Lifted Probabilistic Inference in Relational Models

Guy Van den Broeck
UCLA

Dan Suciu
U. of Washington

IJCAI Tutorial
July 10, 2016

About the Tutorial

Slides available at
http://web.cs.ucla.edu/~guyvdb/talks/IJCAI16-tutorial/
Extensive bibliography at the end.
Your speakers:

http://web.cs.ucla.edu/~guyvdb/

https://homes.cs.washington.edu/~suciu/

I work in AI

I work in DB

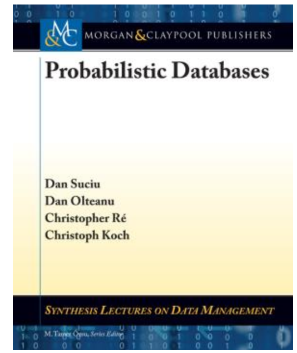
About the Tutorial

- The tutorial is about
  - deep connections between AI and DBs
  - a unified view on probabilistic reasoning
  - a logical approach to prob. reasoning

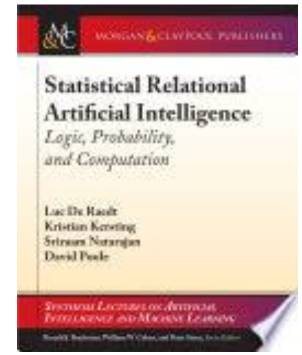
- The tutorial is NOT an exhaustive overview of lifted algorithms for graphical models (see references at the end)

If you want more…

- Books
  - Probabilistic Databases
  - Statistical Relational AI
  - (Lifted Inference Book)



[Suciu'11]



[DeRaedt'16]

- StarAI workshop on Monday
  http://www.starai.org

- Main conference papers

Outline

- Part 1: Motivation
- Part 2: Probabilistic Databases
- Part 3: Weighted Model Counting
- Part 4: Lifted Inference for WFOMC
- Part 5: Completeness of Lifted Inference
- Part 6: Query Compilation
- Part 7: Symmetric Lifted Inference Complexity
- Part 8: Open-World Probabilistic Databases
- Part 9: Discussion & Conclusions

Outline

- Part 1: Motivation
- Part 2: Probabilistic Databases
- Part 3: Weighted Model Counting
- Part 4: Lifted Inference for WFOMC

- Part 5: Completeness of Lifted Inference
- Part 6: Query Compilation
- Part 7: Symmetric Lifted Inference Complexity
- Part 8: Open-World Probabilistic Databases
- Part 9: Discussion & Conclusions

Part 1: Motivation

- Why do we need relational representations of uncertainty?
- Why do we need probabilistic queries?
- Why do we need lifted inference algorithms?

Why Relational Data?

- Our data is already relational!
  - Companies run relational databases
  - Scientific data is relational:
    - Large Hadron Collider generated 25PB in 2012
    - LSST Telescope will produce 30TB per night

- Big data is big business:
  - Oracle: $7.1BN in sales
  - IBM: $3.2BN in sales
  - Microsoft: $2.6BN in sales

[Gartner'06]

Why Probabilistic Relational Data?

- Relational data is increasingly probabilistic
  - NELL machine reading (>50M tuples)
  - Google Knowledge Vault (>2BN tuples)
  - DeepDive (>7M tuples)

- Data is inferred from unstructured information using statistical models
  - Learned from the web, large text corpora, ontologies, etc.
  - The learned/extracted data is relational

[Carlson'10, Dong'14, Niu'12]

Information Extraction

PhD Students Luc De Raedt

- Laura-Andrea Antanas(co-promotor Tinne Tuytelaars)
- Dries Van Daele (co-promotor Kathleen Marchal)
- Thanh Le Van (co-promotor Kathleen Marchal)
- Bogdan Moldovan
- Davide Nitti (co-promotor Tinne De Laet)
- José Antonio Oramas Mogrovejo (key supervisor Tinne Tuytelaars)
- Francesco Orsini (co-supervisor Paol Frasconi)
- Sergey Paramonov
- Joris Renkens
- Mathias Verbeke (with Bettina Berendt)
- Jonas Vlasselaer

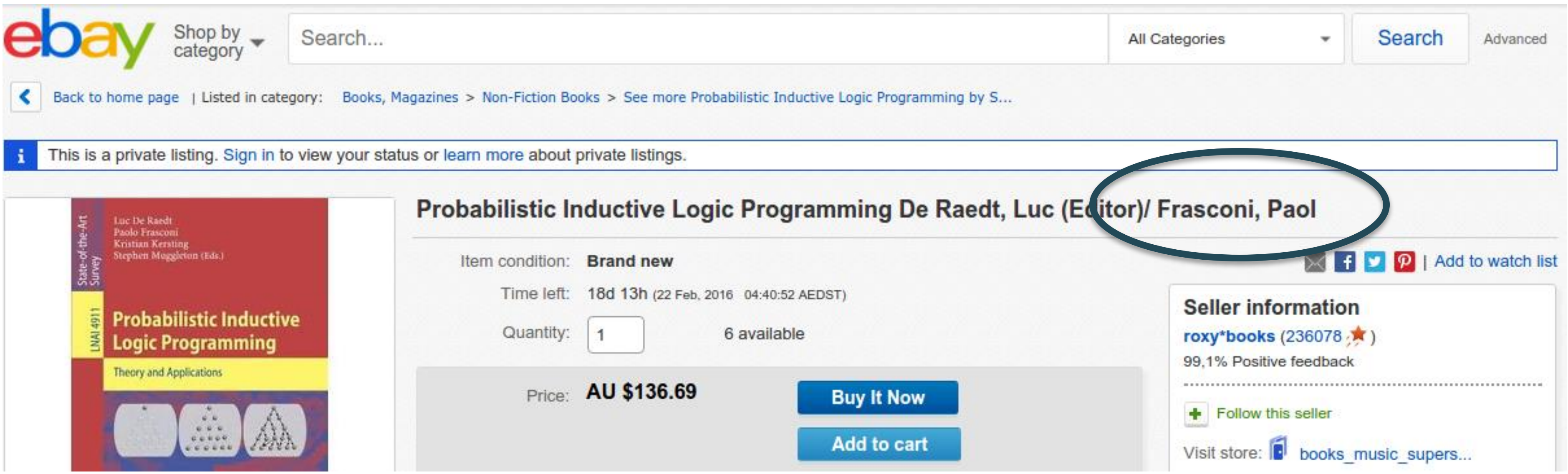
Alumni Luc De Raedt

1. Hendrik Blockeel, *Top-down induction of first order logical decision trees*, Ph.D. thesis, Department of Computer Science, K.U.Leuven, Leuven, Belgium, december 1998, 202+xv pages. (Co-promotor Maurice Bruynooghe)
2. Luc Dehaspe, *Frequent pattern discovery in first-order logic*, Ph.D. thesis, Department of Computer

PublishedWith

X	Y	P
Luc	Laura	0.7
Luc	Hendrik	0.6
Luc	Kathleen	0.3
Luc	Paol	0.3
Luc	Paolo	0.1

Extraction is so Noisy!



Representation: Probabilistic Databases

- Tuple-independent probabilistic databases

Actor

Name	Prob
Brando	0.9
Cruise	0.8
Coppola	0.1

WorkedFor

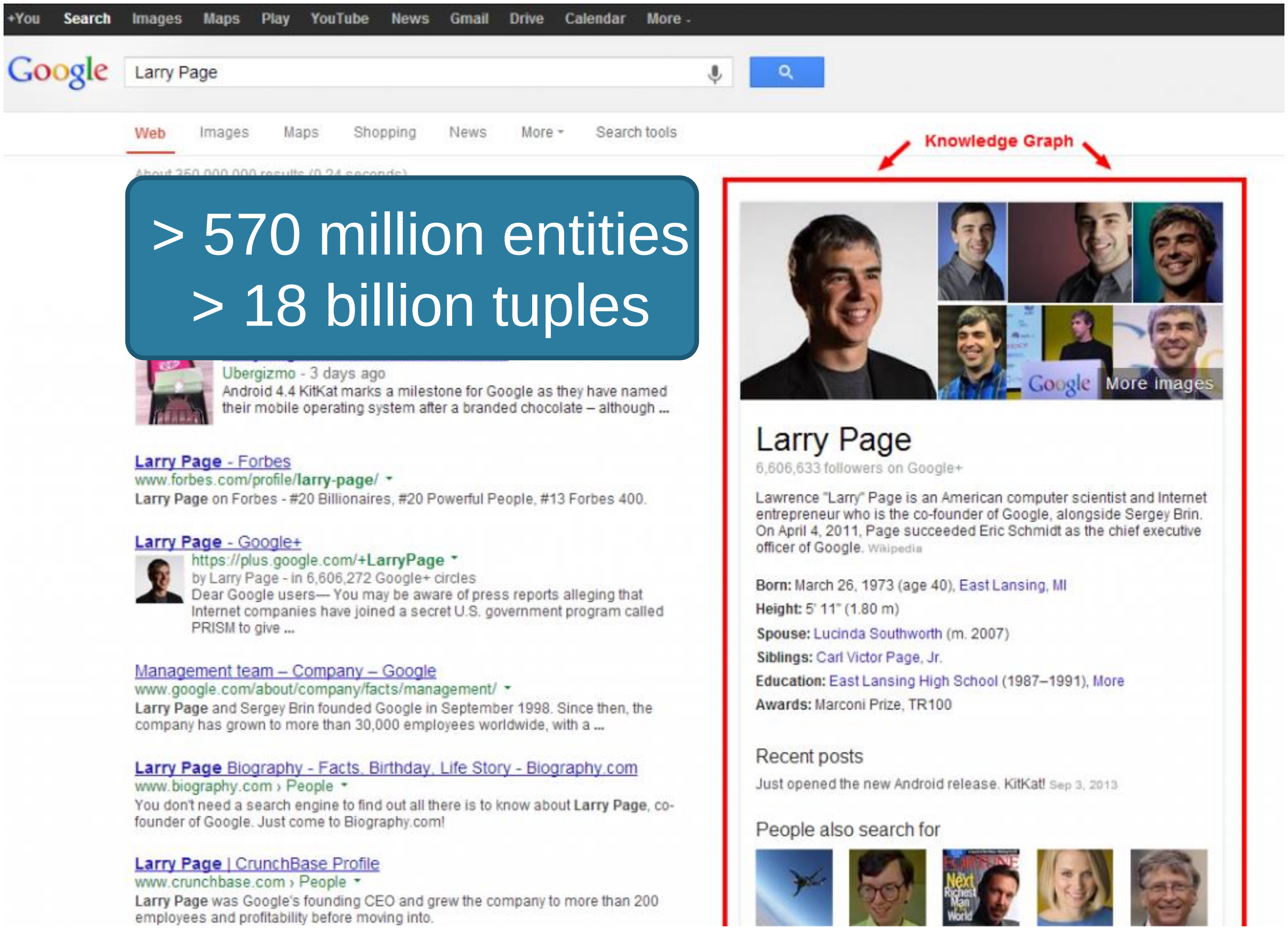
Actor	Director	Prob
Brando	Coppola	0.9
Coppola	Brando	0.2
Cruise	Coppola	0.1

- Query: SQL or First-order logic

```
SELECT Actor.name
FROM Actor, WorkedFor
WHERE Actor.name = WorkedFor.actor
```

$Q(x) = \exists y\ Actor(x) \land WorkedFor(x,y)$

Why Probabilistic Queries?



> 570 million entities
> 18 billion tuples

What we'd like to do…

Has anyone published a paper with both Erdos and Einstein

All News Images Videos Shopping More ▾ Search tools

About 82,400 results (0.73 seconds)

Erdős number - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/**Erdős**_number ▾ Wikipedia ▾
He **published** more **papers** during his lifetime (at least 1,525) than any other ...
Anybody else's **Erdős** number is k + 1 where k is the lowest **Erdős** number of any coauthor. ... Albert **Einstein** and Sheldon Lee Glashow **have** an **Erdős** number of 2. ... and mathematician Ruth Williams, **both** of whom **have** an **Erdős** number of 2.

Erdős–Bacon number - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/**Erdős**–Bacon_number ▾ Wikipedia ▾
This article possibly **contains** previously unpublished synthesis of **published** ... Her **paper** gives her an **Erdős** number of 4, and a Bacon number of 2, **both** of ...

Erdős is in the Knowledge Graph

Paul Erdos

All Images Videos Books News More ▾ Search tools

About 333,000 results (0.35 seconds)

Paul Erdős - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/**Paul_Erdős** ▾ Wikipedia ▾
Paul Erdős was a Hungarian Jewish mathematician. He was one of the most prolific mathematicians of the 20th century. He was known both for his social ...
Fan Chung - Ronald Graham - Béla Bollobás - Category:Paul Erdős

The Man Who Loved Only Numbers - The New York Times
https://www.nytimes.com/books/.../hoffman-man.ht... ▾ The New York Times ▾
Paul Erdös was one of those very special geniuses, the kind who comes along only once in a very long while yet he chose, quite consciously I am sure, to share ...

Paul Erdos | Hungarian mathematician | Britannica.com
www.britannica.com/biography/**Paul-Erdos** ▾ Encyclopaedia Britannica ▾
Paul Erdős, (born March 26, 1913, Budapest, Hungary—died September 20, 1996, Warsaw, Poland), Hungarian “freelance” mathematician (known for his work ...

Paul Erdős - University of St Andrews
www-groups.dcs.st-and.ac.uk/~history/Biographies/**Erdos**.html ▾
Paul Erdős came from a Jewish family (the original family name being Engländer) although neither of his parents observed the Jewish religion. Paul's father ...

[PDF] Paul Erdős Mathematical Genius, Human - UnTruth.org
www.untruth.org/~josh/math/**Paul**%20**Erdös**%20bio-rev2.pdf ▾
by J Hill - 2004 - Related articles

Paul Erdős

Mathematician

Paul Erdős was a Hungarian Jewish mathematician. He was one of the most prolific mathematicians of the 20th century. He was known both for his social practice of mathematics and for his eccentric lifestyle.
Wikipedia

Born: March 26, 1913, Budapest, Hungary

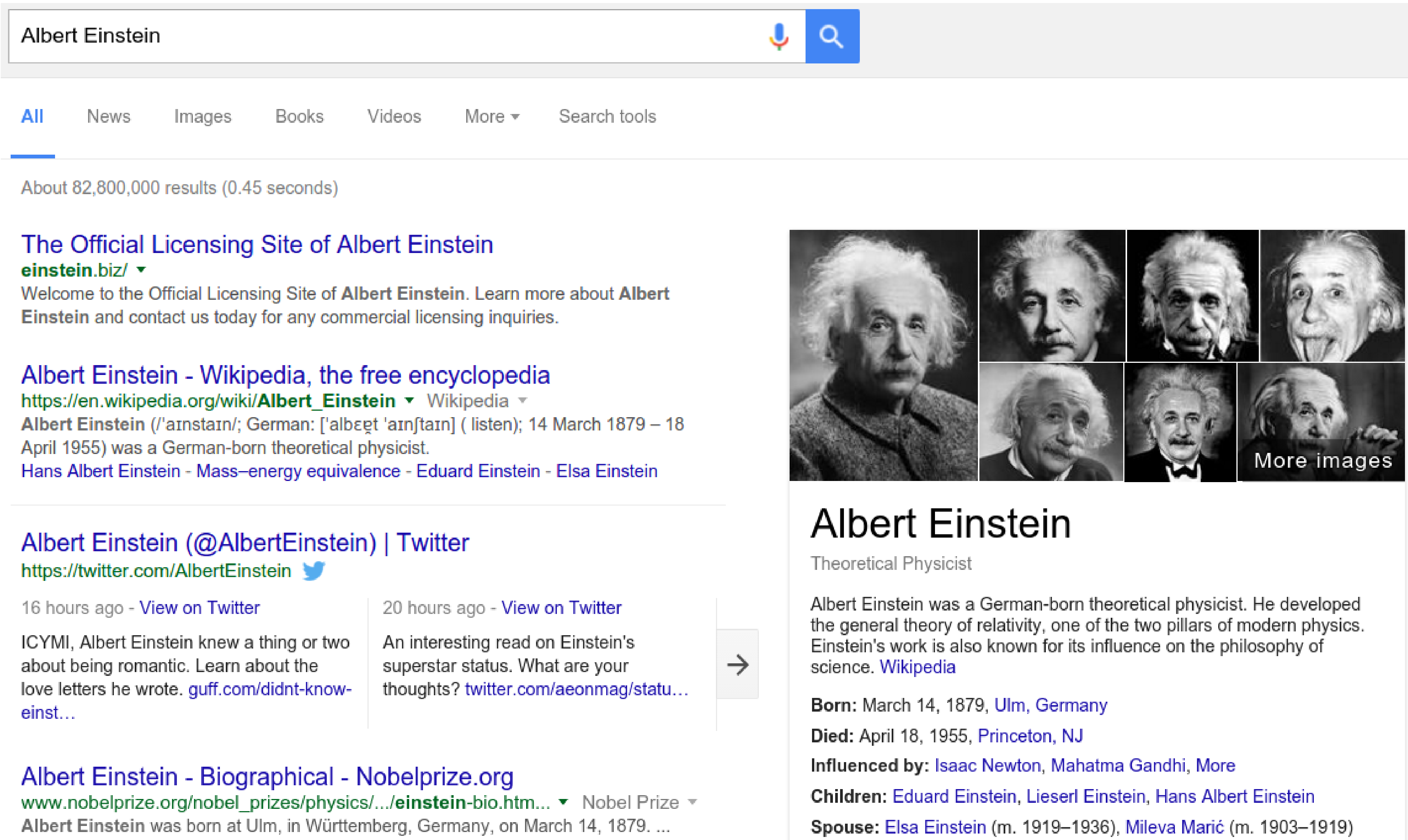
Died: September 20, 1996, Warsaw, Poland

Education: Eötvös Loránd University (1934)

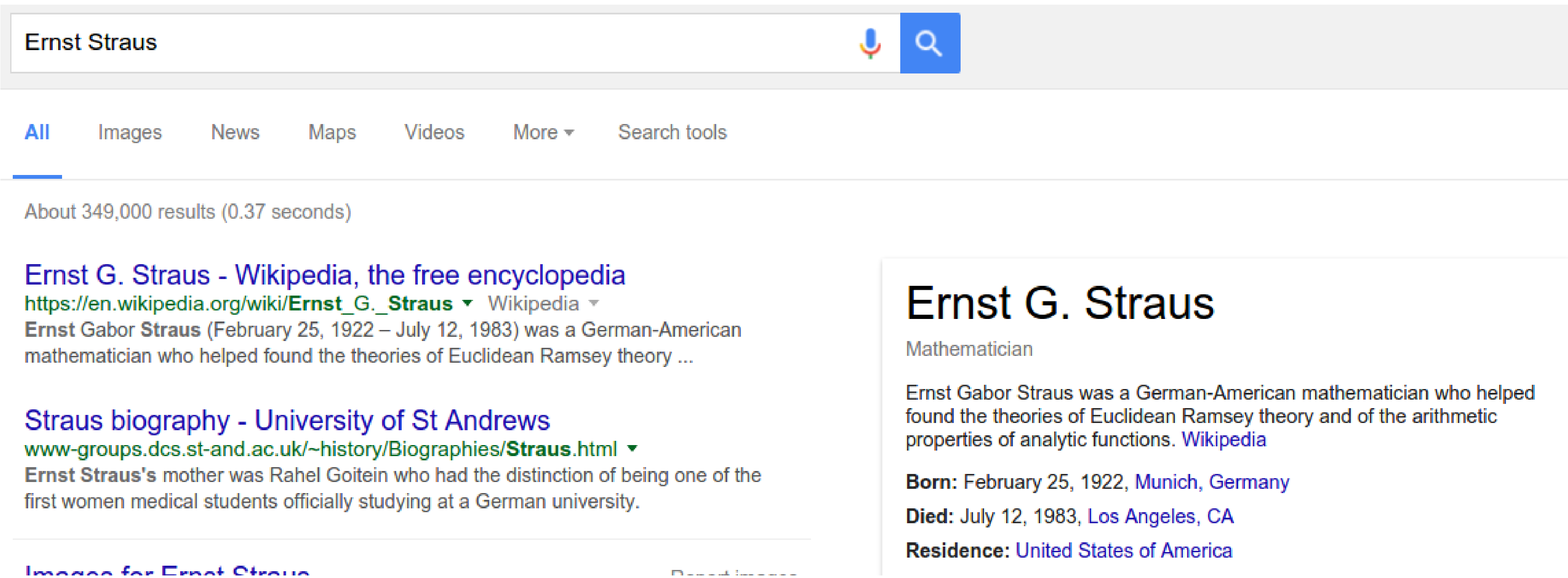
Books: Probabilistic Methods in Combinatorics, More

Notable students: Béla Bollobás, Alexander Soifer, George B. Purdy, Joseph Kruskal

Einstein is in the Knowledge Graph

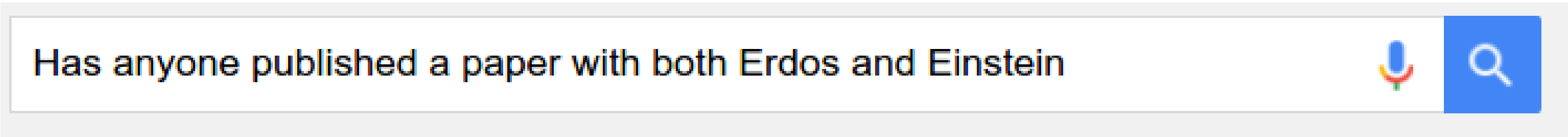


This guy is in the Knowledge Graph



… and he published with both Einstein and Erdos!

Desired Query Answer

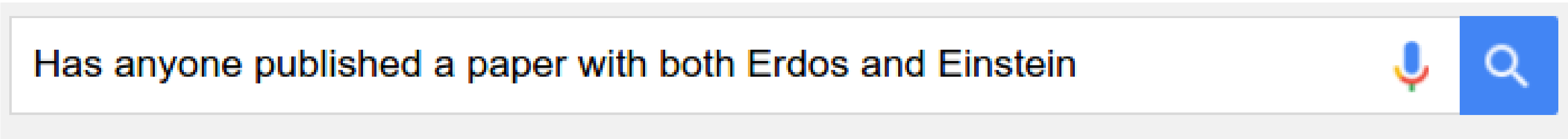


Ernst Straus

Kristian Kersting, …

Justin Bieber, …

Observations



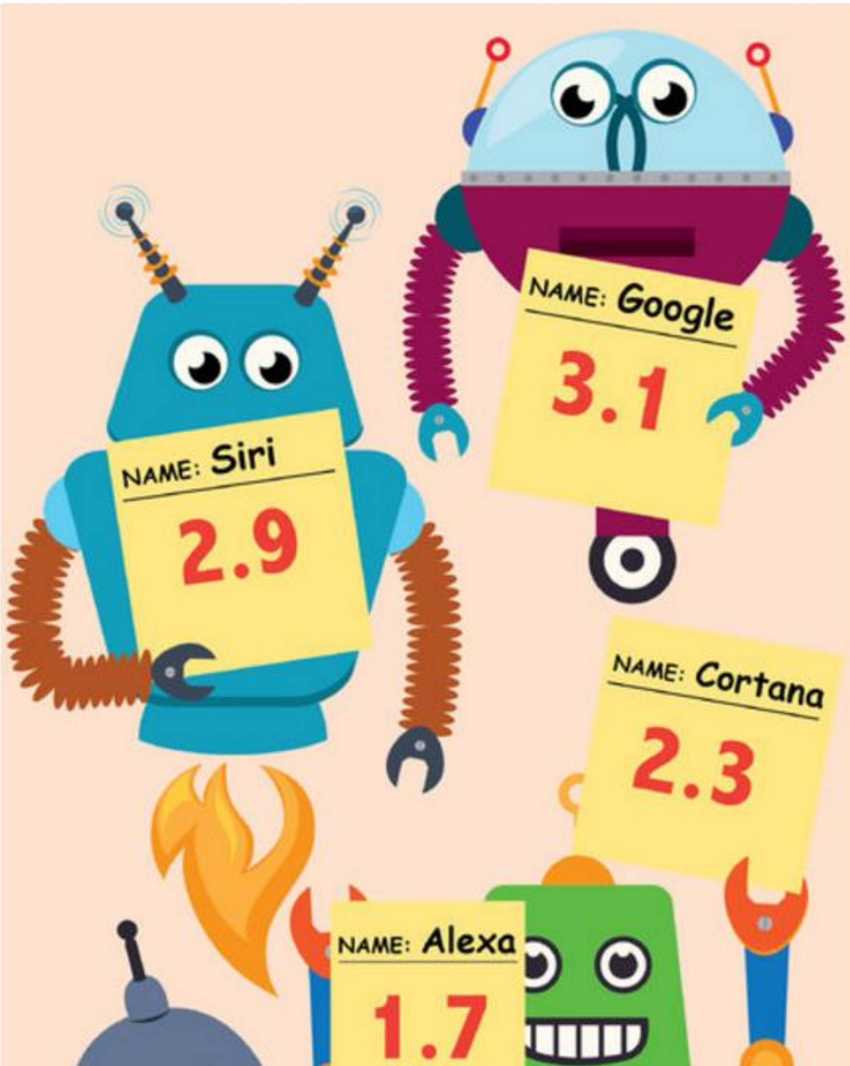
- Cannot come from labeled data
- Fuse uncertain information from many pages

- Expose uncertainty in query answers
  … and risk incorrect answers
- Embrace probability!

Siri, Alexa and Other Virtual Assistants Put to the Test

Tech Fix

By BRIAN X. CHEN JAN. 27, 2016



WHEN I asked Alexa earlier this week who was playing in the Super Bowl, she responded, somewhat monotonously, "Super Bowl 49's winner is New England Patriots."

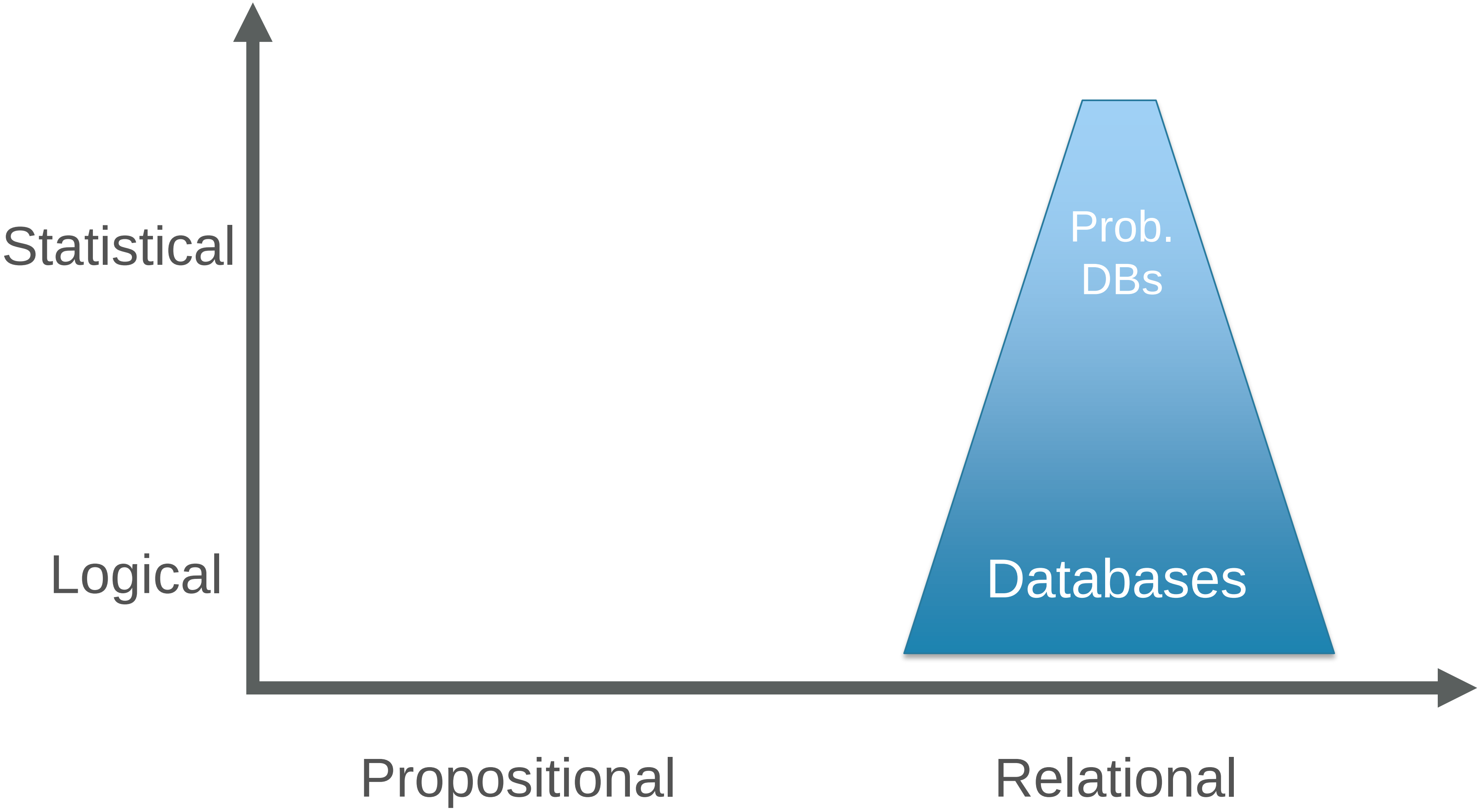
"Come on, that's last year's Super Bowl," I said. "Even I can do better than that."

At the time, I was actually alone in my living room. I was talking to the virtual companion inside Amazon's wireless speaker, Echo, which was released last June. Known as Alexa, she has gained raves from Silicon Valley's tech-obsessed digerati and has become one of the newest members of the virtual assistants club.

All the so-called Frightful Five tech

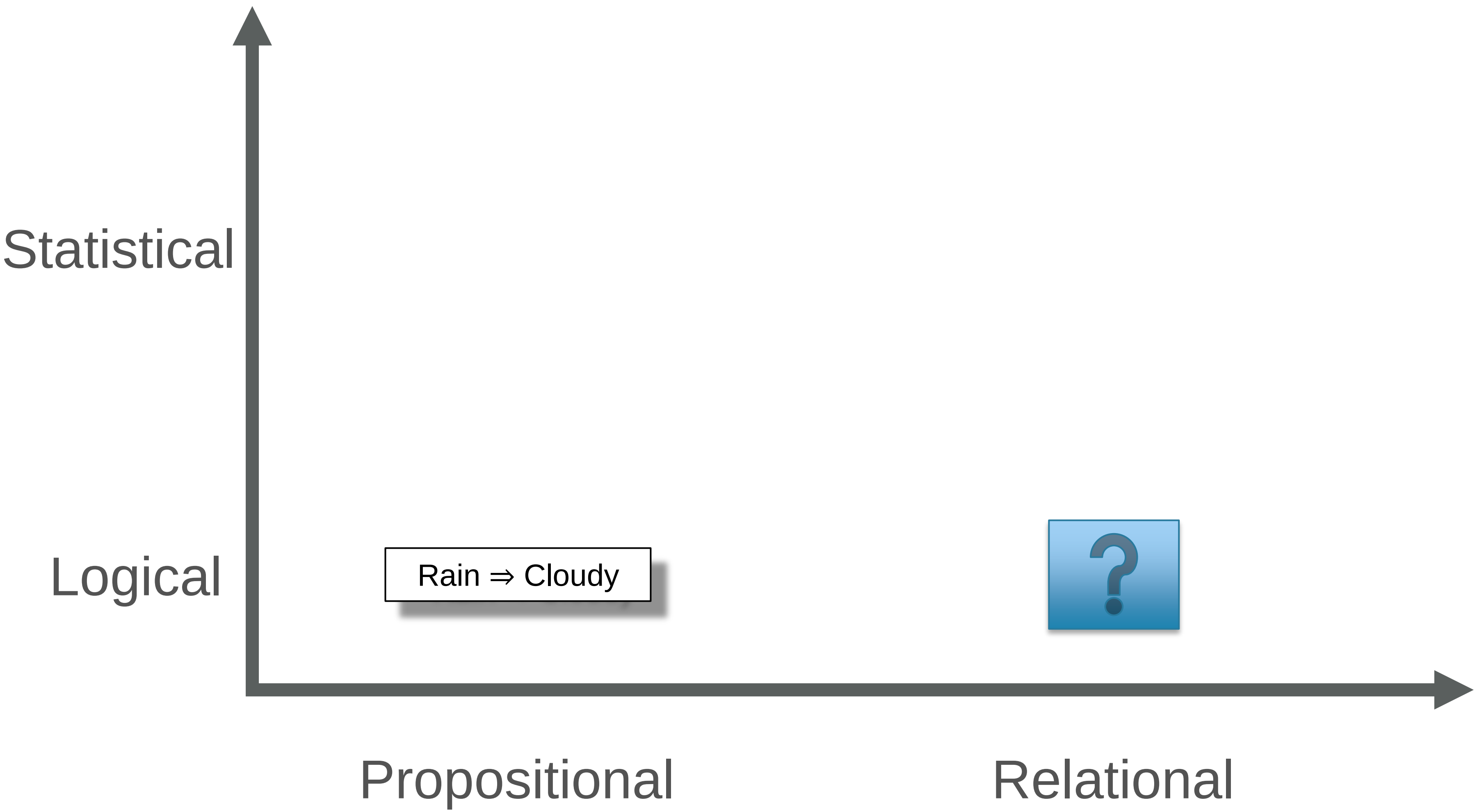
[Chen'16]
(NYTimes)

Summary



Statistical
Logical
Prob.
DBs
Databases
Propositional
Relational

Representations in AI and ML



Graphical Model Learning

Medical Records

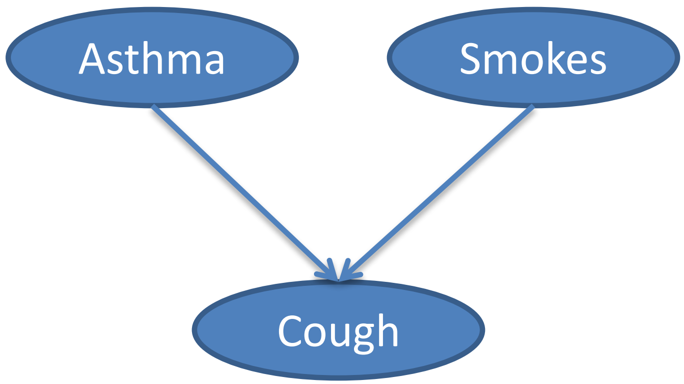
Bayesian Network

Name	Cough	Asthma	Smokes
Alice	1	1	0
Bob	0	0	0
Charlie	0	1	0
Dave	1	0	1
Eve	1	0	0

Frank	1	?	?

Big data

Frank	1	0.3	0.2



Representations in AI and ML

Statistical

Cloudy	Rain	P(Rain|Cloudy)
T	T	0.80
T	F	0.20
F	T	0.01
F	F	0.99

Logical

Rain ⇒ Cloudy

Propositional

Relational

Relational Representations

- Example: First-Order Logic

Formula

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

Atom

Logical Variable

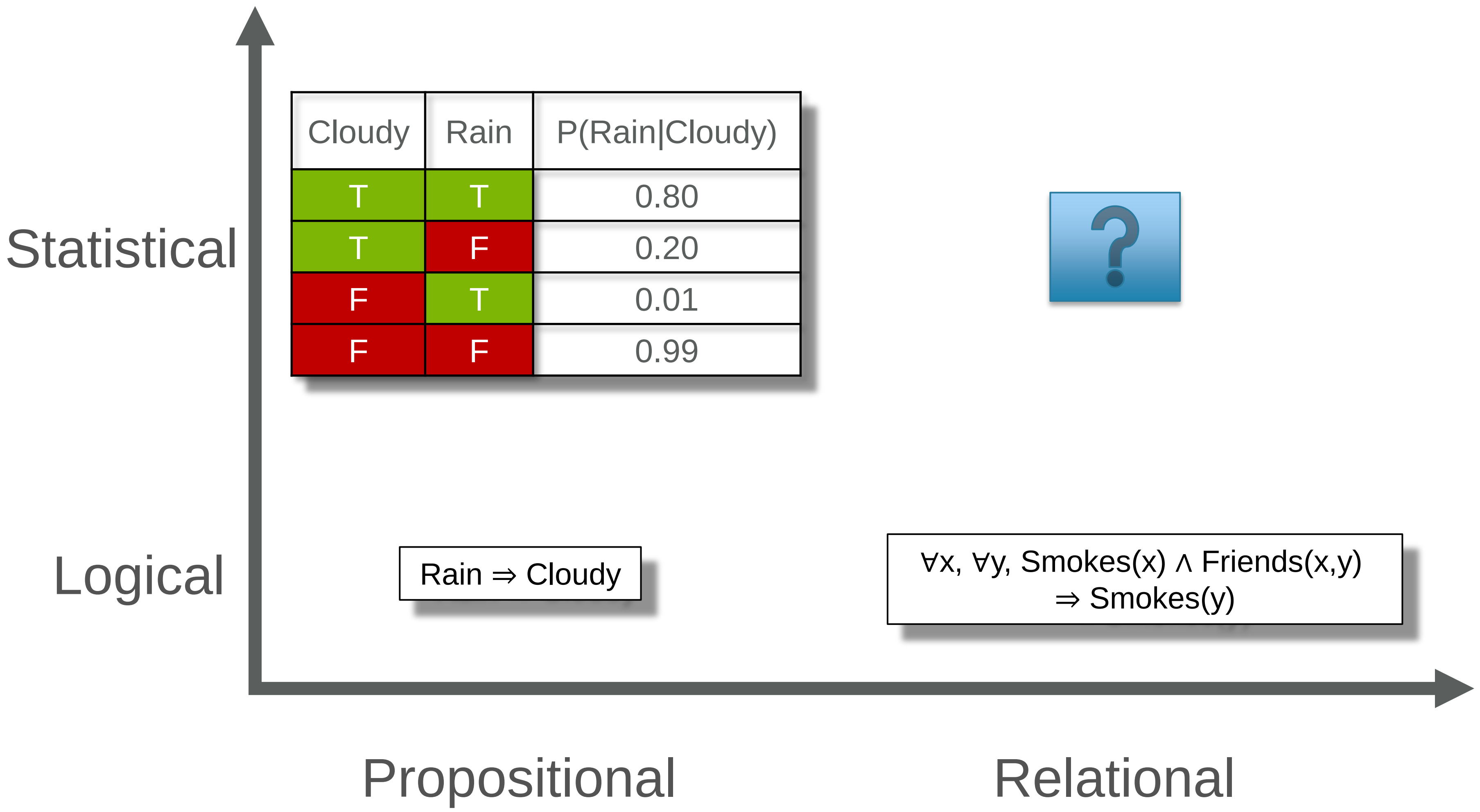
- Logical variables have domain of constants

  x,y range over domain People = {Alice,Bob}

- Ground formula has no logical variables

  Smokes(Alice) ∧ Friends(Alice,Bob) ⇒ Smokes(Bob)

Representations in AI and ML



Cloudy	Rain	P(Rain|Cloudy)
T	T	0.80
T	F	0.20
F	T	0.01
F	F	0.99

Why Statistical Relational Models?

- Probabilistic graphical models
  - ✔ Quantify uncertainty and noise
  - ✘ Not very expressive
    Rules of chess in ~100,000 pages
- First-order logic
  - ✔ Very expressive
    Rules of chess in 1 page
  - ✔ Good match for abundant relational data
  - ✘ Hard to express uncertainty and noise

Graphical Model Learning

Medical Records

Bayesian Network

Name	Cough	Asthma	Smokes
Alice	1	1	0
Bob	0	0	0
Charlie	0	1	0
Dave	1	0	1
Eve	1	0	0

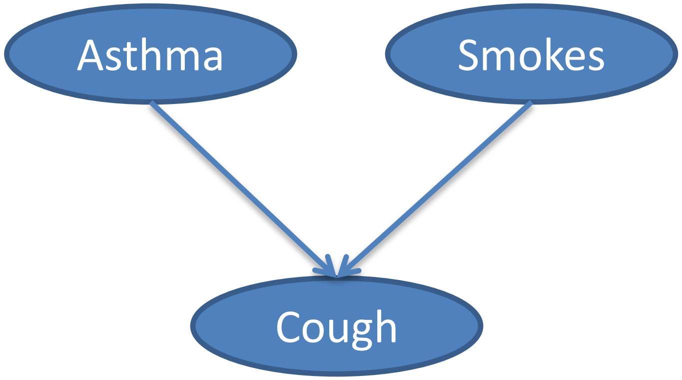
Frank	1	?	?

Friends

Brothers

Frank	1	0.3	0.2

Frank	1	0.2	0.6

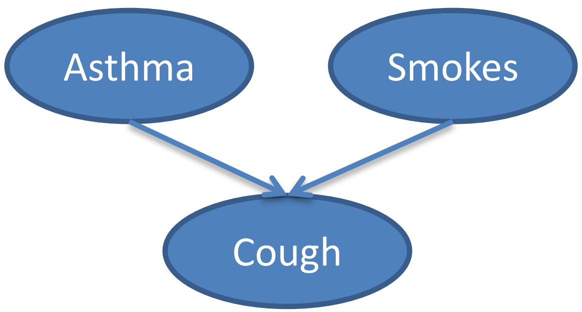


Rows are **independent** during learning and inference!

Statistical Relational Representations

Augment graphical model with relations between entities (rows).

Intuition



+ Friends have similar smoking habits

+ Asthma can be hereditary

Markov Logic

2.1 Asthma ⇒ Cough

3.5 Smokes ⇒ Cough

Logical variables refer to entities

1.9 Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y)

1.5 Asthma (x) ∧ Family(x,y) ⇒ Asthma (y)

Classical Machine Learning

Purchases

Model

Name	Age	Product	Price
Dave	40	Android	€249
Alice	35	iPhone	€799
Bob	32	iPhone	€799
Charlie	22	iPhone	€699
Eve	17	Android	€299
Frank	15	Android	€199

People **older** than **27** probably buy **iPhone**.

People **younger** than **27** probably buy **Android**.

Inference: *Does Guy buy an iPhone?*
Answer: Yes, with probability 66%

Statistical Relational Learning

Purchases

Name	Age	Product	Price
Dave	40	Android	€249
Alice	35	iPhone	€799
Bob	32	iPhone	€799
Charlie	22	iPhone	€699
Eve	17	Android	€299
Frank	15	Android	€199

Relationships

Person A	Person B	Type
Alice	Bob	Spouse
Alice	Charlie	Mother
Bob	Charlie	Father
Dave	Eve	Father
Dave	Frank	Father
Eve	Frank	Siblings

Family 1

Family 2

→ 

Model

Family members probably buy the **same type** of phone.

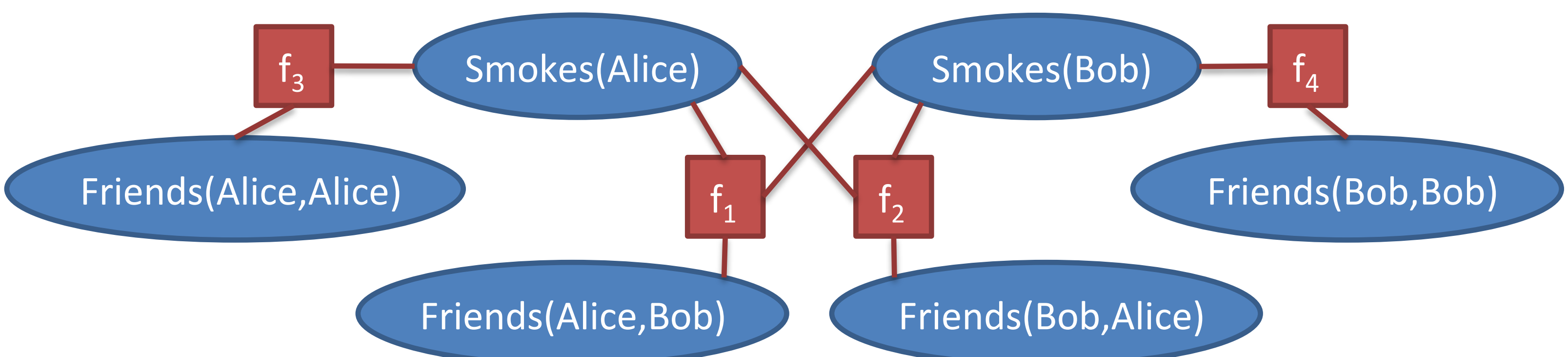
Example: Markov Logic

- Weighted First-Order Logic

Weight or Probability — FOL Formula

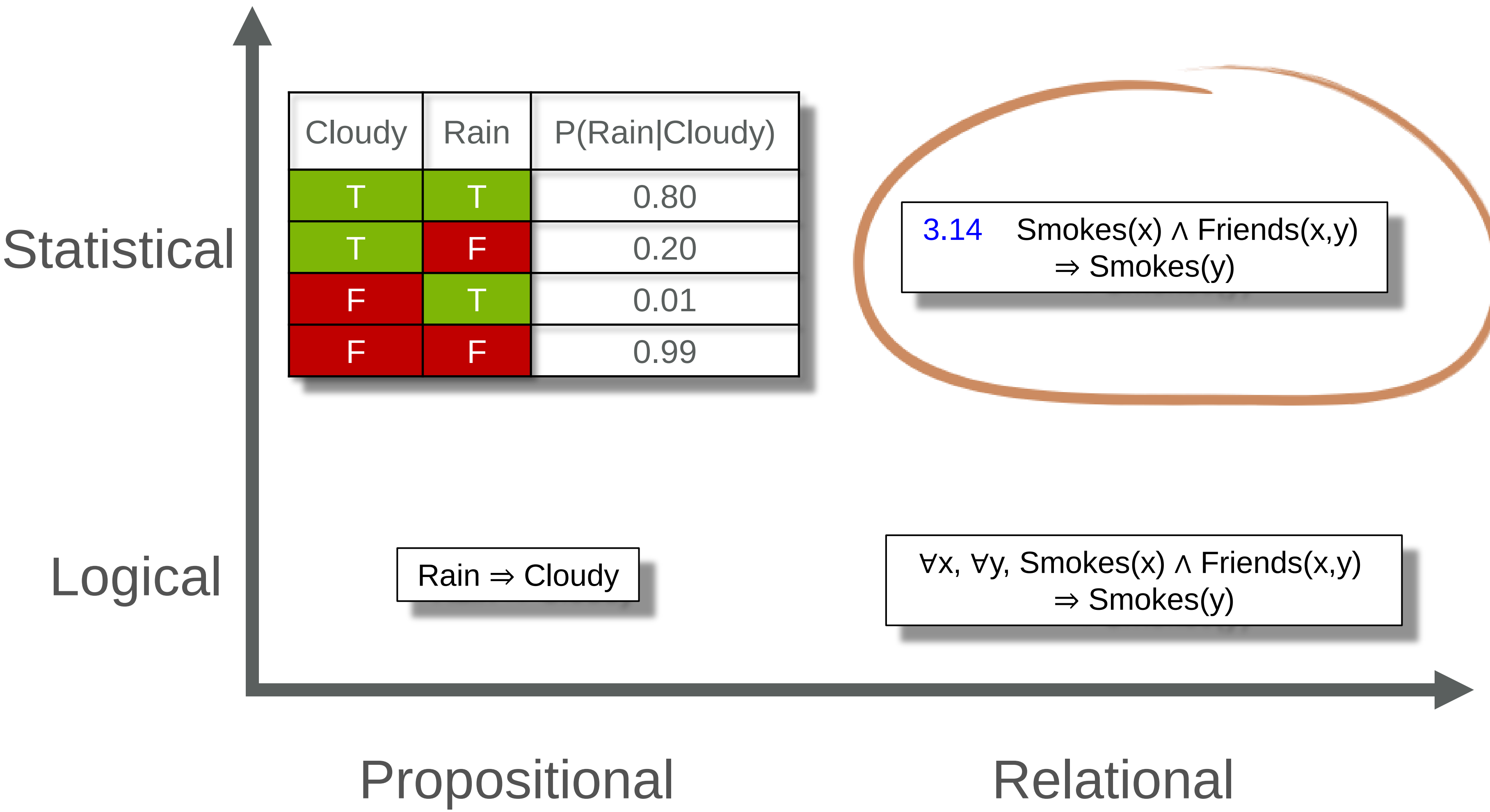
$$3.14 \quad \text{Smokes}(x) \land \text{Friends}(x,y) \Rightarrow \text{Smokes}(y)$$

- Ground atom/tuple = random variable in {true,false}
  e.g., Smokes(Alice), Friends(Alice,Bob), etc.
- Ground formula = factor in propositional factor graph



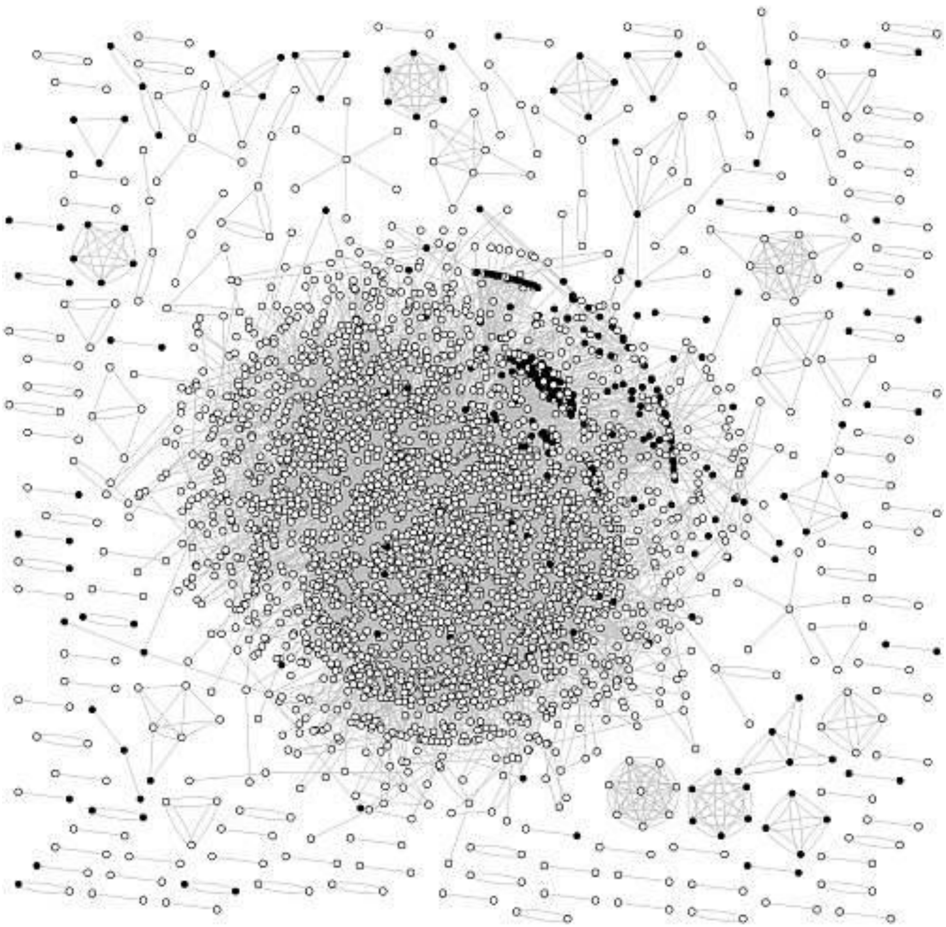
[Richardson'06]

Representations in AI and ML



Cloudy	Rain	P(Rain|Cloudy)
T	T	0.80
T	F	0.20
F	T	0.01
F	F	0.99

Collective Classification

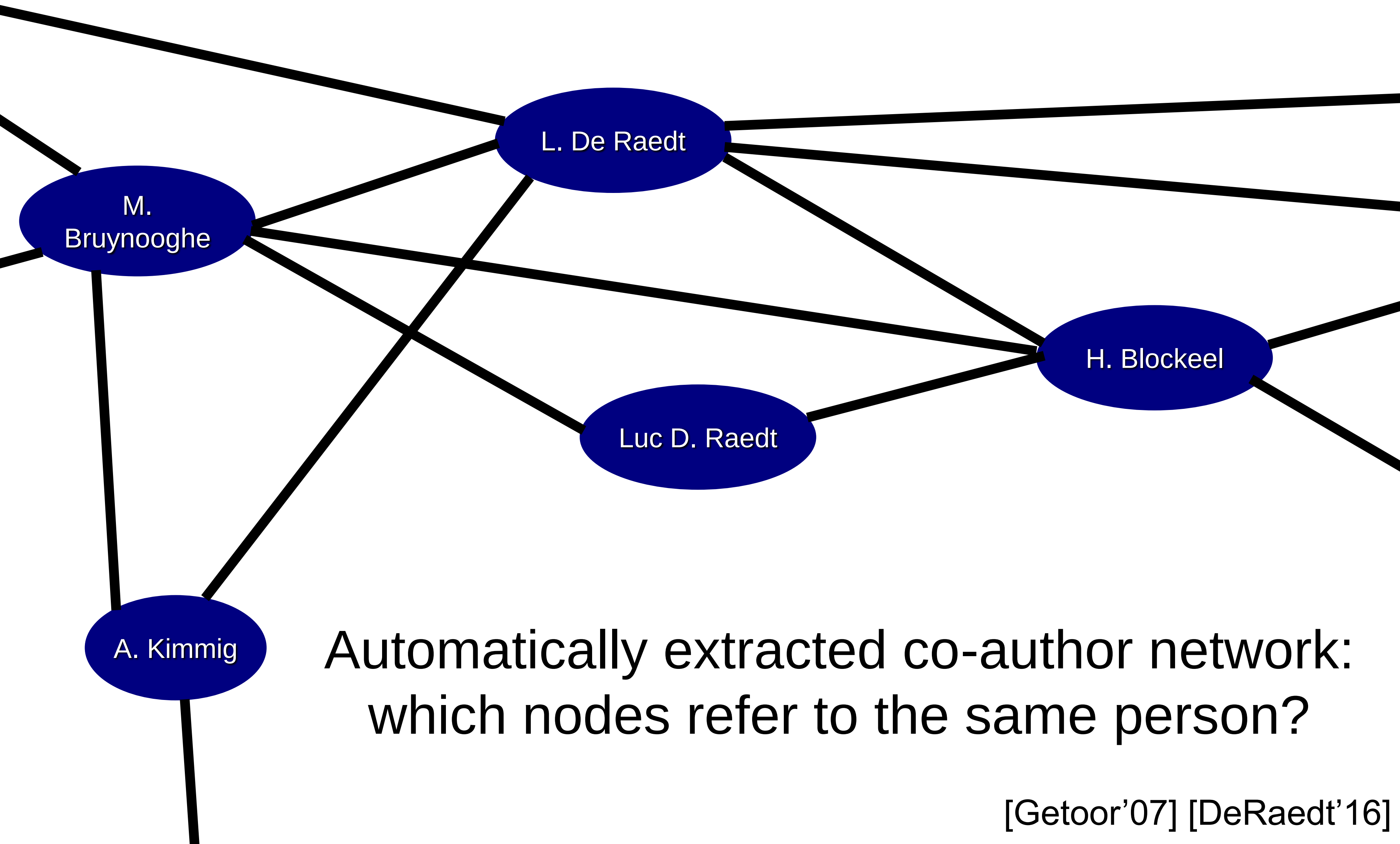


Can we predict the type of the nodes given information on its links and attributes?

E.g., the type of a webpage given its links and the words on the page?

[Getoor'07] [DeRaedt'16]

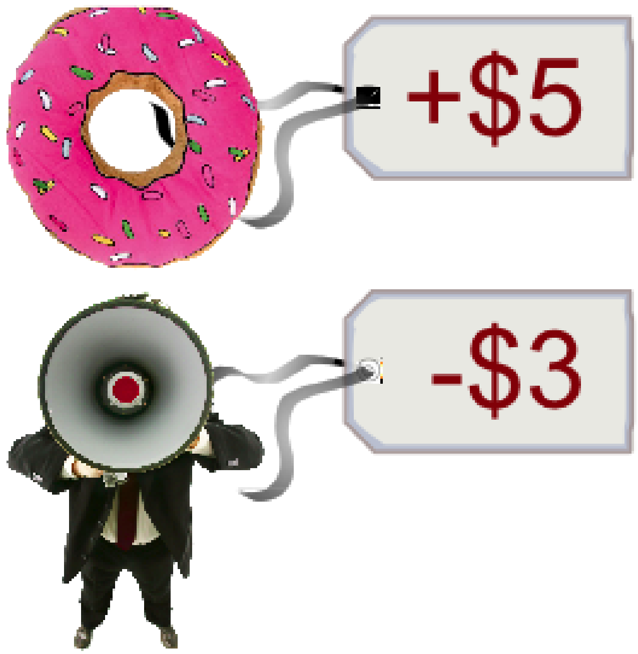
Entity Resolution



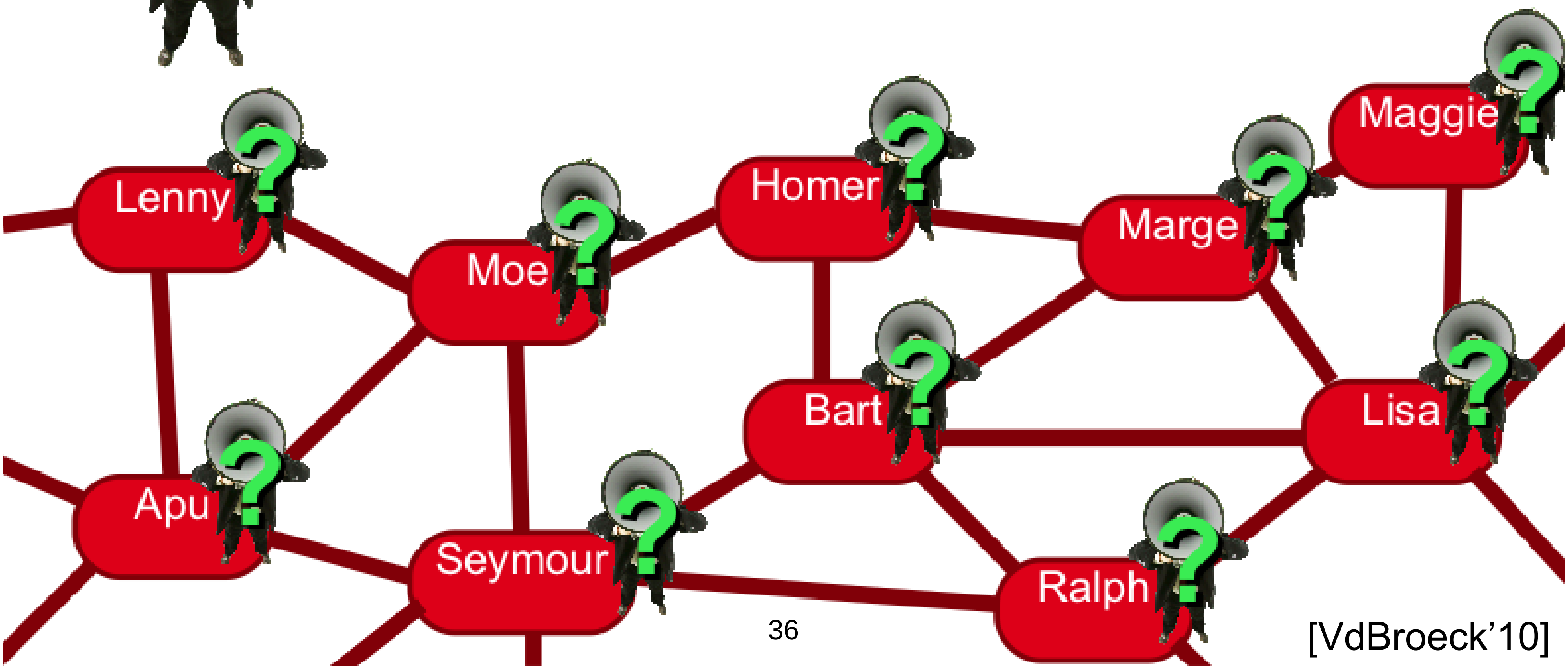
Automatically extracted co-author network:
which nodes refer to the same person?

[Getoor'07] [DeRaedt'16]

Viral Marketing

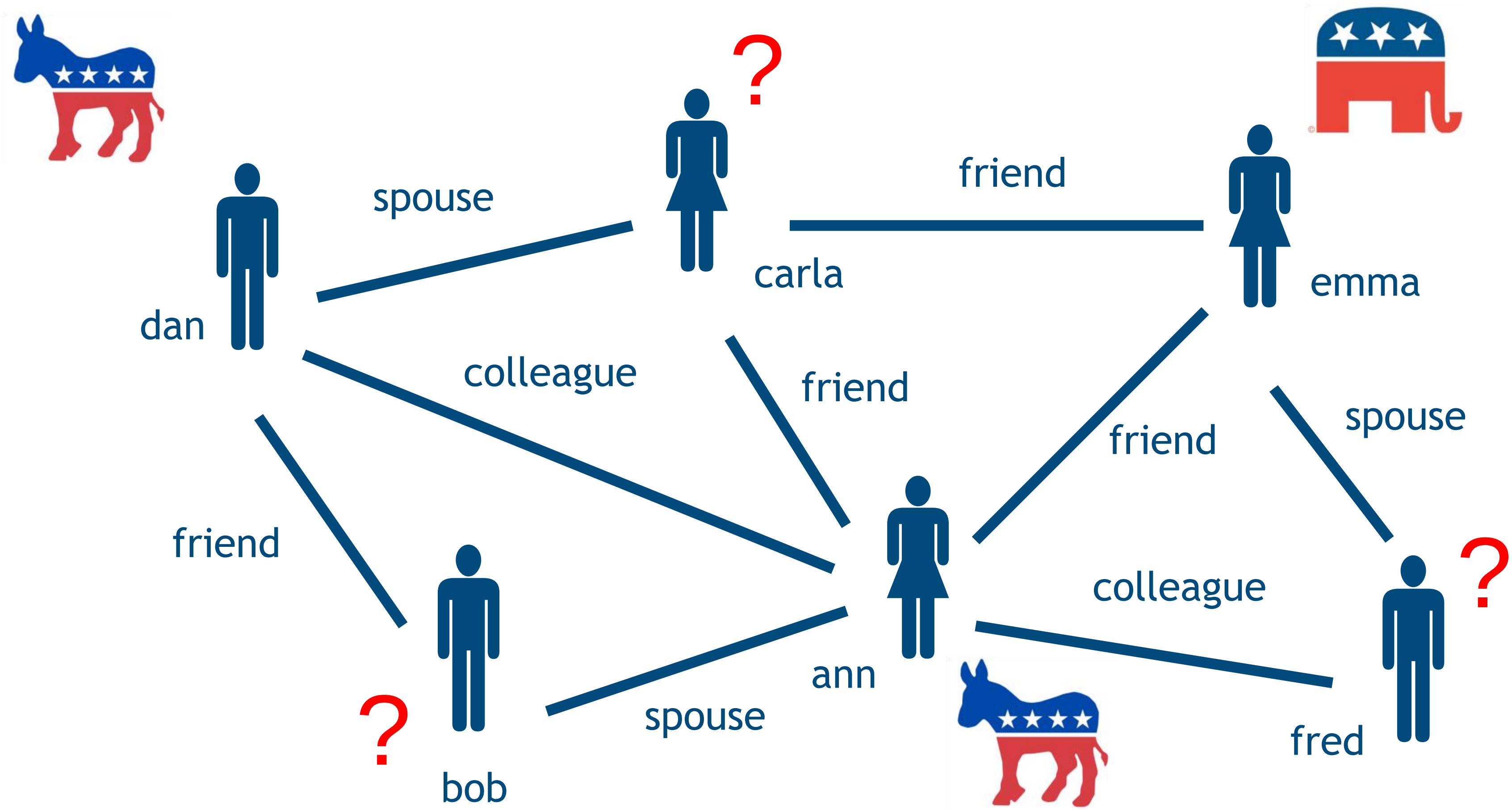


Which advertising strategy maximizes expected profit?



