Symmetry in Probabilistic Databases

Guy Van den Broeck
KU Leuven

Joint work with

Dan Suciu, Paul Beame, Eric Gribkoff, Wannes Meert, Adnan Darwiche

Based on NIPS 2011, KR 2014, and upcoming PODS 2015 paper

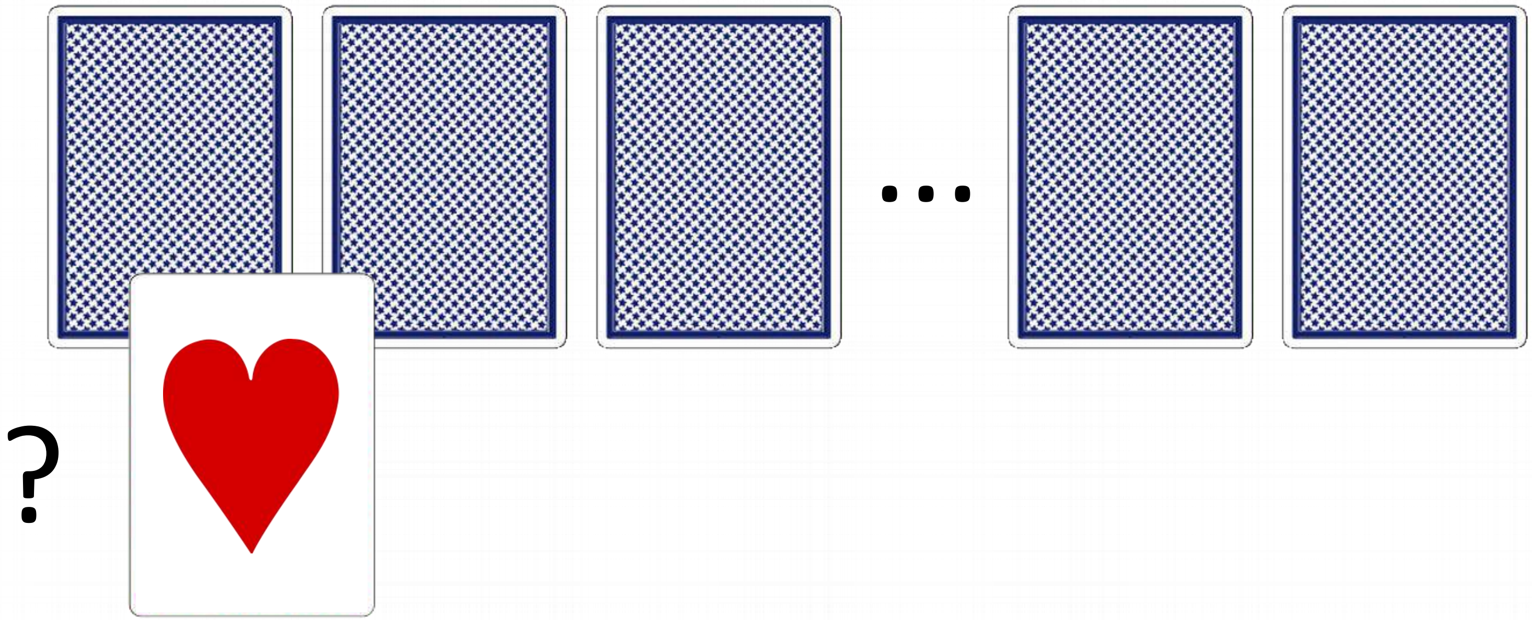
Overview

- Motivation and convergence of
  - The artificial intelligence story (*recap*)
  - The machine learning story (*recap*)
  - The probabilistic database story
  - The database theory story
- Main theoretical results and proof outlines
- Discussion and conclusions
- Dessert

Overview

- Motivation and convergence of
  - **The artificial intelligence story (*recap*)**
  - The machine learning story (*recap*)
  - The probabilistic database story
  - The database theory story
- Main theoretical results and proof outlines
- Discussion and conclusions
- Dessert

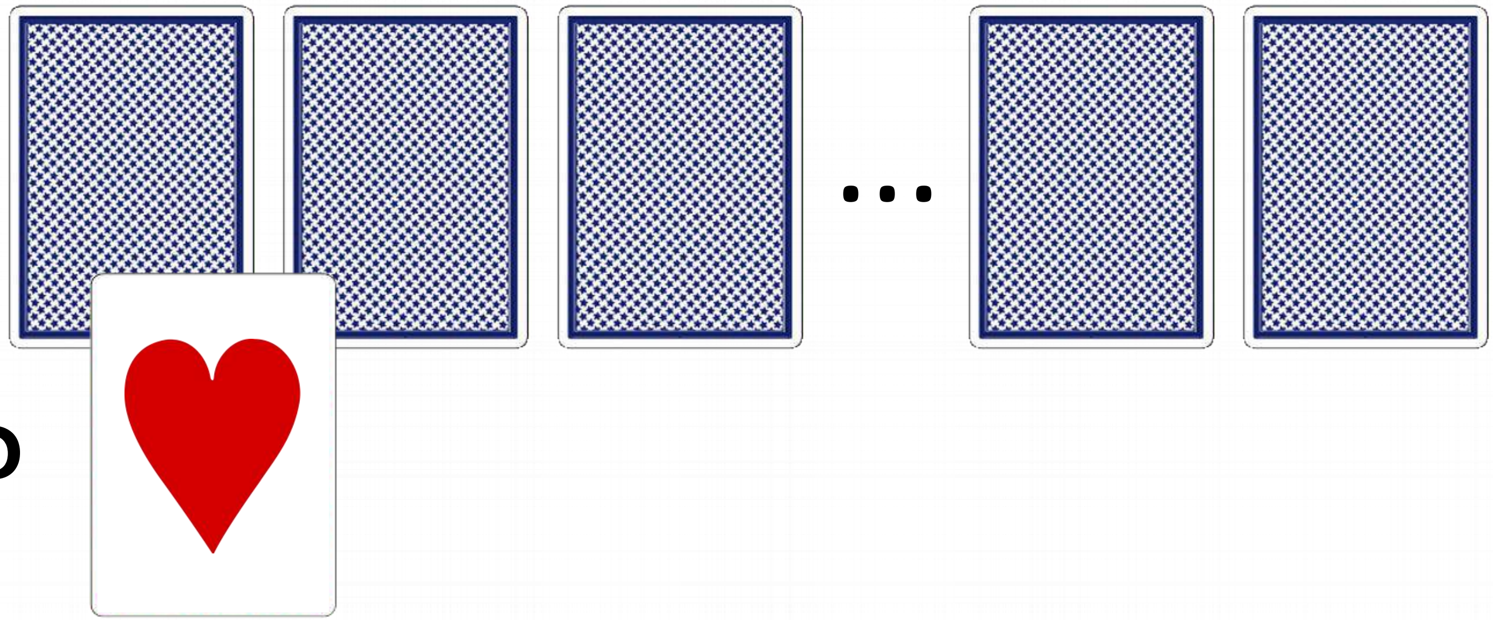
A Simple Reasoning Problem



Probability that Card1 is Hearts?

[Van den Broeck; AAAI-KRR'15]

A Simple Reasoning Problem

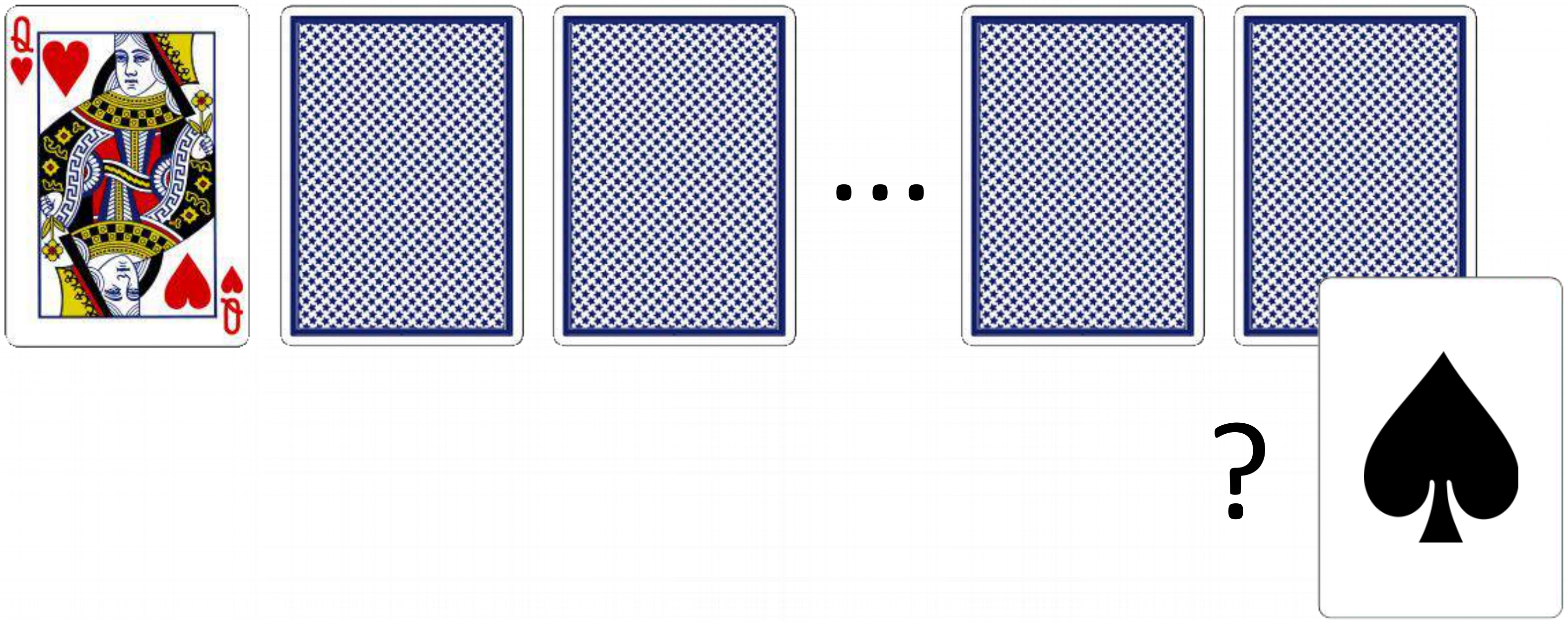


?

Probability that Card1 is Hearts? 1/4

[Van den Broeck; AAAI-KRR'15]

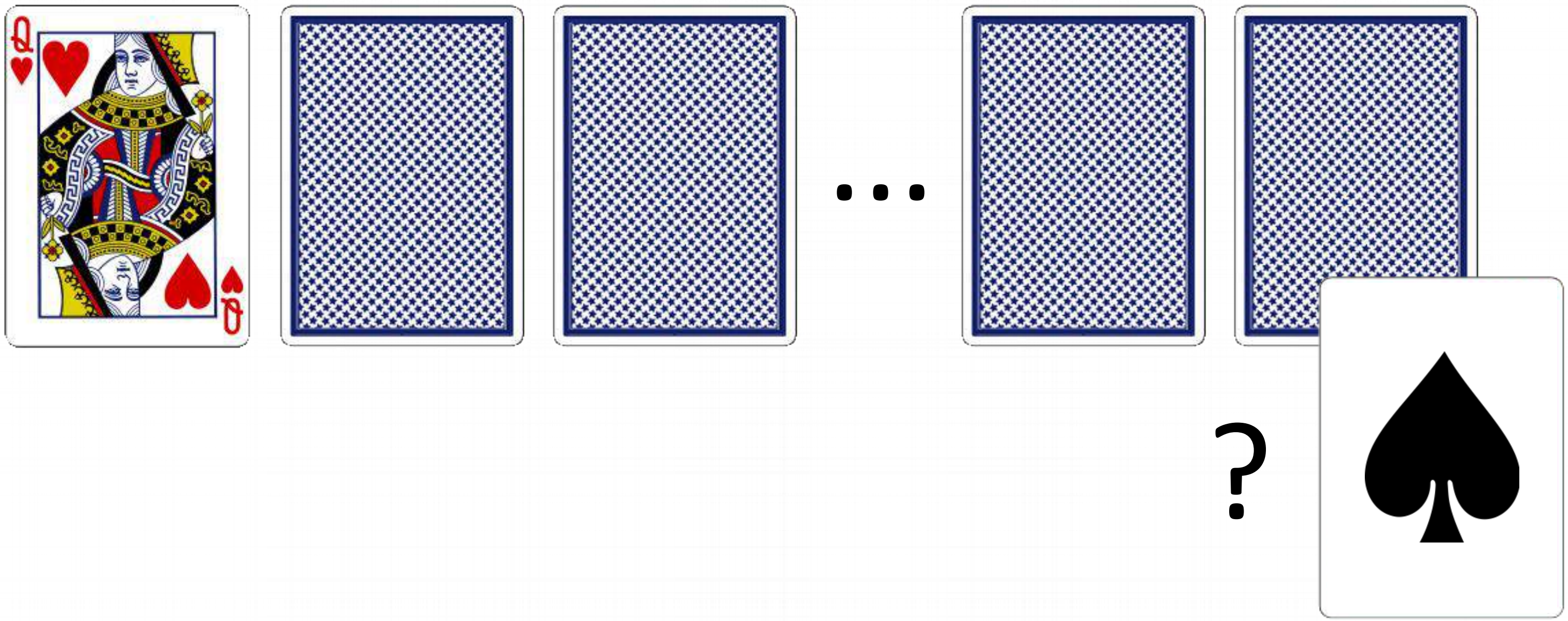
A Simple Reasoning Problem



Probability that Card52 is Spades given that Card1 is QH?

