Weighted Model Counting in First Order Logic

Doc in Progress

@ Department of Mathematics, UNITN

Sagar Malhotra[1,2] and Luciano Serafini [1]

[1]Fondazione Bruno Kessler
[2]University of Trento

Model Counting

Given a logical formula Φ. e.g.

$$\text{Trentino} \rightarrow \text{Climber}$$

MC ?

Model Counting

Given a logical formula Φ. e.g.

$$\text{Trentino} \rightarrow \text{Climber}$$

MC ?

Trentino	Climber	Trentino $\rightarrow$ Climber
1	1	1
1	0	0
0	1	1
0	0	1

$$\text{MC} = 3$$

Weighted Model Counting

Given a logical formula Φ. e.g.

$$\text{Trentino} \rightarrow \text{Climber}$$

And a weight function e.g.

$$w(\text{Trentino}) = 2.5 \qquad w(\neg\text{Trentino}) = 5$$
$$w(\text{Climber}) = 0.5 \qquad w(\neg\text{Climber}) = 1.5$$

WMC ?

Weighted Model Counting

Given a logical formula Φ. e.g.

$$\text{Trentino} \rightarrow \text{Climber}$$

And a weight function e.g.

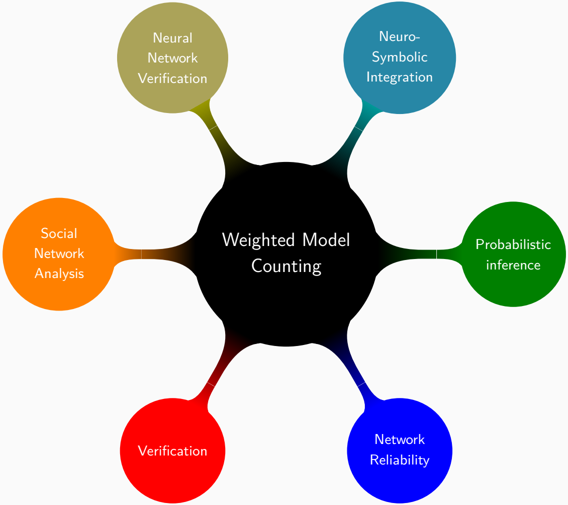
$$w(\text{Trentino}) = 2.5 \qquad w(\neg\text{Trentino}) = 5$$
$$w(\text{Climber}) = 0.5 \qquad w(\neg\text{Climber}) = 1.5$$

WMC ?

Trentino	Climber	Trentino $\rightarrow$ Climber
1	1	$2.5 \times 0.5 = 1.25$
1	0	$2.5 \times 1.5 = 1.25$
0	1	$5 \times 0.5 = 1.5$
0	0	$5 \times 1.5 = 7.5$

WMC $= 11.25$

WMC Applications



Weighted Model Counting: Approaches

- Devising heuristic/approximation algorithms
  WMC is intractable !

Weighted Model Counting: Approaches

- Devising heuristic/approximation algorithms
  WMC is intractable !
- Identifying logical languages that admit efficient WMC
  Today's presentation

Weighted First Order Model Counting

$$\forall x.\text{Trentino}(x) \rightarrow \text{Climber}(x)$$

A weight function associated to predicates:

$$w(\text{Trentino}) = 2.5 \qquad w(\neg\text{Trentino}) = 5$$
$$w(\text{Climber}) = 0.5 \qquad w(\neg\text{Climber}) = 1.5$$

What is the WFOMC for n people ?

Weighted First Order Model Counting

$$\forall x.\text{Trentino}(x) \rightarrow \text{Climber}(x)$$

A weight function associated to predicates:

$$w(\text{Trentino}) = 2.5 \qquad w(\neg\text{Trentino}) = 5$$
$$w(\text{Climber}) = 0.5 \qquad w(\neg\text{Climber}) = 1.5$$

What is the WFOMC for n people ?

Can we compute it in PTIME?

Weighted First Order Model Counting

$$\forall x. \text{Trentino}(x) \rightarrow \text{Climber}(x)$$

A weight function associated to predicates:

$$w(\text{Trentino}) = 2.5 \qquad w(\neg\text{Trentino}) = 5$$
$$w(\text{Climber}) = 0.5 \qquad w(\neg\text{Climber}) = 1.5$$

What is the WFOMC for n people ?

Can we compute it in PTIME?

