Dynamic Probabilistic Relational Models

Lifted Exact Inference: First-Order Knowledge Compilation

Marcel Gehrke
1. Introduction
   - StaRAI: Agent, context, motivation

2. Foundations
   - Logic
   - Probability theory
   - Probabilistic graphical models (PGMs)

3. Probabilistic Relational Models (PRMs)
   - Parfactor models, Markov logic networks
   - Semantics, inference tasks

4. Exact Lifted Inference
   - Lifted Variable Elimination
   - Lifted Junction Tree Algorithm
   - First-Order Knowledge Compilation

5. Lifted Sequential Models and Inference
   - Parameterised models
   - Semantics, inference tasks, algorithm

6. Lifted Decision Making
   - Preferences, utility
   - Decision-theoretic models, tasks, algorithm

7. Approximate Lifted Inference

8. Lifted Learning
   - Parameter learning
   - Relation learning
   - Approximating symmetries
Outline: 4. Lifted Inference

**Exact Inference**

i. Lifted Variable Elimination for Parfactor Models
   • Idea, operators, algorithm, complexity

ii. Lifted Junction Tree Algorithm
    • Idea, helper structure: junction tree, algorithm

iii. First-order Knowledge Compilation for MLNs
    • Idea, helper structure: circuit, algorithm
MLNs: Semantics

- MLN $\Psi = \{(w_i, \psi_i)\}_{i=1}^n$, with $w_i \in \mathbb{R}$, induces a probability distribution over possible interpretations $\omega$ (world) of the grounded atoms in $\Psi$
  
  - $N$ = the number of ground atoms in the grounded $\Psi$
  
  - Probability of one interpretation $\omega$

$$P(\omega) = \frac{1}{Z} \exp \left( \sum_{i=1}^{n} w_i n_i(\omega) \right)$$

- $n_i(\omega) = $ number of propositional sentences of $\psi_i$ that evaluate to true given the assignments of $\omega$

10 $\text{Presents}(X, P, C) \Rightarrow \text{Attends}(X, C)$

3.75 $\text{Publishes}(X, C) \land \text{FarAway}(C) \Rightarrow \text{Attends}(X, C)$
Local Symmetries and Structure

- Consider potential function as given by the table on the right
  \( \phi(T_{\text{Travel}}(X), E_{\text{Epid}}, S_{\text{Sick}}(X)) \)

- Only two weighted formulas \((w, \psi)\) necessary
  - \((\ln 2, \neg T_{\text{Travel}}(X) \lor \neg E_{\text{Epid}} \lor \neg S_{\text{Sick}}(X))\)
  - \((\ln 7, T_{\text{Travel}}(X) \land E_{\text{Epid}} \land S_{\text{Sick}}(X))\)
  - If potential of 1 instead of 2, would reduce to
    - \((\ln 7, T_{\text{Travel}}(X) \land E_{\text{Epid}} \land S_{\text{Sick}}(X))\)
    - Assignments that do not make the formula true automatically get weight of \(0 = \ln 1\)
  - If external knowledge existing, provide FOL formulas directly
    - E.g., \((\ln 2, E_{\text{Epid}} \land S_{\text{Sick}}(X) \Rightarrow \neg T_{\text{Travel}}(X))\)

<table>
<thead>
<tr>
<th>Travel(X)</th>
<th>Epid</th>
<th>Sick(X)</th>
<th>(\phi)</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
<td>2</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>true</td>
<td>2</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>false</td>
<td>2</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
<td>2</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>false</td>
<td>2</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>true</td>
<td>2</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>false</td>
<td>2</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>true</td>
<td>7</td>
</tr>
</tbody>
</table>

Use for efficient inference
Weighted Model Counting

• Solve query answering problem by solving a weighted model counting problem
  – Weighted model count (WMC) given a sentence \( \varphi \) in propositional logic and a weight function \( \text{weight} : L \rightarrow \mathbb{R}_{\geq 0} \) associating a non-negative weight to each literal in \( \varphi \) (set \( L \)) defined by
    \[
    WMC(\varphi, \text{weight}) = \sum_{\omega \in \Omega_\varphi} \prod_{l \in \omega} \text{weight}(l)
    \]
  – where \( \Omega_\varphi \) refers to the set of worlds of \( \varphi \)
  – Probability of a world \( \omega \) of a sentence \( \varphi \) with weight function
    \[
    P(\omega) = \frac{\prod_{l \in \omega} \text{weight}(l)}{WMC(\varphi, \text{weight})} = \frac{WMC(\varphi \land \omega, \text{weight})}{WMC(\varphi, \text{weight})}
    \]
  – A query for literal \( q \) given evidence \( e \) is solved by computing
    \[
    P(q|e) = \frac{WMC(\varphi \land q \land e, \text{weight})}{WMC(\varphi \land e, \text{weight})}
    \]
  – Vgl. \( P(Q|E) = \frac{P(Q,E)}{P(E)} \)
Weighted Model Counting: Example

- Sentence
  - \( \text{sun} \land \text{rain} \Rightarrow \text{rainbow} \)
- Weight function:
  - \( \text{weight}(\text{sun}) = 1 \)
  - \( \text{weight}(\neg \text{sun}) = 5 \)
  - \( \text{weight}(\text{rain}) = 2 \)
  - \( \text{weight}(\neg \text{rain}) = 7 \)
  - \( \text{weight}(\text{rainbow}) = 0 \)
  - \( \text{weight}(\neg \text{rainbow}) = 10 \)

\[
WMC(\varphi, \text{weight}) = \sum_{\omega \in \Omega_{\varphi}} \prod_{l \in \omega} \text{weight}(l)
\]