[Van den Broeck; AAAI-KRR'15]

A Simple Reasoning Problem



Probability that Card52 is Spades given that Card1 is QH?

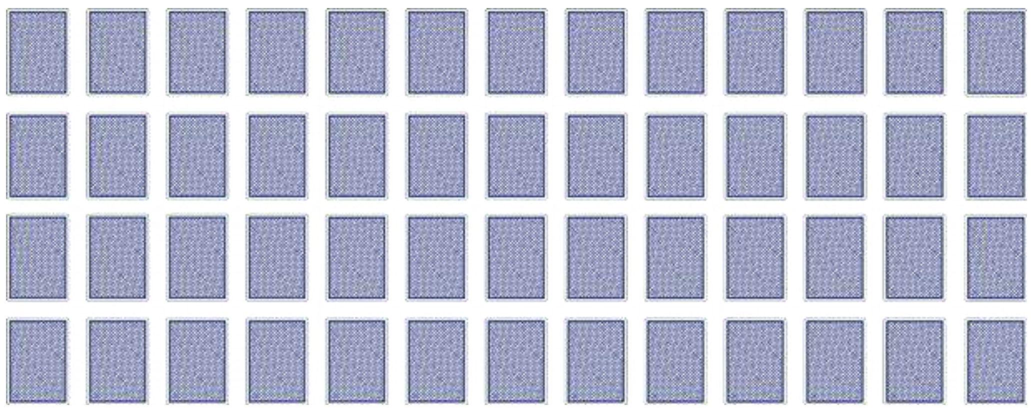
13/51

[Van den Broeck; AAAI-KRR'15]

Automated Reasoning

Let us automate this:

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

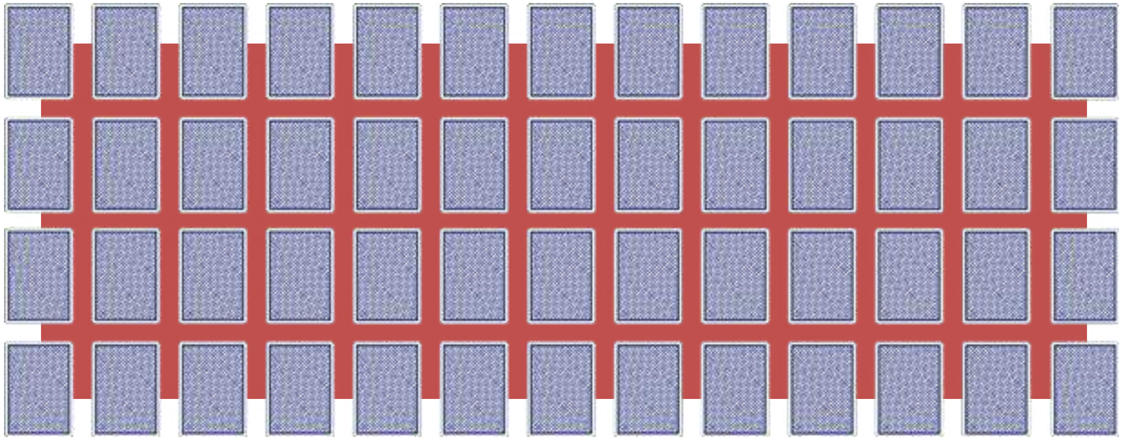


2. Probabilistic inference algorithm
   (e.g., variable elimination or junction tree)

Automated Reasoning

Let us automate this:

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

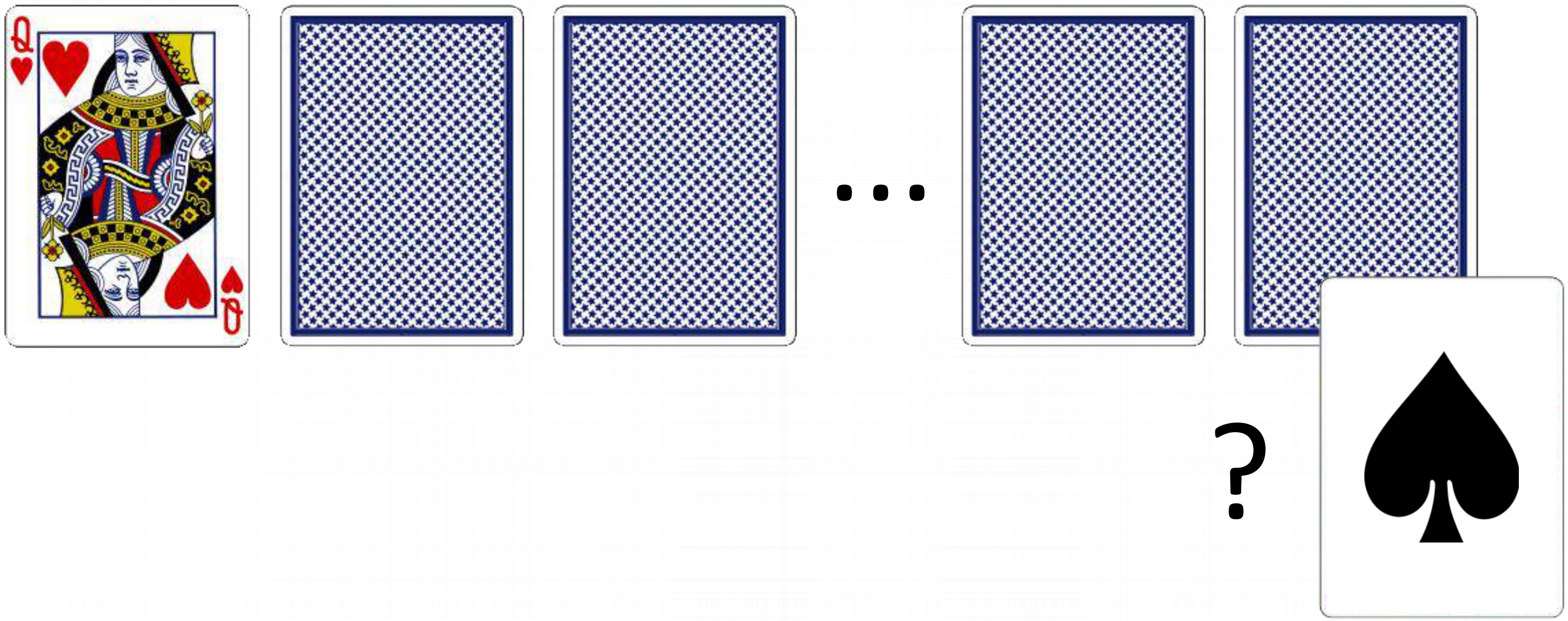


(artist's impression)

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

[Van den Broeck; AAAI-KRR'15]

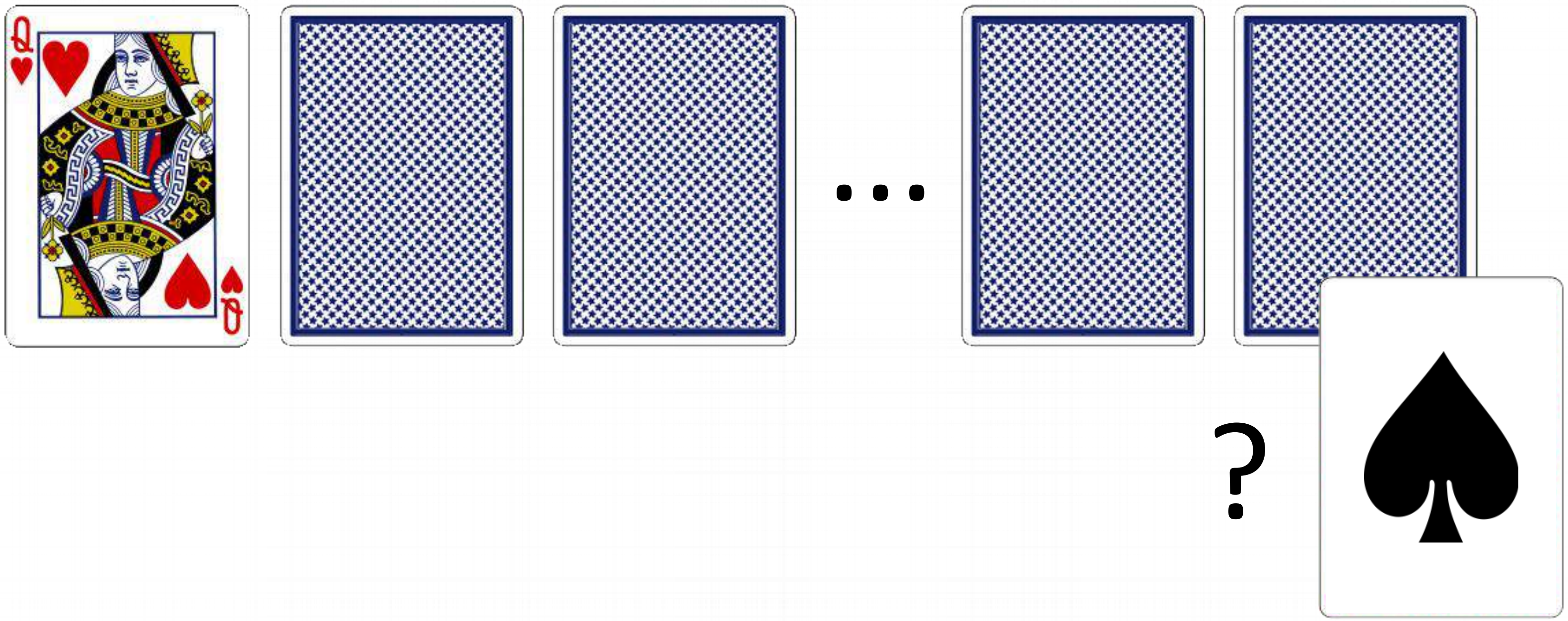
What's Going On Here?



Probability that Card52 is Spades given that Card1 is QH?

[Van den Broeck; AAAI-KRR’15]

What's Going On Here?

