On First-Order Knowledge Compilation

Guy Van den Broeck

UCLA

Beyond NP Workshop

Feb 12, 2016

Overview

1. Why first-order model counting?
2. Why first-order model counters?
3. What first-order circuit languages?
4. How first-order knowledge compilation?
5. Perspectives …

Why do we need
first-order model ***counting****?*

Uncertainty in AI

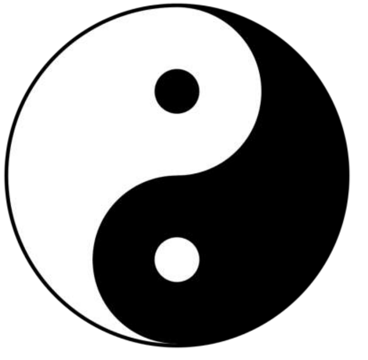
Probability Distribution

=

Qualitative

+

Quantitative



Probabilistic Graphical Models

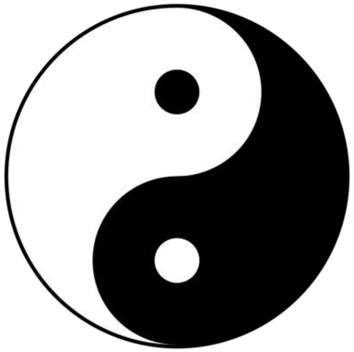
Probability Distribution

=

Graph Structure

+

Parameterization



Probabilistic Graphical Models

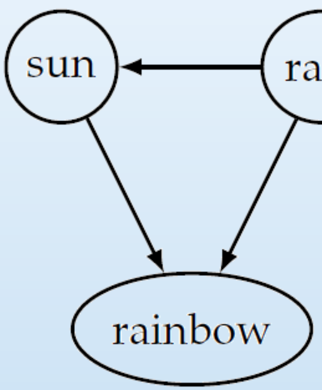
Probability Distribution

=

Graph Structure

+

Parameterization



+

rain	Pr(sun | rain)
T	0.1
F	0.6

rain	sun	Pr(rainbow | rain, sun)
T	T	0.9
T	F	0.05
F	T	0.05
F	F	0

Pr(rain)
0.2

Weighted Model Counting

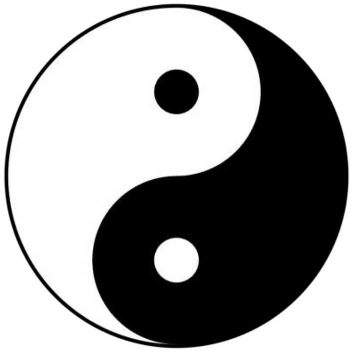
Probability Distribution

=

SAT Formula

+

Weights



[Chavira 2008, Sang 2005]

Weighted Model Counting

Probability Distribution

=

SAT Formula

+

Weights

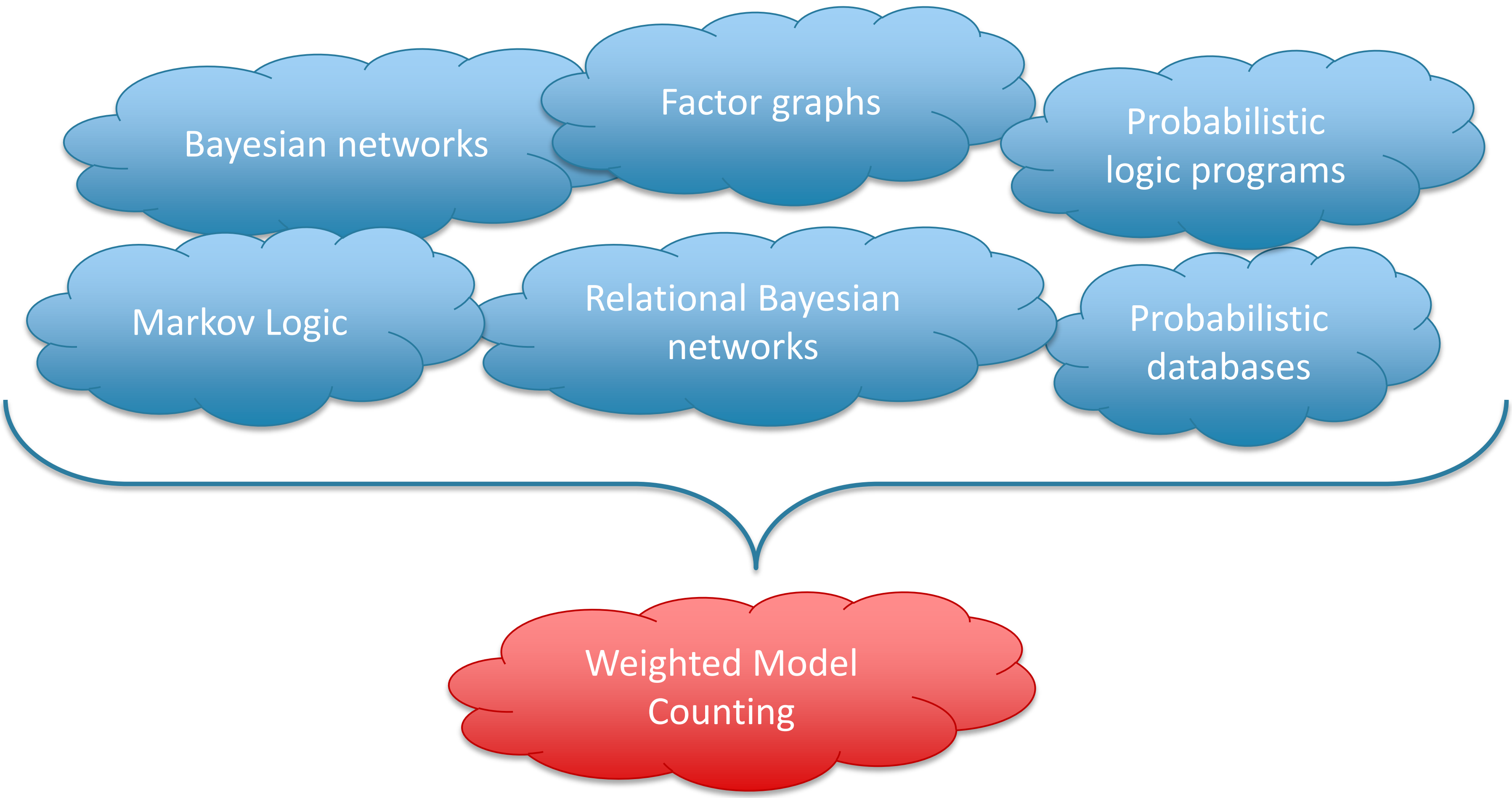
Rain ⇒ Cloudy
Sun ∧ Rain ⇒ Rainbow

+

w(Rain)=1
w(¬Rain)=2
w(Cloudy)=3
w(¬Cloudy)=5
…

[Chavira 2008, Sang 2005]

Beyond NP Pipeline for #P



[Chavira 2006, Chavira 2008, Sang 2005, Fierens 2015]

Generalized Perspective

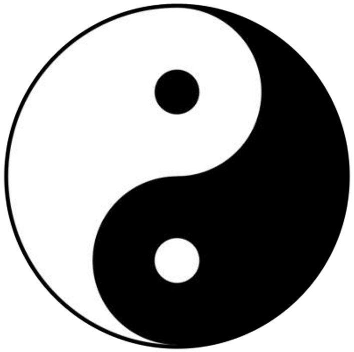
Probability Distribution

=

Logic

+

Weights



Generalized Perspective

Probability Distribution

=

Logic

+

Weights

Logical Syntax
Model-theoretic
Semantics

+

Weight function $w(.)$

Factorized
$Pr(\text{model}) \propto \prod_i w(x_i)$

First-Order Model Counting

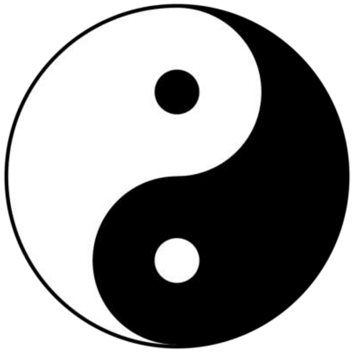
Probability Distribution

=

First-Order Logic

+

Weights



[Van den Broeck 2011, 2013, Gogate 2011]

First-Order Model Counting

Probability Distribution

=

First-Order Logic

+

Weights

Smokes(x) ∧ Friends(x,y)
⇒ Smokes(y)

+

w(Smokes(a))=1
w(¬Smokes(a))=2
w(Smokes(b))=1
w(¬Smokes(b))=2
w(Friends(a,b))=3
w(¬Friends(a,b))=5
…

[Van den Broeck 2011, 2013, Gogate 2011]

Probabilistic Programming

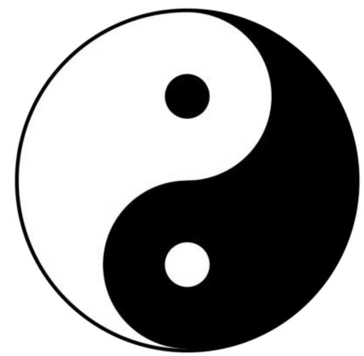
Probability Distribution

=

Logic Programs

+

Weights



[Fierens 2015]

Probabilistic Programming

Probability Distribution

=

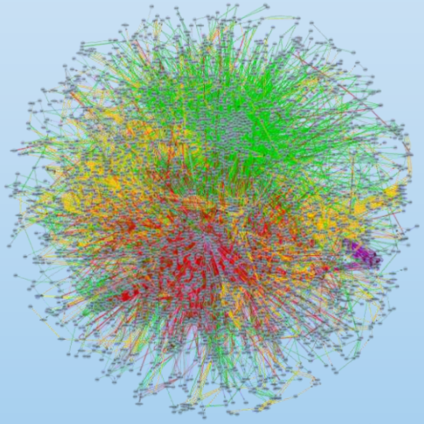
Logic Programs

+

Weights

```
path(X,Y) :-
        edge(X,Y).
path(X,Y) :-
        edge(X,Z), path(Z,Y).
```

+



[Fierens 2015]

Weighted Model Integration

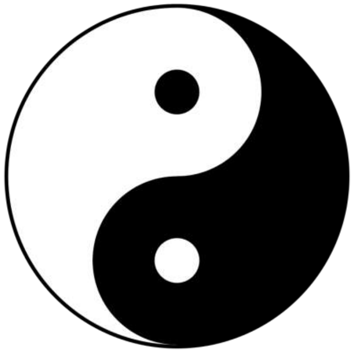
Probability Distribution

=

SMT(LRA)

+

Weights



[Belle 2015]

Weighted Model Integration

Probability Distribution

=

SMT(LRA)

+

Weights

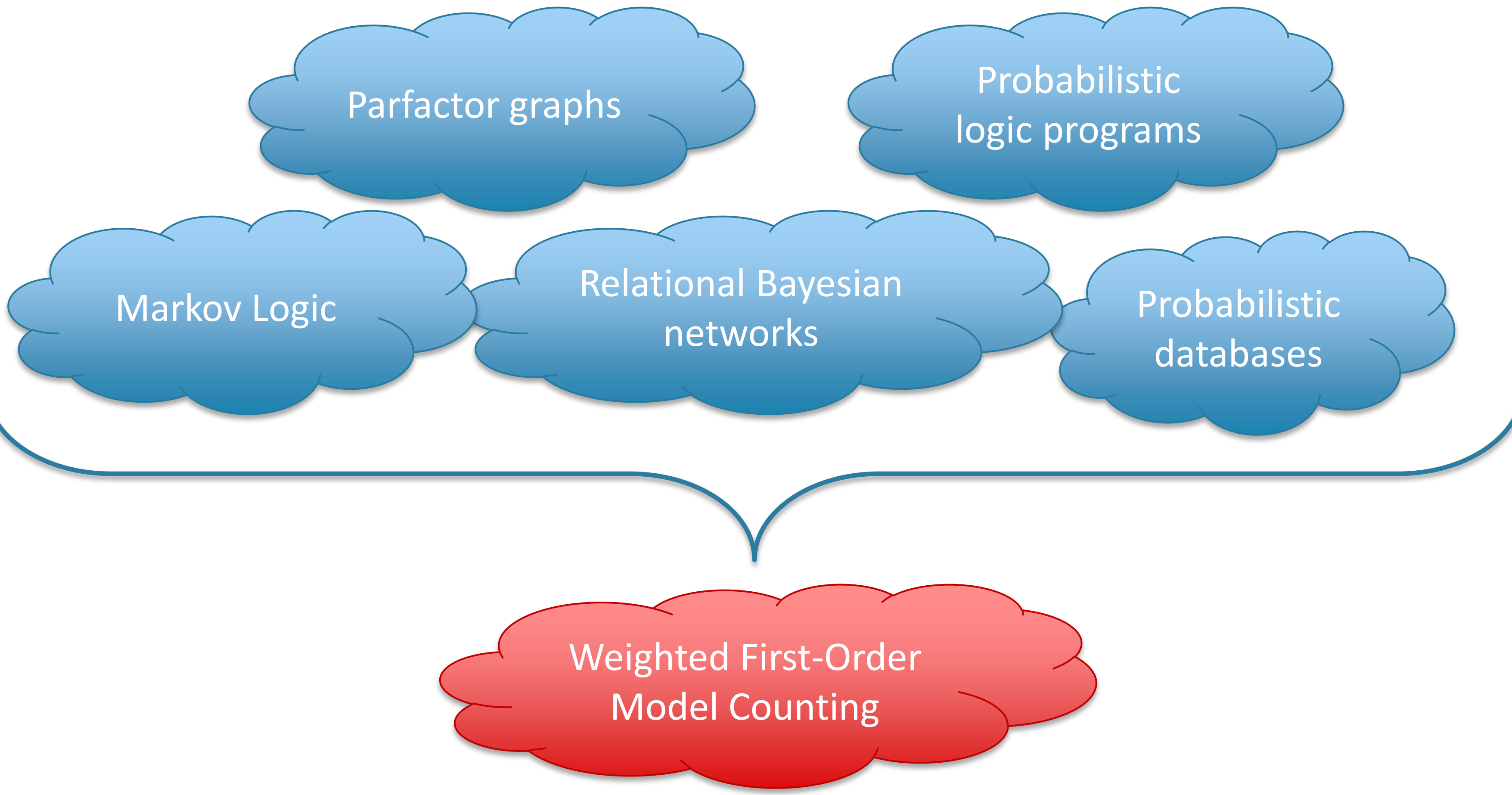
$0 \leq height \leq 200$
$0 \leq weight \leq 200$
$0 \leq age \leq 100$
$age < 1 \Rightarrow$
$height+weight \leq 90$

+

w(height))=height-10
w(¬height)=3*height^2
w(¬weight)=5
...

