UNIVERSITY OF OXFORD
DEPARTMENT OF COMPUTER SCIENCE
LogicBlox

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

Robert Fink and Dan Olteanu

PODS, June 2014

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.

Query Language and Data Model

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

- 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.

The Hierarchical Property for a Query Q

For every pair of distinct attribute equivalence classes $[A]$, $[B]$ 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 $[A]$.

Non-hierarchical queries	Hierarchical queries
$\pi_\emptyset[R(A) \bowtie S(A,B) \bowtie T(B)]$	$\pi_\emptyset[(R(A) \bowtie S(A,B)) - T(A,B)]$
$\pi_\emptyset[\pi_B(R(A) \bowtie S(A,B)) - T(B)]$	$\pi_\emptyset[(R(A) \times T(B)) - (U(A) \times V(B))]$
$\pi_\emptyset[T(B) - \pi_B(R(A) \bowtie S(A,B))]$	$\pi_\emptyset[(M(A) \times N(B)) - [(R(A) \times T(B)) - (U(A) \times V(B))]]$

The hierarchical property can be recognized in LOGSPACE.

The Hard Queries

Reduction from the #P-hard problem #SAT for positive 2DNF.

- Input formula and query: $\Psi = x_1y_1 \vee x_1y_2$, $Q = \pi_\emptyset\left[R(A) - \pi_A(T(B) \bowtie S(A,B))\right]$
- Construct database such that Ψ annotates Q's result:
  - $S(a,b,\phi)$: Clause a has variable b exactly when ϕ is true.
  - $R(a,\top)$ and $T(b,\neg b)$: a is a clause and b is a variable in Ψ.

R

A	Φ
1	$\top$
2	$\top$

T

B	Φ
x_1	$\neg\mathbf{x_1}$
y_1	$\neg\mathbf{y_1}$
y_2	$\neg\mathbf{y_2}$

S

A	B	Φ
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	Φ
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	$\neg\mathbf{y_2}$

$\pi_A(T \bowtie S)$

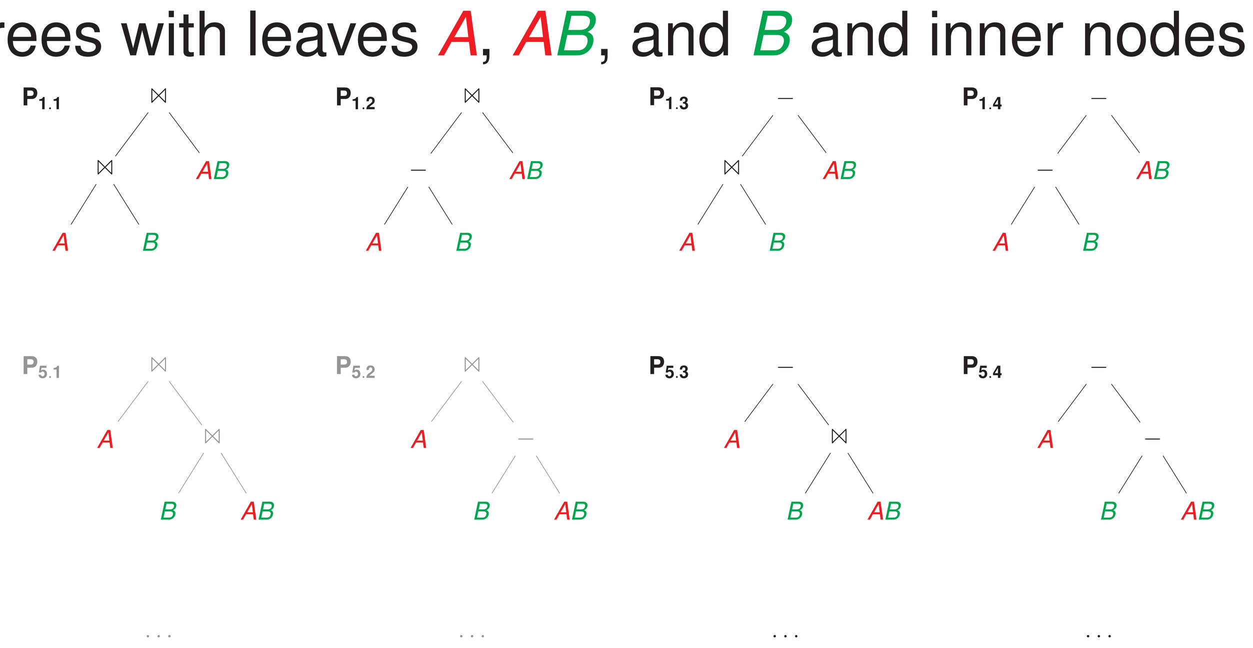
A	Φ
1	$\neg\mathbf{x_1} \vee \neg\mathbf{y_1}$
2	$\neg\mathbf{x_1} \vee \neg\mathbf{y_2}$

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

A	Φ
1	$\mathbf{x_1y_1}$
2	$\mathbf{x_1y_2}$

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

- Binary trees with leaves A, AB, and B and inner nodes $\bowtie$ or $-$.



- There is a database construction scheme for each pattern.

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, B, AB 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.