<table>
<thead>
<tr>
<th>rain</th>
<th>sun</th>
<th>rainbow</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7 \cdot 5 \cdot 10 = 350</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>7 \cdot 5 \cdot 0.1 = 3.5</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>7 \cdot 1 \cdot 10 = 70</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>7 \cdot 1 \cdot 0.1 = 0.7</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2 \cdot 5 \cdot 10 = 100</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2 \cdot 5 \cdot 0.1 = 1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2 \cdot 1 \cdot 10 = 20</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2 \cdot 1 \cdot 0.1 = 0.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+ 525.4</td>
</tr>
</tbody>
</table>
Weighted Model Counting: Example

• Sentence
  - $\text{sun} \land \text{rain} \Rightarrow \text{rainbow}$

• Weight function:
  - $\text{weight}(\text{sun}) = 1$
  - $\text{weight}(\neg \text{sun}) = 5$
  - $\text{weight}(\text{rain}) = 2$
  - $\text{weight}(\neg \text{rain}) = 7$
  - $\text{weight}(\text{rainbow}) = 0.1$
  - $\text{weight}(\neg \text{rainbow}) = 10$

• Probability of worlds:
  
  $$P(\omega) = \frac{\prod_{l \in \omega} \text{weight}(l)}{\text{WMC}(\varphi, \text{weight})} = \frac{\text{WMC}(\varphi \land \omega, \text{weight})}{\text{WMC}(\varphi, \text{weight})}$$

  $\omega = (\text{sun}, \text{rain}, \text{rainbow}) \in \Omega_{\varphi}$

<table>
<thead>
<tr>
<th>rain</th>
<th>sun</th>
<th>rainbow</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7 \cdot 5 \cdot 10 = 350</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>7 \cdot 5 \cdot 0.1 = 3.5</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>7 \cdot 1 \cdot 10 = 70</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>7 \cdot 1 \cdot 0.1 = 0.7</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2 \cdot 5 \cdot 10 = 100</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2 \cdot 5 \cdot 0.1 = 1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2 \cdot 1 \cdot 10 = 20.0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2 \cdot 1 \cdot 0.1 = 0.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+ 525.4</td>
</tr>
</tbody>
</table>

  $$P(\text{sun}, \text{rain}, \text{rainbow}) = \frac{0.2}{525.4} = 0.00038$$
Weighted Model Counting: Example

- **Sentence**
  - \( \text{sun} \land \text{rain} \Rightarrow \text{rainbow} \)

- **Weight function:**
  - \( \text{weight}(\text{sun}) = 1 \)
  - \( \text{weight}(\neg \text{sun}) = 5 \)
  - \( \text{weight}(\text{rain}) = 2 \)
  - \( \text{weight}(\neg \text{rain}) = 7 \)
  - \( \text{weight}(\text{rainbow}) = 0.1 \)
  - \( \text{weight}(\neg \text{rainbow}) = \)

- **Probability of worlds:**
  - \( P(\text{rain}) = \frac{100 + 1 + 0.2}{525.4} = 0.1926 \)

\[
P(q) = \frac{\text{WMC}(\varphi \land q, \text{weight})}{\text{WMC}(\varphi, \text{weight})}
\]

\[
\text{(sun} \land \text{rain} \Rightarrow \text{rainbow}) \land \text{rain}
\]

<table>
<thead>
<tr>
<th>rain</th>
<th>sun</th>
<th>rainbow</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7 \cdot 5 \cdot 10 = 350</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>7 \cdot 5 \cdot 0.1 = 3.5</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>7 \cdot 1 \cdot 10 = 70</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>7 \cdot 1 \cdot 0.1 = 0.7</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2 \cdot 5 \cdot 10 = 100</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2 \cdot 5 \cdot 0.1 = 1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2 \cdot 1 \cdot 10 = 20.0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2 \cdot 1 \cdot 0.1 = 0.2</td>
</tr>
</tbody>
</table>

\[+ 525.4\]

---

All \( \omega \in \Omega_\varphi \) where \( \text{rain} \) holds
WMC and Inference

- Solving a WMC problem for a sentence $\varphi$ as introduced on previous slides is exponential in number of worlds with probability $> 0$ (models).
- To be more efficient, build a helper structure
  - Bring sentence into negation normal form (NNF)
    - NNF: Formulas contain only negations directly in front of variables, conjunctions, and disjunctions
    - E.g.,
      - $\text{sun} \land \text{rain} \Rightarrow \text{rainbow}$ (Apply $A \Rightarrow B \equiv \neg A \lor B$)
      $\equiv \neg (\text{sun} \land \text{rain}) \lor \text{rainbow}$ (Apply De Morgan’s law)
      $\equiv \neg \text{sun} \lor \neg \text{rain} \lor \text{rainbow}$ (NNF)
Circuits

• Represent the NNF sentence as a directed, acyclic graph called circuit with leaves labelled with literals \((l\) or \(\neg l\)) or \(true, false\) with inner nodes being
  – Deterministic disjunctions
    • Only one disjunct (child node) can be true at the same time
      – i.e., their conjunction is unsatisfiable
  – Decomposable conjunctions
    • Each pair of conjuncts (child nodes) must be independent
      – i.e., they cannot share any variables

• Circuit is then in d-DNNF
  – deterministic Decomposable NNF
  – See later why important

Circuits: Example

- **Deterministic** disjunctions
  - Only one disjunct (child node) can be true at the same time
  - I.e., their conjunction is unsatisfiable

- **Decomposable** conjunctions
  - Each pair of conjuncts (child nodes) must be independent
  - I.e., they cannot share any variables
  - E.g., \( \neg \text{sun} \lor \neg \text{rain} \lor \text{rainbow} \)
  - \(<\text{disjunct}> \lor \text{rainbow} \)
    - Determinism: \(<\text{disjunct}> \) can only be true if \( \text{rainbow} \) is not
      - Add \( \neg \text{rainbow} \) to disjunct: \( \neg \text{rainbow} \land <\text{disjunct}> \)
Circuits: Example

