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 1RA^− 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 1RA^−
- 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 Hard Queries

Reduction from the #P-hard problem #SAT for positive 2DNF:
- Input formula and query: \( \Psi = \bigwedge t \bigvee y, Q = \pi x R \land \Psi \)
- Construct database such that \( \Psi \) annotates Q's result:
  - \( S(a, b, \phi) \): 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 \( \Psi \).

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>S</th>
<th>T</th>
<th>( T \times S )</th>
<th>( S )</th>
<th>( T )</th>
<th>( T \times S )</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
<td>x \lor \neg x</td>
<td>y \lor \neg y</td>
<td>x \land \neg y</td>
<td>x \lor \neg x</td>
<td>y \lor \neg y</td>
<td>x \land \neg y</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

There are 48 (!) minimal non-hierarchical query patterns.
- Binary trees with leaves \( A, AB, \) and \( B \) and inner nodes \( \lor \) or \( \land \).
- There is a database construction scheme for each pattern.

Each non-hierarchical query \( Q \) matches a pattern \( P_{xy} \):
- There is a total mapping from \( P_{xy} \) to \( Q \)'s parse tree that
  - is identity on inner nodes \( \lor \) and \( \land \),
  - preserves ancestor-descendant relationships,
  - maps leaves \( A, B, AB \) to relations \( R[AB], S[AB], T[AB] \),
- The match preserves the annotation of the query pattern: \( Q \) and \( P_{xy} \) have the same annotation for any input database.

The Hierarchical Property for a Query Q

For every pair of distinct equivalence classes \([A], [B]\) there is no triple of relation symbols \( R, S, \) and \( T \) in \( Q \) such that
- \( R[AB] \) has attributes in \([A]\) and not in \([B]\),
- \( S[AB] \) has attributes in both \([A]\) and \([B]\), and
- \( T[AB] \) has attributes in \([B]\) and not in \([A]\).

The hierarchical property can be recognized in LOGSPACE.

The Evaluation Algorithm for Hierarchical Queries

- For any database \( D \), the probability \( P_Q(D) \) of a 1RA^− query \( Q \) is the probability \( P_\Psi \) of the query annotation \( \Psi \).

\[
\begin{align*}
P_\Psi &= \frac{1}{|\Omega(D)|} \sum_{D'} P(D') \times M(D') \\
&= \frac{1}{|\Omega(D)|} \sum_{D'} \prod_{(R \times T) \in \Psi} P(R) \times P(T)
\end{align*}
\]

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

\[
\Psi = \bigwedge (\exists X \forall Y R_{XY} \land T_{XY}) \lor (\exists X \forall Y R_{XY} \land T_{XY})
\]

- \( \forall\)-hierarchical: Each disjunct gives rise to a poly-size OBDD.
- \( \forall\)-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 1RA

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 = 1RA + union
- Hierarchical property not enough.
- \( R_{[AB]} \times S_{[AB]} \cup T_{[AB]} \) is hard, though it is equivalent to a union of two hierarchical 1RA^− queries.

Non-repeating relational calculus
- \( S(x, y) \land \neg R(x) \) is tractable, \( S(x, y) \land (R(x) \lor T(y)) \) is hard. Both are non-repeatable, yet not expressible in 1RA^−.
- Possible (though expensive) approach: Translate to RC^\land and check RC-hierarchical and \( \forall\) consistency.