Outline

- Part 1: Motivation
- Part 2: Probabilistic Databases
- Part 3: Weighted Model Counting
- Part 4: Lifted Inference for WFOMC

- Part 5: Completeness of Lifted Inference
- Part 6: Query Compilation
- Part 7: Symmetric Lifted Inference Complexity
- Part 8: Open-World Probabilistic Databases
- Part 9: Discussion & Conclusions

Lifted Inference on Asymmetric DB

Preprocess Q (omitted from this talk; see [Suciu'11]), then apply these rules (some have preconditions)

$$P(\neg Q) = 1 - P(Q)$$

negation

$$P(Q1 \wedge Q2) = P(Q1)P(Q2)$$
$$P(Q1 \vee Q2) = 1 - (1 - P(Q1))(1 - P(Q2))$$

Independent join / union

$$P(\exists z\, Q) = 1 - \prod_{A \in Domain} (1 - P(Q[A/z]))$$
$$P(\forall z\, Q) = \prod_{A \in Domain} P(Q[A/z])$$

Independent project

$$P(Q1 \wedge Q2) = P(Q1) + P(Q2) - P(Q1 \vee Q2)$$
$$P(Q1 \vee Q2) = P(Q1) + P(Q2) - P(Q1 \wedge Q2)$$

Inclusion/ exclusion

Example: Liftable Clause

$Q = \forall x \forall y\ S(x,y) \Rightarrow R(y)$ $= \forall y\ (\exists x\ S(x,y) \Rightarrow R(y))$

Example: Liftable Clause

$Q = \forall x \forall y\ S(x,y) \Rightarrow R(y)$ $= \forall y\ (\exists x\ S(x,y) \Rightarrow R(y))$

$P(Q) = \prod_{B \in \text{Domain}} P(\exists x\ S(x,B) \Rightarrow R(B))$

Indep. $\forall$

Example: Liftable Clause

$Q = \forall x \forall y\ S(x,y) \Rightarrow R(y)$ $= \forall y\ (\exists x\ S(x,y) \Rightarrow R(y))$

$$P(Q) = \prod_{B \in \text{Domain}} P(\exists x\ S(x,B) \Rightarrow R(B))$$

Indep. $\forall$

$$P(Q) = \prod_{B \in \text{Domain}} [1 - P(\exists x\ S(x,B)) \times (1 - P(R(b)))]$$

Indep. or:
$P(X \Rightarrow Y) =$
$= P(\neg X \vee Y)$
$= P(X)\ (1-P(Y))$

Example: Liftable Clause

$Q = \forall x \forall y\ S(x,y) \Rightarrow R(y)$ $= \forall y\ (\exists x\ S(x,y) \Rightarrow R(y))$

$$P(Q) = \prod_{B \in \text{Domain}} P(\exists x\ S(x,B) \Rightarrow R(B))$$

Indep. ∀

$$P(Q) = \prod_{B \in \text{Domain}} [1 - P(\exists x\ S(x,B)) \times (1 - P(R(b)))]$$

Indep. or:
$P(X \Rightarrow Y) =$
$= P(\neg X \vee Y)$
$= P(X)\ (1-P(Y))$

$$P(Q) = \prod_{B \in \text{Domain}} [1 - (1 - \prod_{A \in \text{Domain}} (1 - P(S(A,B)))) \times (1 - P(R(B)))]$$

Indep. ∃

Example: Liftable Clause

$Q = \forall x \forall y\ S(x,y) \Rightarrow R(y)$ $= \forall y\ (\exists x\ S(x,y) \Rightarrow R(y))$

$$P(Q) = \Pi_{B \in Domain}\ P(\exists x\ S(x,B) \Rightarrow R(B))$$

Indep. ∀

$$P(Q) = \Pi_{B \in Domain}\ [1 - P(\exists x\ S(x,B)) \times (1 - P(R(b)))]$$

Indep. or:
$P(X \Rightarrow Y) =$
$= P(\neg X \vee Y)$
$= P(X)\ (1-P(Y))$

$$P(Q) = \Pi_{B \in Domain}\ [1 - (1 - \Pi_{A \in Domain}\ (1 - P(S(A,B)))) \times (1 - P(R(B)))]$$