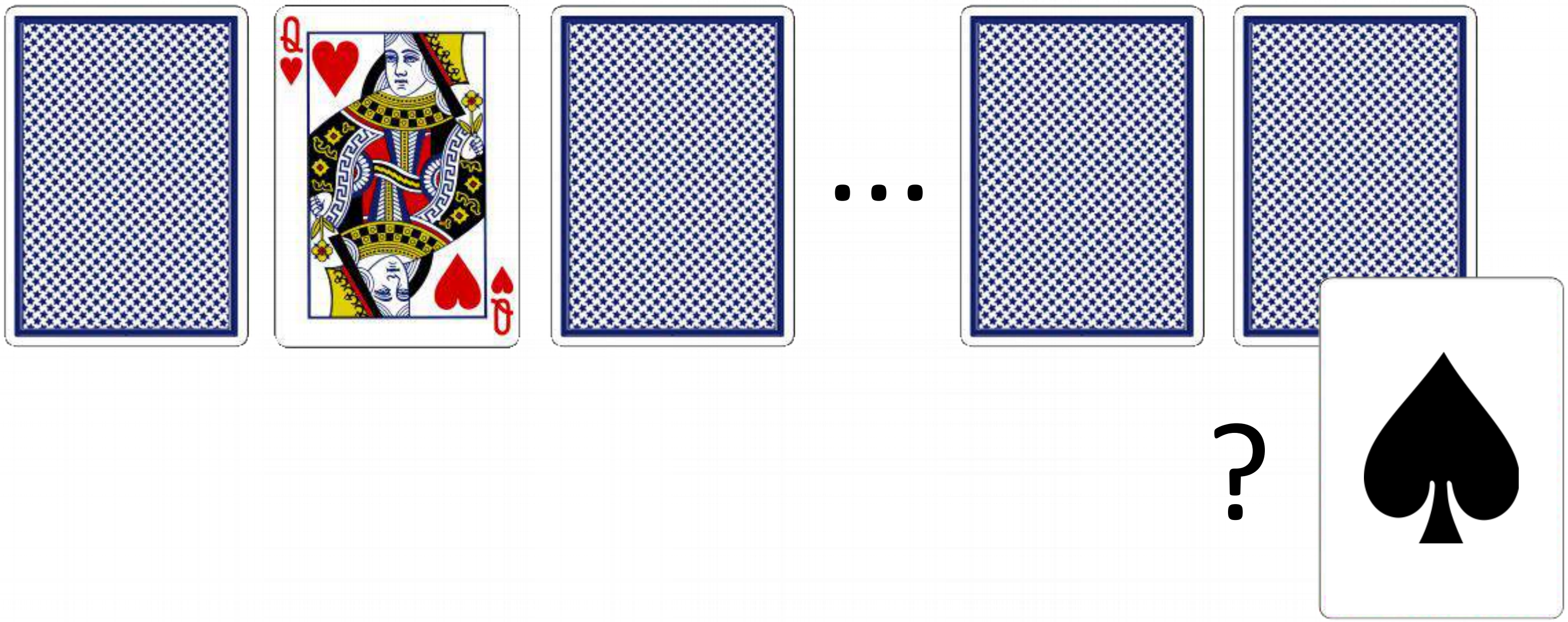


Probability that Card52 is Spades given that Card1 is QH?

13/51

[Van den Broeck; AAAI-KRR'15]

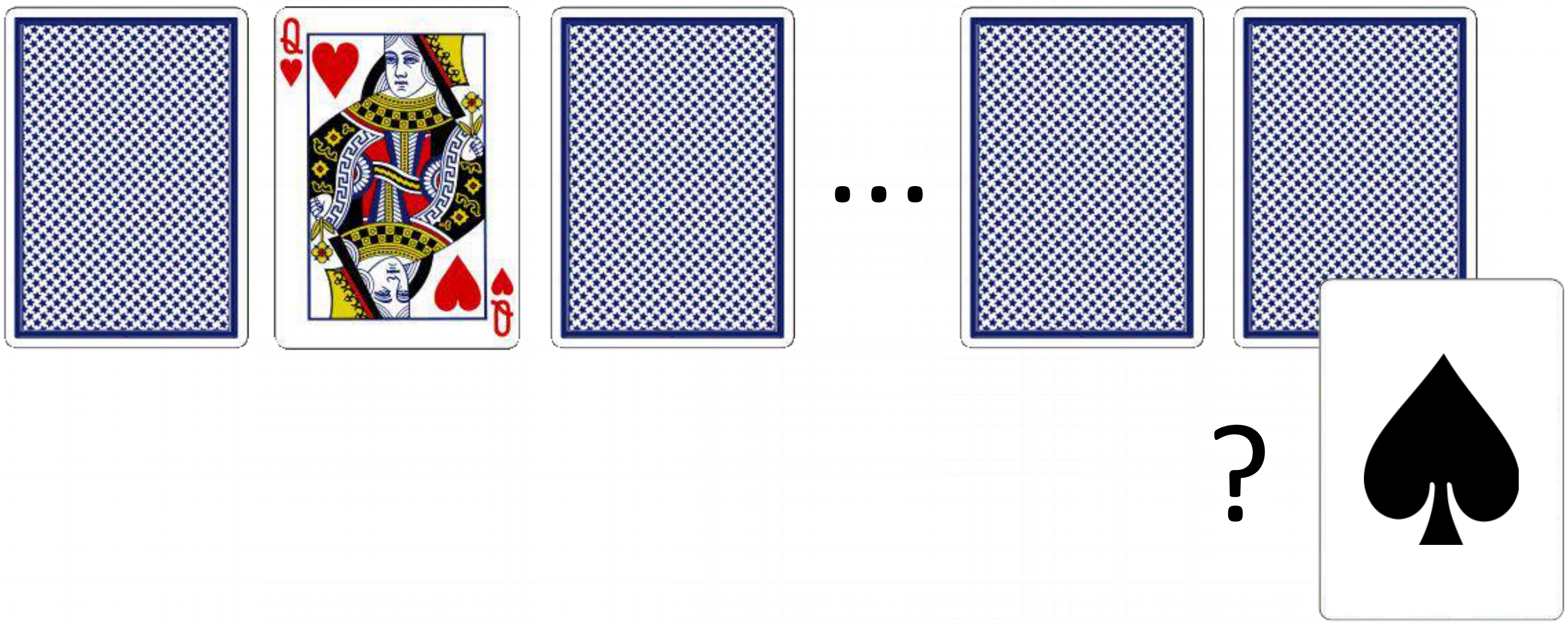
What's Going On Here?



Probability that Card52 is Spades given that Card2 is QH?

[Van den Broeck; AAAI-KRR’15]

What's Going On Here?

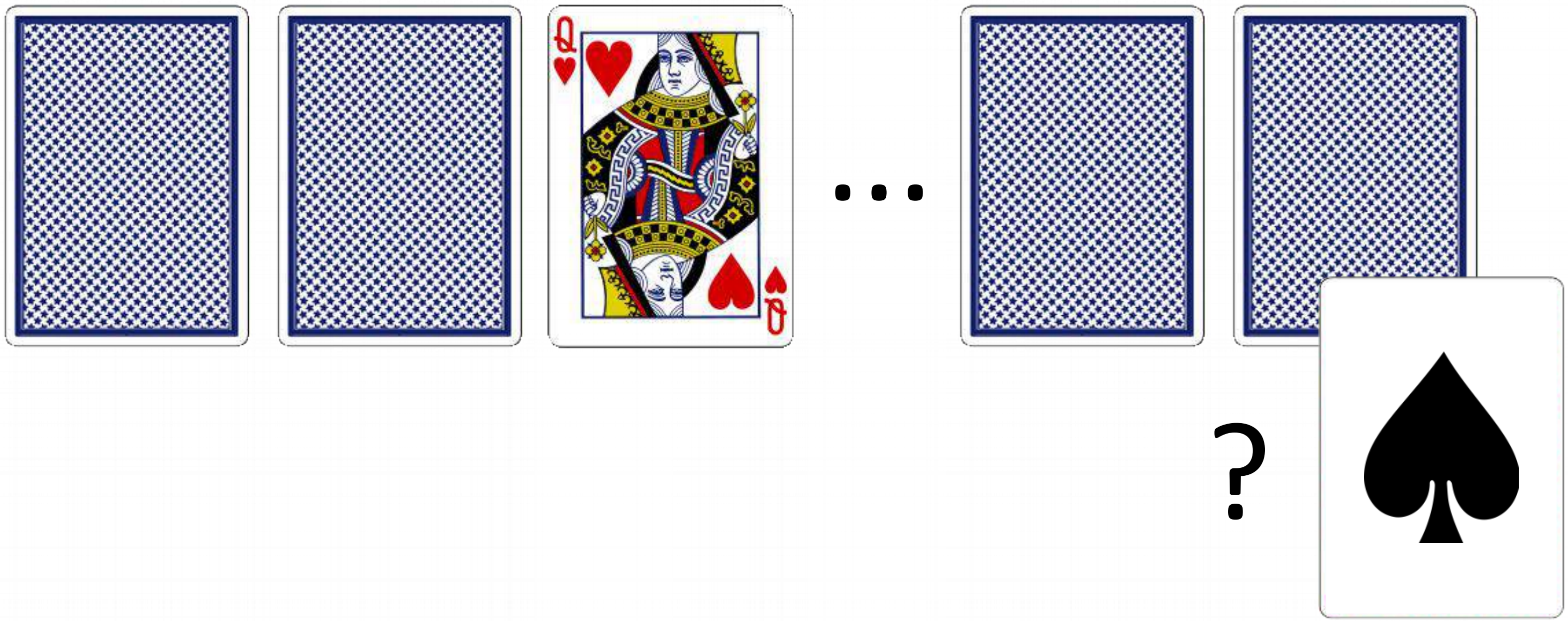


Probability that Card52 is Spades given that Card2 is QH?

13/51

[Van den Broeck; AAAI-KRR’15]

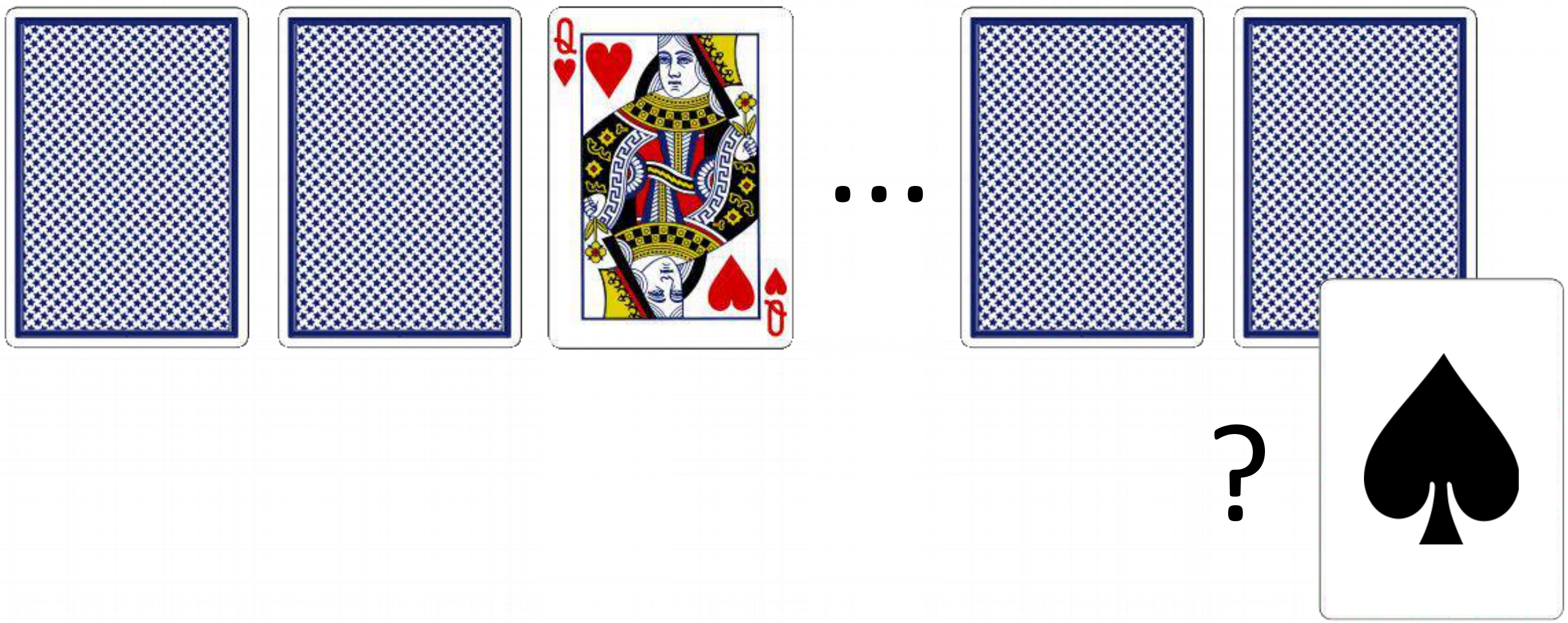
What's Going On Here?



Probability that Card52 is Spades given that Card3 is QH?

[Van den Broeck; AAAI-KRR’15]

What's Going On Here?

