18	Mary

$\}$

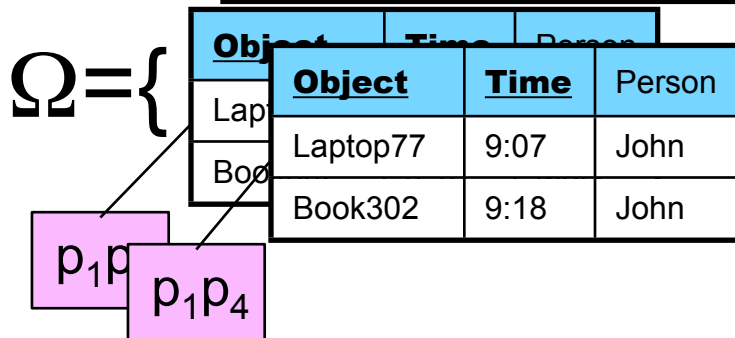
$p_1 p_3$

Possible worlds

Possible Worlds Semantics

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

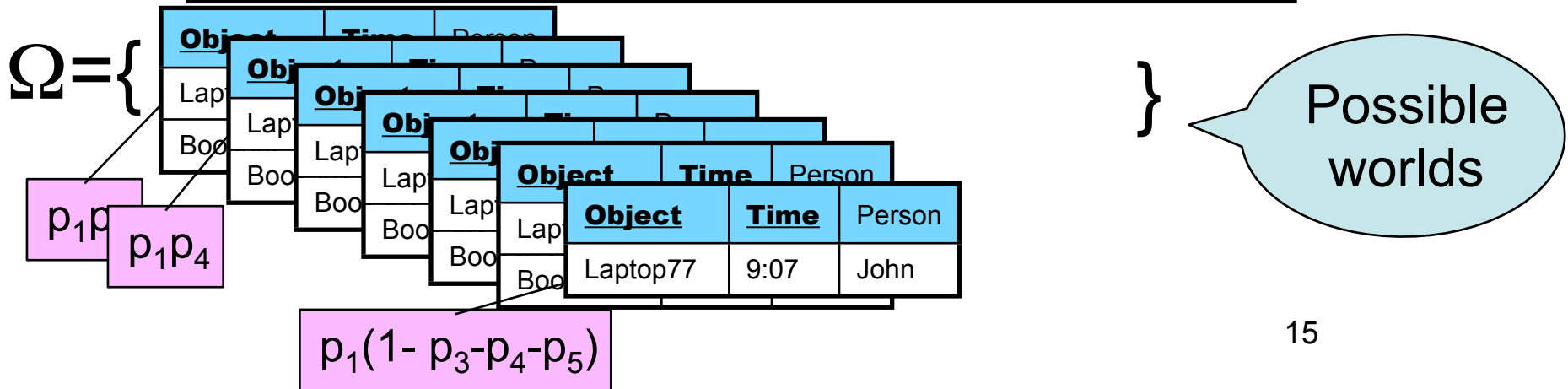


Possible worlds

Possible Worlds Semantics

<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

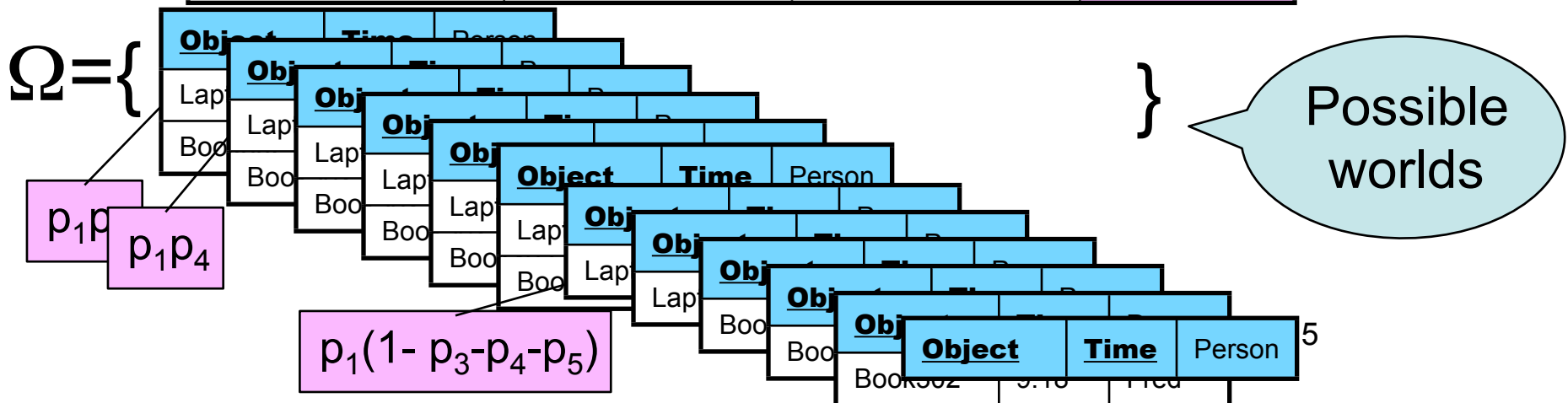
PDB



Possible Worlds Semantics

<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

PDB



Definitions

Definition: A tuple-disjoint/independent table is:

$R(\underline{\mathbf{A1}}, \underline{\mathbf{A2}}, \dots, \underline{\mathbf{Am}}, B1, \dots, Bn, P)$

Definition: A tuple-independent table is:

$R(\underline{\mathbf{A1}}, \underline{\mathbf{A2}}, \dots, \underline{\mathbf{Am}}, P)$

Definition: Semantics is given by possible worlds

HasObject(**Object**, **Time**, 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
Independent

Meets(**Person1**, **Person2**, **Time**, P)

<u>Person1</u>	<u>Person2</u>	<u>Time</u>	P
John	Jim	9:12	p ₁
Mary	Sue	9:20	p ₂
John	Mary	9:20	p ₃

Independent
17

Query Semantics

A boolean query q is an event: $\{\omega \mid \omega \models q\}$

$$P(q) = \sum_{\omega \models q} P(\omega)$$

Did someone take MyBook to the CoffeeRoom ?

$q =$

`HasObject('MyBook',x,t), EnterRoom(x,'CoffeeRoom',t)`



$$P(q) = 0.96$$

(meaning: quite likely !)