Indep. ∃

Lookup the probabilities in D

Example: Liftable Clause

$Q = \forall x \forall y\ S(x,y) \Rightarrow R(y)$ $= \forall y\ (\exists x\ S(x,y) \Rightarrow R(y))$

$$P(Q) = \Pi_{B \in Domain}\ P(\exists x\ S(x,B) \Rightarrow R(B))$$

Indep. ∀

$$P(Q) = \Pi_{B \in Domain}\ [1 - P(\exists x\ S(x,B)) \times (1 - P(R(b)))]$$

Indep. or:
$P(X \Rightarrow Y) =$
$= P(\neg X \vee Y)$
$= P(X)\ (1-P(Y))$

$$P(Q) = \Pi_{B \in Domain}\ [1 - (1 - \Pi_{A \in Domain}\ (1 - P(S(A,B)))) \times (1 - P(R(B)))]$$

Indep. ∃

Lookup the probabilities in D

Runtime = $O(n^2)$.

Two Questions

- Question 1: Are the lifted rules complete?
  - We know that they get stuck on some queries
  - Should we add more rules?

- Question 2: Are lifted rules stronger than grounded?
  - Lifted rules can also be grounded
  - Any advantage over grounded inference?

Two Questions

- Question 1: Are the lifted rules complete?
  - We know that they get stuck on some queries
  - Should we add more rules?

Complete for "unate ∀FO" and for "unate ∃FO"

- Question 2: Are lifted rules stronger than grounded?
  - Lifted rules can also be grounded
  - Any advantage over grounded inference?

Two Questions

- Question 1: Are the lifted rules complete?
  - We know that they get stuck on some queries
  - Should we add more rules?

Complete for "unate ∀FO" and for "unate ∃FO"

- Question 2: Are lifted rules stronger than grounded?
  - Lifted rules can also be grounded
  - Any advantage over grounded inference?

Strictly stronger than DPLL-based algorithms

FO^{un} = Unate FO

An FO sentence is *unate* if:

- Negations occur only on atoms
- Every relational symbol R either occurs only positively, or only negatively

$\forall FO^{un}$ $(\exists FO^{un})$ = restrict quantifiers too

$Q = \forall x \forall y\,(Smoker(x) \vee \neg Friend(x,y))$
$\wedge \forall x \forall y\,(\neg Friend(x,y) \vee Drinker(y))$

Unate

Not unate

$Q = \forall x \forall y\,(Smoker(x) \vee \neg Friend(x,y))$
$\wedge \forall x \forall y\,(Friend(x,y) \vee \neg Drinker(y))$

1. Are the Lifted Rules Complete?

We use complexity classes

- Inference rules: PTIME data complexity
- Some queries: #P-hard data complexity

Dichotomy Theorem for $\forall FO^{un}$ (or $\exists FO^{un}$)

- If lifted rules succeed, then query in PTIME
- If lifted rules fail, then query is #P-hard

Implies lifted rules are complete for $\forall FO^{un}$, $\exists FO^{un}$

Will show in two steps: **Small** and **Big Dichotomy Theorem**

NP v.s. #P

Decision Problems:

- SAT = Satisfiability Problem
- SAT is NP-complete [Cook'71]

Counting Problems:

- #SAT = model counting
- #SAT is #P-complete [Valiant'79]

Note: it would be wrong to say "#SAT is NP-complete"

Positive Partitioned 2CNF

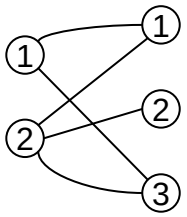
A PP2CNF is:

$$F = \bigwedge_{(i,j) \in E} (x_i \vee y_j)$$

where E = the edge set of a bipartite graph

$$F = (x_1 \vee y_1) \wedge (x_2 \vee y_1) \wedge (x_2 \vee y_3) \wedge (x_1 \vee y_3) \wedge (x_2 \vee y_2)$$

E:



Theorem [Provan'83] #PP2CNF is #P-hard

Unliftable Clause

$H_0 = \forall x \forall y\ (R(x) \vee S(x,y) \vee T(y))$

Independent Project not possible:
For $A_1 \neq A_2$,
$H_0[A_1/x]$ and $H_0[A_2/x]$ are dependent!