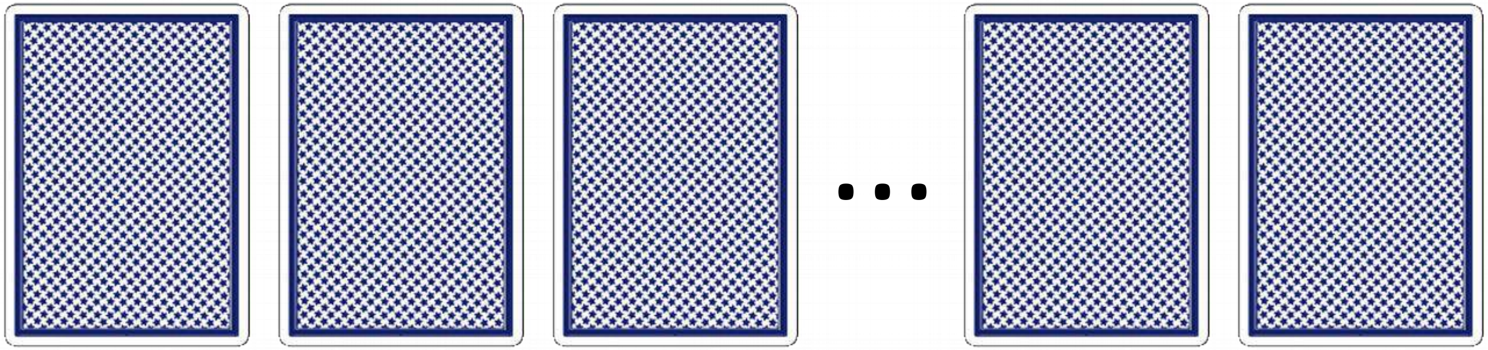


Probability that Card52 is Spades given that Card3 is QH?

13/51

[Van den Broeck; AAAI-KRR'15]

Tractable Probabilistic Inference



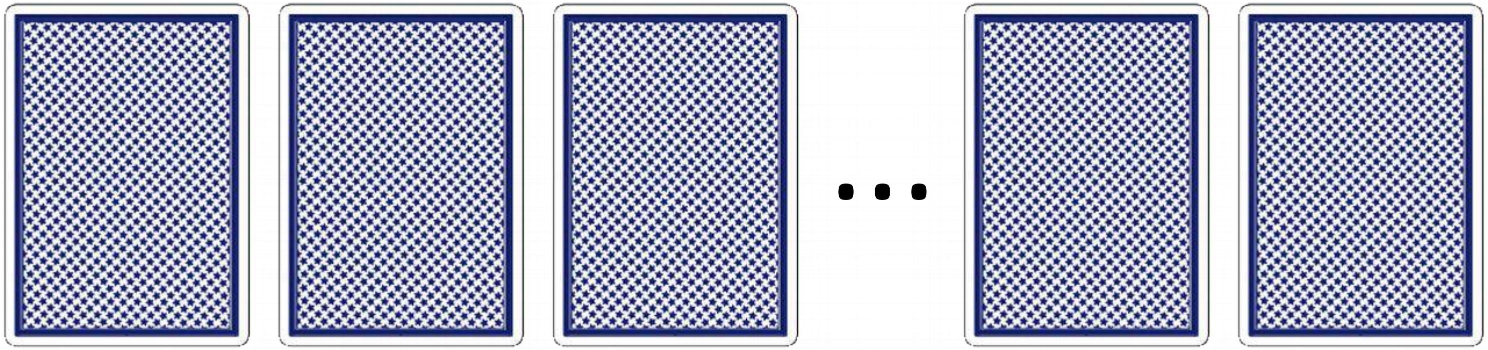
Which property makes inference tractable?

~~Traditional belief: Independence~~

What's going on here?

[Niepert, Van den Broeck; AAAI'14], [Van den Broeck; AAAI-KRR'15]

Tractable Probabilistic Inference



Which property makes inference tractable?

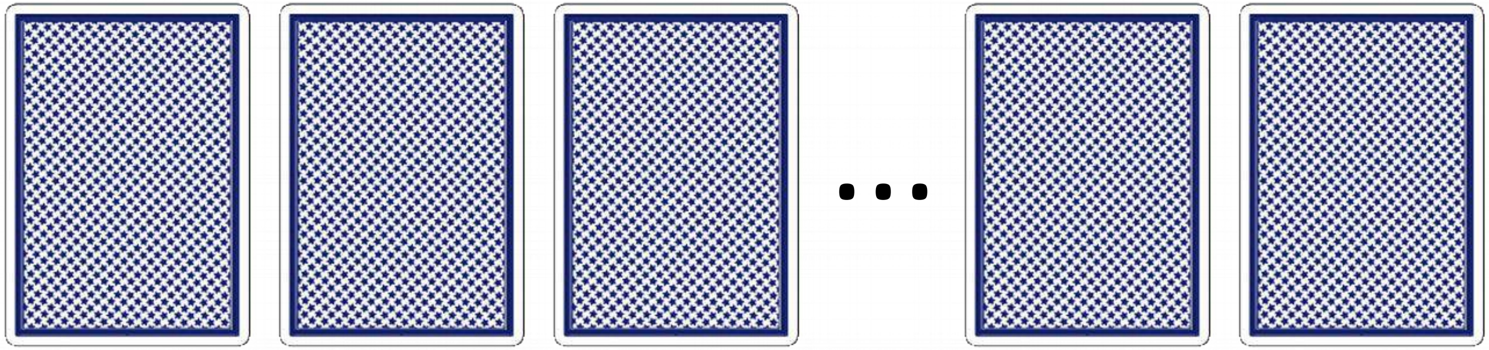
~~Traditional belief: Independence~~

What's going on here?

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

⇒ Lifted Inference

[Niepert, Van den Broeck; AAAI'14], [Van den Broeck; AAAI-KRR'15]



Let us automate this:

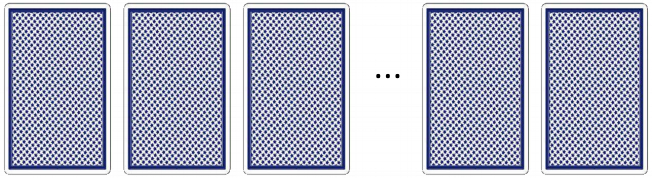
- **Relational** model

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

- **Lifted** probabilistic inference algorithm

Playing Cards Revisited

Let us automate this:



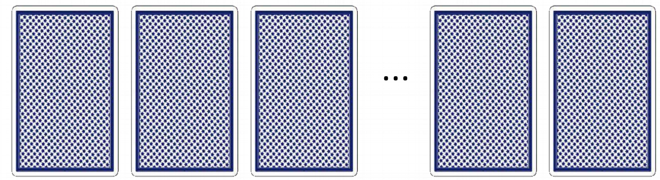
$\forall p, \exists c, Card(p,c)$

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

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

[Van den Broeck.; AAAI-KR'15]

Playing Cards Revisited

Let us automate this: 

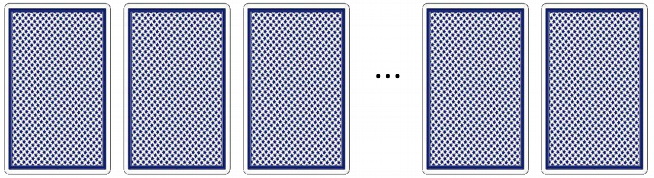
∀p, ∃c, Card(p,c)
∀c, ∃p, Card(p,c)
∀p, ∀c, ∀c’, Card(p,c) ∧ Card(p,c’) ⇒ c = c’

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

[Van den Broeck.; AAAI-KR’15]

Playing Cards Revisited

Let us automate this:



∀p, ∃c, Card(p,c)
∀c, ∃p, Card(p,c)
∀p, ∀c, ∀c’, Card(p,c) ∧ Card(p,c’) ⇒ c = c’

↓

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

Computed in time polynomial in n

[Van den Broeck.; AAAI-KR’15]

Model Counting

- 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

[Valiant] #P-hard, even for 2CNF

First-Order Model Counting

Model = solution to first-order logic formula Δ

Δ = ∀d (Rain(d)
⇒ Cloudy(d))

Days = {Monday}

First-Order Model Counting

Model = solution to first-order logic formula Δ

$\Delta = \forall d\ (\text{Rain}(d) \Rightarrow \text{Cloudy}(d))$

Days = {Monday}

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

+

FOMC = 3

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

First-Order Model Counting

Model = solution to first-order logic formula Δ

$\Delta = \forall d\ (\mathrm{Rain}(d) \Rightarrow \mathrm{Cloudy}(d))$

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

+

FOMC = 9

FOMC Inference: Example 1

FOMC Inference: Example 1

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

Domain = {n people}

FOMC Inference: Example 1

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

Domain = {n people}

→ 3^n models

FOMC Inference: Example 1

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

Domain = {n people}

→ 3^n models

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

D = {n people}

FOMC Inference: Example 1

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

→ 3^n models

2. $\boxed{\Delta = \forall y, (ParentOf(y) \wedge Female \Rightarrow MotherOf(y))}$ $\boxed{D = \{n \text{ people}\}}$

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

FOMC Inference: Example 1

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

Domain = {n people}

→ 3^n models

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

D = {n people}

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

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

FOMC Inference: Example 1

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

$\rightarrow 3^n$ models

2. $\Delta = \forall y, (\text{ParentOf}(y) \wedge \text{Female} \Rightarrow \text{MotherOf}(y))$ D = {n people}

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

If Female = false? Δ = true $\rightarrow 4^n$ models

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

FOMC Inference: Example 1

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

$\rightarrow 3^n$ models

2. $\Delta = \forall y, (\text{ParentOf}(y) \wedge \text{Female} \Rightarrow \text{MotherOf}(y))$ D = {n people}

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

If Female = false? Δ = true $\rightarrow 4^n$ models

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

1. $\Delta = \forall x,y, (\text{ParentOf}(x,y) \wedge \text{Female}(x) \Rightarrow \text{MotherOf}(x,y))$ D = {n people}

FOMC Inference: Example 1

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

Domain = {n people}

$\rightarrow 3^n$ models

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

D = {n people}

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

If Female = false? $\Delta = \text{true}$ $\rightarrow 4^n$ models

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

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

D = {n people}

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

FOMC Inference : Example 2

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

Domain = {n people}

FOMC Inference : Example 2

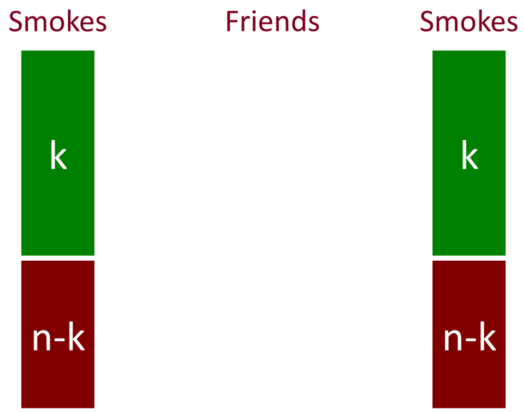
Δ = ∀x,y, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))

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
...



FOMC Inference : Example 2

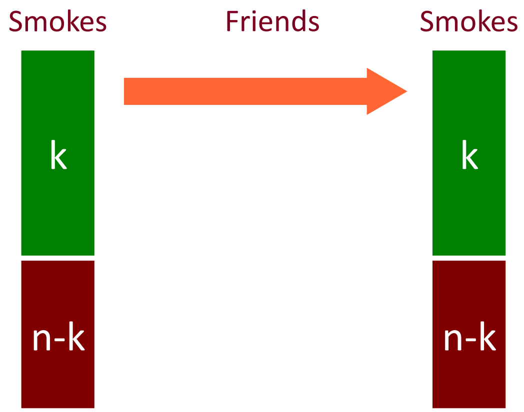
$\Delta = \forall x,y, (Smokes(x) \wedge Friends(x,y) \Rightarrow Smokes(y))$

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
...



FOMC Inference : Example 2

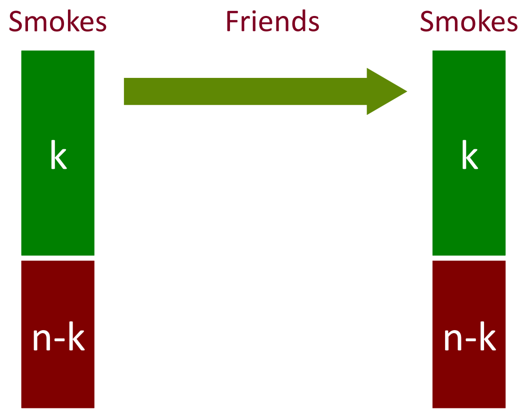
$\Delta = \forall x,y, (Smokes(x) \wedge Friends(x,y) \Rightarrow Smokes(y))$

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
...



