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
Question 2. Are lifted rules stronger than grounded?

Alternative to lifting:

1. Ground the FO sentence
2. Do WMC on the propositional formula

- There is no reason why grounded inference should be weaker than lifted inference
- However, *existing* grounded algorithms are strictly weaker than lifted inference
Algorithms for Model Counting

[Gomes’08] Based on full search DPLL:

• Shannon expansion.
  \[ \#F = \#F[X=0] + \#F[X=1] \]

• Caching.
  Store \#F, look it up later

• Components. If \( \text{Vars}(F_1) \cap \text{Vars}(F_2) = \emptyset \):
  \[ \#(F_1 \land F_2) = \#F_1 \ast \#F_2 \]
Knowledge Compilation

**Definition** (informal): represent the Boolean formula $F$ in a circuit where $\text{WMC}(F)$ is in PTIME in the size of the representation

**Why we care:**

- The trace of any inference algorithm is a knowledge compilation
- Lower bounds on $\text{size}(\text{KC})$ give lower bounds on the algorithm’s runtime
Knowledge Compilation Targets

FBDD: Decision-, sink-nodes

OBDD: fixed variable order

Decision-DNNF add: $\land$-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
  $\rightarrow$ OBDD (fixed variable order)
  $\rightarrow$ FBDD
- DPLL + caching + components
  $\rightarrow$ decision-DNNF

[Huang&Darwiche’2005]
Hard Queries

\[ H_0 = \forall x \forall y \ (R(x) \lor S(x,y) \lor T(y)) = \text{non-hierarchical} \]
\[ H_k = \text{hierarchical, has inversion, for } k \geq 1 \]

Grounded Boolean formulas:
\[ F_n(H_0) = \land_{i \in [n], j \in [n]} (R_i \lor S_{ij} \lor T_j) \]

**Th. [Beame’14]** Any FBDD for \( F_n(H_k) \) has size \( \geq 2^{n-1}/n \).
Same holds for any non-hierarchical query.

What about Decision-DNNFs?
**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**

Optimal [Razgon]
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:
**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: $0 \times 1$

Solution: copy the smaller child
Hard Queries

**Corollary** Any Decision-DNNF for $F_n(H_k)$ has size $2^{\Omega(\sqrt{n})}$

Same holds for any non-hierarchical query.

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

\[ N^{1+\log(N)} > 2^{n-1}/n, \]
\[ \log(N) + \log^2(N) > n - 1 - \log(n) \]
\[ \log^2(N) = \Omega(n) \]
\[ \log(N) = \Omega(\sqrt{n}) \]
**Lifted v.s. Grounded Inference**

<table>
<thead>
<tr>
<th></th>
<th>Non-hierarchical $Q$ (e.g. $H_0$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lifted $P(Q)$</td>
<td>#P-hard</td>
</tr>
<tr>
<td>Grounded $P(F_n(Q))$</td>
<td>$2^\Omega(\sqrt{n})$</td>
</tr>
</tbody>
</table>

What about hierarchical 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:

1. \(\text{at}(x_0) \not\subseteq \text{at}(y_0), \text{at}(x_1) = \text{at}(y_1), \ldots, \text{at}(x_{k-1}) = \text{at}(y_{k-1}), \text{at}(x_k) \not\supseteq \text{at}(y_k)\)
2. 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

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

Theorem Let $Q$ in $\forall \mathbf{FO}^{un}$

1. If $Q$ has inversion then OBDD for $F_n(Q)$ has size $\geq 2^{n-1}/n$
2. Else, $F_n(Q)$ has OBDD of width $2^{\#\text{atoms}(Q)}$ (size $O(n)$)

Proof (part 2 only – next slide)

[Jha&S.11], [Beame’15]
**Theorem** Let $Q$ in $\forall \text{FO}^{un}$

1. If $Q$ has inversion then OBDD for $F_n(Q)$ has size $\geq 2^{n-1}/n$.
2. Else, $F_n(Q)$ has OBDD of width $2^\#\text{atoms}(Q)$ (size $O(n)$).