Unliftable Clause

$H_0 = \forall x \forall y\ (R(x) \vee S(x,y) \vee T(y))$

Independent Project not possible:
For $A_1 \neq A_2$,
$H_0[A_1/x]$ and $H_0[A_2/x]$ are dependent!

Theorem. Computing $P_D(H_0)$ is #P-hard in the size of D

[Dalvi&S.2004]

Unliftable Clause

$H_0 = \forall x \forall y\ (R(x) \vee S(x,y) \vee T(y))$

Independent Project not possible:
For $A_1 \neq A_2$,
$H_0[A_1/x]$ and $H_0[A_2/x]$ are dependent!

Theorem. Computing $P_D(H_0)$ is #P-hard in the size of D

[Dalvi&S.2004]

Proof: PP2CNF: $F = (X_{i1} \vee Y_{j1}) \wedge (X_{i2} \vee Y_{j2}) \wedge \ldots$ reduce #F to computing $P_D(H_0)$

By example:

Unliftable Clause

$H_0 = \forall x \forall y\ (R(x) \vee S(x,y) \vee T(y))$

Independent Project not possible:
For $A_1 \neq A_2$,
$H_0[A_1/x]$ and $H_0[A_2/x]$ are dependent!

Theorem. Computing $P_D(H_0)$ is #P-hard in the size of D

[Dalvi&S.2004]

Proof: PP2CNF: $F = (X_{i1} \vee Y_{j1}) \wedge (X_{i2} \vee Y_{j2}) \wedge \ldots$ reduce #F to computing $P_D(H_0)$

By example:

$F = (X_1 \vee Y_1) \wedge (X_1 \vee Y_2) \wedge (X_2 \vee Y_2)$

Unliftable Clause

$H_0 = \forall x \forall y\ (R(x) \vee S(x,y) \vee T(y))$

Independent Project not possible:
For $A_1 \neq A_2$,
$H_0[A_1/x]$ and $H_0[A_2/x]$ are dependent!

Theorem. Computing $P_D(H_0)$ is #P-hard in the size of D

[Dalvi&S.2004]

Proof: PP2CNF: $F = (X_{i1} \vee Y_{j1}) \wedge (X_{i2} \vee Y_{j2}) \wedge \ldots$ reduce #F to computing $P_D(H_0)$

By example:

$F = (X_1 \vee Y_1) \wedge (X_1 \vee Y_2) \wedge (X_2 \vee Y_2)$

D (tuples not shown have P=1)

R

X	P
x_1	0.5
x_2	0.5

S

X	Y	P
x_1	y_1	0
x_1	y_2	0
x_2	y_2	0

T

Y	P
y_1	0.5
y_2	0.5

Unliftable Clause

$H_0 = \forall x \forall y\ (R(x) \vee S(x,y) \vee T(y))$

Independent Project not possible:
For $A_1 \neq A_2$,
$H_0[A_1/x]$ and $H_0[A_2/x]$ are dependent!

Theorem. Computing $P_D(H_0)$ is #P-hard in the size of D

[Dalvi&S.2004]

Proof: PP2CNF: $F = (X_{i1} \vee Y_{j1}) \wedge (X_{i2} \vee Y_{j2}) \wedge \ldots$ reduce #F to computing $P_D(H_0)$

By example:

$F = (X_1 \vee Y_1) \wedge (X_1 \vee Y_2) \wedge (X_2 \vee Y_2)$

$P_D(H_0) = P(F)$; hence $P_D(H_0)$ is #P-hard

D (tuples not shown have P=1)

R

X	P
x_1	0.5
x_2	0.5

S

X	Y	P
x_1	y_1	0
x_1	y_2	0
x_2	y_2	0

T

Y	P
y_1	0.5
y_2	0.5

Hierarchical Queries

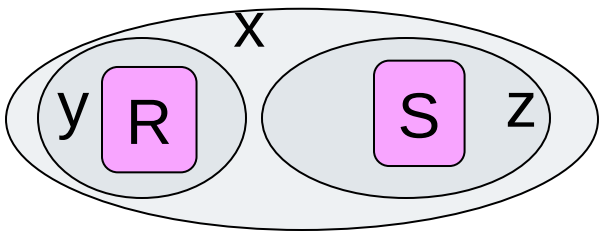
Fix Q; at(x) = set of atoms (=literals) containing the variable x

