First-Order Knowledge Compilation for Probabilistic Reasoning

Guy Van den Broeck

based on joint work with Adnan Darwiche, Dan Suciu, and many others
MOTIVATION 1
A Simple Reasoning Problem

Probability that Card1 is Hearts?
A Simple Reasoning Problem

Probability that Card1 is Hearts? 1/4

[Van den Broeck; AAAI-KRR’15]
A Simple Reasoning Problem

Probability that Card52 is Spades given that Card1 is QH?
A Simple Reasoning Problem

Probability that Card52 is Spades given that Card1 is QH? 13/51

[Van den Broeck; AAAI-KRR’15]
Automated Reasoning

Let us automate this:

1. CNF encoding for deck of cards
2. Compile to tractable knowledge base (e.g., d-DNNF)
3. Condition on observations/questions
   “Card1 is hearts”
4. Model counting
Automated Reasoning

Let us automate this:

1. CNF encoding for deck of cards
2. Compile to tractable knowledge base (e.g., d-DNNF)
3. Condition on observations/questions
   “Card1 is hearts”
4. Model counting

A typical BeyondNP pipeline!
Let us automate this:

1. **CNF encoding for deck of cards**

   Card(p1,c1) v Card(p1,c2) v ...
   Card(p1,c1) v Card(p2,c1) v ...
   ¬Card(p1,c1) v ¬Card(p1,c2)
   ¬Card(p1,c2) v ¬Card(p1,c3)
   ...
   ¬Card(p2,c1) v ¬Card(p2,c2)
   ...
Automated Reasoning

Let us automate this:

1. CNF encoding for deck of cards
2. **Compile to tractable knowledge base (e.g., d-DNNF)**
3. Condition on observations/questions
   - “Card1 is hearts”
4. Model counting

Which language to choose?
Cards problem is easy: we want to be polynomial.
2. Compile to tractable knowledge base
3. Condition on observations/questions
4. Model counting
2. Compile to tractable knowledge base
3. Condition on observations/questions
4. Model counting
2. Compile to tractable knowledge base
3. Condition on observations/questions
4. Model counting
2. Compile to tractable knowledge base
3. **Condition on observations/questions**
4. Model counting

\[ \neg \text{Card(K♥,p14)} \]
2. Compile to tractable knowledge base
3. Condition on observations/questions
4. **Model counting**
2. Compile to tractable knowledge base
3. Condition on observations/questions
4. Model counting: How many perfect matchings?
2. Compile to tractable knowledge base
3. Condition on observations/questions
4. **Model counting**: How many *perfect matchings*?
Observations

• Deck of cards = complete bigraph
• CD = removing edges in bigraph
  Encode any bigraph in cards problem
• CT = counting perfect matchings
• Problem is \#P-complete!

No language with CD and CT can represent the cards problem compactly, unless P=NP.
What's Going On Here?

Probability that Card52 is Spades given that Card1 is QH?

[Van den Broeck; AAAI-KRR’15]
What's Going On Here?

Probability that Card52 is Spades given that Card1 is QH?

13/51

[Van den Broeck; AAAI-KRR’15]
What's Going On Here?

Probability that Card52 is Spades given that Card1 is QH?

13/51

[Van den Broeck; AAAI-KRR’15]
What's Going On Here?

Probability that Card52 is Spades given that Card2 is QH?
What's Going On Here?

Probability that Card52 is Spades given that Card2 is QH?

13/51

[Van den Broeck; AAAI-KRR’15]
What's Going On Here?

Probability that Card52 is Spades given that Card3 is QH?
What's Going On Here?

Probability that Card52 is Spades given that Card3 is QH? 13/51

[Van den Broeck; AAAI-KRR’15]
Tractable Reasoning

What's going on here?
Which property makes reasoning tractable?

[Niepert, Van den Broeck; AAAI’14], [Van den Broeck; AAAI-KRR’15]
What's going on here?

Which property makes reasoning tractable?

- High-level (first-order) reasoning
- Symmetry
- Exchangeability

⇒ Lifted Inference

[Niepert, Van den Broeck; AAAI’14], [Van den Broeck; AAAI-KRR’15]
Let us automate this:

- **Relational/FO** model

\[
\forall p, \exists c, \text{Card}(p,c) \\
\forall c, \exists p, \text{Card}(p,c) \\
\forall p, \forall c, \forall c’, \text{Card}(p,c) \land \text{Card}(p,c’) \Rightarrow c = c’
\]

- **First-Order Knowledge Compilation**
MOTIVATION 2
**Model Counting**

- Model = solution to a propositional logic formula $\Delta$
- Model counting = $\#\text{SAT}$

<table>
<thead>
<tr>
<th>Rain</th>
<th>Cloudy</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

$\Delta = (\text{Rain} \Rightarrow \text{Cloudy})$

$\#\text{SAT} = 3$

[Valiant] $\#P$-hard, even for 2CNF
Weighted Model Counting

- Model = solution to a propositional logic formula $\Delta$
- Model counting = #SAT

$\Delta = (\text{Rain } \Rightarrow \text{Cloudy})$

<table>
<thead>
<tr>
<th>Rain</th>
<th>Cloudy</th>
<th>Model?</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>Yes</td>
</tr>
</tbody>
</table>

