# Probabilistic Relational Models (PRMs)

Statistical Relational Artificial Intelligence
(StaRAI)

$$10 \; Presents(X, P, C) \Rightarrow Attends(X, C)$$

$$3.75 \; Publishes(X, C) \wedge FarAway(C) \Rightarrow Attends(X, C)$$

Tanya Braun

Contents

Outline: 2. Probabilistic Relational Models (PRMs)

A. *Parfactor models (PMs)*

- Motivation: Symmetries and relations
- Syntax, semantics
- Graphical representation
- Inference tasks

B. *Markov logic networks (MLNs)*

- Syntax, semantics of MLNs
- Graphical representation
- Turning MLNs into PMs and vice versa

Exponential Blowup! → Sparse Encoding

Acc	Nat	Treat	Epid	Travel	Sick	P
false	false	false	false	false	false	0.025
false	false	false	false	false	true	0.009
false	false	false	false	true	false	
false	false	false	false	true		
false	false	false	true			
false	false	false				
false	false					
false						

$$3 \cdot 2^3 + 2$$
$$= 26 \text{ entries}$$
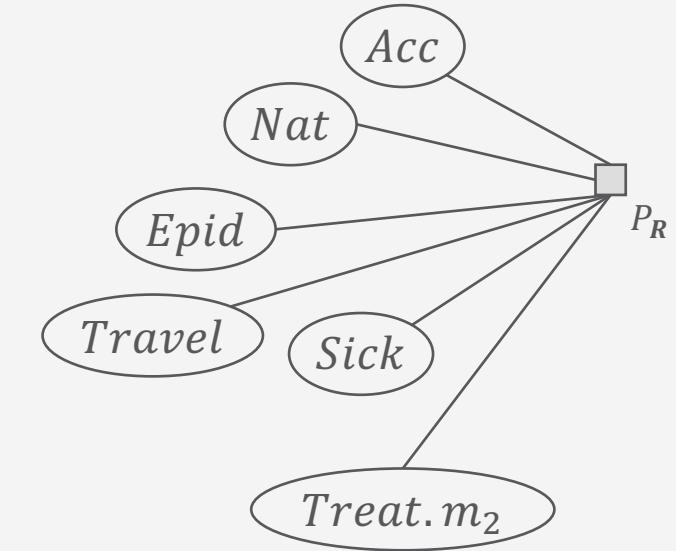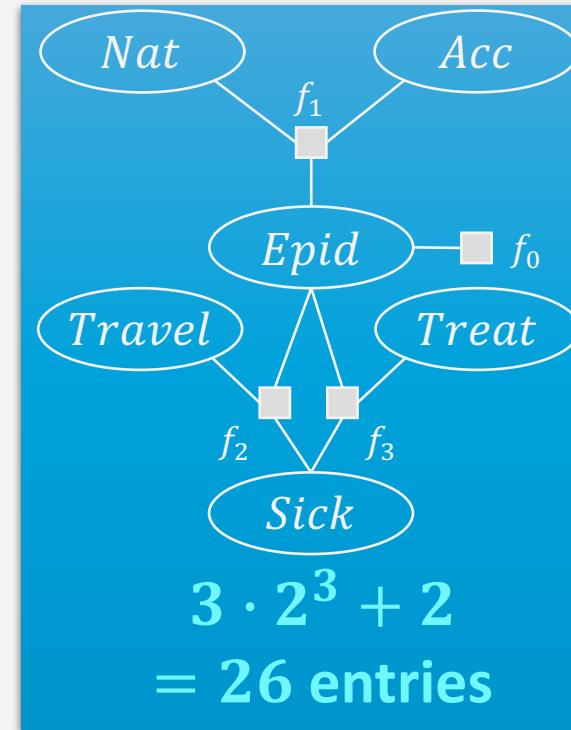
$$2^6 = 64 \text{ possible worlds}$$

Adding relations means adding $Sick, Treat, Travel$
variables for each person, blowing up the model further

Problem: Adding Objects and Relations in Propositional Formalisms Clunky…
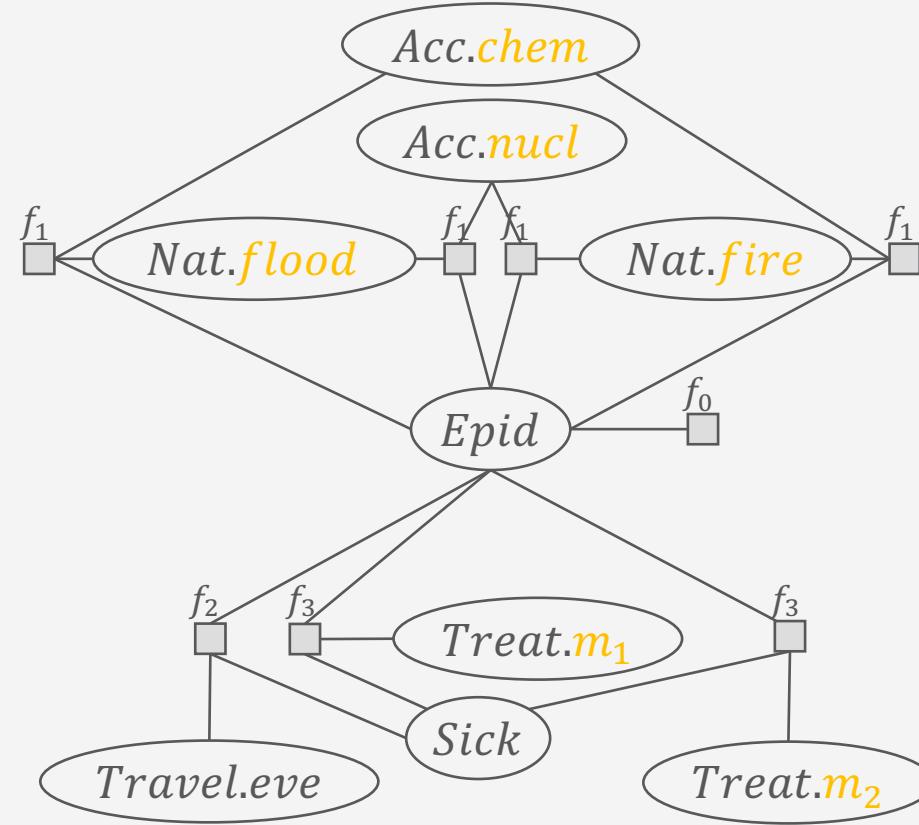
Problem: … and Models Explode with Them
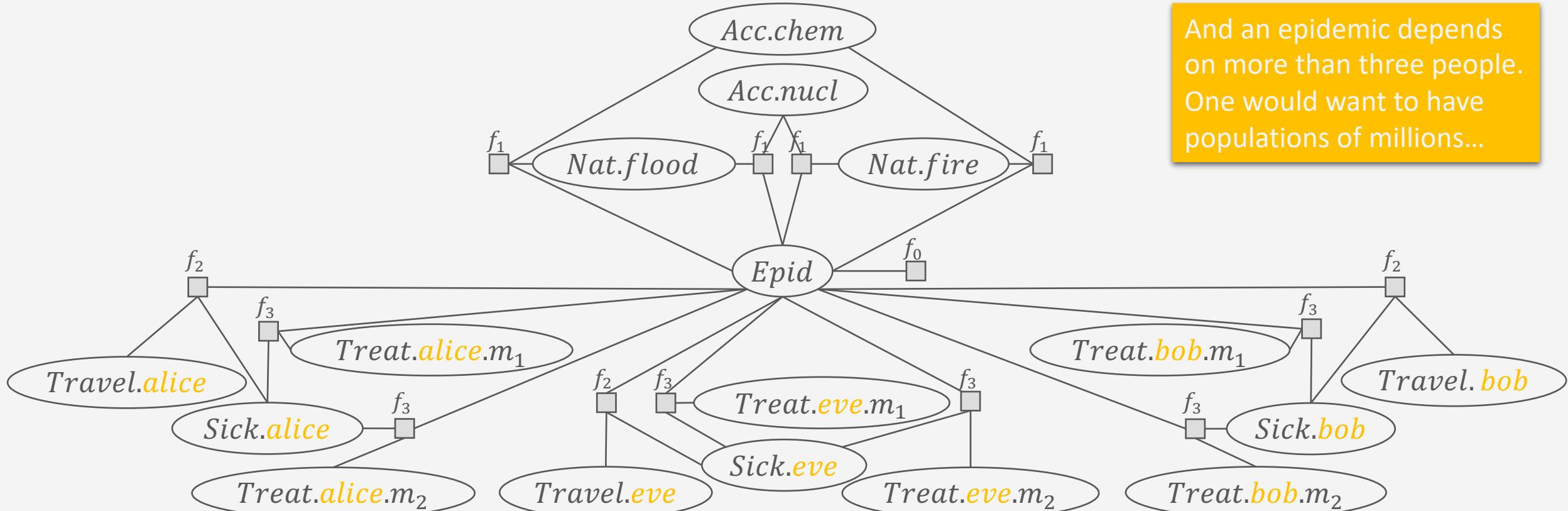
And an epidemic depends on more than three people. One would want to have populations of millions…

$13 \cdot 2^3 + 2 = 104$ **entries in 14 factors, 17 variables**

Propositional → First-order View
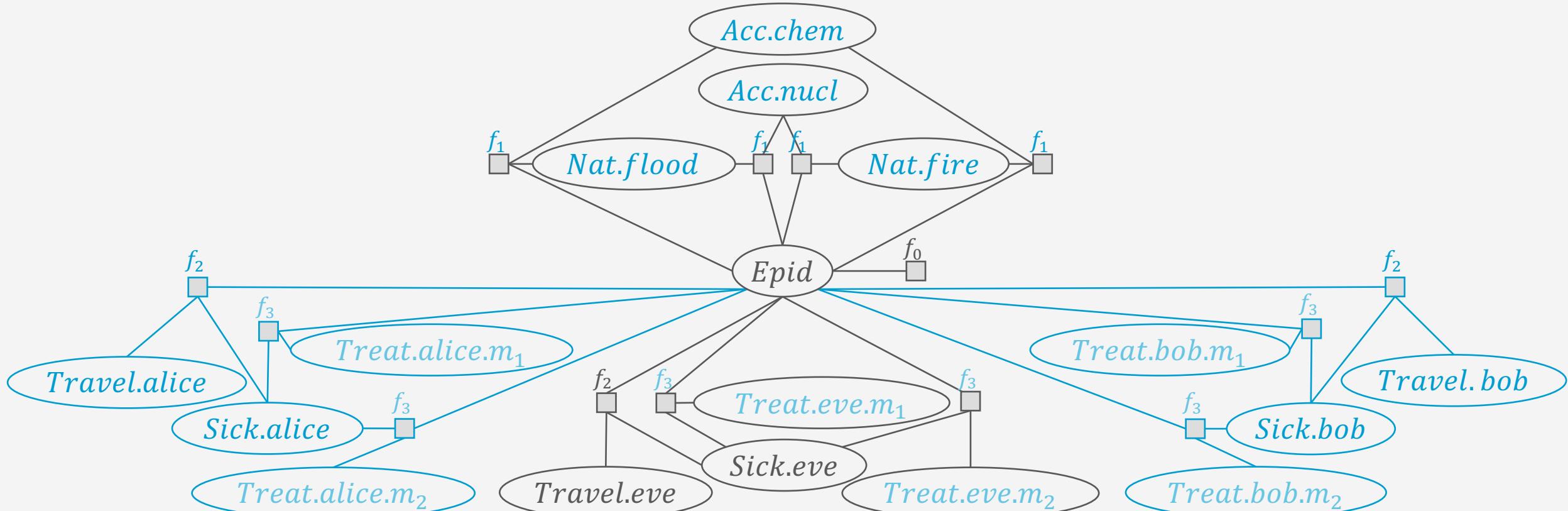
Symmetries in graph / relations in scenario

