# A **Dichotomy** for Non-Repeating Queries with **Negation** in **Probabilistic** Databases

Robert Fink and **Dan Olteanu**

UNIVERSITY OF OXFORD

LogicBlox

PODS June 24, 2014

# Outline

# Problem Setting

**Relational algebra query language fragment $1\text{RA}^-$**

- Included: Equi-joins, selections, projections, **difference**
- Excluded: Repeating relation symbols (self-joins), unions

**Tuple-independent probabilistic model**

- Each tuple associated with a fresh Boolean random variable $x$.
- $P(x)$ is the probability that the tuple exists in the database.
- Simplest probabilistic model in the literature.
  Beyond this model, query tractability is quickly lost.
- Used by real-world large-scale probabilistic repositories,
  e.g., Google Knowledge Vault.

**Query Evaluation Problem:** For a fixed $1\text{RA}^-$ query $Q$:
Given a tuple-independent probabilistic database $D$ and a tuple $t \in Q(D)$, compute its marginal probability.

# The Main Result

Data complexity of any $1\mathrm{RA}^{-}$ query $Q$ on tuple-independent databases:

- Polynomial time if $Q$ is **hierarchical** and
- #P-hard otherwise.

# The Main Result

Data complexity of any $1RA^-$ query $Q$ on tuple-independent databases:

- Polynomial time if $Q$ is **hierarchical** and
- #P-hard otherwise.

This result strictly extends a 2004 result by Dalvi and Suciu:

- We added the relational algebra *difference* operator
  - and moved from conjunctive queries without self-joins to $1RA$.
- Same syntactic characterization of tractable queries.
  - The hierarchical property can be recognized in LOGSPACE.
- The reason for tractability is however *different*.

# Hierarchical 1RA$^-$ Queries

Let $[C]$ be the equivalence class of attribute $C$ in query $Q$ as defined by the transitivity of equi-join conditions and difference operators.

- E.g., $C$ and $D$ are in the same class due to join $X(C) \bowtie_{C=D} Y(D)$ or difference $X(C) -_{C \leftrightarrow D} Y(D)$ under attribute mapping $C \leftrightarrow D$.

# Hierarchical 1RA$^-$ Queries

Let $[C]$ be the equivalence class of attribute $C$ in query $Q$ as defined by the transitivity of equi-join conditions and difference operators.

- E.g., $C$ and $D$ are in the same class due to join $X(C) \bowtie_{C=D} Y(D)$ or difference $X(C) -_{C \leftrightarrow D} Y(D)$ under attribute mapping $C \leftrightarrow D$.

(Boolean$^*$) 1RA$^-$ query $Q$ is *hierarchical* if

For every pair of distinct attribute equivalence classes $[A]$ and $[B]$, there is no triple of relation symbols $R$, $S$, and $T$ in $Q$ such that

- $R^{[A][\neg B]}$ has attributes in $[A]$ and not in $[B]$,
- $S^{[A][B]}$ has attributes in both $[A]$ and $[B]$, and
- $T^{[\neg A][B]}$ has attributes in $[B]$ and not in $[A]$.

$^*$ For non-Boolean queries, we need not check for equivalence classes with attributes in the query result.

# Examples

Examples of hierarchical queries:

- $\pi_\emptyset\big[\big(R(A) \bowtie S(A,B)\big) - T(A,B)\big]$
- $\pi_\emptyset\big[\big(R(A) \times T(B)\big) - \big(U(A) \times V(B)\big)\big]$
- $\pi_\emptyset\Big[\big(M(A) \times N(B)\big) - \big[\big(R(A) \times T(B)\big) - \big(U(A) \times V(B)\big)\big]\Big]$
- $\pi_\emptyset\Big[\big(M(A) \times N(B)\big) - \pi_A\big[\big(R(A) \times T(B)\big) - \big(U(A) \times V(B)\big)\big]\Big]$

# Examples

Examples of hierarchical queries:

- $\pi_\emptyset\big[\big(R(A) \bowtie S(A,B)\big) - T(A,B)\big]$
- $\pi_\emptyset\big[\big(R(A) \times T(B)\big) - \big(U(A) \times V(B)\big)\big]$
- $\pi_\emptyset\Big[\big(M(A) \times N(B)\big) - \big[\big(R(A) \times T(B)\big) - \big(U(A) \times V(B)\big)\big]\Big]$
- $\pi_\emptyset\Big[\big(M(A) \times N(B)\big) - \pi_A\big[\big(R(A) \times T(B)\big) - \big(U(A) \times V(B)\big)\big]\Big]$

Examples of non-hierarchical queries:

- $\pi_\emptyset\big[R(A) \bowtie S(A,B) \bowtie T(B)\big]$
- $\pi_\emptyset\Big[\pi_B\big(R(A) \bowtie S(A,B)\big) - T(B)\Big]$
- $\pi_\emptyset\Big[T(B) - \pi_B\big(R(A) \bowtie S(A,B)\big)\Big]$
- $\pi_\emptyset\Big[X(A) \bowtie \big[R(A) - \pi_A\big(T(B) \bowtie S(A,B)\big)\big]\Big]$

# Outline

# Hardness Proof Idea

Reduction from #P-hard model counting problem for positive 2DNF:

- Given a non-hierarchical 1RA query Q and
- A positive bipartite DNF formula $\Psi$,
- Construct a tuple-independent database $D$ with
  - size polynomial in the number of variables and clauses in $\Psi$, and
  - tuples annotated with variables in $\Psi$ such that $\Psi$ annotates $Q(D)$.
- Then $\#\Psi = 2^n \cdot P_{Q(D)}$, where
  - $P_{Q(D)}$ is the probability of $Q(D)$,
  - $1/2$ is the probability of each variable in $\Psi$, and
  - $n$ is the number of variables in $\Psi$.

# Example of Hardness Reduction

Input formula and query:

- $\Psi = x_1 y_1 \vee x_1 y_2$,
- $Q = \pi_\emptyset \Big[ R(A) - \pi_A \big( T(B) \bowtie S(A, B) \big) \Big]$

Construct database such that $\Psi$ annotates $Q$'s (nullary) result:

- Column $\Phi$ holds annotations over variables in $\Psi$.
  - Special annotations: $\top$ (true), $\perp$ (false)
- Variables used as constants for the attribute $B$ in $T$ and $S$.

- $S(a, b, \phi)$: Clause $a$ has variable $b$ exactly when $\phi$ is true.
- $R(a, \top)$ and $T(b, \neg b)$: $a$ is a clause and $b$ is a variable in $\Psi$.

$R$

| $A$ | $\Phi$ |
|---|---|
| 1 | $\top$ |
| 2 | $\top$ |

$T$

| $B$ | $\Phi$ |
|---|---|
| $x_1$ | $\neg \mathbf{x_1}$ |
| $y_1$ | $\neg \mathbf{y_1}$ |
| $y_2$ | $\neg \mathbf{y_2}$ |

$S$

| $A$ | $B$ | $\Phi$ |
|---|---|---|
| 1 | $x_1$ | $\top$ |
| 1 | $y_1$ | $\top$ |
| 1 | $y_2$ | $\perp$ |
| 2 | $x_1$ | $\top$ |
| 2 | $y_1$ | $\perp$ |
| 2 | $y_2$ | $\top$ |

$T \bowtie S$

| $A$ | $B$ | $\Phi$ |
|---|---|---|
| 1 | $x_1$ | $\neg \mathbf{x_1}$ |
| 1 | $y_1$ | $\neg \mathbf{y_1}$ |
| 1 | $y_2$ | $\perp$ |
| 2 | $x_1$ | $\neg \mathbf{x_1}$ |
| 2 | $y_1$ | $\perp$ |
| 2 | $y_2$ | $\neg \mathbf{y_2}$ |

$\pi_A(T \bowtie S)$

| $A$ | $\Phi$ |
|---|---|
| 1 | $\neg \mathbf{x_1} \vee \neg \mathbf{y_1}$ |
| 2 | $\neg \mathbf{x_1} \vee \neg \mathbf{y_2}$ |

$R - \pi_A(T \bowtie S)$

| $A$ | $\Phi$ |
|---|---|
| 1 | $\mathbf{x_1 y_1}$ |
| 2 | $\mathbf{x_1 y_2}$ |

# Example of Hardness Reduction

Input formula and query:

- $\Psi = x_1 y_1 \vee x_1 y_2$,
- $Q = \pi_{\emptyset}\Big[R(A) - \pi_A\big(T(B) \bowtie S(A, B)\big)\Big]$

Construct database such that $\Psi$ annotates $Q$'s (nullary) result:

- Column $\Phi$ holds annotations over variables in $\Psi$.
  - Special annotations: $\top$ (true), $\bot$ (false)
- Variables used as constants for the attribute $B$ in $T$ and $S$.