Discussion of Data Model

Tuple-disjoint/independent tables:

- Simple model, can store in any DBMS

More advanced models:

- Symbolic boolean expressions

Fuhr and Roellke

- Trio: add lineage

[Widom05, Das Sarma'06, Benjelloun 06]

- Probabilistic Relational Models

[Getoor'2006]

- Graphical models

[Sen&Desphande'07]

Outline

- Data model
- Query evaluation
 - Probability of Boolean expressions
 - From queries to Boolean expressions
 - Data complexity of query evaluation
- Challenges

Probability of Boolean Expressions

$$\Phi = X_1X_2 \vee X_1X_3 \vee X_2X_3$$

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

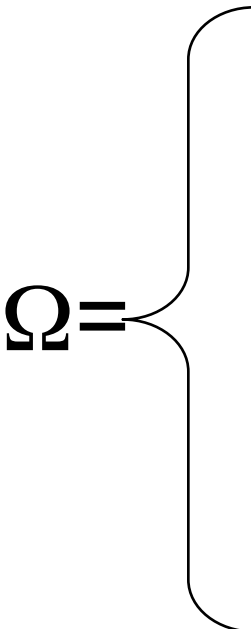
Compute $P(\Phi)$

Probability of Boolean Expressions

$$\Phi = X_1X_2 \vee X_1X_3 \vee X_2X_3$$

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

Compute $P(\Phi)$



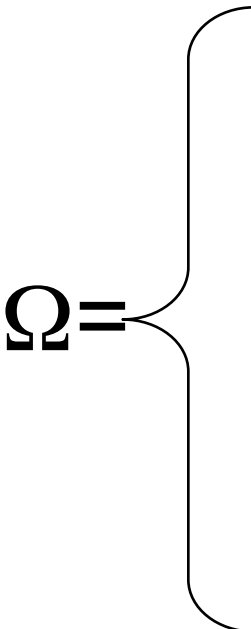
X_1	X_2	X_3	P	Φ
0	0	0		0
0	0	1		0
0	1	0		0
0	1	1	$(1-p_1)p_2p_3$	1
1	0	0		0
1	0	1	$p_1(1-p_2)p_3$	1
1	1	0	$p_1p_2(1-p_3)$	1
1	1	1	$p_1p_2p_3$	1

Probability of Boolean Expressions

$$\Phi = X_1X_2 \vee X_1X_3 \vee X_2X_3$$

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

Compute $P(\Phi)$



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

$$\begin{aligned} \Pr(\Phi) = & (1-p_1)p_2p_3 + \\ & p_1(1-p_2)p_3 + \\ & p_1p_2(1-p_3) + \\ & p_1p_2p_3 \end{aligned}$$

Background

Fix $P(X_1) = P(X_2) = \dots = P(X_n) = 1/2$

Theorem

Exact evaluation of $\Pr(\Phi)$ is #P-complete

[Valiant:1979]

Theorem For DNF Φ

Approximation of $\Pr(\Phi)$ is in PTIME
(FPTRAS)

[Karp&Luby:1983]

Both theorems extend to rational $P(X_1), \dots, P(X_n)$

[Graedel, Gurevitch, Hirsch:1998]

Query q + Database PDB $\rightarrow \Phi$

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

PDB =

R^p		
<u>A</u>	<u>B</u>	P
a_1	b_1	p_1
a_2	b_2	p_2

X_1
 X_2

S^p		
<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

Y_1
 Y_2
 Y_3
 Y_4
 Y_5

23

Query q + Database PDB $\rightarrow \Phi$

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

PDB =

R^p		
<u>A</u>	<u>B</u>	P
a_1	b_1	p_1
a_2	b_2	p_2

X_1
 X_2

S^p

<u>A</u>	<u>C</u>	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



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

Application to Query Evaluation

Corollary Fix FO query q
Exact evaluation of $\text{Pr}(q)$ on input PDB is in $\#P$

Corollary Fix a conjunctive query q .
Approximation of $\text{Pr}(q)$ on input PDB is in PTIME
(FPTRAS)

[Graedel, Gurevitch, Hirsch: 1998]

Background: Probabilistic Networks

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

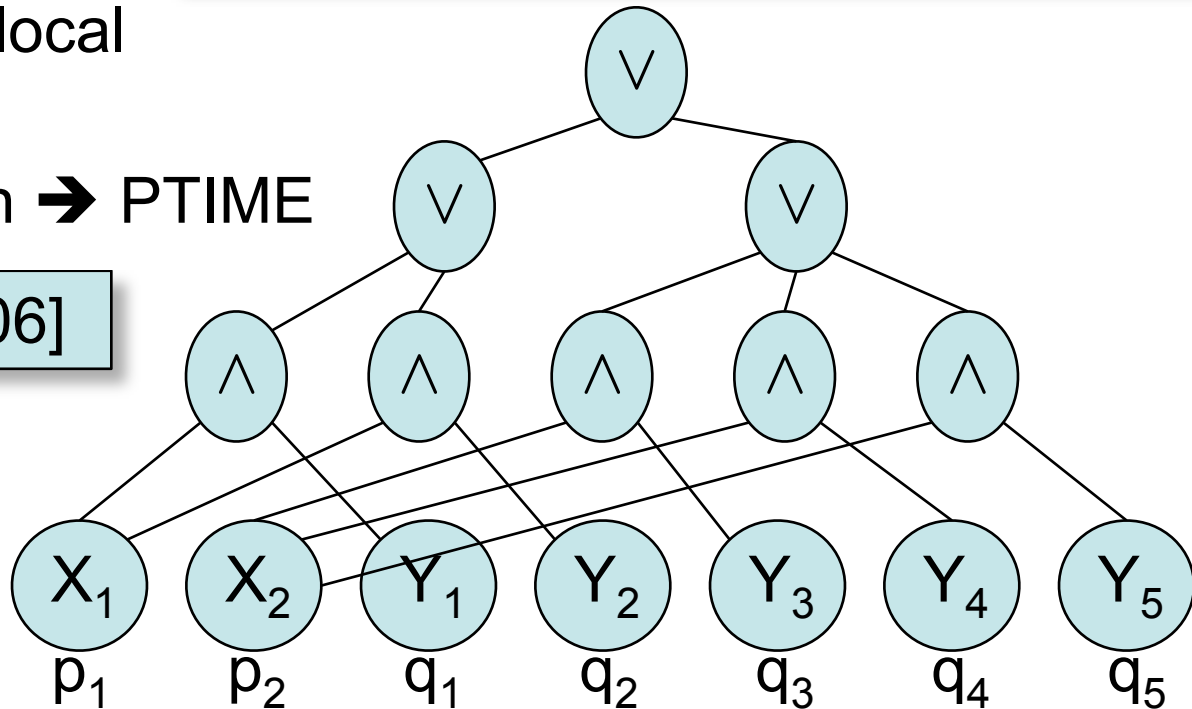
Inference: hard in general
KR techniques exploit local
properties:

$$\Phi = X_1 Y_1 \vee X_1 Y_2 \vee X_2 Y_3 \vee X_2 Y_4 \vee X_2 Y_5$$

E.g. bounded treewidth \rightarrow PTIME

[Zabiyaka&Darwiche'06]