The Evaluation Algorithm for Hierarchical Queries

- For any database D, the probability $P_{Q(D)}$ of a $1\mathrm{RA}^-$ query Q is the probability P_Ψ of the query annotation Ψ.

$$Q = \pi_\emptyset(R(A) \times T(B)) - (U(A) \times V(B))$$

R

A	Φ
1	$\mathbf{r_1}$
2	$\mathbf{r_2}$

T

B	Φ
1	$\mathbf{t_1}$
2	$\mathbf{t_2}$

U

A	Φ
1	$\mathbf{u_1}$
2	$\mathbf{u_2}$

V

B	Φ
1	$\mathbf{v_1}$
2	$\mathbf{v_2}$

$R \bowtie T$

A	B	Φ
1	1	$\mathbf{r_1t_1}$
1	2	$\mathbf{r_1t_2}$
2	1	$\mathbf{r_2t_1}$
2	2	$\mathbf{r_2t_2}$

$R \bowtie T - U \bowtie V$

A	B	Φ
1	1	$\mathbf{r_1t_1}\neg(\mathbf{u_1v_1})$
1	2	$\mathbf{r_1t_2}\neg(\mathbf{u_1v_2})$
2	1	$\mathbf{r_2t_1}\neg(\mathbf{u_2v_1})$
2	2	$\mathbf{r_2t_2}\neg(\mathbf{u_2v_2})$

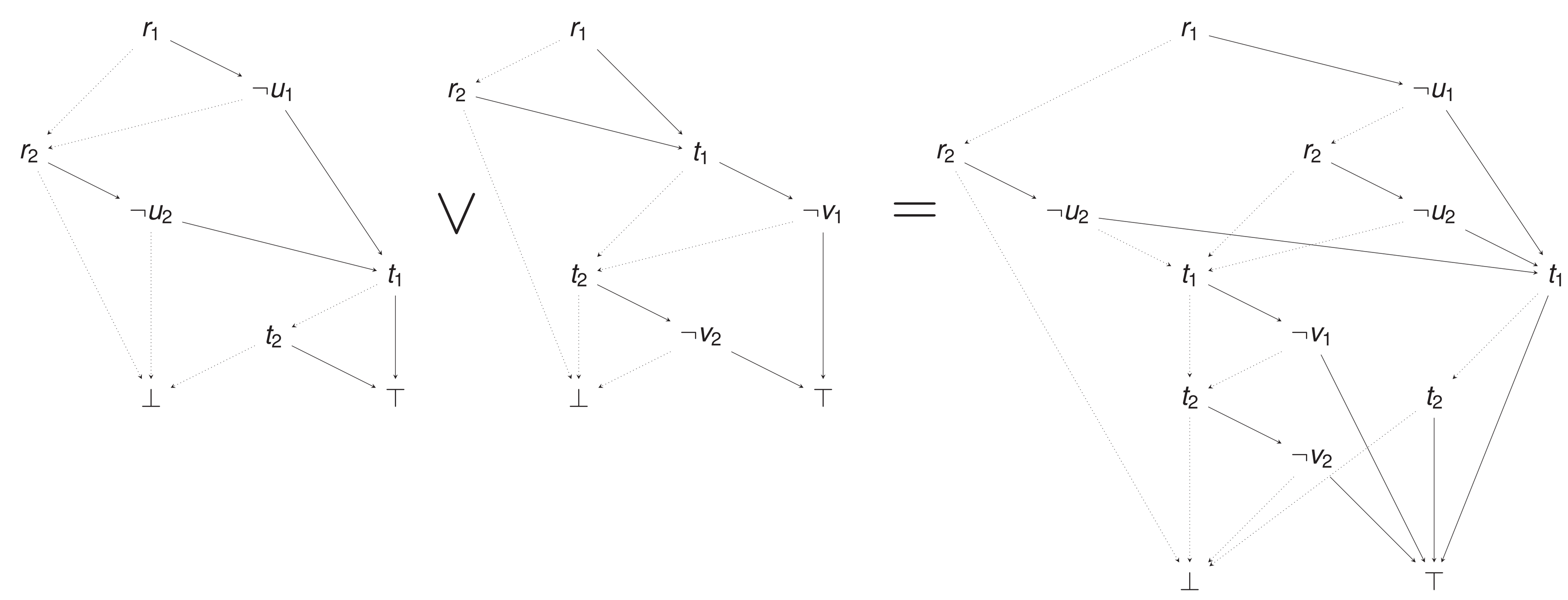
- Translate query Q into equivalent $\mathrm{RC}^\exists$: A disjunction of disjunction-free existential relational calculus queries.

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

  - **RC-hierarchical**: For each quantifier $\exists_X(Q)$, every relation symbol in Q has variable X.
  - **$\exists$-consistent**: All disjuncts have the same nesting order of $\exists$s.
- Compile query annotation into OBDD

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

  - **RC-hierarchical**: Each disjunct gives rise to a poly-size OBDD.
  - **$\exists$-consistent**: All OBDDs have compatible variable orders.
  - The OBDD width grows exponentially with the number of disjuncts, while its height stays linear in the database size.



Dichotomies Beyond $1\mathrm{RA}^-$

Some known dichotomies

- Conjunctive queries w/o self-joins, unions of conjunctive queries [Dalvi & Suciu 2004-2010], quantified queries [F.&O.& Rath 2011]

Full relational algebra

- seems unattainable since it is undecidable whether the union of two equivalent queries, one hard and one tractable, is tractable.

Non-repeating relational algebra = $1\mathrm{RA}^-$ + 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 $1\mathrm{RA}^-$ 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 $1\mathrm{RA}^-$.
- Possible (though expensive) approach: Translate to $\mathrm{RC}^\exists$ and check RC-hierarchical and $\exists$-consistency.