Propositional → First-order View

From logic perspective

- Instead of *propositional* constraints, we have *first-order* constraints:

$$\frac{epid \Rightarrow sick}{epid \Rightarrow sick.eve}$$

$$met.eve.alice \land sick.eve \Rightarrow sick.alice$$
$$met.alice.eve \land sick.alice \Rightarrow sick.eve$$
$$\vdots$$

$$\forall x : epid \Rightarrow sick(x)$$
$$\forall x (\exists y : met(x,y) \land sick(x) \Rightarrow sick(y))$$

- But probabilistic perspective allows soft constraints with a probability assigned

From probabilistic perspective

- Add *logical variables* to random variables

Epid	Sick	P
false	false	0.8
false	true	0.1
true	false	0.6
true	true	0.4

Epid	Sick.eve	ϕ
false	false	
⋮	⋮	

Epid	Sick(X)	ϕ
false	false	9
false	true	1
true	false	6
true	true	4

Encoding Relations and Symmetries

- Logical variables to encode
  - Recurring structures in graphs
  - Relational structures in data
- By parameterising random variables with logical variables
- And using those parameterised random variables (PRVs) as inputs to factors
  - Forming parametric factors (parfactors)
- Allowing to encode
  - Duplicate factors only once
  - That factors apply to multiple individuals (constants)

David Poole: First-order Probabilistic Inference. In *IJCAI-03 Proceedings of the 18th International Joint Conference on Artificial Intelligence*, 2003.
Brian Milch, Luke S. Zettelmoyer, Kristian Kersting, Michael Haimes, and Leslie Pack Kaelbling. Lifted Probabilistic Inference with Counting Formulas AAAI-08 Proceedings of the 23rd Conference on Artificial Intelligence, 2008.
Nima Taghipour: Lifted Probabilistic Inference by Variable Elimination. KU Leuven, 2013.
Tanya B: Rescued from a Sea of Queries: Exact Inference in Probabilistic Relational Models. UzL, 2020.

Notations

- Variable: capitalised character/string
  - Propositional random variable: speaking names or if abstract: R, R_1, R_i (default)
  - PRV: speaking names or if abstract: A, A_1, A_i (default)
  - Logical variable: X (default)
  - Value: lowercase character/string of corresponding variable
- Set: **boldface**
  - $\boldsymbol{V} = \{V_1, \ldots, V_n\}, \boldsymbol{v} = \{v_1, \ldots, v_n\}$
  - Exception to the rule: model (set of factors) F, relational model (set of parfactors) G
- Sequence: $\mathcal{calligraphy\text{-}face}$
  - $\mathcal{V} = (V_1, \ldots, V_n), \mathcal{v} = (v_1, \ldots, v_n)$
  - In abuse of notation, we use set relations ($\in, \subseteq, \supseteq$) and operations ($\cap, \cup, \setminus$) on sequences, assuming a matching order or if not given, a reasonable reordering

Logical Variables, Domains, Universe

- Logical variable L
  - to encode patterns or relations (first-order constructs)
- Possible values (constants) of logical variables = domain $dom(L) = \{c_1, \ldots, c_k\}$
- E.g.,
  - Logical variables D, M, I, X with domains
    - $dom(D) = \{fire, flood\}$
    - $dom(M) = \{m_1, m_2\}$
    - $dom(I) = \{chem, nucl\}$
    - $dom(X) = \{alice, eve, bob\}$
- Universe $\boldsymbol{D} = \{c_1, \ldots, c_K\}$
  - All constants occurring in a scenario, i.e., $dom(L) \subseteq \boldsymbol{D}$

Constraints

- To restrict applicability of domain constants: Constraint $C = (\mathcal{X}, C_{\mathcal{X}})$
  - $\mathcal{X} = (X_1, \ldots, X_n)$ a sequence of logical variables
  - $C_{\mathcal{X}} \subseteq \times_{i=1}^{n} dom(X_i)$ a subset of domain combinations
    - Set of sequences of constants in the order given by $\mathcal{X}$
  - If no restrictions apply, i.e., $C_{\mathcal{X}} = \times_{i=1}^{n} dom(X_i)$: $C = \top$
    - Can be omitted
- E.g.,
  - $(X, \{(alice), (eve)\})$
  - $((X, M),$

$$\{ \ (alice, m_1), (alice, m_2),$$
$$(eve, m_1), (eve, m_2),$$
$$(bob, m_1), (bob, m_2) \quad \}) = \top$$

> Constraints = Abstraction of a database

> The constraint apparatus appears overwhelming but it just tells you for which constants something applies
> - Necessary for proofs but not for general understanding

PRVs

- Random variable R parameterised with logical variables $X_1, \dots, X_n, n \geq 0$ = PRV

$$A = R(X_1, \dots, X_n)$$

  - if $n = 0$, $A = R$ a propositional random variable
- Range as before: $ran(A) = \{v_1, \dots, v_m\}$
- E.g.,
  - $Treat(X, M)$ with $n = 2$
  - $Epid$ with $n = 0$
  - $Treat(X, M), Travel(X),$ $Sick(X)$ sharing the same logical variable

$Nat(D)$

$Acc(I)$

$Epid$

$Travel(X)$
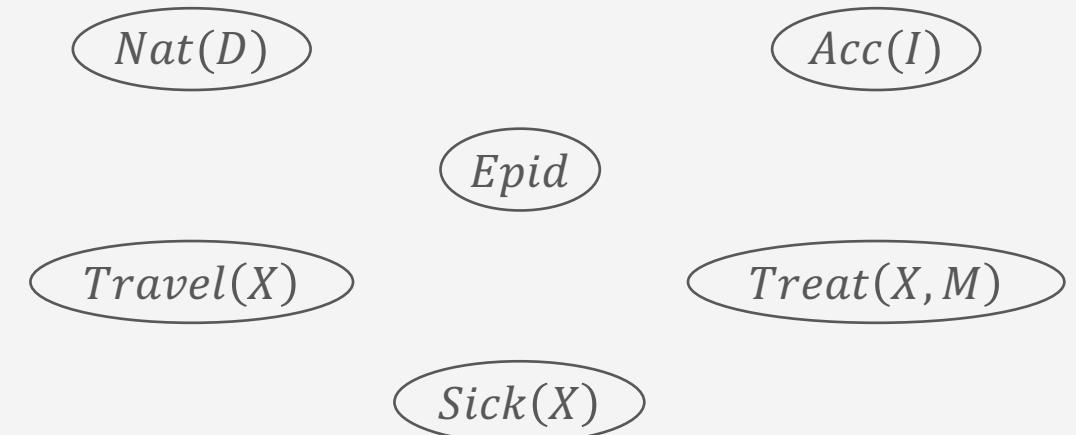
$Treat(X, M)$

$Sick(X)$

Restricting PRVs

- PRV A under constraint C written as $A_{|C}$
  - $C = (\mathcal{X}, C_{\mathcal{X}}), \mathcal{X} \supseteq lv(A)$
    - $lv(.)$ referring to the logical variables of the given input
  - Represents a set of propositional random variables
- E.g.,
  - $Sick(X)_{|C}$ with $C = (X, \{(alice), (eve)\})$
    - Represents $Sick(alice), Sick(eve)$
  - $Treat(X, M)_{|\top}$ with $\top = \left((X, M), dom(X) \times dom(M)\right)$
    - Represents
      $Treat(alice, m_1), Treat(alice, m_2),$
      $Treat(eve, m_1), Treat(eve, m_2),$
      $Treat(bob, m_1), Treat(bob, m_2)$

It may happen that a $C = (\mathcal{X}, C_{\mathcal{X}})$ talks about more logical variables $\mathcal{X}$ than the thing it applies to with logical variables $\mathcal{Y} \subset \mathcal{X}$
- To restrict C to $\mathcal{Y}$, we use the projection operation π from relational algebra:
  $$\text{if } \mathcal{Y} \subseteq \mathcal{X}, \pi_{\mathcal{Y}}(C) := \left(\mathcal{Y}, \pi_{\mathcal{Y}}(C_{\mathcal{X}})\right)$$

$Nat(D)$

$Acc(I)$

$Epid$

$Travel(X)$

$Treat(X, M)$
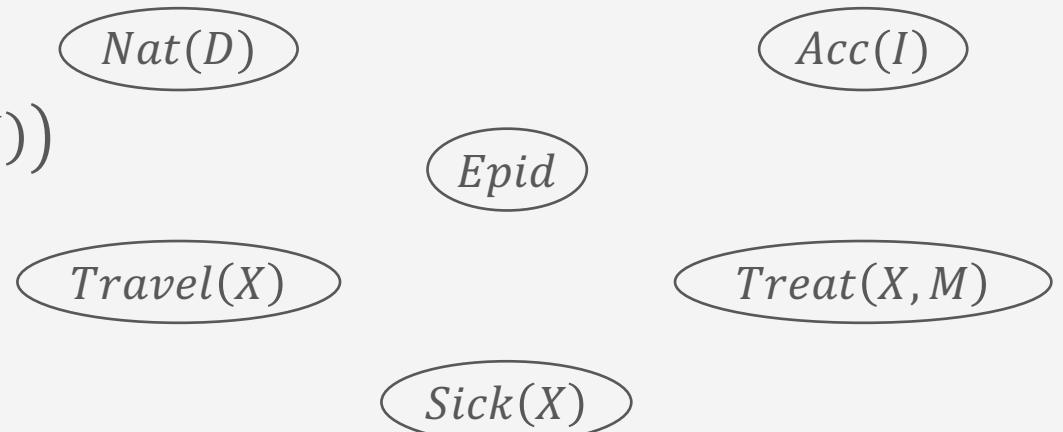
$Sick(X)$

Parfactors

- Factor with PRVs as arguments = parfactor
  $$g = \forall \boldsymbol{x} \in C_{\mathcal{X}} : \phi(\mathcal{A}\theta_{\boldsymbol{x}})_{|(\mathcal{X}, C_{\mathcal{X}})}$$
  - $\mathcal{A} = (A_1, \dots, A_k)$ sequence of PRVs containing logical variables $\mathcal{X}$
  - ϕ as before:
    $$\phi : \times_{i=1}^{k} ran(A_i) \longrightarrow \mathbb{R}^{0,+}$$
    - At least one potential > 0
  - Shorthand: $\phi(\mathcal{A})_{|C}$
    - If $C = \top : \phi(\mathcal{A})$
  - E.g., $g_2 =$
    $$\forall \boldsymbol{x} \in C_{\mathcal{X}} : \phi\big(Travel(x), Epid, Sick(x)\big)_{|((X),\{(alice),(eve),(bob)\})}$$
    $$= \phi\big(Travel(X), Epid, Sick(X)\big)_{|\top}$$
    $$= \phi\big(Travel(X), Epid, Sick(X)\big)$$

$Travel(X)$	$Epid$	$Sick(X)$	ϕ_2
false	false	false	5
false	false	true	0
false	true	false	4
false	true	true	6
true	false	false	4
true	false	true	6
true	true	false	2
true	true	true	9

Grounding of Parfactors

- Grounding of parfactor $gr(g)$
  - Obtain set of factors $\{f_i\}_{i=1}^{m}$ that g represents
  - Replace logical variables in g with constants in constraint of g:

$$gr\left(\phi(\mathcal{A})_{|(\mathcal{X}, C_{\mathcal{X}})}\right) = \bigcup_{\theta_x \in \boldsymbol{\theta}} \phi(\mathcal{A}\theta_x)$$

$$\boldsymbol{\theta} = \bigcup_{x \in C_{\mathcal{X}}} \left\{ \bigcup_{X \in \boldsymbol{x}} \{X \to x\} \right\}$$

  - $\boldsymbol{\theta}$ set of all possible substitutions θ_x to apply to $\mathcal{X}$ in $\mathcal{A}$
- E.g., $gr(g_2) = \{f_2^1, f_2^2, f_2^3\}$
  - $g_2 = \phi\big(Travel(X), Epid, Sick(X)\big)_{|T}$