FOMC Inference : Example 2

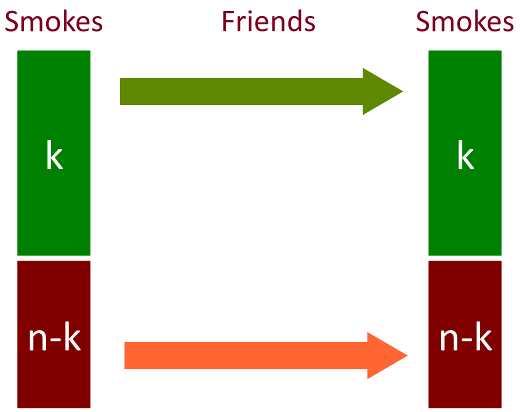
$\Delta = \forall x,y, (\text{Smokes}(x) \wedge \text{Friends}(x,y) \Rightarrow \text{Smokes}(y))$

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
...



FOMC Inference : Example 2

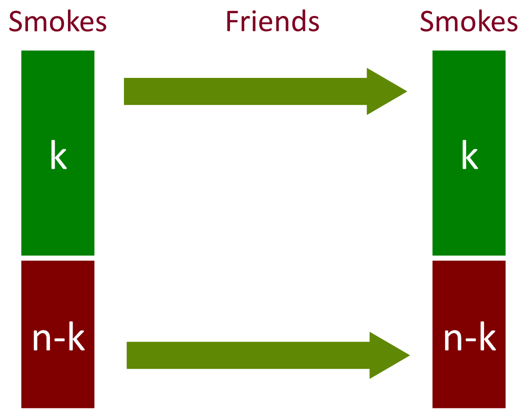
$\Delta = \forall x,y, (Smokes(x) \wedge Friends(x,y) \Rightarrow Smokes(y))$

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
...



FOMC Inference : Example 2

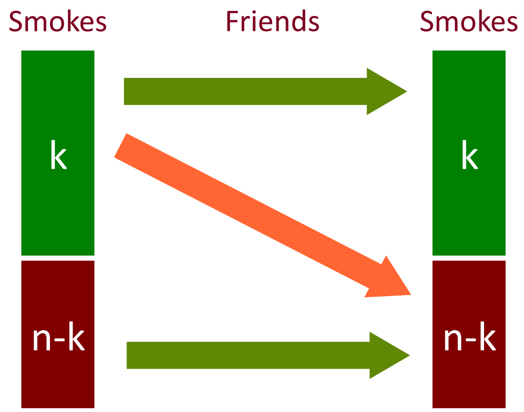
Δ = ∀x,y, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))

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
...



FOMC Inference : Example 2

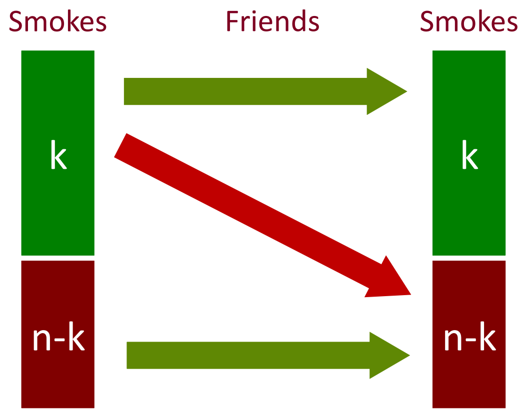
$\Delta = \forall x,y, (\text{Smokes}(x) \wedge \text{Friends}(x,y) \Rightarrow \text{Smokes}(y))$

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
...



FOMC Inference : Example 2

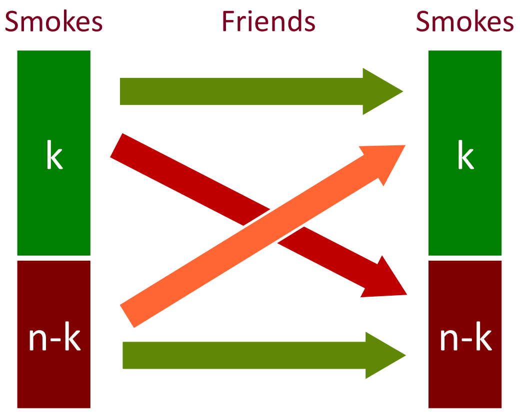
Δ = ∀x,y, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))

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
...



FOMC Inference : Example 2

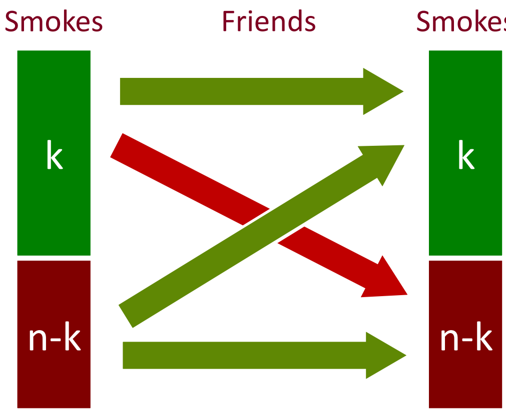
Δ = ∀x,y, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))

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
...



FOMC Inference : Example 2

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

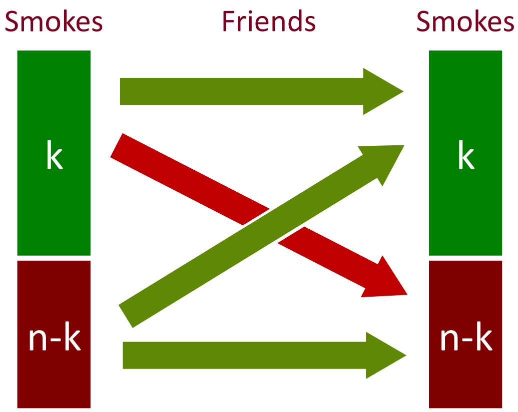
Domain = {n people}

- 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



FOMC Inference : Example 2

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

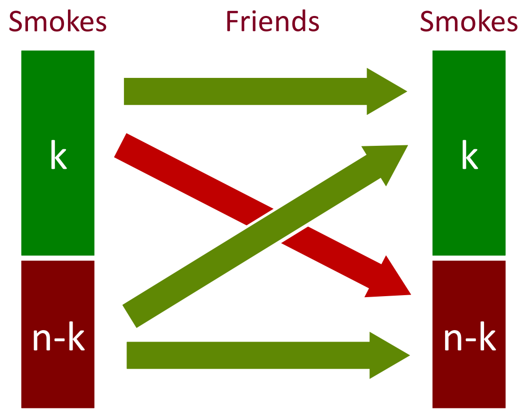
Domain = {n people}

- 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?

FOMC Inference : Example 2

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

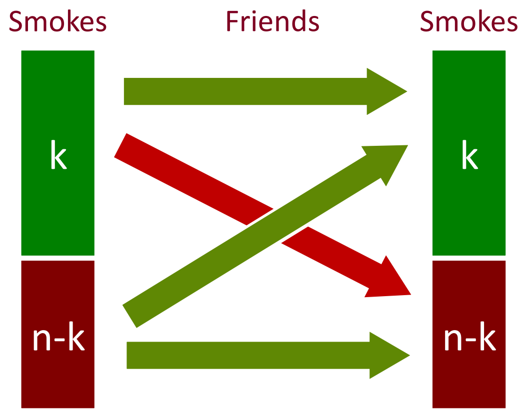
Domain = {n people}

- 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

FOMC Inference : Example 2

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

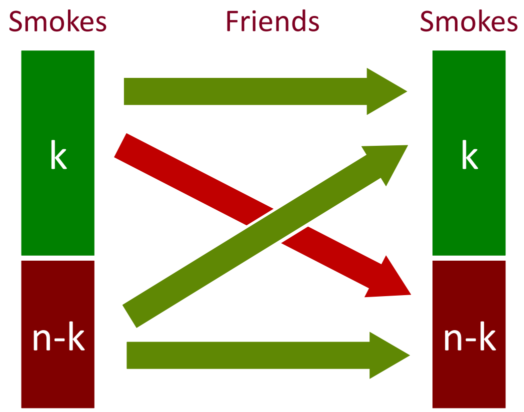
Domain = {n people}

- 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



- If we know that there are *k* smokers? $\rightarrow \binom{n}{k} 2^{n^2-k(n-k)}$ models

- In total…

FOMC Inference : Example 2

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

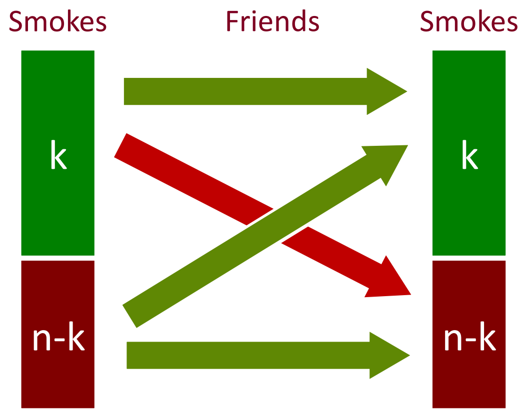
Domain = {n people}

- 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



- If we know that there are *k* smokers? $\rightarrow \binom{n}{k} 2^{n^2-k(n-k)}$ models

- In total… $\rightarrow \sum_{k=0}^{n} \binom{n}{k} 2^{n^2-k(n-k)}$ models

Overview

- Motivation and convergence of
  - The artificial intelligence story (*recap*)
  - **The machine learning story (*recap*)**
  - The probabilistic database story
  - The database theory story
- Main theoretical results and proof outlines
- Discussion and conclusions
- Dessert

Statistical Relational Models

Hard constraint

∞ Smoker(x) $\Rightarrow$ Person(x)

Soft constraint

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

- An MLN = set of constraints (w, $\Gamma(\mathbf{x})$)
- **Weight of a world** = product of w, for all rules (w, $\Gamma(\mathbf{x})$) and groundings $\Gamma(\mathbf{a})$ that hold in the world

$P_{MLN}(Q)$ = [sum of weights of models of Q] / Z

Applications: large KBs, e.g. DeepDive

Weighted Model Counting

- 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

Weighted Model Counting

- 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)

w(R)=1
w(¬R)=2
w(C)=3
w(¬C)=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

Weighted Model Counting

- 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(.)

