Lifted Inference in Probabilistic Logical Models



Kristian Kersting

Fraunhofer IAIS & Univ. of Bonn, Germany



Sriraam Natarajan

Wake Forest University, USA



David Poole

Univ. of British Columbia, Canada

IJCAI-11

TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE

JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

Disclaimer


IJCAI-11
TWENTY-SECOND INTERNATIONAL
JOINT CONFERENCE
ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

[Hermann Rorschach (* Nov 8, 1884; † April 2 1922)]

Rorschach Test



IJCAI-11
TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

Etzioni's Rorschach Test for Computer Scientists





IJCAI-11
TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

Moore's Law?



IJCAI-11
TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

Storage Capacity?



IJCAI-11
TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

Number of Web Pages?



IJCAI-11
TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

Number of Facebook Users?



IJCAI-11
TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

Number of Scientific Publications?



IJCAI-11
TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

[Weikum, Kasneci, Ramanath, Suchanek Commun. ACM 52(4): 56-64 ('09)]

The World-Wide Mind



TextRunner Search http://www.cs.washington.edu/research/textrunner/

Object **Relation** **Uncertainty** **Object**

TextRunner took 3 seconds.

Retrieved **256** results for **paper** in argument 1 and **topic** in argument 2.

Grouping results by argument 1. Group by: predicate | argument 2

paper - 81 results

paper discusses (65), covers (54), addresses (51), *69 more...* the **topic**

paper discusses (34), covers (30), contains (7), *6 more...* the following **topics**

paper focuses on (9), discusses (5), addresses (5), *6 more...* two **topics**

paper will cover (5), explores (2) a variety of **topics**

Paper presented at (7) the Theme issue **topi**

Paper presented at (7) the Special **topic**

white **paper** provides (6) a high-level overvie

paper addresses (5) the **topic** of World Ban

paper describes (3), recommends (2) the sp

Search again:

Argument 1

paper

Predicate

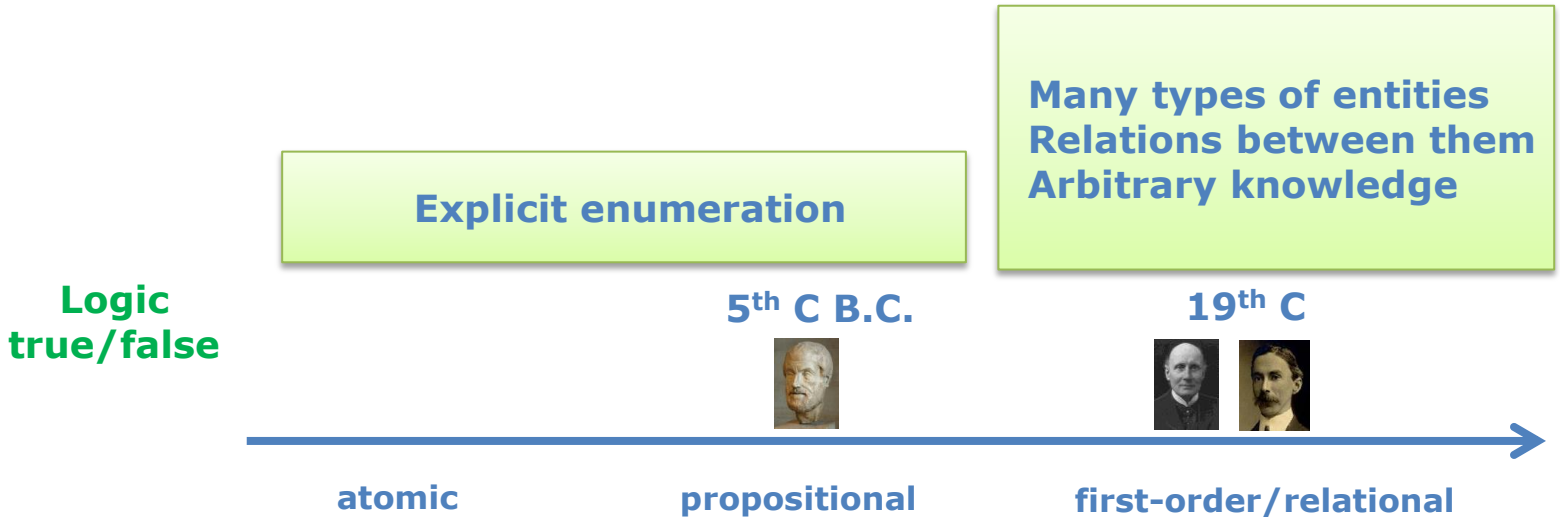
So, how do computer systems deal with uncertainty, objects, and relations? How do we realize the vision of a world-wide-mind?

No complex inference (yet) !
TextRunner: (Turing, born in, London)
+ WordNet: (London, part of, England)
+ Rule: 'born in' is transitive thru 'part of'
Conclusion: (Turing, born in, *England*)

TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011 CCIB BARCELONA, CATALONIA (SPAIN) WWW.IJCAI-11.ORG
IJCAI-11

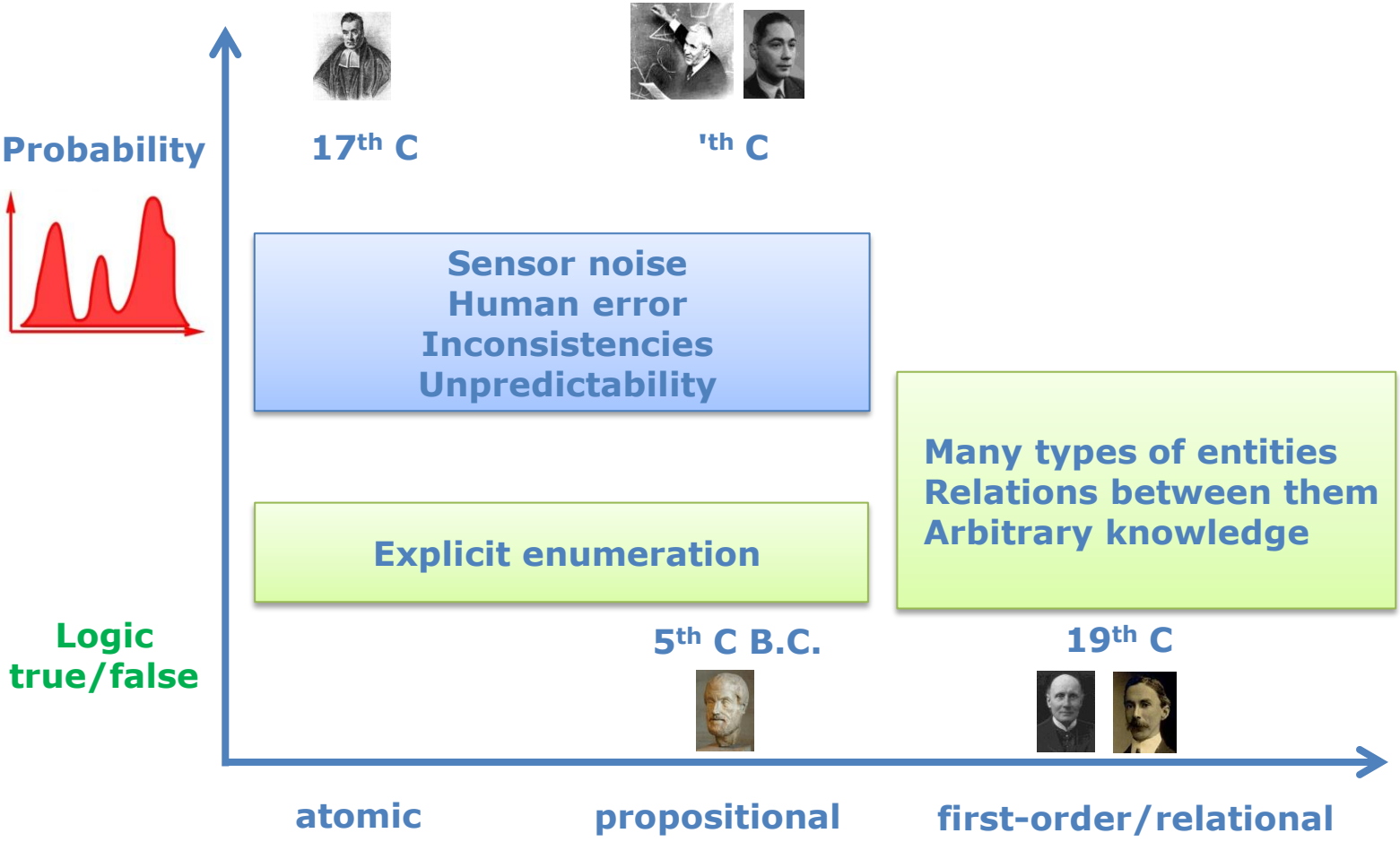
(First-order) Logic handles Complexity

E.g., description complexity to encode the rules of chess: first-order logic << propositional logic<< atomic-state model



TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

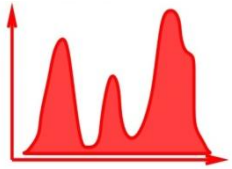
Probability handles Uncertainty



TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG
IJCAI-11

So, will traditional AI scale ? No !

"Scaling up the environment will inevitably overtax the resources of the current AI architecture."



Probability



17th C



'th C

MIND THE GAP

Sensor noise
Human error
Inconsistencies
Unpredictability

Explicit enumeration

Many types of entities
Relations between them
Arbitrary knowledge

Logic
true/false

5th C B.C.



19th C





atomic

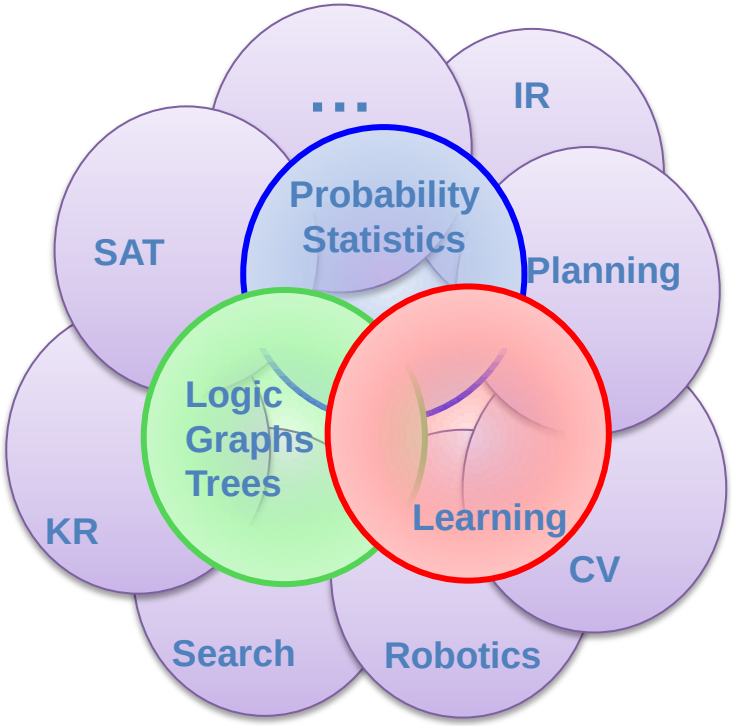
propositional

first-order/relational

IJCAI-11
TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

Statistical Relational AI (StarAI)

Let's deal with **uncertainty**, **objects**, **relations**, and **learning** jointly



The study and design of intelligent agents that act in noisy worlds composed of objects and relations among the objects

IJCAI-11
TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

Pros of StarAI

Relations can reveal additional correlations. Abstraction allows for generalization and compactness

Statistical relational techniques have the potential to lay the foundations of next generation AI systems



+++

- Better performance
- Better understanding of domains
- Growth path for machine learning and artificial intelligence

TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG
IJCAI-11

For example, we can …

- … learn probabilistic relational models automatically from millions of inter-related objects
- … generate optimal plans and learn to act optimally in uncertain environments involving millions of objects and relations among them

IJCAI-11
TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

Pros and Cons of StarAI

Relations can reveal additional correlations. Abstraction allows for generalization and compactness

Statistical relational techniques have the potential to lay the foundations of next generation AI systems



Yes, StarAI is challenging but knowing one of its ingredients is half the battle

+++

- Better performance
- Better understanding of domains
- Growth path for machine learning and artificial intelligence

---

- Learning is much harder
- Inference becomes a crucial issue
- Greater complexity for user

Kristian Kersting, Sriraam Natarajan, David Poole
Lifted Inference in Probabilistic Logical Models

IJCAI-11
TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

So, can we do better?

- Inference in first-order logic is not „ground“ but lifted
  - handling whole sets of indistinguishable objects together



Yes, we can and it is exactly the focus of this tutorial

Resulting lifted approaches are often faster, more compact and provide more structure for optimization

IJCAI-11
TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

So, What is Lifted Inference ?



- An inference algorithm that deals with "*groups*" of random variables at a first-order level
  - Takes a general first-order model as input
  - Automatically answers queries without computational waste
- Reason over a large domains in time independent of the number of objects
- Ability to carry out probabilistic inference in a relational probabilistic model without needing to reason about each individual separately

TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011 CCIB BARCELONA, CATALONIA (SPAIN) WWW.IJCAI-11.ORG
IJCAI-11

Or, What is Lifted Inference?



- Algorithm that exploits interchangeability in the domain
- Queries are answered without instantiating all the objects in the domain
- Exploit shared correlations – Same uncertainties and correlations repeatedly occur in data
- Exploit symmetries in the data and the model
- **Group random variables that send an receive the same messages

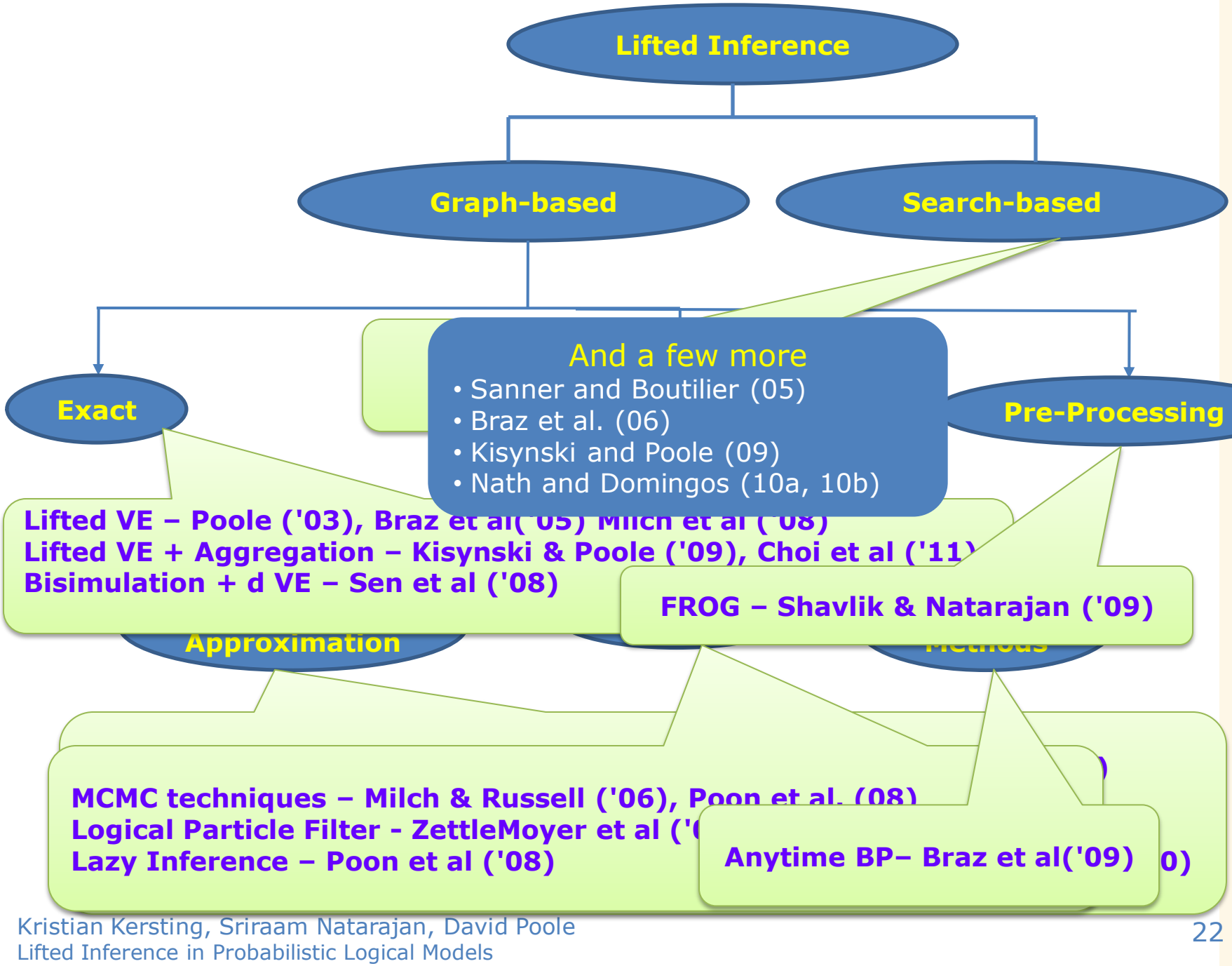
TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

Different definitions – Similar ideas

- Key Ingredients
  - Exploit symmetry (Identify similar sub-groups)
  - Reason as much as possible at a high-level
  - Independent of the number of objects

- This tutorial
  - Algorithms presented by the gurus
  - Unbiased view as a series of "biased" views

IJCAI-11
TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG


Lifted Inference
Graph-based
Search-based
Exact
Pre-Processing
And a few more
• Sanner and Boutilier (05)
• Braz et al. (06)
• Kisynski and Poole (09)
• Nath and Domingos (10a, 10b)
Lifted VE – Poole ('03), Braz et al('05) Milch et al ('08)
Lifted VE + Aggregation – Kisynski & Poole ('09), Choi et al ('11)
Bisimulation + d VE – Sen et al ('08)
FROG – Shavlik & Natarajan ('09)
Approximation
MCMC techniques – Milch & Russell ('06), Poon et al. (08)
Logical Particle Filter - ZettleMoyer et al ('
Lazy Inference – Poon et al ('08)
Anytime BP– Braz et al('09)

Today's Menue

- 9:00 - 9:30 Introduction to Statistical Relational Learning and Inference - Kristian Kersting, Sriraam Natarajan and David Poole
- Morning session Lifted Exact Inference
  - 9:30 - 9:55 Lifted Inference - David Poole
  - 9:55-10:20 First-Order Variable Elimination - Rodrigo de Salvo Braz
  - 10:20-10:40 Break
  - 10:45-11:10 FOVE with counting formulas - Kristian Kersting
  - 11:10-11:35 Bisimulated Variable Elimination - Prithviraj Sen & Lise Getoor
  - 11:35-12:00 Lifted Inference with Continuous Variables - Eyal Amir
- 12:00 - 1:30 Lunch
- Afternoon session Approximate Inference
  - 1:30 - 1:55 Lifted-First Order Belief Propagation - Pedro Domingos
  - 1:55 - 2:20 Counting Belief Propagation - Kristian Kersting
  - 2:20 - 2:45 Bisimulation-based Approximate Lifted Inference - Prithviraj Sen
  - 2:45 - 3:10 Anytime Lifted Belief Propagation - Rodrigo de Salvo Braz
- 3:10 - 3:30 Coffee Break
- Evening Session - Other methods and wrap up
  - 3:30 - 3:55 Preprocessing for Lifted Inference - Sriraam Natarajan
  - 3:55 - 4:20 Probabilistic Theorem Proving - Pedro Domingos
  - 4:20 - 4:45 Overview - David Poole
  - 4:45 - 5:00 Wrap-up and discussion

TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011 CCIB BARCELONA, CATALONIA (SPAIN) WWW.IJCAI-11.ORG
IJCAI'11

First-order probabilistic inference

David Poole

Department of Computer Science,
University of British Columbia

IJCAI 2011 Tutorial

Overview

- Simple representation: parametrized belief networks. Means grounding.
- Inference: combine variable elimination and unification
  - One step of first-order variable elimination corresponds to many VE steps.
- Allows for new queries that depend on population size: probability that someone is guilty of a crime depends on how many other people could have done it.

Outline

1. Background
   - Probability
   - Logic
   - Relational Probabilistic Models
2. First-order Probabilistic Inference
   - Unification and Splitting
   - Lifted VE Operations
3. Conclusions

Bayesians

- Probability is a measure of belief.
- All individuals about which we have the same information should have the same probability.
  - Idea: share probability tables both initially and during inference.

Background: Belief (Bayesian) networks

- Totally order the variables of interest: $X_1, \ldots, X_n$
- $P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|\pi_i)$
  π_i are the parents of X_i: a set of predecessors such that

$$P(X_i|\pi_i) = P(X_i|X_1, \ldots, X_{i-1})$$

Background: variable elimination

To compute the probability of a variable X given evidence $\overline{Z} = \overline{e}$:

$$P(X|\overline{Z} = \overline{e}) = \frac{P(X \wedge \overline{Z} = \overline{e})}{P(\overline{Z} = \overline{e})}$$

Suppose the other variables are $Y_1, \ldots, Y_m$:

$$\begin{aligned} & P(X \wedge \overline{Z}) \\ &= \sum_{Y_m} \cdots \sum_{Y_1} P(X_1, \ldots, X_n) \\ &= \sum_{Y_m} \cdots \sum_{Y_1} \prod_{i=1}^{n} P(X_i|\pi_i) \end{aligned}$$

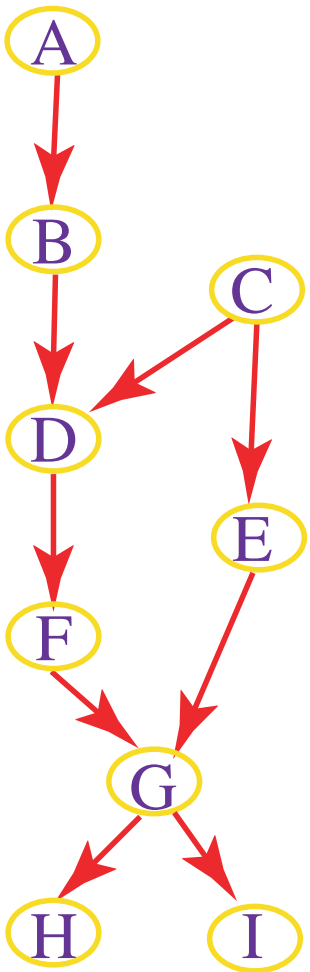
Eliminating a variable

- to compute $AB + AC$ efficiently, distribute out A: $A(B + C)$.
- to compute

$$\sum_{Y_j} \prod_{i=1}^{n} P(X_i|\pi_i)$$

  distribute out those factors that don't involve Y_j.
- Closely related to nonserial dynamic programming [Bertelè & Brioschi, 1972]

Variable Elimination Example



$$\left.\begin{array}{l} P(A) \\ P(B|A) \end{array}\right\} \xrightarrow{\text{elim } A} f_1(B)$$

$$\left.\begin{array}{l} P(C) \\ P(D|BC) \\ P(E|C) \end{array}\right\} \xrightarrow{\text{elim } C} f_2(BDE)$$

$$P(F|D)$$

$$P(G|FE)$$

$$\left.\begin{array}{l} P(H|G) \end{array}\right\} \xrightarrow{\text{obs } H} f_3(G)$$

$$\left.\begin{array}{l} P(I|G) \end{array}\right\} \xrightarrow{\text{elim } I} f_4(G)$$

Variable Elimination: basic operations

- conditioning on observations (local to each factor)
- multiplying factors
- summing a variable from a factor

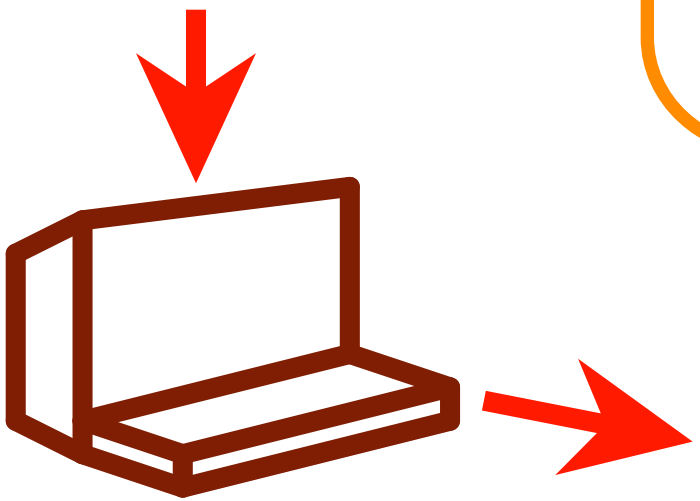
1. Background
   - Probability
   - Logic
   - Relational Probabilistic Models
2. First-order Probabilistic Inference
   - Unification and Splitting
   - Lifted VE Operations
3. Conclusions

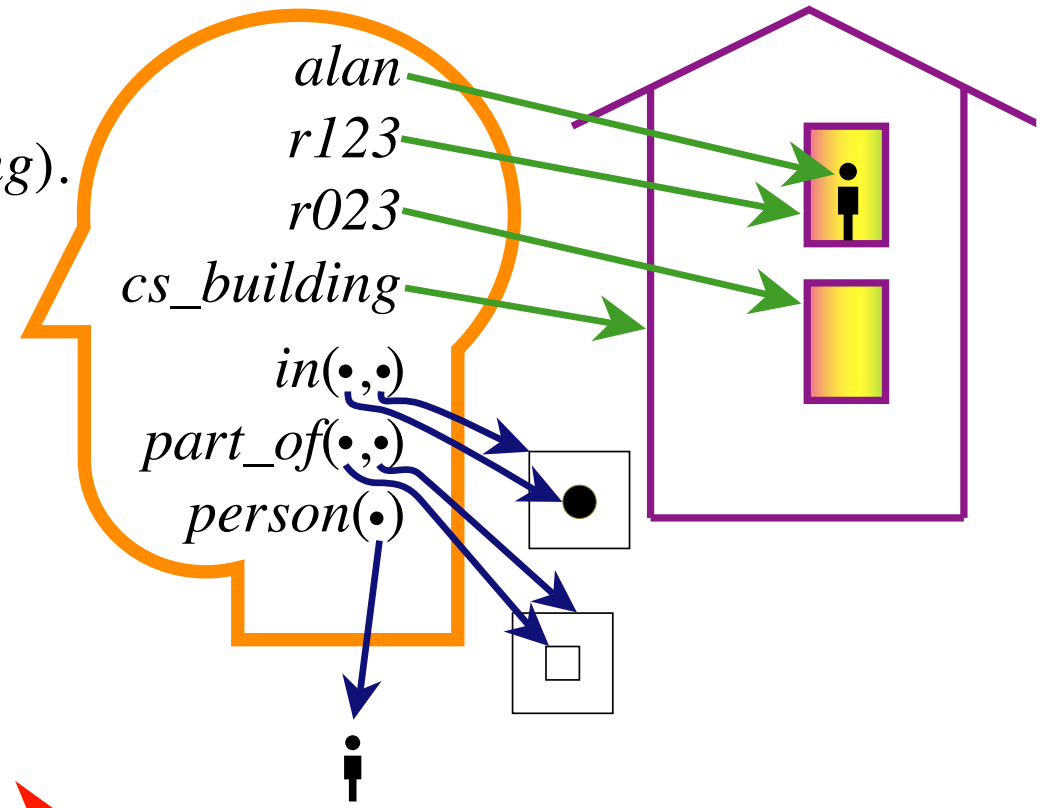
First-order predicate calculus

$in(alan,r123).$

$part\_of(r123,cs\_building).$

$\forall X \forall Y \; in(X,Y) \leftarrow$
$\quad \exists Z \; part\_of(Z,Y) \wedge$
$\quad in(X,Z).$





$in(alan,cs\_building)$

Skolemization and Herbrand's Theorem

Skolemization: give a name for an object said to exist

$\forall X \exists Y \; q(X, Y)$ becomes $q(X, f(X))$

Herbrand's theorem [1930]:

- If a logical theory has a model it has a model where the domain is made of ground terms, and each term denotes itself.
- If a logical theory T is unsatisfiable, there is a finite set of ground instances of formulas of T which is unsatisfiable.

Unique Names Assumption & Negation as Failure

- Unique Names Assumption:
  different names denote different individuals
  different ground terms denote different individuals
- Herbrand's theorem holds even without the unique names assumption.

Theorem Proving and Unification

In 1965, Robinson showed how unification allows many ground steps with one step:

$$\frac{f(X,Z) \vee p(X,a) \qquad \neg p(b,Y) \vee g(Y,W)}{f(b,Z) \vee g(a,W)}$$

Substitution $\{X/b, Y/a\}$ is the most general unifier of $p(X,a)$ and $p(b,Y)$.

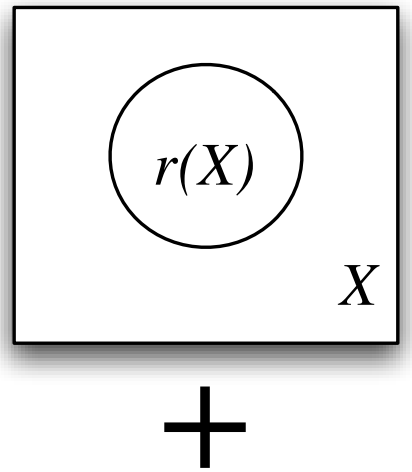
1. Background
   - Probability
   - Logic
   - Relational Probabilistic Models
2. First-order Probabilistic Inference
   - Unification and Splitting
   - Lifted VE Operations
3. Conclusions

Parametrized belief networks

- Allow random variables to be parametrized. $height(X)$
- Parameters correspond to logical variables. X
- Each parameter is typed with a population. $X : person$
- Each population has a size. $|person| = 1000000$
- Parametrized belief network means its grounding.
  Ground instances are random variables:
  $height(p_1) \ldots height(p_{1000000})$
- Instances are independent (but can have common ancestors and descendents).

Parametrized Bayesian networks / Plates

Parametrized Bayes Net:

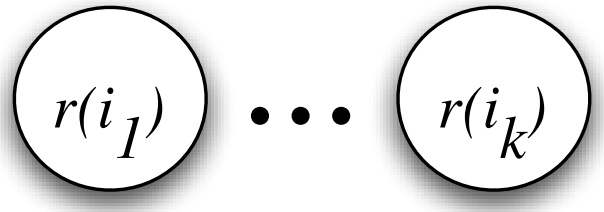


Individuals:

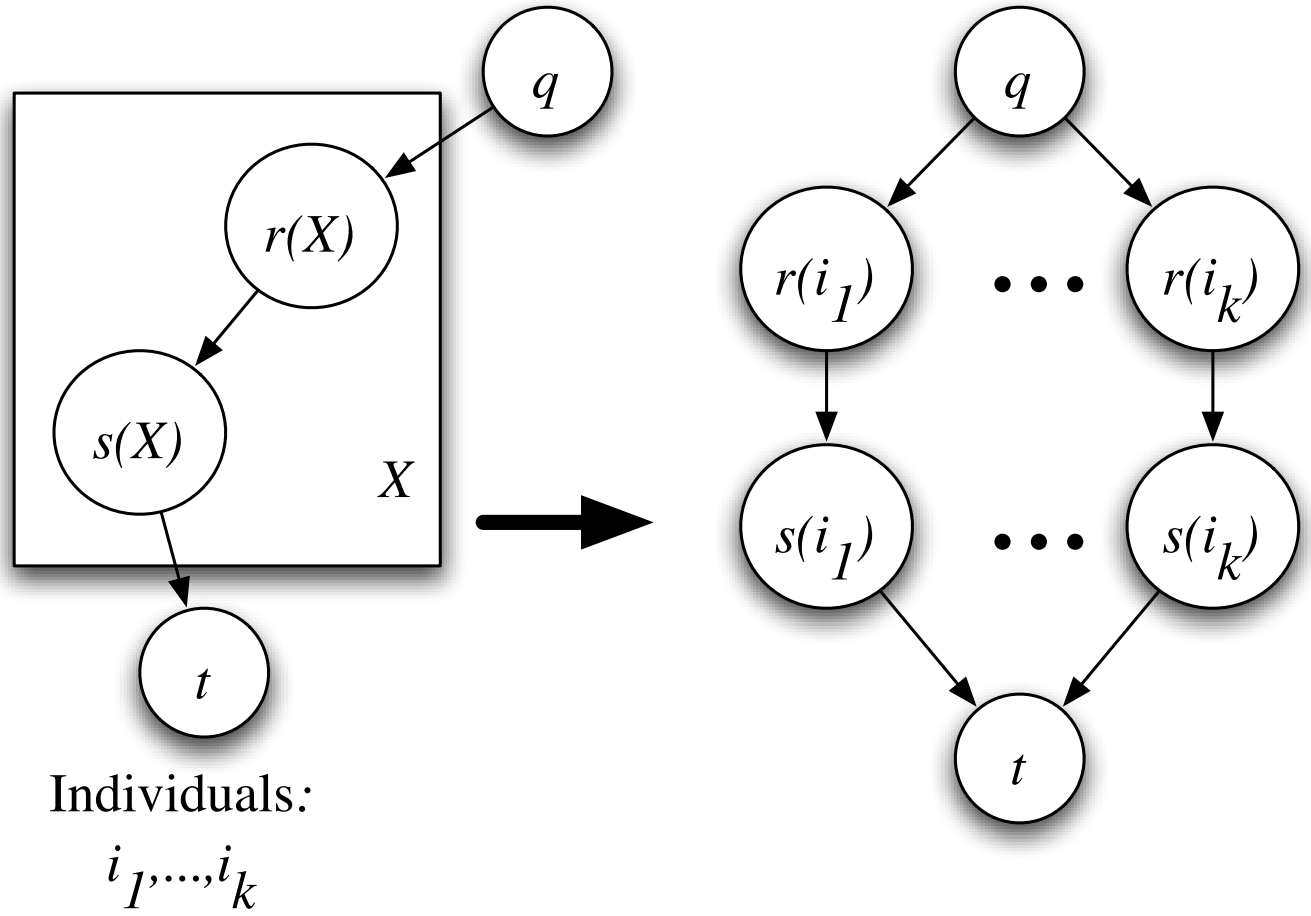
$i_1,\ldots,i_k$



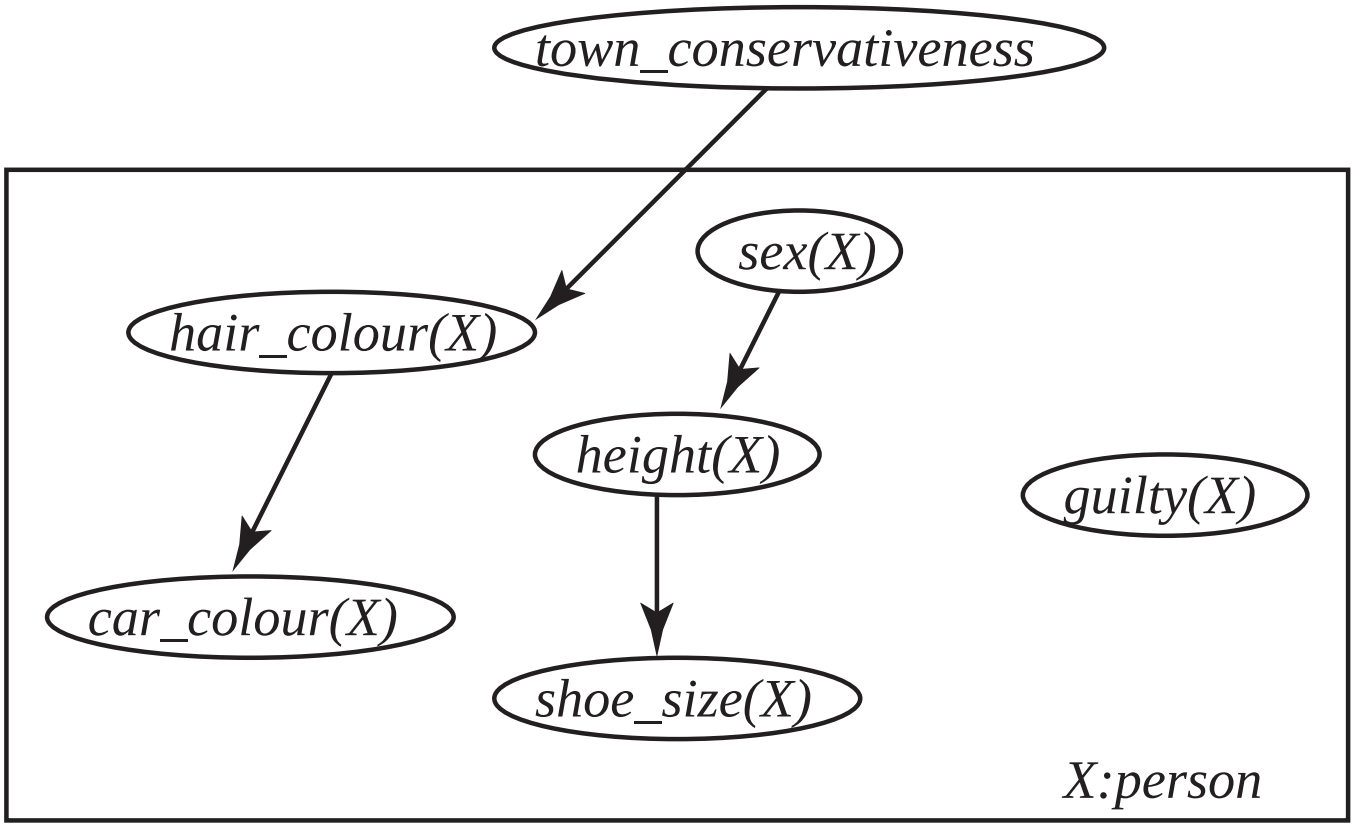
Bayes Net



Parametrized Bayesian networks / Plates (2)



Example parametrized belief network



$\forall X \; P(car\_colour(X)=pink | hair\_colour(X)=pink) = 0.1$

$\forall X \; P(hair\_colour(X)=pink | town\_conservative) = 0.001.$

1 Background
- Probability
- Logic
- Relational Probabilistic Models

2 First-order Probabilistic Inference
- Unification and Splitting
- Lifted VE Operations

3 Conclusions

Theorem Proving and Unification (reprise)

In 1965, Robinson showed how unification allows many ground steps with one step:

$$\frac{f(X,Z) \vee p(X,a) \qquad \neg p(b,Y) \vee g(Y,W)}{f(b,Z) \vee g(a,W)}$$

Substitution $\{X/b, Y/a\}$ is the most general unifier of $p(X,a)$ and $p(b,Y)$.

Variable Elimination and Unification

- Multiplying parametrized factors:

$$\underbrace{[f(X,Z),p(X,a)] \quad \times \quad [p(b,Y),g(Y,W)]}_{[f(b,Z),p(b,a),g(a,W)]}$$

  Doesn't quite work because the first parametrized factor can't be used for $X = b$ but can be used for other instances of X.

- Intuitively, we want to add the constraint $X \neq b$ to $[f(X,Z),p(X,a)]$ after the above multiplication.

Parametric Factors

A parametric factor is a triple $\langle C, V, t \rangle$ where

- C is a set of inequality constraints on parameters,
- V is a set of parametrized random variables
- t is a table representing a factor from the random variables to the non-negative reals.

$$\left\langle \{X \neq sue\}, \{hair\_col(X), cons\}, \right.$$

$hair\_col$	$cons$	Val
$purple$	yes	0.001
$purple$	no	0.01
	...	

$$\left. \right\rangle$$

Splitting

Instead of applying substitutions to parametric factors, we split the parametric factors on the substitution.
A **split** of $\langle C, V, t\rangle$ on $X = \gamma$, results in parametric factors:

$$\langle C[X/\gamma], V[X/\gamma], t\rangle$$
$$\langle \{X \neq \gamma\} \cup C, V, t\rangle \qquad \longleftarrow \text{ residual}$$

where $V[X/\gamma]$ is V with γ substituted for X.

Splitting on a substitution

- Splitting on a substitution, means splitting on each equality in the substitution.
- Different orders of splitting give the same final result, but may give different residuals.
- Example: Split

$$\langle \{\}, \{foo(X, Y, Z)\}, t_1 \rangle$$

on $\{X = Z, Y = b\}$ results in

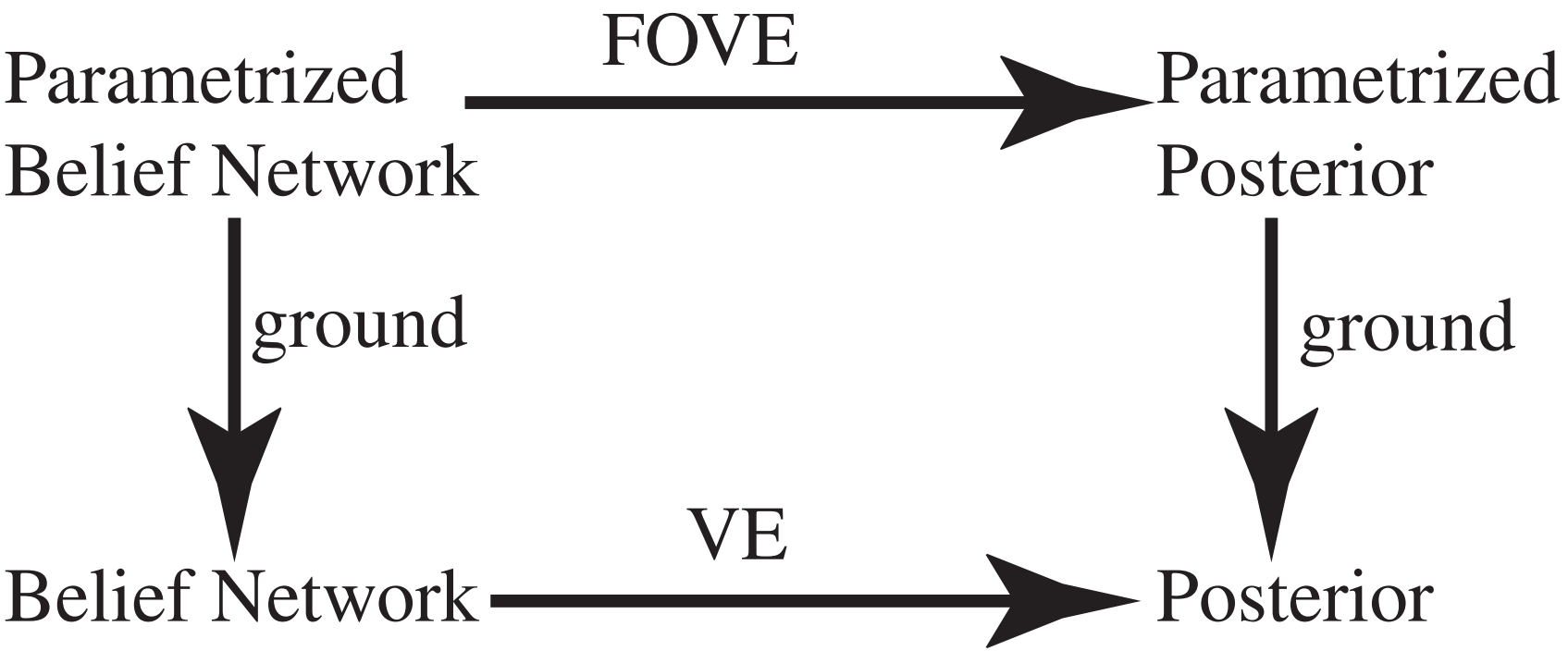
$$\langle \{\}, \{foo(X, b, X)\}, t_1 \rangle$$
$$\langle \{X \neq Z\}, \{foo(X, Y, Z)\}, t_1 \rangle$$
$$\langle \{Y \neq b\}, \{foo(X, Y, X)\}, t_1 \rangle$$

1 Background
- Probability
- Logic
- Relational Probabilistic Models

2 First-order Probabilistic Inference
- Unification and Splitting
- Lifted VE Operations

3 Conclusions

First-order probabilistic inference



Multiplying Parametric Factors

Suppose we were to eliminate p and multiply the two parametric factors:

$$\langle \{\}, \{p(X, a), q(Y, c), s(X, Y)\}, t_1 \rangle$$
$$\langle \{W \neq d\}, \{p(b, Z), q(W, T), r(W, T)\}, t_2 \rangle$$

- If we grounded these, then did VE, some instances of these would be multiplied and some wouldn't.
- We unify $p(X, a)$ and $p(b, Z)$ resulting in the substitution $\theta = \{X/b, Z/a\}$.
- Unification finds the most general instances that need to be multiplied.

Splitting when Multiplying I

We are multiplying the two parametric factors:

$$\langle \{\}, \{p(X, a), q(Y, c), s(X, Y)\}, t_1 \rangle \quad (1)$$
$$\langle \{W \neq d\}, \{p(b, Z), q(W, T), r(W, T)\}, t_2 \rangle \quad (2)$$

We split parametric factor (1) on $\theta = \{X/b, Z/a\}$:

$$\langle \{\}, \{p(b, a), q(Y, c), s(b, Y)\}, t_1 \rangle \quad (3)$$
$$\langle \{X \neq b\}, \{p(X, a), q(Y, c), s(X, Y)\}, t_1 \rangle \quad (4)$$

We can split (2) on θ resulting in:

$$\langle \{W \neq d\}, \{p(b, a), q(W, T), r(W, T)\}, t_2 \rangle \quad (5)$$
$$\langle \{Z \neq a, W \neq d\}, \{p(b, Z), q(W, T), r(W, T)\}, t_2 \rangle \quad (6)$$

Splitting when Multiplying II

When we are multiplying:

$$\langle \{\}, \{p(b,a), q(Y,c), s(b,Y)\}, t_1 \rangle$$
$$\langle \{W \neq d\}, \{p(b,a), q(W,T), r(W,T)\}, t_2 \rangle$$

- All ground instances would need to be multiplied.
- Not all instances have the same number of variables: some will have two different q instances, and some have one.
- We need to split again on the most general unifier of $q(Y,c)$ and $q(W,T)$.

Summing out variables

- If we are not removing a parameter, we sum out as normal. E.g., summing out p:

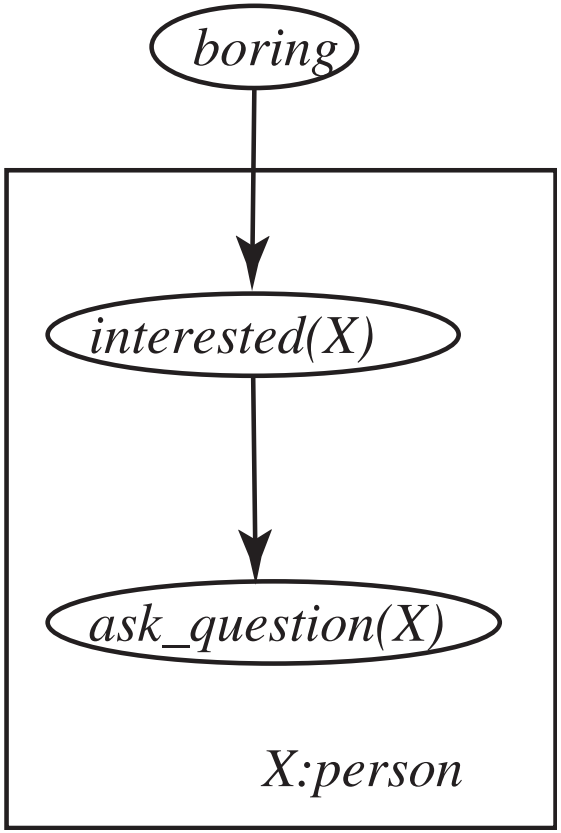
$$\langle \{\}, \{p(X), q(X)\}, t[p, q] \rangle$$

- If we are removing a parameter, we must take to the power of the effective population size. E.g., summing out p:

$$\langle \{Y \neq a\}, \{p(X, Y), q(X)\}, t[p, q] \rangle$$

- Other functions such as noisy-or, you need to take into account the population size.

Removing a parameter when summing



$|people| = 100$

observe no questions

Eliminate *interested*:

$$\langle \{\}, \{boring, interested(X)\}, t_1 \rangle$$
$$\langle \{\}, \{interested(X)\}, t_2 \rangle$$
$$\downarrow$$
$$\langle \{\}, \{boring\}, (t_1 \times t_2)^{100} \rangle$$

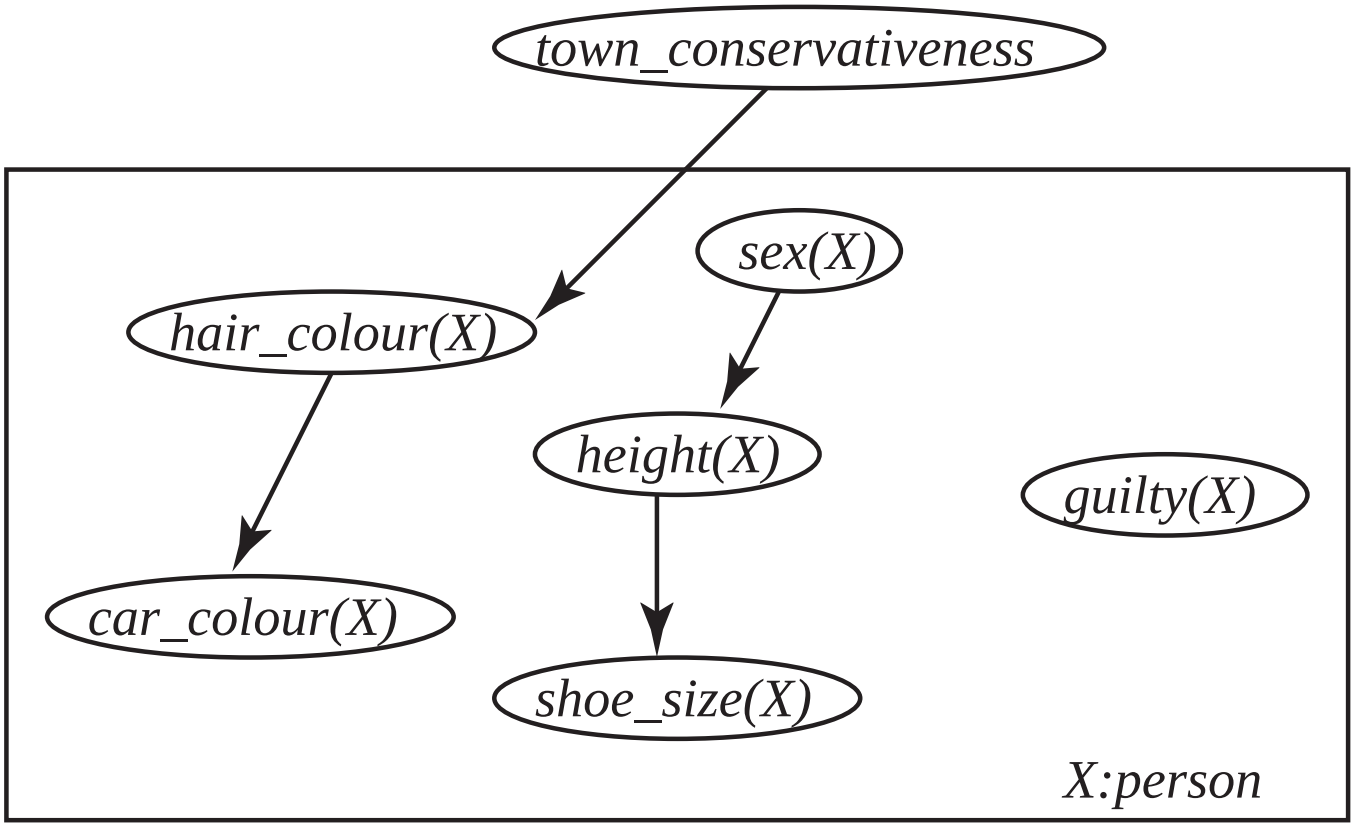
Existential Observations

Suppose we observe:

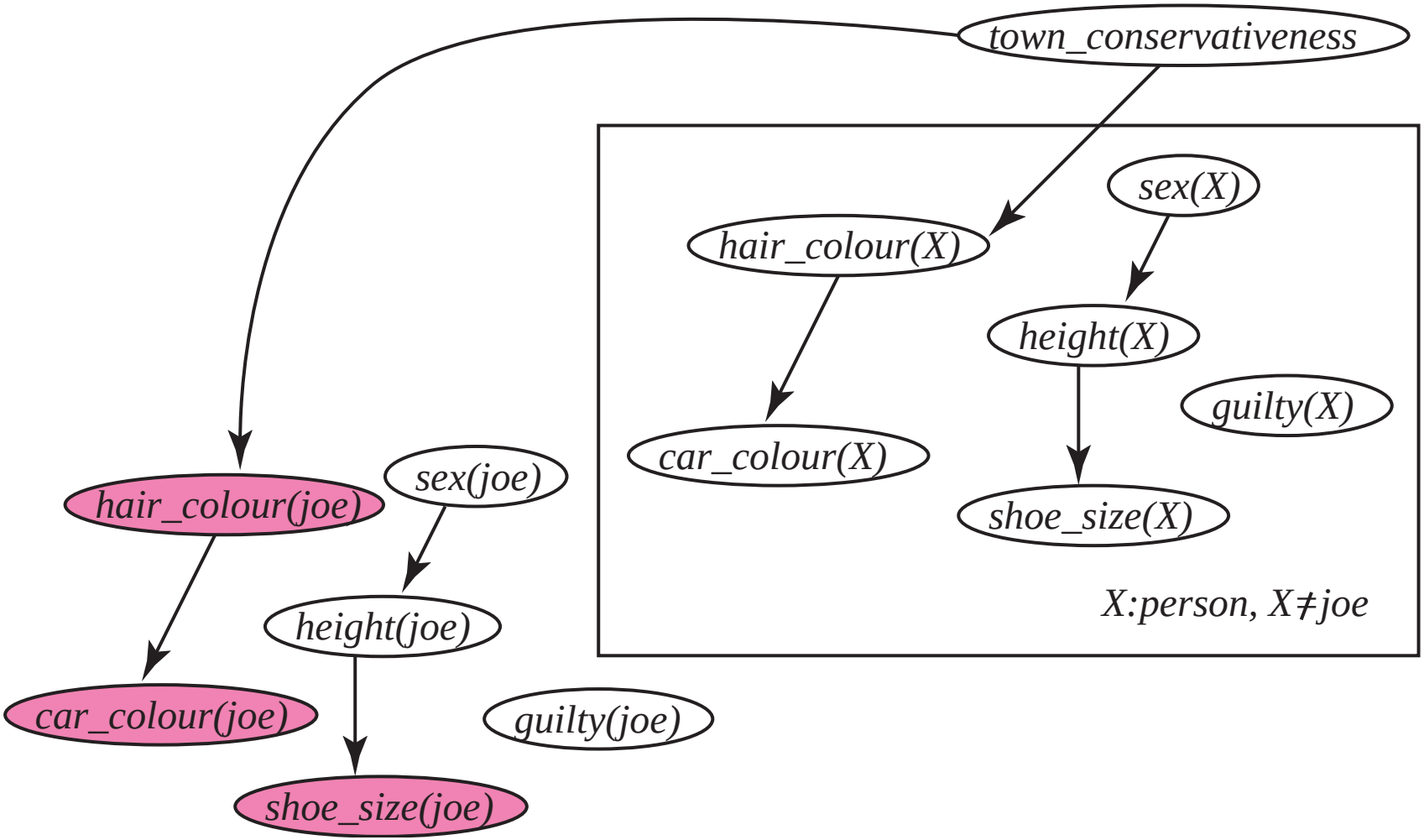
- Joe has purple hair, a purple car, and has big feet.
- A person with purple hair, a purple car, and who is very tall was seen committing a crime.

What is the probability that Joe is guilty?

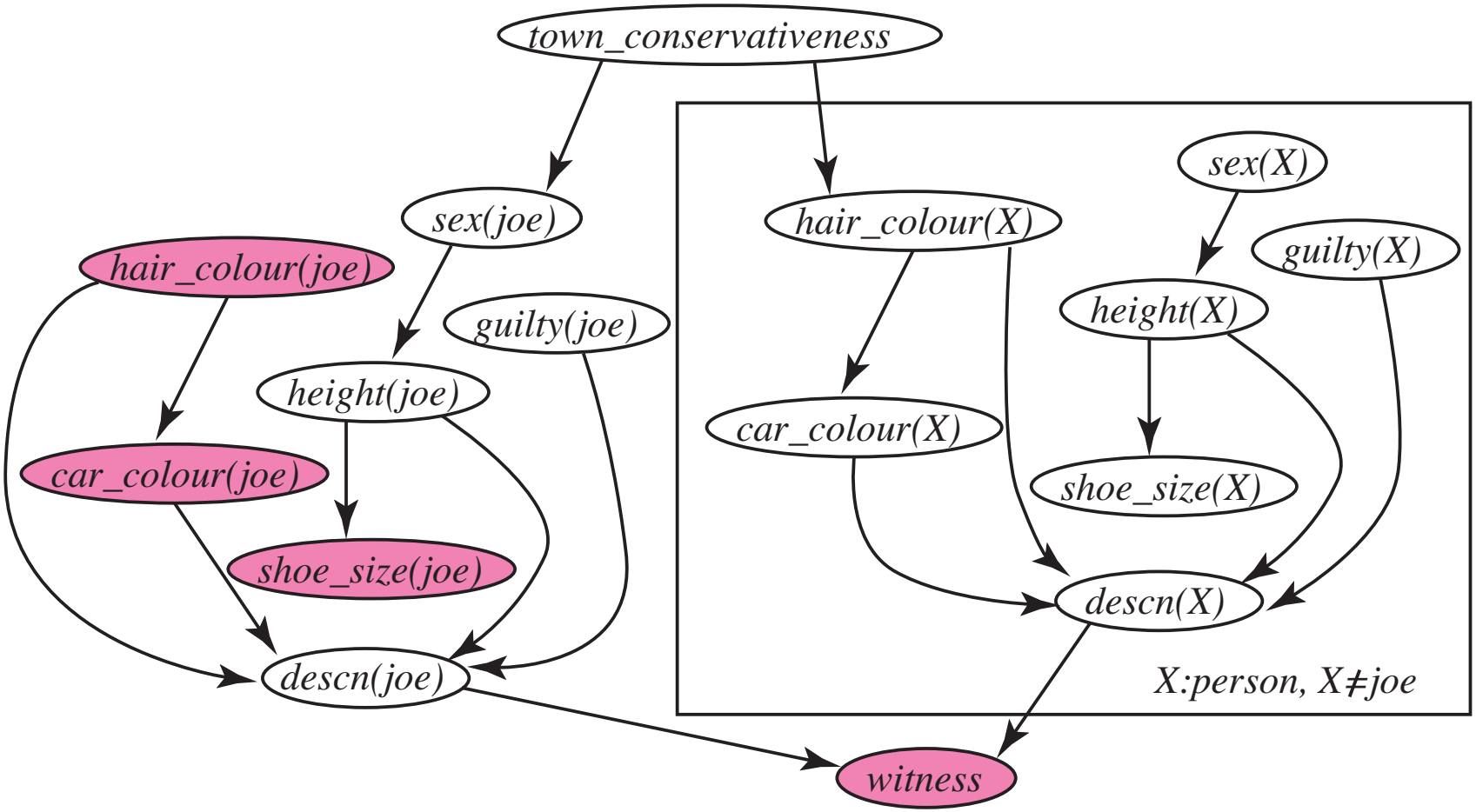
Background parametrized belief network



Observing information about Joe



Observing Joe and the crime



Last Steps

We end up with parametric Factors:

$$\langle \{\}, \{guilty(joe), descn(joe), conservativeness\}, t_1 \rangle$$
$$\langle \{X \neq joe\}, \{descn(X), conservativeness\}, t_2 \rangle$$
$$\langle \{\}, \{descn(X), witness\}, t_3 \rangle$$
$$\langle \{\}, \{conservativeness\}, t_4 \rangle$$

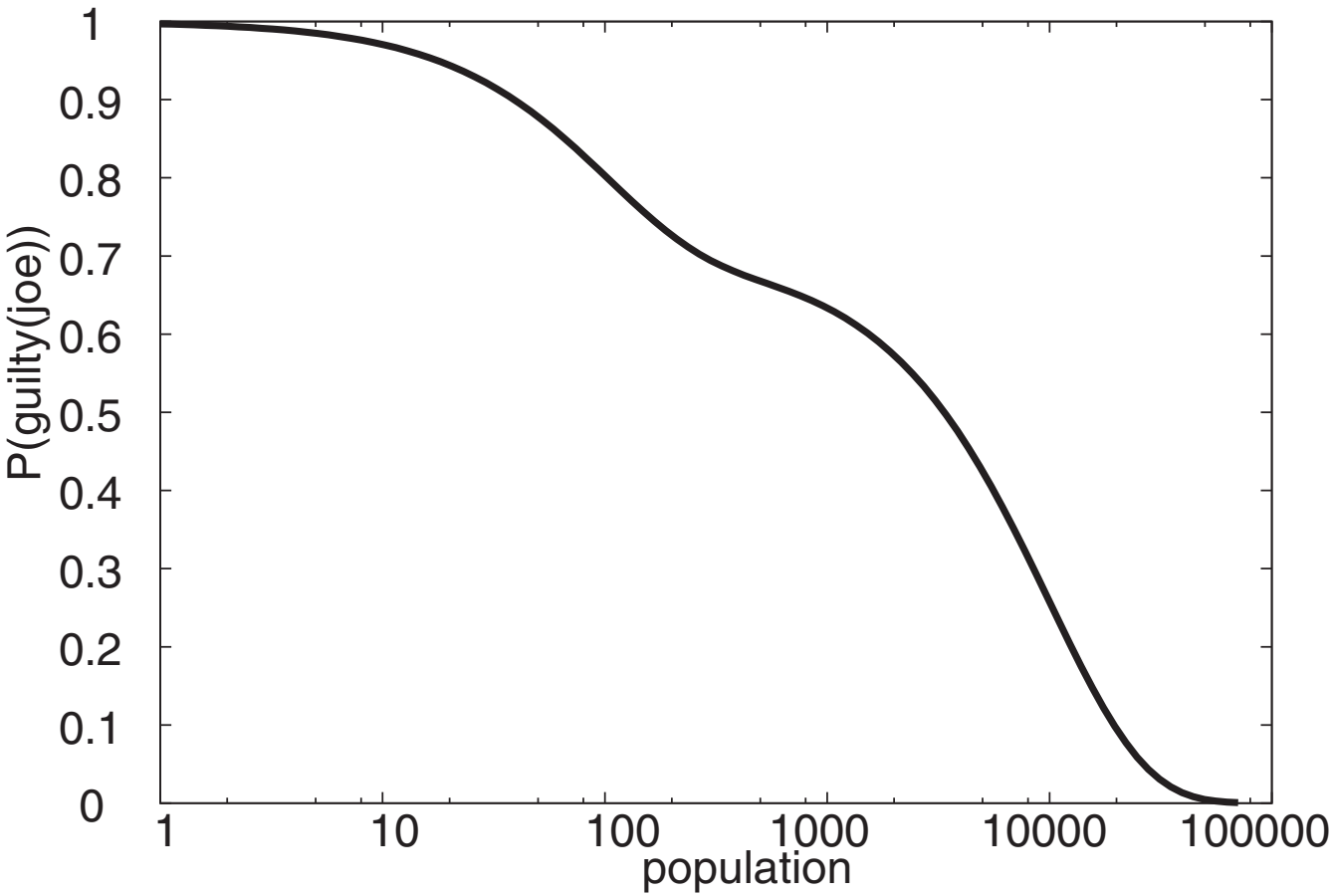
We eliminate $descn(X)$:

$$\langle \{\}, \{guilty(joe), witness, conservativeness\}, t_5 \rangle$$

We sum out *conservativeness* and condition on *witness*:

$$\langle \{\}, \{guilty(joe)\}, t_6 \rangle$$

Guilty as a function of population



1 Background
- Probability
- Logic
- Relational Probabilistic Models

2 First-order Probabilistic Inference
- Unification and Splitting
- Lifted VE Operations

3 Conclusions

Conclusions

- We combine variable elimination + unification.
  - One step of first-order variable elimination corresponds to many steps in ground representation.
  - We can condition on existential and universal observations.
- Contributions of IJCAI-93 paper:
  - parametrized random variables
  - splitting to complement unification
  - parfactor representation of intermediate results
  - an algorithm for multiplying factors in a lifted manner (sometimes)

Lifted First-Order Probabilistic Inference

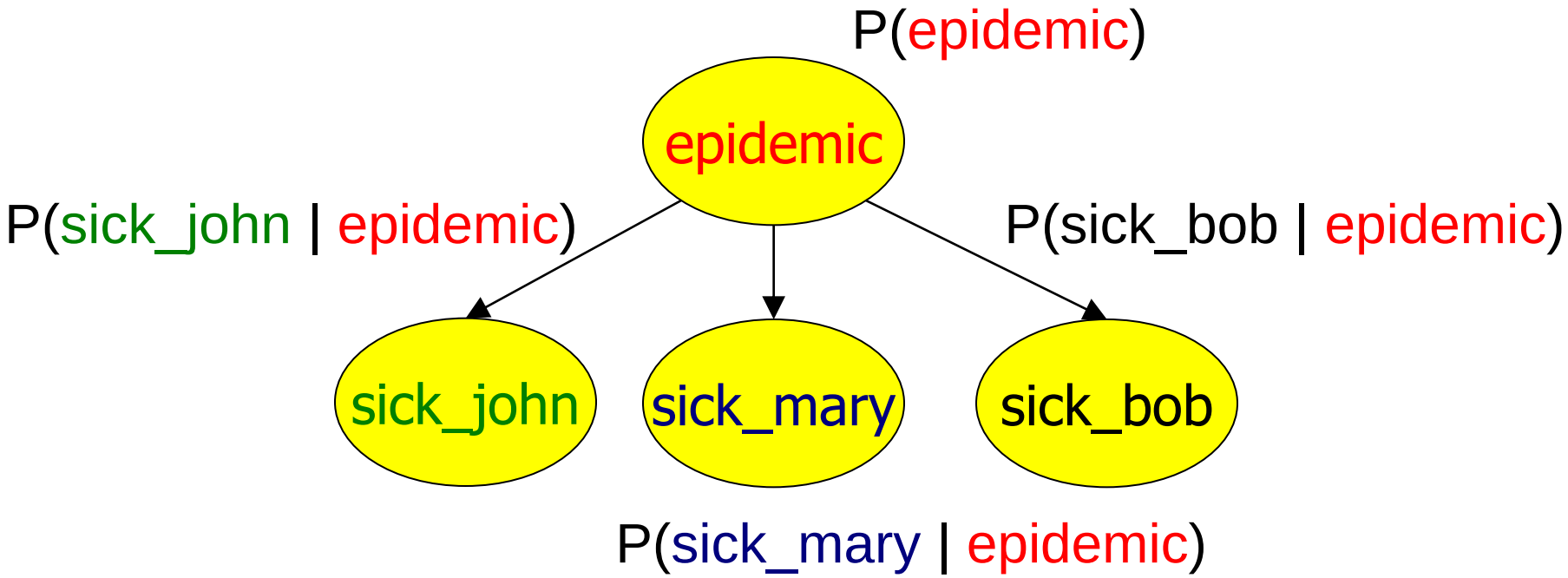
Rodrigo de Salvo Braz
SRI International

based on work with
Eyal Amir and Dan Roth

Outline

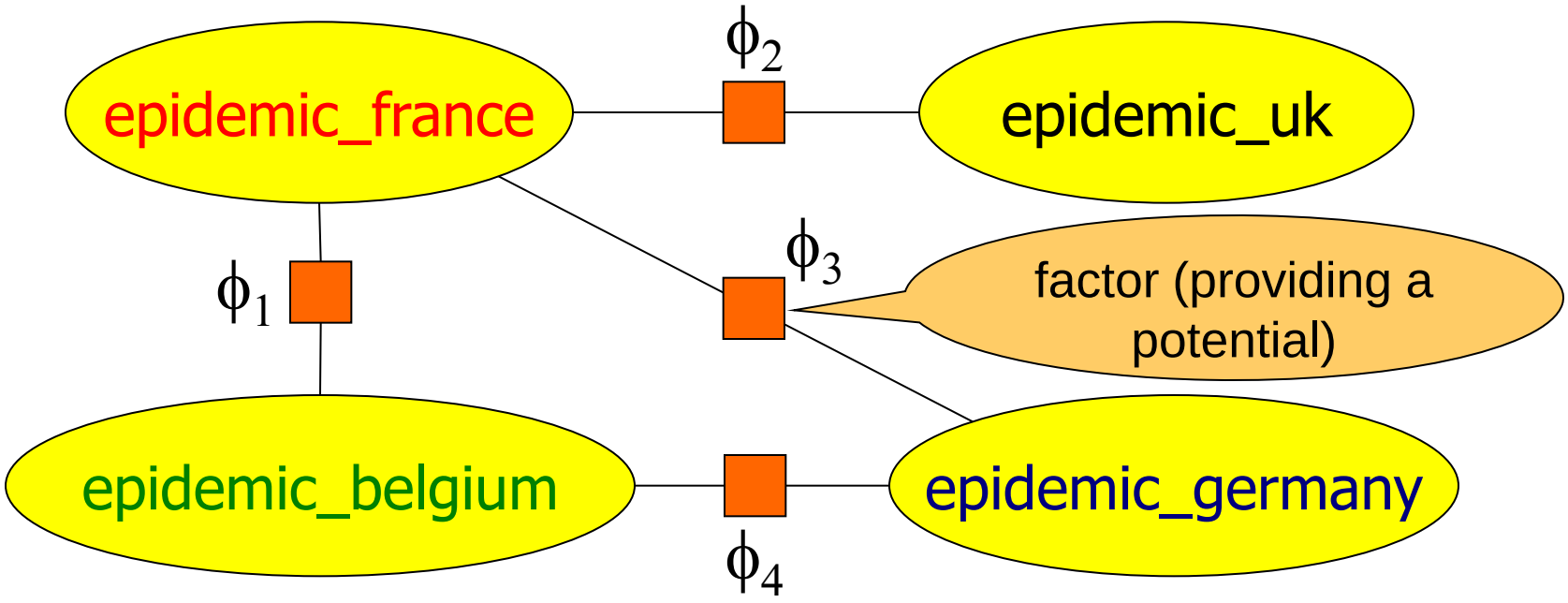
- Motivation
  - Brief background
  - Lifted Inference example
- First-order representations
- Lifted First-Order Probabilistic Inference
  - Inversion Elimination
  - Counting Elimination
  - Partial Inversion
- Conclusion

Bayesian Networks (directed)



P(sick_john, sick_mary, sick_bob, epidemic)
= P(sick_john | epidemic) * P(sick_mary | epidemic)
* P(sick_bob | epidemic) * P(epidemic)

Factor Networks (undirected)

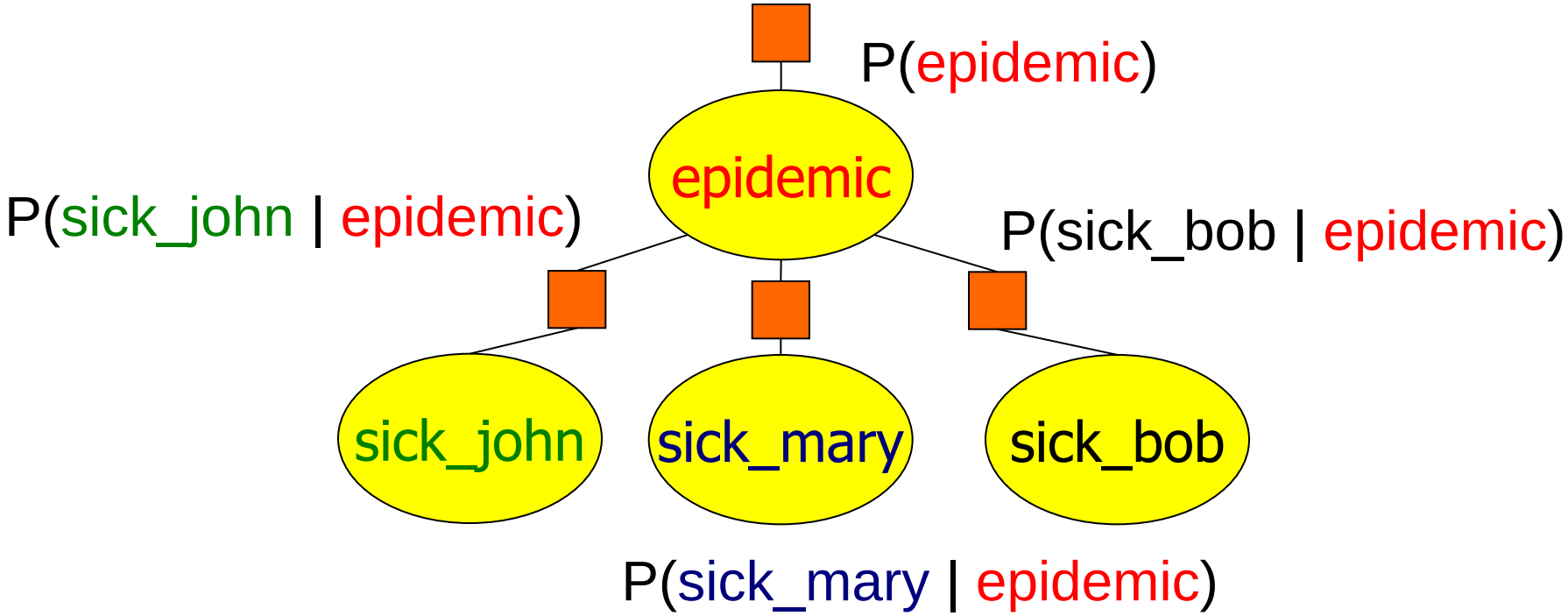


P(epi_france, epi_belgium, epi_uk, epi_germany)

$\propto$ ϕ_1(epi_france, epi_belgium) * ϕ_2(epi_france, epi_uk)

* ϕ_3(epi_france, epi_germany) * ϕ_4(epi_belgium, epi_germany)

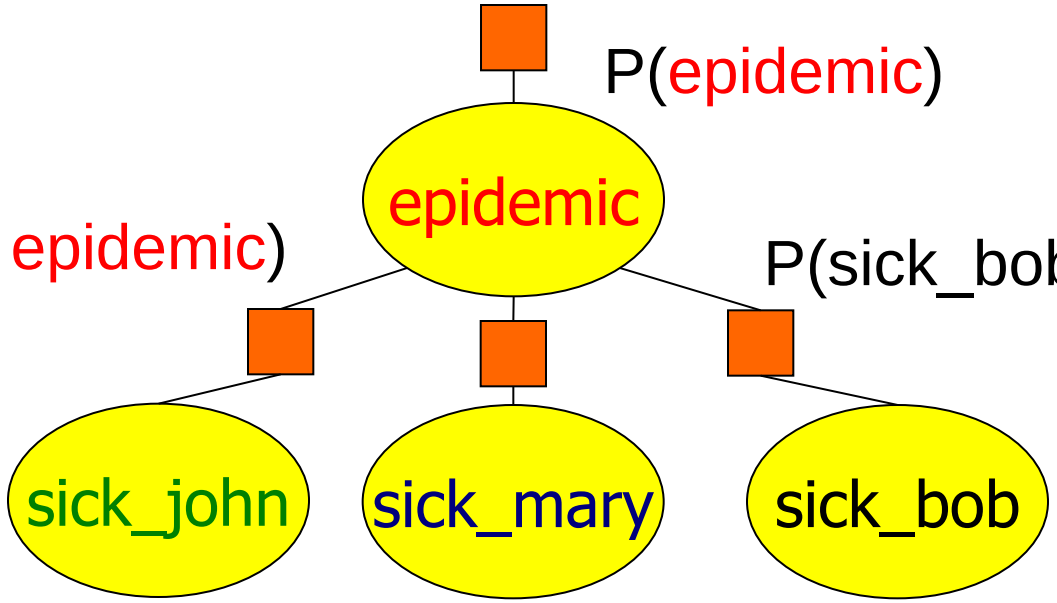
Bayesian Nets as Factor Networks



P(sick_john, sick_mary, sick_bob, epidemic)

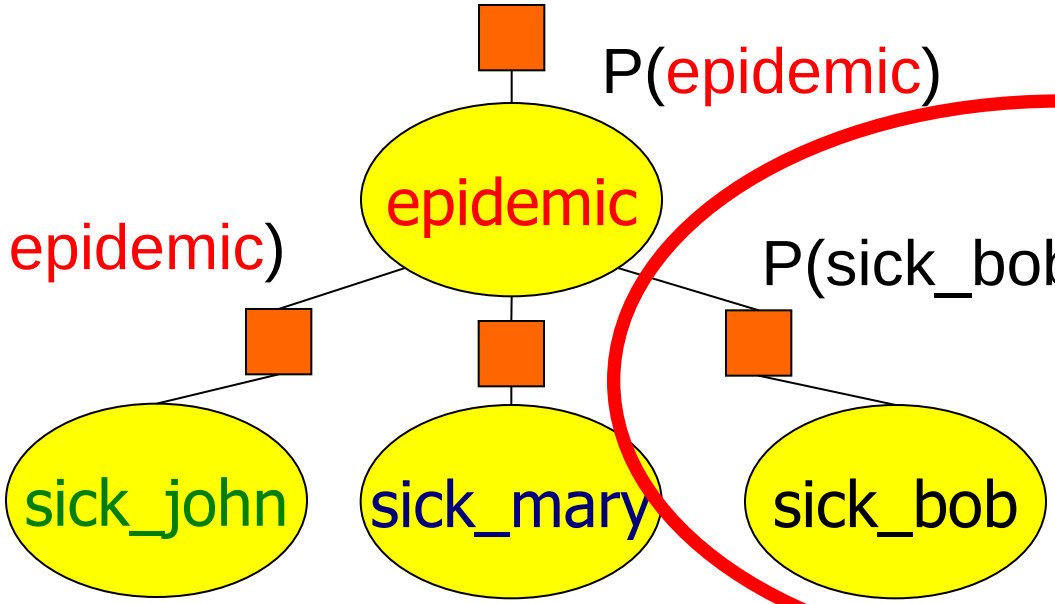
$\propto$ P(sick_john | epidemic) * P(sick_mary | epidemic)
* P(sick_bob | epidemic) * P(epidemic)

Inference: Marginalization



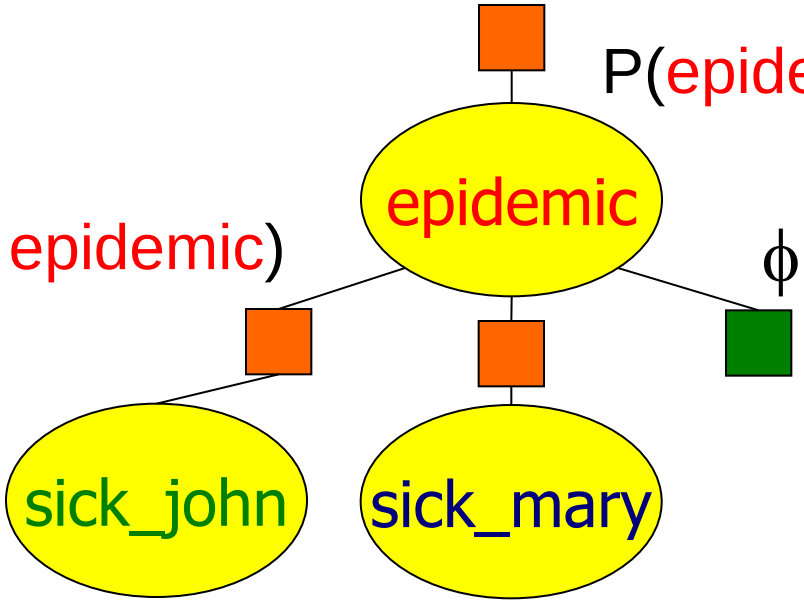
$$P(\text{sick\_john}) \propto \sum_{\text{epidemic}} \sum_{\text{sick\_mary}} \sum_{\text{sick\_bob}} P(\text{sick\_john} \mid \text{epidemic}) * P(\text{sick\_mary} \mid \text{epidemic}) * P(\text{sick\_bob} \mid \text{epidemic}) * P(\text{epidemic})$$