$Travel(X)$	$Epid$	$Sick(X)$	ϕ_2
$false$	$false$	$false$	5
$false$	$false$	$true$	0
$false$	$true$	$false$	4
$false$	$true$	$true$	6
$true$	$false$	$false$	4
$true$	$false$	$true$	6
$true$	$true$	$false$	2
$true$	$true$	$true$	9

Grounding: Example

- E.g., $gr(g_2) = \{f_2^1, f_2^2, f_2^3\}$

Travel(eve)	Epid	Sick(eve)	ϕ_2^1
false	false	false	5
false	false	true	0
false	true	false	4
false	true	true	6
true	false	false	4
true	false	true	6
true	true	false	2
true	true	true	9

Epid	Sick(bob)	ϕ_2^3
false	false	5
false	true	0
true	false	4
true	true	6
false	false	4
false	true	6
true	false	2
true	true	9

Travel(alice)			
false			
false			
false			
false			
true			
true			
true			
true	true	true	9

=

Travel(X)	Epid	Sick(X)	ϕ_2
false	false	false	5
false	false	true	0
false	true	false	4
false	true	true	6
true	false	false	4
true	false	true	6
true	true	false	2
true	true	true	9

Symmetries Within

- Assume four epidemics with identical characteristics
  - $Epid_1, Epid_2, Epid_3, Epid_4$
  - Reasonable to model the epidemics such that it does not matter which $Epid$ variables specifically are $true$ or $false$, i.e., they are interchangeable

  $\#_{true}$ $\#_{false}$

$[0,4]$	8
$[1,3]$	6
$[2,2]$	4
$[3,1]$	2
$[4,0]$	0

  - All $false$ maps to 8
  - 1 $true$, 3 $false$ maps to 6
  - 2 $true$, 2 $false$ maps to 4
  - 3 $true$, 1 $false$ maps to 2
  - All $true$ maps to 0
    - → Five lines enough to describe

$Epid_1$	$Epid_2$	$Epid_3$	$Epid_4$	ϕ
$false$	$false$	$false$	$false$	8
$false$	$false$	$false$	$true$	6
$false$	$false$	$true$	$false$	6
$false$	$false$	$true$	$true$	4
$false$	$true$	$false$	$false$	6
$false$	$true$	$false$	$true$	4
$false$	$true$	$true$	$false$	4
$false$	$true$	$true$	$true$	2
$true$	$false$	$false$	$false$	6
$true$	$false$	$false$	$true$	4
$true$	$false$	$true$	$false$	4
$true$	$false$	$true$	$true$	2
$true$	$true$	$false$	$false$	4
$true$	$true$	$false$	$true$	2
$true$	$true$	$true$	$false$	2
$true$	$true$	$true$	$true$	0

Counting Random Variable

- New PRV type: (Parameterised) counting random variable ((P)CRV) $\#_X[A_{|C}]$
  - $A_{|C}$ a PRV under constraint C
  - $X \in lv(A)$
  - Range values: Histogram $h = \{(v_i, n_i)\}_{i=1}^m$
    - $m = |ran(A)|$ (number of buckets)
    - $n = \sum_{i=1}^m n_i = |gr(A_{|\pi_X(C)})|$ (number of instances to distribute into buckets)
    - $v_i \in ran(A)$ (buckets)
    - $n_i \in \mathbb{N}$ (number of instances in bucket v_i)
    - Shorthand: $[n_1, \ldots, n_m]$

- Range of a (P)CRV = space of histograms fulfilling the conditions on the histograms
  - (All possible ways of distributing n interchangeable instances into m buckets)
- Single histogram encodes several interchangeable assignments at once
  - Given by multinomial coefficient $Mul(h)$
  $$Mul(h) = \frac{n!}{\prod_{i=1}^m n_i!}$$
  - If $m = 2$, binomial coefficient:
  $$\binom{n}{n_1} = \frac{n!}{(n - n_1)! \, n_1!} = \frac{n!}{n_2! \, n_1!}$$

CRV: Example

- (P)CRV $\#_X[A_{|C}]$
  - Range values: Histogram $h = \{(v_i, n_i)\}_{i=1}^m$
    - $m = |ran(A)|$ (number of buckets)
    - $n = \sum_{i=1}^m n_i = |gr(A_{|C})|$ (number of instances to distribute into buckets)
    - $v_i \in ran(A)$ (buckets)
    - $n_i \in \mathbb{N}$ (number of instances in bucket v_i)
    - Shorthand: $[n_1, \dots, n_m]$
  - Single histogram encodes several interchangeable assignments at once:
    $$Mul(h) = \frac{n!}{\prod_{i=1}^m n_i!}$$

- E.g., CRV: $\#_E[Epid(E)]$
  - $ran(Epid(E)) = \{true, false\} \to m = 2$
  - $dom(E) = \{e_1, e_2, e_3, e_4\} \to n = 4$
  - Range values and multiplicities

$\{(true, 0), (false, 4)\} = [0,4] \quad Mul([0,4]) = \frac{4!}{0! \cdot 4!} = 1$

$\{(true, 1), (false, 3)\} = [1,3] \quad Mul([1,3]) = \frac{4!}{1! \cdot 3!} = 4$

$\{(true, 2), (false, 2)\} = [2,2] \quad Mul([2,2]) = \frac{4!}{2! \cdot 2!} = 6$

$\{(true, 3), (false, 1)\} = [3,1] \quad Mul([3,1]) = \frac{4!}{3! \cdot 1!} = 4$

$\{(true, 4), (false, 0)\} = [4,0] \quad Mul([4,0]) = \frac{4!}{4! \cdot 0!} = 1$

CRV: Example

- E.g., (continued)
  - CRV: $\#_E[Epid(E)]$
    - Range values
      $$[0,4], [1,3], [2,2], [3,1], [4,0]$$
      $$1 \quad 4 \quad 6 \quad 4 \quad 1$$
      how many assignments encoded
  - $g' = \phi(\#_E[Epid(E)])$

$\#_E[Epid(E)]$	ϕ'
[0,4]	8
[1,3]	6
[2,2]	4
[3,1]	2
[4,0]	0

$Epid_1$	$Epid_2$	$Epid_3$	$Epid_4$	ϕ
false	false	false	false	8
false	false	false	true	6
false	false	true	false	6
false	false	true	true	4
false	true	false	false	6
false	true	false	true	4
false	true	true	false	4
false	true	true	true	2
true	false	false	false	6
true	false	false	true	4
true	false	true	false	4
true	false	true	true	2
true	true	false	false	4
true	true	false	true	2
true	true	true	false	2
true	true	true	true	0

CRVs Continued

- (P)CRV $\#_X\left[A_{|C}\right]$ with
  - $m = |ran(A)|$ (number of buckets)
  - $n = \sum_{i=1}^{m} n_i = \left|gr\left(A_{|\pi_X(C)}\right)\right|$ (number of instances to distribute into buckets)

- Instead of m^n mappings in the ground factor, the counted factor has

$$\binom{n+m-1}{n-1}$$

Exponential in number of random variables n

mappings
  - Upper bound of range size of a CRV:
$$\binom{n+m-1}{n-1} \leq n^m$$
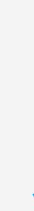
Polynomial in number of random variables n

CRVs Continued

- Counting binds a logical variable, i.e.,

$$lv\big(\#_X\big[A_{|C}\big]\big) = lv(A) \setminus \{X\}$$

- Constants that X represents made explicit through counts in histograms
- Compare the examples on the right
  - $\phi\big(Epid, Sick(X)\big)$ is a representative for the factors behind all groundings of X
  - $\phi'(\#_E[Epid(E)])$ already references all groundings of E in histograms

- Restriction:
  Only one logical variable can be counted at a time in a PRV!
  - In the current formalisation
    (it is possible to define more complex counting operations)

$Epid$	$Sick(X)$	ϕ
false	false	9
false	true	1
true	false	6
true	true	4

$\#_E[Epid(E)]$	ϕ'
[0,4]	8
[1,3]	6
[2,2]	4
[3,1]	2
[4,0]	0

Parfactors Revisited

- Factor with PRVs as arguments = parfactors
  $$g = \forall \boldsymbol{x} \in C_{\mathcal{X}} : \phi(\mathcal{A}\theta_{\boldsymbol{x}})_{|(\mathcal{X},C_{\mathcal{X}})}$$

  - $\mathcal{A} = (A_1, \dots, A_k)$ sequence of P(C)RVs containing logical variables $\mathcal{X}$

  - ϕ as before:
    $$\phi: \times_{i=1}^{k} ran(A_i) \longrightarrow \mathbb{R}^{0,+}$$

    > $\mathcal{A}$ may contain ground random variables, PRVs, CRVs, PCRVs

    - At least one potential > 0

  - Shorthand $\phi(\mathcal{A})_{|C}$

    - If $C = \top$: $\phi(\mathcal{A})$

  - E.g.,
    $$g_2 = \phi\big(Travel(X), Epid, Sick(X)\big)$$
    $$g' = \phi(\#_E[Epid(E)])$$
    $$g'' = \phi(Sick(X), \#_E[Epid(E)])$$

$Sick(X)$	$\#_E[Epid(E)]$	ϕ''
false	[0,4]	9
false	[1,3]	7
false	[2,2]	5
false	[3,1]	3
false	[4,0]	4
true	[0,4]	5
true	[1,3]	4
true	[2,2]	3
true	[3,1]	2
true	[4,0]	6

Parfactor Model

- Set of parfactors = model

$$G = \{g_i\}_{i=1}^n$$

  - Given the definitions of PRVs, it is possible to form a parfactor model that is completely propositional
    - If no logical variables used or domains of size 1
  - E.g., $G = \{g_i\}_{i=0}^3$
    - Plus full specifications of potential functions
    - $\mathbf{3 \cdot 2^3 + 2 = 26}$ **entries** independent of domain sizes!
    - Instead of $\mathbf{13 \cdot 2^3 + 2 = 106}$ **entries** with domains
      $dom(D) = \{fire, flood\}, |dom(D)| = 2$
      $dom(M) = \{m_1, m_2\}, |dom(M)| = 2$
      $dom(I) = \{chem, nucl\}, |dom(I)| = 2$
      $dom(X) = \{alice, eve, bob\}, |dom(D)| = 3$

$g_0 = \phi(Epid)$
$g_1 = \phi(Epid, Nat(D), Acc(I))$
$g_2 = \phi(Travel(X), Epid, Sick(X))$
$g_3 = \phi(Epid, Sick(X), Treat(X, M))$
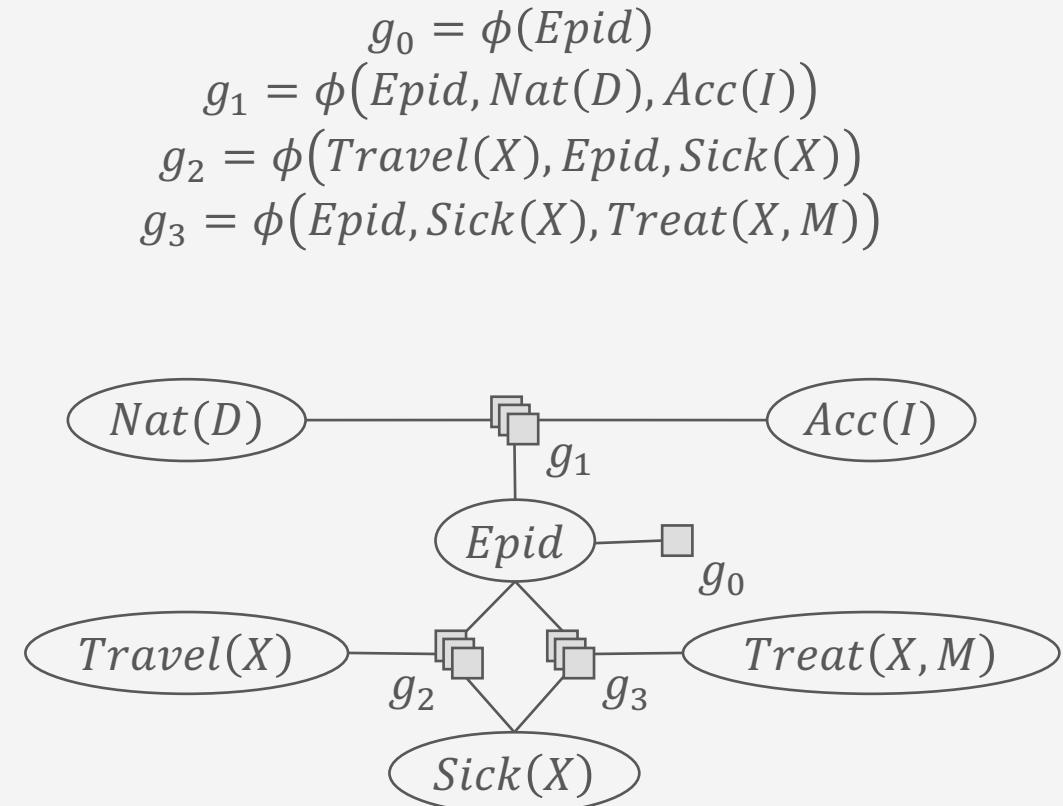
Parfactor Graphs