- **Deterministic** disjunctions
  - Only one disjunct (child node) can be true at the same time
    - I.e., their conjunction is unsatisfiable
- **Decomposable** conjunctions
  - Each pair of conjuncts (child nodes) must be independent
    - I.e., they cannot share any variables
  - E.g., ¬sun ∨ ¬rain ∨ rainbow
    - <disjunct> ∨ rainbow
      - Determinism: <disjunct> can only be true if rainbow is not
        - Add ¬rainbow to disjunct: ¬rainbow ∧ <disjunct>
        - <disjunct> now part of a conjunction with ¬rainbow
          » Decomposability: May not contain Rainbow
Circuits: Example

- **Deterministic** disjunctions
  - Only one disjunct (child node) can be true at the same time
    - I.e., their conjunction is unsatisfiable
- **Decomposable** conjunctions
  - Each pair of conjuncts (child nodes) must be independent
    - I.e., they cannot share any variables
  - E.g., \( \neg\text{sun} \lor \neg\text{rain} \lor \text{rainbow} \)
    - \(<\text{disjunct}> \lor \neg\text{rain} \)
      - Determinism: \(<\text{disjunct}>\) can only be true if \(\neg\text{rain}\) is not, i.e., if \(\text{rain}\) is
        - Add \(\text{rain}\) to disjunct: \(\text{rain} \land <\text{disjunct}>\)
Circuits: Example

- Deterministic disjunctions
  - Only one disjunct (child node) can be true at the same time
    - I.e., their conjunction is unsatisfiable
- Decomposable conjunctions
  - Each pair of conjuncts (child nodes) must be independent
    - I.e., they cannot share any variables
  - E.g., \( \neg sun \lor \neg rain \lor \text{rainbow} \)
    - \(<\text{disjunct}> \lor \neg rain\)
      - Determinism: \(<\text{disjunct}>\) can only be true if \(\neg rain\) is not, i.e., if \(rain\) is
        - Add \(rain\) to disjunct: \(rain \land <\text{disjunct}>\)
        - \(<\text{disjunct}>\) now part of a conjunction with \(rain\)
          » Decomposability: May not contain \(Rain\)
Circuits: Example

- Deterministic disjunctions
  - Only one disjunct (child node) can be true at the same time
    - I.e., their conjunction is unsatisfiable
- Decomposable conjunctions
  - Each pair of conjuncts (child nodes) must be independent
    - I.e., they cannot share any variables
  - E.g., $\neg\text{sun} \lor \neg\text{rain} \lor \text{rainbow}$
    - Add $\neg\text{sun}$ as conjunct
      - Decomposability: Does not share variables with sibling node
Effects of d-DNNF

- Effects of d-DNNF
  - Deterministic disjunctions
    - Only one disjunct (child node) can be true at the same time
      - I.e., their conjunction is unsatisfiable
    - Assume children \( c_i, c_j \) represent probabilities \( p_i, p_j \)
      - Node then represents probability of \( P(c_i \lor c_j) \)
        - \( P(c_i \lor c_j) = P(c_i) + P(c_j) - P(c_i \land c_j) \)
      - If only \( c_i \) or \( c_j \) can be true at a time, \( P(c_i \land c_j) = 0 \), i.e.,
        - \( P(c_i \lor c_j) = P(c_i) + P(c_j) \)
    - Can replace \( \lor \) with \( + \) for inference calculations
Effects of d-DNNF

- Effects of d-DNNF
  - Decomposable conjunctions
    - Each pair of conjuncts (child nodes) must be independent
      - i.e., they cannot share any variables
    - Assume children $c_i, c_j$ represent probabilities $p_i, p_j$
      - Node then represents probability of $P(c_i \land c_j)$
      - If $c_i$ and $c_j$ independent (decomposable), then $P(c_i \land c_j) = P(c_i) \cdot P(c_j)$
    - Can replace $\land$ with $\cdot$ for inference calculations

Diagram:

```
  V
 /\   /
|  |  |
V  -rainbow-\V
 /  /
|  |
V  -rain
 /  /
|  |
V  -sun  rain
```

Rainbow 🌈

Rain 🌧️

Sun ☀️

 ¬Rainbow

 ¬Rain

 ¬Sun

 ¬Rain

 ¬Rainbow
Smooth d-DNNF (sd-DNNF)

- Smooth circuits: constant runtime for certain queries
  - Any pair of disjuncts mentions the same set of variables
  - E.g., \( \neg \text{sun} \lor \neg \text{rain} \lor \text{rainbow} \)
    - Two disjunctions that do not fulfil the smoothness property
- Rules for conversion
  - For each negation of a positive literal \( l \) not appearing, replace \( l \) by
    \[ l \lor (\neg l \land \text{false}) \]
  - For each variable \( A \) not mentioned in a disjunct \( <\text{disjunct}> \), add \( a \lor \neg a \) with a conjunction to \( <\text{disjunct}> \):
    \[ <\text{disjunct}> \land (a \lor \neg a) \]
Smooth d-DNNF (sd-DNNF)

- Add $\text{sun} \lor \neg \text{sun}$ to $\neg \text{rain}$, replacing $\neg \text{rain}$ with $\neg \text{rain} \land (\text{sun} \lor \neg \text{sun})$
Smooth d-DNNF (sd-DNNF)

• Add \( \text{sun} \lor \neg \text{sun} \) and \( \text{rain} \lor \neg \text{rain} \), replacing \text{rainbow} with

\[
\text{rainbow} \land (\text{sun} \lor \neg \text{sun}) \land (\text{rain} \lor \neg \text{rain})
\]
Circuit for Model Counting