- $S(a, b, \phi)$: Clause $a$ has variable $b$ exactly when $\phi$ is true.
- $R(a, \top)$ and $T(b, \neg b)$: $a$ is a clause and $b$ is a variable in $\Psi$.

$R$

| $A$ | $\Phi$ |
|---|---|
| 1 | $\top$ |
| 2 | $\top$ |

$T$

| $B$ | $\Phi$ |
|---|---|
| $x_1$ | $\neg\mathbf{x_1}$ |
| $y_1$ | $\neg\mathbf{y_1}$ |
| $y_2$ | $\neg\mathbf{y_2}$ |

$S$

| $A$ | $B$ | $\Phi$ |
|---|---|---|
| 1 | $x_1$ | $\top$ |
| 1 | $y_1$ | $\top$ |
| 1 | $y_2$ | $\bot$ |
| 2 | $x_1$ | $\top$ |
| 2 | $y_1$ | $\bot$ |
| 2 | $y_2$ | $\top$ |

$T \bowtie S$

| $A$ | $B$ | $\Phi$ |
|---|---|---|
| 1 | $x_1$ | $\neg\mathbf{x_1}$ |
| 1 | $y_1$ | $\neg\mathbf{y_1}$ |
| 1 | $y_2$ | $\bot$ |
| 2 | $x_1$ | $\neg\mathbf{x_1}$ |
| 2 | $y_1$ | $\bot$ |
| 2 | $y_2$ | $\neg\mathbf{y_2}$ |

$\pi_A(T \bowtie S)$

| $A$ | $\Phi$ |
|---|---|
| 1 | $\neg\mathbf{x_1} \vee \neg\mathbf{y_1}$ |
| 2 | $\neg\mathbf{x_1} \vee \neg\mathbf{y_2}$ |

$R - \pi_A(T \bowtie S)$

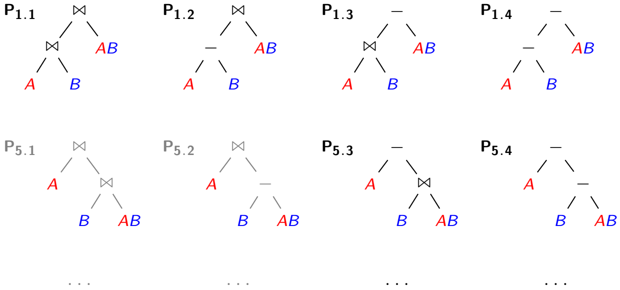
| $A$ | $\Phi$ |
|---|---|
| 1 | $\mathbf{x_1 y_1}$ |
| 2 | $\mathbf{x_1 y_2}$ |

Query $Q$ is already hard when $T$ is the only uncertain input relation!

# Hard Query Patterns

There are 48 (!) minimal non-hierarchical query patterns.

- Binary trees with leaves $A$, $AB$, and $B$ and inner nodes $\bowtie$ or $-$.
  - Some are symmetric and need not be consider separately:
    $A$ and B can be exchanged, joins are commutative and associative.
  - Still, many cases left to consider due to the difference operator.

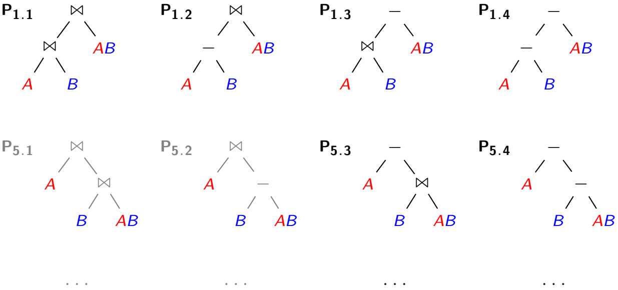



- There is a database construction scheme for each pattern.
- Each non-hierarchical query $Q$ *matches* a pattern $\mathbf{P}_{x.y}$.