[Belle 2015]

Beyond NP Pipeline for #P/#P_1



[Van den Broeck 2011, 2013, Gogate 2011, Gribkoff 2014]

First-Order Model Counting

Model = solution to first-order logic formula Δ

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

Days = {Monday}

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

+

FOMC = 3

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

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

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

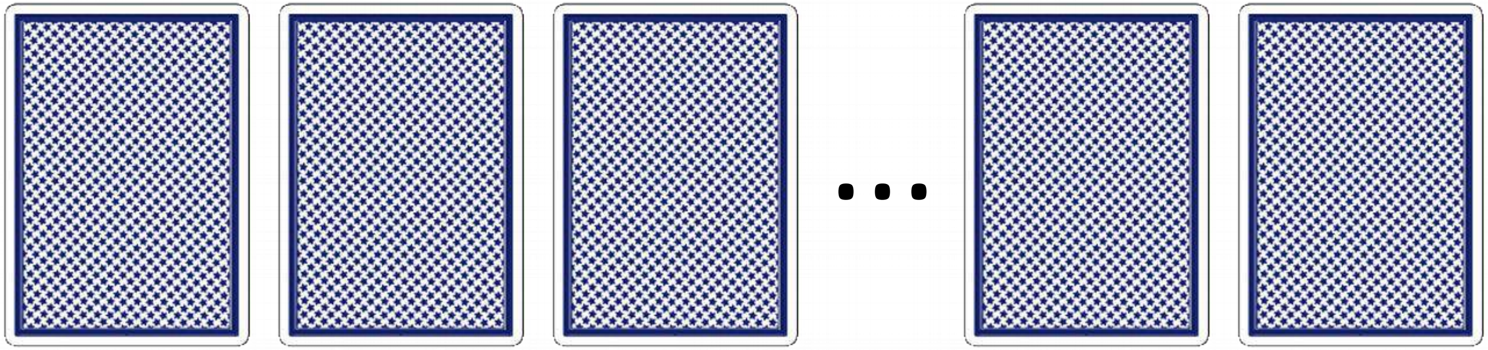
Rain(M)	Cloudy(M)	Rain(T)	Cloudy(T)	Model?	Weight
T	T	T	T	Yes	1 * 1 * 3 * 3 = 9
T	F	T	T	No	0
F	T	T	T	Yes	2 * 1* 3 * 3 = 18
F	F	T	T	Yes	2 * 1 * 5 * 3 = 30
T	T	T	F	No	0
T	F	T	F	No	0
F	T	T	F	No	0
F	F	T	F	No	0
T	T	F	T	Yes	1 * 2 * 3 * 3 = 18
T	F	F	T	No	0
F	T	F	T	Yes	2 * 2 * 3 * 3 = 36
F	F	F	T	Yes	2 * 2 *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

Why do we need
first-order model ***counters****?*

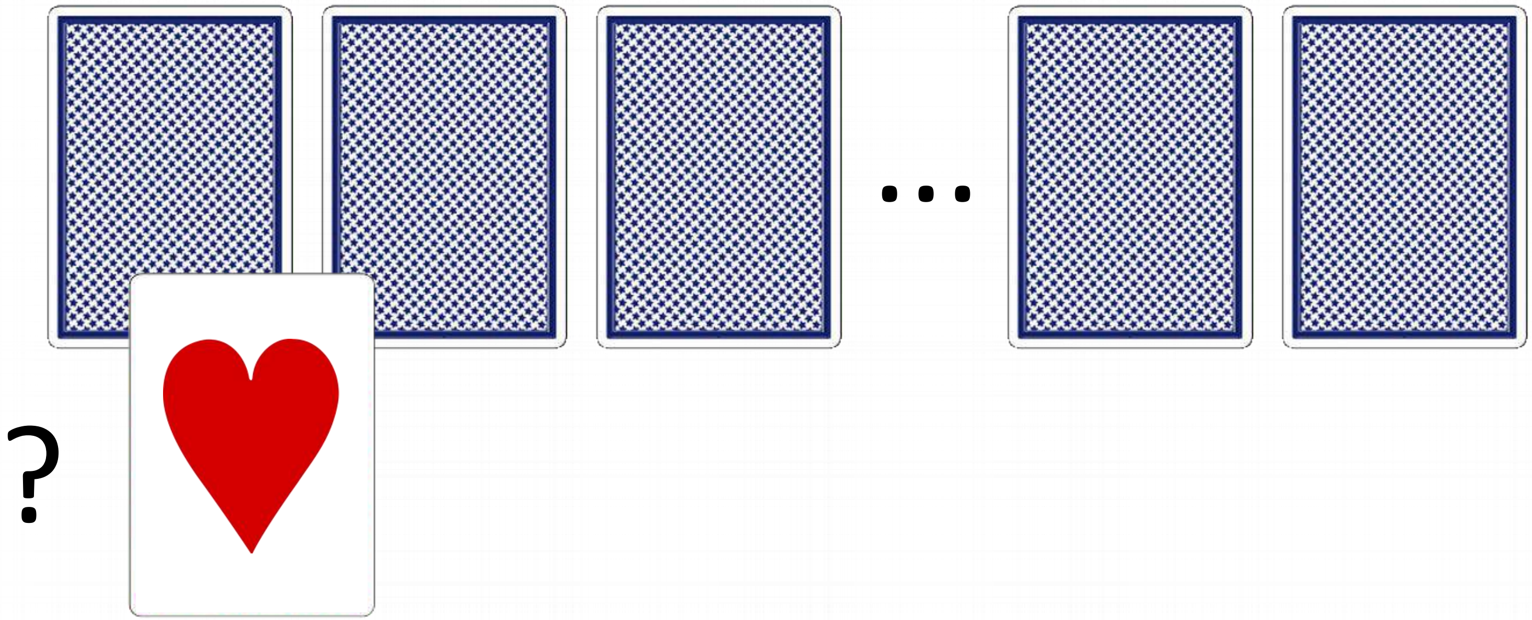
A Simple Reasoning Problem



- 52 playing cards
- Let us ask some simple questions

[Van den Broeck 2015]

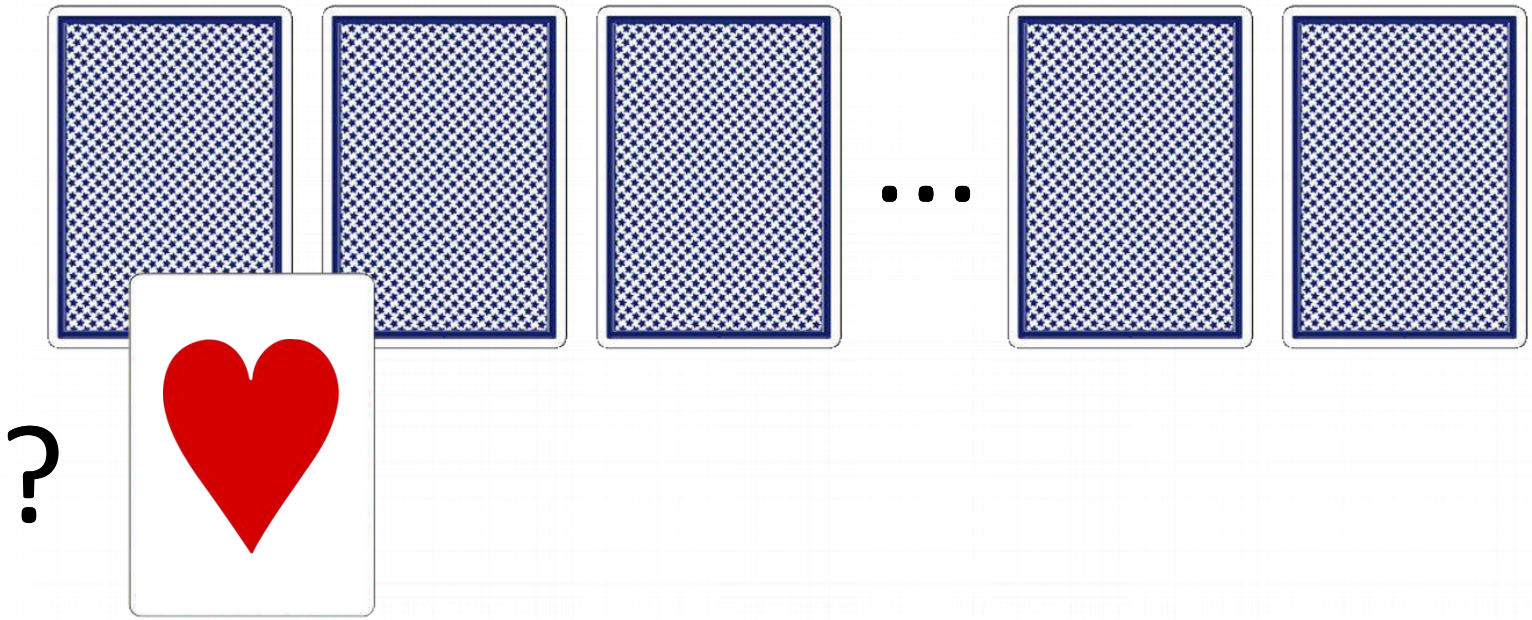
A Simple Reasoning Problem



Probability that Card1 is Hearts?

[Van den Broeck 2015]

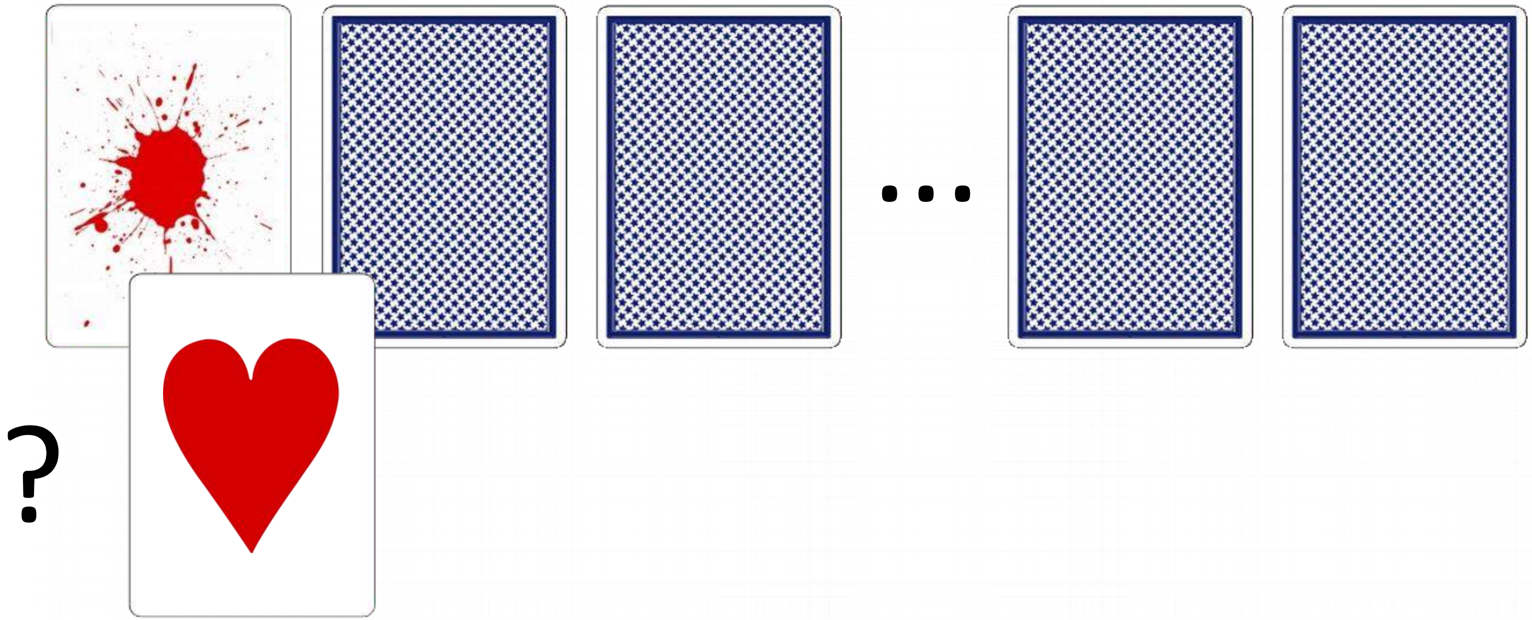
A Simple Reasoning Problem



Probability that Card1 is Hearts? 1/4

[Van den Broeck 2015]

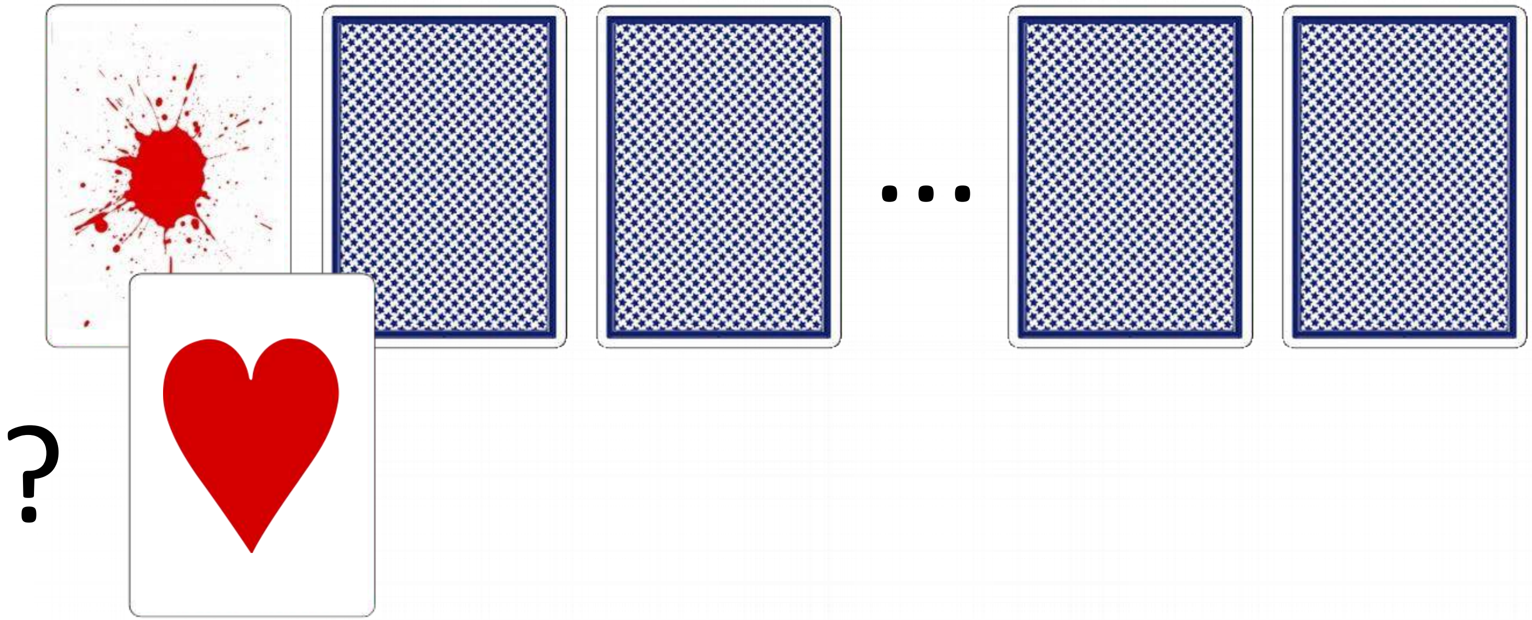
A Simple Reasoning Problem



Probability that Card1 is Hearts given that Card1 is red?

[Van den Broeck 2015]

A Simple Reasoning Problem

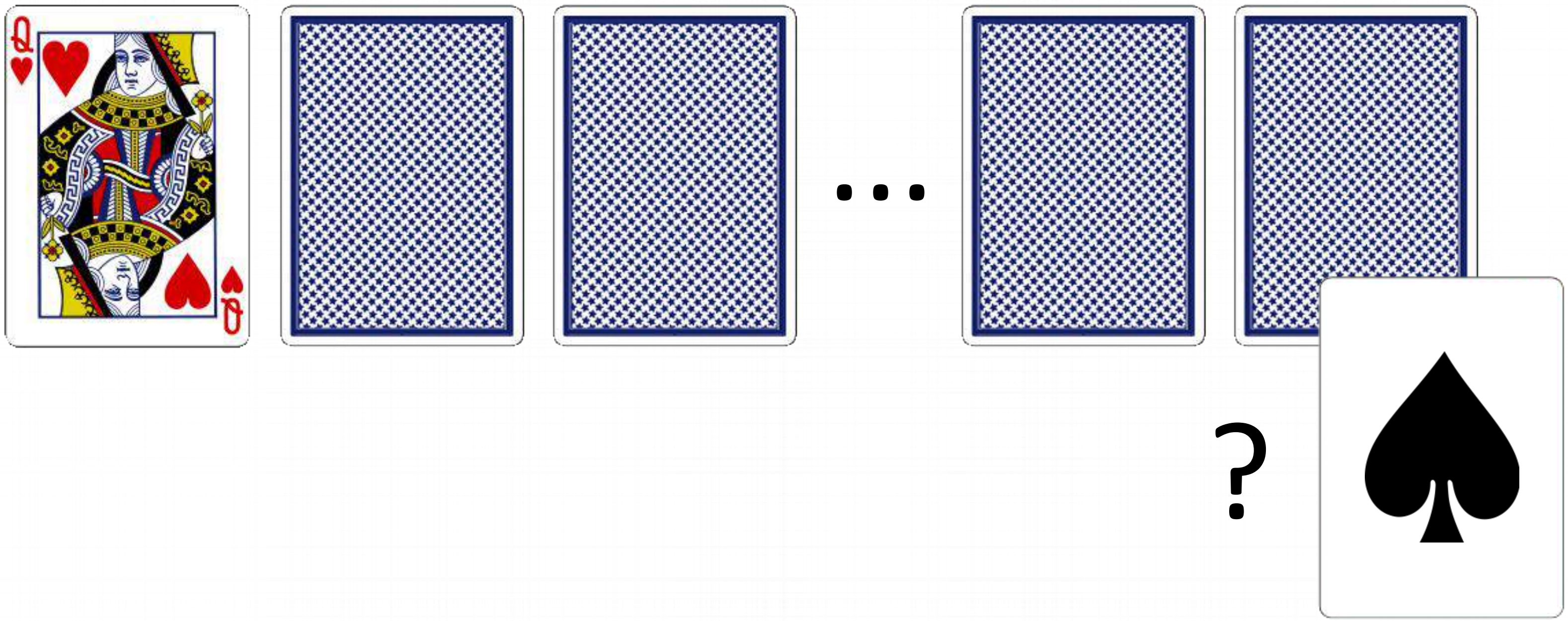


Probability that Card1 is Hearts given that Card1 is red?