Can we get a closed form formula ?

FOMC in the two variable fragment

$\mathbf{FO}^2$ Language: 1-Types

We have a language with **at most two variables** , with the following predicates:

- A unary predicate $\mathtt{FBK}(\mathtt{x})$
- A binary predicate $\mathtt{Shaves}(\mathtt{x}, \mathtt{y})$

FO^2 Language: 1-Types

We have a language with **at most two variables** , with the following predicates:

- A unary predicate $\mathtt{FBK}(\mathtt{x})$
- A binary predicate $\mathtt{Shaves}(\mathtt{x}, \mathtt{y})$

We have the following set of unary properties also called 1-types:

$$\neg\mathtt{FBK}(\mathtt{c}) \wedge \neg\mathtt{Shaves}(\mathtt{c}, \mathtt{c})$$

$$\neg\mathtt{FBK}(\mathtt{c}) \wedge \mathtt{Shaves}(\mathtt{c}, \mathtt{c})$$

$$\mathtt{FBK}(\mathtt{c}) \wedge \neg\mathtt{Shaves}(\mathtt{c}, \mathtt{c})$$

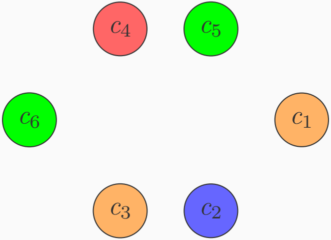
$$\mathtt{FBK}(\mathtt{c}) \wedge \mathtt{Shaves}(\mathtt{c}, \mathtt{c})$$

1- Type Enumeration

An arrangement of 1-Types

$\neg\texttt{FBK}(\texttt{c}) \land \neg\texttt{Shaves}(\texttt{c},\texttt{c})$

$\neg\texttt{FBK}(\texttt{c}) \land \texttt{Shaves}(\texttt{c},\texttt{c})$



$\boldsymbol{k}:\quad k_1 = 1 \quad k_2 = 2 \quad k_3 = 1 \quad k_4 = 2$

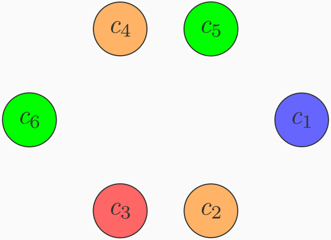
$\texttt{FBK}(\texttt{c}) \land \neg\texttt{Shaves}(\texttt{c},\texttt{c})$

$\texttt{FBK}(\texttt{c}) \land \texttt{Shaves}(\texttt{c},\texttt{c})$

1- Type Enumeration

Another arrangement of 1-Types

$\neg\texttt{FBK}(\texttt{c}) \land \neg\texttt{Shaves}(\texttt{c},\texttt{c})$ $\neg\texttt{FBK}(\texttt{c}) \land \texttt{Shaves}(\texttt{c},\texttt{c})$

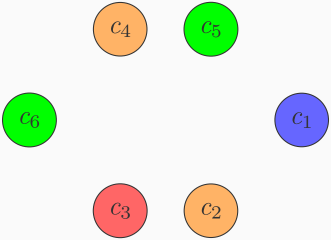


$\boldsymbol{k}$ **:** $k_1 = 1$ $k_2 = 2$ $k_3 = 1$ $k_4 = 2$

$\texttt{FBK}(\texttt{c}) \land \neg\texttt{Shaves}(\texttt{c},\texttt{c})$ $\texttt{FBK}(\texttt{c}) \land \texttt{Shaves}(\texttt{c},\texttt{c})$

1-Type Enumeration

Counting for fixed 1-Type cardinalities



$\boldsymbol{k}: \quad k_1 = 1 \quad k_2 = 2 \quad k_3 = 1 \quad k_4 = 2$

$$\#\text{Similar Arrangements} = \binom{n}{k_1, k_2, k_3, k_4} = \frac{6!}{1!2!1!2!} = 180$$

FO^2 Language: 2-tables

We have an FO^2 language, with the following predicates:

- A unary predicate $\mathtt{FBK}(\mathtt{x})$
- A binary predicate $\mathtt{Shaves}(\mathtt{x}, \mathtt{y})$