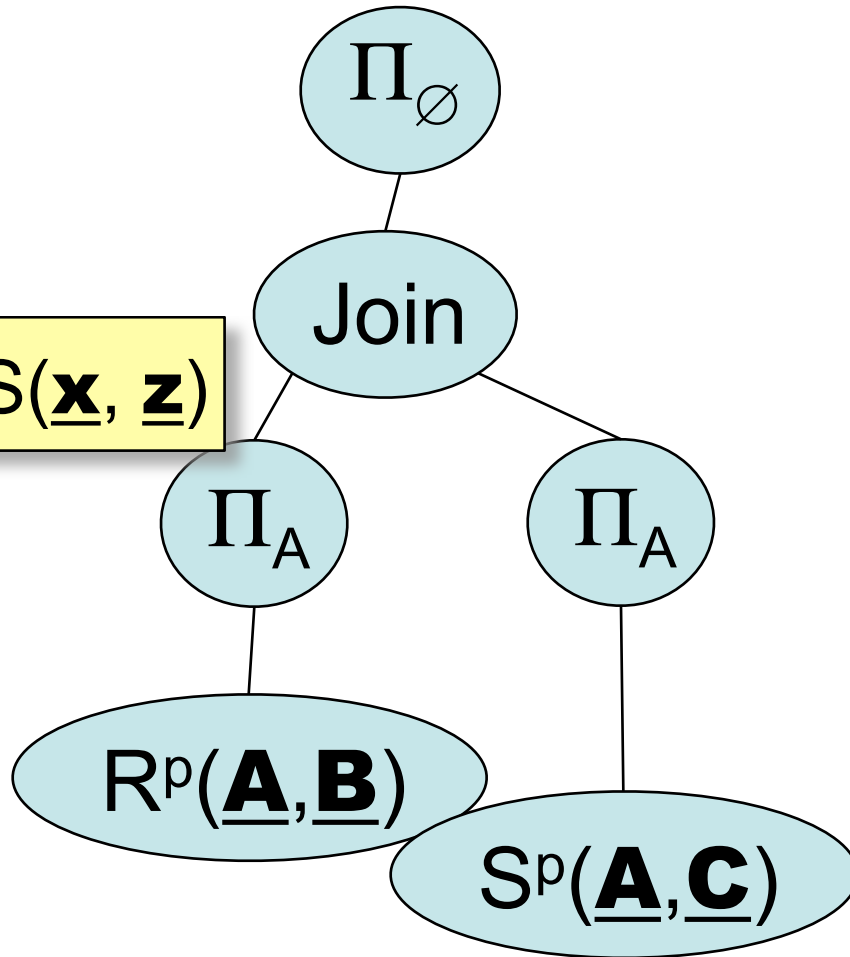
Note: for this query
the treewidth is
unbounded



[D&S'2004]

q =

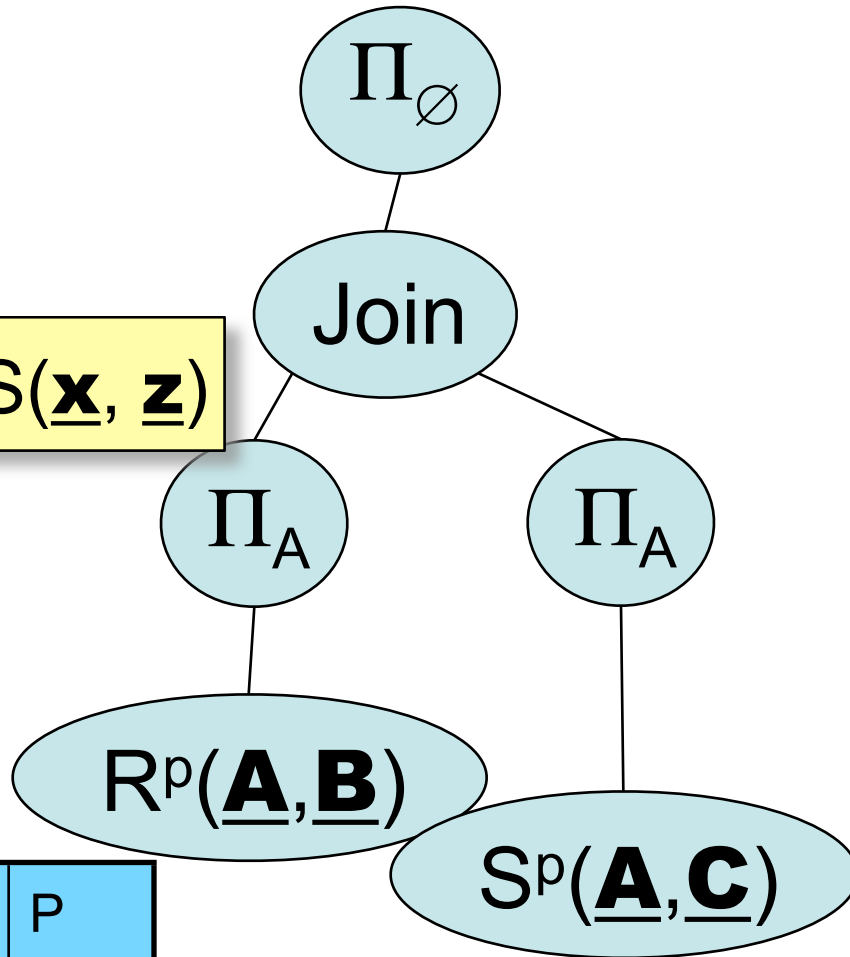
$R(\underline{x}, \underline{y}), S(\underline{x}, \underline{z})$



[D&S'2004]

q =

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



<u>A</u>	<u>B</u>	P
a ₁	b ₁	p ₁
a ₂	b ₂	p ₂

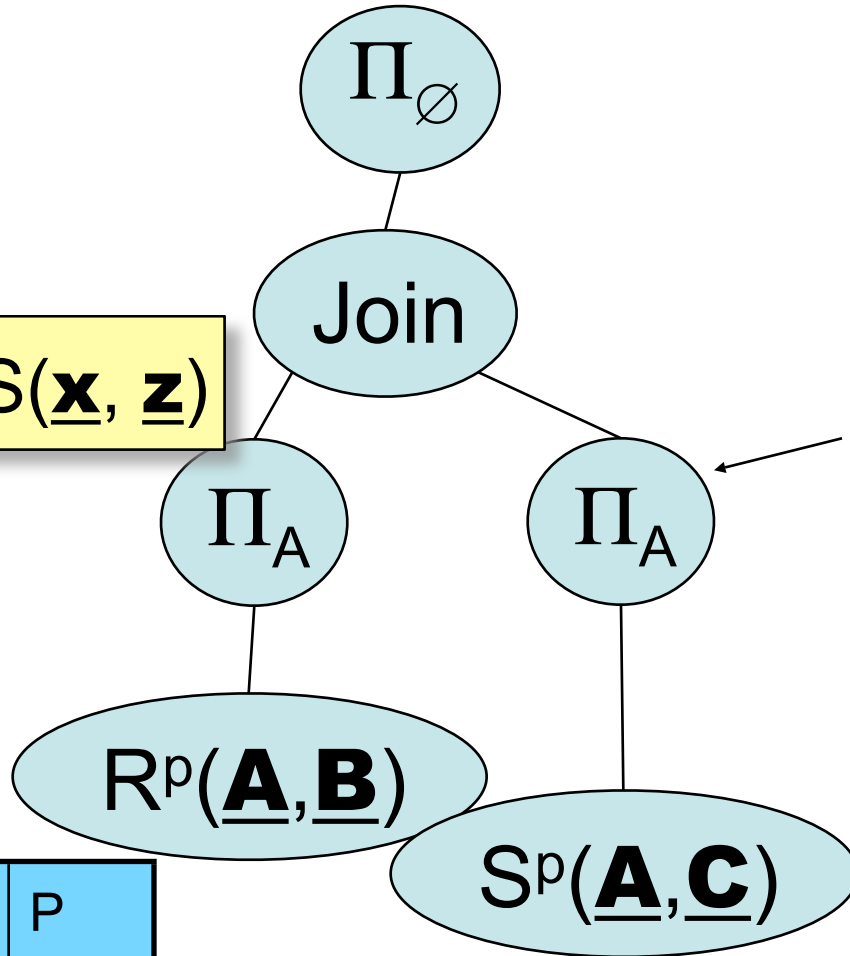
<u>A</u>	<u>C</u>	P
a ₁	c ₁	q ₁
a ₁	c ₂	q ₂
a ₂	c ₃	q ₃
a ₂	c ₄	q ₄
a ₂	c ₅	q ₅