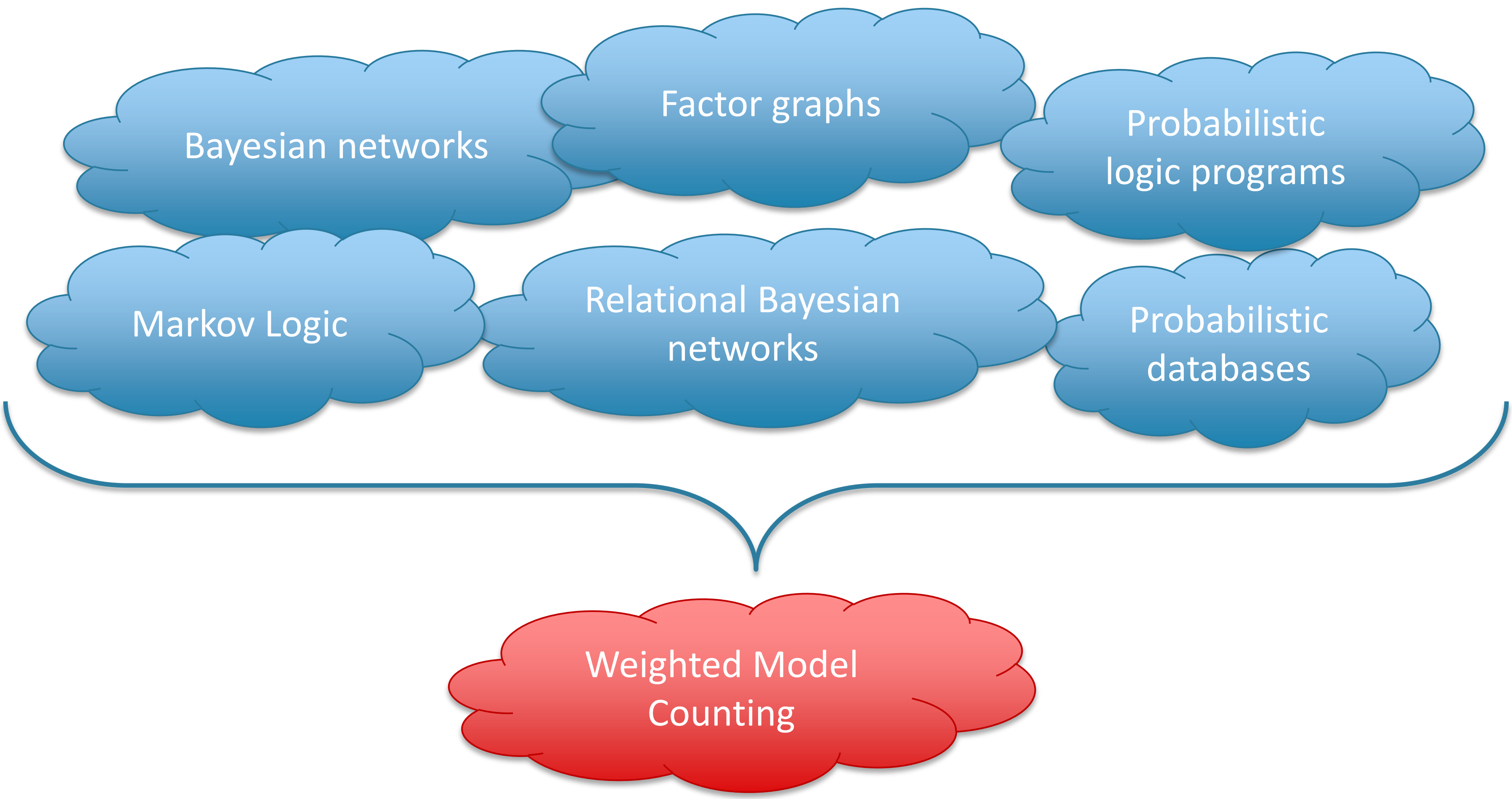
Δ = (Rain ⇒ Cloudy)

w(R)=1
w(¬R)=2
w(C)=3
w(¬C)=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**

Assembly language for probabilistic reasoning and learning



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

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

Weighted First-Order Model Counting

Model = solution to first-order logic formula Δ

$\Delta = \forall d\ (\text{Rain}(d) \Rightarrow \text{Cloudy}(d))$

Days = {Monday **Tuesday**}

$w(R)=1$
$w(\neg R)=2$
$w(C)=3$
$w(\neg C)=5$

Rain(M)	Cloudy(M)	Rain(T)	Cloudy(T)	Model?	Weight
T	T	T	T	Yes	1 * 1 * 3 * 3 = 9
T	F	T	T	No	0
F	T	T	T	Yes	2 * 1* 3 * 3 = 18
F	F	T	T	Yes	2 * 1 * 5 * 3 = 30
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 * 2 * 3 * 3 = 18
T	F	F	T	No	0
F	T	F	T	Yes	2 * 2 * 3 * 3 = 36
F	F	F	T	Yes	2 * 2 * 5 * 3 = 60
T	T	F	F	Yes	1 * 2 * 3 * 5 = 30
T	F	F	F	No	0
F	T	F	F	Yes	2 * 2 * 3 * 5 = 60
F	F	F	F	Yes	2 * 2 * 5 * 5 = 100

+

#SAT = 9

Weighted First-Order Model Counting

Model = solution to first-order logic formula Δ

Δ = ∀d (Rain(d) ⇒ Cloudy(d))

Days = {Monday **Tuesday**}

w(R)=1
w(¬R)=2
w(C)=3
w(¬C)=5

Rain(M)	Cloudy(M)	Rain(T)	Cloudy(T)	Model?	Weight
T	T	T	T	Yes	1 * 1 * 3 * 3 = 9
T	F	T	T	No	0
F	T	T	T	Yes	2 * 1* 3 * 3 = 18
F	F	T	T	Yes	2 * 1 * 5 * 3 = 30
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 * 2 * 3 * 3 = 18
T	F	F	T	No	0
F	T	F	T	Yes	2 * 2 * 3 * 3 = 36
F	F	F	T	Yes	2 * 2 * 5 * 3 = 60
T	T	F	F	Yes	1 * 2 * 3 * 5 = 30
T	F	F	F	No	0
F	T	F	F	Yes	2 * 2 * 3 * 5 = 60
F	F	F	F	Yes	2 * 2 * 5 * 5 = 100

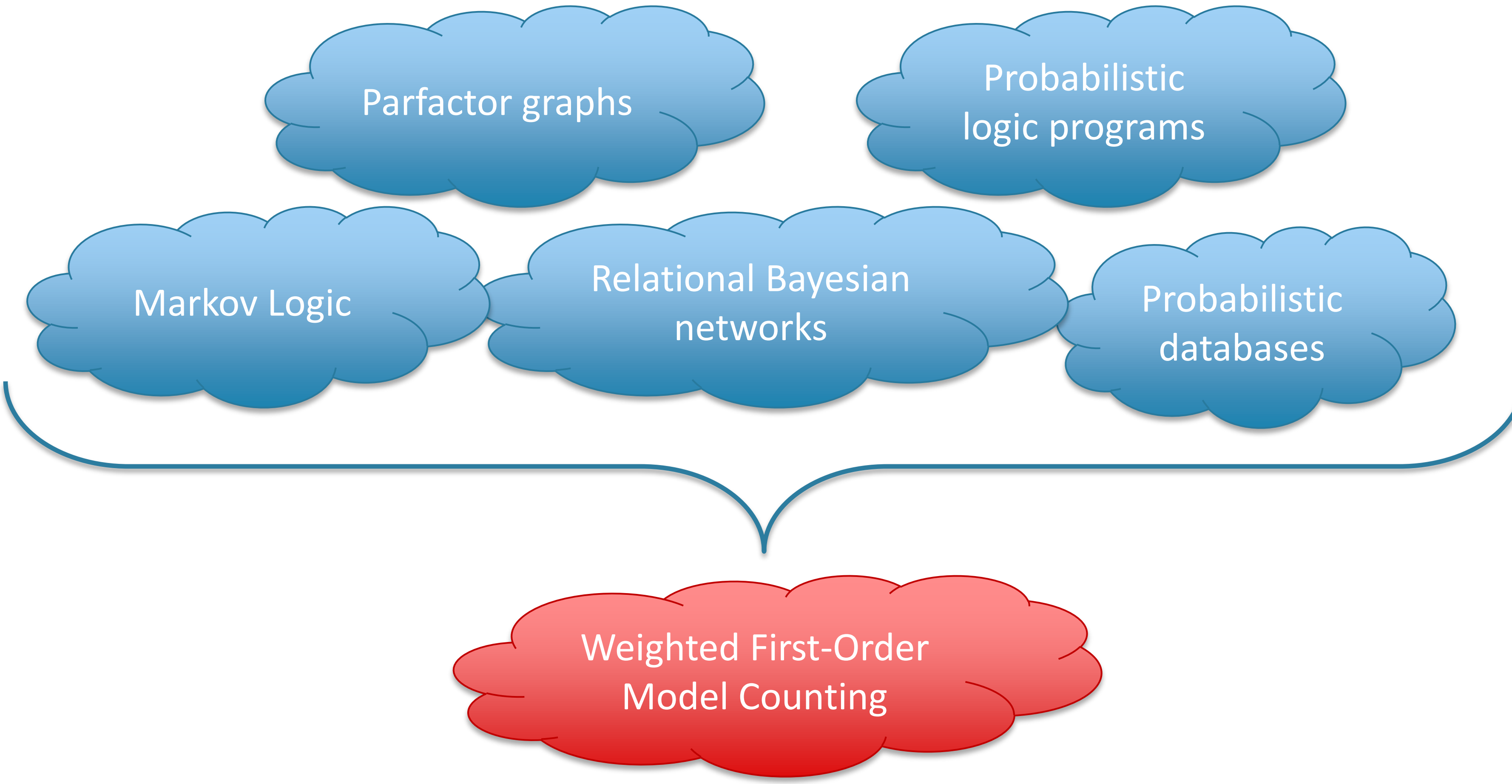
+

#SAT = 9

+

WFOMC = 361

Assembly language for high-level probabilistic reasoning and learning



[VdB et al.; IJCAI'11, PhD'13, KR'14, UAI'14]

Symmetric WFOMC

Def. A weighted vocabulary is ($\mathbf{R}$, $\mathbf{w}$), where

- $\mathbf{R} = (R_1, R_2, \ldots, R_k)$ = relational vocabulary
- $\mathbf{w} = (w_1, w_2, \ldots, w_k)$ = weights

- Fix an FO formula Q, domain of size n
- The weight of a ground tuple t in R_i is w_i

This talk: complexity of FOMC / WFOMC(Q, n)

- Data complexity: fixed Q, input n / and $\mathbf{w}$
- Combined complexity: input (Q, n) / and $\mathbf{w}$

Example

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

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

Computable in PTIME in n

Example

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

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

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

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

$\boxed{\text{Computable in PTIME in } n}$

Example

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

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

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

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

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

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

Computable in PTIME in n

Example

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

Can we compute FOMC(Q, n) in PTIME?

Open problem…

Conjecture FOMC(Q, n) not computable in PTIME in n

[Van den Broeck'2011, Gogate'2011]

From MLN to WFOMC

MLN:

∞	Smoker(x) ⇒ Person(x)
w	~Smoker(x) ∨ ~Friend(x,y) ∨ Smoker(y)

→ MLN':

∞	Smoker(x) ⇒ Person(x)
∞	R(x,y) ⇔ ~Smoker(x) ∨ ~Friend(x,y) ∨ Smoker(y)
w	R(x,y)

Theorem $P_{MLN}(Q) = P(Q \mid \text{hard constraints in } MLN')$
$= WFOMC(Q \wedge MLN') / WFOMC(MLN')$

R is a symmetric relation

Overview

- Motivation and convergence of
  - The artificial intelligence story (*recap*)
  - The machine learning story (*recap*)
  - **The probabilistic database story**
  - The database theory story
- Main theoretical results and proof outlines
- Discussion and conclusions
- Dessert

Probabilistic Databases

- Weights or probabilities given explicitly, for each tuple

- Examples: Knowledge Vault, Nell, Yago

- Dichotomy theorem:
  for any query in UCQ/FO($\exists,\wedge,\vee$) (or FO($\forall,\wedge,\vee$), asymmetric WFOMC is in PTIME or #P-hard.