# Hard Query Patterns

There are 48 (!) minimal non-hierarchical query patterns.

- Binary trees with leaves $A$, $AB$, and $B$ and inner nodes $\bowtie$ or $-$.
  - Some are symmetric and need not be consider separately: A and B can be exchanged, joins are commutative and associative.
  - Still, many cases left to consider due to the difference operator.



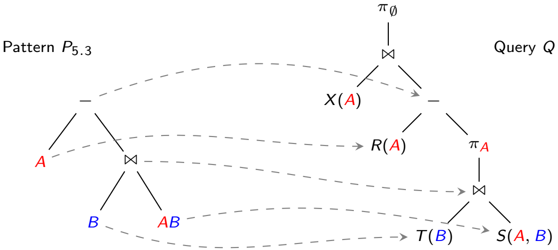

- There is a database construction scheme for each pattern.
- Each non-hierarchical query $Q$ *matches* a pattern $\mathbf{P}_{x.y}$.

$\mathbf{P}_{1.1}$ is the only hard pattern to consider w/o the difference operator!

# Non-hierarchical Queries Match Minimal Hard Patterns

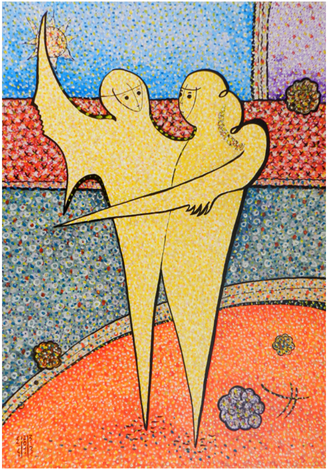
Each non-hierarchical query $Q$ *matches* a pattern $\mathbf{P}_{x.y}$:

- There is a total mapping from $\mathbf{P}_{x.y}$ to $Q$'s parse tree that
  - is identity on inner nodes $\bowtie$ and $-$,
  - preserves ancestor-descendant relationships,
  - maps leaves $A$, $AB$, $B$ to relations $R^{[A][\neg B]}$, $S^{[A][B]}$, $T^{[\neg A][B]}$.




- The match preserves the annotation of the query pattern:
  $Q$ and $\mathbf{P}_{x.y}$ have the same annotation for any input database.

# Outline



The Dichotomy

The Interesting but Hard Queries

The Easy Queries

Leftovers

# Evaluation of Hierarchical 1RA$^-$ Queries

Approach based on knowledge compilation

- For any database $D$, the probability $P_{Q(D)}$ of a 1RA$^-$ query $Q$ is the probability $P_\Psi$ of the query annotation $\Psi$.
- Compile $\Psi$ into poly-size OBDD($\Psi$).
- Compute probability of OBDD($\Psi$) in time linear in its size.

# Evaluation of Hierarchical 1RA$^-$ Queries

Approach based on knowledge compilation

- For any database $D$, the probability $P_{Q(D)}$ of a 1RA$^-$ query $Q$ is the probability $P_\Psi$ of the query annotation $\Psi$.
- Compile $\Psi$ into poly-size OBDD($\Psi$).
- Compute probability of OBDD($\Psi$) in time linear in its size.

Distinction from existing tractability results [O. & Huang 2008]:

- 1RA$^-$ queries w/o difference: Annotations are read-once.
  - Read-once annotations admit linear-size OBBDs.
- 1RA$^-$ queries: Annotations are <u>not</u> read-once.
  - They admit OBBDs of size linear in the database size <u>but</u> exponential in the query size.

# The Inner Workings

From hierarchical $1RA^-$ to RC-hierarchical $\exists$-consistent $RC^{\exists}$:

- Translate query $Q$ into an equivalent disjunction of disjunction-free existential relational calculus queries $Q_1 \vee \cdots \vee Q_k$.
  - $k$ can be very large for queries with projection under difference!

- **RC-hierarchical**:
  For each $\exists_X(Q')$, every relation symbol in $Q'$ has variable $X$.
  - Each of the disjuncts gives rise to a poly-size OBDD.