Inference: Variable Elimination (VE)



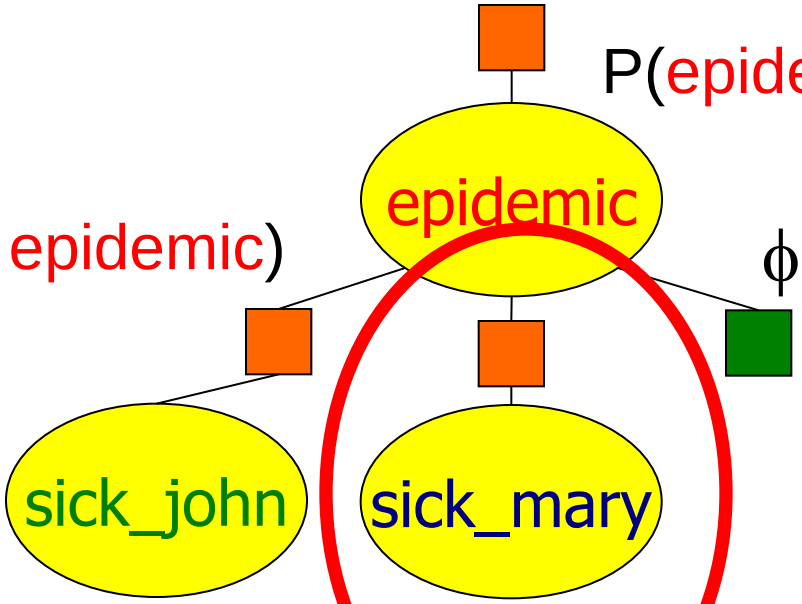
$$P(\text{sick\_john}) \propto \sum_{\text{epidemic}} P(\text{sick\_john} \mid \text{epidemic}) * P(\text{epidemic})$$
$$* \sum_{\text{sick\_mary}} P(\text{sick\_mary} \mid \text{epidemic})$$
$$* \sum_{\text{sick\_bob}} P(\text{sick\_bob} \mid \text{epidemic})$$

Inference: Variable Elimination (VE)



$$P(\text{sick\_john}) \propto \sum_{\text{epidemic}} P(\text{sick\_john} \mid \text{epidemic}) * P(\text{epidemic}) * \sum_{\text{sick\_mary}} P(\text{sick\_mary} \mid \text{epidemic}) * \phi_1(\text{epidemic})$$

Inference: Variable Elimination (VE)



P(epidemic)

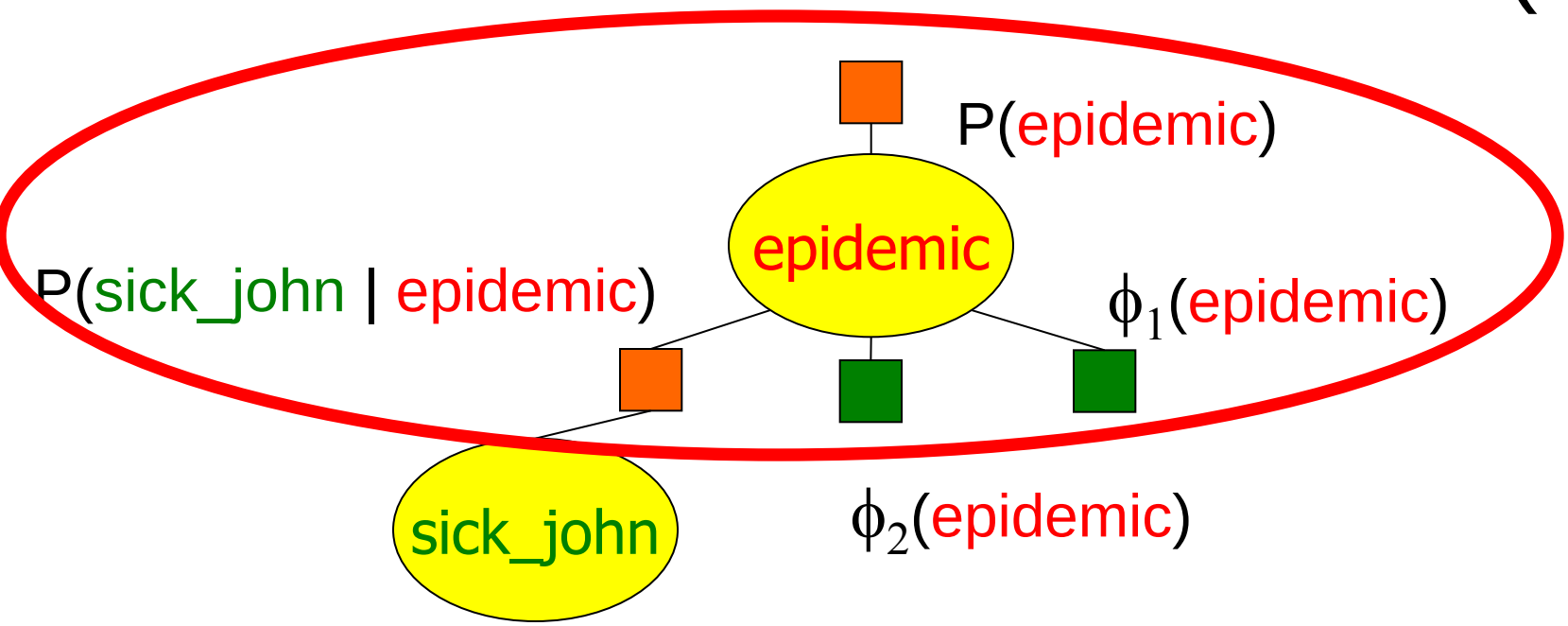
P(sick_john | epidemic)

ϕ_1(epidemic)

P(sick_mary | epidemic)

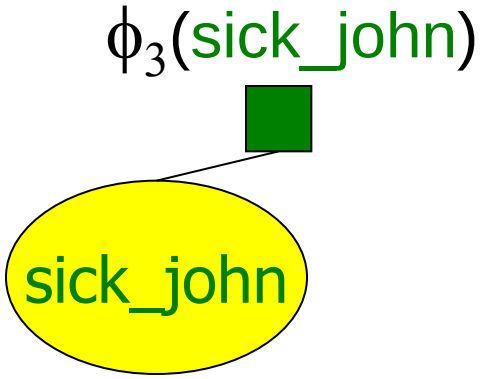
$P(\text{sick\_john}) \propto \sum_{\text{epidemic}} P(\text{sick\_john} \mid \text{epidemic}) * P(\text{epidemic}) * \phi_1(\text{epidemic}) * \sum_{\text{sick\_mary}} P(\text{sick\_mary} \mid \text{epidemic})$

Inference: Variable Elimination (VE)



$$P(\text{sick\_john}) \propto \sum_{\text{epidemic}} P(\text{sick\_john} \mid \text{epidemic}) * P(\text{epidemic}) * \phi_1(\text{epidemic}) * \phi_2(\text{epidemic})$$

Inference: Variable Elimination (VE)



P(sick_john) $\propto$ ϕ_3 (sick_john)

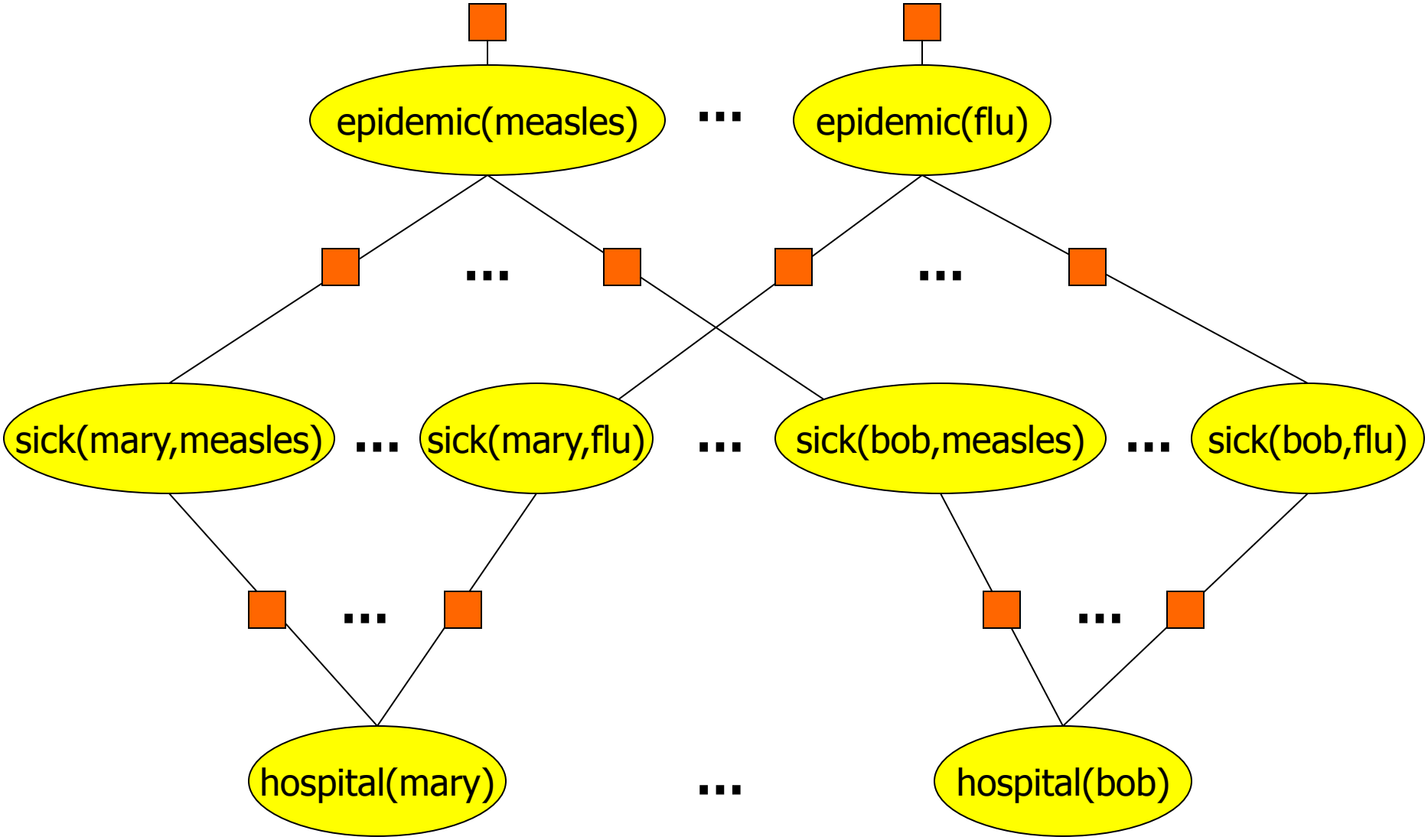
Outline

- Motivation
  - Brief background
  - Lifted Inference example
- First-order representations
- Lifted First-Order Probabilistic Inference
  - Inversion Elimination
  - Counting Elimination
  - Partial Inversion
- Conclusion

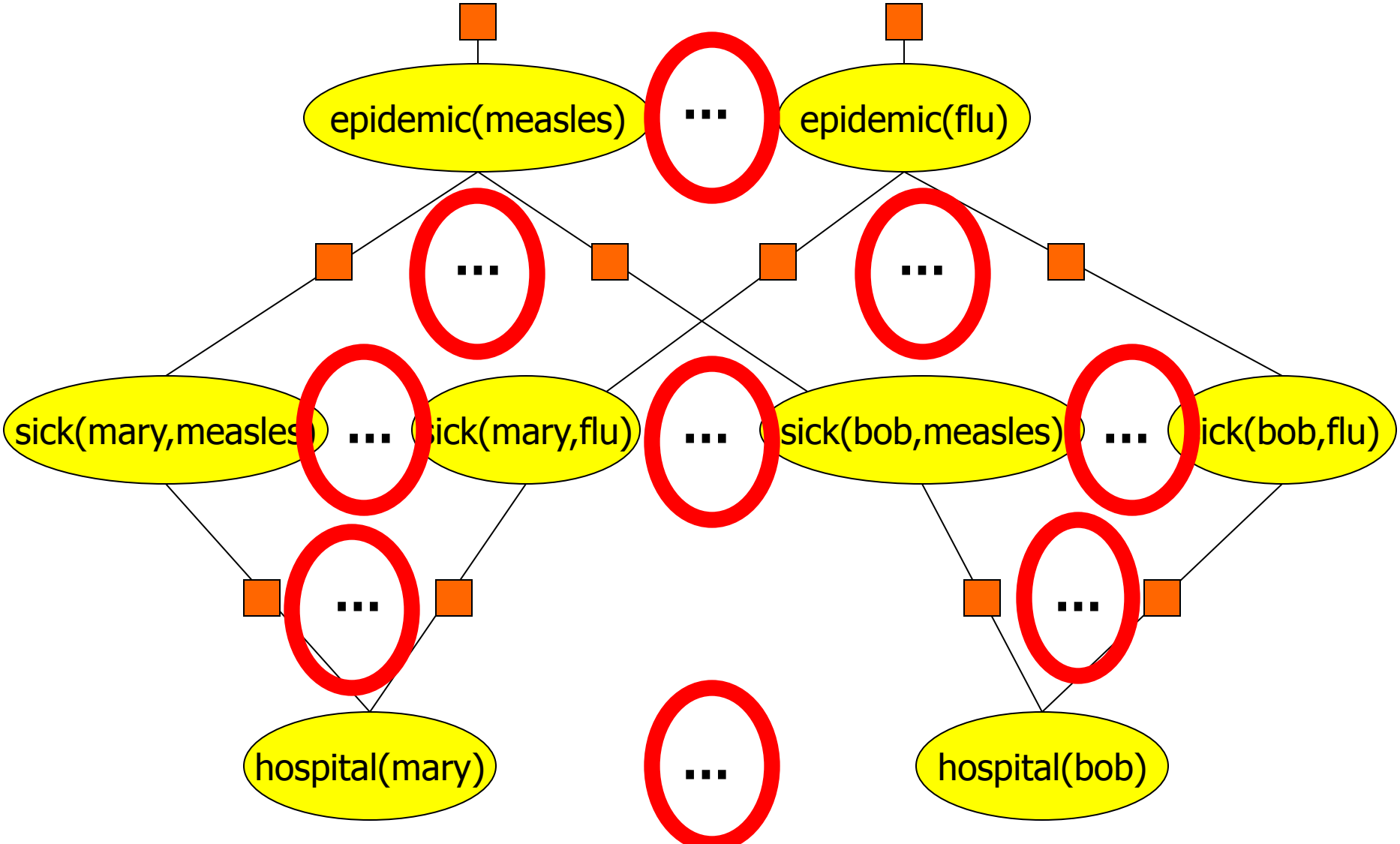
A Lifted inference example

- Dependencies between
  - epidemic(Disease) and sick(Person, Disease)
  - sick(Person, Disease) and hospital(Person)
- Applied to many people and many diseases.

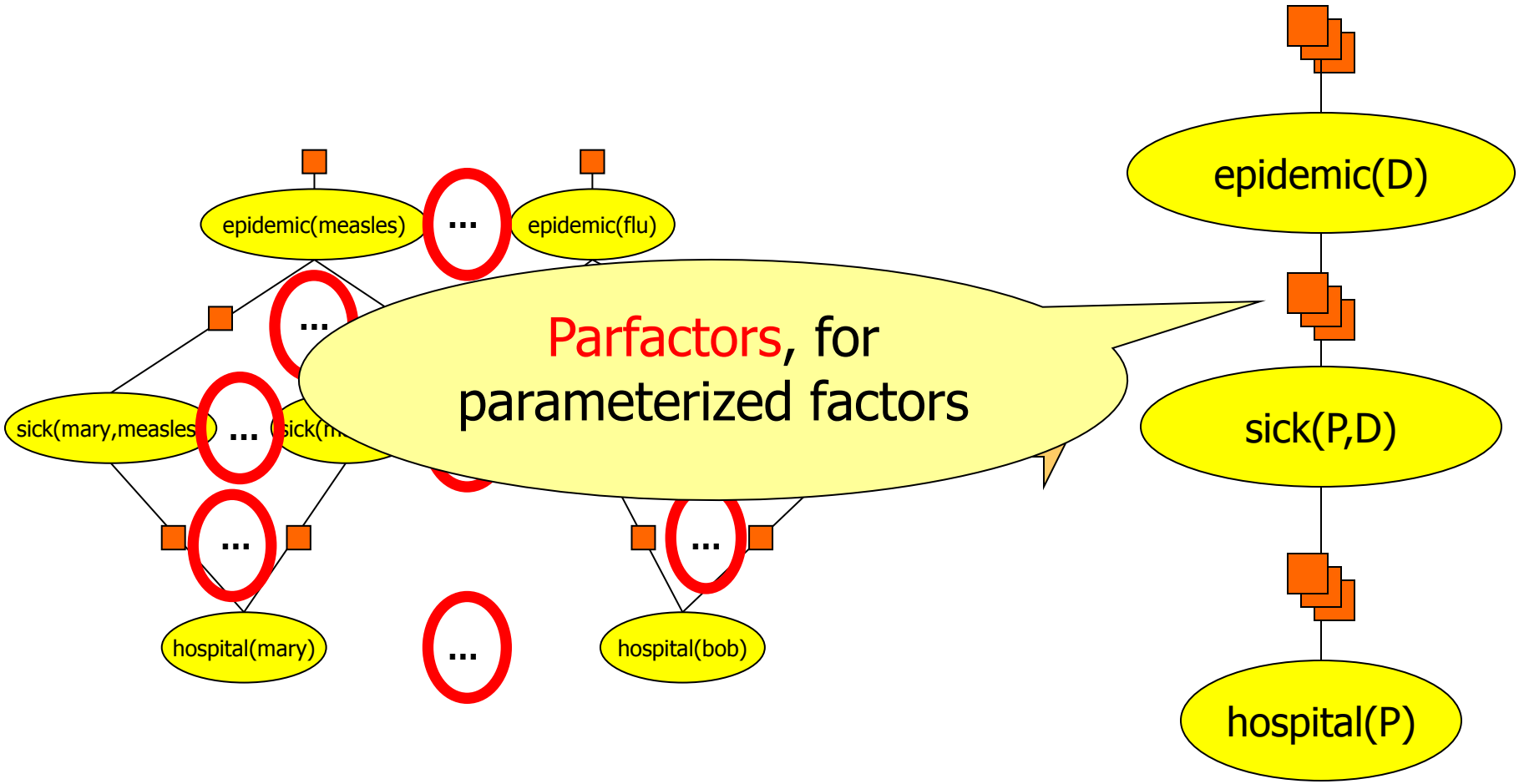
Graphical model size can be large



Much redundancy

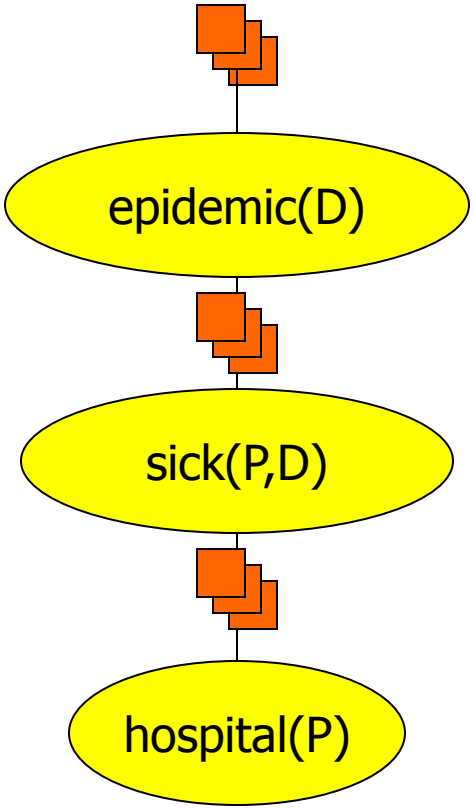


Representing structure



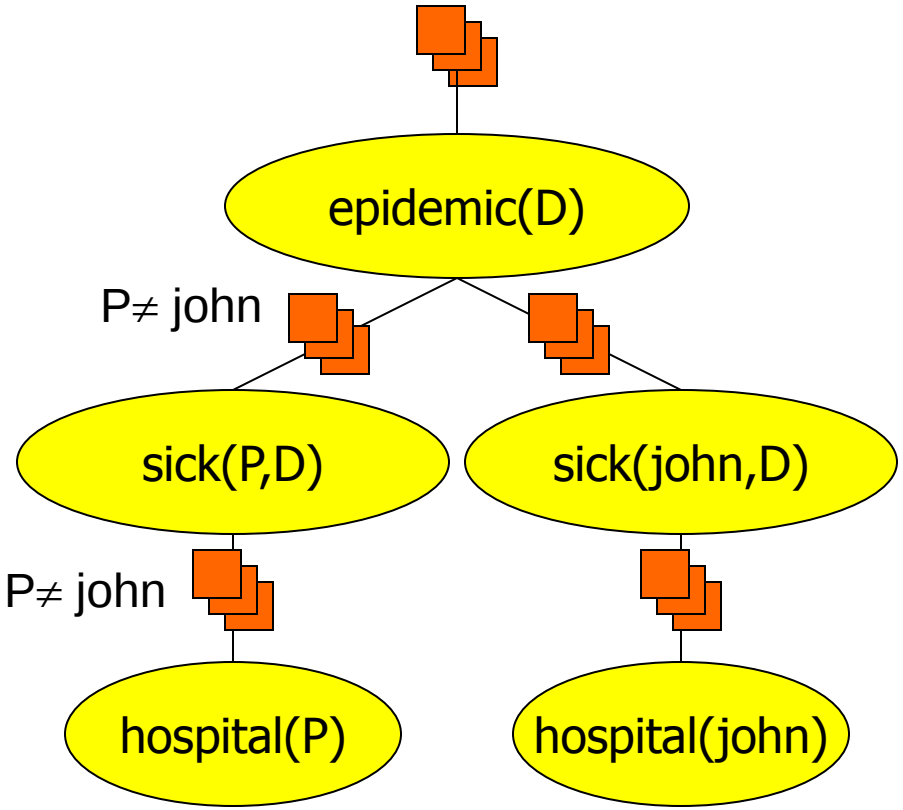
Lifted Inference Example

- P(hospital(john)) = ?



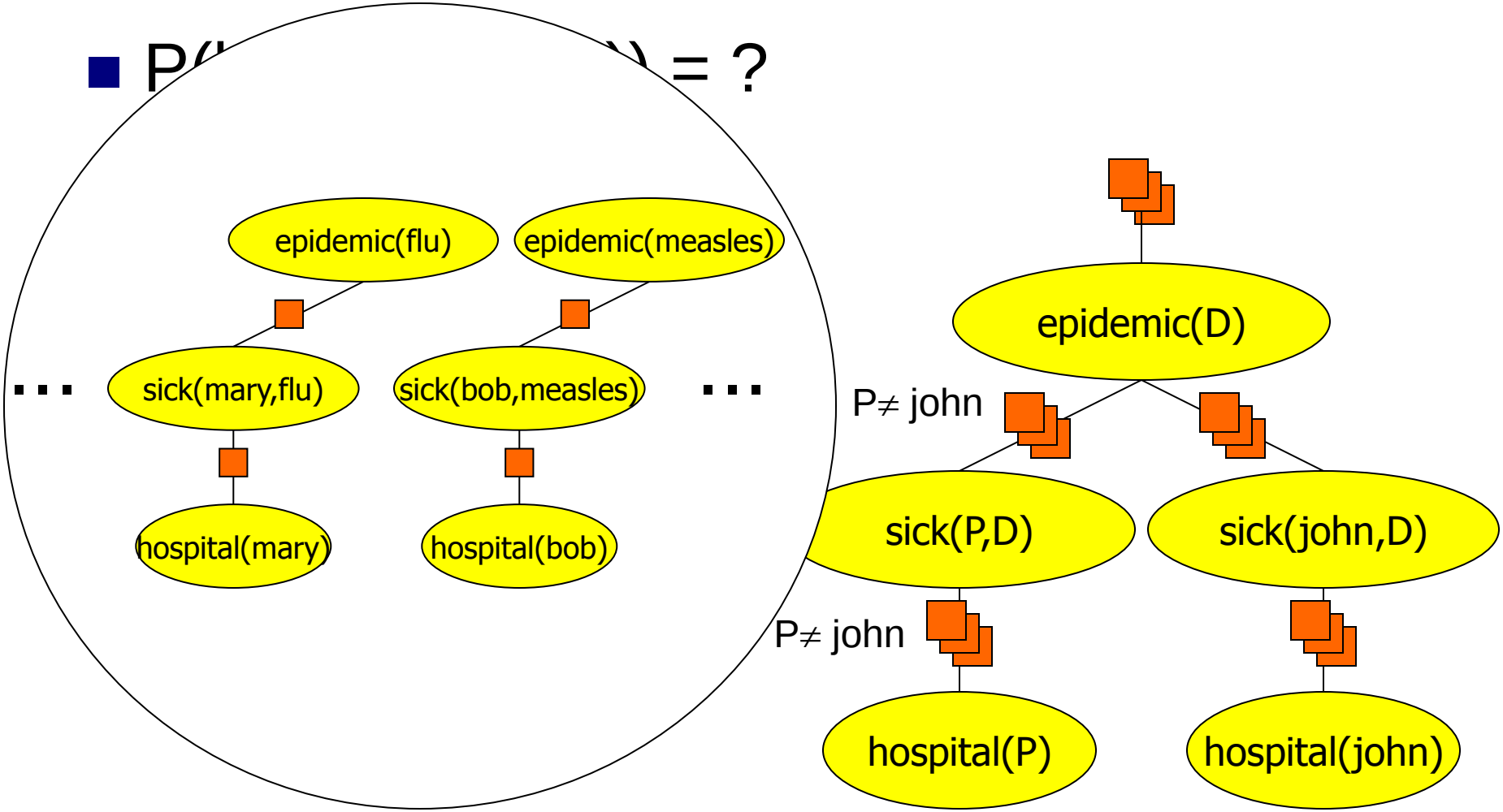
Lifted Inference Example

- P(hospital(john)) = ?



Lifted Inference Example

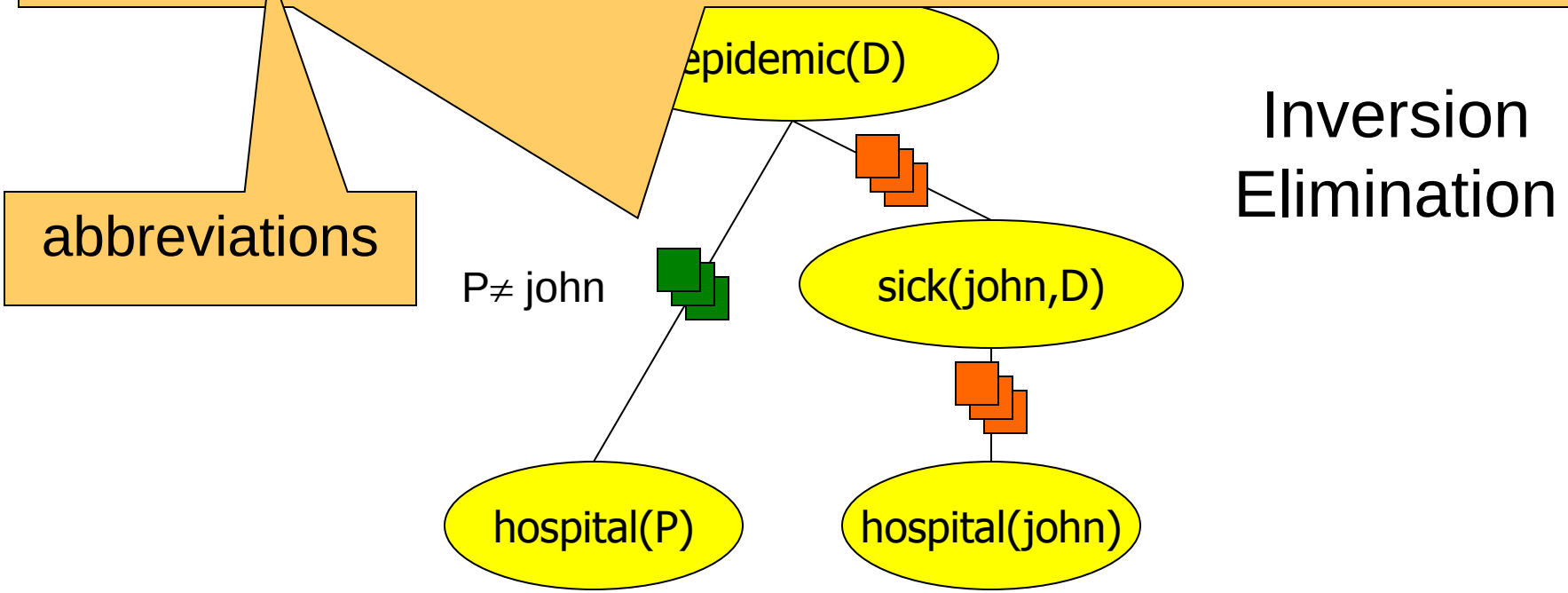
- P( ) = ?



Lifted Inference Example

- P(hospital(john)) = ?

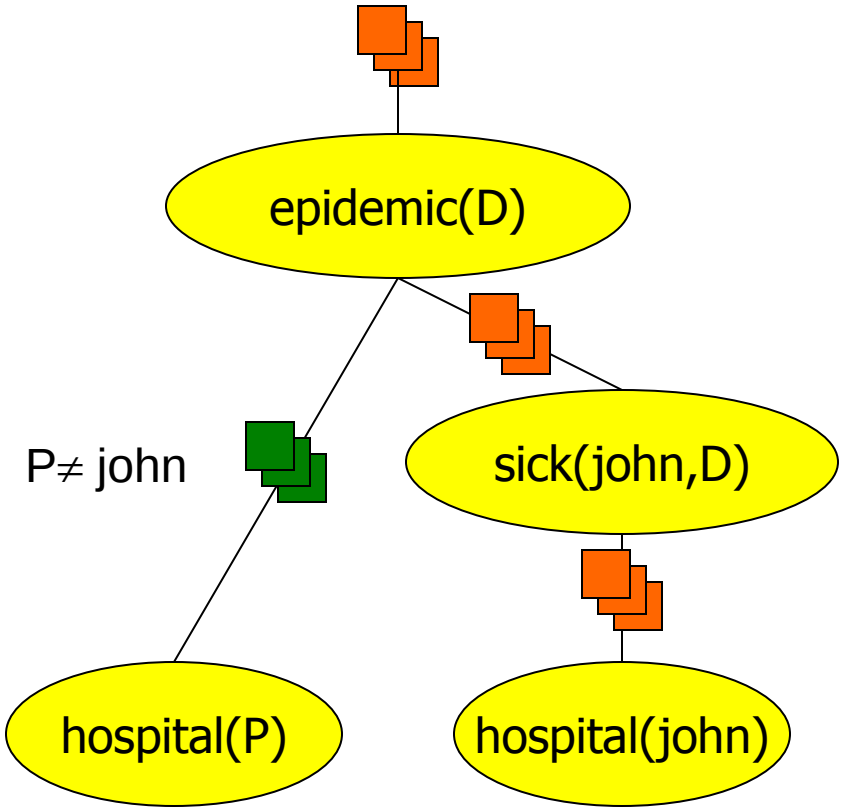
$$\phi'(e(D),h(P)) = \sum_{s(P,D):P \neq \text{john}} \phi_1(s(P,D),e(D),)\ \phi_2(,s(P,D),h(P))$$



Inversion Elimination

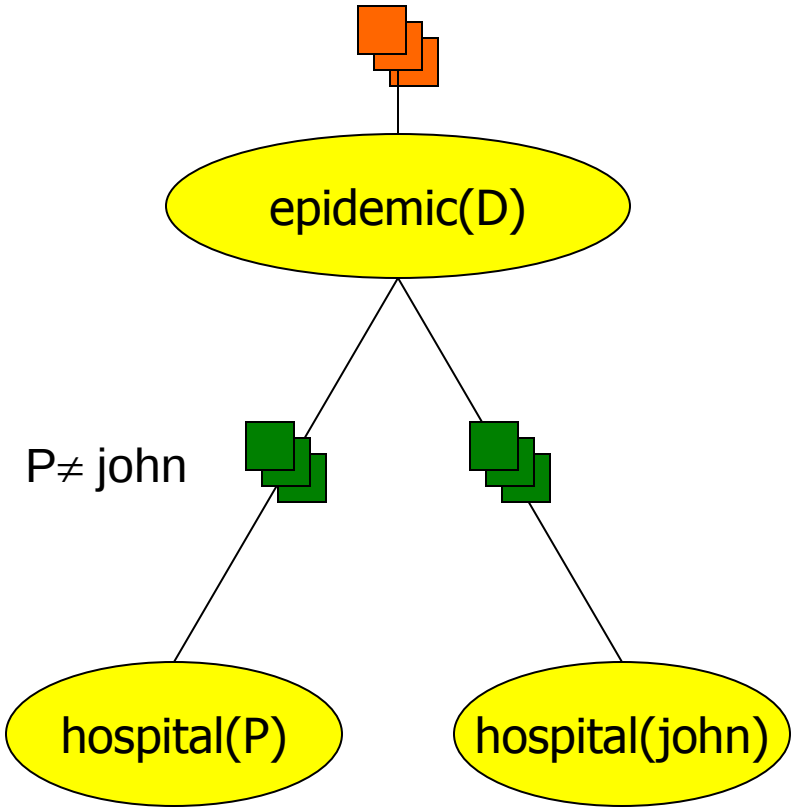
Lifted Inference Example

- P(hospital(john)) = ?



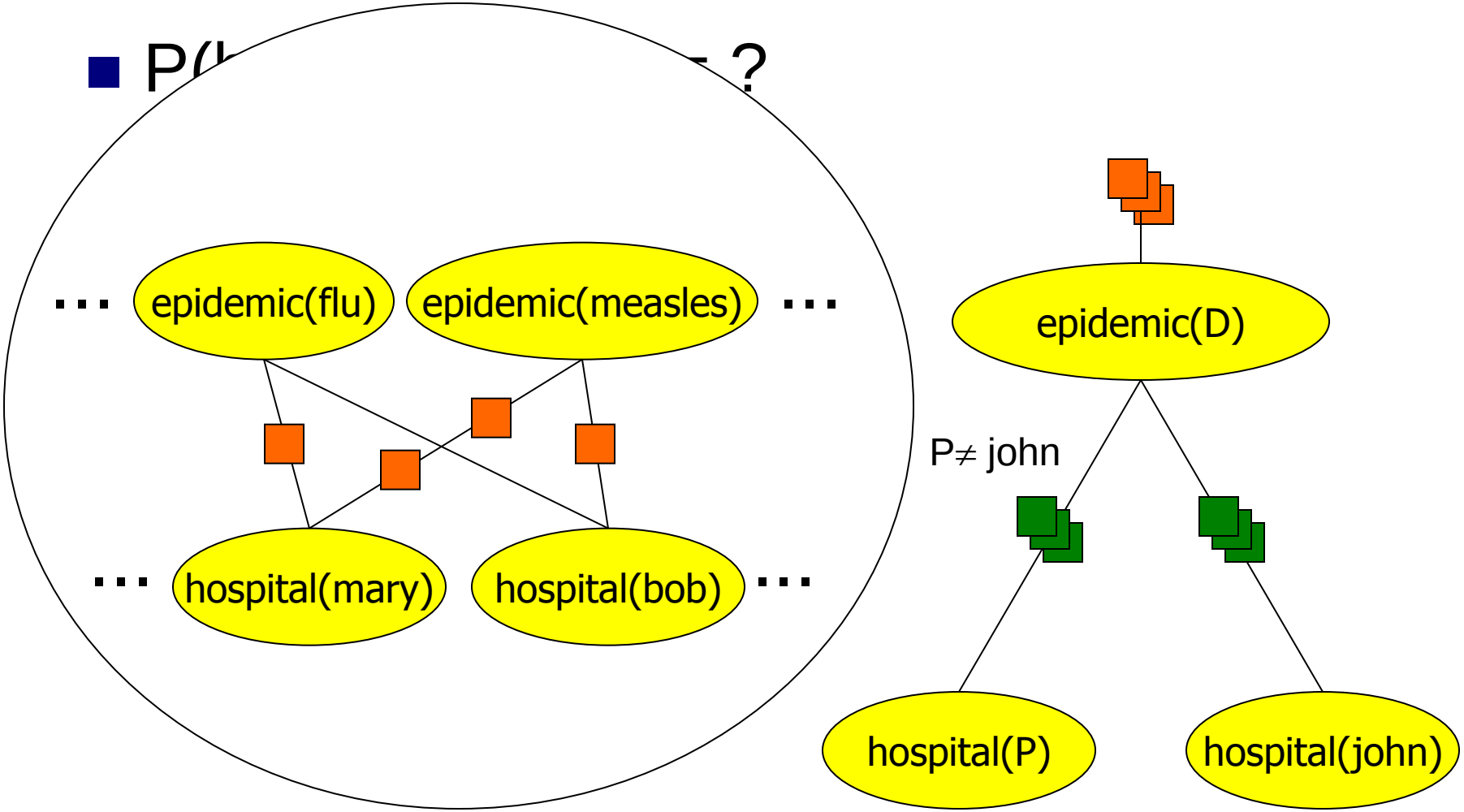
Lifted Inference Example

- P(hospital(john)) = ?



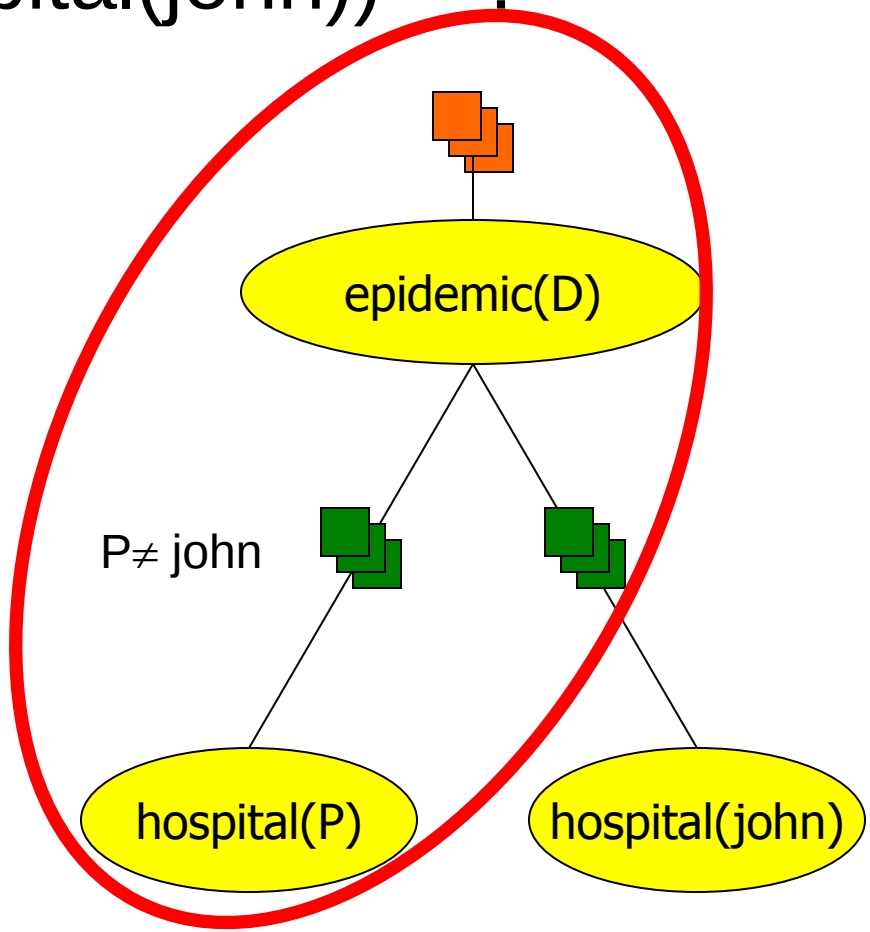
Lifted Inference Example

- P(



Lifted Inference Example

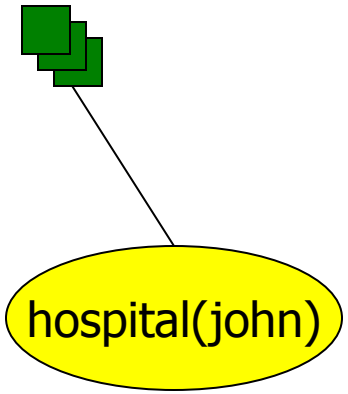
- P(hospital(john)) = ?



Lifted Inference Example

- P(hospital(john)) = ?

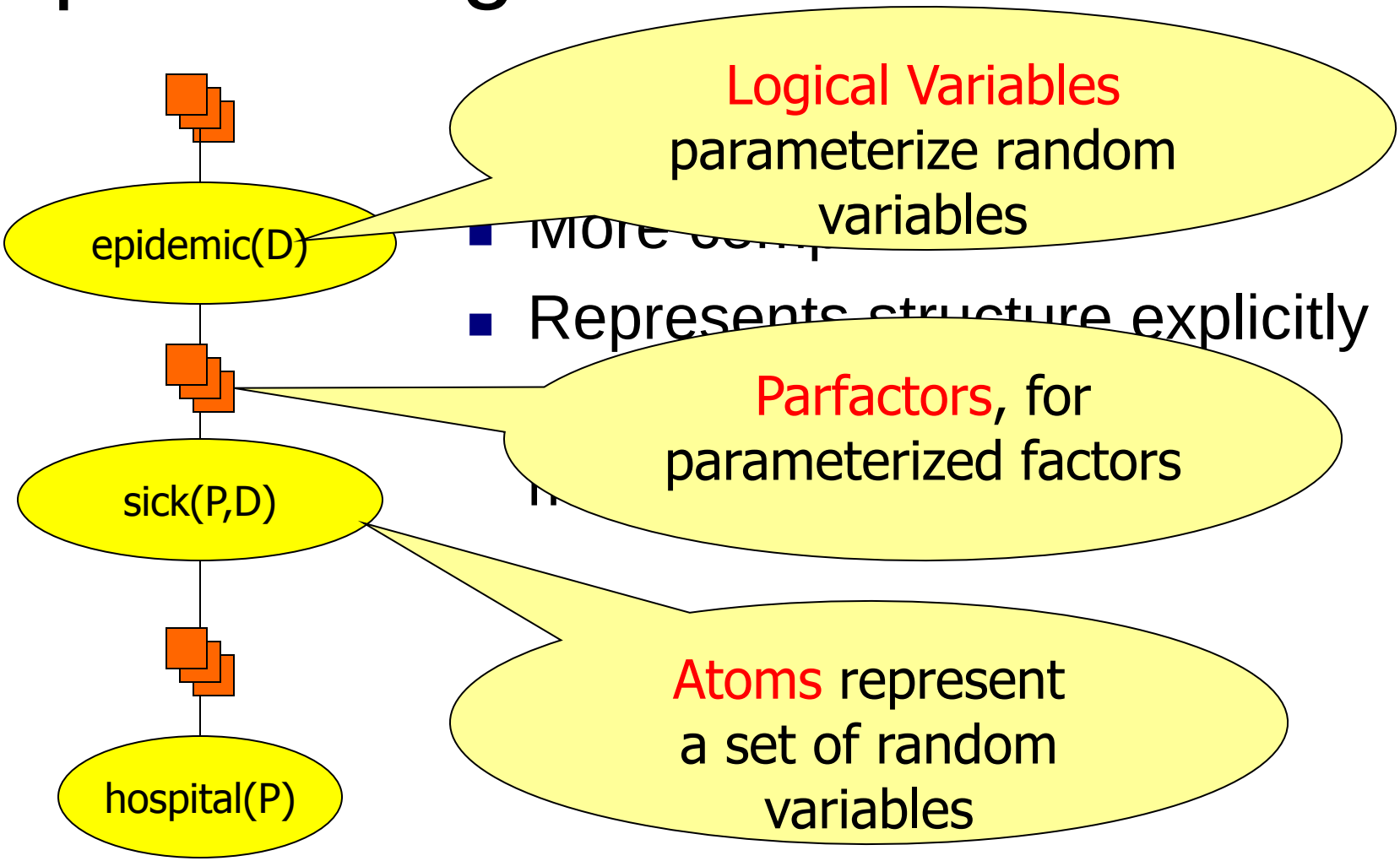
Counting
Elimination



Outline

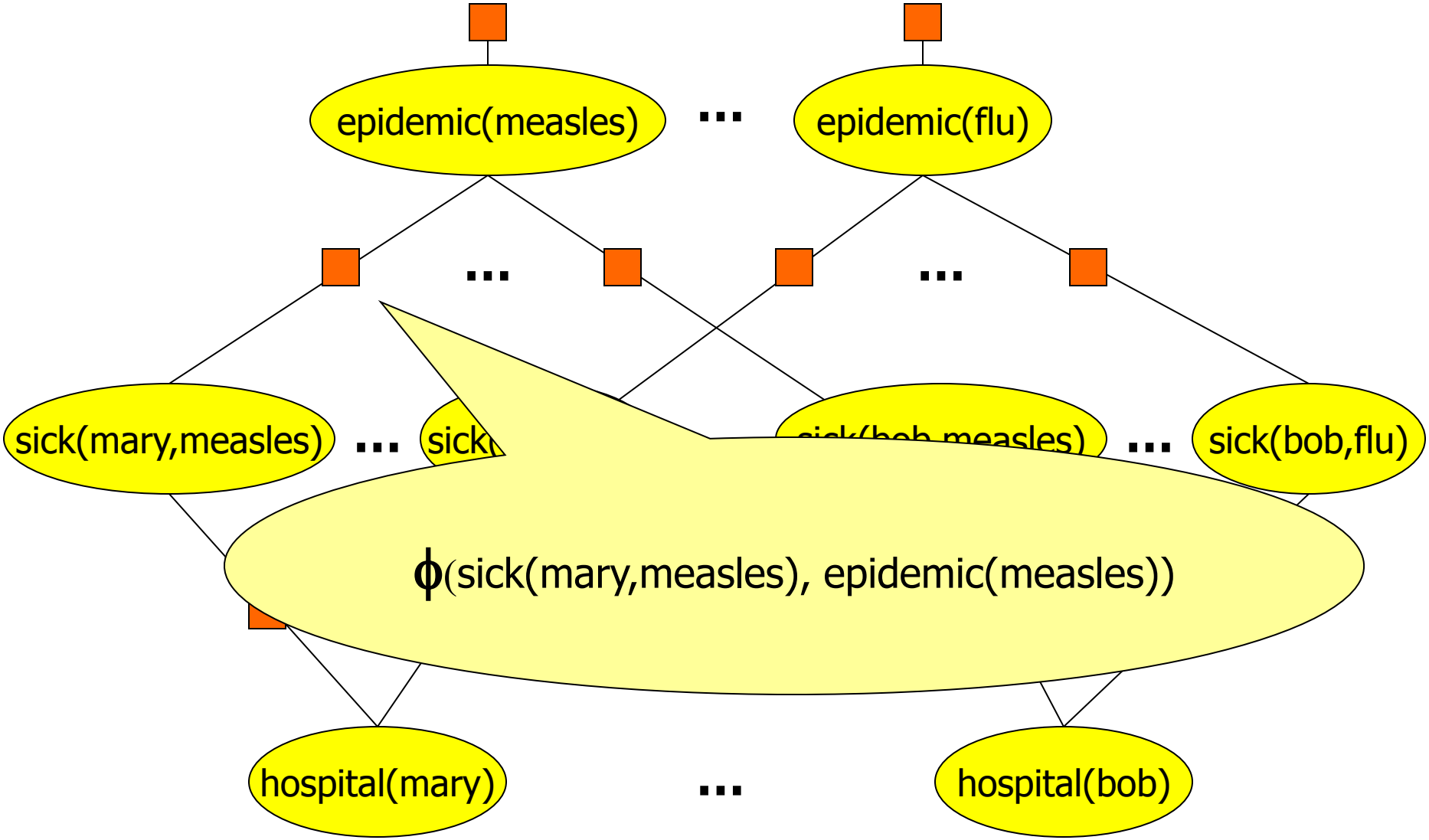
- Motivation
  - Brief background
  - Lifted Inference example
- First-order representations
- Lifted First-Order Probabilistic Inference
  - Inversion Elimination
  - Counting Elimination
  - Partial Inversion
- Conclusion

Representing structure



- More comp
- Represents structure explicitly

Semantics


epidemic(measles)
...
epidemic(flu)
...
...
sick(mary,measles)
...
sick(bob,flu)
ϕ(sick(mary,measles), epidemic(measles))
hospital(mary)
...
hospital(bob)

Making use of structure in Inference

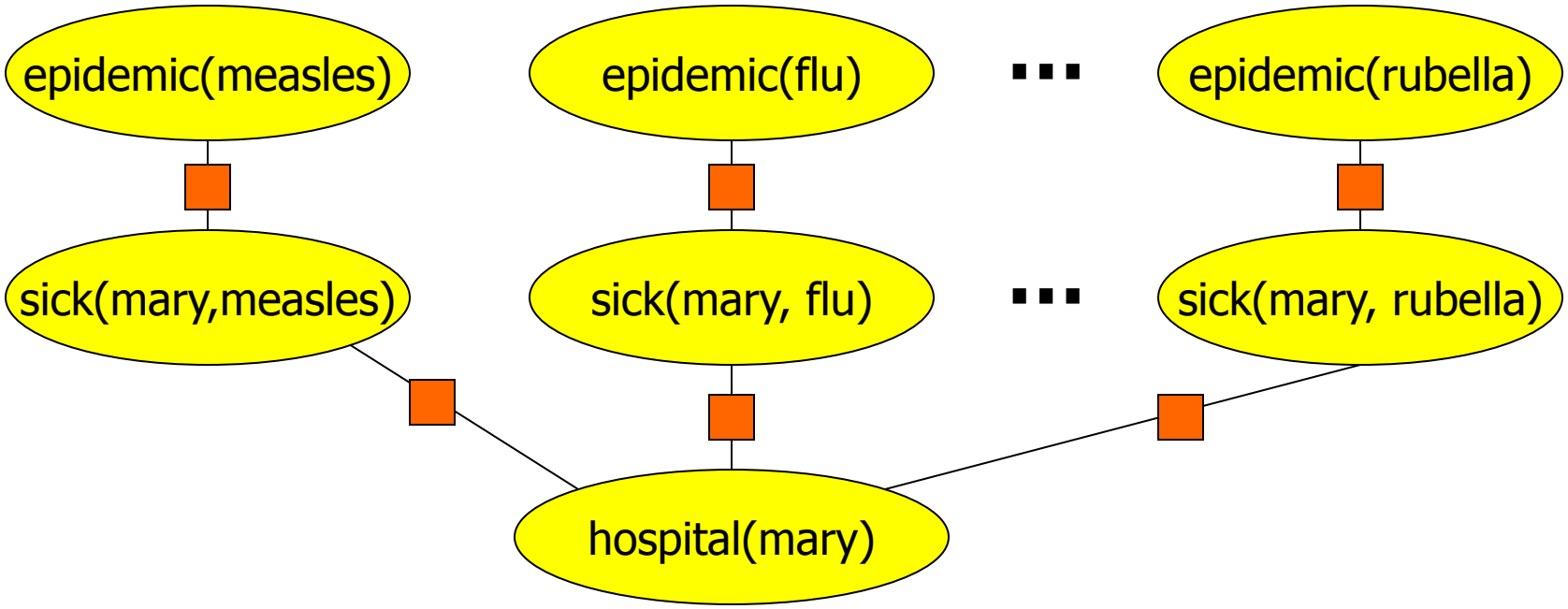
- Task: calculate marginals and posteriors:
  P(sick(bob, measles) | sick(mary,measles)) = ?
- Three approaches
  - plain propositionalization
  - dynamic construction (“smart” propositionalization)
  - lifted inference

Inference - Plain Propositionalization

- Instantiation of potential function for each instantiation
- Lots of redundant computation
- Lots of unnecessary random variables

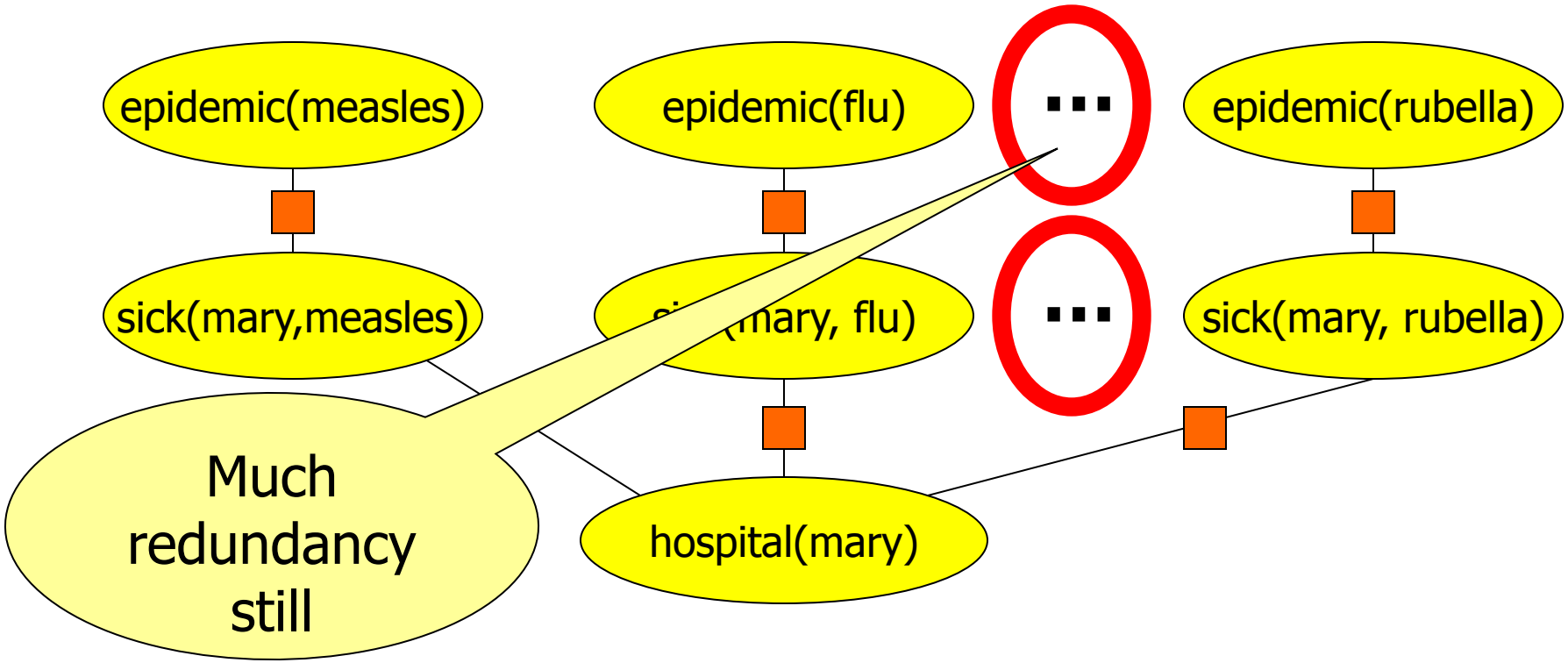
Inference - Dynamic construction

- Instantiation of potential function for relevant parts
- P(hospital(mary) | no epidemics) = ?



Inference - Dynamic construction

- Instantiation of potential function for relevant parts
- P(hospital(mary) | no epidemics) = ?



Inference - Dynamic construction

Used to be the most common approach for exact First-order Probabilistic inference:

- Knowledge-based model construction (Breese, 1992)
- Probabilistic Logic Programming (Ng & Subrahmanian, 1992)
- Probabilistic Logic Programming (Ngo and Haddawy, 1995)
- Probabilistic Relational Models (Friedman et al., 1999)
- Relational Dependency Networks (Neville & Jensen, 2004)
- Relational Bayesian networks (Jaeger, 1997)
- Bayesian Logic Programs (Kersting & DeRaedt, 2001)
- MEBN (Laskey, 2004)
- Markov Logic Networks (Richardson & Domingos, 2004)

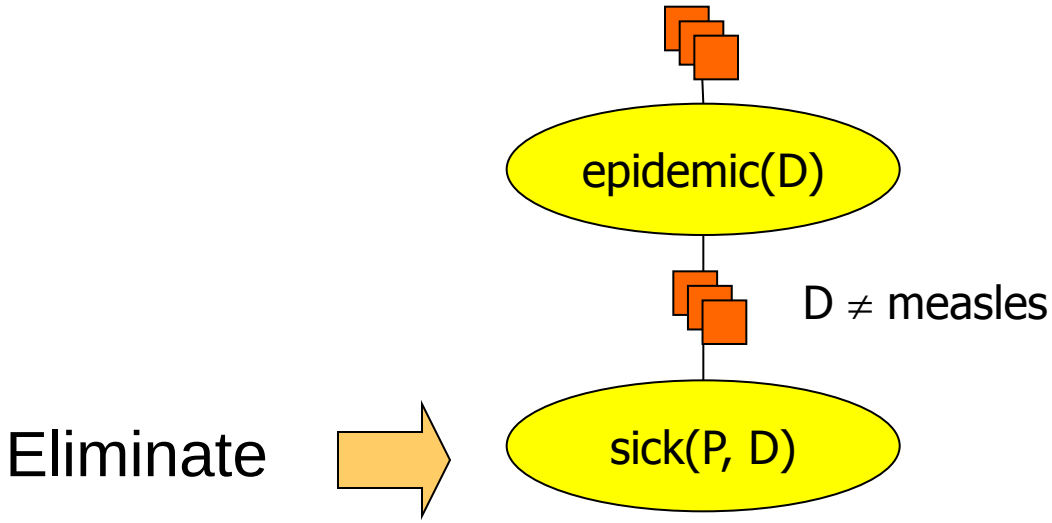
Inference - Lifted inference

- Inference on parameterized, or first-order, level;
- Performs certain inference steps once for a class of random variables;
- Poole (2003) describes a generalized Variable Elimination algorithm which we call Inversion Elimination.
- We formalized Inversion Elimination, showed its limitations and introduced Counting Elimination (IJCAI'05) and Partial Inversion (AAAI'06)

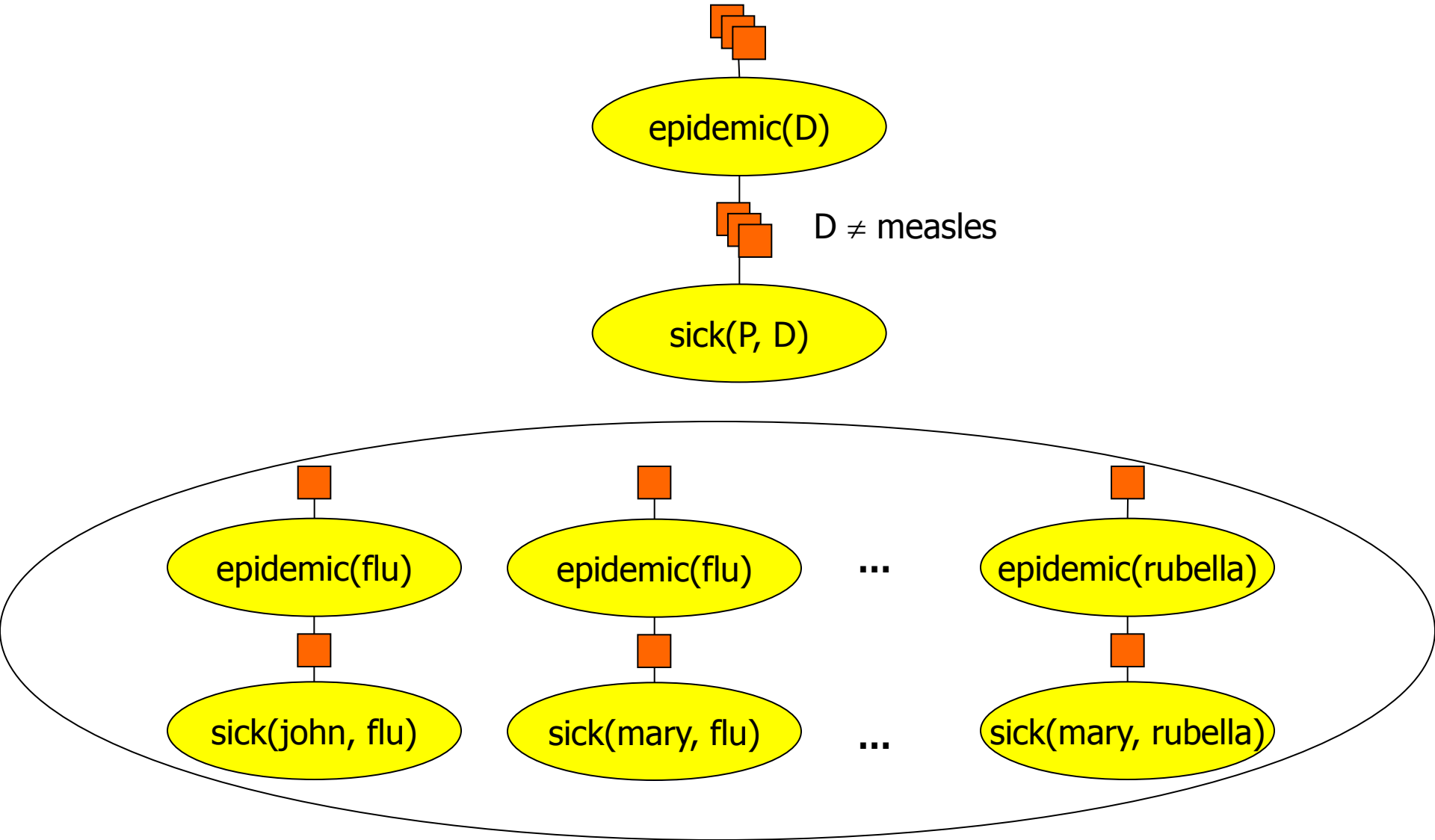
Outline

- Motivation
  - Brief background
  - Lifted Inference example
- First-order representations
- Lifted First-Order Probabilistic Inference
  - Inversion Elimination
  - Counting Elimination
  - Partial Inversion
- Conclusion

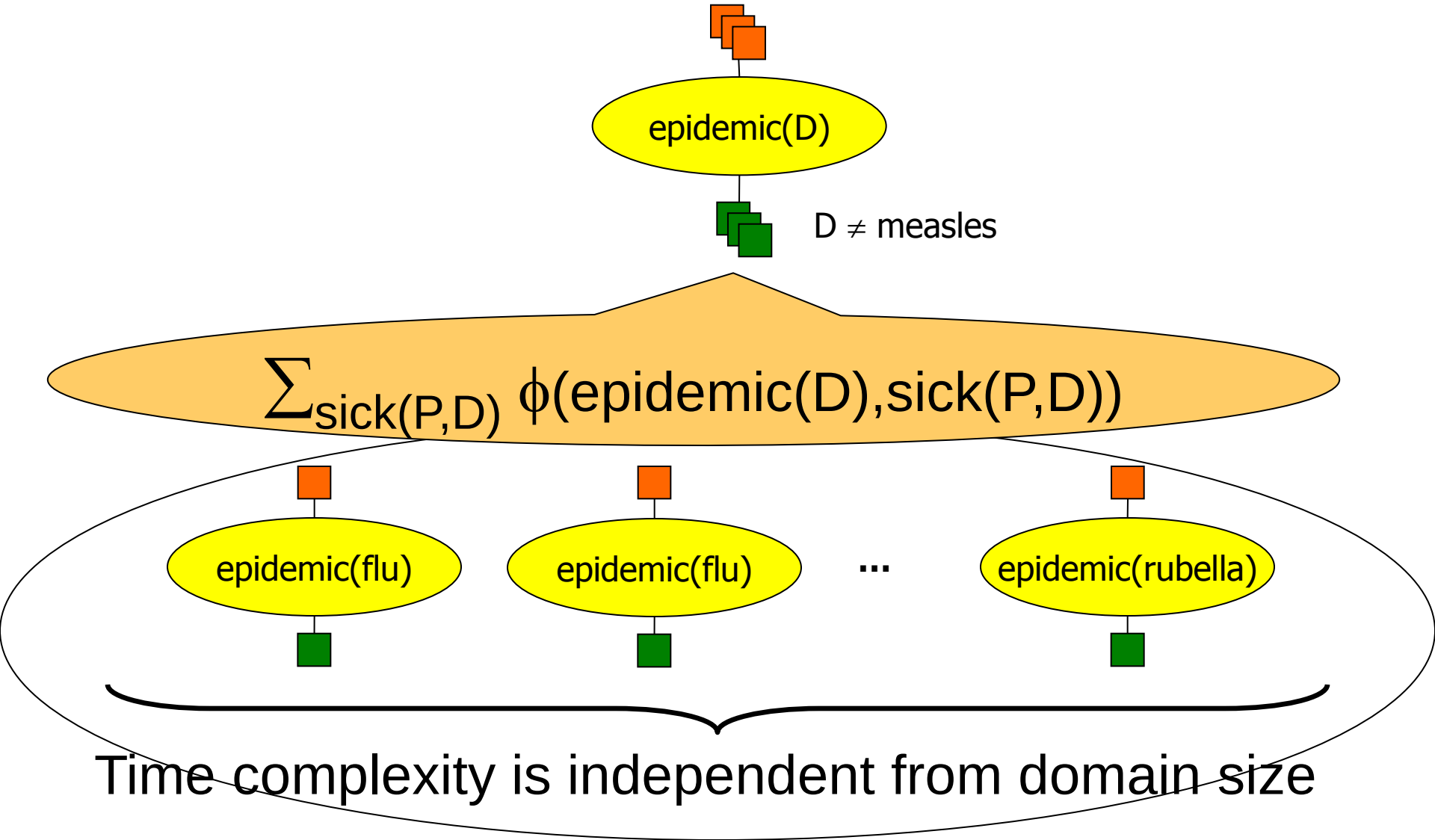
Inversion Elimination (IE)



Inversion Elimination (IE)

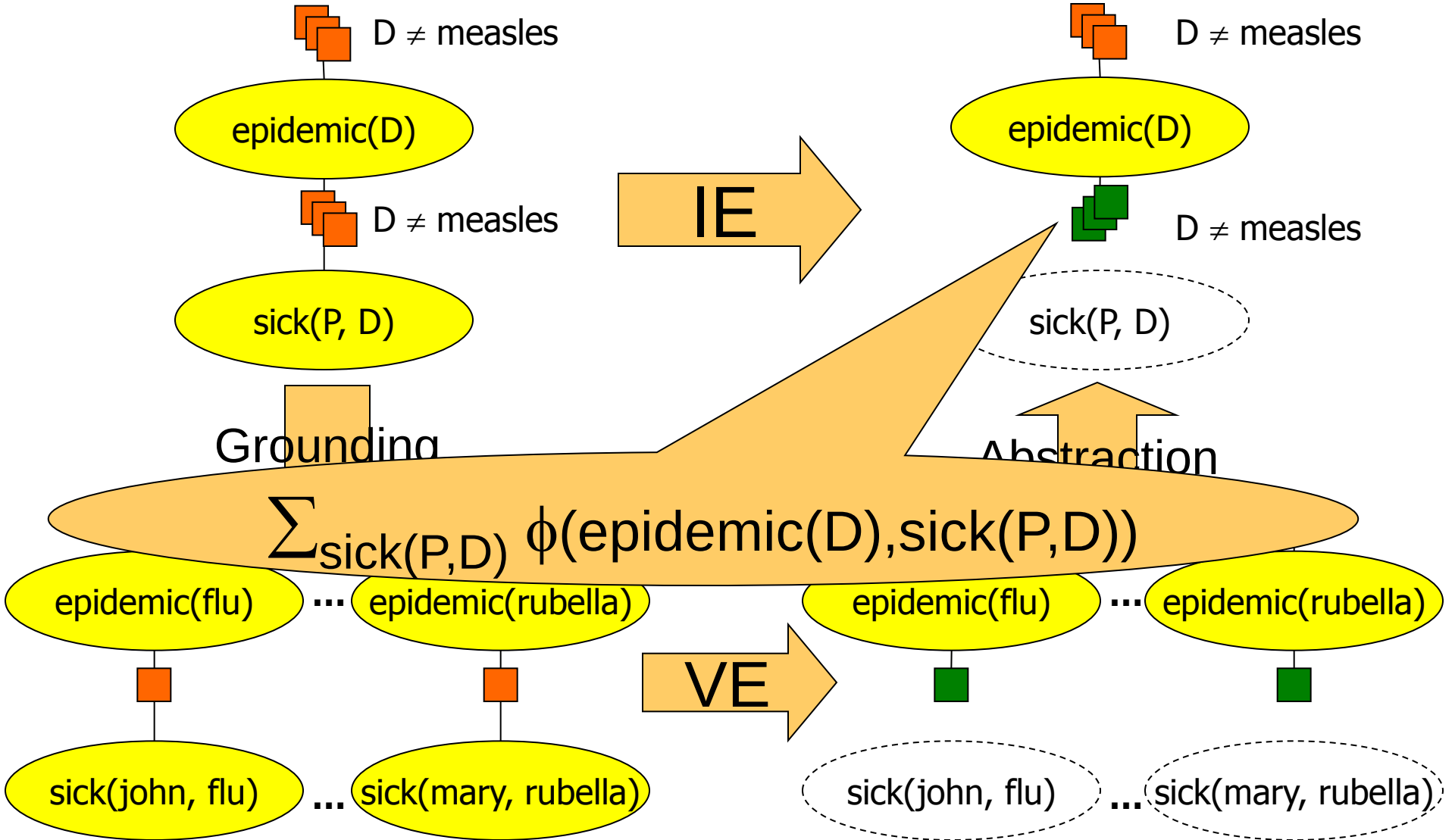


Inversion Elimination (IE)



Time complexity is independent from domain size

Inversion Elimination (IE)



Formalization of IE

- Joint distribution

$$\prod_D \phi_1(e(D)) \prod_{P,D} \phi_2(e(D),s(P,D))$$

- Marginalization by eliminating class s(P,D):

$$\sum_{s(\cdot,\cdot)} \prod_D \phi_1(e(D)) \prod_{P,D} \phi_2(e(D),s(P,D))$$

$$= \prod_D \phi_1(e(D)) \sum_{s(\cdot,\cdot)} \prod_{P,D} \phi_2(e(D),s(P,D))$$

Formalization of IE

$$\sum_{s(\cdot,\cdot)} \prod_{P,D} \phi_2(e(D),s(P,D))$$

$$= \prod_{P,D} \sum_{s(P,D)} \phi_2(e(D),s(P,D))$$

$$= \prod_{P,D} \phi_3(e(D))$$

$$= \prod_{D} \phi_3^{|P|}(e(D))$$

$$= \prod_{D} \phi_4(e(D))$$

Formalization of IE

$$\Sigma_{s(\cdot,\cdot)} \Pi_{P,D} \phi_2(e(D),s(P,D))$$

$$= \Sigma_{s(p1,d1)} \Sigma_{s(p1,d2)} \cdots \Sigma_{s(pn,dm)} \phi_2(e(d_1),s(p_1,d_1)) \phi_2(e(d_1),s(p_1,d_2)) \ldots \phi_2(e(d_n),s(p_n,d_m))$$

$$= (\Sigma_{s(p1,d1)} \phi_2(e(d_1),s(p_1,d_1))) \ldots (\Sigma_{s(p1,d2)} \phi_2(e(d_1),s(p_1,d_2))) \ldots (\Sigma_{s(pn,dm)} \phi_2(e(d_n),s(p_n,d_m)))$$

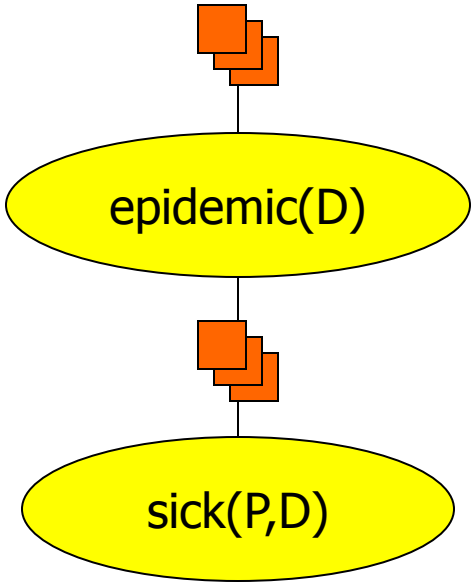
$$= \Pi_{P,D} \Sigma_{s(P,D)} \phi_2(e(D),s(P,D))$$

Formalization of IE

- Inversion Elimination requires certain conditions for its application:
  - Eliminated atom must contain all logical variables in parfactor involved;
  - Eliminated atom instances must not occur together in the same instance of parfactor.
- These are both conditions for parallelism.

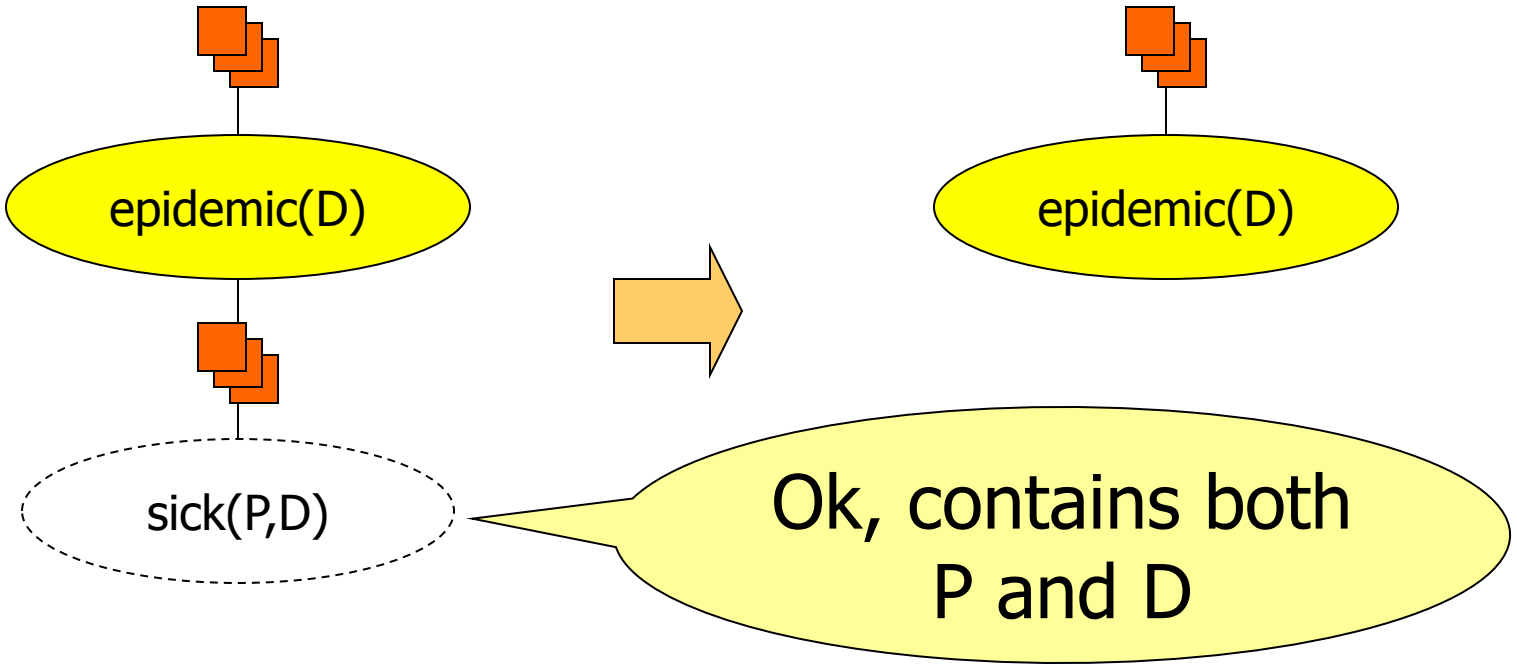
Inversion Elimination - Limitations - I

- Eliminated atom must contain all logical variables in parfactors involved.



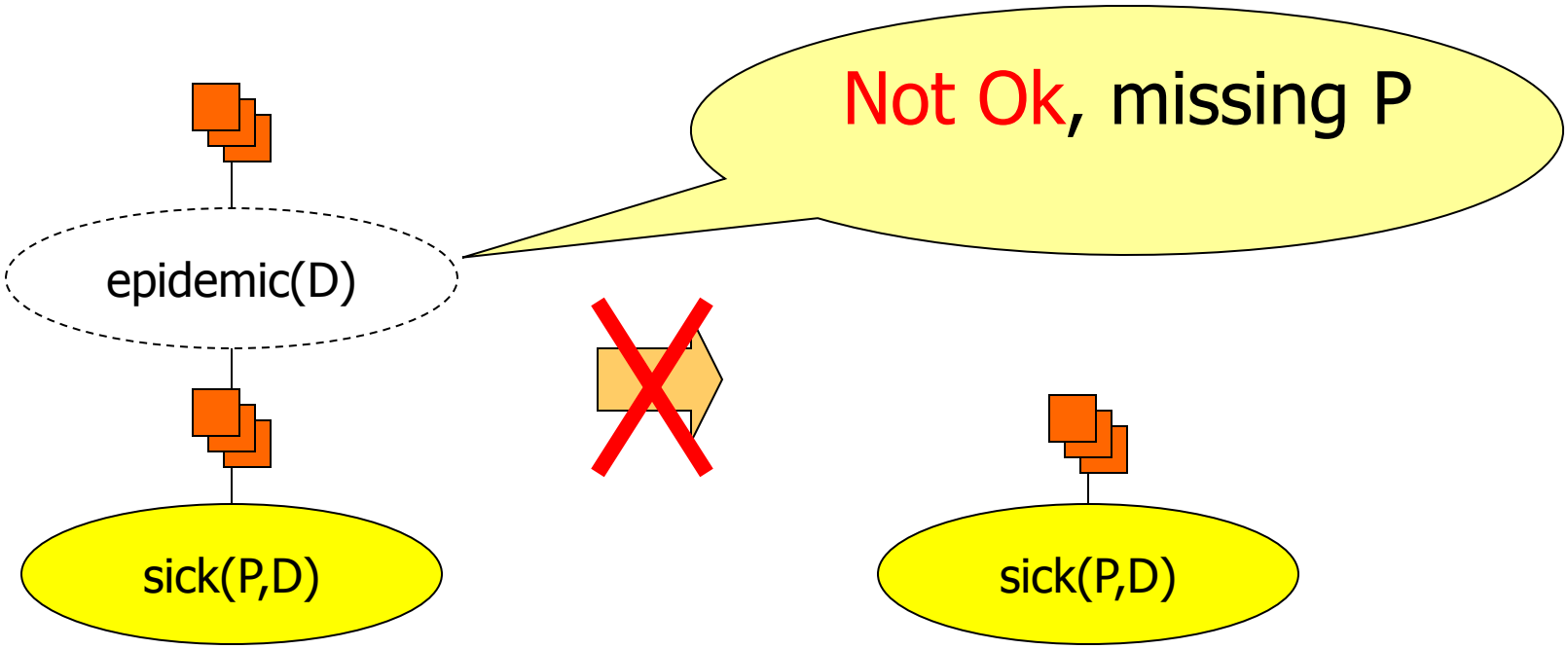
Inversion Elimination - Limitations - I

- Eliminated atom must contain all logical variables in parfactors involved.