#SAT = 3
Weighted Model Counting

- Model = solution to a propositional logic formula $\Delta$
- Model counting = $\#\text{SAT}$
- Weighted model counting (WMC)
  - Weights for assignments to variables
  - Model weight is product of variable weights $w(.)$

$\Delta = (\text{Rain} \Rightarrow \text{Cloudy})$

<table>
<thead>
<tr>
<th>Rain</th>
<th>Cloudy</th>
<th>Model?</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>Yes</td>
<td>1 * 3  = 3</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>Yes</td>
<td>2 * 3  = 6</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>Yes</td>
<td>2 * 5  = 10</td>
</tr>
</tbody>
</table>

$\#\text{SAT} = 3$
Weighted Model Counting

• Model = solution to a propositional logic formula $\Delta$
• Model counting = $\#SAT$
• Weighted model counting (WMC)
  – Weights for assignments to variables
  – Model weight is product of variable weights $w(.)$

$\Delta = (\text{Rain} \Rightarrow \text{Cloudy})$

<table>
<thead>
<tr>
<th></th>
<th>Rain</th>
<th>Cloudy</th>
<th>Model?</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>Yes</td>
<td>$1 \times 3 = 3$</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>No</td>
<td>$2 \times 3 = 6$</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>Yes</td>
<td>$2 \times 5 = 10$</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#SAT = 3  
WMC = 19
Assembly language for probabilistic reasoning and learning

- Bayesian networks
- Factor graphs
- Probabilistic logic programs
- Markov Logic
- Relational Bayesian networks
- Probabilistic databases
- Weighted Model Counting
First-Order Model Counting

Model = solution to first-order logic formula $\Delta$

$\Delta = \forall d \ (\text{Rain}(d) \Rightarrow \text{Cloudy}(d))$

Days = \{Monday\}
First-Order Model Counting

Model = solution to first-order logic formula $\Delta$

$\Delta = \forall d \ (\text{Rain}(d) \Rightarrow \text{Cloudy}(d))$

Days = {Monday}

<table>
<thead>
<tr>
<th>Rain(M)</th>
<th>Cloudy(M)</th>
<th>Model?</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>Yes</td>
</tr>
</tbody>
</table>

$\text{FOMC} = 3$
Weighted First-Order Model Counting

Model = solution to *first-order* logic formula $\Delta$

$$\Delta = \forall d \ (\text{Rain}(d) \Rightarrow \text{Cloudy}(d))$$

Days = \{Monday, Tuesday\}

<table>
<thead>
<tr>
<th>Rain(M)</th>
<th>Cloudy(M)</th>
<th>Rain(T)</th>
<th>Cloudy(T)</th>
<th>Model?</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>No</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>Yes</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>Yes</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Weighted First-Order Model Counting

Model = solution to first-order logic formula $\Delta$

$\Delta = \forall d \ (\text{Rain}(d) \Rightarrow \text{Cloudy}(d))$

Days = \{Monday, Tuesday\}

\[ \text{Rain}(M) \quad \text{Cloudy}(M) \quad \text{Rain}(T) \quad \text{Cloudy}(T) \quad \text{Model?} \]

\[
\begin{array}{ccc}
T & T & T & T & \text{Yes} \\
T & F & T & T & \text{Yes} \\
F & T & T & T & \text{Yes} \\
F & F & T & T & \text{Yes} \\
T & T & T & F & \text{No} \\
T & F & T & F & \text{No} \\
F & T & T & F & \text{No} \\
F & F & T & F & \text{No} \\
T & T & F & T & \text{Yes} \\
T & F & F & T & \text{Yes} \\
F & T & F & T & \text{Yes} \\
F & F & F & T & \text{Yes} \\
T & T & F & F & \text{Yes} \\
T & F & F & F & \text{Yes} \\
F & T & F & F & \text{Yes} \\
F & F & F & F & \text{Yes} \\
\end{array}
\]

$\#\text{SAT} = 9$
Weighted First-Order Model Counting

Model = solution to first-order logic formula $\Delta$

$$\Delta = \forall d \ (\text{Rain}(d) \Rightarrow \text{Cloudy}(d))$$

Days = \{Monday, Tuesday\}

$\begin{align*}
\text{w( R)} &= 1 \\
\text{w( R)} &= 2 \\
\text{w( C)} &= 3 \\
\text{w( C)} &= 5
\end{align*}$

<table>
<thead>
<tr>
<th>Rain(M)</th>
<th>Cloudy(M)</th>
<th>Rain(T)</th>
<th>Cloudy(T)</th>
<th>Model?</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
<td>$1 \times 1 \times 3 \times 3 = 9$</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>No</td>
<td>$2 \times 1 \times 3 \times 3 = 18$</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
<td>$2 \times 1 \times 5 \times 3 = 30$</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
<td>$2 \times 1 \times 5 \times 3 = 30$</td>
</tr>
</tbody>
</table>

$\#\text{SAT} = 9$
Weighted First-Order Model Counting

Model = solution to first-order logic formula $\Delta$

$\Delta = \forall d \ (\text{Rain}(d) \Rightarrow \text{Cloudy}(d))$

Days = {Monday, Tuesday}

$\text{w}(\text{R})=1$
$\text{w}(\neg\text{R})=2$
$\text{w}(\text{C})=3$
$\text{w}(\neg\text{C})=5$