- Model counting problem: Count how many models fulfil a sentence
- Model counting arithmetic circuit
  - Replace $\wedge$ with $\cdot$
  - Replace $\lor$ with $+$
  - Replace leaves with 1’s

\[
\begin{align*}
L & \cdot L + 1 \\
1 & + 1 + 1 \quad \text{rainbow} \\
1 & \cdot L \quad \text{rain} \quad \text{sun} \\
1 & + 1 + 1 \quad \text{sun} \quad \neg \text{rain} \quad \neg \text{sun} \quad \text{rain}
\end{align*}
\]
Circuit for Model Counting

- Propagate 1’s upwards (from leaves to root), using arithmetic operations in inner nodes to combine incoming numbers
  - Result at root: Model count

\[ \text{sun} \land \text{rain} \Rightarrow \text{rainbow} \]

<table>
<thead>
<tr>
<th>rain</th>
<th>sun</th>
<th>rainbow</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ \text{sun} \land \text{rain} \Rightarrow \text{rainbow} \]
Conditioning

- To get model count of models fulfilling certain truth values
  - Replace 1’s with zeros where literal contradicts truth values
  - E.g., condition on \( \neg \text{rainbow} \)

\[
\text{sun} \land \text{rain} \Rightarrow \text{rainbow}
\]

<table>
<thead>
<tr>
<th>(\text{rain})</th>
<th>(\text{sun})</th>
<th>(\text{rainbow})</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Circuit for Weighted Model Counting

- Replace literals with weights in leaves and propagate weights upwards
- Computes $WMC(\varphi, \text{weight})$

$\text{weight}(\text{sun}) = 1$
$\text{weight}(\neg\text{sun}) = 5$
$\text{weight}(\text{rain}) = 2$
$\text{weight}(\neg\text{rain}) = 7$
$\text{weight}(\text{rainbow}) = 0.1$
$\text{weight}(\neg\text{rainbow}) = 10$
Circuit for Weighted Model Counting

- For probabilities of worlds or query terms \( \omega \), condition on truth values
  1. Compute \( WMC(\varphi, \text{weight}) \)
  2. Compute \( WMC(\varphi \land \omega, \text{weight}) \)
  3. Divide the two counts

\[
P(\omega = \{\text{sun, rain, rainbow}\}) = \frac{\text{WMC}(\varphi \land \omega, \text{weight})}{\text{WMC}(\varphi, \text{weight})} \]
\[
= \frac{0.2}{525.4} = 0.00038
\]

Reuse for different queries

\[
\begin{align*}
\text{sun} & \quad \text{rain} & \quad \text{rainbow} & \quad \neg\text{rainbow} \\
1 & \quad 0 & \quad 0 & \quad 2
\end{align*}
\]
Knowledge Compilation

- Given a theory $\Delta$ and a set of queries $\{P(q_i|e)\}_{i=1}^m$
  - Build a circuit for theory $\Delta$ (a conjunction of sentences)
  - Make the circuit a WMC circuit
    - Replace inner nodes with arithmetic operations and leaves with weights
  - Condition on given evidence $e$
    - Replace weights with 0 where literals contradict $e$
  - Calculate $WMC(\Delta \land e, weight)$ in the circuit
    - By propagating the weights upwards
  - For each query $P(q_i|e)$ in the circuit
    - Compute $WMC(\Delta \land e \land q_i, weight)$
    - Return or store $P(q_i|e) = \frac{WMC(\Delta \land e \land q_i, weight)}{WMC(\Delta \land e, weight)}$
Propositional → First-order

• If input theory is in FOL-DC ((function-free) first-order logic with domain constraints), one could ground the theory given domains and build a circuit for the grounded theory
  – FOL-DC includes intensional conjunctions and disjunctions (∀, ∃)
  – Leads to repeated structures in circuit
• Combine repeated structures using new inner node types for intensional conjunctions and disjunctions (∀, ∃)
• We are not going into every detail of FOKC;
  – For complete description, analysis, and discussion, see the PhD thesis by Guy Van den Broeck
Weighted First-order Model Counting

- Define a weighted first-order model counting problem using a weighted first-order model count (WFOMC)

\[ WFOMC(\Delta, w_T, w_F) = \sum_{\omega=\omega_T \cup \omega_F} \prod_{l \in \omega_T} w_T(\text{pred}(l)) \prod_{l \in \omega_F} w_F(\text{pred}(l)) \]

- \( \Delta \) a theory in FOL-DC
- \( w_T \) a weight function for predicates being positive
- \( w_F \) a weight function for predicates being negative
- \( \Omega_\Delta \) the set of worlds (i.e., models in logics) of \( \Delta \)
- \( \text{pred}(l) \) a function mapping a literal \( l \) to its predicate

- Query can be answered by computing

\[ P(q_i|e) = \frac{WFOMC(\Delta \land e \land q_i, w_T, w_F)}{WFOMC(\Delta \land e, w_T, w_F)} \]
Example

• Theory: one sentence
  \( \forall X \in \text{People} : \text{smokes}(X) \Rightarrow \text{cancer}(X) \)
  - People = \{x_1, x_2\}
  - Weight functions
    - \( w_T(\text{smokes}(X)) = 3 \)
    - \( w_F(\neg\text{smokes}(X)) = 1 \)
    - \( w_T(\text{cancer}(X)) = 6 \)
    - \( w_F(\neg\text{cancer}(X)) = 2 \)
  - Model count: 9

\[
WFOMC(\Delta, w_T, w_F) = \sum_{\omega=\omega_T \cup \omega_F} \prod_{l \in \omega_T} w_T(\text{pred}(l)) \prod_{l \in \omega_F} w_F(\text{pred}(l))
\]