Definition Q is hierarchical if forall variables x, y:

$at(x) \subseteq at(y)$ or $at(x) \supseteq at(y)$ or $at(x) \cap at(y) = \emptyset$

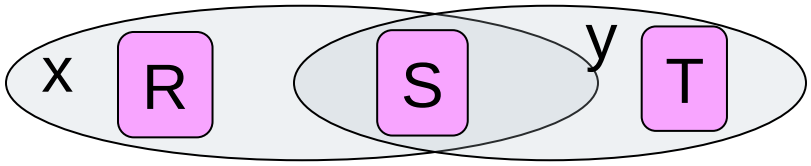
Hierarchical

$Q = \forall x \forall y \forall z (S(x,y) \vee T(x,z))$



Non-hierarchical

$H_0 = \forall x \forall y\ (R(x) \vee S(x,y) \vee T(y))$



The Small Dichotomy Theorem

[Dalvi&S.04]

Theorem Let Q be *one clause, with no repeated symbols*

- If Q is hierarchical, then $P_D(Q)$ is in PTIME.
- If Q is not hierarchical then $P_D(Q)$ is #P-hard.

Checking "Q is hierarchical" is in AC^0 (expression complexity)

The Small Dichotomy Theorem

[Dalvi&S.04]

Theorem Let Q be *one clause, with no repeated symbols*

- If Q is hierarchical, then $P_D(Q)$ is in PTIME.
- If Q is not hierarchical then $P_D(Q)$ is #P-hard.

Checking "Q is hierarchical" is in AC^0 (expression complexity)

[Dalvi,S.'12]

Fact: Any non-hierarchical Q in $\forall FO^{un}$ ($\exists FO^{un}$) is #P-hard

Next: consider only hierarchical queries in $\forall FO^{un}$ ($\exists FO^{un}$)

Clause with Repeated Symbols

$Q_J = \forall x_1 \forall y_1 \forall x_2 \forall y_2 \; (S(x_1,y_1) \vee R(y_1) \vee S(x_2,y_2) \vee T(y_2))$

Clause with Repeated Symbols

$$Q_J = \forall x_1 \forall y_1 \forall x_2 \forall y_2 \; (S(x_1,y_1) \vee R(y_1) \vee S(x_2,y_2) \vee T(y_2))$$

$$= [\forall x_1 \forall y_1 S(x_1,y_1) \vee R(y_1)] \; \vee \; [\forall x_2 \forall y_2 S(x_2,y_2) \vee T(y_2)]$$

Q_1 Q_2

Clause with Repeated Symbols

$$Q_J = \forall x_1 \forall y_1 \forall x_2 \forall y_2 \; (S(x_1,y_1) \vee R(y_1) \vee S(x_2,y_2) \vee T(y_2))$$

$$= [\forall x_1 \forall y_1 S(x_1,y_1) \vee R(y_1)] \; \vee \; [\forall x_2 \forall y_2 S(x_2,y_2) \vee T(y_2)]$$

Q_1 Q_2

$$P(Q_J) = P(Q_1) + P(Q_2) - P(Q_1 \wedge Q_2)$$

PTIME (have seen before)

Clause with Repeated Symbols

$$Q_J = \forall x_1 \forall y_1 \forall x_2 \forall y_2 \; (S(x_1,y_1) \vee R(y_1) \vee S(x_2,y_2) \vee T(y_2))$$

$$= [\forall x_1 \forall y_1 S(x_1,y_1) \vee R(y_1)] \; \vee \; [\forall x_2 \forall y_2 S(x_2,y_2) \vee T(y_2)]$$

Q_1 Q_2

$$P(Q_J) = P(Q_1) + P(Q_2) - P(Q_1 \wedge Q_2)$$

PTIME (have seen before)

$y = y1 = y2$

$$Q_1 \wedge Q_2 = \forall y \; [(\forall x_1 S(x_1,y) \vee R(y)) \wedge (\forall x_2 S(x_2,y)) \vee T(y)]$$
$$= \forall y \; [\forall x \, S(x,y) \vee (R(y) \wedge T(y))]$$