26

[D&S'2004]

q =

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



<u>A</u>	<u>B</u>	P
a ₁	b ₁	p ₁
a ₂	b ₂	p ₂

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

26

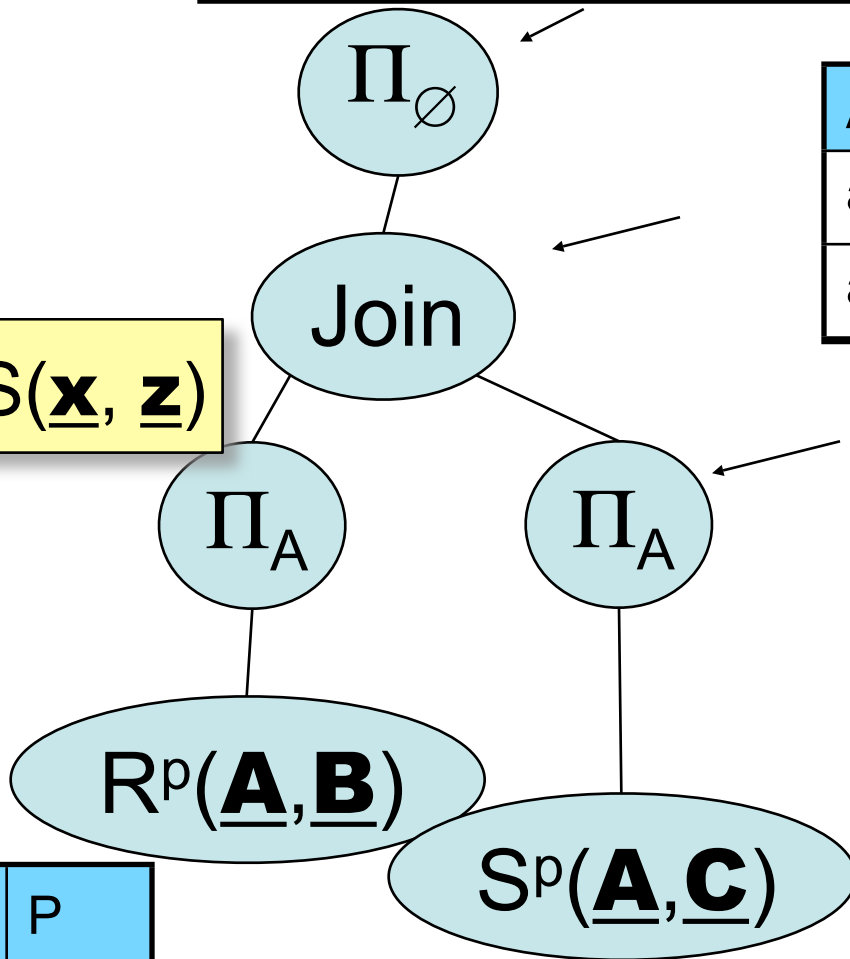
[D&S'2004]

$P(q) =$

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

$q =$

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



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

<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

[D&S'2004]

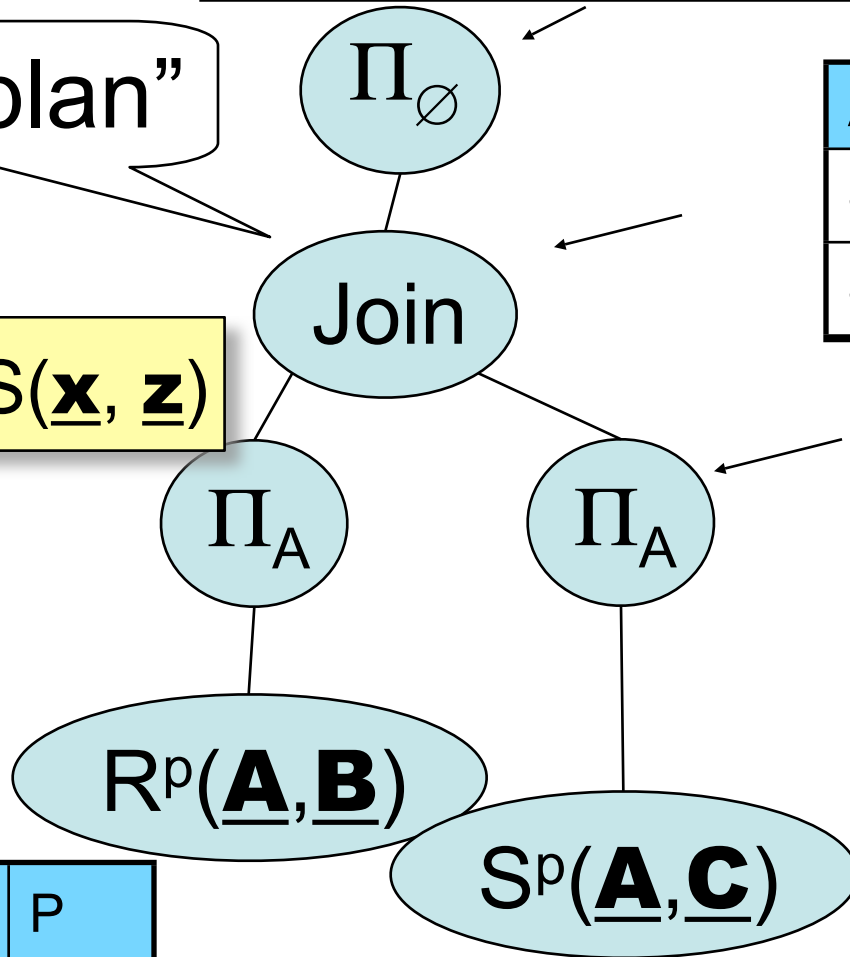
$$P(q) =$$

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

“safe plan”

q =

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



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

<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

The data complexity of this query is PTIME

Dichotomy Theorem

Let q be a conjunctive query without self-joins

Theorem One of the following holds:

- (1) Either q is in PTIME
- (2) Or q is #P hard

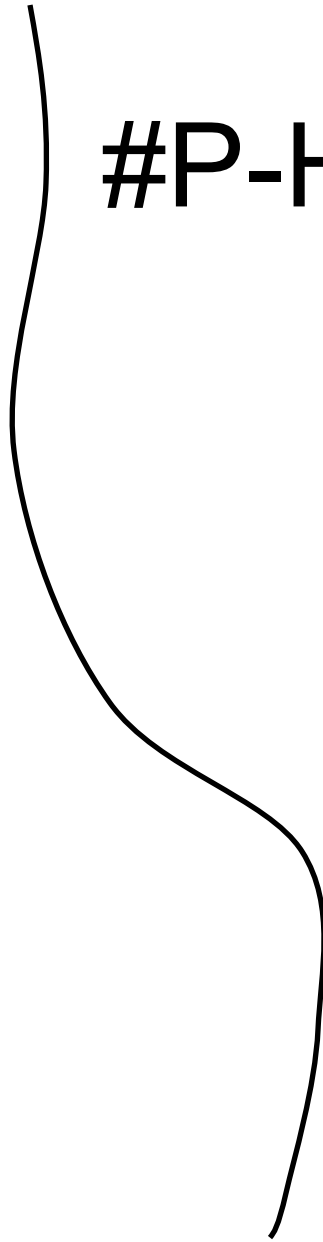
[D&S'2004]

In Case (1) q can be computed by a “*safe plan*” and we call it a “*safe query*”

[Andritsos et al'2006]

PTIME Queries

#P-Hard Queries



PTIME Queries

#P-Hard Queries

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

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

$R(\underline{\mathbf{x}}), S(\underline{\mathbf{x}}, \underline{\mathbf{y}}), T(\underline{\mathbf{y}}), U(\underline{\mathbf{u}}, y), W(\underline{\mathbf{a}}, u)$

• • •

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

$h2 = R(\underline{\mathbf{x}}, y), S(\underline{\mathbf{y}})$

$h3 = R(\underline{\mathbf{x}}, y), S(x, \underline{\mathbf{y}})$

• • •

How do we decide if a query is in PTIME or #P hard ?