1/2

[Van den Broeck 2015]

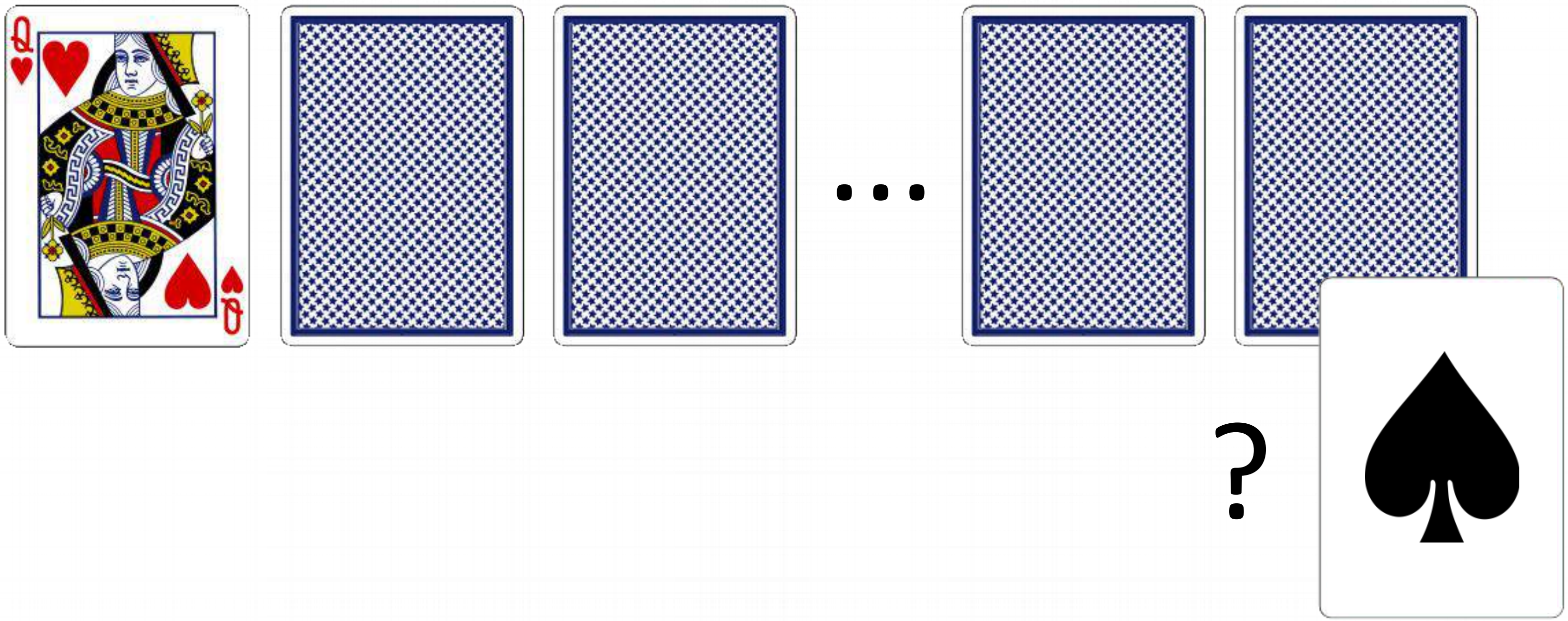
A Simple Reasoning Problem



Probability that Card52 is Spades given that Card1 is QH?

[Van den Broeck 2015]

A Simple Reasoning Problem

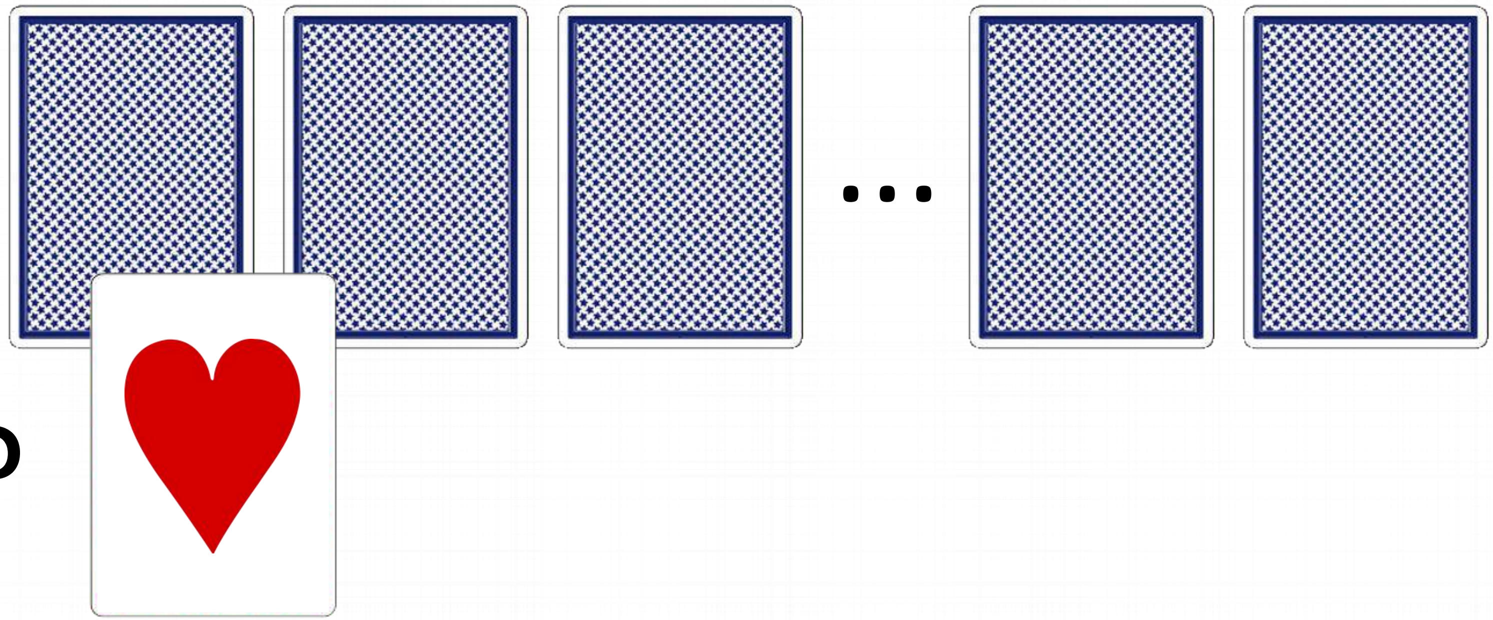


Probability that Card52 is Spades given that Card1 is QH?

13/51

[Van den Broeck 2015]

A Simple Reasoning Problem

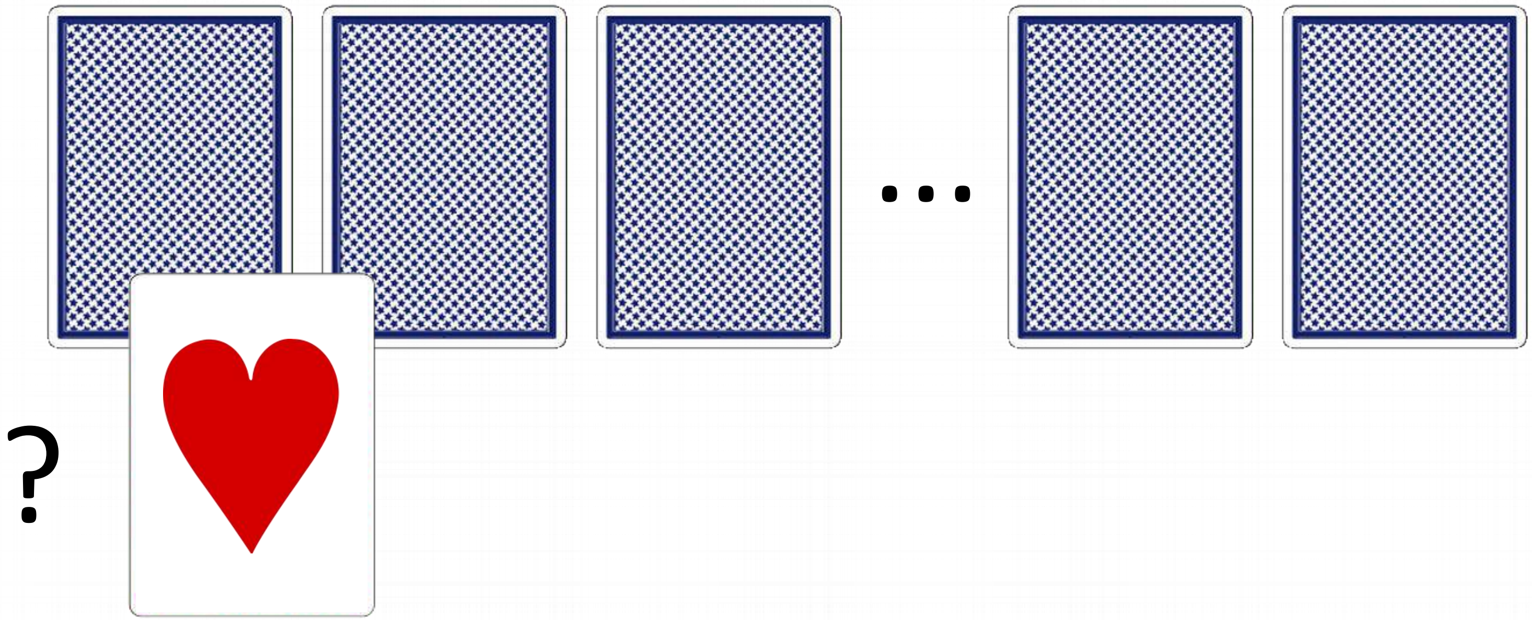


?

Probability that Card1 is Hearts?

[Van den Broeck 2015]

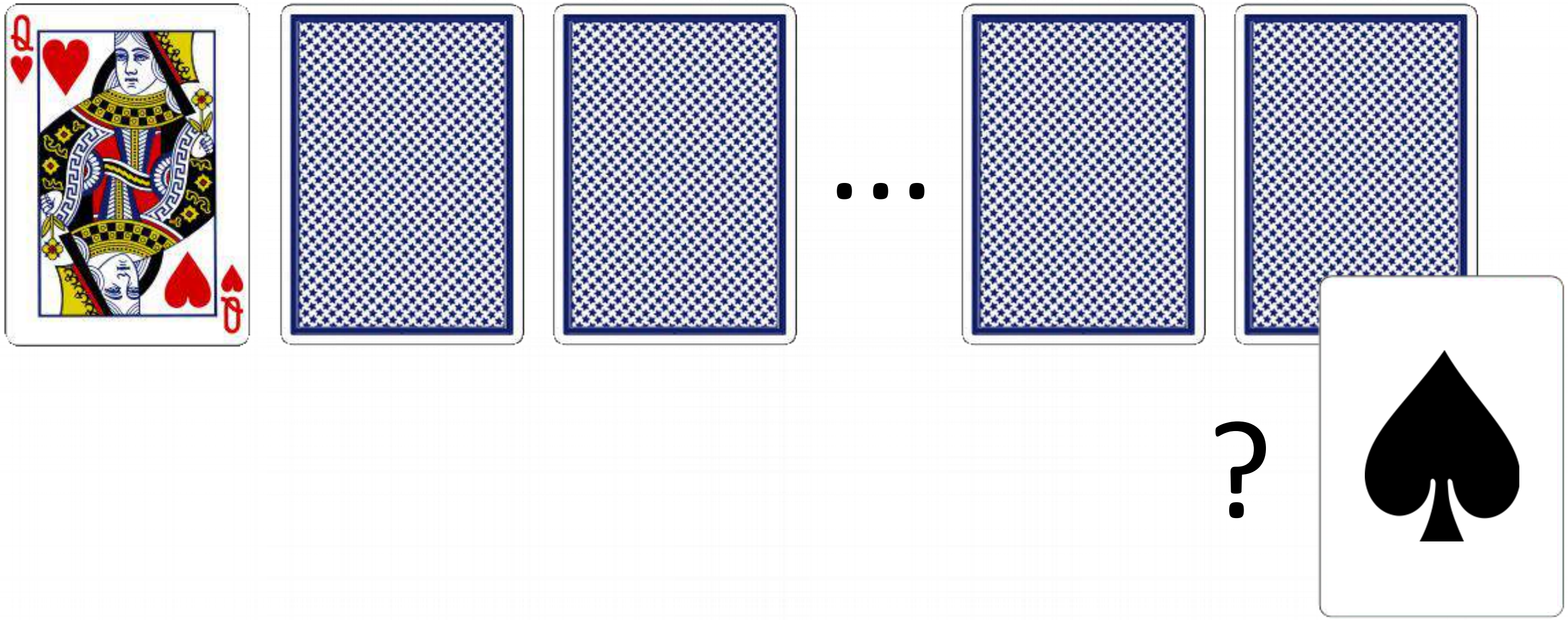
A Simple Reasoning Problem



Probability that Card1 is Hearts? 1/4

[Van den Broeck 2015]

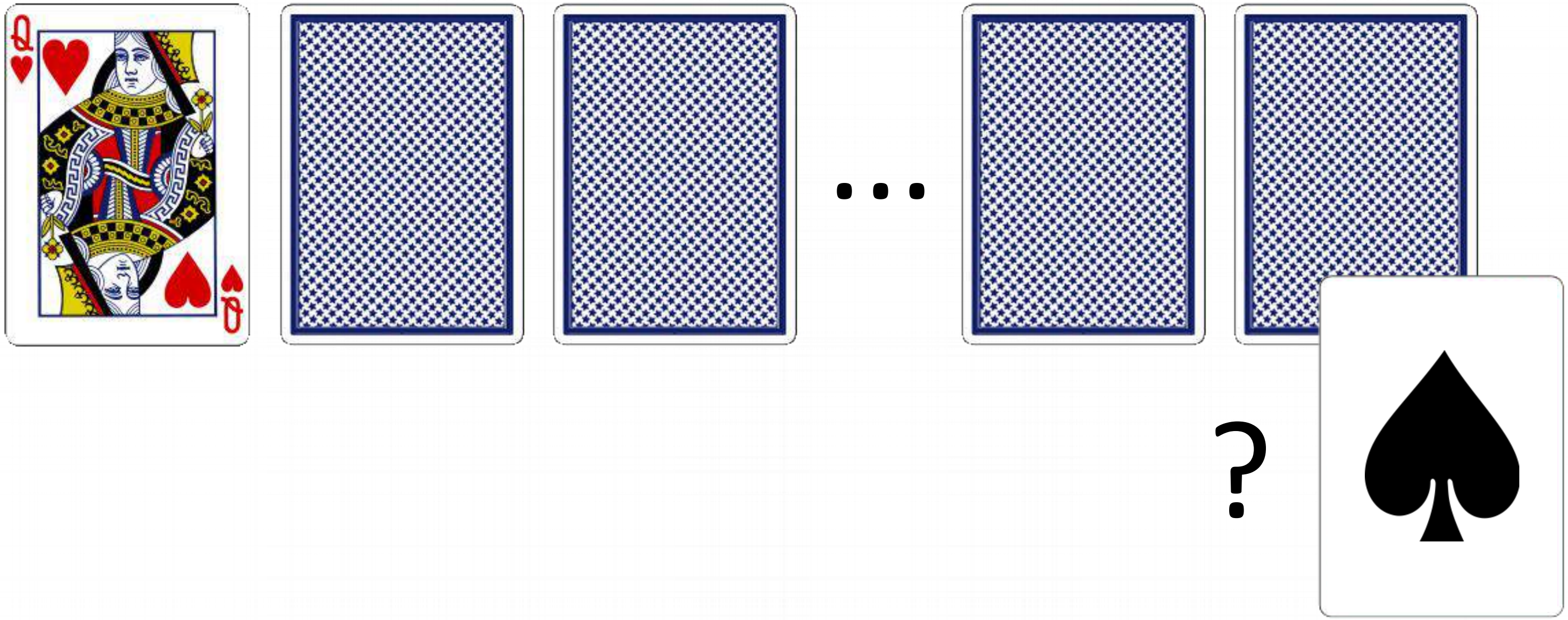
A Simple Reasoning Problem



Probability that Card52 is Spades given that Card1 is QH?