<table>
<thead>
<tr>
<th>Rain(M)</th>
<th>Cloudy(M)</th>
<th>Rain(T)</th>
<th>Cloudy(T)</th>
<th>Model?</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
<td>$1 \times 1 \times 3 \times 3 = 9$</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
<td>$2 \times 1 \times 3 \times 3 = 18$</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>Yes</td>
<td>$2 \times 1 \times 5 \times 3 = 30$</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>Yes</td>
<td>$1 \times 2 \times 3 \times 3 = 18$</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>Yes</td>
<td>$2 \times 2 \times 3 \times 3 = 36$</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>Yes</td>
<td>$2 \times 2 \times 5 \times 3 = 60$</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>Yes</td>
<td>$1 \times 2 \times 3 \times 5 = 30$</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>Yes</td>
<td>$2 \times 2 \times 3 \times 5 = 60$</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>Yes</td>
<td>$2 \times 2 \times 5 \times 5 = 100$</td>
</tr>
</tbody>
</table>

$\#\text{SAT} = 9$  $\text{WFOMC} = 361$
Assembly language for high-level probabilistic reasoning and learning

- Parfactor graphs
- Probabilistic logic programs
- Markov Logic
- Relational Bayesian networks
- Probabilistic databases

Weighted First-Order Model Counting

[VdB et al.; IJCAI’11, PhD’13, KR’14, UAI’14]
An MLN = set of constraints \((w, \Gamma(x))\)

Weight of a world = product of \(w\), for all rules \((w, \Gamma(x))\) and groundings \(\Gamma(a)\) that hold in the world

\[
P_{MLN}(Q) = \frac{\text{[sum of weights of worlds of } Q\text{]}}{Z}
\]

Applications: large probabilistic KBs
FO NNF SYNTAX
First-Order Knowledge Compilation

• Input: Sentence in FOL
• Output: Representation tractable for some class of queries.

• In this work:
  – Function-free FOL
  – Model counting in NNF tradition

• Some pre-KC-map work:
  – FO Horn clauses
  – FO BDDs
Alphabet

• FOL
  – Predicates/relations: Friends
  – Object names: x, y, z
  – Object variables: X, Y, Z
  – Symbols classical FOL (∀, ∃, ∧, ∨, ¬,...)

• Group logic
  – Group variables: X, Y, Z
  – Symbols from basic set theory
    (e.g., ∪, ∩, ∈, ⊆, {, }, complement).
Syntax

- Object terms: $X$, alice, bob
- Group terms: $X$, $\{\text{alice,bob}\}$, $X \cup Y$
- Atom: $\text{Friends(alice,X)}$
- Formulas:
  - $(\alpha)$, $\neg\alpha$, $\alpha \lor \beta$, and $\alpha \land \beta$
  - $\forall X \in G$, $\alpha$ and $\exists X \in G$, $\alpha$
  - $\forall X \subseteq G$, $\alpha$ and $\exists X \subseteq G$, $\alpha$
- Group logic syntactic sugar:
  - $P(G)$ is $\forall X \in G$, $P (X)$
  - $\bar{P}(G)$ is $\forall X \in G$, $\neg P (X)$
Examples:

- $\forall X \in G$, $Y \in \{\text{alice, bob}\}$,
  Enemies($X$, $Y$)
  $\Rightarrow \neg$Friends($X$, $Y$) $\land \neg$Friends($Y$, $X$)

- $\forall X \in G$, $Y \in G$,
  Smokes($X$) $\land$ Friends($X$, $Y$) $\Rightarrow$ Smokes($Y$)

- $\exists G \subseteq \{\text{alice, bob}\}$, Smokes($G$) $\land \overline{\text{Healthy} (G)}$
Semantics

• Template language for propositional logic

• Grounding a sentence: \( \text{gr}(\alpha) \)
  • Replace \( \forall \) by \( \land \)
  • Replace \( \exists \) by \( \lor \)
  • End result: ground sentence = propositional logic

• Grounding is polynomial in group sizes when no \( \forall X \subseteq G \) or \( \exists X \subseteq G \)

Important for polytime reduction to NNF circuits
Decomposability

• **Conjunction:** \( \alpha(X,G) \land \beta(X,G) \)
  For any substitution \( X=c \) and \( G=g \), we have that \( \text{gr}(\alpha(c,g)) \land \text{gr}(\beta(c,g)) \) is decomposable

  Meaning: \( \alpha \) and \( \beta \) can never talk about the same ground atoms

• **Quantifier:** \( \forall Y \in G, \alpha(Y) \)
  For any two \( a,b \in G \), we have that \( \text{gr}(\alpha(a)) \land \text{gr}(\alpha(b)) \) is decomposable
Determinism

- **Disjunction**: $\alpha(X, G) \lor \beta(X, G)$
  
  For any substitution $X=c$ and $G=g$, we have that $\text{gr}(\alpha(c,g)) \lor \text{gr}(\beta(c,g))$ is deterministic