Inversion Elimination - Limitations - I

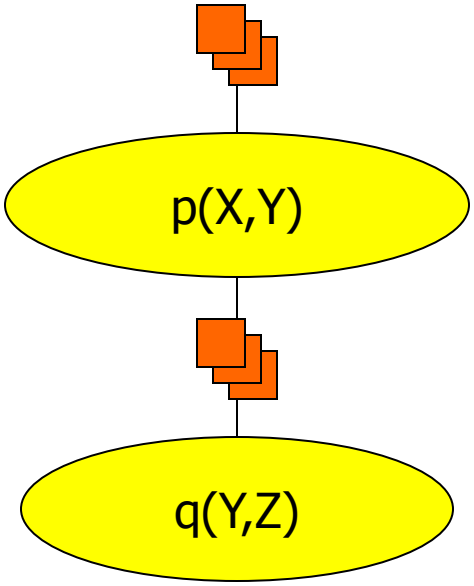
- Eliminated atom must contain all logical variables in parfactors involved.



Inversion Elimination - Limitations - I

- Eliminated atom must contain all logical variables in parfactors involved.

No atom can be eliminated



Inversion Elimination - Limitations - I

- Marginalization by eliminating class e(D):

$$\sum_{e(.)} \prod_{D,P} \phi_2(e(D),s(P,D))$$

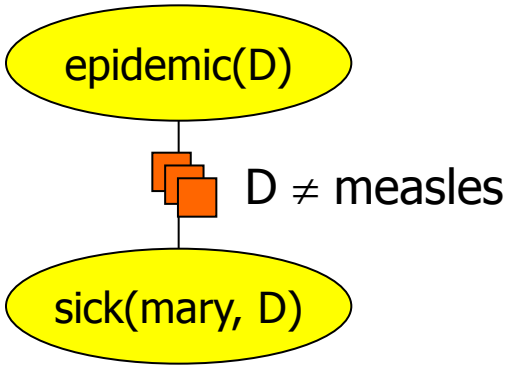
$$= \sum_{e(d1)} \cdots \sum_{e(dn)} \prod_{P} \phi_2(e(d_1),s(P,d_1)) \cdots \prod_{P} \phi_2(e(d_n),s(P,d_n))$$

$$= \left(\sum_{e(d1)} \prod_{P} \phi_2(e(d_1),s(P,d_1))\right) \cdots \left(\sum_{e(dn)} \prod_{P} \phi_2(e(d_n),s(P,d_n))\right)$$

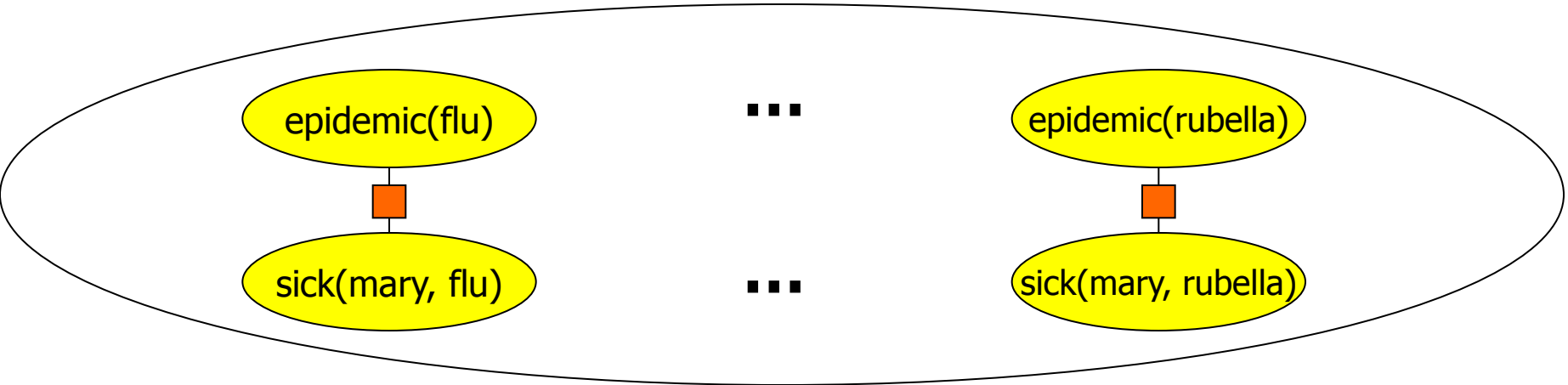
$$= \prod_{D} \sum_{e(D)} \prod_{P} \phi_2(e(D),s(P,D))$$

Inversion Elimination - Limitations - II

- Requires eliminated RVs to occur in separate instances of parfactor

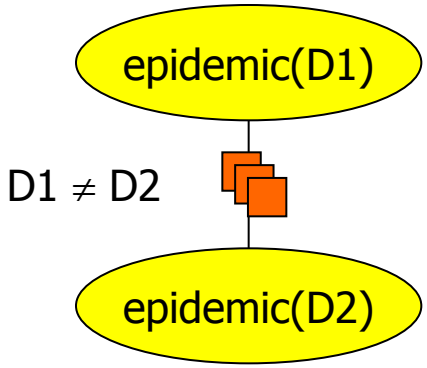


Inversion Elimination Ok

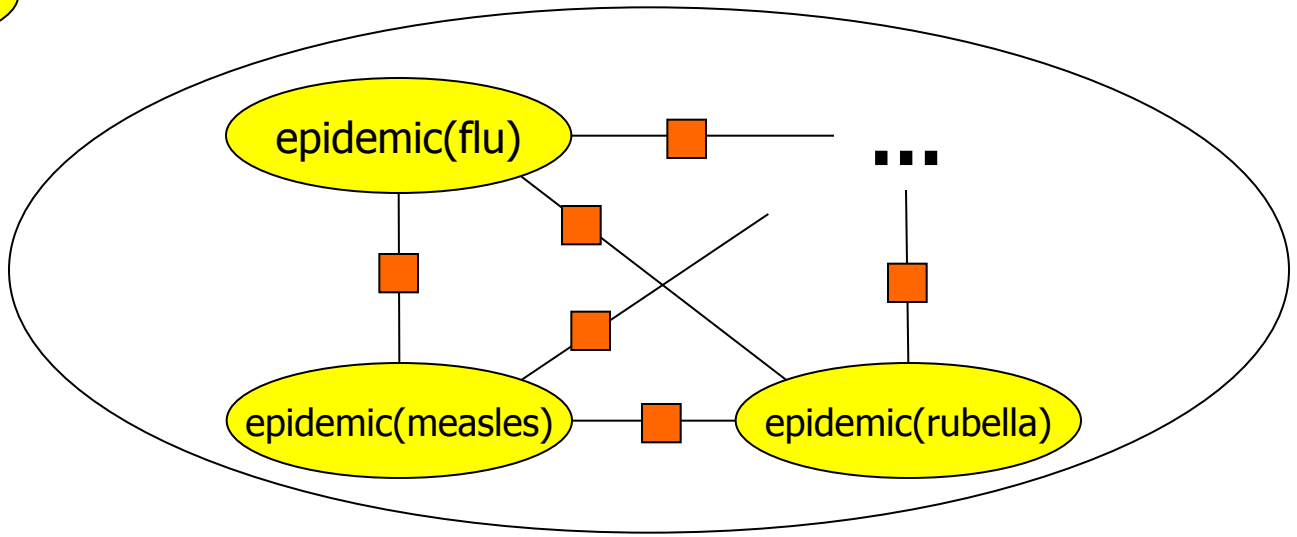


Inversion Elimination - Limitations - II

- Requires eliminated RVs to occur in separate instances of parfactor



Inversion Elimination **Not** Ok



Inversion Elimination - Limitations - II

$$\sum_{e(\cdot)} \prod_{D1 \neq D2} \phi(e(D_1), e(D_2))$$

$$= \sum_{e(d1)} \cdots \sum_{e(dn)} \phi(e(d_1), e(d_2)) \ldots \phi(e(d_{n-1}), e(d_n))$$

$$= \sum_{e(d1)} \phi(e(d_1), e(d_2)) \ldots \sum_{e(dn)} \phi(e(d_{n-1}), e(d_n))$$

Outline

- Motivation
  - Brief background
  - Lifted Inference example
- First-order representations
- Lifted First-Order Probabilistic Inference
  - Inversion Elimination
  - Counting Elimination
  - Partial Inversion
- Conclusion

Counting Elimination

$\sum_{e(\cdot)} \prod_{D1 \neq D2} \phi($

$= \sum_{e(\cdot)} \phi$

$\phi(1,$

$\phi(1,1)$

$= \sum_{e(.)} \prod_{v} \phi(v)^{\#v \text{ in } e(.), D1 \neq D2}$

$= \sum_{i=0}^{|e(.)|} \binom{|e(.)|}{i} \prod_{v} \phi(v)^{\#v \text{ in } e(.), D1 \neq D2 \text{ (from i)}}$

Does depend on domain size, but not exponentially

#(0,1) = 70 * 30

#(1,1) = 30 * (30 – 1)

Counting Elimination Conditions

- No shared logical variables between atoms, so counting can be done independently

$\phi(\text{epidemic}(D_1, \text{Region}), \text{epidemic}(D_2, \text{Region}))$ is not suitable for Counting elimination.

Outline

- Motivation
  - Brief background
  - Lifted Inference example
- First-order representations
- Lifted First-Order Probabilistic Inference
  - Inversion Elimination
  - Counting Elimination
  - Partial Inversion
- Conclusion

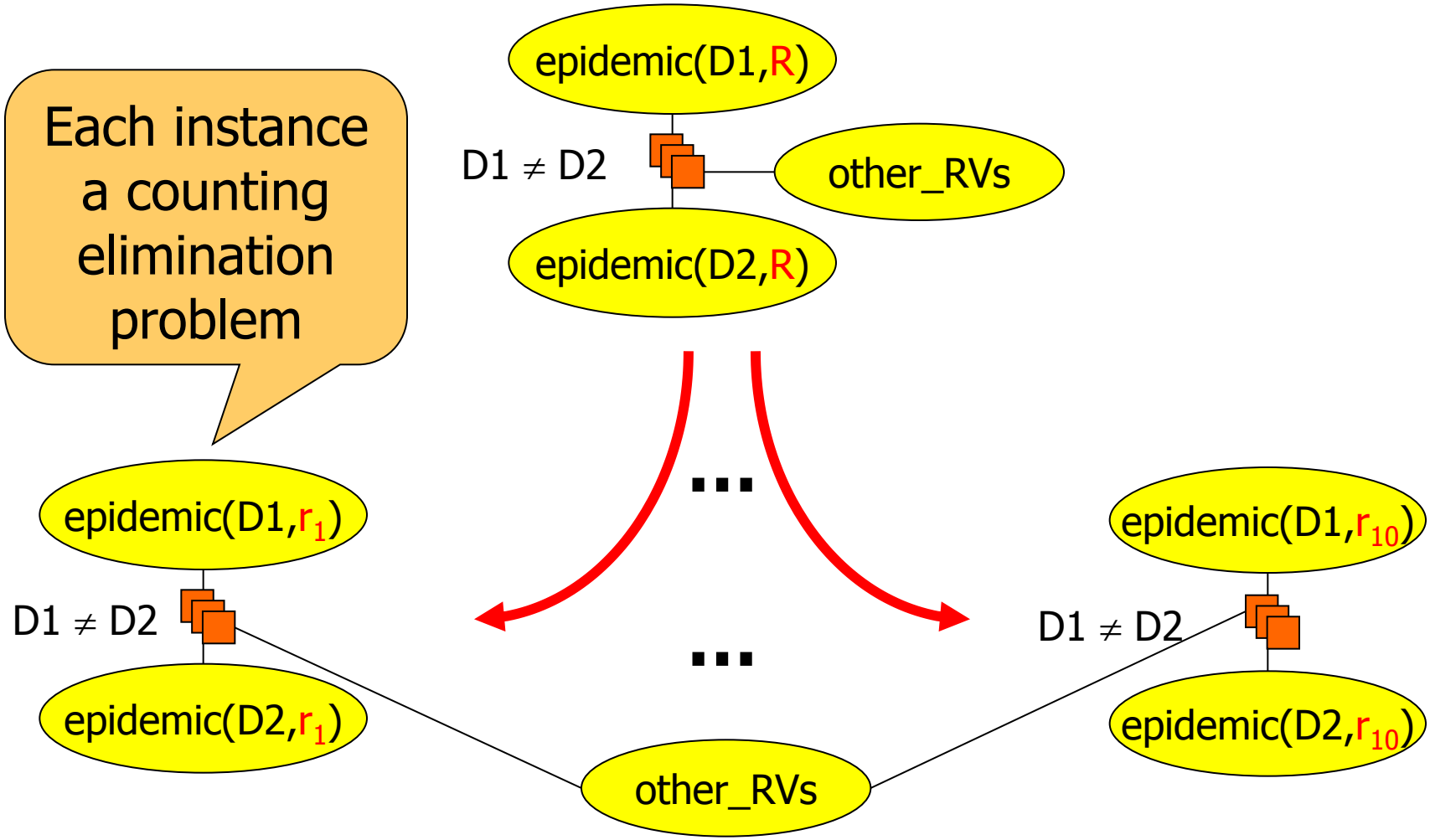
Some things not covered by both Inversion and Counting

- Eliminating epidemic from

  ϕ(epidemic(Disease1,Region),
  epidemic(Disease2,Region),
  other_RVs)

- No atom with all logical variables
  $\Rightarrow$ no Inversion Elimination.
- Shared logical variables
  $\Rightarrow$ no Counting Elimination.

Partial Inversion, graphically



Partial Inversion

$$\sum_{e(.,.)} \prod_{D1 \neq D2, R} \phi(\, e(D_1,R),\, e(D_2,R),\, o\,)$$

$$= \prod_R \sum_{e(.,R)} \prod_{D1 \neq D2} \phi(\, e(D_1,R),\, e(D_2,R),\, o\,)$$

$$= \prod_R \phi'(\, o\,) = \phi'(\, o\,)^{|R|} = \phi''(\, o\,)$$

R is bound
$\Rightarrow$ no shared logical variables
$\Rightarrow$ Counting Elimination

Generalizes both Inversion and Counting Elimination

Partial Inversion Conditions

ϕ(friends(X,Y), friends(Y,X),
sick(X,D), sick(Y,D))

- Cannot partially invert on X,Y because friends(bob,mary) appears in more than one grounding of parfactor.
- But with Partial Inversion we do not need all logical variables in order to invert.

Outline

- Motivation
  - Brief background
  - Lifted Inference example
- First-order representations
- Lifted First-Order Probabilistic Inference
  - Inversion Elimination
  - Counting Elimination
  - Partial Inversion
- Conclusion

Conclusion

- Expressive probabilistic representations used *during* inference as well.
- Lifted inference equivalent to grounded inference; much faster, yet yielding the same exact answer.

Thanks!

Lifted Probabilistic Inference with Counting Formulas

Brian Milch, Luke Zettlemoyer, Kristian Kersting,
Michael Haimes, and Leslie Pack Kaelbling

MIT

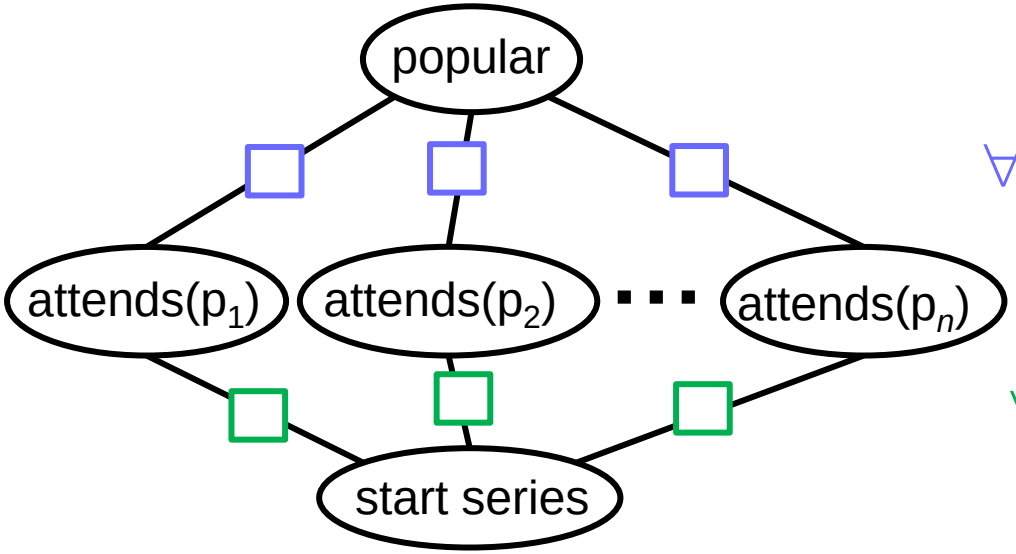
AAAI
July 17, 2008

Lifted Inference

[Pfeffer et al. 1999; Poole 2003; de Salvo Braz et al. 2005]

- Example: Inviting ***n*** people to a workshop

Factor graph



Parfactors

$\forall X.\ \phi_1(\text{popular}, \text{attends}(X))$

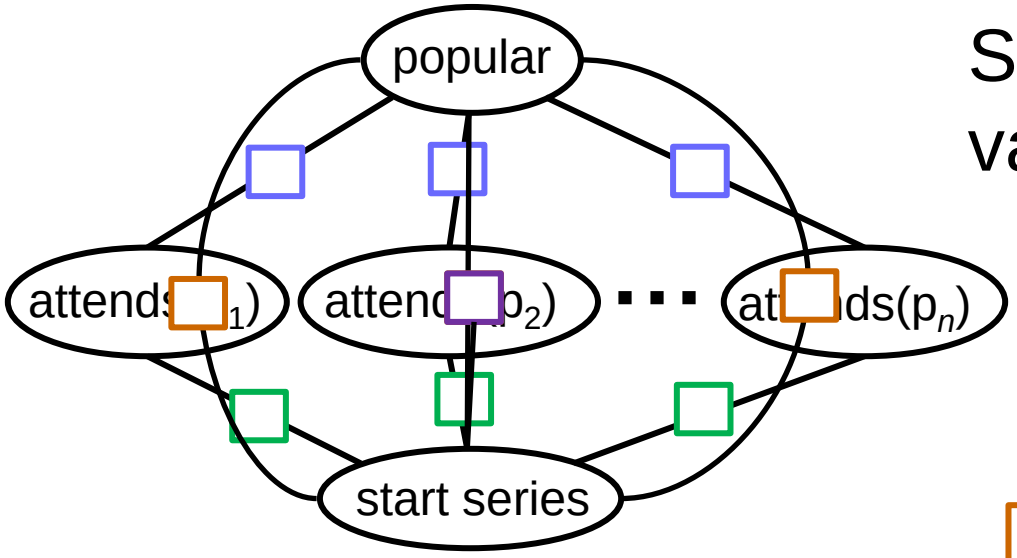
$\forall X.\ \phi_2(\text{attends}(X), \text{series})$

- Lifted inference exploits symmetries revealed by relational model

Outline

- Previous work on exact lifted inference
- Exploiting symmetry within large factors
  - Parfactors defined on counting formulas
  - Inference algorithm that automatically creates counting formulas to represent intermediate results
- Result: asymptotic speed-ups on some problems

Variable Elimination



Sum out non-query variables one by one

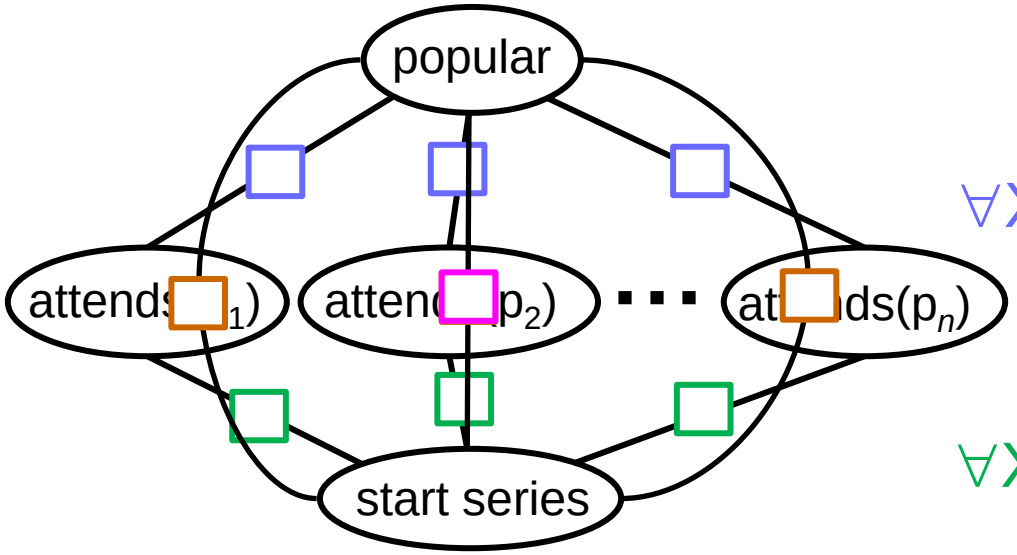
$$\sum_{att(p_1)} \phi_1(pop, att(p_1))\ \phi_2(att(p_1), ser)$$

$$\phi'(pop, ser)$$

Time is linear in number of invitees ***n***

First-Order Variable Elimination

[Poole 2003; de Salvo Braz *et al*. 2005]



$\forall X.\ \phi_1(\text{popular}, \text{attends}(X))$

$\forall X.\ \phi_2(\text{attends}(X), \text{series})$

$\forall X.\ \phi'(\text{popular}, \text{series})$

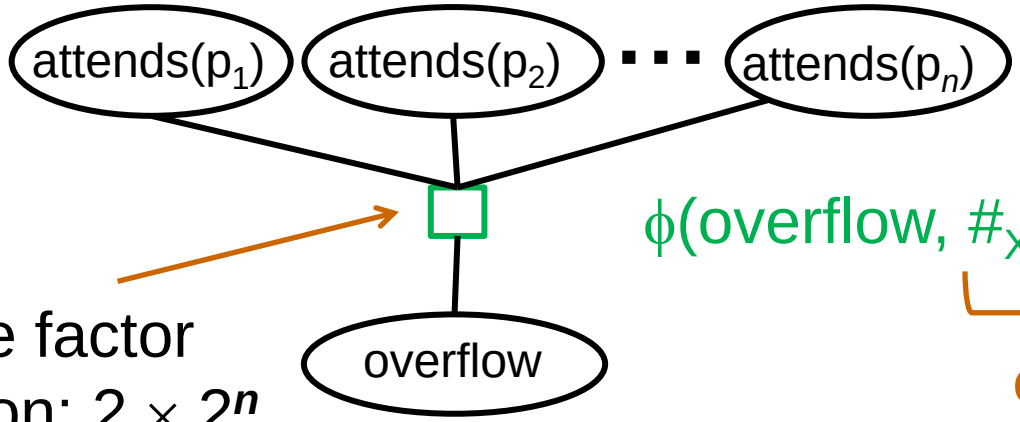
$\phi'(\text{popular}, \text{series})^{n}$

Sum out all attends(X) variables at once

Time is constant in ***n***

Symmetry Within Factors

[cf Zhang & Poole 1996; Gupta et al. 2007]



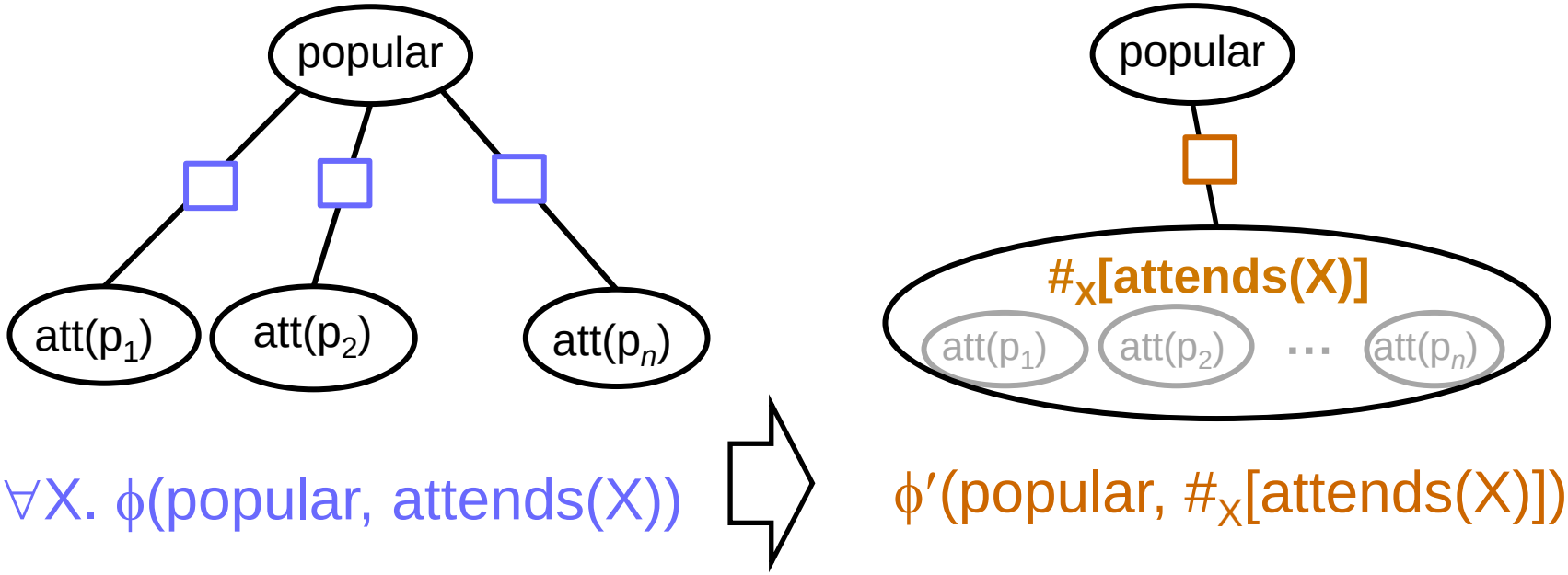
Size of naïve factor representation: 2×2^{n}

ϕ(overflow, $\#_X$[attends(X)])

counting formula

- Values of counting formula are histograms counting how many objects X yield each possible value of attends(X)
  - Only ***n***+1 histograms, e.g., [50, 0], [49, 1], …, [0, 50]
  - Factor size now $2 \times (n+1)$: linear in ***n***

Creating Counting Formulas



$\forall X.\ \phi(\text{popular}, \text{attends}(X))$ ⇨ $\phi'(\text{popular}, \#_X[\text{attends}(X)])$

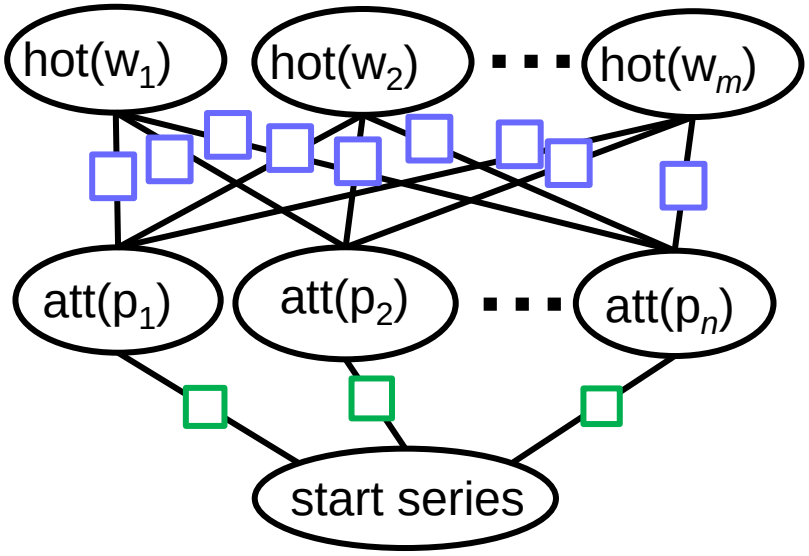
- Convert parfactor to count over X

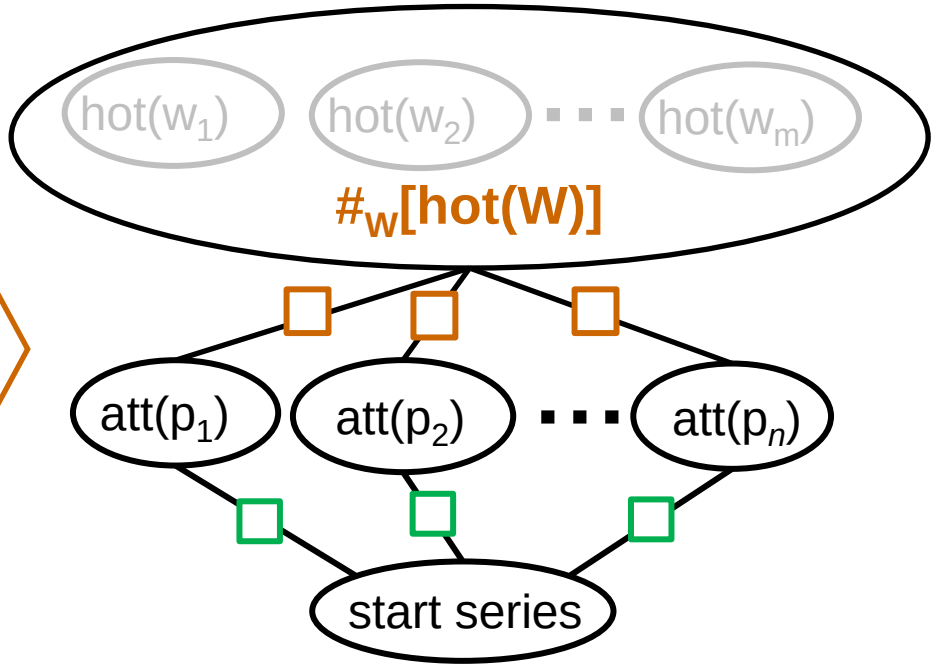
$$\phi'(\text{pop}, [\boldsymbol{k}, \boldsymbol{n}\text{-}\boldsymbol{k}]) = \phi(\text{pop}, \text{true})^{\boldsymbol{k}}\ \phi(\text{pop}, \text{false})^{\boldsymbol{n}\text{-}\boldsymbol{k}}$$

- Can count over any logical variable occuring in just one atom in parfactor

Example: Competing Workshops

$\forall W \; \forall X. \; \phi(\text{hot}(W), \text{att}(X))$





Can't sum out attends(X) without joining all the hot(W) variables

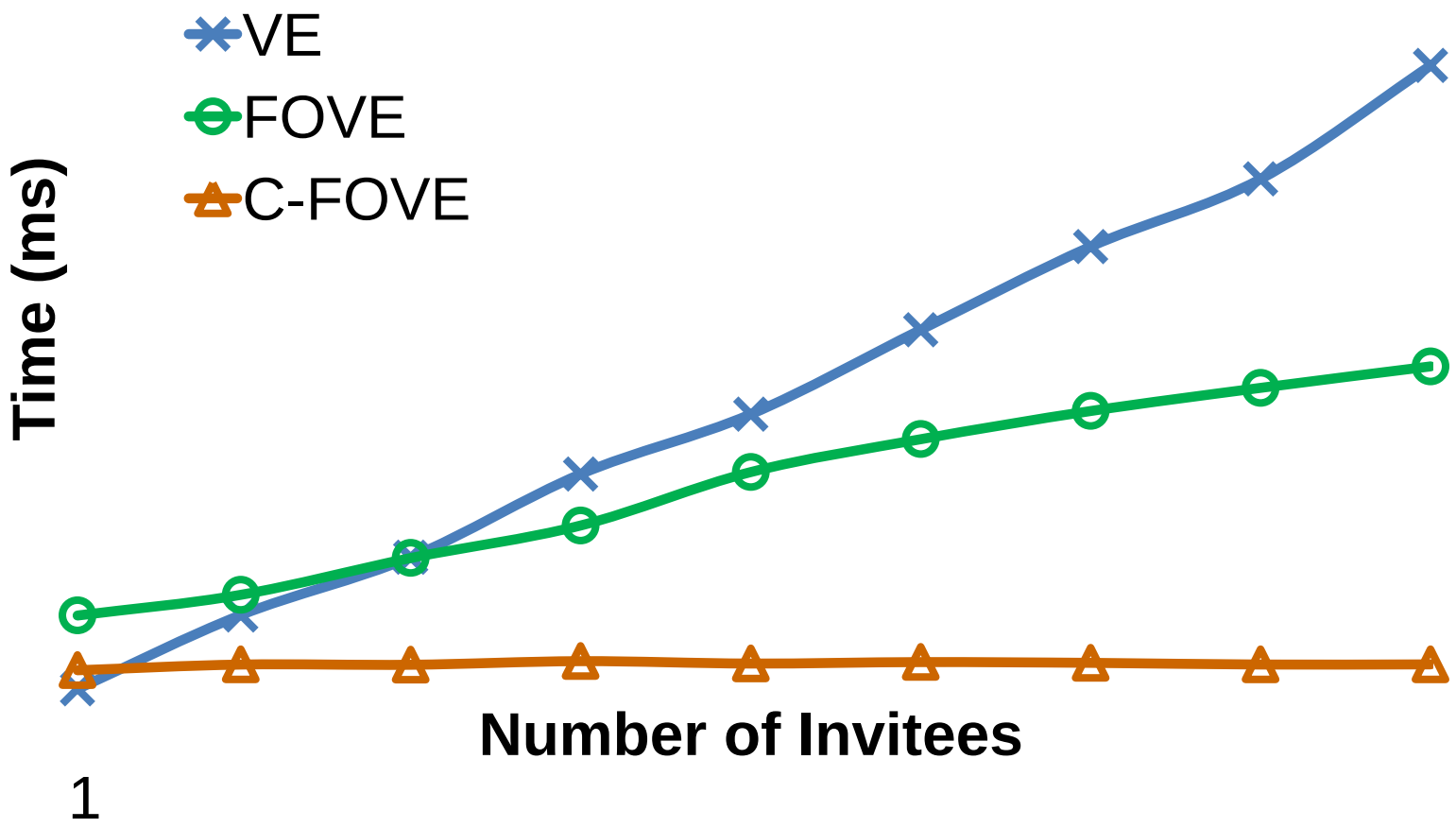
Create counting formula on hot(W), then sum out attends(X) at lifted level

Conversion to counting formulas creates new opportunities for lifted elimination

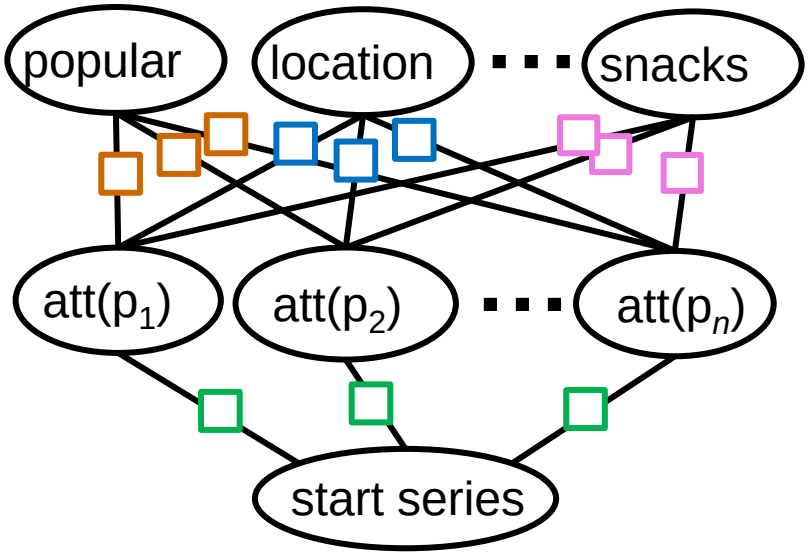
C-FOVE: First-Order VE with Counting Formulas

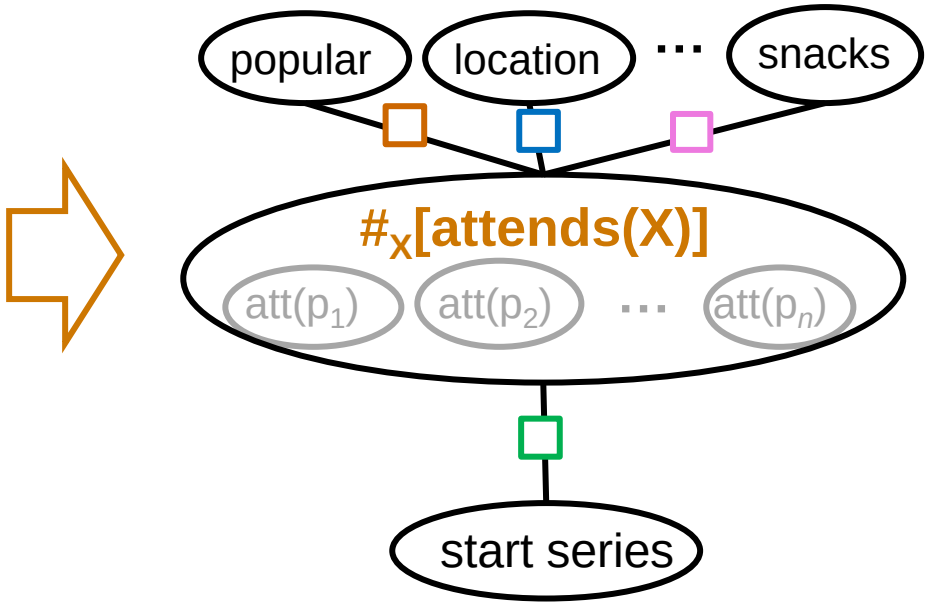
- Maintain set of parfactors
- Sound and complete set of operations
  - Sum out an atom or counting formula
  - Convert a parfactor to count over a variable
  - Propositionalize or expand
- Goal: Apply minimum-cost sequence of operators to sum out non-query RVs
  - Intractable in general
  - Heuristic: choose next operator greedily

Results: Competing Workshops



Example: Workshop Attributes

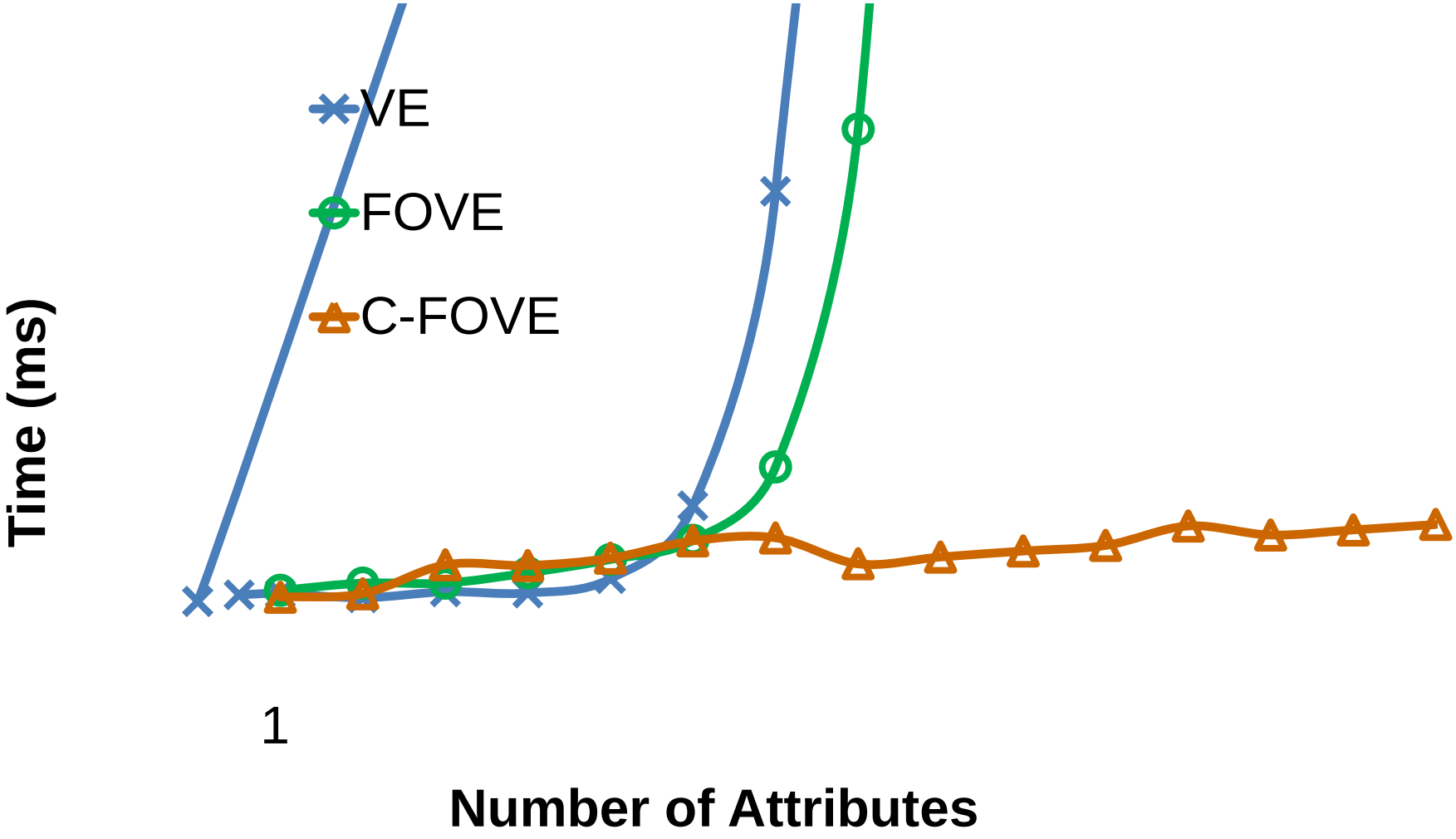




FOVE sums out attends(X) first, creating factor over all attributes

C-FOVE creates counting formula on attends(X), then sums out attributes one by one

Results: Workshop Attributes



Summary

- Parfactors with counting formulas capture symmetries both within and across factors
- C-FOVE algorithm automatically creates counting formulas during inference
  - Asymptotic speed-ups in some cases (linear $\rightarrow$ constant; exponential $\rightarrow$ linear)
  - Distinguishes operator correctness from ordering heuristics

Lifted Inference with Bisimulation

Prithviraj Sen and Lise Getoor

Tutorial on Lifted Inference

International Joint Conference on Artificial Intelligence, 2011

Introduction

- Probabilistic databases
  - Sub-field of Database Systems.
  - Been around since 1990's.
  - Renewed interest since 2004.
- Very powerful, flexible underlying probabilistic model
  - Represents a distribution over many databases.
- Applicable to various real-world applications.
- SQL forms (current) query language of choice.
- Evaluating a query $q \equiv$ evaluating q over many databases.
- Reduces to computing a set of marginal probability distributions.

Information Integration

Employee DB:

Name	Age	Salary
John Smith	39	$1200
Adam Dole	24	$1250
Maddy Bowen	36	$8700
...	...	...

Census DB:

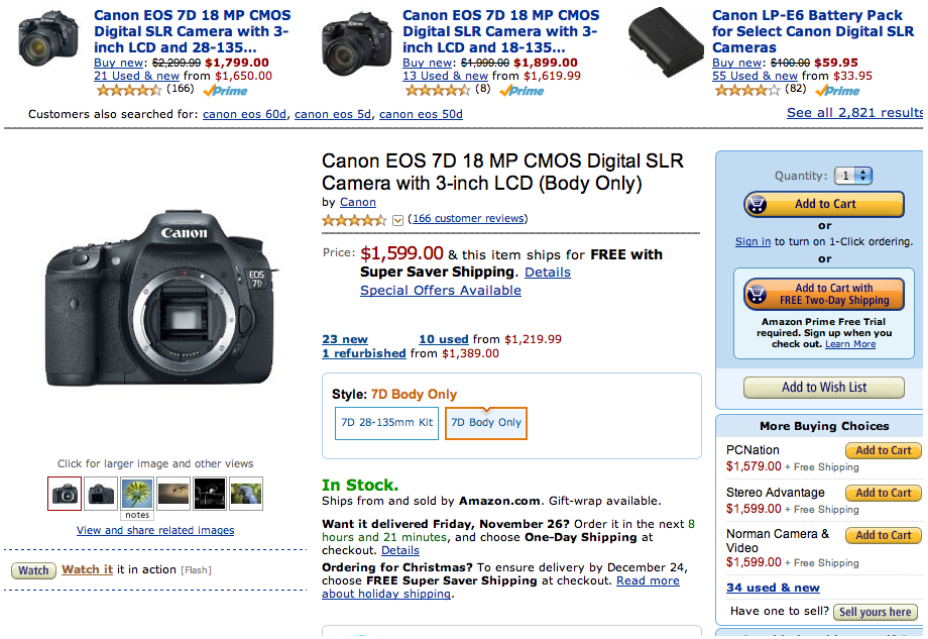
Name	Gender
Johnathan Smith	M
Magdalena Bowen	F
Magda Bowie	F
...	...





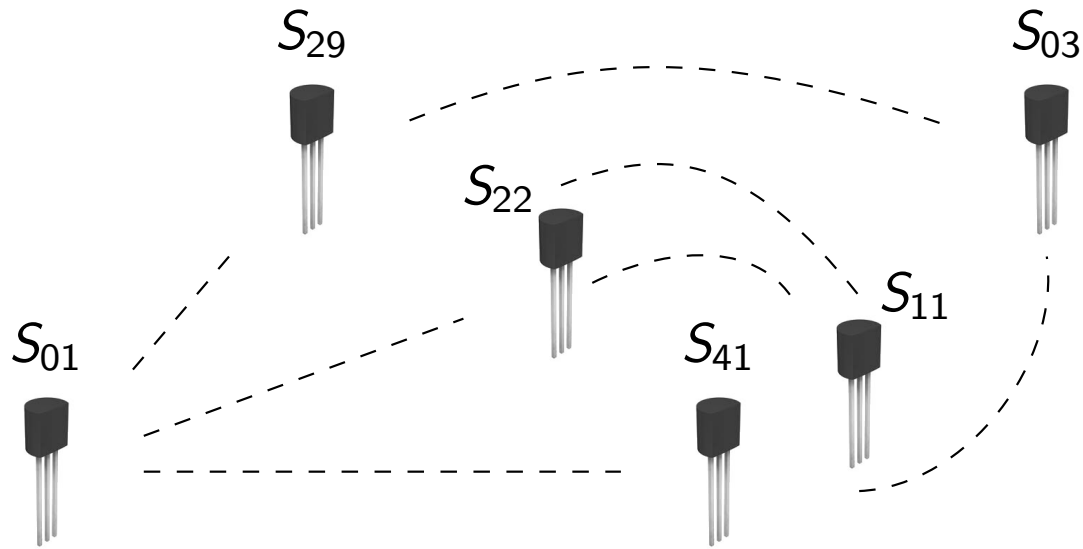
Name	Gender	Age	Salary	
Johnathan Smith	M	39	$1200	0.89
Magdalena Bowen	F	36	$8700	0.95
Magda Bowie	F	36	$8700	0.35
...	...	...	...	...

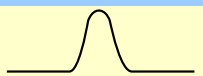
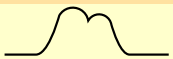
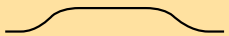
Information Extraction





Sensor Networks



Sensor	Location	Time	Temperature
S_{11}	32°5′N 67°8′E	11:59pm	
S_{22}	33°8′N 66°6′E	12:06pm	?
S_{29}	34°N 65°8′E	12:10pm	
S_{41}	32°3′N 67°4′E	12:01pm	
⋮	⋮	⋮	⋮

Choice of Representation

- Base data is represented using random variables and distributions.
- Usually, there's a choice: tuple-level or attribute-level.
- Probabilistic database represents the (factored) joint distribution.
- Each database is a joint assignment.
- Attribute-level uncertainty produces parsimonious databases.

AdID	Make	Color	Price
1	Honda	?	9,000$

AdID	Make	Color	Price	
1	Honda	Black	9,000$	?
1	Honda	Beige	9,000$	?
		⋮		
1	Honda	⋮	9,000$	?

- Allows for correlated random variables.
- One may use graphical models of any flavour (Bayesian networks, Markov networks, chain graphs).
- Query evaluation requires computing *marginal distributions* of random variables associated with result tuples.

Statistical Relational Learning and Lifted Inference

- Even though, they allow for factored tuple/attribute specific distributions, such representation power is almost never used/required.

AdID	Make	Color	Price
1	Honda	?	9,000$
2	?	?	6,000$
3	?	Beige	8,000$
⋮	⋮	⋮	⋮

Color	f_{color}
Black	0.75
Beige	0.25

Make	f_{make}
Honda	0.55
Toyota	0.45

- Approaches from statistical relational learning (PRMs, MLNs etc.) are applicable
- Since query evaluation requires inference, lifted inference is needed.

Semantics of Probabilistic Databases

S	A	B	
s_1	a_1	1	0.8
s_2	a_2	1	0.8
s_3	a_3	1	0.6

T	B	C	
t_1	1	c	0.5

$$\text{dom}(s_1) = \{\text{t}, \text{f}\}$$
$$\text{dom}(s_2) = \{\text{t}, \text{f}\}$$
$$\text{dom}(s_3) = \{\text{t}, \text{f}\}$$
$$\text{dom}(t_1) = \{\text{t}, \text{f}\}$$

possible world	probability
$\{s_1, s_2, s_3, t_1\}$	0.192
$\{s_1, s_2, s_3\}$	0.192
$\{s_1, s_2, t_1\}$	0.128
$\{s_1, s_2\}$	0.128
$\{s_1, s_3, t_1\}$	0.048
$\{s_1, s_3\}$	0.048
$\{s_1, t_1\}$	0.032
$\{s_1\}$	0.032
$\{s_2, s_3, t_1\}$	0.048
$\{s_2, s_3\}$	0.048
$\{s_2, t_1\}$	0.032
$\{s_2\}$	0.032
$\{s_3, t_1\}$	0.012
$\{s_3\}$	0.012
$\{t_1\}$	0.008
$\emptyset$	0.008

Query Evaluation using Inference

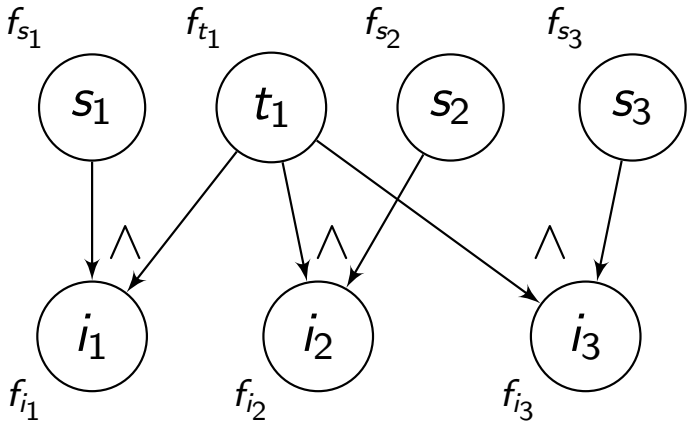
S

	A	B	
s_1	a_1	1	0.8
s_2	a_2	1	0.8
s_3	a_3	1	0.6

T

	B	C	
t_1	1	c	0.5

$S \bowtie_{\mathbf{B}} T$



e.g.,

s_1	f_{s_1}
0	0.2
1	0.8

i_1	s_1	t_1	f_{i_1}
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

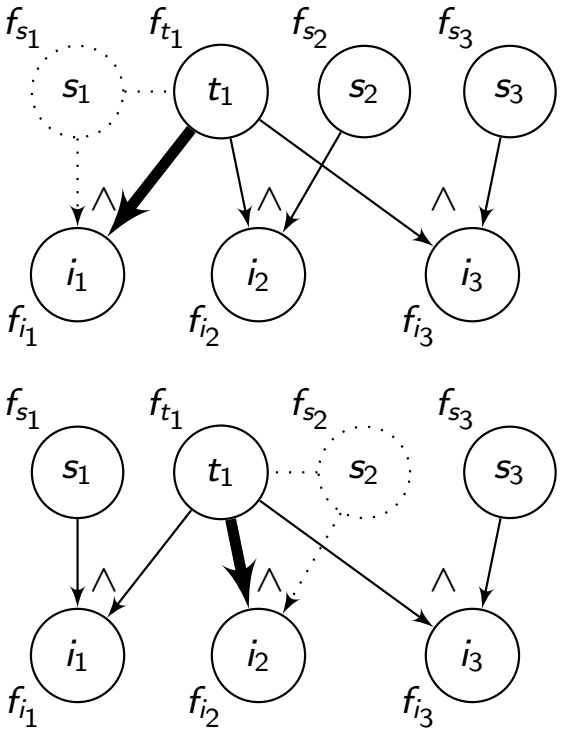
- Inference required:

$$\mu(i_j) = \sum_{s_j, t_1} f_{s_j}(s_j) f_{t_1}(t_1) f_{i_j}(i_j | s_j, t_1), \ \forall j = 1, 2, 3$$

Symmetry in Inference

$$\mu(i_1) = \sum_{t_1} f_{t_1}(t_1) \underbrace{\sum_{s_1} f_{s_1}(s_1) f_{i_1}(i_1|s_1, t_1)}_{m_{s_1}(i_1, t_1)}$$

$$\mu(i_2) = \sum_{t_1} f_{t_1}(t_1) \underbrace{\sum_{s_2} f_{s_2}(s_2) f_{i_2}(i_2|s_2, t_1)}_{m_{s_2}(i_2, t_1)}$$

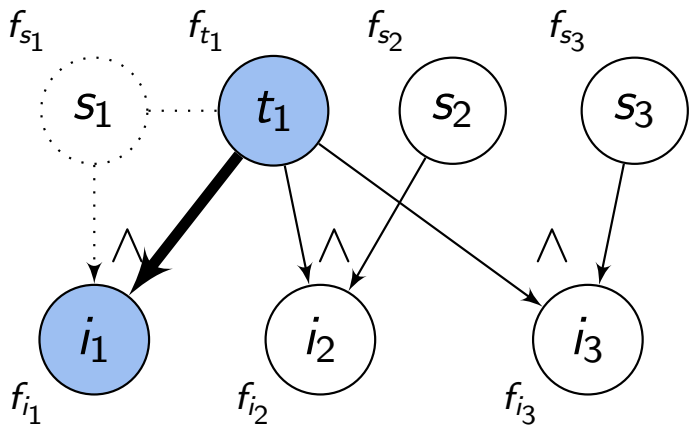


- Two factors f_1 and f_2 are *shared* (or $f_1 \cong f_2$) if they consist of the same input-output mappings.

0	0	1
0	1	0.2
1	0	0
1	1	0.8

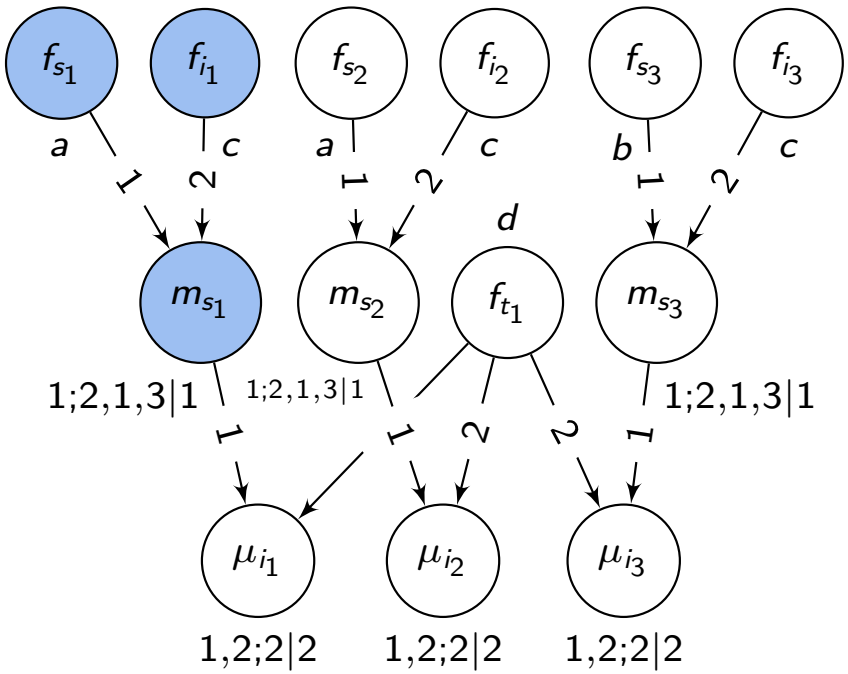
The Random Variable Elimination Graph

Graphical Model



$$\mu(i_1) = \sum_{t_1} f_{t_1}(t_1) \underbrace{\sum_{s_1} f_{s_1}(s_1) f_{i_1}(i_1|s_1, t_1)}_{m_{s_1}(i_1, t_1)}$$

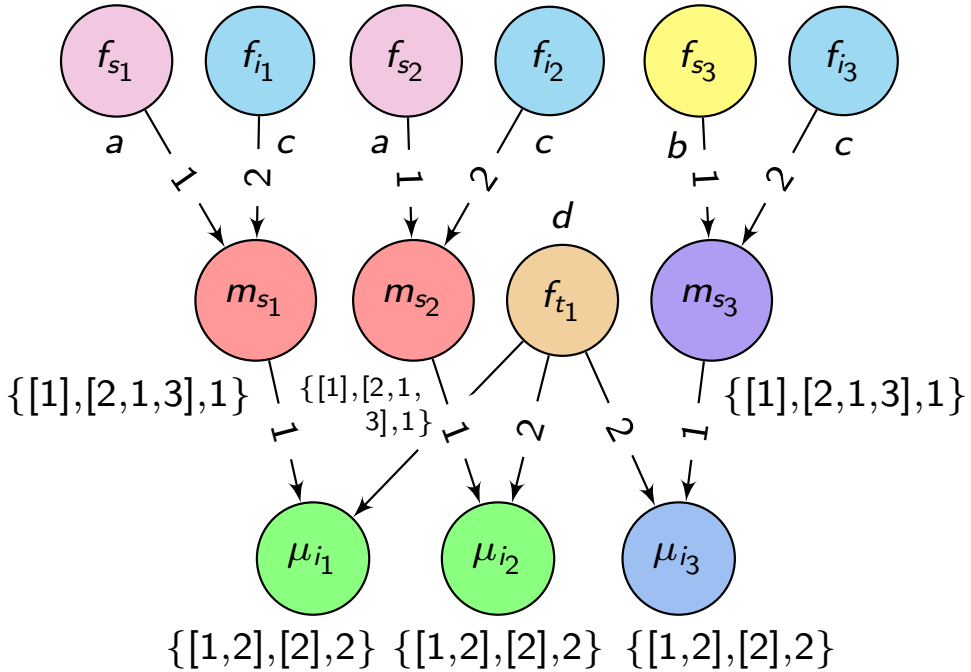
RV-Elim Graph

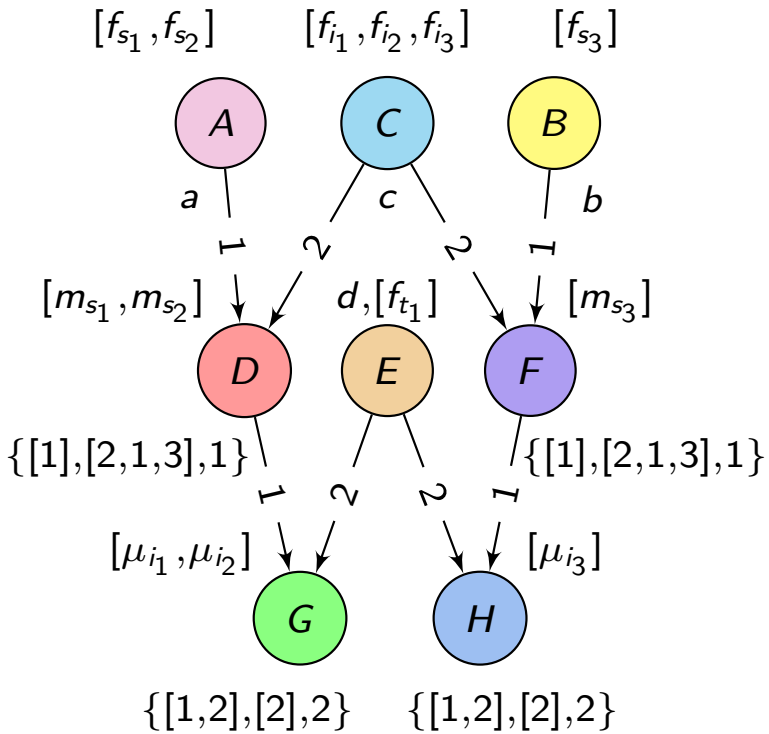


Exploiting Symmetry in RV-Elim Graphs

Two vertices v_1 and v_2 represent shared factors, i.e., $f_{v_1} \cong f_{v_2}$, iff:

- $\forall u_1 \xrightarrow{i} v_1, \exists u_2 \xrightarrow{i} v_2$ s.t. $f_{u_1} \cong f_{u_2}$ and vice versa.
- label(v_1) = label(v_2)

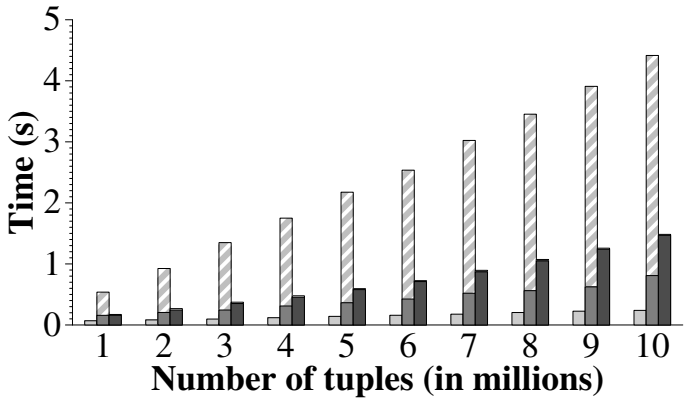


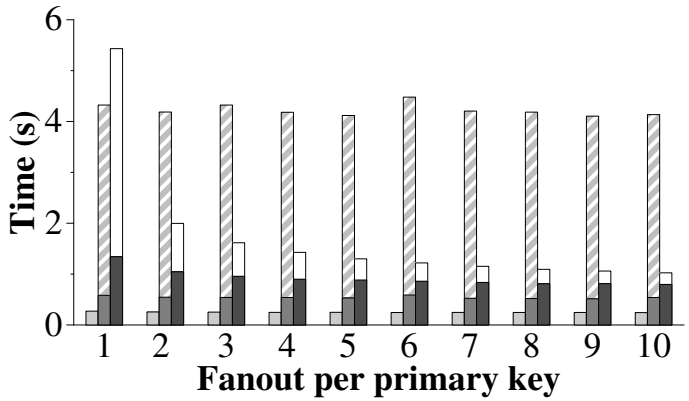


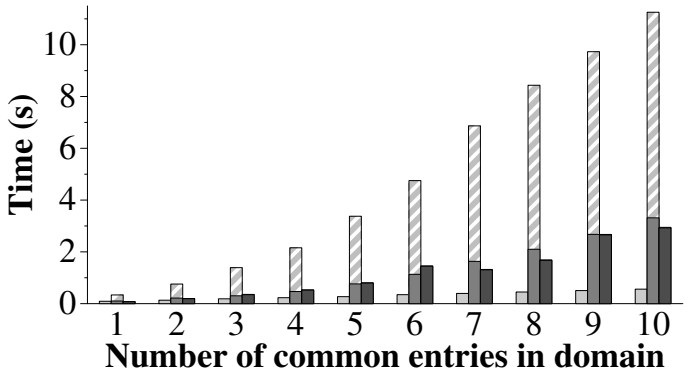
Details

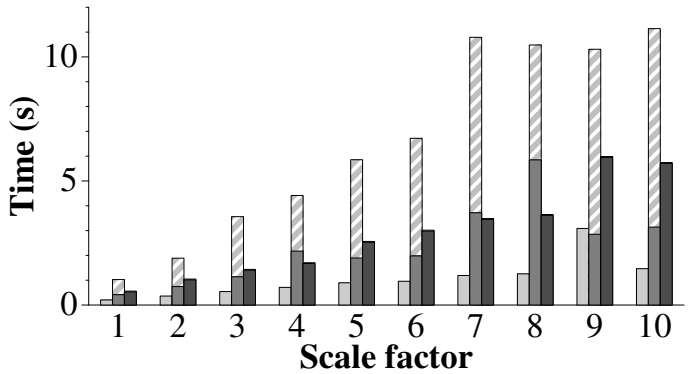
- Factor multiplication is *commutative*: Edge labels and internal vertex labels are constructed on the fly.
- Final inference algorithm is a three-stage approach:
  1. Detect shared factors in the rv-elim graph.
  2. Run inference on the compressed rv-elim graph.
  3. Retrieve relevant marginals.
- Computing "$\cong$" is closely related to *bisimulation* [KS83].
- RV-Elim graphs are DAGs.
- Fast bisimulation algorithms available for DAGs [DPP01].
- Our algorithm runs in $O(|E| \log D + |V|)$ time.
- Choice of elimination order affects symmetry in the rv-elim graph.
- Elimination order heuristic is a combination of:
  - Bisimulation on the probabilistic graphical model.
  - *Min-size heuristic*.

Experiments on Synthetic Data

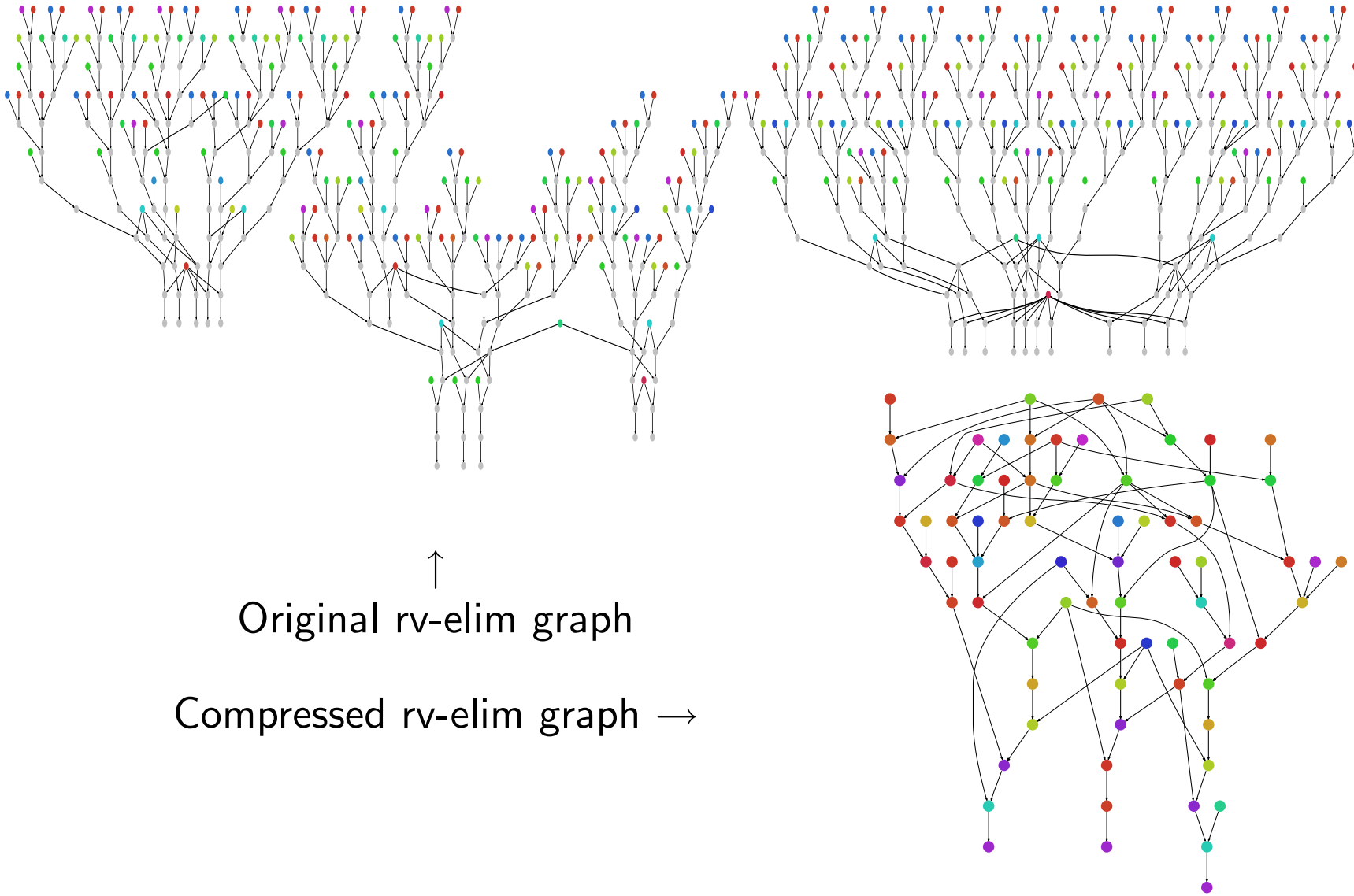








Sample RV-Elim graphs

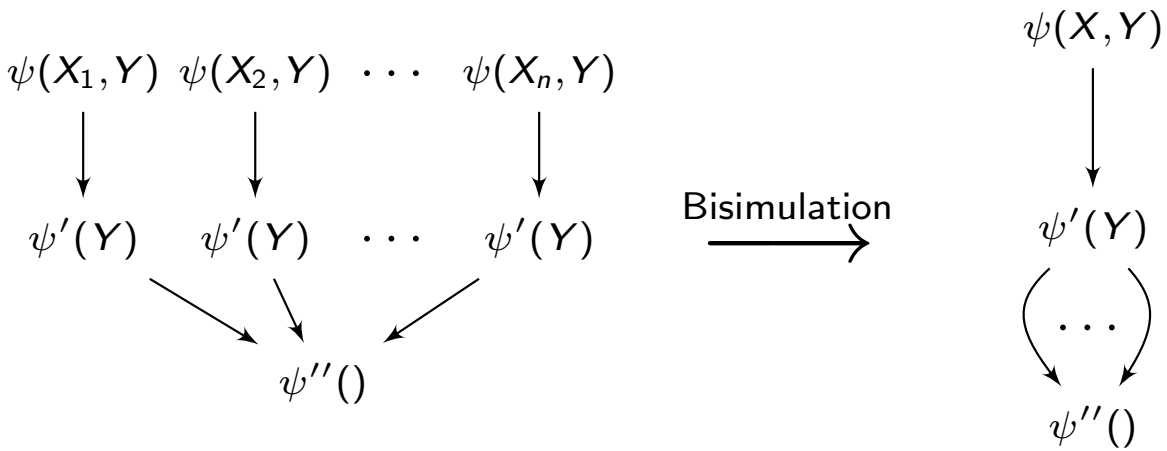


↑

Original rv-elim graph

Compressed rv-elim graph →

Conclusion



- Our lifted inference subsumes inversion elimination.

$$\sum_{X_i} \prod_{i=1}^{n} \psi(X_i, Y) = \prod_{i=1}^{n} \sum_{X_i} \psi(X_i, Y)$$

- Inference $\equiv$ Finite state automaton $\Rightarrow$ minimized FSA $\equiv$ Lifted Inference.
- Real-world data poses significant challenges.
- For approximate and treewidth-reducing inference techniques see next session.

References

[BDHW06] Omar Benjelloun, Anish Das Sarma, Alon Halevy and Jennifer Widom.
ULDBs: Databases with Uncertainty and Lineage.
In *VLDB*, 2006.

[DSS10] Nilesh N. Dalvi, Karl Schnaitter and Dan Suciu
Computing query probability with incidence algebras.
In *PODS*, 2010.

[DS04] Nilesh Dalvi and Dan Suciu.
Efficient Query Evaluation on Probabilistic Databases.
In *VLDB*, 2004.

[DGS08] Amol Deshpande, Lise Getoor and Prithviraj Sen.
Graphical Models for Uncertain Data.
In *Managing and Mining Uncertain Data*, Charu Aggarwal (ed.), Springer, 2008.

[DPP01] Agostino Dovier, Carla Piazza and Alberto Policriti.
A Fast Bisimulation Algorithm.
In *International Conference on Computer Aided Verification*, 2001.

References

[FR97] Norbert Fuhr and Thomas Rolleke.

A probabilistic relational algebra for the integration of information retrieval and database systems.

In *Transactions on Information Systems*, 1997.

[GFKT02] L. Getoor, N. Friedman, D. Koller and B. Taskar.

Learning probabilistic models with link uncertainty.

In *JMLR*, 2002.

[GMR06] M. Golumbic, A. Mintz and U. Rotics.

Factoring and Recognition of Read-Once Functions Using Cographs and Normality and the Readability of Functions Associated with Partial k-Trees.

In *Discrete Applied Mathematics*, 2006.

[JXWPJH08] Ravi Jampani, Fei Xu, Mingxi Wu, Luis Leopoldo Perez, Christopher M. Jermaine and Peter J. Haas.

MCDB: A monte carlo approach to managing uncertain data.

In *SIGMOD*, 2008.

[KS83] Paris Kanellakis and Scott Smolka.

CCS expressions, finite state processes, and three problems of equivalence.

In *PODC*, 1983.

References

[KO08] Christoph Koch and Dan Olteanu.
Conditioning probabilistic databases.
In *VLDB*, 2008.

[OH09] Dan Olteanu and Jiewen Huang.
Secondary-storage confidence computation for conjunctive queries with inequalities.
In *SIGMOD*, 2009.

[OH08] Dan Olteanu and Jiewen Huang.
Using OBDDs for Efficient Query Evaluation on Probabilistic Databases.
In *SUM*, 2008.

[Poole03] David Poole.
First-order probabilistic inference.
In *IJCAI*, 2003.

[RDS07] Christopher Re, Nilesh Dalvi and Dan Suciu.
Efficient Top-k Query Evaluation on Probabilistic Data.
In *ICDE*, 2007.

[RD06] Matthew Richardson and Pedro Domingos.
Markov Logic Networks.
In *Machine Learning*, 2006.

References

[SD07] Prithviraj Sen and Amol Deshpande.
Representing and Querying Correlated Tuples in Probabilistic Databases.
In *ICDE*, 2007.

[SDG08] Prithviraj Sen, Amol Deshpande and Lise Getoor.
Exploiting Shared Correlations in Probabilistic Databases.
In *VLDB*, 2008.

[SDG09] Prithviraj Sen, Amol Deshpande and Lise Getoor.
Bisimulation-based Approximate Lifted Inference.
In *UAI*, 2009

[SDG09] Prithviraj Sen, Amol Deshpande and Lise Getoor.
PrDB: Managing and Exploiting Rich Correlations in Probabilistic Databases.
In VLDB Journal, 2009.

[SDG10] Prithviraj Sen, Amol Deshpande and Lise Getoor.
Read-Once Functions and Query Evaluation in Probabilistic Databases.
In *VLDB*, 2009.

[SDG07] Prithviraj Sen, Amol Deshpande and Lise Getoor.
Representing Tuple and Attribute Uncertainty in Probabilistic Databases.
In *DUNE (ICDM)*, 2007.