[VdBroeck'10]

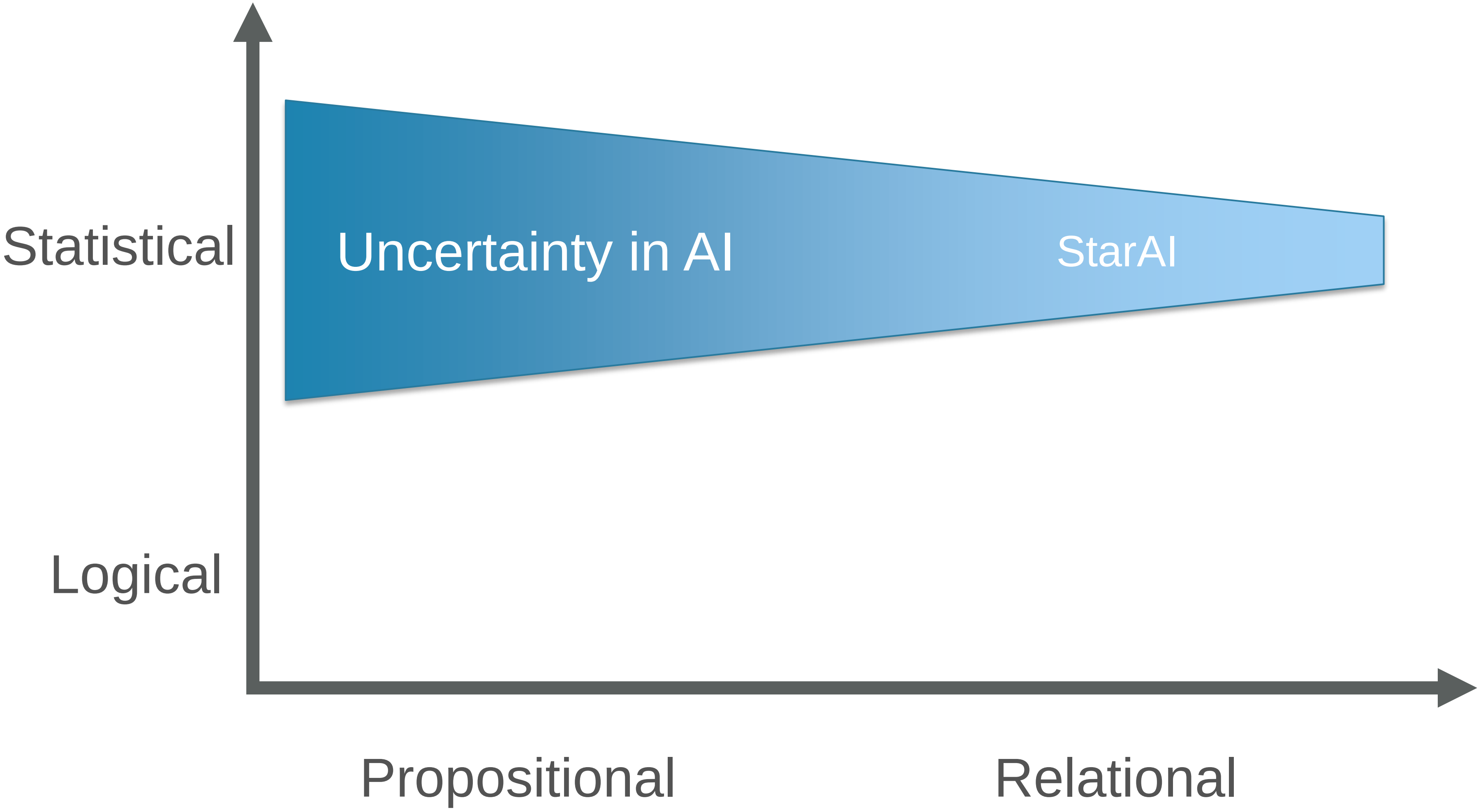
Voter Opinion Modeling



Can we predict preferences?

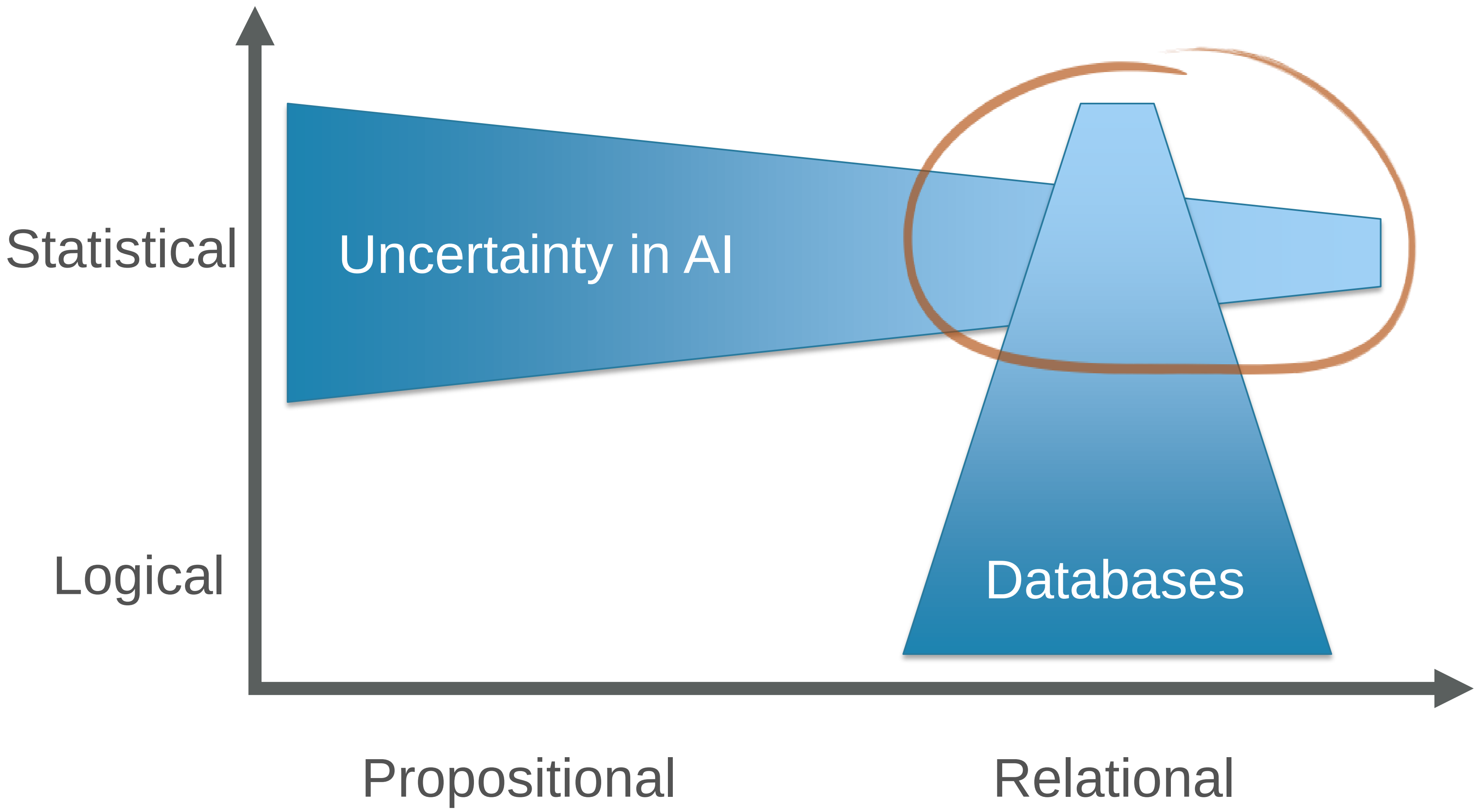
[Bach'15]

Summary



Statistical
Logical
Uncertainty in AI
StarAI
Propositional
Relational

Summary

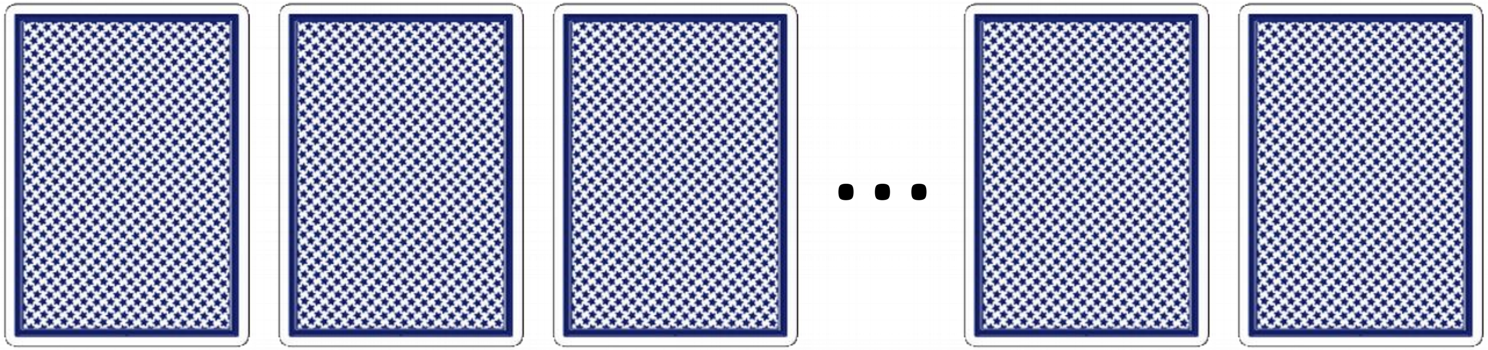

Statistical
Logical
Uncertainty in AI
Databases
Propositional
Relational

Why Lifted Inference?

- Main idea: exploit high level relational representation to speed up reasoning

- Let’s see an example…

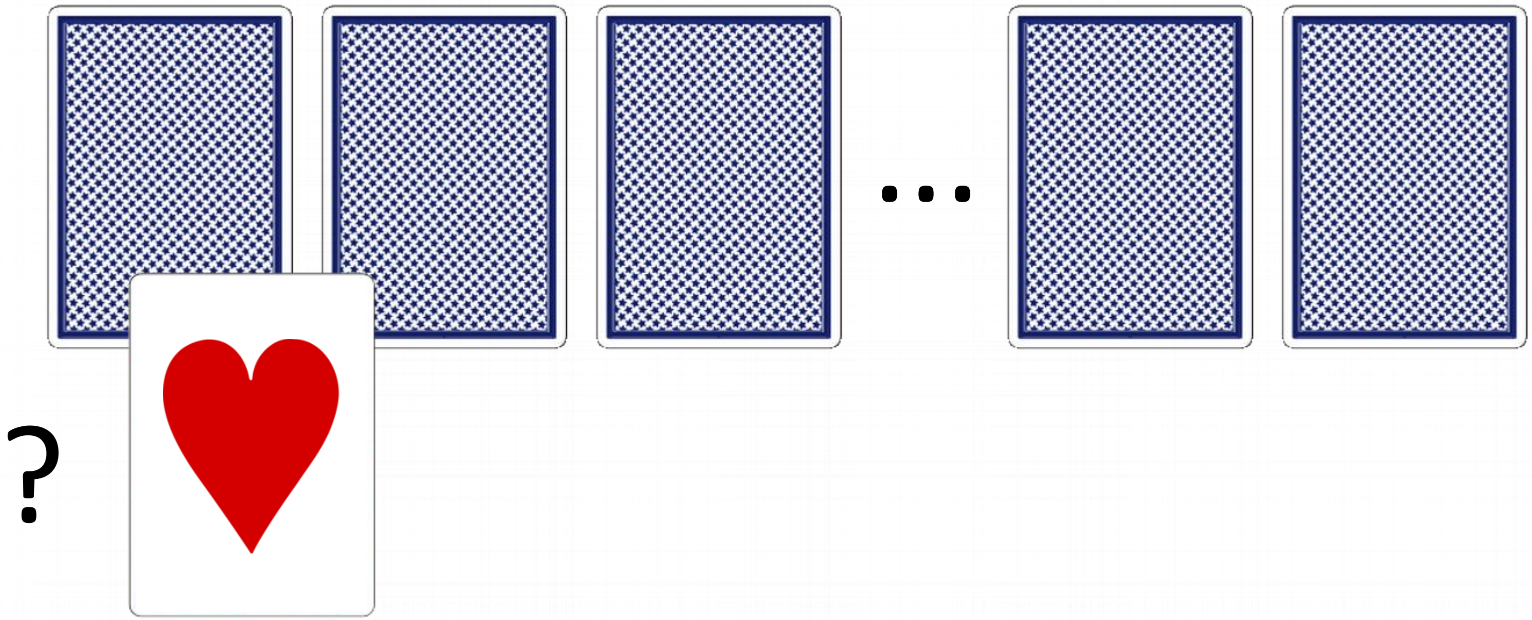
A Simple Reasoning Problem



- 52 playing cards
- Let us ask some simple questions

[VdB'15]

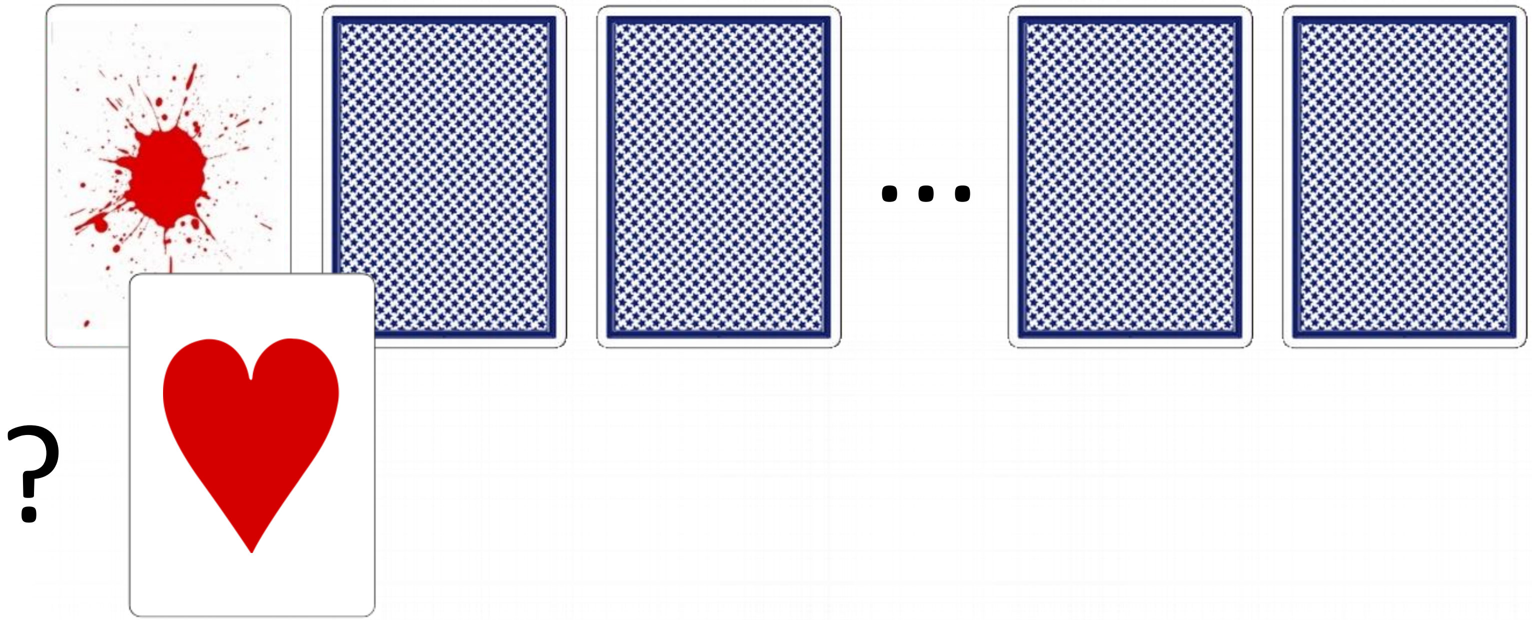
A Simple Reasoning Problem



Probability that Card1 is Hearts? 1/4

[VdB'15]

A Simple Reasoning Problem

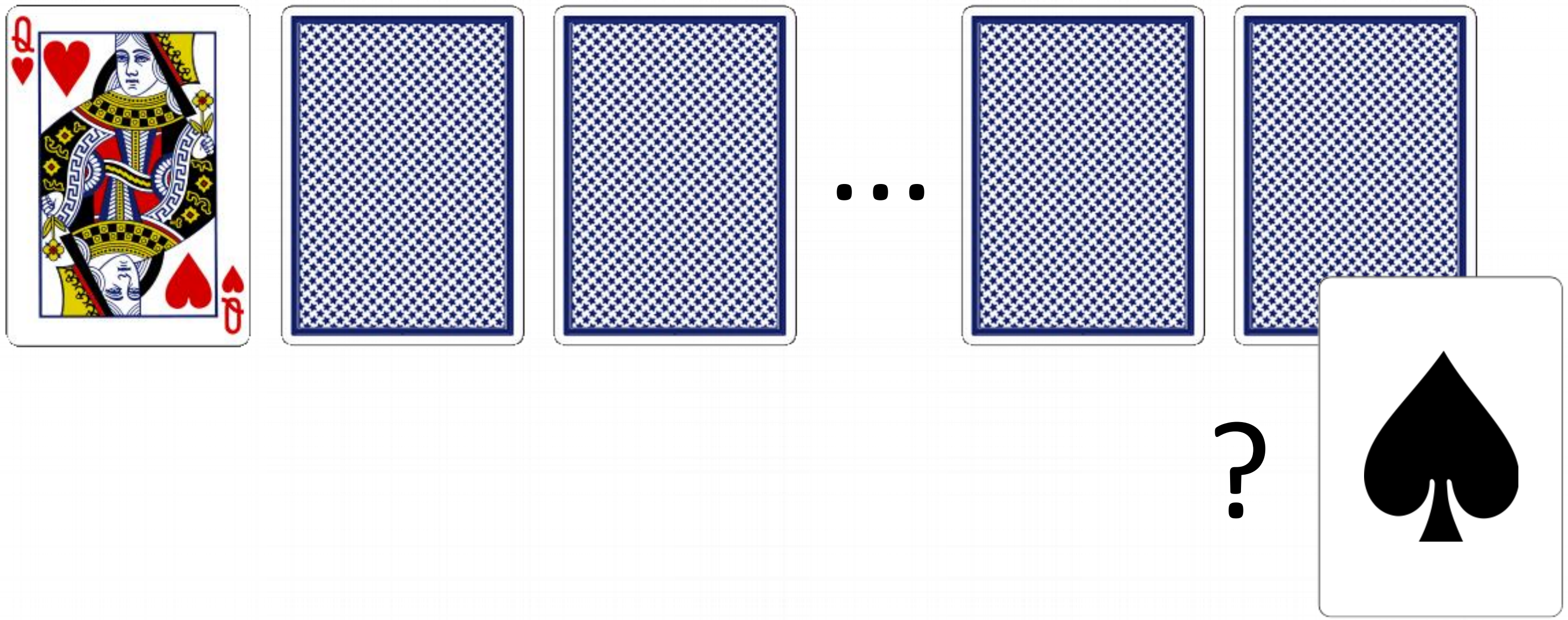


Probability that Card1 is Hearts given that Card1 is red?

1/2

[VdB'15]

A Simple Reasoning Problem

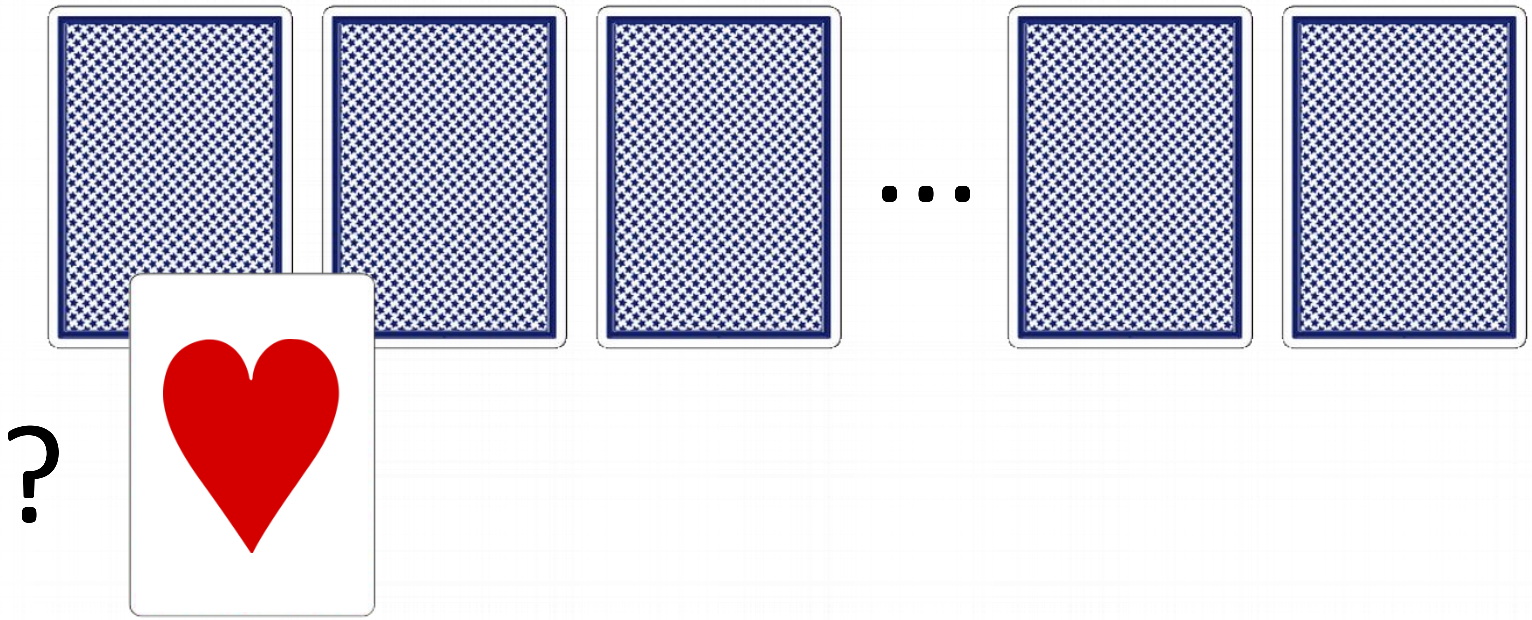


Probability that Card52 is Spades given that Card1 is QH?

13/51

[VdB'15]

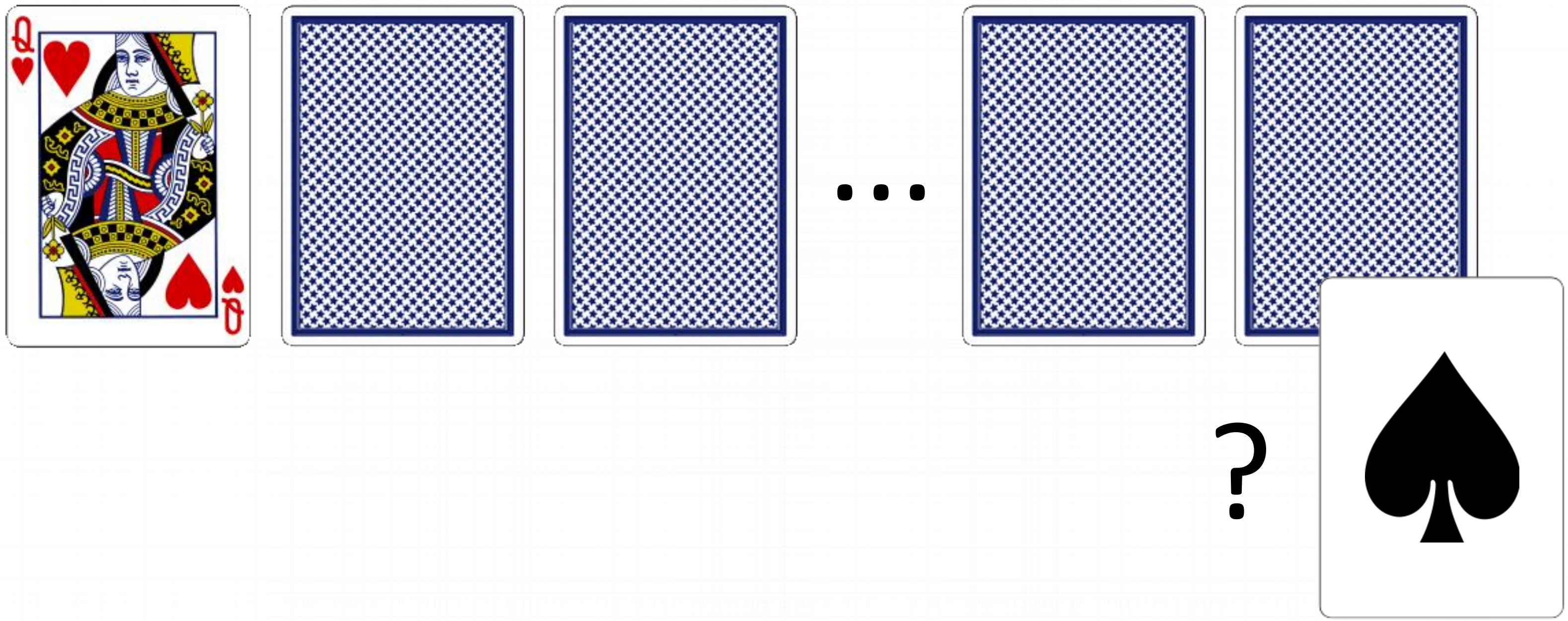
A Simple Reasoning Problem



Probability that Card1 is Hearts? 1/4

[VdB'15]

A Simple Reasoning Problem



Probability that Card52 is Spades given that Card1 is QH?

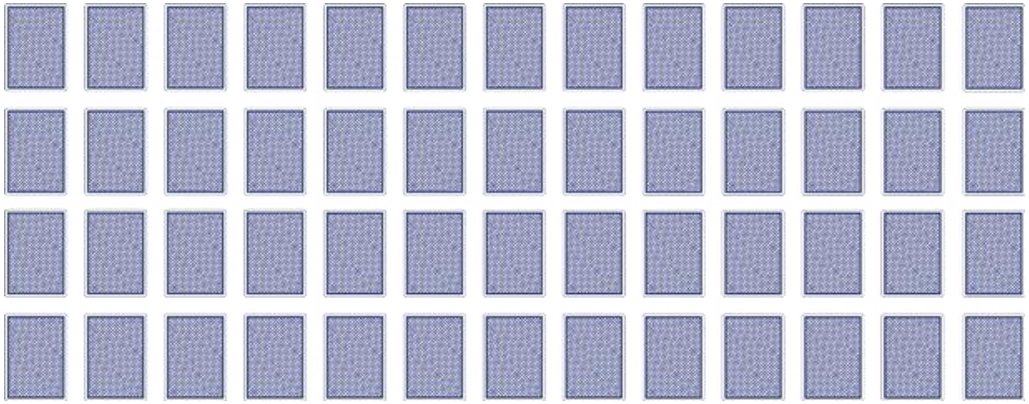
13/51

[VdB'15]

Automated Reasoning

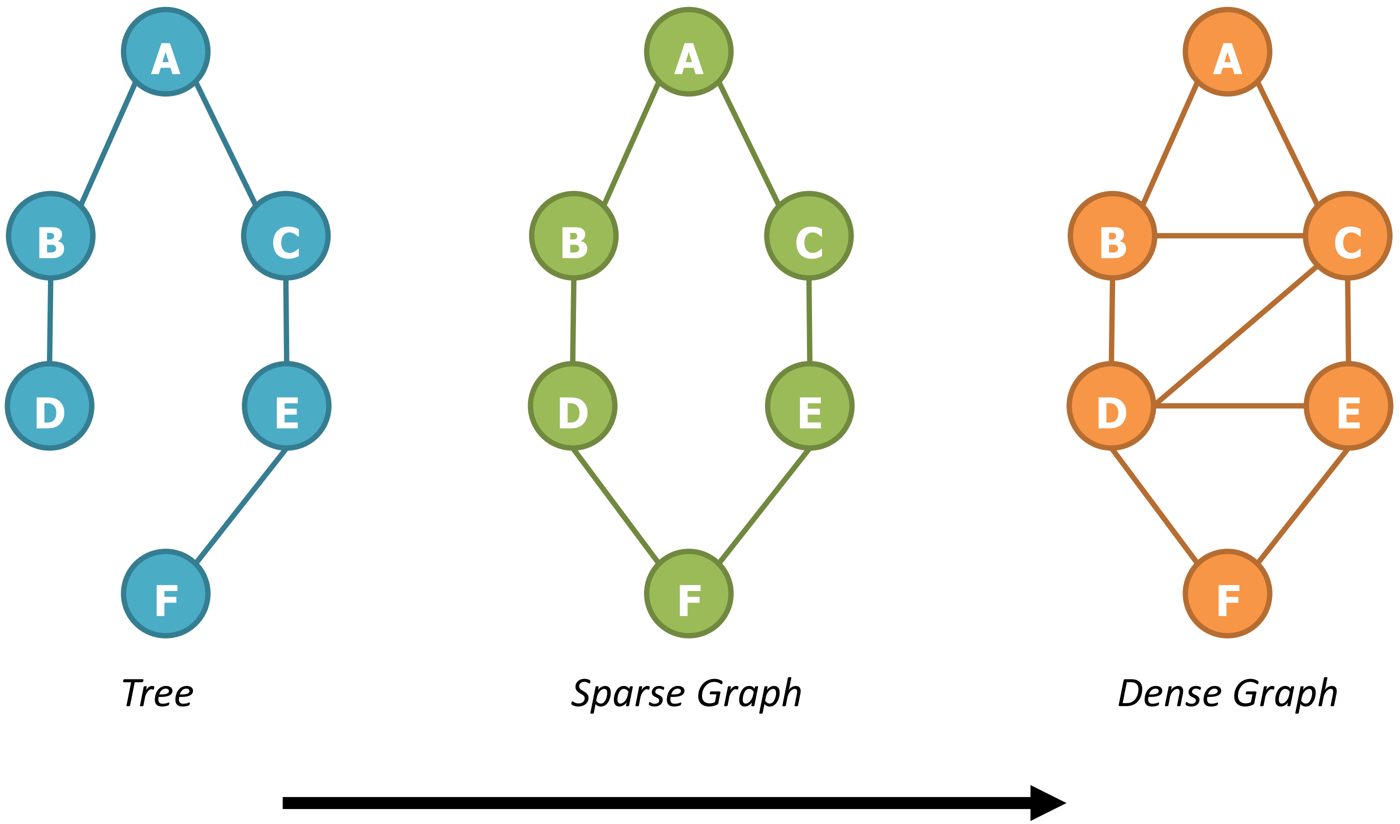
Let us automate this:

1. Probabilistic graphical model (e.g., factor graph)



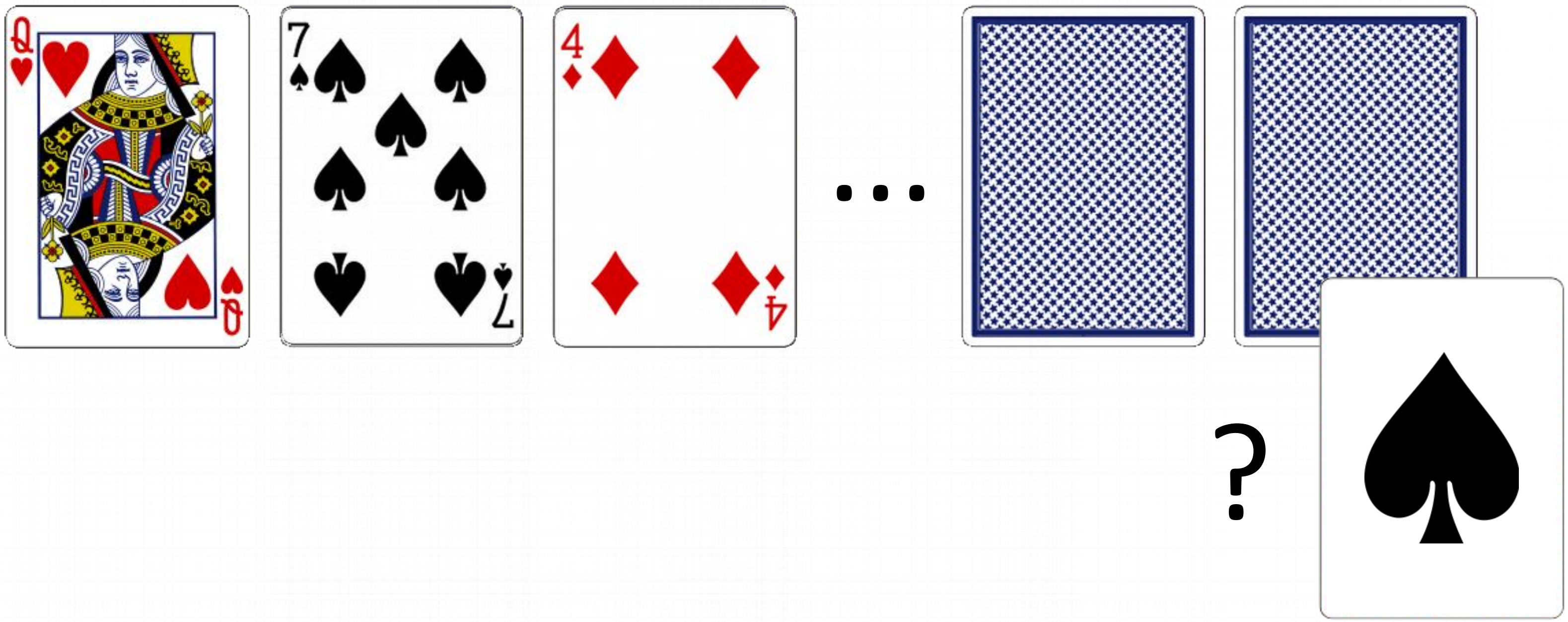
2. Probabilistic inference algorithm
   (e.g., variable elimination or junction tree)

Classical Reasoning



- Higher treewidth
- Fewer conditional independencies
- Slower inference

Is There Conditional Independence?



$P(\text{Card52} \mid \text{Card1}) \neq P(\text{Card52} \mid \text{Card1}, \text{Card2})$

$13/51 \neq 12/50$

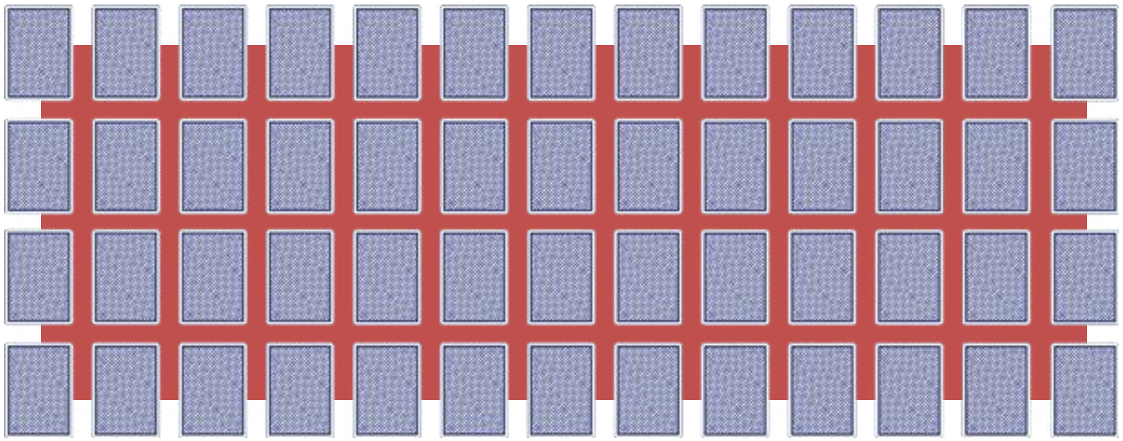
$P(\text{Card52} \mid \text{Card1}, \text{Card2}) \neq P(\text{Card52} \mid \text{Card1}, \text{Card2}, \text{Card3})$

$12/50 \neq 12/49$

Automated Reasoning

Let us automate this:

1. Probabilistic graphical model (e.g., factor graph)
   is fully connected!

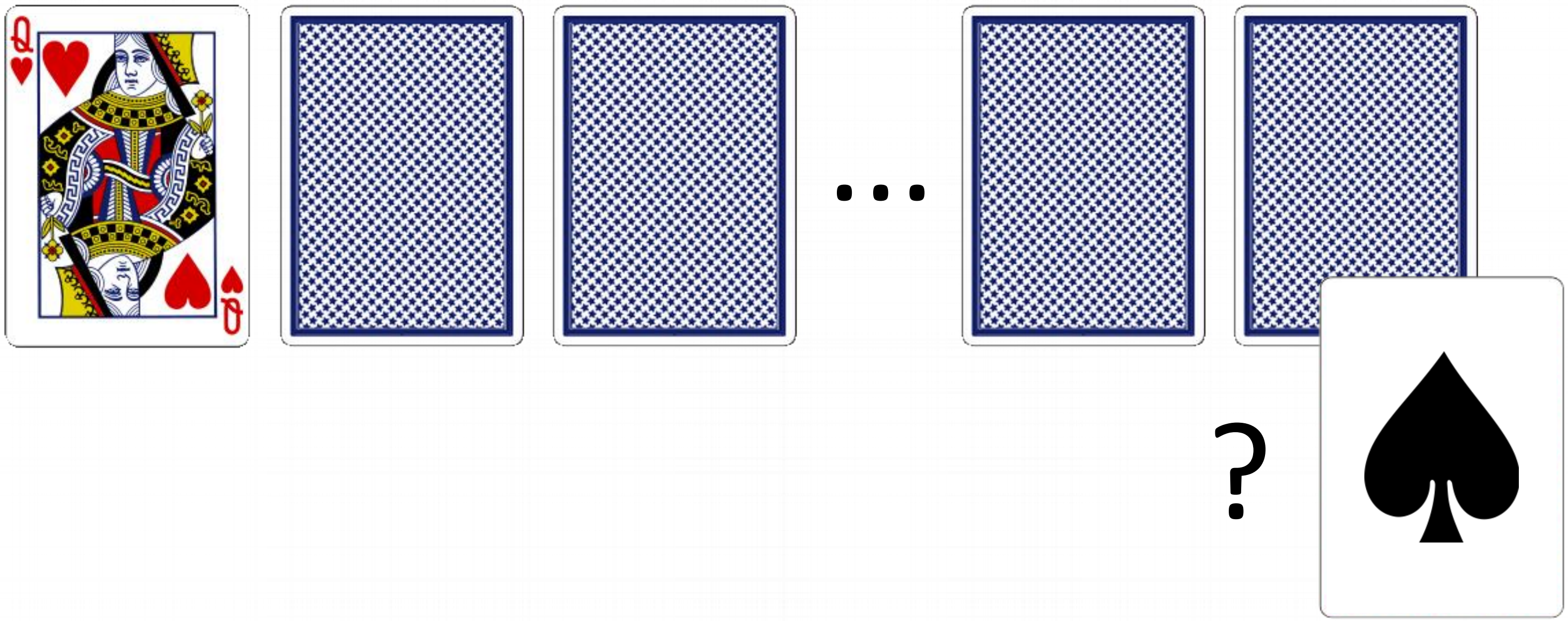


(artist's impression)

2. Probabilistic inference algorithm
   (e.g., variable elimination or junction tree)
   builds a table with **52^{52}** rows

[VdB'15]

What's Going On Here?

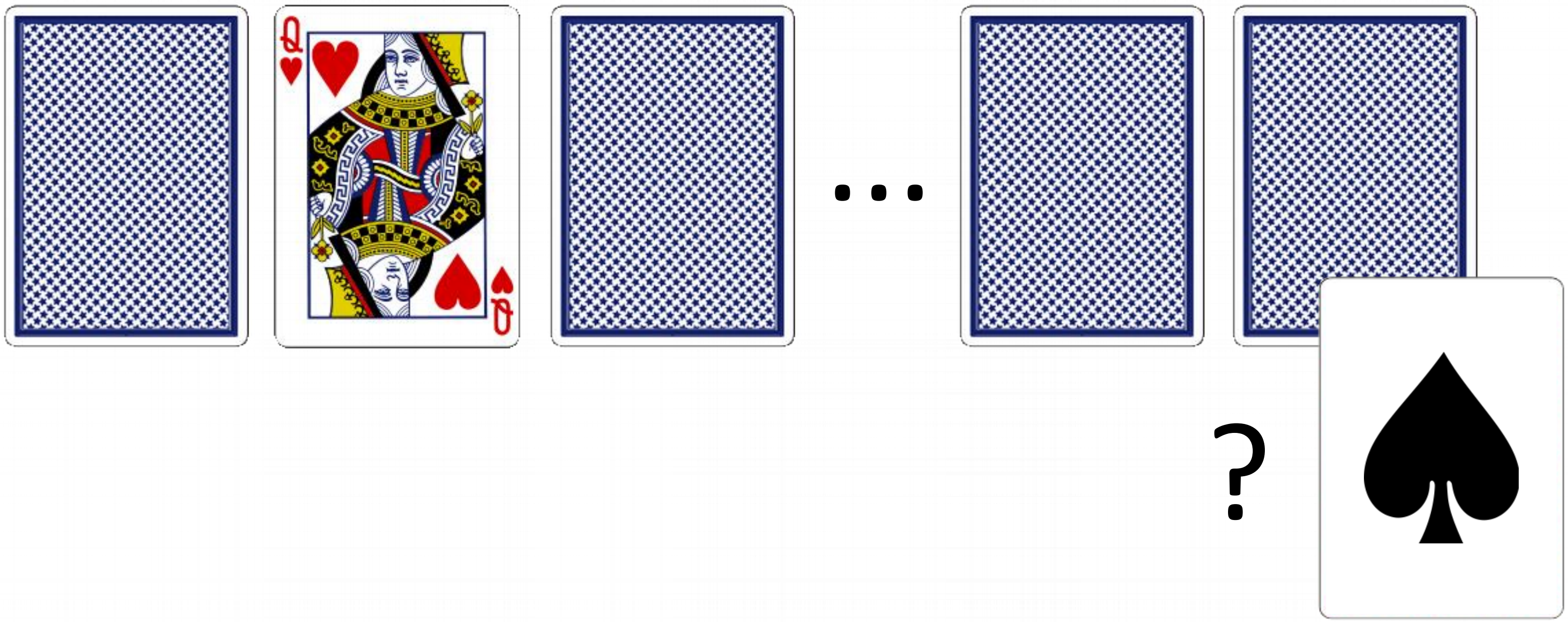


Probability that Card52 is Spades given that Card1 is QH?

13/51

[VdB' 15]

What's Going On Here?

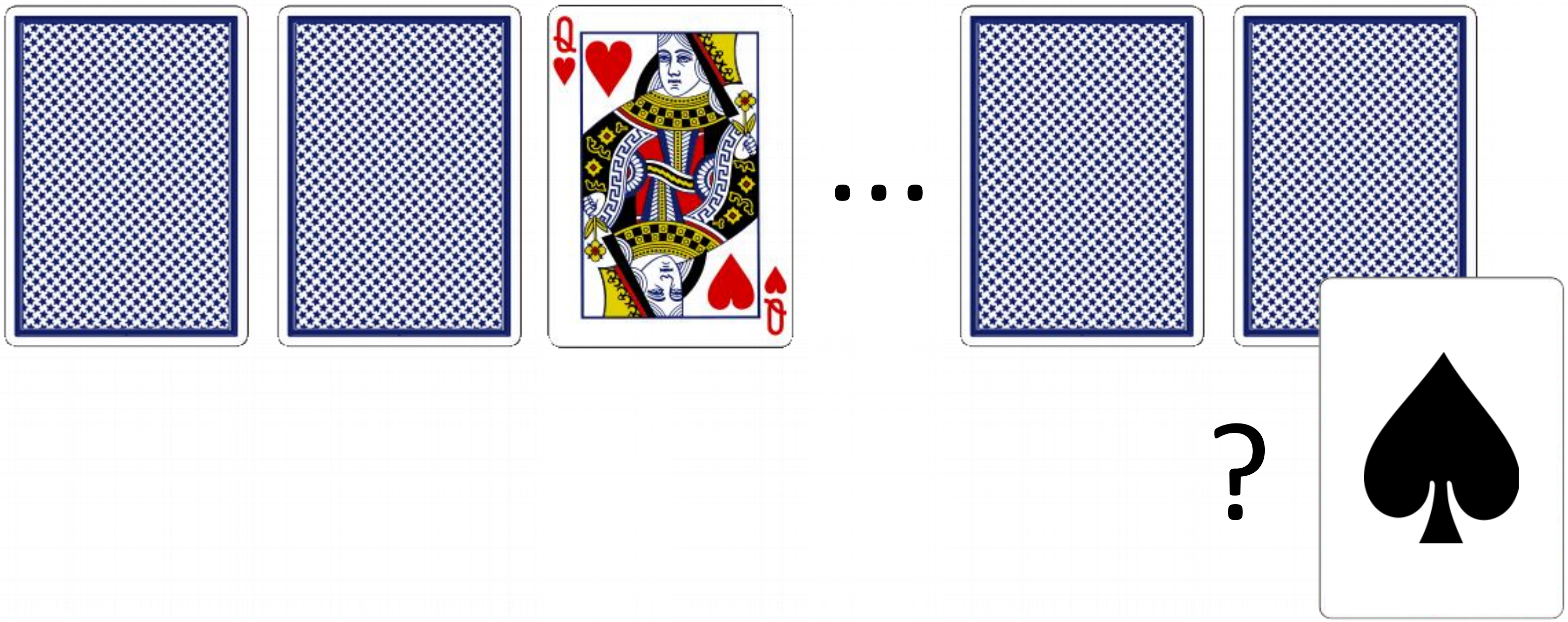


Probability that Card52 is Spades given that Card2 is QH?

13/51

[VdB' 15]

What's Going On Here?

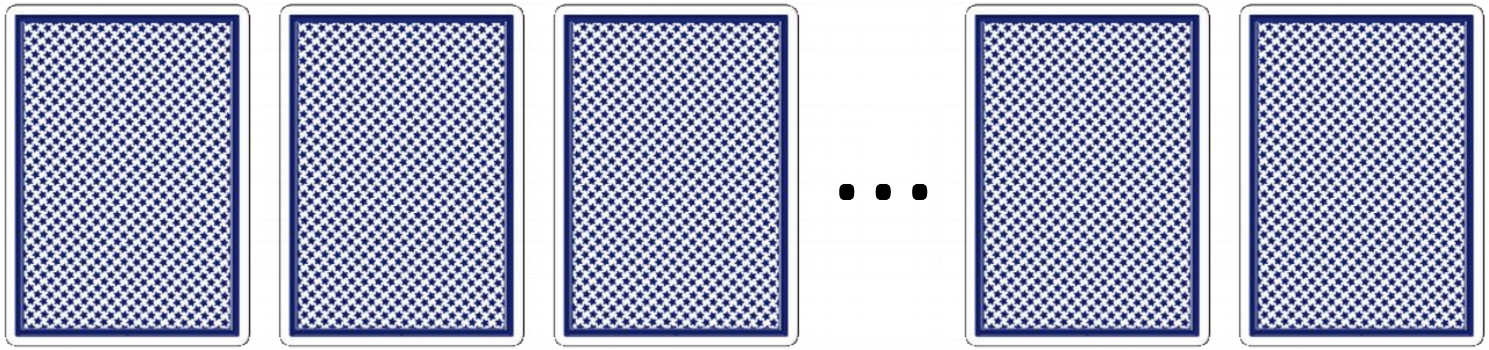


Probability that Card52 is Spades given that Card3 is QH?

13/51

[VdB' 15]

Tractable Reasoning



What's going on here?

Which property makes reasoning tractable?

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

⇒ Lifted Inference

[Niepert'14]

Automated Reasoning

Let us automate this:

- **Relational** model

$$\forall p, \exists c, Card(p,c)$$
$$\forall c, \exists p, Card(p,c)$$
$$\forall p, \forall c, \forall c', Card(p,c) \land Card(p,c') \Rightarrow c = c'$$

- **Lifted** probabilistic inference algorithm

Other Examples of Lifted Inference

- First-order resolution

$$\forall x, \text{Human}(x) \Rightarrow \text{Mortal}(x)$$
$$\forall x, \text{Greek}(x) \Rightarrow \text{Human}(x)$$

implies

$$\forall x, \text{Greek}(x) \Rightarrow \text{Mortal}(x)$$

Other Examples of Lifted Inference

- First-order resolution
- Reasoning about populations

  We are investigating a rare disease. The disease is more rare in women, presenting only in **one in every two billion women** and **one in every billion men**. Then, assuming there are **3.4 billion men** and **3.6 billion women** in the world, the probability that **more than five people** have the disease is

$$1 - \sum_{n=0}^{5} \sum_{f=0}^{n} \binom{3.6 \cdot 10^9}{f} \left(1 - 0.5 \cdot 10^{-9}\right)^{3.6 \cdot 10^9 - f} \left(0.5 \cdot 10^{-9}\right)^{f}$$

$$\times \binom{3.4 \cdot 10^9}{(n-f)} \left(1 - 10^{-9}\right)^{3.4 \cdot 10^9 - (n-f)} \left(10^{-9}\right)^{(n-f)}$$

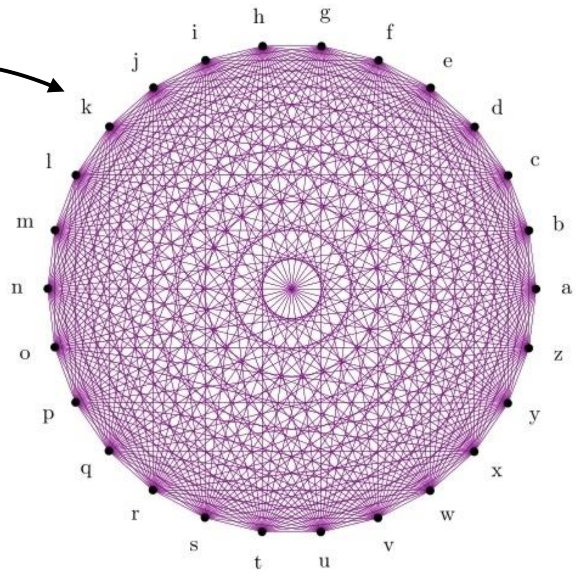
[VdB’ 13]

Lifted Inference in SRL

- Statistical relational model (e.g., MLN)

3.14 FacultyPage(x) ∧ Linked(x,y) ⇒ CoursePage(y)

- As a probabilistic graphical model:
  - 26 pages; 728 variables; 676 factors
  - 1000 pages; 1,002,000 variables; 1,000,000 factors



- Highly intractable?
  - **Lifted inference** in milliseconds!

Statistical Properties

1. Independence

P(

Name	Cough	Asthma	Smokes
Alice	1	1	0
Bob	0	0	0
Charlie	0	1	0

) =

P(

Alice	1	1	0

)

x P(

Bob	0	0	0

)

x P(

Charlie	0	1	0

)

2. Partial Exchangeability

P(

Name	Cough	Asthma	Smokes
Alice	1	1	0
Bob	0	0	0
Charlie	0	1	0

) = P(

Name	Cough	Asthma	Smokes
Charlie	1	1	0
Alice	0	0	0
Bob	0	1	0

)

3. Independent and identically distributed (i.i.d.)

= Independence + Partial Exchangeability

Statistical Properties for Tractability

- Tractable classes independent of representation
- Traditionally:
  - Tractable learning from **i.i.d. data**
  - Tractable inference when **cond. independence**
- New understanding:
  - Tractable learning from **exchangeable data**
  - Tractable inference when
    - **Conditional independence**
    - **Conditional exchangeability**
    - **A combination**

[Niepert'14]

Summary of Motivation

- Relational data is everywhere:
  - Databases in industry and sciences
  - Knowledge bases
  - Probabilistically extracted/learned/queried
- Lifted inference:
  - Use relational structure during reasoning
  - Very efficient where traditional methods break

This tutorial: Lifted Inference in Relational Models

Outline

- Part 1: Motivation
- Part 2: Probabilistic Databases
- Part 3: Weighted Model Counting
- Part 4: Lifted Inference for WFOMC
- Part 5: Completeness of Lifted Inference
- Part 6: Query Compilation
- Part 7: Symmetric Lifted Inference Complexity
- Part 8: Open-World Probabilistic Databases
- Part 9: Discussion & Conclusions

What Everyone Should Know about Databases

- Database = several relations (a.k.a. tables)
- SQL Query = FO Formula
- Boolean Query = FO Sentence

What Everyone Should Know about Databases

Database: relations (= tables)

D =

Smoker

X	Y
Alice	2009
Alice	2010
Bob	2009
Carol	2010

Friend

X	Z
Alice	Bob
Alice	Carol
Bob	Carol
Carol	Bob

What Everyone Should Know about Databases

Database: relations (= tables)

D =

Smoker

X	Y
Alice	2009
Alice	2010
Bob	2009
Carol	2010

Friend

X	Z
Alice	Bob
Alice	Carol
Bob	Carol
Carol	Bob

Query: First Order Formula

$Q(z) = \exists x\,(\text{Smoker}(x,\text{'2009'}) \wedge \text{Friend}(x,z))$

Find friends of smokers in 2009

Query answer: Q(D) =

Z
Bob
Carol

Conjunctive Queries CQ = FO(∃, ∧)
Union of CQs UCQ = FO(∃, ∧, ∨)

What Everyone Should Know about Databases

Database: relations (= tables)

D =

Smoker

x	y
Alice	2009
Alice	2010
Bob	2009
Carol	2010

Friend

x	z
Alice	Bob
Alice	Carol
Bob	Carol
Carol	Bob

Query: First Order Formula

$Q(z) = \exists x\, (\text{Smoker}(x, \text{'2009'}) \wedge \text{Friend}(x,z))$

Find friends of smokers in 2009

Query answer: Q(D) =

z
Bob
Carol

Conjunctive Queries CQ = FO(∃, ∧)
Union of CQs UCQ = FO(∃, ∧, ∨)

Boolean Query: FO Sentence

$Q = \exists x\, (\text{Smoker}(x, \text{'2009'}) \wedge \text{Friend}(x, \text{'Bob'}))$

Query answer: Q(D) = TRUE

What Everyone Should Know about Databases

Declarative Query → Query Plan
"what" → *"how"*

What Everyone Should Know about Databases

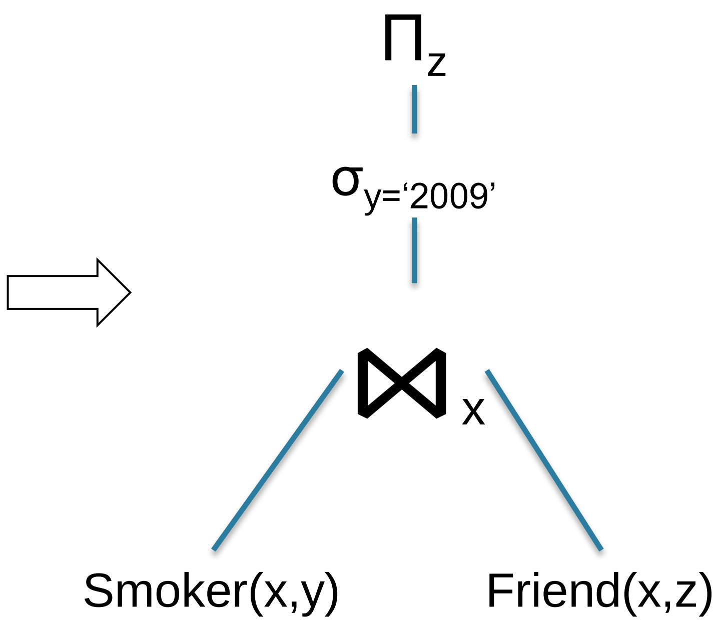
Declarative Query	→	Query Plan
"what"	→	*"how"*

$Q(z) = \exists x\, (\text{Smoker}(x,\text{'2009'}) \land \text{Friend}(x,z))$

What Everyone Should Know about Databases

Declarative Query	→	Query Plan
"what"	→	*"how"*

$Q(z) = \exists x\,(\text{Smoker}(x,\text{'2009'}) \wedge \text{Friend}(x,z))$

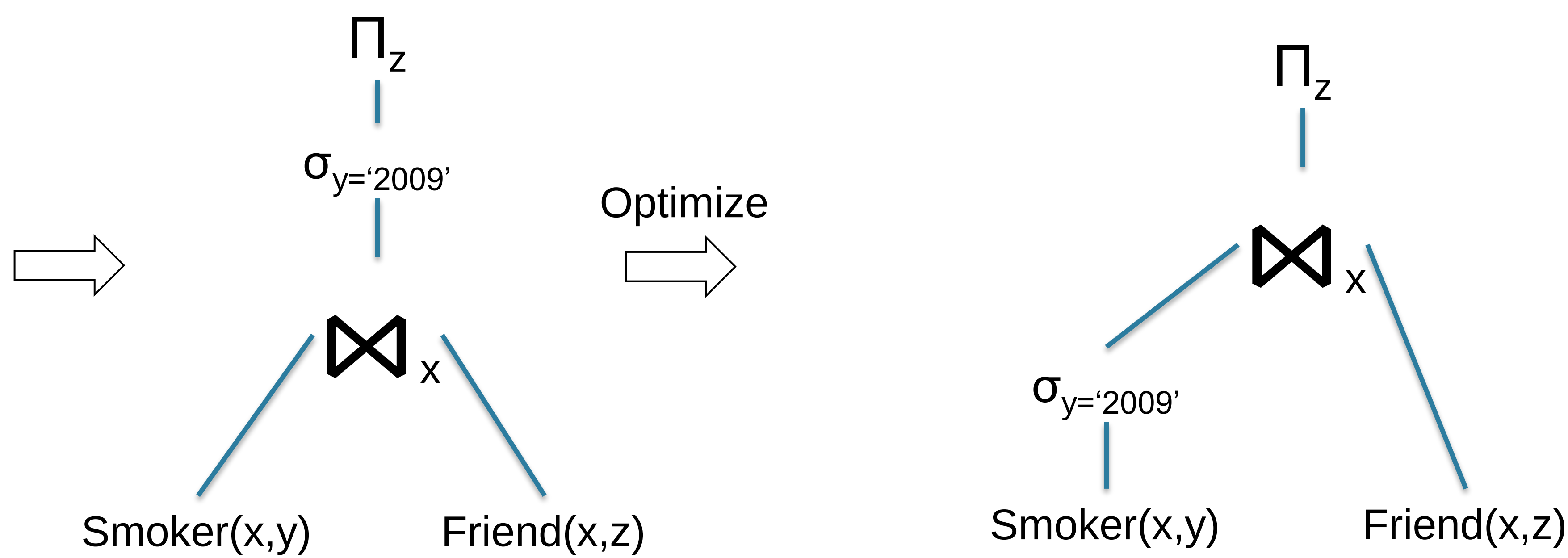


Logical Query Plan

What Everyone Should Know about Databases

Declarative Query	→	Query Plan
"what"	→	*"how"*

$Q(z) = \exists x\, (\text{Smoker}(x,\text{'2009'}) \wedge \text{Friend}(x,z))$

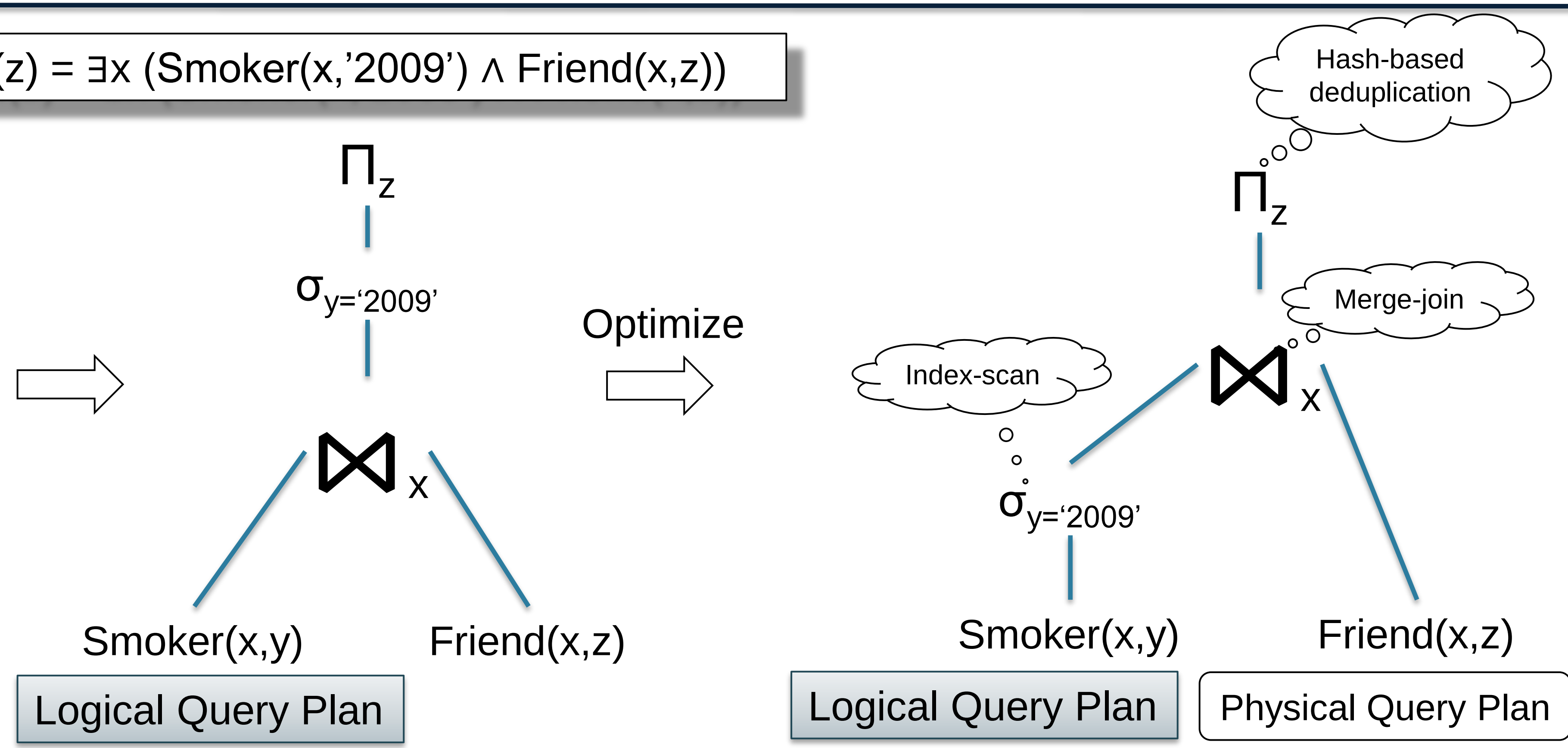


Logical Query Plan   Logical Query Plan

What Everyone Should Know about Databases

Declarative Query	→	Query Plan
"what"	→	*"how"*

Q(z) = ∃x (Smoker(x,'2009') ∧ Friend(x,z))