We have the following set binary properties also called 2-tables:

$$\mathtt{Shaves}(\mathtt{c}, \mathtt{d}) \land \mathtt{Shaves}(\mathtt{d}, \mathtt{c})$$

$$\neg\mathtt{Shaves}(\mathtt{c}, \mathtt{d}) \land \mathtt{Shaves}(\mathtt{d}, \mathtt{c})$$

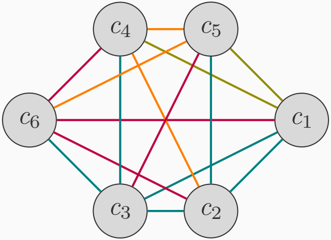
$$\mathtt{Shaves}(\mathtt{c}, \mathtt{d}) \land \neg\mathtt{Shaves}(\mathtt{d}, \mathtt{c})$$

$$\neg\mathtt{Shaves}(\mathtt{c}, \mathtt{d}) \land \neg\mathtt{Shaves}(\mathtt{d}, \mathtt{c})$$

2- Table Enumeration

An arrangement for 2-tables

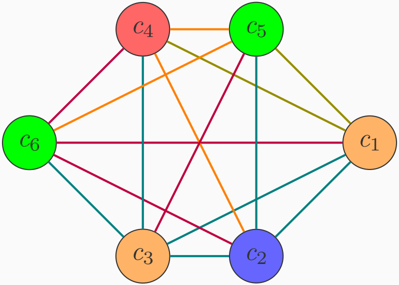
$\texttt{Shaves}(\texttt{c}, \texttt{d}) \wedge \texttt{Shaves}(\texttt{d}, \texttt{c})$ $\neg\texttt{Shaves}(\texttt{c}, \texttt{d}) \wedge \texttt{Shaves}(\texttt{d}, \texttt{c})$



$\texttt{Shaves}(\texttt{c}, \texttt{d}) \wedge \neg\texttt{Shaves}(\texttt{d}, \texttt{c})$ $\neg\texttt{Shaves}(\texttt{c}, \texttt{d}) \wedge \neg\texttt{Shaves}(\texttt{d}, \texttt{c})$

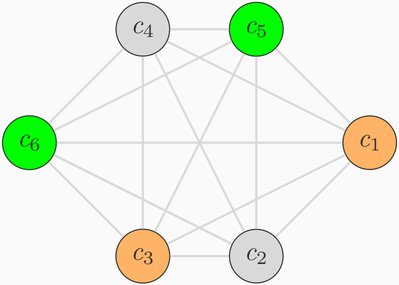
2- Table Enumeration given 1-Types

An arrangement for 2-tables given 1-types



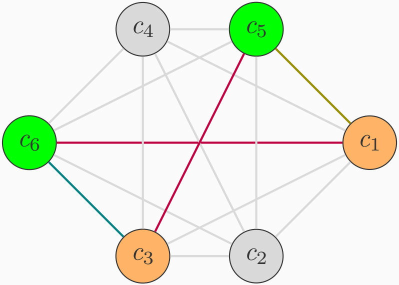
2- Table Enumeration given 1-Types

Picking a sub graph: Pick a pair of 1-Types



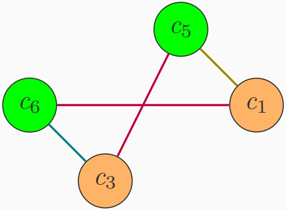
2- Table Enumeration given 1-Types

Picking a sub graph: Pick 2-Tables between them



2- Table Enumeration

Enumerating 2-tables given 1-types

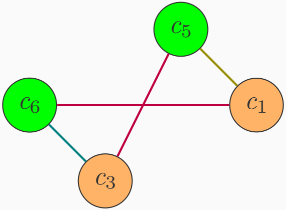


$$k_2 = 2 \quad k_4 = 2$$

$$h_1 = 2 \quad h_2 = 1 \quad h_3 = 1 \quad h_4 = 0$$

2- Table Enumeration

Enumerating 2-tables given 1-types



$$k_2 = 2 \quad k_4 = 2$$

$$h_1 = 2 \quad h_2 = 1 \quad h_3 = 1 \quad h_4 = 0$$

$$\binom{k_2 \times k_4}{h_1 \quad h_2 \quad h_3 \quad h_4} = \frac{(2 \times 2)!}{2!2!1!0!} = 6$$