Lifted Inference for Relational Continuous Models

Eyal Amir*
University of Illinois at Urbana-Champaign

* speaker
(Joint work with Jaesik Choi, David J. Hill, and Abner Guzman-Rivera)

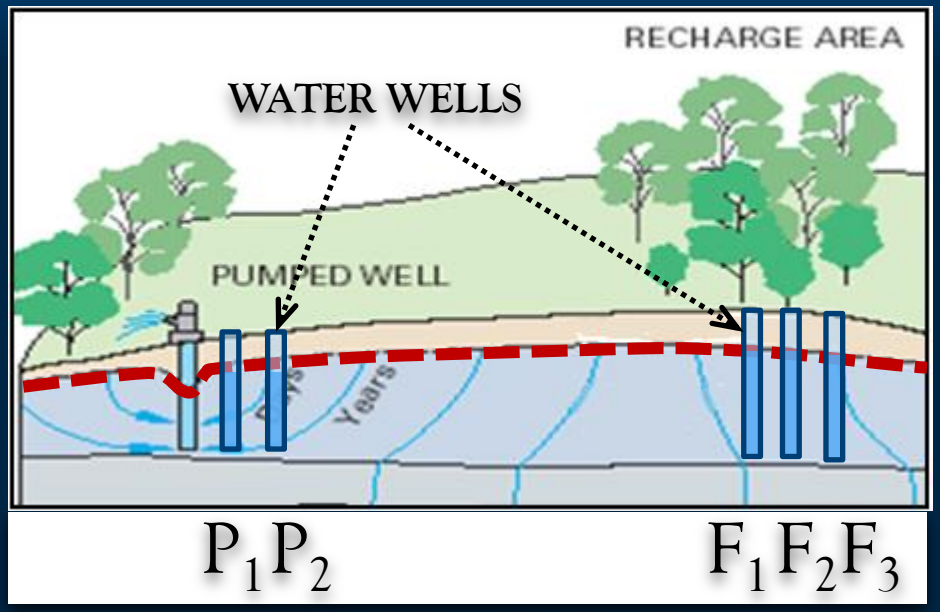
The presentation is based on

[J. Choi, D. J. Hill, and E. Amir, Lifted Inference for Relational Continuous Models, UAI-10]

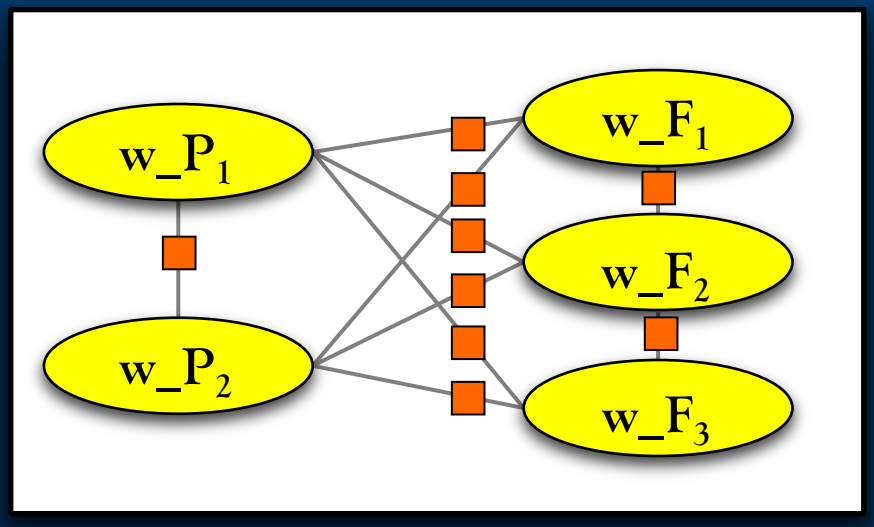
[J. Choi, A. Guzman-Rivera, and E. Amir, Lifted Relational Kalman Filtering, IJCAI-2011]

7/2/2011

Relational Continuous Models: Example 1



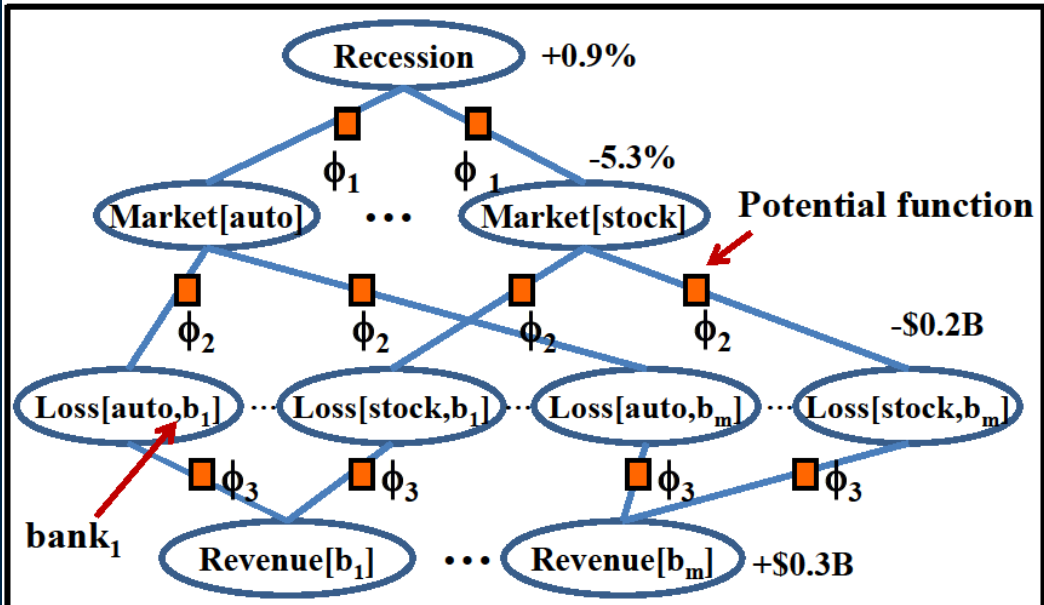
Source: United State Geology Survey (USGS)

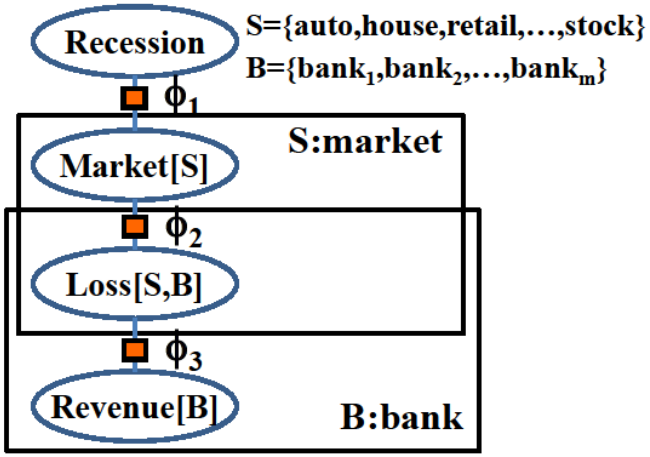


- **Goal**: Estimating levels of water wells (above sea level)
- P_1, P_2: wells near a Pump. F_1, F_2, F_3 well in Forest.
- w_x: water level at x (e.g. w_P_1=300ft)
- Inference problem (variable elimination of w_P_2):

$$P(w_{P1}, w_{F1}, w_{F2}, w_{F3}) = \int P(w_{P1}, w_{P2}, w_{F1}, w_{F2}, w_{F3}) dw_{P2}$$

Relational Continuous Models (RCMs) Example 2

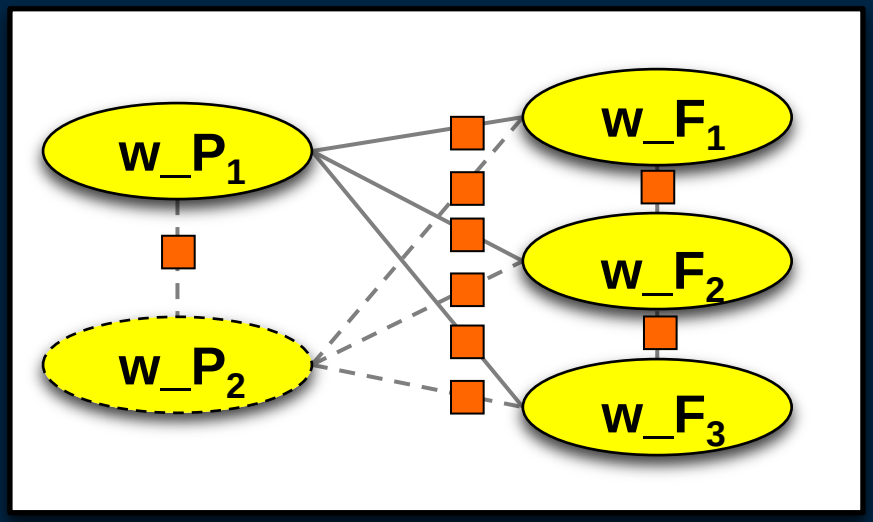


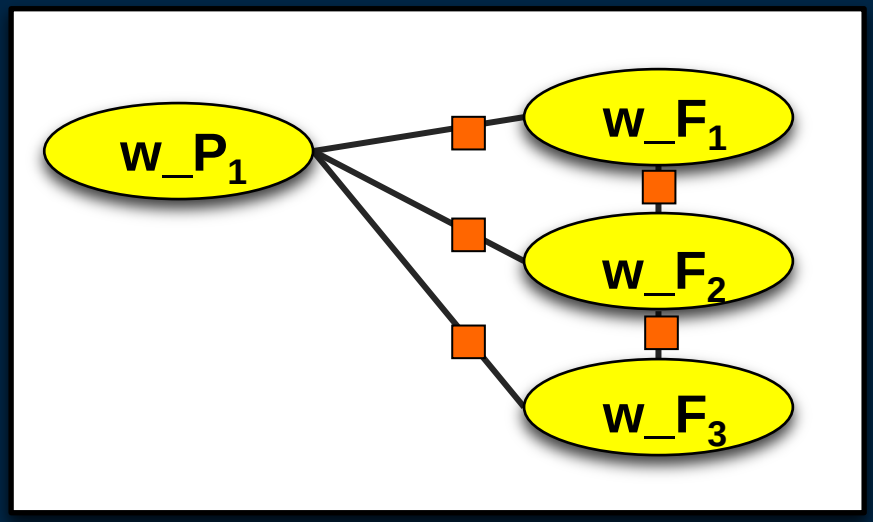


Recession: Recession of a country in [0,1] (i.e +0.9%)
Loss[S,B]: Loss of a bank(B) in a sector(S) (i.e -$0.2B)
Revenue[B]: Revenue of a bank(B) (i.e +$0.3B)
Market[S]: market index of a sector(S) (i.e -5.3%)

This Tutorial's Main Point

- Maintaining pair-wise form during marginalization





$$\int \phi_{PP}(w\_P_1, w\_P_2) \cdot \prod_{j,j'} \phi_{FF}(w\_F_j, w\_F_{j'}) \prod_{i,j} \phi_{PF}(w\_P_i, w\_F_j) \cdot dw\_p_2$$

$$= \prod_{j,j'} \phi'_{FF}(w\_F_j, w\_F_{j'}) \prod_{j} \phi'_{PF}(w\_P_1, w\_F_j)$$

This Tutorial's Main Point

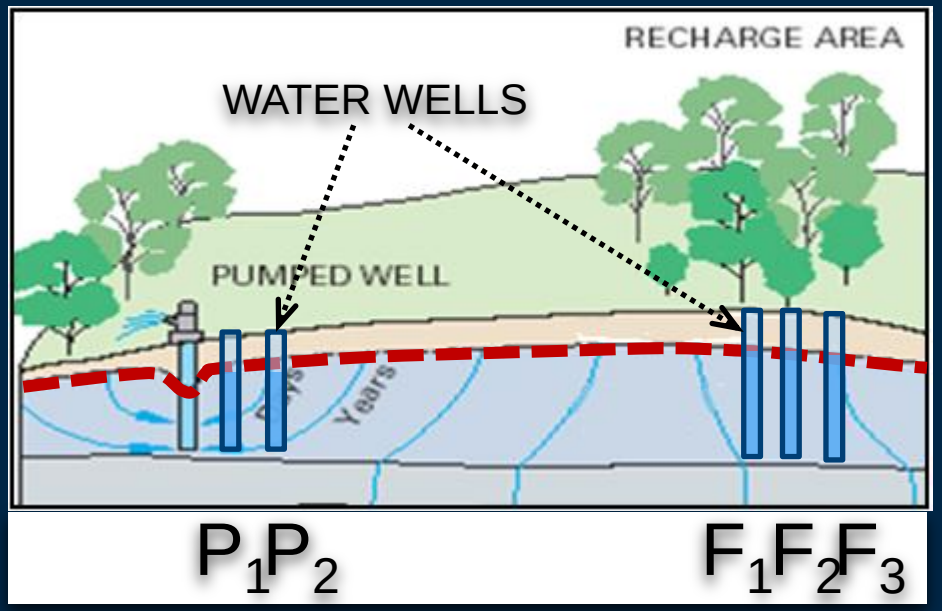
- Relational Continuous models
  - Inference time linear in the number of objects
- Key Insight:
  - Maintain representation with Variable Elimination: *product of pairwise potentials*
- Application: Lifted Kalman Filtering (IJCAI'11 Technical presentation)

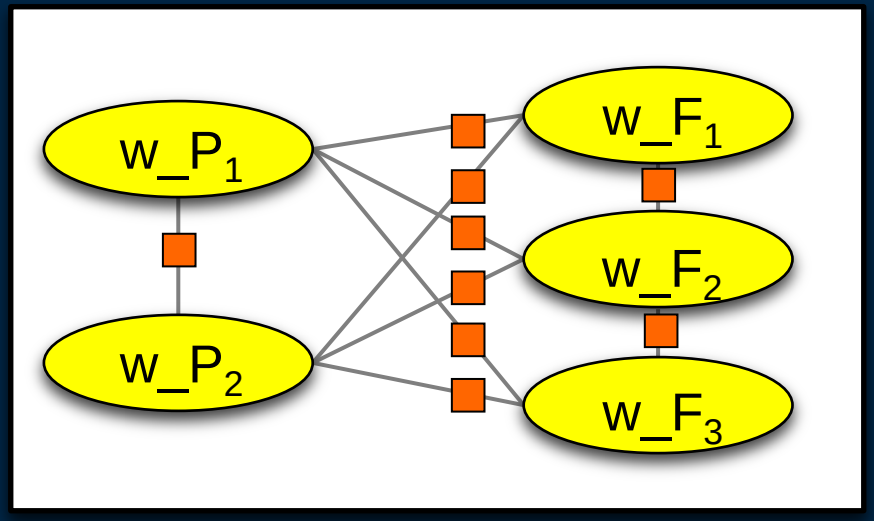
Tutorial Outline

1. Relational Continuous Models (RCMs)
2. Lifted Variable Elimination
3. Conditions for Lifted Inference
4. Special Cases: Relational Gaussian Models
5. Lifted Relational Kalman Filtering (LRKF)

Relational Continuous Models (RCMs)

- Represent problems with a relational graphical model

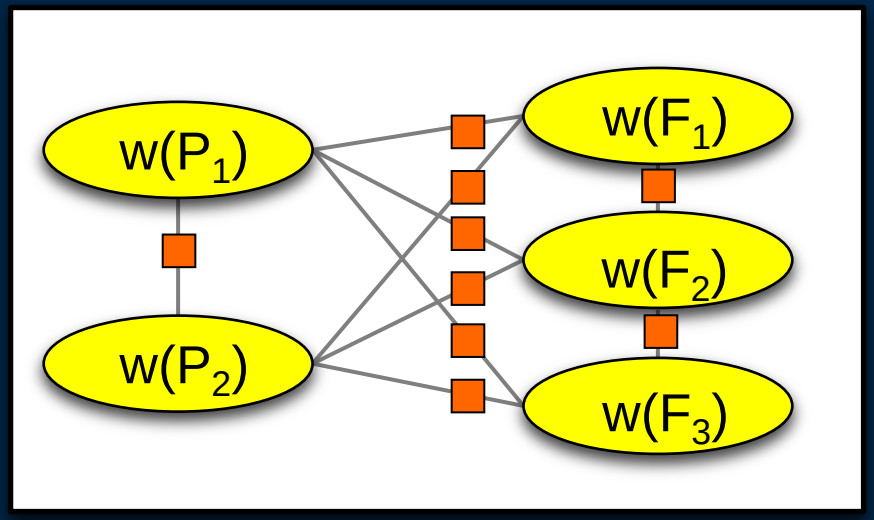


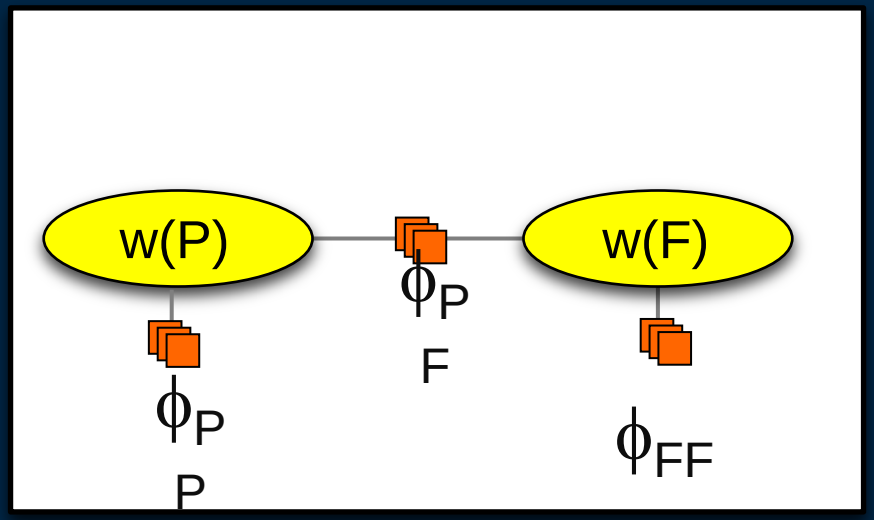


- Nodes: random variables (e.g. w_P_1: the water level at P_1)
- Edges: potential functions represent joint density of two random variables (e.g. $\phi_{PF}(w\_P_1, w\_F_2)$).

Relational Continuous Models (RCMs)

- Observation: some potential functions (edges) are almost identical.

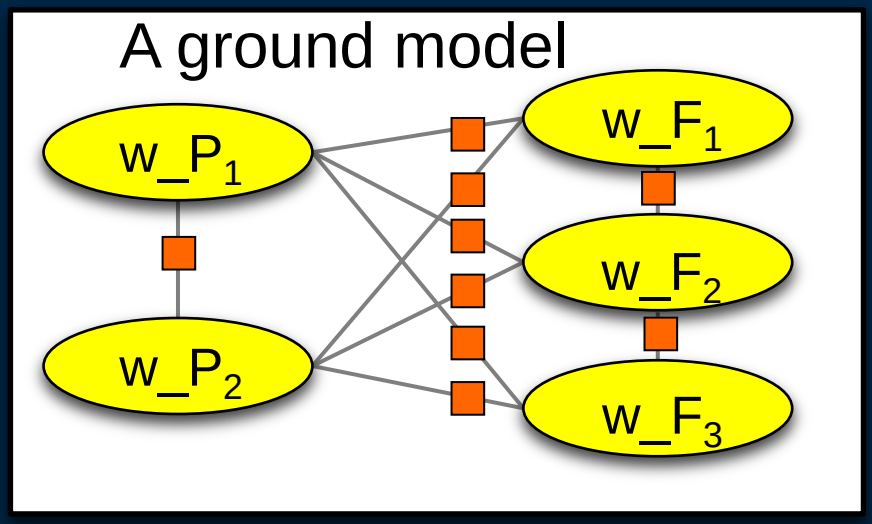


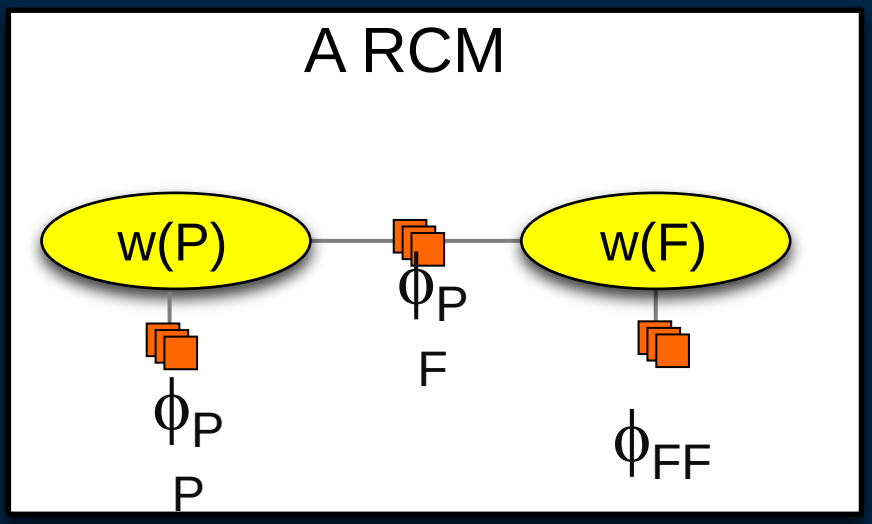


- Thus, we can represent the model compactly (e.g. with only 3 potentials)
- w(P) and w(F) are parameterized random variables.
- ϕ_{PP}, ϕ_{PF} & ϕ_{FF} are potentials between water levels. (e.g. difference of two water levels follows a distribution such as Gaussian).

Relational Continuous Models (RCMs)

- Observation: some potential functions are almost identical.





A ground models

$$\phi_{PP}(w(P_1), w(P_2)) \cdot \phi_{FF}(w(F_1), w(F_2)) \cdot \phi_{FF}(w(F_2), w(F_3)) \cdot \phi_{FF}(w(F_1), w(F_3))$$

$$\phi_{PF}(w(P_1), w(F_2)) \cdot \phi_{PF}(w(P_1), w(F_3)) \cdot \phi_{PF}(w(P_2), w(F_1)) \cdot \phi_{PF}(w(P_2), w(F_2)) \cdot \phi_{PF}(w(P_2), w(F_3))$$

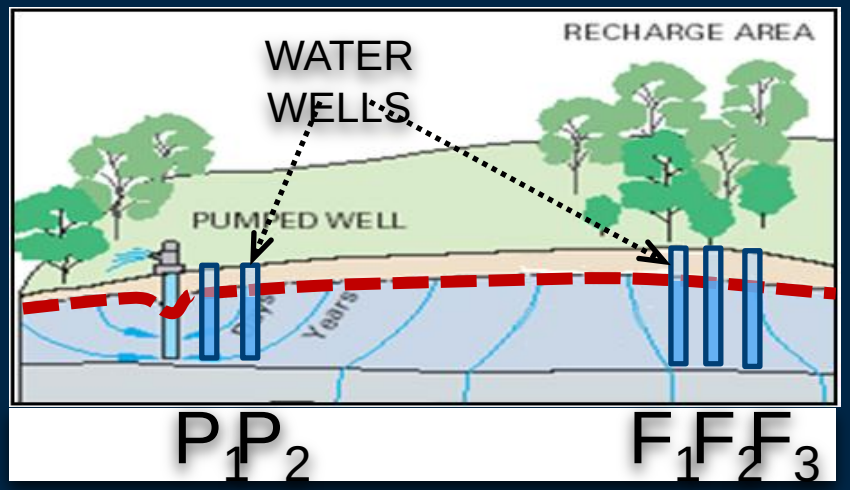
A Relational Continuous Model

$$= \phi_{PP}(w(P_1), w(P_2)) \cdot \prod_{j, j'(j \neq j')} \phi_{FF}(w(F_j), w(F_{j'})) \cdot \prod_{i,j} \phi_{PF}(w(P_i), w(F_j))$$

A RCM with Gaussian Potentials

- Two parameters, (μ, σ^2), are selected for each RN.

$$\phi_{RN}(X(\ldots),Y(\ldots)) = c \cdot \exp(-(X(\ldots)-Y(\ldots)-\mu)^2 / \sigma^2)$$



μ: the difference of water levels

σ^2 : the parameter proportional to the covariance.

$$\phi_{PP}(w(P_1),w(P_2)) = \exp\left(-(w(P_1)-w(P_2))^2 / 0.8\right)$$

$$\phi_{FF}(w(F_j),w(F_{j'})) = \exp\left(-(w(F_j)-w(F_{j'})^2 / 1.1\right)$$

Small covariances in a relational atom

$$\phi_{PF}(w(P_i),w(F_j)) = \exp\left(-(w(P_i)-w(F_j)-12)^2 / 5.3\right)$$

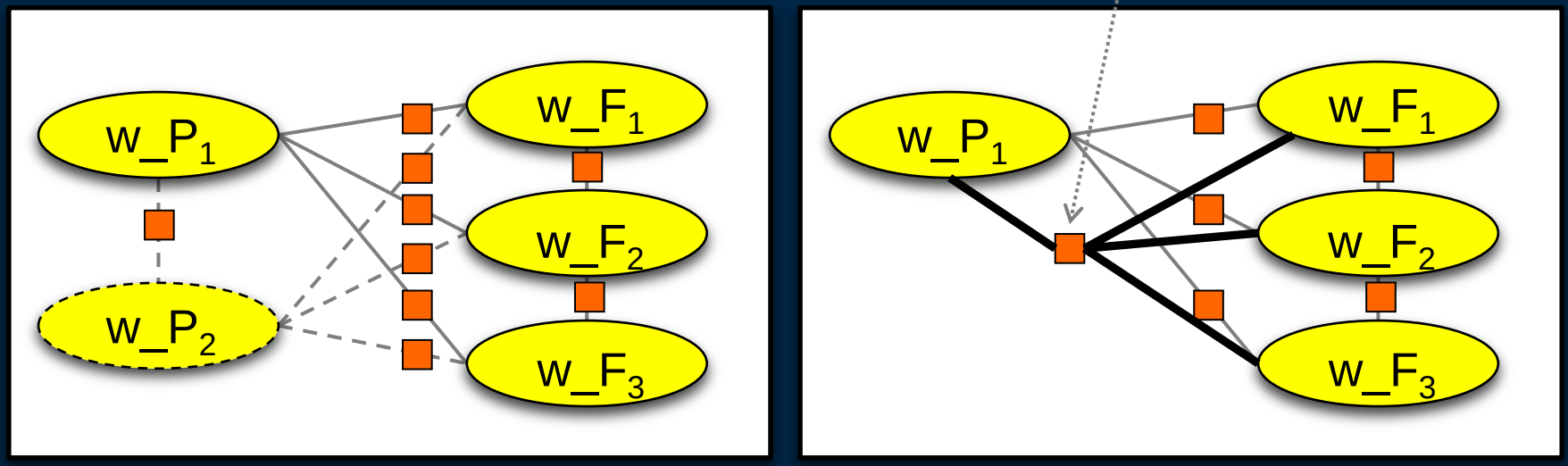
Large covariances between two relational atoms

Tutorial Progress

1. Relational Continuous Models (RCMs)
2. Lifted Variable Elimination
3. Conditions for Lifted Inference
4. Special Cases: Relational Gaussian Models
5. Lifted Relational Kalman Filtering (LRKF)

Inference with Ground RVs

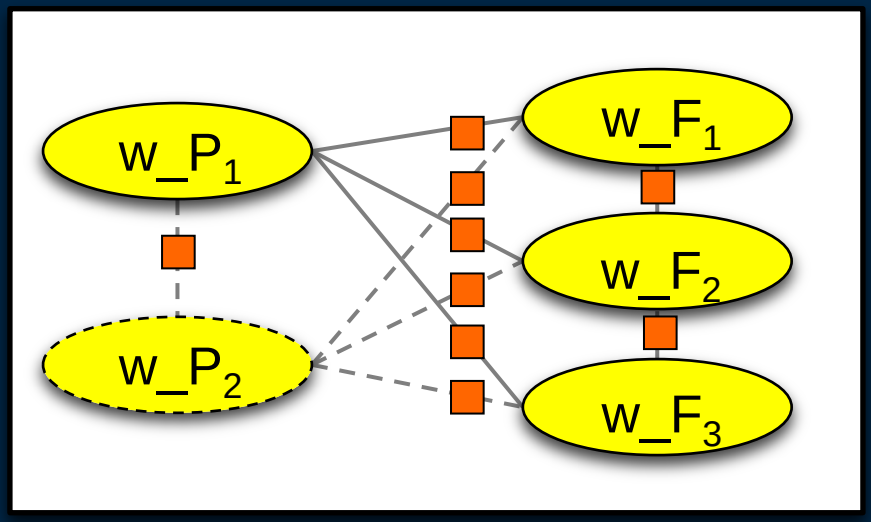
- Eliminating w_P_2 results in $\phi(w\_P_1, w\_F_1, w\_F_2, w\_F_3)$

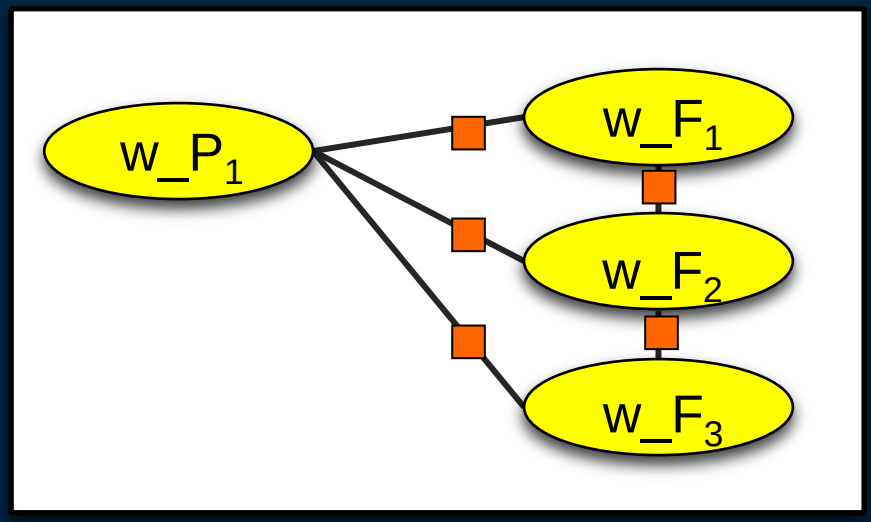


- The new potential $\phi(w\_P_1, w\_F_1, w\_F_2, w\_F_3)$ hard to use and represented in a lifted way because it includes all objects & RVs

Some RCMs can Maintain Pair-wise Product Representation

- Maintaining pair-wise form during marginalizations





$$\int \phi_{PP}(w\_P_1, w\_P_2) \cdot \prod_{j,j'} \phi_{FF}(w\_F_j, w\_F_{j'}) \prod_{i,j} \phi_{PF}(w\_P_i, w\_F_j) \cdot dw\_p_2$$

$$= \prod_{j,j'} \phi'_{FF}(w\_F_j, w\_F_{j'}) \prod_{j} \phi'_{PF}(w\_P_1, w\_F_j)$$

Example: Relational Normal

- Pair-wise Relational Normal (RNs) potentials
  - A family of Gaussian potentials in the following form
  - Potential defined over two relational atoms (X(…), Y(…))

$$\phi_{RN}(X(\ldots),Y(\ldots)) = c \cdot \exp(-(X(\ldots) - Y(\ldots) - \mu)^2 / \sigma^2)$$

- An example

$$e.g. \phi_{PF}(w(P),w(F)) = c \cdot \exp(-(w(P) - w(F) - \mu_{PF})^2 / \sigma_{PF}^2)$$

Eliminating random variables in RNs

- Reminder: Gaussian Potentials are exponential quadratic
  - All RVs (e.g. $w(P_i)$) are in quadratic form

$$\exp(-a w(P_i)^2 + b w(P_i) + c)$$

  - a is a positive constant, and b,c are terms without $w(P_i)$.
- Reminder: Marginalizing out RV $w(P_i)$

$$\int \exp(-a w(P_i)^2 + b w(P_i) + c)\, dw(P_i) = \sqrt{\frac{\pi}{a}} \exp\left(\frac{b^2}{4a} + c\right)$$

Lifted Elimination of RVs

$$\int \exp(-a w(P_i)^2 + b w(P_i) + c)\, dw(P_i) = \sqrt{\frac{\pi}{a}} \exp\left(\frac{b^2}{4a} + c\right)$$

- Detail of marginalization

$$\int \phi_{PP}(w\_P_1, w\_P_2) \cdot \prod_{i,j} \phi_{PF}(w\_P_i, w\_F_j) \cdot dw\_p_2$$

$$= \int \exp\left(-\left(\frac{1}{\sigma_{PP}^2} + \frac{|F|}{\sigma_{PF}^2} \right) w(P_2)^2 + 2\left(\frac{w(P_1)}{\sigma_{PP}^2} + \frac{\sum_j w(F_j)}{\sigma_{PF}^2} \right) w(P_2) - \left(\frac{w(P_1)^2}{\sigma_{PP}^2} + \frac{\sum_j w(F_j)^2}{\sigma^2} \right) \right) dw(P_2)$$

a constant

sum of variables

sum of squares of variables

$$= \exp\left(\frac{w(P_1)^2}{c_{PP}^2} - \frac{\sum_j w(F_j)^2}{c_F^2} - \frac{\sum_j w(P_1) w(F_j)}{c_{PF}^2} - \frac{\sum_{j,j'(j \neq j')} w(F_j) w(F_{j'})}{c_{FF}^2} \right)$$

$$= \prod_{j,j'} \phi'_{FF}(w(F_j), w(F_{j'})) \prod_j \phi'_{PF}(w(P_1), w(F_j))$$

Conditions for Lifted Inference

- **Lifted inference in pair-wise form** applies if potentials are
  - Analytically integrable

$$\int \phi_{PP}(w(P_2), w(P_2)) \prod_j \phi_{PF}(w(P_2), w(F_j)) dw(P_2) = \phi_{new}(w(P_1), w(F_1), w(F_2), w(F_3))$$

  - Closed under marginalization*

$$*\phi \in \Theta, \int \phi = \phi' \rightarrow \phi' \in \Theta$$

$$\phi_{new}(w(P_1), w(F_1), w(F_2), w(F_3)) = \prod_j \phi'_{PF}(w(P_1), w(F_j)) \cdot \prod_{j, j'(j \neq j')} \phi'_{FF}(w(F_j), w(F_{j'}))$$

  - Closed under the product operation**

$$**\phi \in \Theta, \phi' \in \Theta \rightarrow \phi \cdot \phi' \in \Theta$$

$$\phi_{PF}(w(P_1), w(F)) \phi'_{PF}(w(P_1), w(F)) = \phi''_{PF}(w(P_1), w(F))$$

Computational Complexity

- Given
  - |U| is the number of ground random variables
  - |R| is the number of relational atoms
  - Normally, |U|>>|R|
- Marginalization of n random variables in a ground form

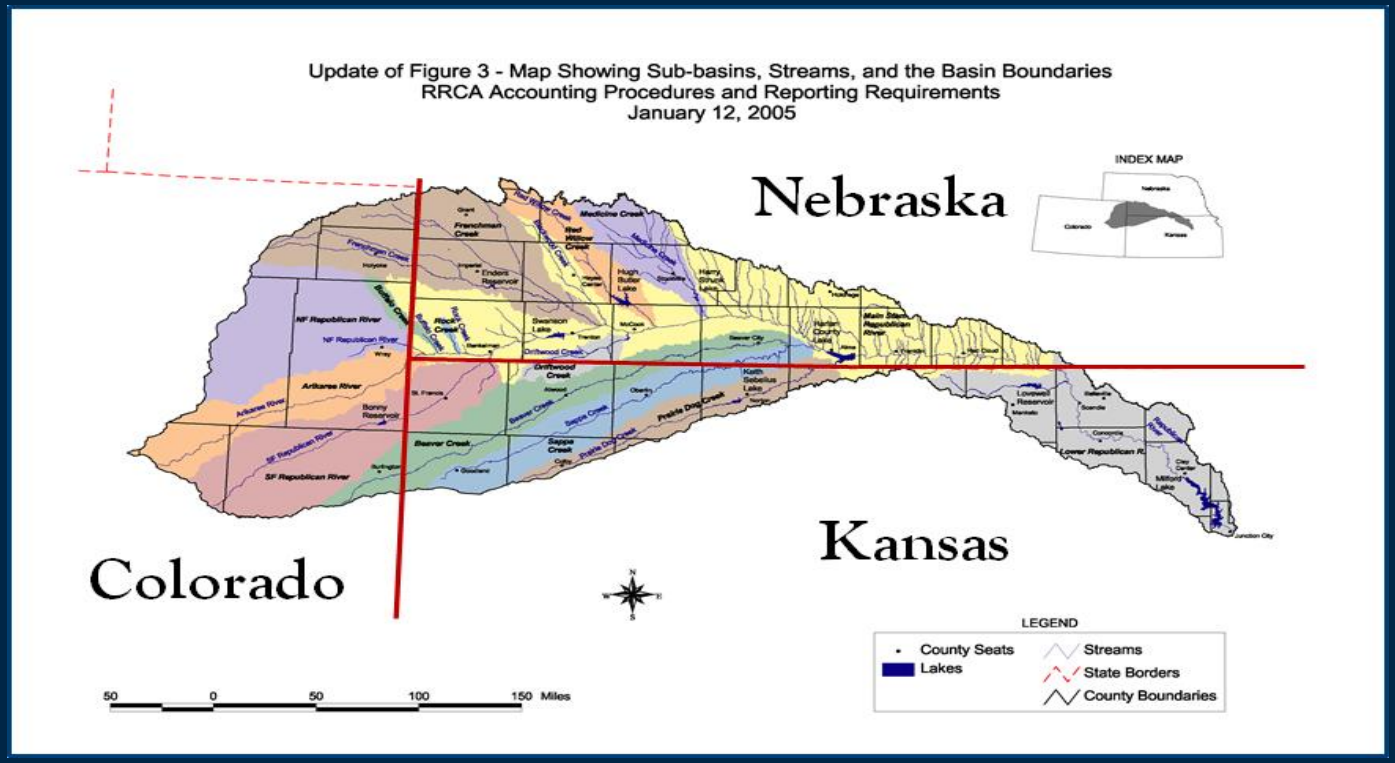
$$O(n \cdot |U|^2)$$

- Marginalization of n random variables in pair-wise form

$$O(n \cdot |R|^2)$$

Experiment 1: Estimating Water Levels of Wells

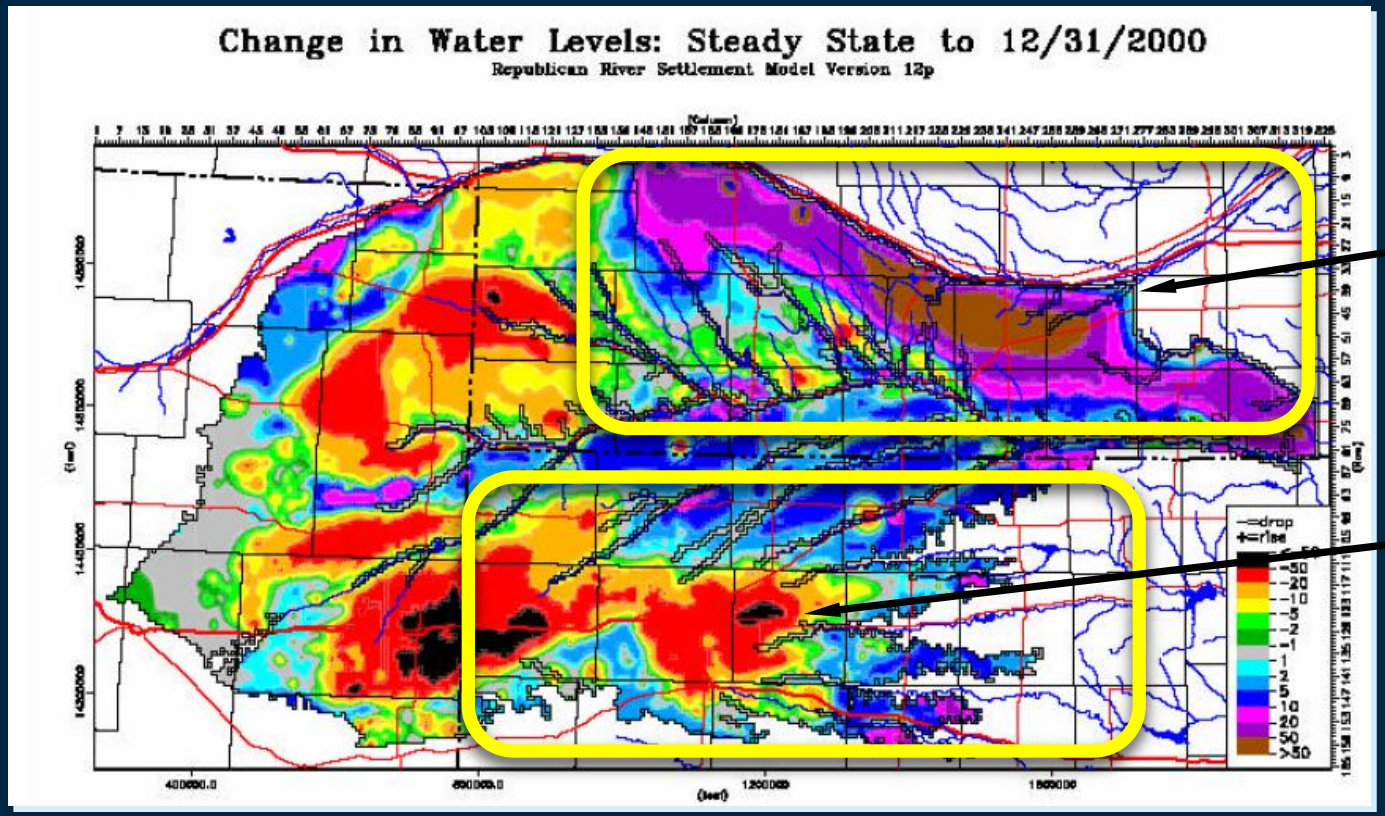
- Water wells near the Republican River (spans Colorado, Kansa, and Nebraska in the USA)



- Dataset: a total of 350,233 water level records, 10,835 different water wells or sites.

Experiment 1: Estimating Water Levels of Wells

- Why is this important?
  - Regions suffer lack of water due to excessive water pumping in other regions. (In fact, state of Kansas filed a lawsuit against state of Nebraska)

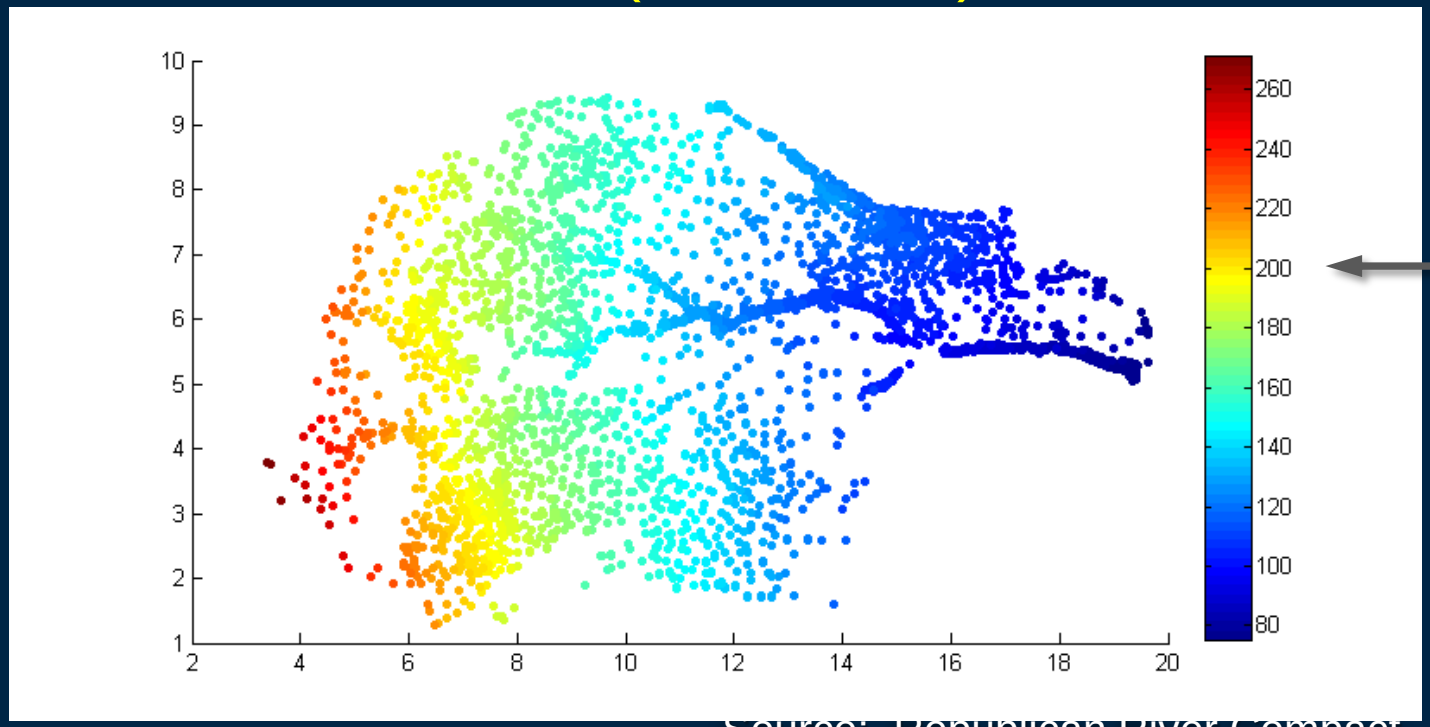


Nebraska
(water surplus)

Kansas
(water depletion

Distribution of water wells

- 10,835 different sites (water wells)



Source: Republican River Compact Agreement

- Estimation is challenging because
  - many wells to estimate, but only some measurements
  - many factors (precipitation, pumping, stream, …) to consider

Experiment 1: Estimating Water Levels of Wells

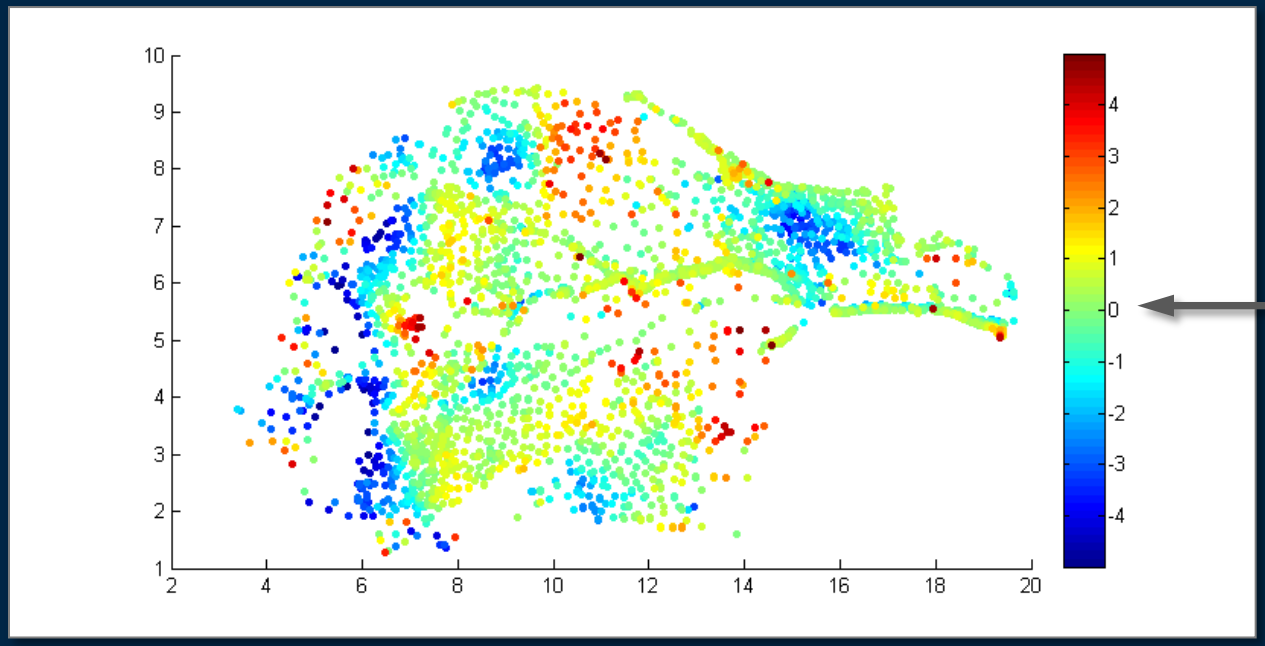
Limitations of existing models for RRCA

- ***[**Joint work with Albert Valocchi, Environmental Eng, University of Illinois, Urbana-Champaign]***

- Experts in environmental science builds an elaborate estimating model, **Groundwater Model**, which consider various factor such as streams, evapotranspiration, well pumping, and recharge.
- However, for some reasons, the estimations still have errors (or water residuals) from measurements.
- Here, we focus on **efficient (lifted) estimations of water residuals with a RCM of RNs**.

Building a Relational Continuous Model (RCM) for RRCA

- We first grouped the set of water wells which have the similar statistics (e.g. means, variances)

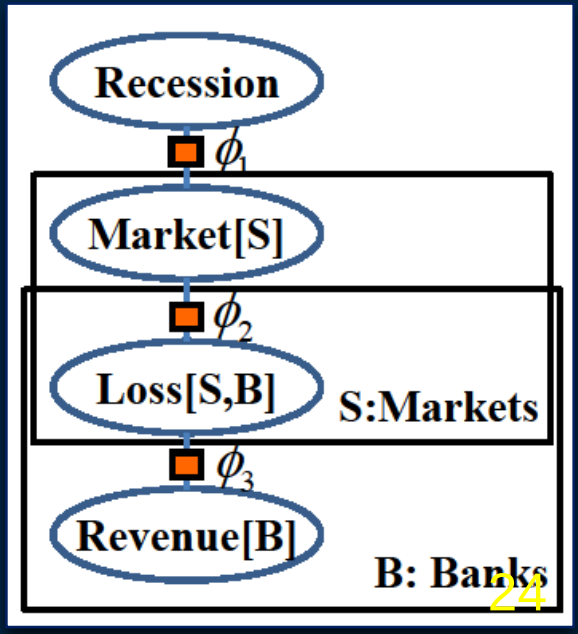


Average residuals

- Then, we learned the parameters of RN potentials.
- Experiment with different models using lifted inference

Experiment 2: Economics Simulations

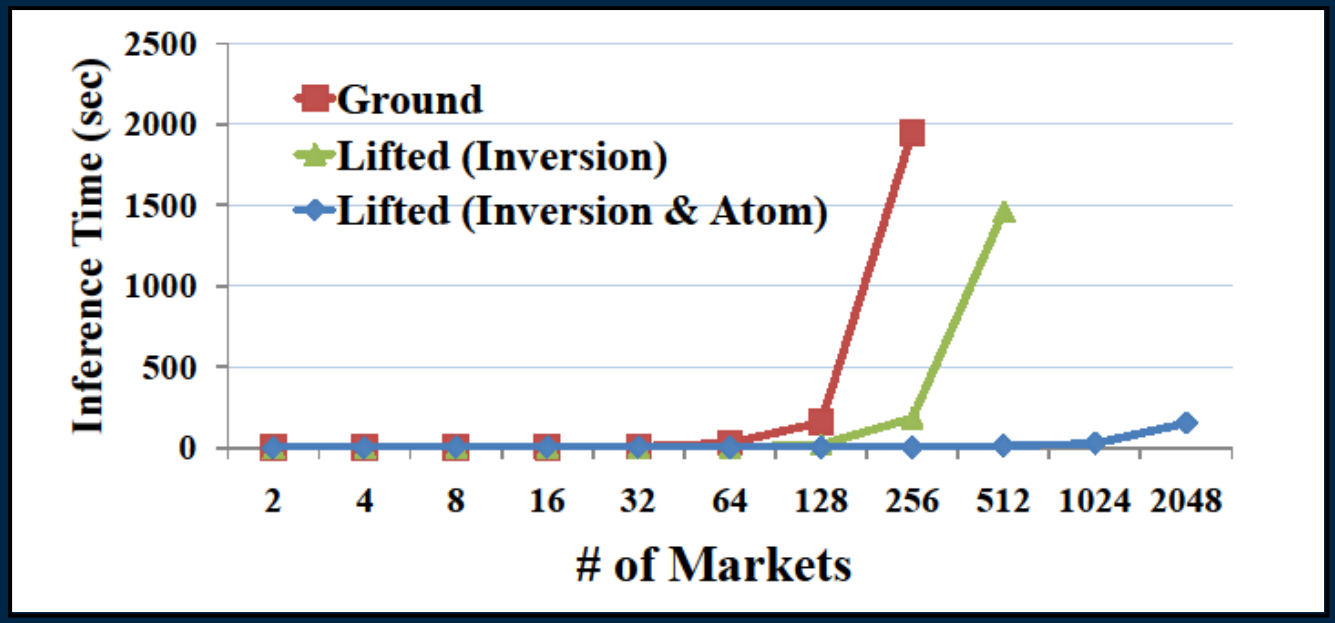
- Experimental settings
  - 2 logical variables: S, a set of 'market sectors' and B, a set of 'banks'
  - 4 relational atoms: Recession, Market(S), Loss(S,B), Revenue(B)
  - 3 parfactors in RNs: ϕ_1(Recession, Market(S)), ϕ_2(Market(S),Loss(S,B)), and ϕ_3(Loss(S,B), Revenue(B))
- An inference problem
  - Observations: Market(s_1) and Revenue(b_1)
  - Output: Recession (eliminating all others)
- Compare three inference algorithms
  - Grounding and Variable Elimination
  - Discrete-model lifted inference (Inversion)
  - Discrete-model lifted inference (Inversion, atom)

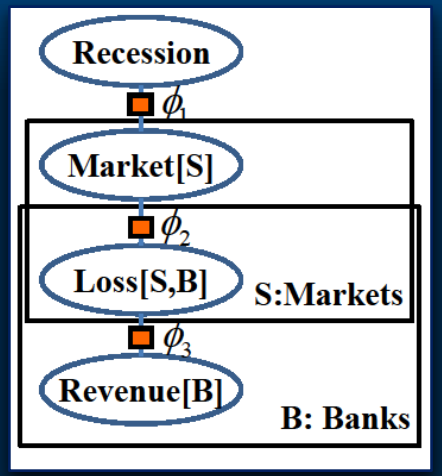


Experiment 2: Economics Simulations

Increase the number of market sectors, |S|

- Time for exact inference

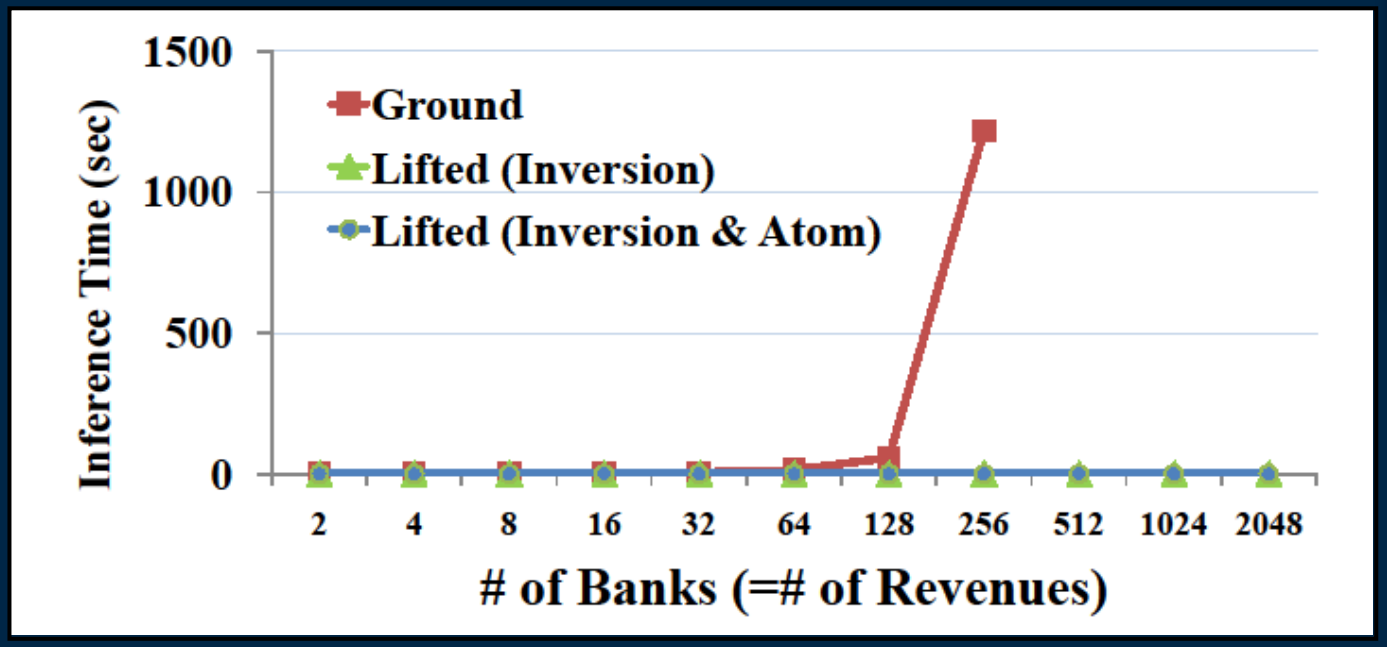


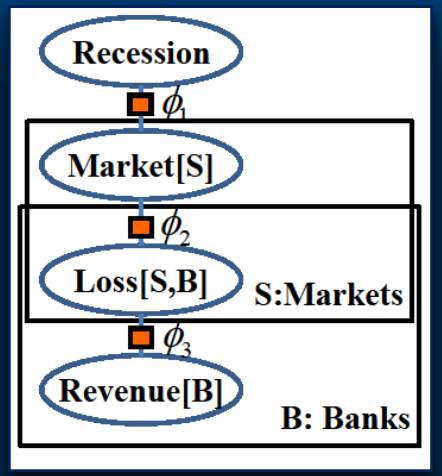


Experiment 2: Economic Simulations

Increase the number of banks, |B|

- Time for exact inference





- The computational complexity for the algorithm based on ground random variables is quadratic.

Tutorial Progress

1. Relational Continuous Models (RCMs)
2. Lifted Variable Elimination
3. Conditions for Lifted Inference
4. Special Cases: Relational Gaussian Models
5. Lifted Relational Kalman Filtering (LRKF)

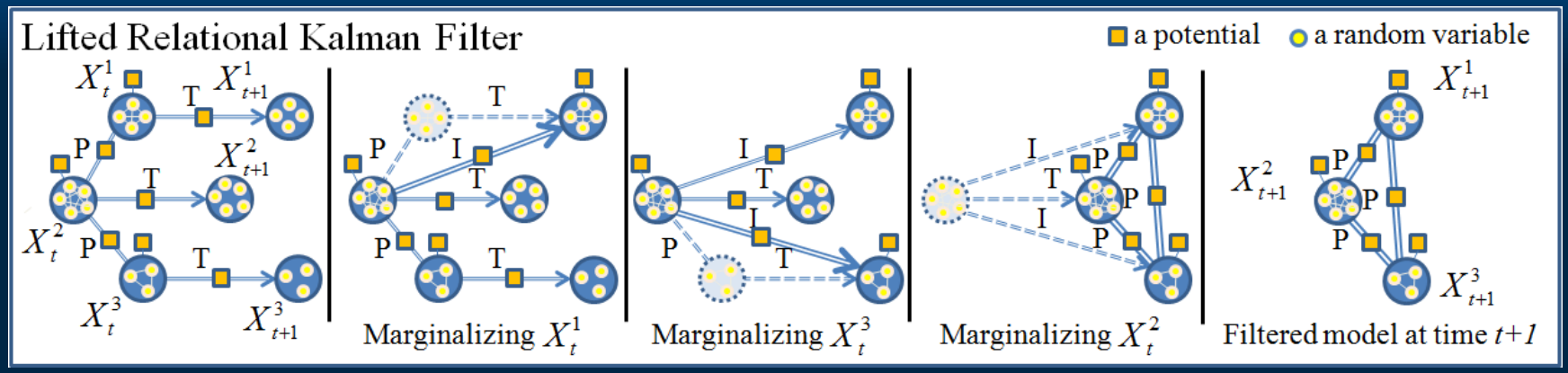
Lifted Relational Kalman Filtering
An extension to dynamic systems (IJCAI'11)



- RCMs can be extended to dynamic systems
  - State estimation: recursive elimination of all random variables in time t to yield a distribution over time t+1.
  - Assumption: Pairwise form for transition, observation models
- Limitations of lifted inference above (for RCMs)
  - **Individual observations** degenerate the model into a ground one (SPLITing or SHATTERing (Braz etal IJCAI'05))
- For Relational Kalman Filter:
  - Maintain a generalized relational structure given observations
  - Observations do not deteriorate the computational complexity.

Lifted Relational Kalman Filtering

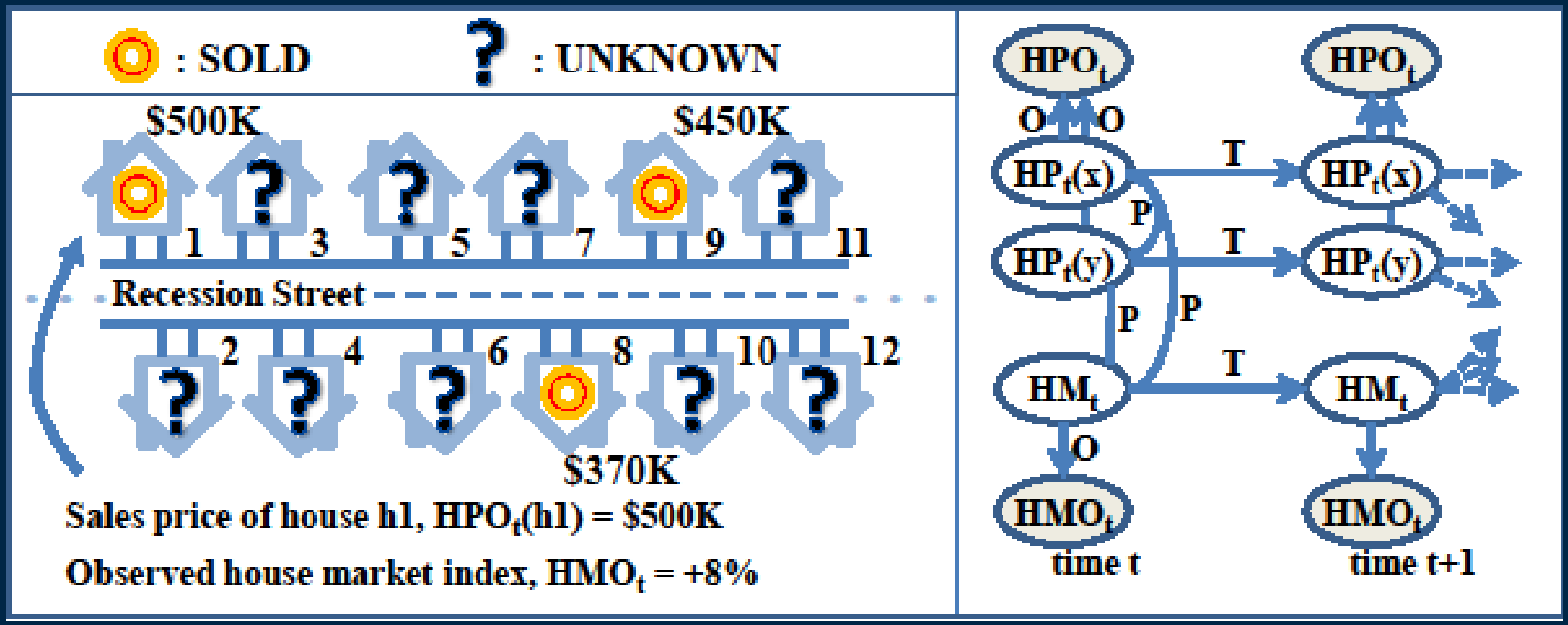
An extension to dynamic systems (IJCAI'11)



- Limitations of lifted inference above (for RCMs)
  - **Individual observations** degenerate the model into a ground one (SPLITing or SHATTERing (Braz etal IJCAI'05))
- For Relational Kalman Filter:
  - Maintain a generalized relational structure given observations
  - Observations do not deteriorate the computational complexity.

Lifted Relational Kalman Filtering
Example

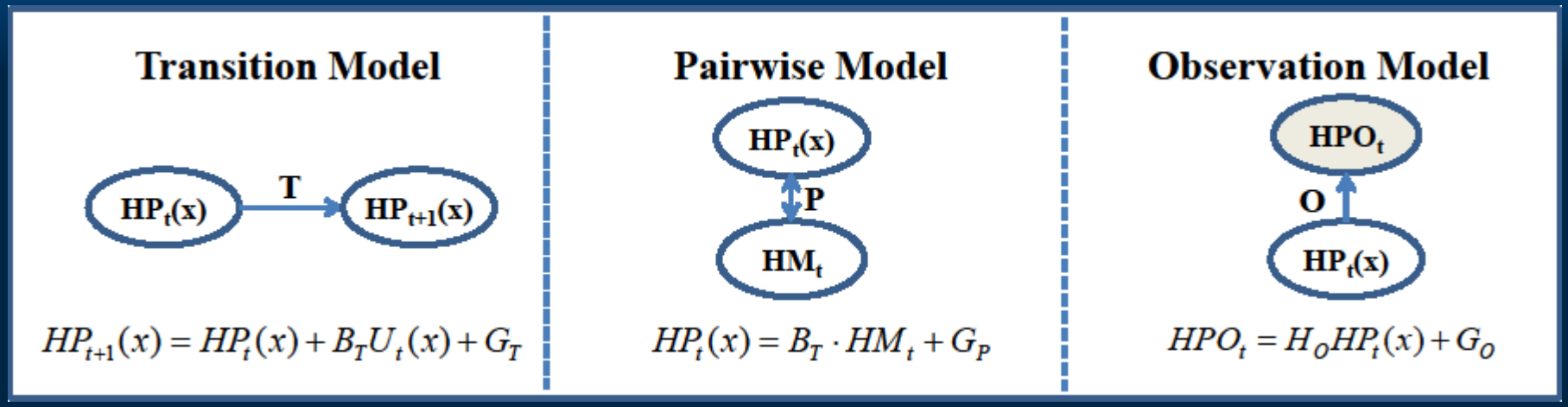
- A housing market example (updated every year)



O: Observation Model
P: Pairwise Model
T: Transition Model

Lifted Relational Kalman Filtering

Definitions: Relational Gaussian Models (RGMs)



- Relational Transition Models (RTM or T)
  - $U_t(x)$: user input for x at time t
  - G_r: Gaussian noise of the transition model.
- Relational Pairwise Models (RPM or P): i.e. RCM
- Relational Observation Models (ROM or O)
  - G_o: Gaussian noise of the observation model.

Examples of equations in LRKF

- Parameters for RTM, ROM & RPM.
- RTM: user inputs (e.g. the inflation rate*current price) and the variance.

$$RTM(HP_t(x), HP_{t+1}(x)) = \exp\left(-(HP_{t+1}(x) - HP_t(x) - \$8{,}000)^2 / 5000\right)$$

- ROM: observations (e.g. sold price) and the variance.

$$ROM(HP_t(x), HPO_t(x)) = \exp\left(-(HP_{t+1}(x) - \$500{,}000)^2 / 2000\right)$$

Small covariances for observations models

- RPM: almost identical to RCMs

$$RPM(HP_t(x), HP_t(x')) = \exp\left(-(HP_t(x) - HP_t(x'))^2 / 7000\right)$$

Small covariances in a relational atom

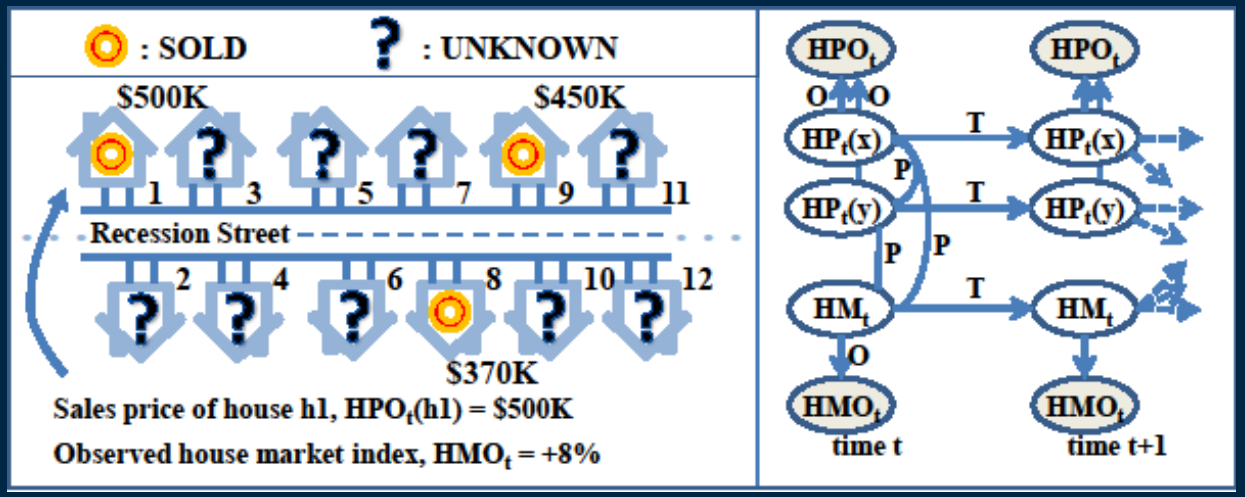
$$RPM(HP_t(x), HP_t(y)) = \exp\left(-(HP_t(x) - HP_t(y))^2 / 40000\right)$$

Large covariances between two relational atoms

Lifted Relational Kalman Filtering

Observations

- Remark 1: if two random variables, $HP_{t-1}(x_i)$ and $HP_{t-1}(x_j)$, have the same variance and covariances at t-1, given a same type of observations (observed or not), two random variables, $HP_t(x_i)$ and $HP_t(x_j)$, have the same variance and covariances at t.
- Remark 2: $HP_t(x_i)$ and $HP_t(x_j)$ may have different means.
- Example: HP_{2010}(1 recessions st) and HP_{2010}(9 recessions st) are sold in 2010. Then, the two have the same variance and covariances at 2011.



Lifted Relational Kalman Filtering

Main results



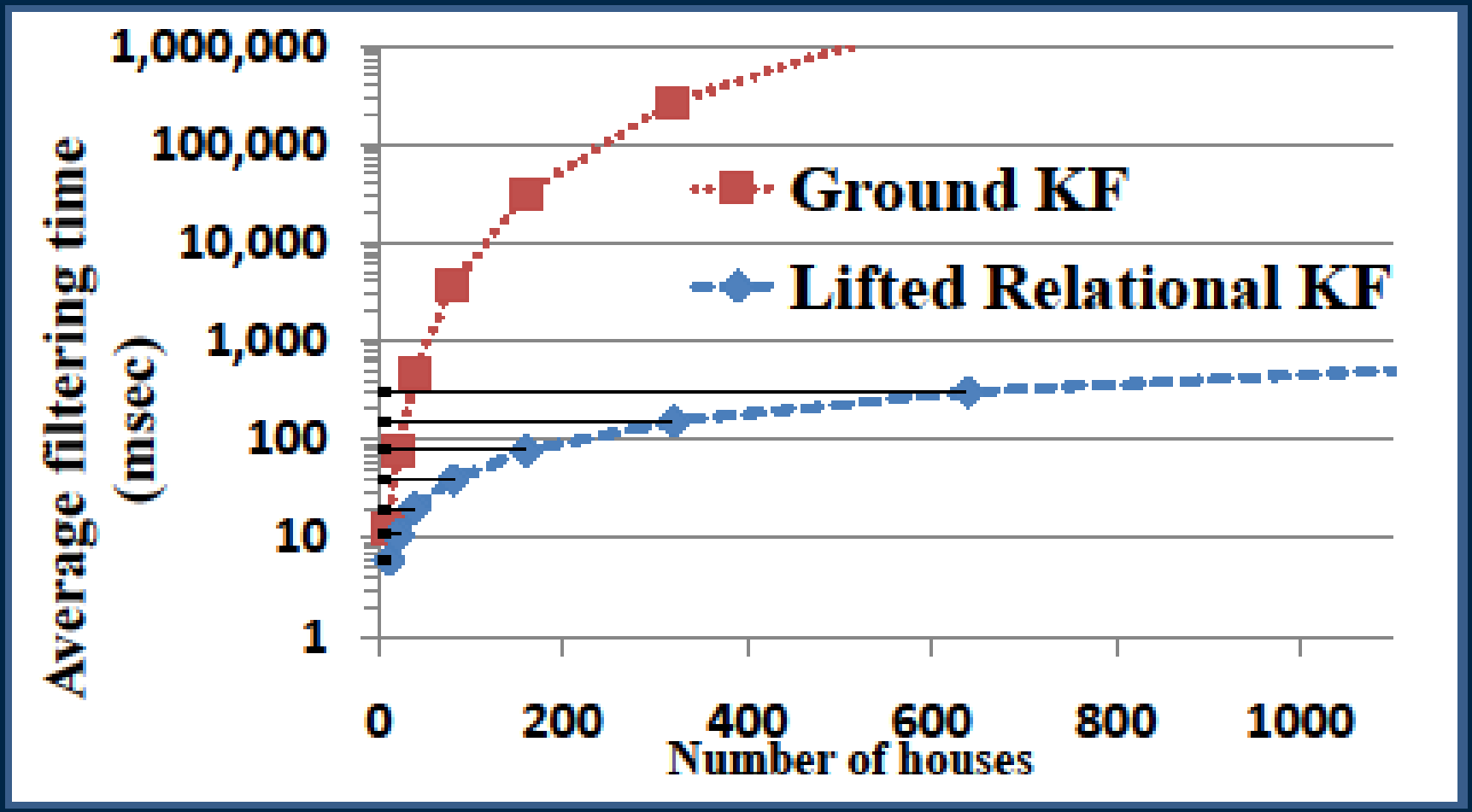
- Our method: maintain pair-wise relational structure between relational atoms with only variance and covariances. The means are stored at a ground level.
- Main results: exact inference (filtering) algorithm for LRKF takes $O(n|R|^2)$ (n = number of random variables, |R| = number of relational atoms*).
- This is the first exact lifted inference algorithm which handles individual observations without shattering.

* Individual observations do not degenerate the relational atoms.

Lifted Relational Kalman Filtering

Experimental results

Increase the number of houses



Conclusions

- Relational Continuous models
  - Inference time linear in the number of objects
- Key Insight:
  - Maintain representation with Variable Elimination: *product of pairwise potentials*
- Application: Lifted Kalman Filtering (IJCAI'11 Technical presentation)

Open Questions

- Which continuous models (beyond Gaussian CRMs) permit such methods?
- What is the penalty when approximating other models via pairwise CRMs?
- What is analogous to Counting Elimination in CRMs?

Appendix – More Details

Additional Details in the Papers

- The papers include
  - A condition which makes the relational normal is a probability density function.
  - A proof that the product of potentials is a probability density function given the condition.
  - Three conditions for exact lifted inference for general potentials (including non-Gaussian)
    - Analytically integrable
    - Closed under product operations
    - Closed under marginalization, thus represented with the product of relational pairwise form

Eliminating random variables at a ground level

- Details of marginalization

$$\int_{R(b_i)} \prod_{s \in S} \varphi_4(M(s), R(b_i)) dR(b_i) = \int_{R(b_i)} \exp\left(\sum_{s \in S} -\frac{(R(b_i) - M(s))^2}{\sigma^2} \right)$$

$$= \int_{R(b_i)} -\frac{|S|}{\sigma^2} R(b_i)^2 + \frac{2\left(\sum_{s \in S} M(s)\right)}{\sigma^2} R(b_i) - \frac{\left(\sum_{s \in S} M(s)^2\right)}{\sigma^2}$$

$$= c \cdot \exp\left(\frac{\left(\sum_{s \in S} M(s)\right)^2}{2\sigma^2 \cdot |S|} - \frac{\sum_{s \in S} M(s)^2}{2\sigma^2} \right)$$

$$= \varphi_5\left(M(s_1), M(s_2), \cdots, M(s_{|S|})\right)$$

Our method – Eliminating random variables in a lifted level with the Pair-wise form

- Generalization
  - We made some algebraic notations

$$X_{[m]} = \sum_{1 \le i \le m} x_i \qquad \text{e.g. } M_{[|S|]} = \sum_{s \in S} M(s)$$

$$X_{[m]^2} = \sum_{1 \le i \le m} x_i^2 \qquad \text{e.g. } M_{[|S|]^2} = \sum_{s \in S} M(s)^2$$

$$X_{[m][m]} = \sum_{1 \le i \le m} x_i \cdot x_j$$

$$\left(X_{[m]^2} \right)^2 = X_{[m]^2} + 2X_{[m][m]}$$

Eliminating random variables in a lifted level

- Details of marginalization

$$\int \prod_{F_j} \phi_{PF}(w(P_i), w(F_j))\, dw(P_i) = \int \exp\left(\sum_{F_j} \frac{(w(P_i) - w(F_j))^2}{\sigma^2} \right) dw(P_i)$$

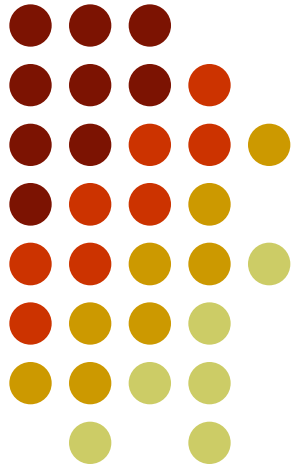
$$= \int \exp\left(-\frac{|F|}{\sigma^2} w(P_i)^2 + \frac{2\left(\sum_{F_j} w(F_j)\right)}{\sigma^2} w(P_i) - \frac{\left(\sum_{F_j} w(F_j)^2\right)}{\sigma^2} \right) dw(P_i)$$

$$= c \cdot \exp\left(\frac{\left(\sum_{F_j} w(F_j)\right)^2}{2\sigma^2 \cdot |F|} - \frac{\sum_{F_j} w(F_j)^2}{2\sigma^2} \right)$$

$$= c \cdot \prod_{F_j, F_{j'} (j \neq j')} \exp\left(-\frac{(w(F_j) - w(F_{j'}))^2}{2\sigma^2 \cdot |F|} \right) = c \cdot \prod_{F_j, F_{j'} (j \neq j')} \phi_{FF'}(w(F_j), w(F_{j'}))$$

Lifted Belief Propagation

Pedro Domingos
Dept. of Computer Science & Eng.
University of Washington



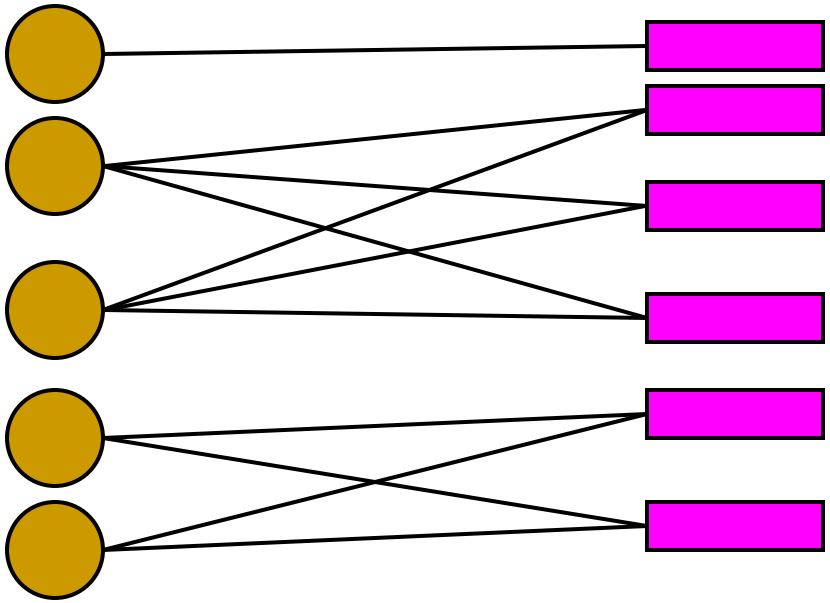
Motivation



- Exact inference in real-world probabilistic logical models is seldom feasible
- Need lifted approximate inference
- Also: Can we understand better how lifted probabilistic inference works?