- Graphical representation of $G = \{g_i\}_{i=1}^n$:
  Parfactor graph
  - Analogous to factor graph
  - Each $A \in rv(G)$: variable node in graph (ellipse)
  - Each $g \in G$: factor node in graph (box)
    - Layered if $lv(g) \neq \emptyset$
  - For each argument A in $g \in G$: edge between variable node for A and factor node for g
  - Constraints are not depicted
  - Example
    - Six variable nodes
    - Four factor nodes, three of them layered

$$g_0 = \phi(Epid)$$
$$g_1 = \phi(Epid, Nat(D), Acc(I))$$
$$g_2 = \phi(Travel(X), Epid, Sick(X))$$
$$g_3 = \phi(Epid, Sick(X), Treat(X, M))$$
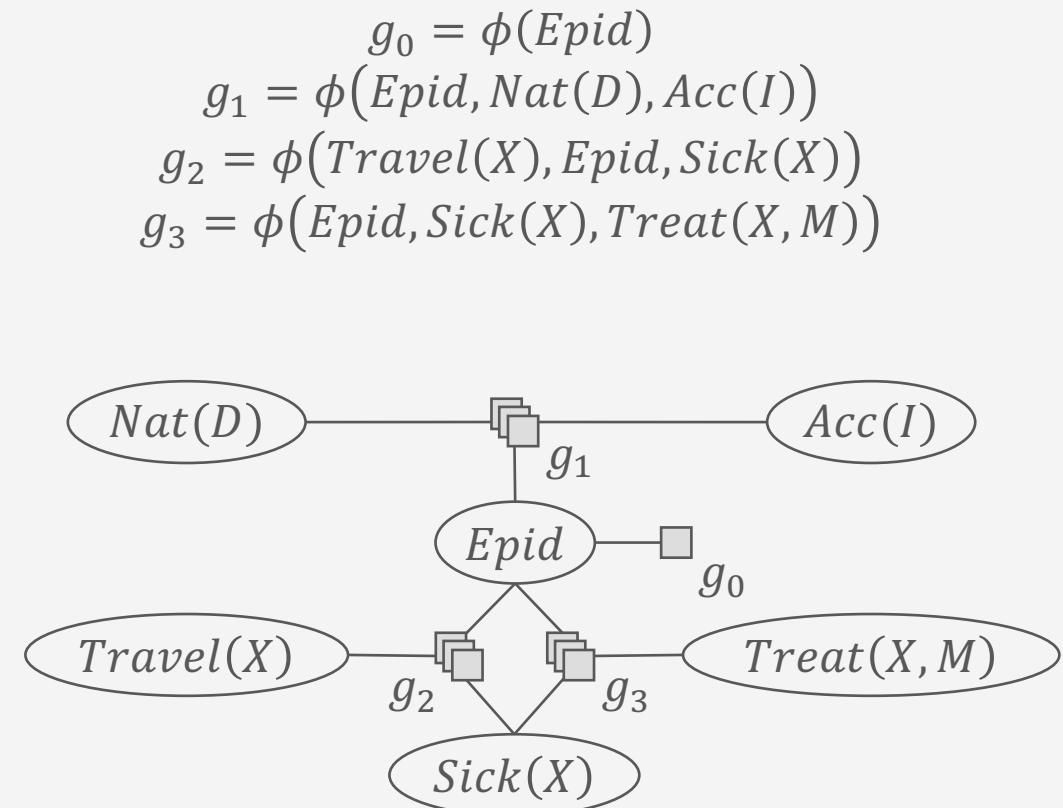
Grounding of Models

- Grounding of model G

$$gr(G) = \bigcup_{g \in G} gr(g)$$

- Depends on constraints and thus, domains
- E.g., example model with $\top$ constraints and domains
  - $dom(D) = \{fire, flood\}$
  - $dom(I) = \{chem, nucl\}$
  - $dom(M) = \{m_1, m_2\}$
  - $dom(X) = \{alice, eve, bob\}$

$Nat(D)$ —— g_1 —— $Acc(I)$

$Epid$ —— g_0

$Travel(X)$ —— g_2 —— g_3 —— $Treat(X, M)$

$Sick(X)$
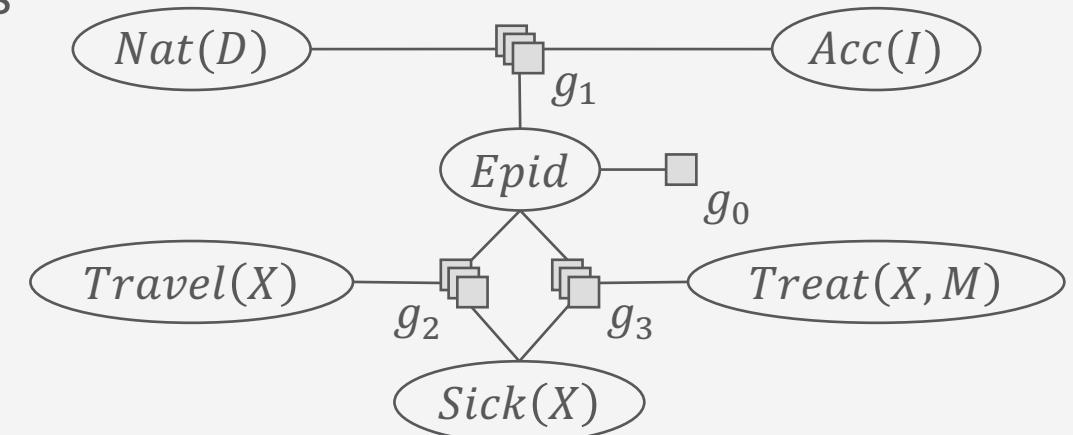
Grounding of Models

- Graphical representation of grounding
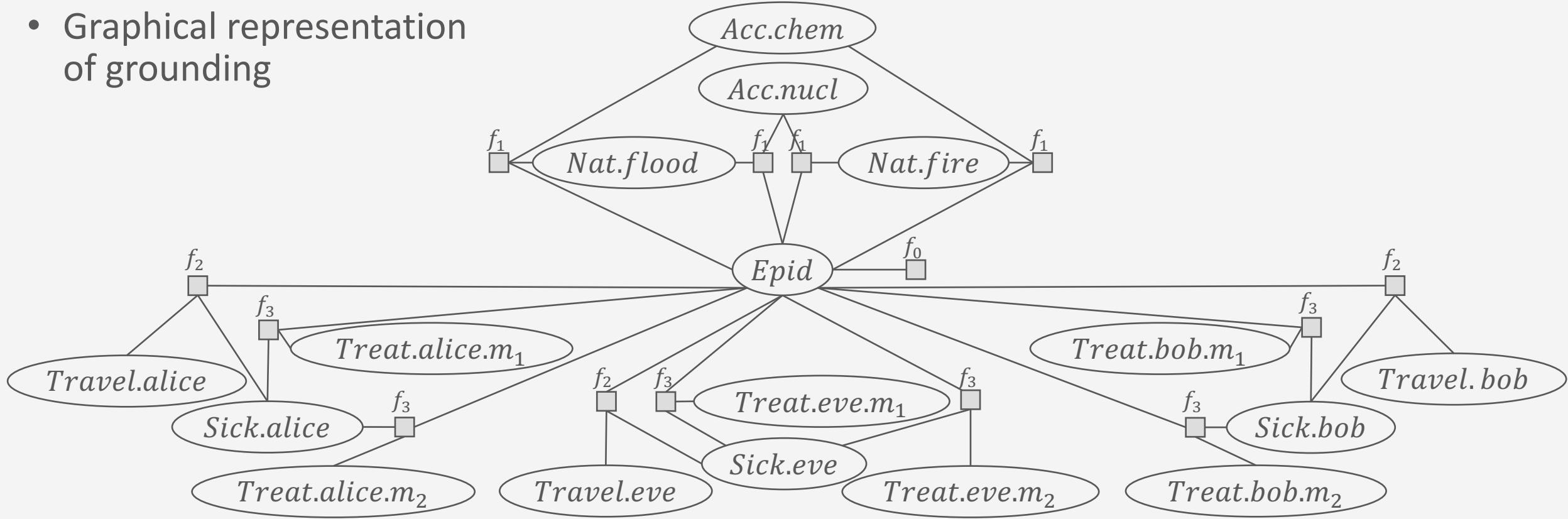
Semantics

- Given model $G = \{g_i\}_{i=1}^n$
  - Over PRVs $rv(G)$
    - Equivalent propositional model $F = gr(G)$ over random variables $gr(rv(G)) = \{R_1, \dots, R_N\}$ with semantics $P_{\boldsymbol{R}}$
- Semantics: Full joint probability distribution P_G

$$P_G = \frac{1}{Z} \prod_{f \in gr(G)} f$$

$$Z = \sum_{r_1 \in ran(R_1)} \dots \sum_{r_N \in ran(R_N)} \prod_{f \in gr(G)} f$$

- Equivalence over semantics:

$$P_G = P_F = P_{\boldsymbol{R}}$$
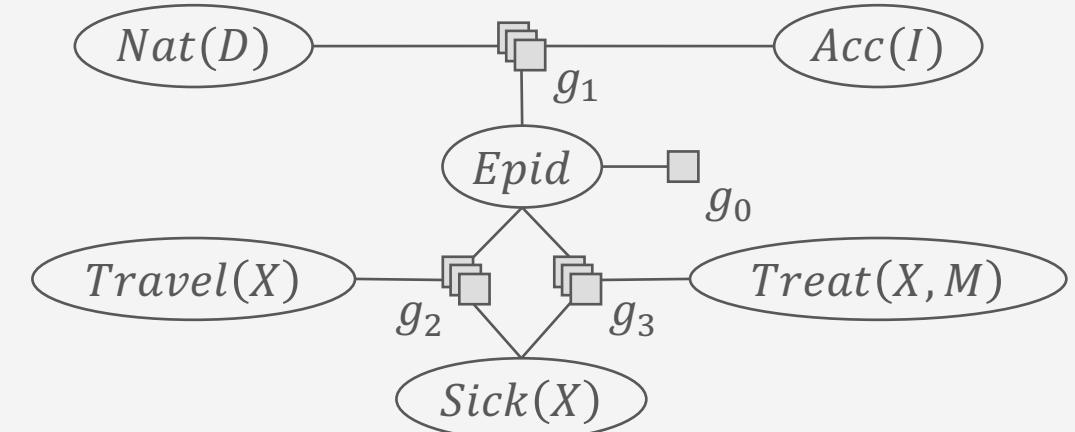
Inference Tasks

- Query Answering Problem (as before)
  - Compute an answer to a query $P(\boldsymbol{S}|\boldsymbol{T})$ given a model G representing the full joint probability distribution P_G
    - Avoid grounding (parts of) G
  - E.g.,
    - $P(Treat(eve, m_1))$
    - $P(Travel(eve), Epid)$
    - $P(Sick(eve)|Epid)$
    - $P(Epid|Sick(eve) = true)$
    - $P(\#_E[Epid(E)])$
    - $P(\#_E[Epid(E)] = [2,2])$