  Meaning: $\alpha \land \beta$ is UNSAT

- **Quantifier**: $\exists Y \in G, \alpha(Y)$
  
  For any two $a, b \in G$, we have that $\text{gr}(\alpha(a)) \lor \text{gr}(\alpha(b))$ is decomposable
Group Quantifiers

- **Decomposability**: \( \forall X \subseteq G, \alpha(X) \)
  
  For any two \( A, B \subseteq G \), we have that \( \text{gr}(\alpha(A)) \lor \text{gr}(\alpha(B)) \) is decomposable

- **Determinism**: \( \exists X \subseteq G, \alpha(X) \)
  
  For any two \( A, B \subseteq G \), we have that \( \text{gr}(\alpha(A)) \lor \text{gr}(\alpha(B)) \) is deterministic
Automorphism

- Object permutation $\sigma : D \rightarrow D$ is a one-to-one mapping from objects to objects.
- Permuting $\alpha$ using $\sigma$ replaces $o$ in $\alpha$ by $\sigma(o)$.
- Sentences $\alpha$ and $\beta$ are p-equivalent iff $\alpha$ is equivalent to an object permutation of $\beta$.
  Smokes(alice) and Smokes(bob) are p-equivalent.
- Group quantifiers: $\forall X \subseteq G$, $\alpha(X)$ or $\exists X \subseteq G$, $\alpha(X)$ are automorphic iff for any two $A,B \subseteq G$ s.t. $|A|=|B|$, gr($\alpha(A)$) and gr($\alpha(B)$) are p-equivalent.
First-Order NNF

∀X, X ∈ People : belgian(X) ⇒ likes(X, chocolate)
First-Order NNF

$$\forall X, X \in \text{People} : \text{belgian}(X) \Rightarrow \text{likes}(X, \text{chocolate})$$
First-Order DNNF

$$\forall X, X \in \text{People} : \text{belgian}(X) \Rightarrow \text{likes}(X, \text{chocolate})$$
First-Order DNNF

\[ \forall X, X \in \text{People} : \text{belgian}(X) \Rightarrow \text{likes}(X, \text{chocolate}) \]
First-Order d-DNNF

\( \forall X, X \in \text{People} : \text{belgian}(X) \Rightarrow \text{likes}(X, \text{chocolate}) \)
First-Order d-DNNF

\[ \forall X, X \in \text{People} : \text{belgian}(X) \Rightarrow \text{likes}(X, \text{chocolate}) \]
First-Order d-DNNF

∀X, X ∈ People : belgian(X) ⇒ likes(X, chocolate)
First-Order ad-DNNF

∀X, X ∈ People : belgian(X) ⇒ likes(X, chocolate)
FO NNF Languages

- FO NNF: group logic circuits, negation only on atoms
- FO d-DNNF: determinism and decomposability
  Grounding generates a d-DNNF
- FO DNNF
  Grounding generates a DNNF
- FO ad-DNNF: automorphic
  Powerful properties!
FO NNF TRACTABILITY
Symmetric WFOMC

Def. A weighted vocabulary is \((\mathbf{R}, \mathbf{w})\), where

- \(\mathbf{R} = (R_1, R_2, \ldots, R_k)\) = relational vocabulary
- \(\mathbf{w} = (w_1, w_2, \ldots, w_k)\) = weights

• Fix an FO formula \(Q\), domain of size \(n\)
• The weight of a ground tuple \(t\) in \(R_i\) is \(w_i\)

Complexity of FOMC / WFOMC\((Q, n)\)?

Data/domain complexity:

fixed \(Q\), input \(n\) / and \(w\)
Symmetric WFOMC on FO ad-DNNF

\[ U(\alpha) = \begin{cases} 
0 & \text{when } \alpha = \text{false} \\
1 & \text{when } \alpha = \text{true} \\
0.5 & \text{when } \alpha \text{ is a literal} \\
U(\ell_1) \times \cdots \times U(\ell_n) & \text{when } \alpha = \ell_1 \land \cdots \land \ell_n \\
U(\ell_1) + \cdots + U(\ell_n) & \text{when } \alpha = \ell_1 \lor \cdots \lor \ell_n \\
\prod_{i=1}^{n} U(\beta\{X/x_i\}) & \text{when } \alpha = \forall X \in \tau, \beta \text{ and } x_1, \ldots, x_n \text{ are the objects in } \tau. \\
\sum_{i=1}^{n} U(\beta\{X/x_i\}) & \text{when } \alpha = \exists X \in \tau, \beta \text{ and } x_1, \ldots, x_n \text{ are the objects in } \tau. \\
\prod_{i=0}^{\vert \tau \vert} U(\beta\{X/x_i\})^{\vert \tau \vert} & \text{when } \alpha = \forall X \subseteq \tau, \beta, \text{ and } x_i \text{ is any subset of } \tau \text{ such that } \vert x_i \vert = i. \\
\sum_{i=0}^{\vert \tau \vert} \binom{\vert \tau \vert}{i} \cdot U(\beta\{X/x_i\}) & \text{when } \alpha = \exists X \subseteq \tau, \beta, \text{ and } x_i \text{ is any subset of } \tau \text{ such that } \vert x_i \vert = i. 
\end{cases} \]

Complexity polynomial in domain size!
Polynomial in NNF size for bounded depth.
FOMC Query: Example