- $\exists$-**consistent**:
  The nesting order of the quantifiers is the same in $Q_1, \cdots, Q_k$.
  - All OBDDs have compatible variable orders and
    their disjunction is a poly-size OBDD.

- The OBDD width grows exponentially with $k$,
  its height stays linear in the size of the database.
  - Width = maximum number of edges crossing the section between any two consecutive levels.

# Query Evaluation Example

Consider the following query and tuple-independent database:

$$Q = \pi_{\emptyset}\Big[\big(R(A) \times T(B)\big) - \big(U(A) \times V(B)\big)\Big]$$

| $R$ | |
|---|---|
| $A$ | $\Phi$ |
| 1 | $\mathbf{r_1}$ |
| 2 | $\mathbf{r_2}$ |

| $T$ | |
|---|---|
| $B$ | $\Phi$ |
| 1 | $\mathbf{t_1}$ |
| 2 | $\mathbf{t_2}$ |

| $U$ | |
|---|---|
| $A$ | $\Phi$ |
| 1 | $\mathbf{u_1}$ |
| 2 | $\mathbf{u_2}$ |

| $V$ | |
|---|---|
| $B$ | $\Phi$ |
| 1 | $\mathbf{v_1}$ |
| 2 | $\mathbf{v_2}$ |

| $R \bowtie T$ | | |
|---|---|---|
| $A$ | $B$ | $\Phi$ |
| 1 | 1 | $\mathbf{r_1 t_1}$ |
| 1 | 2 | $\mathbf{r_1 t_2}$ |
| 2 | 1 | $\mathbf{r_2 t_1}$ |
| 2 | 2 | $\mathbf{r_2 t_2}$ |

| $R \bowtie T - U \bowtie V$ | | |
|---|---|---|
| $A$ | $B$ | $\Phi$ |
| 1 | 1 | $\mathbf{r_1 t_1} \neg(\mathbf{u_1 v_1})$ |
| 1 | 2 | $\mathbf{r_1 t_2} \neg(\mathbf{u_1 v_2})$ |
| 2 | 1 | $\mathbf{r_2 t_1} \neg(\mathbf{u_2 v_1})$ |
| 2 | 2 | $\mathbf{r_2 t_2} \neg(\mathbf{u_2 v_2})$ |

# Query Evaluation Example

Consider the following query and tuple-independent database:

$$Q = \pi_{\emptyset}\Big[\big(R(A) \times T(B)\big) - \big(U(A) \times V(B)\big)\Big]$$

$R$

| $A$ | $\Phi$ |
|---|---|
| 1 | $\mathbf{r_1}$ |
| 2 | $\mathbf{r_2}$ |

$T$

| $B$ | $\Phi$ |
|---|---|
| 1 | $\mathbf{t_1}$ |
| 2 | $\mathbf{t_2}$ |

$U$

| $A$ | $\Phi$ |
|---|---|
| 1 | $\mathbf{u_1}$ |
| 2 | $\mathbf{u_2}$ |

$V$

| $B$ | $\Phi$ |
|---|---|
| 1 | $\mathbf{v_1}$ |
| 2 | $\mathbf{v_2}$ |

$R \bowtie T$

| $A$ | $B$ | $\Phi$ |
|---|---|---|
| 1 | 1 | $\mathbf{r_1 t_1}$ |
| 1 | 2 | $\mathbf{r_1 t_2}$ |
| 2 | 1 | $\mathbf{r_2 t_1}$ |
| 2 | 2 | $\mathbf{r_2 t_2}$ |

$R \bowtie T - U \bowtie V$

| $A$ | $B$ | $\Phi$ |
|---|---|---|
| 1 | 1 | $\mathbf{r_1 t_1} \neg(\mathbf{u_1 v_1})$ |
| 1 | 2 | $\mathbf{r_1 t_2} \neg(\mathbf{u_1 v_2})$ |
| 2 | 1 | $\mathbf{r_2 t_1} \neg(\mathbf{u_2 v_1})$ |
| 2 | 2 | $\mathbf{r_2 t_2} \neg(\mathbf{u_2 v_2})$ |

The annotation of $Q$ is:

$$\Psi = r_1\big[t_1(\neg u_1 \vee \neg v_1) \vee t_2(\neg u_1 \vee \neg v_2)\big] \vee r_2\big[t_1(\neg u_2 \vee \neg v_1) \vee t_2(\neg u_2 \vee \neg v_2)\big].$$

- Variables entangle in $\Psi$ beyond read-once factorization.
- This is the pivotal intricacy introduced by the difference operator.