Clause with Repeated Symbols

$$Q_J = \forall x_1 \forall y_1 \forall x_2 \forall y_2 \; (S(x_1,y_1) \vee R(y_1) \vee S(x_2,y_2) \vee T(y_2))$$

$$= [\underbrace{\forall x_1 \forall y_1 S(x_1,y_1) \vee R(y_1)}_{Q_1}] \vee [\underbrace{\forall x_2 \forall y_2 S(x_2,y_2) \vee T(y_2)}_{Q_2}]$$

$$P(Q_J) = P(Q_1) + P(Q_2) - P(Q_1 \wedge Q_2)$$

PTIME (have seen before)

$y = y1 = y2$

$$Q_1 \wedge Q_2 = \forall y \; [(\forall x_1 S(x_1,y) \vee R(y)) \wedge (\forall x_2 S(x_2,y)) \vee T(y)]$$
$$= \forall y \; [\forall x \, S(x,y) \vee (R(y) \wedge T(y))]$$

$$P(Q_1 \wedge Q_2) = \Pi_{B \in \text{Domain}} \; P[\forall x.S(x,B) \vee (R(B) \wedge T(B))] = \ldots \text{etc}$$

Runtime = $O(n^2)$.

Unliftable Queries H_k

$H_0 = R(x) \vee S(x,y) \vee T(y)$

$H_1 = [R(x_0) \vee S(x_0,y_0)] \wedge [S(x_1,y_1) \vee T(y_1)]$

Will drop ∀ to reduce clutter

Every H_k, $k \geq 1$ is hierarchical

Unliftable Queries H_k

Will drop ∀ to reduce clutter

$H_0 = R(x) \vee S(x,y) \vee T(y)$

$H_1 = [R(x_0) \vee S(x_0,y_0)] \wedge [S(x_1,y_1) \vee T(y_1)]$

$H_2 = [R(x_0) \vee S_1(x_0,y_0)] \wedge [S_1(x_1,y_1) \vee S_2(x_1,y_1)] \vee [S_2(x_2,y_2) \vee T(y_2)]$

Every H_k, $k \geq 1$ is hierarchical

Unliftable Queries H_k

Will drop ∀ to reduce clutter

$H_0 = R(x) \vee S(x,y) \vee T(y)$

$H_1 = [R(x_0) \vee S(x_0,y_0)] \wedge [S(x_1,y_1) \vee T(y_1)]$

$H_2 = [R(x_0) \vee S_1(x_0,y_0)] \wedge [S_1(x_1,y_1) \vee S_2(x_1,y_1)] \vee [S_2(x_2,y_2) \vee T(y_2)]$

$H_3 = [R(x_0) \vee S_1(x_0,y_0)] \wedge [S_1(x_1,y_1) \vee S_2(x_1,y_1)] \wedge [S_2(x_2,y_2) \vee S_3(x_2,y_2)] \wedge [S_3(x_3,y_3) \vee T(y_3)]$

. . .

Every H_k, $k \geq 1$ is hierarchical

Unliftable Queries H_k

Will drop ∀ to reduce clutter

$H_0 = R(x) \vee S(x,y) \vee T(y)$

$H_1 = [R(x_0) \vee S(x_0,y_0)] \wedge [S(x_1,y_1) \vee T(y_1)]$

$H_2 = [R(x_0) \vee S_1(x_0,y_0)] \wedge [S_1(x_1,y_1) \vee S_2(x_1,y_1)] \vee [S_2(x_2,y_2) \vee T(y_2)]$

$H_3 = [R(x_0) \vee S_1(x_0,y_0)] \wedge [S_1(x_1,y_1) \vee S_2(x_1,y_1)] \wedge [S_2(x_2,y_2) \vee S_3(x_2,y_2)] \wedge [S_3(x_3,y_3) \vee T(y_3)]$

. . .