What Every Researcher Should Know about Databases

Problem: compute Q(D)

Moshe Vardi [Vardi'82]
2008 ACM SIGMOD Contribution Award

This talk: query = blue, data = red

What Every Researcher Should Know about Databases

Problem: compute Q(D)

Moshe Vardi [Vardi’82]
2008 ACM SIGMOD Contribution Award

- Data complexity:
  fix Q, complexity = f(D)

This talk: query = blue, data = red

What Every Researcher Should Know about Databases

Problem: compute Q(D)

Moshe Vardi [Vardi'82]
2008 ACM SIGMOD Contribution Award

- Data complexity:
  fix Q, complexity = f(D)

- Query complexity: (expression complexity)
  fix D, complexity = f(Q)

- Combined complexity:
  complexity = f(D, Q)

This talk: query = blue, data = red

Probabilistic Databases

- A probabilistic database = relational database where each tuple is a random variable
- Semantics = probability distribution over possible worlds (deterministic databases)
- In this talk: tuples are independent events

Example

Probabilistic database D:

Friend

x	y	P
A	B	p_1
A	C	p_2
B	C	p_3

Example

Probabilistic database D:

Friend

x	y	P
A	B	p_1
A	C	p_2
B	C	p_3

Possible worlds semantics:

$p_1 p_2 p_3$

x	y
A	B
A	C
B	C

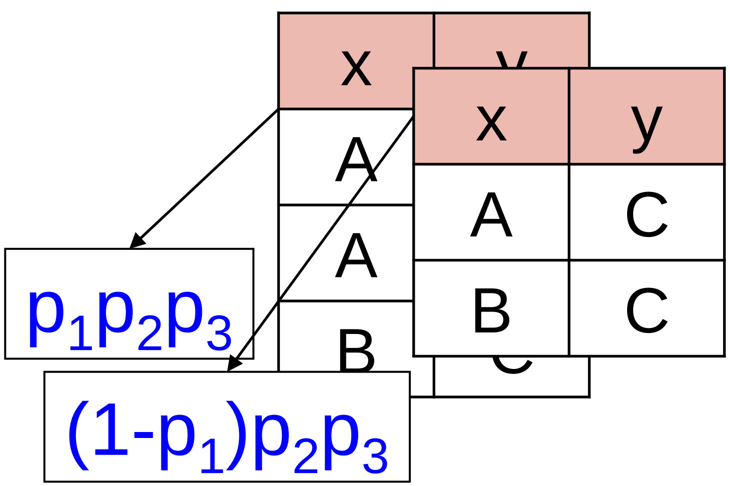
Example

Probabilistic database D:

Friend

x	y	P
A	B	p_1
A	C	p_2
B	C	p_3

Possible worlds semantics:



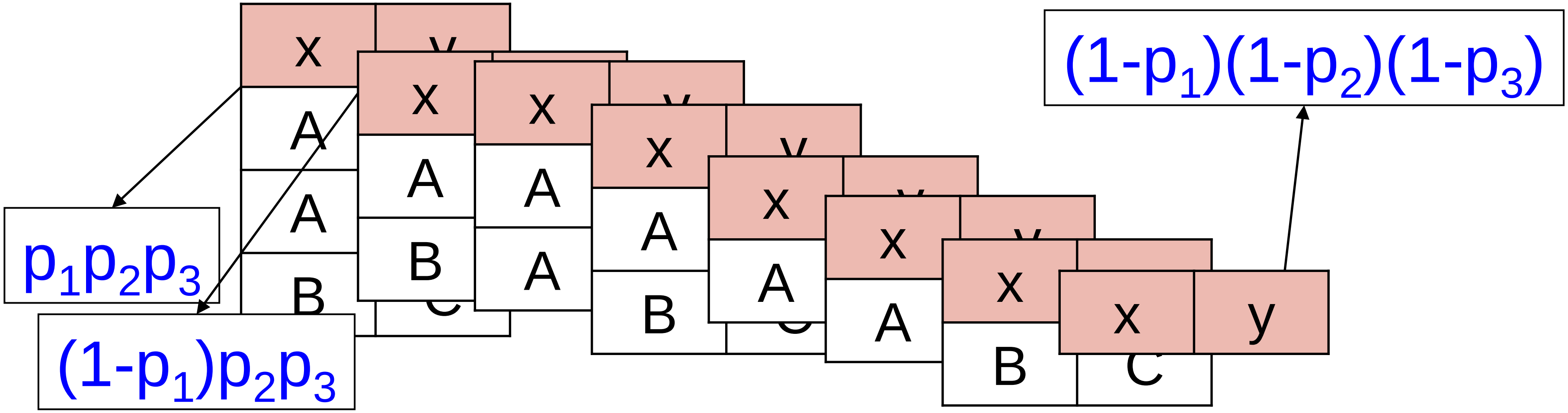
Example

Probabilistic database D:

Friend

x	y	P
A	B	p_1
A	C	p_2
B	C	p_3

Possible worlds semantics:



Query Semantics

Fix a Boolean query Q, probabilistic database D:

$P(Q \mid D) = P_D(Q)$ = marginal probability of Q on possible words of D

$Q = \exists x \exists y \; Smoker(x) \wedge Friend(x,y)$

An Example

$P(Q \mid D) =$

Friend

x	y	P
A	D	q_1
A	E	q_2
B	F	q_3
B	G	q_4
B	H	q_5

Smoker

x	P
A	p_1
B	p_2
C	p_3

$Q = \exists x \exists y \; Smoker(x) \wedge Friend(x,y)$

An Example

$P(Q \mid D) = 1-(1-q_1)*(1-q_2)$

Smoker

x	P
A	p_1
B	p_2
C	p_3

Friend

x	y	P
A	D	q_1
A	E	q_2
B	F	q_3
B	G	q_4
B	H	q_5

$Q = \exists x \exists y \; Smoker(x) \wedge Friend(x,y)$

An Example

$$P(Q \mid D) = p_1 * [\; 1-(1-q_1)*(1-q_2) \;]$$

Friend

x	y	P
A	D	q_1
A	E	q_2
B	F	q_3
B	G	q_4
B	H	q_5

Smoker

x	P
A	p_1
B	p_2
C	p_3

$Q = \exists x \exists y \; Smoker(x) \wedge Friend(x,y)$

An Example

$$P(Q \mid D) = p_1 * [\; 1-(1-q_1)*(1-q_2) \;]$$
$$1-(1-q_3)*(1-q_4)*(1-q_5)$$

Smoker

x	P
A	p_1
B	p_2
C	p_3

Friend

x	y	P
A	D	q_1
A	E	q_2
B	F	q_3
B	G	q_4
B	H	q_5

$Q = \exists x \exists y \; Smoker(x) \wedge Friend(x,y)$

An Example

$$P(Q \mid D) = p_1 * [\; 1-(1-q_1)*(1-q_2) \;] \; p_2 * [\; 1-(1-q_3)*(1-q_4)*(1-q_5) \;]$$

Smoker

x	P
A	p_1
B	p_2
C	p_3

Friend

x	y	P
A	D	q_1
A	E	q_2
B	F	q_3
B	G	q_4
B	H	q_5

$Q = \exists x \exists y \; Smoker(x) \wedge Friend(x,y)$

An Example

$$P(Q \mid D) = 1 - \{1 - p_1 * [\; 1-(1-q_1)*(1-q_2) \;]\} * \{1 - p_2 * [\; 1-(1-q_3)*(1-q_4)*(1-q_5) \;]\}$$

Friend

x	y	P
A	D	q_1
A	E	q_2
B	F	q_3
B	G	q_4
B	H	q_5

Smoker

x	P
A	p_1
B	p_2
C	p_3

$Q = \exists x \exists y\ Smoker(x) \wedge Friend(x,y)$

An Example

$$P(Q \mid D) = 1 - \{1 - p_1 * [\ 1-(1-q_1)*(1-q_2)\ ]\} * \{1 - p_2 * [\ 1-(1-q_3)*(1-q_4)*(1-q_5)\ ]\}$$

One can compute P(Q | D) in PTIME in the size of the database D

Friend

x	y	P
A	D	q_1
A	E	q_2
B	F	q_3
B	G	q_4
B	H	q_5

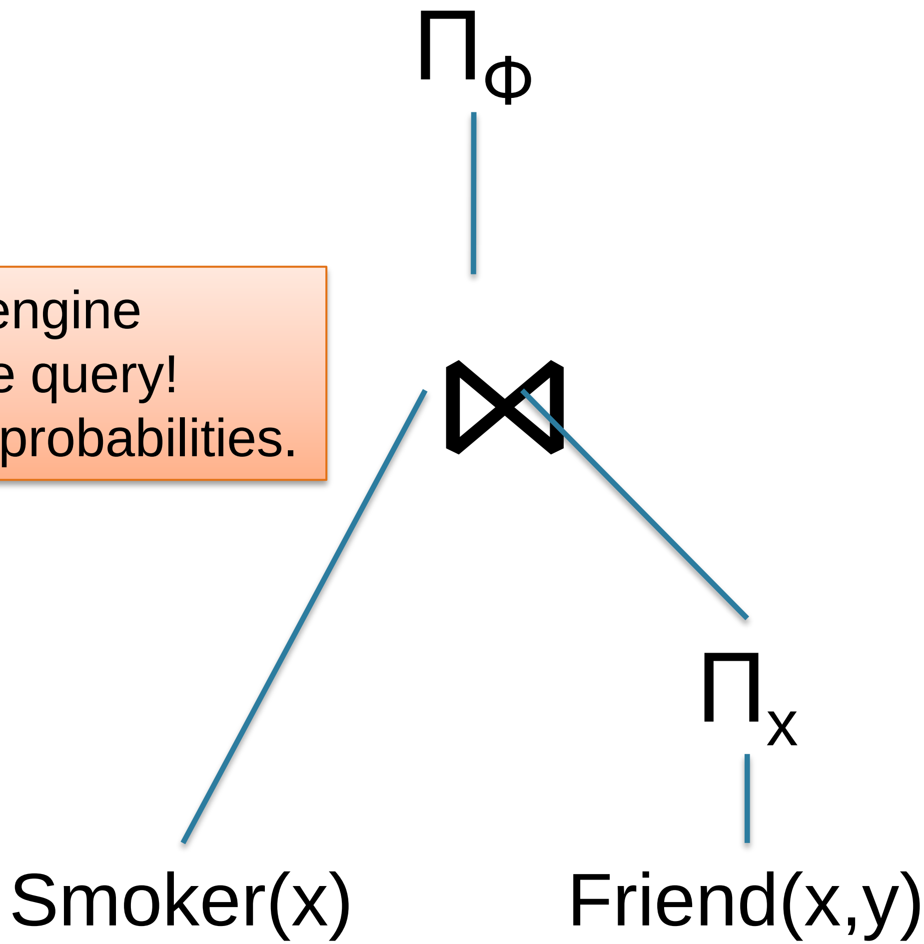
Smoker

x	P
A	p_1
B	p_2
C	p_3

$Q = \exists x \exists y \; Smoker(x) \land Friend(x,y)$

An Example

Use the SQL engine
to compute the query!
Aggregate on probabilities.



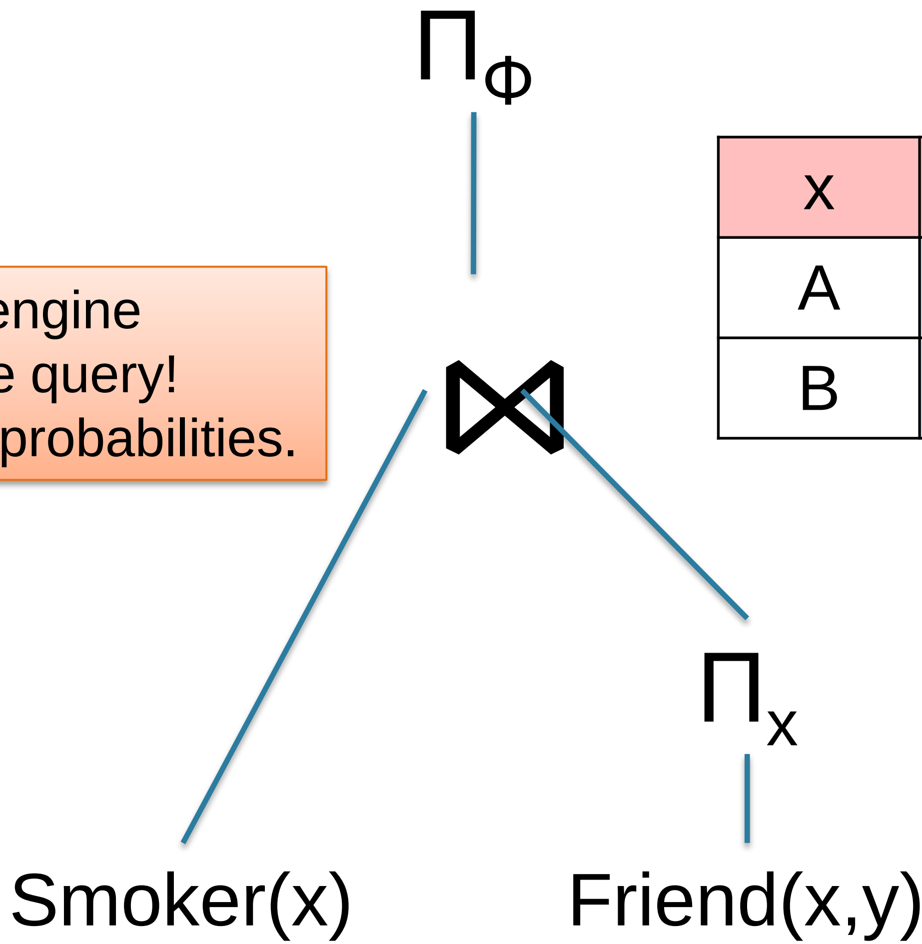
x	P
A	p_1
B	p_2
C	p_3

x	y	P
A	D	q_1
A	E	q_2
B	F	q_3
B	G	q_4
B	H	q_5

$Q = \exists x \exists y \; Smoker(x) \wedge Friend(x,y)$

An Example

Use the SQL engine
to compute the query!
Aggregate on probabilities.



x	P
A	p_1
B	p_2
C	p_3

x	y	P
A	D	q_1
A	E	q_2
B	F	q_3
B	G	q_4
B	H	q_5

x	P
A	$1-(1-q_1)(1-q_2)$
B	$1-(1-q_4)(1-q_5)\ (1-q_6)$

$Q = \exists x \exists y\ Smoker(x) \wedge Friend(x,y)$

An Example

$$1-\{1-p_1[1-(1-q_1)(1-q_2)]\} * \{1-p_2[1-(1-q_4)(1-q_5)(1-q_6)]\}$$

Use the SQL engine
to compute the query!
Aggregate on probabilities.

Π_Φ

$\bowtie$

Π_x

Smoker(x)

Friend(x,y)

x	P
A	$1-(1-q_1)(1-q_2)$
B	$1-(1-q_4)(1-q_5)(1-q_6)$

x	P
A	p_1
B	p_2
C	p_3

x	y	P
A	D	q_1
A	E	q_2
B	F	q_3
B	G	q_4
B	H	q_5

Problem Statement

Given: probabilistic database D, query Q

Compute: P(Q | D)

Data complexity: fix Q, complexity = f(|D|)

Approaches to Compute P(Q | D)

- Propositional inference:
  - Ground the query Q → $F_{Q,D}$, compute P($F_{Q,D}$)
  - This is Weighted Model Counting (later…)
  - Works for every query Q
  - But: may be exponential in |D| (data complexity)

- Lifted inference:
  - Compute a query plan for Q, execute plan on D
  - Always polynomial time in |D| (data complexity)
  - But: does not work for all queries Q

[Olteanu'08, Jha'13, Dalvi'04, Dalvi'12]

Lifted Inference Rules

Preprocess Q (omitted from this talk; see [Suciu'11]), then apply these rules (some have preconditions)

$P(\neg Q) = 1 - P(Q)$ negation

$P(Q1 \wedge Q2) = P(Q1)P(Q2)$
$P(Q1 \vee Q2) = 1 - (1 - P(Q1))(1 - P(Q2))$

Independent join / union

$P(\exists z\ Q) = 1 - \Pi_{A \in Domain} (1 - P(Q[A/z]))$
$P(\forall z\ Q) = \Pi_{A \in Domain} P(Q[A/z])$

Independent project

$P(Q1 \wedge Q2) = P(Q1) + P(Q2) - P(Q1 \vee Q2)$
$P(Q1 \vee Q2) = P(Q1) + P(Q2) - P(Q1 \wedge Q2)$

Inclusion/ exclusion

Example

$Q = \forall x \forall y\, (Smoker(x) \vee Friend(x,y))$ $= \forall x\, (Smoker(x) \vee \forall y\, Friend(x,y))$

∀-Rule

$P(Q) = \prod_{A \in Domain} P(Smoker(A) \vee \forall y\, Friend(A,y))$

Check independence:

$Smoker(Alice) \vee \forall y\, Friend(Alice,y)$

$Smoker(Bob) \vee \forall y\, Friend(Bob,y)$

Example

$Q = \forall x \forall y\, (\text{Smoker}(x) \lor \text{Friend}(x,y))$ $= \forall x\, (\text{Smoker}(x) \lor \forall y\, \text{Friend}(x,y))$

∀-Rule

$P(Q) = \prod_{A \in \text{Domain}} P(\text{Smoker}(A) \lor \forall y\, \text{Friend}(A,y))$

Check independence:

Smoker(Alice) ∨ ∀y Friend(Alice,y)

Smoker(Bob) ∨ ∀y Friend(Bob,y)

$P(Q) = \prod_{A \in \text{Domain}} [1 - (1 - P(\text{Smoker}(A))) \times (1 - P(\forall y\, \text{Friend}(A,y)))]$

∨-Rule

Example

$Q = \forall x \forall y\, (\text{Smoker}(x) \vee \text{Friend}(x,y))$ $= \forall x\, (\text{Smoker}(x) \vee \forall y\, \text{Friend}(x,y))$

∀-Rule

$P(Q) = \prod_{A \in \text{Domain}} P(\text{Smoker}(A) \vee \forall y\, \text{Friend}(A,y))$

Check independence:
$\text{Smoker}(\text{Alice}) \vee \forall y\, \text{Friend}(\text{Alice},y)$
$\text{Smoker}(\text{Bob}) \vee \forall y\, \text{Friend}(\text{Bob},y)$

$P(Q) = \prod_{A \in \text{Domain}} [1 - (1 - P(\text{Smoker}(A))) \times (1 - P(\forall y\, \text{Friend}(A,y)))]$

∨-Rule

$P(Q) = \prod_{A \in \text{Domain}} [1 - (1 - P(\text{Smoker}(A))) \times (1 - \prod_{B \in \text{Domain}} P(\text{Friend}(A,B)))]$

∀-Rule

Example

$Q = \forall x \forall y\, (\text{Smoker}(x) \lor \text{Friend}(x,y))$ $= \forall x\, (\text{Smoker}(x) \lor \forall y\, \text{Friend}(x,y))$

∀-Rule

$P(Q) = \prod_{A \in \text{Domain}} P(\text{Smoker}(A) \lor \forall y\, \text{Friend}(A,y))$

Check independence:
Smoker(Alice) ⋁ ∀y Friend(Alice,y)
Smoker(Bob) ⋁ ∀y Friend(Bob,y)

$P(Q) = \prod_{A \in \text{Domain}} [1 - (1 - P(\text{Smoker}(A))) \times (1 - P(\forall y\, \text{Friend}(A,y)))]$

⋁-Rule

$P(Q) = \prod_{A \in \text{Domain}} [1 - (1 - P(\text{Smoker}(A))) \times (1 - \prod_{B \in \text{Domain}} P(\text{Friend}(A,B)))]$

Lookup the probabilities in the database

∀-Rule

Example

$Q = \forall x \forall y\,(\text{Smoker}(x) \vee \text{Friend}(x,y))$ $= \forall x\,(\text{Smoker}(x) \vee \forall y\,\text{Friend}(x,y))$

∀-Rule

$P(Q) = \prod_{A \in \text{Domain}} P(\text{Smoker}(A) \vee \forall y\,\text{Friend}(A,y))$

Check independence:
Smoker(Alice) ∨ ∀y Friend(Alice,y)
Smoker(Bob) ∨ ∀y Friend(Bob,y)

$P(Q) = \prod_{A \in \text{Domain}} [1 - (1 - P(\text{Smoker}(A))) \times (1 - P(\forall y\,\text{Friend}(A,y)))]$

∨-Rule

$P(Q) = \prod_{A \in \text{Domain}} [1 - (1 - P(\text{Smoker}(A))) \times (1 - \prod_{B \in \text{Domain}} P(\text{Friend}(A,B)))]$

Lookup the probabilities in the database

∀-Rule

Runtime = $O(n^2)$.

Discussion: CNF vs. DNF

Databases		KR/AI	
Conjunctive Queries CQ	FO(∃, ∧)	Positive Clause	FO(∀, ∨)
Union of Conjunctive Queries UCQ	FO(∃, ∧, ∨) = ∃ Positive-DNF	Positive FO	FO(∀, ∧, ∨) = ∀ Positive-CNF
UCQ with "safe negation" UCQ¬	∃ DNF	First Order CNF	∀ CNF
Q = ∃x,∃y, Smoker(x) ∧ Friend(x,y)		Q = ∀x∀y (Smoker(x) ∨ Friend(x,y))	

∃x,∃y, Smoker(x) ∧ Friend(x,y) = ¬ ∀x,∀y, (¬Smoker(x) ∨ ¬Friend(x,y))

Discussion

Lifted Inference Sometimes Fails.

$H_0 = \forall x \forall y\, (\text{Smoker}(x) \vee \text{Friend}(x,y) \vee \text{Jogger}(y))$

The ∀-rule does not apply: H_0[Alice/x] and H_0[Bob/x] are dependent:

$H_0[\text{Alice}/x] = \forall y\, (\text{Smoker}(\text{Alice}) \vee \text{Friend}(\text{Alice},y) \vee \text{Jogger}(y))$

$H_0[\text{Bob}/x] = \forall y\, (\text{Smoker}(\text{Bob}) \vee \text{Friend}(\text{Bob},y) \vee \text{Jogger}(y))$

Dependent

Computing P(H_0 | D) is #P-hard in |D|
(Proof: later…)

Discussion

Lifted Inference Sometimes Fails.

$H_0 = \forall x \forall y\, (\text{Smoker}(x) \lor \text{Friend}(x,y) \lor \text{Jogger}(y))$

The ∀-rule does not apply: H_0[Alice/x] and H_0[Bob/x] are dependent:

$H_0[\text{Alice}/x] = \forall y\, (\text{Smoker}(\text{Alice}) \lor \text{Friend}(\text{Alice},y) \lor \text{Jogger}(y))$

$H_0[\text{Bob}/x] = \forall y\, (\text{Smoker}(\text{Bob}) \lor \text{Friend}(\text{Bob},y) \lor \text{Jogger}(y))$

Dependent

Computing $P(H_0 \mid D)$ is #P-hard in $|D|$
(Proof: later…)

Consequence: assuming PTIME ≠ #P, H_0 is not liftable!

Summary

- Database D = relations
- Query Q = FO
- Query plans, query optimization
- Data complexity: fix Q, complexity f(D)
- Probabilistic DB's = independent tuples
- Lifted inference: simple, but fails sometimes

Outline

- Part 1: Motivation
- Part 2: Probabilistic Databases
- Part 3: Weighted Model Counting
- Part 4: Lifted Inference for WFOMC

- Part 5: Completeness of Lifted Inference
- Part 6: Query Compilation
- Part 7: Symmetric Lifted Inference Complexity
- Part 8: Open-World Probabilistic Databases
- Part 9: Discussion & Conclusions

WMC Probabilistic Inference

- Model = solution to a propositional logic formula Δ
- Model counting = #SAT

Δ = (Rain ⇒ Cloudy)

Rain	Cloudy	Model?
T	T	Yes
T	F	No
F	T	Yes
F	F	Yes

+ ———

#SAT = 3

WMC Probabilistic Inference

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

Δ = (Rain ⇒ Cloudy)

Rain

w(R)	w(¬R)
1	2

Cloudy

w(C)	w(¬C)
3	5

Rain	Cloudy	Model?	Weight
T	T	Yes	1 * 3 = 3
T	F	No	0
F	T	Yes	2 * 3 = 6
F	F	Yes	2 * 5 = 10

+ ———

#SAT = 3

WMC Probabilistic Inference

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

Δ = (Rain ⇒ Cloudy)

Rain

w(R)	w(¬R)
1	2

Cloudy

w(C)	w(¬C)
3	5

Rain	Cloudy	Model?	Weight
T	T	Yes	1 * 3 = 3
T	F	No	0
F	T	Yes	2 * 3 = 6
F	F	Yes	2 * 5 = 10

#SAT = 3 **WMC = 19**

Weighted Model Counting

- Assembly language for non-lifted inference
- Reductions to WMC for inference in
  - Bayesian networks [Chavira'05, Sang'05 , Chavira'08]
  - Factor graphs [Choi'13]
  - Relational Bayesian networks [Chavira'06]
  - Probabilistic logic programs [Fierens'11, Fierens'15]
  - Probabilistic databases [Olteanu'08, Jha'11]
- State-of-the-art exact solvers
  - Knowledge compilation (WMC → d-DNNF → AC)

    Winner of the UAI'08 exact inference competition!
  - DPLL counters

Weighted First-Order Model Counting

Model = solution to first-order logic formula Δ

Δ = ∀d (Rain(d)
⇒ Cloudy(d))

Days = {Monday}

[VdB'11, Gogate'11]

Weighted First-Order Model Counting

Model = solution to first-order logic formula Δ

Δ = ∀d (Rain(d) ⇒ Cloudy(d))

Days = {Monday}

Rain(M)	Cloudy(M)	Model?
T	T	Yes
T	F	No
F	T	Yes
F	F	Yes

+

#SAT = 3

[VdB'11, Gogate'11]

Weighted First-Order Model Counting

Model = solution to first-order logic formula Δ

Δ = ∀d (Rain(d) ⇒ Cloudy(d))

Days = {Monday **Tuesday**}

Rain(M)	Cloudy(M)	Rain(T)	Cloudy(T)	Model?
T	T	T	T	Yes
T	F	T	T	No
F	T	T	T	Yes
F	F	T	T	Yes
T	T	T	F	No
T	F	T	F	No
F	T	T	F	No
F	F	T	F	No
T	T	F	T	Yes
T	F	F	T	No
F	T	F	T	Yes
F	F	F	T	Yes
T	T	F	F	Yes
T	F	F	F	No
F	T	F	F	Yes
F	F	F	F	Yes

[VdB'11, Gogate'11]

Weighted First-Order Model Counting

Model = solution to first-order logic formula Δ

Δ = ∀d (Rain(d) ⇒ Cloudy(d))

Days = {Monday Tuesday}

Rain(M)	Cloudy(M)	Rain(T)	Cloudy(T)	Model?
T	T	T	T	Yes
T	F	T	T	No
F	T	T	T	Yes
F	F	T	T	Yes
T	T	T	F	No
T	F	T	F	No
F	T	T	F	No
F	F	T	F	No
T	T	F	T	Yes
T	F	F	T	No
F	T	F	T	Yes
F	F	F	T	Yes
T	T	F	F	Yes
T	F	F	F	No
F	T	F	F	Yes
F	F	F	F	Yes

+ ———

#SAT = 9

[VdB'11, Gogate'11]

Weighted First-Order Model Counting

Model = solution to first-order logic formula Δ

Δ = ∀d (Rain(d) ⇒ Cloudy(d))

Days = {Monday Tuesday}

Rain

d	w(R(d))	w(¬R(d))
M	1	2
T	4	1

Cloudy

d	w(C(d))	w(¬C(d))
M	3	5
T	6	2

Rain(M)	Cloudy(M)	Rain(T)	Cloudy(T)	Model?
T	T	T	T	Yes
T	F	T	T	No
F	T	T	T	Yes
F	F	T	T	Yes
T	T	T	F	No
T	F	T	F	No
F	T	T	F	No
F	F	T	F	No
T	T	F	T	Yes
T	F	F	T	No
F	T	F	T	Yes
F	F	F	T	Yes
T	T	F	F	Yes
T	F	F	F	No
F	T	F	F	Yes
F	F	F	F	Yes

+

#SAT = 9

[VdB'11, Gogate'11]

Weighted First-Order Model Counting

Model = solution to first-order logic formula Δ

Δ = ∀d (Rain(d) ⇒ Cloudy(d))

Days = {Monday, Tuesday}

Rain

d	w(R(d))	w(¬R(d))
M	1	2
T	4	1

Cloudy

d	w(C(d))	w(¬C(d))
M	3	5
T	6	2

Rain(M)	Cloudy(M)	Rain(T)	Cloudy(T)	Model?	Weight
T	T	T	T	Yes	1 * 3 * 4 * 6 = 72
T	F	T	T	No	0
F	T	T	T	Yes	2 * 3 * 4 * 6 = 144
F	F	T	T	Yes	2 * 5 * 4 * 6 = 240
T	T	T	F	No	0
T	F	T	F	No	0
F	T	T	F	No	0
F	F	T	F	No	0
T	T	F	T	Yes	1 * 3 * 1 * 6 = 18
T	F	F	T	No	0
F	T	F	T	Yes	2 * 3 * 1 * 6 = 36
F	F	F	T	Yes	2 * 5 * 1 * 6 = 60
T	T	F	F	Yes	1 * 3 * 1 * 2 = 6
T	F	F	F	No	0
F	T	F	F	Yes	2 * 3 * 1 * 2 = 12
F	F	F	F	Yes	2 * 5 * 1 * 2 = 20

+

#SAT = 9

[VdB'11, Gogate'11]

Weighted First-Order Model Counting

Model = solution to first-order logic formula Δ

Δ = ∀d (Rain(d) ⇒ Cloudy(d))

Days = {Monday **Tuesday**}

Rain

d	w(R(d))	w(¬R(d))
M	1	2
T	4	1

Cloudy

d	w(C(d))	w(¬C(d))
M	3	5
T	6	2

Rain(M)	**Cloudy(M)**	**Rain(T)**	**Cloudy(T)**	**Model?**	**Weight**
T	T	T	T	Yes	1 * 3 * 4 * 6 = 72
T	F	T	T	No	0
F	T	T	T	Yes	2 * 3 * 4 * 6 = 144
F	F	T	T	Yes	2 * 5 * 4 * 6 = 240
T	T	T	F	No	0
T	F	T	F	No	0
F	T	T	F	No	0
F	F	T	F	No	0
T	T	F	T	Yes	1 * 3 * 1 * 6 = 18
T	F	F	T	No	0
F	T	F	T	Yes	2 * 3 * 1 * 6 = 36
F	F	F	T	Yes	2 * 5 * 1 * 6 = 60
T	T	F	F	Yes	1 * 3 * 1 * 2 = 6
T	F	F	F	No	0
F	T	F	F	Yes	2 * 3 * 1 * 2 = 12
F	F	F	F	Yes	2 * 5 * 1 * 2 = 20

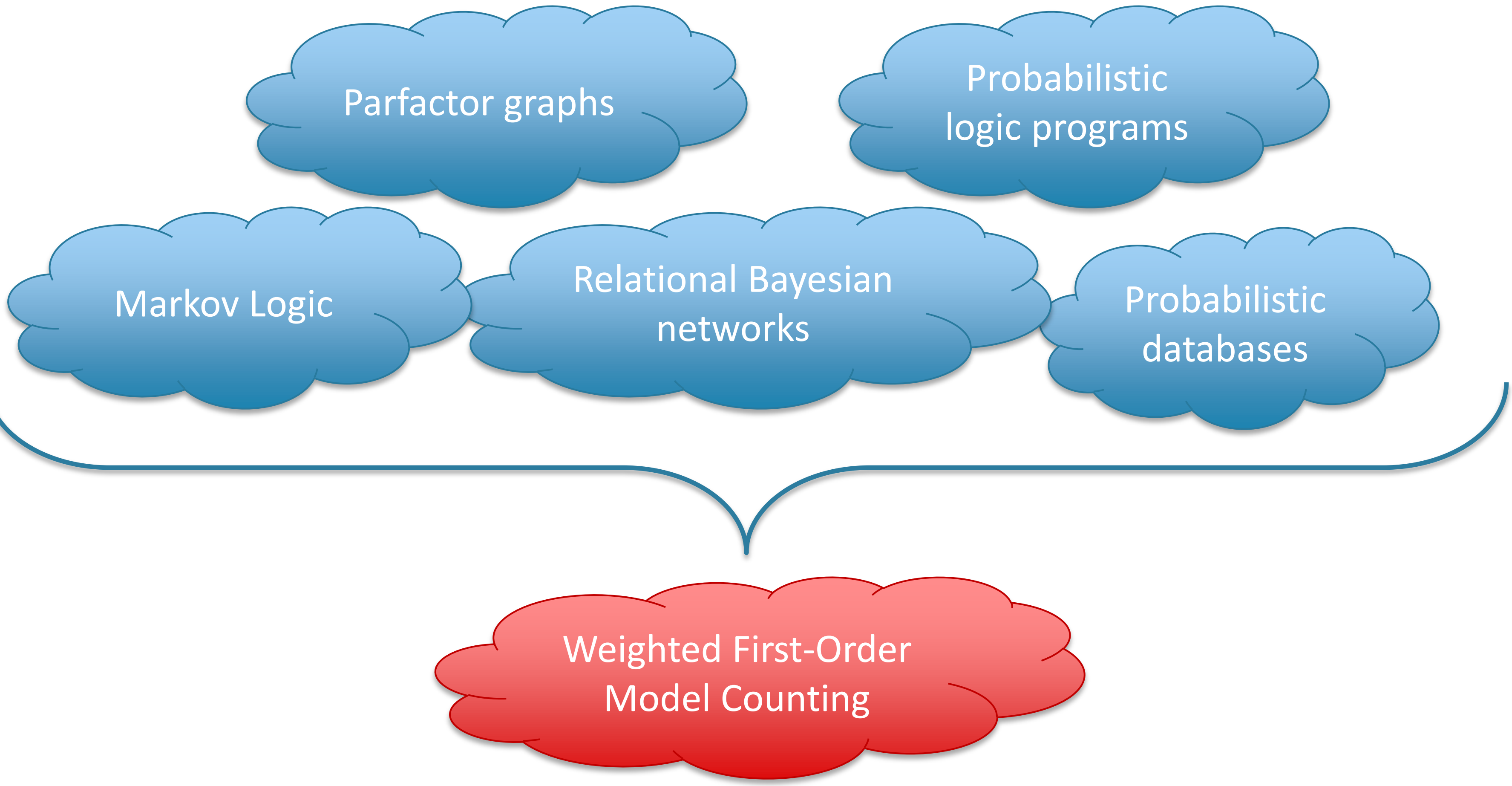
#SAT = 9 **WFOMC = 608**

[VdB'11, Gogate'11]

WFOMC Probabilistic Inference

- Assembly language for lifted inference
- Reduction to WFOMC for lifted inference in
  - Markov logic networks [VdB'11,Gogate'11]
  - Parfactor graphs [VdB'13]
  - Probabilistic logic programs [VdB'14]
  - Probabilistic databases [Gribkoff'14]

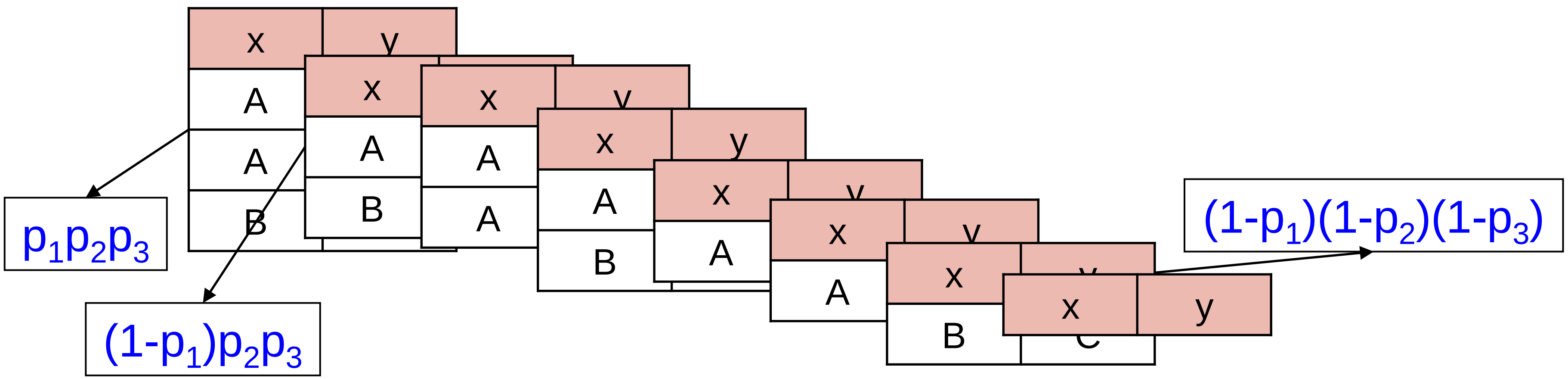
Assembly language for high-level probabilistic reasoning



From Probabilities to Weights

Friend

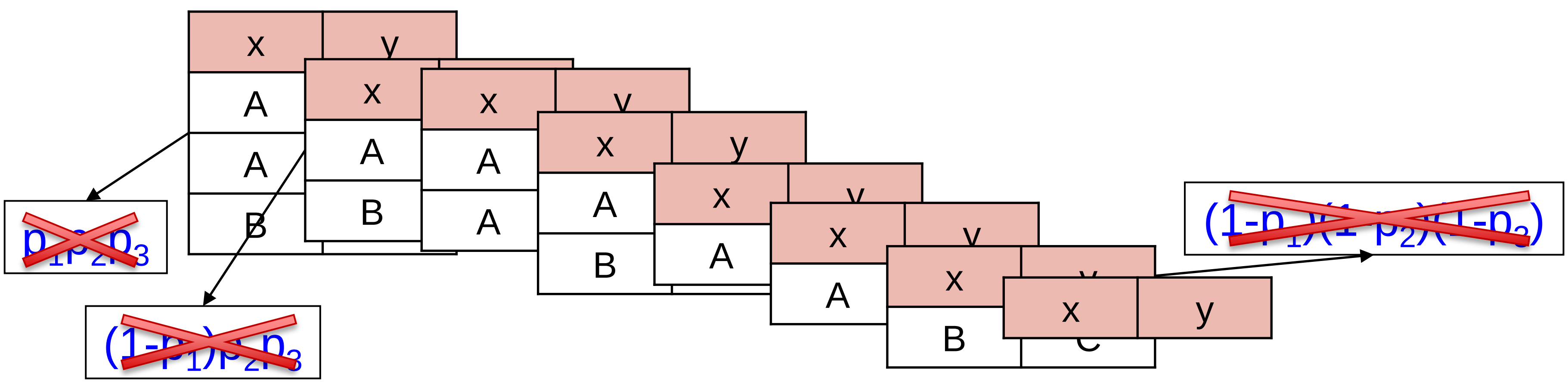
x	y	P
A	B	p_1
A	C	p_2
B	C	p_3



From Probabilities to Weights

Friend

x	y	P
A	B	~~p_1~~
A	C	~~p_2~~
B	C	~~p_3~~



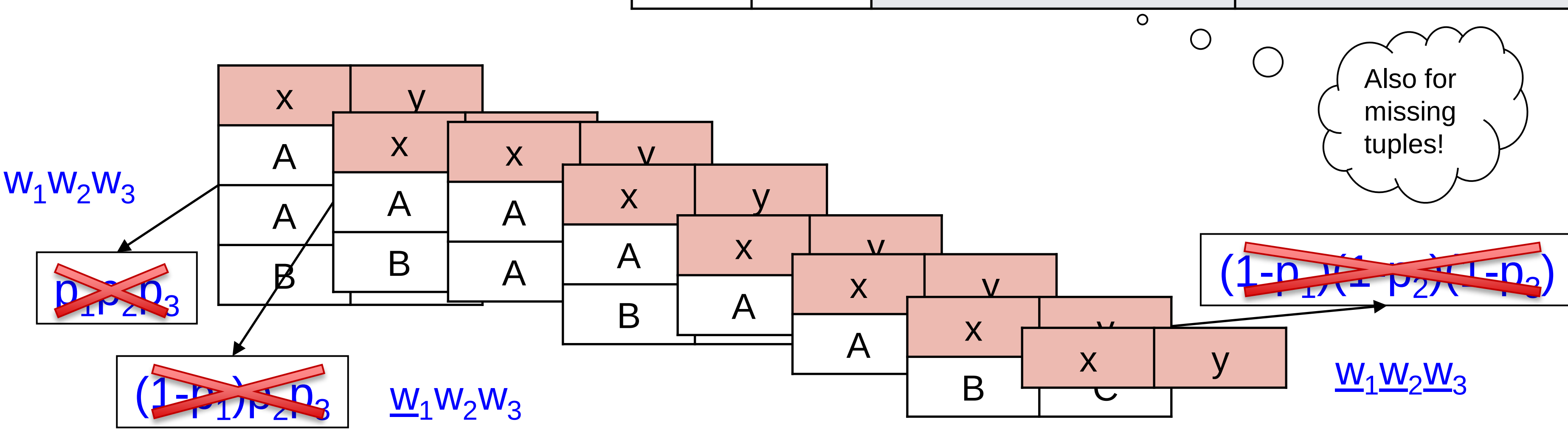
From Probabilities to Weights

Friend

x	y	P
A	B	~~p_1~~
A	C	~~p_2~~
B	C	~~p_3~~

→

x	y	w(Friend(x,y))	w(¬Friend(x,y))
A	B	$w_1 = p_1$	$\underline{w}_1 = 1-p_1$
A	C	$w_2 = p_2$	$\underline{w}_2 = 1-p_2$
B	C	$w_3 = p_3$	$\underline{w}_3 = 1-p_3$
A	A	$w_4 = 0$	$\underline{w}_4 = 1$
A	C	$w_5 = 0$	$\underline{w}_5 = 1$
	…	…	



Discussion

- Simple idea: replace p, $1-p$ by w, $\underline{w}$
- Query computation becomes WFOMC

- To obtain a probability space, divide the weight of each world by Z = sum of weights of all worlds:

  $Z = (w_1+\underline{w}_1)\ (w_2+\underline{w}_2)\ (w_3+\underline{w}_3)\ \ldots$

- Why weights instead of probabilities?
  They can describe complex correlations (next)

Markov Logic

Capture knowledge through soft constraints (a.k.a. “features”):

Hard constraint

∞ Smoker(x) $\Rightarrow$ Person(x)

Soft constraint,
weight = exp(3.73)

3.75 Smoker(x) $\wedge$ Friend(x,y) $\Rightarrow$ Smoker(y)

[Richardson’06]

Markov Logic

Capture knowledge through soft constraints (a.k.a. "features"):

Hard constraint

∞ Smoker(x) ⇒ Person(x)

Soft constraint,
weight = exp(3.73)

3.75 Smoker(x) ∧ Friend(x,y) ⇒ Smoker(y)

An MLN is a set of constraints (w, $\Gamma(\mathbf{x})$), where w=weight, $\Gamma(\mathbf{x})$=FO formula

[Richardson'06]

Markov Logic

Capture knowledge through soft constraints (a.k.a. “features”):

Hard constraint

∞	Smoker(x) ⇒ Person(x)

Soft constraint, weight = exp(3.73)

3.75	Smoker(x) ∧ Friend(x,y) ⇒ Smoker(y)

An MLN is a set of constraints (w, $\Gamma(\mathbf{x})$), where w=weight, $\Gamma(\mathbf{x})$=FO formula

Weight of a world = product of exp(w), for all MLN rules (w, $\Gamma(\mathbf{x})$) and grounding $\Gamma(\mathbf{a})$ that hold in that world

[Richardson’06]

Markov Logic

Capture knowledge through soft constraints (a.k.a. "features"):

Hard constraint

∞	Smoker(x) ⇒ Person(x)

Soft constraint, weight = exp(3.73)

3.75	Smoker(x) ∧ Friend(x,y) ⇒ Smoker(y)

An MLN is a set of constraints (w, $\Gamma(\mathbf{x})$), where w=weight, $\Gamma(\mathbf{x})$=FO formula

Weight of a world = product of exp(w), for all MLN rules (w, $\Gamma(\mathbf{x})$) and grounding $\Gamma(\mathbf{a})$ that hold in that world

Probability of a world = Weight / Z

Z = sum of weights of all worlds (no longer a simple expression!)

[Richardson'06]

Discussion

- Probabilistic databases = independence
  MLN = complex correlations

- To translate weights to probabilities we need to divide by Z, which often is difficult to compute

- However, we can reduce the Z-computation problem to WFOMC (next)

Z → WFOMC(Δ)

1. **<u>Formula Δ</u>**

2. **<u>Weight function w(.)</u>**

Z → WFOMC(Δ)

1. **<u>Formula Δ</u>**

If all MLN constraints are hard: $\Delta = \bigwedge_{(\infty,\Gamma(\mathbf{x}))\in MLN} (\forall \mathbf{x}\ \Gamma(\mathbf{x}))$

2. **<u>Weight function w(.)</u>**

$Z \rightarrow \text{WFOMC}(\Delta)$

1. Formula Δ

If all MLN constraints are hard: $\Delta = \bigwedge_{(\infty, \Gamma(\mathbf{x})) \in MLN} (\forall \mathbf{x}\ \Gamma(\mathbf{x}))$

If $(w_i, \Gamma_i(\mathbf{x}))$ is a soft MLN constraint, then:

a) Remove $(w_i, \Gamma_i(\mathbf{x}))$ from the MLN

b) Add new probabilistic relation $F_i(\mathbf{x})$

c) Add hard constraint $(\infty, \forall \mathbf{x}\ (F_i(\mathbf{x}) \Leftrightarrow \Gamma_i(\mathbf{x})))$

2. Weight function w(.)

$Z \rightarrow \mathrm{WFOMC}(\Delta)$

1. Formula Δ

If all MLN constraints are hard: $\Delta = \bigwedge_{(\infty,\Gamma(\mathbf{x}))\in MLN} (\forall \mathbf{x}\ \Gamma(\mathbf{x}))$

If $(w_i, \Gamma_i(\mathbf{x}))$ is a soft MLN constraint, then:

a) Remove $(w_i, \Gamma_i(\mathbf{x}))$ from the MLN
b) Add new probabilistic relation $F_i(\mathbf{x})$
c) Add hard constraint $(\infty, \forall \mathbf{x}\ (F_i(\mathbf{x}) \Leftrightarrow \Gamma_i(\mathbf{x})))$

2. Weight function w(.)

For all constants **A**, relations F_i,
set $w(F_i(\mathbf{A})) = \exp(w_i)$, $w(\neg F_i(\mathbf{A})) = 1$

Better rewritings in
[Jha'12],[V.d.Broeck'14]

$Z \rightarrow \mathrm{WFOMC}(\Delta)$

1. Formula Δ

If all MLN constraints are hard: $\Delta = \bigwedge_{(\infty, \Gamma(\mathbf{x})) \in MLN} (\forall \mathbf{x}\ \Gamma(\mathbf{x}))$

If $(w_i, \Gamma_i(\mathbf{x}))$ is a soft MLN constraint, then:

a) Remove $(w_i, \Gamma_i(\mathbf{x}))$ from the MLN
b) Add new probabilistic relation $F_i(\mathbf{x})$
c) Add hard constraint $(\infty, \forall \mathbf{x}\ (F_i(\mathbf{x}) \Leftrightarrow \Gamma_i(\mathbf{x})))$

2. Weight function w(.)

For all constants **A**, relations F_i,
set $w(F_i(\mathbf{A})) = \exp(w_i)$, $w(\neg F_i(\mathbf{A})) = 1$

Theorem: $Z = \mathrm{WFOMC}(\Delta)$

Better rewritings in [Jha’12],[V.d.Broeck’14]

Example

1. **Formula Δ**

2. **Weight function w(.)**

Example

1. **<u>Formula Δ</u>**

∞	Smoker(x) ⇒ Person(x)

2. **<u>Weight function w(.)</u>**

Example

1. **Formula Δ**

∞ Smoker(x) ⇒ Person(x)

$\Delta = \forall x\,(\text{Smoker}(x) \Rightarrow \text{Person}(x))$

2. **Weight function w(.)**

Example

1. **<u>Formula Δ</u>**

∞	Smoker(x) ⇒ Person(x)
3.75	Smoker(x) ∧ Friend(x,y) ⇒ Smoker(y)

$\Delta = \forall x\ (\text{Smoker}(x) \Rightarrow \text{Person}(x))$

2. **<u>Weight function w(.)</u>**

Example

1. **<u>Formula Δ</u>**

∞	Smoker(x) ⇒ Person(x)
3.75	Smoker(x) ∧ Friend(x,y) ⇒ Smoker(y)

$$\Delta = \forall x\,(\text{Smoker}(x) \Rightarrow \text{Person}(x))$$
$$\wedge\ \forall x \forall y\,(F(x,y) \Leftrightarrow [\text{Smoker}(x) \wedge \text{Friend}(x,y) \Rightarrow \text{Smoker}(y)])$$

2. **<u>Weight function w(.)</u>**

Example

1. Formula Δ

∞	Smoker(x) ⇒ Person(x)
3.75	Smoker(x) ∧ Friend(x,y) ⇒ Smoker(y)

$$\Delta = \forall x\,(\text{Smoker}(x) \Rightarrow \text{Person}(x))$$
$$\wedge\ \forall x \forall y\,(F(x,y) \Leftrightarrow [\text{Smoker}(x) \wedge \text{Friend}(x,y) \Rightarrow \text{Smoker}(y)])$$

2. Weight function w(.)

F

x	y	w(F(x,y))	w(¬F(x,y))
A	A	exp(3.75)	1
A	B	exp(3.75)	1
A	C	exp(3.75)	1
B	A	exp(3.75)	1
	. . .	. . .	

Note: if no tables given for Smoker, Person, etc, (i.e. no evidence) then set their w = <u>w</u> = 1

Example

1. Formula Δ

∞	Smoker(x) ⇒ Person(x)
3.75	Smoker(x) ∧ Friend(x,y) ⇒ Smoker(y)

Δ = ∀x (Smoker(x) ⇒ Person(x))
∧ ∀x∀y (F(x,y) ⇔ [Smoker(x) ∧ Friend(x,y) ⇒ Smoker(y)])

2. Weight function w(.)

F

x	y	w(F(x,y))	w(¬F(x,y))
A	A	exp(3.75)	1
A	B	exp(3.75)	1
A	C	exp(3.75)	1
B	A	exp(3.75)	1
	. . .	. . .	

Note: if no tables given for Smoker, Person, etc, (i.e. no evidence) then set their w = w = 1

Z = WFOMC(Δ)

Lessons

- Weighed Model Counting:
  - Unified framework for probabilistic inference tasks
  - Independent variables
- Weighed FO Model Counting:
  - Formula described by a concise FO sentence
  - Still independent variables
- MLNs:
  - Weighted formulas
  - Correlations!
  - Can be converted to WFOMC

Lessons

- Weighed Model Counting:
  - Unified framework for probabilistic inference tasks
  - Independent variables
- Weighed FO Model Counting:
  - Formula described by a concise FO sentence
  - Still independent variables
- MLNs:
  - Weighted formulas
  - Correlations!
  - Can be converted to WFOMC

Tuple-independence is not a severe representational restriction!
It is a convenience for building inference algorithms.

Symmetric vs. Asymmetric

Symmetric WFOMC:

- In every relation R, all tuples have same weight
- Example: converting MLN "without evidence" into WFOMC leads to a symmetric weight function

F

x	y	w(F(x,y))	w(¬F(x,y))
A	A	exp(3.75)	1
A	B	exp(3.75)	1
A	C	exp(3.75)	1
B	A	exp(3.75)	1
	. . .	. . .	

Asymmetric WFOMC:

- Each relation R is given explicitly
- Example: Probabilistic Databases
- Example: MLN's plus evidence

[Gribkoff'14]

Comparison

	MLNs	Prob. DBs
Random variable is a	Ground atom	DB Tuple
Weights w associated with	Formulas	DB Tuples
Typical query Q is a	Single atom	FO formula/SQL
Data is encoded into	Evidence (Query)	Distribution
Correlations induced by	Model formulas	Query
Model generalizes across domains?	Yes	No
Query generalizes across domains?	No	Yes
Sum of weights of worlds is 1 (normalized)?	No	Yes

Outline

- Part 1: Motivation
- Part 2: Probabilistic Databases
- Part 3: Weighted Model Counting
- Part 4: Lifted Inference for WFOMC
- Part 5: Completeness of Lifted Inference
- Part 6: Query Compilation
- Part 7: Symmetric Lifted Inference Complexity
- Part 8: Open-World Probabilistic Databases
- Part 9: Discussion & Conclusions

Defining Lifted Inference

- Informal:

  Exploit symmetries, Reason at first-order level, Reason about groups of objects, Scalable inference, High-level probabilistic reasoning, etc.

- A formal definition: **Domain-lifted inference**

> Inference runs in time polynomial
> in the number of objects in the domain.

  - Polynomial in #people, #webpages, #cards
  - Not polynomial in #predicates, #formulas, #logical variables
  - Related to data complexity in databases

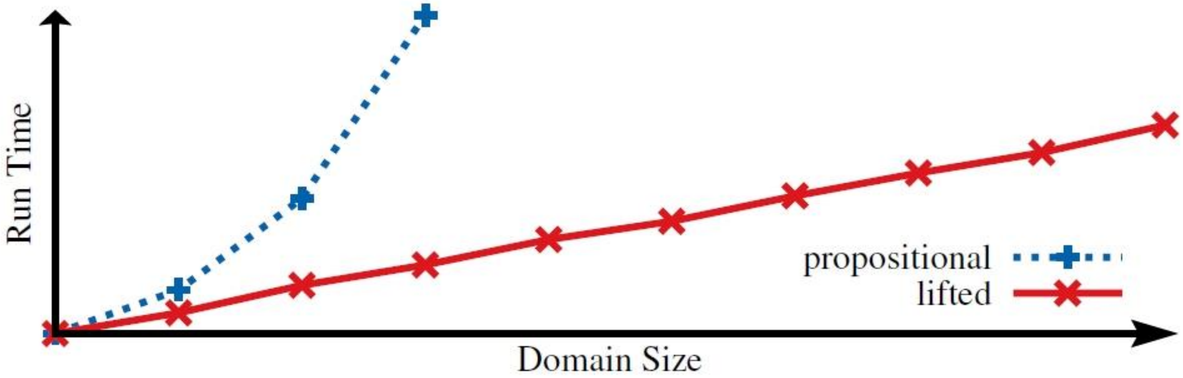
[VdB'11, Jaeger'12]

Defining Lifted Inference

- Informal:

  Exploit symmetries, Reason at first-order level, Reason about groups of objects, Scalable inference, High-level probabilistic reasoning, etc. [Poole'03, etc.]

- A formal definition: **Domain-lifted inference**



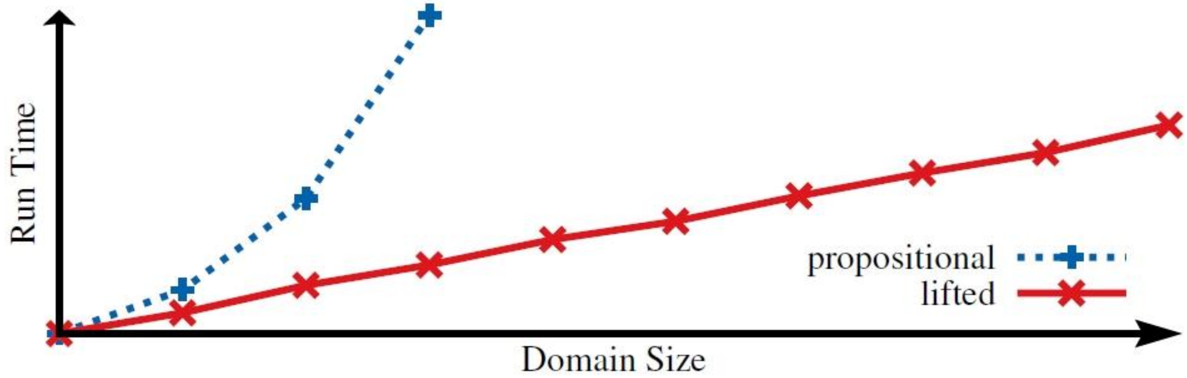
[VdB'11, Jaeger'12]

Defining Lifted Inference

- Informal:

  Exploit symmetries, Reason at first-order level, Reason about groups of objects, Scalable inference, High-level probabilistic reasoning, etc. [Poole'03, etc.]