<table>
<thead>
<tr>
<th>( s(x_1) )</th>
<th>( c(x_1) )</th>
<th>( s(x_2) )</th>
<th>( c(x_2) )</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1\cdot2\cdot1\cdot2 4</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1\cdot2\cdot1\cdot6 12</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1\cdot2\cdot3\cdot2 12</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1\cdot2\cdot3\cdot6 36</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1\cdot6\cdot1\cdot2 12</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1\cdot6\cdot1\cdot6 36</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1\cdot6\cdot3\cdot2 36</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1\cdot6\cdot3\cdot6 108</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3\cdot2\cdot1\cdot2 12</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3\cdot2\cdot1\cdot6 36</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3\cdot2\cdot3\cdot2 36</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3\cdot2\cdot3\cdot6 108</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3\cdot6\cdot1\cdot2 36</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>3\cdot6\cdot1\cdot6 108</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>3\cdot6\cdot3\cdot2 108</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3\cdot6\cdot3\cdot6 324</td>
</tr>
</tbody>
</table>

\[ + 676 \]
Example

- Theory: one sentence
  \[ \forall X \in \text{People} : \text{smokes}(X) \Rightarrow \text{cancer}(X) \]
- People = \{x_1, x_2\}
- Weight functions
  - \( w_T(\text{smokes}(X)) = 3 \)
  - \( w_F(\neg \text{smokes}(X)) = 1 \)
  - \( w_T(\text{cancer}(X)) = 6 \)
  - \( w_F(\neg \text{cancer}(X)) = 2 \)
- Model count: 9
  \[ P(s(x_1)) = \frac{WFOMC(\Delta \land s(x_1), w_T, w_F)}{WFOMC(\Delta, w_T, w_F)} = \frac{36 + 108 + 324}{676} = \frac{468}{676} \]

<table>
<thead>
<tr>
<th>( s(x_1) )</th>
<th>( c(x_1) )</th>
<th>( s(x_2) )</th>
<th>( c(x_2) )</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1·2·1·2·4·1 = 4</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1·2·1·6·1 = 12</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1·2·3·2·1 = 12</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1·2·3·6·1 = 36</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1·6·1·2·1 = 12</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1·6·1·6·1 = 36</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1·6·3·2·1 = 36</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1·6·3·6·1 = 108</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3·2·1·2·1 = 12</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3·2·1·6·1 = 36</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3·2·3·2·1 = 36</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3·2·3·6·1 = 108</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3·6·1·2·1 = 36</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>3·6·1·6·1 = 108</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>3·6·3·2·1 = 108</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3·6·3·6·1 = 324</td>
</tr>
</tbody>
</table>

\[ \sum_{\text{all } s, c} = 676 \]
First-order (FO) Circuits

• Assume theory in Skolem normal form + CNF
  – Sequence of intensional conjunctions in CNF
  – E.g., with \( s = \text{smokes}, c = \text{cancer} \)
    \[
    \forall X \in \text{People} : s(X) \Rightarrow c(X)
    \equiv \forall X \in \text{People} : \neg s(X) \lor c(X)
    \]

• FO circuit (excerpt)
  – Inner nodes:
    • Extensional conjunctions/disjunctions (as before)
    • Set conjunctions
  – Leaf nodes
    • Positive and negative predicates, *true*, *false*
  – Full + construction: see PhD thesis by Guy Van den Broeck

\[
\forall X \in \text{People} \quad \neg s(X) \lor c(X)
\]
Smooth FO d-DNNF Circuits

- Properties
  - Deterministic disjunctions
    - Only one disjunct (child node) can be true at the same time
  - Decomposable conjunctions
    - Each pair of conjuncts (child nodes) must be independent
  - Smoothness
    - Each disjunct contains the same variables

\[
\forall X
\begin{align*}
X \in \text{People} \\
c(X) \land s(X) \land \neg s(X) \land \neg c(X)
\end{align*}
\]

\[
\forall X
\begin{align*}
X \in \text{People} \\
c(X) \land \neg s(X) \land \neg c(X)
\end{align*}
\]
Arithmetic FO d-DNNF Circuits

- Replace
  - Replace $\land$ with $\cdot$
  - Replace $\lor$ with $+$
  - Replace $\forall$ with exponentiation for $|\text{Domain}|$
  - Replace leaves with 1’s
- E.g., with $|\text{People}| = |\{x_1, x_2\}| = 2$
WFOMC Circuits

- Replace
  - Replace $\land$ with $\cdot$
  - Replace $\lor$ with $+$
  - Replace $\forall$ with exponentiation for $\text{Domain}$
  - Replace leaves with weights
  - E.g., with $|\text{People}| = |\{x_1, x_2\}| = 2$

$$ WFOMC(\Delta, w_T, w_F) = \sum_{\omega \in \omega_T} \prod_{l \in \omega} w_T(\text{pred}(l)) \prod_{l \in \omega} w_F(\text{pred}(l)) $$

$w_T(\text{smokes}(X)) = 3$
$w_F(\neg\text{smokes}(X)) = 1$
$w_T(\text{cancer}(X)) = 6$
$w_F(\neg\text{cancer}(X)) = 2$
Given $P(q_i)$
- Basically, compile a circuit for $\Delta \land q_i$ reusing components from the circuit of $\Delta$
- E.g., $P(s(x_1))$ with $|\text{People}| = |\{x_1, x_2\}| = 2$

\[
P(s(x_1)) = \frac{\text{WFOMC}(\Delta \land s(x_1), w_T, w_F)}{\text{WFOMC}(\Delta, w_T, w_F)}
\]
\[
= \frac{468}{676} = 0.692
\]
Conditioning in FO Circuits