[Van den Broeck 2015]

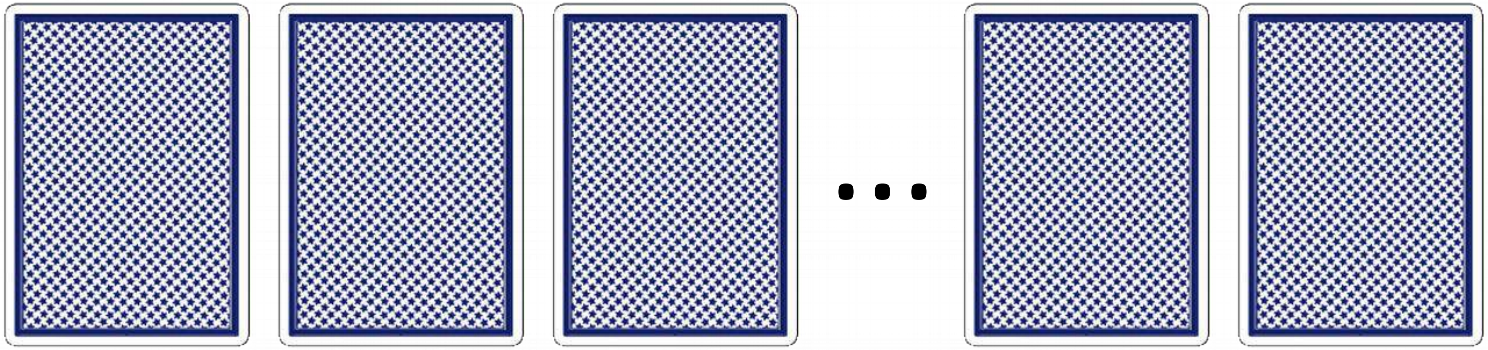
A Simple Reasoning Problem



Probability that Card52 is Spades given that Card1 is QH?

13/51

[Van den Broeck 2015]



Model distribution by FOMC:

$$\Delta = \begin{array}{r} \forall p, \exists c, \mathrm{Card}(p,c) \\ \forall c, \exists p, \mathrm{Card}(p,c) \\ \forall p, \forall c, \forall c', \mathrm{Card}(p,c) \wedge \mathrm{Card}(p,c') \Rightarrow c = c' \end{array}$$

[Van den Broeck 2015]

Beyond NP Pipeline for #P

Reduce to propositional model counting:

[Van den Broeck 2015]

Beyond NP Pipeline for #P

Reduce to propositional model counting:

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

[Van den Broeck 2015]

Beyond NP Pipeline for #P

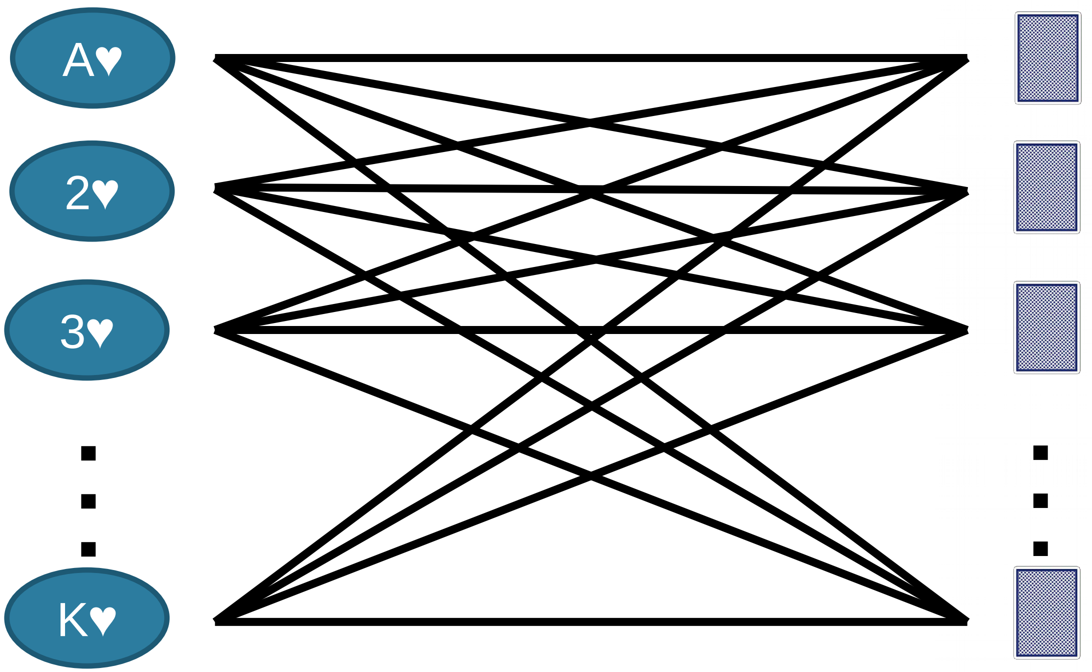
Reduce to propositional model counting:

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?

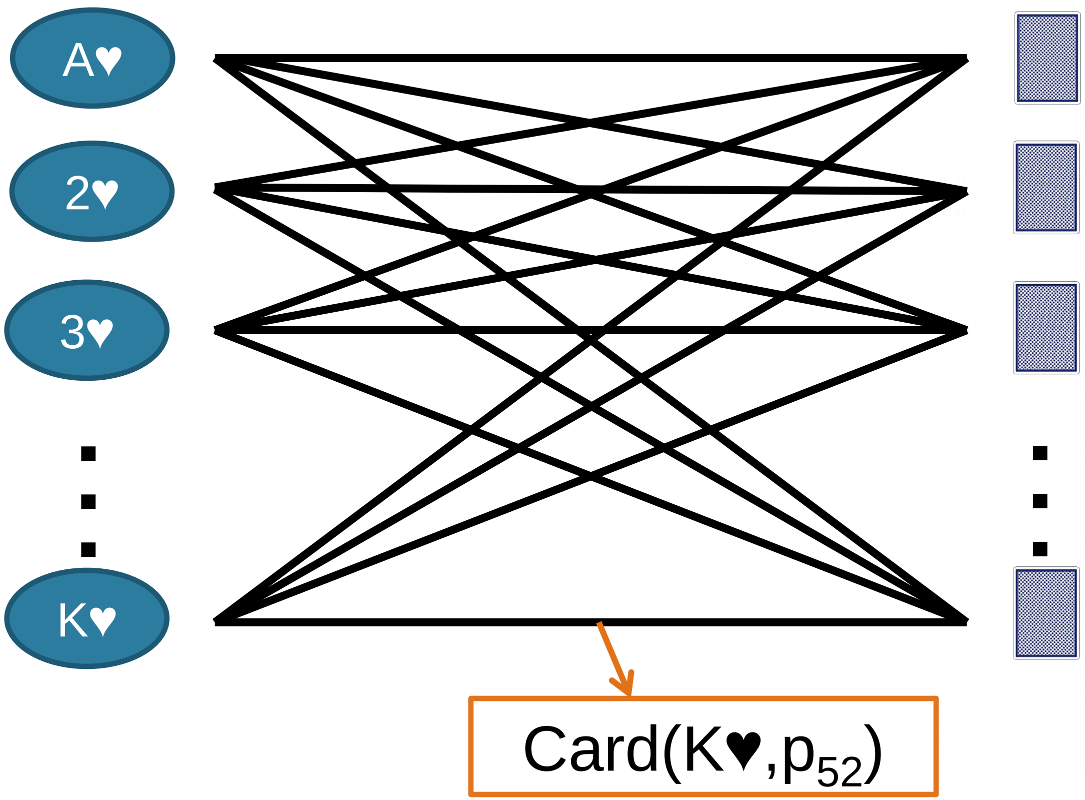
[Van den Broeck 2015]

Deck of Cards Graphically



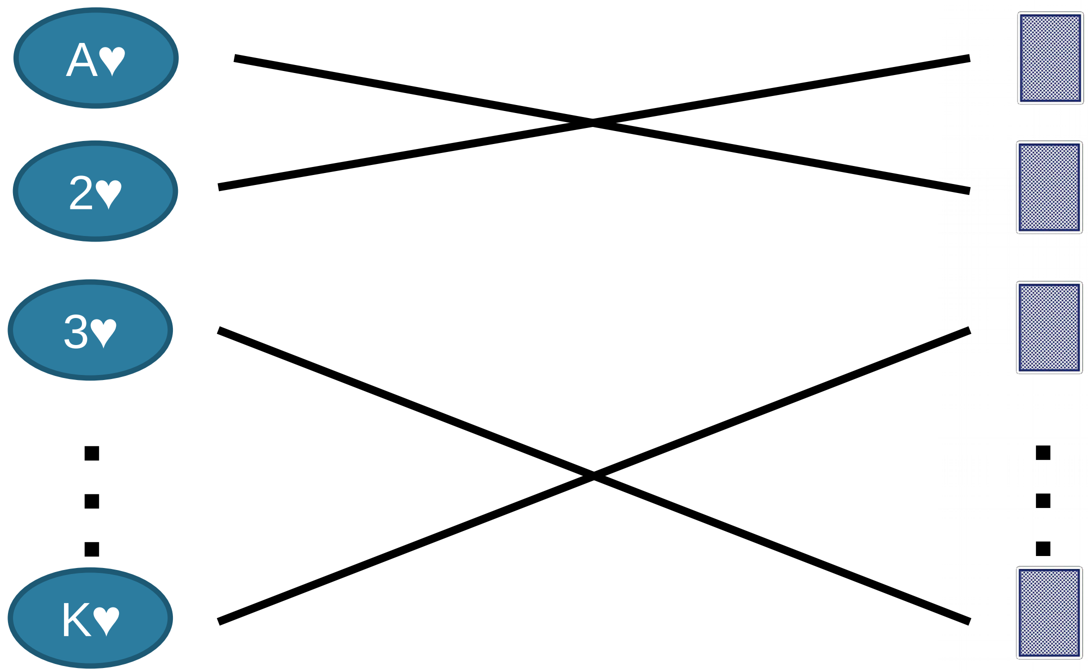
[Van den Broeck 2015]

Deck of Cards Graphically



[Van den Broeck 2015]

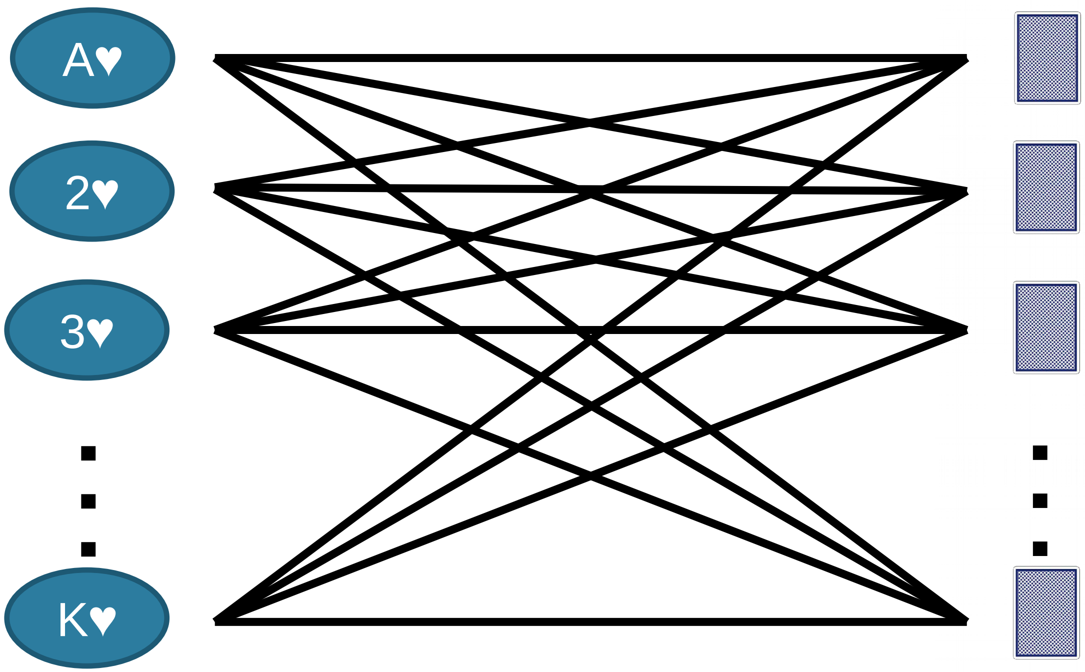
Deck of Cards Graphically



One model/*perfect matching*

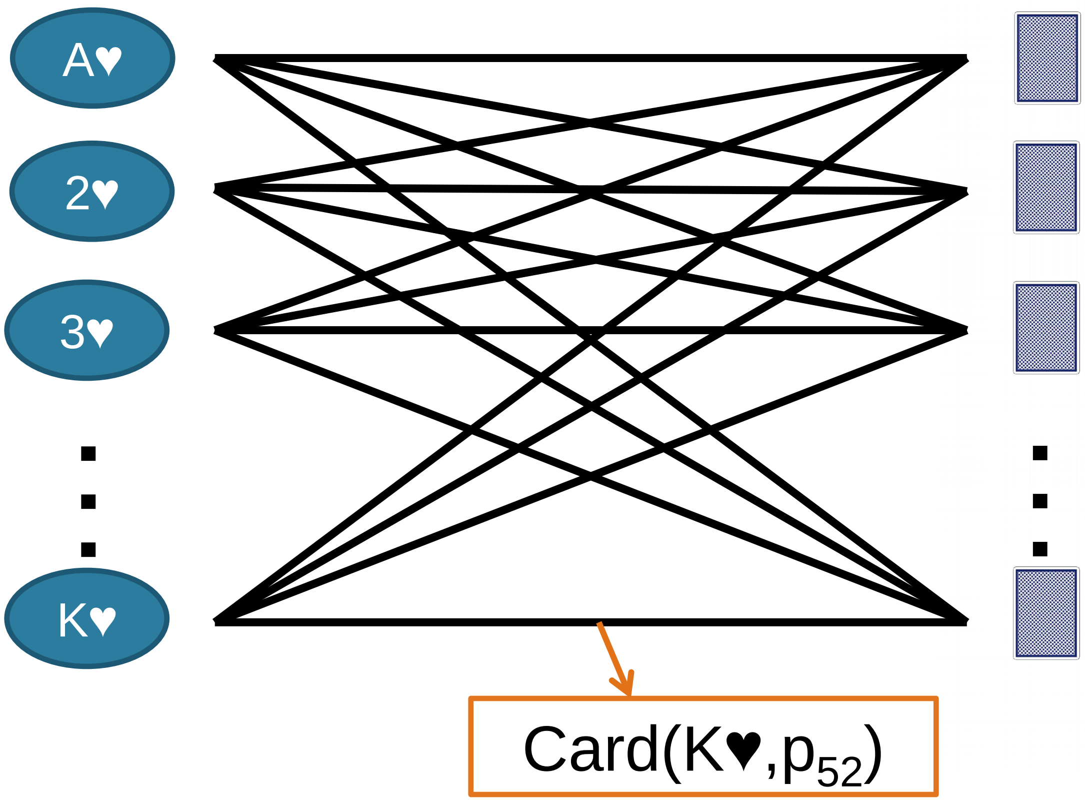
[Van den Broeck 2015]

