Query Compilation: the View from the Database Side

Dan Suciu
University of Washington

Joint with: P. Beame, N. Dalvi, A. Jha, J. Li, S. Roy
The View from the Database Side

- The large Boolean formula $F$ is generated by some much smaller program $Q$

- Each $Q$ defines a different problem:
  - KC, or SAT-solver, or … for formulas $F$ produced by $Q$
  - *Data complexity*: query $Q$, database $D$

- This talk:
  - $Q$ is a sentence in $\text{FO}(\land, \lor, \forall)$
  - The problem is model counting $\#F$ and KC

- Color code: blue=fixed, red=input
Sources

- Jha, S., ICDT 2011
- Dalvi, S., JACM 2012
- Beame, Li, Roy, S. UAI’2013
- Beame, Li, Roy, S. ICDT’2014

- Background on probabilistic databases:
Model Counting

• Given Boolean formula $F$, compute the number of models $\#F$

Example:
$F = X_1 X_2 \lor X_2 X_3 \lor X_3 X_1$

$\#F = 4$

[Valiant] $\#P$-hard, even for 2CNF
Probability of a Formula

- Each variable $X$ has a probability $p(X)$;
- $P(F) =$ probability that $F=\text{true}$

Example:
$F = X_1 \lor X_2 \lor X_2 X_3 \lor X_3 X_1$

$P(F) = (1-p_1)p_2p_3 + p_1(1-p_2)p_3 + p_1p_2(1-p_3) + p_1p_2p_3$

$P(F) = \#F / 2^n$, when $p(X) = \frac{1}{2}$ for all $X$
Grounding of an FO Sentence

Let $Q$, be an FO sentence, $n$ a natural number.

**Def** The grounding, $F_n(Q)$ is:

- $F_n(\forall x Q) = \bigwedge_{i \in [n]} F_n(Q[i/x])$
- $F_n(\exists x Q) = \bigvee_{i \in [n]} F_n(Q[i/x])$
- $F_n(Q_1 \text{ op } Q_2) = F_n(Q_1) \text{ op } F_n(Q_1)$ \hspace{1cm} $\text{op} = \land, \lor, \neg$

**Example:** $Q = \forall d (\text{Rain}(d) \Rightarrow \text{Cloudy}(d)) \quad n = 7$

$F_7(Q) = (\text{Rain}_1 \Rightarrow \text{Cloudy}_1) \land \ldots \land (\text{Rain}_7 \Rightarrow \text{Cloudy}_7)$

Probabilistic Databases, Markov Logic Networks, …
Research Question

Given an FO sentence $Q$ determine the complexity of $P(F_n(Q))$; PTIME? #P-hard?

**Data complexity**: assume fixed $Q$, input given by $n$

In practice, $Q$ is small:
- SQL query: 10-20 joins
- MLN’s: 10-15 rules

**Next**: knowledge compilation for $F_n(Q)$
Outline

• Problem statement

• Review: FBDD, Decision-DNNF

• Hard Queries

• Easy Queries

• Hard/Easy Queries

• Conclusion
Knowledge Compilation Targets

- FBDD (Free Binary Decision Diagram)
- Decision-DNNF (Decomposable Negation Normal Form)

$P(F)$ computable in linear time in the KC
Knowledge Compilation Targets

FBDD:
Decision-, sink-nodes

Decision-DNNF
add: decomposable-\(\land\)-nodes

Children of \(\land\) have disjoint sets of variables
DPLL and Knowledge Compilation

Fact: Trace of full-search DPLL $\rightarrow$ KC:

- Basic DPLL $\rightarrow$ decision trees
- DPLL + caching $\rightarrow$ FBDD
- DPLL + caching + components $\rightarrow$ decision-DNNF

Our interest in KC: lower bounds for DPLL.
Research Question

Given an FO sentence $Q$

<table>
<thead>
<tr>
<th>Determine the complexity of $P(F_n(Q))$; PTIME? $#P$-hard?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determine the size of $KC$ for $F_n(Q)$</td>
</tr>
</tbody>
</table>

“Data complexity”: fixed $Q$, input given by $n$
Outline

• Problem statement
• Review: FBDD, Decision-DNNF
  • Hard Queries
• Easy Queries
• Hard/Easy Queries
• Conclusion
Theorem [Bollig&Wegener’98] Any FBDD for $F = \bigwedge_{(i,j) \in E} (R_i \lor T_j)$ has size $2^{\Omega(\sqrt{n})}$