- **Evidence on propositional variables** $L$
  - Replace leaf values with 0 where literal contradicts observation as in propositional circuits
- **Evidence on unary variable** $L(X)$
  - For each variable $L(X)$ that one wants to condition on,
    - Replace FOL-DC formula with three copies with additional domain constraints, simplify based on observation
      1. $X \in D_T$ for observations $l(x)$
      2. $X \in D_\perp$ for observations $\neg l(x)$
      3. $X \notin D_T \land X \notin D_\perp$ no observations
  - Compile a circuit for the extended theory
  - Given specific evidence, domains for $D_T, D_\perp$ are determined
    - Might be empty
- **Evidence on binary variable** $L(X, Y)$
  - Can compile a circuit, no longer polynomial in time (reduction of #2SAT problem)
Conditioning in FO Circuits

- E.g., ∀X ∈ People : s(X) ⇒ c(X) and S(X)
  1. ∀X ∈ People_\top : s(X) ⇒ c(X) \equiv ∀X ∈ People_\top : c(X)
  2. ∀X ∈ People_\bot : s(X) ⇒ c(X) \equiv ∀X ∈ People_\bot : true
  3. ∀X ∈ People, X \notin People_\top, X \notin People_\bot : s(X) ⇒ c(X)
    - Delete Formula 2 as it is always true
    - If one also wants to condition on C(X), theory becomes larger again:
      - Formulas (1) and (3) contain C(X) and therefore need to be replaced by three formulas, then simplify
First-order Knowledge Compilation (FOKC)

- **Given**
  - Theory $\Delta$ in FOL-DC in Skolem NNF
  - Weight function $w_T$ for positive predicates, weight function $w_F$ for negative predicates
  - Set of queries $\{P(q_i|e)\}_{i=1}^{m}$

- Build a WFOMC circuit $C_\Delta$ for $\Delta$, also preparing for evidence on $rv(e)$
- Condition on $e$
- Calculate $WFOMC(\Delta \land e, w_T, w_F)$ in $C_\Delta$
- For each query $P(q_i|e)$
  - Build a WFOMC circuit $C_{\Delta,q_i}$ for $\Delta \land q_i$ conditioned on $e$
  - Compute $WFOMC(\Delta \land e \land q_i, w_T, w_F)$ in $C_{\Delta,q_i}$
  - Return or store $P(q_i|e) = \frac{WFOMC(\Delta e \land q_i, w_T, w_F)}{WFOMC(\Delta e, w_T, w_F)}$
MLNs for WFOMCs

- Weights in MLNs specified for formulas instead of single predicates
  - E.g., example from the beginning
    - \((\ln 7, travel(X) \land epid \land sick(X))\),
    - \((\ln 2, \neg travel(X) \lor \neg epid \lor \neg sick(X))\)
  - Trick:
    - Introduce a new predicate \(\theta_i\) containing all free variables of \(\psi_i\) as equivalent to \(\psi_i\)
      - \(\forall X \in \text{People} : \theta_1(X) \iff (travel(X) \land epid \land sick(X))\)
      - \(\forall X \in \text{People} : \theta_2(X) \iff (\neg travel(X) \lor \neg epid \lor \neg sick(X))\)
    - Specify weight functions such that \(\theta_i\) takes the weight of \(\psi_i\)
      - \(w_T(\theta_1(X)) = \exp(\ln 7) = 7\)
      - \(w_T(\theta_2(X)) = \exp(\ln 2) = 2\)
      - All other predicates and \(\neg \theta_1, \neg \theta_2\) are mapped to 1 by both \(w_T, w_F\)
WFOMC Reduction

• Formally, given an MLN $\Psi = \{(w_i, \psi_i)\}_{i=1}^{n}$
  – Transform each weighted formula $(w_i, \psi_i)$ into an FOL-DC formula
    $\forall X_i, cs_i : \theta_i(X_i) \iff \psi_i$
  – where
    • $X_i$ are the free variables in $\psi_i$
    • $cs_i$ is the constraint set that enforces the domain constraints as given by the MLN
    • $\theta_i(X_i)$ is a new predicate containing all free variables of $\psi_i$
  – Specify weight functions $w_T, w_F$ such that for each
    • $w_T(\theta_i(X_i)) = \exp(w_i)$
    • $w_T(p_i) = 1$ for all predicates $p_i$ occurring in $\Psi$
    • $w_F(\theta_i(X_i)) = 1$
  • Continue with knowledge compilation
Example

• Given
  – \((\ln 7, \text{travel}(X) \land \text{epid} \land \text{sick}(X))\)
  – \((\ln 2, \neg\text{travel}(X) \lor \neg\text{epid} \lor \neg\text{sick}(X))\)

• Resulting theory \((t = \text{travel}, e = \text{epid}, s = \text{sick})\)
  – \(\forall X \in \text{People} : \theta_1(X) \iff (t(X) \land e \land s(X))\)
  – \(\forall X \in \text{People} : \theta_2(X) \iff (\neg t(X) \lor \neg e \lor \neg s(X))\)
  – with weight functions
    • \(w_T(\theta_1(X)) = 7\)
    • \(w_T(\theta_2(X)) = 2\)
    • Rest mapped to 1 by both \(w_T, w_F\)

• Transform formulas into CNF
Example: Normal Form

- Transform formulas into CNF: \( \forall X \in \text{People} : \theta_1 (X) \iff (t(X) \land e \land s(X)) \)

\[
\begin{align*}
\theta_1 (X) & \iff (t(X) \land e \land s(X)) & \text{(resolve } \iff) \\
\equiv (\theta_1 (X) \Rightarrow (t(X) \land e \land s(X))) & \land (\theta_1 (X) \Leftarrow (t(X) \land e \land s(X))) & \text{(De Morgan on } \Rightarrow) \\
\equiv (\neg \theta_1 (X) \lor (t(X) \land e \land s(X))) & \land (\theta_1 (X) \lor \neg(t(X) \land e \land s(X))) & \text{(move } \neg \text{ inward)} \\
\equiv (\neg \theta_1 (X) \lor (t(X) \land e \land s(X))) & \land (\theta_1 (X) \lor \neg t(X) \lor \neg e \lor \neg s(X)) & \text{(distribute } \lor) \\
\equiv (\neg \theta_1 (X) \lor t(X)) & \land (\neg \theta_1 (X) \lor e) & \land (\neg \theta_1 (X) \lor s(X)) & \land (\theta_1 (X) \lor \neg t(X) \lor \neg e \lor \neg s(X)) & \text{(CNF)} \\
\end{align*}
\]