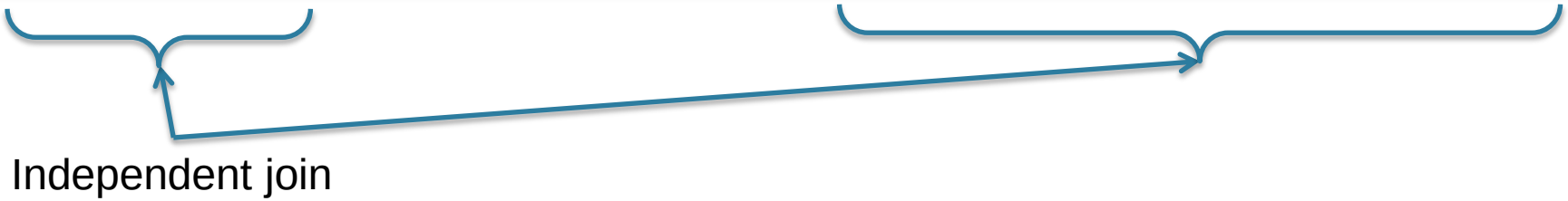
Every H_k, $k \geq 1$ is hierarchical

Theorem. [Dalvi&S'12] Every query H_k is #P-hard

A Closer Look at H_k

If we drop any one clause → in PTIME

$H_3 = [R(x_0) \vee S_1(x_0,y_0)] \wedge [S_1(x_1,y_1) \vee S_2(x_1,y_1)] \wedge [S_2(x_2,y_2) \vee S_3(x_2,y_2)] \wedge [S_3(x_3,y_3) \vee T(y_3)]$

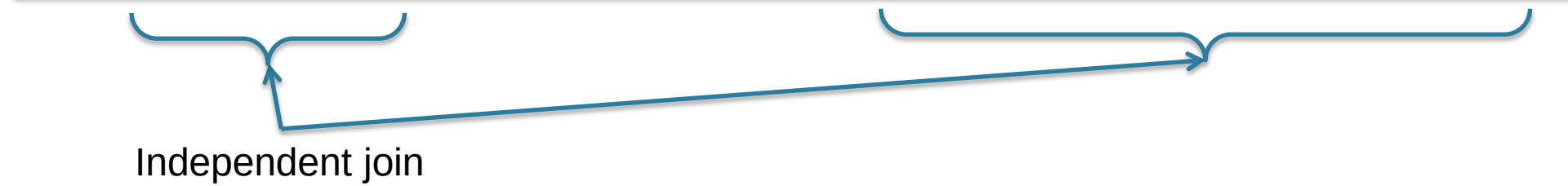


Independent join

A Closer Look at H_k

If we drop any one clause → in PTIME

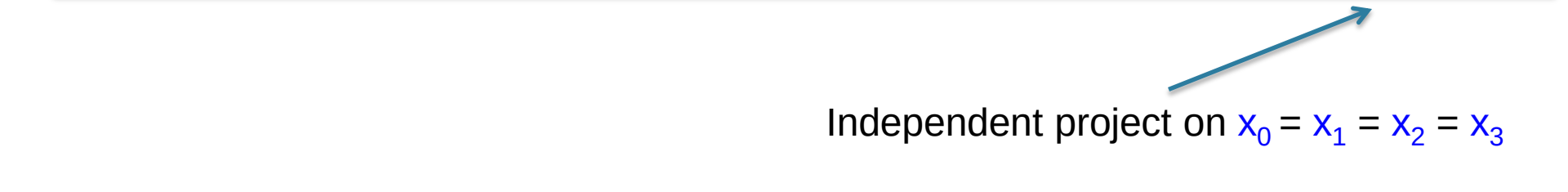
$H_3 = [R(x_0) \vee S_1(x_0,y_0)] \wedge [S_1(x_1,y_1) \vee S_2(x_1,y_1)] \wedge [S_2(x_2,y_2) \vee S_3(x_2,y_2)] \wedge [S_3(x_3,y_3) \vee T(y_3)]$



Independent join

If we replace $T(y_3)$ with $\mathbf{T(x_3)}$ then in PTIME

$[R(x_0) \wedge S_1(x_0,y_0)] \wedge [S_1(x_1,y_1) \vee S_2(x_1,y_1)] \wedge [S_2(x_2,y_2) \vee S_3(x_2,y_2)] \wedge [S_3(x_3,y_3) \vee \mathbf{T(x_3)}]$