Motivation 2: Probabilistic Databases

Probabilistic database D:

x	y	P
a1	b1	p_1
a1	b2	p_2
a2	b2	p_3

Motivation 2: Probabilistic Databases

Probabilistic database D:

x	y	P
a1	b1	p_1
a1	b2	p_2
a2	b2	p_3

Possible worlds semantics:

x	y
a1	b1
a1	b2
a2	b2

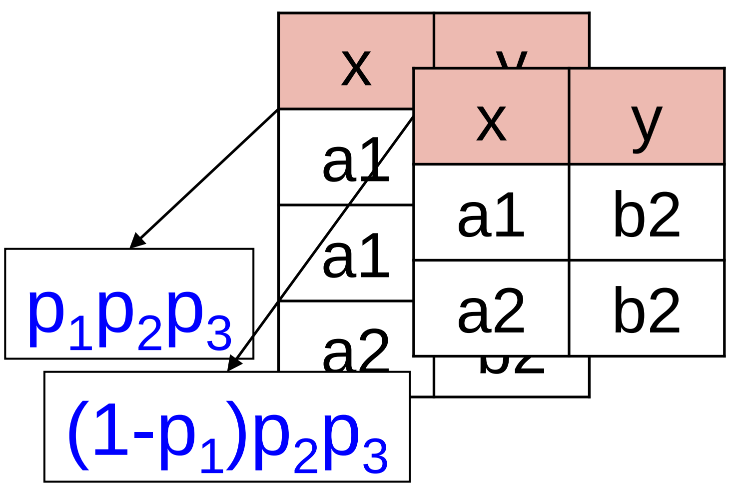
$p_1p_2p_3$

Motivation 2: Probabilistic Databases

Probabilistic database D:

x	y	P
a1	b1	p_1
a1	b2	p_2
a2	b2	p_3

Possible worlds semantics:

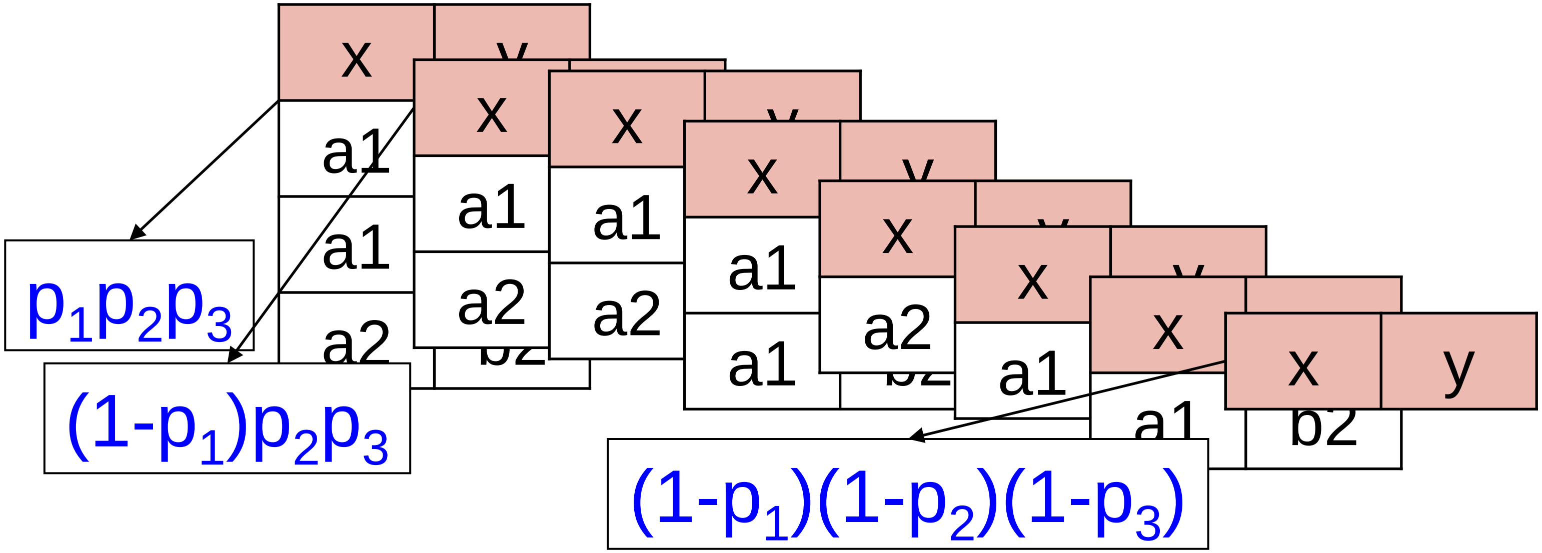


Motivation 2: Probabilistic Databases

Probabilistic database D:

x	y	P
a1	b1	p_1
a1	b2	p_2
a2	b2	p_3

Possible worlds semantics:



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

An Example

$P(Q) =$

R

x	P
a_1	p_1
a_2	p_2
a_3	p_3

S

x	y	P
a_1	b_1	q_1
a_1	b_2	q_2
a_2	b_3	q_3
a_2	b_4	q_4
a_2	b_5	q_5

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

An Example

$$P(Q) = 1-(1-q_1)*(1-q_2)$$

R

x	P
a_1	p_1
a_2	p_2
a_3	p_3

S

x	y	P
a_1	b_1	q_1
a_1	b_2	q_2
a_2	b_3	q_3
a_2	b_4	q_4
a_2	b_5	q_5

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

An Example

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

R

x	P
a_1	p_1
a_2	p_2
a_3	p_3

S

x	y	P
a_1	b_1	q_1
a_1	b_2	q_2
a_2	b_3	q_3
a_2	b_4	q_4
a_2	b_5	q_5

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

An Example

$$P(Q) = \quad p_1 * [\ 1-(1-q_1)*(1-q_2)\ ]$$

$$1-(1-q_3)*(1-q_4)*(1-q_5)$$

R

x	P
a_1	p_1
a_2	p_2
a_3	p_3

S

x	y	P
a_1	b_1	q_1
a_1	b_2	q_2
a_2	b_3	q_3
a_2	b_4	q_4
a_2	b_5	q_5

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

An Example

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

S

x	y	P
a_1	b_1	q_1
a_1	b_2	q_2
a_2	b_3	q_3
a_2	b_4	q_4
a_2	b_5	q_5

R

x	P
a_1	p_1
a_2	p_2
a_3	p_3

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

An Example

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

R

x	P
a_1	p_1
a_2	p_2
a_3	p_3

S

x	y	P
a_1	b_1	q_1
a_1	b_2	q_2
a_2	b_3	q_3
a_2	b_4	q_4
a_2	b_5	q_5

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

An Example

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

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

R

x	P
a_1	p_1
a_2	p_2
a_3	p_3

S

x	y	P
a_1	b_1	q_1
a_1	b_2	q_2
a_2	b_3	q_3
a_2	b_4	q_4
a_2	b_5	q_5

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

An Example

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

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

R

x	P	
a_1	p_1	X_1
a_2	p_2	X_2
a_3	p_3	X_3

S

x	y	P	
a_1	b_1	q_1	Y_1
a_1	b_2	q_2	Y_2
a_2	b_3	q_3	Y_3
a_2	b_4	q_4	Y_4
a_2	b_5	q_5	Y_5

Probabilistic Database Inference

Preprocess Q (omitted from this talk; see book [S.'2011])

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

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

If rules succeed, WFOMC(Q,n) in PTIME; else, #P-hard

#P-hardness no longer holds for symmetric WFOMC

Overview

- Motivation and convergence of
  - The artificial intelligence story (*recap*)
  - The machine learning story (*recap*)
  - The probabilistic database story
  - **The database theory story**
- Main theoretical results and proof outlines
- Discussion and conclusions
- Dessert

Motivation: 0/1 Laws

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

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

Theorem.
For every Q in FO, $\lim_{n \to \infty} \mu_n(Q) = 0$ or 1

Example: Q = ∀x∃y R(x,y);
FOMC(Q,n) = $(2^n-1)^n$
$\mu_n(Q) = (2^n-1)^n / 2^{n^2} \to 1$

Motivation: 0/1 Laws

In 1976 Fagin proved the 0/1 law for FO using a transfer theorem.

But is there an elementary proof? Find explicit formula for $\mu_n(Q)$, then compute the limit. [Fagin communicated to us that he tried this first]

Overview

- Motivation and convergence of
  - The artificial intelligence story (*recap*)
  - The machine learning story (*recap*)
  - The probabilistic database story
  - The database theory story
- **Main theoretical results and proof outlines**
- Discussion and conclusions
- Dessert

Class FO^2

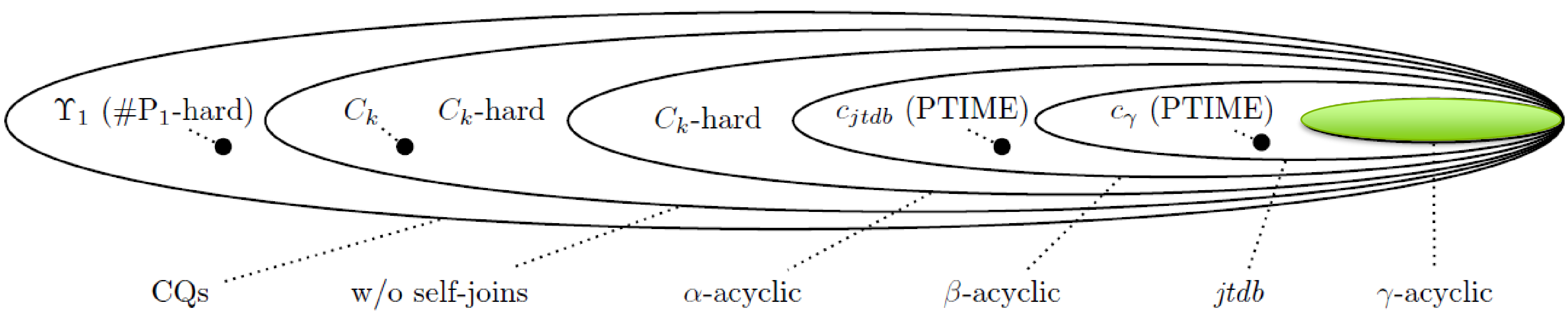
- FO^2 = FO restricted to two variables
- Intuition: SQL queries that have a plan where all temp tables have arity ≤ 2
- "The graph has a path of length 10":

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

Main Positive Results

Data complexity:

- for any formula Q in FO^2, WFOMC(Q, n) is in PTIME [see NIPS'11, KR'13]
- for any γ-acyclic conjunctive query w/o self-joins Q, WFOMC(Q, n) is in PTIME



Main Negative Results

Data complexity:

- There exists an FO formula Q s.t. symmetric FOMC(Q, n) is $\#P_1$ hard
- There exists Q in FO^3 s.t. FOMC(Q, n) is $\#P_1$ hard
- There exists a conjunctive query Q s.t. symmetric WFOMC(Q, n) is $\#P_1$ hard
- There exists a positive clause Q w.o. '=' s.t. symmetric WFOMC(Q, n) is $\#P_1$ hard

Combined complexity:

- FOMC(Q, n) is #P-hard

Review: $\#P_1$

- $\#P_1$ = class of functions in #P over a unary input alphabet
- Valiant 1979: there exists $\#P_1$ complete problems
- Bertoni, Goldwurm, Sabatini 1988:
  counting strings of a given length in some CFG is $\#P_1$ complete
- Goldberg: “no natural combinatorial problems known to be $\#P_1$ complete”

Main Result 1

Theorem 1. There exists an FO^3 sentence Q s.t. $FOMC(Q,n)$ is $\#P_1$-hard

Proof

- Step 1. Construct a Turing Machine U s.t.
  - U is in $\#P_1$ and runs in linear time in n
  - U computes a $\#P_1$ –hard function
- Step 2. Construct an FO^3 sentence Q s.t. $FOMC(Q,n) / n! = U(n)$

Main Result 2

Theorem 2 There exists a Conjunctive Query Q s.t. WFOMC(Q,n) is $\#P_1$-hard

- Note: the decision problem is trivial (Q has a model iff $n > 0$)
- *Unweighted* Model Counting for CQ: open

Proof Start with a formula Q that is $\#P_1$-hard for FOMC, and transform it to a CQ in five steps (next)

Start: Q s.t. FOMC(Q, n) is $\#P_1$-hard

Step 1: Remove ∃

Rewrite $Q = \forall x \exists y\ \psi(x,y)$
to $Q' = \forall x\ \forall y\ (\neg\psi(x,y) \lor \neg A(x))$

where A = new symbol with weight $w = -1$

Claim: WFOMC(Q, n) = WFOMC(Q', n)
Proof Consider a model for Q', and a constant $x=a$

- If $\exists b\ \psi(a,b)$, then $A(a)$=false; contributes $w=1$
- Otherwise, $A(a)$ can be either true or false, contributing either $w=1$ or $w=-1$, and $1 - 1 = 0$.