- Result (each conjunct as own formula):
  - \( \forall X \in \text{People} : \neg \theta_1 (X) \lor t(X) \)
  - \( \forall X \in \text{People} : \neg \theta_1 (X) \lor e \)
  - \( \forall X \in \text{People} : \neg \theta_1 (X) \lor s(X) \)
  - \( \forall X \in \text{People} : \theta_1 (X) \lor \neg t(X) \lor \neg e \lor \neg s(X) \)
Example: Normal Form

• Transform formulas into CNF: \( \forall X \in \text{People} : \theta_2(X) \iff (\neg t(X) \lor \neg e \lor \neg s(X)) \)

\[
\begin{align*}
\theta_2(X) & \iff (\neg t(X) \lor \neg e \lor \neg s(X)) \\
\equiv & \left( \theta_2(X) \Rightarrow (\neg t(X) \lor \neg e \lor \neg s(X)) \right) \land \left( \theta_2(X) \Leftarrow (\neg t(X) \lor \neg e \lor \neg s(X)) \right) \\
\equiv & (\neg \theta_2(X) \lor \neg t(X) \lor \neg e \lor \neg s(X)) \land (\theta_2(X) \lor \neg (\neg \theta_2(X) \lor \neg t(X) \lor \neg e \lor \neg s(X))) \\
\equiv & (\neg \theta_2(X) \lor \neg t(X) \lor \neg e \lor \neg s(X)) \land (\theta_2(X) \lor t(X) \land e \land s(X)) \\
\equiv & (\neg \theta_2(X) \lor \neg t(X) \lor \neg e \lor \neg s(X)) \land (\theta_2(X) \lor t(X)) \land (\theta_2(X) \lor e) \land (\theta_2(X) \lor s(X))
\end{align*}
\]

– Result (each conjunct as own formula):
  • \( \forall X \in \text{People} : \neg \theta_2(X) \lor \neg t(X) \lor \neg e \lor \neg s(X) \)
  • \( \forall X \in \text{People} : \theta_2(X) \lor t(X) \)
  • \( \forall X \in \text{People} : \theta_2(X) \lor e \)
  • \( \forall X \in \text{People} : \theta_2(X) \lor s(X) \)
Example: FO d-DNNF Circuit

- Given theory in CNF
  - $\forall X \in \text{People} : \neg \theta_1(X) \lor t(X)$
  - $\forall X \in \text{People} : \neg \theta_1(X) \lor e$
  - $\forall X \in \text{People} : \neg \theta_1(X) \lor s(X)$
  - $\forall X \in \text{People} : \theta_1(X) \lor \neg t(X) \lor \neg e \lor \neg s(X)$
  - $\forall X \in \text{People} : \neg \theta_2(X) \lor \neg t(X) \lor \neg e \lor \neg s(X)$
  - $\forall X \in \text{People} : \theta_2(X) \lor t(X)$
  - $\forall X \in \text{People} : \theta_2(X) \lor e$
  - $\forall X \in \text{People} : \theta_2(X) \lor s(X)$

- Resulting FO d-DNNF circuit generated by the FOKC implementation
  - Some leaves repeated for readability
Example: FO d-DNNF Circuit

• Given theory in CNF
  1. \( \forall X \in \text{People} : \neg \theta_2(X) \lor \neg t(X) \lor \neg s(X) \lor \neg e \)
  2. \( \forall X \in \text{People} : \theta_1(X) \lor \neg t(X) \lor \neg e \lor \neg s(X) \)
  3. \( \forall X \in \text{People} : \neg \theta_1(X) \lor t(X) \)
  4. \( \forall X \in \text{People} : \neg \theta_1(X) \lor e \)
  5. \( \forall X \in \text{People} : \neg \theta_1(X) \lor s(X) \)
  6. \( \forall X \in \text{People} : \theta_2(X) \lor t(X) \)
  7. \( \forall X \in \text{People} : \theta_2(X) \lor e \)
  8. \( \forall X \in \text{People} : \theta_2(X) \lor s(X) \)
Example: FO d-DNNF Circuit

- Given theory in CNF
  1. $\forall X \in \text{People} : \neg \theta_2(X) \lor \neg t(X) \lor \neg s(X) \lor \neg e$
  2. $\forall X \in \text{People} : \theta_1(X) \lor \neg t(X) \lor \neg e \lor \neg s(X)$
  3. $\forall X \in \text{People} : \neg \theta_1(X) \lor t(X)$
  4. $\forall X \in \text{People} : \neg \theta_1(X) \lor e$
  5. $\forall X \in \text{People} : \neg \theta_1(X) \lor s(X)$
  6. $\forall X \in \text{People} : \theta_2(X) \lor t(X)$
  7. $\forall X \in \text{People} : \theta_2(X) \lor e$
  8. $\forall X \in \text{People} : \theta_2(X) \lor s(X)$
Example: FO d-DNNF Circuit