FO-Model Counting: $w(R) = w(\neg R) = 1$

FO ad-DNNF sentences
FO-Model Counting: \( w(R) = w(\neg R) = 1 \)

FO ad-DNNF sentences

4. \( \Delta = (\text{Stress}(\text{Alice}) \Rightarrow \text{Smokes}(\text{Alice})) \)

Domain = \{Alice\}
FO-Model Counting: $w(R) = w(\neg R) = 1$
FO ad-DNNF sentences

4. $\Delta = (\text{Stress}(Alice) \Rightarrow \text{Smokes}(Alice))$
   Domain = \{Alice\}

$\rightarrow$ 3 models
FOMC Query: Example

FO-Model Counting: \( w(R) = w(\neg R) = 1 \)

FO ad-DNNF sentences

4. \( \Delta = (\text{Stress(Alice)} \Rightarrow \text{Smokes(Alice)}) \)  
   Domain = \{Alice\}  
   \( \rightarrow \) 3 models

3. \( \Delta = \forall x, (\text{Stress}(x) \Rightarrow \text{Smokes}(x)) \)  
   Domain = \{n people\}
FOMC Query: Example

FO-Model Counting: $w(R) = w(\neg R) = 1$

FO ad-DNNF sentences

3. $\Delta = \forall x, (\text{Stress}(x) \Rightarrow \text{Smokes}(x))$
   
   Domain = \{n people\}

   $\rightarrow 3^n$ models

4. $\Delta = (\text{Stress}(\text{Alice}) \Rightarrow \text{Smokes}(\text{Alice}))$
   
   Domain = \{Alice\}

   $\rightarrow 3$ models
FOMC Query: Example

3. \[ \Delta = \forall x, (\text{Stress}(x) \Rightarrow \text{Smokes}(x)) \]

\[ \rightarrow 3^n \text{ models} \]

Domain = \{n people\}
FOMC Query: Example

3. \( \Delta = \forall x, (\text{Stress}(x) \Rightarrow \text{Smokes}(x)) \)  
   \[ \Rightarrow 3^n \text{ models} \]
   Domain = \{n people\}

2. \( \Delta = \forall y, (\text{ParentOf}(y) \land \text{Female} \Rightarrow \text{MotherOf}(y)) \)  
   D = \{n people\}
FOMC Query: Example

3. \( \Delta = \forall x, (\text{Stress}(x) \Rightarrow \text{Smokes}(x)) \)  
   \[ \text{Domain} = \{n \text{ people}\} \]
   \[ \rightarrow 3^n \text{ models} \]

2. \( \Delta = \forall y, (\text{ParentOf}(y) \land \text{Female} \Rightarrow \text{MotherOf}(y)) \)  
   \[ \text{D} = \{n \text{ people}\} \]
   If Female = true? \[ \Delta = \forall y, (\text{ParentOf}(y) \Rightarrow \text{MotherOf}(y)) \]
   \[ \rightarrow 3^n \text{ models} \]
### FOMC Query: Example

#### 3. \( \Delta = \forall x, (\text{Stress}(x) \Rightarrow \text{Smokes}(x)) \)

\[ \rightarrow 3^n \text{ models} \]

#### 2. \( \Delta = \forall y, (\text{ParentOf}(y) \land \text{Female} \Rightarrow \text{MotherOf}(y)) \)

- If Female = true? \( \Delta = \forall y, (\text{ParentOf}(y) \Rightarrow \text{MotherOf}(y)) \)
  \[ \rightarrow 3^n \text{ models} \]
- If Female = false? \( \Delta = \text{true} \)
  \[ \rightarrow 4^n \text{ models} \]
## FOMC Query: Example

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>
| 2. | \( \Delta = \forall y, (\text{ParentOf}(y) \land \text{Female} \Rightarrow \text{MotherOf}(y)) \)  
|   | \( D = \{n \text{ people}\} \)  

If Female = true?

\( \Delta = \forall y, (\text{ParentOf}(y) \Rightarrow \text{MotherOf}(y)) \)  
\( \rightarrow 3^n \text{ models} \)

If Female = false?

\( \Delta = \text{true} \)  
\( \rightarrow 4^n \text{ models} \)

\( \rightarrow 3^n + 4^n \text{ models} \)

| 3. | \( \Delta = \forall x, (\text{Stress}(x) \Rightarrow \text{Smokes}(x)) \)  
|   | Domain = \( \{n \text{ people}\} \)  

\( \rightarrow 3^n \text{ models} \)
FOMC Query: Example

1. \[ \Delta = \forall x, \forall y, (\text{ParentOf}(x,y) \land \text{Female}(x) \Rightarrow \text{MotherOf}(x,y)) \]
   \text{Domain} = \{n \text{ people}\}
   \Rightarrow 3^n \text{ models}

2. \[ \Delta = \forall y, (\text{ParentOf}(y) \land \text{Female} \Rightarrow \text{MotherOf}(y)) \]
   \text{Domain} = \{n \text{ people}\}
   \Rightarrow 3^n \text{ models}

   If Female = true?
   \[ \Delta = \forall y, (\text{ParentOf}(y) \Rightarrow \text{MotherOf}(y)) \]
   \Rightarrow 3^n \text{ models}