Proof (part 2 only – next slide)

[Beame&Liew’15] Extended to SDD. Thus, over $\forall \text{FO}^{un}$, OBDD $\approx$ SDD

[Jha&S.11], [Beame’15]
Easy Queries

Theorem Let $Q$ in $\forall FO^{un}$
1. If $Q$ has inversion then OBDD for $F_n(Q)$ has size $\geq 2^{n-1}/n$
2. Else, $F_n(Q)$ has OBDD of width $2^{\#atoms(Q)}$ (size $O(n)$)

Proof (part 2 only – next slide)

[Beame’15] SDD more succinct than OBDD (HWB)

[Beame&Liew’15] Extended to SDD. Thus, over $\forall FO^{un}$, OBDD $\approx$ SDD

[Jha&S.11], [Beame’15]

[Bova’16] SDD more succinct than OBDD (HWB)
\[ Q = [R(x) \lor S(x,y)] \land [T(x') \lor S(x',y')] \]
$Q = [R(x) \lor S(x, y)] \land [T(x') \lor S(x', y')]$

$n = 2$

$\Pi = R_1 T_1 S_{11} S_{12} R_2 T_2 S_{21} S_{22}$

$x = 1$

$x = 2$
\[ Q = [R(x) \lor S(x,y)] \land [T(x') \lor S(x',y')] \]

\( n = 2 \)

\( \Pi = R_1 T_1 S_{11} S_{12} R_2 T_2 S_{21} S_{22} \)

\( x = 1 \)

\( x = 2 \)

\( C_1 = R(x) \lor S(x,y) \land C_2 = T(x') \land S(x',y') \)

\( C_2 = T(x') \land S(x',y') \land Q = [R(x) \lor S(x,y)] \land [T(x') \lor S(x',y')] \)
\[ C_1 = R(x) \lor S(x,y) \quad \land \quad C_2 = T(x') \land S(x',y') \quad \land \quad Q = [R(x) \lor S(x,y)] \land [T(x') \lor S(x',y')] \]

\[ F_2(C_1) = (R_1 \lor S_{11}) \land (R_1 \lor S_{12}) \land (R_2 \lor S_{21}) \land (R_2 \lor S_{22}) \]

\[ n = 2 \]

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

\[ x = 1 \quad \quad x = 2 \]
\[ C_1 = R(x) \lor S(x, y) \quad \land \quad C_2 = T(x') \land S(x', y') = Q = [R(x) \lor S(x, y)] \land [T(x') \lor S(x', y')] \]

\[
F_2(C_1) = (R_1 \lor S_{11}) \land (R_1 \lor S_{12}) \land (R_2 \lor S_{21}) \land (R_2 \lor S_{22})
\]

\[n = 2\]

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

\[
x = 1 \quad \quad x = 2
\]
\[ Q = C_1 \land C_2 \]

\[ Q = [R(x) \lor S(x,y)] \land [T(x') \land S(x',y')] \]

\[ F_2(C_1) = (R_1 \lor S_{11}) \land (R_1 \lor S_{12}) \land (R_2 \lor S_{21}) \land (R_2 \lor S_{22}) \]

\[ n = 2 \]

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

Same variable order \( \Pi \) in both OBDDs!

OBDD for \( Q = C_1 \land C_2 \) has width = width1 \( \times \) width2
# Lifted v.s. Grounded Inference

<table>
<thead>
<tr>
<th></th>
<th>Non-hierarchical ( Q ) (e.g. ( H_0 ))</th>
<th>Inversion-free ( Q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lifted ( P(Q) )</td>
<td>#P-hard</td>
<td>PTIME</td>
</tr>
<tr>
<td>Grounded ( P(F_n(Q)) )</td>
<td>( 2^{\Omega(\sqrt{n})} )</td>
<td>PTIME</td>
</tr>
</tbody>
</table>
Easy/Hard Queries

Main result: a class of queries $Q$ such that:

- Lifted inference: $P((Q))$ in PTIME
- Grounded inference: $P(F_n(Q))$ exponential time

Significance: limitation of DPLL-based algorithms for model counting
Clauses of $H_k$

\[
\begin{align*}
H_{k0} &= \forall x \forall y \ R(x) \lor S_1(x,y) \\
H_{k1} &= \forall x \forall y \ S_1(x,y) \lor S_2(x,y) \\
H_{k2} &= \forall x \forall y \ S_2(x,y) \lor S_3(x,y) \\
&\quad \quad \cdots \\
&\quad \quad \cdots \\
H_{kk} &= \forall x \forall y \ S_k(x,y) \lor T(y)
\end{align*}
\]
Clauses of $H_k$