Where

- $R_1, \ldots, R_n, T_1, \ldots, T_n = \text{Boolean Variables}$
- $E = \ldots(\text{some complex relation } \subseteq [n] \times [n])$

For $p = \text{a prime}$, $n = p^2$,

$E = \{(1+i, 1+j)| i = a + bp, j = c + dp, c = a + bd \text{ mod } p\}$

$|E| = p^3 = n^{3/2}$
$H_0$ is Hard for FBDDs

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

$$F_n(H_0) = \bigwedge_{i \in [n], j \in [n]} (R_i \lor S_{ij} \lor T_j)$$

By [B&W], any FBDD has size $2^{\Omega(\sqrt{n})}$. We strengthen:

**Th.** [Beame’14] Any FBDD for $F_n(H_0)$ has size $\geq 2^{n-1}/n$.

What about Decision-DNNFs?
Decision-DNNF to FBDD

We proved this in [Beame’13]:

**Theorem** If $F$ has a Decision-DNNF with $N$ nodes, then $F$ has an FBDD with at most $N^{1+\log(N)}$ nodes.

**Proof idea**

Problem: $0 \land 1 \land 0 \land 1$

Solution: copy the smaller child

Optimal [Razgon]
H₀ is Hard for Decision-DNNFs

Corollary Any Decision-DNNF for \( F_n(H₀) \) has size \( 2^{Ω(\sqrt{n})} \)

Proof. \( N \)-node Decision-DNNF to \( N^{1+\log(N)} \) nodes FBDD.

\[
N^{1+\log(N)} > 2^{n-1}/n , \\
\log(N) + \log^2(N) > n - 1 - \log(n) \\
\log^2(N) = Ω(n) \\
\log(N) = Ω(\sqrt{n})
\]
Generalization

\[ C = \text{a positive clause; } \quad at(x) = \text{set of atoms containing variable } x \]

**Definition**  \( C \) is **hierarchical** if for all \( x, y \):
\[ at(x) \subseteq at(y) \quad \text{or} \quad at(x) \supseteq at(y) \quad \text{or} \quad at(x) \cap at(y) = \emptyset \]

A query \( Q \) in FO(\( \land, \lor, \forall \)) is hierarchical if all its clauses are

**Hierarchical**

\[ Q = R(x,y) \lor S(x,z) \]

**Non-hierarchical**

\[ H_0 = R(x) \lor S(x,y) \lor T(y) \]

**Thrm.** If \( Q \) is non-hierarchical, any Decision-DNNF has size \( 2^{\Omega(\sqrt{n})} \).
Discussion

Exponential size of KC not surprising, because:

**Theorem** \(#F_n(H_0)\) is \#P-hard.
(Same holds for any non-hierarchical \(Q\))

**Proof:**
[Provan&Ball’82] PP2CNF is \#P-complete:

\[
F = \bigwedge_{(i,j) \in E} (R_i \lor T_j)
\]
Research Question

Given an FO sentence $Q$ in FO($\wedge, \vee, \forall$)

<table>
<thead>
<tr>
<th>Non-hierarchical $Q$ (e.g. $H_0$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is $P(F_n(Q))$ in PTIME? Or #$P$-hard?</td>
</tr>
<tr>
<td>How large is Knowledge Compilation for $F_n(Q)$?</td>
</tr>
</tbody>
</table>

What about hierarchical queries?
Outline

• Problem statement
• Review: FBDD, Decision-DNNF
• Hard Queries
  • Easy Queries
• Hard/Easy Queries
• Conclusion
Easy Queries

• Let \( Q \) in FO(\( \land, \lor, \forall \)). Then \( F_n(Q) \) has a polynomial-size OBDD iff it is both hierarchical \textit{and} inversion-free.