   If Female = false?
   \[ \Delta = \text{true} \]
   \Rightarrow 4^n \text{ models}

   \Rightarrow 3^n + 4^n \text{ models}

3. \[ \Delta = \forall x, (\text{Stress}(x) \Rightarrow \text{Smokes}(x)) \]
   \text{Domain} = \{n \text{ people}\}
   \Rightarrow 3^n \text{ models}
### FOMC Query: Example

<table>
<thead>
<tr>
<th>1.</th>
<th>( \Delta = \forall x, \forall y, (\text{ParentOf}(x, y) \land \text{Female}(x) \implies \text{MotherOf}(x, y)) )</th>
<th>( D = {n \text{ people}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \rightarrow (3^n + 4^n)^n \text{ models} )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2.</th>
<th>( \Delta = \forall y, (\text{ParentOf}(y) \land \text{Female} \implies \text{MotherOf}(y)) )</th>
<th>( D = {n \text{ people}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>If Female = true?</td>
<td>( \Delta = \forall y, (\text{ParentOf}(y) \implies \text{MotherOf}(y)) )</td>
<td>( \rightarrow 3^n \text{ models} )</td>
</tr>
<tr>
<td>If Female = false?</td>
<td>( \Delta = \text{true} )</td>
<td>( \rightarrow 4^n \text{ models} )</td>
</tr>
<tr>
<td></td>
<td>( \rightarrow 3^n + 4^n \text{ models} )</td>
<td></td>
</tr>
</tbody>
</table>

| 3. | \( \Delta = \forall x, (\text{Stress}(x) \implies \text{Smokes}(x)) \) | \( \text{Domain} = \{n \text{ people}\} \) |
|  | \( \rightarrow 3^n \text{ models} \) |  |
Group Quantifiers: Example

\[ \Delta = \forall x, y \in D, (\text{Smokes}(x) \land \text{Friends}(x, y) \Rightarrow \text{Smokes}(y)) \]

Domain = \{n people\}

- Not decomposable!
- Rewrite as FO ad-DNNF:

\[ \exists G \subseteq D, \text{Smokes}(G) \land \overline{\text{Smokes}}(\overline{G}) \land \overline{\text{Friends}}(G, \overline{G}) \]

- Not possible to ground to d-DNNF!
- How to do tractable CT?

\[ \sum_{i=0}^{|\tau|} \binom{|\tau|}{i} \cdot U(\beta\{X/x_i\}) \text{ when } \alpha = \exists X \subseteq \tau, \beta, \text{ and } x_i \text{ is any subset of } \tau \text{ such that } |x_i| = i \]
Group Quantifiers: Example

\[ \exists G \subseteq D, \text{Smokes}(G) \land \overline{\text{Smokes}}(\overline{G}) \land \text{Friends}(G, \overline{G}) \]
Group Quantifiers: Example

\[ \exists G \subseteq D, \text{Smokes}(G) \land \neg \text{Smokes}(\neg G) \land \text{Friends}(G, \neg G) \]

- If we know G precisely: who smokes, and there are k smokers?

**Database:**

<table>
<thead>
<tr>
<th></th>
<th>Smokes</th>
<th>Friends</th>
<th>Smokes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smokes(Alice)</td>
<td>1</td>
<td></td>
<td>k</td>
</tr>
<tr>
<td>Smokes(Bob)</td>
<td>0</td>
<td></td>
<td>k</td>
</tr>
<tr>
<td>Smokes(Charlie)</td>
<td>0</td>
<td></td>
<td>n-k</td>
</tr>
<tr>
<td>Smokes(Dave)</td>
<td>1</td>
<td></td>
<td>n-k</td>
</tr>
<tr>
<td>Smokes(Eve)</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Group Quantifiers: Example

\[ \exists G \subseteq D, \text{Smokes}(G) \land \text{Smokes}(\overline{G}) \land \text{Friends}(G, \overline{G}) \]

- If we know G precisely: who smokes, and there are k smokers?

**Database:**
- Smokes(Alice) = 1
- Smokes(Bob) = 0
- Smokes(Charlie) = 0
- Smokes(Dave) = 1
- Smokes(Eve) = 0
- ...

![Diagram showing the relationship between Smokes, Friends, and Smokes with k and n-k values.](image-url)
Group Quantifiers: Example

\[ \exists G \subseteq D, \text{Smokes}(G) \land \text{Smokes}(\bar{G}) \land \text{Friends}(G, \bar{G}) \]

- If we know \( G \) precisely: who smokes, and there are \( k \) smokers?

**Database:**
- Smokes(Alice) = 1
- Smokes(Bob) = 0
- Smokes(Charlie) = 0
- Smokes(Dave) = 1
- Smokes(Eve) = 0
  ...

[Diagram showing relationships between Smokes, Friends, and Smokes]
Group Quantifiers: Example

\[ \exists G \subseteq D, \text{Smokes}(G) \land \neg \text{Smokes}(G) \land \text{Friends}(G, \bar{G}) \]

- If we know G precisely: who smokes, and there are k smokers?