Deck of Cards Graphically



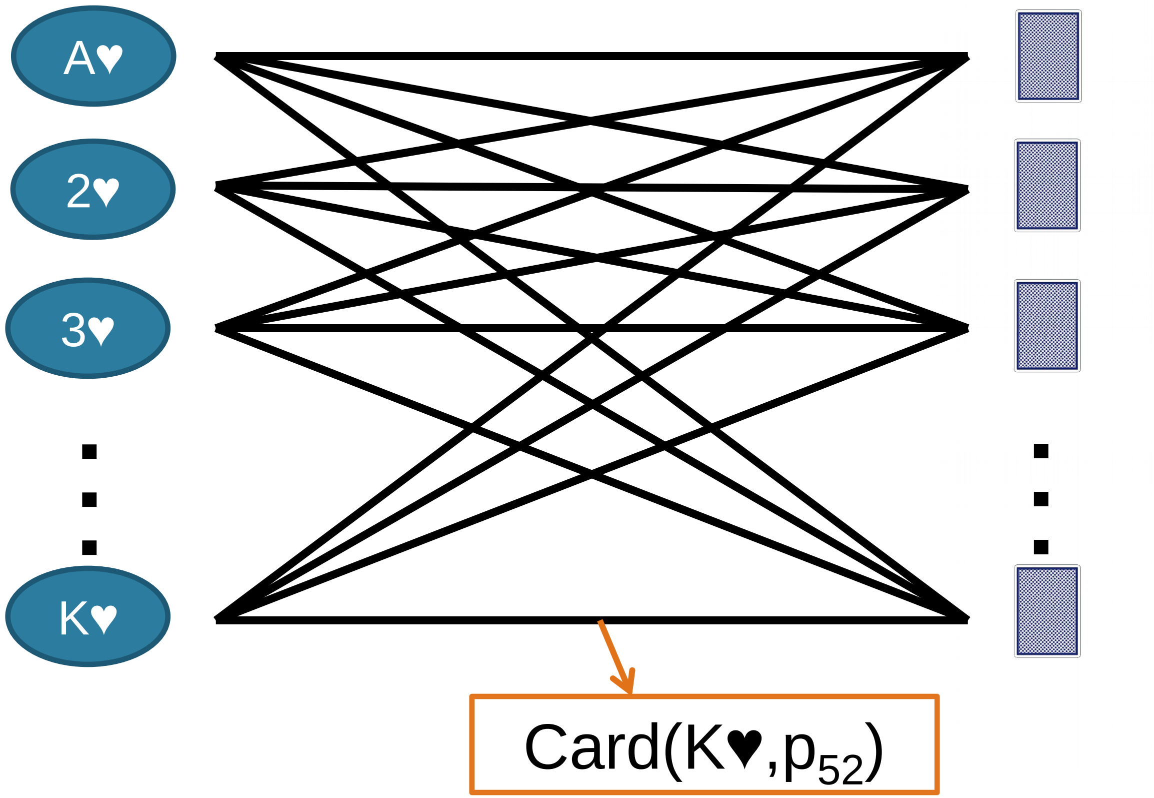
[Van den Broeck 2015]

Deck of Cards Graphically



[Van den Broeck 2015]

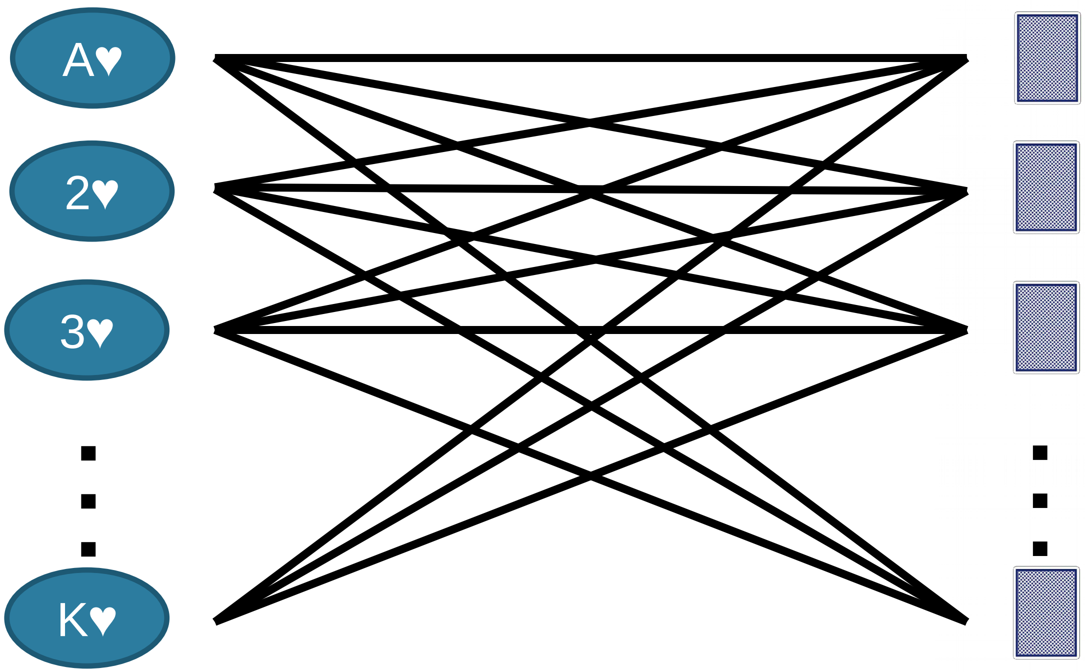
Deck of Cards Graphically



Model counting: How many *perfect matchings*?

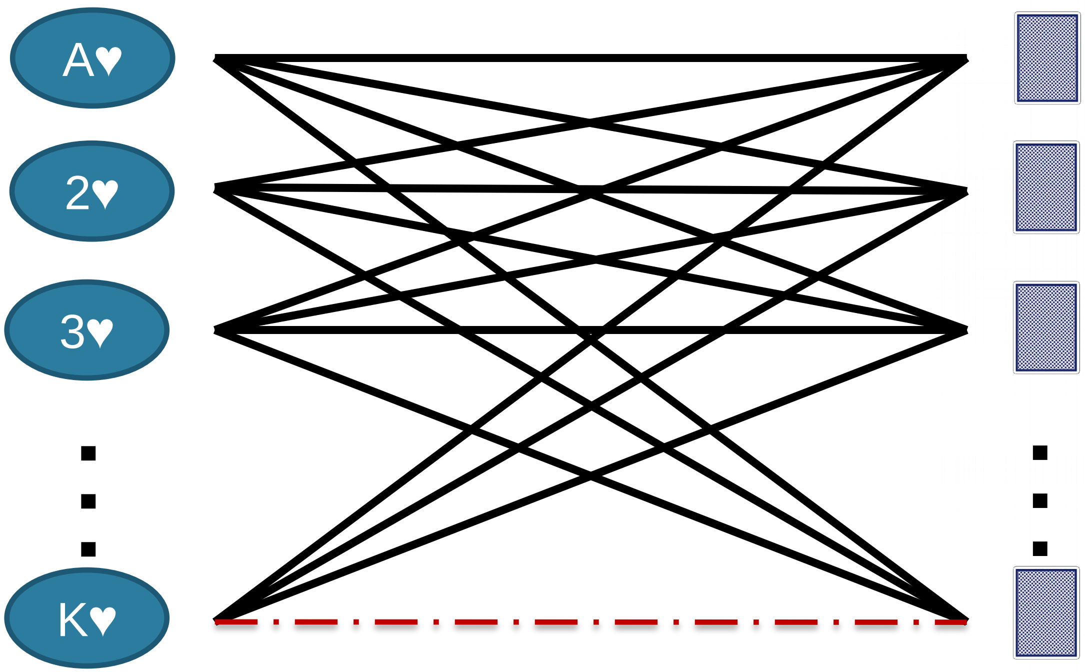
[Van den Broeck 2015]

Deck of Cards Graphically



[Van den Broeck 2015]

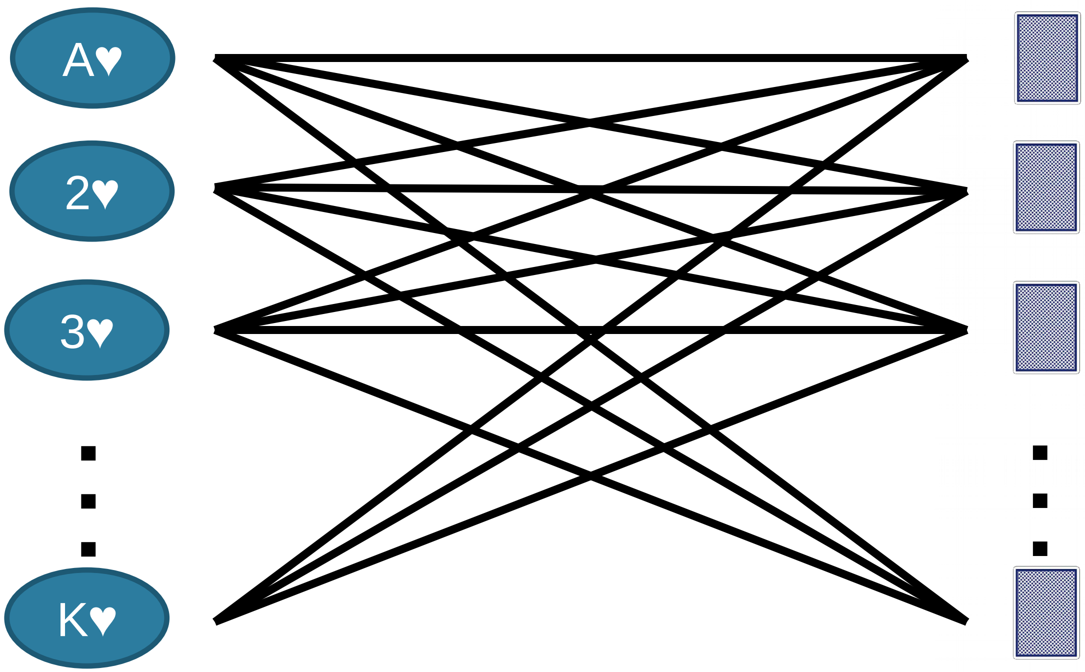
Deck of Cards Graphically



What if I add the unit clause
¬Card(K♥,p_{52}) to my CNF?

[Van den Broeck 2015]

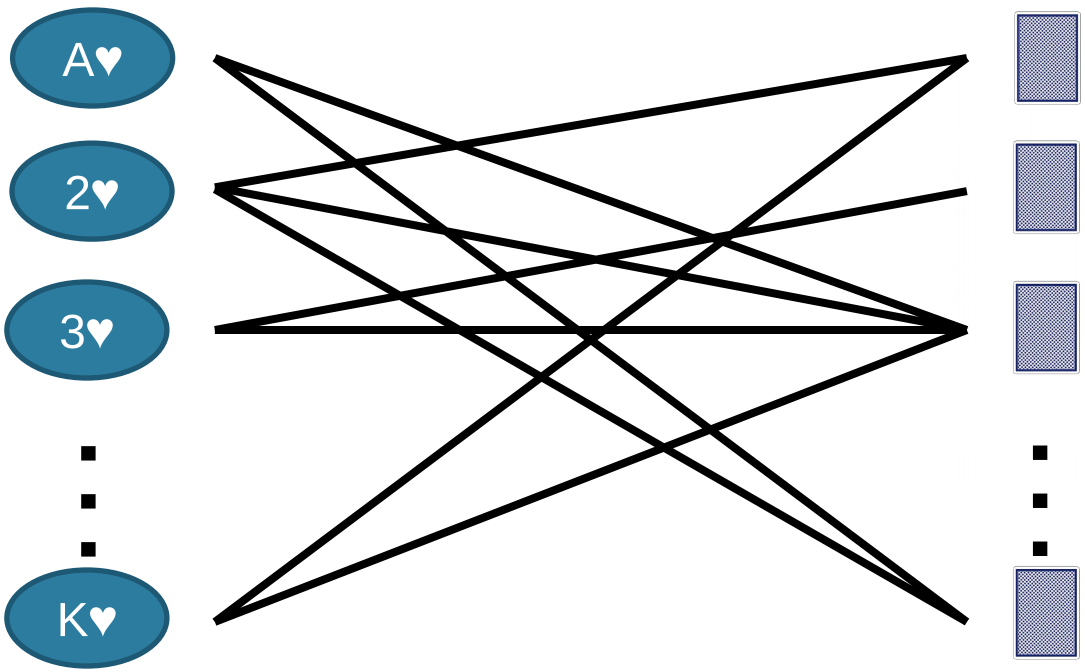
Deck of Cards Graphically



What if I add the unit clause
¬Card(K♥,p_{52}) to my CNF?

[Van den Broeck 2015]

Deck of Cards Graphically



What if I add unit clauses to my CNF?

[Van den Broeck 2015]

Observations

- Deck of cards = complete bigraph
- Unit clause removes edge

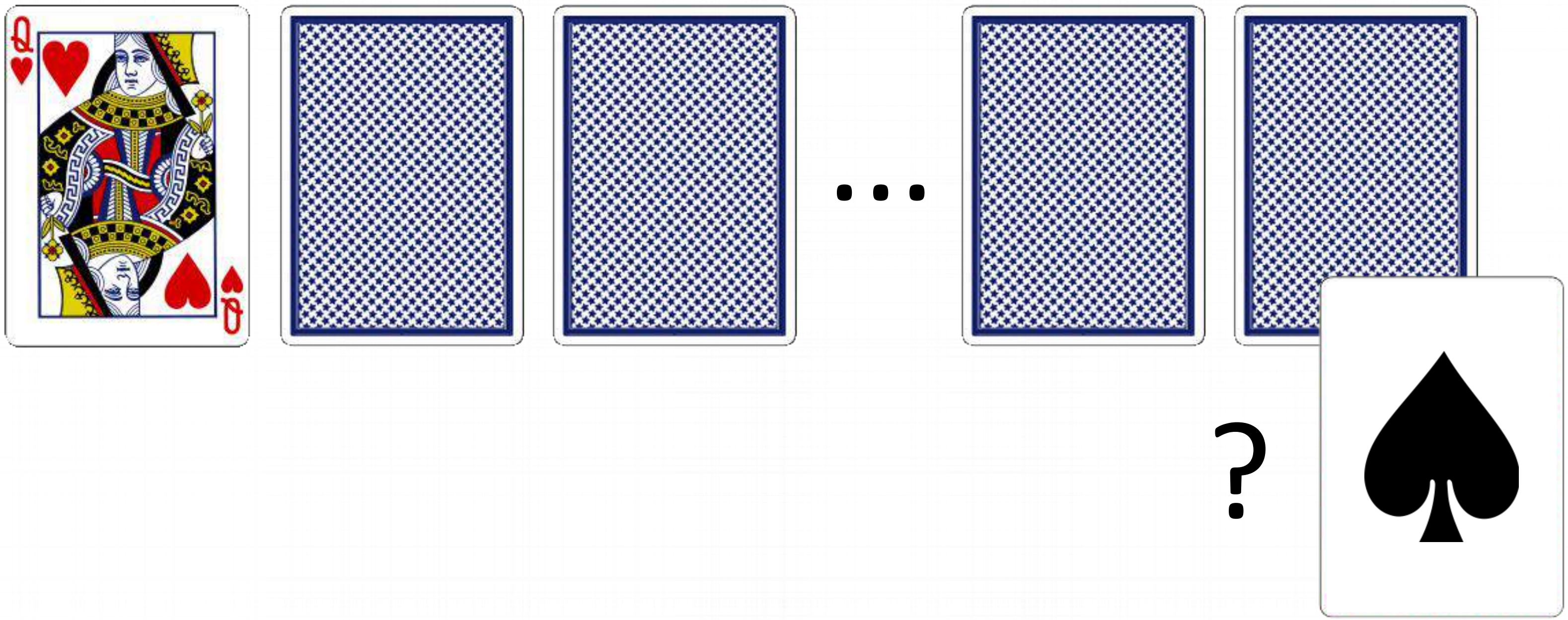
  Encode any bigraph

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

- All solvers efficiently handle unit clauses
- No solver can do cards problem efficiently!