- A formal definition: **Domain-lifted inference**



- Alternative in this tutorial:

Lifted inference = ∃Query Plan = ∃FO Compilation

[VdB'11, Jaeger'12]

Asymmetric WFOMC Rules

Preprocess Q (omitted from this talk; see [Suciu'11]), then apply these rules (some have preconditions)

WMC(¬Δ) = Z-WMC(Δ)

Negation

Normalization constant Z
(easy to compute)

Asymmetric WFOMC Rules

Preprocess Q (omitted from this talk; see [Suciu'11]), then apply these rules (some have preconditions)

$$\mathrm{WMC}(\neg\Delta) = Z - \mathrm{WMC}(\Delta)$$

Negation

Normalization constant Z (easy to compute)

$$\mathrm{WMC}(\Delta_1 \wedge \Delta_2) = \mathrm{WMC}(\Delta_1) * \mathrm{WMC}(\Delta_2)$$
$$\mathrm{WMC}(\Delta_1 \vee \Delta_2) = Z - (Z_1 - \mathrm{WMC}(\Delta_1)) * (Z_2 - \mathrm{WMC}(\Delta_2))$$

Independent join / union

Asymmetric WFOMC Rules

Preprocess Q (omitted from this talk; see [Suciu'11]), then apply these rules (some have preconditions)

$$WMC(\neg\Delta) = Z - WMC(\Delta)$$

Negation

Normalization constant Z (easy to compute)

$$WMC(\Delta_1 \wedge \Delta_2) = WMC(\Delta_1) * WMC(\Delta_2)$$
$$WMC(\Delta_1 \vee \Delta_2) = Z - (Z_1 - WMC(\Delta_1))*(Z_2 - WMC(\Delta_2))$$

Independent join / union

$$WMC(\exists z\ \Delta) = Z - \Pi_{C \in Domain} (Z_C - WMC(\Delta[C/z]))$$
$$WMC(\forall z\ \Delta) = \Pi_{C \in Domain} WMC(\Delta[C/z])$$

Independent project

Asymmetric WFOMC Rules

Preprocess Q (omitted from this talk; see [Suciu'11]), then apply these rules (some have preconditions)

$$WMC(\neg\Delta) = Z\text{-}WMC(\Delta)$$

Negation

Normalization constant Z (easy to compute)

$$WMC(\Delta_1 \wedge \Delta_2) = WMC(\Delta_1) * WMC(\Delta_2)$$
$$WMC(\Delta_1 \vee \Delta_2) = Z - (Z_1 - WMC(\Delta_1)) * (Z_2 - WMC(\Delta_2))$$

Independent join / union

$$WMC(\exists z\ \Delta) = Z - \Pi_{C \in Domain} (Z_C - WMC(\Delta[C/z]))$$
$$WMC(\forall z\ \Delta) = \Pi_{C \in Domain}\ WMC(\Delta[C/z])$$

Independent project

$$WMC(\Delta_1 \wedge \Delta_2) = WMC(\Delta_1) + WMC(\Delta_2) - WMC(\Delta_1 \vee \Delta_2)$$
$$WMC(\Delta_1 \vee \Delta_2) = WMC(\Delta_1) + WMC(\Delta_2) - WMC(\Delta_1 \wedge \Delta_2)$$

Inclusion/ exclusion

Symmetric WFOMC Rules

- Simplification to *independent project*:

If $\Delta[C_1/x]$, $\Delta[C_2/x]$, … are independent

$$\mathrm{WMC}(\exists z\ \Delta) = Z - (Z_{C_1} - \mathrm{WMC}(\Delta[C_1/z])^{|\mathrm{Domain}|}$$

$$\mathrm{WMC}(\forall z\ \Delta) = \mathrm{WMC}(\Delta[C_1/z])^{|\mathrm{Domain}|}$$

[VdB'11]

Symmetric WFOMC Rules

- Simplification to *independent project*:

  If $\Delta[C_1/x]$, $\Delta[C_2/x]$, … are independent

  $\mathrm{WMC}(\exists z\ \Delta) = Z - (Z_{C_1} - \mathrm{WMC}(\Delta[C_1/z])^{|\mathrm{Domain}|}$

  $\mathrm{WMC}(\forall z\ \Delta) = \mathrm{WMC}(\Delta[C_1/z])^{|\mathrm{Domain}|}$

- A powerful new inference rule: *atom counting*
  Only possible with symmetric weights
  Intuition: **Remove unary relations**

The workhorse of
Symmetric WFOMC

[VdB’11]

WFOMC Inference: Example

- FO-Model Counting: $w(R) = w(\neg R) = 1$
- Apply inference rules backwards (step 4-3-2-1)

WFOMC Inference: Example

- FO-Model Counting: w(R) = w(¬R) = 1
- Apply inference rules backwards (step 4-3-2-1)

---

4. Δ = (Stress(Alice) ⇒ Smokes(Alice))

Domain = {Alice}

WFOMC Inference: Example

- FO-Model Counting: $w(R) = w(\neg R) = 1$
- Apply inference rules backwards (step 4-3-2-1)

---

4. Δ = (Stress(Alice) ⇒ Smokes(Alice))

Domain = {Alice}

→ 3 models

WFOMC Inference: Example

- FO-Model Counting: $w(R) = w(\neg R) = 1$
- Apply inference rules backwards (step 4-3-2-1)

---

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

Domain = {Alice}

$$\text{WMC}(\neg\text{Stress(Alice)} \lor \text{Smokes(Alice)}) =$$
$$= Z - \text{WMC}(\text{Stress(Alice)}) \times \text{WMC}(\neg\text{Smokes(Alice)})$$
$$= 4 - 1 \times 1 = 3 \text{ models}$$

WFOMC Inference: Example

- FO-Model Counting: w(R) = w(¬R) = 1
- Apply inference rules backwards (step 4-3-2-1)

---

4. Δ = (Stress(Alice) ⇒ Smokes(Alice)) | Domain = {Alice}

WMC(¬Stress(Alice) ∨ Smokes(Alice))) =
= Z – WMC(Stress(Alice)) × WMC(¬Smokes(Alice))
= 4 – 1 × 1 = 3 models

---

3. Δ = ∀x, (Stress(x) ⇒ Smokes(x)) | Domain = {n people}

WFOMC Inference: Example

- FO-Model Counting: w(R) = w(¬R) = 1
- Apply inference rules backwards (step 4-3-2-1)

---

4. Δ = (Stress(Alice) ⇒ Smokes(Alice))

Domain = {Alice}

WMC(¬Stress(Alice) ∨ Smokes(Alice))) =
= Z – WMC(Stress(Alice)) × WMC(¬Smokes(Alice))
= 4 – 1 × 1 = 3 models

---

3. Δ = ∀x, (Stress(x) ⇒ Smokes(x))

Domain = {n people}

→ 3^n models

WFOMC Inference: Example

3. $\Delta = \forall x, (Stress(x) \Rightarrow Smokes(x))$

Domain = {n people}

→ 3^n models

WFOMC Inference: Example

3. Δ = ∀x, (Stress(x) ⇒ Smokes(x))

Domain = {n people}

→ 3^n models

2. Δ = ∀y, (ParentOf(y) ∧ Female ⇒ MotherOf(y))

D = {n people}

WFOMC Inference: Example

3. $\Delta = \forall x, (Stress(x) \Rightarrow Smokes(x))$

Domain = {n people}

→ 3^n models

2. $\Delta = \forall y, (ParentOf(y) \wedge Female \Rightarrow MotherOf(y))$

D = {n people}

If Female = true? $\Delta = \forall y, (ParentOf(y) \Rightarrow MotherOf(y))$ → 3^n models

WFOMC Inference: Example

3. $\Delta = \forall x, (Stress(x) \Rightarrow Smokes(x))$

Domain = {n people}

→ 3^n models

2. $\Delta = \forall y, (ParentOf(y) \land Female \Rightarrow MotherOf(y))$

D = {n people}

If Female = true? $\Delta = \forall y, (ParentOf(y) \Rightarrow MotherOf(y))$ → 3^n models

If Female = false? $\Delta = true$ → 4^n models

WFOMC Inference: Example

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

Domain = {n people}

→ 3^n models

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

D = {n people}

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

If Female = false? Δ = true → 4^n models

→ $3^n + 4^n$ models

WFOMC Inference: Example

3. Δ = ∀x, (Stress(x) ⇒ Smokes(x))

Domain = {n people}

→ 3^n models

2. Δ = ∀y, (ParentOf(y) ∧ Female ⇒ MotherOf(y))

D = {n people}

WMC(Δ) = WMC(¬ Female ∨ ∀y, (ParentOf(y) ⇒ MotherOf(y)))

$= 2 * 2^n * 2^n - (2 – 1) * (2^n * 2^n –$ WMC(∀y, (ParentOf(y) ⇒ MotherOf(y))))

$= 2 * 4^n – (4^n – 3^n)$

→ $3^n + 4^n$ models

WFOMC Inference: Example

3. $\Delta = \forall x, (Stress(x) \Rightarrow Smokes(x))$

Domain = {n people}

$\rightarrow 3^n$ models

2. $\Delta = \forall y, (ParentOf(y) \land Female \Rightarrow MotherOf(y))$

D = {n people}

$$WMC(\Delta) = WMC(\neg Female \lor \forall y, (ParentOf(y) \Rightarrow MotherOf(y)))$$
$$= 2 * 2^n * 2^n - (2 - 1) * (2^n * 2^n - WMC(\forall y, (ParentOf(y) \Rightarrow MotherOf(y))))$$
$$= 2 * 4^n - (4^n - 3^n)$$

$\rightarrow 3^n + 4^n$ models

1. $\Delta = \forall x,y, (ParentOf(x,y) \land Female(x) \Rightarrow MotherOf(x,y))$

D = {n people}

WFOMC Inference: Example

3. $\Delta = \forall x, (Stress(x) \Rightarrow Smokes(x))$

Domain = {n people}

→ 3^n models

2. $\Delta = \forall y, (ParentOf(y) \wedge Female \Rightarrow MotherOf(y))$

D = {n people}

$$WMC(\Delta) = WMC(\neg Female \vee \forall y, (ParentOf(y) \Rightarrow MotherOf(y)))$$
$$= 2 * 2^n * 2^n - (2 - 1) * (2^n * 2^n - WMC(\forall y, (ParentOf(y) \Rightarrow MotherOf(y))))$$
$$= 2 * 4^n - (4^n - 3^n)$$

→ $3^n + 4^n$ models

1. $\Delta = \forall x,y, (ParentOf(x,y) \wedge Female(x) \Rightarrow MotherOf(x,y))$

D = {n people}

→ $(3^n + 4^n)^n$ models

Atom Counting: Example

$\Delta = \forall x,y, (Smokes(x) \land Friends(x,y) \Rightarrow Smokes(y))$

Domain = {n people}

Atom Counting: Example

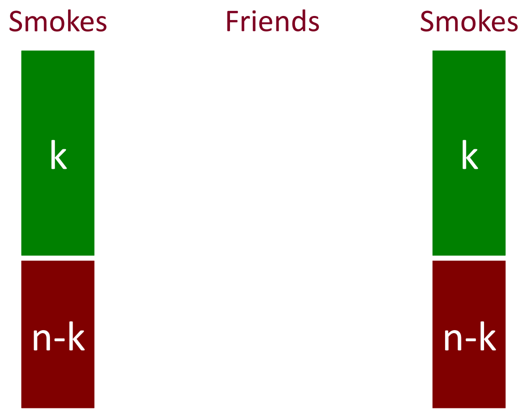
Δ = ∀x,y, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))

Domain = {n people}

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

Database:

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



Atom Counting: Example

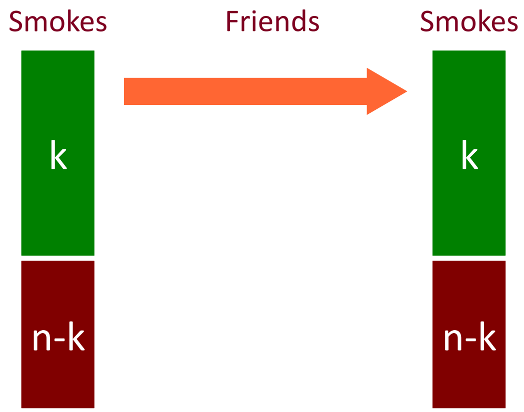
Δ = ∀x,y, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))

Domain = {n people}

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

Database:

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



Atom Counting: Example

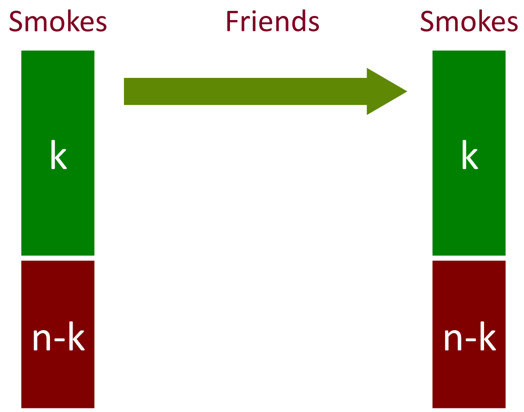
$\Delta = \forall x,y, (Smokes(x) \wedge Friends(x,y) \Rightarrow Smokes(y))$

Domain = {n people}

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

Database:

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



Atom Counting: Example

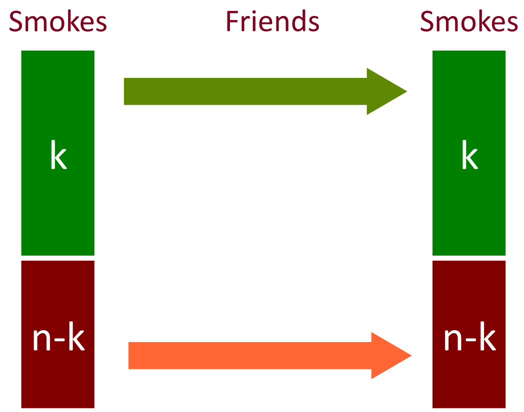
$\Delta = \forall x,y, (Smokes(x) \wedge Friends(x,y) \Rightarrow Smokes(y))$

Domain = {n people}

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

Database:

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



Atom Counting: Example

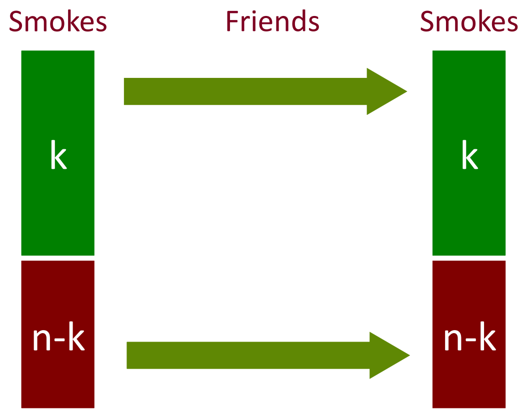
$\Delta = \forall x,y, (Smokes(x) \land Friends(x,y) \Rightarrow Smokes(y))$

Domain = {n people}

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

Database:

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



Atom Counting: Example

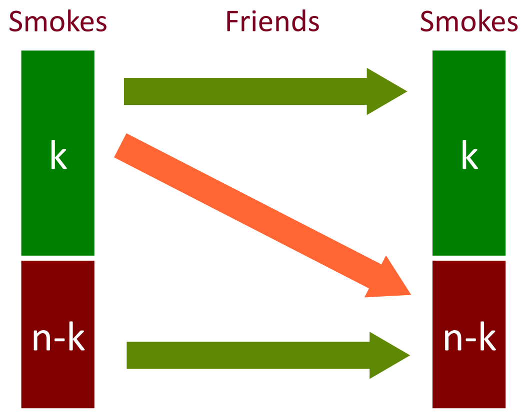
$\Delta = \forall x,y, (Smokes(x) \wedge Friends(x,y) \Rightarrow Smokes(y))$

Domain = {n people}

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

Database:

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



Atom Counting: Example

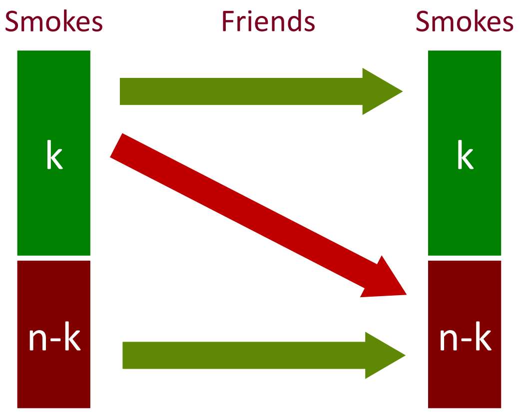
Δ = ∀x,y, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))

Domain = {n people}

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

Database:

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



Atom Counting: Example

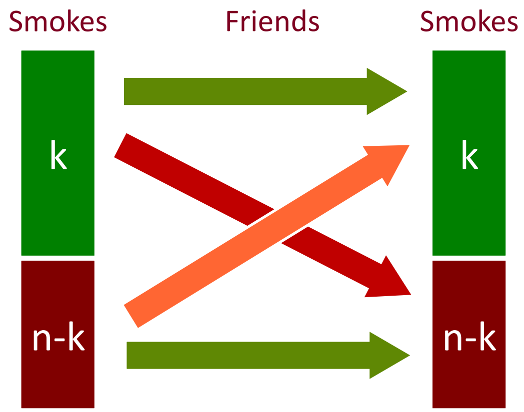
$\Delta = \forall x,y, (\text{Smokes}(x) \wedge \text{Friends}(x,y) \Rightarrow \text{Smokes}(y))$

Domain = {n people}

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

Database:

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



Atom Counting: Example

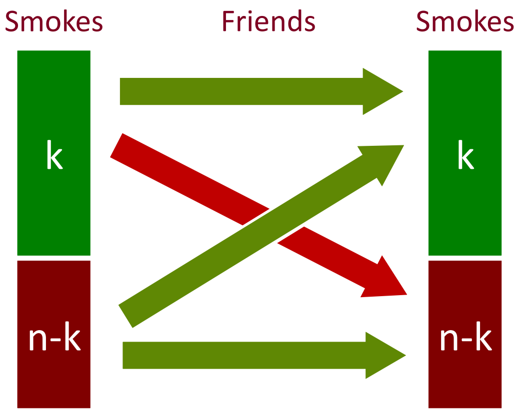
Δ = ∀x,y, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))

Domain = {n people}

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

Database:

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



Atom Counting: Example

Δ = ∀x,y, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))

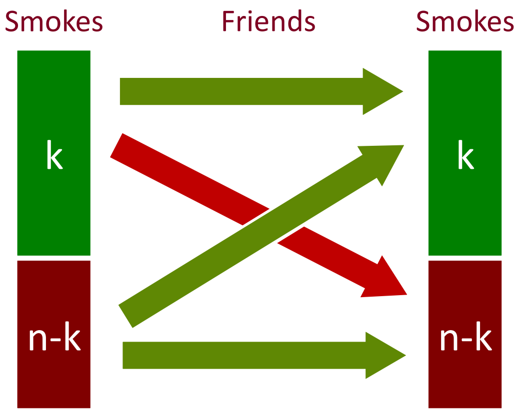
Domain = {n people}

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

Database:

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

$\rightarrow 2^{n^2 - k(n-k)}$ models



Atom Counting: Example

Δ = ∀x,y, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))

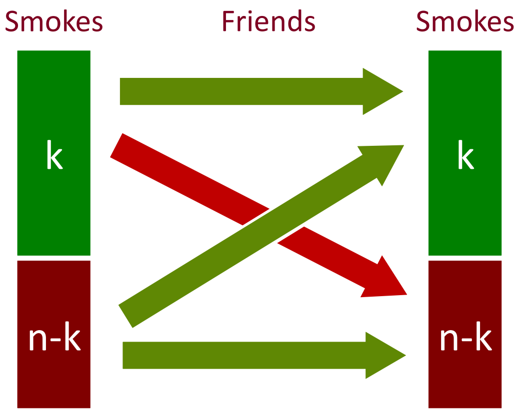
Domain = {n people}

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

Database:

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

→ $2^{n^2-k(n-k)}$ models



- If we know that there are *k* smokers?

Atom Counting: Example

Δ = ∀x,y, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))

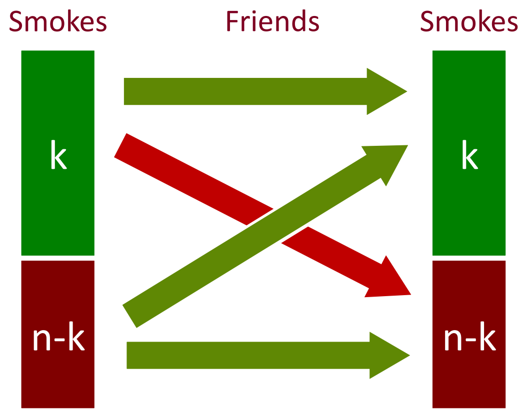
Domain = {n people}

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

Database:

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

→ $2^{n^2 - k(n-k)}$ models



- If we know that there are *k* smokers?

→ $\binom{n}{k} 2^{n^2 - k(n-k)}$ models

Atom Counting: Example

Δ = ∀x,y, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))

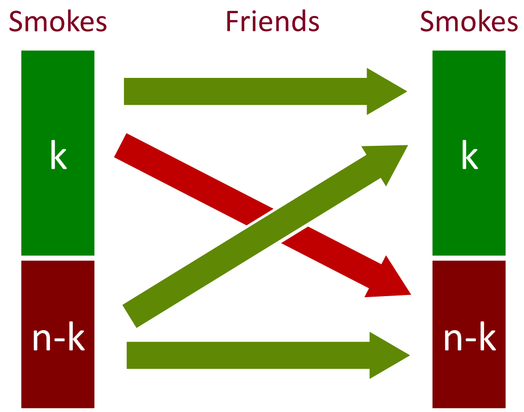
Domain = {n people}

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

Database:

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

→ $2^{n^2 - k(n-k)}$ models



- If we know that there are *k* smokers?

→ $\binom{n}{k} 2^{n^2 - k(n-k)}$ models

- In total…

Atom Counting: Example

Δ = ∀x,y, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))

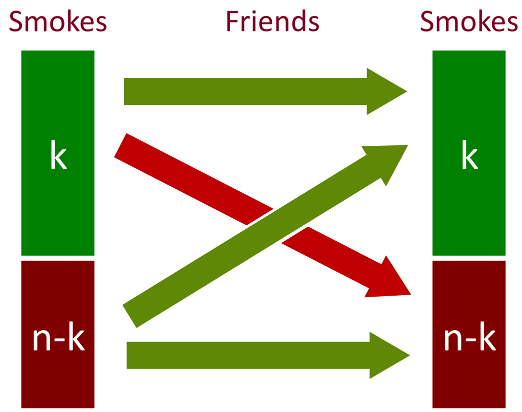
Domain = {n people}

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

Database:

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

→ $2^{n^2-k(n-k)}$ models



- If we know that there are *k* smokers? → $\binom{n}{k} 2^{n^2-k(n-k)}$ models
- In total… → $\sum_{k=0}^{n} \binom{n}{k} 2^{n^2-k(n-k)}$ models

Augment Rules with Logical Rewritings

[Suciu’11]

Augment Rules with Logical Rewritings

1. Remove constants (shattering)

$\Delta = \forall x\ (\text{Friend}(\text{Alice}, x) \lor \text{Friend}(x, \text{Bob}))$

[Suciu'11]

Augment Rules with Logical Rewritings

$F_1(x)$ = Friend(Alice,x)
$F_2(x)$ = Friend(x,Bob)
F_3 = Friend(Alice, Alice)
F_4 = Friend(Alice,Bob)
F_5 = Friend(Bob,Bob)

1. Remove constants (shattering)

$\Delta = \forall x\, (\text{Friend}(\text{Alice}, x) \lor \text{Friend}(x, \text{Bob}))$

⇒ $\Delta = \forall x\, (F_1(x) \lor F_2(x)) \land (F_3 \lor F_4) \land (F_4 \lor F_5)$

[Suciu'11]

Augment Rules with Logical Rewritings

$F_1(x)$ = Friend(Alice,x)
$F_2(x)$ = Friend(x,Bob)
F_3 = Friend(Alice, Alice)
F_4 = Friend(Alice,Bob)
F_5 = Friend(Bob,Bob)

1. Remove constants (shattering)

$\Delta = \forall x\, (\text{Friend}(\text{Alice}, x) \vee \text{Friend}(x, \text{Bob}))$

$\Rightarrow$ $\Delta = \forall x\, (F_1(x) \vee F_2(x)) \wedge (F_3 \vee F_4) \wedge (F_4 \vee F_5)$

---

2. "Rank" variables (= occur in the same order in each atom)

$\Delta = (\text{Friend}(x,y) \vee \text{Enemy}(x,y)) \quad \wedge \quad (\text{Friend}(x,y) \vee \text{Enemy}(y,x))$

Wrong order

[Suciu'11]

Augment Rules with Logical Rewritings

$F_1(x)$ = Friend(Alice,x)
$F_2(x)$ = Friend(x,Bob)
F_3 = Friend(Alice, Alice)
F_4 = Friend(Alice,Bob)
F_5 = Friend(Bob,Bob)

1. Remove constants (shattering)

$\Delta = \forall x\,(\text{Friend}(\text{Alice}, x) \vee \text{Friend}(x, \text{Bob}))$

⇨ $\Delta = \forall x\,(F_1(x) \vee F_2(x)) \wedge (F_3 \vee F_4) \wedge (F_4 \vee F_5)$

---

2. "Rank" variables (= occur in the same order in each atom)

$\Delta = (\text{Friend}(x,y) \vee \text{Enemy}(x,y)) \quad \wedge \quad (\text{Friend}(x,y) \vee \text{Enemy}(y,x))$ — Wrong order

$F_1(u,v)$ = Friend(u,v),u<v
$F_2(u)$ = Friend(u,u)
$F_3(u,v)$ = Friend(v,u),v<u

$E_1(u,v)$ = Friend(u,v),u<v
$E_2(u)$ = Friend(u,u)
$E_3(u,v)$ = Friend(v,u),v<u

⇨

$$\Delta = (F_1(x,y) \vee E_1(x,y)) \wedge (F_1(x,y) \vee E_3(x,y)) \wedge (F_2(x) \vee E_2(x)) \wedge (F_3(x,y) \vee E_3(x,y)) \wedge (F_3(x,y) \vee E_1(x,y))$$

[Suciu'11]

Augment Rules with Logical Rewritings

3. Perform Resolution [Gribkoff'14]

$\Delta = \forall x \forall y\ (R(x) \lor \neg S(x,y)) \land \forall x \forall y\ (S(x,y) \lor T(y))$

Rules stuck…

Resolution on S(x,y): $\forall x \forall y\ (R(x) \lor T(y))$

⇩

Add resolvent:

$$\Delta = \forall x \forall y\ (R(x) \lor \neg S(x,y)) \land \forall x \forall y\ (S(x,y) \lor T(y)) \land \forall x \forall y\ (R(x) \lor T(y))$$

Now apply I/E!

Augment Rules with Logical Rewritings

4. Skolemization [VdB'14]

$\Delta = \forall p, \exists c, Card(p,c)$

Inference rules assume one type of quantifier!

Mix ∀/∃ in encodings of MLNs with quantifiers and probabilistic programs

Datalog

smokes(X) :- friends(X,Y), smokes(Y).

FOL

$\Delta = \forall x, Smokes(x) \Leftrightarrow \exists y, Friends(x,y), Smokes(y).$

⇩

Skolemization Input: Mix ∀/∃ Output: Only ∀

BUT: cannot introduce Skolem constants or functions!

$\forall p, Card(p,S(p))$

Skolemization: Example

Δ = ∀p, ∃c, Card(p,c)

[VdB'14]

Skolemization: Example

$\Delta = \forall p, \exists c, \text{Card}(p,c)$

Skolemization

$\Delta' = \forall p, \forall c, \text{Card}(p,c) \Rightarrow S(p)$

[VdB'14]

Skolemization: Example

$\Delta = \forall p, \exists c, Card(p,c)$

Skolemization

$\Delta' = \forall p, \forall c, Card(p,c) \Rightarrow S(p)$

$w(S) = 1$ and $w(\neg S) = -1$

Skolem predicate

[VdB'14]

Skolemization: Example

$\Delta = \forall p, \exists c, \mathrm{Card}(p,c)$

Skolemization

$\Delta' = \forall p, \forall c, \mathrm{Card}(p,c) \Rightarrow S(p)$

$w(S) = 1$ and $w(\neg S) = -1$

Skolem predicate

Consider one position p:

$\exists c, \mathrm{Card}(p,c) = \text{true}$

$\exists c, \mathrm{Card}(p,c) = \text{false}$

[VdB'14]

Skolemization: Example

$\Delta = \forall p, \exists c, Card(p,c)$

Skolemization

$\Delta' = \forall p, \forall c, Card(p,c) \Rightarrow S(p)$

$w(S) = 1$ and $w(\neg S) = -1$

Skolem predicate

Consider one position p:

$\exists c, Card(p,c) = true$

$S(p) = true$

Also model of Δ, weight * 1

$\exists c, Card(p,c) = false$

[VdB'14]

Skolemization: Example

$\Delta = \forall p, \exists c, Card(p,c)$

Skolemization

$\Delta' = \forall p, \forall c, Card(p,c) \Rightarrow S(p)$

$w(S) = 1$ and $w(\neg S) = -1$

Skolem predicate

Consider one position p:

$\exists c, Card(p,c) = true$

- $S(p) = true$ — Also model of Δ, weight $* 1$

$\exists c, Card(p,c) = false$

- $S(p) = true$ — No model of Δ, weight $* 1$
- $S(p) = false$ — No model of Δ, weight $* -1$

Extra models

Cancel out

[VdB'14]

First-Order Knowledge Compilation

Markov Logic

3.14 Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y)

[Vdb'11,'13]

First-Order Knowledge Compilation

Markov Logic | 3.14 Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y)

Weight Function

w(Smokes)=1
w(¬Smokes)=1
w(Friends)=1
w(¬Friends)=1
w(F)=exp(3.14)
w(¬F)=1

FOL Sentence

∀x,y, F(x,y) ⇔ [Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y)]

[Vdb'11,'13]

First-Order Knowledge Compilation

Markov Logic

3.14   Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y)

Weight Function

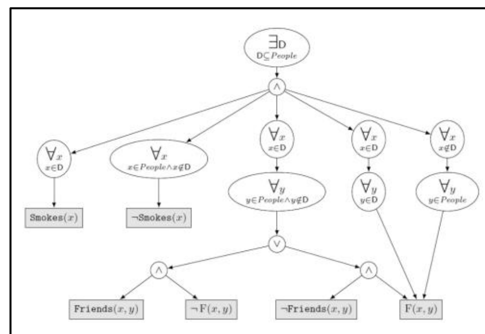
w(Smokes)=1
w(¬Smokes)=1
w(Friends)=1
w(¬Friends)=1
w(F)=exp(3.14)
w(¬F)=1

FOL Sentence

∀x,y, F(x,y) ⇔ [Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y)]

Compile?

First-Order d-DNNF Circuit



[Vdb'11,'13]

First-Order Knowledge Compilation

Markov Logic: 3.14 Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y)

Weight Function

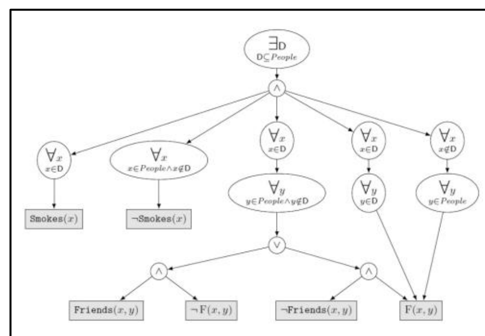
w(Smokes)=1
w(¬Smokes)=1
w(Friends)=1
w(¬Friends)=1
w(F)=exp(3.14)
w(¬F)=1

FOL Sentence

∀x,y, F(x,y) ⇔ [Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y)]

Compile?

First-Order d-DNNF Circuit



Domain

Alice
Bob
Charlie

Z = WFOMC = 1479.85

[Vdb'11,'13]

First-Order Knowledge Compilation

Markov Logic

3.14 Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y)

Weight Function

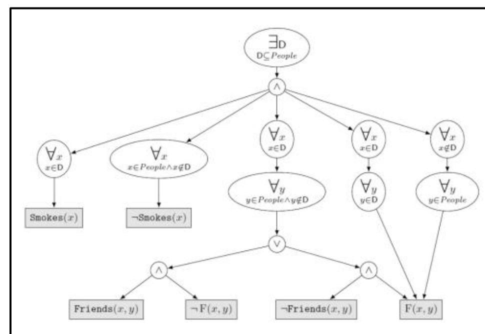
w(Smokes)=1
w(¬Smokes)=1
w(Friends)=1
w(¬Friends)=1
w(F)=exp(3.14)
w(¬F)=1

FOL Sentence

∀x,y, F(x,y) ⇔ [Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y)]

Compile?

First-Order d-DNNF Circuit



Domain

Alice
Bob
Charlie

Z = WFOMC = 1479.85

Evaluation in time polynomial in domain size

[Vdb'11,'13]

First-Order Knowledge Compilation

Markov Logic

3.14 Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y)

Weight Function

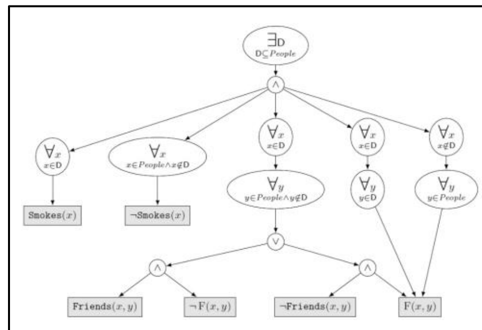
w(Smokes)=1
w(¬Smokes)=1
w(Friends)=1
w(¬Friends)=1
w(F)=exp(3.14)
w(¬F)=1

FOL Sentence

∀x,y, F(x,y) ⇔ [Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y)]

Compile?

First-Order d-DNNF Circuit



Domain

Alice
Bob
Charlie

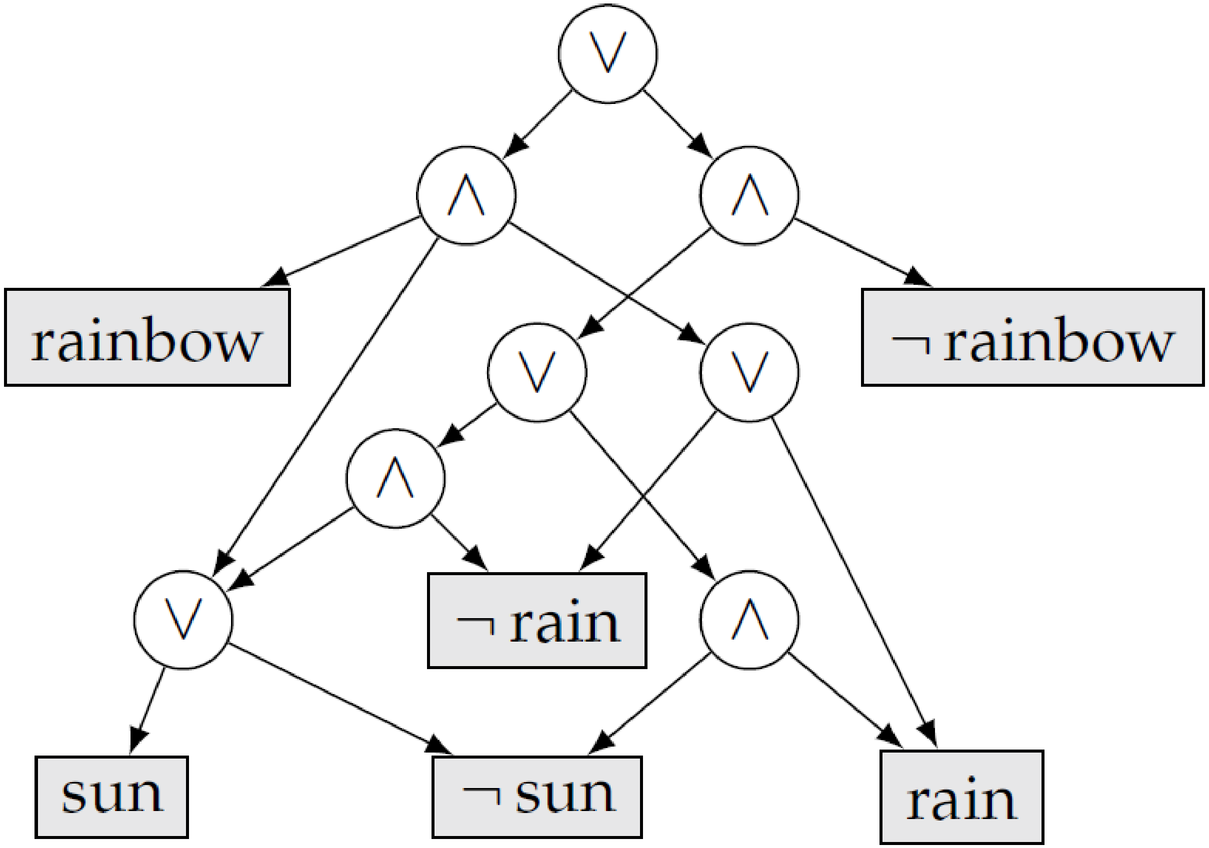
Z = WFOMC = 1479.85

Evaluation in time polynomial in domain size

Domain-lifted!

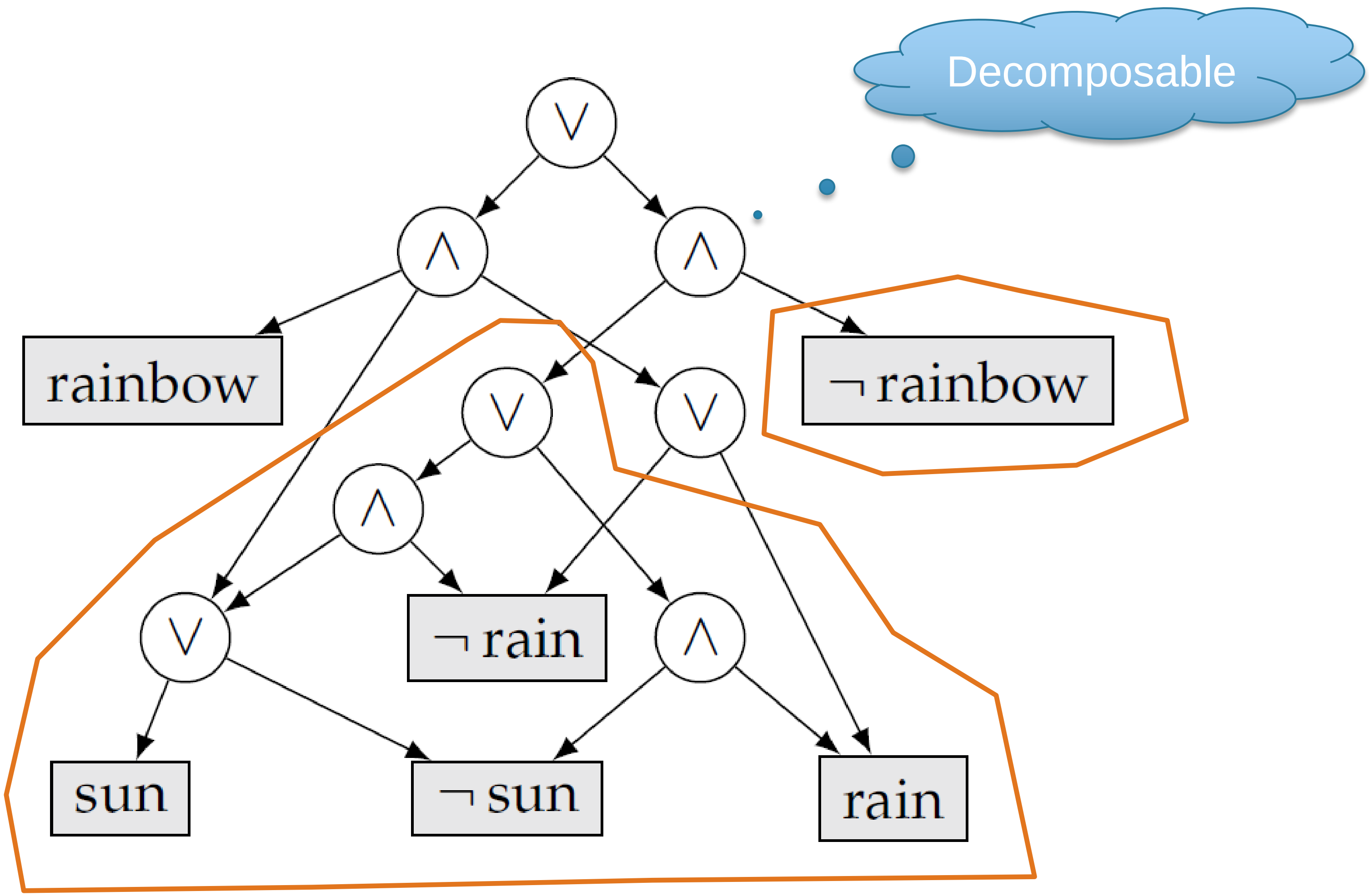
[Vdb'11,'13]

Negation Normal Form



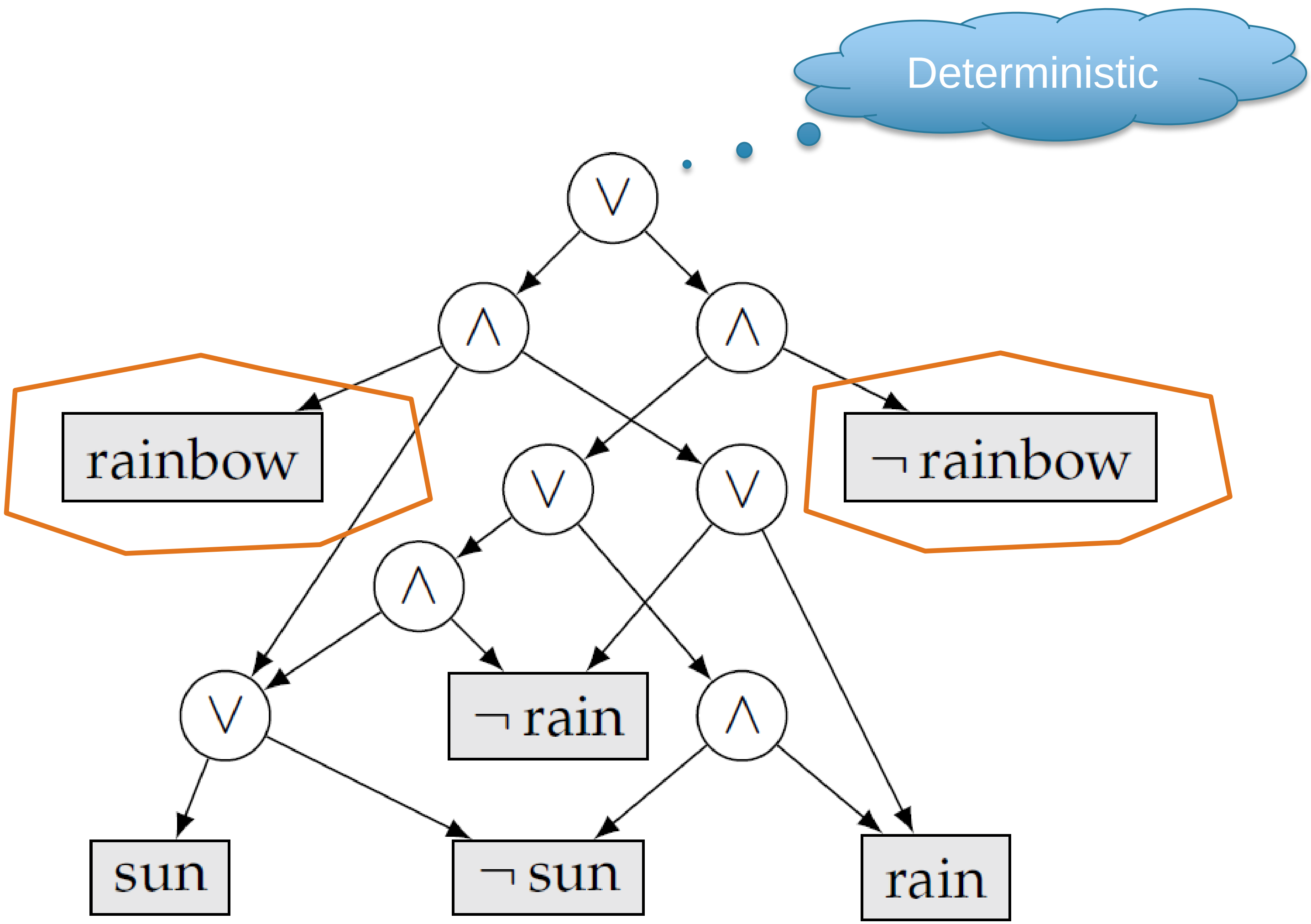
[Darwiche'01]

Decomposable NNF



[Darwiche'01]

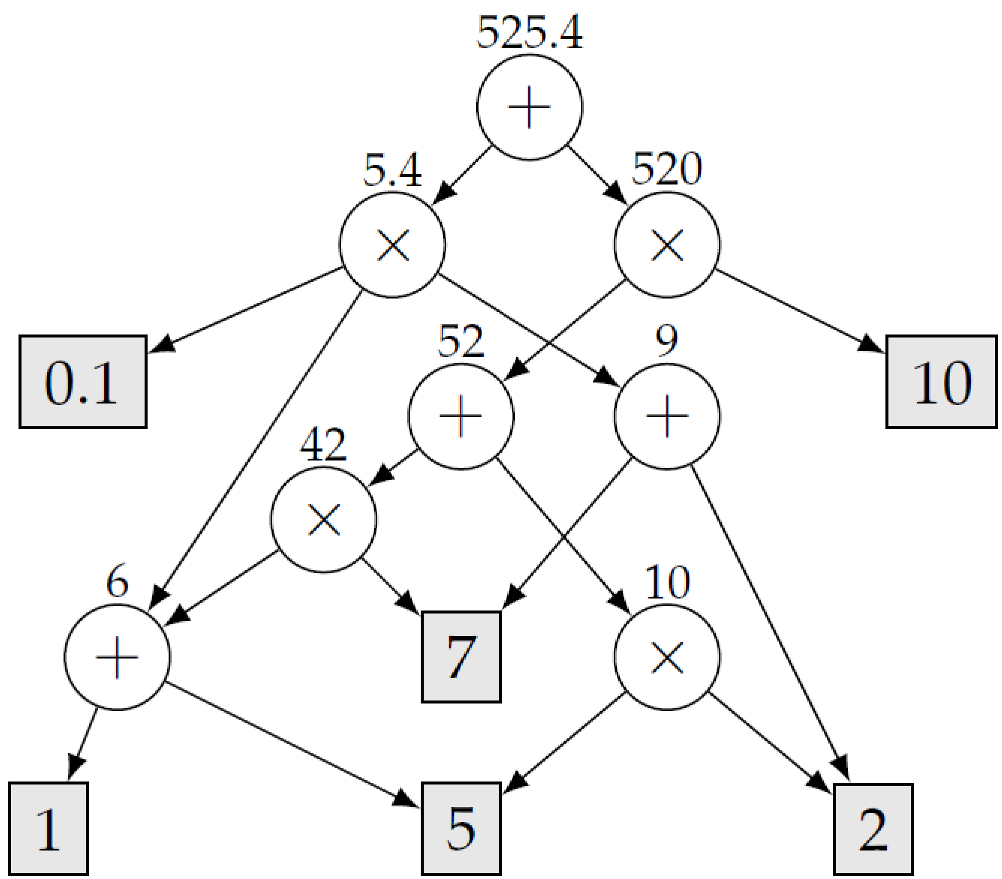
Deterministic Decomposable NNF



[Darwiche'01]

Deterministic Decomposable NNF

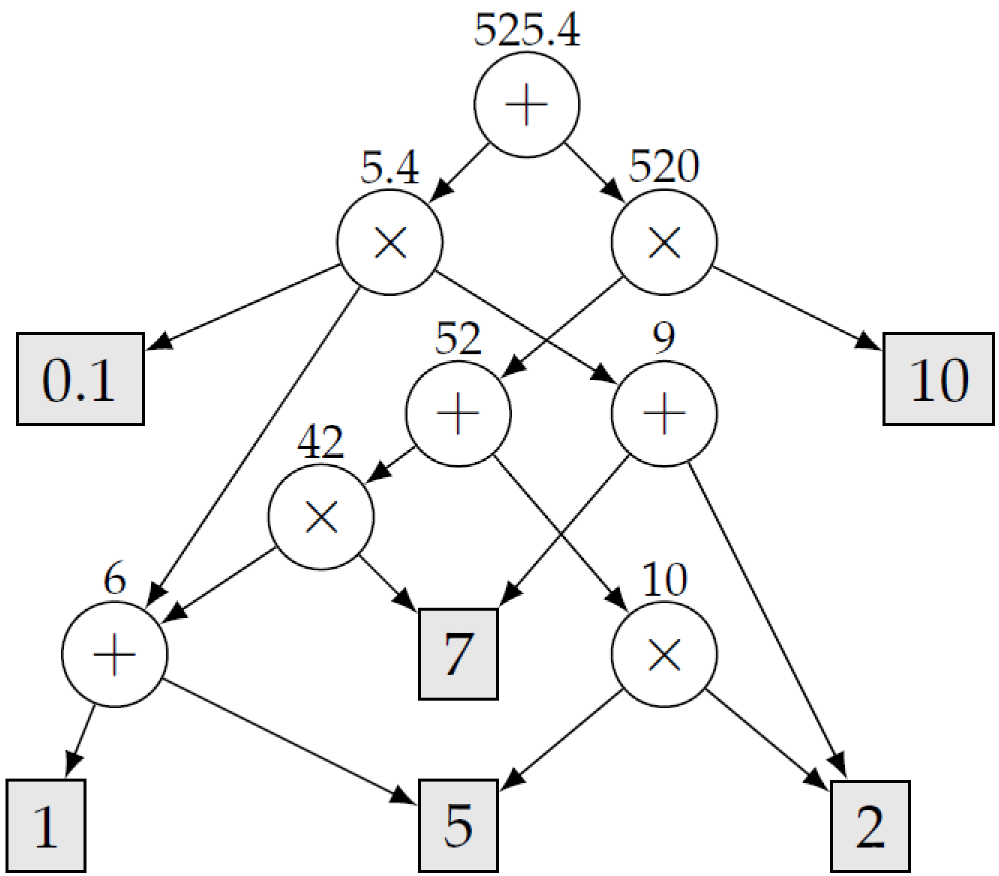
Weighted Model Counting



[Darwiche'01]

Deterministic Decomposable NNF

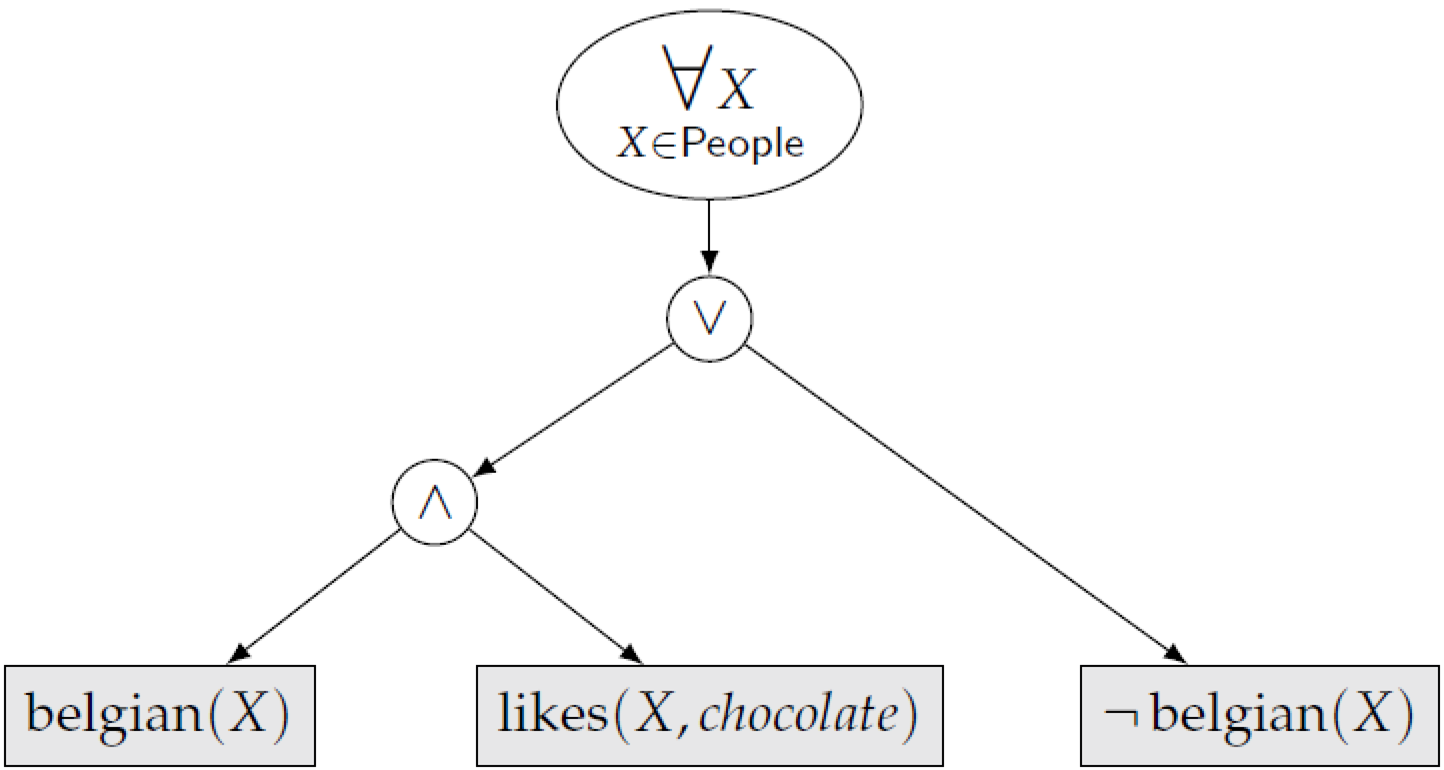
Weighted Model Counting **and much more!**



[Darwiche'01]

First-Order NNF

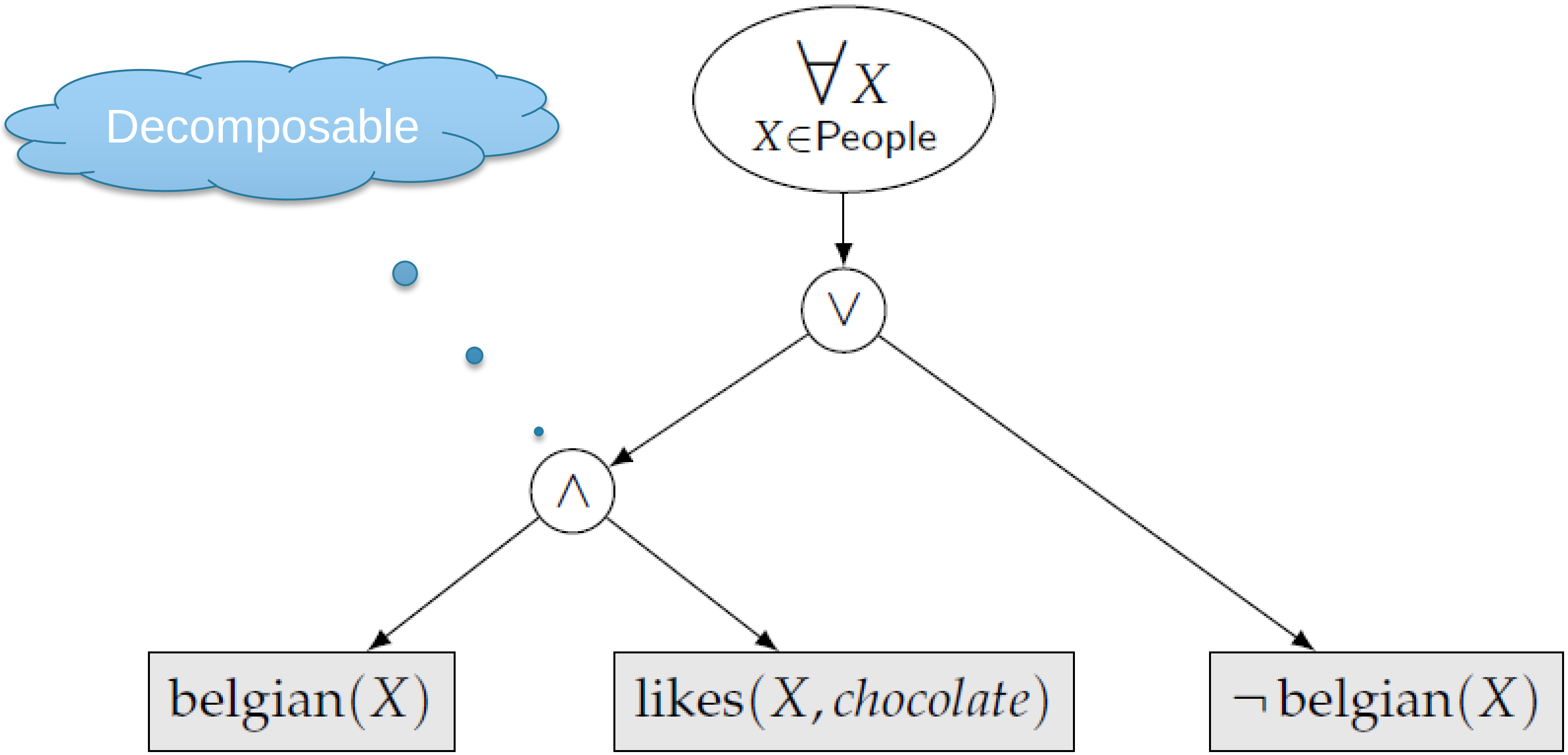
$$\forall X, X \in \text{People} : \text{belgian}(X) \Rightarrow \text{likes}(X, \mathit{chocolate})$$



[VdB'13]

First-Order Decomposability

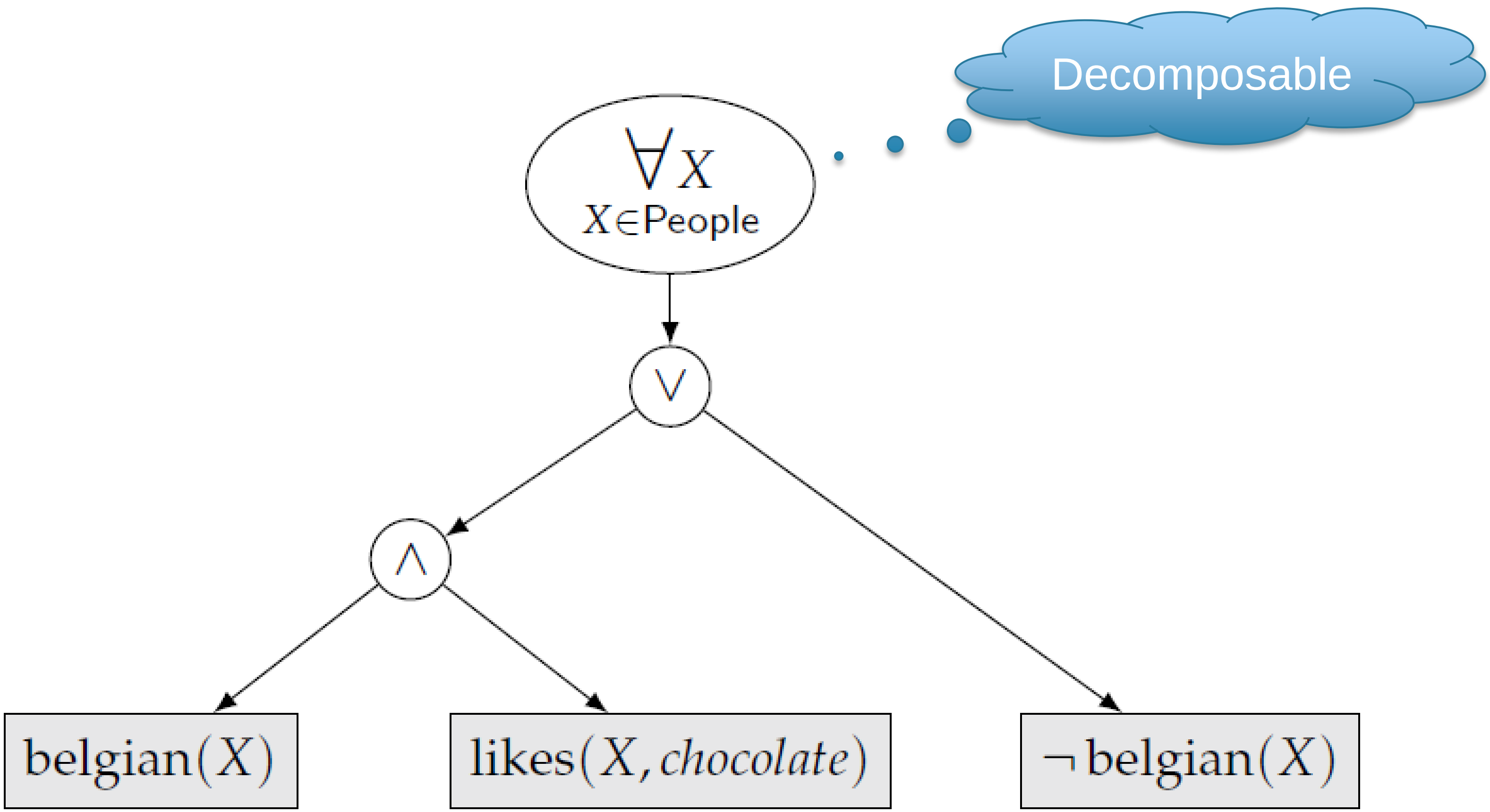
$$\forall X, X \in \text{People} : \text{belgian}(X) \Rightarrow \text{likes}(X, \mathit{chocolate})$$



[VdB'13]

First-Order Decomposability

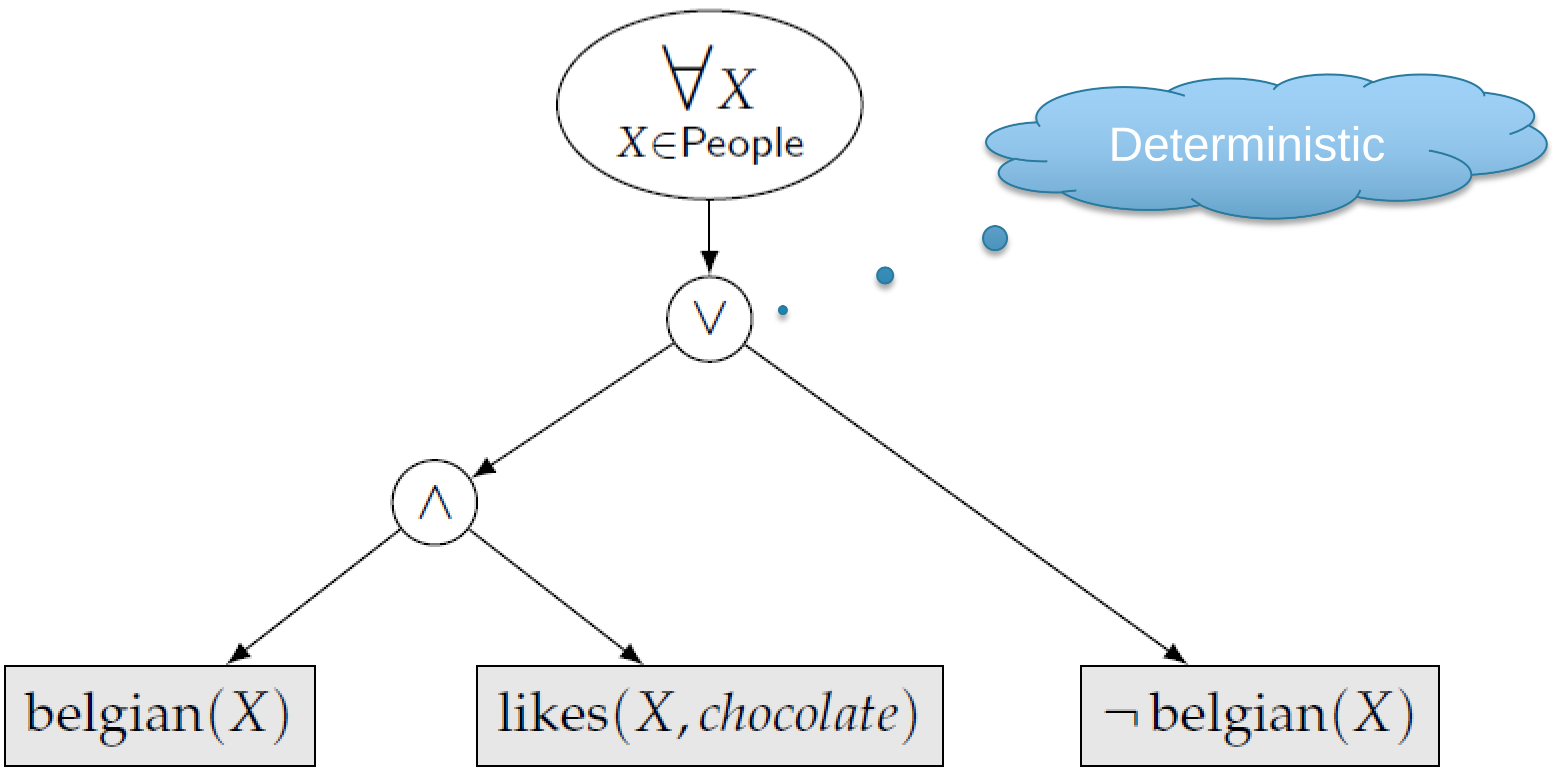
$$\forall X, X \in \text{People} : \text{belgian}(X) \Rightarrow \text{likes}(X, \textit{chocolate})$$



[VdB'13]