$Q = \forall^* \ldots$, WFOMC(Q, n) is $\#P_1$-hard

Start: Q s.t. FOMC(Q, n) is $\#P_1$-hard

Step 2: Remove Negation

- Transform Q to Q' w/o negation s.t. WFOMC(Q, n) = WFOMC(Q', n)

- Similarly to step 1 and omitted

Q = ∀*[positive], WFOMC(Q, n) is $\#P_1$-hard

Start: Q s.t. FOMC(Q, n) is $\#P_1$-hard

Step 3: Remove "="

Rewrite Q to Q' as follows:

- Add new binary symbol E with weight w
- Define: $Q' = Q[\,E\ /\ \text{"="}\,] \wedge (\forall x\, E(x,x))$

Claim: WFOMC(Q,n) computable using oracle for WFOMC(Q', n)
(coefficient of w^n in polynomial WFOMC(Q', n)

$Q = \forall^*$[positive, w/o =], WFOMC(Q, n) is $\#P_1$-hard

Start: Q s.t. FOMC(Q, n) is #P_1-hard

Step 4: To UCQ

- Write $Q = \forall^* (C_1 \wedge C_2 \wedge \ldots)$ where each C_i is a positive clause
- The dual $Q' = \exists^* (C_1' \vee C_2' \vee \ldots)$ is a UCQ

UCQ Q, WFOMC(Q, n) is #P_1-hard

Start: Q s.t. FOMC(Q, n) is $\#P_1$-hard

Step 5: from UCQ to CQ

- UCQ: $Q = C_1 \vee C_2 \vee \ldots \vee C_k$

- $P(Q) = \ldots. + (-1)^S P(\wedge_{i \in S} C_i) + \ldots.$

- 2^k-1 CQs $P(Q_1), P(Q_2), \ldots P(Q_{2\text{^}k-1})$

- 1 CQ (using fresh copies of symbols): $P(Q'_1 Q'_2 \ldots Q'_{2\text{^}k-1}) = P(Q'_1) P(Q'_2) \ldots P(Q'_{2\text{^}k-1})$

CQ Q' $(= Q'_1 Q'_2 \ldots Q'_{2\text{^}k-1})$ WFOMC(Q', n) is $\#P_1$-hard

Overview

- Motivation and convergence of
  - The artificial intelligence story (*recap*)
  - The machine learning story (*recap*)
  - The probabilistic database story
  - The database theory story
- Main theoretical results and proof outlines
- **Discussion and conclusions**
- Dessert

Motivation: 0/1 Laws

In 1976 Fagin proved the 0/1 law for FO using a transfer theorem.

But is there an elementary proof? Find explicit formula for $\mu_n(Q)$, then compute the limit. [Fagin communicated to us that he tried this first]

Motivation: 0/1 Laws

In 1976 Fagin proved the 0/1 law for FO using a transfer theorem.

But is there an elementary proof? Find explicit formula for $\mu_n(Q)$, then compute the limit. [Fagin communicated to us that he tried this first]

A: unlikely when FOMC(Q,n) is $\#P_1$-hard

Discussion

Fagin (1974) restated:

1. NP = ∃SO
   (Fagin's classical characterization of NP)
2. $NP_1 = \{Spec(\Phi) \mid \Phi \in FO\}$ in tally notation
   (less well known!)

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

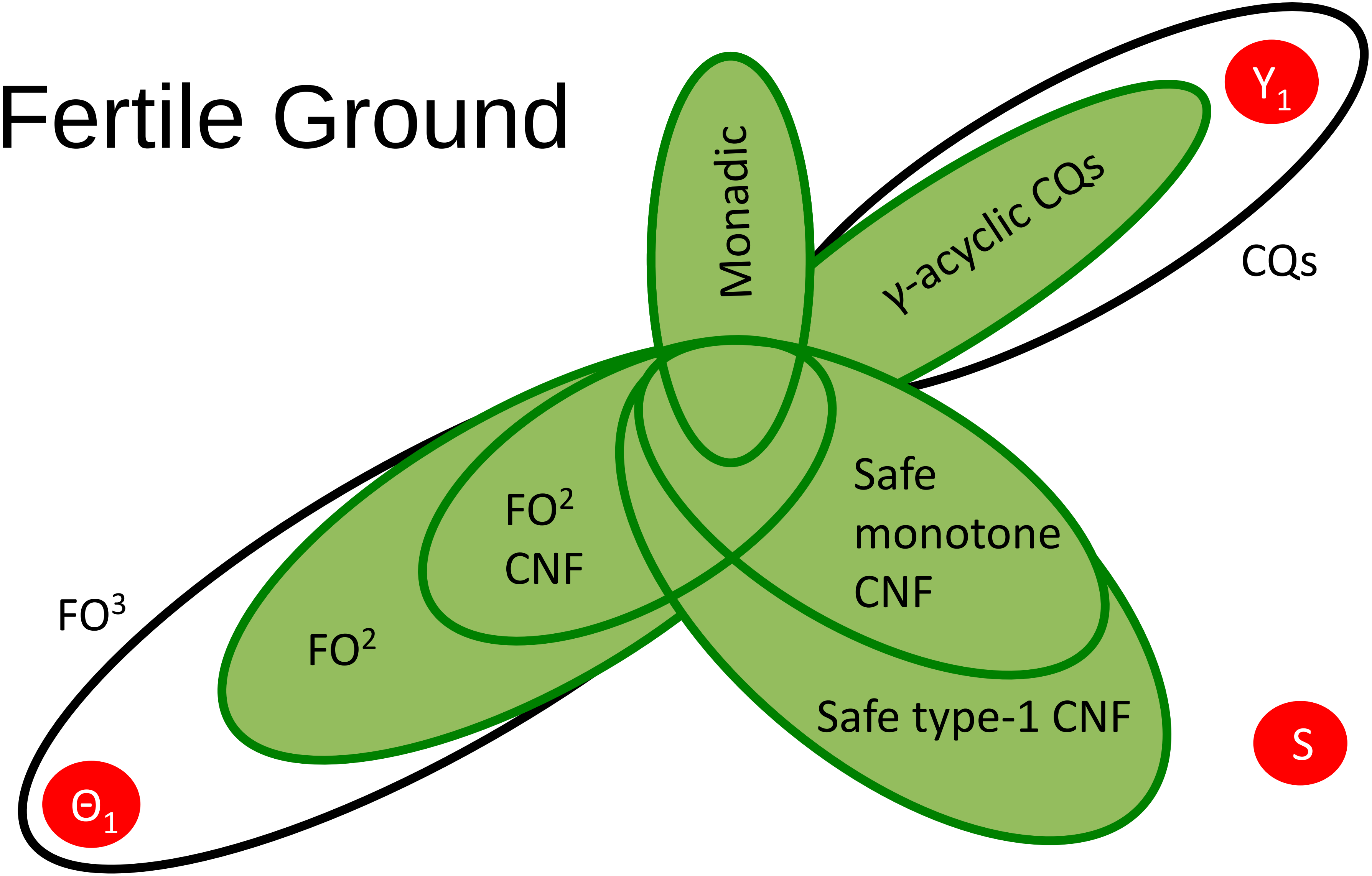
Discussion

- Convergence of AI/ML/DB/theory
- First-order model counting is a basic problem that touches all these areas
- Under-investigated
- Hardness proofs are more difficult than for #P

Open problems:

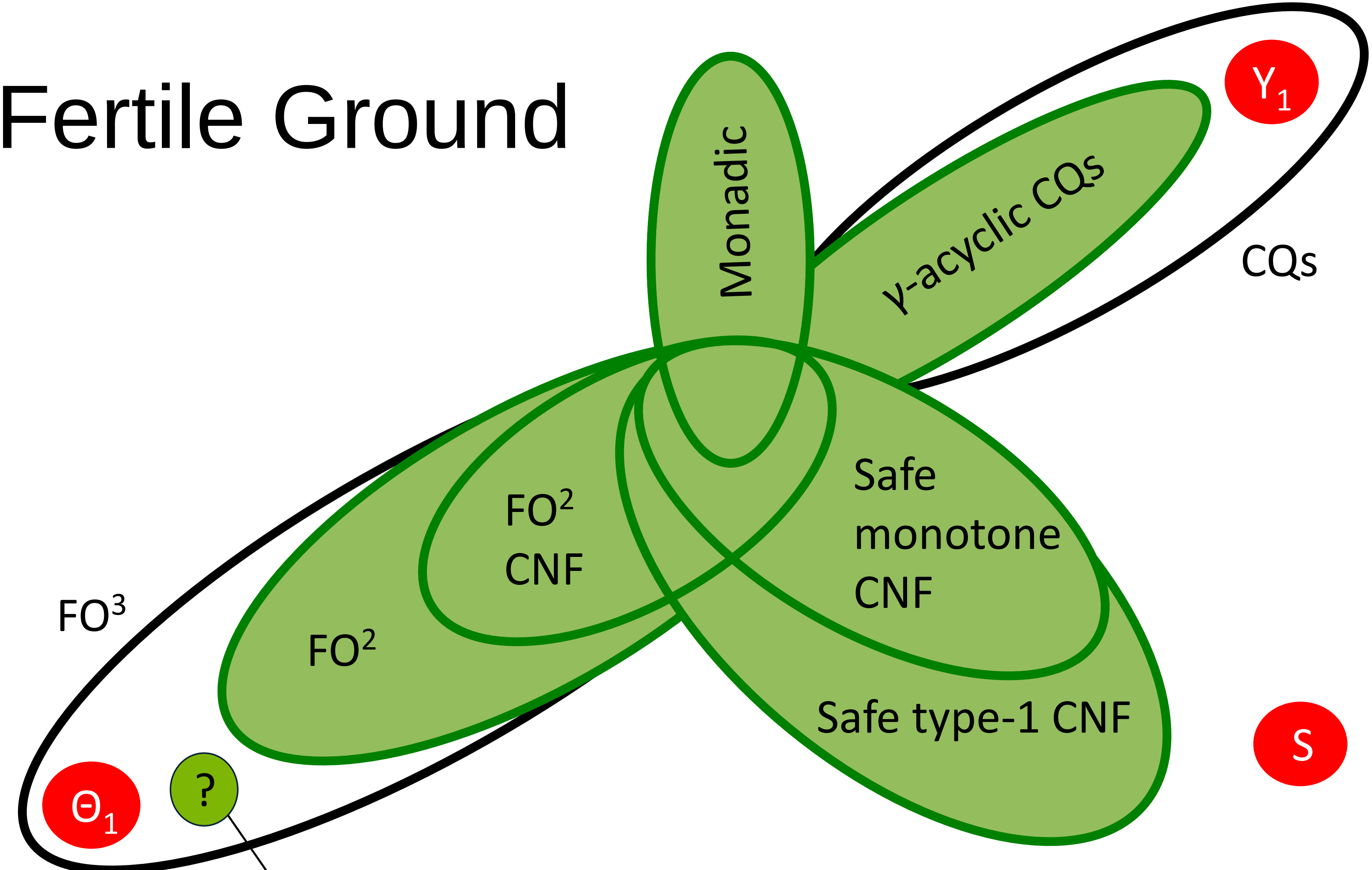
- New algorithm for symmetric model counting
- New hardness reduction techniques

Fertile Ground



[VdB; NIPS’11], [VdB et al.; KR’14], [Gribkoff, VdB, Suciu; UAI’15], [Beame, VdB, Gribkoff, Suciu; PODS’15], etc.

Fertile Ground



$\Delta = \forall x,y,z,\ \mathrm{Friends}(x,y) \wedge \mathrm{Friends}(y,z) \Rightarrow \mathrm{Friends}(x,z)$

[VdB; NIPS'11], [VdB et al.; KR'14], [Gribkoff, VdB, Suciu; UAI'15], [Beame, VdB, Gribkoff, Suciu; PODS'15], etc.

Overview

- Motivation and convergence of
  - The artificial intelligence story (*recap*)
  - The machine learning story (*recap*)
  - The probabilistic database story
  - The database theory story
- Main theoretical results and proof outlines
- Discussion and conclusions
- **Dessert**

The Decision Problem

- Counting problem
  "count the number of XXX s.t…"

- Decision problem
  "does there exists an XXX s.t. …?"

- #3SAT and 3SAT:
  - counting is #P-complete, decision is NP-hard
- #2SAT and 2SAT:
  - counting is #P-hard, decision is in PTIME

Counting/Decision Problems for FO

- **Counting**: given Q,n, count the number of models of Q over a domain of size n
- **Decision**: given Q,n, does there exists a model of Q over a domain of size n?
- **Data complexity**: fix Q, input = n
- **Combined complexity**: input = Q, n

The Spectrum

Definition. [Scholz 1952]
Spec(Q)= {n | Q has a model over domain [n]}

Example: Q says "(D, +, *, 0, 1) is a field":

Spec(Q) = $\{p^k \mid p \text{ prime}, k \geq 1\}$

Spectra studied intensively for over 50 years

The FO decision problem is precisely spectrum membership

The Data Complexity

Suppose n is given in binary representation:

- Jones&Selman'72: spectra = NETIME

$$\text{NETIME} = \bigcup_{c \geq 0} \text{NTIME}(2^{cn}) \qquad \text{NEXPTIME} = \bigcup_{c \geq 0} \text{NTIME}(2^{c^n})$$

Suppose n is given in unary representation:

- Fagin'74: spectra = NP_1

Combined Complexity

Consider the combined complexity for FO^2
"given Q, n, check if $n \in Spec(Q)$"

We prove its complexity:

- NP-complete for FO^2,
- PSPACE-complete for FO

Thanks!