- New syntactical construct of a query: parameterised query $P(\boldsymbol{A}_{|C}|\boldsymbol{T})$
  - Represents $P(gr(\boldsymbol{A}_{|C})|\boldsymbol{T})$
  - E.g., $P(Sick(X)_{|\top})$
    - $P(Sick(eve), Sick(alice), Sick(bob))$
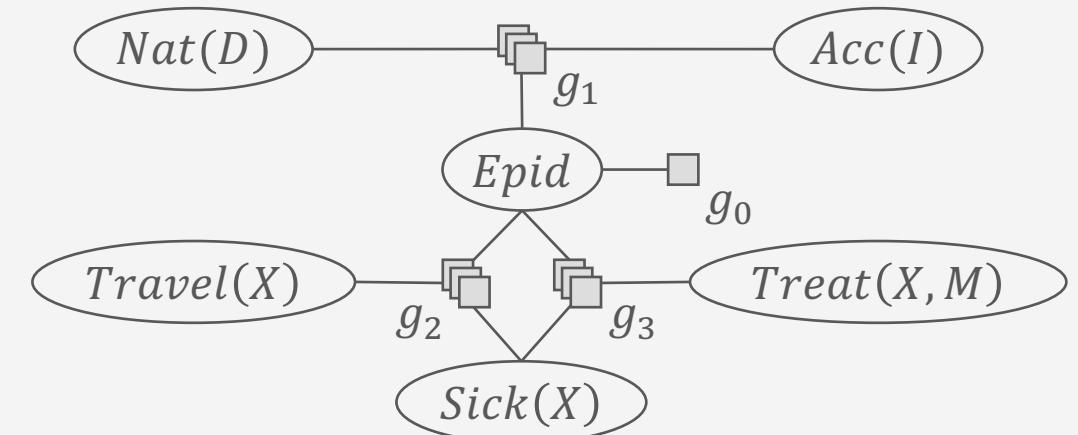
Evidence in Parameterised Models

- Observations for groundings of a PRV
  - Can be
    - One of the range values
    - Not available
  - E.g., for $Sick(X)$ with $dom(X) = \{x_1, \dots, x_n\}$
    - $Sick(x_1) = Sick(x_2) = \cdots = Sick(x_{10}) = true$
    - $Sick(x_{11}) = Sick(x_{12}) = \cdots = Sick(x_{20}) = false$
    - Observations for $Sick(x_{21}) \dots Sick(x_n)$ not available
- Compactly encode evidence with PRVs and parfactors
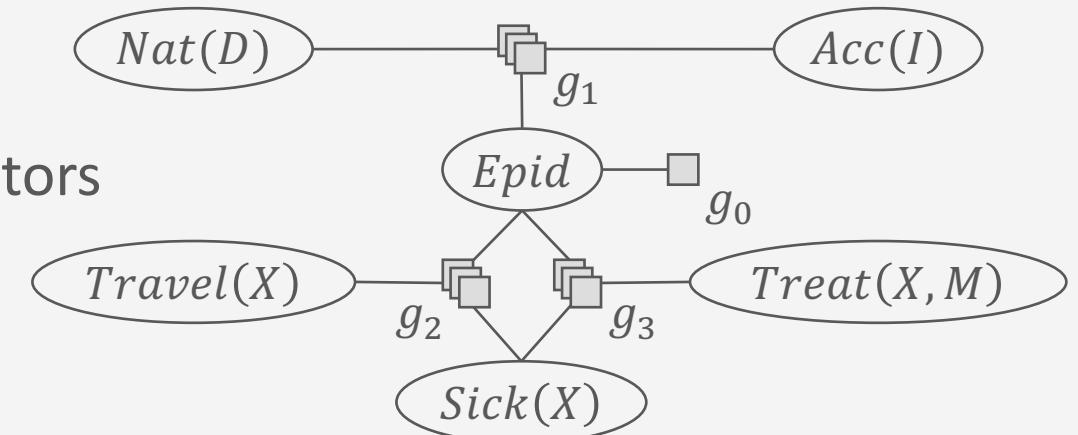
Evidence Parfactors: Example

- Observations of $Sick(X)$ with range value *true*
  - Parfactor $g_e^T = \phi_e^T\big(Sick(X)\big)_{|(X,C_X)}$
    - $C_X = \{x_1, \dots, x_{10}\}$
    - Or: Use new logical variable Y, $dom(Y) = \{x_1, \dots, x_{10}\}$ $\rightarrow \phi_e^T\big(Sick(Y)\big)_{|\top}$

$Sick(X)$	ϕ_e^T
false	0
true	1

- Observations of $Sick(X)$ with range value *false*
  - Parfactor $g_e^F = \phi_e^F\big(Sick(X)\big)_{|(X,C_X)}$
    - $C_X = \{x_{11}, \dots, x_{20}\}$
    - Or: Use new logical variable Z, $dom(Z) = \{x_{11}, \dots, x_{20}\}$ $\rightarrow \phi_e^F\big(Sick(Z)\big)_{|\top}$

$Sick(X)$	ϕ_e^F
false	1
true	0

Observations for $Sick(X)$

$Sick(x_1) = Sick(x_2) = \cdots = Sick(x_{10}) = true$
$Sick(x_{11}) = Sick(x_{12}) = \cdots = Sick(x_{20}) = false$

Graphical representation does not depict constraints
- g_2, g_3 have T constraints, g_e^T, g_e^F not
- g_e^T, g_e^F apply to a disjoint subset of constants occurring in g_2, g_3
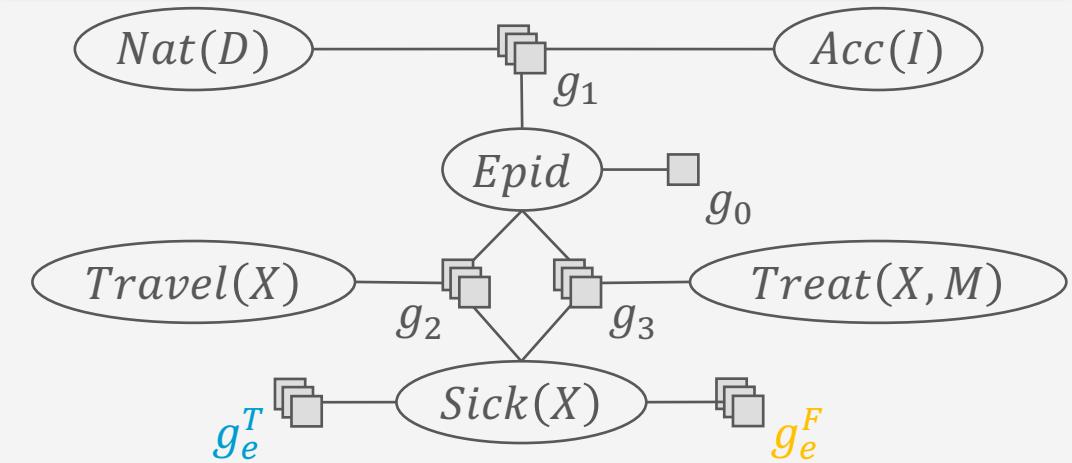
Evidence Parfactors

- Let e be the set of observations (evidence)
- For each PRV A referenced in $rv(e)$
  - For each range value r observed for a grounding of A
    - Set up a parfactor g_e with
      - A as argument
      - Range value r mapped to 1 and all range values $r' \neq r$ to 0
      - $C_\mathcal{X}$ in constraint $(\mathcal{X}, C_\mathcal{X})$ restricting the constants of $\mathcal{X}$ to the constants observed of A with r
      - Alternative: Use new logical variables $\mathcal{X}'$ in A with domains corresponding to observed constants and $\top$ constraints in g_e

Observations for $Sick(X)$
$$Sick(x_1) = Sick(x_2) = \cdots = Sick(x_{10}) = true$$
$$Sick(x_{11}) = Sick(x_{12}) = \cdots = Sick(x_{20}) = false$$
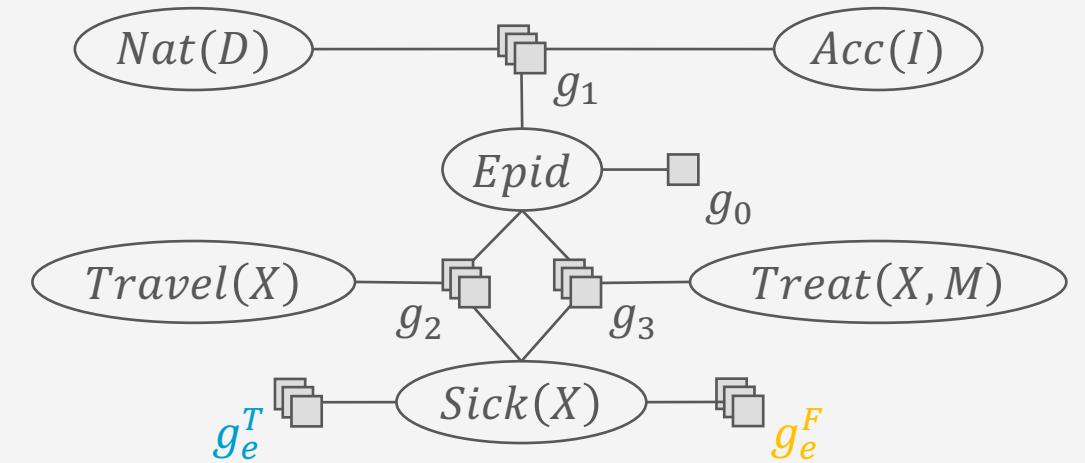
Interim Summary

- First-order view on probabilistic modelling
  - Relations in data
  - Symmetries in graph
- Parameterised models
  - Logical variables with domains of constants
    - Universe
  - Constraints to restrict domains to certain constants
  - PRVs to encode sets of propositional random variables
  - CRVs to encode symmetries within factors
  - Parfactors to build a model with recurring patterns
    - Semantics over grounding and full joint distribution
  - Inference tasks
    - Query terms with logical variables, evidence parfactors

Outline: 2. Probabilistic Relational Models (PRMs)

A. *Parfactor models (PMs)*
- Motivation: Symmetries and relations
- Syntax, semantics
- Graphical representation
- Inference tasks

B. *Markov logic networks (MLNs)*
- Syntax, semantics of MLNs
- Graphical representation
- Turning MLNs into PMs and vice versa

Markov Logic Networks (MLNs)

- Again: First-order view on probabilistic modelling
- Use logical formulas to specify potential functions
  - Weights for each formula induce a full joint again
    - Lower weight if constraint does not hold all the time
    - ∞ weight for hard constraints, in practice either a high weight or only worlds considered that satisfy hard constraints
- Next slides
  - Syntax of MLNs
  - Semantics of MLNs

Hard constraint

Soft constraint,
weight = exp(3.75)

$$\infty \; Presents(X, P, C) \Rightarrow Attends(X, C)$$

$$3.75 \; Publishes(X, C) \wedge FarAway(C) \Rightarrow Attends(X, C)$$

Matthew Richardson and Pedro Domingos: Markov Logic Networks. In: *Machine Learning*, 2006.