[Van den Broeck 2015]

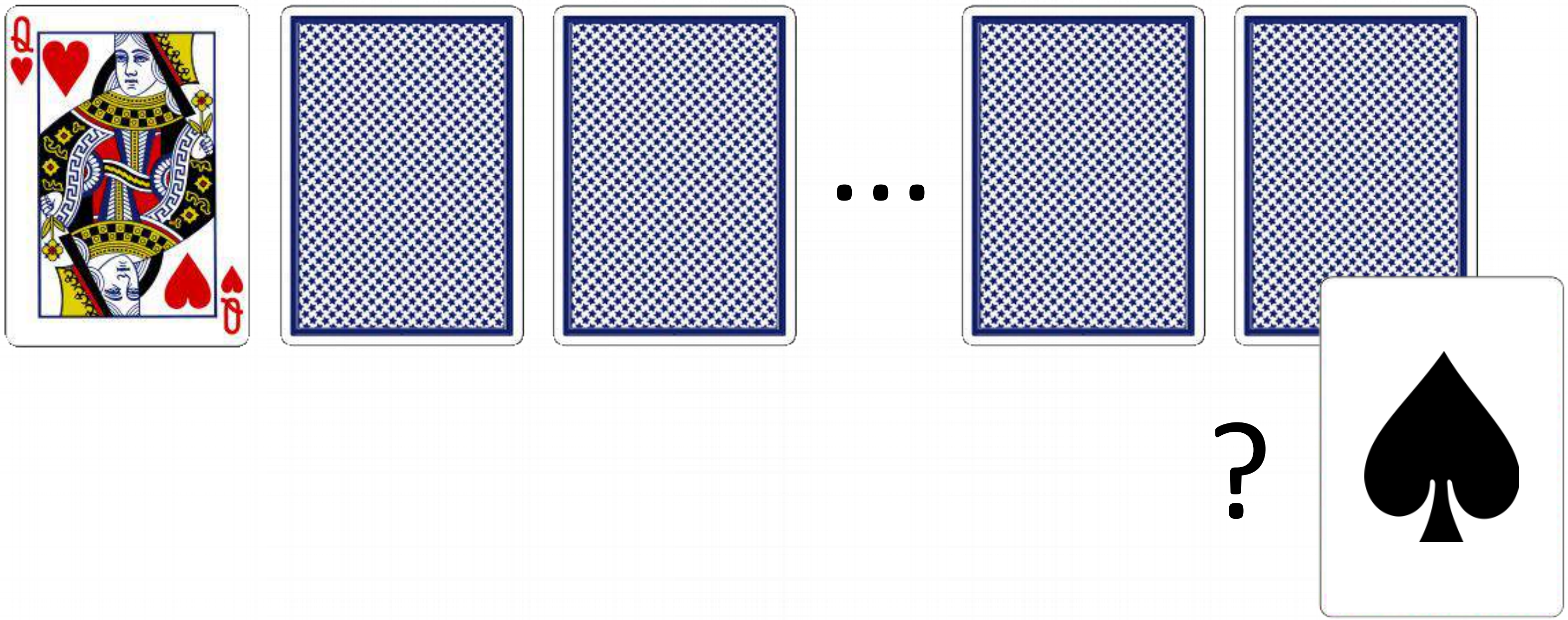
What's Going On Here?



Probability that Card52 is Spades given that Card1 is QH?

[Van den Broeck 2015]

What's Going On Here?

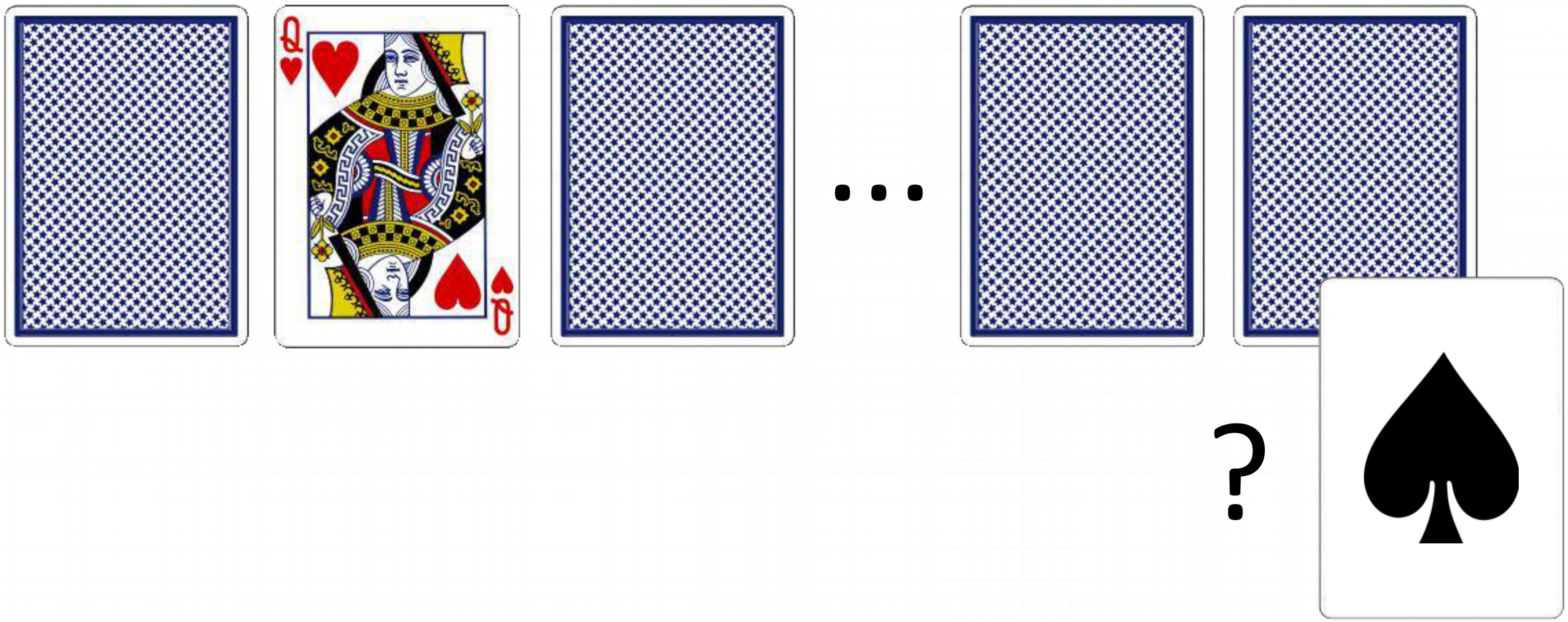


Probability that Card52 is Spades given that Card1 is QH?

13/51

[Van den Broeck 2015]

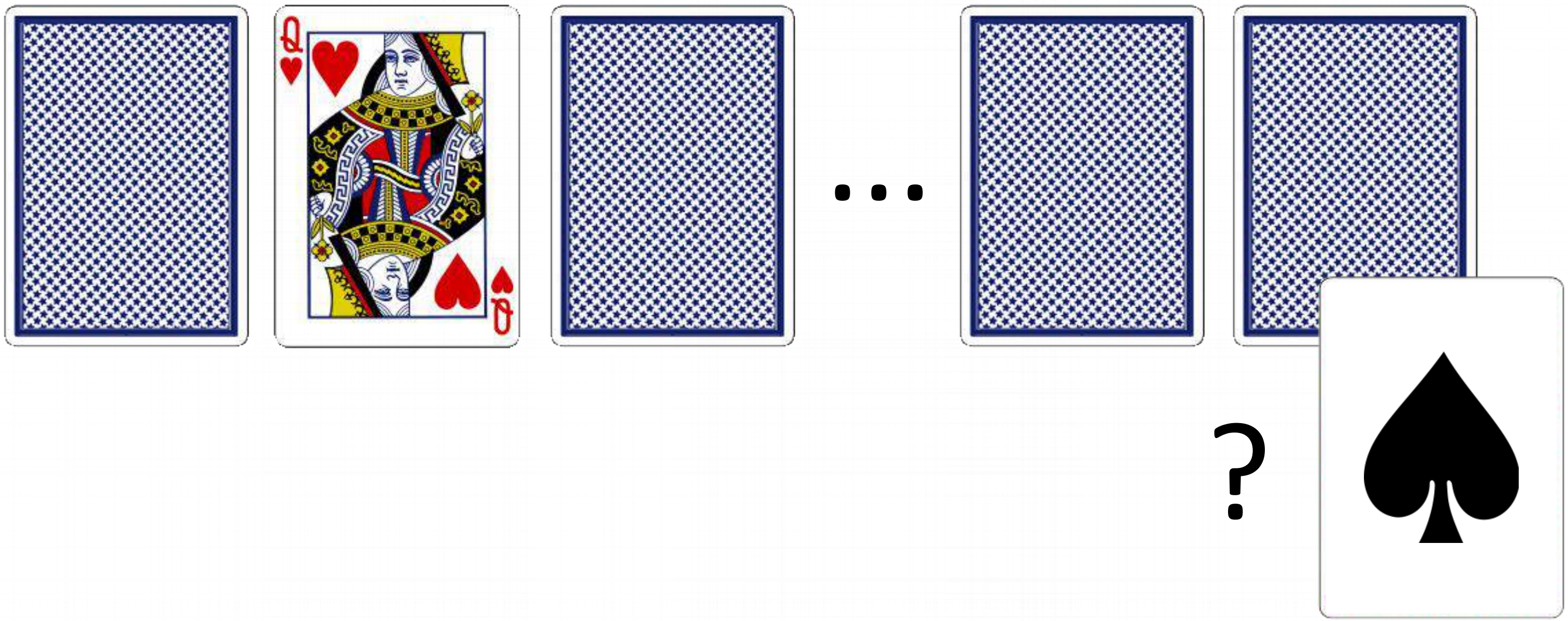
What's Going On Here?



Probability that Card52 is Spades given that Card2 is QH?

[Van den Broeck 2015]

What's Going On Here?

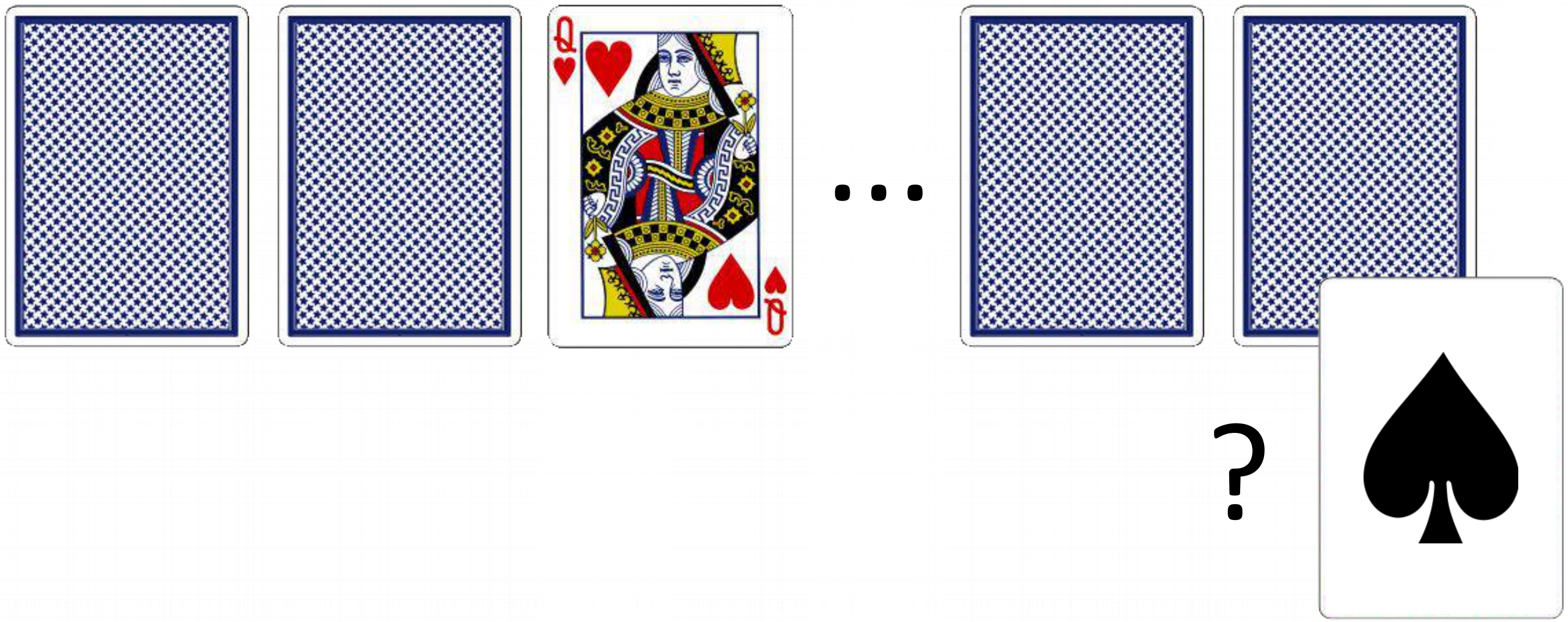


Probability that Card52 is Spades given that Card2 is QH?

13/51

[Van den Broeck 2015]

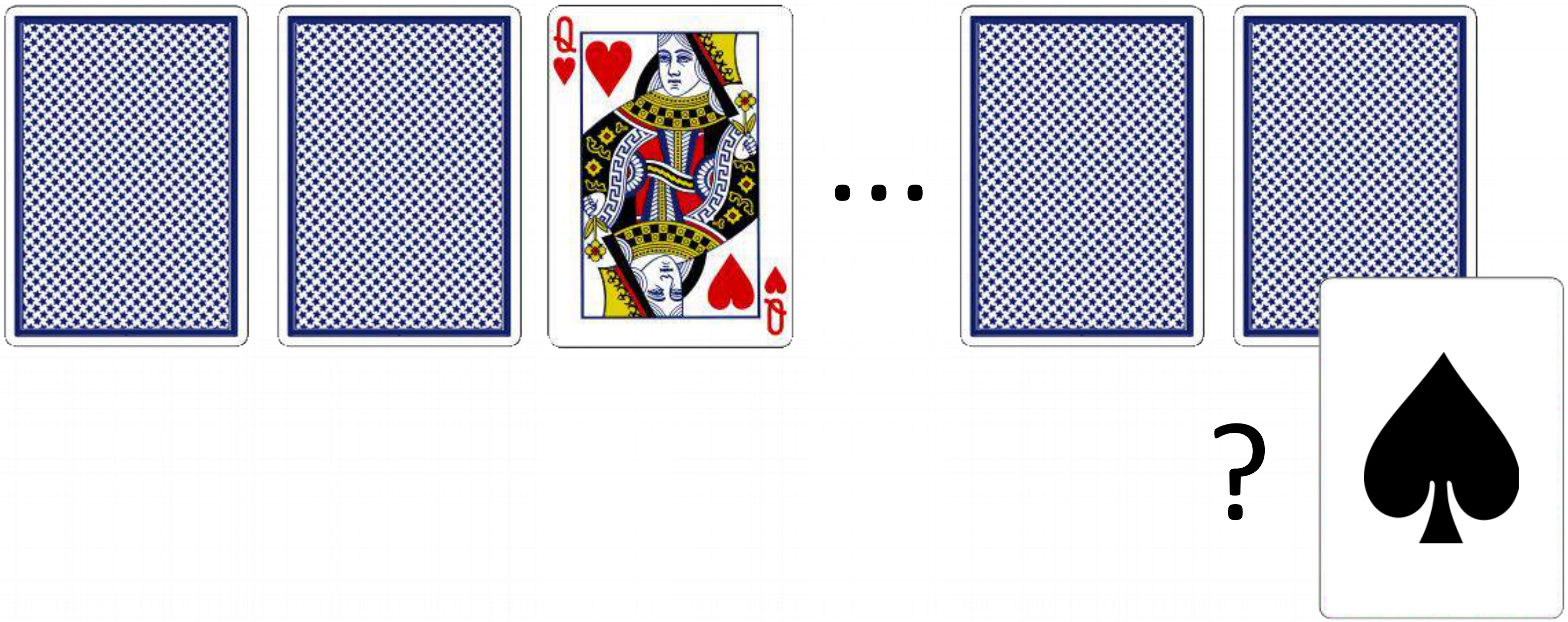
What's Going On Here?



Probability that Card52 is Spades given that Card3 is QH?

[Van den Broeck 2015]

What's Going On Here?