**Database:**
- Smokes(Alice) = 1
- Smokes(Bob) = 0
- Smokes(Charlie) = 0
- Smokes(Dave) = 1
- Smokes(Eve) = 0
- ...

![Diagram showing the relationship between smokers and non-smokers connected through friendship]
Group Quantifiers: Example

\[ \exists G \subseteq D, \text{Smokes}(G) \land \lnot \text{Smokes}(\bar{G}) \land \text{Friends}(G, \bar{G}) \]

- If we know G precisely: who smokes, and there are \( k \) smokers?

**Database:**
- Smokes(Alice) = 1
- Smokes(Bob) = 0
- Smokes(Charlie) = 0
- Smokes(Dave) = 1
- Smokes(Eve) = 0
- ...

![Diagram showing the relationship between Smokes, Friends, and Smokes with variables k and n-k]
Group Quantifiers: Example

\[ \exists G \subseteq D, \text{Smokes}(G) \land \neg \text{Smokes}(\neg G) \land \text{Friends}(G, \neg G) \]

- If we know \( G \) precisely: who smokes, and there are \( k \) smokers?

**Database:**
- Smokes(Alice) = 1
- Smokes(Bob) = 0
- Smokes(Charlie) = 0
- Smokes(Dave) = 1
- Smokes(Eve) = 0
- ...
Group Quantifiers: Example

$$\exists G \subseteq D, \text{Smokes}(G) \land \neg \text{Smokes}(\overline{G}) \land \text{Friends}(G, \overline{G})$$

- If we know $G$ precisely: who smokes, and there are $k$ smokers?

**Database:**
- Smokes(Alice) = 1
- Smokes(Bob) = 0
- Smokes(Charlie) = 0
- Smokes(Dave) = 1
- Smokes(Eve) = 0
- ...

![Diagram showing relationships between smokers and non-smokers]
Group Quantifiers: Example

\[ \exists G \subseteq D, \text{Smokes}(G) \land \overline{\text{Smokes}}(\bar{G}) \land \text{Friends}(G, \bar{G}) \]

- If we know \( G \) precisely: who smokes, and there are \( k \) smokers?

**Database:**
- Smokes(Alice) = 1
- Smokes(Bob) = 0
- Smokes(Charlie) = 0
- Smokes(Dave) = 1
- Smokes(Eve) = 0
- ...

![Diagram showing relationships between groups and friends](image-url)
Group Quantifiers: Example

\[ \exists G \subseteq D, \text{Smokes}(G) \land \neg\text{Smokes}(\bar{G}) \land \text{Friends}(G, \bar{G}) \]

- If we know $G$ precisely: who smokes, and there are $k$ smokers?

**Database:**

- Smokes(Alice) = 1
- Smokes(Bob) = 0
- Smokes(Charlie) = 0
- Smokes(Dave) = 1
- Smokes(Eve) = 0

...
Group Quantifiers: Example

\[ \exists G \subseteq D, \text{Smokes}(G) \land \text{Smokes}(\bar{G}) \land \text{Friends}(G, \bar{G}) \]

- If we know G precisely: who smokes, and there are k smokers?

**Database:**
- Smokes(Alice) = 1
- Smokes(Bob) = 0
- Smokes(Charlie) = 0
- Smokes(Dave) = 1
- Smokes(Eve) = 0
- ...

\[ \rightarrow 2^{n^2 - k(n-k)} \] models
Group Quantifiers: Example

\[ \exists G \subseteq D, \text{Smokes}(G) \land \neg\text{Smokes}(\bar{G}) \land \text{Friends}(G, \bar{G}) \]

- If we know \( G \) precisely: who smokes, and there are \( k \) smokers?

**Database:**
- Smokes(Alice) = 1
- Smokes(Bob) = 0
- Smokes(Charlie) = 0
- Smokes(Dave) = 1
- Smokes(Eve) = 0
- ...

\[ \Rightarrow 2^{\binom{n}{2} - k(n-k)} \] models

- If we know that there are \( k \) smokers?
Group Quantifiers: Example

\[ \exists G \subseteq D, \text{Smokes}(G) \land \neg \text{Smokes}(\bar{G}) \land \text{Friends}(G, \bar{G}) \]

- If we know G precisely: who smokes, and there are \(k\) smokers?
  - Database:
    - Smokes(Alice) = 1
    - Smokes(Bob) = 0
    - Smokes(Charlie) = 0
    - Smokes(Dave) = 1
    - Smokes(Eve) = 0
    ...
  - \(2^{n^2 - k(n-k)}\) models

- If we know that there are \(k\) smokers?
  \(\binom{n}{k}2^{n^2 - k(n-k)}\) models
Group Quantifiers: Example

\[ \exists G \subseteq D, \text{Smokes}(G) \land \tilde{\text{Smokes}}(\tilde{G}) \land \text{Friends}(G, \tilde{G}) \]

- If we know \( G \) precisely: who smokes, and there are \( k \) smokers?

  **Database:**
  - Smokes(Alice) = 1
  - Smokes(Bob) = 0
  - Smokes(Charlie) = 0
  - Smokes(Dave) = 1
  - Smokes(Eve) = 0
  ...

  \[ 2n^2 - k(n-k) \] models

- If we know that there are \( k \) smokers?

  \[ \binom{n}{k} 2n^2 - k(n-k) \] models

- In total...

\[ 2n^2 - k(n-k) \] models
Group Quantifiers: Example

\[ \exists G \subseteq D, \text{Smokes}(G) \land \neg \text{Smokes}(\overline{G}) \land \text{Friends}(G, \overline{G}) \]

- If we know \( G \) precisely: who smokes, and there are \( k \) smokers?