Markov Logic Networks (MLNs)

- Weighted formulas for modelling constraints ($\neq$ domain constraints)
- An MLN is a set of constraints $\Psi = \{(w_i, \psi_i)\}_{i=1}^{n}$
  - $w_i \in \mathbb{R}^+$ weight
  - ψ_i FOL formula
    - Originally without a constraint language for domains
    - Mostly used with implicit all-quantifiers
  - Implicitly connected via conjunction
    - I.e., set of formulas ψ_i = knowledge base/theory

Hard constraint

Soft constraint,
weight = exp(3.75)

$$\infty \; Presents(X, P, C) \Rightarrow Attends(X, C)$$

$$3.75 \; Publishes(X, C) \wedge FarAway(C) \Rightarrow Attends(X, C)$$

Intuition behind Weights

- Soften logic using weights
  - Worlds that violate constraint become less likely but not impossible
    - As w_i increases, so does the strength of ψ_i
    - Infinite weight: Hard constraint = pure logic formula
      - Probabilities of worlds that do not satisfy hard constraint set to 0
    - Standard MLNs only use weights in $\mathbb{R}$; otherwise semantics break (special handling of hard constraints required)
      - E.g., use hard constraints to filter worlds, probabilities of remaining worlds based on soft constraints

Hard constraint

Soft constraint, weight = exp(3.75)

$$\infty \; Presents(X, P, C) \Rightarrow Attends(X, C)$$

$$3.75 \; Publishes(X, C) \land FarAway(C) \Rightarrow Attends(X, C)$$

Grounding

- Each (w_i, ψ_i) represents a set of *propositional* sentences, each sentence with weight w_i
  - One sentence for each possible substitution of the free variables $free(\psi_i)$ in ψ_i given a finite domain (or a constraint set) D over $free(\psi_i)$
    - $\theta_D = \bigcup_{d \in D}\{\bigcup_{t \in d}\{X_d \rightarrow t\}\}$
  - Example: MLN $\Psi = \{(w_i, \psi_i)\}_{i=1}^2$
    - Domains
      - $dom(X) = \{alice\}$
      - $dom(P) = \{p_1, p_2\}$
      - $dom(C) = \{ijcai, ki\}$
    - Groundings on the right

$$\big(10, Presents(alice, p_1, ijcai) \Rightarrow Attends(alice, ijcai)\big)$$
$$\big(10, Presents(alice, p_1, ki) \Rightarrow Attends(alice, ki)\big)$$
$$\big(10, Presents(alice, p_2, ijcai) \Rightarrow Attends(alice, ijcai)\big)$$
$$\big(10, Presents(alice, p_2, ki) \Rightarrow Attends(alice, ki)\big)$$
$$\big(3.75, Publishes(alice, ijcai) \wedge FarAway(ijcai) \Rightarrow Attends(alice, ijcai)\big)$$
$$\big(3.75, Publishes(alice, ki) \wedge FarAway(ki) \Rightarrow Attends(alice, ki)\big)$$

$$10 \; Presents(X, P, C) \Rightarrow Attends(X, C)$$

$$3.75 \; Publishes(X, C) \wedge FarAway(C) \Rightarrow Attends(X, C)$$

MLN: Graphical Representation?

- Usually not depicted by a graph but by the logical formulas with their weights to the left

- Since the name invokes Markov networks, let us build an analogue:
  - Logical atoms as nodes
  - Edges between atoms whenever atoms occur together in a formula
    - Form cliques in graph
    - Potential functions (factors) for cliques from weights
    - Can also build an equivalent parfactor graph as graphical representation
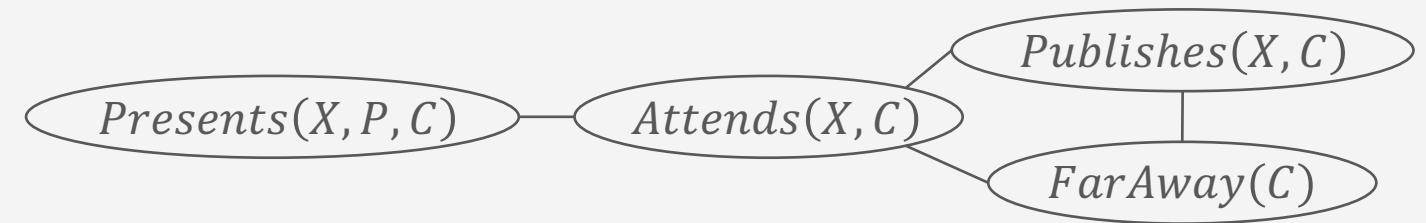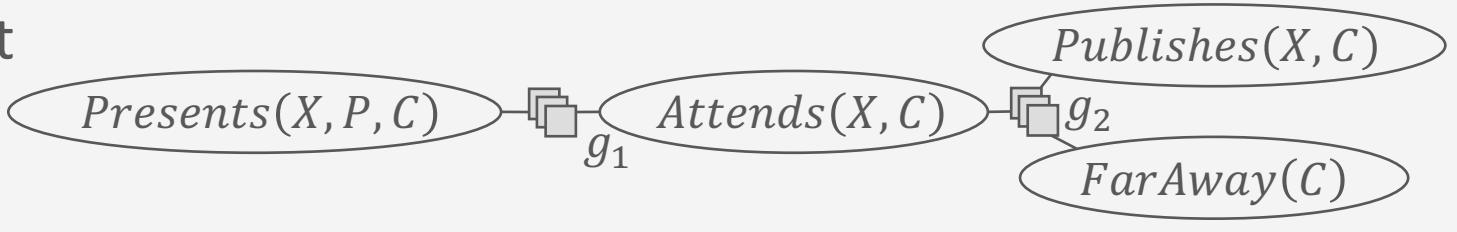
$Presents(X, P, C)$ —g_1— $Attends(X, C)$ —g_2— $Publishes(X, C)$
$FarAway(C)$

$Presents(X, P, C)$ — $Attends(X, C)$ — $Publishes(X, C)$ / $FarAway(C)$

$10 \; Presents(X, P, C) \Rightarrow Attends(X, C)$

$3.75 \; Publishes(X, C) \land FarAway(C) \Rightarrow Attends(X, C)$

Weights to Potential Functions (Factors)

- How to get from weights to potential functions (factors)?
  - Arguments = Atoms
    - Only Boolean ranges
  - Map input to $\exp(w_i)$ if input makes ψ_i true
    - Input = assignments to atoms in ψ_i
  - Otherwise map to $\exp(0) = 1$
  - If indeed $w_i = \infty$: choose large number
    - In implementation: maximum number of chosen number format
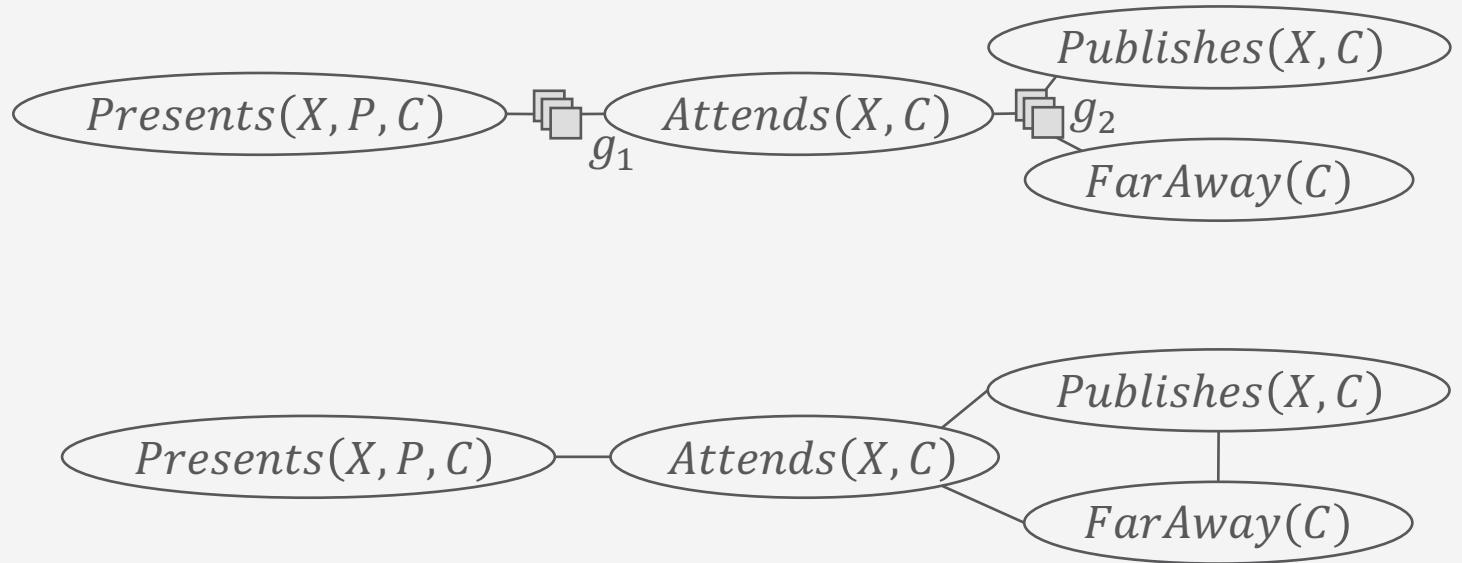
$Publishes(X, C)$

$Presents(X, P, C)$ — g_1 — $Attends(X, C)$ — g_2

$FarAway(C)$

$Publishes(X, C)$

$Presents(X, P, C)$ — $Attends(X, C)$

$FarAway(C)$

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

$3.75\ Publishes(X, C) \wedge FarAway(C) \Rightarrow Attends(X, C)$

Weights to Potential Functions (Factors) – Examples

| $Presents(X,P,C)$	$Attends(X,C)$	ϕ_1
$false$	$false$	$\exp 10$
$false$	$true$	$\exp 10$
$true$	$false$	$\exp 0$
$true$	$true$	$\exp 10$
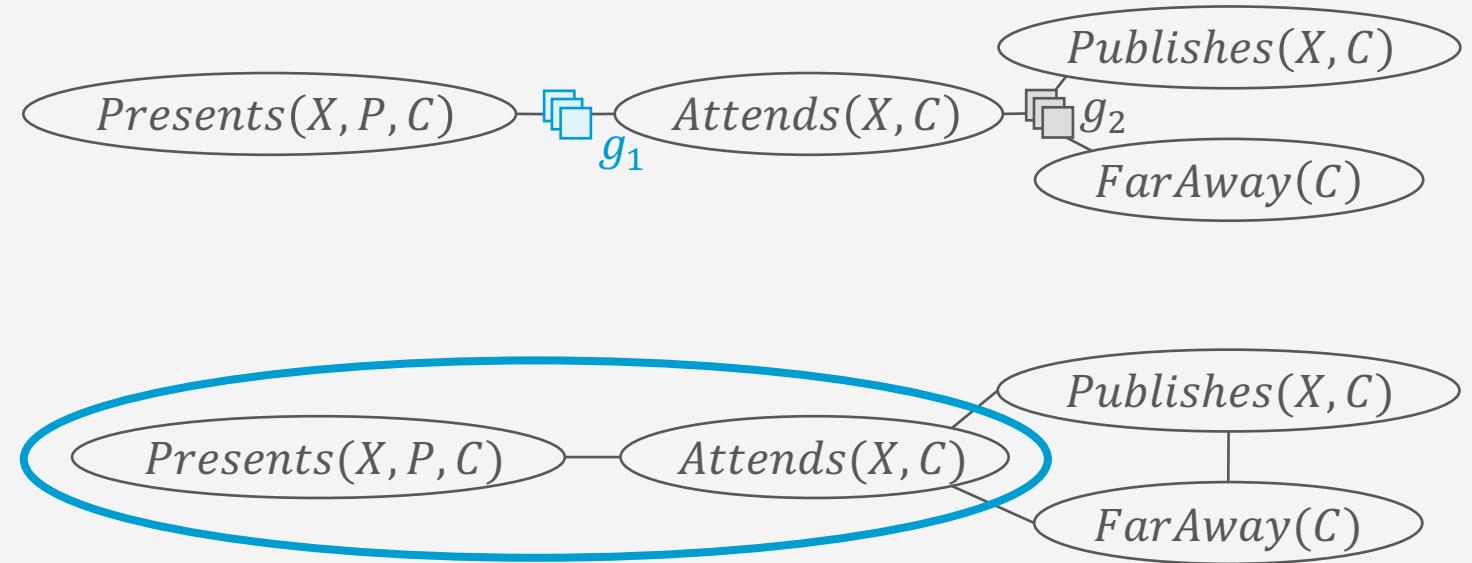
$$10\ Presents(X,P,C) \Rightarrow Attends(X,C)$$

$$3.75\ Publishes(X,C) \wedge FarAway(C) \Rightarrow Attends(X,C)$$

Weights to Potential Functions (Factors) – Examples

$Publishes(X,C)$	$FarAway(C)$	$Attends(X,C)$	ϕ
false	false	false	exp 3.75
false	false	true	exp 3.75
false	true	false	exp 3.75
false	true	true	exp 3.75
true	false	false	exp 3.75
true	false	true	exp 3.75
true	true	false	exp 0
true	true	true	exp 3.75