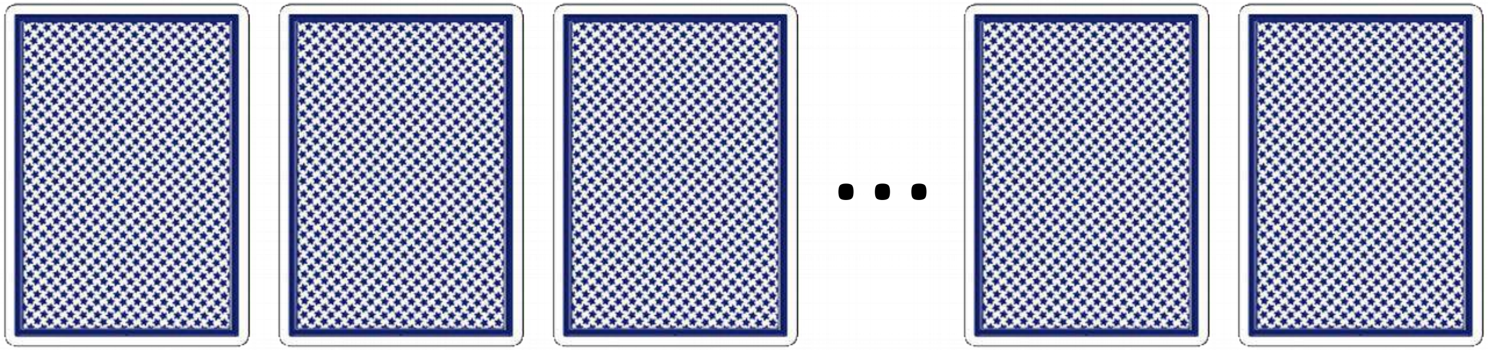


Probability that Card52 is Spades given that Card3 is QH?

13/51

[Van den Broeck 2015]

Tractable Reasoning

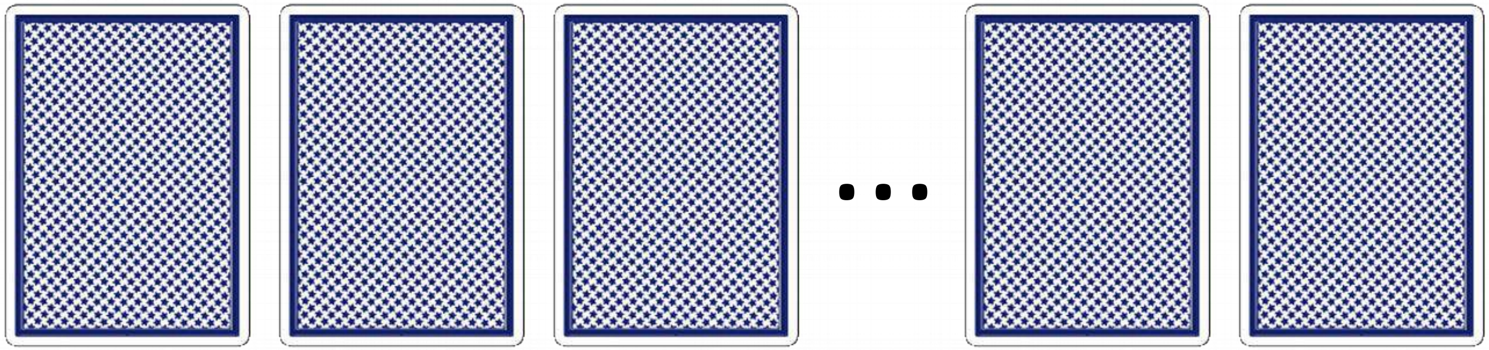


What's going on here?

Which property makes reasoning tractable?

[Niepert 2014, Van den Broeck 2015]

Tractable Reasoning



What's going on here?

Which property makes reasoning tractable?

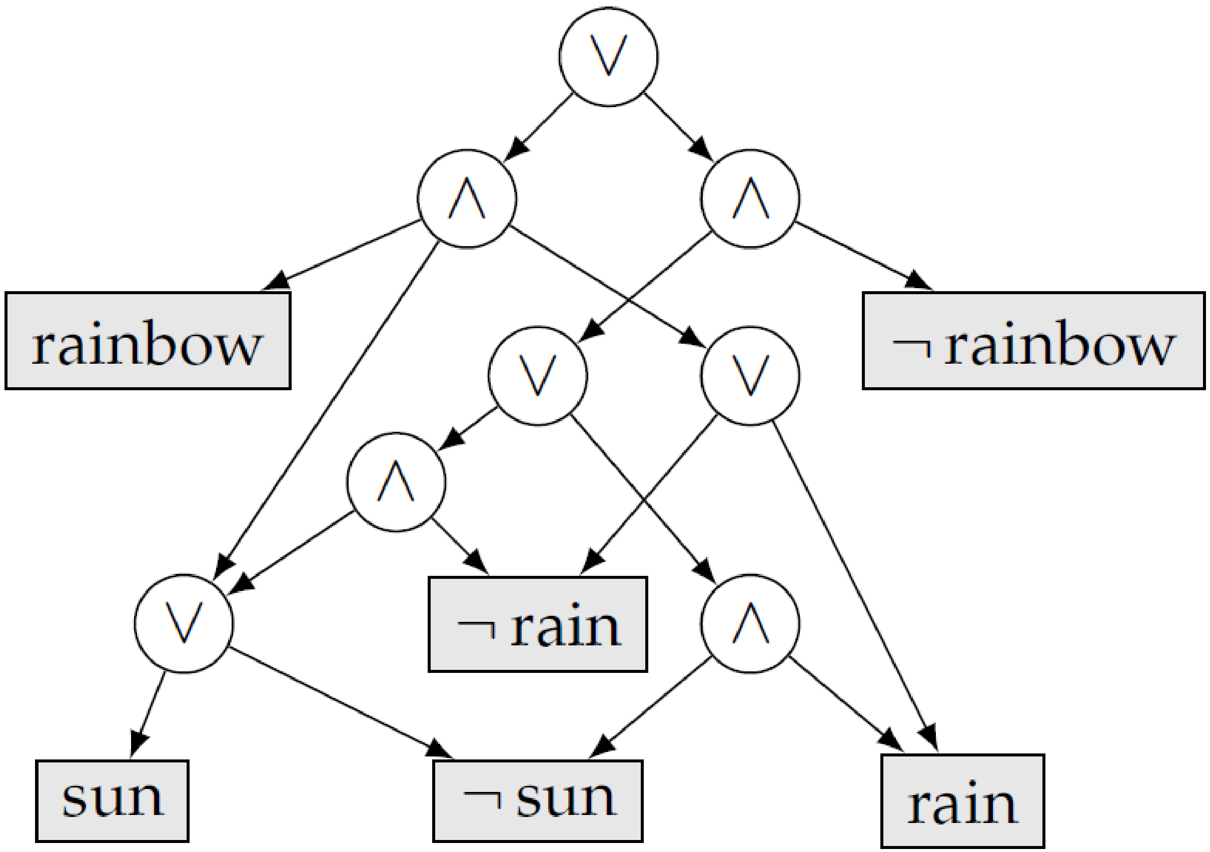
- High-level (first-order) reasoning
- Symmetry
- Exchangeability

⇒ **Lifted Inference**

[Niepert 2014, Van den Broeck 2015]

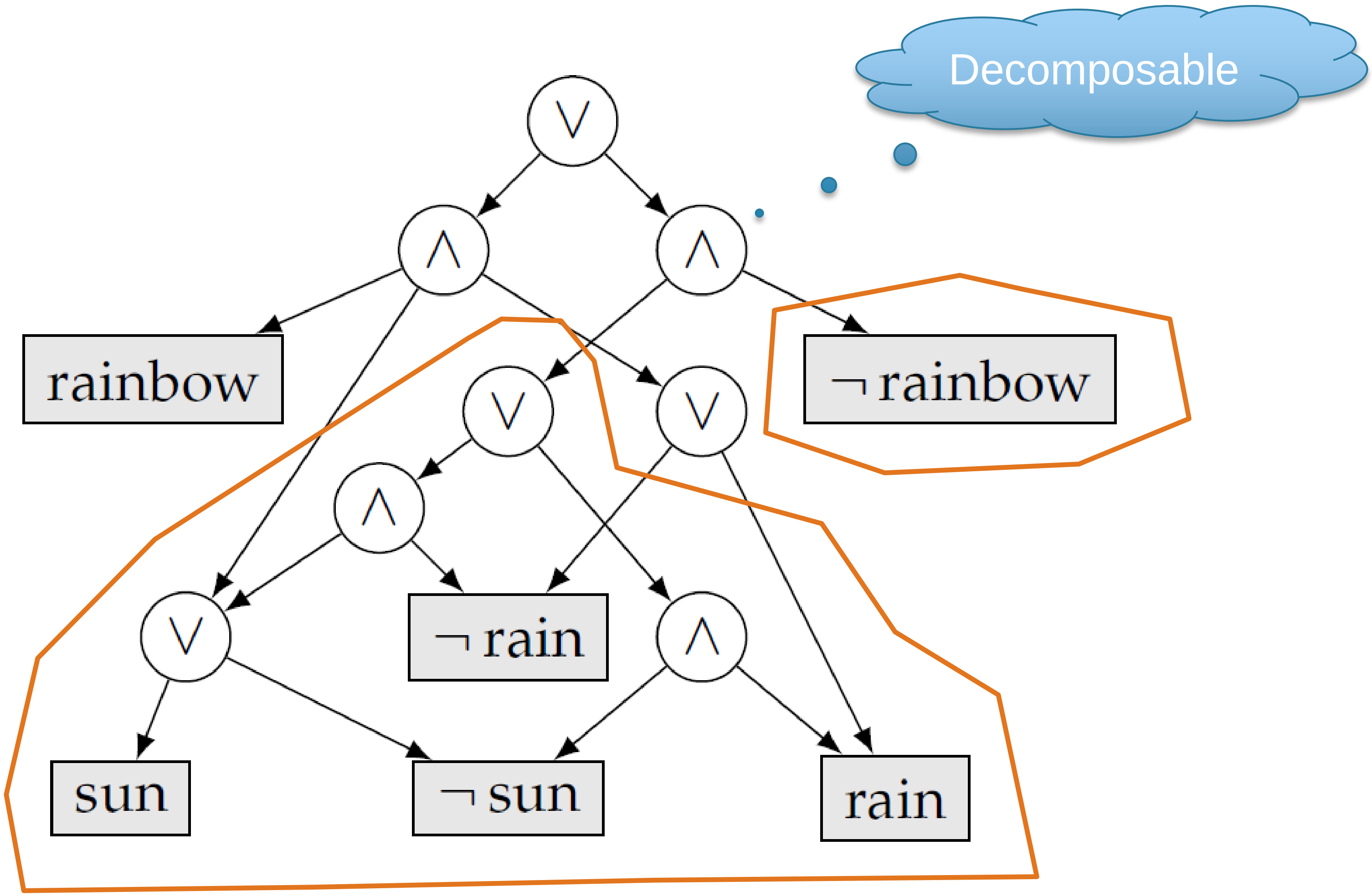
What are first-order circuit ***languages****?*

Negation Normal Form



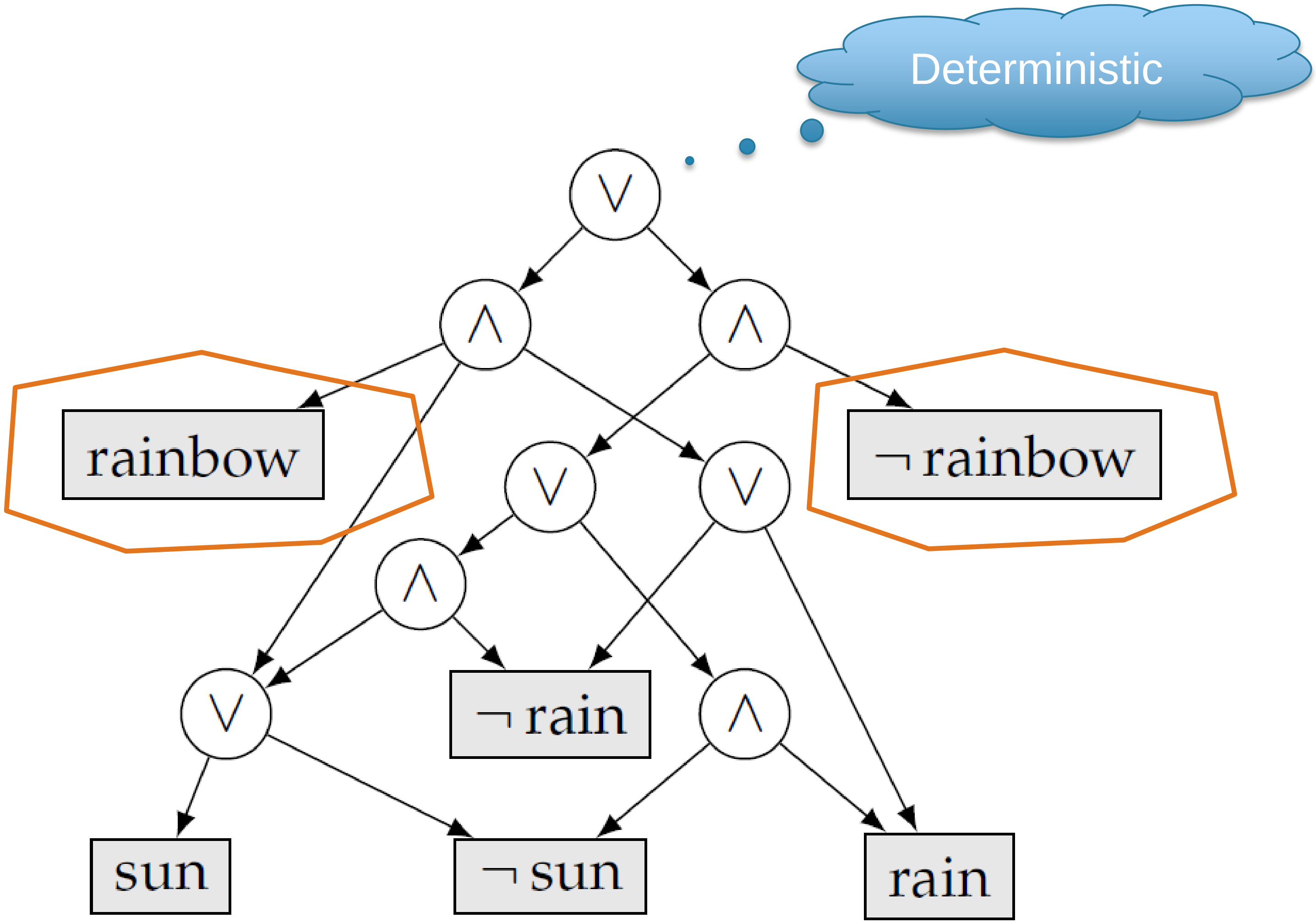
[Darwiche 2002]

Decomposable NNF



[Darwiche 2002]