\[
\begin{align*}
H_{k0} &= \forall x \forall y \ R(x) \lor S_1(x,y) \\
H_{k1} &= \forall x \forall y \ S_1(x,y) \lor S_2(x,y) \\
H_{k2} &= \forall x \forall y \ S_2(x,y) \lor S_3(x,y) \\
\ldots \\
\ldots \\
H_{kk} &= \forall x \forall y \ S_k(x,y) \lor T(y)
\end{align*}
\]

\[f(Z_0, Z_1, \ldots, Z_k) = \text{a Boolean function}\]
Clauses of $H_k$

\[
H_{k0} = \forall x \forall y \ R(x) \lor S_1(x,y) \\
H_{k1} = \forall x \forall y \ S_1(x,y) \lor S_2(x,y) \\
H_{k2} = \forall x \forall y \ S_2(x,y) \lor S_3(x,y) \\
\ldots \\
\ldots \\
H_{kk} = \forall x \forall y \ S_k(x,y) \lor T(y)
\]

$f(Z_0, Z_1, \ldots, Z_k) = \text{a Boolean function}$

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

$H_{k0} = \forall x \forall y \ R(x) \lor S_1(x,y)$
$H_{k1} = \forall x \forall y \ S_1(x,y) \lor S_2(x,y)$
$H_{k2} = \forall x \forall y \ S_2(x,y) \lor S_3(x,y)$

$\ldots$

$\ldots$

$H_{kk} = \forall x \forall y \ S_k(x,y) \lor T(y)$

Examples:

$f = Z_0 \land Z_1 \land \ldots \land Z_k$ then $f(H_{k0}, H_{k1}, \ldots, H_{kk}) = H_k$

$f = Z_0 \land Z_2 \lor Z_0 \land Z_3 \lor Z_1 \land Z_3$ then $f(H_{30}, H_{31}, H_{31}, H_{33}) = Q_W$

$f(Z_0, Z_1, \ldots, Z_k) = \text{a Boolean function}$

$Q = f(H_{k0}, H_{k1}, \ldots, H_{kk})$
Easy/Hard Queries

[Beame’14]

**Theorem** For any Boolean function \( f(Z_0, Z_1, \ldots, Z_k) \), denoting \( Q = f(H_{k0}, H_{k1}, \ldots, H_{kk}) \):

- Any FBDD for \( F_n(Q) \) has size \( 2^{\Omega(n)} \)
- Any Decision-DNNF has size \( 2^{\Omega(\sqrt{n})} \).

Consequence:

- Lifted inference computes \( P(Q_W) \) in PTIME
- Any DPLL-based algorithm takes time \( 2^{\Omega(\sqrt{n})} \)

Many other queries are like \( Q_W \)
### Lifted v.s. Grounded Inference

<table>
<thead>
<tr>
<th></th>
<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>Lifted P($Q$)</td>
<td>#P-hard</td>
<td>PTIME</td>
<td>PTIME or #P-hard</td>
</tr>
<tr>
<td>Grounded P($F_n(Q)$)</td>
<td>$2^{\Omega(\sqrt{n})}$</td>
<td>PTIME</td>
<td>$2^{\Omega(\sqrt{n})}$</td>
</tr>
</tbody>
</table>
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
Möbius Über Alles

∀FO^{un}, ∃FO^{un}

#P-hard
Möbius Über Alles

∀FO^{un}, ∃FO^{un}

#P-hard

PTIME

Q_W

Q_9
Möbius Über Alles

∀FO_{un}, ∃FO_{un}

#P-hard

PTIME

∀FO_{un}, ∃FO_{un}

Q_w

Q_9

Read Once

Q_u
Möbius Über Alles

∀ FO^{un}, ∃ FO^{un}

#P-hard

PTIME

Poly-size OBDD, SDD = = inversion-free

Read Once

∀ FO^{un}, ∃ FO^{un}
Möbius Über Alles

∀FO^{un}, ∃FO^{un}

#P-hard

PTIME

Poly-size FBDD, dec-DNNF

Poly-size OBDD, SDD = inversion-free

Read Once
Möbius Über Alles

Non-hierarchical

#P-hard

PTIME

Poly-size FBDD, dec-DNNF

Poly-size OBDD, SDD = inversion-free

Read Once

∀FO^{un}, ∃FO^{un}

Non- hierarchical

hierarchical

H_0

H_1

H_2

H_3

Q_w

Q_9

Q_v

Q_j

Q_u

Q_F

Open