$X, P, C)$ — g_1 — $Attends(X,C)$ — g_2

$Publishes(X,C)$

$FarAway(C)$

$s(X,P,C)$ — $Attends(X,C)$ — $Publishes(X,C)$ — $FarAway(C)$
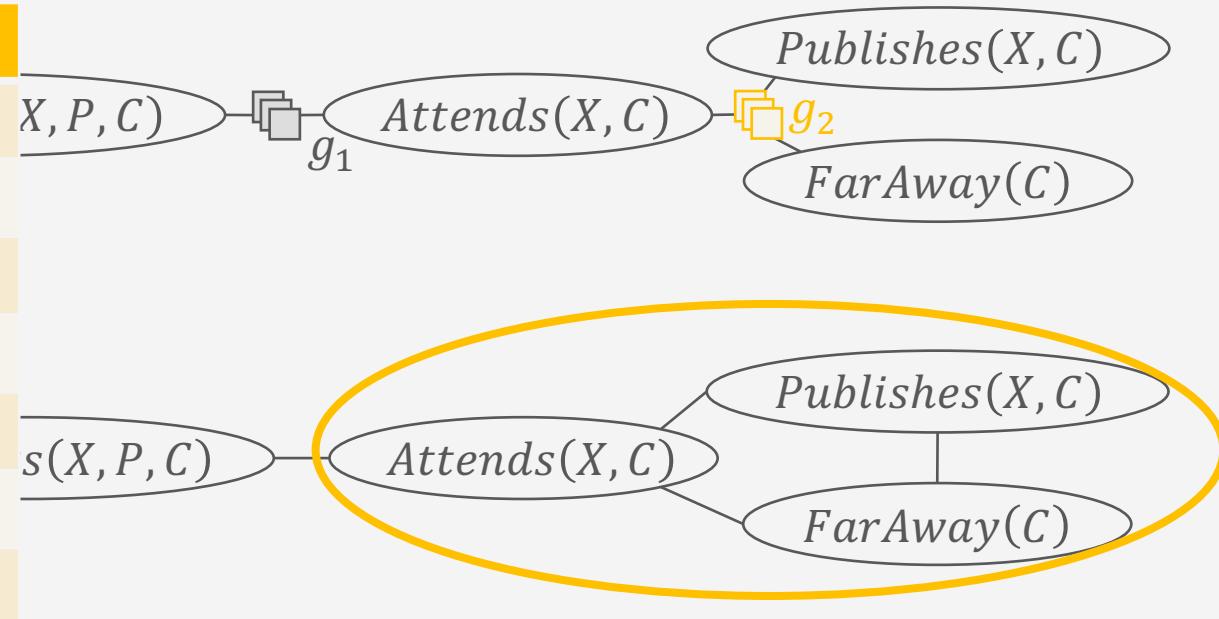
$10 \; Presents(X,P,C) \Rightarrow Attends(X,C)$

$3.75 \; Publishes(X,C) \wedge FarAway(C) \Rightarrow Attends(X,C)$

Graphical Representation for Groundings

- Each (w_i, ψ_i) represents a set of propositional sentences, each sentence with weight w_i
- Can also form a Markov network graphical representation (or factor graph) for this set of propositional sentences
  - Nodes = random variables = ground atoms
  - Edges connect literals that appear together in a sentence
  - Potential functions (as before, now with instances)
    - Map to $\exp(w_i)$ if assignment to ground atoms makes ψ_i true
   - Otherwise map to $\exp(0) = 1$
    - If indeed $w_i = \infty$: choose large number

$$10 \; Presents(X, P, C) \Rightarrow Attends(X, C)$$

$$3.75 \; Publishes(X, C) \wedge FarAway(C) \Rightarrow Attends(X, C)$$

Instances: Example – MN

- $\big(10, Presents(alice, p_1, ijcai) \Rightarrow Attends(alice, ijcai)\big)$
- $\big(10, Presents(alice, p_1, ki) \Rightarrow Attends(alice, ki)\big)$
- $\big(10, Presents(alice, p_2, ijcai) \Rightarrow Attends(alice, ijcai)\big)$
- $\big(10, Presents(alice, p_2, ki) \Rightarrow Attends(alice, ki)\big)$
- $\big(3.75, Publishes(alice, ijcai) \wedge FarAway(ijcai) \Rightarrow Attends(alice, ijcai)\big)$
- $\big(3.75, Publishes(alice, ki) \wedge FarAway(ki) \Rightarrow Attends(alice, ki)\big)$
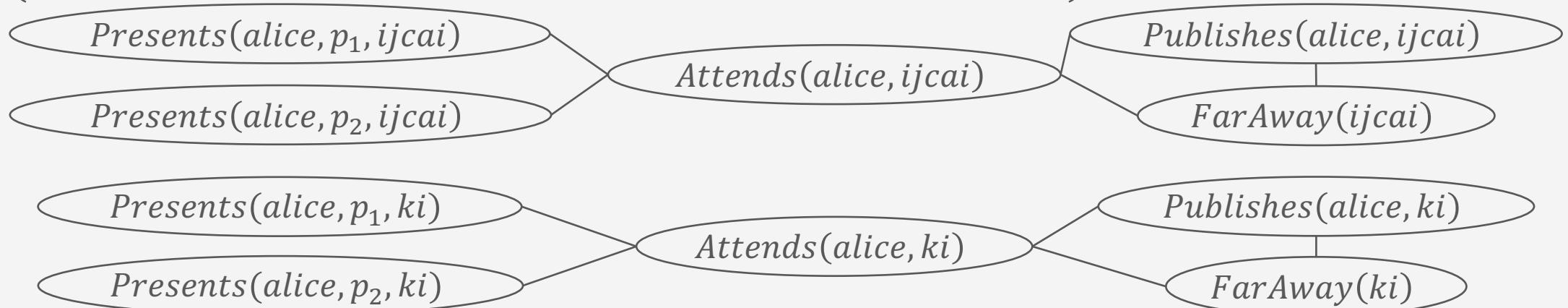
Example – Potential Functions

$Presents(x, p, c)$	$Attends(x, c)$	ϕ_1
false	false	exp 10
false	true	exp 10
true	false	exp 0
true	true	exp 10

$Publishes(x, c)$	$FarAway(c)$	$Attends(x, c)$	ϕ_2
false	false	false	exp 3.75
false	false	true	exp 3.75
false	true	false	exp 3.75
false	true	true	exp 3.75
true	false	false	exp 3.75
true	false	true	exp 3.75
true	true	false	exp 0
true	true	true	exp 3.75

Example – Factor Graph

$Presents(x,p,c)$	$Attends(x,c)$	ϕ_1
false	false	exp 10
false	true	exp 10
true	false	exp 0
true	true	exp 10

$Publishes(x,c)$	$FarAway(c)$	$Attends(x,c)$	ϕ_2
false	false	false	exp 3.75
false	false	true	exp 3.75
false	true	false	exp 3.75
false	true	true	exp 3.75
true	false	false	exp 3.75
true	false	true	exp 3.75
true	true	false	exp 0
true	true	true	exp 3.75

MLNs: Semantics

- MLN $\Psi = \{(w_i, \psi_i)\}_{i=1}^{n}$, with $w_i \in \mathbb{R}$, induces a probability distribution over possible interpretations ω (world) of the grounded atoms in Ψ

$$\omega \in \{true, false\}^N$$

  - N = the number of ground atoms in the grounded Ψ

  - Probability of one interpretation ω

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

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

MLNs: Semantics – Derivation

- Let an MLN $\Psi = \{(w_i, \psi_i)\}_{i=1}^{n}$ and a domain D be given
- The grounded MLN is the following

$$gr(\Psi) = \bigcup_{i=1}^{n} \bigcup_{\theta \in \theta_D} \{(w_i, \psi_i \theta)\}$$

- The semantics of $gr(\Psi)$ induces a probability distribution over possible interpretations ω
  - ω assigns a truth value to each (ground) atom in $gr(\Psi)$
  - I.e., a normalised product over all weights of formulas (potential functions of cliques) with $\exp w_j$ if ω makes $\psi_j \theta$ true and $\exp 0$ otherwise, i.e.,

$$P(\omega) = \frac{1}{Z} \prod_{j=1}^{|gr(\Psi)|} \exp(w_j) = \frac{1}{Z} weight(\omega)$$

  - Product called weight of a world

MLNs: Semantics – Derivation

- Consider the weight of ω

$$weight(\omega) = \prod_{j=1}^{|gr(\Psi)|} \exp(w_j)$$

- To simplify, use the following:
  - Sentences in $gr(\Psi)$ are groundings of n formulas where each set of propositional sentences per formula carries the same weight
  - If a propositional sentences evaluates to $false$, its contribution to the product is $\exp(0) = 1$.
  - → Rewrite expression:

$$weight(\omega) = \prod_{i=1}^{n} \exp(w_i)^{n_i(\omega)} = \prod_{i=1}^{n} \exp(w_i \cdot n_i(\omega))$$

  - $n_i(\omega)$ = number of propositional sentences of ψ_i that evaluate to $true$ given ω

MLNs: Semantics – Derivation

- Therefore, $\Psi = \{(w_i, \psi_i)\}_{i=1}^{n}$ induces a probability distribution over possible worlds ω

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

- $Z = \sum_{\omega \in \{true, false\}^N} weight(\omega)$
  - N = the number of ground atoms in $gr(\Psi)$

Why exp?

- Weight of a world ω:

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

- Taking the logarithm yields

$$lweight(\omega) = \ln \exp\left(\sum_{i=1}^{n} w_i n_i(\omega)\right) = \sum_{i=1}^{n} w_i n_i(\omega)$$

- Sum allows for component-wise optimisation during weight learning
- Referred to as log-linear models
- Semantics:

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

- $Z' = \sum_{\omega \in \{true, false\}^N} lweight(\omega) = \sum_{\omega \in \{true, false\}^N} \sum_{i=1}^{n} w_i n_i(\omega)$

Transforming MLNs into PMs

- Follows the same idea of generating a graphical representation
  - Logical atoms = PRVs $\mathcal{A}$
  - Potential function
    - Map a to $\exp(w_i)$ if truth values in a as assignments to atoms makes ψ_i true
    - Otherwise map to $\exp(0) = 1$
    - If indeed $w_i = \infty$: choose large number
      - In implementation: maximum number of chosen number encoding
  - Constraints
    - If no domain constraints given: $(\mathcal{X}, C_{\mathcal{X}}) = \top$, i.e.,
      $$\mathcal{X} = lv(\mathcal{A}), C_{\mathcal{X}} = \times_{X \in \mathcal{X}} dom(X)$$
    - Otherwise: build constraint according to domain constraints

$10 \; Presents(X, P, C) \Rightarrow Attends(X, C)$

$3.75 \; Publishes(X, C) \wedge FarAway(C) \Rightarrow Attends(X, C)$

Transforming MLNs into PMs: Example

- E.g.,
  - $g_1 = \phi\big(Presents(X, P, C), Attends(X, C)\big)_{|C_1}$
    - $C_1 = \big((X, P, C), dom(X) \times dom(P) \times dom(C)\big)$

$Presents(X, P, C)$	$Attends(X, C)$	ϕ_1
$false$	$false$	$\exp 10$
$false$	$true$	$\exp 10$
$true$	$false$	$\exp 0$
$true$	$true$	$\exp 10$