Enumerating all models over 1-types and 2-tables

$$\sum_{\vec{k},\vec{h}} \binom{n}{k_1 \ldots k_u} \prod_{1 \leq i \leq j \leq u} \binom{k(i,j)}{h_1^{ij} \ldots h_b^{ij}}$$

$$\boldsymbol{k}(i,j) = \begin{cases} k_i k_j & \text{if } i \neq j \\ \frac{k_i(k_i-1)}{2} & \text{if } i = j \end{cases}$$

Adding Formulas : $\forall xy.\Phi(x, y)$

A formula $\forall xy.\Phi(x, y)$, allows some and disallows other 1-Type and 2-Table configuration. For Example:

$$\neg \texttt{Shaves}(\texttt{x}, \texttt{x})$$

$$\texttt{Shaves}(\texttt{x}, \texttt{y}) \rightarrow \texttt{Shaves}(\texttt{y}, \texttt{x})$$

$$\texttt{FBK}(\texttt{x}) \wedge \texttt{Shaves}(\texttt{x}, \texttt{y}) \rightarrow \texttt{FBK}(\texttt{y})$$

Adding Formulas : $\forall xy.\Phi(x, y)$

$$\neg \mathtt{Shaves}(\mathtt{x}, \mathtt{x})$$
$$\mathtt{Shaves}(\mathtt{x}, \mathtt{y}) \rightarrow \mathtt{Shaves}(\mathtt{y}, \mathtt{x})$$
$$\mathtt{FBK}(\mathtt{x}) \wedge \mathtt{Shaves}(\mathtt{x}, \mathtt{y}) \rightarrow \mathtt{FBK}(\mathtt{y})$$

Not allowed:

$$\neg \mathtt{FBK}(\mathtt{c}) \wedge \neg \mathtt{Sh}(\mathtt{c}, \mathtt{c}) \wedge \mathtt{FBK}(\mathtt{d}) \wedge \neg \mathtt{Sh}(\mathtt{d}, \mathtt{d}) \wedge \mathtt{Sh}(\mathtt{c}, \mathtt{d}) \wedge \mathtt{Sh}(\mathtt{d}, \mathtt{c})$$

Allowed:

$$\mathtt{FBK}(\mathtt{c}) \wedge \neg \mathtt{Sh}(\mathtt{c}, \mathtt{c}) \wedge \mathtt{FBK}(\mathtt{d}) \wedge \neg \mathtt{Sh}(\mathtt{d}, \mathtt{d}) \wedge \mathtt{Sh}(\mathtt{c}, \mathtt{d}) \wedge \mathtt{Sh}(\mathtt{d}, \mathtt{c})$$

FOMC in FO^2: $\forall x \forall y. \Phi(x, y)$

$$\mathrm{FOMC}(\Phi, n) =$$

$$\sum_{\vec{\boldsymbol{k}}, \vec{\boldsymbol{h}}} \binom{n}{k_1, \ldots, k_u} \prod_{1 \leq i \leq j \leq u} \binom{\boldsymbol{k}(i,j)}{h_1^{ij}, \ldots, h_b^{ij}} \prod_{1 \leq v \leq b} {n_{ijv}}^{h_v^{ij}}$$