• Recall: OBDD = FBDD with fixed variable order \( \Pi \)
Inversion-Free Queries

**Definition** An inversion in Q is a sequence of co-occurring vars:

\[(x_0,y_0), (x_1,y_1), \ldots, (x_k,y_k),\] such that:

- \(\text{at}(x_0) \not\subseteq \text{at}(y_0), \text{at}(x_1) = \text{at}(y_1), \ldots, \text{at}(x_{k-1}) = \text{at}(y_{k-1}), \text{at}(x_k) \nsubseteq \text{at}(y_k)\)
- For all \(i=1,\ldots,k-1\) there exists two atoms in Q of the form: \(S_i(\ldots,x_{i-1},\ldots,y_{i-1},\ldots)\) and \(S_i(\ldots,x_i,\ldots,y_i,\ldots)\)

Inversion-free implies hierarchical, but converse fails

\[Q = [R(x_0) \lor S(x_0,y_0)] \land [S(x_1,y_1) \lor T(x_1)]\]

Inversion-free

\[H_1 = [R(x_0) \lor S(x_0,y_0)] \land [S(x_1,y_1) \lor T(y_1)]\]

Inversion
Theorem [Jha&S.11] Let $Q$ in $\text{FO}(\land, \lor, \forall)$
1. If $Q$ has inversion then OBDD for $F_n(Q)$ has size $2^{\Omega(n)}$
2. Else, $F_n(Q)$ has OBDD of width $2^{\#\text{atoms}(Q)}$ (linear size)

Proof (part 2 only – next slide)
\[ C_1 = R(x) \lor S(x, y) \quad \land \quad C_2 = T(x') \land S(x', y') \quad = \quad Q = [R(x) \lor S(x, y)] \land [T(x') \lor S(x', y')] \]

\[ F(C_1) = (R_1 \lor S_{11}) \land (R_1 \lor S_{12}) \land (R_2 \lor S_{21}) \land (R_2 \lor S_{22}) \]

\[ n = 2 \quad \Pi = R_1 T_1 S_{11} S_{12} R_2 T_2 S_{21} S_{22} \]

\[ x = 1 \quad x = 2 \]

OBDD for \( Q = C_1 \land C_2 \) has width = \( \text{width}_1 \times \text{width}_2 \)

Same variable order \( \Pi \) in both OBDDs!
## Research Question

Given an FO sentence $Q$ in $\text{FO}(\land, \lor, \forall)$

<table>
<thead>
<tr>
<th>Non-hierarchical $Q$ (e.g. $H_0$)</th>
<th>Inversion-free $Q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is $P(F_n(Q))$ in PTIME? Or $#P$-hard?</td>
<td>$#P$-hard</td>
</tr>
<tr>
<td>How large is knowledge compilation for $F_n(Q)$?</td>
<td>decision-DNNF has size $2^{\Omega(\sqrt{n})}$</td>
</tr>
</tbody>
</table>

What about hierarchical queries w/ inversion?
Outline

• Problem statement

• Review: FBDD, Decision-DNNF

• Hard Queries

• Easy Queries

• **Hard/Easy Queries**

• Conclusion
Easy/Hard Queries

Will describe a class of queries $Q$ such that:

- Computing probability is easy $(P(F_n(Q))$ in PTIME)
- Compiling $F_n(Q)$ is hard (Exponential-size Decision-DNNF)

- Implication: inherent limitation of DPLL-based algorithms for model counting
The Queries $H_k$

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

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

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

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

Non-hierarchical

Hierarchical

Inversion: $at(x_0) \supset at(y_0)$, $at(x_1) \subset at(y_1)$

Longer inversion:

$at(x_0) \supset at(y_0)$, $at(x_1) = at(y_1)$, $at(x_2) = at(y_2)$, $at(x_3) \subset at(y_3)$:
Easy/Hard Queries

The clauses of $H_k$ (dropping $\forall$)

\[ H_{k0} = R(x_0) \lor S_1(x_0,y_0) \]
\[ H_{k1} = S_1(x_1,y_1) \lor S_2(x_1,y_1) \]
\[ H_{k2} = S_2(x_2,y_2) \lor S_3(x_2,y_2) \]
\[ \ldots \]
\[ \ldots \]
\[ H_{kk} = S_k(x_k,y_k) \lor T(y_k) \]

$f(Z_0, Z_1, \ldots, Z_k)$ = a Boolean function in $k+1$ variables

$Q = f(H_{k0}, H_{k1}, \ldots, H_{kk})$

Example: $f = Z_0 \land Z_1 \land \ldots \land Z_k$ then $f(H_{k0}, H_{k1}, \ldots, H_{kk}) = H_k$
Easy/Hard Queries

\[ f(Z_0, Z_1, \ldots, Z_k) = \text{Boolean function in } k+1 \text{ vars} \]
\[ Q = f(H_{k0}, H_{k1}, \ldots, H_{kk}) \]

**Theorem** [Beame’14] Any FBDD for \( F_n(Q) \) has size \( 2^{\Omega(n)} \)
Any Decision-DNNF has size \( \geq 2^{\Omega(\sqrt{n})} \).