First-Order Determinism

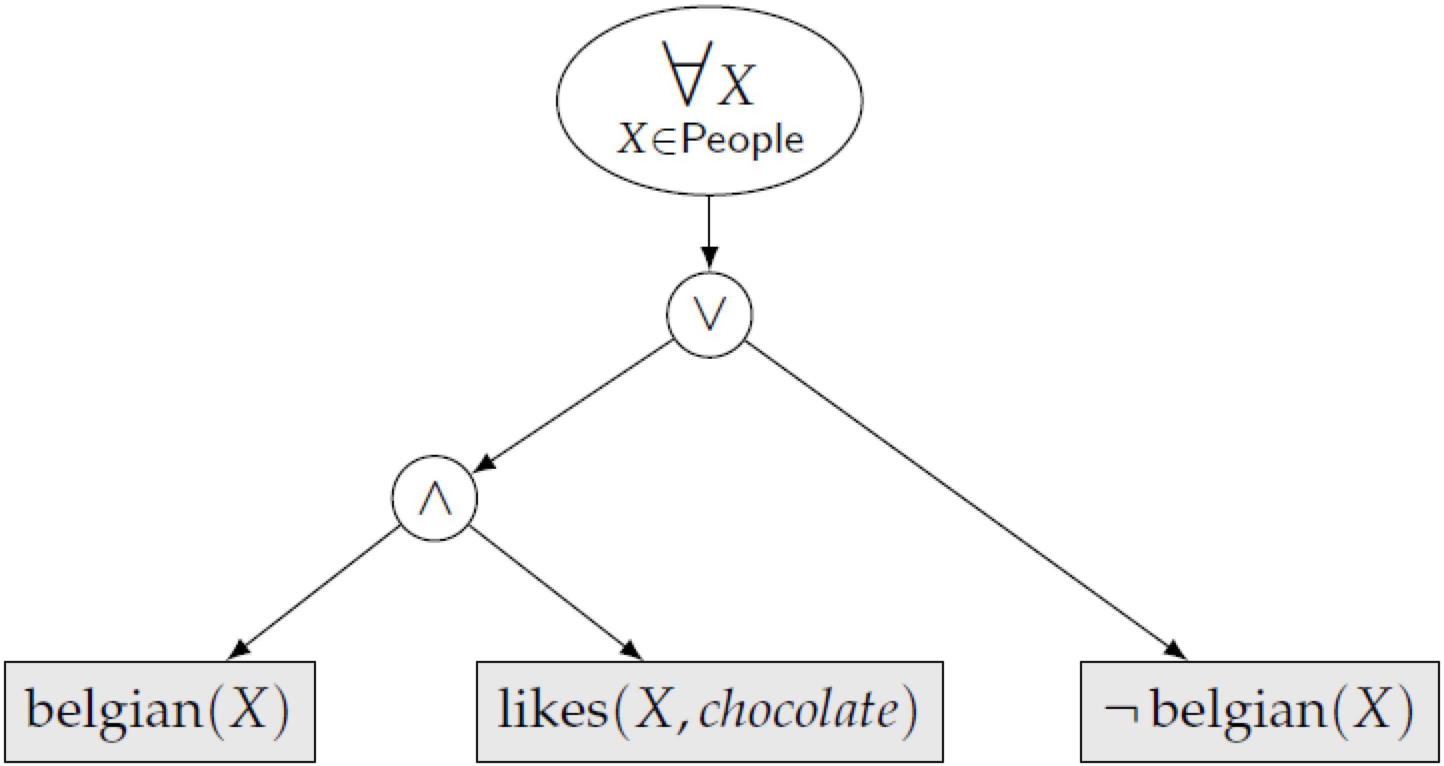
$$\forall X, X \in \text{People} : \text{belgian}(X) \Rightarrow \text{likes}(X, \textit{chocolate})$$



[VdB'13]

First-Order NNF = Query Plan

$$\forall X, X \in \text{People} : \text{belgian}(X) \Rightarrow \text{likes}(X, \textit{chocolate})$$

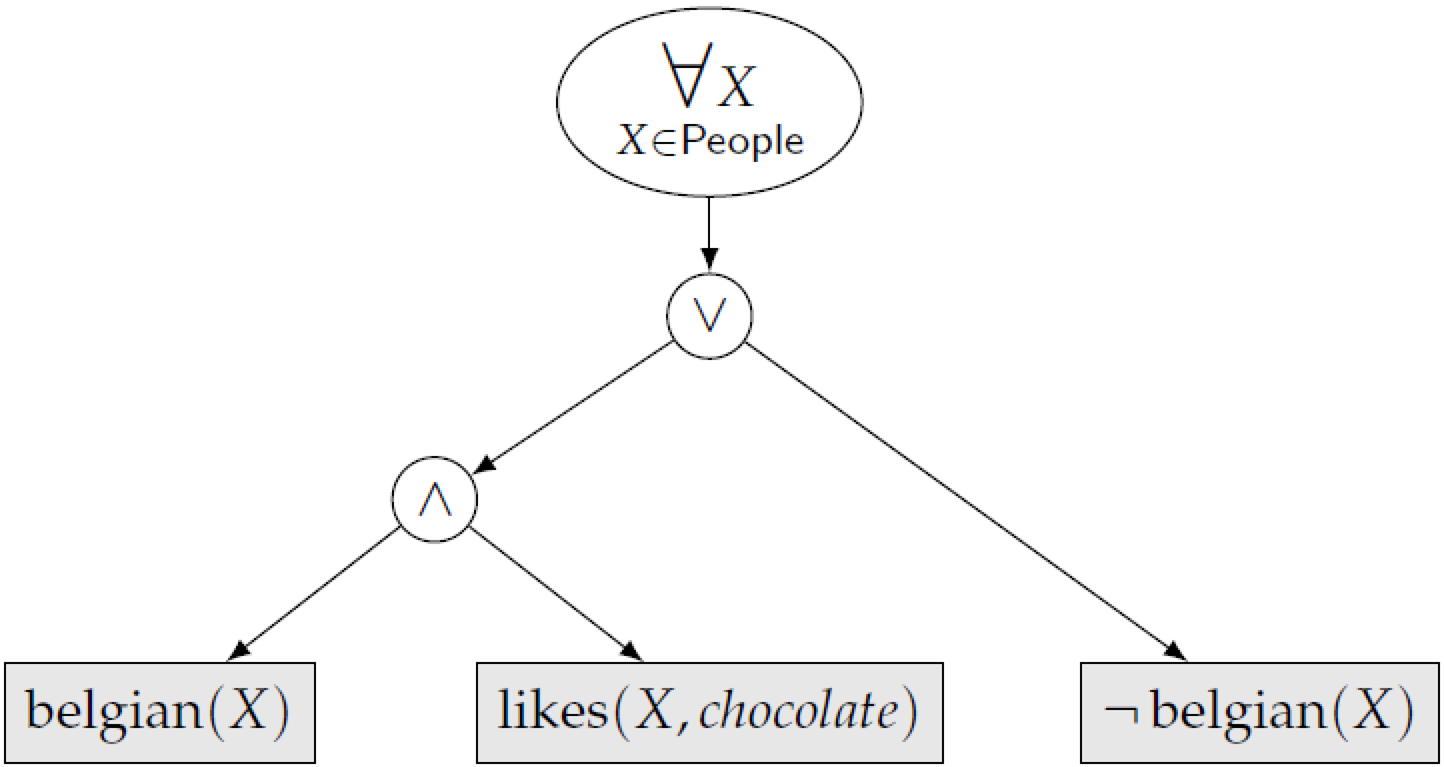


[VdB'13]

Deterministic Decomposable FO NNF

$$\forall X, X \in \text{People} : \text{belgian}(X) \Rightarrow \text{likes}(X, \textit{chocolate})$$

Weighted Model Counting

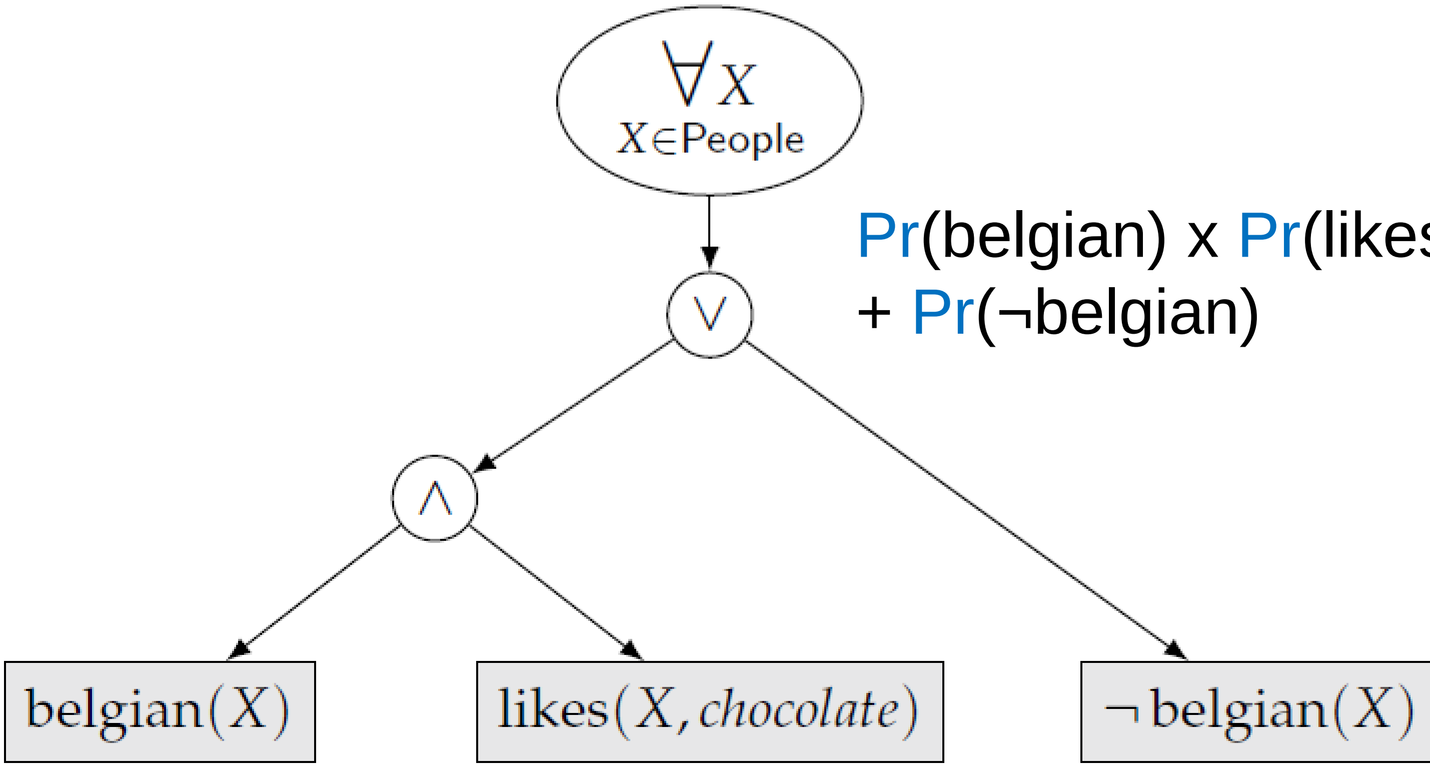


[VdB'13]

Deterministic Decomposable FO NNF

$$\forall X, X \in \mathsf{People} : \mathsf{belgian}(X) \Rightarrow \mathsf{likes}(X, \mathit{chocolate})$$

Weighted Model Counting



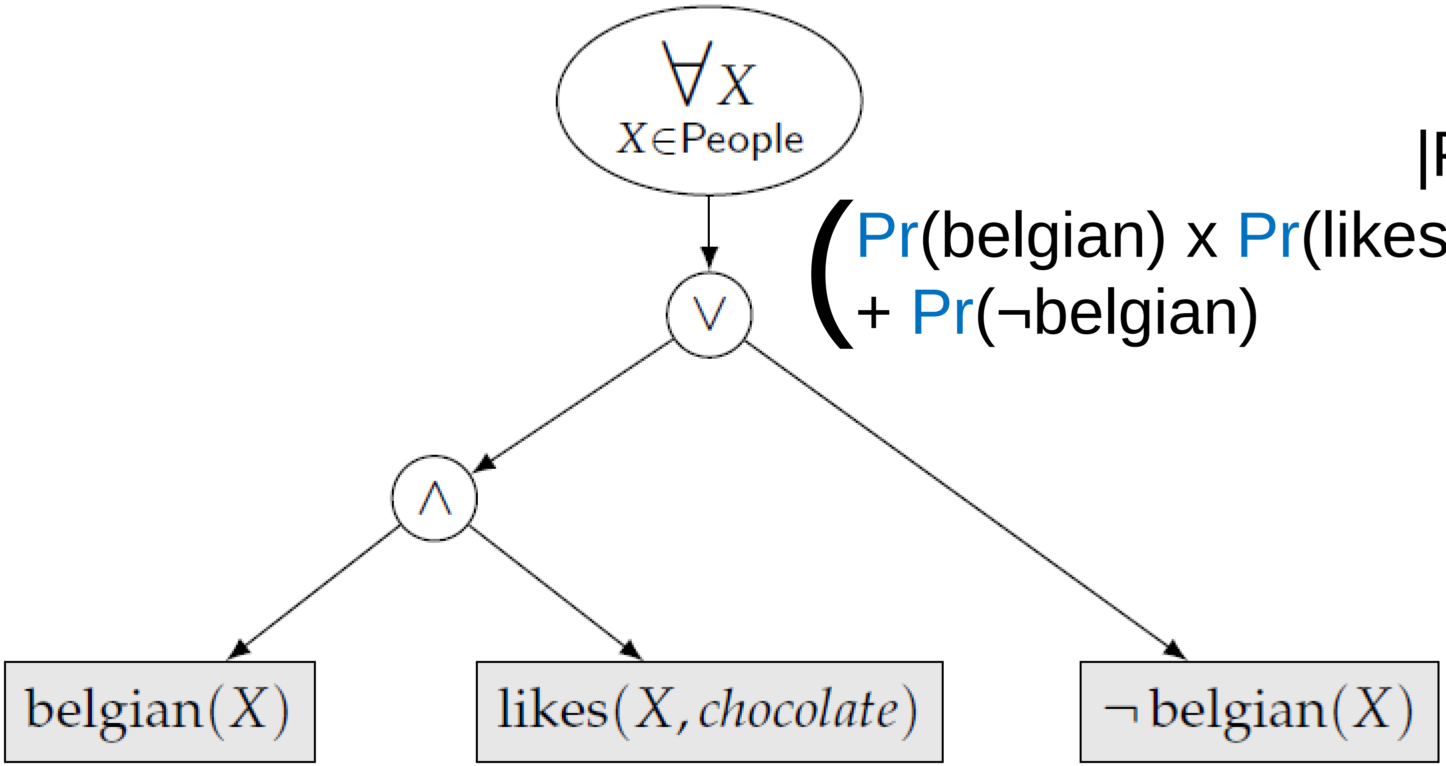
Pr(belgian) x Pr(likes)
+ Pr(¬belgian)

[VdB'13]

Deterministic Decomposable FO NNF

$$\forall X, X \in \text{People} : \text{belgian}(X) \Rightarrow \text{likes}(X, \textit{chocolate})$$

Weighted Model Counting



$$\left(\text{Pr}(\text{belgian}) \times \text{Pr}(\text{likes}) + \text{Pr}(\neg\text{belgian})\right)^{|\text{People}|}$$

[VdB'13]

Symmetric WFOMC on FO NNF

$$U(\alpha) = \begin{cases} 0 & \textit{when } \alpha = \mathsf{false} \\ 1 & \textit{when } \alpha = \mathsf{true} \\ 0.5 & \textit{when } \alpha \textit{ is a literal} \\ U(\ell_1) \times \cdots \times U(\ell_n) & \textit{when } \alpha = \ell_1 \wedge \cdots \wedge \ell_n \\ U(\ell_1) + \cdots + U(\ell_n) & \textit{when } \alpha = \ell_1 \vee \cdots \vee \ell_n \\ \prod_{i=1}^{n} U(\beta\{X/x_i\}) & \textit{when } \alpha = \forall X \in \tau, \beta \textit{ and } x_1, \ldots, x_n \textit{ are the objects in } \tau. \\ \sum_{i=1}^{n} U(\beta\{X/x_i\}) & \textit{when } \alpha = \exists X \in \tau, \beta \textit{ and } x_1, \ldots, x_n \textit{ are the objects in } \tau. \\ \prod_{i=0}^{|\tau|} U(\beta\{\mathbf{X}/\mathbf{x}_i\})^{\binom{|\tau|}{i}} & \textit{when } \alpha = \forall \mathbf{X} \subseteq \tau, \beta, \textit{ and } \mathbf{x}_i \textit{ is any subset of } \tau \textit{ such that } |\mathbf{x}_i| = i. \\ \sum_{i=0}^{|\tau|} \binom{|\tau|}{i} \cdot U(\beta\{\mathbf{X}/\mathbf{x}_i\}) & \textit{when } \alpha = \exists \mathbf{X} \subseteq \tau, \beta, \textit{ and } \mathbf{x}_i \textit{ is any subset of } \tau \textit{ such that } |\mathbf{x}_i| = i. \end{cases}$$

Complexity polynomial in domain size!
Polynomial in NNF size for bounded depth.

[VdB'13]

*How to do first-order knowledge **compilation**?*

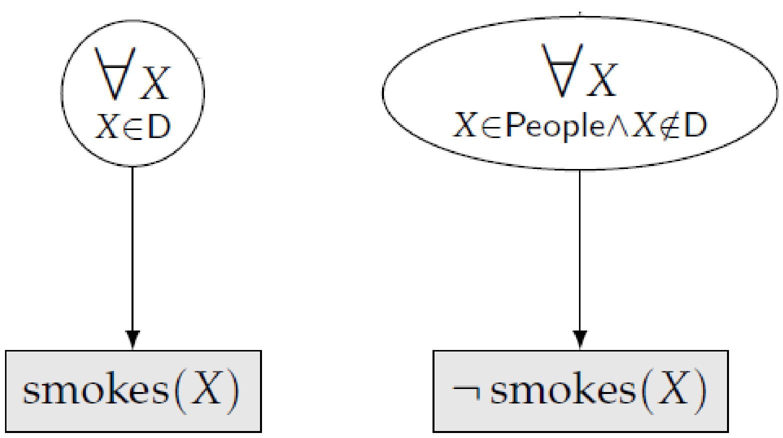
Deterministic Decomposable FO NNF

Δ = ∀x ,y ∈ **People**, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))

[VdB'13]

Deterministic Decomposable FO NNF

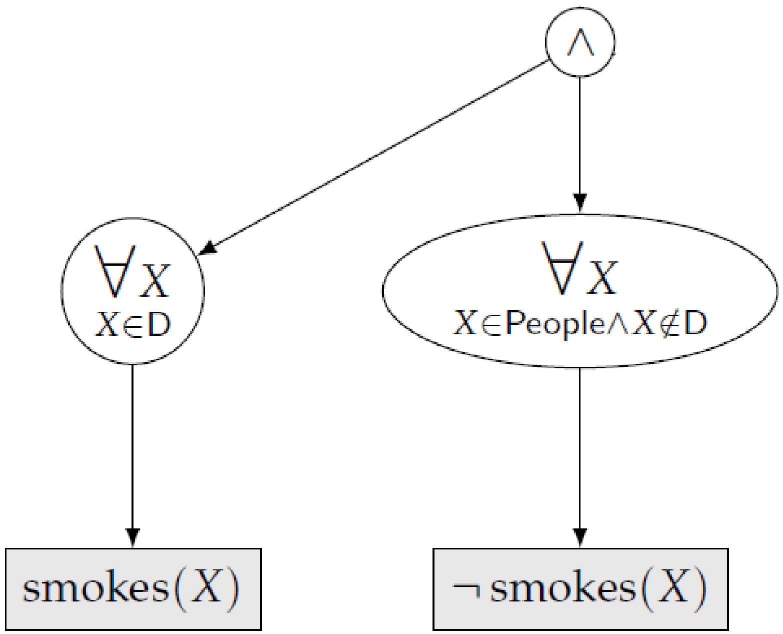
Δ = ∀x ,y ∈ **People**, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))



[VdB'13]

Deterministic Decomposable FO NNF

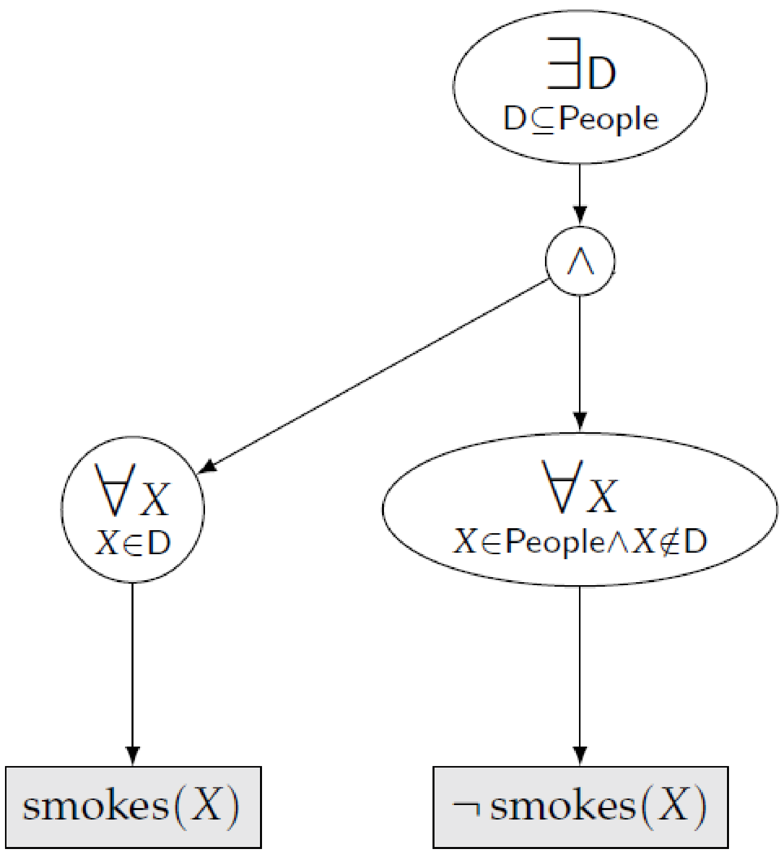
Δ = ∀x ,y ∈ **People**, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))



[VdB'13]

Deterministic Decomposable FO NNF

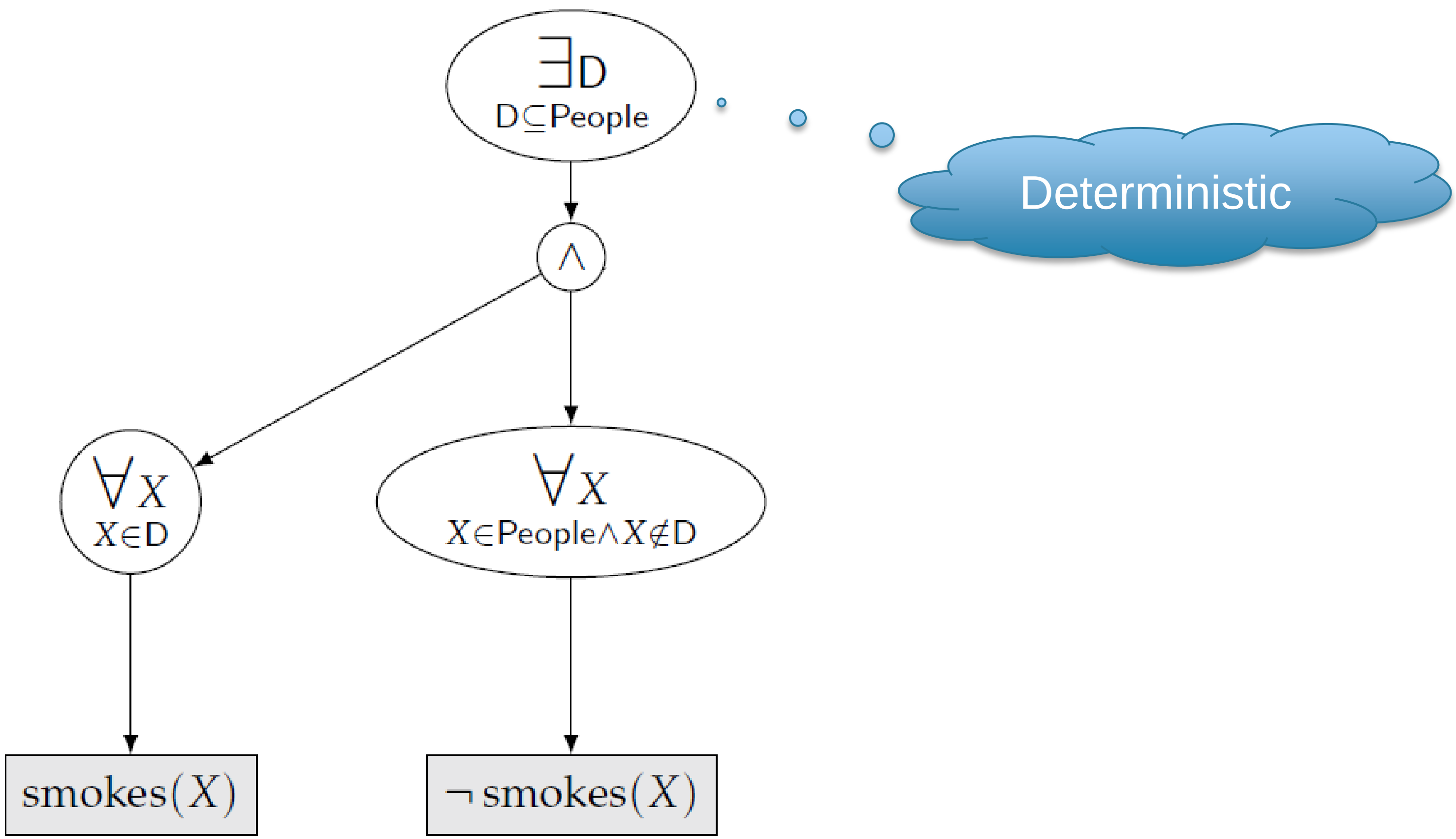
Δ = ∀x ,y ∈ **People**, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))



[VdB'13]

Deterministic Decomposable FO NNF

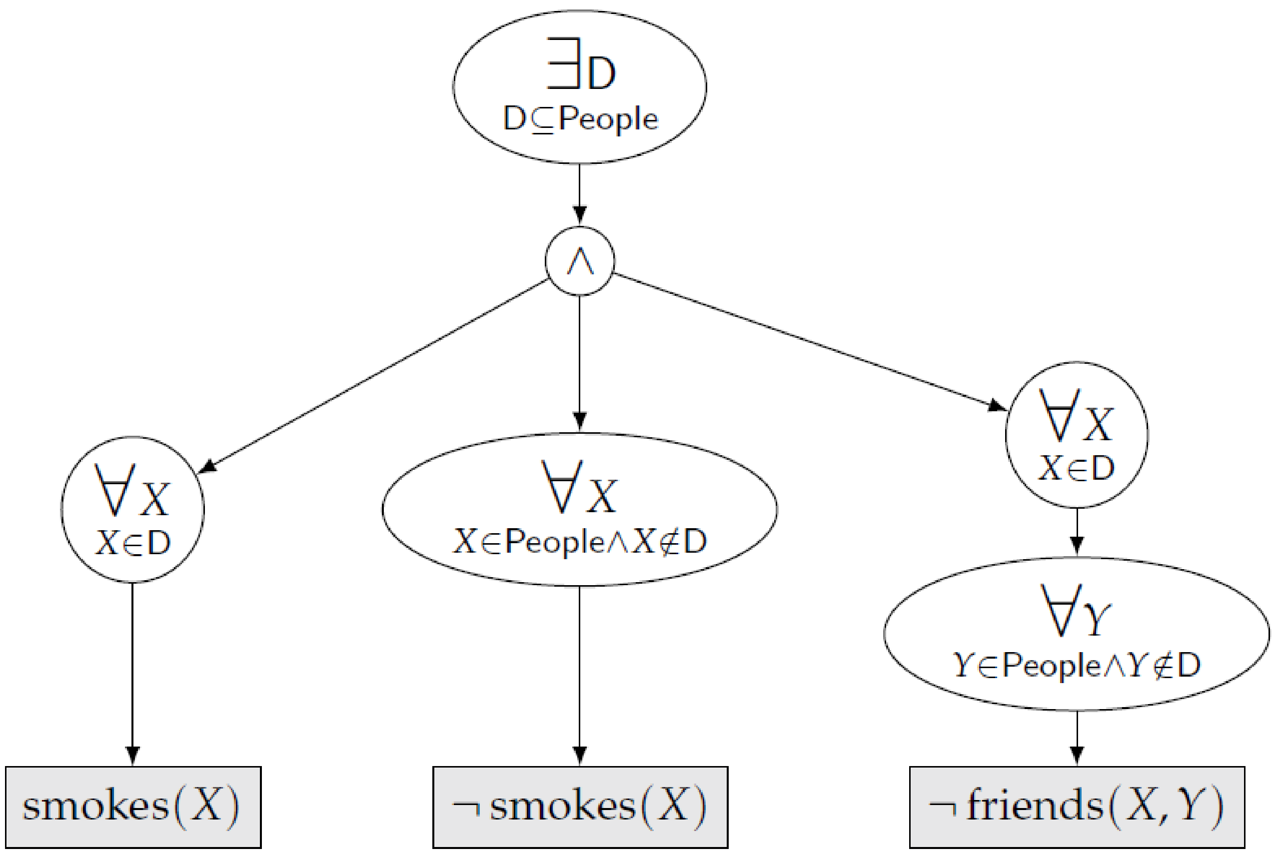
$\Delta = \forall x, y \in$ **People**, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))



[VdB'13]

Deterministic Decomposable FO NNF

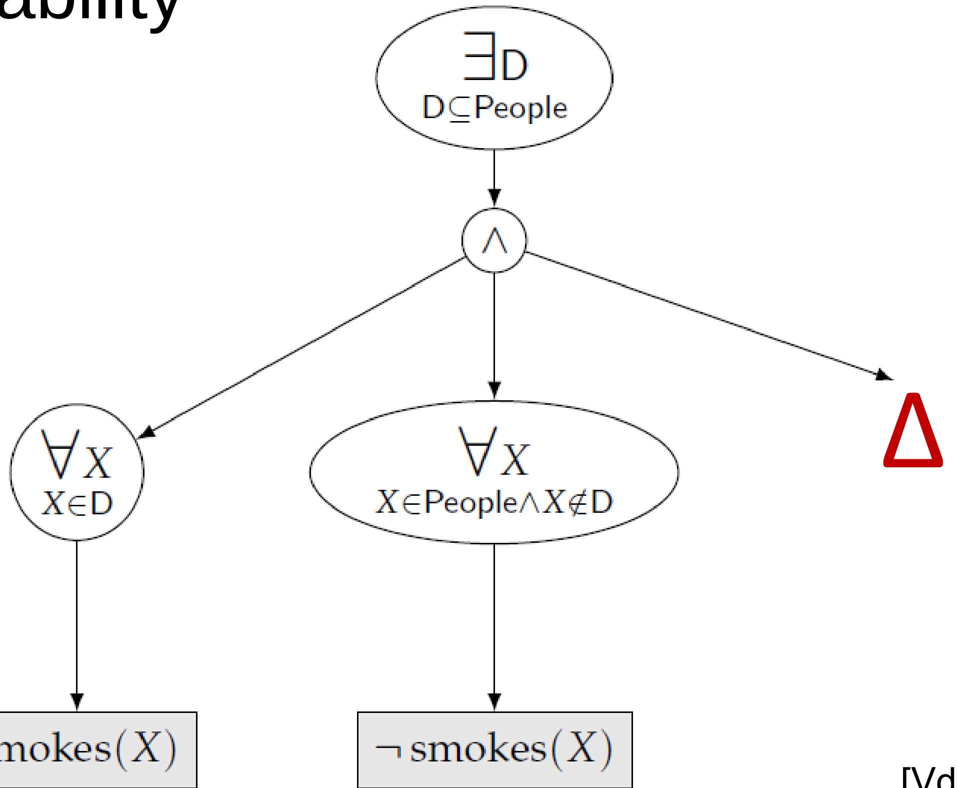
Δ = ∀x ,y ∈ **People**, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))



[VdB'13]

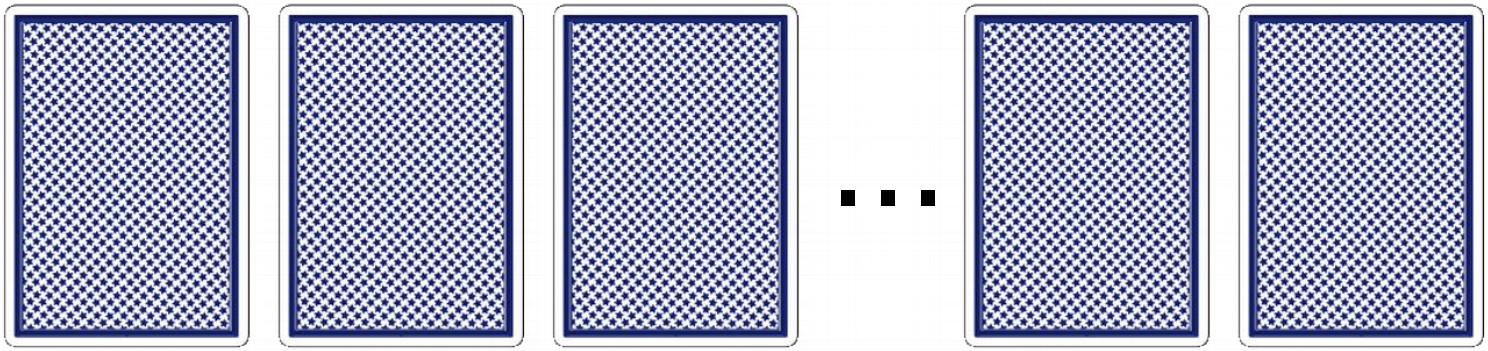
Compilation Rules

- Standard rules
  - Shannon decomposition (DPLL)
  - Detect decomposability
  - Etc.
- FO Shannon decomposition:



[VdB'13]

Playing Cards Revisited



Let us automate this:

- **Relational** model

$$\forall p, \exists c, \text{Card}(p,c)$$
$$\forall c, \exists p, \text{Card}(p,c)$$
$$\forall p, \forall c, \forall c', \text{Card}(p,c) \wedge \text{Card}(p,c') \Rightarrow c = c'$$

- **Lifted** probabilistic inference algorithm

Why not do propositional WMC?

Reduce to propositional model counting:

[VdB'15]

Why not do propositional WMC?

Reduce to propositional model counting:

$$
\begin{aligned}
\Delta = \;& \text{Card}(A♥,p_1) \vee \ldots \vee \text{Card}(2♣,p_1) \\
& \text{Card}(A♥,p_2) \vee \ldots \vee \text{Card}(2♣,p_2) \\
& \ldots \\
& \text{Card}(A♥,p_1) \vee \ldots \vee \text{Card}(A♥,p_{52}) \\
& \text{Card}(K♥,p_1) \vee \ldots \vee \text{Card}(K♥,p_{52}) \\
& \ldots \\
& \neg\text{Card}(A♥,p_1) \vee \neg\text{Card}(A♥,p_2) \\
& \neg\text{Card}(A♥,p_1) \vee \neg\text{Card}(A♥,p_3) \\
& \ldots
\end{aligned}
$$

[VdB'15]

Why not do propositional WMC?

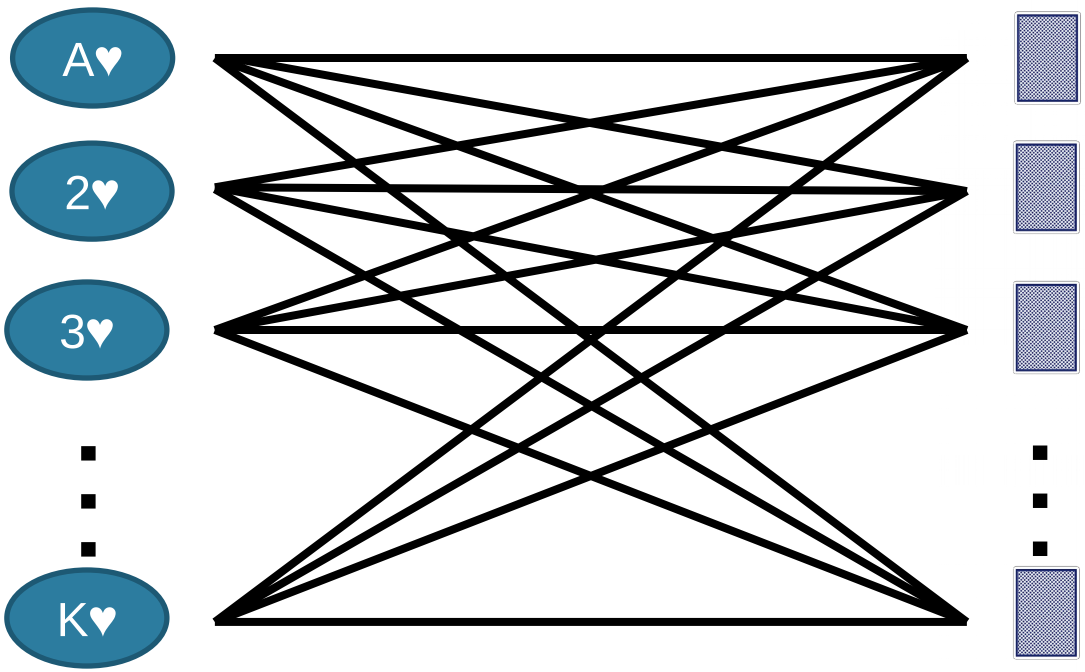
Reduce to propositional model counting:

Δ = Card(A♥,p_1) v … v Card(2♣,p_1)
Card(A♥,p_2) v … v Card(2♣,p_2)
…
Card(A♥,p_1) v … v Card(A♥,p_{52})
Card(K♥,p_1) v … v Card(K♥,p_{52})
…
¬Card(A♥,p_1) v ¬Card(A♥,p_2)
¬Card(A♥,p_1) v ¬Card(A♥,p_3)
…

What will happen?

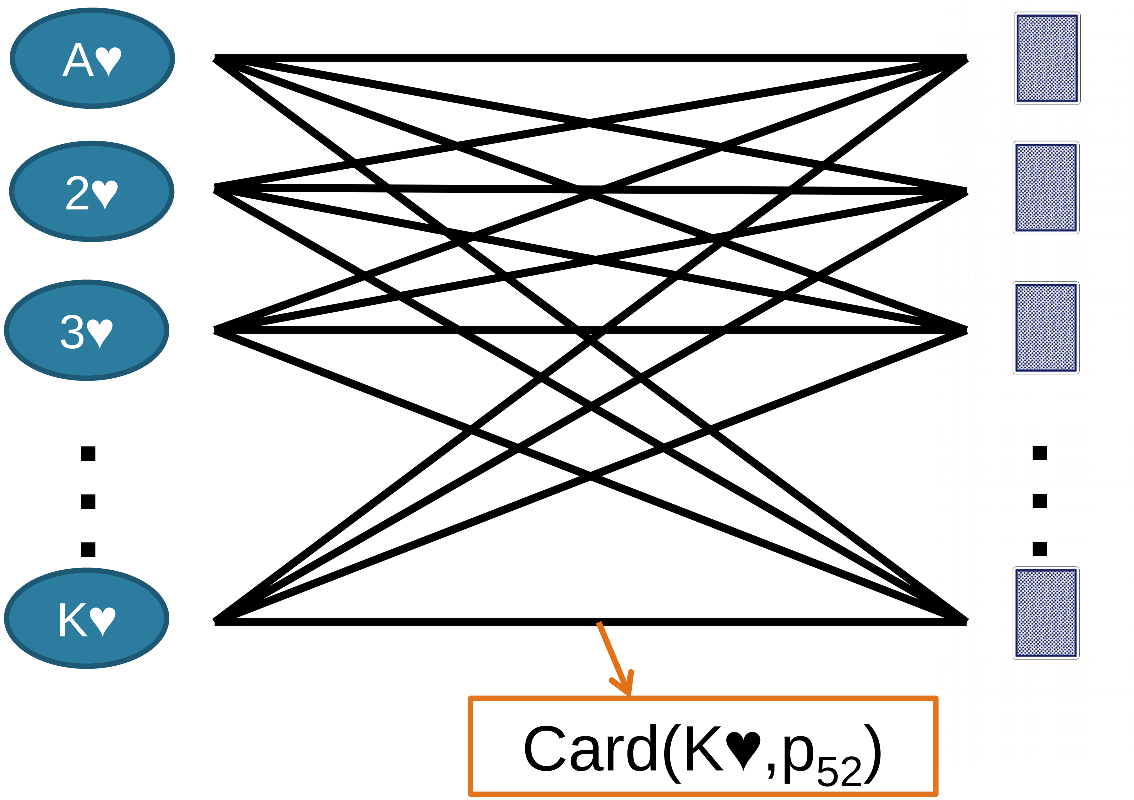
[VdB'15]

Deck of Cards Graphically



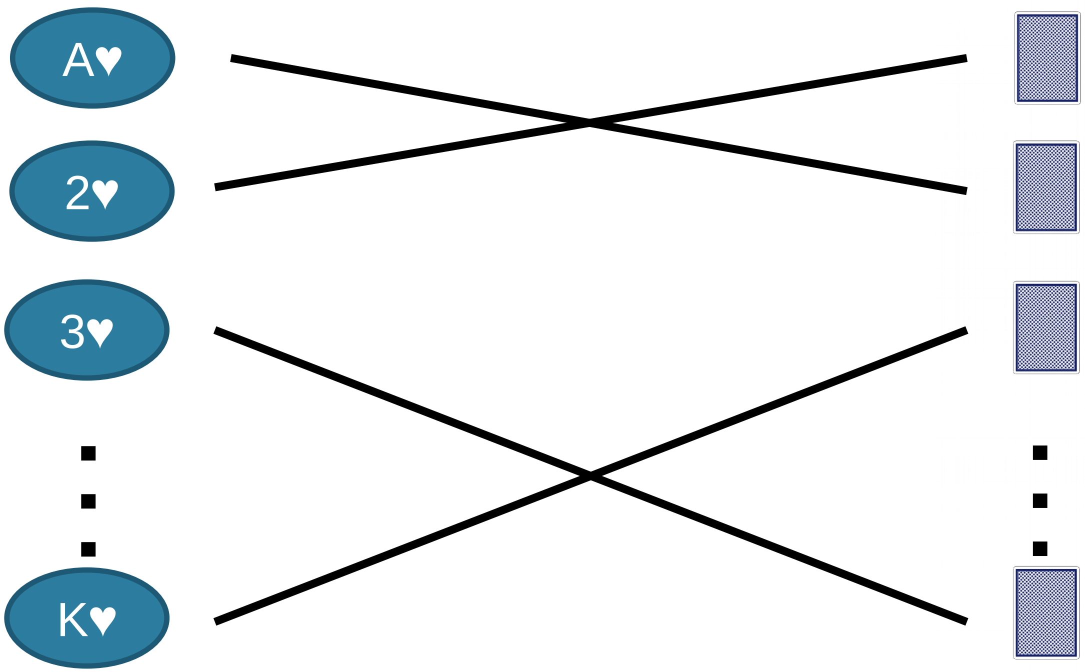
[VdB'15]

Deck of Cards Graphically



[VdB'15]

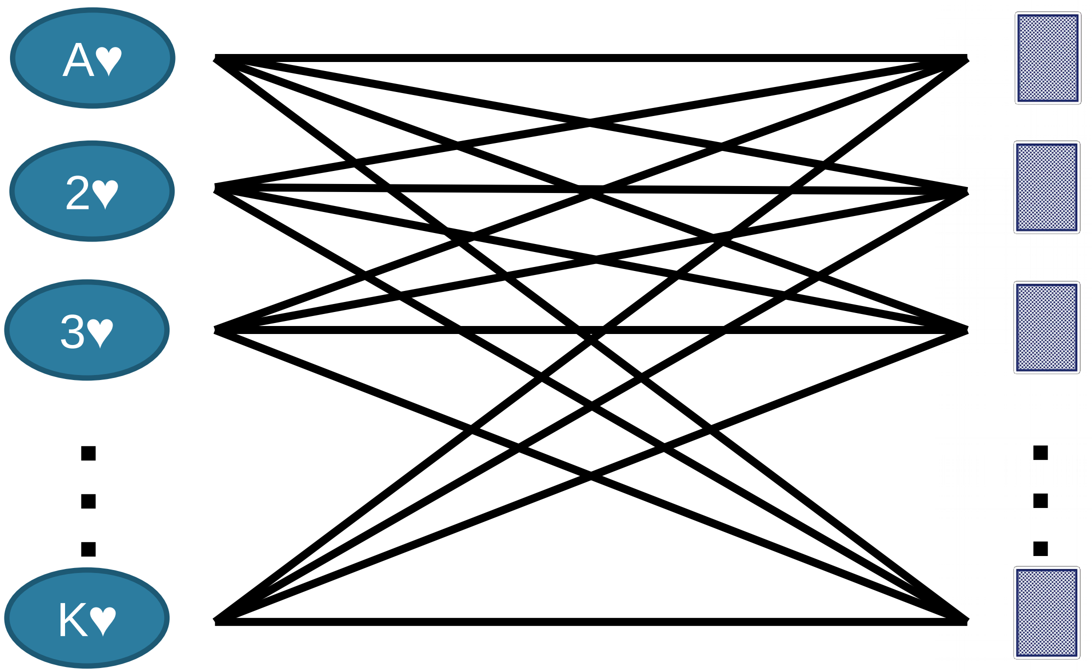
Deck of Cards Graphically



One model/*perfect matching*

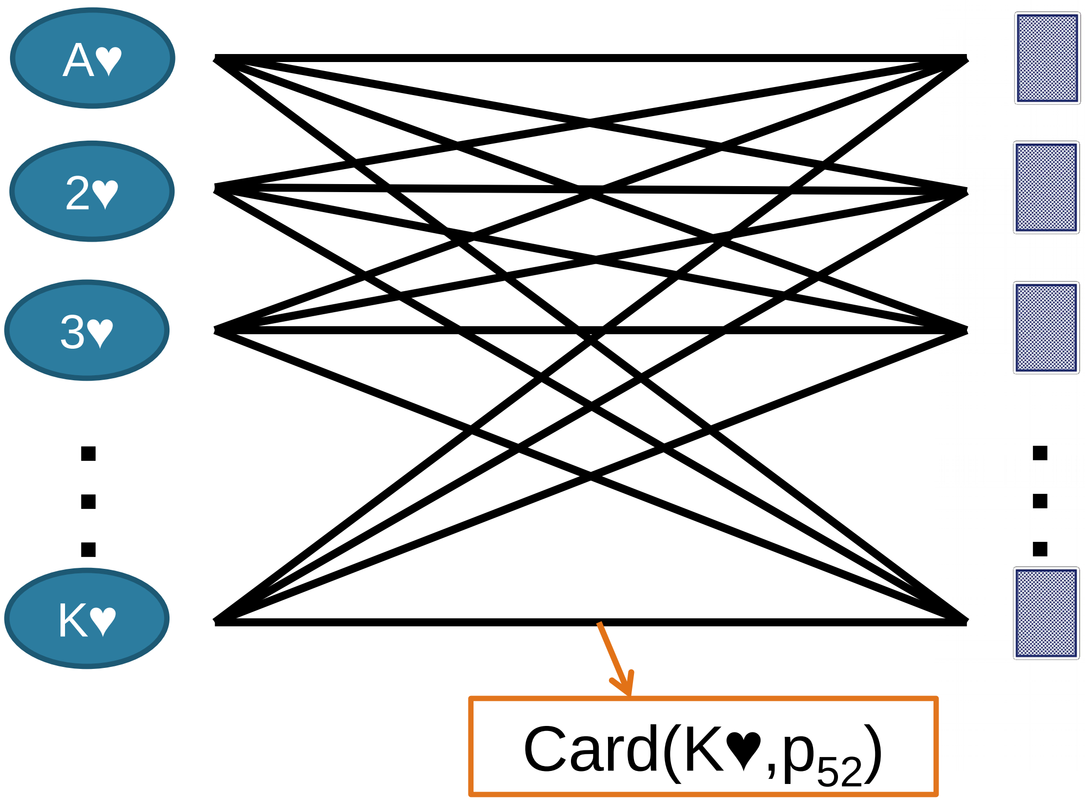
[VdB'15]

Deck of Cards Graphically



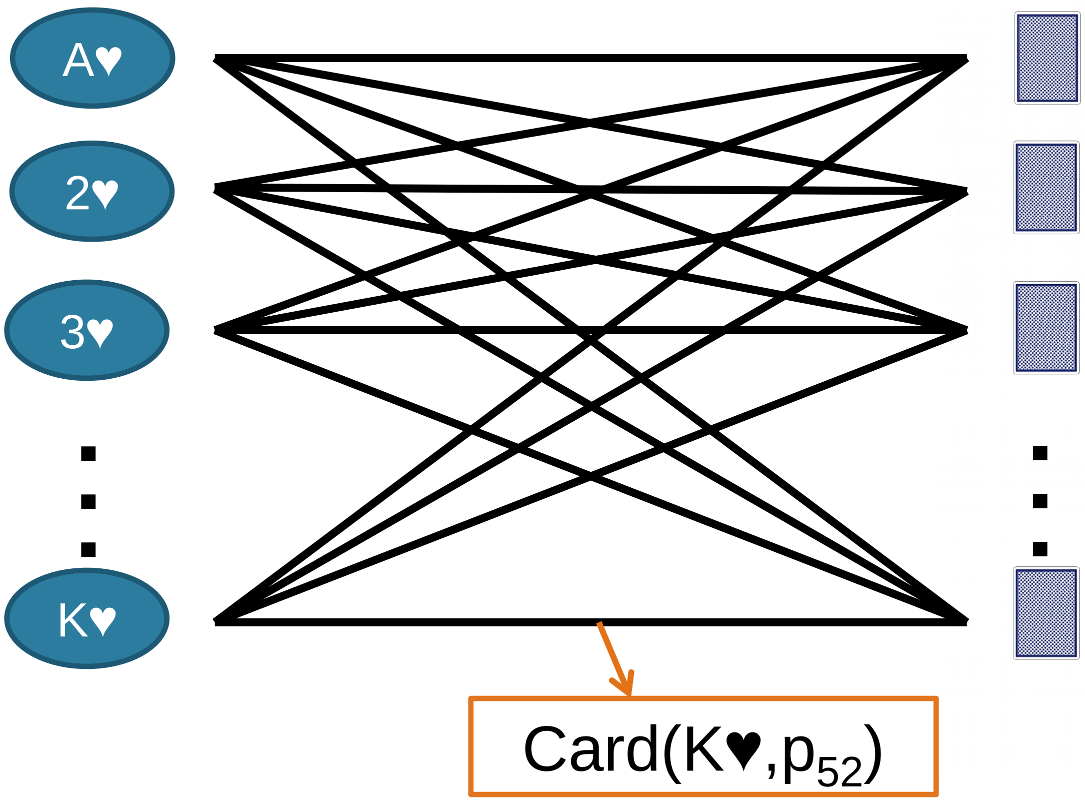
[VdB’15]

Deck of Cards Graphically



[VdB'15]

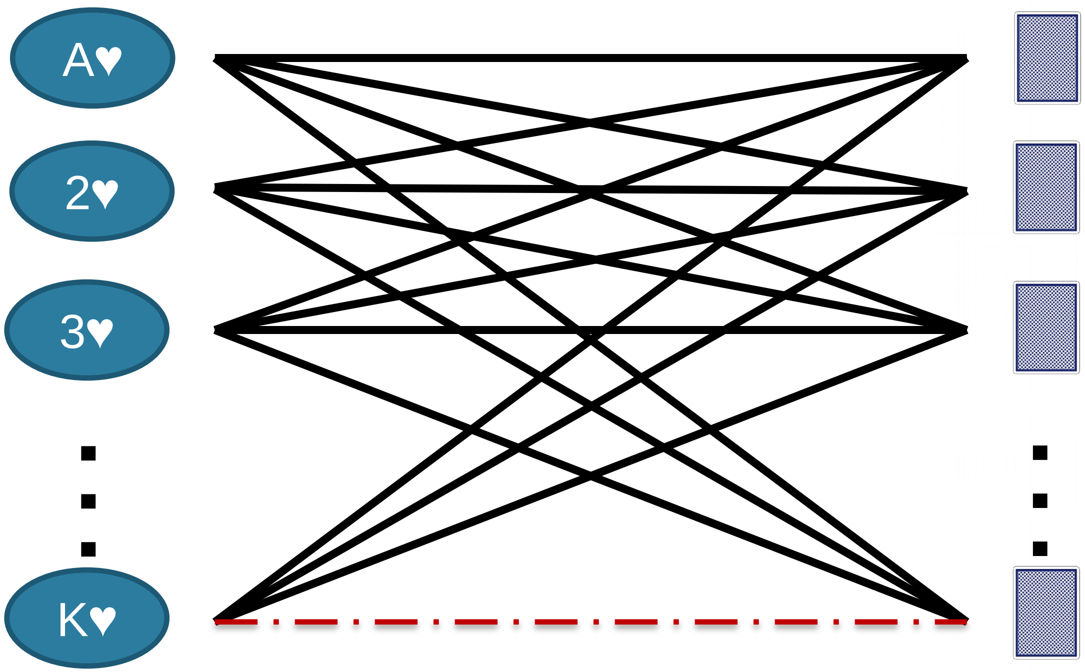
Deck of Cards Graphically



Model counting: How many *perfect matchings*?

[VdB'15]

Deck of Cards Graphically

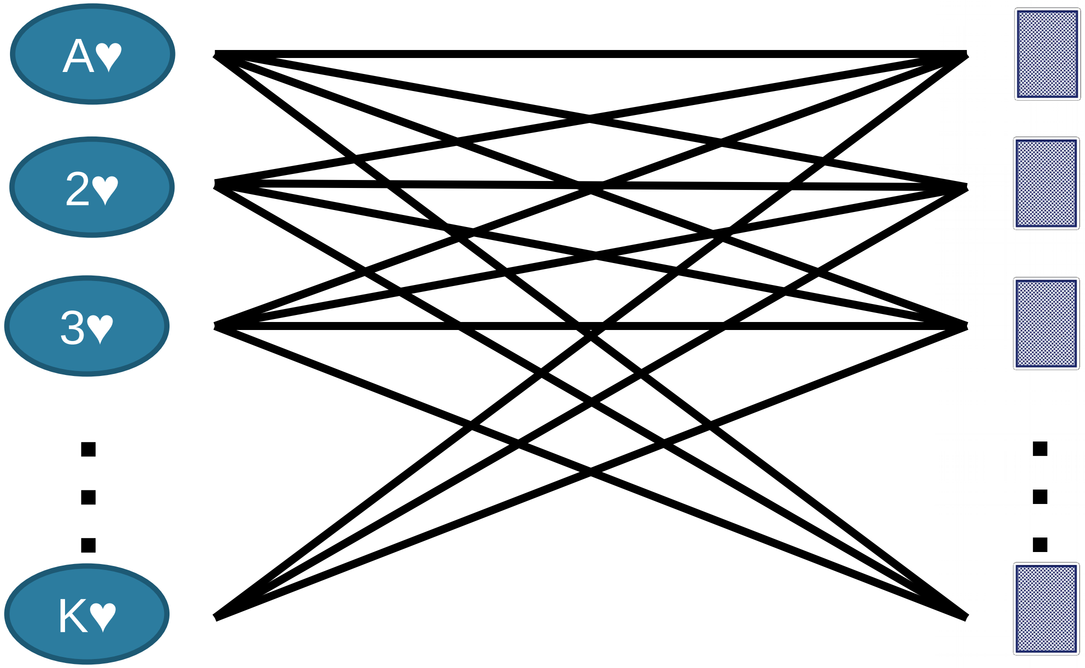


What if I set

w(Card(K♥,p_{52})) = 0?

[VdB'15]

Deck of Cards Graphically

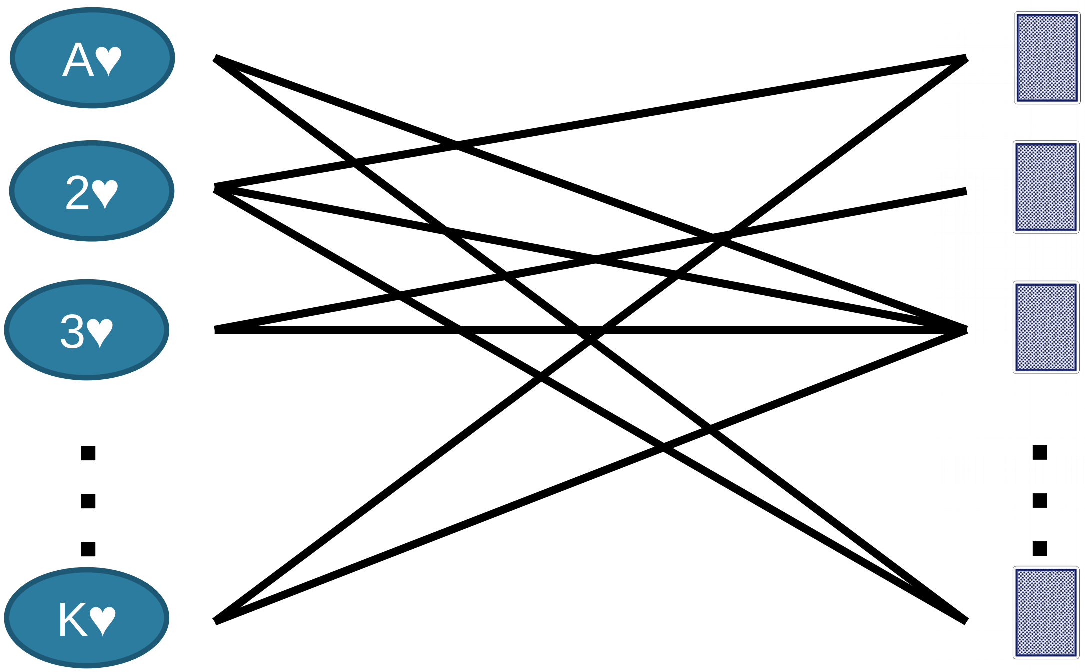


What if I set

w(Card(K♥,p_{52})) = 0?

[VdB'15]

Deck of Cards Graphically



What if I set can set any asymmetric weight function?

[VdB'15]

Observations

- Asymmetric weight function can remove edge
  Encode any bigraph

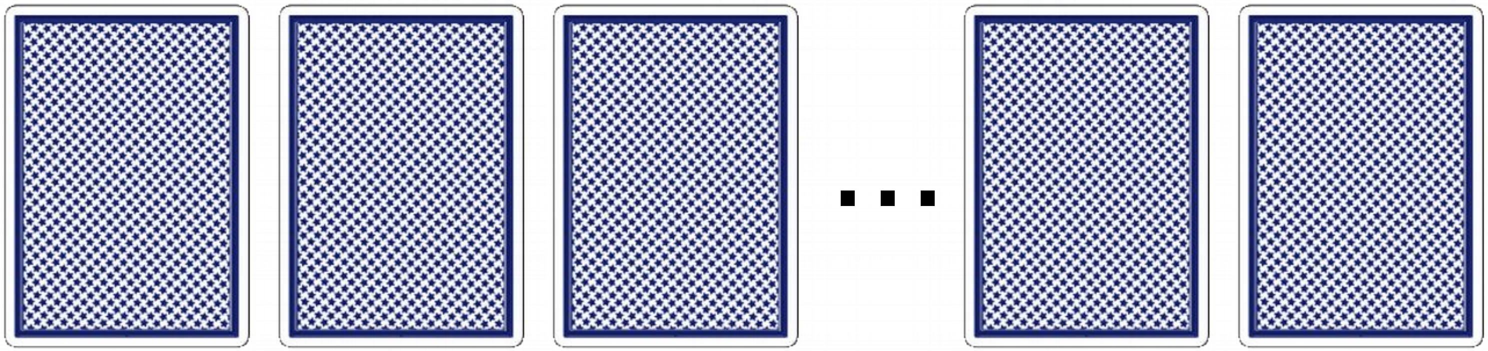
- Counting models = perfect matchings
- Problem is #P-complete! ☹

- All non-lifted WMC solvers efficiently handle asymmetric weights
- No solver does cards problem efficiently!