Unary Properties

Constraints: Φ

Binary Properties

Extensions

Cardinality Constraints

$$\neg \mathrm{Shaves}(x, x)$$

$$\mathrm{Shaves}(x, y) \rightarrow \mathrm{Shaves}(y, x)$$

$$\mathrm{FBK}(x) \wedge \mathrm{Shaves}(x, y) \rightarrow \mathrm{FBK}(y)$$

$$|\mathrm{FBK}(x)| = 2$$

Cardinality Constraints

$$\neg \mathtt{Shaves}(x, x)$$

$$\mathtt{Shaves}(x, y) \rightarrow \mathtt{Shaves}(y, x)$$

$$\mathtt{FBK}(x) \land \mathtt{Shaves}(x, y) \rightarrow \mathtt{FBK}(y)$$

$$|\mathtt{FBK}(x)| = 2$$

$$k_1 : \neg\mathtt{FBK}(c) \land \neg\mathtt{Shaves}(c, c)$$

$$k_2 : \neg\mathtt{FBK}(c) \land \mathtt{Shaves}(c, c)$$

$$k_3 : \mathtt{FBK}(c) \land \neg\mathtt{Shaves}(c, c)$$

$$k_4 : \mathtt{FBK}(c) \land \mathtt{Shaves}(c, c)$$

Cardinality Constraints

$$\neg \mathtt{Shaves}(x, x)$$

$$\mathtt{Shaves}(x, y) \to \mathtt{Shaves}(y, x)$$

$$\mathtt{FBK}(x) \wedge \mathtt{Shaves}(x, y) \to \mathtt{FBK}(y)$$

$$|\mathtt{FBK}(x)| = 2$$

$$k_1 : \neg \mathtt{FBK}(c) \wedge \neg \mathtt{Shaves}(c, c)$$

$$k_2 : \neg \mathtt{FBK}(c) \wedge \mathtt{Shaves}(c, c)$$

$$k_3 : \mathtt{FBK}(c) \wedge \neg \mathtt{Shaves}(c, c)$$

$$k_4 : \mathtt{FBK}(c) \wedge \mathtt{Shaves}(c, c)$$

$$\boldsymbol{k} : k_3 + k_4 = 2$$

Principle of Inclusion Exclusion

- Let Ω be a set of objects
- $\mathcal{S} = \{S_1, \ldots, S_m\}$ be a set of properties of Ω
- e_0 : The count of objects with **NONE** of the properties in S
- Let $Q \subseteq S$, then N_Q is the count of objects with **AT LEAST** the properties in Q

Principle of Inclusion Exclusion

- Let Ω be a set of objects
- $\mathcal{S} = \{S_1, \ldots, S_m\}$ be a set of properties of Ω
- e_0 : The count of objects with **NONE** of the properties in S
- Let $Q \subseteq S$, then N_Q is the count of objects with **AT LEAST** the properties in Q

We define,

$$s_l = \sum_{|\mathcal{Q}|=l} N_Q \tag{1}$$

Principle of Inclusion Exclusion

- Let Ω be a set of objects
- $\mathcal{S} = \{S_1, \ldots, S_m\}$ be a set of properties of Ω
- e_0 : The count of objects with **NONE** of the properties in S
- Let $Q \subseteq S$, then N_Q is the count of objects with **AT LEAST** the properties in Q

We define,

$$s_l = \sum_{|\mathcal{Q}|=l} N_Q \tag{1}$$

Then the following relation holds:

$$e_0 = \sum_{l=0}^{m} (-1)^l s_l \tag{2}$$

FOMC Existential Quantifiers (Example)

$$\forall \mathtt{xy}.\Phi(x, y) \land \forall \mathtt{x}\ \exists \mathtt{y}.\mathtt{Sh}(\mathtt{x}, \mathtt{y})$$

FOMC Existential Quantifiers (Example)

$$\forall \mathtt{xy}.\Phi(x, y) \wedge \forall \mathtt{x}\ \exists \mathtt{y}.\mathtt{Sh}(\mathtt{x}, \mathtt{y})$$

$$\Omega = \{\omega : \omega \models \forall xy.\Phi(x, y)\}$$

FOMC Existential Quantifiers (Example)

$$\forall \mathtt{xy}.\Phi(x,y) \wedge \forall \mathtt{x}\ \exists \mathtt{y}.\mathtt{Sh}(\mathtt{x},\mathtt{y})$$

$$\Omega = \{\omega : \omega \models \forall xy.\Phi(x,y)\}$$

$$S_c = \{\omega : \omega \models \forall xy.\Phi(x,y) \wedge \forall y.\neg Sh(c,y)\}$$

FOMC Existential Quantifiers (Example)

$$\forall \mathtt{xy}.\Phi(x,y) \wedge \forall \mathtt{x}\ \exists \mathtt{y}.\mathtt{Sh}(\mathtt{x},\mathtt{y})$$

$$\Omega = \{\omega : \omega \models \forall xy.\Phi(x,y)\}$$

$$S_c = \{\omega : \omega \models \forall xy.\Phi(x,y) \wedge \forall y.\neg Sh(c,y)\}$$

$$e_0 = \text{FOMC}(\forall xy.\Phi(x,y) \wedge \forall x\ \exists y.Sh(x,y))$$

FOMC Existential Quantifiers (Example)

$$\forall \mathtt{xy}.\Phi(x, y) \land \forall \mathtt{x}\ \exists \mathtt{y}.\mathtt{Sh}(\mathtt{x}, \mathtt{y})$$

$$\Omega = \{\omega : \omega \models \forall xy.\Phi(x, y)\}$$

$$S_c = \{\omega : \omega \models \forall xy.\Phi(x, y) \land \forall y.\neg Sh(c, y)\}$$

$$e_0 = \mathrm{FOMC}(\forall xy.\Phi(x, y) \land \forall x\ \exists y.Sh(x, y))$$

$$s_l = \mathrm{FOMC}(\forall xy.\Phi(x, y) \land P(x) \rightarrow \neg Sh(x, y) \land (|P| = l))$$