Belief Propagation

$$\mu_{x \to f}(x) = \prod_{h \in n(x) \setminus \{f\}} \mu_{h \to x}(x)$$



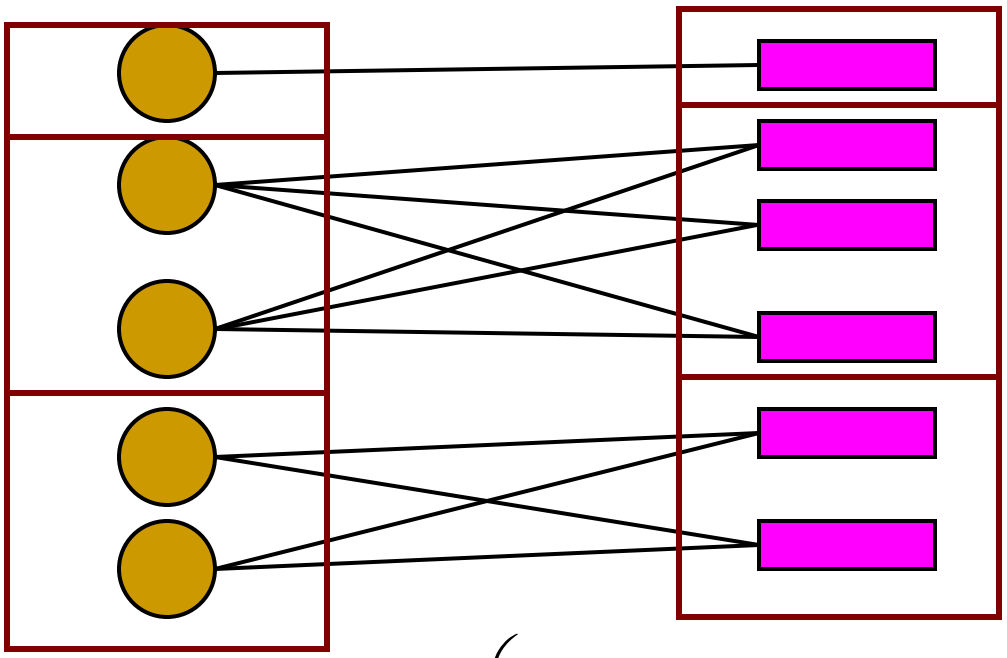
Nodes (x)

Features (f)

$$\mu_{f \to x}(x) = \sum_{\sim\{x\}} \left(e^{wf(x)} \prod_{y \in n(f) \setminus \{x\}} \mu_{y \to f}(y) \right)$$

Lifted Belief Propagation

$$\mu_{x \to f}(x) = \prod_{h \in n(x) \setminus \{f\}} \mu_{h \to x}(x)$$

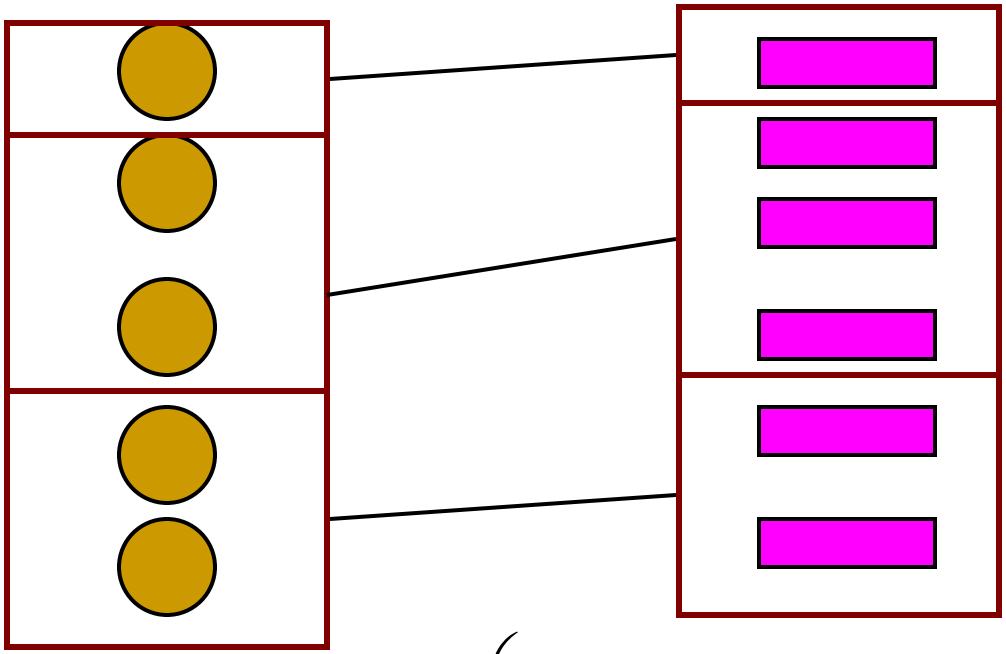


$$\mu_{f \to x}(x) = \sum_{\sim\{x\}} \left(e^{wf(x)} \prod_{y \in n(f) \setminus \{x\}} \mu_{y \to f}(y) \right)$$

Lifted Belief Propagation



$$\mu_{x\to f}(x) = \prod_{h\in n(x)\setminus\{f\}} \mu_{h\to x}(x)$$



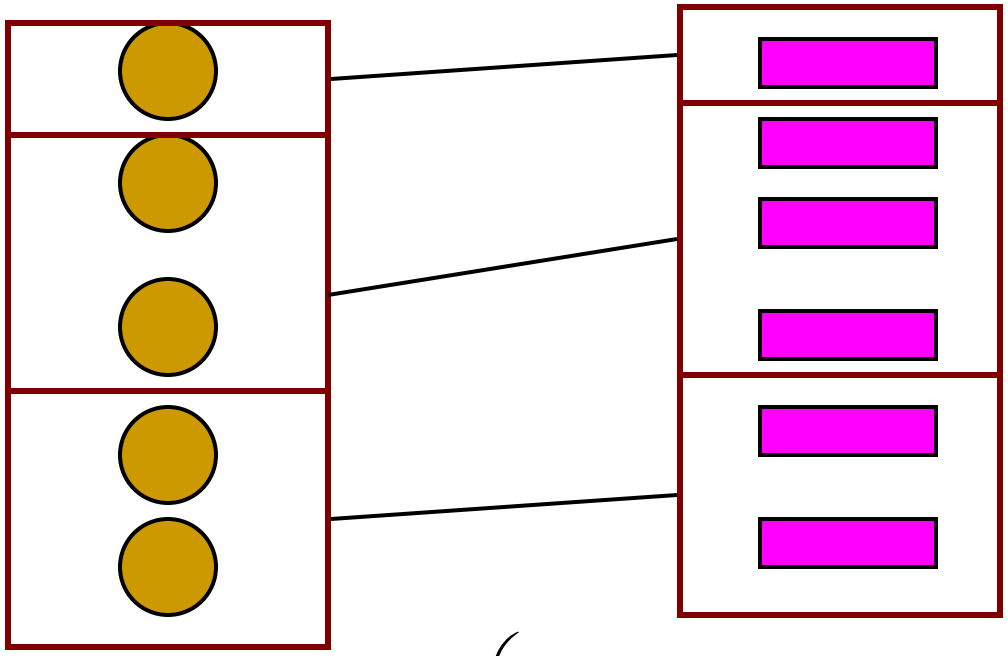
$$\mu_{f\to x}(x) = \sum_{\sim\{x\}} \left(e^{wf(x)} \prod_{y\in n(f)\setminus\{x\}} \mu_{y\to f}(y) \right)$$

Lifted Belief Propagation

α,β : Functions of edge counts

α

$$\mu_{x\to f}(x) = \beta \prod_{h\in n(x)\setminus\{f\}} \mu_{h\to x}(x)$$



Nodes (x)

Features (f)

$$\mu_{f\to x}(x) = \sum_{\sim\{x\}} \left(e^{wf(x)} \prod_{y\in n(f)\setminus\{x\}} \mu_{y\to f}(y) \right)$$

Lifted Belief Propagation



- Form **lifted network** composed of supernodes and superfeatures
  - **Supernode:** Set of ground atoms that all send and receive same messages throughout BP
  - **Superfeature:** Set of ground clauses that all send and receive same messages throughout BP
- Run belief propagation on lifted network
- Guaranteed to produce same results as ground BP
- Time and memory savings can be huge

Forming the Lifted Network



1. **Form initial supernodes**
   One per predicate and truth value
   (true, false, unknown)
2. **Form superfeatures by doing joins of their supernodes**
3. **Form supernodes by projecting superfeatures down to their predicates**
   Supernode = Groundings of a predicate with same number of projections from each superfeature
4. **Repeat until convergence**

Example

$$\forall x, y\ Smokes(x) \wedge Friends(x, y) \Rightarrow Smokes(y)$$

Evidence: Smokes(Ana)
Friends(Bob,Charles), Friends(Charles,Bob)

N people in the domain

Example

$$\forall x, y\ Smokes(x) \wedge Friends(x, y) \Rightarrow Smokes(y)$$

Evidence: Smokes(Ana)
Friends(Bob,Charles), Friends(Charles,Bob)

Intuitive Grouping :

Smokes(Ana)	Smokes(Bob) Smokes(Charles)	Smokes(James) Smokes(Harry) …

Initialization

Supernodes

Smokes(Ana)

Smokes(X)
X ≠ Ana

Friends(Bob, Charles)
Friends(Charles, Bob)

Friends(Ana, X)
Friends(X, Ana)

Friends(Bob,X)
X ≠ Charles
….

Superfeatures



Joining the Supernodes



Supernodes

Smokes(Ana)

Smokes(X)
X ≠ Ana

Friends(Bob, Charles)
Friends(Charles, Bob)

Friends(Ana, X)
Friends(X, Ana)

Friends(Bob,X)
X ≠ Charles
….

Superfeatures

Smokes(Ana)

Joining the Supernodes



Supernodes

Smokes(Ana)

Smokes(X)
X ≠ Ana

Friends(Bob, Charles)
Friends(Charles, Bob)

Friends(Ana, X)
Friends(X, Ana)

Friends(Bob,X)
X ≠ Charles
….

Superfeatures

Smokes(Ana) ∧
Friends(Ana, X)

Joining the Supernodes



Supernodes

Smokes(Ana)

Smokes(X)
X ≠ Ana

Friends(Bob, Charles)
Friends(Charles, Bob)

Friends(Ana, X)
Friends(X, Ana)

Friends(Bob,X)
X ≠ Charles
….

Superfeatures

Smokes(Ana) ∧
Friends(Ana, X) ⇒ Smokes(X)
X ≠ Ana

Joining the Supernodes



Supernodes

Smokes(Ana)

Smokes(X)
X ≠ Ana

Friends(Bob, Charles)
Friends(Charles, Bob)

Friends(Ana, X)
Friends(X, Ana)

Friends(Bob,X)
X ≠ Charles
….

Superfeatures

Smokes(Ana) ∧
Friends(Ana, X) ⇒ Smokes(X)
X ≠ Ana

Smokes(X) ∧ Friends(X, Ana)
⇒ Smokes(Ana)
X ≠ Ana

Joining the Supernodes



Supernodes

Smokes(Ana)

Smokes(X)
X ≠ Ana

Friends(Bob, Charles)
Friends(Charles, Bob)

Friends(Ana, X)
Friends(X, Ana)

Friends(Bob,X)
X ≠ Charles
….

Superfeatures

Smokes(Ana) ∧
Friends(Ana, X) ⇒ Smokes(X)
X ≠ Ana

Smokes(X) ∧ Friends(X, Ana)
⇒ Smokes(Ana)
X ≠ Ana

Smokes(Bob) ∧ Friends(Bob, Charles)
⇒ Smokes(Charles)
…

Joining the Supernodes



Supernodes

Smokes(Ana)

Smokes(X)
X ≠ Ana

Friends(Bob, Charles)
Friends(Charles, Bob)

Friends(Ana, X)
Friends(X, Ana)

Friends(Bob,X)
X ≠ Charles
….

Superfeatures

Smokes(Ana) ∧
Friends(Ana, X) ⇒ Smokes(X)
X ≠ Ana

Smokes(X) ∧ Friends(X, Ana)
⇒ Smokes(Ana)
X ≠ Ana

Smokes(Bob) ∧ Friends(Bob, Charles)
⇒ Smokes(Charles)
…

Smokes(Bob) ∧ Friends(Bob, X)
⇒ Smokes(X)
X ≠ Charles
…

Projecting the Superfeatures



Supernodes

Superfeatures

Smokes(Ana) ∧ Friends(Ana, X)
⇒ Smokes(X)
X ≠ Ana

Smokes(X) ∧ Friends(X, Ana)
⇒ Smokes(Ana)
X ≠ Ana

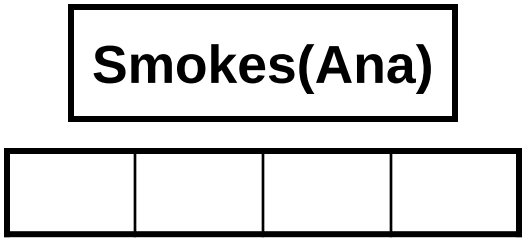
Smokes(Bob) ∧ Friends(Bob, Charles)
⇒ Smokes(Charles)
…

Smokes(Bob) ∧ Friends(Bob,X)
⇒ Smokes(X)
X ≠ Charles
…

Projecting the Superfeatures



Supernodes



Superfeatures

Smokes(Ana) ∧ Friends(Ana, X)
⇒ Smokes(X)
X ≠ Ana

Smokes(X) ∧ Friends(X, Ana)
⇒ Smokes(Ana)
X ≠ Ana

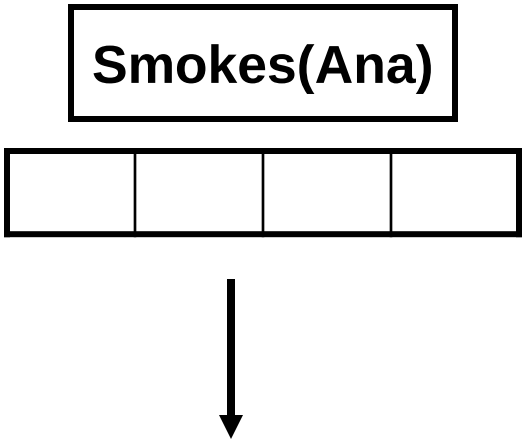
Smokes(Bob) ∧ Friends(Bob, Charles)
⇒ Smokes(Charles)
…

Smokes(Bob) ∧ Friends(Bob,X)
⇒ Smokes(X)
X ≠ Charles
…

Projecting the Superfeatures



Supernodes



Populate with projection counts

Superfeatures

Smokes(Ana) ∧ Friends(Ana, X)
⇒ Smokes(X)
X ≠ Ana

Smokes(X) ∧ Friends(X, Ana)
⇒ Smokes(Ana)
X ≠ Ana

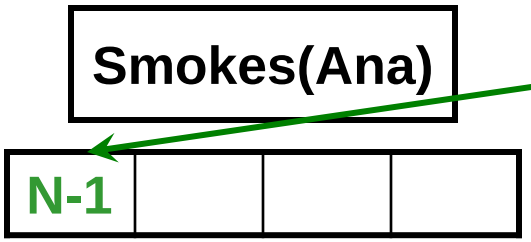
Smokes(Bob) ∧ Friends(Bob, Charles)
⇒ Smokes(Charles)
…

Smokes(Bob) ∧ Friends(Bob,X)
⇒ Smokes(X)
X ≠ Charles
…

Projecting the Superfeatures



Supernodes



Superfeatures

Smokes(Ana) ∧ Friends(Ana, X)
⇒ Smokes(X)
X ≠ Ana

Smokes(X) ∧ Friends(X, Ana)
⇒ Smokes(Ana)
X ≠ Ana

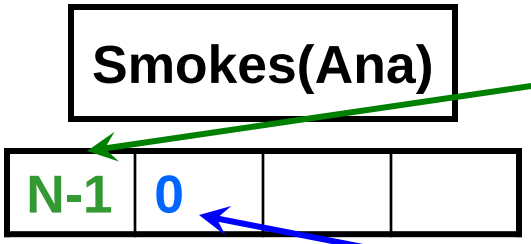
Smokes(Bob) ∧ Friends(Bob, Charles)
⇒ Smokes(Charles)
…

Smokes(Bob) ∧ Friends(Bob,X)
⇒ Smokes(X)
X ≠ Charles
…

Projecting the Superfeatures



Supernodes



Superfeatures

Smokes(Ana) ∧ Friends(Ana, X)
⇒ Smokes(X)
X ≠ Ana

Smokes(X) ∧ Friends(X, Ana)
⇒ Smokes(Ana)
X ≠ Ana

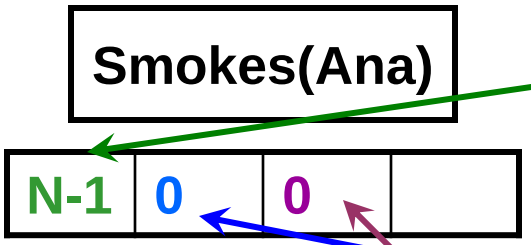
Smokes(Bob) ∧ Friends(Bob, Charles)
⇒ Smokes(Charles)
…

Smokes(Bob) ∧ Friends(Bob,X)
⇒ Smokes(X)
X ≠ Charles
…

Projecting the Superfeatures



Supernodes



Superfeatures

Smokes(Ana) ∧ Friends(Ana, X)
⇒ Smokes(X)
X ≠ Ana

Smokes(X) ∧ Friends(X, Ana)
⇒ Smokes(Ana)
X ≠ Ana

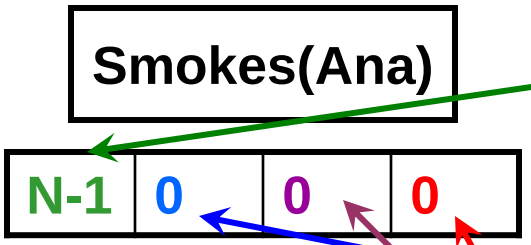
Smokes(Bob) ∧ Friends(Bob, Charles)
⇒ Smokes(Charles)
…

Smokes(Bob) ∧ Friends(Bob,X)
⇒ Smokes(X)
X ≠ Charles
…

Projecting the Superfeatures



Supernodes



Superfeatures

Smokes(Ana) ∧ Friends(Ana, X)
⇒ Smokes(X)
X ≠ Ana

Smokes(X) ∧ Friends(X, Ana)
⇒ Smokes(Ana)
X ≠ Ana

Smokes(Bob) ∧ Friends(Bob, Charles)
⇒ Smokes(Charles)
…

Smokes(Bob) ∧ Friends(Bob,X)
⇒ Smokes(X)
X ≠ Charles
…

Projecting the Superfeatures



Supernodes

Smokes(Ana)

N-1	0	0	0

Smokes(Bob)
Smokes(Charles)

0	1	1	N-3

Superfeatures

Smokes(Ana) ∧ Friends(Ana, X)
⇒ Smokes(X)
X ≠ Ana

Smokes(X) ∧ Friends(X, Ana)
⇒ Smokes(Ana)
X ≠ Ana

Smokes(Bob) ∧ Friends(Bob, Charles)
⇒ Smokes(Charles)
…

Smokes(Bob) ∧ Friends(Bob,X)
⇒ Smokes(X)
X ≠ Charles
…

Projecting the Superfeatures



Supernodes

Smokes(Ana)

N-1	0	0	0

Smokes(Bob)
Smokes(Charles)

0	1	1	N-3

Smokes(X)
X ≠ Ana, Bob, Charles

0	1	0	N-2

Superfeatures

Smokes(Ana) ∧ Friends(Ana, X)
⇒ Smokes(X)
X ≠ Ana

Smokes(X) ∧ Friends(X, Ana)
⇒ Smokes(Ana)
X ≠ Ana

Smokes(Bob) ∧ Friends(Bob, Charles)
⇒ Smokes(Charles)
…

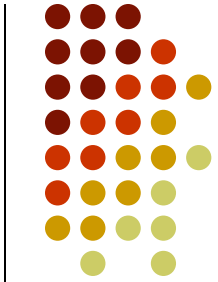
Smokes(Bob) ∧ Friends(Bob,X)
⇒ Smokes(X)
X ≠ Charles
…

Theorem



- There exists a unique minimal lifted network
- The lifted network construction algorithm finds it
- BP on lifted network gives same result as on ground network

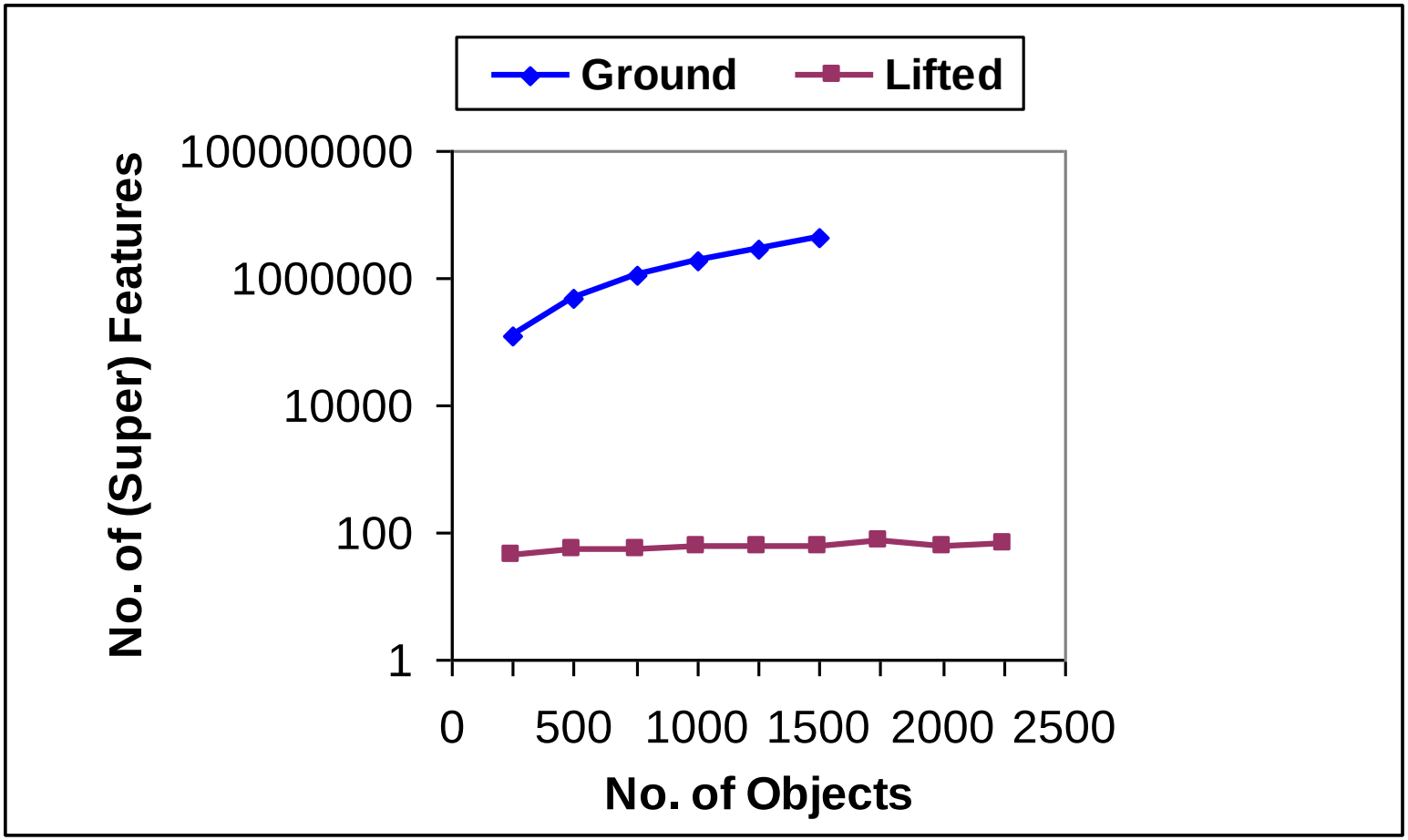
Representing Supernodes And Superfeatures



- **List of tuples:** Simple but inefficient
- **Resolution-like:** Use equality and inequality
- **Hypercubes:** Products of sets of values
- Beyond this: Open problem

Comparing Network Size

Friends & Smokers

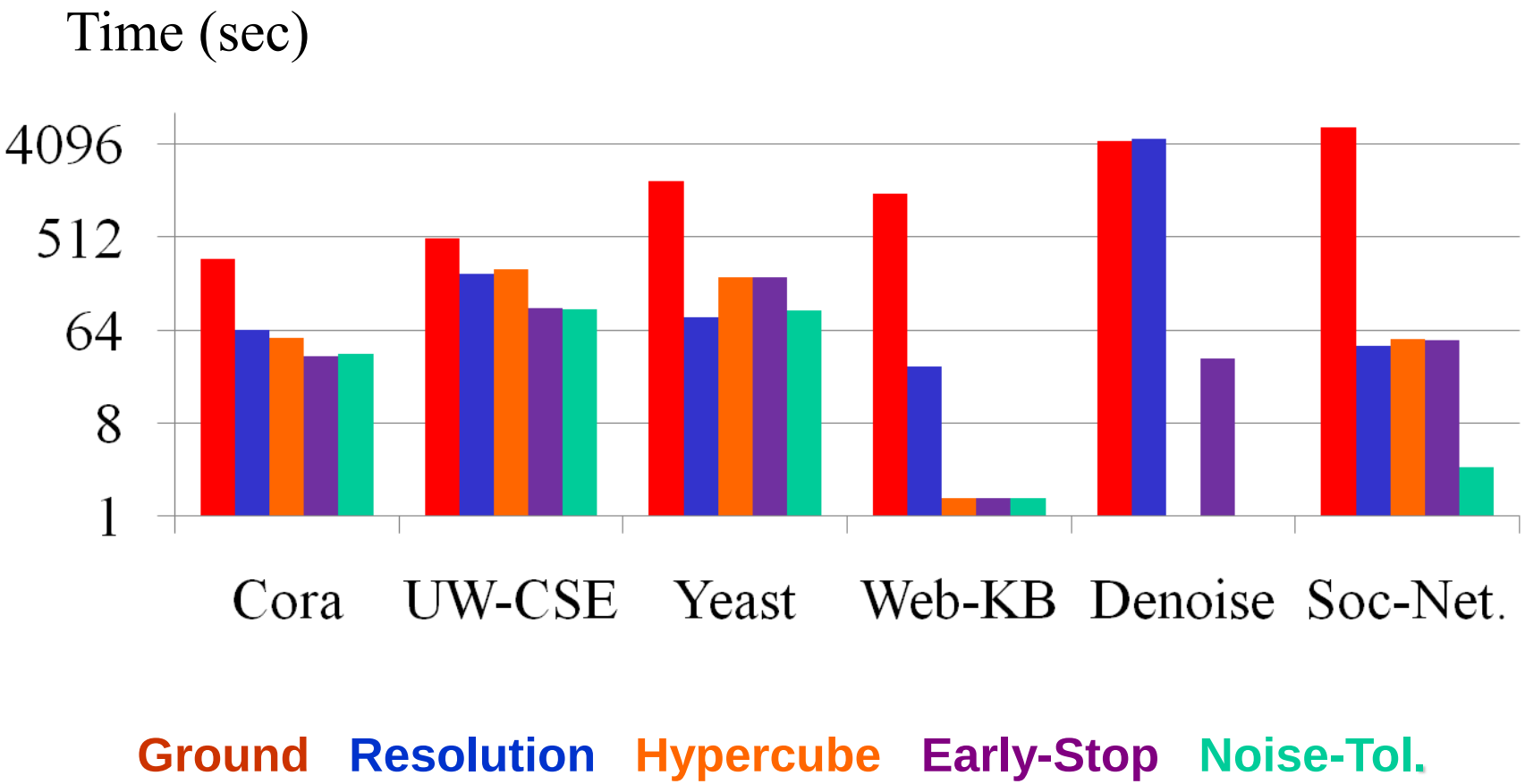


Approximate Lifting



- Group together nodes that send/receive **approximately** the same messages
- Two methods to date:
  - **Early stopping**
  - **Noise-tolerant hypercubes**
- **Pro:** Potentially much greater savings of time & memory
- **Con:** Potential loss of accuracy

Experimental Results



Online Lifting



- Avoid constructing lifted network from scratch at each time step
- Proposed solution: ΔLNC
- Idea:
  - Start with coarsest lifted network
  - Retrace LNC's steps
  - Only move tuples whose messages change

Video Segmentation

Cricket	BP	LBP-4
Initial number of nodes	76800	4520
Initial number of factors	306080	19965
Initial LNC time (s)	-	870.69
Average ΔLNC time (s)	-	18.29

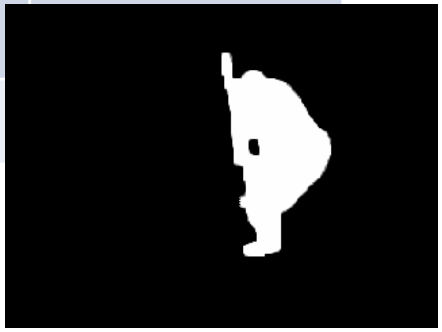
~48X



Last frame



BP labeling



LBP-4 labeling

[http://commons.wikimedia.org/wiki/File:This is batting.OGG]

Limitations of Lifted BP



- May not converge
- Convergence diagnosis is difficult
- May converge to poor approximation
- Cannot handle determinism
- Does not exploit logical structure
- Unclear relation to logical inference

Solved by **Probabilistic Theorem Proving**


Babak Ahmadi


Sriraam Natarajan


Youssef El Massaoudi


Fabian Hadiji

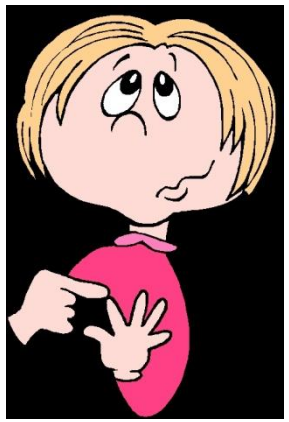

Scott Sanner


Marion Neumann



+

Counting Belief Propagation



=

0.6%



Kristian Kersting

Fraunhofer IAIS & Univ. of Bonn, Germany

universitätbonn Fraunhofer IAIS

TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

Motivation

- Exact inference approaches are rather complex, so far do not easily scale to realistic domains,and hence have only been applied to rather small artificial problems
- Moreover, they as well as lifted first-order BP assume a relational model over boolean random variables to be given

- Can we perform lifted BP on an arbitrary factor graph? If so, this would instantly yield lifted version of many well-known message passing algorithms such as Gaussian BP.

TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG
IJCAI-11

How do you spend your spare time?



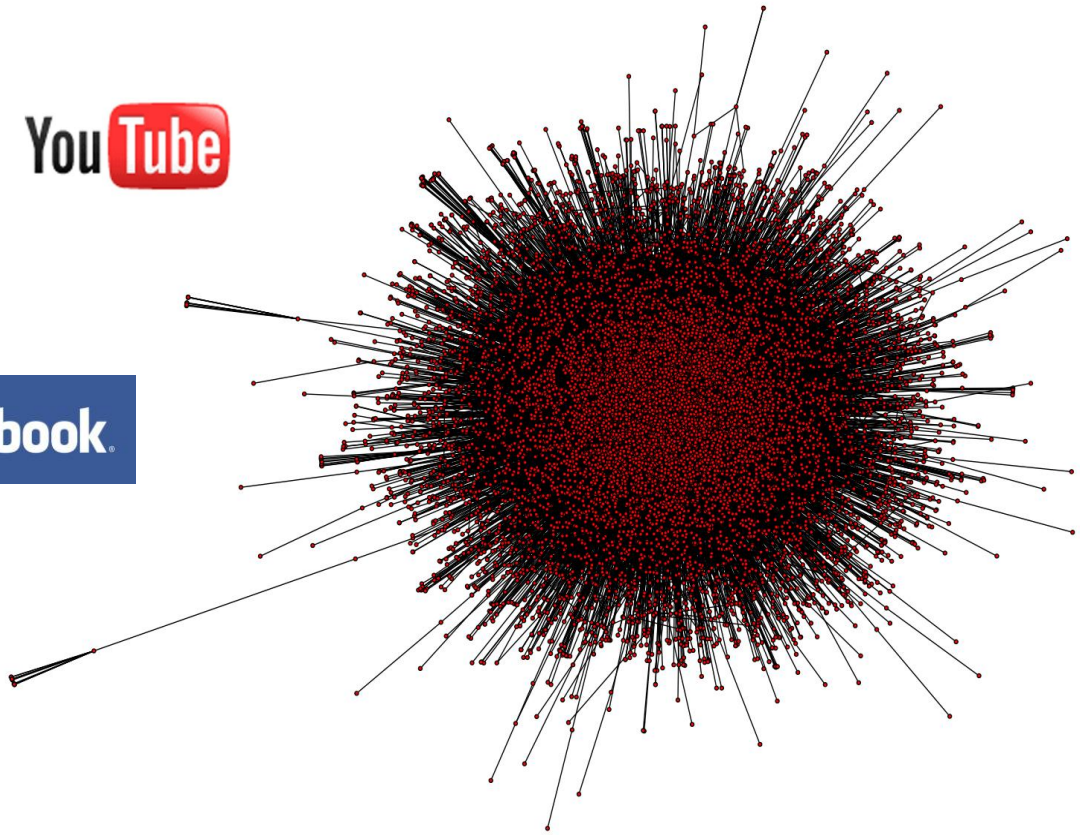
YouTube like media portals have changed the way users access media content in the Internet

Every day, millions of people visit social media sites such as Flickr, YouTube, and Jumpcut, among others, to share their photos and videos, …

while others enjoy themselves by searching, watching, commenting, and rating the photos and videos; what your friends like will bear great significance for you.

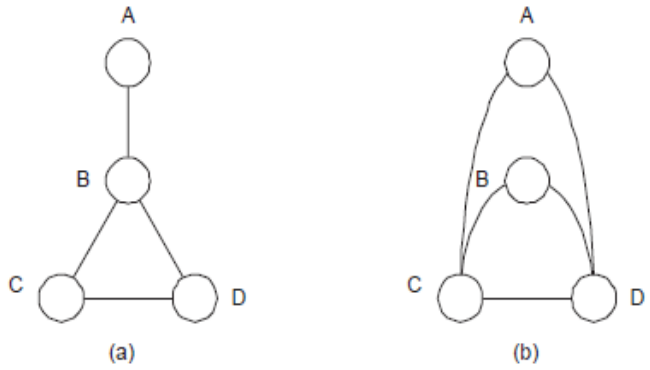
TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

But, how do we efficiently broadcast information in such networks?



[D. Bickson, D. Malkhi and D. Rabinowitz WDAS04]

Content Distribution using Stochastic Policies



Add an edge potential between any two nodes which can take a part from a common neighboring node.

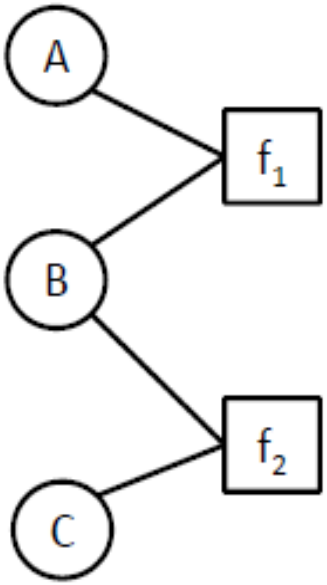
Large (distributed) networks, so Bickson et al. propose to use (loopy) belief propagation

TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG
IJCAI-11

The Sum-Product Algorithm aka Belief Propagation

- Iterative process in which neighboring variables "talk" to each other, passing **messages such as:**

"I (variable *x3) think that you (variable x2)* belong in these states with various likelihoods…"



A	B	f_1
True	True	1.2
True	False	1.4
False	True	2.0
False	False	0.4

C	B	f_2
True	True	1.2
True	False	1.4
False	True	2.0
False	False	0.4



$$\mu_{X\to f}(x) = \prod_{h\in \mathrm{nb}(X)\setminus\{f\}} \mu_{h\to X}(x)$$

$$\mu_{f\to X}(x) = \sum_{\neg\{x\}} \left(f(\mathbf{x}) \prod_{y\in \mathrm{nb}(f)\setminus\{X\}} \mu_{y\to f}(y) \right)$$

IJCAI-11 TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE JULY 16-22, 2011 CCIB BARCELONA, CATALONIA (SPAIN) WWW.IJCAI-11.ORG

Loopy Belief Propagation

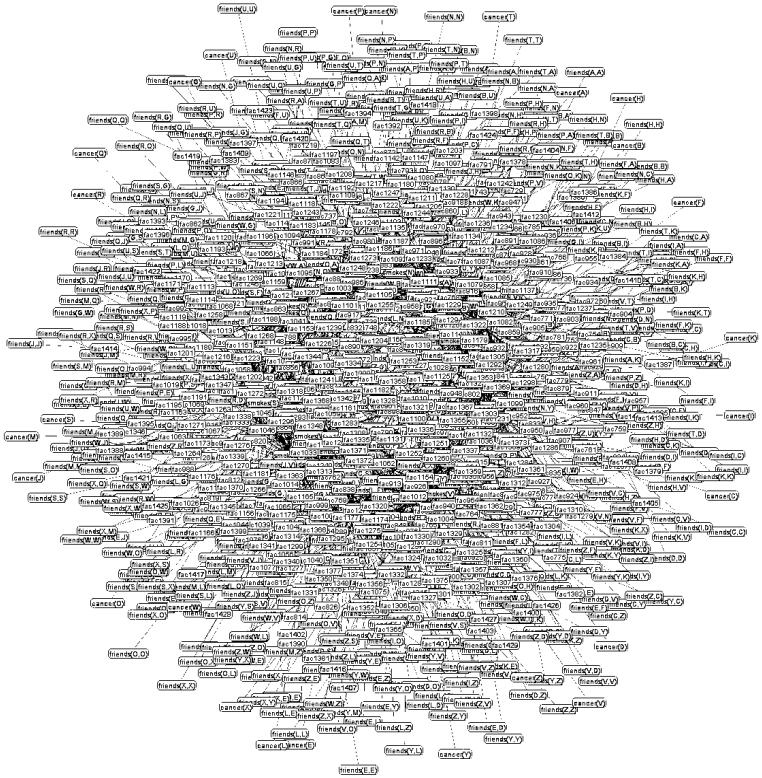
- After enough iterations, this series of conversations is **likely** to converge to a consensus that determines the marginal probabilities of all the variables.

- Sum-Product/BP
  - **(1) update messages until convergence**
  - **(2) compute single node marginals**

- Variants exist for solving
  - SAT problems,
  - systems of linear equations,
  - matching problems and
- for abitrary distributions (based on sampling)

TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011 CCIB BARCELONA, CATALONIA (SPAIN) WWW.IJCAI-11.ORG
IJCAI-11

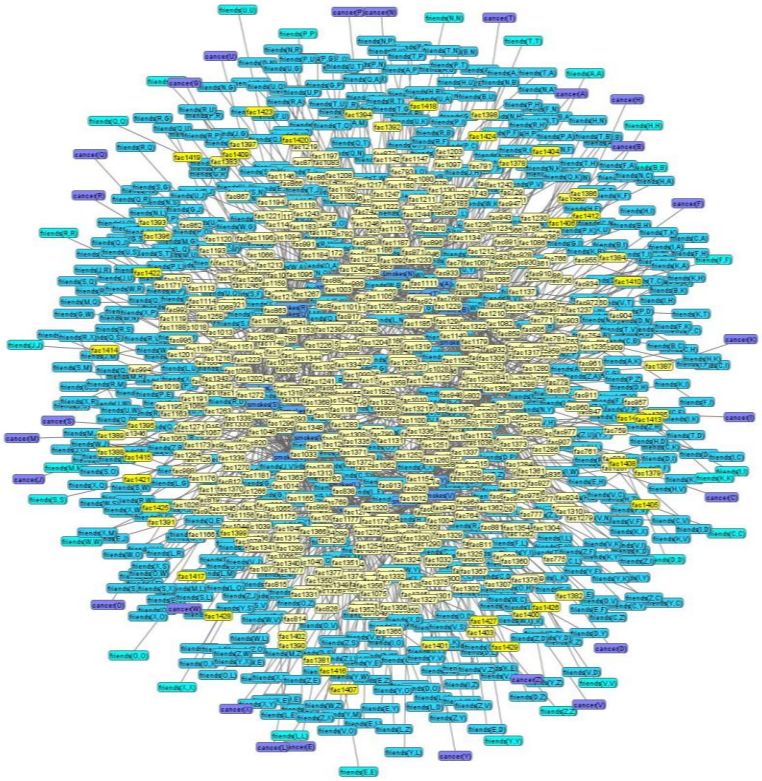
Social Network Analysis



IJCAI-11
TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

Social Network Analysis

A lot of shared factors!



Can we exploite the symmetries to speed up belief propagation?

IJCAI-11
TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

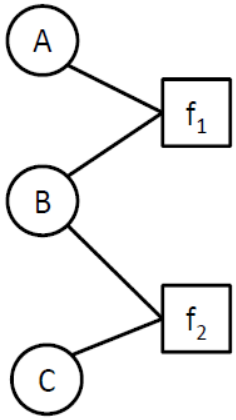
http://www-kd.iai.uni-bonn.de/index.php?page=software_details&id=16

Yes, we can: Lifted (Loopy) Belief Propagation

[Singla, Domingos AAAI08; Kersting, Ahmadi, Natarajan UAI09]

Counting shared factors can result in great efficiency gains for (loopy) belief propagation





A	B	f_1
True	True	1.2
True	False	1.4
False	True	2.0
False	False	0.4

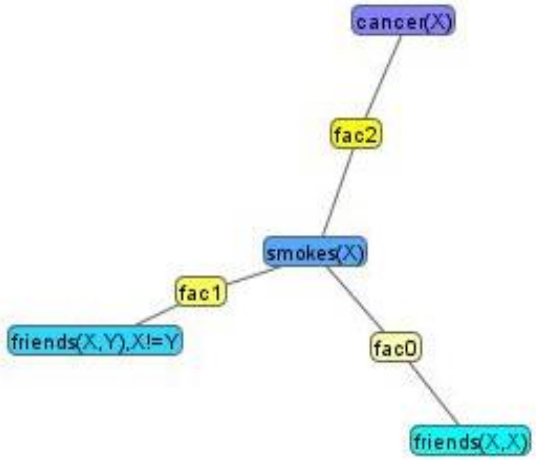
C	B	f_2
True	True	1.2
True	False	1.4
False	True	2.0
False	False	0.4

identical

Shared factors appear more often than you think in relevant real world problems

TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG
IJCAI-11

Social Network Analysis



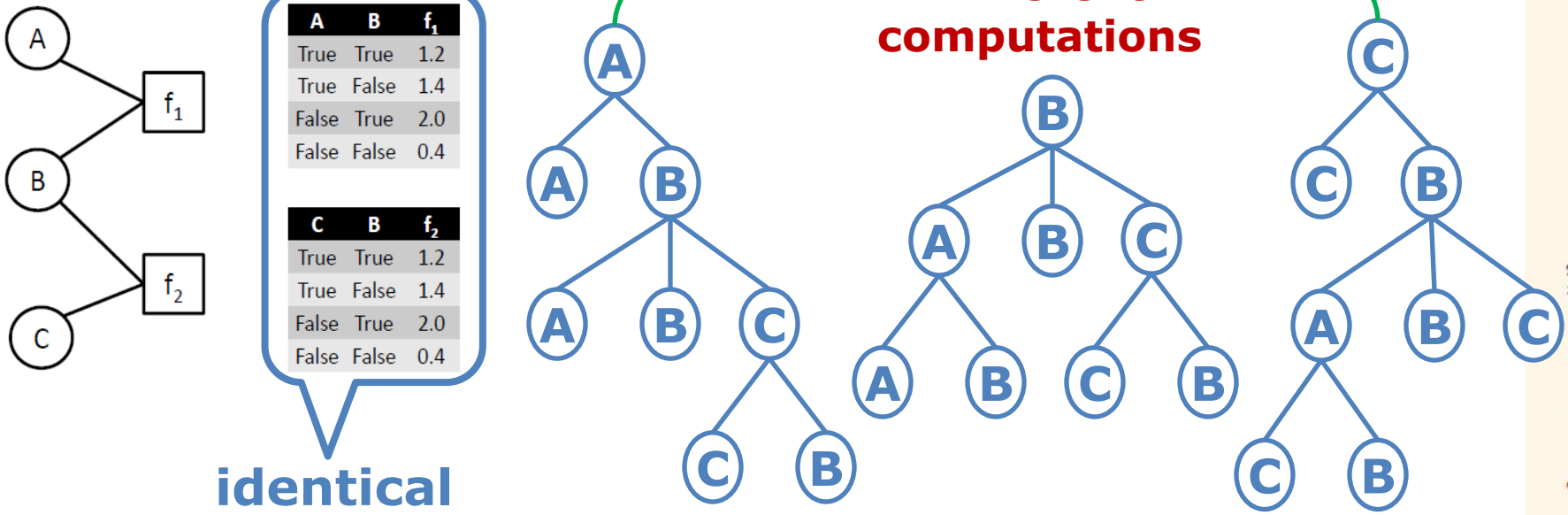
But how do we get there?

Actually, the complex network collapses to a beautifully small network

TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG
IJCAI-11

Computation Trees …

… the tree-structured unrolling of the underlying graph rooted at the nodes

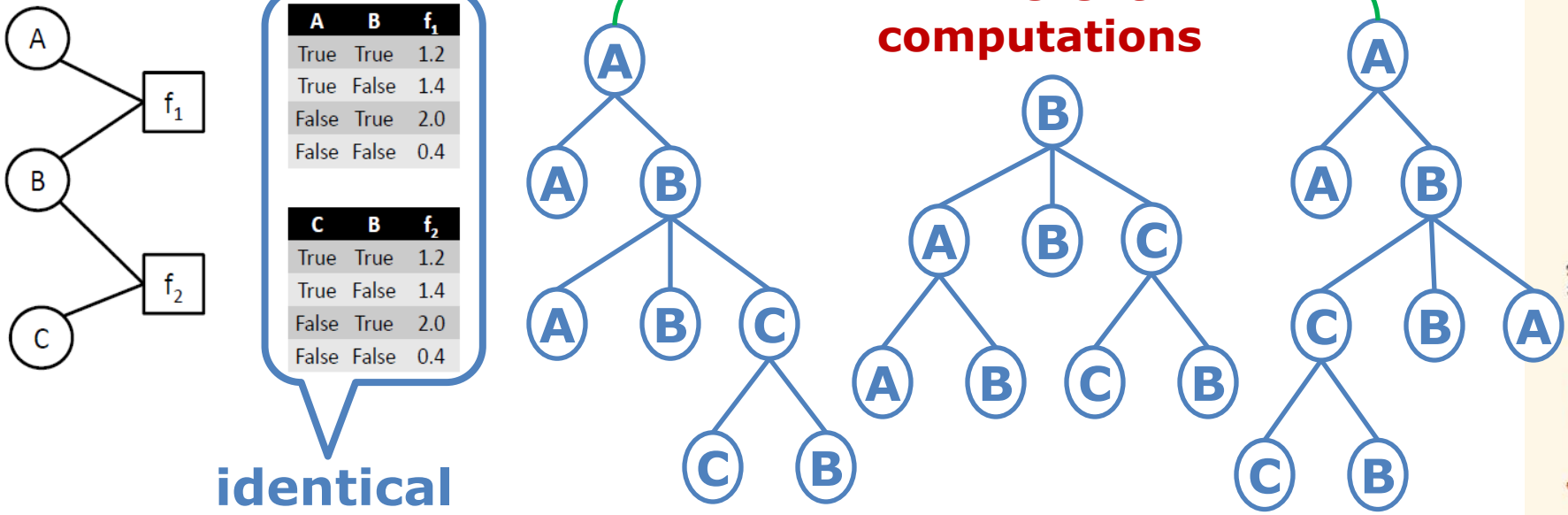


Group nodes together that have isomorphic computation trees

TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

Computation Trees …

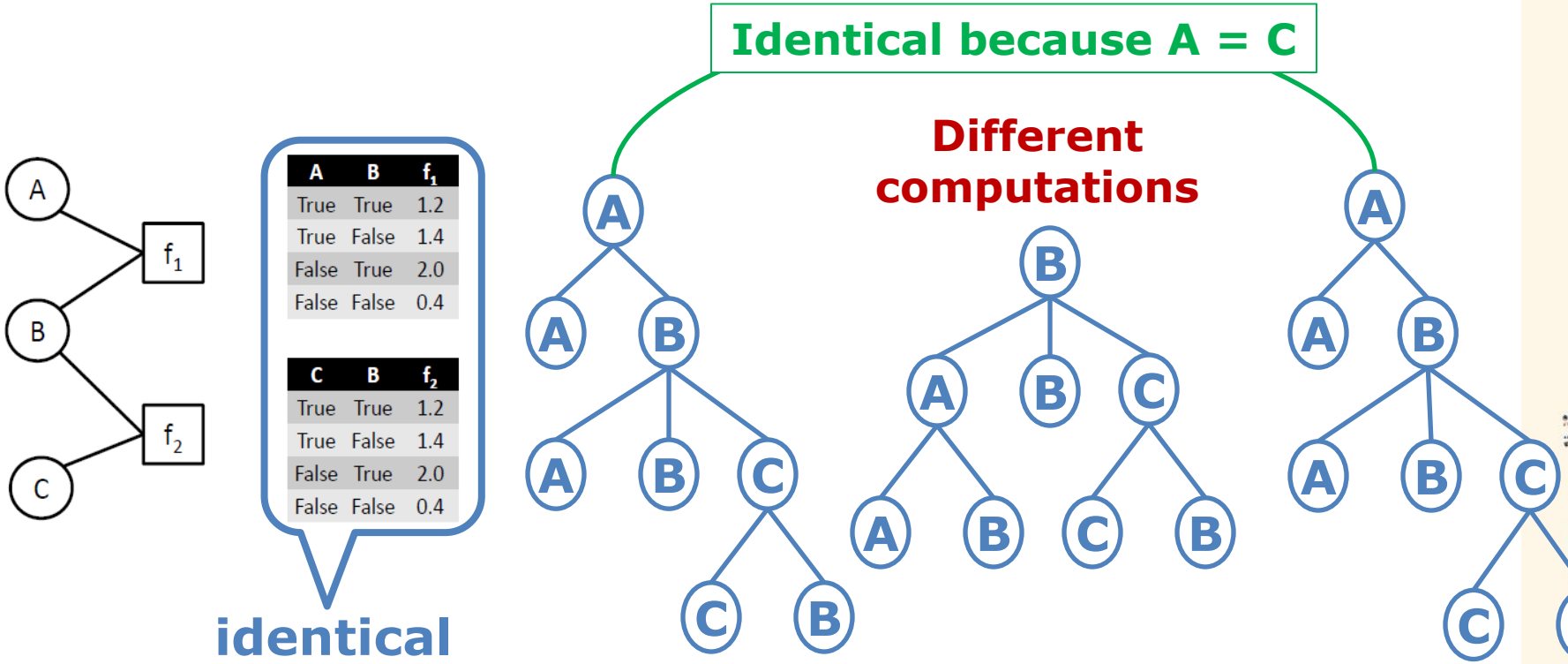
… the tree-structured unrolling of the underlying graph rooted at the nodes



TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

Computation Trees …

… the tree-structured unrolling of the underlying graph rooted at the nodes



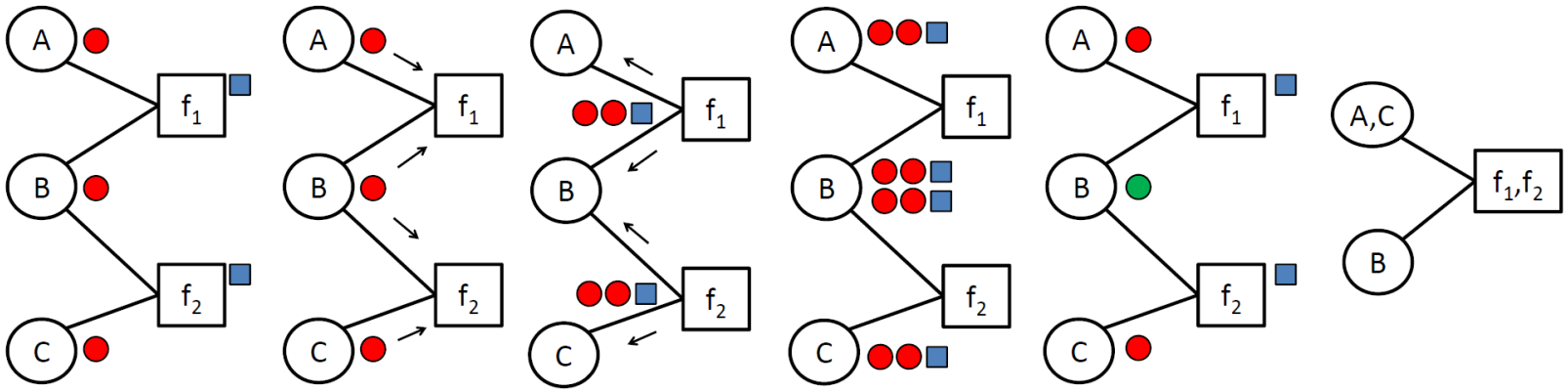
How do we compute this efficiently?

Reorder children

TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG
IJCAI-11

Step 1: Compression/Bisimulation

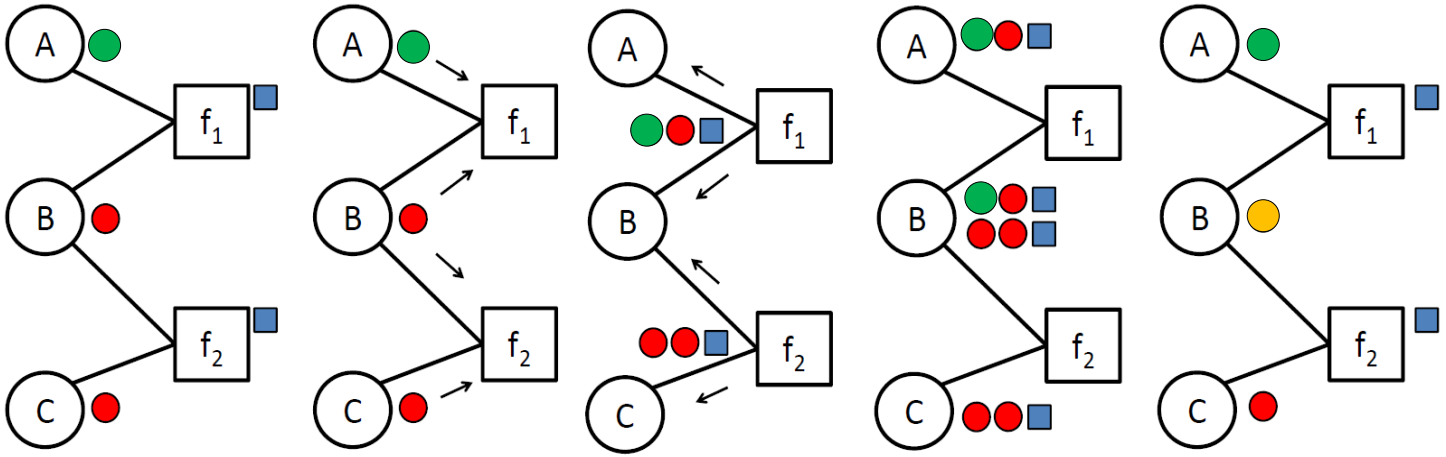
Instead of sending BP messages, send around color messages



- Guaranteed to converge in O(|E|*N)
- Closely related to computing equitable partitions of graphs

TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG
IJCAI-11

Indeed, evidence influences the lifting

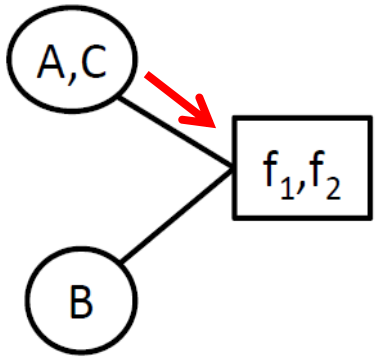


- Assume, we have only observed A

So, we have a compressed factor graph. What do we do with it?

IJCAI-11 TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE JULY 16-22, 2011 CCIB BARCELONA, CATALONIA (SPAIN) WWW.IJCAI-11.ORG

Step 2: Modified Belief Propagation



$$\mu_{\mathfrak{x}\to\mathfrak{f}}(x) = \mu_{\mathfrak{f}\to\mathfrak{x}}(x)^{c(\mathfrak{f},\mathfrak{x})-1} \cdot \prod_{\mathfrak{h}\in \mathrm{nb}(\mathfrak{x})\setminus\{\mathfrak{f}\}} \mu_{\mathfrak{h}\to\mathfrak{x}}(x)^{c(\mathfrak{h},\mathfrak{x})}$$

$$b_i(x_i) = \prod_{\mathfrak{f}\in \mathrm{nb}(\mathfrak{x}_i)} \mu_{\mathfrak{f}\to\mathfrak{x}_i}(x_i)^{c(\mathfrak{f},\mathfrak{x})}$$

Only difference* : counts c(f,x)
How often do elements of superfactor F send messages to elements of supernode X

***actually also depends on the position at which the element appears in the factor**

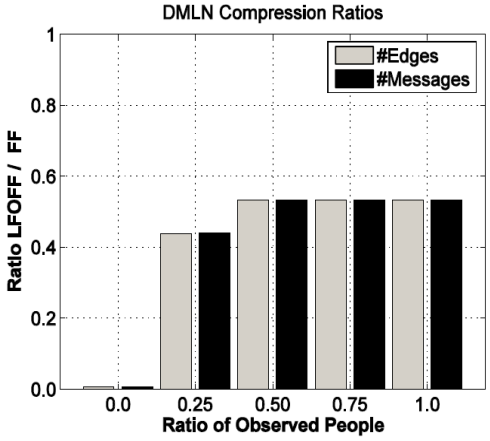
TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011 CCIB BARCELONA, CATALONIA (SPAIN) WWW.IJCAI-11.ORG

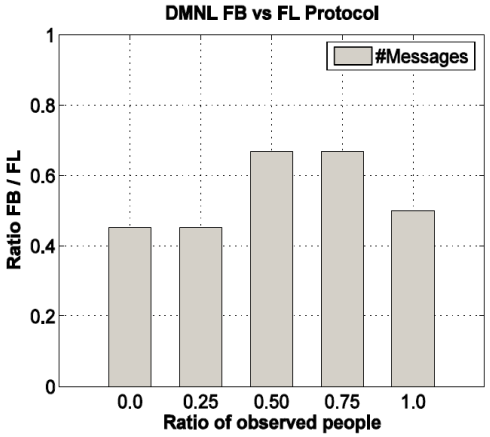
Lifted Factored Frontier

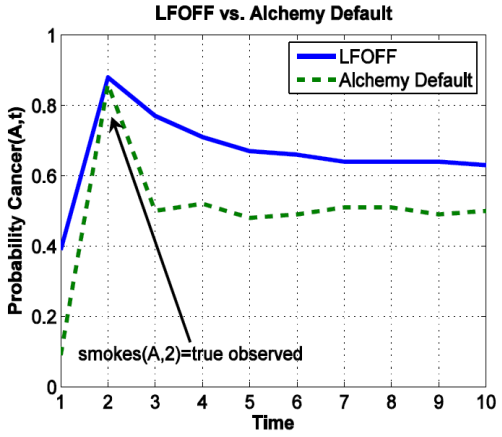
Temporal Models: Random variables easily become correlated over time by virtue of sharing common influences in the past.

Apriori most people do not smoke	$\neg\texttt{Smokes}(x,0)$
Apriori most people do not have cancer	$\neg\texttt{Cancer}(x,0)$
Apriori most people are not friends	$\neg\texttt{Friends}(x,y,0)$
Smoking causes cancer	$\texttt{Smokes}(x,t) \Rightarrow \texttt{Cancer}(x,t)$
Friends have similar smoking habits	$\texttt{Friends}(x,y,t) \Rightarrow (\texttt{Smokes}(x,t) <=> \texttt{Smokes}(y,t))$
Most friends stay friends	$\texttt{Friends}(x,y,t) \Leftrightarrow \texttt{Friends}(x,y,\texttt{succ}(t))$
Most smokers stay smokers	$\texttt{Smokes}(x,t) \Leftrightarrow \texttt{Smokes}(x,\texttt{succ}(t))$

20 people over 10 time steps. Max number of friends 5. Cancer never observed. Time step randomly selected.

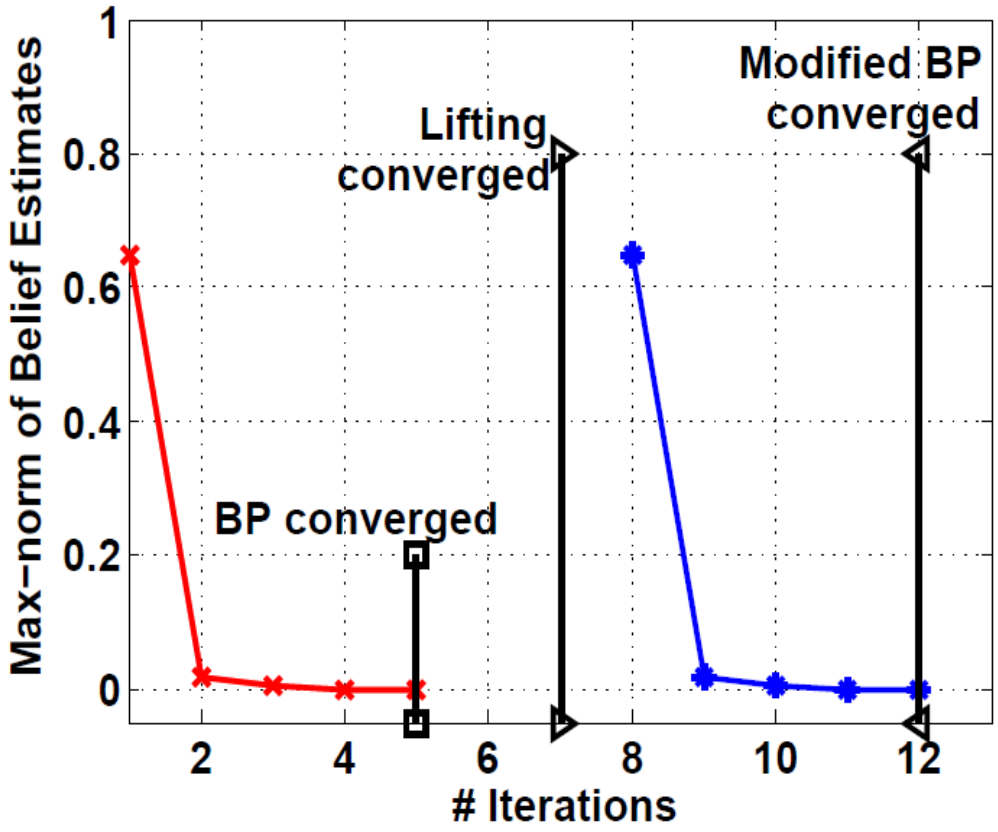






Are we satisfied by lifted message passing?

Content Distribution (Gnutella): Lifted BP vs. BP



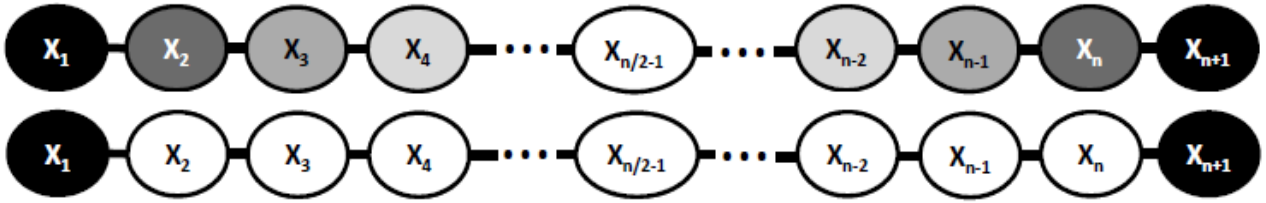


IJCAI-11 TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE JULY 16-22, 2011 CCIB BARCELONA, CATALONIA (SPAIN) WWW.IJCAI-11.ORG

[El Massaoudi, Kersting, Ahmadi, Hadiji AAAI10]

Message Errors to the Rescue

Ihler et al. 05:
BP message errors decay along paths
LBP may spuriously assume some nodes send and receive different messages and, hence, produce pessimistic lifted network

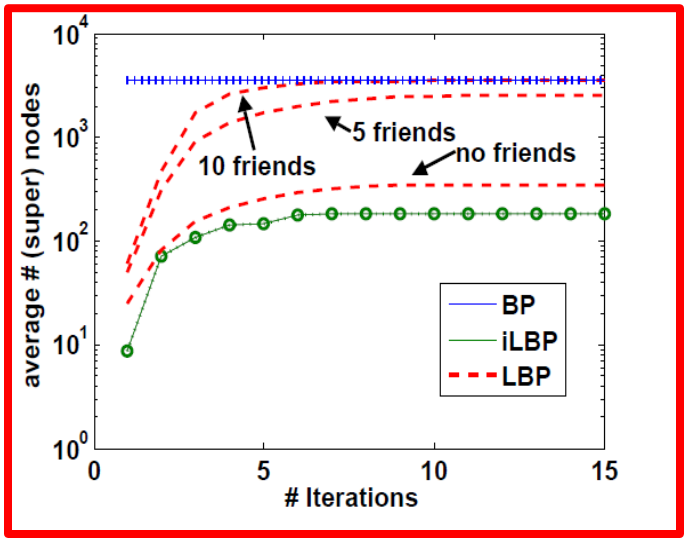


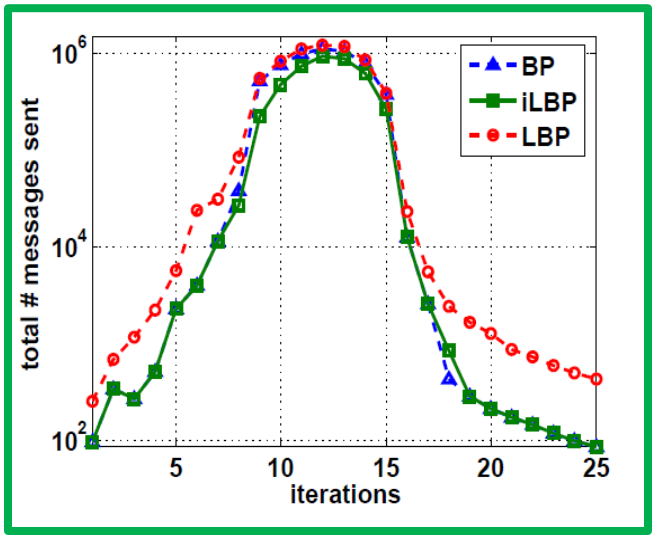
Informed Lifted BP:
(1)Alternate Lifting and Modified BP
(2)Cluster according to actually computed messages

Make use of decaying errors already at lifting time

TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

Social Networks/Lifted Content Distribution





- 1 file, Gnutella snapshort

- 10876 nodes **and** 39994 edges

iLBP 4.3 mio mess.< BP 5.8 mio mess.< LBP 6.4 mio mess.

- On a different network:

iLBP 1.9mio < LBP 2.9mio < BP 5.8mio

Are we really satisfied by LBP?

TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011 CCIB BARCELONA, CATALONIA (SPAIN) WWW.IJCAI-11.ORG
IJCAI-11

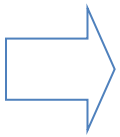
[Kesting, Ahmadi, Natarajan UAI09, Hadiji, Kersting, Ahmadi StarAI10]

Yes, because we can also lift satisfiability

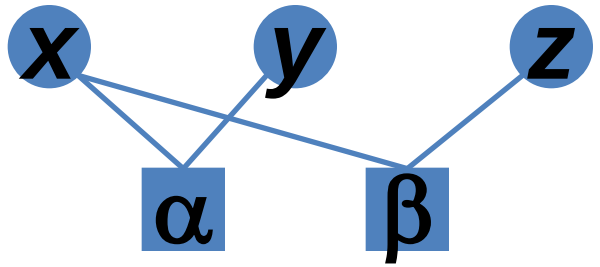
- BPCount, warning and survey propagation can also be lifted

SAT Problem:

$$\underbrace{(x \vee y)}_{\alpha} \wedge \underbrace{(\neg x \vee z)}_{\beta}$$



Factor Graph:



x	y	$f_\alpha(x,y)$
0	0	0
0	1	1
1	0	1
1	1	1

x	z	$f_\beta(x,z)$
0	0	1
0	1	1
1	0	0
1	1	1

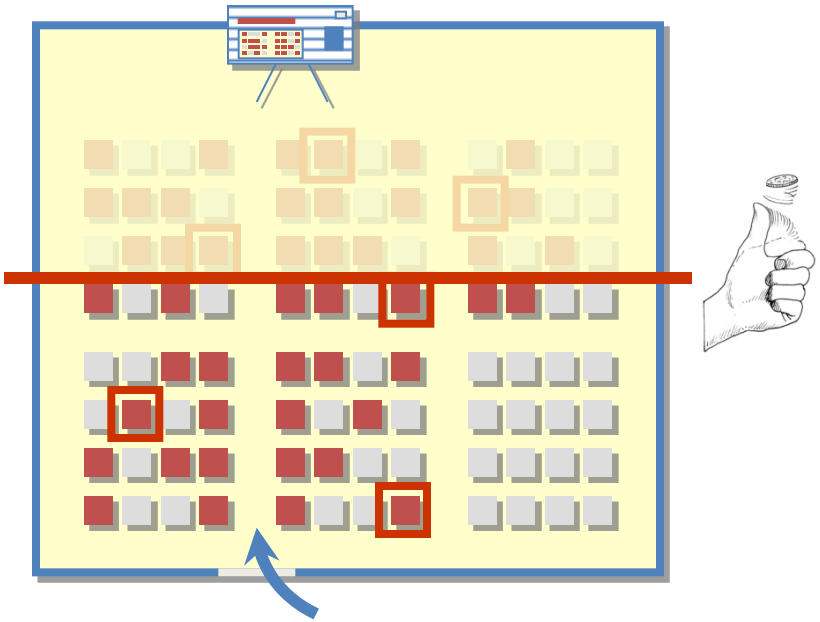
[Slides due to Lukas Kroc, Ashish Sabharwal, Bart Selman]

TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011 CCIB BARCELONA, CATALONIA (SPAIN) WWW.IJCAI-11.ORG
IJCAI-11

[Kesting, Ahmadi, Natarajan UAI09, Hadiji, Kersting, Ahmadi StarAI10]

Yes, because we can also lift satisfiability

- BPCount, warning and survey propagation can also be lifted



Idea:

- **Identify a "balanced" row split or column split (roughly equal number of solutions on each side)**
  - **Use marginals for estimate**
- **Pick one side *at random***
- **Count on that side recursively**
- **Multiply result by 2**

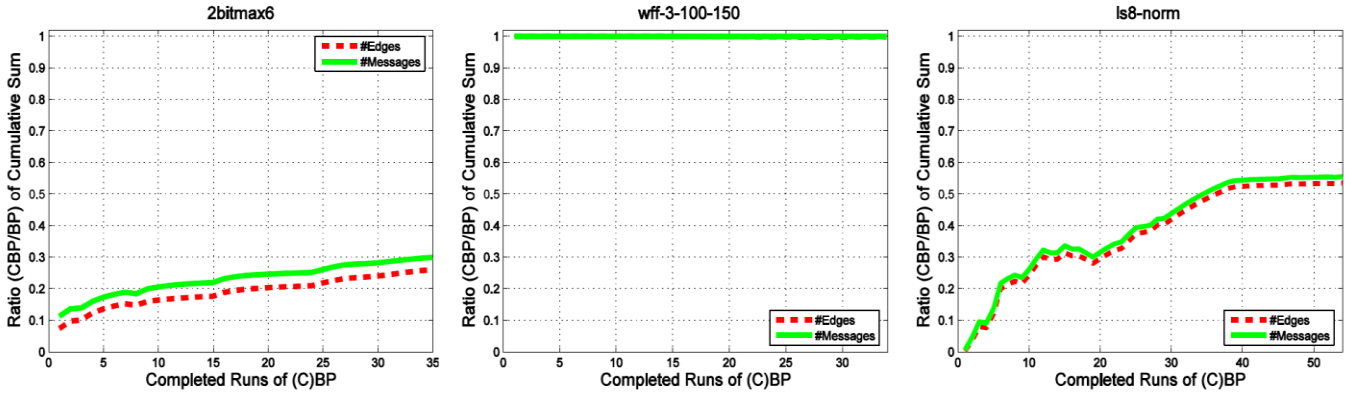
[similar to decimation]

[Slides due to Lukas Kroc, Ashish Sabharwal, Bart Selman]

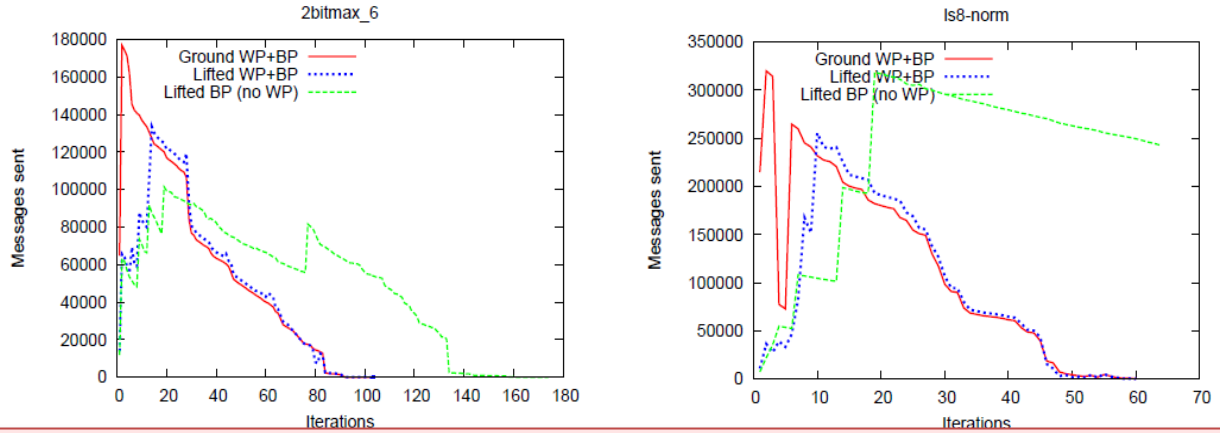
TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011 CCIB BARCELONA, CATALONIA (SPAIN) WWW.IJCAI-11.ORG
IJCAI-11

[Kesting, Ahmadi, Natarajan UAI09, Hadiji, Kersting, Ahmadi StarAI10]

Model Counting



SAT



Similar, we can lift solvers for systems of linear equations and matrix inversion

TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG
IJCAI-11

[Shental, Bickson, Siegel, Wolf, Dolev. ISIT08]

Solving Systems of Linear equations

- We can rewrite the task $A \cdot x - b = 0$ as a probabilistic inference problem

$$q(x) = x^t \cdot A \cdot x / 2 - b^t \cdot x \qquad \frac{\partial q(x)}{\partial x} = A \cdot x - b$$

- Thus, the minimum of q(x) is our solution. This, however, is equivalent to the maximum of

$$p(x) \propto \exp\left(-x^t \cdot A \cdot x / 2 + b^t \cdot x\right)$$

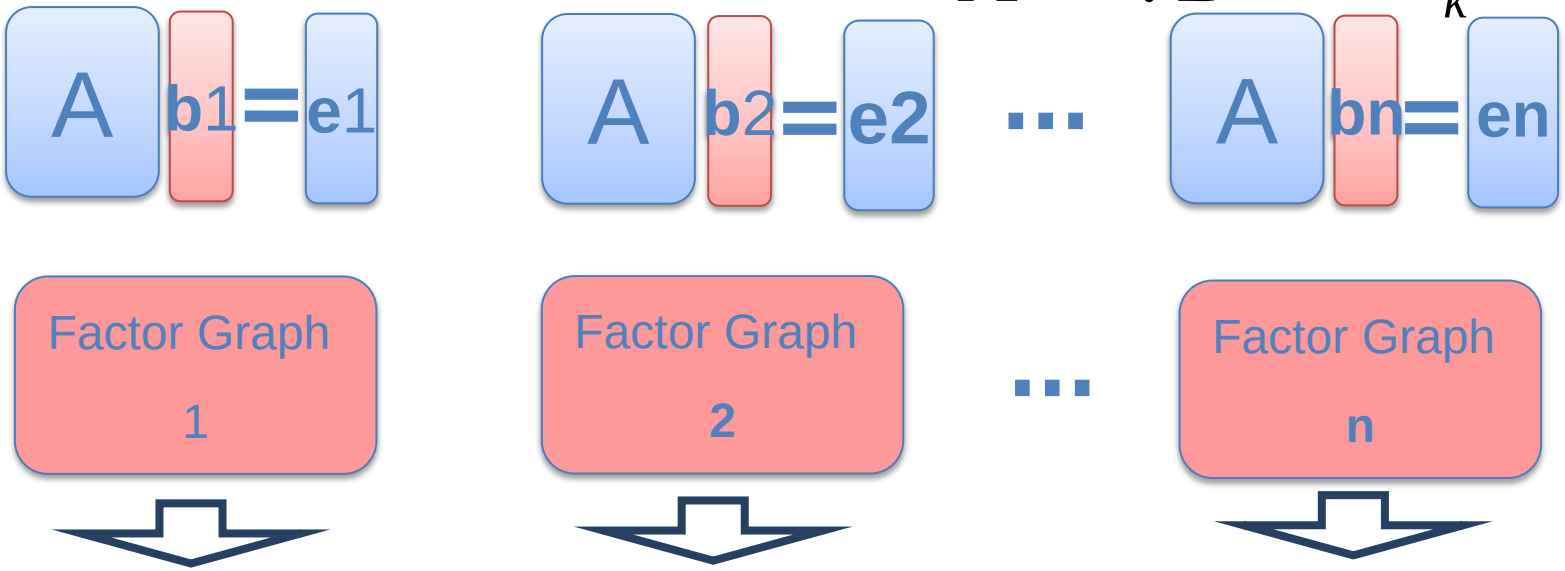
- This can be viewed as pairwise MRF over Gaussians and leads to Gaussian BP.

IJCAI-11 TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE JULY 16-22, 2011 CCIB BARCELONA, CATALONIA (SPAIN) WWW.IJCAI-11.ORG

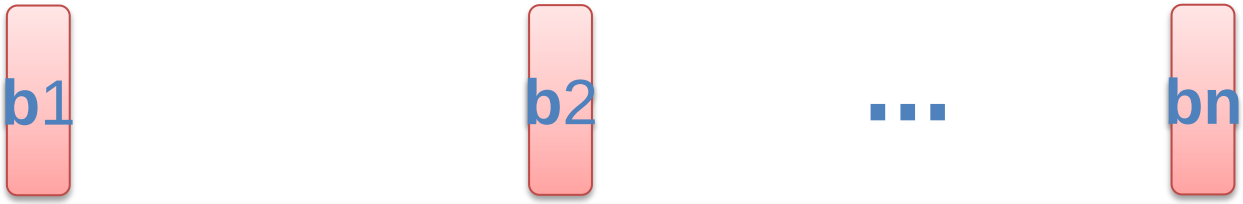
[Shental, Bickson, Siegel, Wolf, Dolev. ISIT08]

Matrix Inversion using GaBP

$$A^{-1} =: B \qquad A \cdot b_k = e_k$$



Gaussian BP



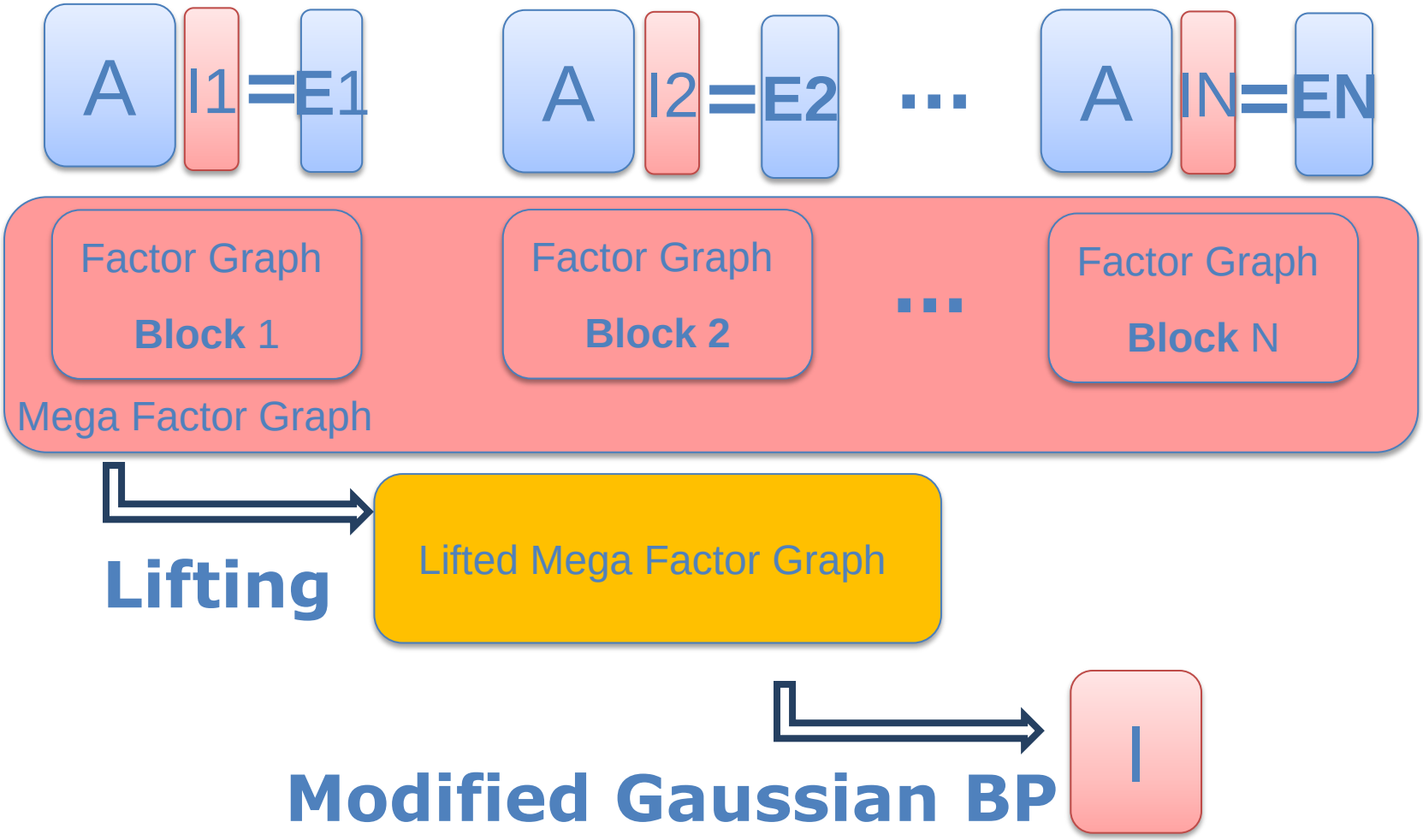
So, let's run Lifted GaBP

Color Passing+modified GaBP = Lifted GaBP

TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

[Ahmadi, Kersting, Sanner IJCAI11]

But we can do better: Multi-Evidence Lifting

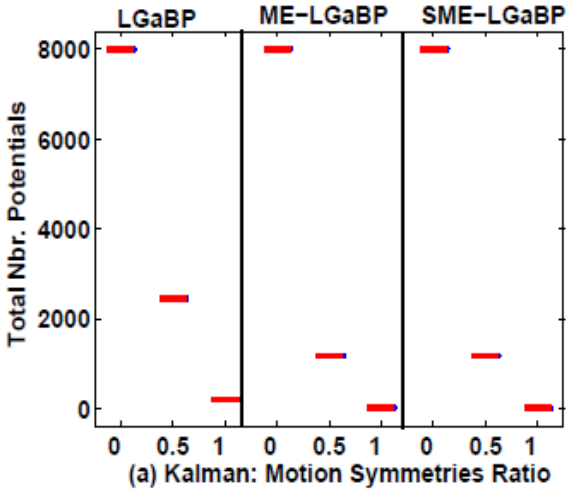


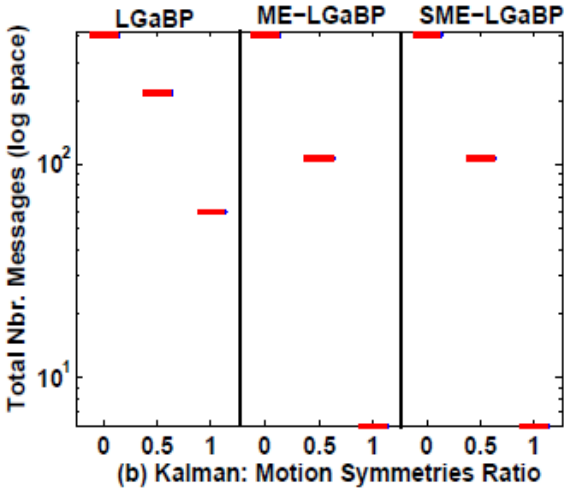
Exploits symmetries across similar inference tasks!