Later: Power of lifted vs. ground inference and complexities

[VdB'15]

Playing Cards Revisited



Let us automate this:

– **Relational** model

$$\forall p, \exists c, Card(p,c)$$
$$\forall c, \exists p, Card(p,c)$$
$$\forall p, \forall c, \forall c', Card(p,c) \wedge Card(p,c') \Rightarrow c = c'$$

– **Lifted** probabilistic inference algorithm

Playing Cards Revisited

$\forall p, \exists c, Card(p,c)$
$\forall c, \exists p, Card(p,c)$
$\forall p, \forall c, \forall c', Card(p,c) \wedge Card(p,c') \Rightarrow c = c'$

[VdB'15]

Playing Cards Revisited

$$\forall p, \exists c, Card(p,c)$$

$$\forall c, \exists p, Card(p,c)$$

$$\forall p, \forall c, \forall c', Card(p,c) \land Card(p,c') \Rightarrow c = c'$$

↓ Skolemization

[VdB'15]

Playing Cards Revisited

$\forall p, \exists c, \text{Card}(p,c)$
$\forall c, \exists p, \text{Card}(p,c)$
$\forall p, \forall c, \forall c', \text{Card}(p,c) \wedge \text{Card}(p,c') \Rightarrow c = c'$

↓ ∘ ○ ○ Skolemization

$\forall p, \forall c, \text{Card}(p,c) \Rightarrow S_1(p)$
$\forall c, \forall p, \text{Card}(p,c) \Rightarrow S_2(c)$
$\forall p, \forall c, \forall c', \text{Card}(p,c) \wedge \text{Card}(p,c') \Rightarrow c = c'$

[VdB'15]

Playing Cards Revisited

$\forall p, \exists c, Card(p,c)$
$\forall c, \exists p, Card(p,c)$
$\forall p, \forall c, \forall c', Card(p,c) \land Card(p,c') \Rightarrow c = c'$

↓ Skolemization

$\forall p, \forall c, Card(p,c) \Rightarrow S_1(p)$
$\forall c, \forall p, Card(p,c) \Rightarrow S_2(c)$
$\forall p, \forall c, \forall c', Card(p,c) \land Card(p,c') \Rightarrow c = c'$

$w(S_1) = 1$ and $w(\neg S_1) = -1$

$w(S_2) = 1$ and $w(\neg S_2) = -1$

[VdB'15]

Playing Cards Revisited

$$\forall p, \exists c, Card(p,c)$$
$$\forall c, \exists p, Card(p,c)$$
$$\forall p, \forall c, \forall c', Card(p,c) \wedge Card(p,c') \Rightarrow c = c'$$

↓ Skolemization

$$\forall p, \forall c, Card(p,c) \Rightarrow S_1(p)$$
$$\forall c, \forall p, Card(p,c) \Rightarrow S_2(c)$$
$$\forall p, \forall c, \forall c', Card(p,c) \wedge Card(p,c') \Rightarrow c = c'$$

$w(S_1) = 1$ and $w(\neg S_1) = -1$

$w(S_2) = 1$ and $w(\neg S_2) = -1$

↓ Atom counting

[VdB'15]

Playing Cards Revisited

$\forall p, \exists c, Card(p,c)$
$\forall c, \exists p, Card(p,c)$
$\forall p, \forall c, \forall c', Card(p,c) \wedge Card(p,c') \Rightarrow c = c'$

↓ Skolemization

$\forall p, \forall c, Card(p,c) \Rightarrow S_1(p)$
$\forall c, \forall p, Card(p,c) \Rightarrow S_2(c)$
$\forall p, \forall c, \forall c', Card(p,c) \wedge Card(p,c') \Rightarrow c = c'$

$w(S_1) = 1$ and $w(\neg S_1) = -1$

$w(S_2) = 1$ and $w(\neg S_2) = -1$

↓ Atom counting

$\forall p, \forall c, \forall c', Card(p,c) \wedge Card(p,c') \Rightarrow c = c'$

[VdB'15]

Playing Cards Revisited

$$\forall p, \exists c, Card(p,c)$$
$$\forall c, \exists p, Card(p,c)$$
$$\forall p, \forall c, \forall c', Card(p,c) \wedge Card(p,c') \Rightarrow c = c'$$

↓ Skolemization

$$\forall p, \forall c, Card(p,c) \Rightarrow S_1(p)$$
$$\forall c, \forall p, Card(p,c) \Rightarrow S_2(c)$$
$$\forall p, \forall c, \forall c', Card(p,c) \wedge Card(p,c') \Rightarrow c = c'$$

$w(S_1) = 1$ and $w(\neg S_1) = -1$

$w(S_2) = 1$ and $w(\neg S_2) = -1$

↓ Atom counting

$$\forall p, \forall c, \forall c', Card(p,c) \wedge Card(p,c') \Rightarrow c = c'$$

↓ ∀-Rule

[VdB'15]

Playing Cards Revisited

$\forall p, \exists c, Card(p,c)$
$\forall c, \exists p, Card(p,c)$
$\forall p, \forall c, \forall c', Card(p,c) \land Card(p,c') \Rightarrow c = c'$

↓ Skolemization

$\forall p, \forall c, Card(p,c) \Rightarrow S_1(p)$
$\forall c, \forall p, Card(p,c) \Rightarrow S_2(c)$
$\forall p, \forall c, \forall c', Card(p,c) \land Card(p,c') \Rightarrow c = c'$

$w(S_1) = 1$ and $w(\neg S_1) = -1$

$w(S_2) = 1$ and $w(\neg S_2) = -1$

↓ Atom counting

$\forall p, \forall c, \forall c', Card(p,c) \land Card(p,c') \Rightarrow c = c'$

↓ ∀-Rule

$\forall c, \forall c', Card(c) \land Card(c') \Rightarrow c = c'$

[VdB'15]

Playing Cards Revisited

$\forall p, \exists c, Card(p,c)$
$\forall c, \exists p, Card(p,c)$
$\forall p, \forall c, \forall c', Card(p,c) \wedge Card(p,c') \Rightarrow c = c'$

↓ Skolemization

$\forall p, \forall c, Card(p,c) \Rightarrow S_1(p)$
$\forall c, \forall p, Card(p,c) \Rightarrow S_2(c)$
$\forall p, \forall c, \forall c', Card(p,c) \wedge Card(p,c') \Rightarrow c = c'$

$w(S_1) = 1$ and $w(\neg S_1) = -1$

$w(S_2) = 1$ and $w(\neg S_2) = -1$

↓ Atom counting

$\forall p, \forall c, \forall c', Card(p,c) \wedge Card(p,c') \Rightarrow c = c'$

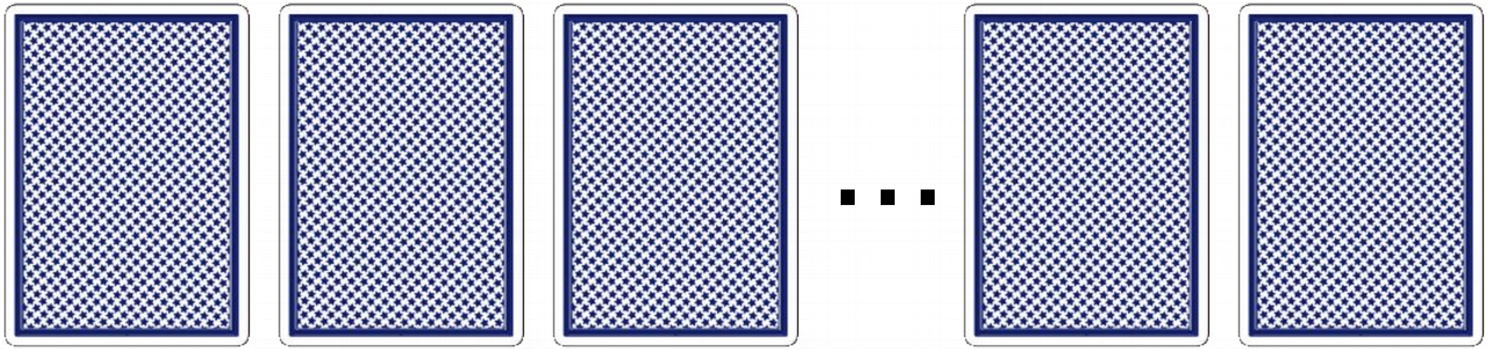
↓ ∀-Rule

$\forall c, \forall c', Card(c) \wedge Card(c') \Rightarrow c = c'$

↓ …

[VdB'15]

Playing Cards Revisited



Let us automate this:

- **Lifted** probabilistic inference algorithm

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

Computed in time polynomial in n

[VdB'15]

Summary Lifted Inference

- By definition: PTIME data complexity
  Also: ∃ FO compilation = ∃ Query Plan
- However: only works for "liftable" queries
- Preprocessing based on logical rewriting
- The rules: Deceptively simple: the only surprising rules are I/E and atom counting
- Rules are captured by a query plan
  or first-order NNF circuit

Outline

- Part 1: Motivation
- Part 2: Probabilistic Databases
- Part 3: Weighted Model Counting
- Part 4: Lifted Inference for WFOMC
- Part 5: Completeness of Lifted Inference
- Part 6: Query Compilation
- Part 7: Symmetric Lifted Inference Complexity
- Part 8: Open-World Probabilistic Databases
- Part 9: Discussion & Conclusions

Lifted Inference on Asymmetric DB

Preprocess Q (omitted from this talk; see [Suciu'11]), then apply these rules (some have preconditions)

$$P(\neg Q) = 1 - P(Q)$$

negation

$$P(Q1 \wedge Q2) = P(Q1)P(Q2)$$
$$P(Q1 \vee Q2) = 1 - (1 - P(Q1))(1 - P(Q2))$$

Independent join / union

$$P(\exists z\, Q) = 1 - \Pi_{A \in Domain} (1 - P(Q[A/z]))$$
$$P(\forall z\, Q) = \Pi_{A \in Domain} P(Q[A/z])$$

Independent project

$$P(Q1 \wedge Q2) = P(Q1) + P(Q2) - P(Q1 \vee Q2)$$
$$P(Q1 \vee Q2) = P(Q1) + P(Q2) - P(Q1 \wedge Q2)$$

Inclusion/ exclusion

Example: Liftable Clause

$Q = \forall x \forall y\ S(x,y) \Rightarrow R(y)$

$= \forall y\ (\exists x\ S(x,y) \Rightarrow R(y))$

Example: Liftable Clause

$Q = \forall x \forall y\ S(x,y) \Rightarrow R(y)$

$= \forall y\ (\exists x\ S(x,y) \Rightarrow R(y))$

$P(Q) = \Pi_{B \in Domain}\ P(\exists x\ S(x,B) \Rightarrow R(B))$

Indep. ∀

Example: Liftable Clause

$Q = \forall x \forall y\ S(x,y) \Rightarrow R(y)$ $\quad = \forall y\ (\exists x\ S(x,y) \Rightarrow R(y))$

$P(Q) = \Pi_{B \in Domain}\ P(\exists x\ S(x,B) \Rightarrow R(B))$

Indep. ∀

$P(Q) = \Pi_{B \in Domain}\ [1 - P(\exists x\ S(x,B)) \times (1 - P(R(b)))]$

Indep. or:
$P(X \Rightarrow Y) =$
$= P(\neg X \vee Y)$
$= P(X)\ (1-P(Y))$

Example: Liftable Clause

$Q = \forall x \forall y\, S(x,y) \Rightarrow R(y)$ $\quad = \forall y\, (\exists x\, S(x,y) \Rightarrow R(y))$

$$P(Q) = \Pi_{B \in Domain} P(\exists x\, S(x,B) \Rightarrow R(B))$$

Indep. ∀

$$P(Q) = \Pi_{B \in Domain} [1 - P(\exists x\, S(x,B)) \times (1 - P(R(b)))]$$

Indep. or:
$P(X \Rightarrow Y) =$
$= P(\neg X \vee Y)$
$= P(X)\,(1-P(Y))$

$$P(Q) = \Pi_{B \in Domain} [1 - (1 - \Pi_{A \in Domain} (1-P(S(A,B)))) \times (1 - P(R(B)))]$$

Indep. ∃

Example: Liftable Clause

$Q = \forall x \forall y\ S(x,y) \Rightarrow R(y)$

$= \forall y\ (\exists x\ S(x,y) \Rightarrow R(y))$

$P(Q) = \Pi_{B \in Domain}\ P(\exists x\ S(x,B) \Rightarrow R(B))$

Indep. ∀

$P(Q) = \Pi_{B \in Domain}\ [1 - P(\exists x\ S(x,B)) \times (1 - P(R(b)))]$

Indep. or:
$P(X \Rightarrow Y) =$
$= P(\neg X \vee Y)$
$= P(X)\ (1-P(Y))$

$P(Q) = \Pi_{B \in Domain}\ [1 - (1 - \Pi_{A \in Domain}\ (1-P(S(A,B)))) \times (1 - P(R(B)))]$

Indep. ∃

Lookup the probabilities in D

Example: Liftable Clause

$Q = \forall x \forall y\ S(x,y) \Rightarrow R(y)$ $\quad = \forall y\ (\exists x\ S(x,y) \Rightarrow R(y))$

$P(Q) = \Pi_{B \in Domain}\ P(\exists x\ S(x,B) \Rightarrow R(B))$ Indep. ∀

$P(Q) = \Pi_{B \in Domain}\ [1 - P(\exists x\ S(x,B)) \times (1 - P(R(b)))]$

Indep. or:
P(X⇒Y) =
= P(¬X ∨ Y)
= P(X) (1-P(Y))

$P(Q) = \Pi_{B \in Domain}\ [1 - (1 - \Pi_{A \in Domain}(1-P(S(A,B)))) \times (1 - P(R(B)))]$

Indep. ∃

Lookup the probabilities in D

Runtime = $O(n^2)$.

Two Questions

- Question 1: Are the lifted rules complete?
  - We know that they get stuck on some queries
  - Should we add more rules?
- Question 2: Are lifted rules stronger than grounded?
  - Lifted rules can also be grounded
  - Any advantage over grounded inference?

Two Questions

- Question 1: Are the lifted rules complete?
  - We know that they get stuck on some queries
  - Should we add more rules?

Complete for "unate ∀FO" and for "unate ∃FO"

- Question 2: Are lifted rules stronger than grounded?
  - Lifted rules can also be grounded
  - Any advantage over grounded inference?

Two Questions

- Question 1: Are the lifted rules complete?
  - We know that they get stuck on some queries
  - Should we add more rules?

Complete for "unate ∀FO" and for "unate ∃FO"

- Question 2: Are lifted rules stronger than grounded?
  - Lifted rules can also be grounded
  - Any advantage over grounded inference?

Strictly stronger than DPLL-based algorithms

FO^{un} = Unate FO

An FO sentence is ***unate*** if:

- Negations occur only on atoms
- Every relational symbol R either occurs only positively, or only negatively

$\forall FO^{un}$ ($\exists FO^{un}$) = restrict quantifiers too

$$Q = \forall x \forall y\, (Smoker(x) \vee \neg Friend(x,y)) \wedge \forall x \forall y\, (\neg Friend(x,y) \vee Drinker(y))$$

Unate

$$Q = \forall x \forall y\, (Smoker(x) \vee \neg Friend(x,y)) \wedge \forall x \forall y\, (Friend(x,y) \vee \neg Drinker(y))$$

Not unate

1. Are the Lifted Rules Complete?

We use complexity classes

- Inference rules: PTIME data complexity
- Some queries: #P-hard data complexity

Dichotomy Theorem for $\forall FO^{un}$ (or $\exists FO^{un}$)

- If lifted rules succeed, then query in PTIME
- If lifted rules fail, then query is #P-hard

Implies lifted rules are complete for $\forall FO^{un}$, $\exists FO^{un}$

Will show in two steps: **Small** and **Big Dichotomy Theorem**

NP v.s. #P

Decision Problems:

- SAT = Satisfiability Problem
- SAT is NP-complete [Cook'71]

Counting Problems:

- #SAT = model counting
- #SAT is #P-complete [Valiant'79]

Note: it would be wrong to say "#SAT is NP-complete"

Positive Partitioned 2CNF

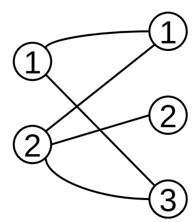
A PP2CNF is:

$$F = \bigwedge_{(i,j) \in E} (x_i \vee y_j)$$

where E = the edge set of a bipartite graph

$F = (x_1 \vee y_1) \wedge (x_2 \vee y_1) \wedge (x_2 \vee y_3) \wedge (x_1 \vee y_3) \wedge (x_2 \vee y_2)$

E:



Theorem [Provan'83] #PP2CNF is #P-hard

Unliftable Clause

$H_0 = \forall x \forall y\ (R(x) \lor S(x,y) \lor T(y))$

Independent Project not possible:
For $A_1 \neq A_2$,
$H_0[A_1/x]$ and $H_0[A_2/x]$ are dependent!

Unliftable Clause

$H_0 = \forall x \forall y\, (R(x) \vee S(x,y) \vee T(y))$

Independent Project not possible:
For $A_1 \neq A_2$,
$H_0[A_1/x]$ and $H_0[A_2/x]$
are dependent!

Theorem. Computing $P_D(H_0)$ is #P-hard in the size of D

[Dalvi&S.2004]

Unliftable Clause

Independent Project not possible:
For $A_1 \neq A_2$,
$H_0[A_1/x]$ and $H_0[A_2/x]$
are dependent!

$H_0 = \forall x \forall y\ (R(x) \vee S(x,y) \vee T(y))$

Theorem. Computing $P_D(H_0)$ is #P-hard in the size of D

[Dalvi&S.2004]

Proof: PP2CNF: $F = (X_{i1} \vee Y_{j1}) \wedge (X_{i2} \vee Y_{j2}) \wedge \ldots$ reduce #F to computing $P_D(H_0)$

By example:

Unliftable Clause

$H_0 = \forall x \forall y\ (R(x) \lor S(x,y) \lor T(y))$

Independent Project not possible:
For $A_1 \neq A_2$,
$H_0[A_1/x]$ and $H_0[A_2/x]$
are dependent!

Theorem. Computing $P_D(H_0)$ is #P-hard in the size of D

[Dalvi&S.2004]

Proof: PP2CNF: $F = (X_{i1} \lor Y_{j1}) \land (X_{i2} \lor Y_{j2}) \land \ldots$ reduce #F to computing $P_D(H_0)$

By example:

$$F = (X_1 \lor Y_1) \land (X_1 \lor Y_2) \land (X_2 \lor Y_2)$$

Unliftable Clause

$H_0 = \forall x \forall y\ (R(x) \vee S(x,y) \vee T(y))$

Independent Project not possible:
For $A_1 \neq A_2$,
$H_0[A_1/x]$ and $H_0[A_2/x]$
are dependent!

Theorem. Computing $P_D(H_0)$ is #P-hard in the size of D

[Dalvi&S.2004]

Proof: PP2CNF: $F = (X_{i1} \vee Y_{j1}) \wedge (X_{i2} \vee Y_{j2}) \wedge \ldots$ reduce #F to computing $P_D(H_0)$

By example:

$F = (X_1 \vee Y_1) \wedge (X_1 \vee Y_2) \wedge (X_2 \vee Y_2)$

D (tuples not shown have P=1)

R

X	P
x_1	0.5
x_2	0.5

S

X	Y	P
x_1	y_1	0
x_1	y_2	0
x_2	y_2	0

T

Y	P
y_1	0.5
y_2	0.5

Unliftable Clause

$H_0 = \forall x \forall y\, (R(x) \vee S(x,y) \vee T(y))$

Independent Project not possible:
For $A_1 \neq A_2$,
$H_0[A_1/x]$ and $H_0[A_2/x]$
are dependent!

Theorem. Computing $P_D(H_0)$ is #P-hard in the size of D

[Dalvi&S.2004]

Proof: PP2CNF: $F = (X_{i1} \vee Y_{j1}) \wedge (X_{i2} \vee Y_{j2}) \wedge \ldots$ reduce #F to computing $P_D(H_0)$

By example:

$F = (X_1 \vee Y_1) \wedge (X_1 \vee Y_2) \wedge (X_2 \vee Y_2)$

$P_D(H_0) = P(F)$; hence $P_D(H_0)$ is #P-hard

D (tuples not shown have P=1)

R

X	P
x_1	0.5
x_2	0.5

S

X	Y	P
x_1	y_1	0
x_1	y_2	0
x_2	y_2	0

T

Y	P
y_1	0.5
y_2	0.5

Hierarchical Queries

Fix Q; at(x) = set of atoms (=literals) containing the variable x

Definition Q is hierarchical if forall variables x, y:

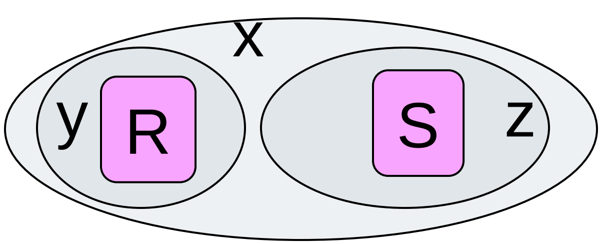
$at(x) \subseteq at(y)$ or $at(x) \supseteq at(y)$ or $at(x) \cap at(y) = \emptyset$

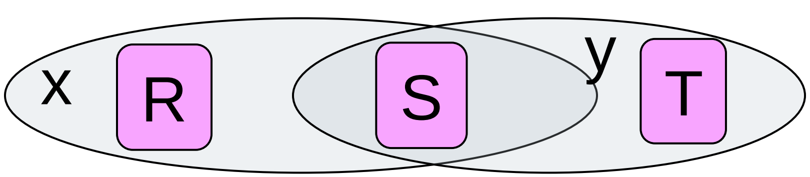
Hierarchical

$Q = \forall x \forall y \forall z (S(x,y) \vee T(x,z))$

Non-hierarchical

$H_0 = \forall x \forall y\ (R(x) \vee S(x,y) \vee T(y))$





The Small Dichotomy Theorem

[Dalvi&S.04]

<u>Theorem</u> Let Q be <u>*one clause, with no repeated symbols*</u>

- If Q is hierarchical, then $P_D(Q)$ is in PTIME.
- If Q is not hierarchical then $P_D(Q)$ is #P-hard.

Checking "Q is hierarchical" is in AC^0 (expression complexity)

The Small Dichotomy Theorem

[Dalvi&S.04]

<u>Theorem</u> Let Q be *<u>one clause, with no repeated symbols</u>*
- If Q is hierarchical, then $P_D(Q)$ is in PTIME.
- If Q is not hierarchical then $P_D(Q)$ is #P-hard.

Checking "Q is hierarchical" is in AC^0 (expression complexity)

[Dalvi,S.'12]

Fact: Any non-hierarchical Q in $\forall FO^{un}$ ($\exists FO^{un}$) is #P-hard

Next: consider only hierarchical queries in $\forall FO^{un}$ ($\exists FO^{un}$)

Clause with Repeated Symbols

$Q_J = \forall x_1 \forall y_1 \forall x_2 \forall y_2\ (S(x_1,y_1) \lor R(y_1) \lor S(x_2,y_2) \lor T(y_2))$

Clause with Repeated Symbols

$Q_J = \forall x_1 \forall y_1 \forall x_2 \forall y_2 \ (S(x_1,y_1) \lor R(y_1) \lor S(x_2,y_2) \lor T(y_2))$

$= \underbrace{[\forall x_1 \forall y_1 S(x_1,y_1) \lor R(y_1)]}_{Q_1} \lor \underbrace{[\forall x_2 \forall y_2 S(x_2,y_2) \lor T(y_2)]}_{Q_2}$

Clause with Repeated Symbols

$Q_J = \forall x_1 \forall y_1 \forall x_2 \forall y_2\ (S(x_1,y_1) \vee R(y_1) \vee S(x_2,y_2) \vee T(y_2))$

$= \underbrace{[\forall x_1 \forall y_1 S(x_1,y_1) \vee R(y_1)]}_{Q_1} \vee \underbrace{[\forall x_2 \forall y_2 S(x_2,y_2) \vee T(y_2)]}_{Q_2}$

$P(Q_J) = P(Q_1) + P(Q_2) - P(Q_1 \wedge Q_2)$

PTIME (have seen before)

Clause with Repeated Symbols

$$Q_J = \forall x_1 \forall y_1 \forall x_2 \forall y_2\ (S(x_1,y_1) \vee R(y_1) \vee S(x_2,y_2) \vee T(y_2))$$

$$= \underbrace{[\forall x_1 \forall y_1 S(x_1,y_1) \vee R(y_1)]}_{Q_1} \vee \underbrace{[\forall x_2 \forall y_2 S(x_2,y_2) \vee T(y_2)]}_{Q_2}$$

$$P(Q_J) = P(Q_1) + P(Q_2) - P(Q_1 \wedge Q_2)$$

PTIME (have seen before)

$y = y1 = y2$

$$Q_1 \wedge Q_2 = \forall y\ [(\forall x_1 S(x_1,y) \vee R(y)) \wedge (\forall x_2 S(x_2,y)) \vee T(y)]$$

$$= \forall y\ [\forall x\ S(x,y) \vee (R(y) \wedge T(y))]$$

Clause with Repeated Symbols

$$Q_J = \forall x_1 \forall y_1 \forall x_2 \forall y_2\ (S(x_1,y_1) \vee R(y_1) \vee S(x_2,y_2) \vee T(y_2))$$

$$= \underbrace{[\forall x_1 \forall y_1 S(x_1,y_1) \vee R(y_1)]}_{Q_1} \vee \underbrace{[\forall x_2 \forall y_2 S(x_2,y_2) \vee T(y_2)]}_{Q_2}$$

$$P(Q_J) = P(Q_1) + P(Q_2) - P(Q_1 \wedge Q_2)$$

PTIME (have seen before)

$y = y1 = y2$

$$Q_1 \wedge Q_2 = \forall y\ [(\forall x_1 S(x_1,y) \vee R(y)) \wedge (\forall x_2 S(x_2,y)) \vee T(y)]$$
$$= \forall y\ [\forall x\ S(x,y) \vee (R(y) \wedge T(y))]$$

$$P(Q_1 \wedge Q_2) = \Pi_{B \in Domain}\ P[\forall x.S(x,B) \vee (R(B) \wedge T(B))] = \ldots etc$$

Runtime = $O(n^2)$.

Unliftable Queries H_k

Will drop ∀ to reduce clutter

$H_0 = R(x) \vee S(x,y) \vee T(y)$

$H_1 = [R(x_0) \vee S(x_0,y_0)] \wedge [S(x_1,y_1) \vee T(y_1)]$

Every H_k, $k \geq 1$ is hierarchical

Unliftable Queries H_k

Will drop ∀ to reduce clutter

$H_0 = R(x) \vee S(x,y) \vee T(y)$

$H_1 = [R(x_0) \vee S(x_0,y_0)] \wedge [S(x_1,y_1) \vee T(y_1)]$

$H_2 = [R(x_0) \vee S_1(x_0,y_0)] \wedge [S_1(x_1,y_1) \vee S_2(x_1,y_1)] \vee [S_2(x_2,y_2) \vee T(y_2)]$

Every H_k, $k \geq 1$
is hierarchical

Unliftable Queries H_k

Will drop ∀ to reduce clutter

H_0= R(x)∨S(x,y)∨T(y)

H_1= [R(x_0)∨S(x_0,y_0)] ∧ [S(x_1,y_1)∨T(y_1)]

H_2= [R(x_0)∨S_1(x_0,y_0)] ∧ [S_1(x_1,y_1)∨S_2(x_1,y_1)] ∨ [S_2(x_2,y_2)∨T(y_2)]

H_3= [R(x_0)∨S_1(x_0,y_0)]∧[S_1(x_1,y_1)∨S_2(x_1,y_1)]∧[S_2(x_2,y_2)∨S_3(x_2,y_2)]∧[S_3(x_3,y_3)∨T(y_3)]

. . .

Every H_k, $k \geq 1$
is hierarchical

Unliftable Queries H_k

Will drop ∀ to reduce clutter

$H_0 = R(x) \vee S(x,y) \vee T(y)$

$H_1 = [R(x_0) \vee S(x_0,y_0)] \wedge [S(x_1,y_1) \vee T(y_1)]$

$H_2 = [R(x_0) \vee S_1(x_0,y_0)] \wedge [S_1(x_1,y_1) \vee S_2(x_1,y_1)] \vee [S_2(x_2,y_2) \vee T(y_2)]$

$H_3 = [R(x_0) \vee S_1(x_0,y_0)] \wedge [S_1(x_1,y_1) \vee S_2(x_1,y_1)] \wedge [S_2(x_2,y_2) \vee S_3(x_2,y_2)] \wedge [S_3(x_3,y_3) \vee T(y_3)]$

. . .

Every H_k, $k \geq 1$ is hierarchical

Theorem. [Dalvi&S'12] Every query H_k is #P-hard

A Closer Look at H_k

If we drop any one clause → in PTIME

$H_3 = [R(x_0) \vee S_1(x_0,y_0)] \wedge [$~~$S_1(x_1,y_1) \vee S_2(x_1,y_1)$~~$] \wedge [S_2(x_2,y_2) \vee S_3(x_2,y_2)] \wedge [S_3(x_3,y_3) \vee T(y_3)]$

Independent join

A Closer Look at H_k

If we drop any one clause → in PTIME

$$H_3 = [R(x_0) \vee S_1(x_0,y_0)] \wedge \cancel{[S_1(x_1,y_1) \vee S_2(x_1,y_1)]} \wedge [S_2(x_2,y_2) \vee S_3(x_2,y_2)] \wedge [S_3(x_3,y_3) \vee T(y_3)]$$

Independent join

If we replace $T(y_3)$ with $\mathbf{T(x_3)}$ then in PTIME

$$[R(x_0) \wedge S_1(x_0,y_0)] \wedge [S_1(x_1,y_1) \vee S_2(x_1,y_1)] \wedge [S_2(x_2,y_2) \vee S_3(x_2,y_2)] \wedge [S_3(x_3,y_3) \vee \mathbf{T(x_3)}]$$

Independent project on $x_0 = x_1 = x_2 = x_3$

Cancellations

Q_W = a Boolean expression
over the clauses in H_3 Yet, in PTIME

$$Q_W = [(R(x_0) \vee S_1(x_0,y_0)) \wedge (S_2(x_2,y_2) \vee S_3(x_2,y_2))] \vee \quad /* Q_1 */$$

$$[(R(x_0) \vee S_1(x_0,y_0)) \wedge (S_3(x_3,y_3) \vee T(y_3))] \vee \quad /* Q_2 */$$

$$[(S_1(x_1,y_1) \vee S_2(x_1,y_1)) \wedge (S_3(x_3,y_3) \vee T(y_3))] \quad /* Q_3 */$$

Cancellations

Q_W = a Boolean expression over the clauses in H_3 Yet, in PTIME

$$
\begin{aligned}
Q_W = &[(R(x_0) \vee S_1(x_0,y_0)) \wedge (S_2(x_2,y_2) \vee S_3(x_2,y_2))] \vee &&/* Q_1 */\\
&[(R(x_0) \vee S_1(x_0,y_0)) \wedge (S_3(x_3,y_3) \vee T(y_3))] \vee &&/* Q_2 */\\
&[(S_1(x_1,y_1) \vee S_2(x_1,y_1)) \wedge (S_3(x_3,y_3) \vee T(y_3))] &&/* Q_3 */
\end{aligned}
$$

$$
\begin{aligned}
P(Q_W) = \; & P(Q_1) + P(Q_2) + P(Q_3) + \\
& - P(Q_1 \wedge Q_2) - P(Q_2 \wedge Q_3) - P(Q_1 \wedge Q_3) \\
& + P(Q_1 \wedge Q_2 \wedge Q_3)
\end{aligned}
$$

= H_3 (hard !)

Also = H_3

Cancellations

Q_W = a Boolean expression over the clauses in H_3 Yet, in PTIME

$$Q_W = [(R(x_0) \vee S_1(x_0,y_0)) \wedge (S_2(x_2,y_2) \vee S_3(x_2,y_2))] \vee \quad /* Q_1 */$$
$$[(R(x_0) \vee S_1(x_0,y_0)) \wedge (S_3(x_3,y_3) \vee T(y_3))] \vee \quad /* Q_2 */$$
$$[(S_1(x_1,y_1) \vee S_2(x_1,y_1)) \wedge (S_3(x_3,y_3) \vee T(y_3))] \quad /* Q_3 */$$

$$P(Q_W) = P(Q_1) + P(Q_2) + P(Q_3) + - P(Q_1 \wedge Q_2) - P(Q_2 \wedge Q_3) - \cancel{P(Q_1 \wedge Q_3)} + \cancel{P(Q_1 \wedge Q_2 \wedge Q_3)}$$

= H_3 (hard !)

Also = H_3

Need to cancel terms to compute the query in PTIME

Using Mobius' function in the the lattice of Q's minterms [Suciu'11]

The Big Dichotomy Theorem

Call Q *liftable* if the rules don't get stuck.

Dichotomy Theorem [Dalvi'12] Fix a $\forall FO^{un}$ query Q.
1. If Q is liftable, then P(Q) is in PTIME
2. If Q is not liftable, then P(Q) is #P-complete

Note Original formulation for UCQ;
Immediate extension to $\forall FO^{un}$ and for $\exists FO^{un}$

Discussion

- This answers Question 1: lifted inference rules are complete for $\forall FO^{un}$ (and for $\exists FO^{un}$)

- Notice: we did not use any symmetries!

- Beyond unate FO? Conjectures:
  - Rules+resolution* complete for CNF-FO
  - No complete set of rules for FO

* $Q = \forall x \forall y\, (R(x) \vee S(x,y)) \wedge \forall x \forall y\, (\neg S(x,y) \vee T(y))$

$= \forall x \forall y\, (R(x) \vee S(x,y)) \wedge \forall x \forall y\, (\neg S(x,y) \vee T(y)) \wedge \boldsymbol{\forall x \forall y\, (R(x) \vee T(y))}$

Outline

- Part 1: Motivation
- Part 2: Probabilistic Databases
- Part 3: Weighted Model Counting
- Part 4: Lifted Inference for WFOMC
- Part 5: Completeness of Lifted Inference
- Part 6: Query Compilation
- Part 7: Symmetric Lifted Inference Complexity
- Part 8: Open-World Probabilistic Databases
- Part 9: Discussion & Conclusions

Question 2. Are lifted rules stronger than grounded?

Alternative to lifting:

1. Ground the FO sentence
2. Do WMC on the propositional formula

- There is no reason why grounded inference should be weaker than lifted inference
- However, *existing* grounded algorithms are strictly weaker than lifted inference

Algorithms for Model Counting

[Gomes'08] Based on full search DPLL:

- Shannon expansion.

  $\#F = \#F[X=0] + \#F[X=1]$

- Caching.

  Store #F, look it up later

- Components. If $Vars(F1) \cap Vars(F2) = \emptyset$:

  $\#(F1 \wedge F2) = \#F1 * \#F2$

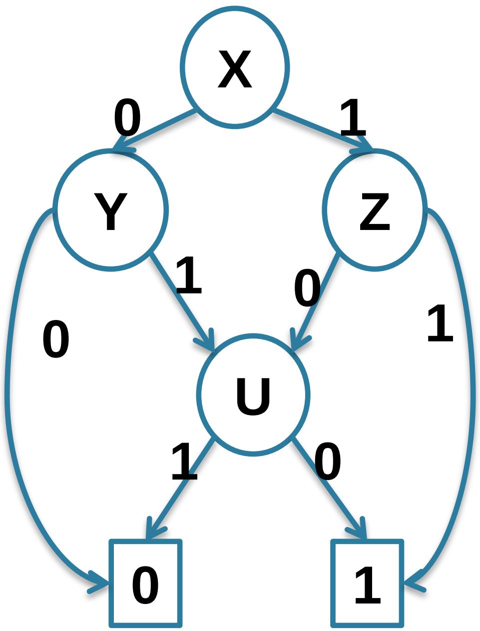
Knowledge Compilation

Definition (informal): represent the Boolean formula F in a circuit where WMC(F) is in PTIME in the size of the representation

Why we care:

- The trace of any inference algorithm is a knowledge compilation
- Lower bounds on size(KC) give lower bounds on the algorithm's runtime

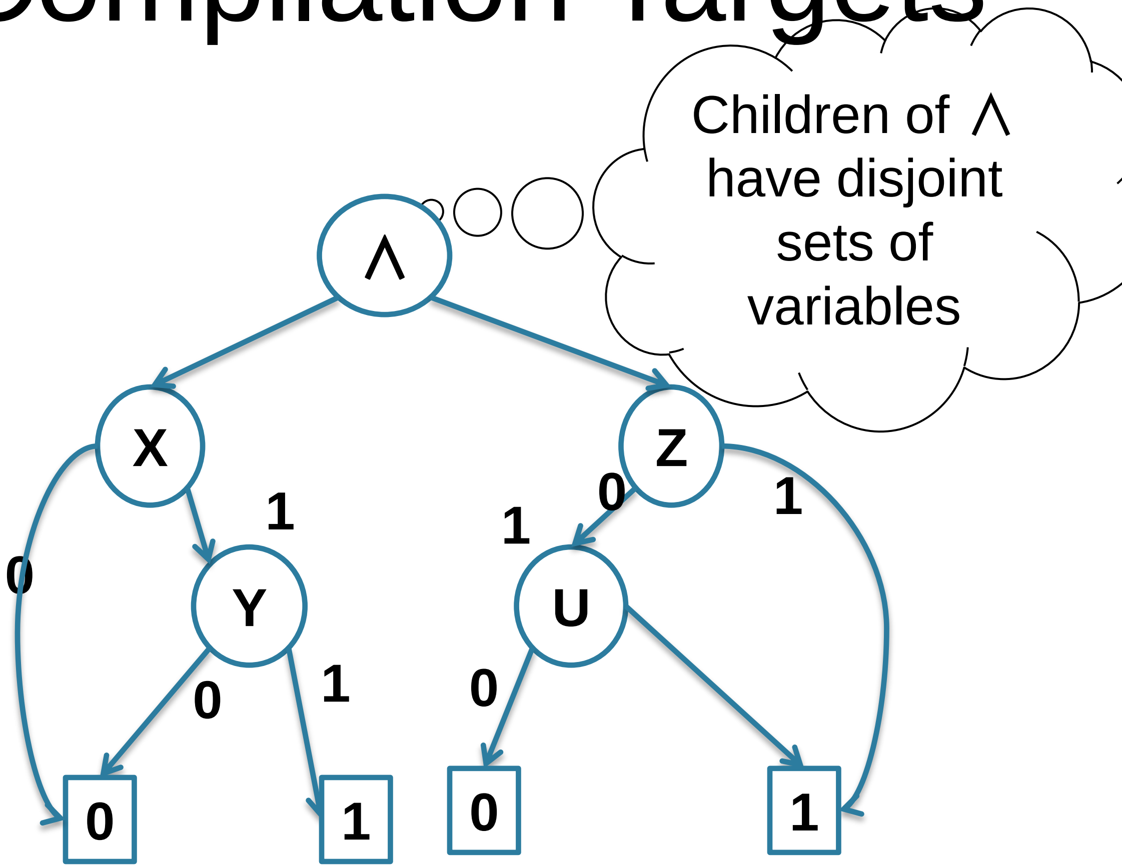
Knowledge Compilation Targets



FBDD:

Decision-, sink-nodes

OBDD: fixed variable order



Decision-DNNF

add: ∧-nodes

[Huang&Darwiche'2005]

DPLL and Knowledge Compilation

Fact: Trace of full-search DPLL → KC:

- Basic DPLL
  - → decision trees
- DPLL + caching
  - → OBDD (fixed variable order)
  - → FBDD
- DPLL + caching + components
  - → decision-DNNF

Hard Queries

H_0 = ∀x∀y (R(x) ∨ S(x,y) ∨ T(y)) = non-hierarchical
H_k = hierarchical, has inversion, for $k \geq 1$

Grounded Boolean formulas:
$F_n(H_0) = \wedge_{i\in[n],\, j\in[n]} (R_i \vee S_{ij} \vee T_j)$

Th. [Beame'14] Any FBDD for $F_n(H_k)$ has size $\geq 2^{n-1}/n$.
Same holds for any non-hierarchical query.

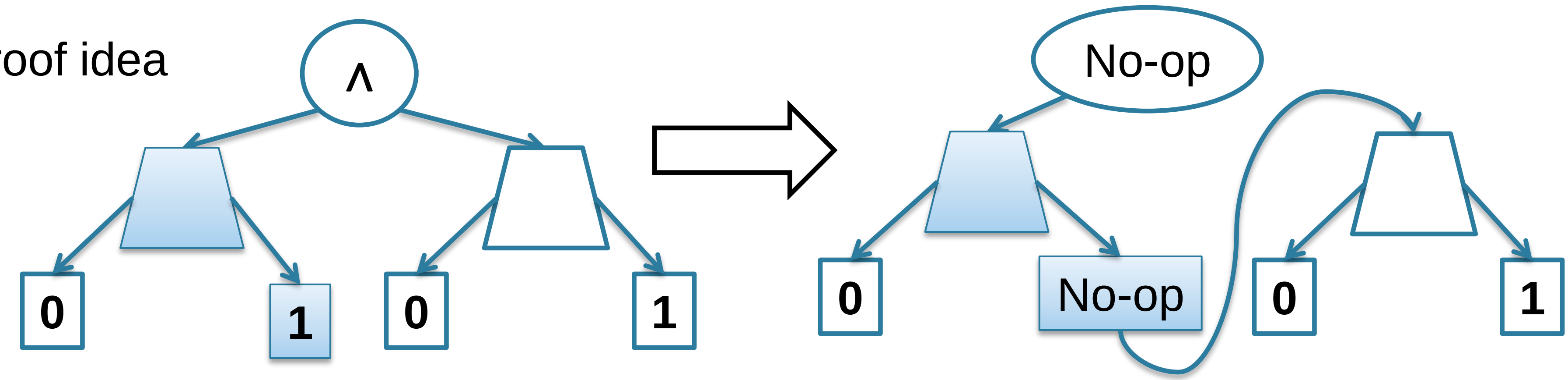
What about Decision-DNNFs?

Decision-DNNF to FBDD

Optimal [Razgon]

Theorem If F has a Decision-DNNF with N nodes, then F has an FBDD with at most $N^{1+\log(N)}$ nodes.

Proof idea

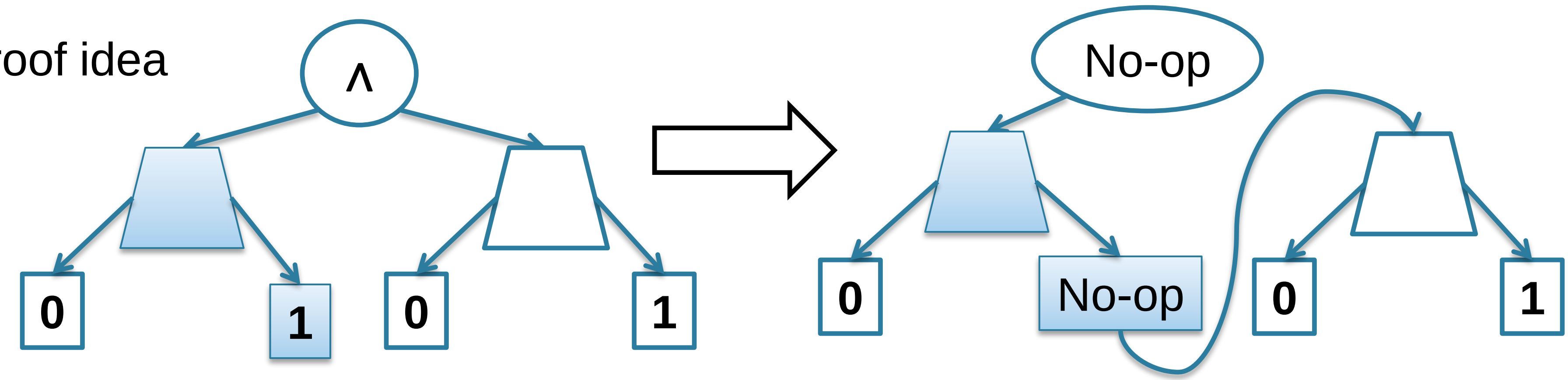


Decision-DNNF to FBDD

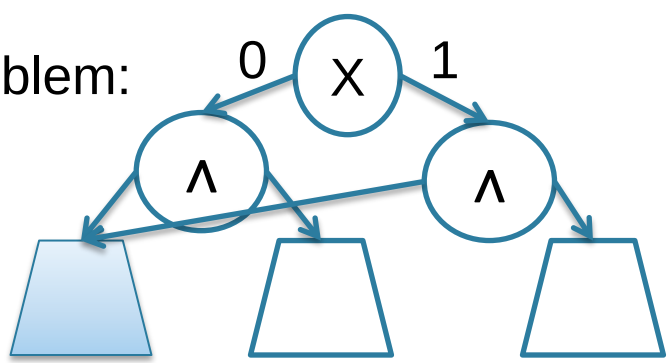
Optimal [Razgon]

Theorem If F has a Decision-DNNF with N nodes, then F has an FBDD with at most $N^{1+\log(N)}$ nodes.

Proof idea



Problem:

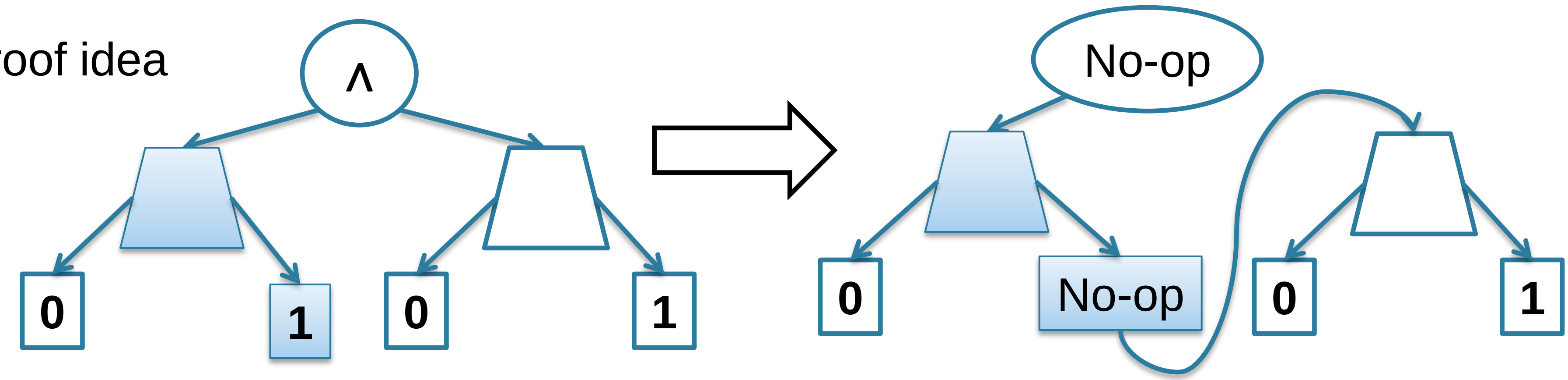


Decision-DNNF to FBDD

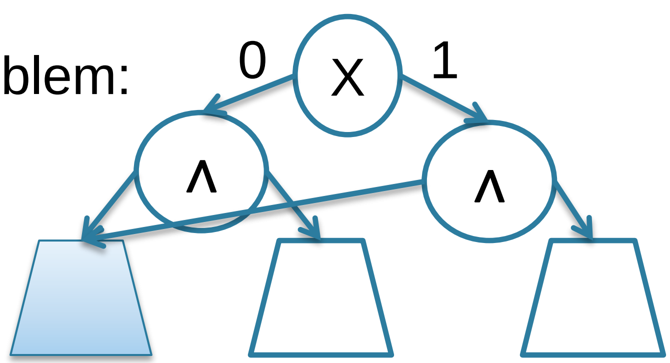
Optimal [Razgon]

Theorem If F has a Decision-DNNF with N nodes, then F has an FBDD with at most $N^{1+\log(N)}$ nodes.

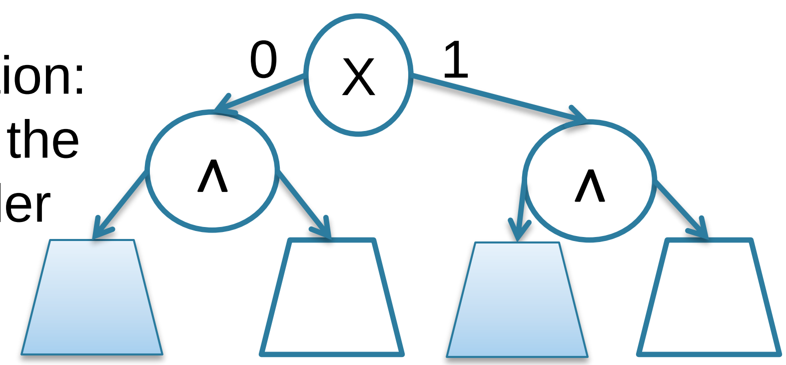
Proof idea



Problem:



Solution: copy the smaller child



Hard Queries

Corollary Any Decision-DNNF for $F_n(H_k)$ has size $2^{\Omega(\sqrt{n})}$
Same holds for any non-hierarchical query.

Proof. N-node Decision-DNNF to $N^{1+\log(N)}$ nodes FBDD.

$$N^{1+\log(N)} > 2^{n-1}/n \, ,$$
$$\log(N) + \log^2(N) > n - 1 - \log(n)$$
$$\log^2(N) = \Omega(n)$$
$$\log(N) = \Omega(\sqrt{n})$$

Lifted v.s. Grounded Inference

	Non-hierarchical Q (e.g. H_0)
Lifted P(Q)	#P-hard
Grounded $P(F_n(Q))$	$2^{\Omega(\sqrt{n})}$

What about hierarchical queries ?

Inversion-Free Queries

Definition An inversion in Q is a sequence of co-occurring vars:

$(x_0,y_0), (x_1,y_1), \ldots, (x_k,y_k)$, such that:

- $at(x_0) \not\subseteq at(y_0)$, $at(x_1)=at(y_1), \ldots, at(x_{k-1})=at(y_{k-1})$, $at(x_k) \not\supseteq at(y_k)$
- For all i=1,..,k-1 there exists two atoms in Q of the form: $S_i(\ldots,x_{i-1},\ldots,y_{i-1},\ldots)$ and $S_i(\ldots,x_i, \ldots, y_i, \ldots)$

Inversion-free implies hierarchical, but converse fails

$Q=[R(x_0) \lor S(x_0,y_0)] \land [S(x_1,y_1) \lor T(x_1)]$

Inversion-free

Inversion

$H_1=[R(x_0) \lor S(x_0,y_0)] \land [S(x_1,y_1) \lor T(y_1)]$

Easy Queries

[Jha&S.11], [Beame'15]

Theorem Let Q in $\forall FO^{un}$

1. If Q has inversion then OBDD for $F_n(Q)$ has size $\geq 2^{n-1}/n$
2. Else, $F_n(Q)$ has OBDD of width $2^{\#atoms(Q)}$ (size $O(n)$)

Proof (part 2 only – next slide)

Easy Queries

[Beame&Liew'15] Extended to SDD.
Thus, over $\forall FO^{un}$, OBDD ≈ SDD

[Jha&S.11], [Beame'15]

Theorem Let Q in $\forall FO^{un}$

1. If Q has inversion then OBDD for $F_n(Q)$ has size $\geq 2^{n-1}/n$
2. Else, $F_n(Q)$ has OBDD of width $2^{\#atoms(Q)}$ (size $O(n)$)

Proof (part 2 only – next slide)

Easy Queries

[Bova'16] SDD more succint than OBDD (HWB)

[Beame&Liew'15] Extended to SDD.
Thus, over $\forall FO^{un}$, OBDD ≈ SDD

[Jha&S.11], [Beame'15]

Theorem Let Q in $\forall FO^{un}$

1. If Q has inversion then OBDD for $F_n(Q)$ has size ≥ $2^{n-1}/n$
2. Else, $F_n(Q)$ has OBDD of width $2^{\#atoms(Q)}$ (size $O(n)$)

Proof (part 2 only – next slide)

$Q = [R(x) \vee S(x,y)] \wedge [T(x') \vee S(x',y')]$

$Q = [R(x) \vee S(x,y)] \wedge [T(x') \vee S(x',y')]$

$n = 2$

$\Pi = \underbrace{R_1 T_1 S_{11} S_{12}}_{x = 1} \underbrace{R_2 T_2 S_{21} S_{22}}_{x = 2}$

$$C_1 = R(x) \vee S(x,y) \quad \wedge \quad C_2 = T(x') \ \wedge S(x',y') \quad = \quad Q = [R(x) \vee S(x,y)] \wedge [T(x') \vee S(x',y')]$$

$$n = 2$$

$$\Pi = \underbrace{R_1 T_1 S_{11} S_{12}}_{x = 1} \underbrace{R_2 T_2 S_{21} S_{22}}_{x = 2}$$

$C_1 = R(x) \vee S(x,y)$ $\wedge$ $C_2 = T(x') \wedge S(x',y')$ $=$ $Q = [R(x) \vee S(x,y)] \wedge [T(x') \vee S(x',y')]$

$$F_2(C_1) = (R_1 \vee S_{11}) \wedge (R_1 \vee S_{12}) \wedge (R_2 \vee S_{21}) \wedge (R_2 \vee S_{22})$$

$n = 2$

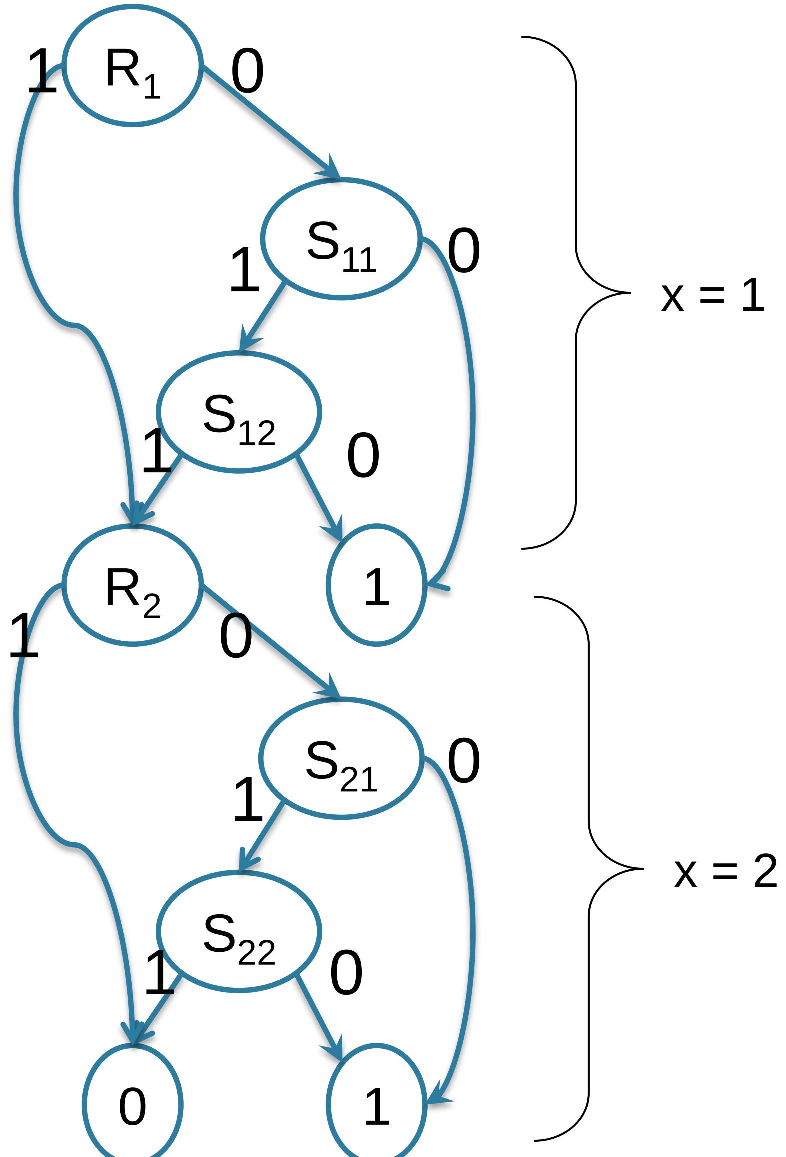
$$\Pi = \underbrace{R_1 T_1 S_{11} S_{12}}_{x = 1} \underbrace{R_2 T_2 S_{21} S_{22}}_{x = 2}$$

$C_1 = R(x) \vee S(x,y)$ $\wedge$ $C_2 = T(x') \wedge S(x',y')$ $=$ $Q = [R(x) \vee S(x,y)] \wedge [T(x') \vee S(x',y')]$

$F_2(C_1) = (R_1 \vee S_{11}) \wedge (R_1 \vee S_{12}) \wedge (R_2 \vee S_{21}) \wedge (R_2 \vee S_{22})$

$n = 2$

$\Pi = \underbrace{R_1 T_1 S_{11} S_{12}}_{x=1} \underbrace{R_2 T_2 S_{21} S_{22}}_{x=2}$



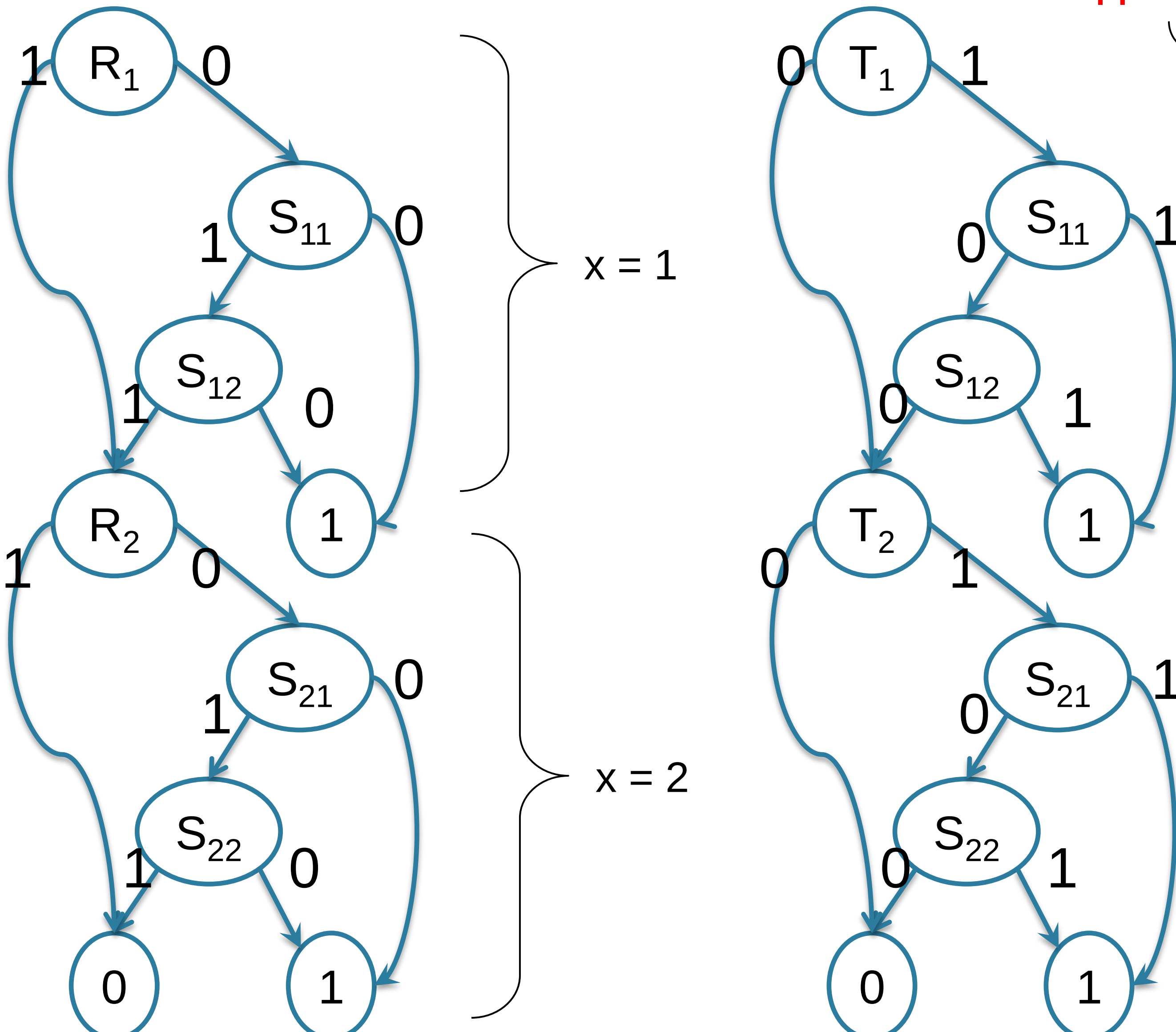
$C_1 = R(x) \vee S(x,y)$ $\wedge$ $C_2 = T(x') \wedge S(x',y')$ $=$ $Q = [R(x) \vee S(x,y)] \wedge [T(x') \vee S(x',y')]$

$F_2(C_1) = (R_1 \vee S_{11}) \wedge (R_1 \vee S_{12}) \wedge (R_2 \vee S_{21}) \wedge (R_2 \vee S_{22})$

$n = 2$

$\Pi = R_1 T_1 S_{11} S_{12} R_2 T_2 S_{21} S_{22}$

x = 1 x = 2



Same variable order Π in both OBDDs!

OBDD for
$Q = C_1 \wedge C_2$
has width =
width1 × width2

Lifted v.s. Grounded Inference

	Non-hierarchical Q (e.g. H_0)	Inversion-free Q
Lifted P(Q)	#P-hard	PTIME
Grounded $P(F_n(Q))$	$2^{\Omega(\sqrt{n})}$	PTIME

Easy/Hard Queries

Main result: a class of queries Q such that:

- Lifted inference: P((Q)) in PTIME
- Grounded inference: P(F_n(Q)) exponential time

Significance: limitation of DPLL-based algorithms for model counting

Clauses of H_k

$H_{k0} = \forall x \forall y\ R(x) \vee S_1(x,y)$

$H_{k1} = \forall x \forall y\ S_1(x,y) \vee S_2(x,y)$

$H_{k2} = \forall x \forall y\ S_2(x,y) \vee S_3(x,y)$

…

…

$H_{kk} = \forall x \forall y\ S_k(x,y) \vee T(y)$

Clauses of H_k

$H_{k0} = \forall x \forall y\ R(x) \lor S_1(x,y)$
$H_{k1} = \forall x \forall y\ S_1(x,y) \lor S_2(x,y)$
$H_{k2} = \forall x \forall y\ S_2(x,y) \lor S_3(x,y)$
…
…
$H_{kk} = \forall x \forall y\ S_k(x,y) \lor T(y)$

$f(Z_0, Z_1, \ldots, Z_k)$ = a Boolean function

Clauses of H_k

$H_{k0} = \forall x \forall y\, R(x) \lor S_1(x,y)$
$H_{k1} = \forall x \forall y\, S_1(x,y) \lor S_2(x,y)$
$H_{k2} = \forall x \forall y\, S_2(x,y) \lor S_3(x,y)$
…
…
$H_{kk} = \forall x \forall y\, S_k(x,y) \lor T(y)$

$f(Z_0, Z_1, \ldots, Z_k)$ = a Boolean function

$Q = f(H_{k0}, H_{k1}, \ldots, H_{kk})$

Clauses of H_k

$H_{k0} = \forall x \forall y\, R(x) \vee S_1(x,y)$
$H_{k1} = \forall x \forall y\, S_1(x,y) \vee S_2(x,y)$
$H_{k2} = \forall x \forall y\, S_2(x,y) \vee S_3(x,y)$
…
…
$H_{kk} = \forall x \forall y\, S_k(x,y) \vee T(y)$

$f(Z_0, Z_1, \ldots, Z_k)$ = a Boolean function

$Q = f(H_{k0}, H_{k1}, \ldots, H_{kk})$

Examples:

$f = Z_0 \wedge Z_1 \wedge \ldots \wedge Z_k$ then $f(H_{k0}, H_{k1}, \ldots, H_{kk}) = H_k$

$f = Z_0 \wedge Z_2 \vee Z_0 \wedge Z_3 \vee Z_1 \wedge Z_3$ then $f(H_{30}, H_{31}, H_{31}, H_{33}) = Q_W$

Easy/Hard Queries

[Beame'14]

Theorem For any Boolean function $f(Z_0, Z_1, \ldots, Z_k)$, denoting $Q = f(H_{k0}, H_{k1}, \ldots, H_{kk})$:

- Any FBDD for $F_n(Q)$ has size $2^{\Omega(n)}$
- Any Decision-DNNF has size $2^{\Omega(\sqrt{n})}$.