Hierarchical Queries

$sg(x)$ = set of subgoals containing the variable x in key positions

Definition A query q is *hierarchical* if for all 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 Queries

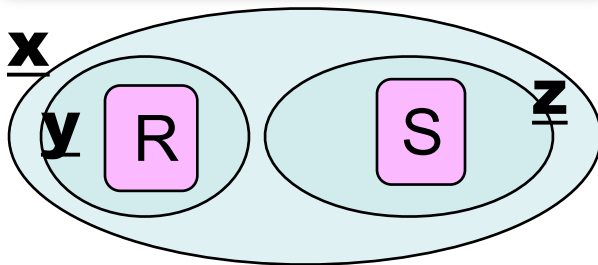
$sg(x)$ = set of subgoals containing the variable x in key positions

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

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

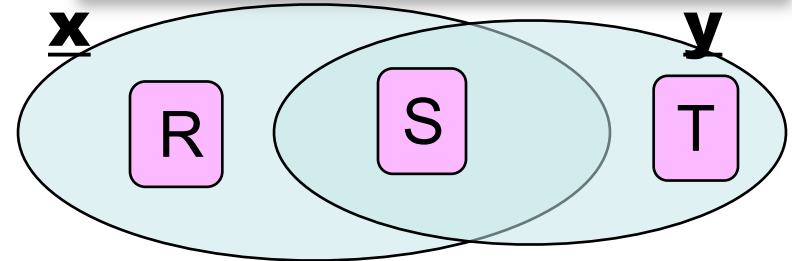
Hierarchical

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



Non-hierarchical

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



[D&S'2004]

Case 1: Independent Tuples Only

PTIME Queries:

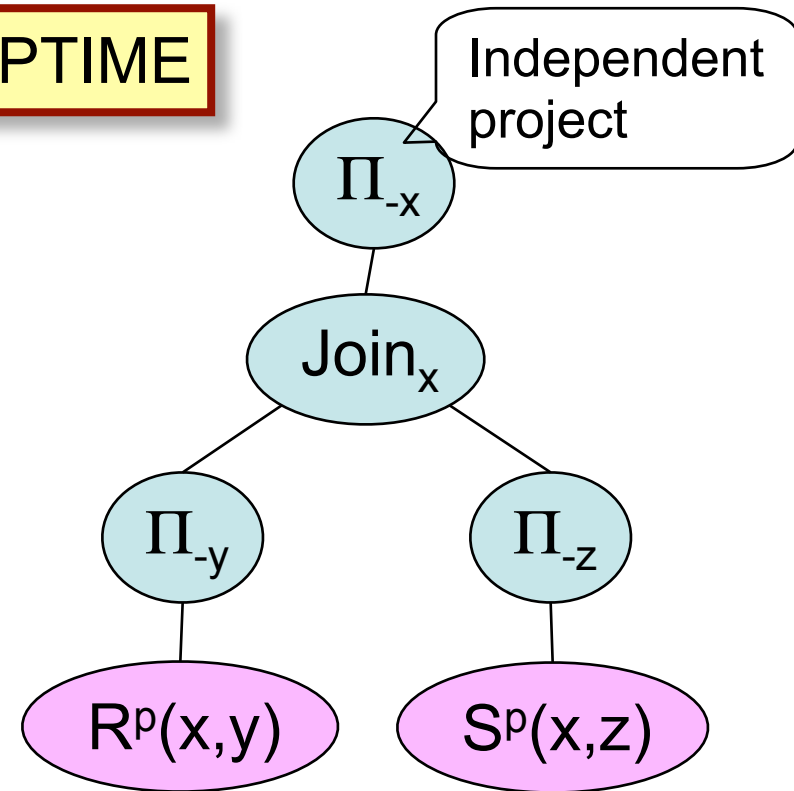
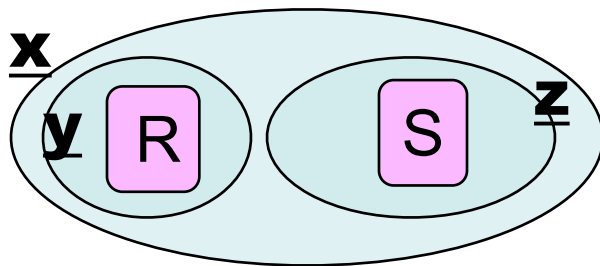
Fact If q is hierarchical then q is in PTIME

Case 1: Independent Triples Only

PTIME Queries:

Fact If q is hierarchical then q is in PTIME

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



The hierarchy gives the safe plan !

1. Root variable $u \rightarrow \Pi_{-u}$
2. Connected components \rightarrow Join

[D&S'2004]

Case 1: Independent Tuples Only

#P-hard Queries:

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

$h1$ is #P-hard (reduction from Partitioned Positive 2DNF)

[Provan&Ball'83]

Fact If q is non-hierarchical then it is #P-hard.

Proof: it “contains” $h1$:

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

[D&S'2004]

Case 1: Independent Tuples Only

#P-hard Queries:

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

$h1$ is #P-hard (reduction from Partitioned Positive 2DNF)

[Provan&Ball'83]

Fact If q is non-hierarchical then it is #P-hard.

Proof: it “contains” $h1$:

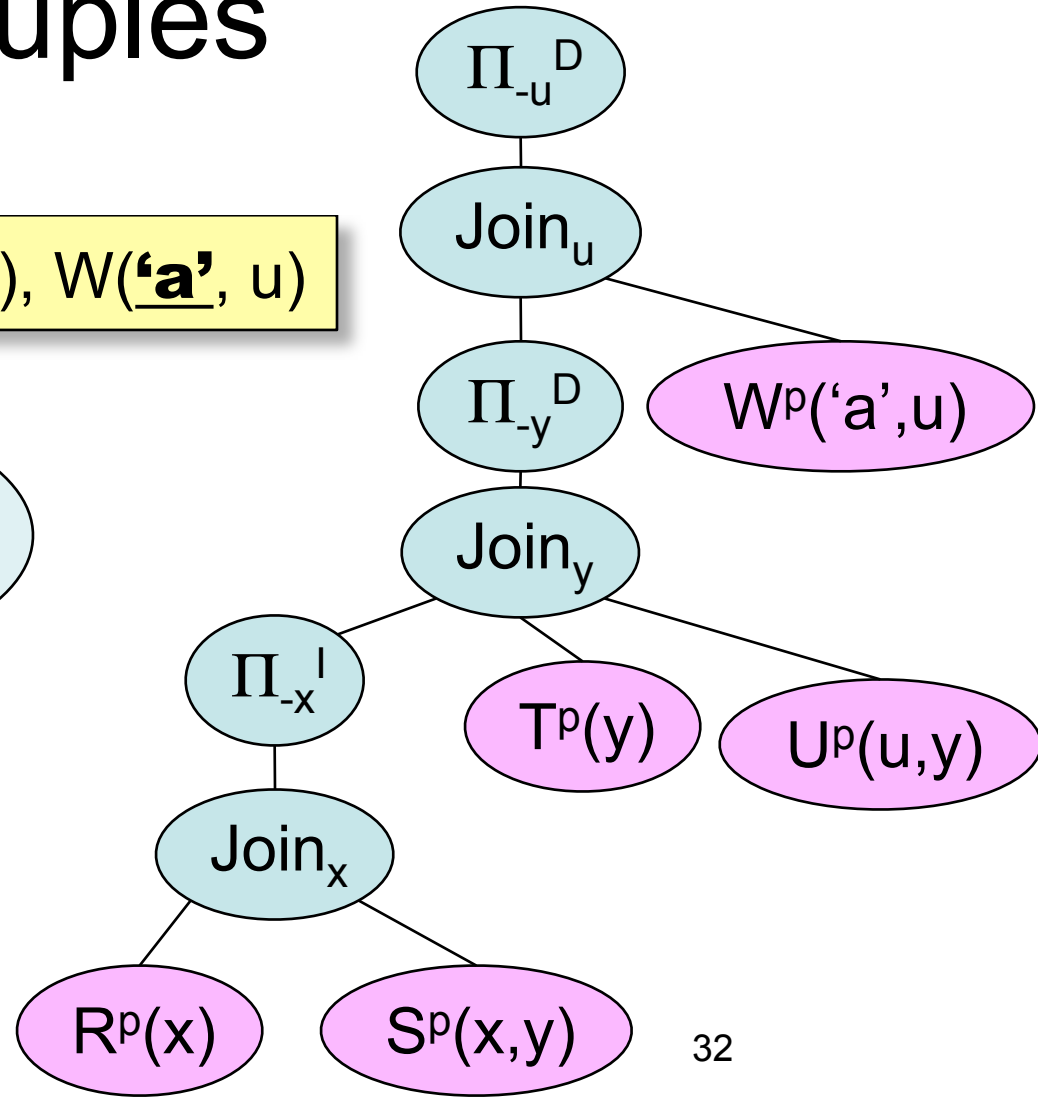
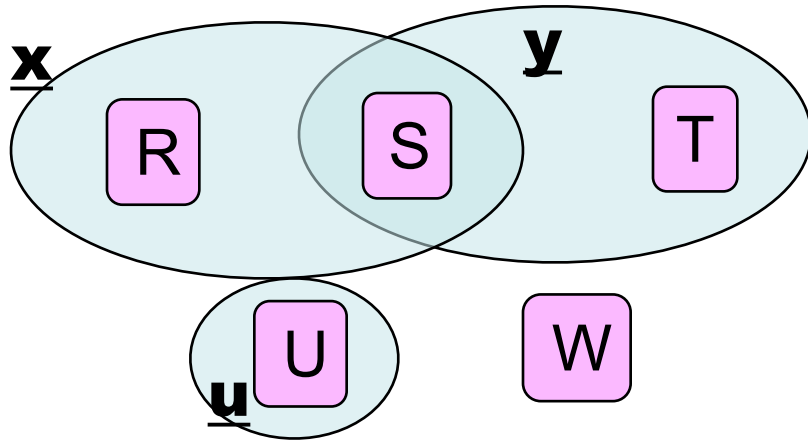
$q = \dots R(\underline{\mathbf{x}}, \dots), S(\underline{\mathbf{x}}, \underline{\mathbf{y}}, \dots), T(\underline{\mathbf{y}}, \dots) \dots$

Theorem Testing if q is PTIME or #P-hard is in AC^0

Case 2: Independent/disjoint Tuples

PTIME Queries:

$R(\underline{x}), S(\underline{x}, \underline{y}), T(\underline{y}), U(\underline{u}, y), W(\underline{a}, u)$