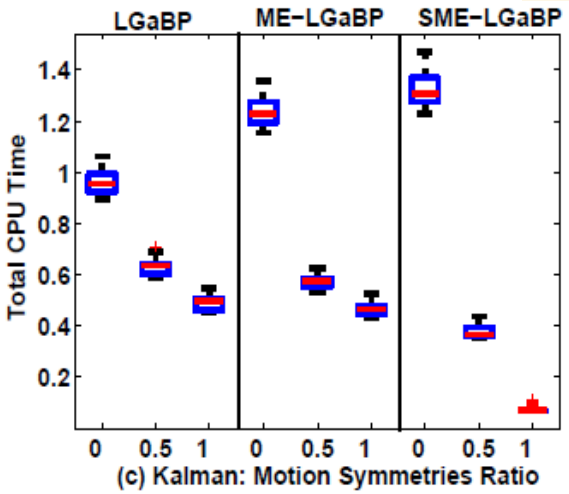
TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

[Ahmadi, Kersting, Sanner IJCAI11]

Lifted Kalman Filter





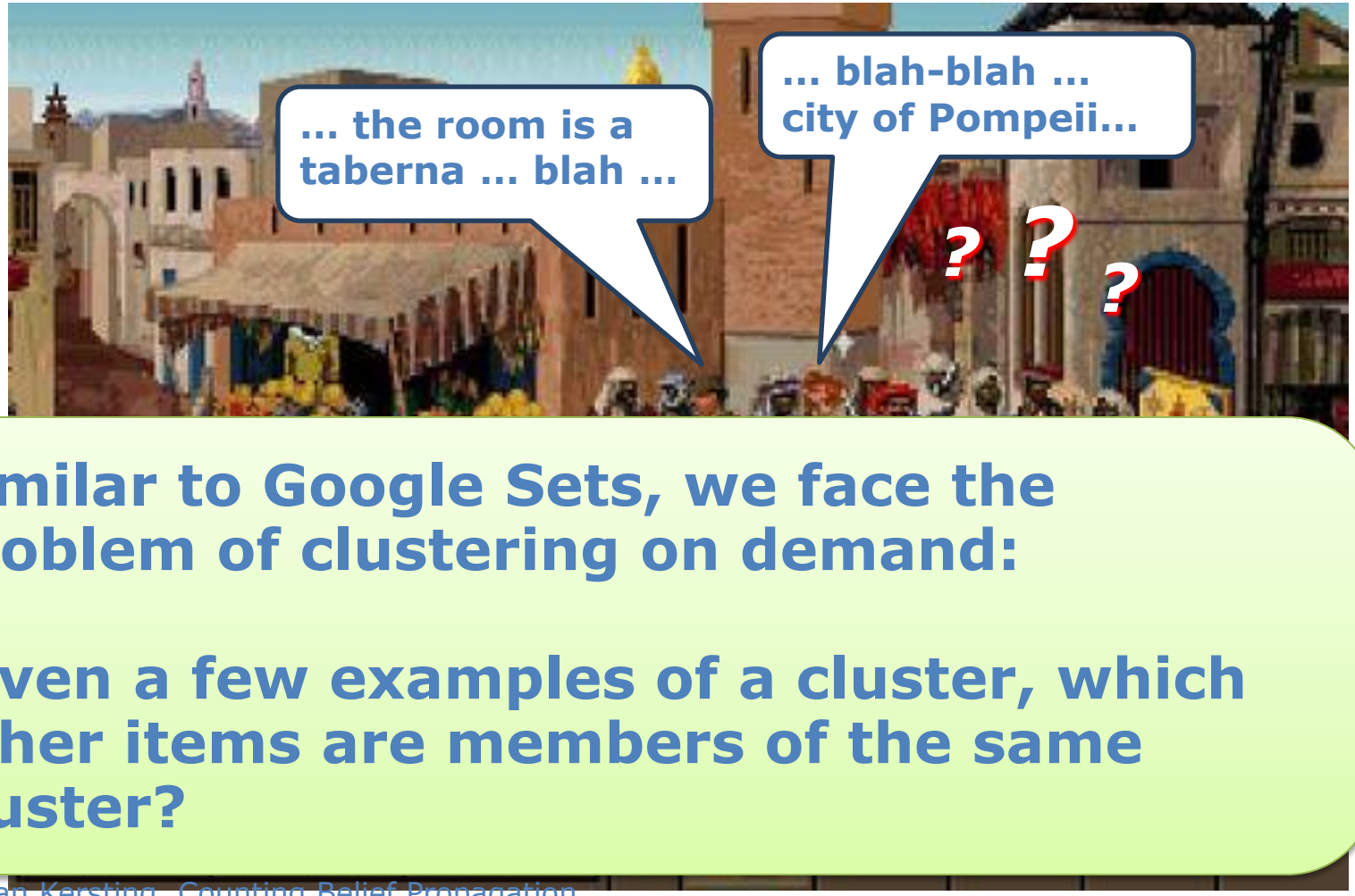


Similar we can lift random walk approaches such as PageRank and Label Propagation

TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG
IJCAI-11

[Neumann, Ahmadi, Kersting, AAAI11]

Markov Logic Sets: Towards Lifted Information Retrieval



Similar to Google Sets, we face the problem of clustering on demand:

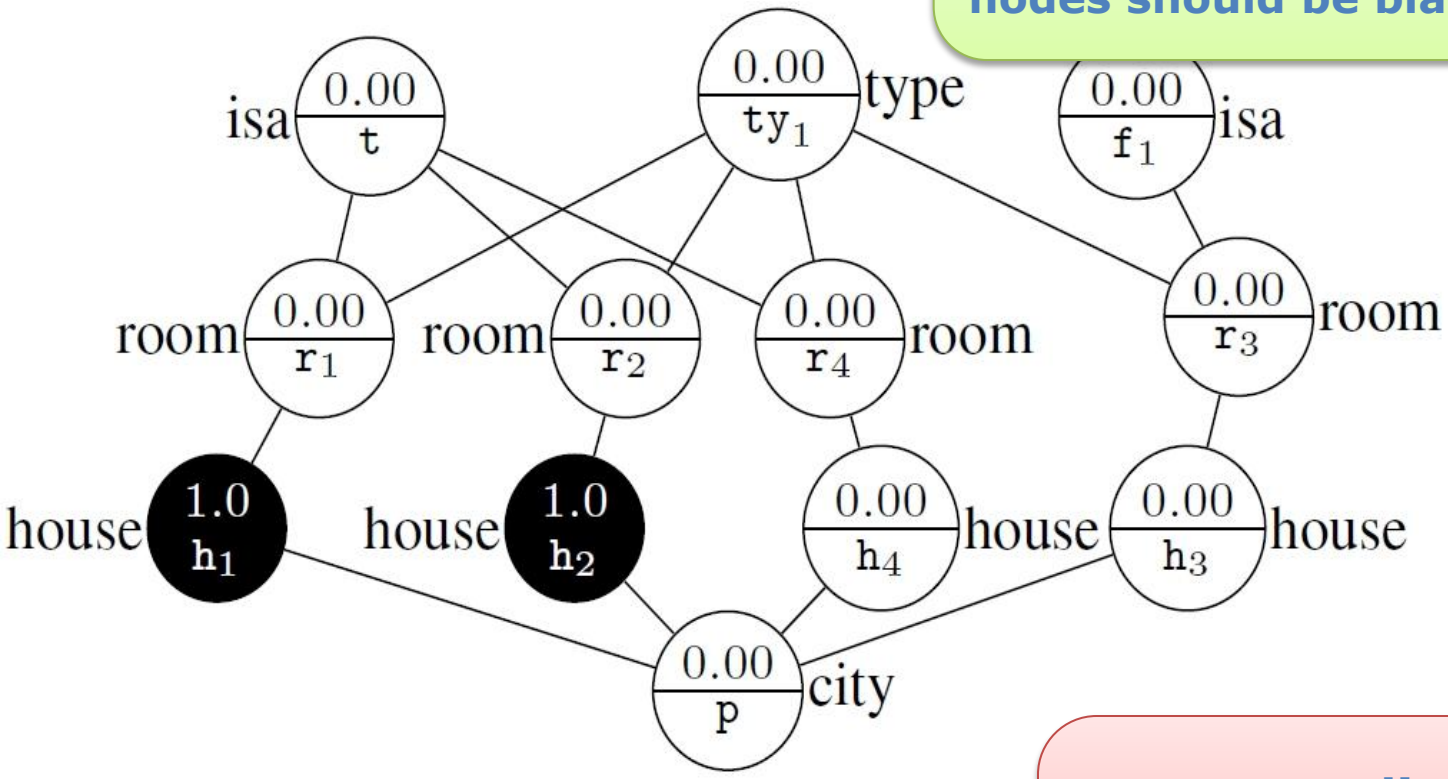
Given a few examples of a cluster, which other items are members of the same cluster?

IJCAI-11 TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE JULY 16-22, 2011 CCIB BARCELONA, CATALONIA (SPAIN) WWW.IJCAI-11.ORG

Relational Clustering on Demand

$$\mathcal{Q} = \{\text{house}(h_1), \text{house}(h_2)\}$$

Which set completion do you expect? Which other nodes should be black



Database represented as hypergraph
Nodes = atoms
Black nodes = given evidence

- same predicate
- connected items
- same structure

TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

Relational Pathfinding for Generalization

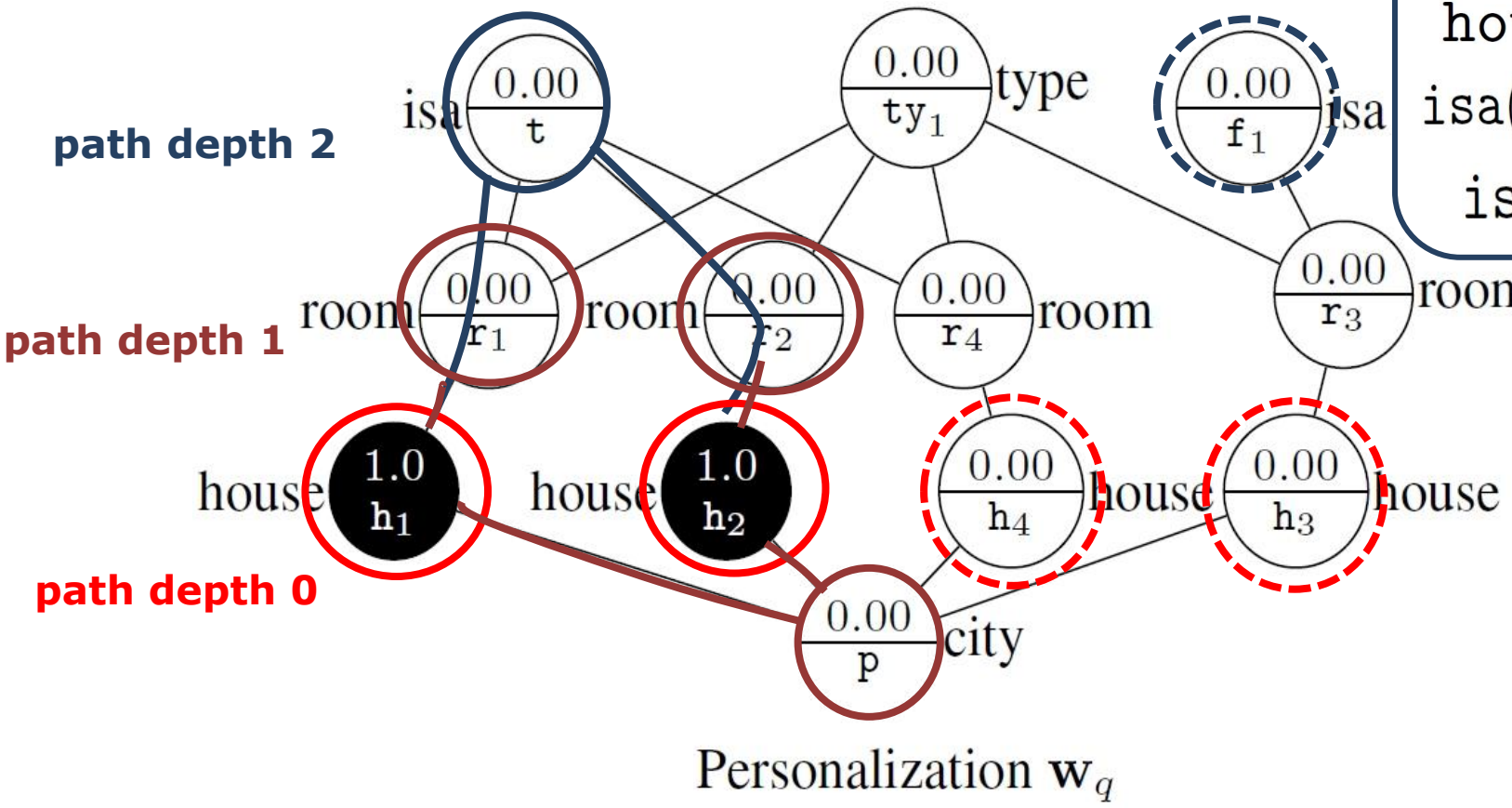
$$\mathcal{Q} = \{\text{house}(h_1), \text{house}(h_2)\}$$

Logical Features:

house(X)

isa(taberna)

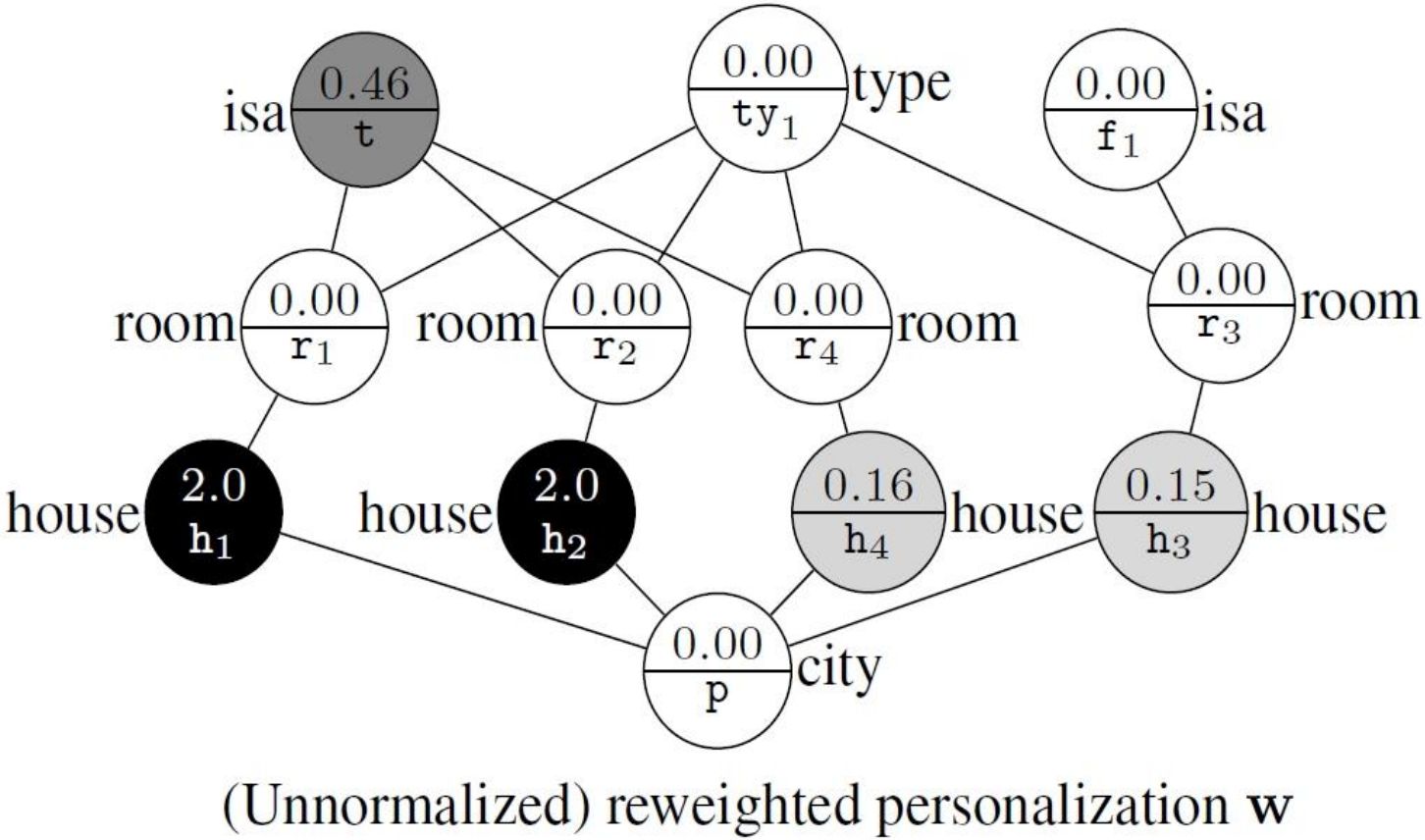
isa(X)



PageRank values do not generalize to similar items!

Personalization with Original PageRank

$$\mathcal{Q} = \{\text{house}(h_1), \text{house}(h_2)\}$$



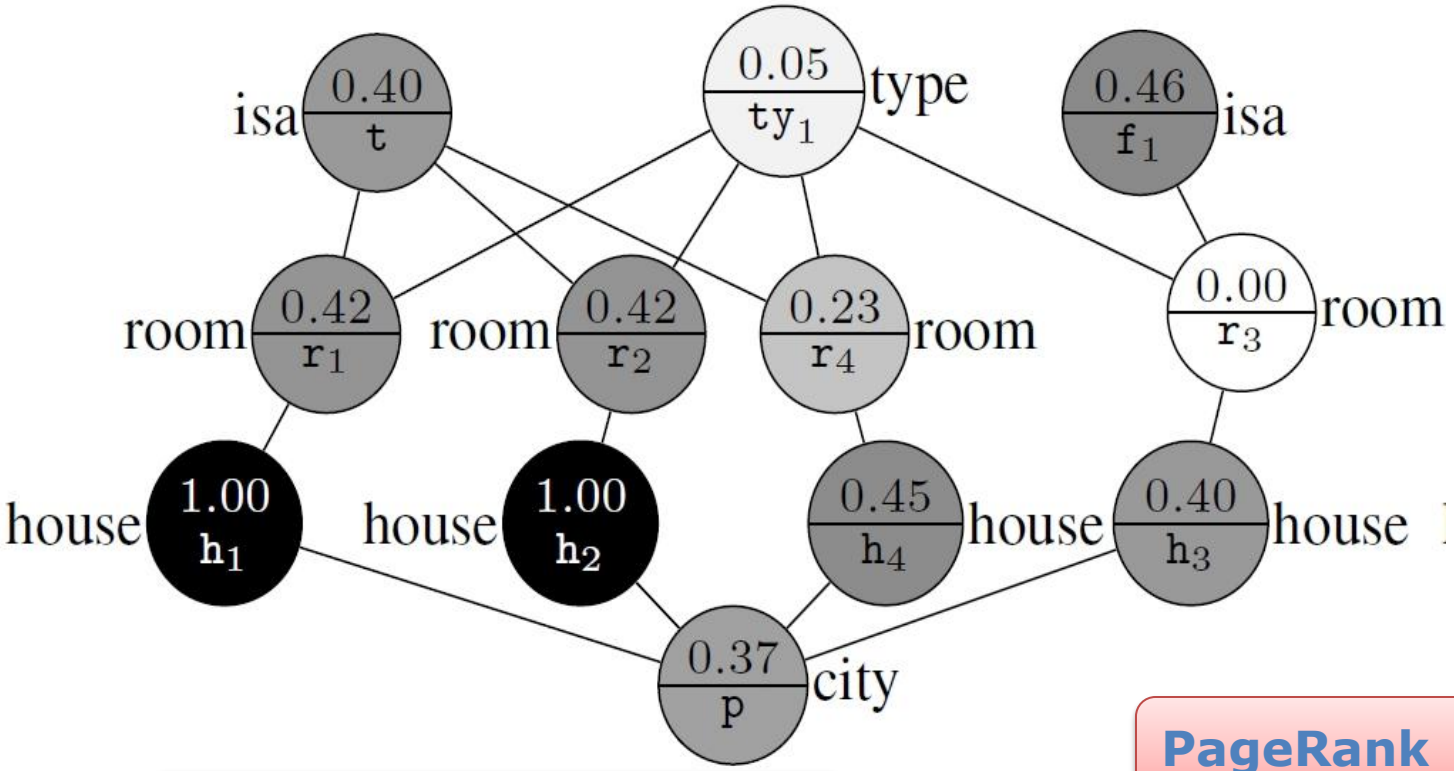
(Unnormalized) reweighted personalization w

Add PageRank values for each feature that fires

IJCAI-11
TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

We are only interested in things that behave differently

$$\mathcal{Q} = \{\text{house}(\text{h}_1), \text{house}(\text{h}_2)\}$$



Differential PageRank

$$\mathbf{x}_d = \mathbf{x}_p - \mathbf{x}_u$$

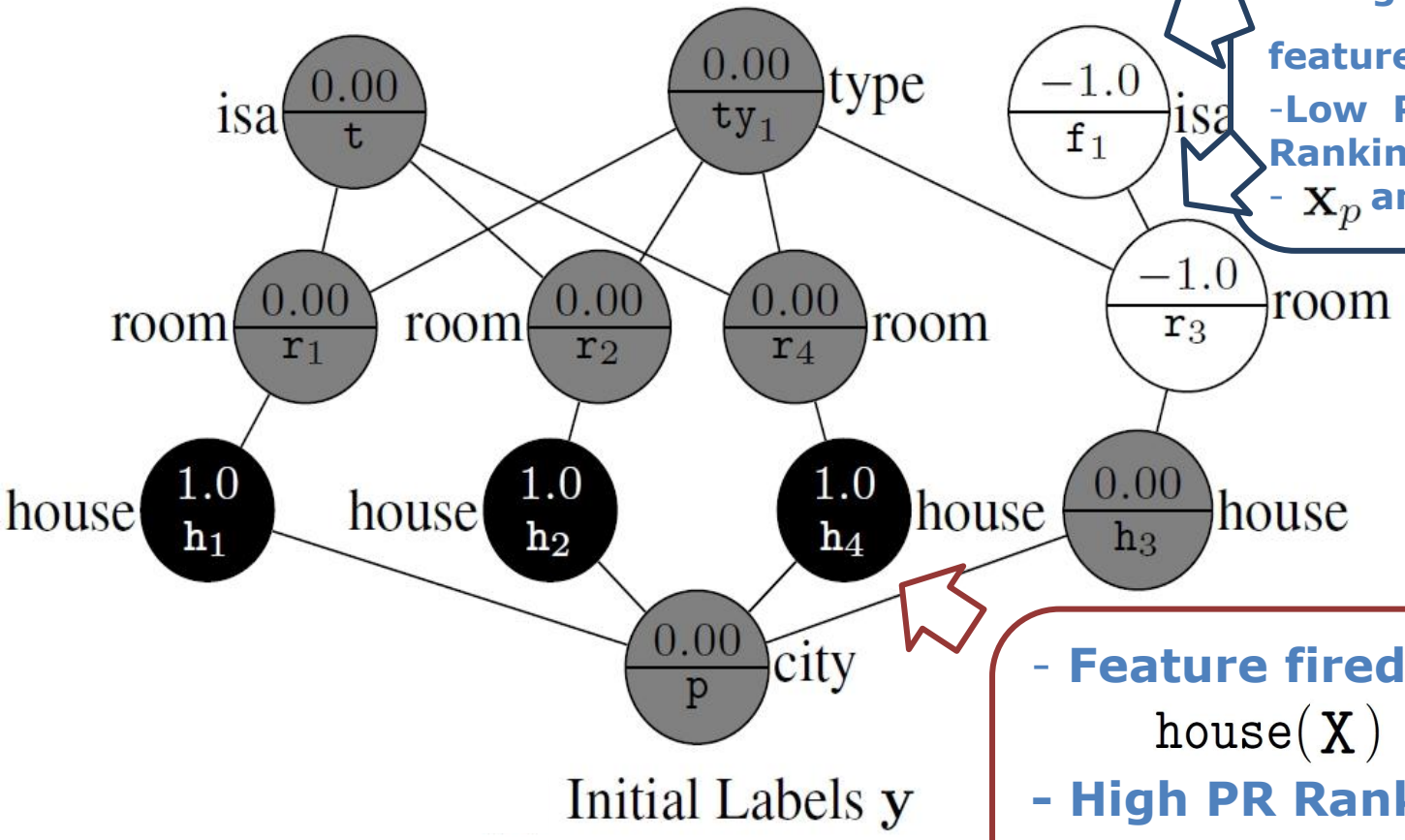
PageRank

Personalized PageRank

TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

Finally, we run Label Propagation with inital labels derived from PageRank

$$\mathcal{Q} = \{\text{house}(h_1), \text{house}(h_2)\}$$



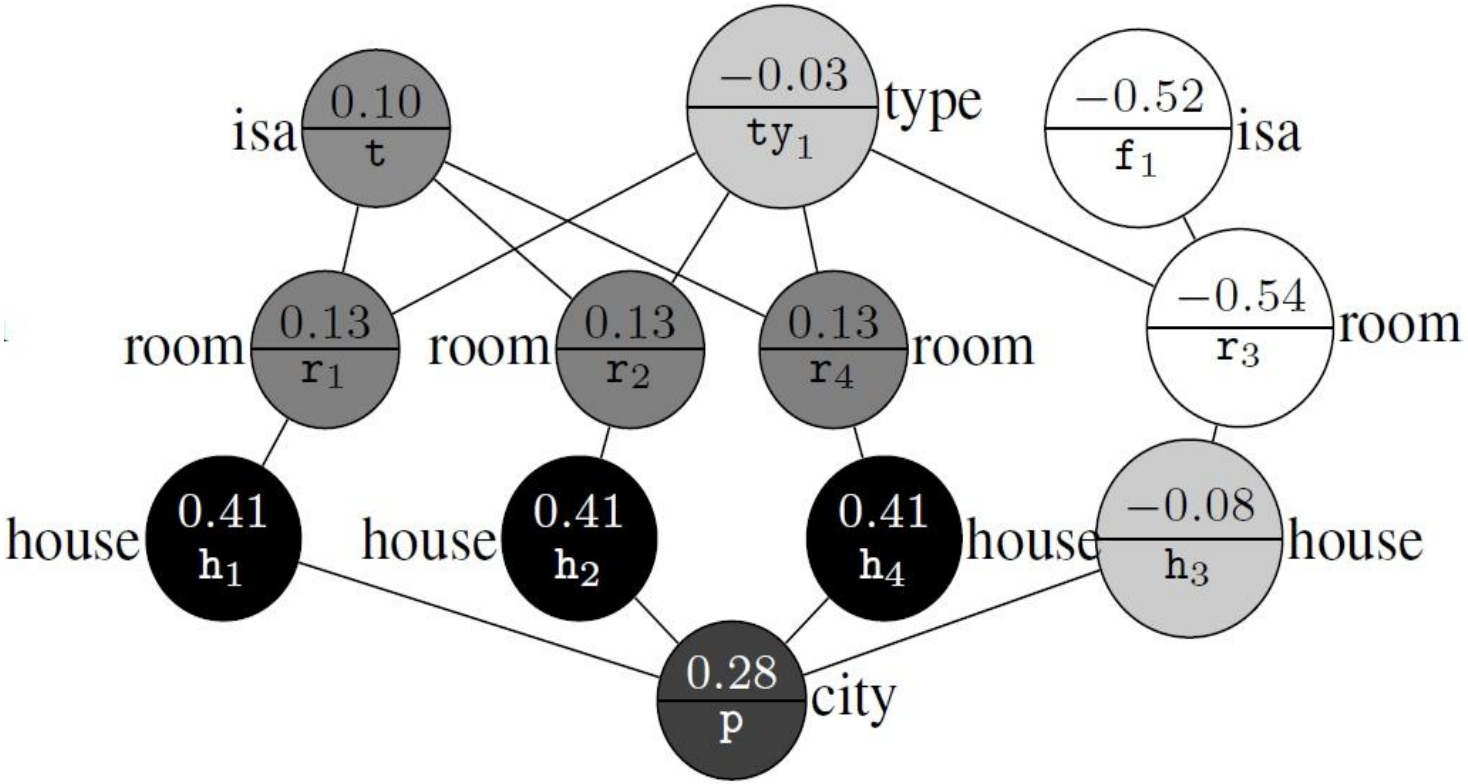
- NO ground feature fired
- Low PR Ranking
- $\mathbf{x}_p$ and/or $\mathbf{x}_d$

- Feature fired: $\text{house}(X)$
- High PR Ranking

$\mathbf{x}_d$ and/or $\mathbf{x}_p$

Final Ranking

$$\mathcal{Q} = \{\mathrm{house}(\mathrm{h}_1), \mathrm{house}(\mathrm{h}_2)\}$$



$$\mathcal{C} = \{\mathrm{h}_1, \mathrm{h}_2, \mathrm{h}_4, \mathrm{p}, \mathrm{r}_1, \mathrm{r}_2, \mathrm{r}_4, \mathrm{t}\}$$

IJCAI-11 TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE JULY 16-22, 2011 CCIB BARCELONA, CATALONIA (SPAIN) WWW.IJCAI-11.ORG

Lifted Random Walks with Restart

All main computations involve matrix inversion. So, let's apply lifted GaBP using multi-evidence lifting!

Personalized PageRank

$$(\mathbf{I} - \alpha \mathbf{M}^T)\mathbf{x} = \mathbf{v}$$

Label Propagation

$$(\mathbf{I} - \alpha \mathbf{L})\mathbf{x} = (1 - \alpha)\mathbf{y}$$

where

$$\mathbf{L} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$$

$$\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$$

IJCAI-11
TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

Conclusions

- Lifted/efficient reasoning crucial to StarAI
  - Exploit symmetries revealed by (relational) model
- CBP bisimulates BP
  - Group together nodes that send and receive the same messages
- Can be applied to any factor graph / CNF
  - SAT, Systems of Linear Equations, Matrix Inversion
  - Kalman Filter, PageRank, Label Propagation, …

TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE
JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG
IJCAI-11

Lifted Inference with Approximate Bisimulation

Prithviraj Sen and Lise Getoor

Tutorial on Lifted Inference

International Joint Conference on Artificial Intelligence, 2011

Introduction

- Exact lifted inference may not help when
  - The graphical model has very little symmetry.
  - Exact inference is infeasible.
- In such cases: Approximate lifted inference.

Goals

- "Lift" more than what's possible with exact bisimulation
- Incur a small loss in inference accuracy.
- Combine with a bounded memory inference procedure.

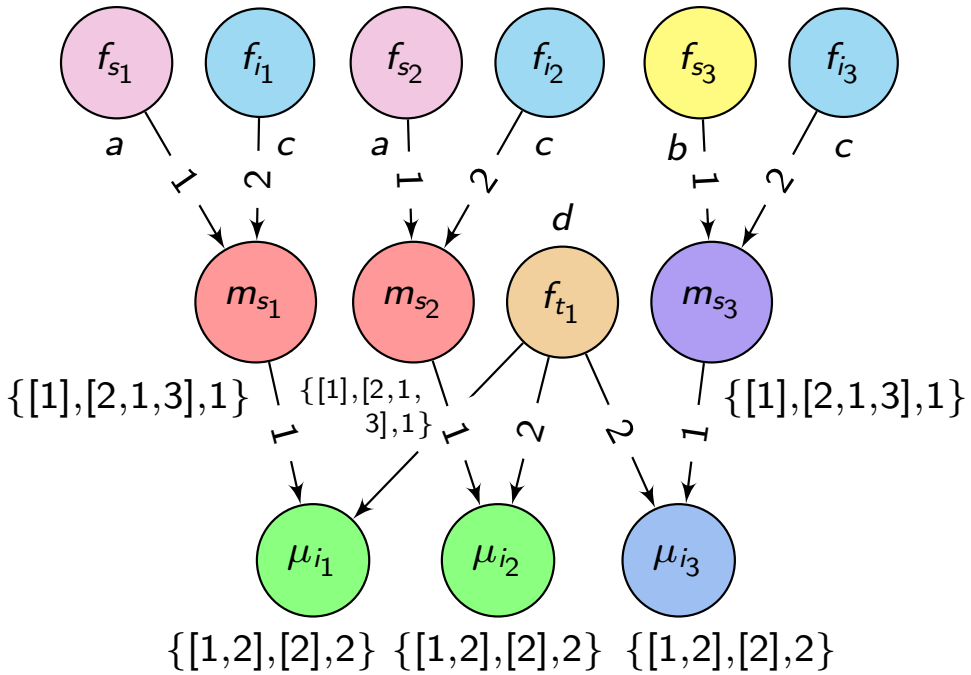
Outline

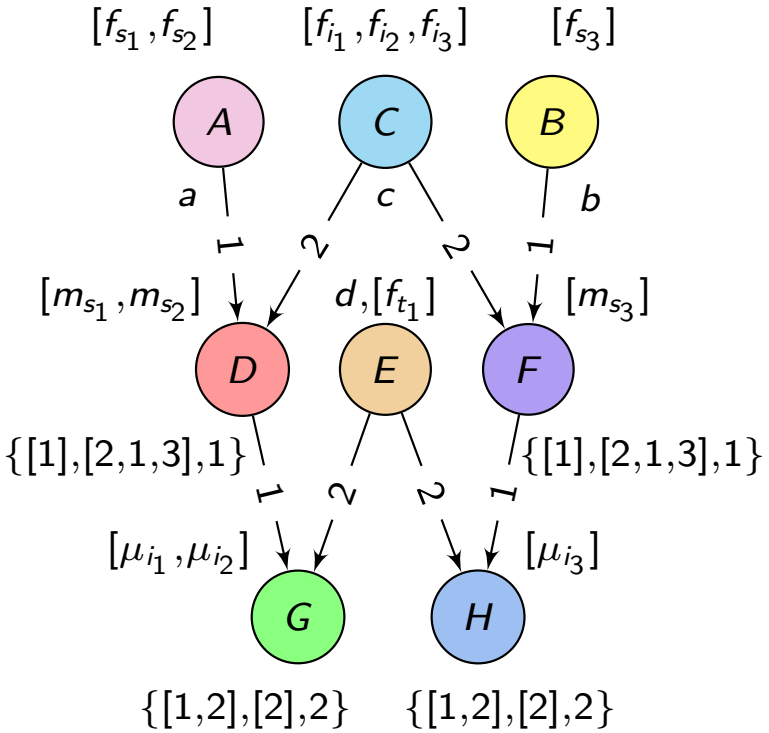
- Recap bisimulation-based exact lifted inference
- Extend definition of shared factors
- Approximate Lifted Inference
  - with Approximate Bisimulation
  - with Factor Binning
- Bounded Complexity Inference
- Experimental Results
- Future Work

Exploiting Symmetry in RV-Elim Graphs

- Compute

$$\mu(i_1) = \sum_{t_1} f_{t_1}(t_1) \sum_{s_1} f_{s_1}(s_1) f_{i_1}(i_1 | s_1, t_1)$$





Shared Factors: Re-interpreted

Earlier definition

Two vertices v_1 and v_2 represent shared factors, i.e., $f_{v_1} \cong f_{v_2}$, iff:

- parents are shared.
- labels are identical.

- Label path: formed by concatenating labels from the RV-Elim graph.

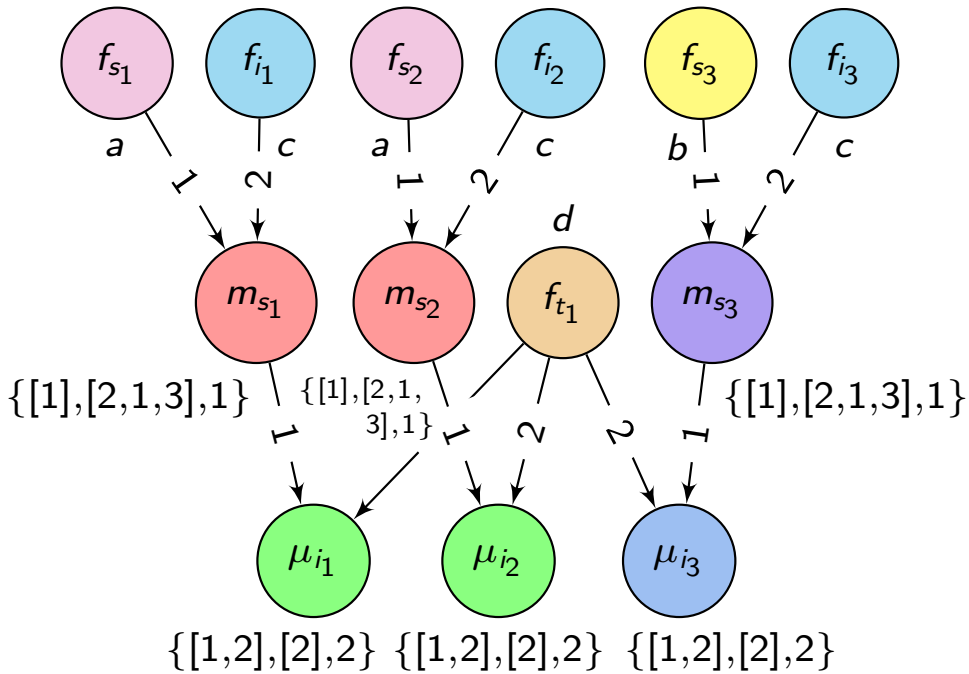
Equivalent Definition

Two vertices v_1 and v_2 represent shared factors, i.e., $f_{v_1} \cong f_{v_2}$, iff:

- They share the same set of incoming label paths.

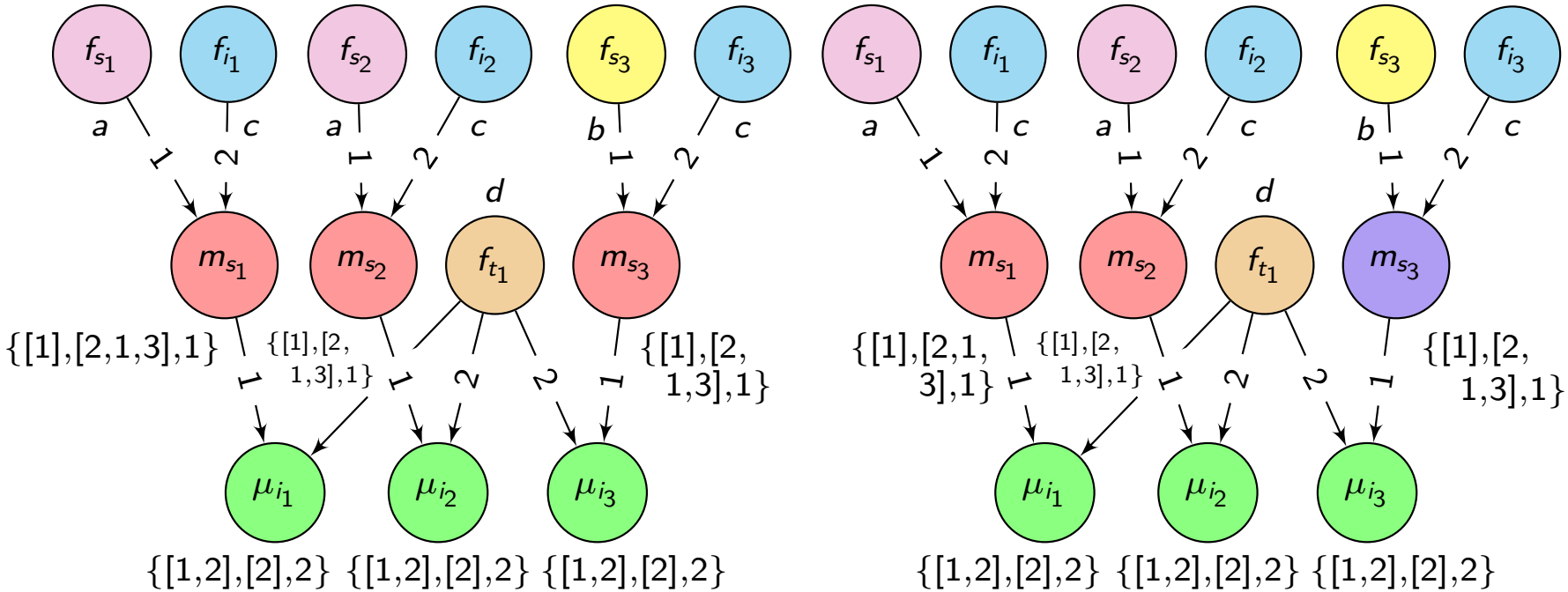
Path-based Compression of RV-Elim Graphs

- $m_{s_1} \cong m_{s_2}$
  - $\rightsquigarrow m_{s_1}$
    - $a \rightarrow 1 \rightarrow \{[1],[2,1,3],1\}$
    - $c \rightarrow 2 \rightarrow \{[1],[2,1,3],1\}$
  - $\rightsquigarrow m_{s_2}$
    - $a \rightarrow 1 \rightarrow \{[1],[2,1,3],1\}$
    - $c \rightarrow 2 \rightarrow \{[1],[2,1,3],1\}$
- $\mu_{i_2} \ncong \mu_{i_3}$
- $a \rightarrow 1 \rightarrow \{[1],[2,1,3],1\} \rightarrow 1 \rightarrow \{[1,2],[2],2\} \rightsquigarrow \mu_{i_2}$
- $b \rightarrow 1 \rightarrow \{[1],[2,1,3],1\} \rightarrow 1 \rightarrow \{[1,2],[2],2\} \rightsquigarrow \mu_{i_3}$



Lifted Inference with Approximate Bisimulation

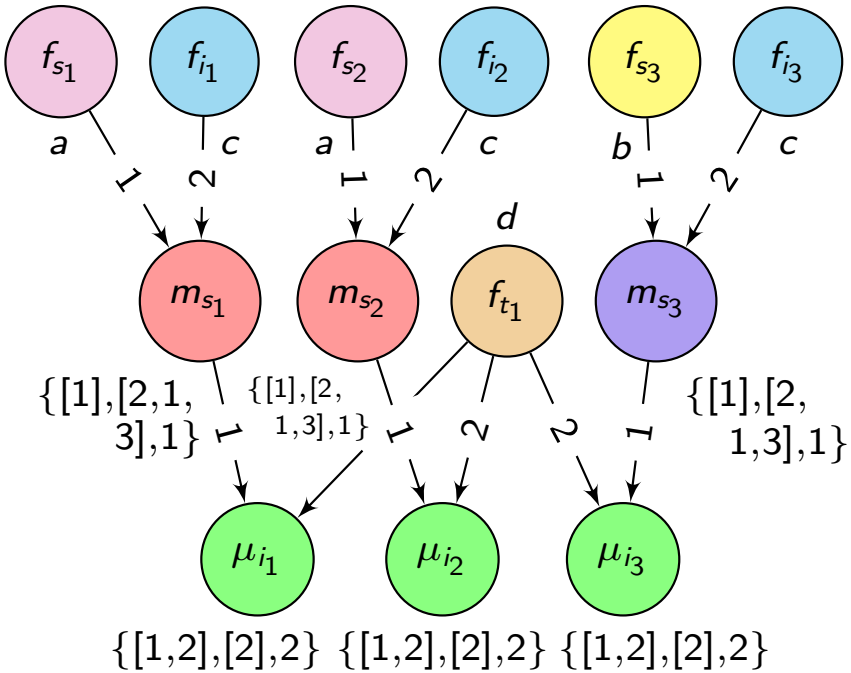
- Approximate bisimulation: $f_1 \cong^k f_2$ if all k-length paths are identical.
- k is a parameter (*path-length*).

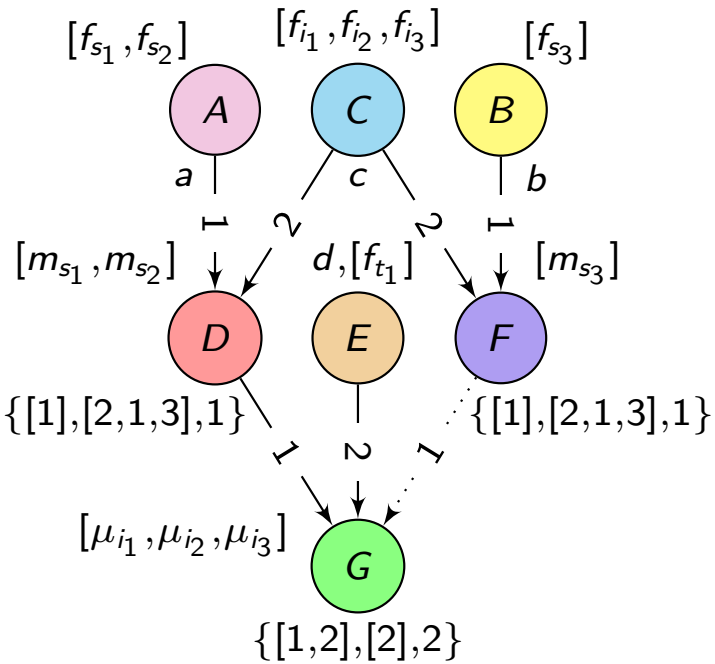


$k = 0$ (partition of size 6) $k = 1$ (partition of size 7)

- Path-length=2 results in exact bisimulation, partition of size 8.

Compressed RV-Elim Graph





Lifted Inference with Factor Binning

- Exact bisimulation: $f_1 \cong f_2$ if *distance* between them is 0.
- Factor binning: $f_1 \cong^{\epsilon} f_2$ if the *distance* between them $\leq \epsilon$.
- ϵ is a parameter, *distance* is user-defined.
- Note that, determining if $f_1 \cong^{\epsilon} f_2$ requires evaluating them.
- Better is:
  - Implement exact bisimulation-based lifted inference.
  - After computing a factor from each partition, perform factor binning.

Implementing Factor Binning

- Factor binning problem:

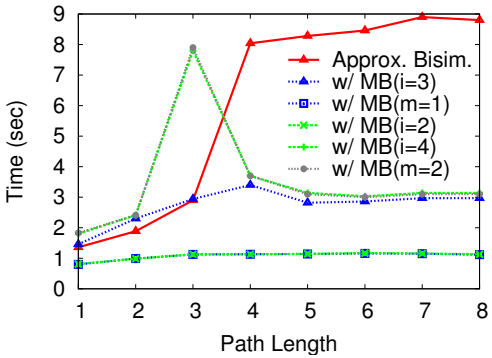
$$\begin{aligned} \text{Given:} \quad & \text{set of factors } \mathcal{F} = \{f_1, \ldots f_n\} \\ & \text{threshold } \epsilon, \text{ distance function dist}(\cdot,\cdot) \\ \text{Return:} \quad & \text{argmin}_{\mathbf{F} \subseteq \mathcal{F}} |\mathbf{F}| \\ \text{such that} \quad & \forall f_i \in \mathcal{F} \setminus \mathbf{F} \; \exists f \in \mathbf{F} \text{ s.t. } \text{dist}(f_i, f) \leq \epsilon \end{aligned}$$

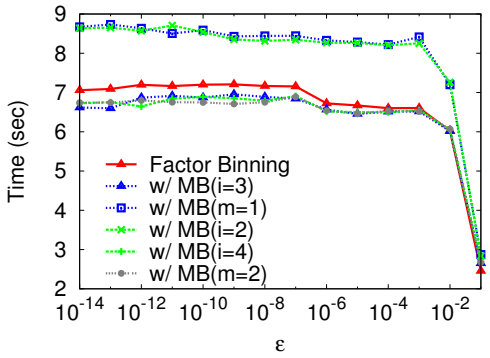
- Is equivalent to dominating set problem.
- Dominating set is NP-Complete.
- Greedy heuristic is $\log(n)$-approximate, the best we have.
- Possible to be smarter if distance function satisfies special properties.
- E.g., algorithms available for Euclidean distance metrics.

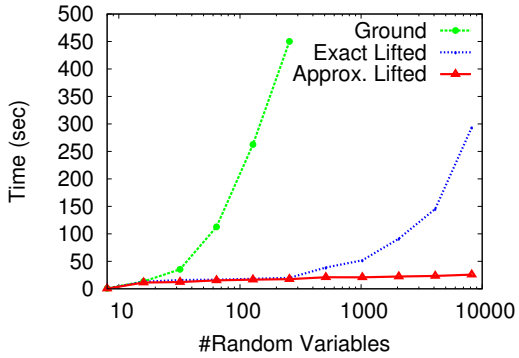
Bounded Complexity Inference

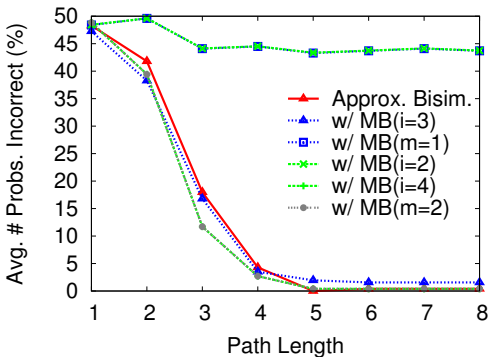
- All our techniques incur full inference complexity.
- If treewidth of graphical model is too large (≥ 30), we can't run lifted inference.
- Solution: Combine with bounded complexity inference techniques.
- One choice: Mini-buckets (Dechter and Rish, '03).
- Instead of multiplying all factors containing x as argument, mini-buckets:
  - pools them into disjoint buckets.
  - eliminates x from each bucket separately.
- Each non-source node represents an elimination operation.
- To incorporate mini-buckets, instead we form nodes corresponding to factors produced by mini-buckets.

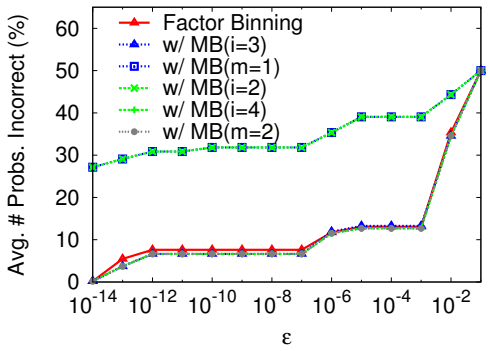
Performance of Approximate Lifted Inference

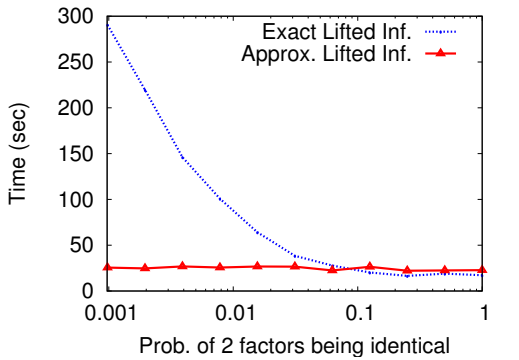










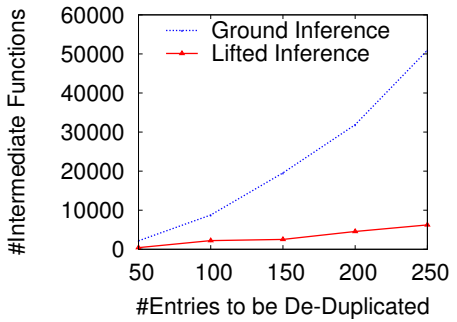


- Either approach gives an order of magnitude speedup over ground inference.
- Combined, gives two orders of magnitude speedup.

Experimental Results: Real-World Data

Dataset	Inf. Alg.	Time (s)	Arith. Ops.	Rem. Ops.	Acc.
Cora	Ground Inf.	163.5	163	0.5	77.8
	Lifted Inf.	60.6	59.9	0.7	73
CiteSeer	Ground Inf.	101.0	100.8	0.2	68.7
	Lifted Inf.	65.0	63.9	1.1	66.8

- Cora and CiteSeer, bibliographic datasets with citation networks.
- Inference problem involves evidence and highly irregular networks.
- Also, tried with Cora for entity resolution.
- Dataset with unclean references to publications, authors and venues.
- Lifted inference reduces memory consumption.



Conclusion

- We described bisimulation-based lifted inference.
- Treats inference as an automaton, minimizes it to perform "lifting".
- Exact version subsumes inversion elimination.
- Very simple approach.
- Allows various ways of approximations.
- Can be combined with other inference techniques.
- Counting elimination is out of scope.
- Future work includes intelligent setting of parameters.

Thank you.

References

[KS83] Paris Kanellakis and Scott Smolka.
CCS expressions, finite state processes, and three problems of equivalence.
In *PODC*, 1983.

[ZP94] Nevin Zhang and David Poole.
A simple approach to Bayesian network computations.
In *Canadian Conference on Artificial Intelligence*, 1994.

[DPP01] Agostino Dovier, Carla Piazza and Alberto Policriti.
A Fast Bisimulation Algorithm.
In *International Conference on Computer Aided Verification*, 2001.

[Poole03] David Poole.
First-order probabilistic inference.
In *IJCAI*, 2003.

[SD08] Parag Singla and Pedro Domingos.
Lifted First-Order Belief Propagation.
In *AAAI*, 2008.

Anytime Lifted Belief Propagation

Rodrigo de Salvo Braz	SRI International
Hung Bui	SRI International
Sriraam Natarajan	University of Wisconsin
Jude Shavlik	University of Wisconsin
Stuart Russell	UC Berkeley

Slides online

http://www.ai.sri.com/~braz/

and go to “Presentations”

What we are doing

- Regular lifted inference (de Salvo Braz et al, MLNs, Milch) shatters models (exhaustively splits them) before inference starts
- Needs to consider entire model before giving a result
- In this work, we interleave splitting and inference, obtaining exact bounds on query as we go
- We usually will not shatter or consider entire model before yielding answer

Outline

- Background
  - Relational Probabilistic Models
  - Propositionalization
  - Belief Propagation
  - Lifted Belief Propagation and shattering
- Anytime Lifted Belief Propagation
  - Intuition
  - Box propagation
  - Example
- Final Remarks
  - Connection to Theorem Proving
  - Conclusion
  - Future directions

Background

Relational Probabilistic Model

- Compact representation for graphical models
- A parameterized factor (parfactor) stands for all its instantiations

  for all Y ϕ_1(funny(Y))

  for all X,Y, X≠Y : ϕ_2(funny(Y),likes(X,Y))

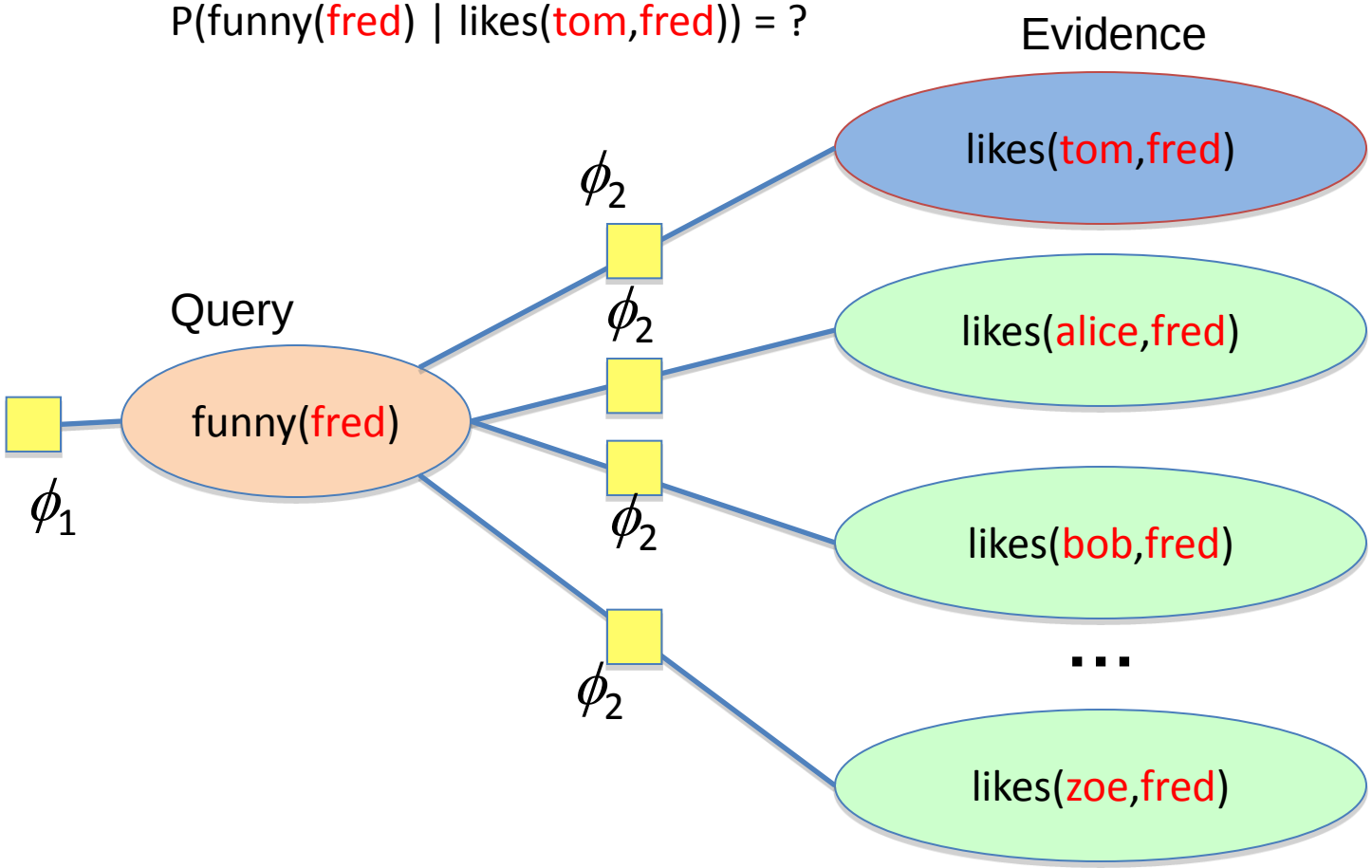
  stands for

- ϕ_1(funny(a)), ϕ_1(funny(b)), ϕ_1(funny(c)),…,
  ϕ_2(funny(a),likes(b,a)),
  ϕ_2(funny(a),likes(c,a)),…,
  ϕ_2(funny(z),likes(a,z)),…

Propositionalization

for all Y ϕ_1(funny(Y))
for all X,Y, X≠Y : ϕ_2(funny(Y),likes(X,Y))

P(funny(fred) | likes(tom,fred)) = ?

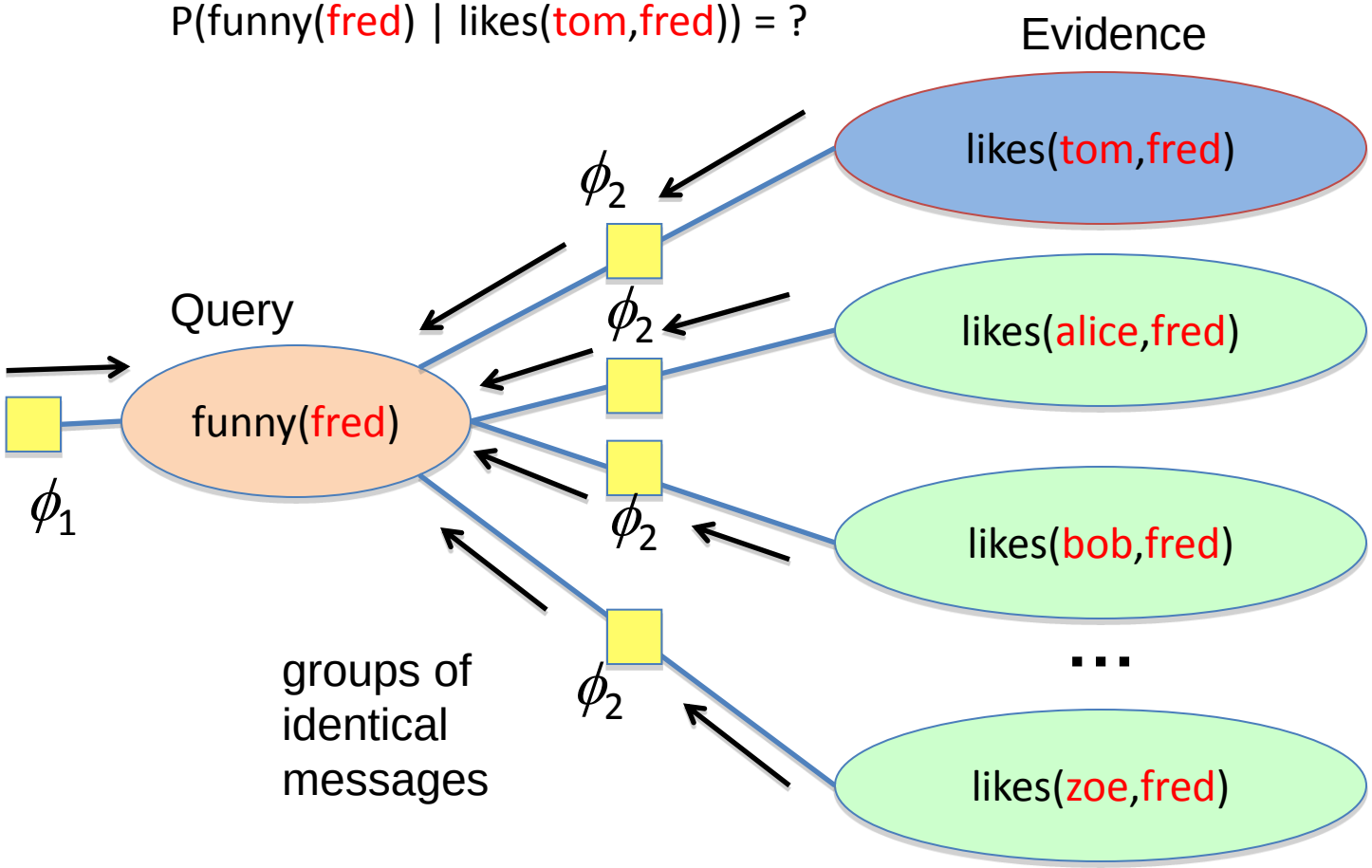


Belief Propagation

for all Y ϕ_1(funny(Y))
for all X,Y, X≠Y : ϕ_2(funny(Y),likes(X,Y))

Propagates messages all the way to query

P(funny(fred) | likes(tom,fred)) = ?



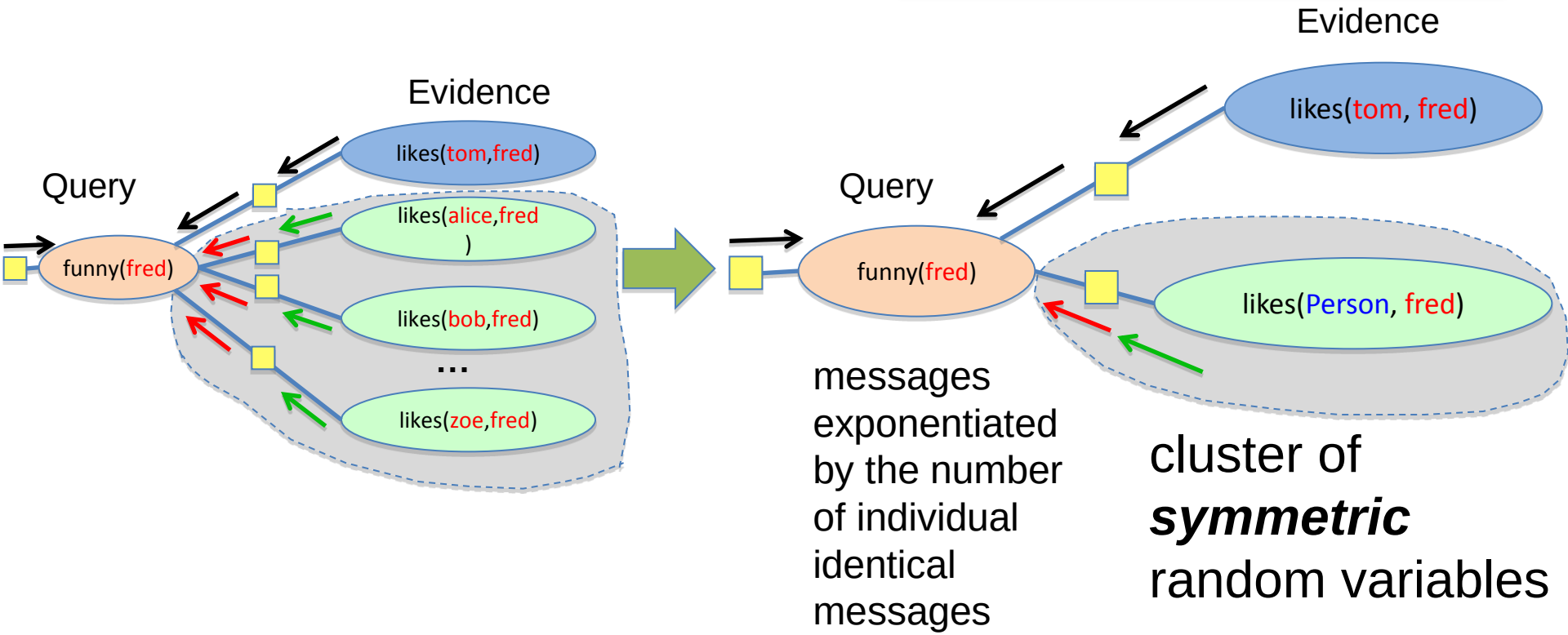
sends a different message because it has evidence on it

Lifted Belief Propagation

for all Y ϕ_1(funny(Y))
for all X,Y, X≠Y : ϕ_2(funny(Y),likes(X,Y))

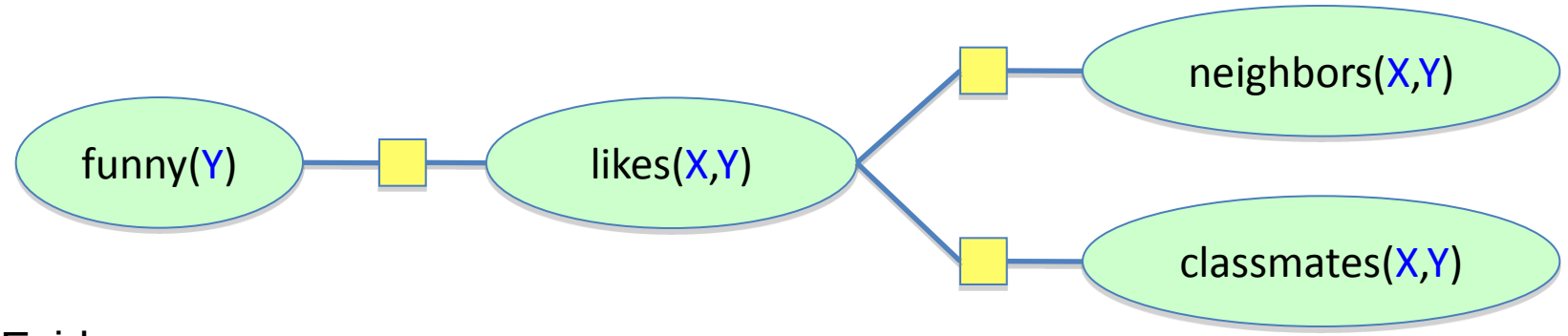
P(funny(fred) | likes(tom,fred)) = ?

Groups identical messages and computes them once



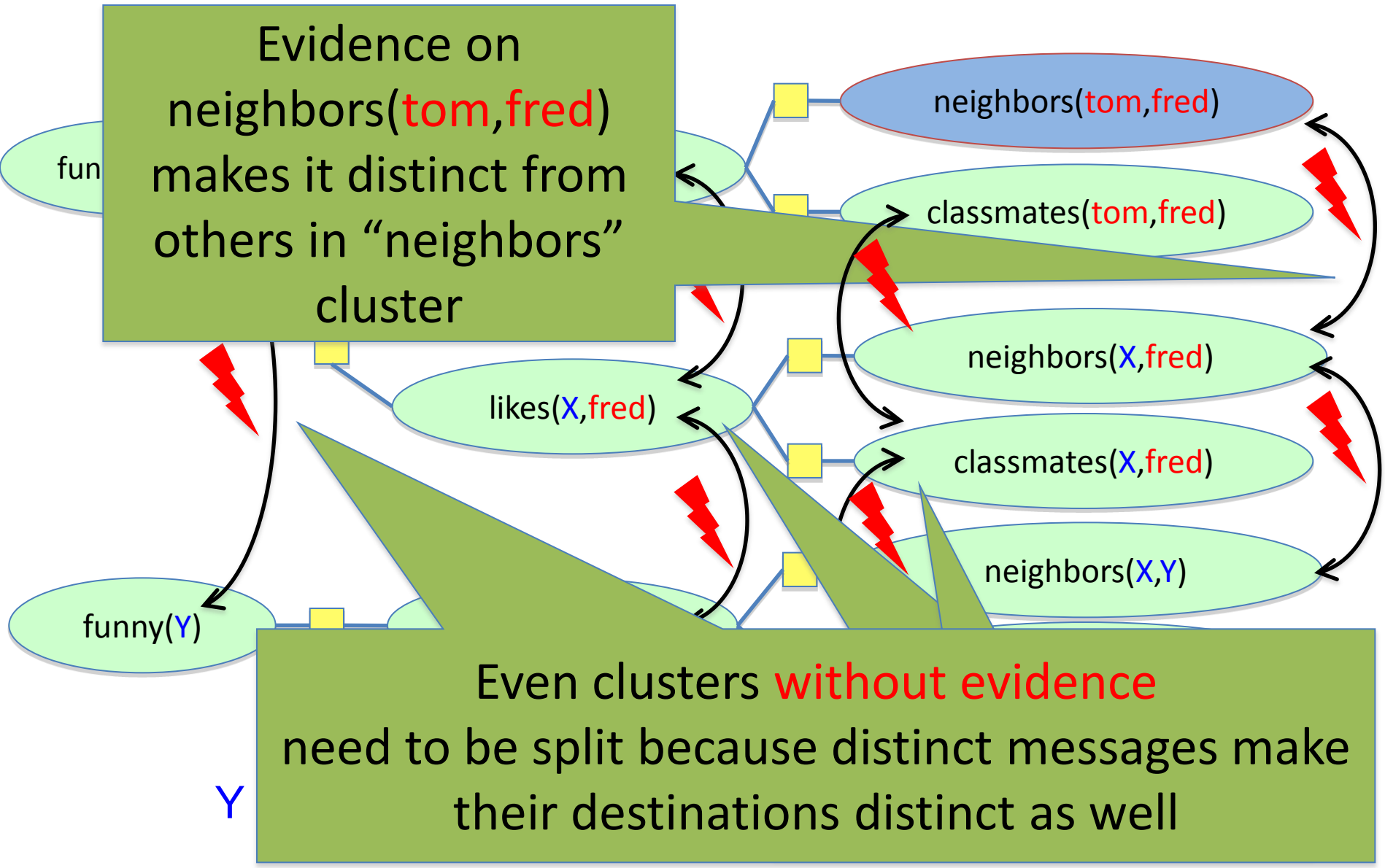
The Need for Shattering

- Lifted BP depends on clusters of variables being *symmetric*, that is,
  sending and receiving identical messages.
- In other words, it is about
  dividing random variables in cases

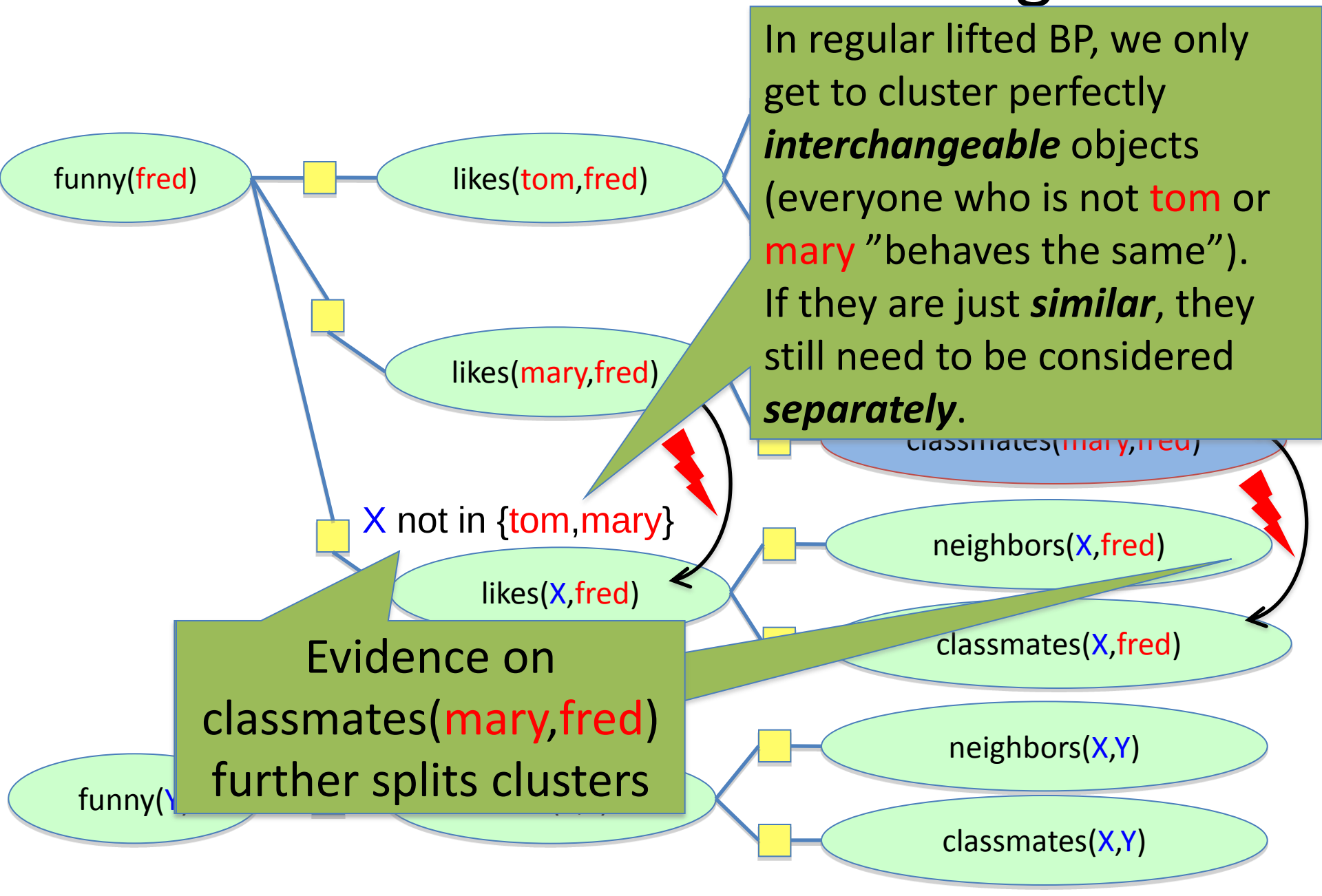


Evidence:
neighbors(tom,fred),
classmates(mary,fred)

The Need for Shattering

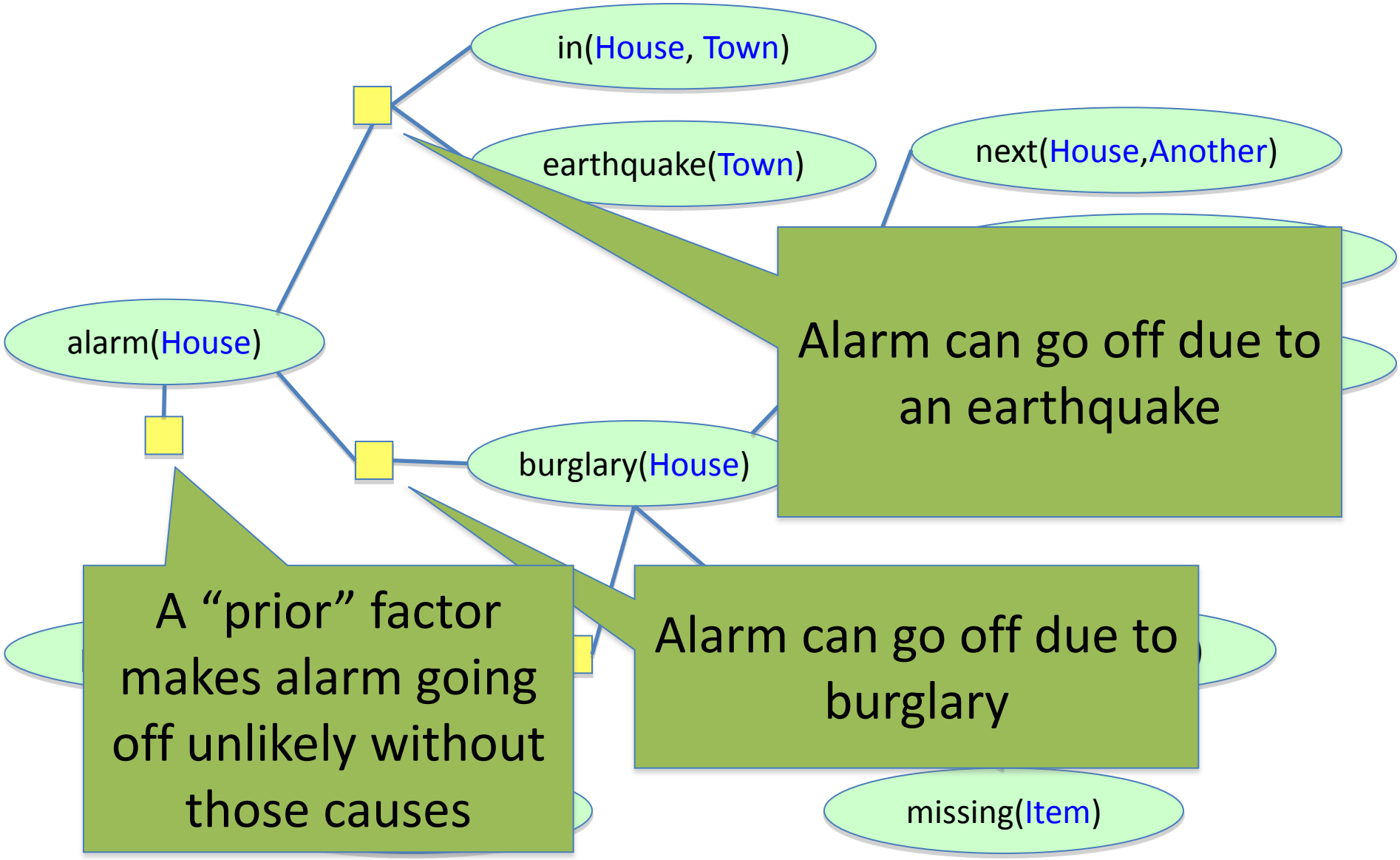


The Need for Shattering

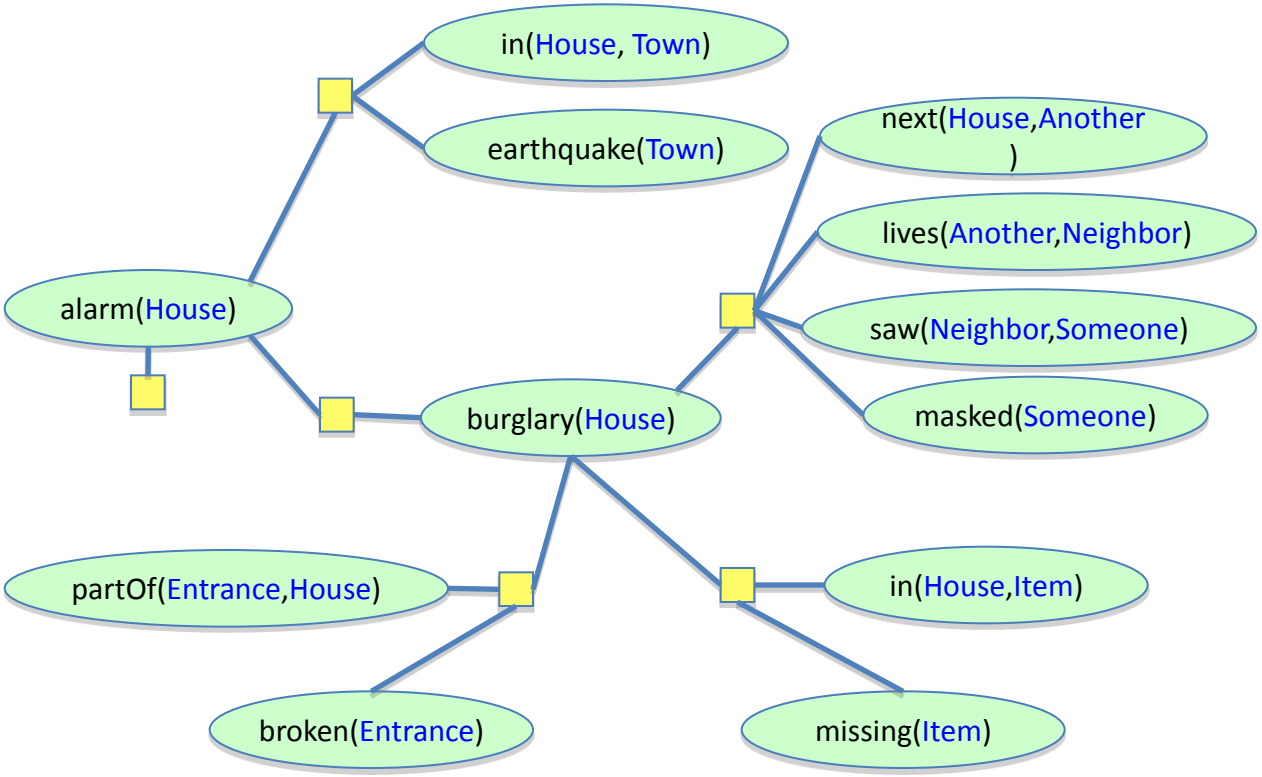


Anytime
Lifted
Belief Propagation

Intuition for Anytime Lifted BP

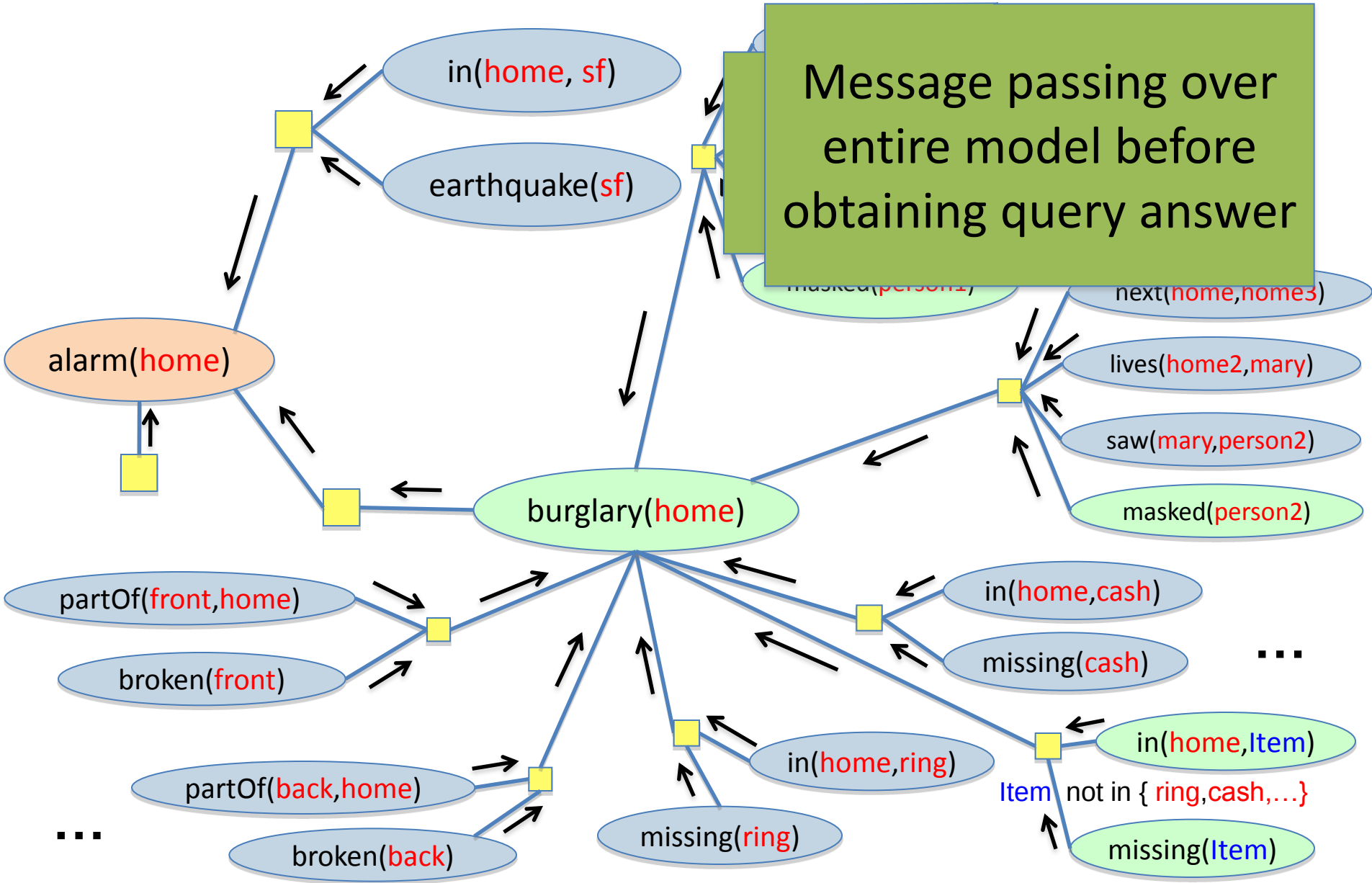


Intuition for Anytime Lifted BP

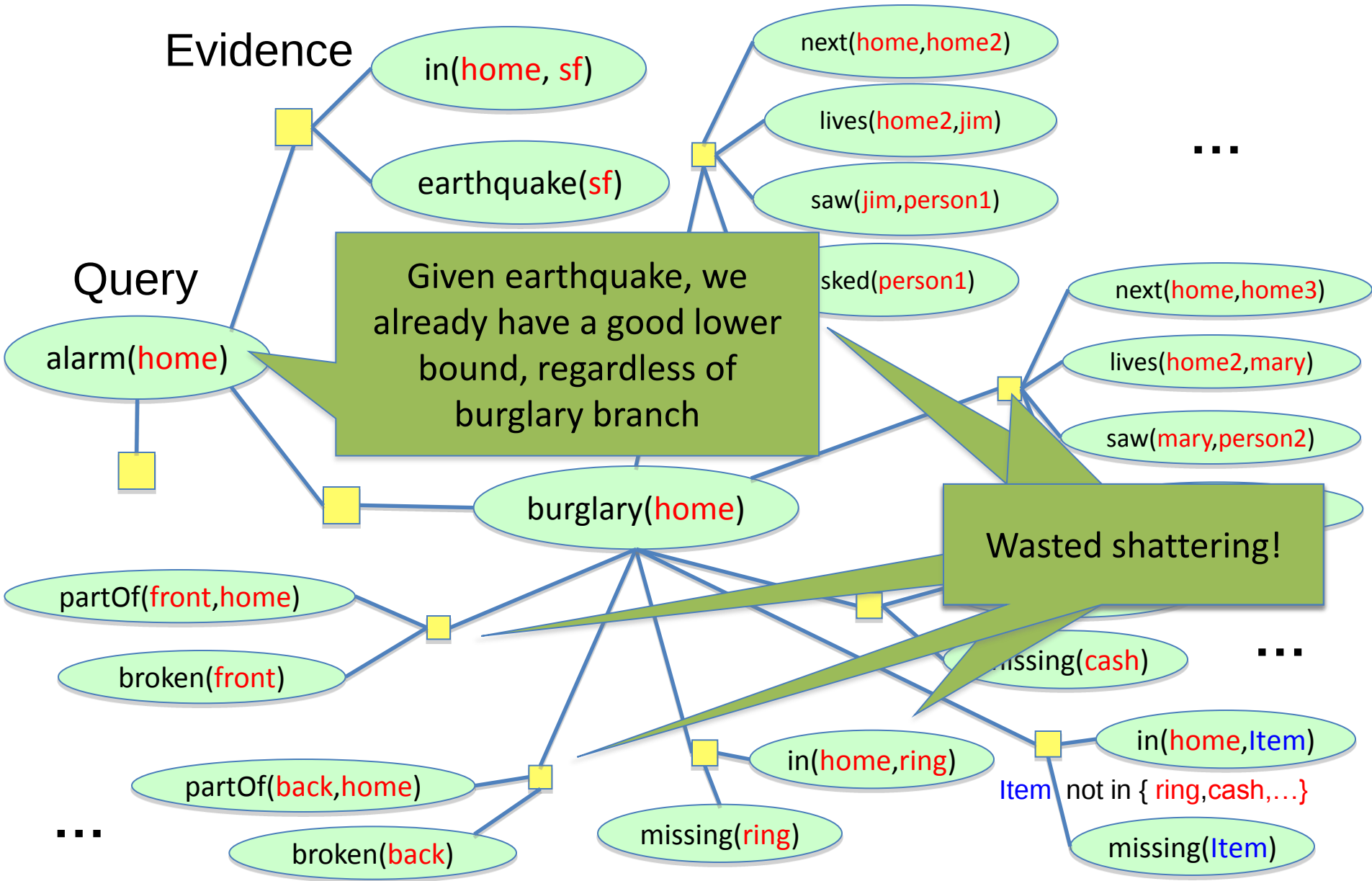


Given
a home in sf with home2 and home3 next to it with neighbors jim and mary,
each seeing person1 and person2,
several items in home, including a missing ring and non-missing cash,
broken front but not broken back entrances to home,
an earthquake in sf,
what is the probability that home's alarm goes off?