Consequence:

- Lifted inference computes compute $P(Q_W)$ in PTIME
- Any DPLL-based algorithm takes time $2^{\Omega(\sqrt{n})}$

Many other queries are like Q_W

Lifted v.s. Grounded Inference

	Non-hierarchical Q (e.g. H_0)	Inversion-free Q	Q = $f(H_{k0},\ldots,H_{kk})$
Lifted P(Q)	#P-hard	PTIME	PTIME or #P-hard
Grounded $P(F_n(Q))$	$2^{\Omega(\sqrt{n})}$	PTIME	$2^{\Omega(\sqrt{n})}$

Two Questions

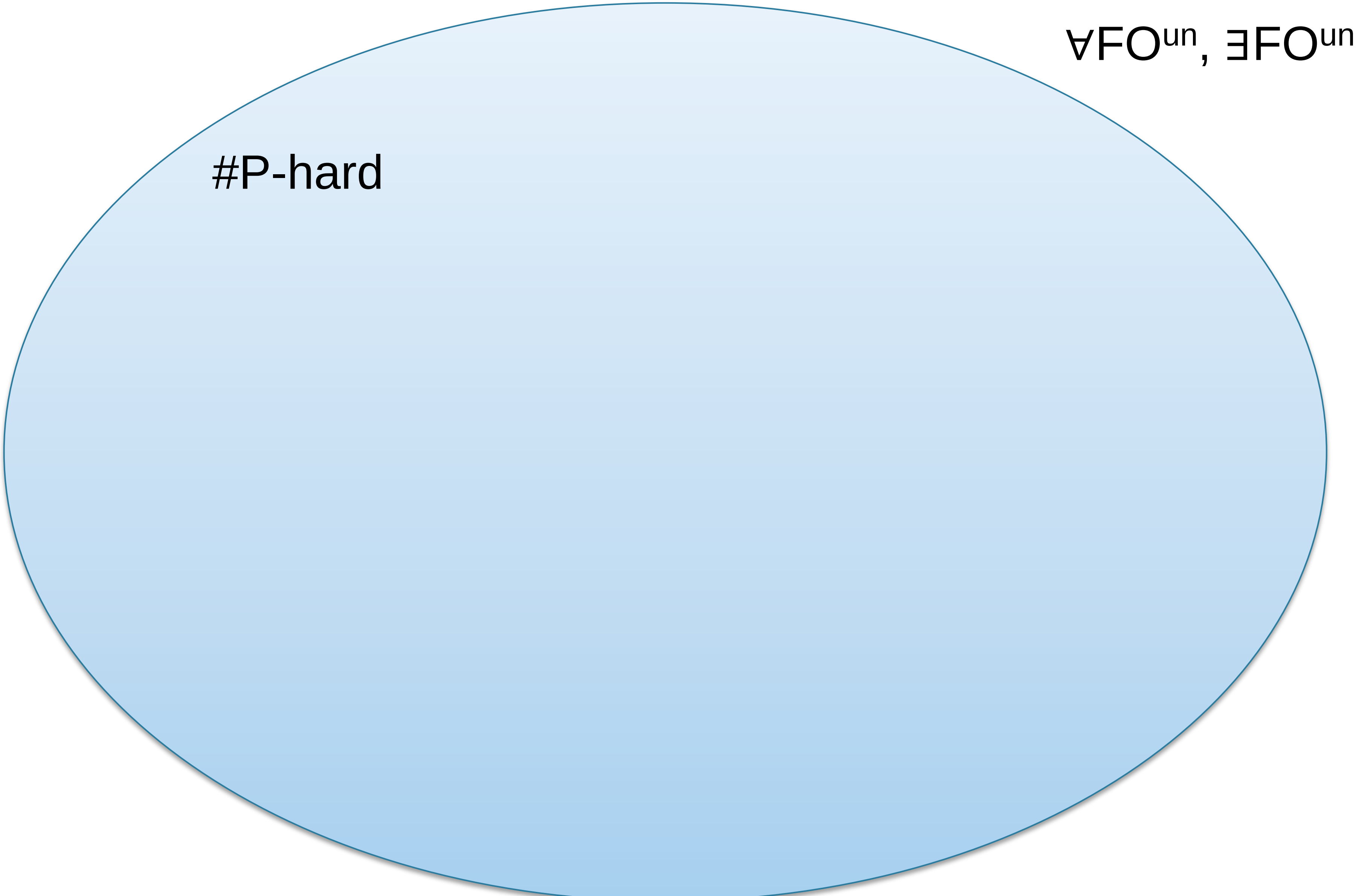
- Question 1: Are the lifted rules complete?
  - We know that they get stuck on some queries
  - Should we add more rules?

Complete for "unate ∀FO" and for "unate ∃FO"

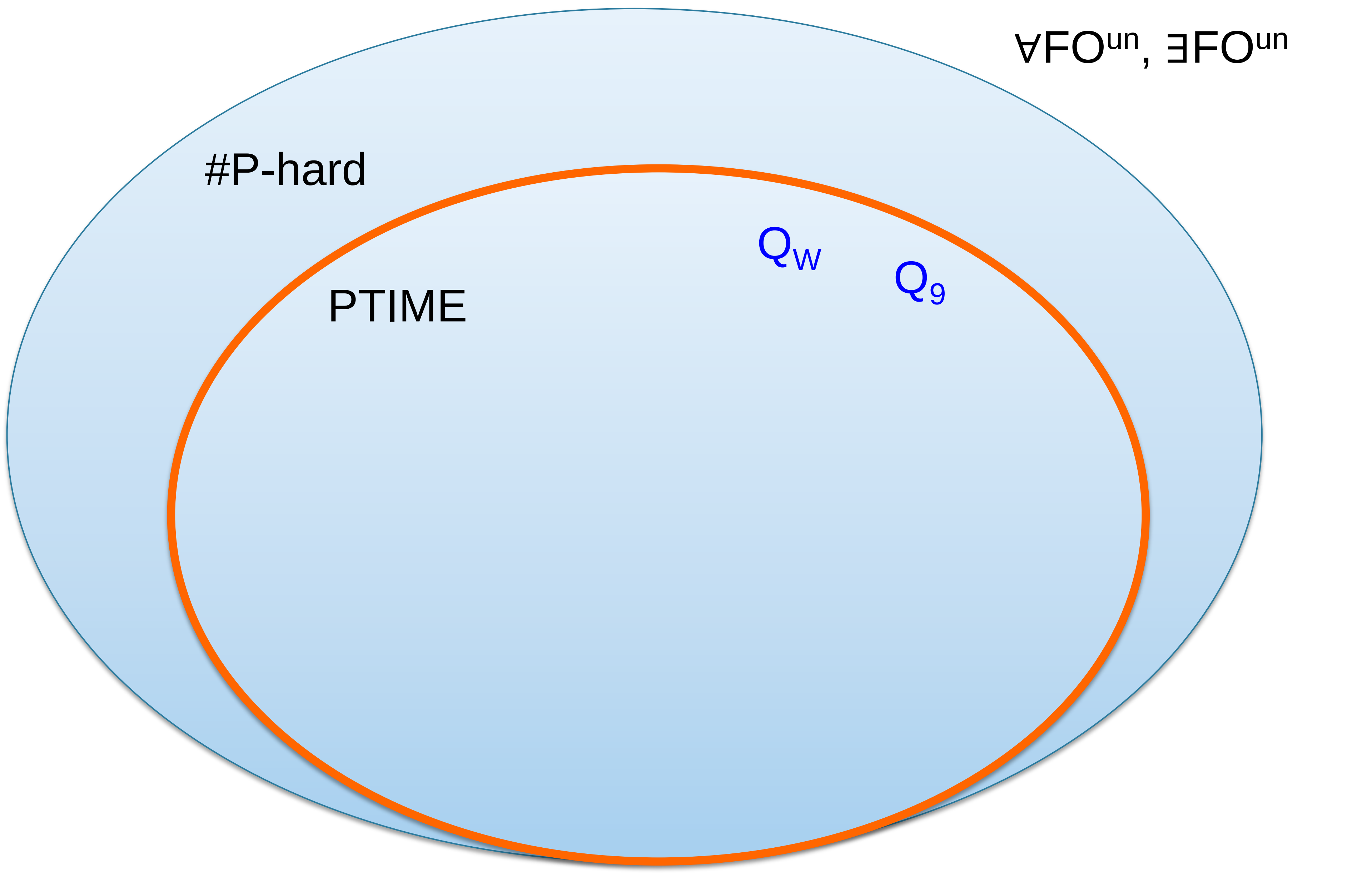
- Question 2: Are lifted rules stronger than grounded?
  - Lifted rules can also be grounded
  - Any advantage over grounded inference?

Strictly stronger than DPLL-based algorithms

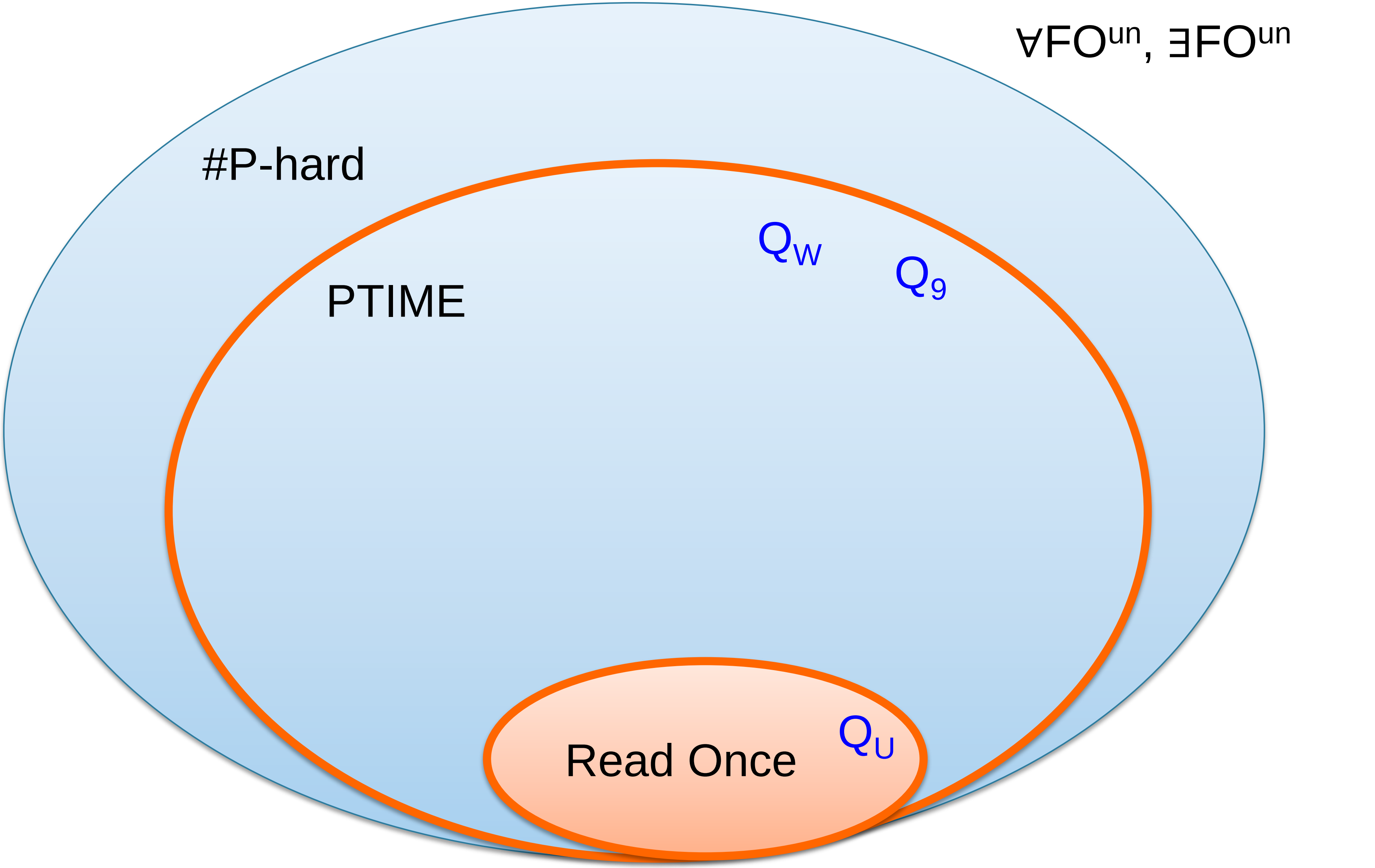
Möbius Über Alles



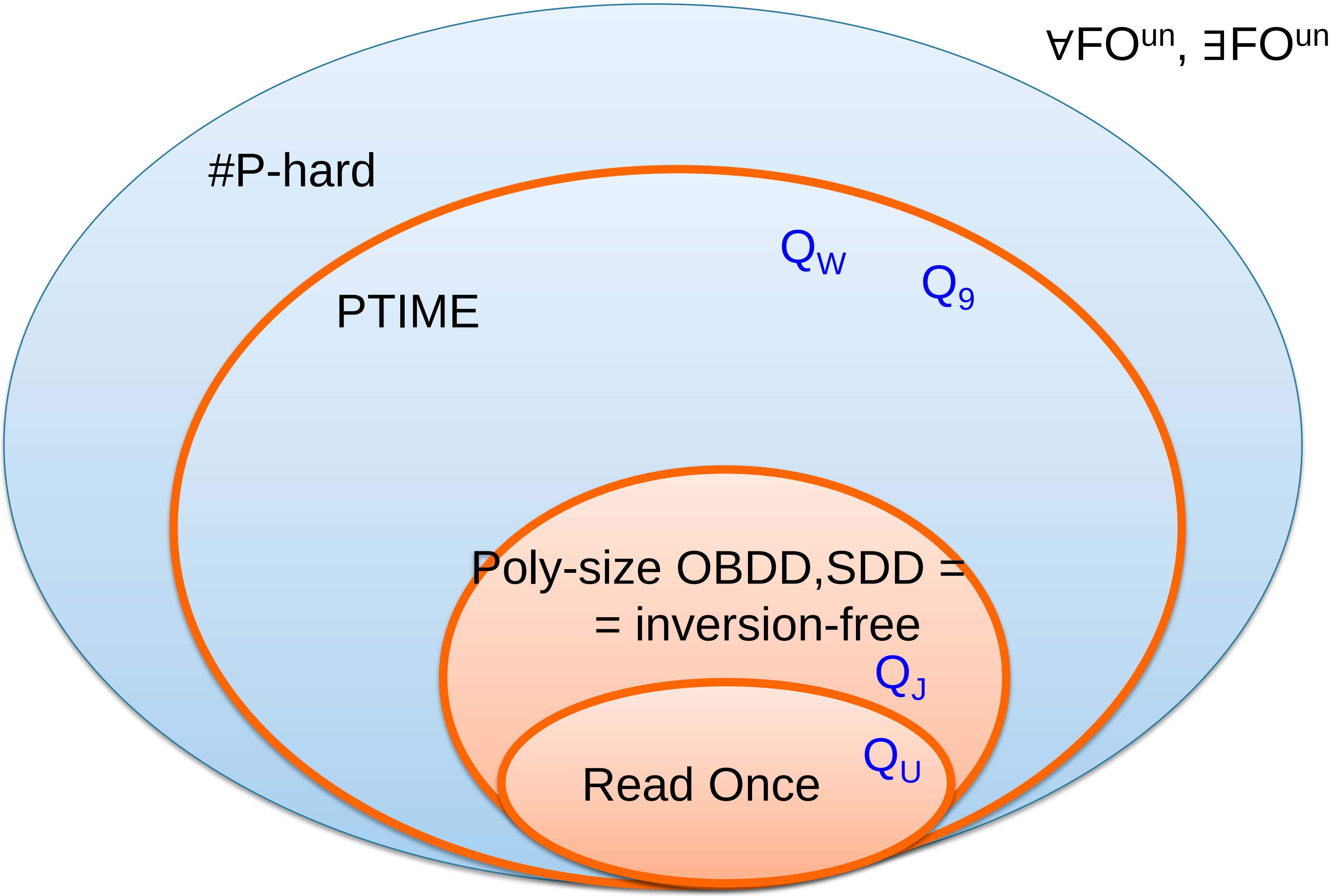
Möbius Über Alles



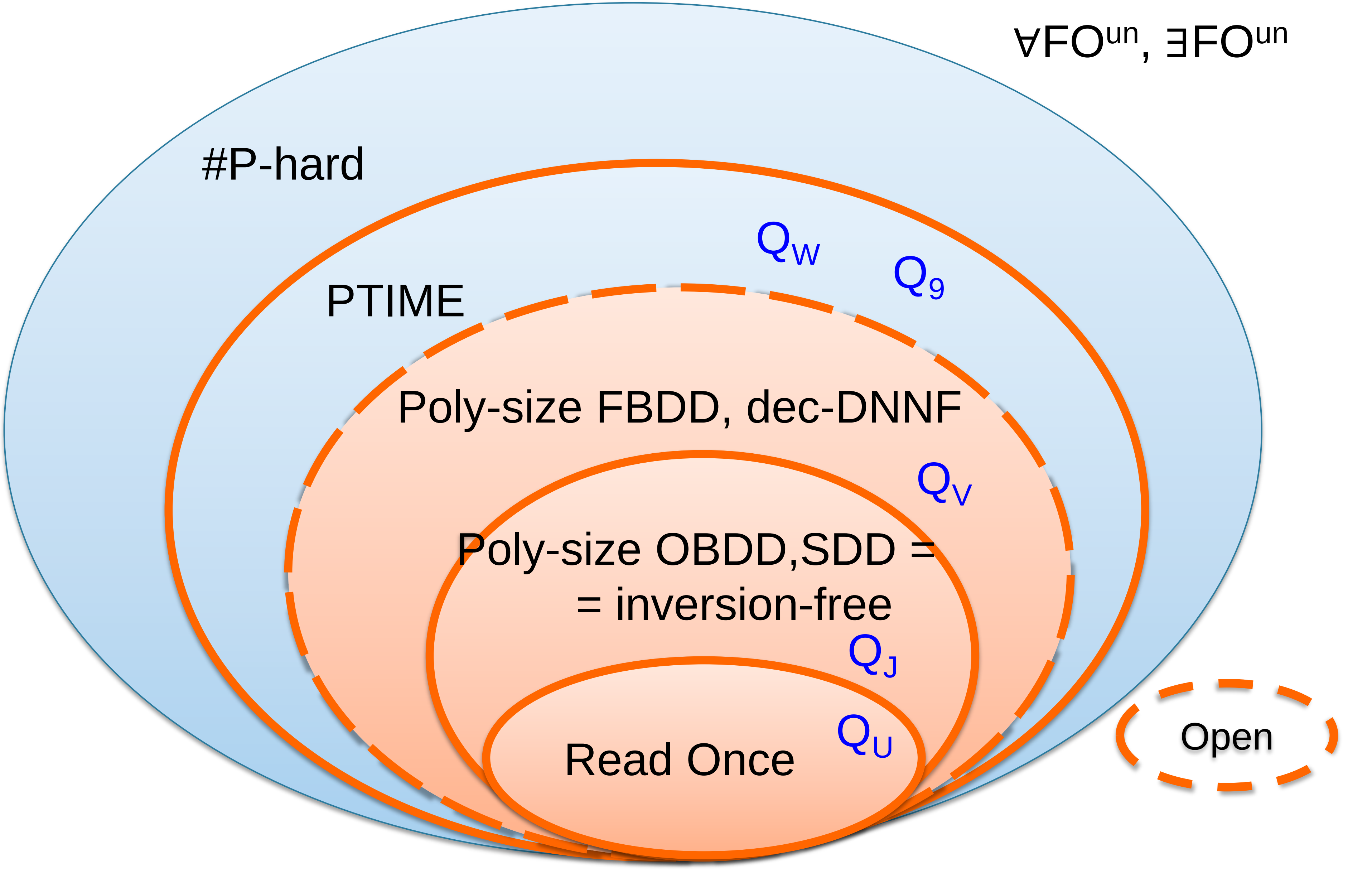
Möbius Über Alles



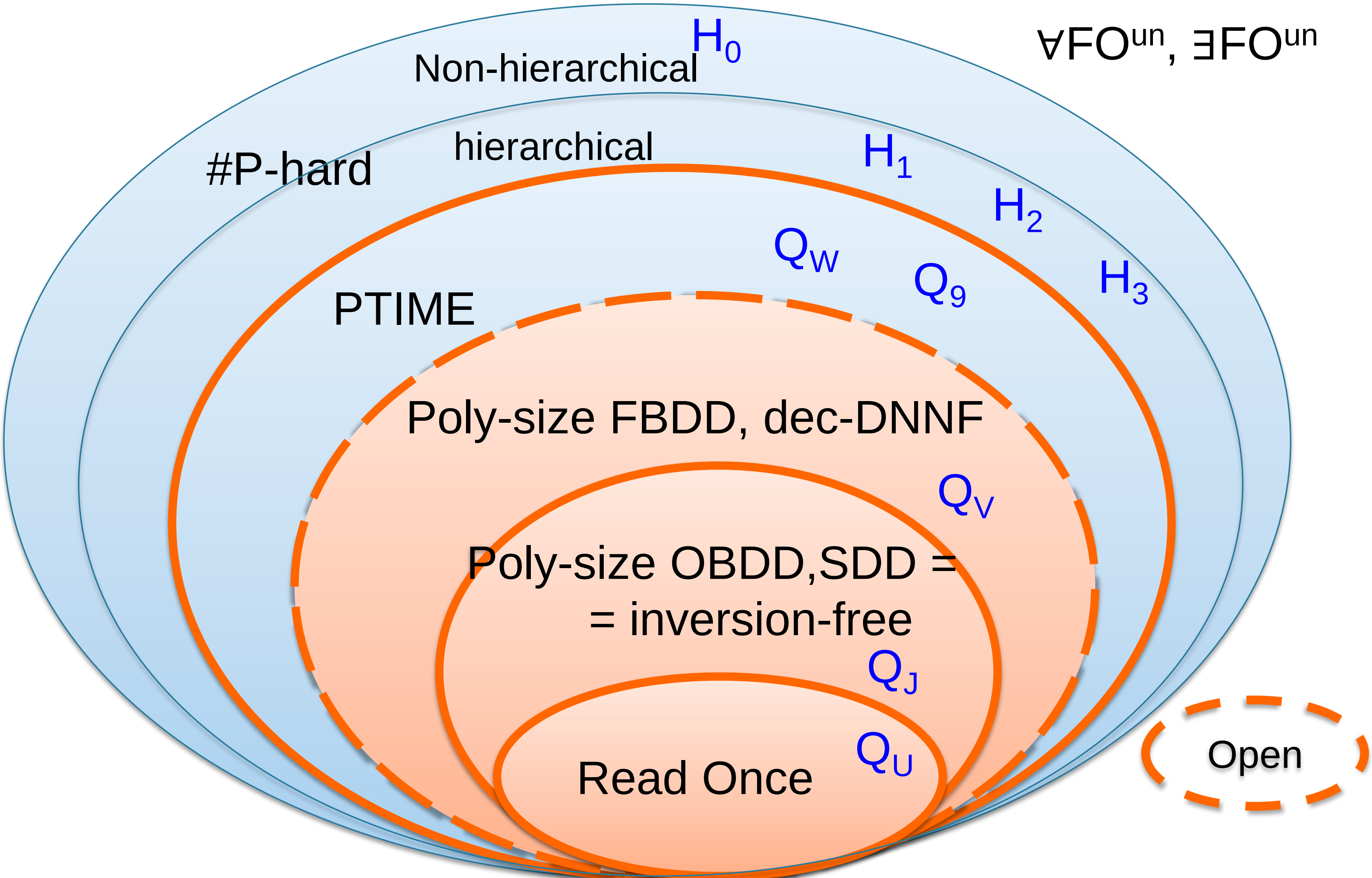
Möbius Über Alles



Möbius Über Alles



Möbius Über Alles



Outline

- Part 1: Motivation
- Part 2: Probabilistic Databases
- Part 3: Weighted Model Counting
- Part 4: Lifted Inference for WFOMC

- Part 5: Completeness of Lifted Inference
- Part 6: Query Compilation
- Part 7: Symmetric Lifted Inference Complexity
- Part 8: Open-World Probabilistic Databases
- Part 9: Discussion & Conclusions

Complexity over Symmetric DBs

Recall: in a symmetric DB all ground facts have the same probability

- We can apply new rules that exploit symmetries
- Dichotomy into PTIME / #P-hard no longer applies
- Lower bounds on query compilation no loner apply

Symmetric WFOMC

No database!

Def. A *weighted vocabulary* is (**R**, **w**), where

- **R** = (R_1, R_2, …, R_k) = relational vocabulary
- **w** = (w_1, w_2, …, w_k) = weights

Fix domain of size n;

- Implicit weights: $w(t) = w_i$, $\forall t \in [n]^{|arity(R_i)|}$

Complexity of symmetric WFOMC(Q,n): fixed Q, input n

Examples

$Q = \forall x \exists y\ R(x,y)$

Computable in PTIME in n

Examples

$Q = \forall x \exists y\, R(x,y)$

$\text{FOMC}(Q,n) = (2^n-1)^n$ $\text{WOMC}(Q,n) = ((1+w_R)^n-1)^n$

Computable in PTIME in n

Examples

Q = ∀x∃y R(x,y)

FOMC(Q,n) = $(2^n-1)^n$ WOMC(Q,n) = $((1+w_R)^n-1)^n$

Q = ∃x∃y [R(x) ∧ S(x,y) ∧T(y)]

$$\text{FOMC}(Q, n) = \sum_{i=0,n} \sum_{j=0,n} \binom{n}{i} \binom{n}{j} 2^{n^2 - ij} (2^{ij} - 1)$$

Computable in PTIME in n

Examples

$Q = \forall x \exists y\ R(x,y)$

$\text{FOMC}(Q,n) = (2^n-1)^n$ $\quad$ $\text{WOMC}(Q,n) = ((1+w_R)^n-1)^n$

$Q = \exists x \exists y\ [R(x) \wedge S(x,y) \wedge T(y)]$

$$\text{FOMC}(Q,n) = \sum_{i=0,n} \sum_{j=0,n} \binom{n}{i}\binom{n}{j} 2^{n^2-ij}\,(2^{ij}-1)$$

$$\text{WFOMC}(Q,n) =$$

$$\sum_{i=0,n} \sum_{j=0,n} \binom{n}{i}\binom{n}{j} {w_R}^i {w_T}^j (1+w_S)^{n-ij}\,((1+w_S)^{ij}-1)$$

Computable in PTIME in n

Hardness is Hard

Triangle = $\exists x \exists y \exists z\ [R(x,y) \land S(y,z) \land T(z,x)]$

Complexity of FOMC(Triangle, n) = open problem

Hardness is Hard

Triangle = $\exists x \exists y \exists z\, [R(x,y) \wedge S(y,z) \wedge T(z,x)]$

It is hard to prove that Triangle is hard!

- The input = just one number n, runtime = f(n)
- In unary: n = 111…11, runtime = f(size of input)
- FOMC(Q, n) in $\#P_1$
- Unlikely #P-hard [Valiant'79]

Complexity of FOMC(Triangle, n) = open problem

The Class $\#P_1$

- $\#P_1$ = functions in #P over a unary input alphabet
  Also called *tally problems*

- Valiant [1979]: *there exists* $\#P_1$ complete problems

- Bertoni, Goldwurm, Sabadini [1991]:
  there exists a CFG s.t. counting # strings of a given length is $\#P_1$ complete

- What about a natural problem?
  - Goldsmith: “no natural combinatorial problems known to be $\#P_1$ complete”

The Logic FO^k

FO^k = FO restricted to k variables

- Note: may reuse variables!
- "The graph has a path of length 10":

$$\exists x \exists y (R(x,y) \wedge \exists x (R(y,x) \wedge \exists y (R(x,y) \wedge \exists x (R(y,x) \ldots)))$$

What is known about FO^k

- Satisfiability is decidable for FO^2
- Satisfiability is undecidable for FO^k, $k \geq 3$

Results for Symmetric Inference

[Van den Broeck’14 ,Beame’15]

Results for Symmetric Inference

Theorem
There exists Q in FO^3 s.t. FOMC(Q, n) is $\#P_1$ hard
There exists CQ Q s.t. WFOMC(Q, n) is $\#P_1$ hard

[Van den Broeck'14 ,Beame'15]

Results for Symmetric Inference

Theorem
There exists Q in FO^3 s.t. FOMC(Q, n) is $\#P_1$ hard
There exists CQ Q s.t. WFOMC(Q, n) is $\#P_1$ hard

Theorem WFOMC(Q, n) is in PTIME

- For any Q in FO^2
- For any gamma-acyclic Q

[Van den Broeck'14 ,Beame'15]

Results for Symmetric Inference

Theorem
There exists Q in FO^3 s.t. FOMC(Q, n) is $\#P_1$ hard
There exists CQ Q s.t. WFOMC(Q, n) is $\#P_1$ hard

Theorem WFOMC(Q, n) is in PTIME
- For any Q in FO^2
- For any gamma-acyclic Q

Corresponding decision problem = the spectrum problem
Data complexity: { Spec(Q) | Q in FO} = NP_1 [Fagin'74]
Combined complexity: NP-complete for FO^2, PSPACE-complete for FO

[Van den Broeck'14 ,Beame'15]

(Non-)Application: 0/1 Laws

Def. $\mu_n(Q)$ = fraction of structures over a domain of size n that are models of Q

$$\mu_n(Q) = \text{FOMC}(Q, n) \ / \ \text{FOMC}(\text{TRUE}, n)$$

Theorem. [Fagin'76]
For all Q in FO (w/o constants) $\lim_{n \rightarrow \infty} \mu_n(Q) = 0$ or 1

Example: $Q = \forall x \exists y\, R(x,y)$;

$$\text{FOMC}(Q,n) = (2^n-1)^n$$

$$\mu_n(Q) = (2^n-1)^n / 2^{n\char94 2} \rightarrow 1$$

(Non-)Application: 0/1 Laws

How does one proof the 0/1 law?

- Attempt: find explicit formula $\mu_n(Q)$, compute limit.
- Fails! because $\mu_n(Q)$ is $\#P_1$-hard in general! Very unlikely to admit a simple closed form formula
- Fagin's proof: beautiful argument involving infinite models, the compactness theorem, and completeness of a theory with a categorical model

Discussion

Fagin 1974

THEOREM 6. *Assume that* $A \subseteq \mathrm{Fin}(S)$, *and that* A *is closed under isomorphism.*

1. *If* $S \neq \varnothing$, *then* A *is an* S*-spectrum iff* $E(A) \in NP$.
2. *If* $S = \varnothing$, *then* A *is a spectrum iff* $E(A) \in NP_1$.

Here: S is a vocabulary, S-spectrum of Q = set of structures that satisfy Q

$\#P_1$ corresponds to {FOMC(Q,n) | Q in FO }

Discussion

Fagin 1974

THEOREM 6. *Assume that* $\mathcal{A} \subseteq \text{Fin}(S)$, *and that* $\mathcal{A}$ *is closed under isomorphism,*

1. *If* $S \neq \emptyset$, *then* $\mathcal{A}$ *is an* S*-spectrum iff* $E(\mathcal{A}) \in NP$.
2. *If* $S = \emptyset$, *then* $\mathcal{A}$ *is a spectrum iff* $E(\mathcal{A}) \in NP_1$.

Here: S is a vocabulary, S-spectrum of Q = set of structures that satisfy Q

Restated:

1. NP = ∃SO — Fagin's classic result
2. NP_1 = ∃SO(empty-vocabulary) — less well known

$\#P_1$ corresponds to {FOMC(Q,n) | Q in FO }

Summary

Exploiting symmetries gives us more power:

- Some queries that are hard over asymmetric databases become easy over symmetric ones: e.g. FO^2 is in PTIME

Limitations:

- Proving hardness is very hard
- Real data is never completely symmetric

Outline

- Part 1: Motivation
- Part 2: Probabilistic Databases
- Part 3: Weighted Model Counting
- Part 4: Lifted Inference for WFOMC

- Part 5: Completeness of Lifted Inference
- Part 6: Query Compilation
- Part 7: Symmetric Lifted Inference Complexity
- Part 8: Open-World Probabilistic Databases
- Part 9: Discussion & Conclusions

What we'd like to do…

Has anyone published a paper with both Erdos and Einstein

All News Images Videos Shopping More ▾ Search tools

About 82,400 results (0.73 seconds)

Erdős number - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/**Erdős**_number ▾ Wikipedia ▾
He **published** more **papers** during his lifetime (at least 1,525) than any other ...
Anybody else's **Erdős** number is k + 1 where k is the lowest **Erdős** number of any coauthor. ... Albert **Einstein** and Sheldon Lee Glashow **have** an **Erdős** number of 2. ... and mathematician Ruth Williams, **both** of whom **have** an **Erdős** number of 2.

Erdős–Bacon number - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/**Erdős**–Bacon_number ▾ Wikipedia ▾
This article possibly **contains** previously unpublished synthesis of **published** ... Her **paper** gives her an **Erdős** number of 4, and a Bacon number of 2, **both** of ...

What we'd like to do…

∃x Coauthor(Einstein,x) ∧ Coauthor(Erdos,x)

All News Images Videos Shopping More Search tools

About 82,400 results (0.73 seconds)

Erdős number - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/**Erdős**_number Wikipedia
He **published** more **papers** during his lifetime (at least 1,525) than any other ... **Anybody** else's **Erdős** number is k + 1 where k is the lowest **Erdős** number of any coauthor. ... Albert **Einstein** and Sheldon Lee Glashow **have** an **Erdős** number of 2. ... and mathematician Ruth Williams, **both** of whom **have** an **Erdős** number of 2.

Erdős–Bacon number - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/**Erdős**–Bacon_number Wikipedia
This article possibly **contains** previously unpublished synthesis of **published** ... Her **paper** gives her an **Erdős** number of 4, and a Bacon number of 2, **both** of ...

What we’d like to do…

$\exists x$ Coauthor(Einstein,x) $\wedge$ Coauthor(Erdos,x)

★★★★★

Ernst Straus

★★☆☆☆

Kristian Kersting, …

★☆☆☆☆

Justin Bieber, …

Problem: Queries

- What if fact missing?
- Probability 0 for:

Coauthor

X	Y	P
Einstein	Straus	0.7
~~**Erdos**~~	~~**Straus**~~	~~**0.6**~~
Einstein	Pauli	0.9
Erdos	Renyi	0.7
Kersting	Natarajan	0.8
Luc	Paol	0.1
…	…	…

Q1 = ∃x Coauthor(Einstein,**x**) ∧ Coauthor(Erdos,**x**)

Problem: Queries

- What if fact missing?
- Probability 0 for:

Coauthor

X	Y	P
Einstein	Straus	0.7
~~**Erdos**~~	~~**Straus**~~	~~**0.6**~~
Einstein	Pauli	0.9
Erdos	Renyi	0.7
Kersting	Natarajan	0.8
Luc	Paol	0.1
…	…	…

Q1 = ∃x Coauthor(Einstein,**x**) ∧ Coauthor(Erdos,**x**)

Q2 = Coauthor(Einstein,**Straus**) ∧ Coauthor(Erdos,**Straus**)

Problem: Queries

- What if fact missing?
- Probability 0 for:

Coauthor

X	Y	P
Einstein	Straus	0.7
~~**Erdos**~~	~~**Straus**~~	~~**0.6**~~
Einstein	Pauli	0.9
Erdos	Renyi	0.7
Kersting	Natarajan	0.8
Luc	Paol	0.1
…	…	…

Q1 = ∃x Coauthor(Einstein,**x**) ∧ Coauthor(Erdos,**x**)

Q2 = Coauthor(Einstein,**Straus**) ∧ Coauthor(Erdos,**Straus**)

Q3 = Coauthor(Einstein,**Kersting**) ∧ Coauthor(Erdos,**Kersting**)

Problem: Queries

- What if fact missing?
- Probability 0 for:

Coauthor

X	Y	P
Einstein	Straus	0.7
~~**Erdos**~~	~~**Straus**~~	~~**0.6**~~
Einstein	Pauli	0.9
Erdos	Renyi	0.7
Kersting	Natarajan	0.8
Luc	Paol	0.1
…	…	…

Q1 = $\exists x$ Coauthor(Einstein,**x**) $\wedge$ Coauthor(Erdos,**x**)

Q2 = Coauthor(Einstein,**Straus**) $\wedge$ Coauthor(Erdos,**Straus**)

Q3 = Coauthor(Einstein,**Kersting**) $\wedge$ Coauthor(Erdos,**Kersting**)

Q4 = Coauthor(Einstein,**Bieber**) $\wedge$ Coauthor(Erdos,**Bieber**)

Problem: Queries

- What if fact missing?
- Probability 0 for:

Coauthor

X	Y	P
Einstein	Straus	0.7
~~**Erdos**~~	~~**Straus**~~	~~**0.6**~~
Einstein	Pauli	0.9
Erdos	Renyi	0.7
Kersting	Natarajan	0.8
Luc	Paol	0.1
…	…	…

Q1 = ∃x Coauthor(Einstein,**x**) ∧ Coauthor(Erdos,**x**)

Q2 = Coauthor(Einstein,**Straus**) ∧ Coauthor(Erdos,**Straus**)

Q3 = Coauthor(Einstein,**Kersting**) ∧ Coauthor(Erdos,**Kersting**)

Q4 = Coauthor(Einstein,**Bieber**) ∧ Coauthor(Erdos,**Bieber**)

Q5 = Coauthor(Einstein,**Bieber**) ∧ ¬Coauthor(**Einstein**,**Bieber**)

Intuition

X	Y	P
Einstein	Straus	0.7
~~**Erdos**~~	~~**Straus**~~	~~0.6~~
Einstein	Pauli	0.9
Erdos	Renyi	0.7
Kersting	Natarajan	0.8
Luc	Paol	0.1
…	…	…

Q1 = ∃x Coauthor(Einstein,**x**) ∧ Coauthor(Erdos,**x**)

Q2 = Coauthor(Einstein,**Straus**) ∧ Coauthor(Erdos,**Straus**)

Q3 = Coauthor(Einstein,**Kersting**) ∧ Coauthor(Erdos,**Kersting**)

Q4 = Coauthor(Einstein,**Bieber**) ∧ Coauthor(Erdos,**Bieber**)

Q5 = Coauthor(Einstein,**Bieber**) ∧ ¬Coauthor(**Einstein**,**Bieber**)

[Ceylan'16]

Intuition

X	Y	P
Einstein	Straus	0.7
~~**Erdos**~~	~~**Straus**~~	~~**0.6**~~
Einstein	Pauli	0.9
Erdos	Renyi	0.7
Kersting	Natarajan	0.8
Luc	Paol	0.1
…	…	…

Q1 = ∃x Coauthor(Einstein,**x**) ∧ Coauthor(Erdos,**x**)

Q2 = Coauthor(Einstein,**Straus**) ∧ Coauthor(Erdos,**Straus**)

Q3 = Coauthor(Einstein,**Kersting**) ∧ Coauthor(Erdos,**Kersting**)

Q4 = Coauthor(Einstein,**Bieber**) ∧ Coauthor(Erdos,**Bieber**)

Q5 = Coauthor(Einstein,**Bieber**) ∧ ¬Coauthor(**Einstein**,**Bieber**)

We know for sure that P(Q1) ≥ P(Q2), P(Q1) ≥ P(Q3), P(Q1) ≥ P(Q4)

[Ceylan'16]

Intuition

X	Y	P
Einstein	Straus	0.7
~~**Erdos**~~	~~**Straus**~~	~~**0.6**~~
Einstein	Pauli	0.9
Erdos	Renyi	0.7
Kersting	Natarajan	0.8
Luc	Paol	0.1
…	…	…

Q1 = ∃x Coauthor(Einstein,**x**) ∧ Coauthor(Erdos,**x**)

Q2 = Coauthor(Einstein,**Straus**) ∧ Coauthor(Erdos,**Straus**)

Q3 = Coauthor(Einstein,**Kersting**) ∧ Coauthor(Erdos,**Kersting**)

Q4 = Coauthor(Einstein,**Bieber**) ∧ Coauthor(Erdos,**Bieber**)

Q5 = Coauthor(Einstein,**Bieber**) ∧ ¬Coauthor(**Einstein**,**Bieber**)

We know for sure that P(Q1) ≥ P(Q2), P(Q1) ≥ P(Q3), P(Q1) ≥ P(Q4)

and P(Q2) ≥ P(Q5), P(Q3) ≥ P(Q5), P(Q4) ≥ P(Q5)

[Ceylan'16]

Intuition

X	Y	P
Einstein	Straus	0.7
~~**Erdos**~~	~~**Straus**~~	~~**0.6**~~
Einstein	Pauli	0.9
Erdos	Renyi	0.7
Kersting	Natarajan	0.8
Luc	Paol	0.1
…	…	…

Q1 = ∃x Coauthor(Einstein,**x**) ∧ Coauthor(Erdos,**x**)

Q2 = Coauthor(Einstein,**Straus**) ∧ Coauthor(Erdos,**Straus**)

Q3 = Coauthor(Einstein,**Kersting**) ∧ Coauthor(Erdos,**Kersting**)

Q4 = Coauthor(Einstein,**Bieber**) ∧ Coauthor(Erdos,**Bieber**)

Q5 = Coauthor(Einstein,**Bieber**) ∧ ¬Coauthor(**Einstein**,**Bieber**)

We know for sure that P(Q1) ≥ P(Q2), P(Q1) ≥ P(Q3), P(Q1) ≥ P(Q4)

and P(Q2) ≥ P(Q5), P(Q3) ≥ P(Q5), P(Q4) ≥ P(Q5)   and P(Q5) = 0.

[Ceylan'16]

Intuition

X	Y	P
Einstein	Straus	0.7
~~**Erdos**~~	~~**Straus**~~	~~**0.6**~~
Einstein	Pauli	0.9
Erdos	Renyi	0.7
Kersting	Natarajan	0.8
Luc	Paol	0.1
…	…	…

Q1 = ∃x Coauthor(Einstein,**x**) ∧ Coauthor(Erdos,**x**)

Q2 = Coauthor(Einstein,**Straus**) ∧ Coauthor(Erdos,**Straus**)

Q3 = Coauthor(Einstein,**Kersting**) ∧ Coauthor(Erdos,**Kersting**)

Q4 = Coauthor(Einstein,**Bieber**) ∧ Coauthor(Erdos,**Bieber**)

Q5 = Coauthor(Einstein,**Bieber**) ∧ ¬Coauthor(**Einstein**,**Bieber**)

We know for sure that P(Q1) ≥ P(Q2), P(Q1) ≥ P(Q3), P(Q1) ≥ P(Q4)

and P(Q2) ≥ P(Q5), P(Q3) ≥ P(Q5), P(Q4) ≥ P(Q5) and P(Q5) = 0.

We have strong evidence that P(Q2) ≥ P(Q3) ≥ P(Q4).

[Ceylan'16]

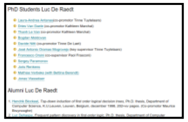
Problem: Broken Learning Loop

Bayesian view on learning:

- Prior belief:

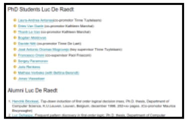
  Pr(HasStudent(Luc,Paol)) = 0.01

- Observe page

  Pr(HasStudent(Luc,Paol) | 

) = 0.2

- Observe page

  Pr(HasStudent(Luc,Paol) |  , 

) = 0.3

Principled and sound reasoning!

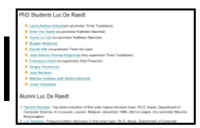
Problem: Broken Learning Loop

Current view on Knowledge Base Completion:

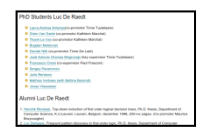
- Prior belief:

  Pr(HasStudent(Luc,Paol)) = 0

- Observe page

  Pr(HasStudent(Luc,Paol) | ) = 0.2

- Observe page

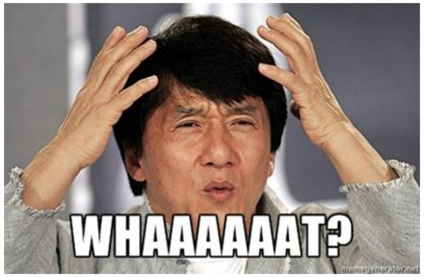
  Pr(HasStudent(Luc,Paol) |  , ) = 0.3

Problem: Broken Learning Loop

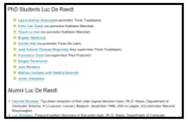
Current view on Knowledge Base Completion:

– Prior belief:

Pr(`HasStudent(Luc,Paol)`) = 0

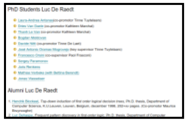


– Observe page

Pr(`HasStudent(Luc,Paol)` | 

) = 0.2

– Observe page

Pr(`HasStudent(Luc,Paol)` |  , 

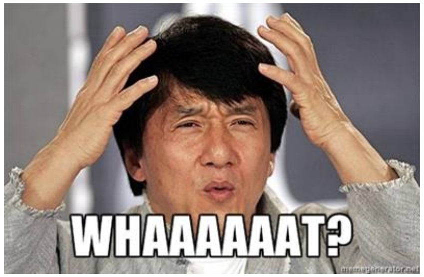
) = 0.3

Problem: Broken Learning Loop

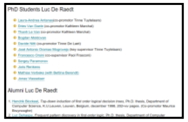
Current view on Knowledge Base Completion:

– Prior belief:

Pr(HasStudent(Luc,Paol)) = 0

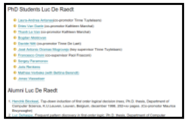


– Observe page

Pr(HasStudent(Luc,Paol) | 

) = 0.2

– Observe page

Pr(HasStudent(Luc,Paol) |  , 

) = 0.3

This is mathematical nonsense!

Knowledge Base Completion

Given:

LivesIn

X	Y
Luc	Belgium
Guy	USA
Kristian	Germany

LocatedIn

X	Y
Siemens	Germany
Siemens	Belgium
UCLA	USA
TUDortmund	Germany
KU Leuven	Belgium

WorksFor

X	Y
Luc	KU Leuven
Guy	UCLA
Kristian	TUDortmund
Ingo	Siemens

Learn:

0.8::LivesIn(x,y) :- WorksFor(x,z) ∧ LocatedIn(z,x).

How to measure success?

WorksFor

X	Y	P
Luc	KU Leuven	0.7
Guy	UCLA	0.6
Kristian	TUDortmund	0.3
Ingo	Siemens	0.3

LocatedIn

X	Y	P
Siemens	Germany	0.7
Siemens	Belgium	0.5
UCLA	USA	0.8
TUDortmund	Germany	0.6
KU Leuven	Belgium	0.7

0.8::LivesIn(x,y) :- WorksFor(x,z) ∧ LocatedIn(z,x).

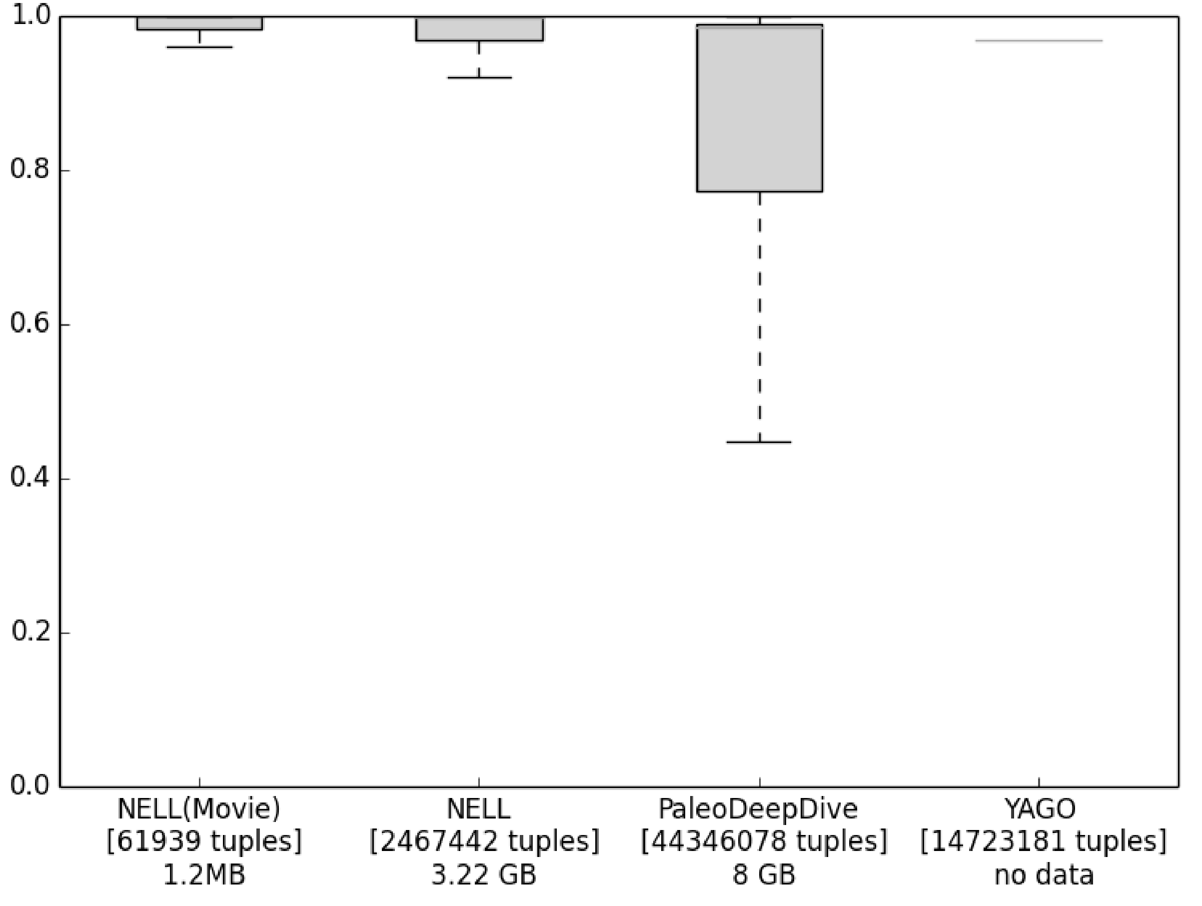
or

0.5::LivesIn(x,y) :- BornIn(x,y).

What is the likelihood, precision, accuracy, …?

Problem: Curse of Superlinearity

- Reality is worse!
- Tuples are intentionally missing!
- Every tuple has 99% pr.



Problem: Curse of Superlinearity

"This is all true, Guy,
but it's just a temporary issue"

"No it's not!"

Problem: Curse of Superlinearity

- A single table

Sibling

X	Y	P
...	...	...

- At the scale of facebook (billions of people)
- Real Bayesian belief about everyone

  I.e., all non-zero probabilities

$\Rightarrow$ 200 Exabytes of data

Problem: Curse of Superlinearity

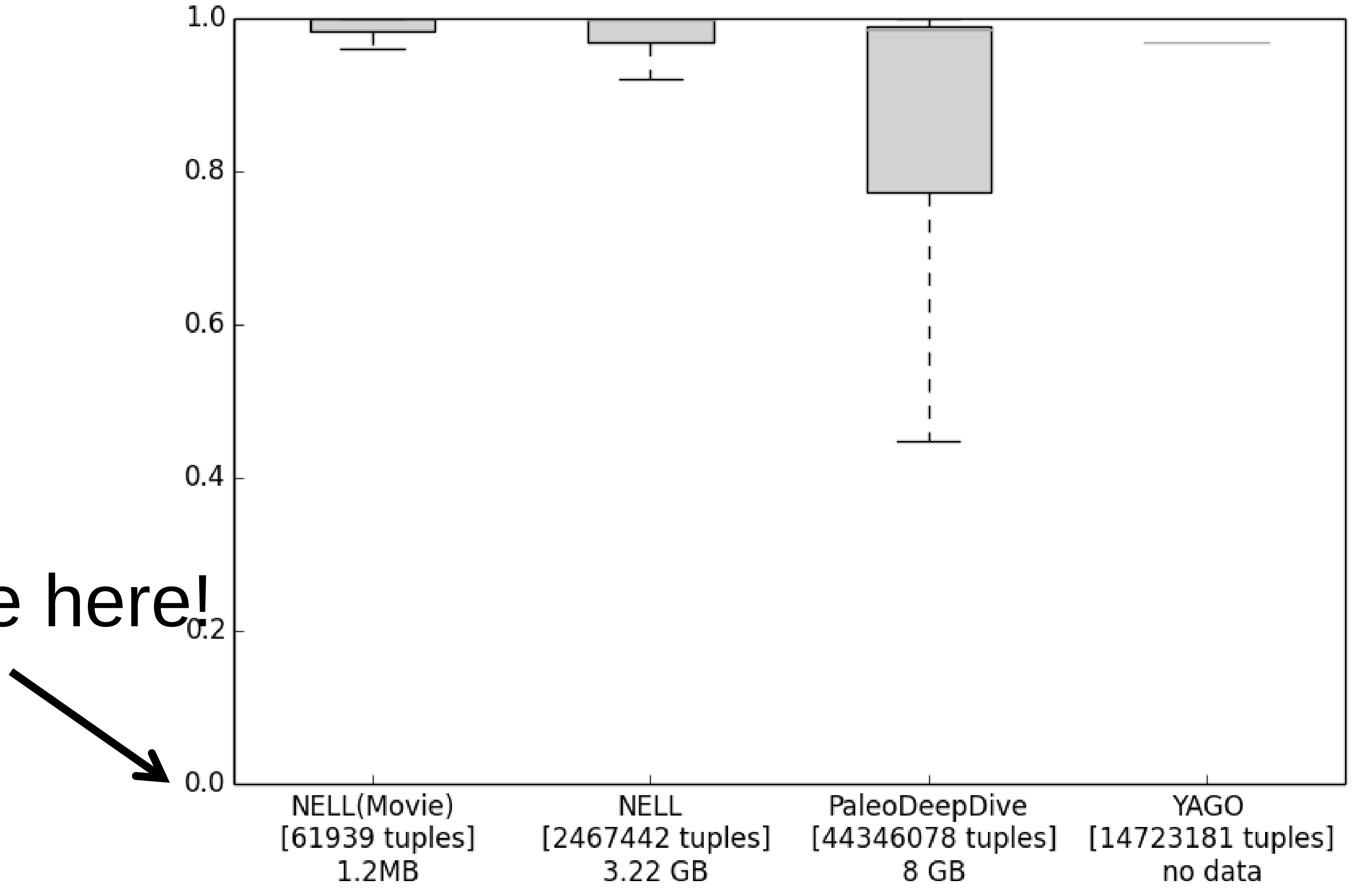
All Google storage is
a couple exabytes…

ing. In *Proc. of AAAI'15*. AAAI Press, 2015.

Randall Munroe. Google's datacenters on punch cards, 2015.

James D Park and Adnan Darwiche. Complexity Results and

Problem: Curse of Superlinearity



We should be here!

Closed-World Prob. Databases

A PDB $\mathcal{P}$ induces a *unique probability distribution* over worlds ω:

$$\mathrm{P}_{\mathcal{P}}(\omega) = \prod_{t \in \omega} \mathrm{P}_{\mathcal{P}}(t) \prod_{t \notin \omega} (1 - \mathrm{P}_{\mathcal{P}}(t)),$$

where for every tuple t, it holds that

$$\mathrm{P}_{\mathcal{P}}(t) = \begin{cases} p & \text{if } \langle t : p \rangle \in \mathcal{P} \\ 0 & \text{otherwise. [Probabilistic CWA]} \end{cases}$$

Open-World Prob. Databases

An *OpenPDB* is a pair $\mathcal{G} = (\mathcal{P}, \lambda)$, where $\mathcal{P}$ is a PDB

$$\mathrm{P}_{\mathcal{G}}(t) = \begin{cases} p & \text{if } \langle t : p \rangle \in \mathcal{P} \\ [0, \lambda] & \text{otherwise.} \end{cases}$$

A *λ-completion* of $\mathcal{G}$ contains a tuple $\langle t : p \rangle$ for some $p \in [0, \lambda]$ for every $t \notin \mathcal{P}$. $\mathcal{G}$ induces *a set of probability distributions* $\mathrm{K}_{\mathcal{G}}$:

$$\underline{\mathrm{P}}_{\mathcal{G}}(\mathrm{Q}) = \min_{\mathrm{P} \in \mathrm{K}_{\mathcal{G}}} \mathrm{P}(\mathrm{Q}) \quad \text{and} \quad \overline{\mathrm{P}}_{\mathcal{G}}(\mathrm{Q}) = \max_{\mathrm{P} \in \mathrm{K}_{\mathcal{G}}} \mathrm{P}(\mathrm{Q}).$$

[Ceylan'16]

Open-World Prob. Databases

Intuition: tuples can be added with prob < λ

Q2 = Coauthor(Einstein,**Straus**) ∧ Coauthor(Erdos,**Straus**)

Coauthor

X	Y	P
Einstein	Straus	0.7
Einstein	Pauli	0.9
Erdos	Renyi	0.7
Kersting	Natarajan	0.8
Luc	Paol	0.1
…	…	…

Open-World Prob. Databases

Intuition: tuples can be added with prob < λ

Q2 = Coauthor(Einstein,**Straus**) ∧ Coauthor(Erdos,**Straus**)

Coauthor

X	Y	P
Einstein	Straus	0.7
Einstein	Pauli	0.9
Erdos	Renyi	0.7
Kersting	Natarajan	0.8
Luc	Paol	0.1
…	…	…

Coauthor

X	Y	P
Einstein	Straus	0.7
Einstein	Pauli	0.9
Erdos	Renyi	0.7
Kersting	Natarajan	0.8
Luc	Paol	0.1
…	…	…
Erdos	**Straus**	**λ**

Open-World Prob. Databases

Intuition: tuples can be added with prob < λ

Q2 = Coauthor(Einstein,**Straus**) ∧ Coauthor(Erdos,**Straus**)

Coauthor

X	Y	P
Einstein	Straus	0.7
Einstein	Pauli	0.9
Erdos	Renyi	0.7
Kersting	Natarajan	0.8
Luc	Paol	0.1
…	…	…

$0.7 * \lambda \geq P(Q2) \geq 0$

Coauthor

X	Y	P
Einstein	Straus	0.7
Einstein	Pauli	0.9
Erdos	Renyi	0.7
Kersting	Natarajan	0.8
Luc	Paol	0.1
…	…	…
Erdos	**Straus**	**λ**

Monotone Queries

- E.g., Unions of Conjunctive Queries (UCQ)

- Lower bound = closed world probability
- Upper bound = probability after adding all tuples with probability λ

- Quadratic blow-up ☹
- Lifted inference to the rescue!

Independent Project

$Q = \forall x \forall y\, (\mathrm{Smoker}(x) \lor \mathrm{Friend}(x,y))$

[Ceylan'16]

Independent Project

$Q = \forall x \forall y\, (\text{Smoker}(x) \lor \text{Friend}(x,y))$

$P(Q) = \prod_{A \in \text{Domain}} P(\text{Smoker}(A) \lor \forall y\, \text{Friend}(A,y))$

∀-Rule

Check independence:

$\text{Smoker}(\text{Alice}) \lor \forall y\, \text{Friend}(\text{Alice},y)$

$\text{Smoker}(\text{Bob}) \lor \forall y\, \text{Friend}(\text{Bob},y)$

[Ceylan'16]

Independent Project

$Q = \forall x \forall y\, (Smoker(x) \lor Friend(x,y))$

∀-Rule

Check independence:

$Smoker(Alice) \lor \forall y\, Friend(Alice,y)$

$Smoker(Bob) \lor \forall y\, Friend(Bob,y)$

$$P(Q) = \Pi_{A \in Domain} P(Smoker(A) \lor \forall y\, Friend(A,y))$$

$$\begin{aligned} = \ & P(Smoker(A) \lor \forall y\, Friend(A,y)) \\ \times\ & P(Smoker(B) \lor \forall y\, Friend(B,y)) \\ \times\ & P(Smoker(C) \lor \forall y\, Friend(C,y)) \\ \times\ & P(Smoker(D) \lor \forall y\, Friend(D,y)) \\ \times\ & P(Smoker(E) \lor \forall y\, Friend(E,y)) \\ \times\ & P(Smoker(F) \lor \forall y\, Friend(F,y)) \\ & \ldots \end{aligned}$$

[Ceylan'16]

Independent Project

$Q = \forall x \forall y\, (\text{Smoker}(x) \lor \text{Friend}(x,y))$

$P(Q) = \prod_{A \in \text{Domain}} P(\text{Smoker}(A) \lor \forall y\, \text{Friend}(A,y))$

∀-Rule

Check independence:
Smoker(Alice) ∨ ∀y Friend(Alice,y)
Smoker(Bob) ∨ ∀y Friend(Bob,y)

$$\begin{aligned} = \quad & P(\text{Smoker}(A) \lor \forall y\, \text{Friend}(A,y)) \\ \times\ & P(\text{Smoker}(B) \lor \forall y\, \text{Friend}(B,y)) \\ \times\ & P(\text{Smoker}(C) \lor \forall y\, \text{Friend}(C,y)) \\ \times\ & P(\text{Smoker}(D) \lor \forall y\, \text{Friend}(D,y)) \\ \times\ & P(\text{Smoker}(E) \lor \forall y\, \text{Friend}(E,y)) \\ \times\ & P(\text{Smoker}(F) \lor \forall y\, \text{Friend}(F,y)) \\ & \ldots \end{aligned}$$

Complexity PTIME?