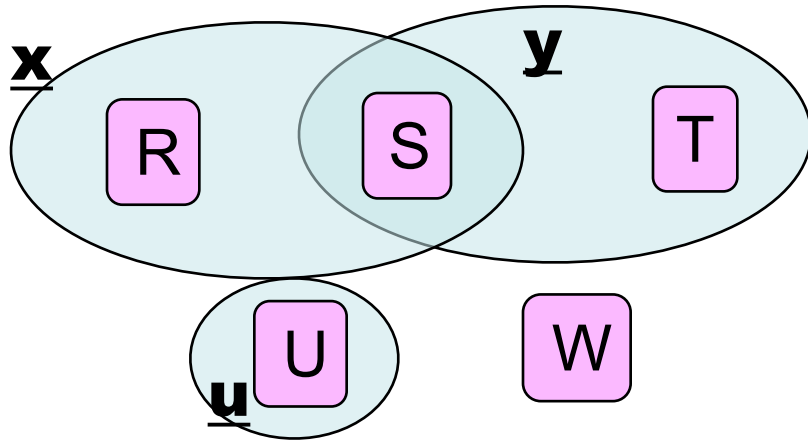


1. Root variable $\rightarrow \Pi^I$
2. CC's $\rightarrow \text{Join}$
3. Constant key attrs $\rightarrow \Pi^D$

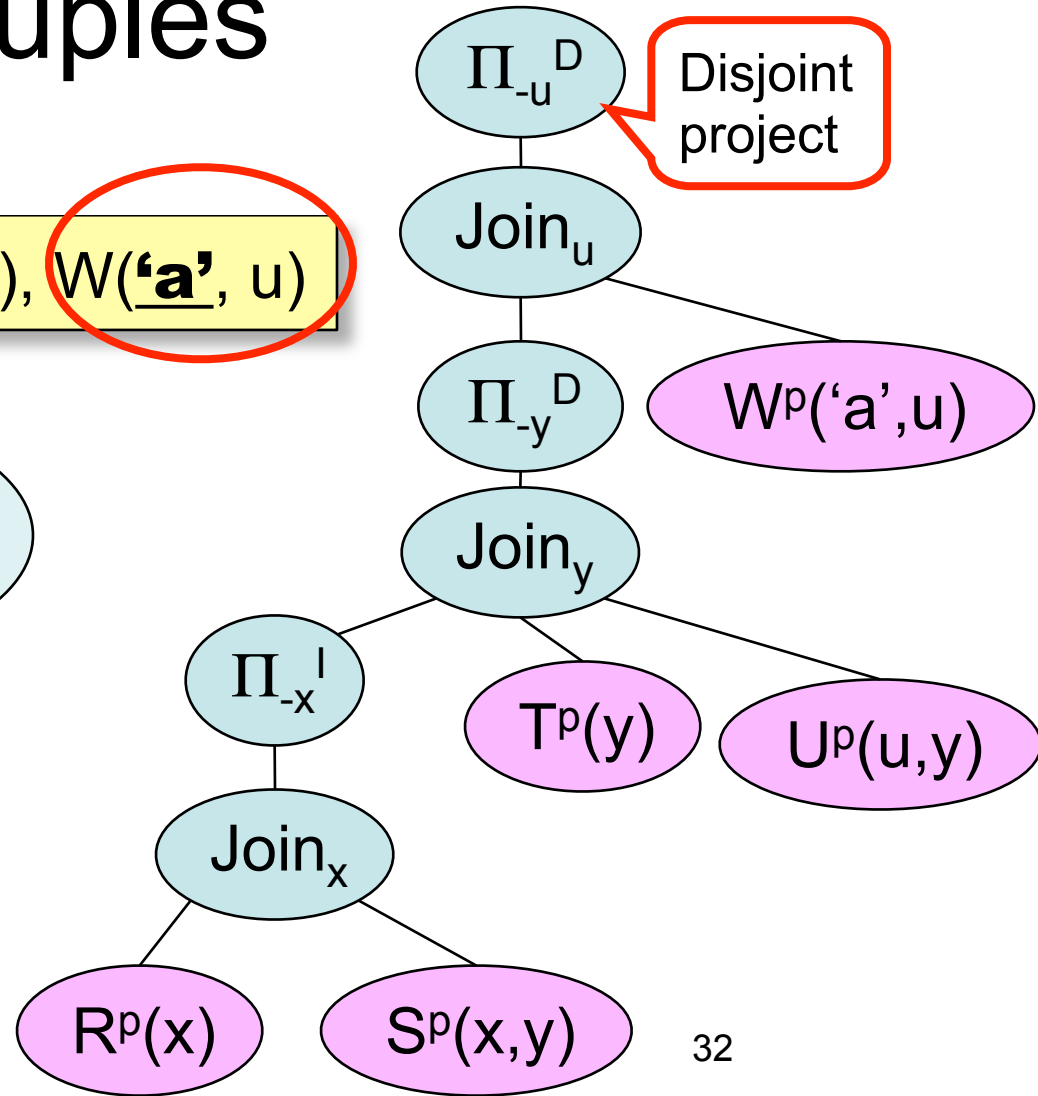
Case 2: Independent/disjoint Tuples

PTIME Queries:

$R(\underline{x}), S(\underline{x}, \underline{y}), T(\underline{y}), U(\underline{u}, y), W(\underline{a}, u)$



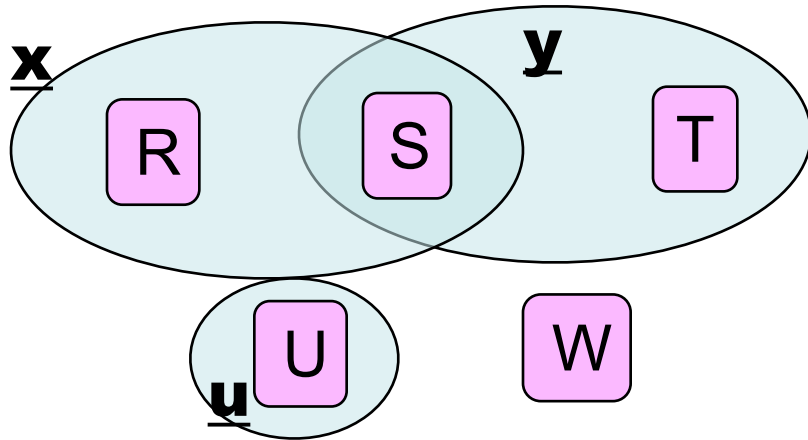
1. Root variable $\rightarrow \Pi^I$
2. CC's $\rightarrow \text{Join}$
3. Constant key attrs $\rightarrow \Pi^D$



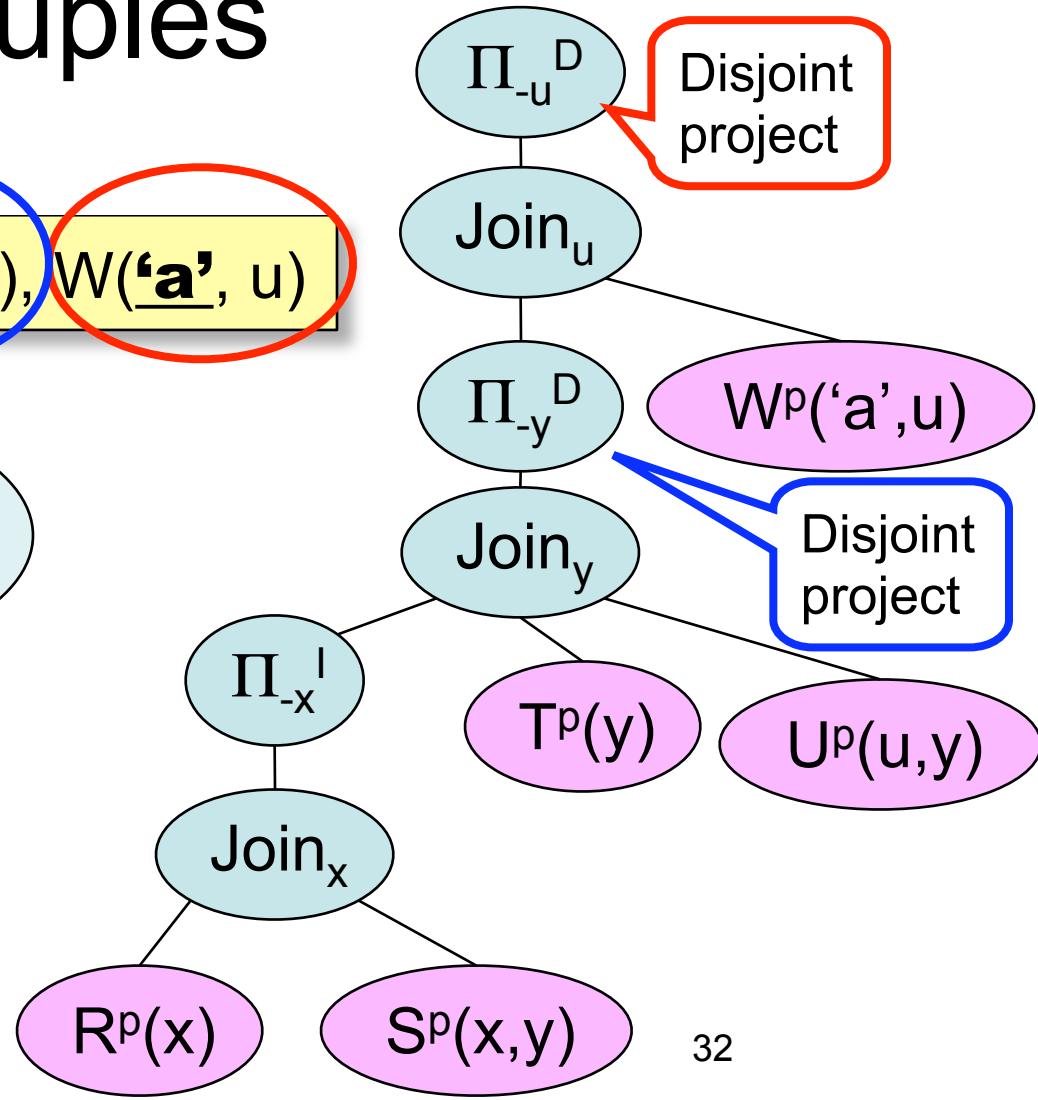
Case 2: Independent/disjoint Tuples

PTIME Queries:

$R(\underline{x}), S(\underline{x}, \underline{y}), T(\underline{y}), U(\underline{u}, y), W(\underline{a}, u)$