# Query Evaluation Example (2)

Translate $Q = \pi_\emptyset\Big[(R(A) \times T(B)) - (U(A) \times V(B))\Big]$ into $\mathrm{RC}^\exists$:

$$Q_{RC} = \underbrace{\exists_A \big(R(A) \wedge \neg U(A)\big) \wedge \exists_B T(B)}_{Q_1} \quad \vee \quad \underbrace{\exists_A R(A) \wedge \exists_B \big(T(B) \wedge \neg V(B)\big)}_{Q_2}.$$

- Both $Q_1$ and $Q_2$ are RC-hierarchical.
- $Q_1 \vee Q_2$ is $\exists$-consistent: Same order $\exists_A \exists_B$ for $Q_1$ and $Q_2$.

Query annotation:

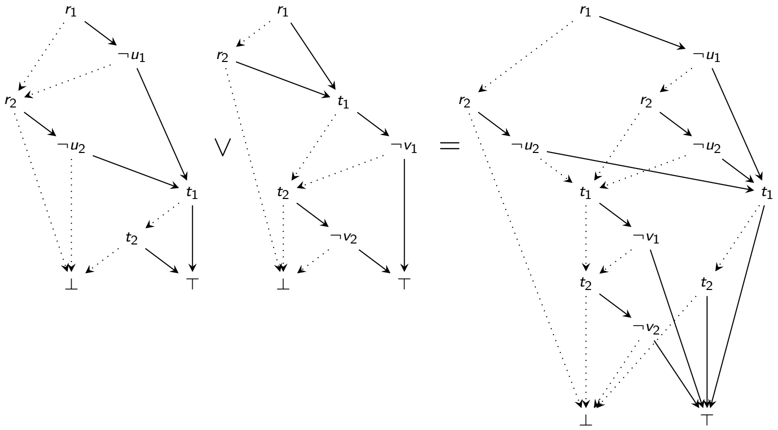
$$\Psi = \underbrace{(r_1 \neg u_1 \vee r_2 \neg u_2) \wedge (t_1 \vee t_2)}_{\Psi_1} \quad \vee \quad \underbrace{(r_1 \vee r_2) \wedge (t_1 \neg v_1 \vee t_2 \neg v_2)}_{\Psi_2}.$$

- Both $\Psi_1$ and $\Psi_2$ admit linear-size OBDDs.
- The two OBDDs have compatible orders and their disjunction is an OBDD whose width is the product of the widths of the two OBDDs.

# Query Evaluation Example (3)

Compile query annotation into OBDD:

$$\Psi = \underbrace{(r_1 \neg u_1 \vee r_2 \neg u_2) \wedge (t_1 \vee t_2)}_{\Psi_1} \quad \vee \quad \underbrace{(r_1 \vee r_2) \wedge (t_1 \neg v_1 \vee t_2 \neg v_2)}_{\Psi_2}.$$



# Outline

# Dichotomies Beyond $1RA^-$

Some known dichotomies

- Conjunctive queries w/o self-joins [Dalvi & Suciu 2004]
- Unions of conjunctive queries [Dalvi & Suciu 2010]
- Quantified relational algebra queries [F. & O. & Rath 2011]

Full relational algebra

- It is undecidable whether the union of two equivalent relational algebra queries, one hard and one tractable, is tractable.

Non-repeating relational algebra = $1RA^-$ + union.

- Hierarchical property not enough.
- $\pi_\emptyset[(R(A) \bowtie S_1(A,B) \cup T(B) \bowtie S_2(A,B)) - S(A,B)]$ is hard, though it is equivalent to a union of two hierarchical $1RA^-$ queries.

Non-repeating relational calculus

- $S(x,y) \wedge \neg R(x)$ is tractable, $S(x,y) \wedge (R(x) \vee T(y))$ is hard.
  - Both are non-repeatable, yet not expressible in $1RA^-$.
- Possible (though expensive) approach:
  - Translate to $RC^\exists$ and check RC-hierarchical and $\exists$-consistency.

**Thank you!**