FOMC Existential Quantifiers (Example)

$$\forall \mathtt{xy}.\Phi(x, y) \wedge \forall \mathtt{x}\ \exists \mathtt{y}.\mathtt{Sh}(\mathtt{x}, \mathtt{y})$$

$$\Omega = \{\omega : \omega \models \forall xy.\Phi(x, y)\}$$

$$S_c = \{\omega : \omega \models \forall xy.\Phi(x, y) \wedge \forall y.\neg Sh(c, y)\}$$

$$e_0 = \mathrm{FOMC}(\forall xy.\Phi(x, y) \wedge \forall x\ \exists y.Sh(x, y))$$

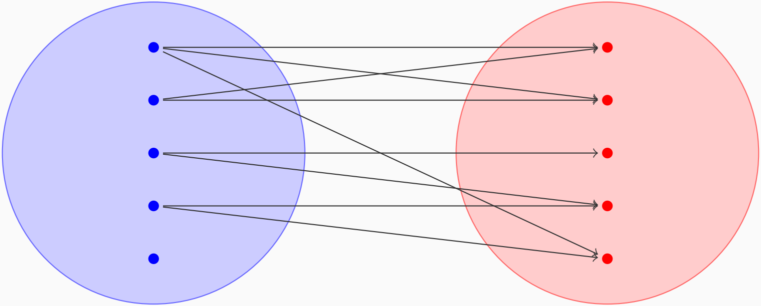
$$s_l = \mathrm{FOMC}(\forall xy.\Phi(x, y) \wedge P(x) \rightarrow \neg Sh(x, y) \wedge (|P| = l))$$

From principle of inclusion-exclusion:

$$e_0 = \sum_{l=1}^{n} (-1)^l s_l \tag{3}$$

Counting Quantifiers: Expressing Functionality

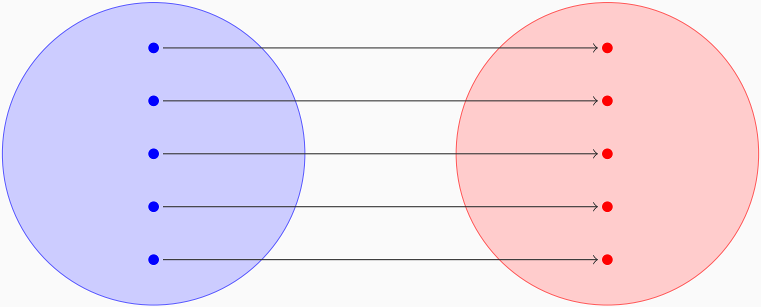
$R(x, y) : \Delta \to \Delta$



Δ Δ

Counting Quantifiers: Expressing Functionality

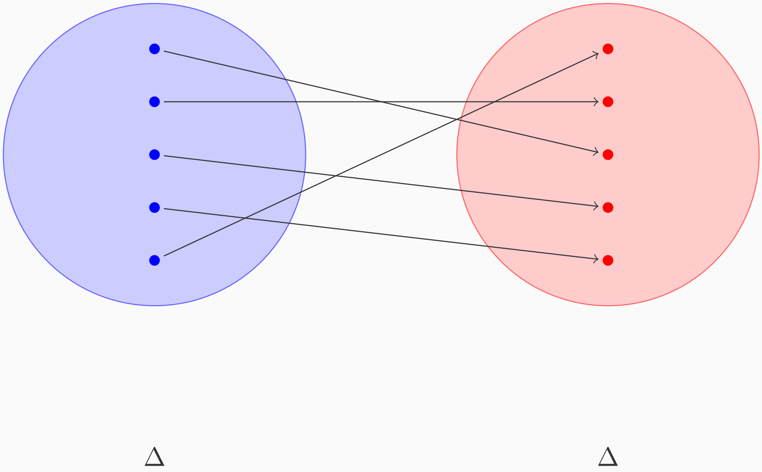
$$\forall x\, \exists^{=1} y.\ R(x, y)$$