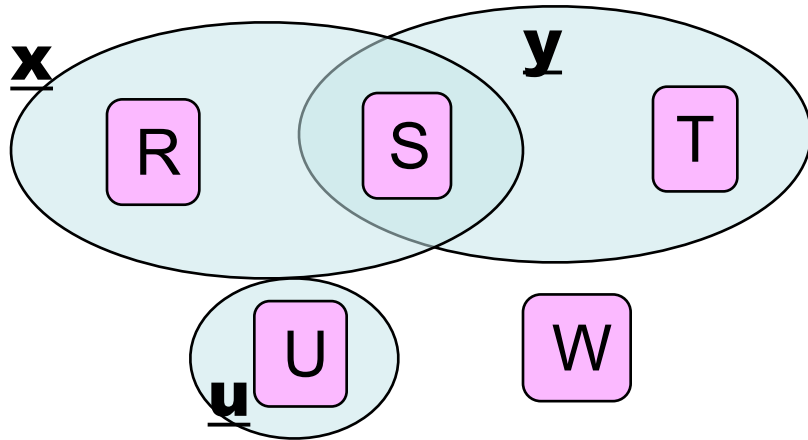
1. Root variable $\rightarrow \Pi^I$
2. CC's \rightarrow Join
3. Constant key attrs $\rightarrow \Pi^D$



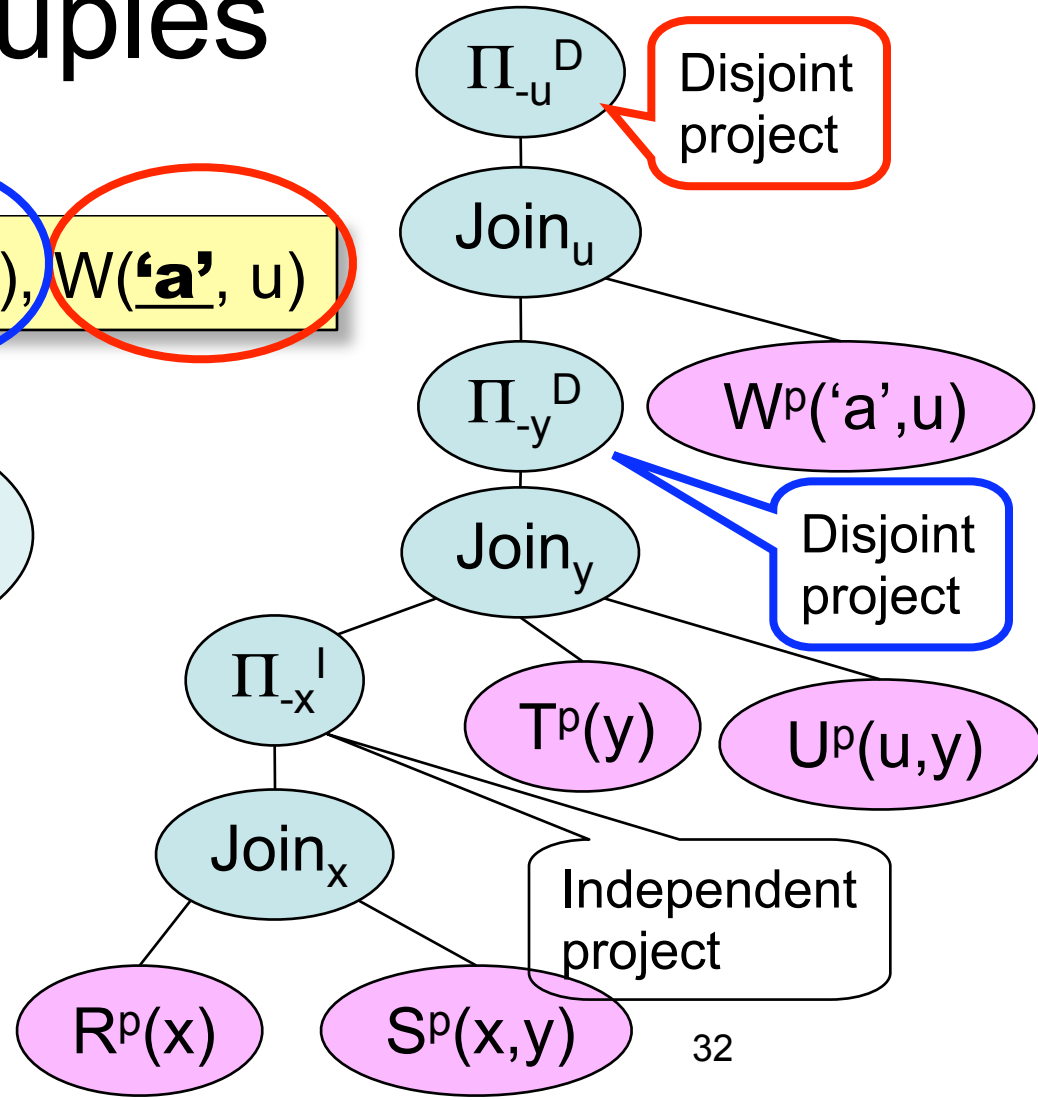
Case 2: Independent/disjoint Tuples

PTIME Queries:

$R(\underline{x}), S(\underline{x}, \underline{y}), T(\underline{y}), U(\underline{u}, y), W('a', u)$



1. Root variable $\rightarrow \Pi^I$
2. CC's \rightarrow Join
3. Constant key attrs $\rightarrow \Pi^D$



Case 2: Independent/disjoint Tuples

#P-hard Queries:

Recall:

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

$$h2 = R(\underline{\mathbf{x}}, y), S(\underline{\mathbf{y}})$$

$$h3 = R(\underline{\mathbf{x}}, y), S(x, \underline{\mathbf{y}})$$

} #P-hard by reduction
from PERMANENT

If the safe-plan algorithm fails on q , then q can be “rewritten” to either $h1$ or $h2$ or $h3$ and hence is #P-hard (see paper for details)

Theorem Testing if q is PTIME or #P-hard is PTIME complete

Summary on Query Evaluation

We understand completely only queries w/o self-joins

Lessons learned from our system MystiQ:

- When the query is safe:
 - Evaluate it exactly, in the database engine
 - Performance: close to regular SQL
- When the query is unsafe
 - Approximate it, compute only top-k
 - Performance: one or two orders of magnitude worse

[Re'2007]

Outline

- Data model
- Query evaluation
- **Challenges**

Query Optimization

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

- Combine safe plans + probabilistic inference
- “Interesting independence/disjointness”
- Model a probabilistic engine as black-box

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

Probabilistic Inference Algorithms

Open the box ! Logical to physical

Examine specific algorithms from KR:

- Variable elimination
- Junction trees
- Bounded treewidth

[Sen&Deshpande'2007]

[Bravo&Ramakrishnan'2007]

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

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 ($<$, \leq , \neq), aggregates
- Do hardness results still hold for $\text{Pr} = 1/2$?

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

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: Explore the connection between complex models and views

[Verma&Pearl'1990]

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

Information Leakage

A view V should not leak information about a secret S

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

- Issues: Which prior P ? What is \approx ?

Probability Logic:

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

[Pearl'88, Adams'98]

CHALLENGE: Define a probability logic for reasoning about information leakage

Conclusions

- Prohibitive cost of cleaning data
- Represent uncertainties explicitly
- Need to re-examine many assumptions

Conclusions

- Prohibitive cost of cleaning data
- Represent uncertainties explicitly
- Need to re-examine many assumptions

A call to arms:

The management of probabilistic data