**Theorem** [Dalvi’12] Assume \( f \) is monotone, let \( L \) be its DNF lattice, \( \mu \) its Möbius function

- If \( \mu(0, 1) = 0 \) then \( P(F_n(Q)) \) is in PTIME
- If \( \mu(0, 1) \neq 0 \) then \( P(F_n(Q)) \) is in \#P-hard
Proof Highlights

**Theorem** [Beame’14] Any FBDD for $F_n(Q)$ has size $2^{\Omega(n)}$.

**Proof part 1:** any FBDD for $F_n(H_k)$ has size $\geq 2^{n-1}/n$.

**Proof part 2:**
Convert a $N$-node FBDD for $F_n(f(H_{k0}, H_{k1}, \ldots, H_{kk}))$, to a $O(n^3N)$-node multi-output FBDD for $k+1$ functions: $F_n(H_{k0}), F_n(H_{k1}), \ldots, F_n(H_{kk})$.

Convert the latter to an FBDD for $F_n(H_k)$.
Proof Highlights

**Theorem** [Dalvi’12] If \( \mu = 0 \) then \( P(F_n(Q)) \) is in PTIME

By example on \( f = Z_0 \land Z_2 \lor Z_0 \land Z_3 \lor Z_1 \land Z_3 \)

\[
Q_W = H_{30} \land H_{32} \lor /* Q_1 */ \\
H_{30} \land H_{33} \lor /* Q_2 */ \\
H_{31} \land H_{33} /* Q_3 */
\]

Recall:
\( H_3 = H_{30} \land \ldots \land H_{33} \)

\[
P(Q_W) = P(Q_1) + P(Q_2) + P(Q_3) + \\
- P(Q_1 \land Q_2) - P(Q_2 \land Q_3) - P(Q_1 \land Q_3) \\
+ P(Q_1 \land Q_2 \land Q_3)
\]

Also = \( H_3 \)

The remaining terms are inversion-free, hence PTIME
The DNF Lattice

Definition.
The DNF lattice $L$ of a monotone DNF $f = t_1 \lor t_2 \lor \ldots$ is:
• Elements of $L$ are terms $t_{i_1} \land t_{i_2} \land \ldots$;
• Order is logical implication

$$f = Z_0 \land Z_2 \lor Z_0 \land Z_3 \lor Z_1 \land Z_3$$

Fact: if $\mu = 0$ then $P(F_n(Q))$ is in PTIME

Nodes • in PTIME,
Nodes • #P hard.
Research Question

Given an FO sentence $Q$

<table>
<thead>
<tr>
<th>Non-hierarchical $Q$ (e.g. $H_0$)</th>
<th>Inversion-free $Q$</th>
<th>$Q = f(H_{k0}, \ldots, H_{kk})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is $P(F_n(Q))$ in PTIME? Or #P-hard?</td>
<td>#P-hard</td>
<td>PTIME or #P-hard</td>
</tr>
<tr>
<td>How large is Knowledge Compilation for $F_n(Q)$?</td>
<td>size $2^{\Omega(\sqrt{n})}$</td>
<td>Poly-size</td>
</tr>
</tbody>
</table>
Outline

• Problem statement
• Review: FBDD, Decision-DNNF
• Hard Queries
• Easy Queries
• Hard/Easy Queries
• Conclusion
The View from the Database Side

High level idea:
• Boolean function $F$ generated by small program $Q$

For FO sentence $Q$ in FO($\land$, $\lor$, $\forall$)
• Hard/hard
• Easy/easy
• Easy/hard

Separation of grounded v.s. lifted inference:
• Limitation of DPLL-based algorithms
• Inclusion/exclusion possible only on the FO sentence
Möbius Über Alles

PTIME

Poly-size FBDD, dec-DNNF

Poly-size OBDD = inversion-free

Read Once

Non-hierarchical

hierarchical

#P-hard