Lifted Belief Propagation



Intuition for Anytime Lifted BP



Using only a portion of a model

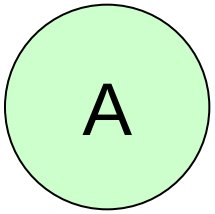
- By using only a portion, I don’t have to shatter other parts of the model.
- How can use only a portion?
- A solution for propositional models already exists: box propagation.

Box Propagation

- A way of getting bounds on query without examining entire network.

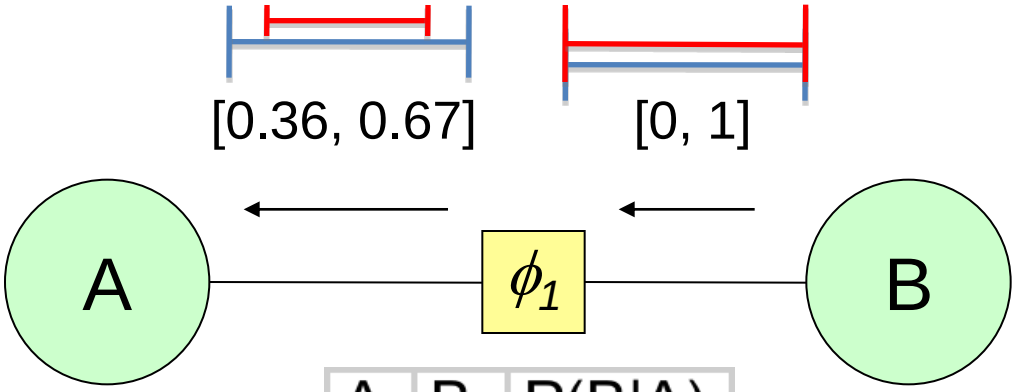


[0, 1]



Box Propagation

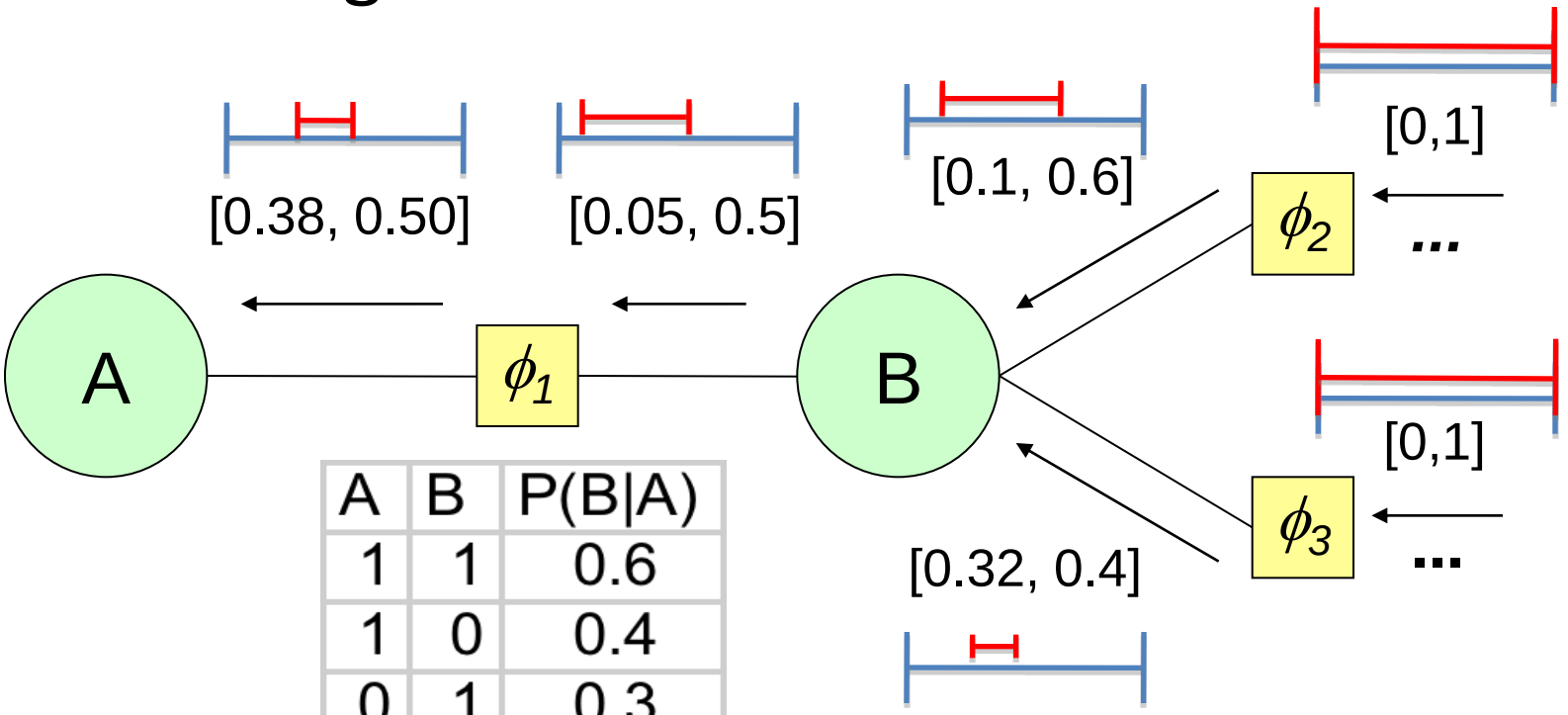
- A way of getting bounds on query without examining entire network.



A	B	P(B|A)
1	1	0.6
1	0	0.4
0	1	0.3
0	0	0.7

Box Propagation

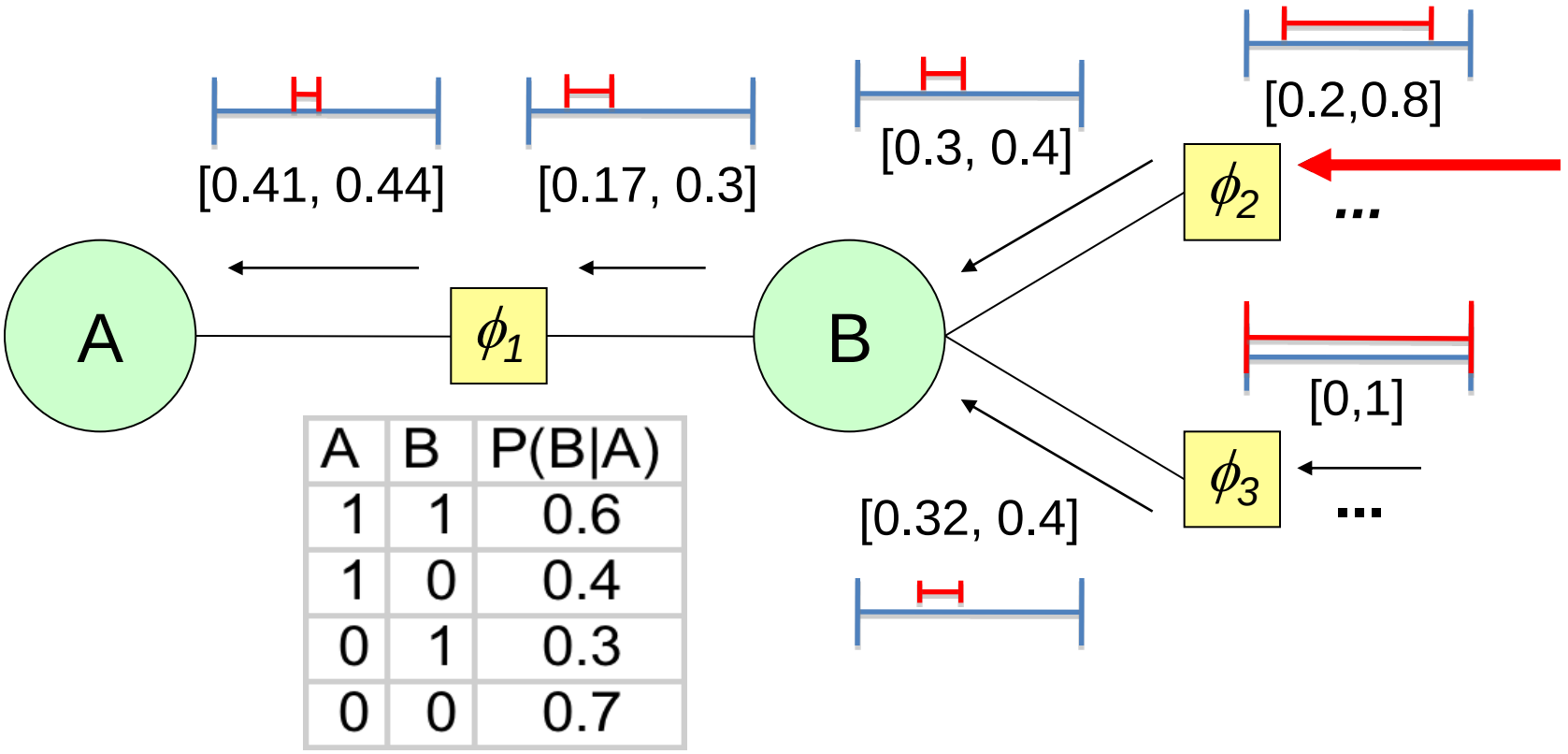
- A way of getting bounds on query without examining entire network.



A	B	P(B|A)
1	1	0.6
1	0	0.4
0	1	0.3
0	0	0.7

Box Propagation

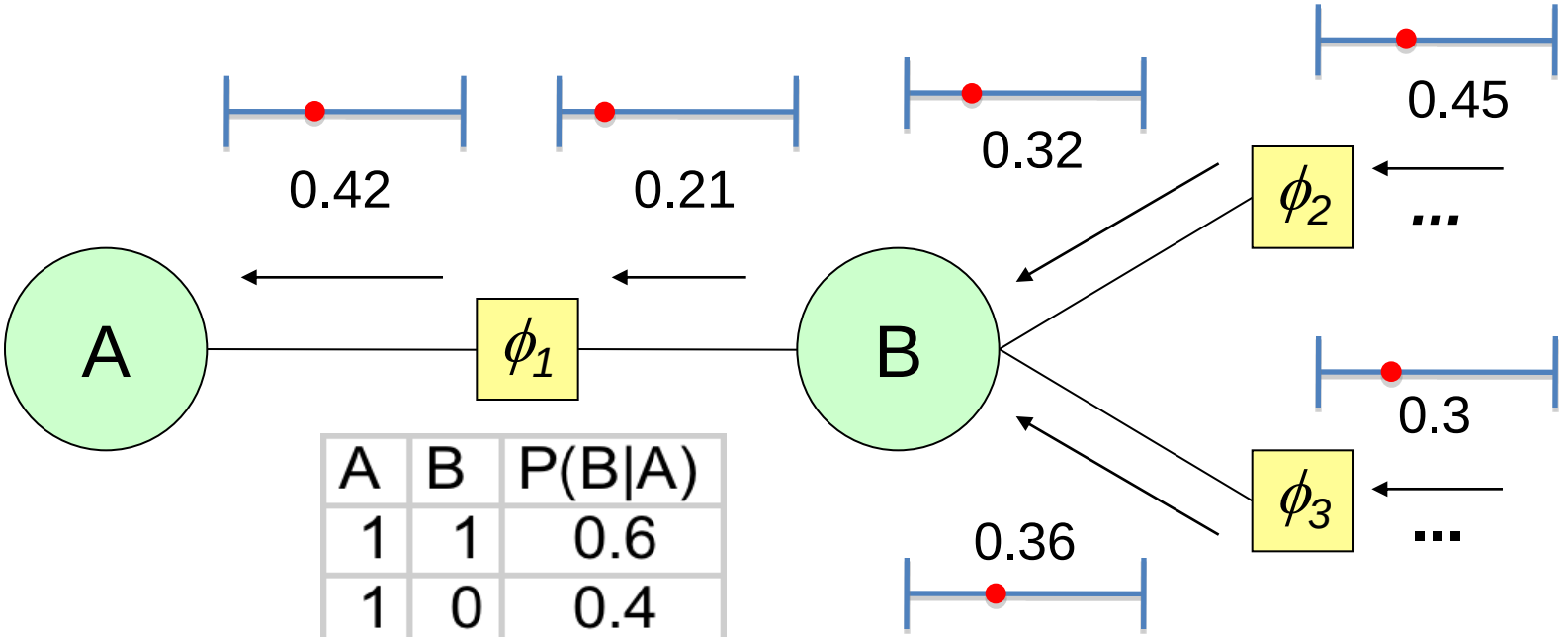
- A way of getting bounds on query without examining entire network.



A	B	P(B|A)
1	1	0.6
1	0	0.4
0	1	0.3
0	0	0.7

Box Propagation

- A way of getting bounds on query without examining entire network.



A	B	P(B|A)
1	1	0.6
1	0	0.4
0	1	0.3
0	0	0.7

Convergence after all messages are collected

Anytime Lifted BP

Incremental shattering + box propagation

Anytime Lifted Belief Propagation

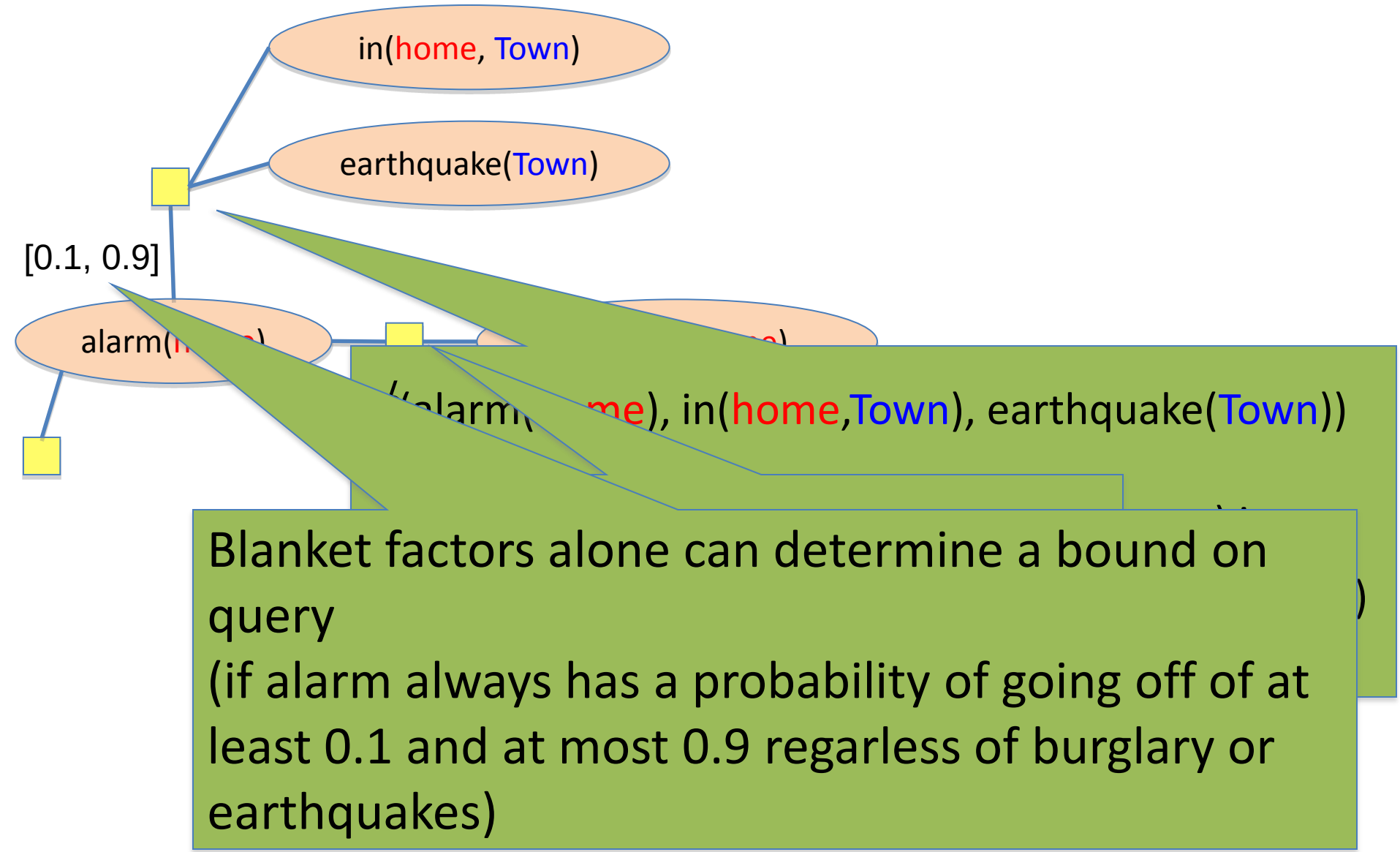
[0,1]



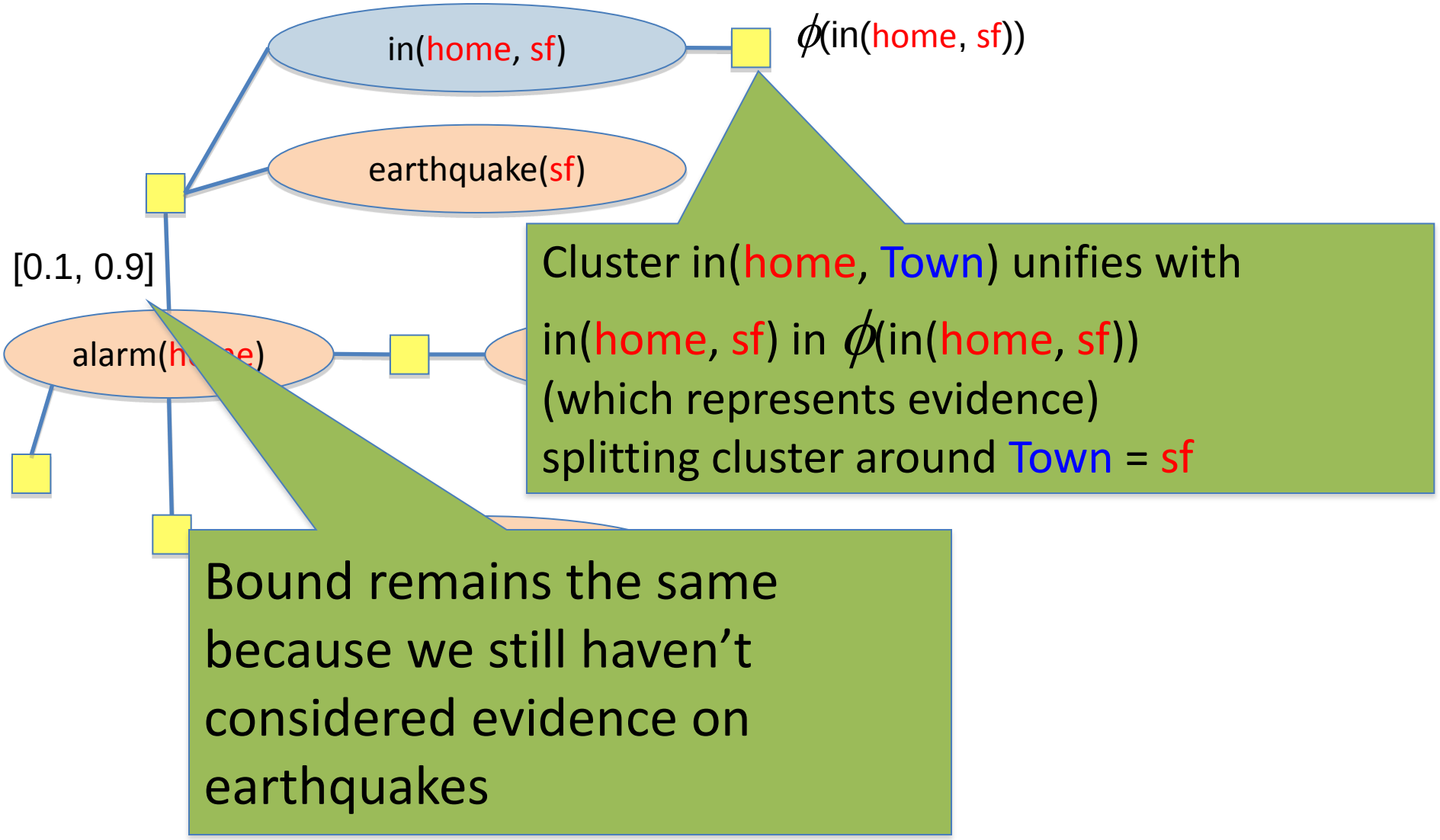
Start from query alone

The algorithm works by picking a cluster variable and including the factors in its blanket

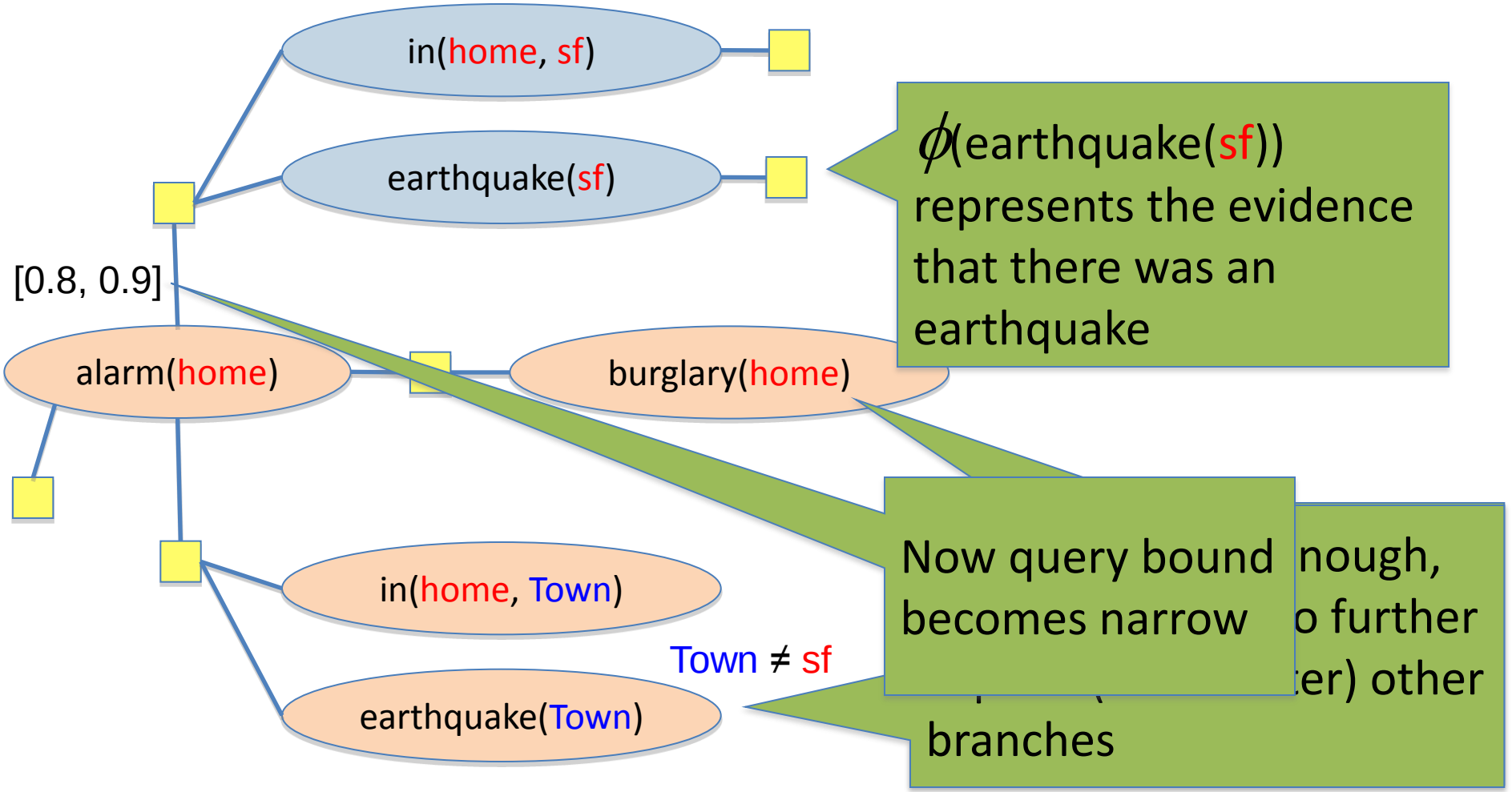
Anytime Lifted Belief Propagation



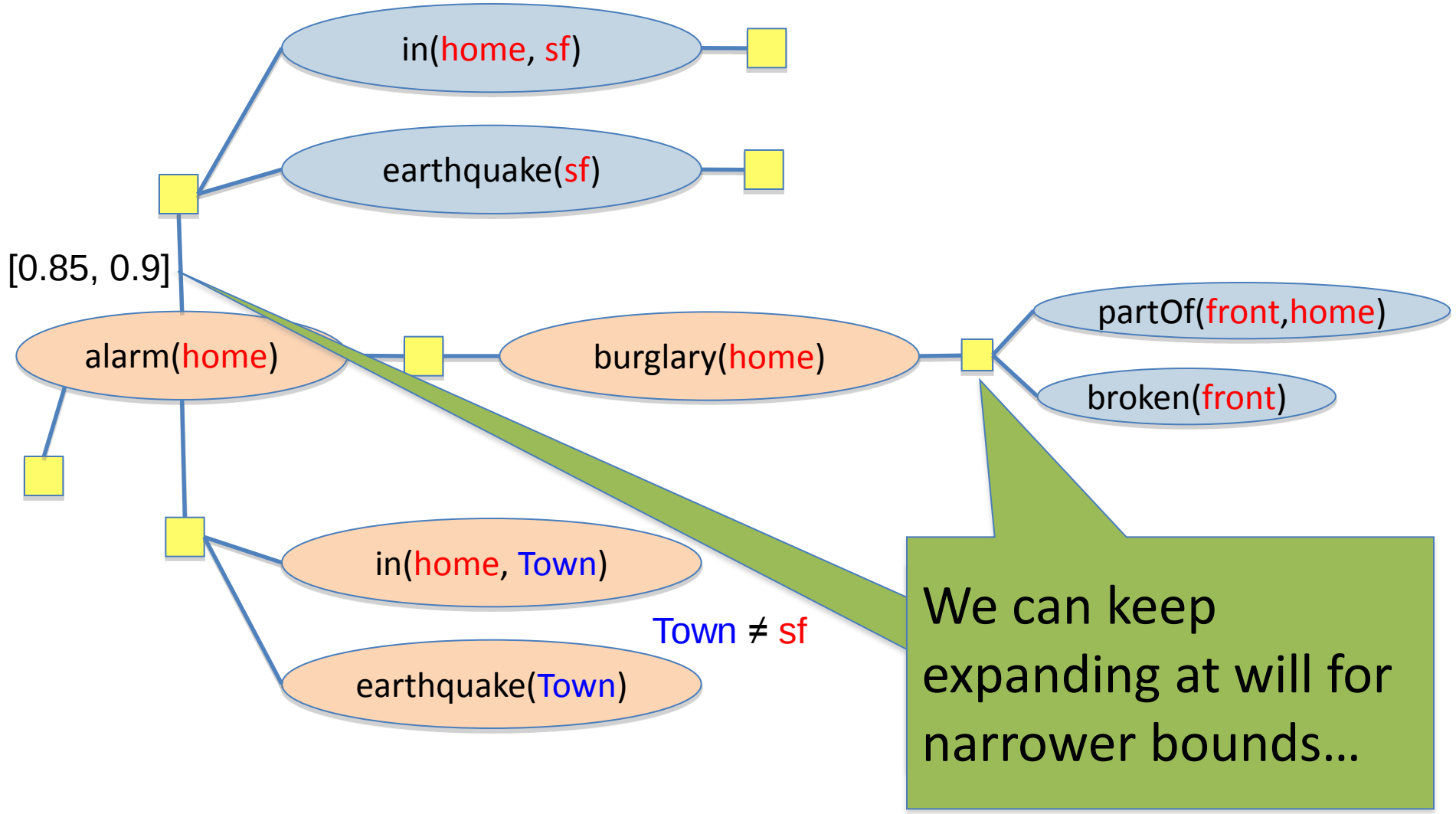
Anytime Lifted Belief Propagation



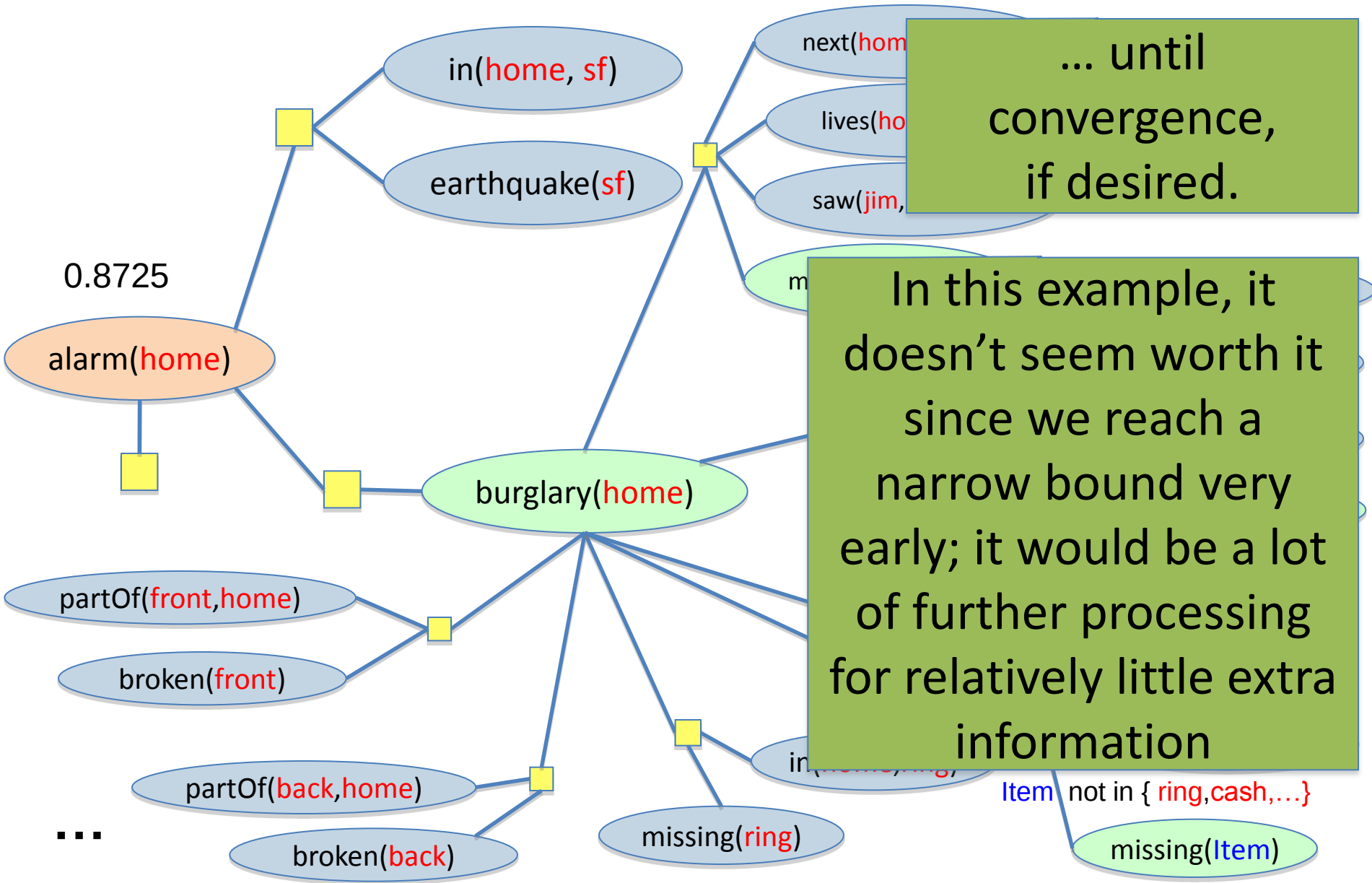
Anytime Lifted Belief Propagation



Anytime Lifted Belief Propagation



Anytime Lifted Belief Propagation



Another Anytime Lifted BP example

- A more realistic example
  - Large commonsense knowledge base
  - Large number of facts on many different constants, making shattering very expensive

Anytime Lifted BP Intuition

Let's consider a large knowledge base formed by parfactors.

```
φ(hasGoodOffer(Person),   offer(Job,Person),goodFor(Person,Job))
φ(goodFor(Person,Job),    cityPerson(Person),inCity(Job))
φ(goodFor(Person,Job),    goodEmployer(Job))
φ(goodFor(Person,Job),    involves(Subject,Job),likes(Subject,Person))
φ(goodEmployer(Job),      in(Subject,Job),profitable(Subject))
φ(likes(Subject,Person),  takesTeamWork(Subject),social(Person))
...
<many more parfactors representing rules>
...
0.9: offer(mary,Job), Job in {a,b,c}.
1.0: not offer(mary,Job), Job not in {a,b,
0.8: goodEmployer(Job), Job in {a,c}.
1.0: social(mary).
0.7: involves(ai,a).
1.0: likes(theory,frank).
1.0: likes(graphics, john).
1.0: inCity(c).
... <and many more such facts from a database, for example>
```

This is shorthand for a parfactor placing potentials 0.9 and 0.1 on `offer(mary,Job)` being true or false

Expensive shattering

Any two constants among the ones shown have distinct properties,
so their clusters become singletons,
so these singleton clusters appear isolated for each parfactor.
For example

```
φ(goodFor(Person,Job),
                    involves(Subject,Job),likes(Subject,Person))
```

gets shattered into

```
φ(goodFor(mary,a),     involves(theory,a),likes(theory,mary))
φ(goodFor(mary,b),     involves(theory,b),likes(theory,mary))
φ(goodFor(mary,c),     involves(theory,c),likes(theory,mary))
…
φ(goodFor(mary,a),     involves(ai,a),likes(ai,mary))
φ(goodFor(mary,b),     involves(ai,b),likes(ai,mary))
φ(goodFor(mary,c),     involves(ai,c),likes(ai,mary))
…
φ(goodFor(frank,a),    involves(theory,a),likes(theory,frank))
φ(goodFor(frank,b),    involves(theory,b),likes(theory,frank))
φ(goodFor(frank,c),    involves(theory,c),likes(theory,frank))
…
φ(goodFor(Person,Job), involves(Subject,Job),likes(Subject,Person)),
  Person not in {mary,frank, …},Subject not in {theory,ai, …},Job not in {a,b,c, …}
```

And that's just a single parfactor!

Anytime Lifted BP Intuition

We can usually tell a lot from a tiny fraction of a model.

```
φ(hasGoodOffer(Person),   offer(Job,Person),goodFor(Person,Job))
φ(goodFor(Person,Job),    cityPerson(Person),inCity(Job))
φ(goodFor(Person,Job),    goodEmployer(Job))
φ(goodFor(Person,Job),    involves(Subject,Job),likes(Subject,Person))
φ(goodEmployer(Job),      in(Subject,Job),profitable(Subject))
φ(likes(Subject,Person),  takesTeamWork(Subject),social(Person))
...
<many more parfactors representing rules>
...
0.9: offer(mary,Job), Job in {a,b,c}.
1.0: not offer(mary,Job), Job not in {a,b,c}.
0.8: goodEmployer(Job), Job in {a,c}.
1.0: social(mary).
0.7: involves(ai,a).
1.0: likes(theory,frank).
1.0: likes(graphics, john).
1.0: inCity(c).
... <and many more such facts from a database, for example>
```

Anytime Lifted BP Intuition

We can usually tell a lot from a tiny fraction of a model.

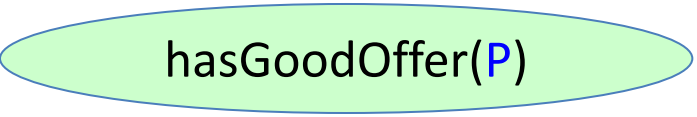
```
φ(hasGoodOffer(Person),   offer(Job,Person),goodFor(Person,Job))
φ(goodFor(Person,Job),    cityPerson(Person),inCity(Job))
φ(goodFor(Person,Job),    goodEmployer(Job))
φ(goodFor(Person,Job),    involves(Subject,Job),likes(Subject,Person))
φ(goodEmployer(Job),      in(Subject,Job),profitable(Subject))
φ(likes(Subject,Person),  takesTeamWork(S
...
<many more parfactors representing rule
...
0.9: offer(mary,Job), Job in {a,b,c}.
1.0: not offer(mary,Job), Job not in {a
0.8: goodEmployer(Job), Job in {a,c}.
1.0: social(mary).
0.7: involves(ai,a).
1.0: likes(theory,frank).
1.0: likes(graphics, john).
1.0: inCity(c).
... <and many more such facts from a database, for example>
```

If either **a** or **c** is indeed a good employer (0.8), has made an offer to **mary** (0.9), then it is likely that she has a good offer

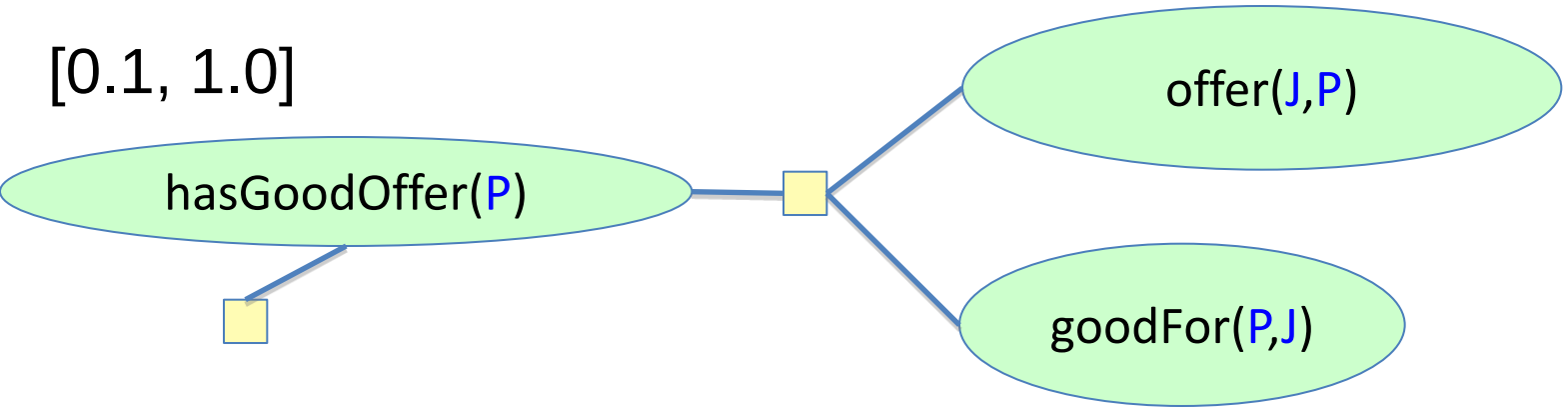
huge model!

Anytime Lifted BP Example

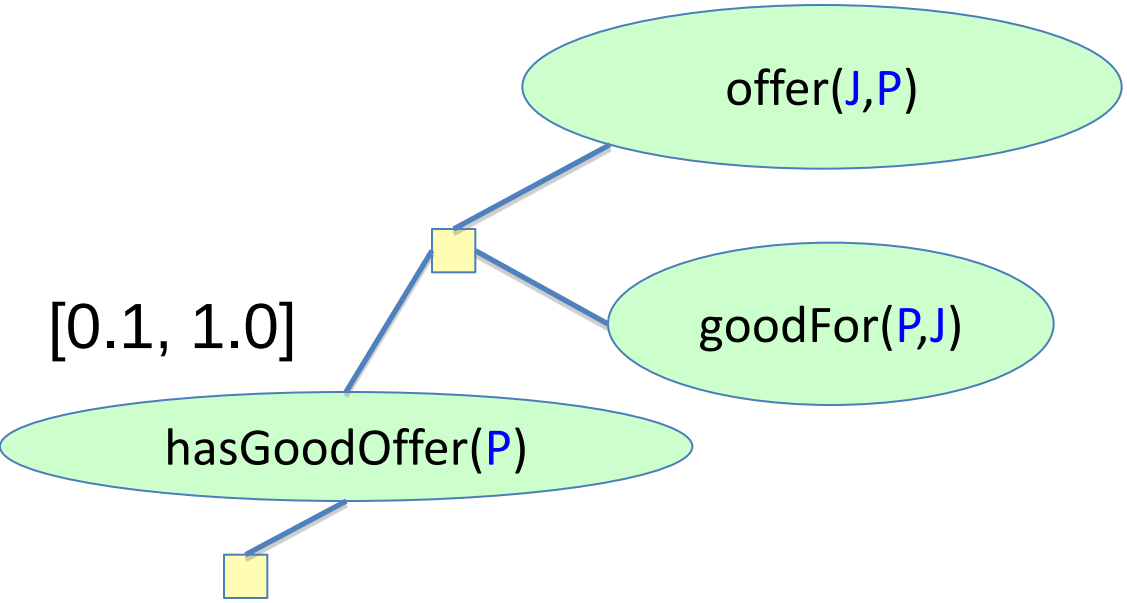
[0,1]



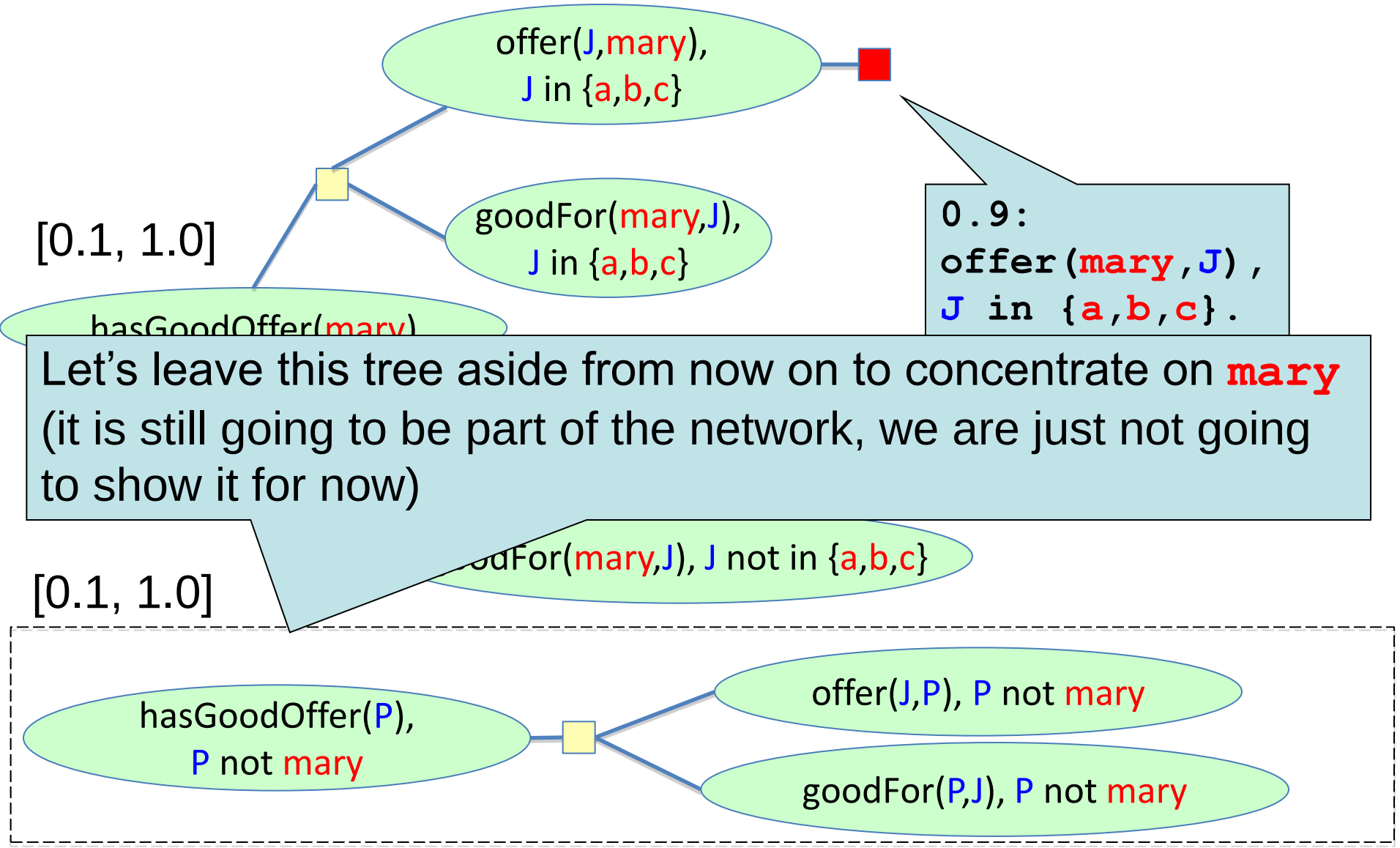
Anytime Lifted BP Example



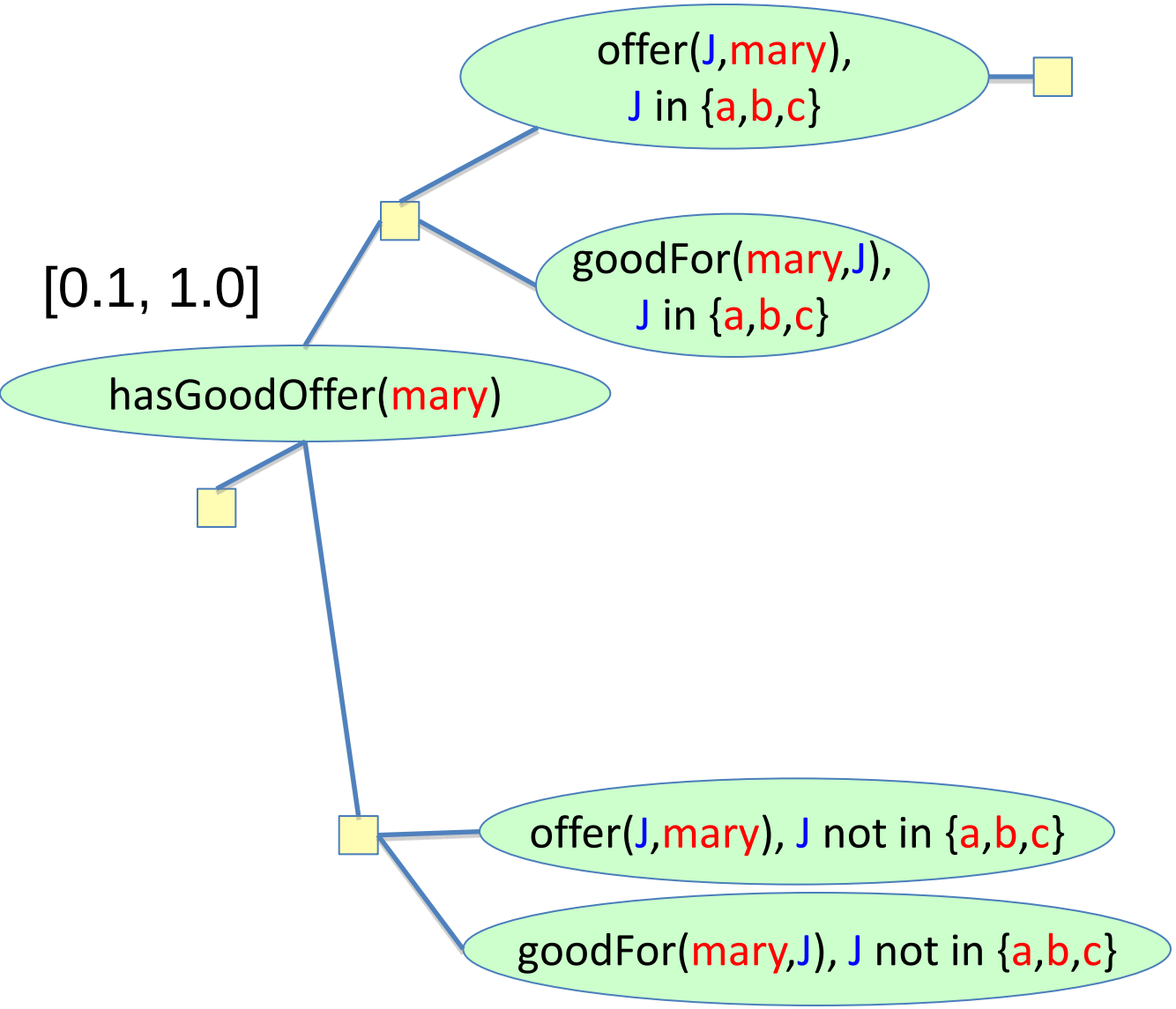
Anytime Lifted BP Example



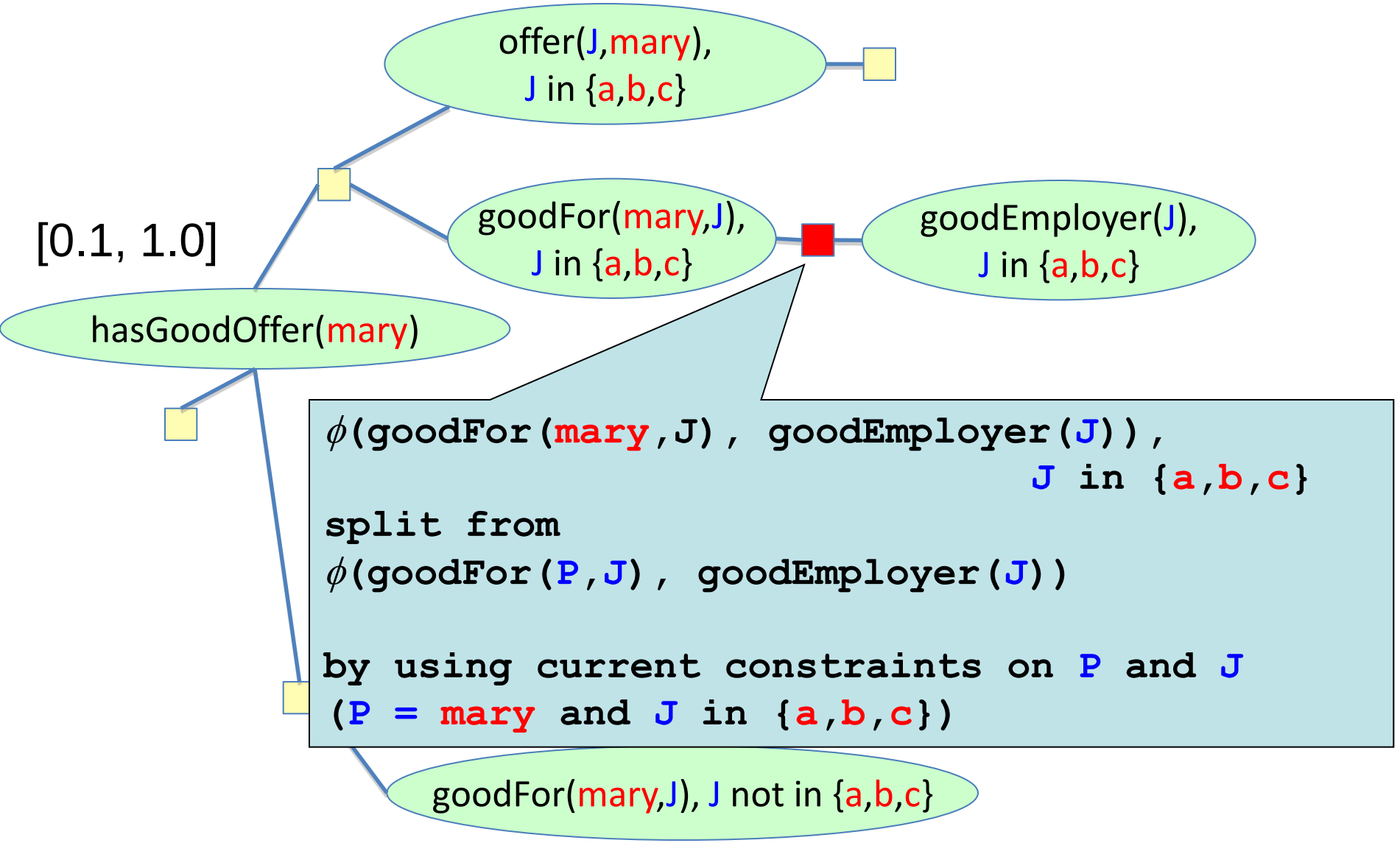
Anytime Lifted BP Example



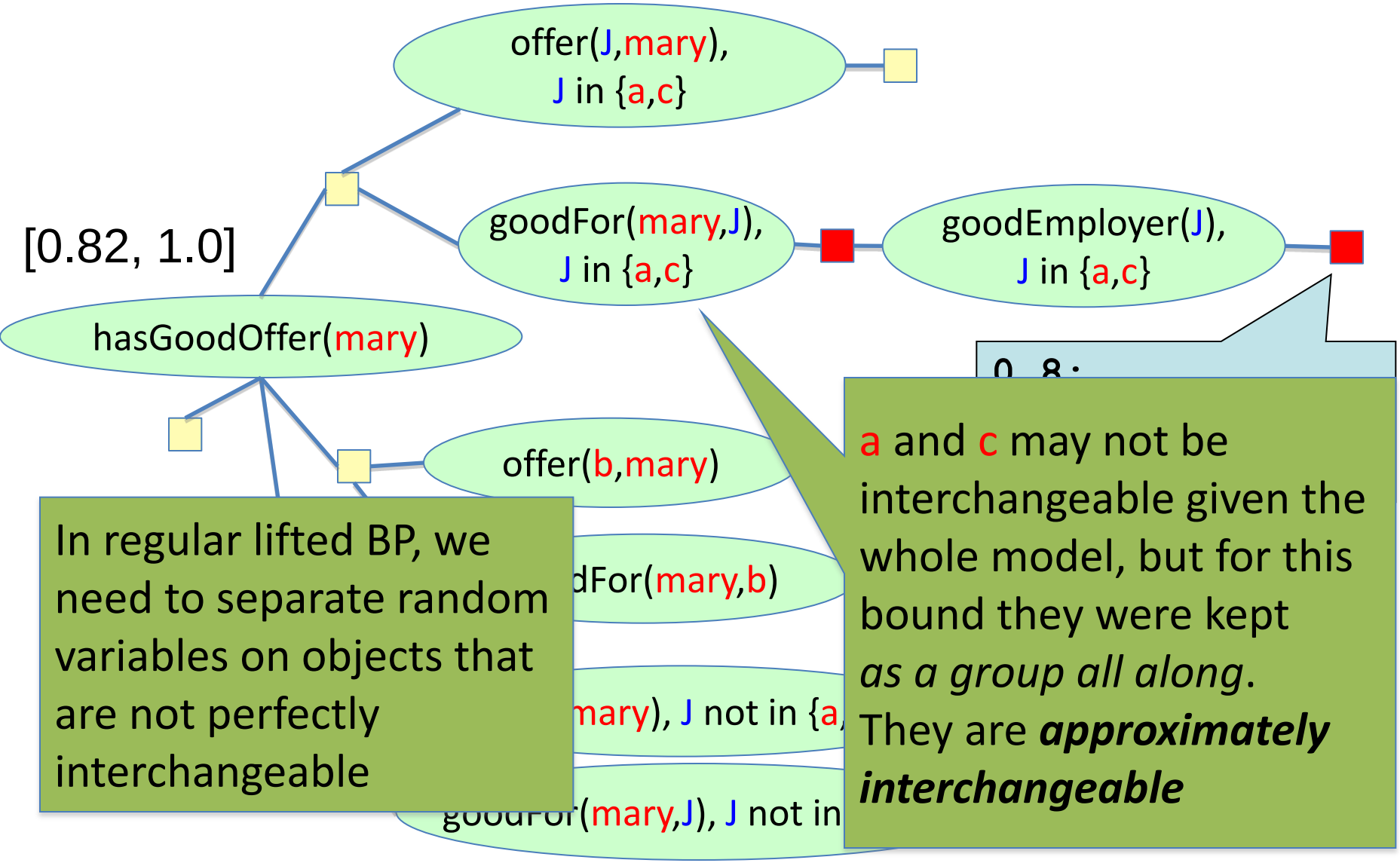
Anytime Lifted BP Example



Anytime Lifted BP Example



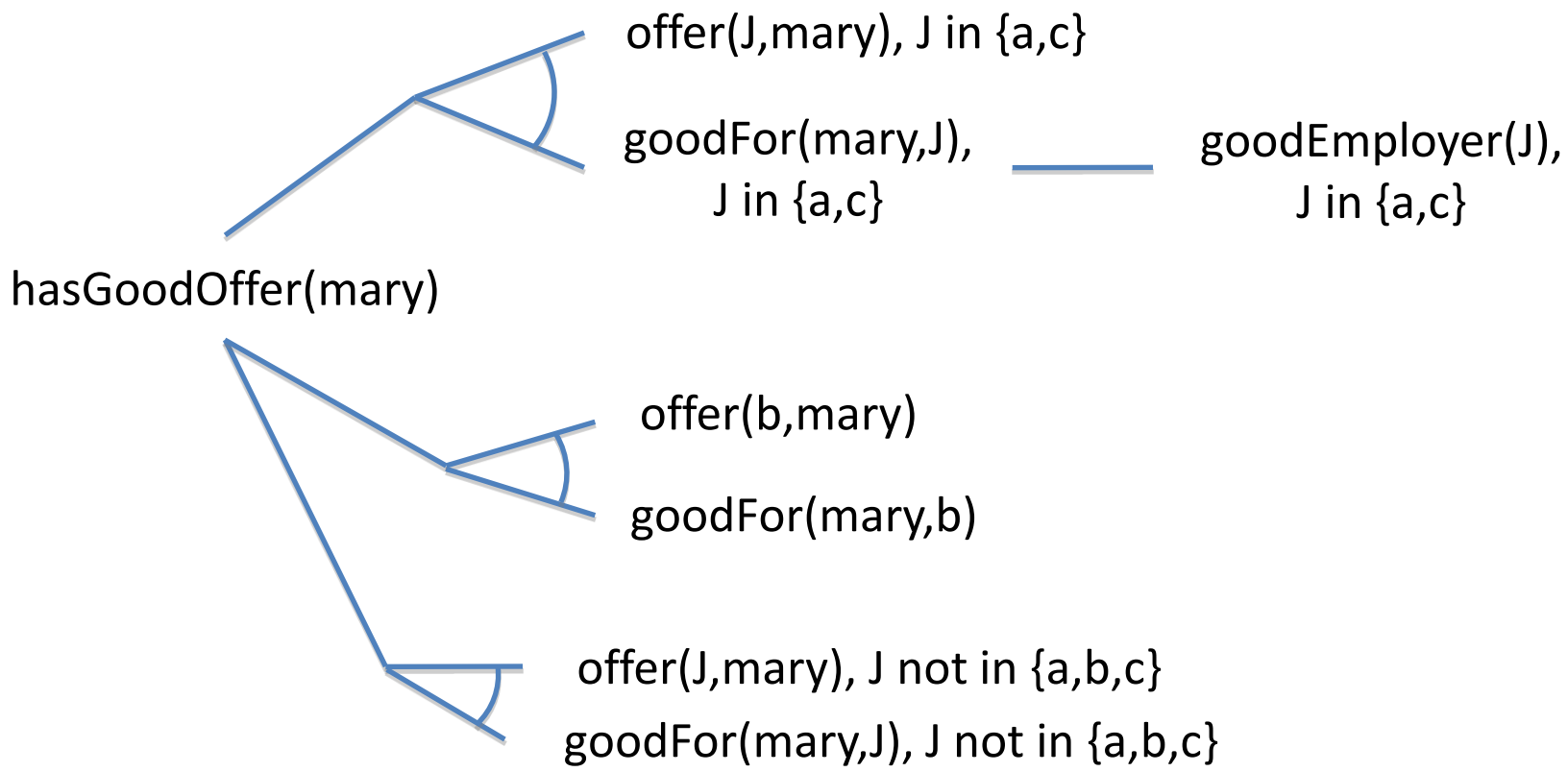
Anytime Lifted BP Example



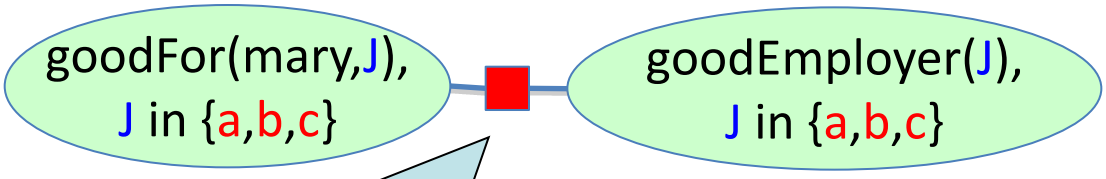
Final Remarks

Connection to Theorem Proving

- Incremental shattering corresponds to building a proof tree



Connection to Theorem Proving




```
ϕ(goodFor(mary,J), goodEmployer(J)),
                                  J in {a,b,c}
split from
ϕ(goodFor(P,J), goodEmployer(J))
```

This results from a unification step between
ϕ(goodFor(P,J), goodEmployer(J))
and
goodFor(mary,J), J in {a,b,c}
where **goodFor(P,J)** is unified with **goodFor(mary,J)**
through **P** = **mary**

Conclusions

- Most of query answer computed (potentially) much sooner than in Lifted BP
- Only the most necessary fraction of model considered and shatttered
- Sets of non-interchangeable objects still treated as groups
- Theorem proving-like probabilistic inference narrowing the gap from logic
- More intuitive algorithm,
  with natural explanations and proofs.

Future directions

- Which factor to expand next (for example using utilities).
- More flexible bounds (belief may be outside bounds but only with a small probability, for example)

Questions?