- Given theory in CNF
  1. \( \forall X \in \text{People} : \) 
     \[ \neg \theta_2(X) \lor \neg t(X) \lor \neg s(X) \lor \neg e \]
  2. \( \forall X \in \text{People} : \) 
     \[ \theta_1(X) \lor \neg t(X) \lor \neg e \lor \neg s(X) \]
  3. \( \forall X \in \text{People} : \) 
     \[ \neg \theta_1(X) \lor t(X) \]
  4. \( \forall X \in \text{People} : \) 
     \[ \neg \theta_1(X) \lor e \]
  5. \( \forall X \in \text{People} : \) 
     \[ \neg \theta_1(X) \lor s(X) \]
  6. \( \forall X \in \text{People} : \) 
     \[ \theta_2(X) \lor t(X) \]
  7. \( \forall X \in \text{People} : \) 
     \[ \theta_2(X) \lor e \]
  8. \( \forall X \in \text{People} : \) 
     \[ \theta_2(X) \lor s(X) \]
Example: Smoothed FO d-DNNF Circuit

- As generated by the FOKC implementation
  - Grey parts new to not-smoothed version
  - Abbreviated depiction of $p \lor \neg p$ in one node
- Not-smoothed version for comparison

\[
\begin{align*}
\neg \text{epid} & \land \forall x, x \in \text{person} \\
\neg \text{epid} & \land \forall x, x \in \text{person} \\
\text{travel}(x) & \land \neg \text{travel}(x) \\
\theta_2(x), x \in \text{person} & \land \neg \theta_1(x), x \in \text{person} \\
\neg \text{sick}(x) & \land \text{sick}(x) \\
\theta_2(x) & \land \neg \theta_1(x) \\
\text{travel}(x) & \land \neg \text{travel}(x) \\
\theta_1(x) & \land \neg \theta_2(x) \\
\neg \theta_1(x) & \land \theta_2(x) \\
\theta_1(x) & \land \neg \theta_2(x) \\
\theta_2(x) & \land \neg \theta_1(x)
\end{align*}
\]
Theoretical Results

• Compilation independent of domain sizes
  – Just like construction of FO jtree is also independent of domain sizes
• Inference
  – Polynomial in domain sizes
    • Based on the computations that are computed at different node types
• Completeness as before
  – $M^{2lv}$
    • Two-logvar theories with max. two logical variables per formula
  – $M^{1prv}$
    • One logical variable per predicate
Implementation

• Available at
  – [https://github.com/UCLA-StarAI/Forclift](https://github.com/UCLA-StarAI/Forclift)
    • May no longer work according to Guy so you may have to try
      – [https://github.com/tanyabraun/wfomc](https://github.com/tanyabraun/wfomc)
  – Officially three input formats
    • Based on the normal form required (.wmc)
    • Early version of parfactor graphs (.fg)
    • MLN version (.mln)
    → MLN file format only one I got the compiled version to parse
Runtimes: Increasing Domain Sizes

• Example model with all domain sizes $\in \{2, 4, \ldots, 20, 30, \ldots, 100, 200, \ldots, 1000\}$
• No evidence
• Queries: $P(\text{Travel}(x_1)), P(\text{Sick}(x_1)), P(\text{Treat}(x_1, m_1)), P(\text{Nat}(d_1)), P(\text{Man}(w_1)), P(\text{Epid})$
• Compare query answering times of different inference algorithms
  - Propositional: VE, JT
  - Lifted: LVE, LJT, FOKC
    • Compare trade-off (overhead vs. fast inference) between single / multi-query algs.
• Test
  - Increasing
    • Ground width $w_g$
      – Default: 3
    • Counting width $w_\#$
      – Default: 1
    • Number of nodes $n_J$
      – Default: 3
    • Domain size $n$
      – Default: 1000
    • Based on $O(n_J \cdot \log_2(n) \cdot r^{w_g} \cdot n^{w_\#})$
**Queries Answering**

FOKC almost invariant w.r.t. domain sizes whereas count conversion hits LVE-based algorithms

FOKC does not build histograms, which blow up the representation for LVE-based algs.

Runtimes in milliseconds
Default: $n = 1000, n_j = 3, w_g = 3, w_# = 1$
Trade-off Evaluation: Criteria

• For multi-query algorithms
  – Overhead to set off (model is compiled into a helper structure)
  vs.
  – Shorter individual query answering time

• With
  – $t_{q,cpl}$ runtime for answering single query with an algorithm that uses compilation
  – $t_{q,uncpl}$ runtime for answering single query with an algorithm without compilation
  – $t_{c,cpl}$ runtime for compilation with an algorithm that uses compilation

What is the ratio between individual query answering times?

$$\alpha = \frac{t_{q,cpl}}{t_{q,uncpl}}$$

How many queries does it take to offset the overhead?

$$\beta = \frac{t_{c,cpl}}{t_{q,uncpl} - t_{q,cpl}}$$

• Makes only sense if $\alpha < 1$
Trade-off

Default: $n_f = 3, w_g = 3, w_\# = 1$
Probabilistic Theorem Proving (PTP)

• Based on theorem proving in logics
• Solves lifted weighted model counting problem
  – Similar to the weighted first-order model counting problem by Guy Van den Broeck
  – MLNs as input

• Implementation available: Alchemy
  – http://alchemy.cs.washington.edu
  – Input format: MLNs
Summary

• Propositional (weighted) model counting
  – WMC definition
  – Circuits:
    • Inner nodes: conjunctions/disjunctions
    • Leaves: literals, \textit{true}, \textit{false}
    • Properties: d-DNNF, smooth
    • Model counts, WMC by propagation
  – Knowledge compilation: Inference in circuits, i.e., query answering by weighted model counting in circuits
• Lifted (weighted) model counting
  – WFOMC definition
  – FO circuits: Inner nodes can also be set conjunctions/disjunctions
  – FOKC: Inference in FO circuits
Outline: 4. Lifted Inference

A. Exact Inference
   i. Lifted Variable Elimination for Parfactor Models
      • Idea, operators, algorithm, complexity
   ii. Lifted Junction Tree Algorithm
      • Idea, helper structure: junction tree, algorithm
   iii. First-order Knowledge Compilation for MLNs
      • Idea, helper structure: circuit, algorithm