Independent project on $x_0 = x_1 = x_2 = x_3$

Cancellations

Q_W = a Boolean expression
over the clauses in H_3 Yet, in PTIME

$$Q_W = [(R(x_0) \vee S_1(x_0,y_0)) \wedge (S_2(x_2,y_2) \vee S_3(x_2,y_2))] \vee \text{/* } Q_1 \text{ */}$$
$$[(R(x_0) \vee S_1(x_0,y_0)) \wedge (S_3(x_3,y_3) \vee T(y_3))] \vee \text{/* } Q_2 \text{ */}$$
$$[(S_1(x_1,y_1) \vee S_2(x_1,y_1)) \wedge (S_3(x_3,y_3) \vee T(y_3))] \text{ /* } Q_3 \text{ */}$$

Cancellations

Q_W = a Boolean expression over the clauses in H_3 Yet, in PTIME

$$Q_W = [(R(x_0) \vee S_1(x_0,y_0)) \wedge (S_2(x_2,y_2) \vee S_3(x_2,y_2))] \vee \text{/* } Q_1 \text{ */}$$
$$[(R(x_0) \vee S_1(x_0,y_0)) \wedge (S_3(x_3,y_3) \vee T(y_3))] \vee \text{/* } Q_2 \text{ */}$$
$$[(S_1(x_1,y_1) \vee S_2(x_1,y_1)) \wedge (S_3(x_3,y_3) \vee T(y_3))] \text{ /* } Q_3 \text{ */}$$

$$P(Q_W) = P(Q_1) + P(Q_2) + P(Q_3) +$$
$$- P(Q_1 \wedge Q_2) - P(Q_2 \wedge Q_3) - P(Q_1 \wedge Q_3)$$
$$+ P(Q_1 \wedge Q_2 \wedge Q_3)$$

Also = H_3

= H_3 (hard !)

Cancellations

Q_W = a Boolean expression
over the clauses in H_3 Yet, in PTIME

$$Q_W = [(R(x_0) \vee S_1(x_0,y_0)) \wedge (S_2(x_2,y_2) \vee S_3(x_2,y_2))] \vee \text{/* } Q_1 \text{ */}$$
$$[(R(x_0) \vee S_1(x_0,y_0)) \wedge (S_3(x_3,y_3) \vee T(y_3))] \vee \text{/* } Q_2 \text{ */}$$
$$[(S_1(x_1,y_1) \vee S_2(x_1,y_1)) \wedge (S_3(x_3,y_3) \vee T(y_3))] \text{ /* } Q_3 \text{ */}$$

$$P(Q_W) = P(Q_1) + P(Q_2) + P(Q_3) + \\ - P(Q_1 \wedge Q_2) - P(Q_2 \wedge Q_3) - P(Q_1 \wedge Q_3) \\ + P(Q_1 \wedge Q_2 \wedge Q_3)$$

Also = H_3

= H_3 (hard !)

Need to cancel terms to compute the query in PTIME
Using Mobius' function in the the lattice of Q's minterms [Suciu'11]

The Big Dichotomy Theorem

Call Q *liftable* if the rules don't get stuck.

Dichotomy Theorem [Dalvi'12] Fix a $\forall FO^{un}$ query Q.

1. If Q is liftable, then P(Q) is in PTIME
2. If Q is not liftable, then P(Q) is #P-complete

Note Original formulation for UCQ;
Immediate extension to $\forall FO^{un}$ and for $\exists FO^{un}$

Discussion

- This answers Question 1: lifted inference rules are complete for $\forall FO^{un}$ (and for $\exists FO^{un}$)
- Notice: we did not use any symmetries!
- Beyond unate FO? Conjectures:
  - Rules+resolution* complete for CNF-FO
  - No complete set of rules for FO

* $Q = \forall x \forall y\,(R(x) \vee S(x,y)) \wedge \forall x \forall y\,(\neg S(x,y) \vee T(y))$

$= \forall x \forall y\,(R(x) \vee S(x,y)) \wedge \forall x \forall y\,(\neg S(x,y) \vee T(y)) \wedge \mathbf{\forall x \forall y\,(R(x) \vee T(y))}$