**Database:**
- Smokes(Alice) = 1
- Smokes(Bob) = 0
- Smokes(Charlie) = 0
- Smokes(Dave) = 1
- Smokes(Eve) = 0
...

\[ \rightarrow 2^{n^2 - k(n-k)} \text{ models} \]

- If we know that there are \( k \) smokers?

\[ \rightarrow \binom{n}{k} 2^{n^2 - k(n-k)} \text{ models} \]

- In total...

\[ \rightarrow \sum_{k=0}^{n} \binom{n}{k} 2^{n^2 - k(n-k)} \text{ models} \]
Playing Cards Revisited

Let us automate this:

\[
\forall p, \exists c, \text{Card}(p,c) \\
\forall c, \exists p, \text{Card}(p,c) \\
\forall p, \forall c, \forall c', \text{Card}(p,c) \land \text{Card}(p,c') \Rightarrow c = c'
\]

Playing Cards Revisited

Let us automate this:

\[ \forall p, \exists c, \text{Card}(p,c) \]
\[ \forall c, \exists p, \text{Card}(p,c) \]
\[ \forall p, \forall c, \forall c', \text{Card}(p,c) \land \text{Card}(p,c') \Rightarrow c = c' \]

\[ \text{#SAT} = \sum_{k=0}^{n} \binom{n}{k} \sum_{l=0}^{n} \binom{n}{l} (l + 1)^k (-1)^{2n-k-l} = n! \]

Playing Cards Revisited

Let us automate this:

\[
\forall p, \exists c, \text{Card}(p,c) \\
\forall c, \exists p, \text{Card}(p,c) \\
\forall p, \forall c, \forall c', \text{Card}(p,c) \land \text{Card}(p,c') \Rightarrow c = c'
\]

\[\#\text{SAT} = \sum_{k=0}^{n} \binom{n}{k} \sum_{l=0}^{n} \binom{n}{l} (l + 1)^k (-1)^{2n-k-l} = n!\]

Computed in time polynomial in n

FO COMPILATION
Compilation Rules

• Lots of preprocessing
• Shannon decomposition/Boole’s expansion
• Detect propositional decomposability
• FO Shannon decomposition:
  \[ \exists X \subseteq \tau, P(X) \land \overline{P(X)} \land \beta \]
  Simplify \( \beta \) (remove atoms subsumed by \( P(X) \))
  Always deterministic! Ensure automorphic \( \exists \)
• Detect FO decomposability
FO NNF EXPRESSIVENESS
Main Positive Result: FO^2

- FO^2 = FO restricted to two variables
- “The graph has a path of length 10”:
  \[ \exists x \exists y (R(x,y) \land \exists x (R(y,x) \land \exists y (R(x,y) \land \ldots ))) \]

- Theorem: Compilation algorithm to FO ad-DNNF is complete for FO^2
- Model counting for FO^2 in PTIME domain complexity
Main Negative Results

Domain complexity:

• There exists an FO formula $Q$ s.t. symmetric $\text{FOMC}(Q, n)$ is #P$_1$ hard
• There exists $Q$ in FO$^3$ s.t. $\text{FOMC}(Q, n)$ is #P$_1$ hard
• There exists a conjunctive query $Q$ s.t. symmetric $\text{WFOMC}(Q, n)$ is #P$_1$ hard
• There exists a positive clause $Q$ w.o. ‘=‘ s.t. symmetric $\text{WFOMC}(Q, n)$ is #P$_1$ hard

Therefore, no FO ad-DNNF can exist 😞
Proof

**Theorem.** There exists an FO$^3$ sentence $Q$ s.t. $\text{FOMC}(Q,n)$ is #$P_1$-hard

**Proof**

- **Step 1.** Construct a Turing Machine $U$ s.t.
  - $U$ is in #$P_1$ and runs in linear time in $n$
  - $U$ computes a #$P_1$-hard function
- **Step 2.** Construct an FO$^3$ sentence $Q$ s.t. $\text{FOMC}(Q,n) / n! = U(n)$
Fertile Ground

Monadic

$\Theta_1$

$\Theta_1$

$\Theta_1$

$\Theta_1$

$\Theta_1$

$\Theta_1$
\[ \Delta = \forall x, y, z, \text{Friends}(x, y) \land \text{Friends}(y, z) \Rightarrow \text{Friends}(x, z) \]

[VdB; NIPS’11], [VdB et al.; KR’14], [Gribkoff, VdB, Suciu; UAI’15], [Beame, VdB, Gribkoff, Suciu; PODS’15], etc.
Other Queries and Transformations

• What if all ground atoms have different weights? Asymmetric WFOMC
• FO d-DNNF complete for all monotone FO CNFs that support efficient CT
• No clausal entailment
• No conditioning
Conclusions

• Very powerful already!
• We need to solve this!

THANKS
References

• **Cards Example:**

• **First-Order Knowledge Compilation:**

• **Expressiveness:**