$10 \; Presents(X, P, C) \Rightarrow Attends(X, C)$

$3.75 \; Publishes(X, C) \wedge FarAway(C) \Rightarrow Attends(X, C)$
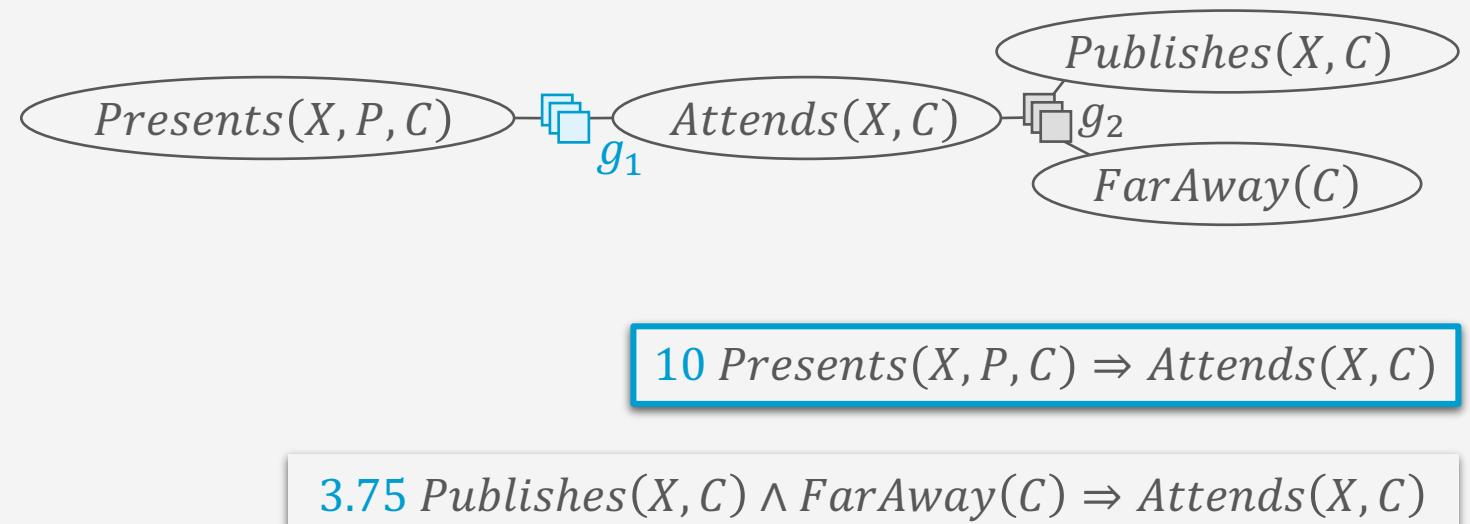
Transforming MLNs into PMs: Example

- $g_2 = \phi\big(Publishes(X, C), FarAway(C), Attends(X, C)\big)_{|C_2}$
  - $C_2 = \big((X, C), dom(X) \times dom\mathcal{D}(C)\big)$

$Publishes(X, C)$	$FarAway(C)$	$Attends(X, C)$	ϕ
false	false	false	exp 3.75
false	false	true	exp 3.75
false	true	false	exp 3.75
false	true	true	exp 3.75
true	false	false	exp 3.75
true	false	true	exp 3.75
true	true	false	exp 0
true	true	true	exp 3.75

(X, P, C) — g_1 — $Attends(X, C)$ — g_2

$Publishes(X, C)$

$FarAway(C)$

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

$3.75\ Publishes(X, C) \wedge FarAway(C) \Rightarrow Attends(X, C)$

Transforming PMs into MLNs

- Inverse of Transforming MLNs into PMs
- Given $G = \left\{ \phi_i(\mathcal{A}_i)_{|C_i} \right\}_{i=1}^{n}$
  - Precondition: All $A \in rv(G)$ need to have Boolean ranges
  - PRVs $\mathcal{A}$ = logical atoms
  - Constraint C_i
    - If $C_i = \top$: original MLN without further restrictions
    - Otherwise: encode in constraint set per formula

- Potential function ϕ_i
  - For each input-output mapping $a \to p$
    - Build formula $\psi = \Lambda_{A \in \mathcal{A}} \, t(A, a)$ with weight $w = \ln p$
      - $t(A, a) = \begin{cases} a & \text{Assignment of } A \text{ in } a \text{ is } true \\ \neg a & \text{Assignment of } A \text{ in } a \text{ is } false \end{cases}$
    - If the potentials are in $[0,1]$, then w is negative
    - If potential is 0, choose large negative number
    - Compare to the inverse with a weight of infinity
    - Best: ensure that potentials are in either $(0,1]$ or $\mathbb{R}_{\geq 1}$

Inverse Example

- E.g., $G = \{g_i\}_{i=0}^3$
  - $g_1 = \phi(Epid)$
  - $g_1 = \phi\big(Epid, Nat(D), Acc(I)\big)$
  - $g_2 = \phi\big(Travel(X), Epid, Sick(X)\big)$
  - $g_3 = \phi\big(Epid, Sick(X), Treat(X, M)\big)$
  - Logical atoms:
    $$Epid, Nat(D), Acc(I),$$
    $$Travel(X), Sick(X),$$
    $$Treat(X, M)$$
  - Domains as given for G
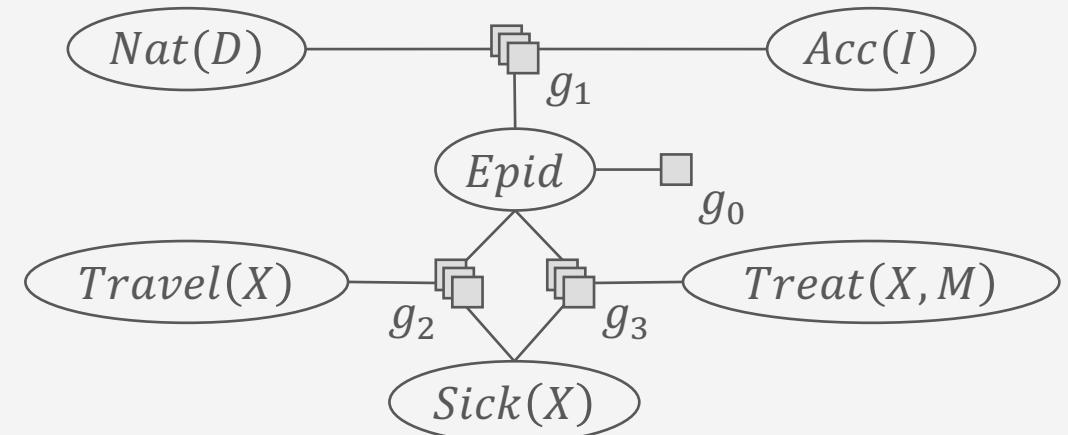
Inverse Example

- Weighted formulas (w_i, ψ_i) for g_2
  - $\left(\ln 5, \neg travel(X) \wedge \neg epid \wedge \neg sick(X) \right)$
  - $\left(\ln 1, \neg travel(X) \wedge \neg epid \wedge sick(X) \right)$
  - $\left(\ln 4, \neg travel(X) \wedge epid \wedge \neg sick(X) \right)$
  - $\left(\ln 6, \neg travel(X) \wedge epid \wedge sick(X) \right)$
  - $\left(\ln 4, travel(X) \wedge \neg epid \wedge \neg sick(X) \right)$
  - $\left(\ln 6, travel(X) \wedge \neg epid \wedge sick(X) \right)$
  - $\left(\ln 2, travel(X) \wedge epid \wedge \neg sick(X) \right)$
  - $\left(\ln 9, travel(X) \wedge epid \wedge sick(X) \right)$
- Same has to be done for g_0, g_1, g_3

$Travel(X)$	$Epid$	$Sick(X)$	ϕ_2
$false$	$false$	$false$	5
$false$	$false$	$true$	1
$false$	$true$	$false$	4
$false$	$true$	$true$	6
$true$	$false$	$false$	4
$true$	$false$	$true$	6
$true$	$true$	$false$	2
$true$	$true$	$true$	9

If some potentials appear multiple times, use an algorithm such as Quine-McCluskey to minimise formulas needed to encode the same information [Marwitz et al. (2021)]

Florian Marwitz, Tanya B, and Ralf Möller: A First Step Towards Even More Sparse Encodings of Probability Distributions. In *ILP-21 Proceedings of the 30th International Conference on Inductive Logic Programming*, 2021.

Inverse Example

- Consider $\phi\big(Travel(X), Epid, Sick(X)\big)$ as given by the table on the right
- Only two weighted formulas (w_i, ψ_i) necessary
  - $\big(\ln 2, \neg travel(X) \lor \neg epid \lor \neg sick(X)\big)$
  - $\big(\ln 7, travel(X) \land epid \land sick(X)\big)$
  - If potential of 1 instead of 2, would reduce to
    - $\big(\ln 7, travel(X) \land epid \land sick(X)\big)$
    - Assignments that do not make the formula true automatically get weight of $0 = \ln 1$

$Travel(X)$	$Epid$	$Sick(X)$	ϕ
false	false	false	2
false	false	true	2
false	true	false	2
false	true	true	2
true	false	false	2
true	false	true	2
true	true	false	2
true	true	true	7

Size of resulting MLN depends on the *local symmetries* in the potentials!

MLNs vs. PMs

- MLNs: soft constraints in FOL-DC
  - Requires knowing interactions to build formulas with logical connectives
  - Allows humans to interpret a knowledge base more easily
  - Space-efficient encoding *if* different truth value assignments have the same weight *and* fewer rules can encode them
- PMs are more blurred
  - Interaction described via distributions
    - Does not enforce connectives between variables
    - But: Interpretation not that easy

- In propositional case
  - MNs and FGs are different graphical representations of the same factor-based model
- In relational case
  - MLNs and PMs are more different
    - Semantics still the same: full joint
    - Can transform one into the other → same expressivity
    - But the syntax is considered differently
- Possible to start with PMs
  - Extract rules if necessary

Interim Summary

- MLNs
  - Weighted formulas
  - Semantics again over groundings and full joint distribution
  - Log-linear version allows for local optimisation during learning
- Transformations between MLNs and PMs
  - One weighted rule → one potential function
  - One potential function → possibly # of input-output mappings different rules
  - If local symmetries exist (in potentials or weights), then MLNs offer compact encoding

Other Formalisms*

- Relational BNs
  - Nodes in BNs are n-ary relations, labelled with so-called probability formulas, which define the semantics as a probability measure over interpretations of the relations
  - Implementation: http://people.cs.aau.dk/~jaeger/Primula/

- Hinge-loss MNs
  - Continuous variables in the $[0,1]$ unit interval combined in constraints of first-order logic syntax
    - Distance of each constraint to satisfaction = hinge loss
  - Implemented with Probabilistic Soft Logic https://psl.linqs.org

- ProbLog: Probabilistic Prolog (probabilistic logic programming)
  - Defines a probability distribution over logic programs
  - Typically Horn clauses annotated with probabilities
  - Implementation: https://dtai.cs.kuleuven.be/problog/

> * Relations not necessarily used for efficient inference (point of the next lectures)

Manfred Jaeger: Relational Bayesian Networks. In: *UAI-97 Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence*, 1997.
Stephen H. Bach, Matthias Broecheler, Bert Huang, and Lise Getoor: Hinge-loss Markov Random Fields and Probabilistic Soft Logic. In: *Journal of Machine Learning Research*, 2017
Luc De Raedt, Angelika Kimmig, and Hannu Toivonen: ProbLog: A Probabilistic Prolog and its Application in Link Discovery. In: IJCAI-07 Proceedings of the 20th International Joint Conference on Artificial Intelligence, 2007.

Outline: 2. Probabilistic Relational Models (PRMs)

A. *Parfactor models (PMs)*
- Motivation: Symmetries and relations
- Syntax, semantics
- Graphical representation
- Inference tasks

B. *Markov logic networks (MLNs)*
- Syntax, semantics of MLNs
- Graphical representation
- Turning MLNs into PMs and vice versa

→ Lifted Inference