[Ceylan'16]

Closed-World Independent Project

$Q = \forall x \forall y\, (Smoker(x) \vee Friend(x,y))$

∀-Rule

Check independence:
Smoker(Alice) ⋁ ∀y Friend(Alice,y)
Smoker(Bob) ⋁ ∀y Friend(Bob,y)

$$P(Q) = \prod_{A \in Domain} P(Smoker(A) \vee \forall y\, Friend(A,y))$$

$$\begin{aligned} = \; & P(Smoker(A) \vee \forall y\, Friend(A,y)) \\ \times\; & P(Smoker(B) \vee \forall y\, Friend(B,y)) \\ \times\; & P(Smoker(C) \vee \forall y\, Friend(C,y)) \\ \times\; & P(Smoker(D) \vee \forall y\, Friend(D,y)) \\ \times\; & P(Smoker(E) \vee \forall y\, Friend(E,y)) \\ \times\; & P(Smoker(F) \vee \forall y\, Friend(F,y)) \\ & \dots \end{aligned}$$

Closed-World Independent Project

$Q = \forall x \forall y\,(\text{Smoker}(x) \lor \text{Friend}(x,y))$

∀-Rule

Check independence:
Smoker(Alice) ⋁ ∀y Friend(Alice,y)
Smoker(Bob) ⋁ ∀y Friend(Bob,y)

$P(Q) = \prod_{A \in \text{Domain}} P(\text{Smoker}(A) \lor \forall y\,\text{Friend}(A,y))$

$$
\begin{aligned}
= \quad & P(\text{Smoker}(A) \lor \forall y\,\text{Friend}(A,y)) \\
\times\ & P(\text{Smoker}(B) \lor \forall y\,\text{Friend}(B,y)) \\
\times\ & P(\text{Smoker}(C) \lor \forall y\,\text{Friend}(C,y)) \\
\times\ & P(\text{Smoker}(D) \lor \forall y\,\text{Friend}(D,y)) \\
\times\ & P(\text{Smoker}(E) \lor \forall y\,\text{Friend}(E,y)) \\
\times\ & P(\text{Smoker}(F) \lor \forall y\,\text{Friend}(F,y)) \\
& \ldots
\end{aligned}
$$

⇨ No supporting facts in database!

Closed-World Independent Project

$Q = \forall x \forall y\, (\text{Smoker}(x) \lor \text{Friend}(x,y))$

∀-Rule

Check independence:

$\text{Smoker}(\text{Alice}) \lor \forall y\, \text{Friend}(\text{Alice},y)$

$\text{Smoker}(\text{Bob}) \lor \forall y\, \text{Friend}(\text{Bob},y)$

$$P(Q) = \prod_{A \in \text{Domain}} P(\text{Smoker}(A) \lor \forall y\, \text{Friend}(A,y))$$

$$\begin{aligned} = \; & P(\text{Smoker}(A) \lor \forall y\, \text{Friend}(A,y)) \\ \times\; & P(\text{Smoker}(B) \lor \forall y\, \text{Friend}(B,y)) \\ \times\; & P(\text{Smoker}(C) \lor \forall y\, \text{Friend}(C,y)) \\ \times\; & P(\text{Smoker}(D) \lor \forall y\, \text{Friend}(D,y)) \\ \times\; & P(\text{Smoker}(E) \lor \forall y\, \text{Friend}(E,y)) \\ \times\; & P(\text{Smoker}(F) \lor \forall y\, \text{Friend}(F,y)) \\ & \ldots \end{aligned}$$

⇨ No supporting facts in database!

⇨ Probability 0 in closed world

Closed-World Independent Project

$Q = \forall x \forall y\, (\text{Smoker}(x) \lor \text{Friend}(x,y))$

∀-Rule

Check independence:
Smoker(Alice) ∨ ∀y Friend(Alice,y)
Smoker(Bob) ∨ ∀y Friend(Bob,y)

$$P(Q) = \prod_{A \in \text{Domain}} P(\text{Smoker}(A) \lor \forall y\, \text{Friend}(A,y))$$

$$\begin{aligned} = \; & P(\text{Smoker}(A) \lor \forall y\, \text{Friend}(A,y)) \\ \times\; & P(\text{Smoker}(B) \lor \forall y\, \text{Friend}(B,y)) \\ \times\; & P(\text{Smoker}(C) \lor \forall y\, \text{Friend}(C,y)) \\ \times\; & P(\text{Smoker}(D) \lor \forall y\, \text{Friend}(D,y)) \\ \times\; & P(\text{Smoker}(E) \lor \forall y\, \text{Friend}(E,y)) \\ \times\; & P(\text{Smoker}(F) \lor \forall y\, \text{Friend}(F,y)) \\ & \ldots \end{aligned}$$

⇒ No supporting facts in database!

⇒ Probability 0 in closed world

⇒ Ignore these queries!

Closed-World Independent Project

$Q = \forall x \forall y\, (\text{Smoker}(x) \vee \text{Friend}(x,y))$

∀-Rule

Check independence:
Smoker(Alice) ∨ ∀y Friend(Alice,y)
Smoker(Bob) ∨ ∀y Friend(Bob,y)

$$P(Q) = \prod_{A \in \text{Domain}} P(\text{Smoker}(A) \vee \forall y\, \text{Friend}(A,y))$$

$$
\begin{aligned}
= \; & P(\text{Smoker}(A) \vee \forall y\, \text{Friend}(A,y)) \\
\times\; & P(\text{Smoker}(B) \vee \forall y\, \text{Friend}(B,y)) \\
\times\; & P(\text{Smoker}(C) \vee \forall y\, \text{Friend}(C,y)) \\
\times\; & P(\text{Smoker}(D) \vee \forall y\, \text{Friend}(D,y)) \\
\times\; & P(\text{Smoker}(E) \vee \forall y\, \text{Friend}(E,y)) \\
\times\; & P(\text{Smoker}(F) \vee \forall y\, \text{Friend}(F,y)) \\
& \ldots
\end{aligned}
$$

⇒ No supporting facts in database!

⇒ Probability 0 in closed world

⇒ Ignore these queries!

Complexity linear time!

Open-World Independent Project

$Q = \forall x \forall y\, (\text{Smoker}(x) \vee \text{Friend}(x,y))$

∀-Rule

$P(Q) = \prod_{A \in \text{Domain}} P(\text{Smoker}(A) \vee \forall y\, \text{Friend}(A,y))$

Check independence:

$\text{Smoker}(\text{Alice}) \vee \forall y\, \text{Friend}(\text{Alice},y)$

$\text{Smoker}(\text{Bob}) \vee \forall y\, \text{Friend}(\text{Bob},y)$

$$
\begin{aligned}
= \quad & P(\text{Smoker}(A) \vee \forall y\, \text{Friend}(A,y)) \\
\times\ & P(\text{Smoker}(B) \vee \forall y\, \text{Friend}(B,y)) \\
\times\ & P(\text{Smoker}(C) \vee \forall y\, \text{Friend}(C,y)) \\
\times\ & P(\text{Smoker}(D) \vee \forall y\, \text{Friend}(D,y)) \\
\times\ & P(\text{Smoker}(E) \vee \forall y\, \text{Friend}(E,y)) \\
\times\ & P(\text{Smoker}(F) \vee \forall y\, \text{Friend}(F,y)) \\
& \ldots
\end{aligned}
$$

⇒ No supporting facts in database!

Open-World Independent Project

$Q = \forall x \forall y\, (\mathrm{Smoker}(x) \lor \mathrm{Friend}(x,y))$

$P(Q) = \prod_{A \in \mathrm{Domain}} P(\mathrm{Smoker}(A) \lor \forall y\, \mathrm{Friend}(A,y))$

∀-Rule

Check independence:
Smoker(Alice) ∨ ∀y Friend(Alice,y)
Smoker(Bob) ∨ ∀y Friend(Bob,y)

$= \quad P(\mathrm{Smoker}(A) \lor \forall y\, \mathrm{Friend}(A,y))$
$\times\, P(\mathrm{Smoker}(B) \lor \forall y\, \mathrm{Friend}(B,y))$
$\times\, P(\mathrm{Smoker}(C) \lor \forall y\, \mathrm{Friend}(C,y))$
$\times\, P(\mathrm{Smoker}(D) \lor \forall y\, \mathrm{Friend}(D,y))$
$\times\, P(\mathrm{Smoker}(E) \lor \forall y\, \mathrm{Friend}(E,y))$
$\times\, P(\mathrm{Smoker}(F) \lor \forall y\, \mathrm{Friend}(F,y))$
…

⇒ No supporting facts in database!

⇒ Probability p in closed world

Open-World Independent Project

$Q = \forall x \forall y\, (Smoker(x) \lor Friend(x,y))$

∀-Rule

Check independence:
Smoker(Alice) ⋁ ∀y Friend(Alice,y)
Smoker(Bob) ⋁ ∀y Friend(Bob,y)

$$P(Q) = \prod_{A \in Domain} P(Smoker(A) \lor \forall y\, Friend(A,y))$$

$$\begin{aligned} = \quad & P(Smoker(A) \lor \forall y\, Friend(A,y)) \\ \times\, & P(Smoker(B) \lor \forall y\, Friend(B,y)) \\ \times\, & P(Smoker(C) \lor \forall y\, Friend(C,y)) \\ \times\, & P(Smoker(D) \lor \forall y\, Friend(D,y)) \\ \times\, & P(Smoker(E) \lor \forall y\, Friend(E,y)) \\ \times\, & P(Smoker(F) \lor \forall y\, Friend(F,y)) \\ & \ldots \end{aligned}$$

⇒ No supporting facts in database!

⇒ Probability p in closed world

Complexity PTIME!

Open-World Independent Project

$Q = \forall x \forall y\, (\text{Smoker}(x) \lor \text{Friend}(x,y))$

∀-Rule

Check independence:

Smoker(Alice) ∨ ∀y Friend(Alice,y)

Smoker(Bob) ∨ ∀y Friend(Bob,y)

$$P(Q) = \prod_{A \in \text{Domain}} P(\text{Smoker}(A) \lor \forall y\, \text{Friend}(A,y))$$

$$\begin{aligned} = \quad & P(\text{Smoker}(A) \lor \forall y\, \text{Friend}(A,y)) \\ \times\, & P(\text{Smoker}(B) \lor \forall y\, \text{Friend}(B,y)) \\ \times\, & P(\text{Smoker}(C) \lor \forall y\, \text{Friend}(C,y)) \\ \times\, & P(\text{Smoker}(D) \lor \forall y\, \text{Friend}(D,y)) \\ \times\, & P(\text{Smoker}(E) \lor \forall y\, \text{Friend}(E,y)) \\ \times\, & P(\text{Smoker}(F) \lor \forall y\, \text{Friend}(F,y)) \\ & \ldots \end{aligned}$$

⇒ No supporting facts in database!

⇒ Probability p in closed world

Open-World Independent Project

$Q = \forall x \forall y\,(\text{Smoker}(x) \lor \text{Friend}(x,y))$

$P(Q) = \prod_{A \in \text{Domain}} P(\text{Smoker}(A) \lor \forall y\, \text{Friend}(A,y))$

∀-Rule

Check independence:
Smoker(Alice) ∨ ∀y Friend(Alice,y)
Smoker(Bob) ∨ ∀y Friend(Bob,y)

$= \quad P(\text{Smoker}(A) \lor \forall y\, \text{Friend}(A,y))$
$\times P(\text{Smoker}(B) \lor \forall y\, \text{Friend}(B,y))$
$\times P(\text{Smoker}(C) \lor \forall y\, \text{Friend}(C,y))$
$\times P(\text{Smoker}(D) \lor \forall y\, \text{Friend}(D,y))$
$\times P(\text{Smoker}(E) \lor \forall y\, \text{Friend}(E,y))$
$\times P(\text{Smoker}(F) \lor \forall y\, \text{Friend}(F,y))$
…

⇒ No supporting facts in database!

⇒ Probability p in closed world

⇒ All together, probability p^k
Do symmetric lifted inference

Open-World Independent Project

$Q = \forall x \forall y\, (\mathrm{Smoker}(x) \vee \mathrm{Friend}(x,y))$

∀-Rule

Check independence:
Smoker(Alice) ⋁ ∀y Friend(Alice,y)
Smoker(Bob) ⋁ ∀y Friend(Bob,y)

$$P(Q) = \prod_{A \in \mathrm{Domain}} P(\mathrm{Smoker}(A) \vee \forall y\, \mathrm{Friend}(A,y))$$

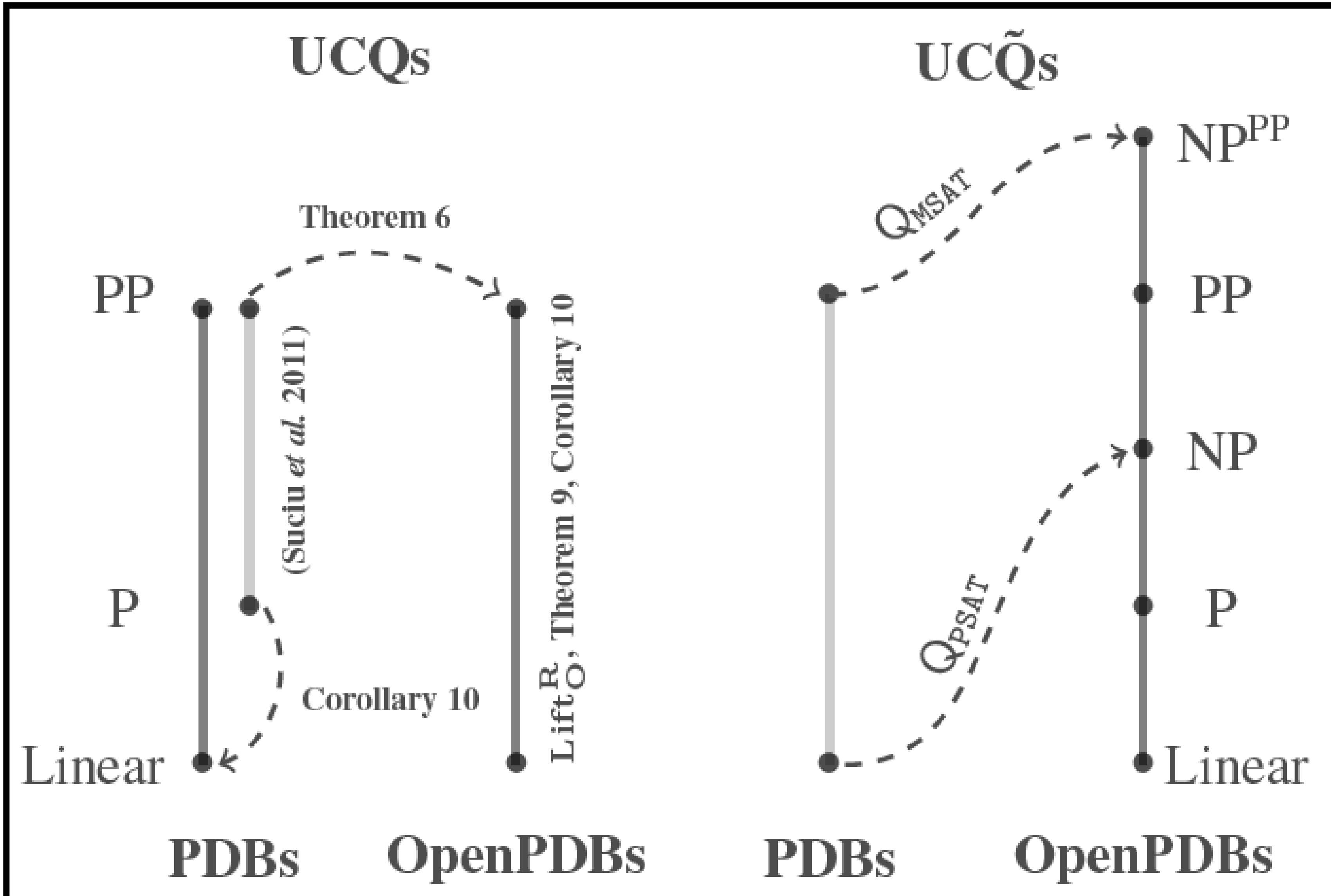
$$= P(\mathrm{Smoker}(A) \vee \forall y\, \mathrm{Friend}(A,y))$$
$$\times P(\mathrm{Smoker}(B) \vee \forall y\, \mathrm{Friend}(B,y))$$
$$\times P(\mathrm{Smoker}(C) \vee \forall y\, \mathrm{Friend}(C,y))$$
$$\times P(\mathrm{Smoker}(D) \vee \forall y\, \mathrm{Friend}(D,y))$$
$$\times P(\mathrm{Smoker}(E) \vee \forall y\, \mathrm{Friend}(E,y))$$
$$\times P(\mathrm{Smoker}(F) \vee \forall y\, \mathrm{Friend}(F,y))$$
$$\ldots$$

⇒ No supporting facts in database!

⇒ Probability p in closed world

⇒ All together, probability p^k
Do symmetric lifted inference

Complexity linear time!



Linear ⊆ P ⊆ NP ⊆ PP ⊆ P^{PP} ⊆ NP^{PP} ⊆ PSpace ⊆ ExpTime

[Ceylan'16]

Summary

- Open-world semantics make sense
- Matches how systems are employed
- Open-world reasoning is FREE for UCQs
- Beyond UCQs, can pay a hefty price
- Future work:
  More refined models of the open world
  E.g., (types, MLNs, additional statistics)

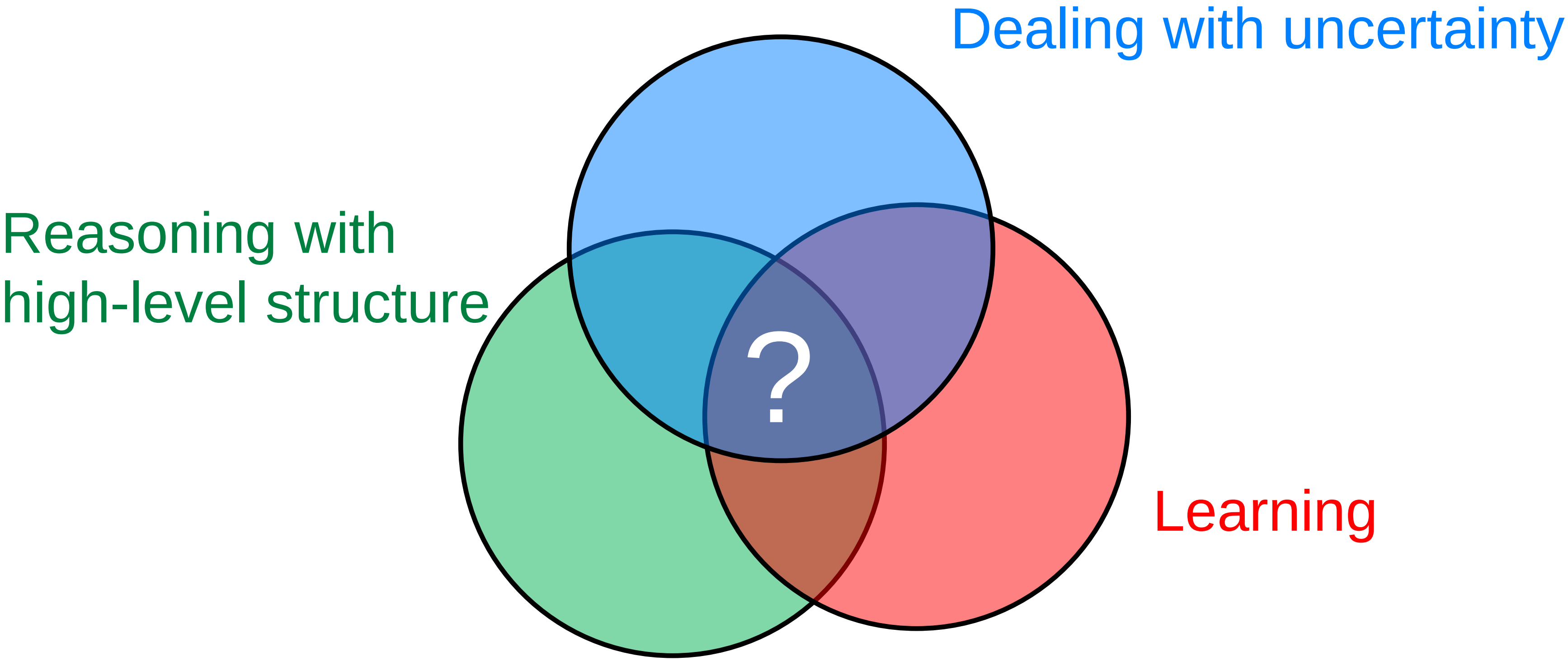
Outline

- Part 1: Motivation
- Part 2: Probabilistic Databases
- Part 3: Weighted Model Counting
- Part 4: Lifted Inference for WFOMC
- Part 5: Completeness of Lifted Inference
- Part 6: Query Compilation
- Part 7: Symmetric Lifted Inference Complexity
- Part 8: Open-World Probabilistic Databases
- Part 9: Discussion & Conclusions

Summary

- Relational models = the vast majority of data today, plus probabilistic Databases
- Weighted Model Counting = Uniform approach to Probabilistic Inference
- Lifted Inference = really simple rules
- The Power of Lifted Inference = we can prove that lifted inference is better

Challenges for the Future

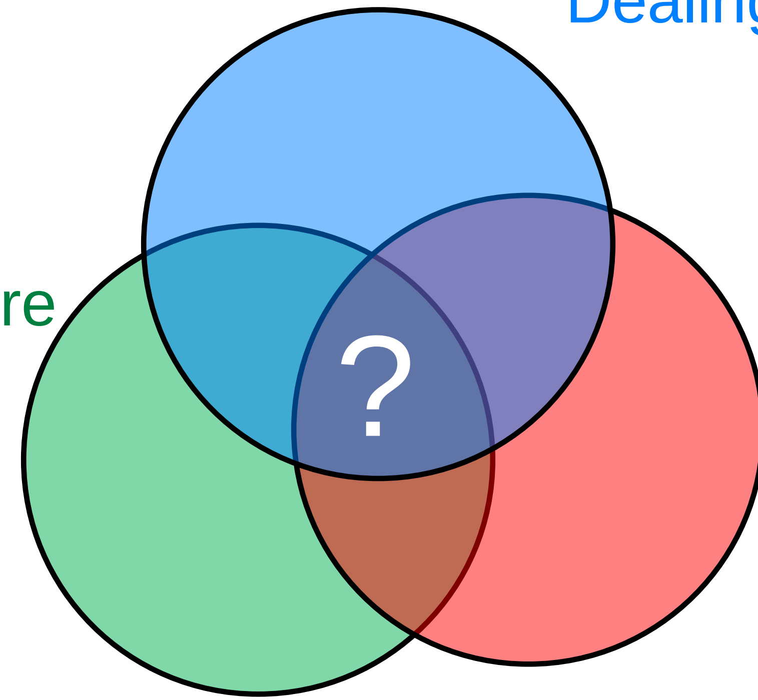


Challenges for the Future

Dealing with uncertainty

Reasoning with
high-level structure

- logic
- databases
- programming
- ...



Learning

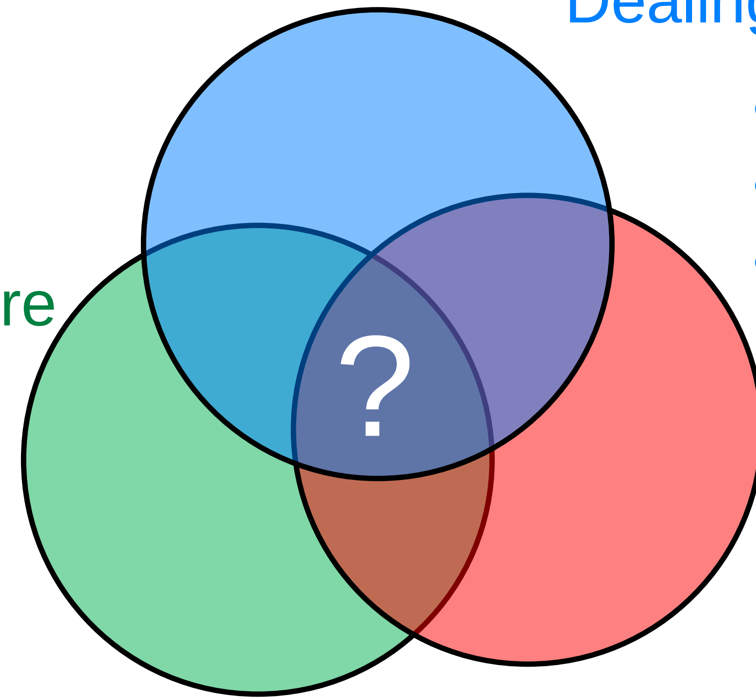
Challenges for the Future

Dealing with uncertainty

- probability theory
- graphical models
- ...

Reasoning with
high-level structure

- logic
- databases
- programming
- ...



Learning

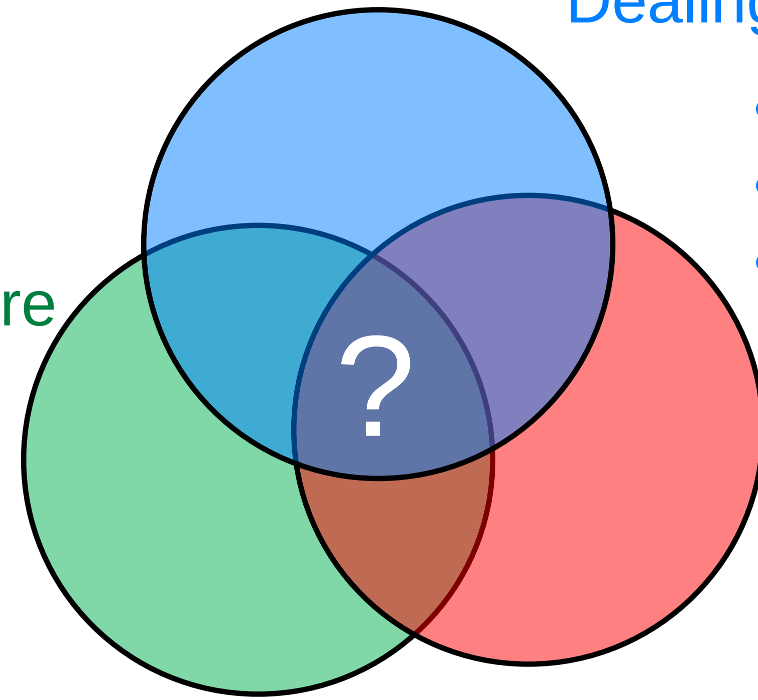
Challenges for the Future

Dealing with uncertainty

- probability theory
- graphical models
- ...

Reasoning with
high-level structure

- logic
- databases
- programming
- ...



Learning

- parameters
- structure

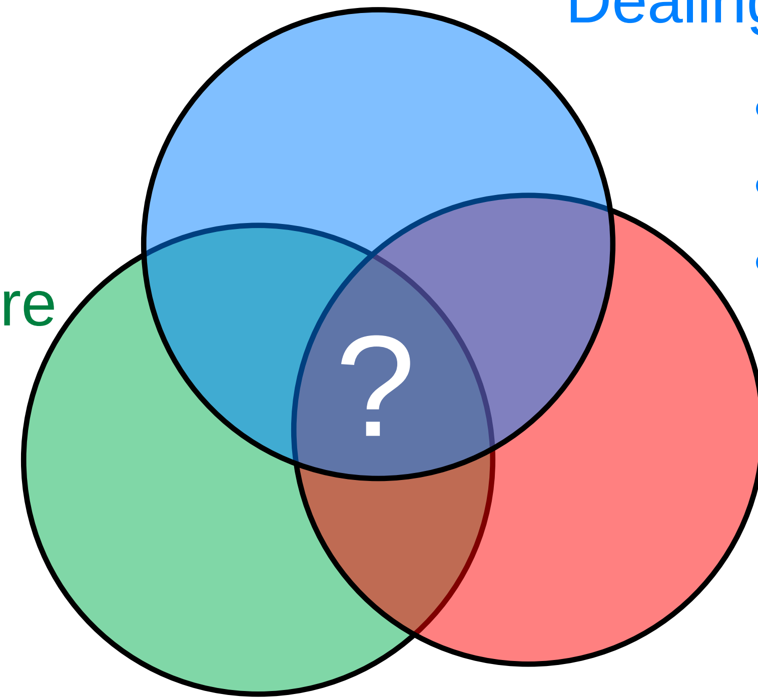
Challenges for the Future

Dealing with uncertainty

- probability theory
- graphical models
- ...

Reasoning with high-level structure

- logic
- databases
- programming
- ...



Learning

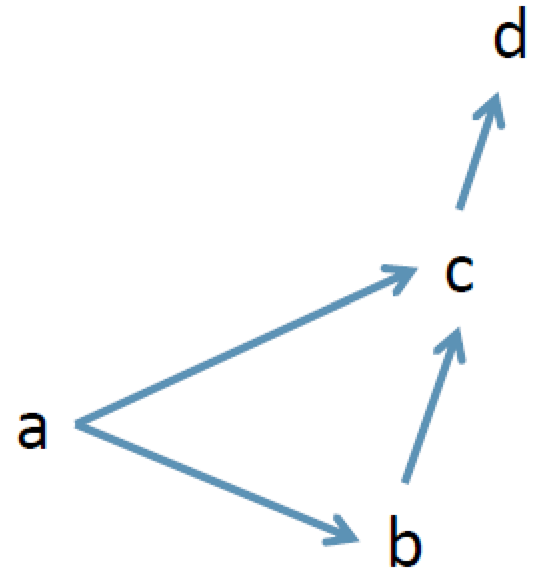
- parameters
- structure

Statistical relational learning, probabilistic logic learning, probabilistic programming, probabilistic databases, ...

Datalog

Edge

x	y
a	c
a	b
b	c
c	d



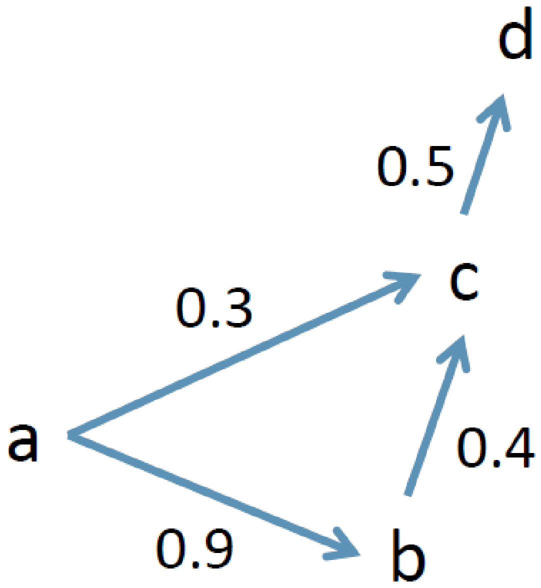

```
path(X,Y):- edge(X,Y).
path(X,Y):- edge(X,Z), path(Z,Y).
```

`path(a,d)` = Yes

Probabilistic Datalog

Edge

x	y	P
a	c	0.3
a	b	0.9
b	c	0.4
c	d	0.5




```
path(X,Y) :- edge(X,Y).
path(X,Y) :- edge(X,Z), path(Z,Y).
```

P(`path(a,d)`) = ??

[De Raedt'07, Fierens'15]

Probabilistic Programming

- Programming language + random variables
- Reason about distribution over executions

As going from hardware circuits to programming languages

```
sample(L,N,S) :- permutation(S,T), sample_ordered(L,N,T).

sample_ordered(_, 0, []).
sample_ordered([X|L], N, [X|S]) :-
     N > 0, sample_now([X|L],N), N2 is N-1,
     sample_ordered(L,N2,S).
sample_ordered([H|L], N, S) :-
     N > 0, \+ sample_now([H|L],N), sample_ordered(L,N,S).

P::sample_now(L,N) :- length(L, M), M >= N, P is N/M.
```

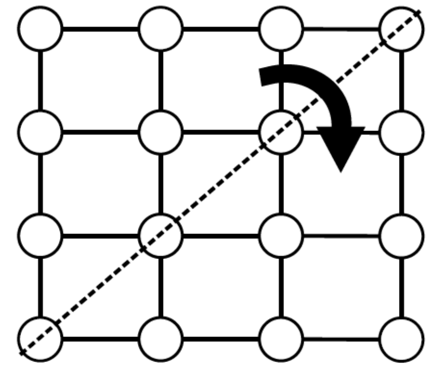
```
P(sample([c,a,c,t,u,s],3,[c,a,t])) = 0.1
```

[De Raedt'07, Fierens'15]

Approximate Symmetries

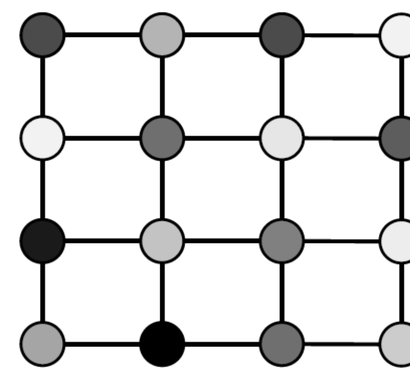
- What if not liftable? Asymmetric graph?
- Exploit approximate symmetries:
  - Exact symmetry g: $\Pr(\mathbf{x}) = \Pr(\mathbf{x}^g)$

    E.g. Ising model
    without external field



  - Approximate symmetry g: $\Pr(\mathbf{x}) \approx \Pr(\mathbf{x}^g)$

    E.g. Ising model with external field



P [] ≈ P []

[VdB'13,'15,Gogate'14]

Example: Statistical Relational Model

- WebKB: Classify pages given links and words
- Very large Markov logic network

$$1.3 \quad \texttt{Page}(\mathsf{x}, \mathsf{Faculty}) \Rightarrow \texttt{HasWord}(\mathsf{x}, \mathsf{Hours})$$

$$1.5 \quad \texttt{Page}(\mathsf{x}, \mathsf{Faculty}) \wedge \texttt{Link}(\mathsf{x}, \mathsf{y}) \Rightarrow \texttt{Page}(\mathsf{y}, \mathsf{Course})$$

and 5000 more …

- No symmetries with evidence on Link or Word
- Where do approx. symmetries come from?

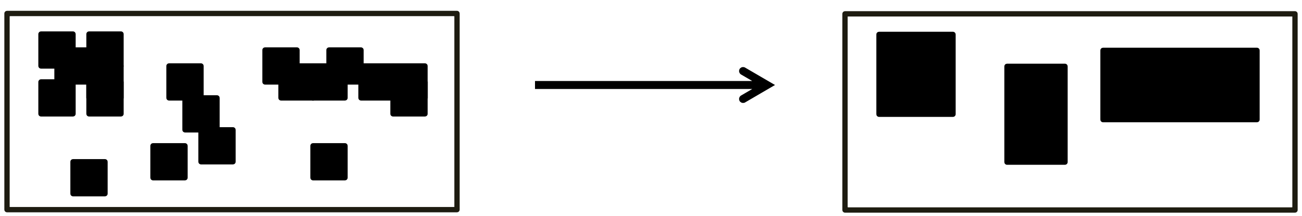
[VdB'13,'15]

Over-Symmetric Approximations

- OSA makes model more symmetric
- E.g., low-rank Boolean matrix factorization

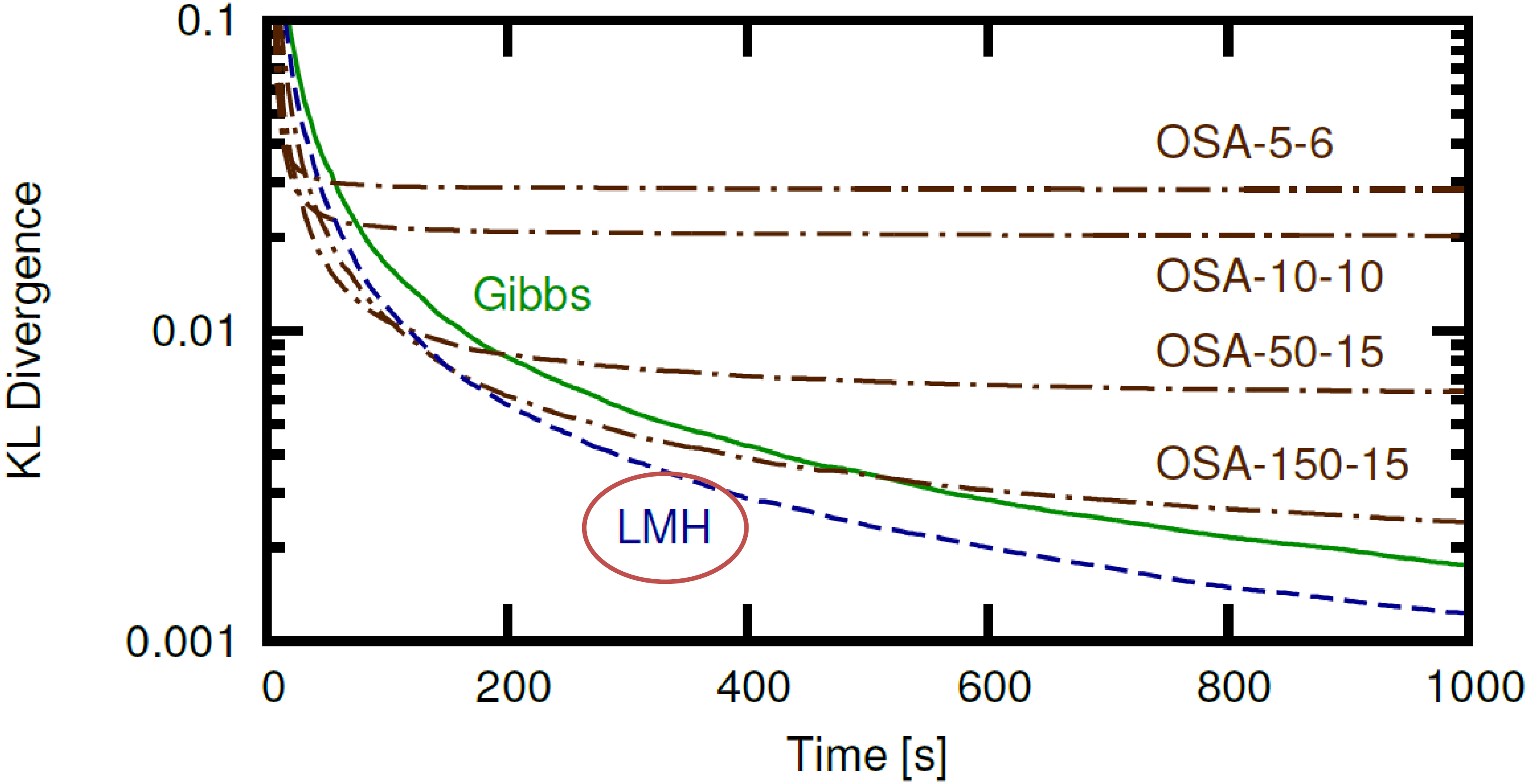
Link ("aaai.org", "google.com") Link ("google.com", "aaai.org") Link ("google.com", "gmail.com") Link ("ibm.com", "aaai.org")	→	Link ("aaai.org", "google.com") Link ("google.com", "aaai.org") - ~~Link ("google.com", "gmail.com")~~ + Link ("aaai.org", "ibm.com") Link ("ibm.com", "aaai.org")

google.com and ibm.com become symmetric!



[VdB'13,'15]

Experiments: WebKB



[VdB'13,'15]

Lifted Weight Learning

- **Given:** A set of first-order logic **formulas**

  w FacultyPage(x) ∧ Linked(x,y) ⇒ CoursePage(y)

  A set of training **databases**

- **Learn:** The associated maximum-likelihood **weights**

$$\frac{\partial}{\partial w_j} \log \Pr_w(db) = n_j(db) - \mathbb{E}_w[n_j]$$

Count in databases
Efficient

Expected counts
Requires inference

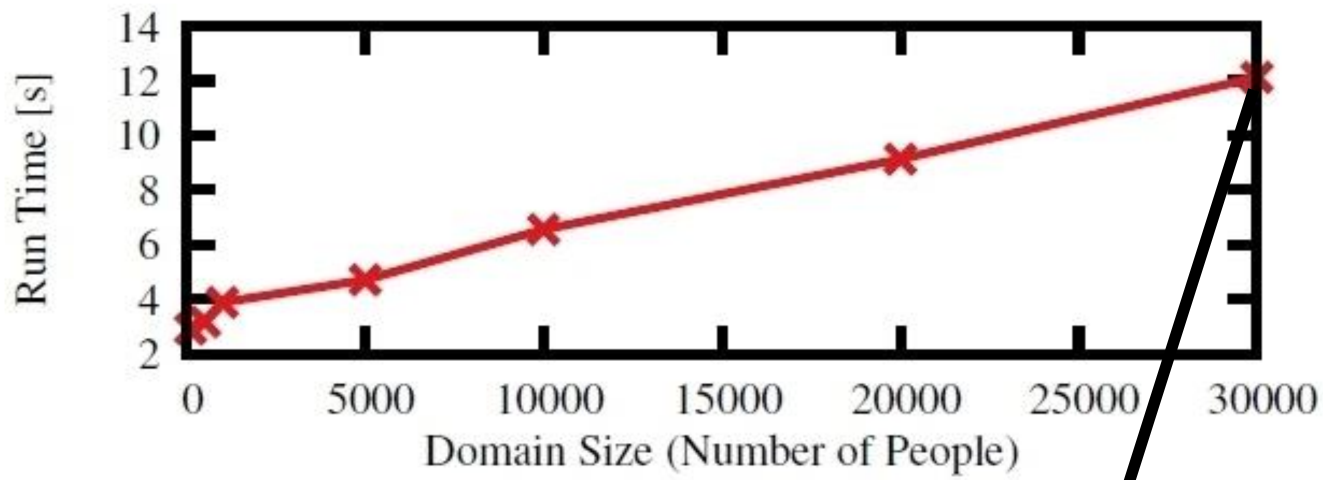
$$\mathbb{E}_w[n_F] = \Pr(F\theta_1) + \cdots + \Pr(F\theta_m)$$

- **Idea:** Lift the computation of $\mathbb{E}_w[n_j]$

[Van Haaren'16]

Learning Time

w Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y)



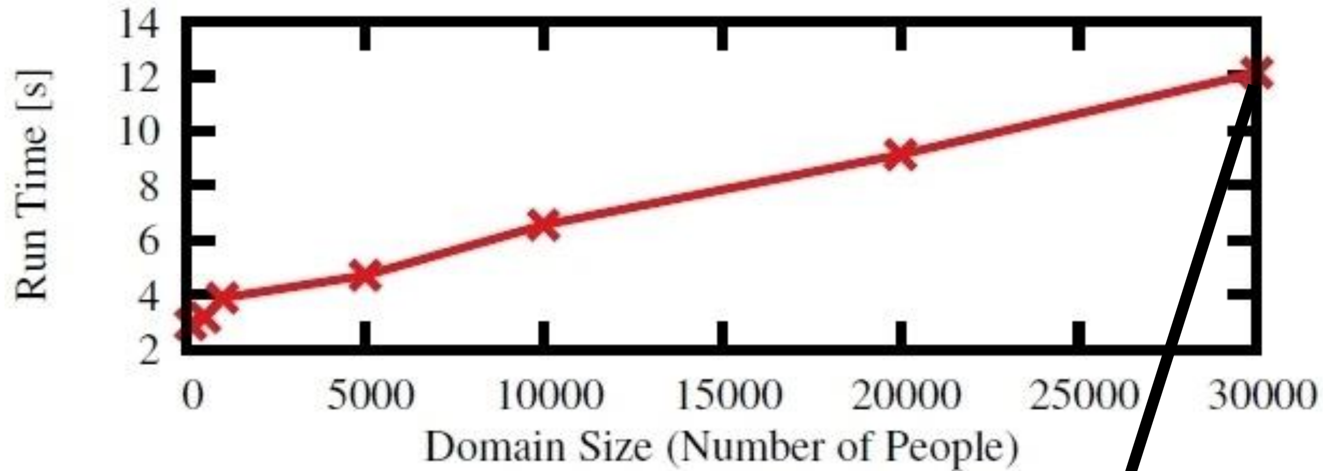
Big data

Learns a model over 900,030,000 random variables

[Van Haaren'16]

Learning Time

w Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y)



~~Big data~~

Big models

Learns a model over
900,030,000 random variables

[Van Haaren'16]

More Lifted Algorithms

- Exact Inference (AI)
  - First-Order Variable Elimination [Poole’03, deSalvoBraz’05, Milch’08, Taghipour’13]
  - First-Order Knowledge Compilation [V.d.Broeck’11,’12,’13]
  - Probabilistic Theorem Proving [Gogate’11]
  - MPE/MAP Inference [deSalvoBraz’06,Apsel’12,Sarkhel’14,Kopp’15]

More Lifted Algorithms

- Approximate Inference (AI)
  - Lifted Belief Propagation [Jaimovich’07, Singla’08, Kersting’09]
  - Lifted Bisimulation/Mini-buckets [Sen’08,‘09]
  - Lifted Importance Sampling [Gogate’11,’12]
  - Lifted Relax, Compensate & Recover [V.d.Broeck’12]
  - Lifted MCMC [Niepert’13,Venugopal’12,VdB’15]
  - Lifted Variational Inference [Choi’12, Bui’12]
  - Lifted MAP-LP [Mladenov’14, Apsel’14]

More Lifted Algorithms

- Other Tasks (AI)
  - Lifted Kalman Filter [Ahmadi'11, Choi'11]
  - Lifted Linear Programming [Mladenov'12]
- Surveys [Kersting'12,Kimmig'15]
- Approximate Query Evaluation (DB)
  - Dissociation [Gatterbauer'13,'14,'15]
  - Collapsed Sampling [Gribkoff'15]
  - Approximate Compilation [Olteanu'10, Dylla'13]

Conclusions

- A radically new reasoning paradigm
- Lifted inference is **frontier** and **integration** of AI, KR, ML, DBs, theory, etc.
- We need
  - relational databases and logic
  - probabilistic models and statistical learning
  - algorithms that scale
- Many theoretical open problems
- Recently cool practical applications

Symmetric Open Problems

- Rules are complete beyond FO^2?
- Lifted approximations
  - Over-symmetric approx. with guarantees
  - Combined with Learning
- Mixed symmetric and asymmetric
- Theoretical computer science connections
  - Understanding #P1
- More SRL applications
- More expressive logics and programs
- Continuous random variables + Logic

Asymmetric Open Problems

- Extensions of the Dichotomy theorem
  - For 0, ½, 1 probabilities
  - FDs, Deterministic tables
  - Negations: ∀FO, ∃FO, or full FO
- Lifted approximation algorithms
- Characterize queries with tractable compilation to: FBDD, SDD, d-DNNF
- Circuit language supporting dichotomy
- Characterize queries with tractable most likely world (MAP = maximum a posterior)

Long-Term Outlook

Probabilistic inference and learning exploit

~ 1988: conditional independence

~ 2000: contextual independence (local structure)

Long-Term Outlook

Probabilistic inference and learning exploit

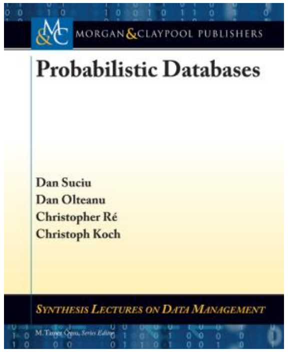
~ 1988: conditional independence

~ 2000: contextual independence (local structure)

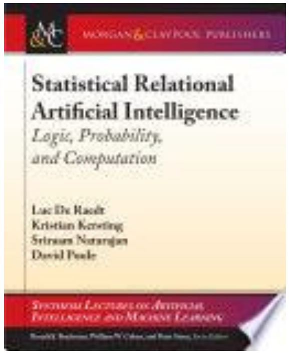
~ 201?: **symmetry & exchangeability & first-order**

If you want more…

- Books
  - Probabilistic Databases
  - Statistical Relational AI
  - (Lifted Inference Book)



[Suciu’11]



[DeRaedt’16]

- StarAI workshop on Monday
  http://www.starai.org

- Main conference papers

Thank You!

Questions?

References

- Suciu, Dan, Dan Olteanu, Christopher Ré, and Christoph Koch. "Probabilistic databases." Synthesis Lectures on Data Management 3, no. 2 (2011): 1-180.
- De Raedt, Luc, Kristian Kersting, Sriraam Natarajan, and David Poole. "Statistical Relational Artificial Intelligence: Logic, Probability, and Computation." Synthesis Lectures on Artificial Intelligence and Machine Learning 10, no. 2 (2016): 1-189.
- Gartner. "Market Share: Relational Database Management Systems by Operating System, Worldwide" (2006).
- Carlson, Andrew, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr, and Tom M. Mitchell. "Toward an Architecture for Never-Ending Language Learning." In AAAI, vol. 5, p. 3. 2010.
- Dong, Xin, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. "Knowledge vault: A web-scale approach to probabilistic knowledge fusion." In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 601-610. ACM, 2014.

References

- Niu, Feng, Ce Zhang, Christopher Ré, and Jude W. Shavlik. "DeepDive: Web-scale Knowledge-base Construction using Statistical Learning and Inference." VLDS 12 (2012): 25-28.
- Chen, Brian X. "Siri, Alexa and Other Virtual Assistants Put to the Test" The New York Times (2016).
- Richardson, Matthew, and Pedro Domingos. "Markov logic networks." Machine learning 62, no. 1-2 (2006): 107-136.
- Getoor, Lise, Ben Taskar. Introduction to statistical relational learning. MIT press, 2007.
- Van den Broeck, Guy, Ingo Thon, Martijn Van Otterlo, and Luc De Raedt. "DTProbLog: A decision-theoretic probabilistic Prolog." In Proceedings of the twenty-fourth AAAI conference on artificial intelligence, pp. 1217-1222. AAAI Press, 2010.
- Bach, Stephen, Bert Huang, Ben London, and Lise Getoor. "Hinge-loss Markov random fields: Convex inference for structured prediction." arXiv preprint arXiv:1309.6813 (2015).

References

- Van den Broeck, Guy. "Towards high-level probabilistic reasoning with lifted inference." AAAI Spring Symposium on KRR (2015).
- Niepert, Mathias, and Guy Van den Broeck. "Tractability through exchangeability: A new perspective on efficient probabilistic inference." AAAI (2014).
- Van den Broeck, Guy. Lifted inference and learning in statistical relational models. Diss. Ph. D. Dissertation, KU Leuven, 2013.
- Vardi, Moshe Y. "The complexity of relational query languages." In Proceedings of the fourteenth annual ACM symposium on Theory of computing, pp. 137-146. ACM, 1982.
- Olteanu, Dan, and Jiewen Huang. "Using OBDDs for efficient query evaluation on probabilistic databases." In International Conference on Scalable Uncertainty Management, pp. 326-340. Springer Berlin Heidelberg, 2008.

References

- Jha, Abhay, and Dan Suciu. "Knowledge compilation meets database theory: compiling queries to decision diagrams." ICDT (2011).
- Dalvi, Nilesh, and Dan Suciu. "The dichotomy of probabilistic inference for unions of conjunctive queries." Journal of the ACM (JACM) 59, no. 6 (2012): 30.
- Chavira, Mark, and Adnan Darwiche. "Compiling Bayesian networks with local structure." In IJCAI, vol. 5, pp. 1306-1312. 2005.
- Sang, Tian, Paul Beame, and Henry A. Kautz. "Performing Bayesian inference by weighted model counting." In AAAI, vol. 5, pp. 475-481. 2005.
- Chavira, Mark, and Adnan Darwiche. "On probabilistic inference by weighted model counting." Artificial Intelligence 172, no. 6 (2008): 772-799.

References

- Choi, Arthur, Doga Kisa, and Adnan Darwiche. "Compiling probabilistic graphical models using sentential decision diagrams." In European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty, pp. 121-132. Springer Berlin Heidelberg, 2013.

- Chavira, Mark, Adnan Darwiche, and Manfred Jaeger. "Compiling relational Bayesian networks for exact inference." International Journal of Approximate Reasoning 42, no. 1 (2006): 4-20.

- Fierens, Daan, Guy Van den Broeck, Ingo Thon, Bernd Gutmann, and Luc De Raedt. "Inference in probabilistic logic programs using weighted CNF's." UAI (2011).

- Fierens, Daan, Guy Van den Broeck, Joris Renkens, Dimitar Shterionov, Bernd Gutmann, Ingo Thon, Gerda Janssens, and Luc De Raedt. "Inference and learning in probabilistic logic programs using weighted boolean formulas." Theory and Practice of Logic Programming 15, no. 03 (2015): 358-401.

- Van den Broeck, Guy, Nima Taghipour, Wannes Meert, Jesse Davis, and Luc De Raedt. "Lifted probabilistic inference by first-order knowledge compilation." In Proceedings of the Twenty-Second international joint conference on Artificial Intelligence, pp. 2178-2185. AAAI Press/International Joint Conferences on Artificial Intelligence, 2011.

References

- Gogate, Vibhav, and Pedro Domingos. "Probabilistic theorem proving." UAI , 2011.
- Van den Broeck, Guy, Wannes Meert, and Adnan Darwiche. "Skolemization for weighted first-order model counting." In Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR). 2014.
- Gribkoff, Eric, Guy Van den Broeck, and Dan Suciu. "Understanding the complexity of lifted inference and asymmetric weighted model counting." UAI, 2014.
- Jha, Abhay, and Dan Suciu. "Probabilistic databases with MarkoViews." Proceedings of the VLDB Endowment 5, no. 11 (2012): 1160-1171.
- Van den Broeck, Guy. "On the completeness of first-order knowledge compilation for lifted probabilistic inference." In Advances in Neural Information Processing Systems, pp. 1386-1394. 2011.
- Jaeger, Manfred, and Guy Van den Broeck. "Liftability of probabilistic inference: Upper and lower bounds." In Proceedings of the 2nd international workshop on statistical relational AI. 2012.

References

- Darwiche, Adnan. "Decomposable negation normal form." Journal of the ACM (JACM) 48, no. 4 (2001): 608-647.
- Cook, Stephen A. "The complexity of theorem-proving procedures." In Proceedings of the third annual ACM symposium on Theory of computing, pp. 151-158. ACM, 1971.
- Valiant, Leslie G. "The complexity of computing the permanent." Theoretical computer science 8, no. 2 (1979): 189-201.
- Provan, J. Scott, and Michael O. Ball. "The complexity of counting cuts and of computing the probability that a graph is connected." SIAM Journal on Computing 12, no. 4 (1983): 777-788.
- Dalvi, Nilesh, and Dan Suciu. "Efficient query evaluation on probabilistic databases." The VLDB Journal 16, no. 4 (2004): 523-544.

References

- Gomes, Carla P., Ashish Sabharwal, and Bart Selman. "Model counting." (2008).
- Huang, Jinbo, and Adnan Darwiche. "DPLL with a trace: From SAT to knowledge compilation." In IJCAI, vol. 5, pp. 156-162. 2005.
- Razgon, Igor. "On the read-once property of branching programs and CNFs of bounded treewidth." Algorithmica (2015): 1-18.
- Beame, Paul, Jerry Li, Sudeepa Roy, and Dan Suciu. "Model counting of query expressions: Limitations of propositional methods." ICDT (2014).
- Bova, Simone. "SDDs are Exponentially More Succinct than OBDDs." AAAI (2016).
- Beame, Paul, and Vincent Liew. "New Limits for Knowledge Compilation and Applications to Exact Model Counting." UAI (2015).

References

- Valiant, Leslie G. "The complexity of enumeration and reliability problems." SIAM Journal on Computing 8, no. 3 (1979): 410-421.
- Bertoni, Alberto, Massimiliano Goldwurm, and Nicoletta Sabadini. "The complexity of computing the number of strings of given length in context-free languages." Theoretical Computer Science 86, no. 2 (1991): 325-342.
- Beame, Paul, Guy Van den Broeck, Eric Gribkoff, and Dan Suciu. "Symmetric weighted first-order model counting." In Proceedings of the 34th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, pp. 313-328. ACM, 2015.
- Fagin, Ronald. "Generalized first-order spectra and polynomial-time recognizable sets." (1974).
- Ceylan, Ismail Ilkan, Adnan Darwiche, and Guy Van den Broeck. "Open-world probabilistic databases." Proc. of KR 16 (2016).

References

- De Raedt, Luc, Angelika Kimmig, and Hannu Toivonen. "ProbLog: A Probabilistic Prolog and Its Application in Link Discovery." In IJCAI, vol. 7, pp. 2462-2467. 2007.

- Van den Broeck, Guy, and Adnan Darwiche. "On the complexity and approximation of binary evidence in lifted inference." In Advances in Neural Information Processing Systems, pp. 2868-2876. 2013.

- Van den Broeck, Guy, and Mathias Niepert. "Lifted probabilistic inference for asymmetric graphical models.“ AAAI (2015).

- Venugopal, Deepak, and Vibhav Gogate. "Evidence-based clustering for scalable inference in markov logic." In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 258-273. Springer Berlin Heidelberg, 2014.

- Van Haaren, Jan, Guy Van den Broeck, Wannes Meert, and Jesse Davis. "Lifted generative learning of Markov logic networks." Machine Learning 103, no. 1 (2016): 27-55.

References

- Poole, David. "First-order probabilistic inference." In IJCAI, vol. 3, pp. 985-991. 2003.
- de Salvo Braz, Rodrigo, Eyal Amir, and Dan Roth. "Lifted first-order probabilistic inference." In Proceedings of the 19th international joint conference on Artificial intelligence, pp. 1319-1325. 2005.
- Milch, Brian, Luke S. Zettlemoyer, Kristian Kersting, Michael Haimes, and Leslie Pack Kaelbling. "Lifted Probabilistic Inference with Counting Formulas." In Aaai, vol. 8, pp. 1062-1068. 2008.
- Taghipour, Nima, Daan Fierens, Jesse Davis, and Hendrik Blockeel. "Lifted Variable Elimination with Arbitrary Constraints." In AISTATS, pp. 1194-1202. 2012.
- Van den Broeck, Guy, and Jesse Davis. "Conditioning in first-order knowledge compilation and lifted probabilistic inference." In Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, pp. 1-7. AAAI Press, 2012.

References

- de Salvo Braz, Rodrigo, Eyal Amir, and Dan Roth. "MPE and partial inversion in lifted probabilistic variable elimination." In AAAI, vol. 6, pp. 1123-1130. 2006.
- Apsel, Udi, and Ronen I. Brafman. "Exploiting uniform assignments in first-order mpe." UAI (2012).
- Sarkhel, Somdeb, Deepak Venugopal, Parag Singla, and Vibhav Gogate. "Lifted MAP Inference for Markov Logic Networks." In AISTATS, pp. 859-867. 2014.
- Kopp, Timothy, Parag Singla, and Henry Kautz. "Lifted symmetry detection and breaking for map inference." In Advances in Neural Information Processing Systems, pp. 1315-1323. 2015.
- Jaimovich, Ariel, Ofer Meshi, and Nir Friedman. "Template based inference in symmetric relational Markov random fields." UAI (2007).
- Singla, Parag, and Pedro M. Domingos. "Lifted First-Order Belief Propagation." In AAAI, vol. 8, pp. 1094-1099. 2008.

References

- Kersting, Kristian, Babak Ahmadi, and Sriraam Natarajan. "Counting belief propagation." In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pp. 277-284. AUAI Press, 2009.

- Sen, Prithviraj, Amol Deshpande, and Lise Getoor. "Exploiting shared correlations in probabilistic databases." Proceedings of the VLDB Endowment 1, no. 1 (2008): 809-820.

- Sen, Prithviraj, Amol Deshpande, and Lise Getoor. "Bisimulation-based approximate lifted inference." In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pp. 496-505. AUAI Press, 2009.

- Gogate, Vibhav, Abhay Kumar Jha, and Deepak Venugopal. "Advances in Lifted Importance Sampling." In AAAI. 2012.

- Van den Broeck, Guy, Arthur Choi, and Adnan Darwiche. "Lifted relax, compensate and then recover: From approximate to exact lifted probabilistic inference." UAI (2012).

References

- Niepert, Mathias. "Symmetry-aware marginal density estimation." AAAI (2013).
- Venugopal, Deepak, and Vibhav Gogate. "On lifting the gibbs sampling algorithm." In Advances in Neural Information Processing Systems, pp. 1655-1663. 2012.
- Choi, Jaesik, and Eyal Amir. "Lifted relational variational inference." UAI (2012).
- Bui, Hung Hai, Tuyen N. Huynh, and Sebastian Riedel. "Automorphism groups of graphical models and lifted variational inference." StarAI (2012).
- Mladenov, Martin, Kristian Kersting, and Amir Globerson. "Efficient Lifting of MAP LP Relaxations Using k-Locality." In AISTATS, pp. 623-632. 2014.
- Apsel, Udi, Kristian Kersting, and Martin Mladenov. "Lifting Relational MAP-LPs Using Cluster Signatures." In AAAI Workshop: Statistical Relational Artificial Intelligence. 2014.

References

- Ahmadi, Babak, Kristian Kersting, and Scott Sanner. "Multi-evidence lifted message passing, with application to pagerank and the kalman filter." In IJCAI Proceedings-International Joint Conference on Artificial Intelligence, vol. 22, no. 1, p. 1152. 2011.

- Choi, Jaesik, Abner Guzman-Rivera, and Eyal Amir. "Lifted Relational Kalman Filtering." In IJCAI, pp. 2092-2099. 2011.

- Mladenov, Martin, Babak Ahmadi, and Kristian Kersting. "Lifted Linear Programming." In AISTATS, pp. 788-797. 2012.

- Kimmig, Angelika, Lilyana Mihalkova, and Lise Getoor. "Lifted graphical models: a survey." Machine Learning 99, no. 1 (2015): 1-45.

- Kersting, Kristian. "Lifted Probabilistic Inference." In ECAI, pp. 33-38. 2012.

- Gatterbauer, Wolfgang, and Dan Suciu. "Dissociation and propagation for efficient query evaluation over probabilistic databases." arXiv preprint arXiv:1310.6257 (2013).

References

- Gatterbauer, Wolfgang, and Dan Suciu. "Oblivious bounds on the probability of boolean functions." ACM Transactions on Database Systems (TODS) 39, no. 1 (2014): 5.
- Gatterbauer, Wolfgang, and Dan Suciu. "Approximate lifted inference with probabilistic databases." Proceedings of the VLDB Endowment 8, no. 5 (2015): 629-640.
- Gribkoff, Eric, and Dan Suciu. "SlimShot: Probabilistic inference for web-scale knowledge bases." (2015).
- Olteanu, Dan, Jiewen Huang, and Christoph Koch. "Approximate confidence computation in probabilistic databases." In 2010 IEEE 26th International Conference on Data Engineering (ICDE 2010), pp. 145-156. IEEE, 2010.
- Dylla, Maximilian, Iris Miliaraki, and Martin Theobald. "Top-k query processing in probabilistic databases with non-materialized views." In Data Engineering (ICDE), 2013 IEEE 29th International Conference on, pp. 122-133. IEEE, 2013.