Δ Δ

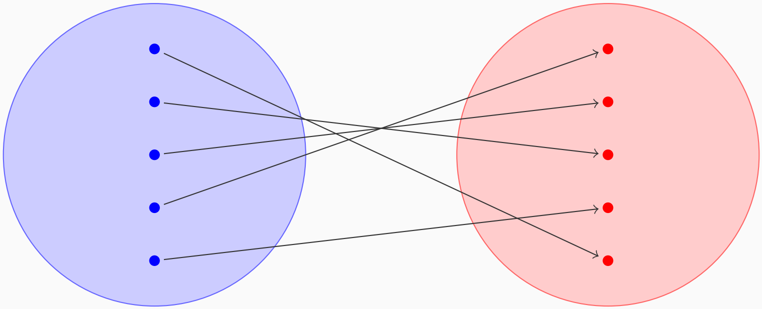
Counting Quantifiers: Expressing Functionality

$$\forall x\, \exists^{=1} y.\ R(x, y)$$



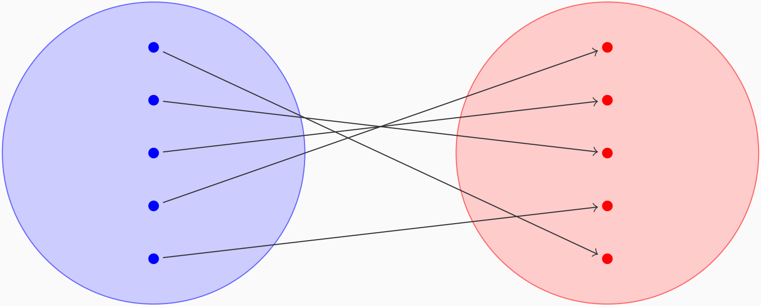
Counting Quantifiers: Expressing Functionality

$$\forall x\, \exists^{=1} y.\ R(x, y)$$



Counting Quantifiers: Expressing Functionality

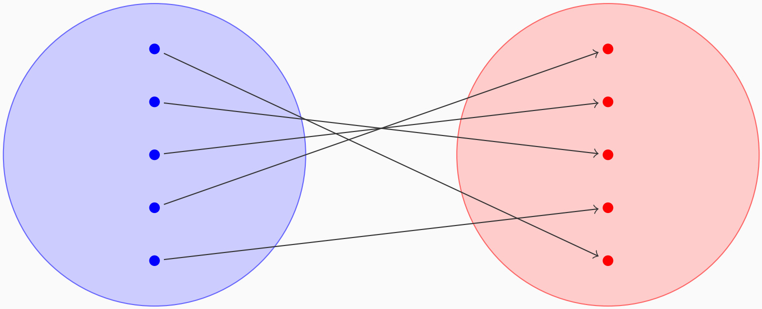
$$\forall x\, \exists^{=1} y.\ R(x, y)$$



What is the cardinality of R in all functions ??

Counting Quantifiers: Expressing Functionality

$$\forall x\, \exists^{=1} y.\ R(x, y)$$



What is the cardinality of R in all functions ??

$$|\Delta|$$

Counting Quantifiers: Functionality Constraint

$$\forall x\, \exists^{=1} y. R(x, y)$$

$$\forall x\, \exists y. R(x, y) \land (|R| = |n|)$$

Counting Quantifiers: Functionality Constraint

$$\forall x \, \exists^{=1} y. R(x, y)$$

$$\forall x \, \exists y. R(x, y) \wedge (|R| = |n|)$$

Given any formula Φ, $\Phi \wedge \forall x \, \exists y. R(x, y) \wedge (|R| = |n|)$ allows only the models where R is functional.

Conclusion

Weighted Model Counting is an assembly language to a vast array of problems.

Conclusion

Weighted Model Counting is an assembly language to a vast array of problems.

Only first-order logic fragments have ever been shown to admit exact tractable counting.

Conclusion

Weighted Model Counting is an assembly language to a vast array of problems.

Only first-order logic fragments have ever been shown to admit exact tractable counting.

In this work, we expand the fragments admitting tractable counting and provide a combinatorial framework that admits closed form formulae.

Future Works

- Consistency of probabilistic inference [Preprint Online]
- Expressing Scalable Consistent Models [Work in Progress]
- Approximate counting with guarantees [Work in Progress]

Thank You !