Speeding Up Inference in Markov Logic Networks by Preprocessing to Reduce the Size of the Resulting Grounded Network

Jude Shavlik



Sriraam Natarajan



Computer Sciences Department
University of Wisconsin, Madison USA

THE UNIVERSITY of WISCONSIN MADISON

Markov Logic Networks
(Richardson & Domingos, MLj 2006)



- A **probabilistic**, first-order logic
- Key idea

  compactly represent large graphical models

  using **weight = *w*** $\forall$ ***x, y, z f(x, y, z)***
- Standard approach

  1) assume **finite** number of constants

  2) create **all possible groundings**

  3) perform statistical inference (often via **sampling**)

The Challenge We Address

- Creating all possible groundings can be daunting

- A story …

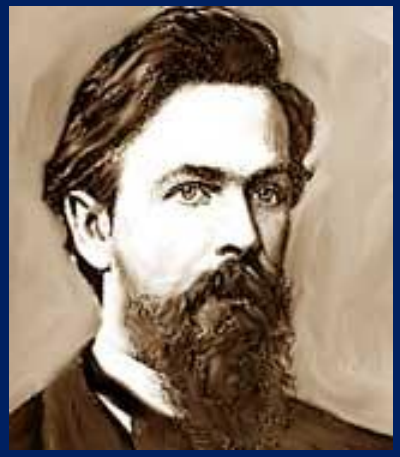


Given: an MLN and data

Do: quickly find an equivalent, reduced MLN

Computing Probabilities in MLNs

Probability(World S)



$$= (1 / Z) \times$$

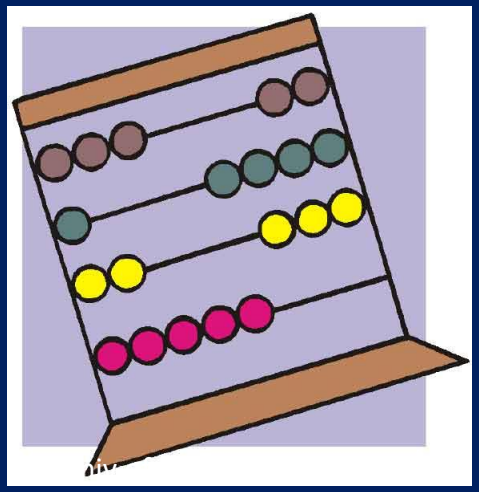
$$\exp\{ \sum_{i \in \text{formulae}} \text{weight}_i \times \text{numberTimesTrue}(f_i, S) \}$$

Counting Satisfied Groundings

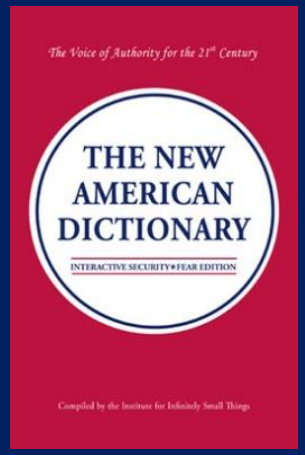
Typically lots of redundancy in FOL sentences

$$\forall\, x, y, z \;\; p(x) \wedge q(x, y, z) \wedge r(z) \;\rightarrow\; w(x, y, z)$$

If p(John) = false,
then formula = true
for all Y and Z values



Some Terminology



Three kinds of literals ('predicates')

Evidence: truth value known

Query: want to know prob's of these

Hidden: other

Factoring Out the Evidence

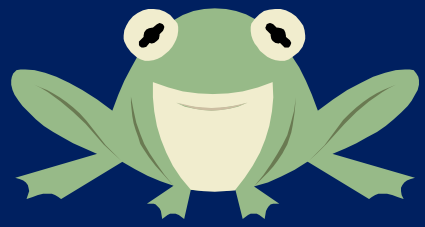


Let A = weighted sum of formula satisfied by evidence

Let B_i = weighted sum of formula in world i **not** satisfied by evidence

$$\text{Prob(world i)} = \frac{e^{A} e^{Bi}}{e^{A} e^{B1} + \ldots + e^{An} e^{Bn}}$$

Key Idea of Our FROG Algorithm



Efficiently factor out those formula groundings that evidence satisfies

- Can produce many orders-of-magnitude smaller Markov networks
- Can eliminate need for approximate inference, if resulting Markov net small/disconnected enough
- Resulting Markov net compatible with other speed-up methods, such as lifted and lazy inference, knowledge-based model construction

Worked Example

$\forall$ x, y, z GradStudent(x) $\wedge$ Prof(y) $\wedge$ Prof(z) $\wedge$ TA(x, z) $\wedge$ SameGroup(y, z)
$\rightarrow$ AdvisedBy(x, y)

10,000	People at some school	**The Evidence**
2000	Graduate students	
1000	Professors	
1000	TAs	
500	Pairs of professors in the same group	

Total Num of Groundings = $|x| \times |y| \times |z| = 10^{12}$

10^{12}

GradStudent(x) ∧ Prof(y) ∧ Prof(z) ∧ TA(x,z) ∧ SameGroup(y,z) → AdvisedBy(x,y)

FROG keeps only these *X* values

GradStudent(x)

GradStudent(P1)
¬ GradStudent(P2)
GradStudent(P3)
…

True

GradStudent(P1)
GradStudent(P3)
…

2000 Grad Students

False

¬ GradStudent(P2)
¬ GradStudent(P4)
…

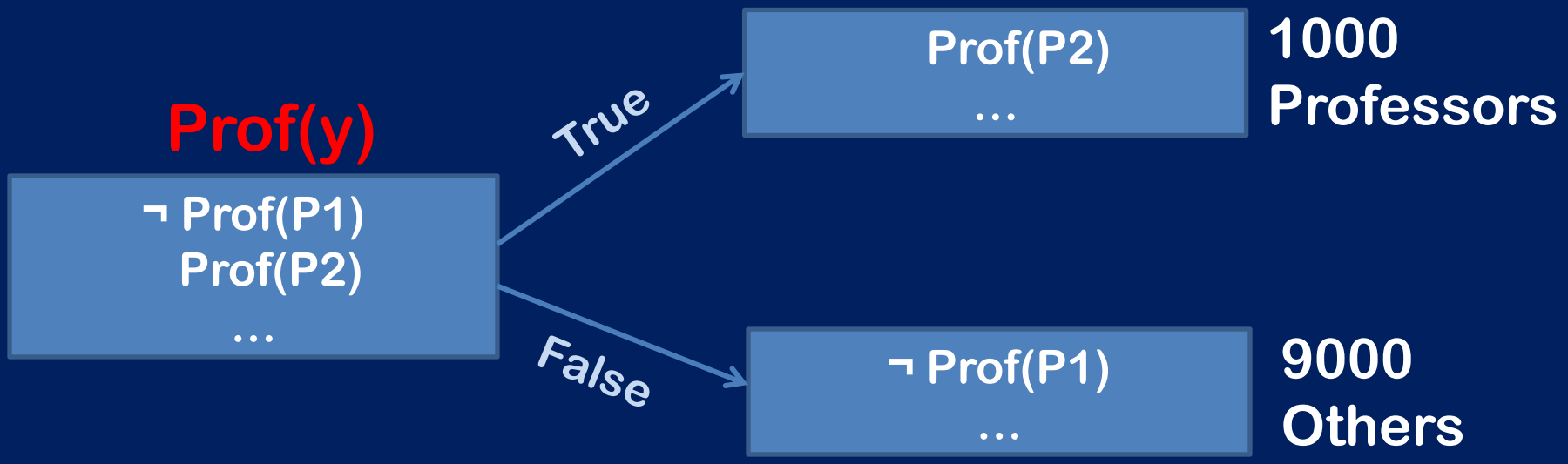
8000 Others

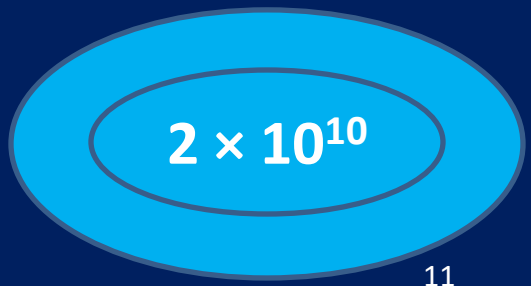
All these values for *X* satisfy the clause, <u>regardless</u> of *Y* and *Z*

2×10^{11}

Instead of 10^4 values for *X*, have 2×10^3

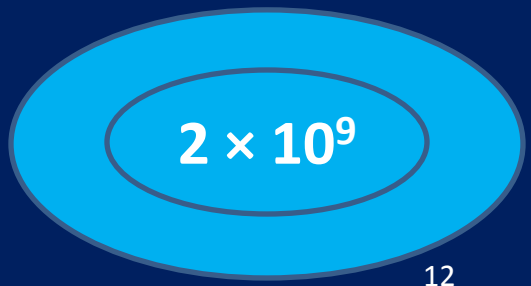
GradStudent(x) ∧ Prof(y) ∧ Prof(z) ∧ TA(x,z) ∧ SameGroup(y,z) → AdvisedBy(x,y)



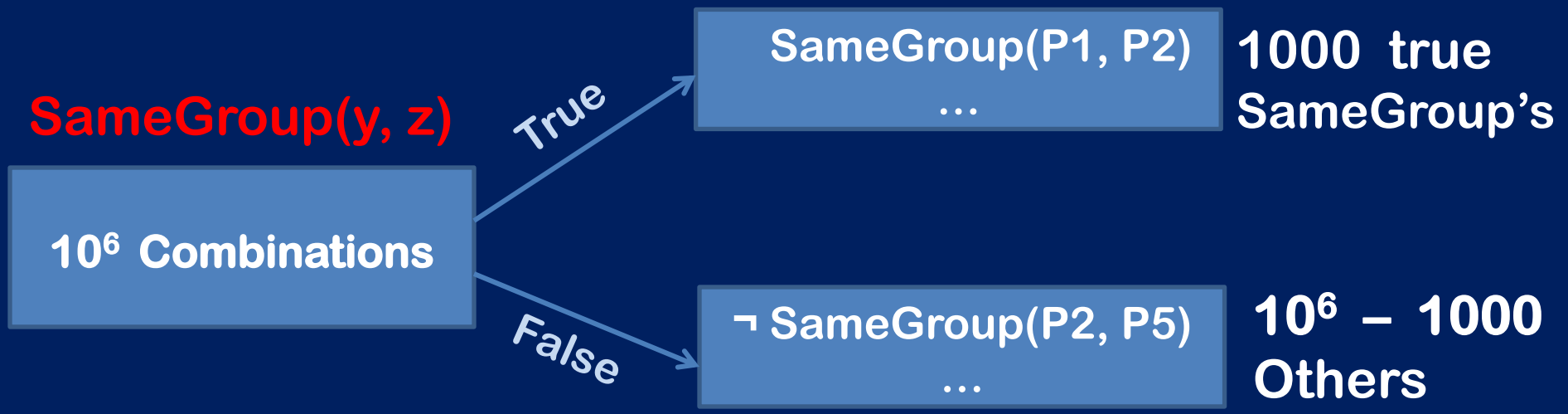


GradStudent(x) ∧ Prof(y) ∧ Prof(z) ∧ TA(x,z) ∧ SameGroup(y,z) → AdvisedBy(x,y)

<<< Same as Prof(y) >>>



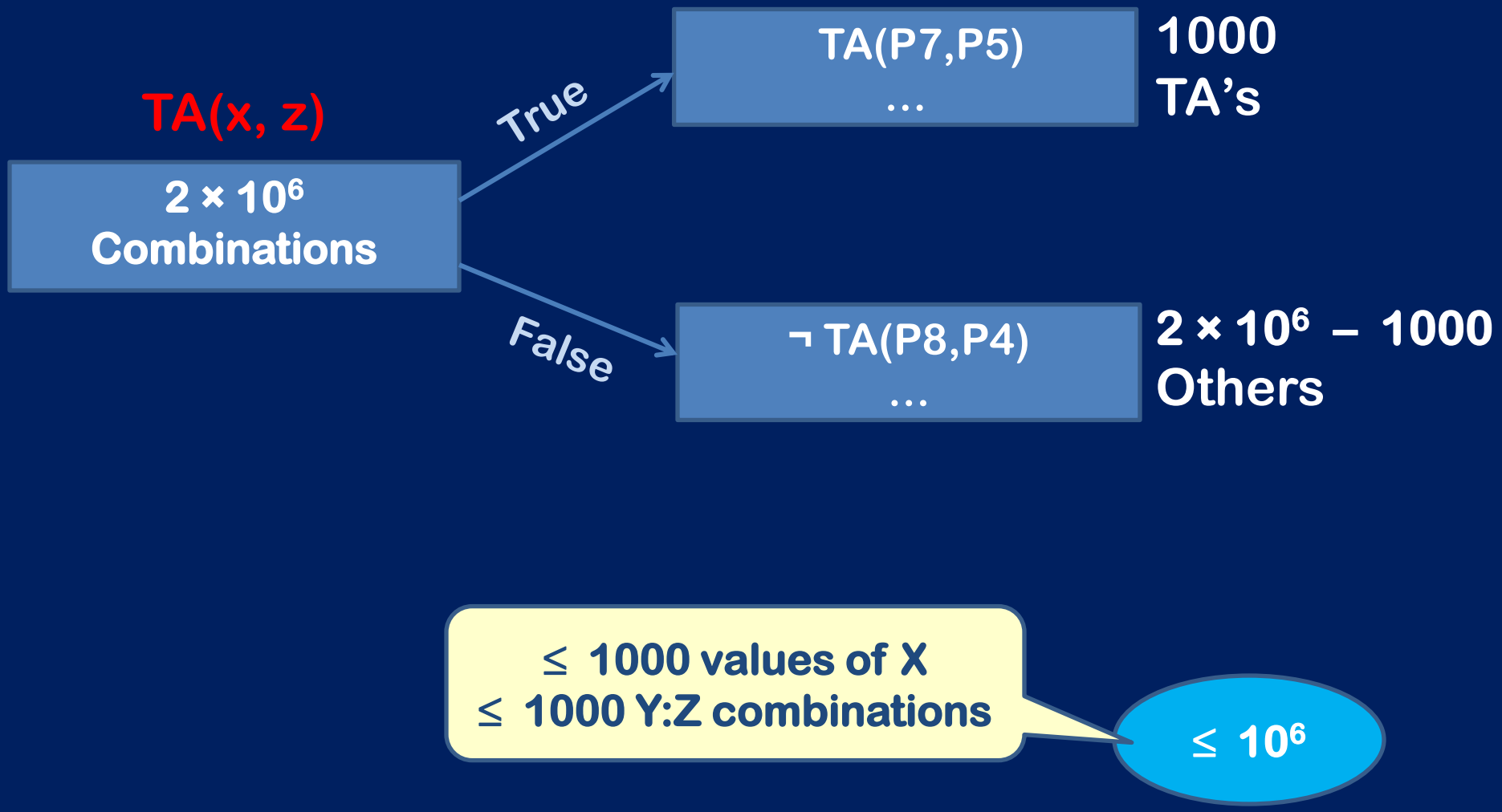
GradStudent(x) ∧ Prof(y) ∧ Prof(z) ∧ TA(x,z) ∧ SameGroup(y,z) → AdvisedBy(x,y)



2000 values of X
1000 Y:Z combinations

2×10^6

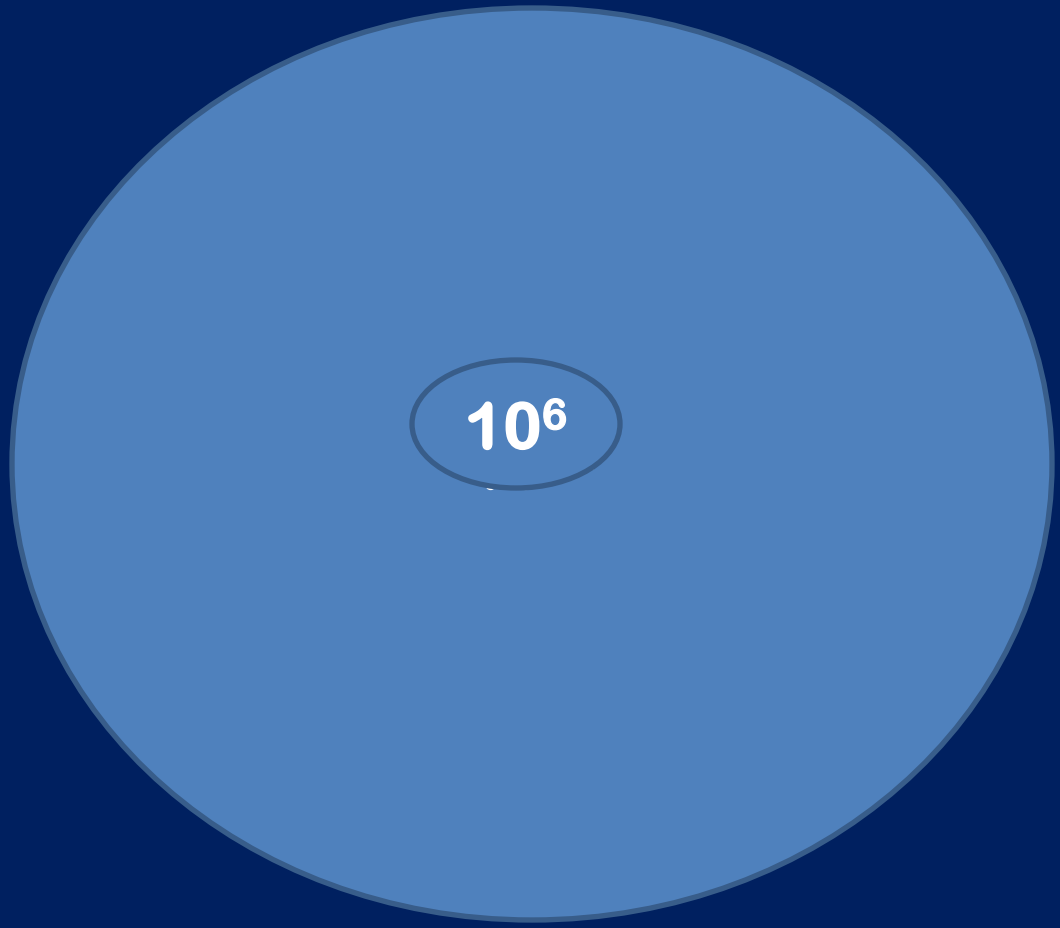
GradStudent(x) ∧ Prof(y) ∧ Prof(z) ∧ TA(x,z) ∧ SameGroup(y,z) → AdvisedBy(x,y)



GradStudent(x) ∧ Prof(y) ∧ Prof(z) ∧ TA(x,z) ∧ SameGroup(y,z) → AdvisedBy(x,y)

Original number of groundings = 10^{12}

Final number of groundings ≤ 10^{6}

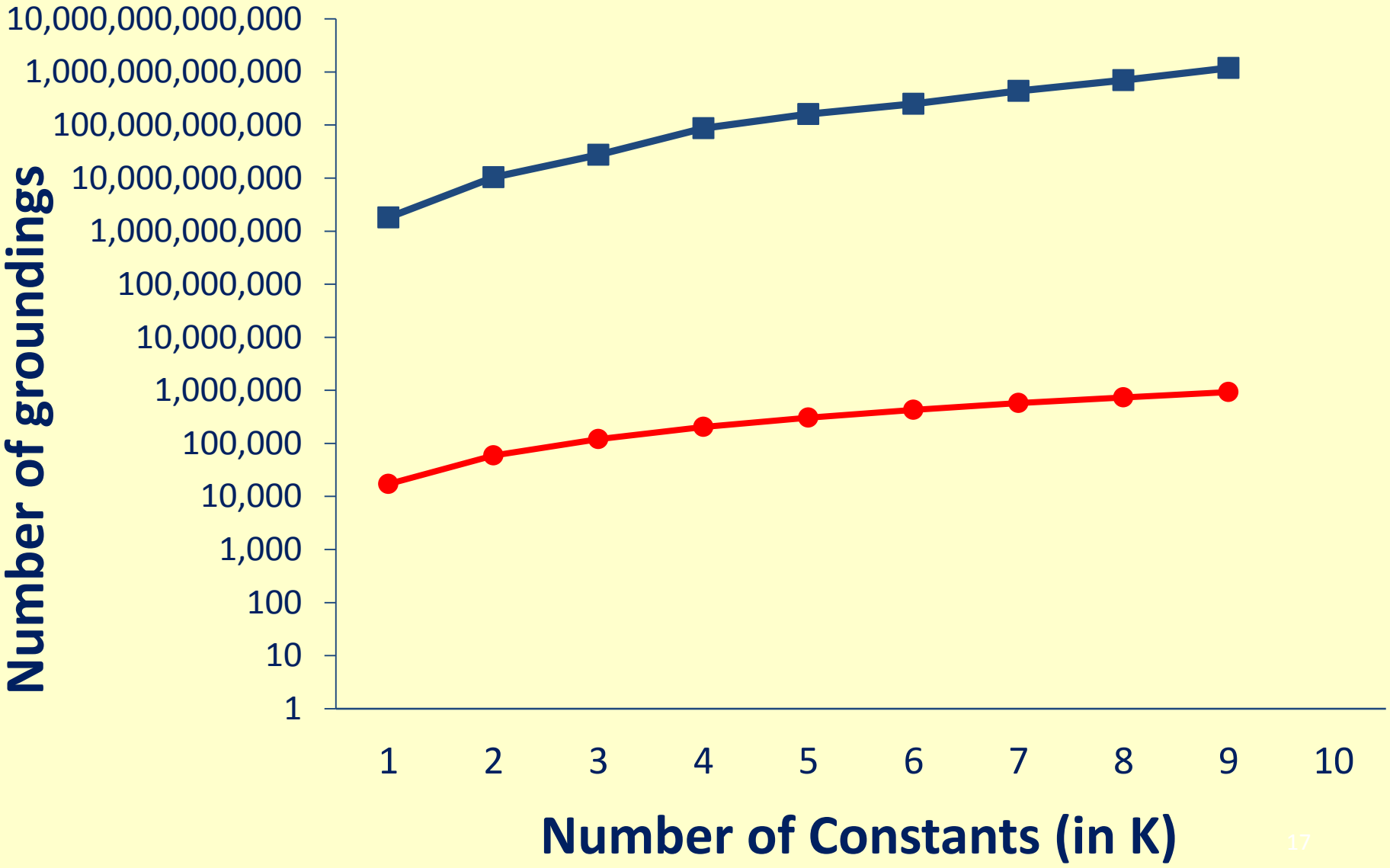


Some Algorithmic Details

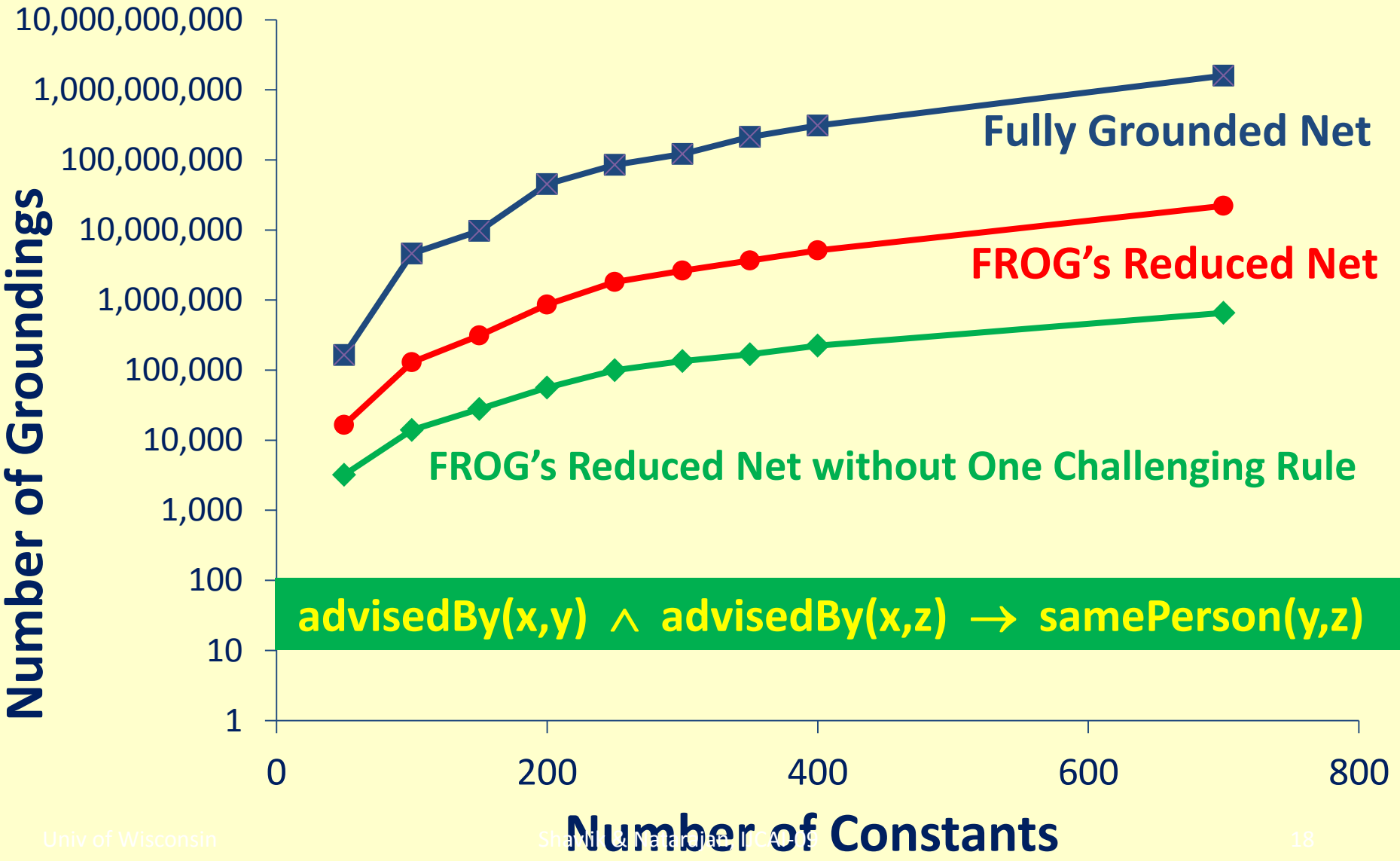


- Initially store 10^{12} groundings with 10^4 space
- Storage needs grow because literals cause variables to 'interact'
  - P(x, y, z) might require O(10^{12}) space
- Order literals 'reduced' impacts storage needs
  - Simple heuristic (see paper) chooses literal to process next – or try all permutations
- Can merge inference rules after reduction
  - After reduction, sample rule only has **advisedBy(x,y)**

Empirical Results: CiteSeer



Empirical Results: UWash-CSE



Runtimes



- On Full UWash-CSE (27 rules)
  - FROG takes 4.2 sec
- On CORA (2K rules) and CiteSeer (8K rules)
  - FROG takes less than 700 msec per rule
- On CORA
  - Alchemy's Lazy Inference takes 94 mins to create its initial network
  - FROG takes 30 mins and produces small enough network (10^6 nodes) that lazy inference not needed

Related Work



- Lazy MLN inference
  - Singla & Domingos (2006), Poon et al (2008)
  - FROG: precompute instead of lazily calculate
- Lifted inference
  - Braz et al (2005), Singla & Domingos (2008), Milch et al (2008), Riedel (2008), Kisynski & Poole (2009), Kersting et al (2009)
- Knowledge-based model construction
  - Wellman et al (1992)
  - FROG also exploits KBMC

Future Work



- Efficiently handle *small changes* to truth values of evidence
- Combine FROG with *Lifted Inference*
- Exploit commonality *across* rules
- Integrate with weight and rule *learning*

Conclusion

- MLN's count the satisfied groundings of FOL formula
- Many ways a formula can be satisfied

$$P(x) \vee Q(x, y) \vee R(x, y, z) \vee \neg S(y) \vee \neg T(x, y)$$

- Our FROG algorithm efficiently counts groundings satisfied by evidence
- FROG can reduce number of groundings by several orders of magnitude
- Reduced network compatible with lifted and lazy inference, etc

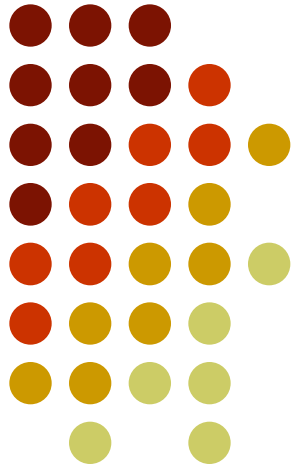


Probabilistic Theorem Proving

Pedro Domingos

Dept. of Computer Science & Eng.

University of Washington



Desiderata for Lifted Inference



- Exploit redundancy
- Scale to real domains
- Handle determinism
- Exploit logical structure
- Do for probabilistic logical models everything theorem proving does for logic

Probabilistic Theorem Proving

meets all these desiderata.

Probabilistic Theorem Proving



Given Probabilistic knowledge base *K*
Query formula *Q*

Output *P(Q|K)*

Propositional Logic



- **Atoms:** Symbols representing propositions
- **Logical connectives:** ¬, Λ, V, etc.
- **Knowledge base:** Set of formulas
- Every KB can be converted to **CNF**
  - *CNF:* Conjunction of clauses
  - *Clause:* Disjunction of literals
  - *Literal:* Atom or its negation
- **Entailment:** Does KB entail query?

Theorem Proving

TP(*KB*, *Query*)

$KB_Q \leftarrow KB \cup \{\neg Query\}$

return ¬SAT(CNF(KB_Q))



Satisfiability (DPLL)



SAT(*CNF*)

if *CNF* is empty **return** *True*

if *CNF* contains empty clause **return** *False*

choose an atom *A*

return SAT(CNF(*A*)) V SAT(CNF(¬*A*))

First-Order Logic



- Atom: Predicate(Variables,Constants)
  E.g.: $Friends(Anna, x)$
- Ground atom: All arguments are constants
- Quantifiers: $\forall$, $\exists$
- This talk: Finite, Herbrand interpretations

First-Order Theorem Proving



- **Propositionalization**
  1. Form all possible ground atoms
  2. Apply propositional theorem prover
- **Lifted Inference: Resolution**
  - Resolve pairs of clauses until empty clause derived
  - Unify literals by substitution, e.g.: $x = Bob$ unifies $Friends(Anna, x)$ and $Friends(Anna, Bob)$

$$\frac{\neg Friends(Anna, x) \vee Happy(x) \qquad Friends(Anna, Bob)}{Happy(Bob)}$$

Probabilistic Knowledge Bases



PKB = Set of formulas and their probabilities

+ Consistency + Maximum entropy

= Set of formulas and their weights

(a.k.a. Markov logic network)

= Set of formulas and their potentials

(1 if formula true, ϕ_i if formula false)

$$P(world) = \frac{1}{Z} \prod_i \phi_i^{n_i(world)}$$

Weighted Model Counting

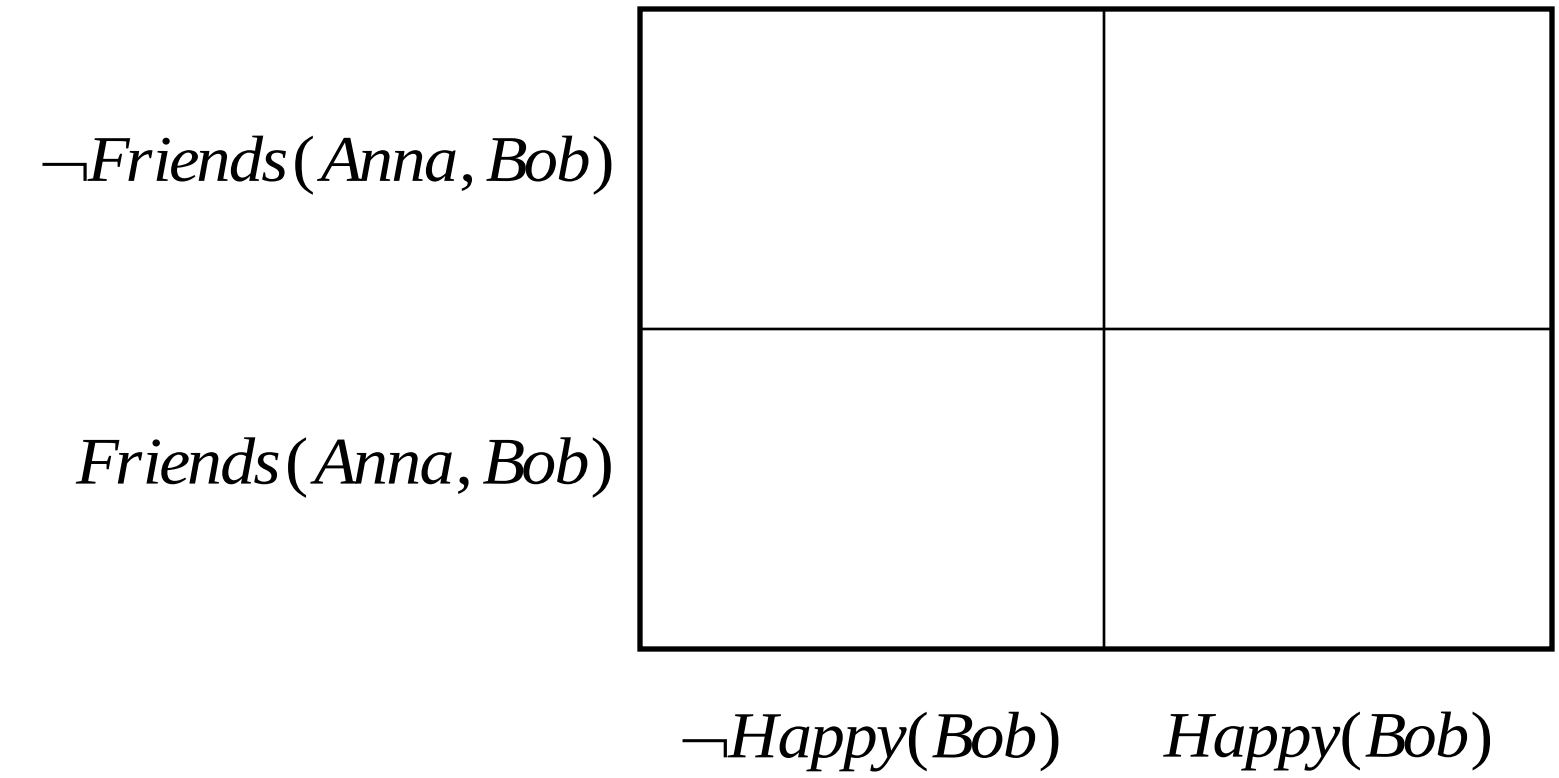


- ModelCount(CNF) = # worlds that satisfy CNF
- Assign a weight to each literal
- Weight(world) = $\prod$ weights(true literals)
- Weighted model counting:
  Given CNF *C* and literal weights *W*

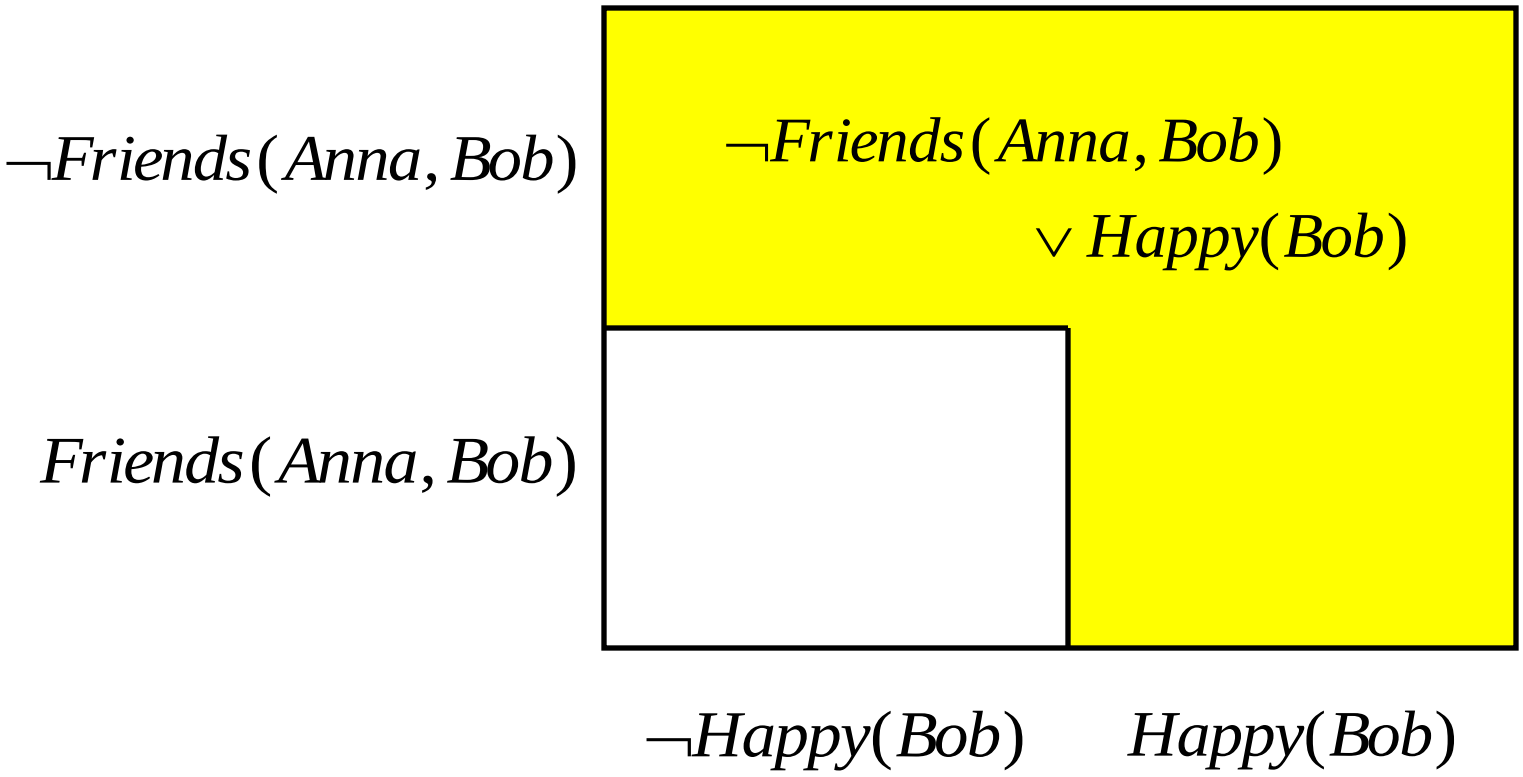
  Output Σ weights(worlds that satisfy *C*)

PTP is reducible to lifted WMC

Example

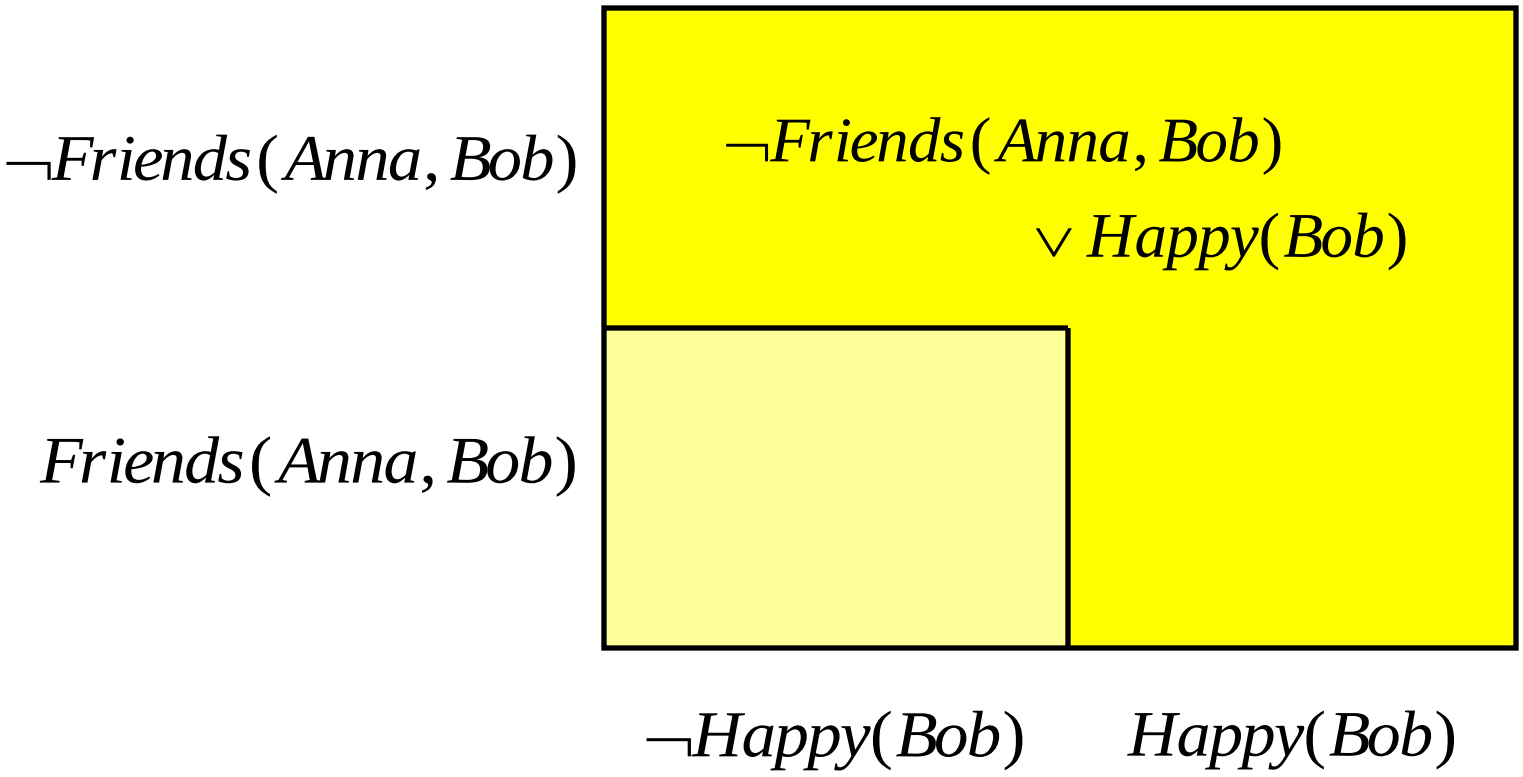

$\neg Friends(Anna, Bob)$
$Friends(Anna, Bob)$
$\neg Happy(Bob)$
$Happy(Bob)$

Example


$\neg Friends(Anna, Bob)$
$\neg Friends(Anna, Bob)$
$\vee Happy(Bob)$
$Friends(Anna, Bob)$
$\neg Happy(Bob)$
$Happy(Bob)$

Example

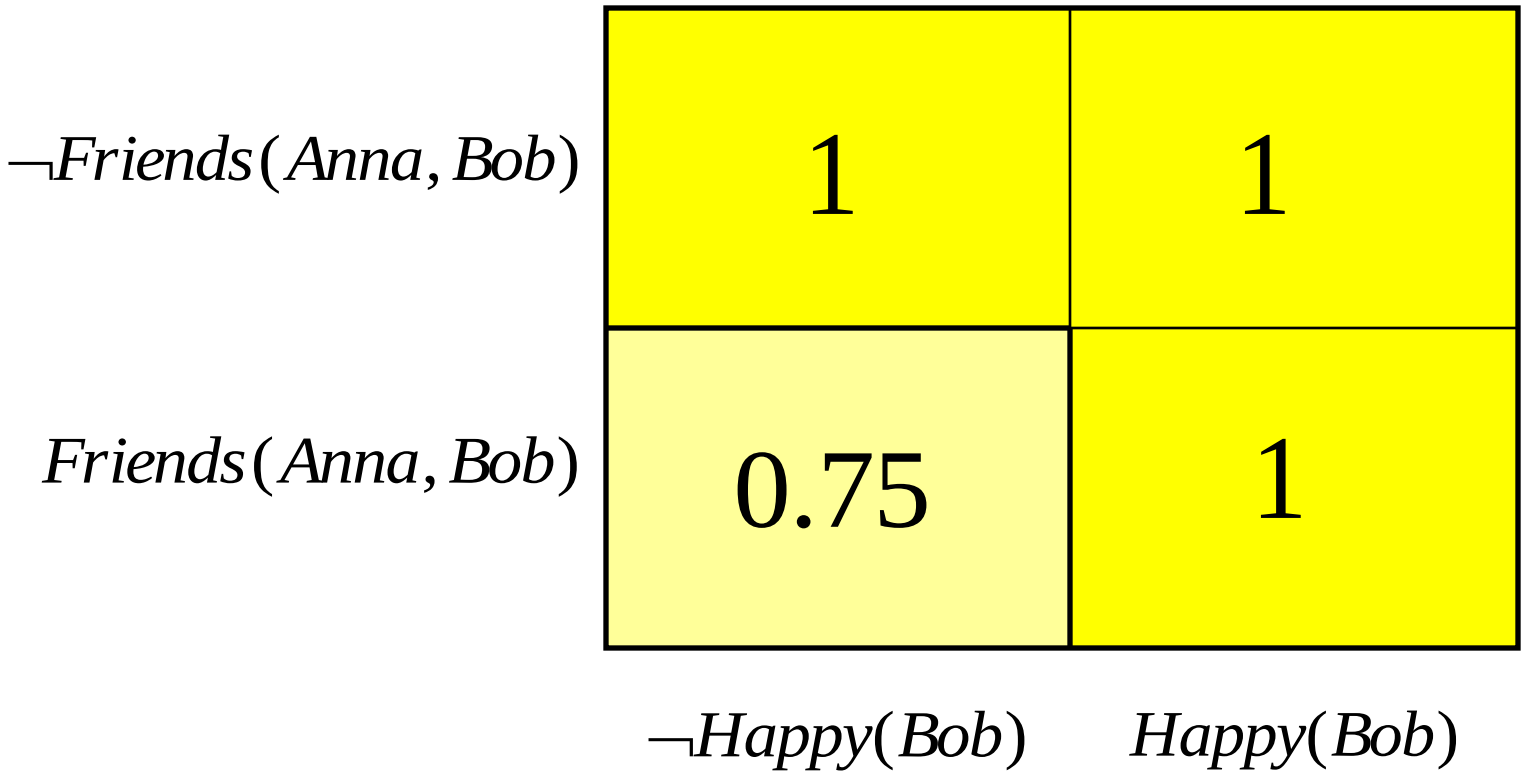
$$P(\neg Friends(Anna, Bob) \vee Happy(Bob)) = 0.8$$



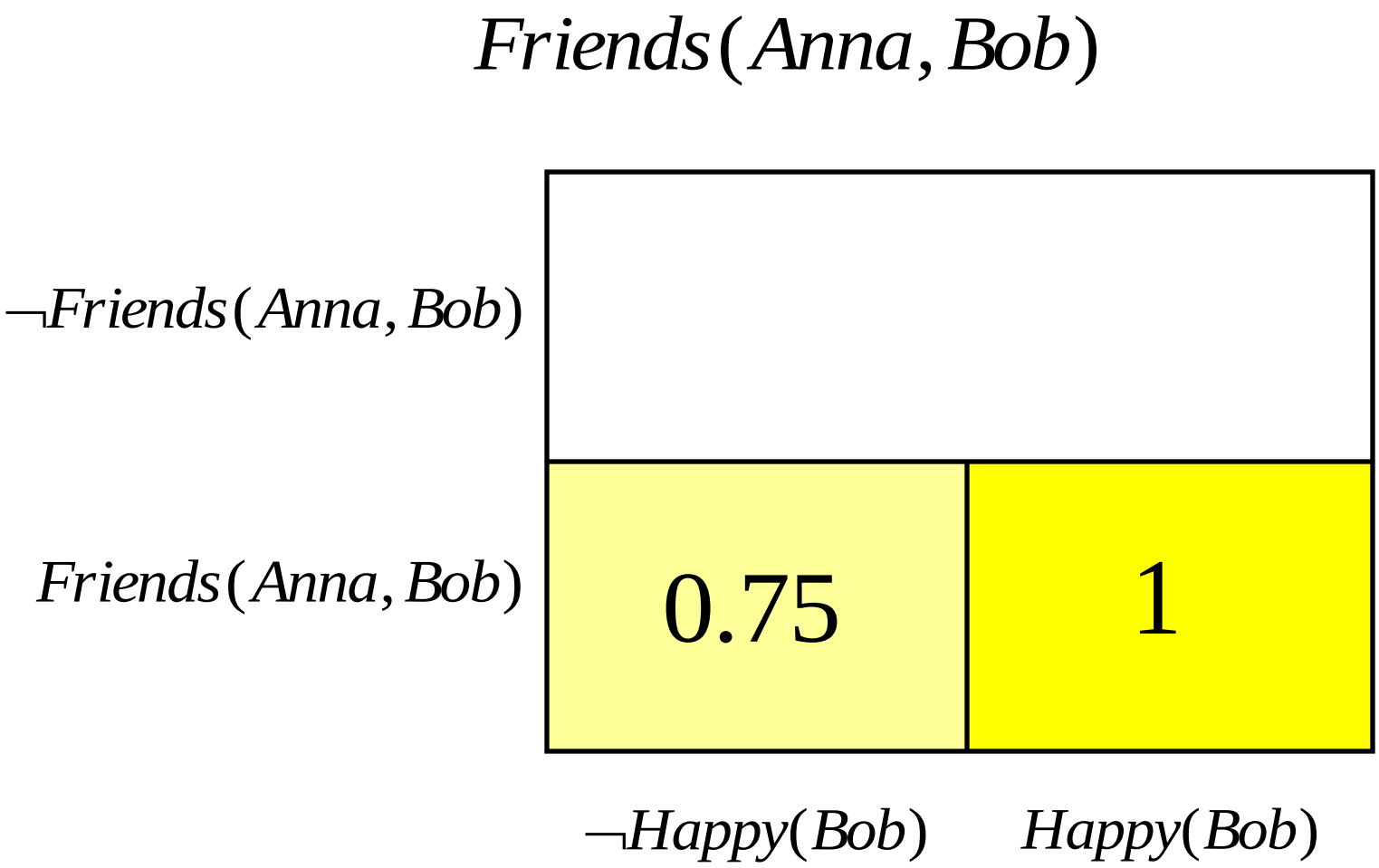
Example

$$\Phi(\neg Friends(Anna, Bob) \vee Happy(Bob)) = 1$$

$$\Phi(Friends(Anna, Bob) \wedge \neg Happy(Bob)) = 0.75$$

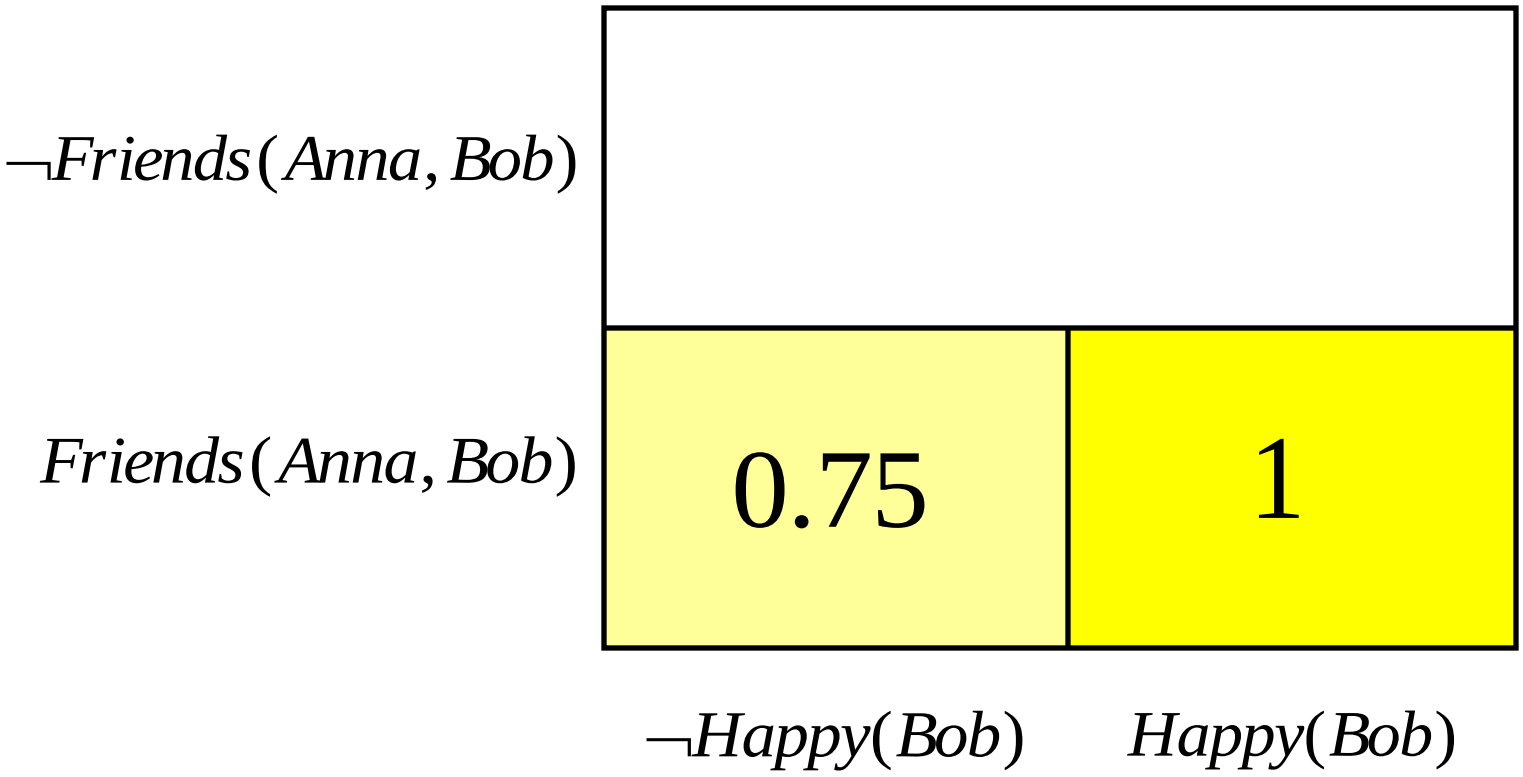


Example


Friends(Anna, Bob)
¬Friends(Anna, Bob)
Friends(Anna, Bob)
0.75
1
¬Happy(Bob)
Happy(Bob)

Example

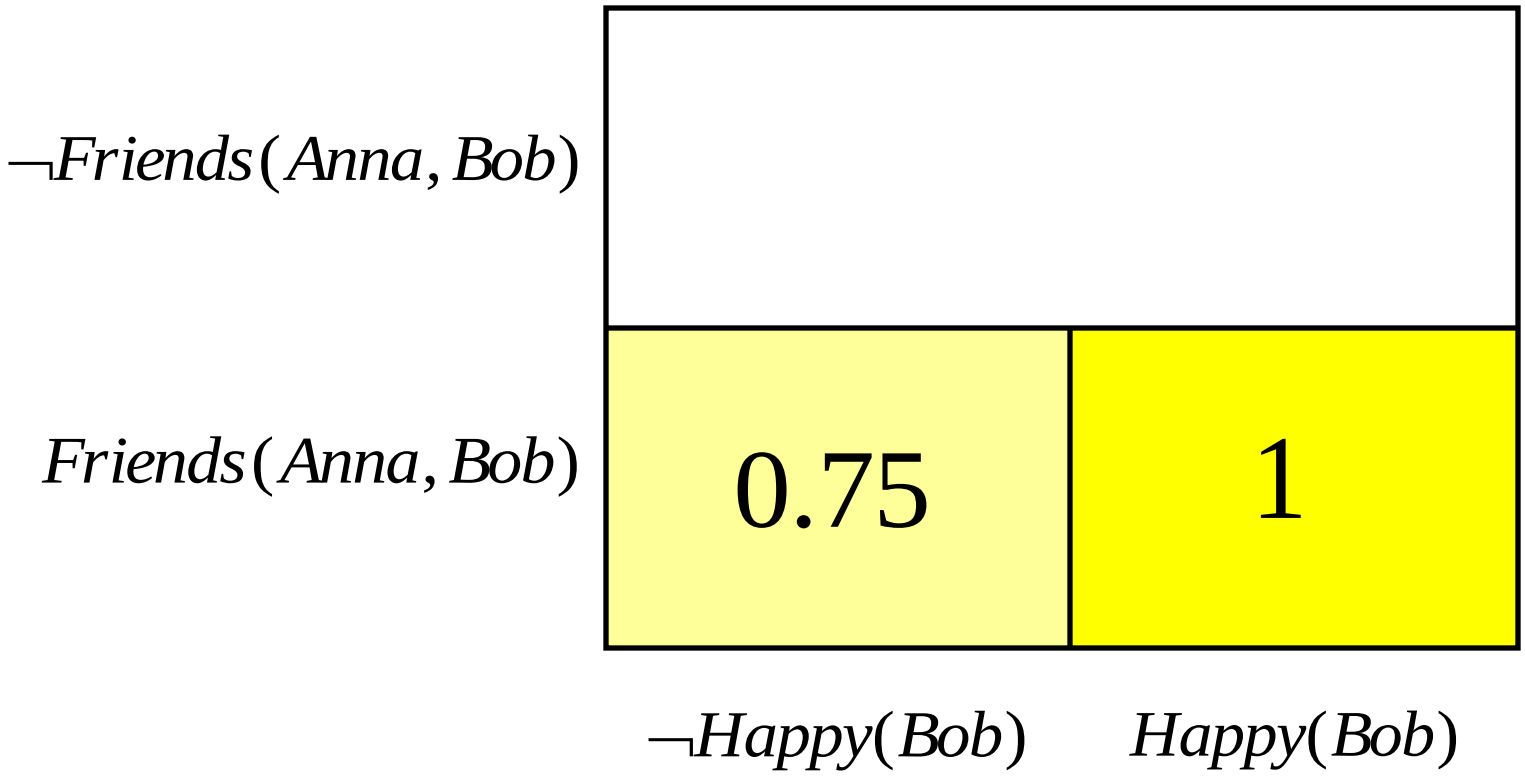
$$P(Happy(Bob) \mid Friends(Anna, Bob)) = \frac{1}{1+0.75} \approx 0.57$$



Example

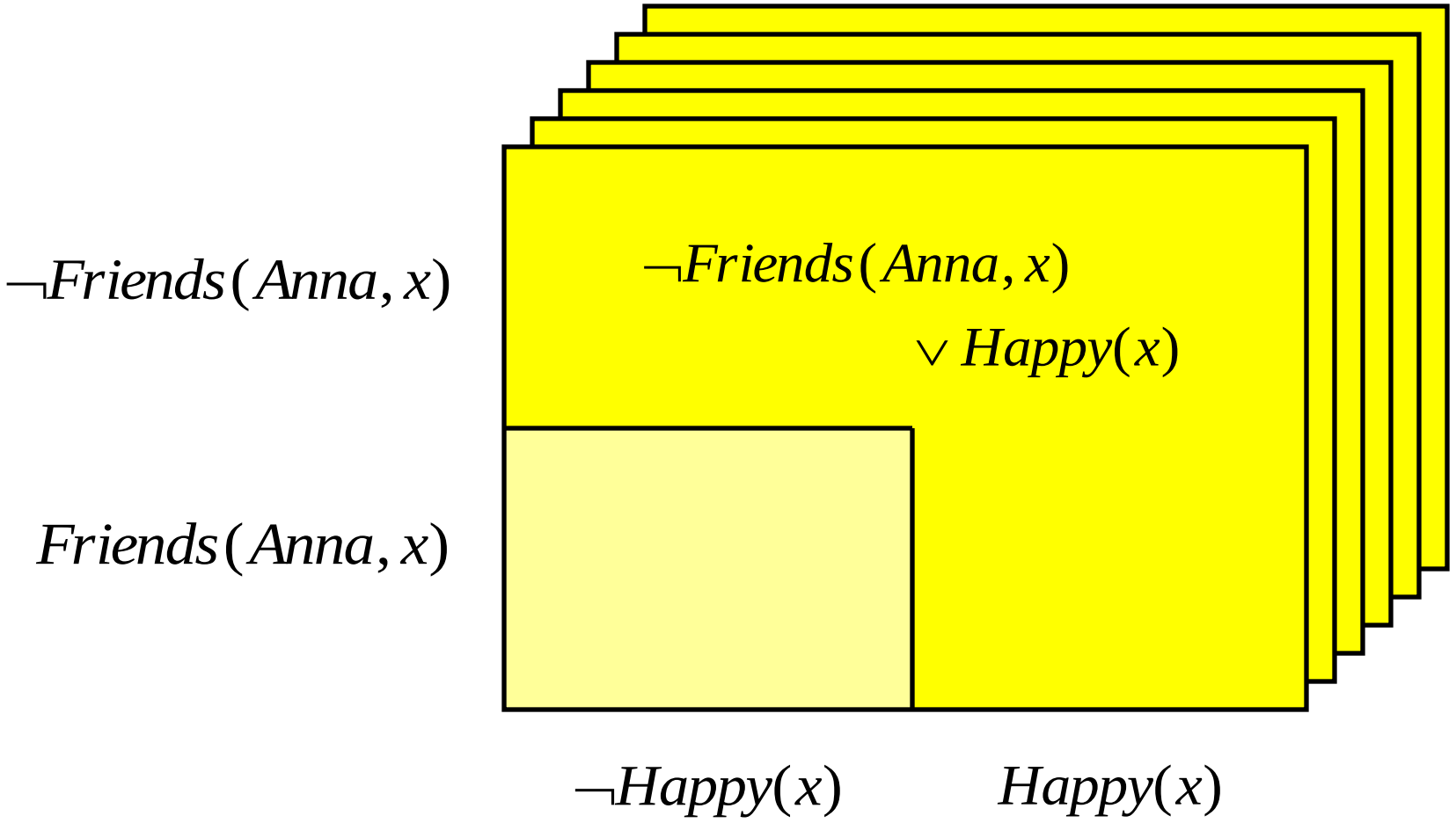
If $P(\neg Friends(Anna, Bob) \vee Happy(Bob)) = 0.8$

Then $P(Happy(Bob) \mid Friends(Anna, Bob)) = \frac{1}{1+0.75} \approx 0.57$



Example

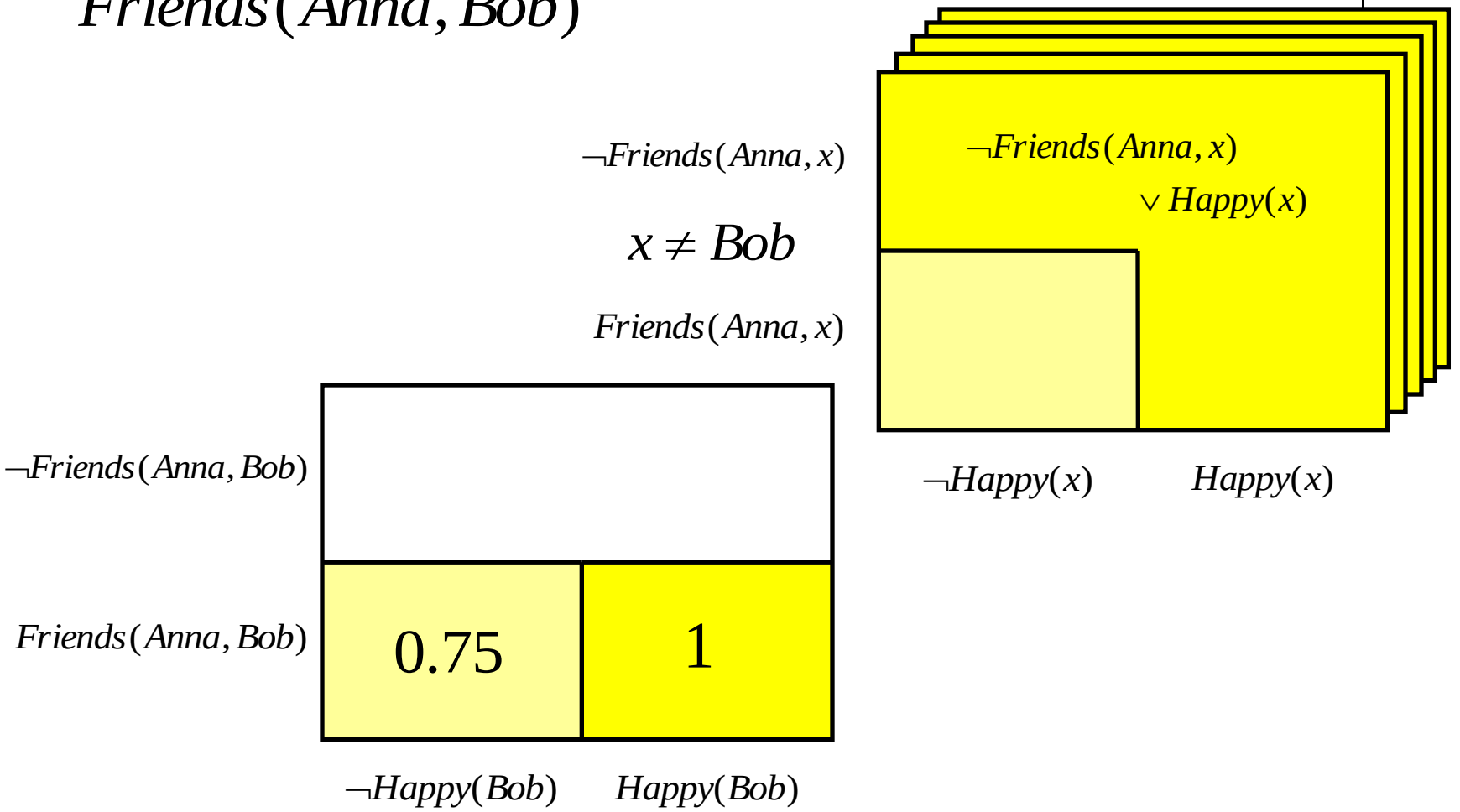
$$P(\neg Friends(Anna, x) \vee Happy(x)) = 0.8$$



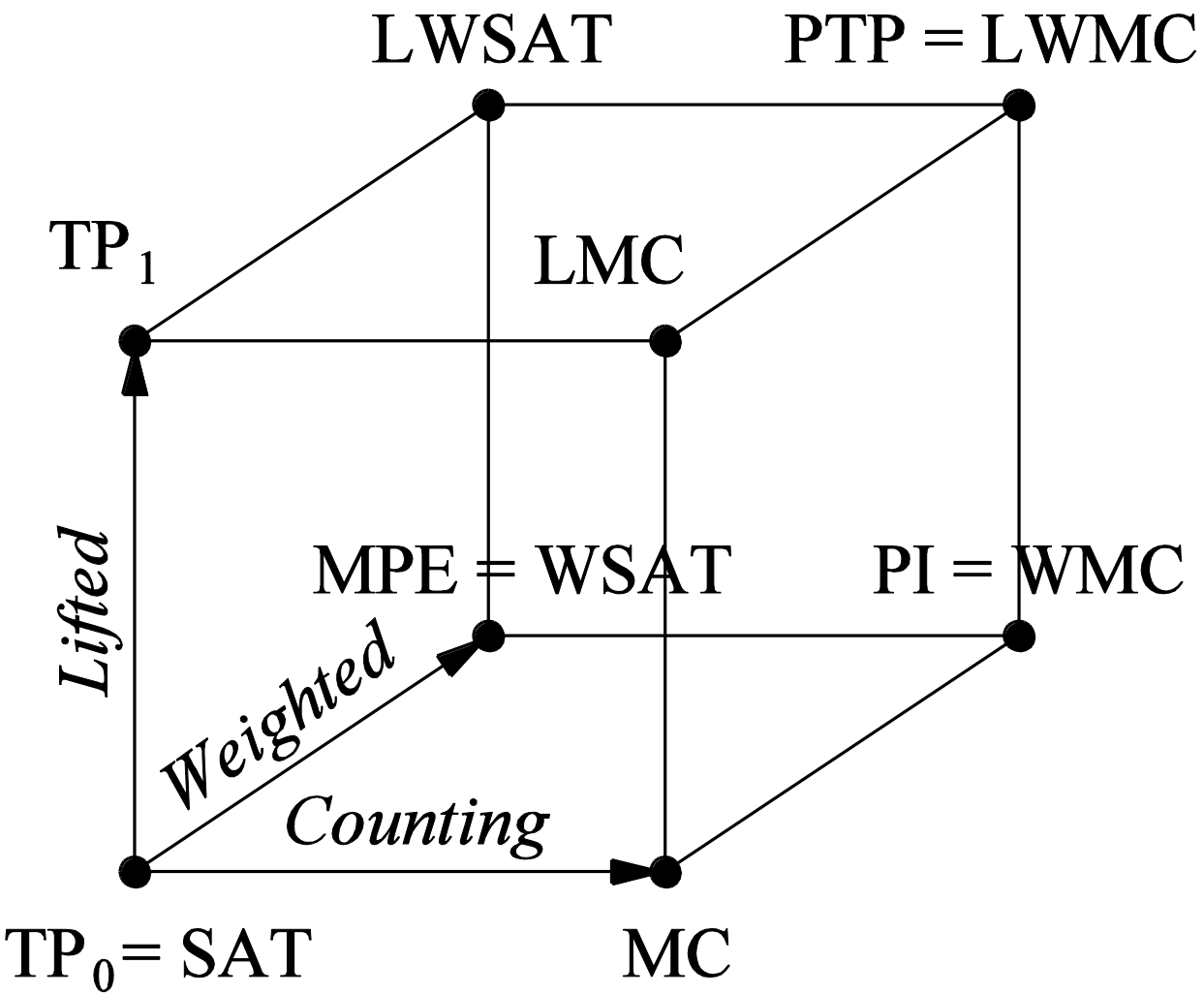
Example

$$P(\neg Friends(Anna, x) \vee Happy(x)) = 0.8$$

$$Friends(Anna, Bob)$$



Inference Problems



Propositional Case

- All conditional probabilities are ratios of partition functions:

$$P(Query \mid PKB) = \frac{\sum_{worlds} 1_{Query}(world) \prod_i \Phi_i(world)}{Z(PKB)}$$

$$= \frac{Z(PKB \cup \{(Query, 0)\})}{Z(PKB)}$$

- All partition functions can be computed by weighted model counting

Conversion to CNF + Weights



WCNF(*PKB*)

for all (F_i, Φ_i) ϵ *PKB* s.t. $\Phi_i > 0$ **do**

$PKB \leftarrow PKB \cup \{(F_i \Leftrightarrow A_i, 0)\} \setminus \{(F_i, \Phi_i)\}$

$CNF \leftarrow$ CNF(*PKB*)

for all $\neg A_i$ literals do $W_{\neg Ai} \leftarrow \Phi_i$

for all other literals *L* do $w_L \leftarrow 1$

return (*CNF*, *weights*)

Probabilistic Theorem Proving



PTP(*PKB*, *Query*)

$PKB_Q \leftarrow PKB \cup \{(Query, 0)\}$

return WMC(WCNF(PKB_Q))
/ WMC(WCNF(*PKB*))

Probabilistic Theorem Proving



PTP(*PKB*, *Query*)

$PKB_Q \leftarrow PKB \cup \{(Query, 0)\}$

return WMC(WCNF(PKB_Q)) / WMC(WCNF(*PKB*))

Compare:

TP(*KB*, *Query*)

$KB_Q \leftarrow KB \cup \{\neg Query\}$

return ¬SAT(CNF(KB_Q))

Weighted Model Counting

WMC(*CNF*, *weights*)

if all clauses in *CNF* are satisfied

return $\prod_{A \in \mathrm{A}(CNF)} (w_A + w_{\neg A})$

Base Case

if *CNF* has empty unsatisfied clause **return** 0

Weighted Model Counting

WMC(*CNF*, *weights*)

if all clauses in CNF are satisfied

return $\prod_{A \in \mathrm{A}(CNF)} (w_A + w_{\neg A})$

if *CNF* has empty unsatisfied clause **return** 0

if *CNF* can be partitioned into CNFs $C_1, \ldots, C_k$ sharing no atoms

return $\prod_{i=1}^{k} WMC(C_i, weights)$

Decomp. Step

Weighted Model Counting

WMC(*CNF*, *weights*)

if all clauses in *CNF* are satisfied

return $\prod_{A \in \mathrm{A}(CNF)} (w_A + w_{\neg A})$

if *CNF* has empty unsatisfied clause **return** 0

if *CNF* can be partitioned into CNFs $C_1, \ldots, C_k$ sharing no atoms

return $\prod_{i=1}^{k} WMC(C_i, weights)$

choose an atom *A*

return $w_A \, WMC(CNF \mid A, weights) + w_{\neg A} \, WMC(CNF \mid \neg A, weights)$

Splitting Step

First-Order Case



- PTP schema remains the same
- Conversion of PKB to hard CNF and weights:
  New atom in $F_i \Leftrightarrow A_i$ is now
  $Predicate_i(variables\ in\ F_i,\ constants\ in\ F_i)$
- New argument in WMC:
  Set of substitution constraints of the form
  $x = A,\ x \neq A,\ x = y,\ x \neq y$
- Lift each step of WMC

Lifted Weighted Model Counting

LWMC(*CNF, substs, weights*)

if all clauses in *CNF* are satisfied

return $\prod_{A \in \mathrm{A}(CNF)} (w_A + w_{\neg A})^{n_A(substs)}$

if *CNF* has empty unsatisfied clause **return** 0

Base Case

Lifted Weighted Model Counting

LWMC(*CNF, substs, weights*)

if all clauses in *CNF* are satisfied

return $\prod_{A \in \mathrm{A}(CNF)} (w_A + w_{\neg A})^{n_A(substs)}$

if *CNF* has empty unsatisfied clause **return** 0

if there exists a lifted decomposition of *CNF*

return $\prod_{i=1}^{k} [LWMC(CNF_{i,1}, substs, weights)]^{m_i}$

Decomp. Step

Lifted Weighted Model Counting

LWMC(*CNF, substs, weights*)

if all clauses in *CNF* are satisfied

return $\prod_{A \in \mathrm{A}(CNF)} (w_A + w_{\neg A})^{n_A(substs)}$

if *CNF* has empty unsatisfied clause **return** 0

if there exists a lifted decomposition of *CNF*

return $\prod_{i=1}^{k} [LWMC(CNF_{i,1}, substs, weights)]^{m_i}$

choose an atom *A*

return

$\sum_{i=1}^{l} n_i w_A^{t_i} w_{\neg A}^{f_i} LWMC(CNF \mid \sigma_j, substs_j, weights)$

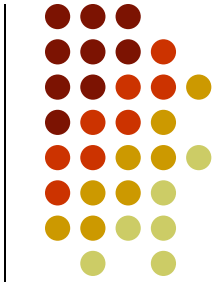
Splitting Step

Extensions



- Unit propagation, etc.
- Caching / Memoization
- Knowledge-based model construction

Formal properties



- PTP can be exponentially more efficient than FOVE (first-order variable elimination)
- PTP has the same worst-case time and space complexity as FOVE

Approximate Inference

WMC(*CNF*, *weights*)

if all clauses in *CNF* are satisfied

return $\prod_{A \in \mathrm{A}(CNF)} (w_A + w_{\neg A})$

if *CNF* has empty unsatisfied clause **return** 0

if *CNF* can be partitioned into CNFs $C_1, \ldots, C_k$ sharing no atoms

return $\prod_{i=1}^{k} WMC(C_i, weights)$

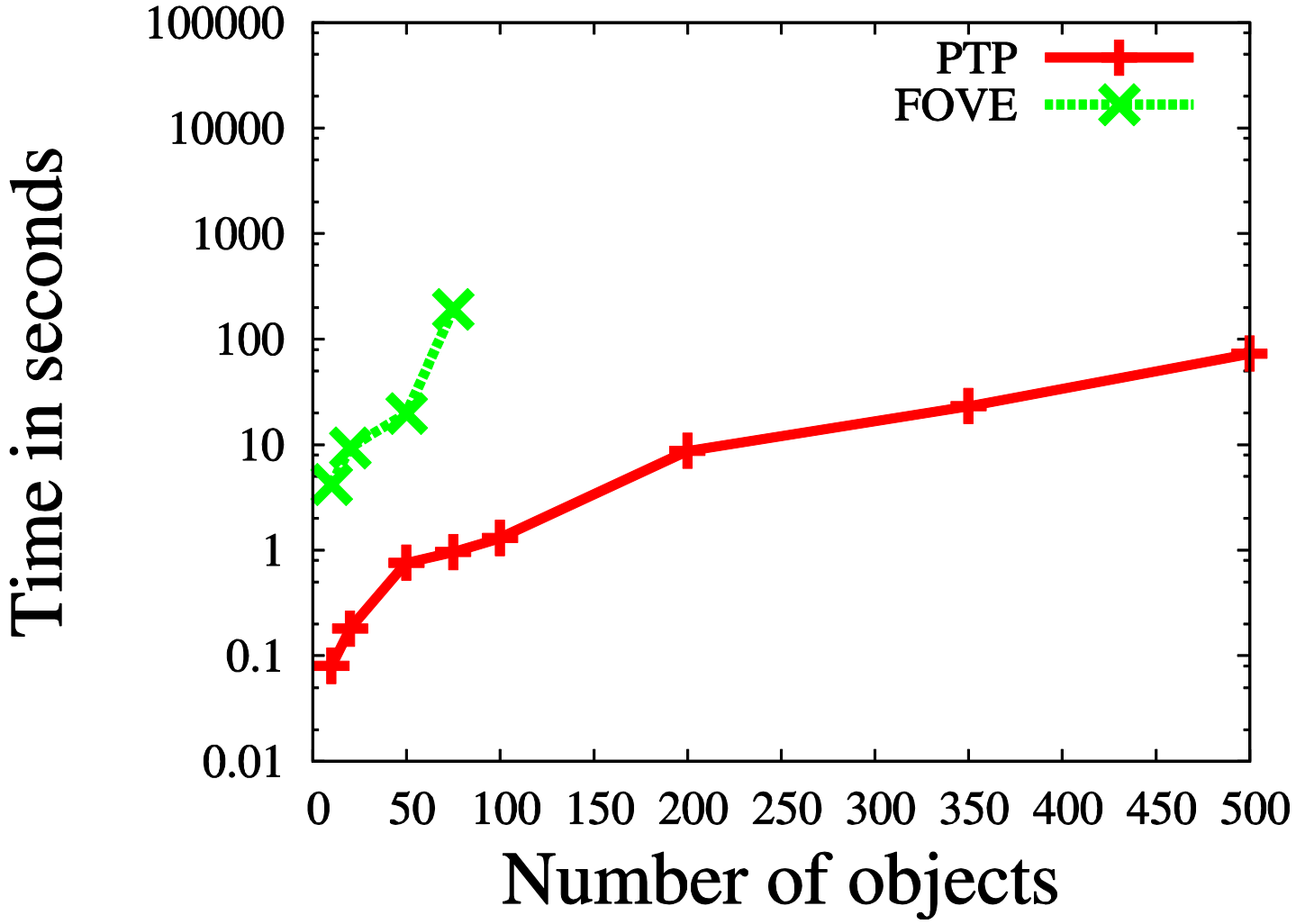
Splitting Step

choose an atom *A*

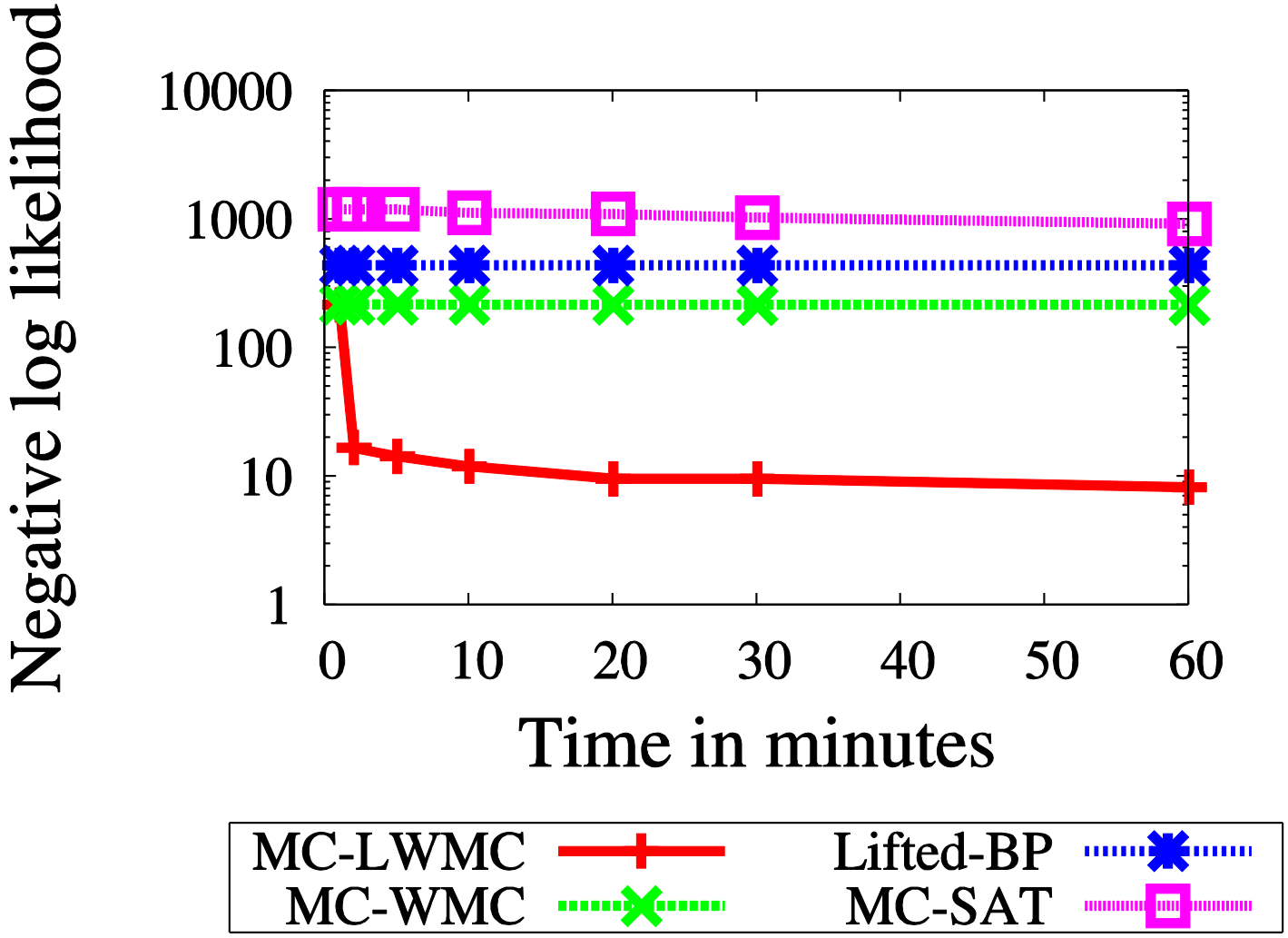
return $\frac{w_A}{Q(A \mid CNF, weights)} WMC(CNF \mid A, weights)$

with probability $Q(A \mid CNF, weights)$, etc.

Link Prediction



Coreference (Cora)



Conclusion



- Probabilistic theorem proving is the most powerful lifted inference approach to date
- PTP is reducible to lifted weighted model counting
- A single algorithm solves a lot of problems
- Empirically very efficient
- Available soon in Alchemy system **(alchemy.cs.washington.edu)**

Limitations and potential of lifted probabilistic inference

David Poole

Department of Computer Science,
University of British Columbia

IJCAI 2011 Tutorial

Overview

- Lifted Variable Elimination versus Lifted Search
- Limitations of Lifted Inference
- Number Uncertainty
- Richer Languages
- Identity Uncertainty
- Existence Uncertainty

Assumptions behind current lifted algorithms

- Known population size
- Conditioning and querying on conjunctions of ground atoms
- Unique names assumption: we can identify when individuals are the same. There is no identity uncertainty.
- No querying about equality
- No existence uncertainty

Outline

1. Lifted VE vs Search
2. Unknown Number of Individuals

Problems with Lifted VE

- Intermediate representations (parfactors, counting formulae...) are not closed under lifted operations.
  - We need to ground sometimes
  - We need better intermediate representations
  - We need alternatives to lifted VE.

Lifted Search

- Variable elimination is the dynamic programming variant of recursive conditioning (and related search methods).
- VE creates intermediate representations.
- Search just evaluates factors when fully instantiated.
- In search, everything is evaluated in a particular context — perhaps we don't need complex intermediate representations that need to anticipate all eventualities.

Limitations to Lifted Inference

Jaeger [AIJ 2000] show that

- If the query/observation language includes first-order logic with equality
- then there are queries that are not polynomial in population size.

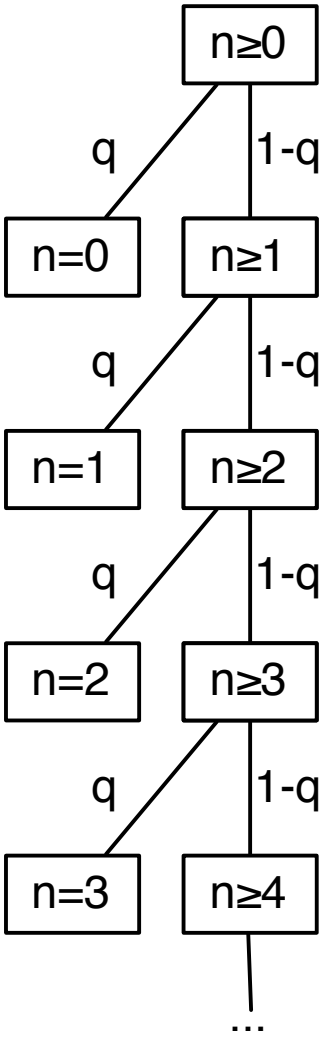
Outline

1. Lifted VE vs Search
2. Unknown Number of Individuals

What if we don't know the population size?

- The previous methods assumed that we know how many individuals there are.
- What if we don't know the population size, but only have a distribution over population size?

Geometric population size



If n is distributed according to a geometric distribution with $\forall k\ q = P(n = k|n \geq k)$, the expected value of p^n is

$$\frac{q}{1 - p(1 - q)}$$

Proof:

$$\mathcal{E}_n(p^n) = \sum_{n=0}^{\infty} q(1-q)^n p^n$$

$$= q \sum_{n=0}^{\infty} (p(1-q))^n$$

$$= \frac{q}{1 - p(1 - q)}$$

Poisson population size

If n is distributed according to a Poisson distribution $f(n, \lambda) = \frac{\lambda^n e^{-\lambda}}{n!}$ (λ is the expected number of individuals) the expected value of p^n is

$$e^{-\lambda(1-p)}.$$

Proof:

$$\begin{aligned}
\mathcal{E}_n(p^n) &= \sum_{n=0}^{\infty} \frac{\lambda^n e^{-\lambda}}{n!} p^n \\
&= e^{-\lambda} e^{\lambda p} \sum_{n=0}^{\infty} \frac{(\lambda p)^n e^{-\lambda p}}{n!} \\
&= e^{-\lambda(1-p)}
\end{aligned}$$

Counting for unknown population

- Open problem: Is there an analog of counting formulae for unknown population?
- Can we do lifted inference after finding some evidence about the number of objects?