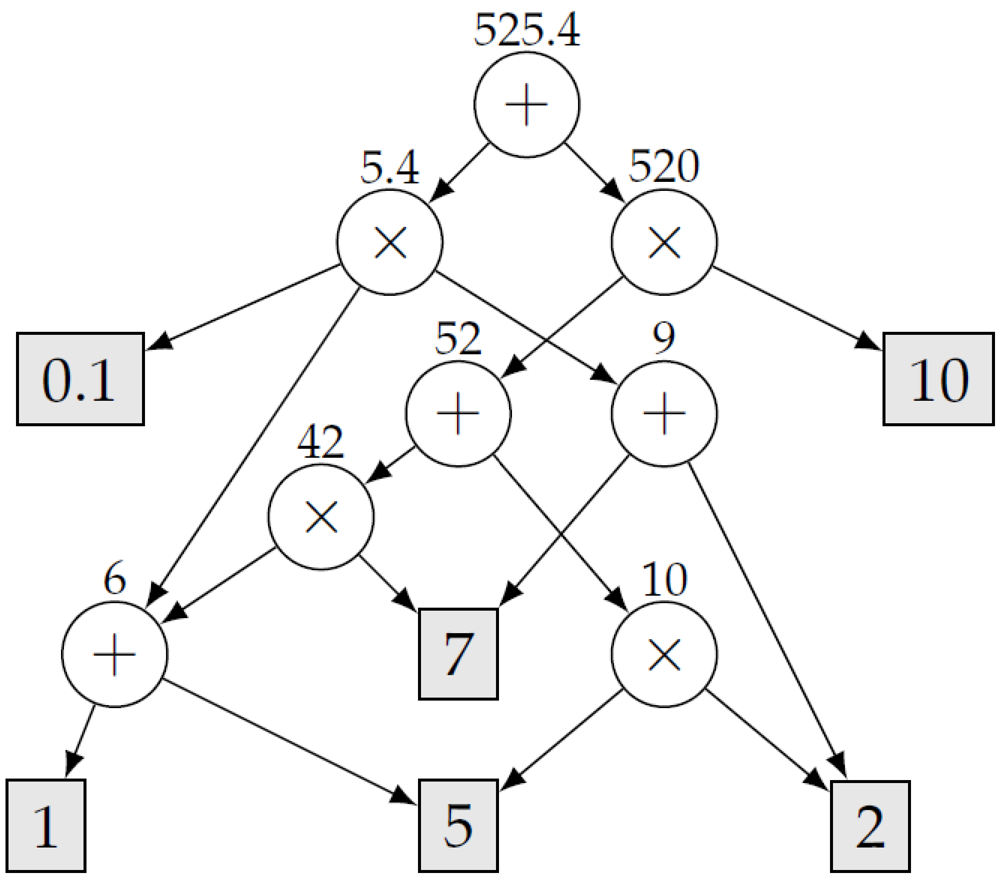
Deterministic Decomposable NNF



[Darwiche 2002]

Deterministic Decomposable NNF

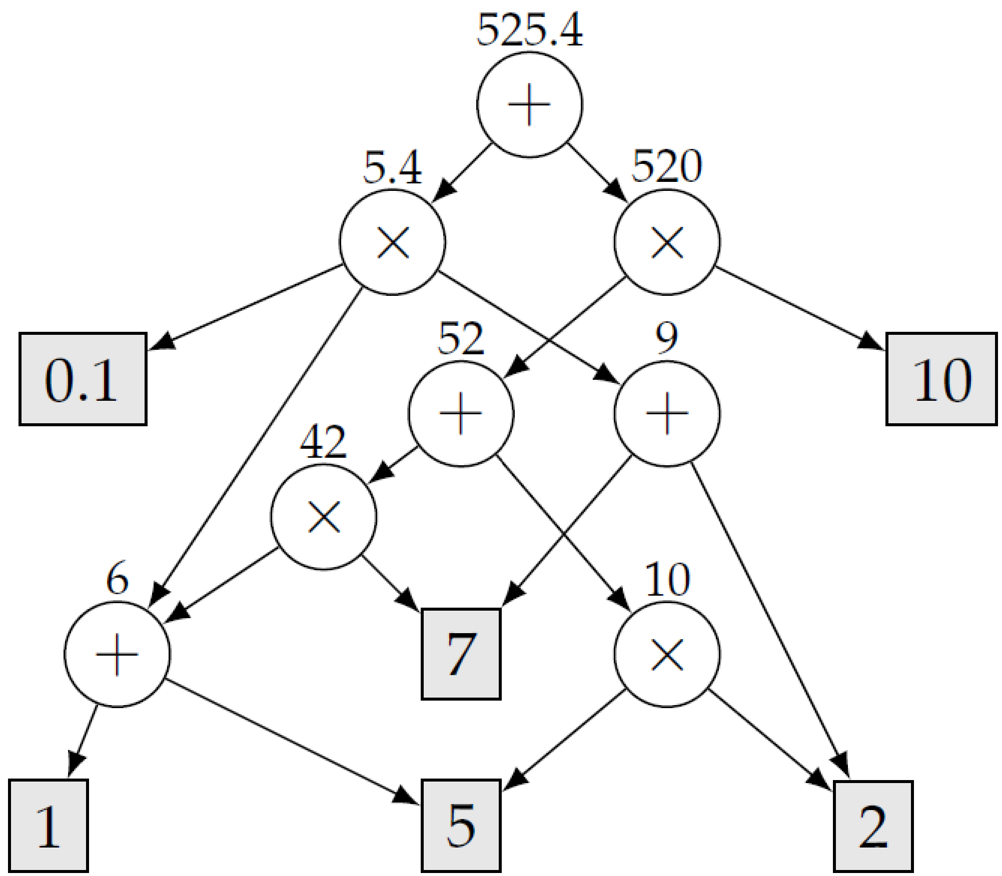
Weighted Model Counting



[Darwiche 2002]

Deterministic Decomposable NNF

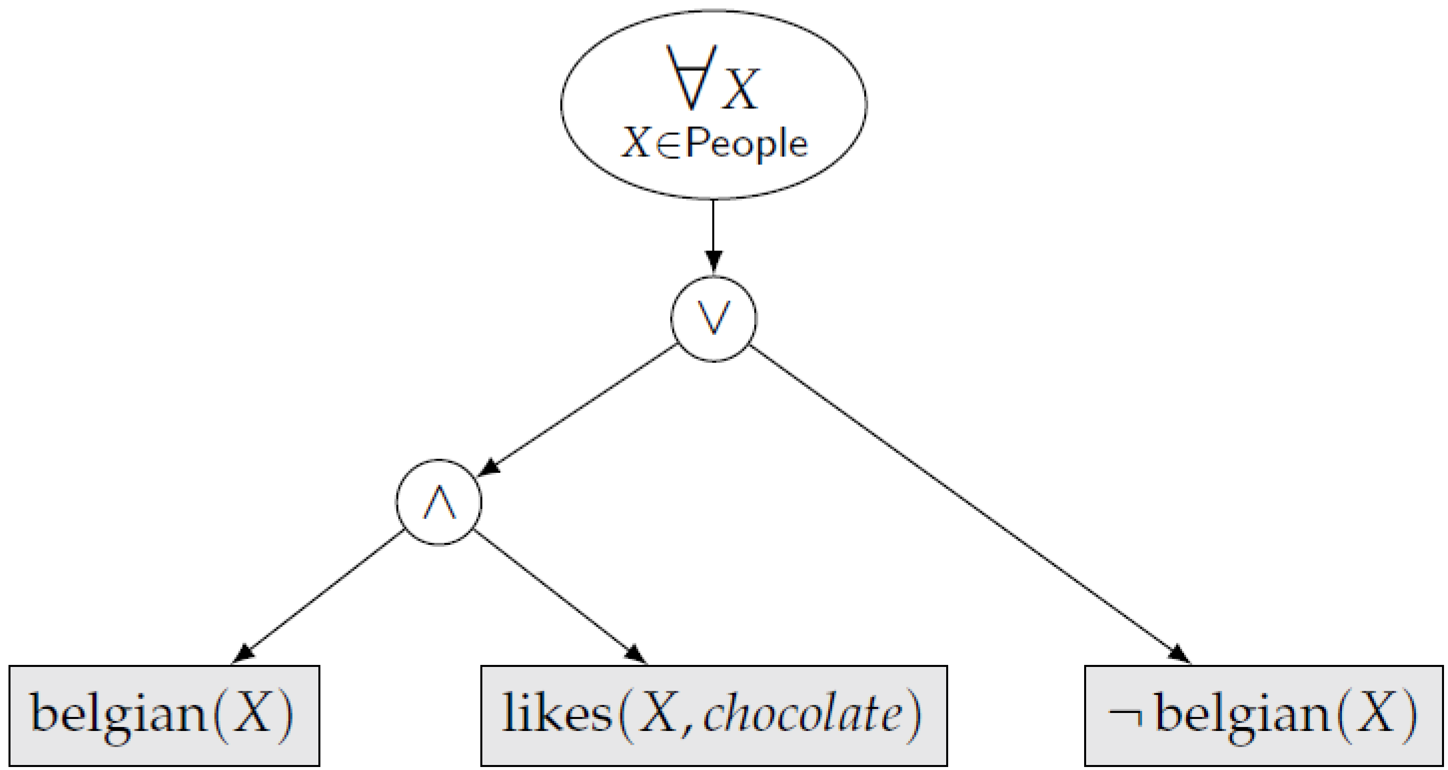
Weighted Model Counting **and much more!**



[Darwiche 2002]

First-Order NNF

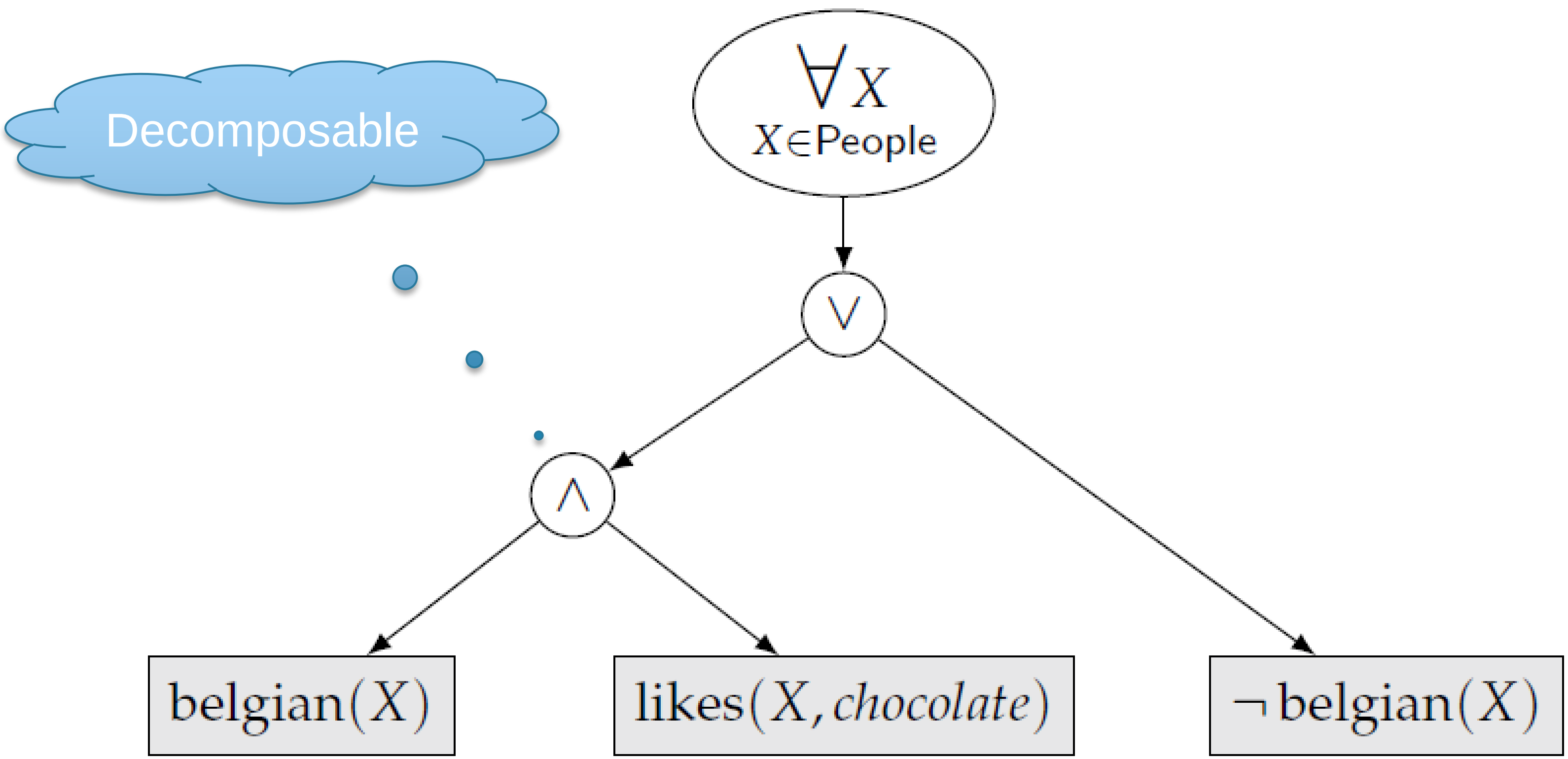
$$\forall X, X \in \text{People} : \text{belgian}(X) \Rightarrow \text{likes}(X, \textit{chocolate})$$



[Van den Broeck 2013]

First-Order Decomposability

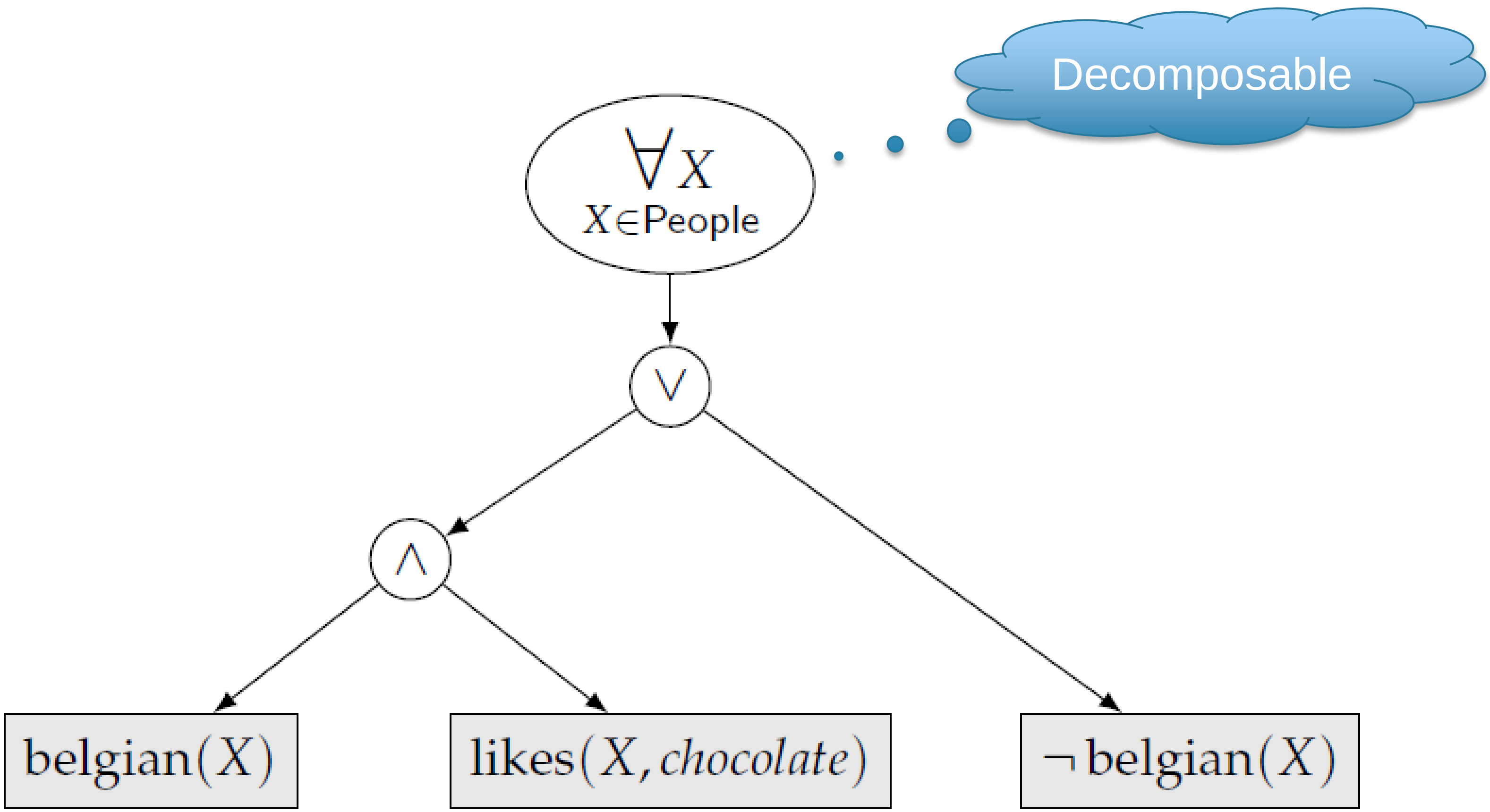
$$\forall X, X \in \text{People} : \text{belgian}(X) \Rightarrow \text{likes}(X, \textit{chocolate})$$



[Van den Broeck 2013]

First-Order Decomposability

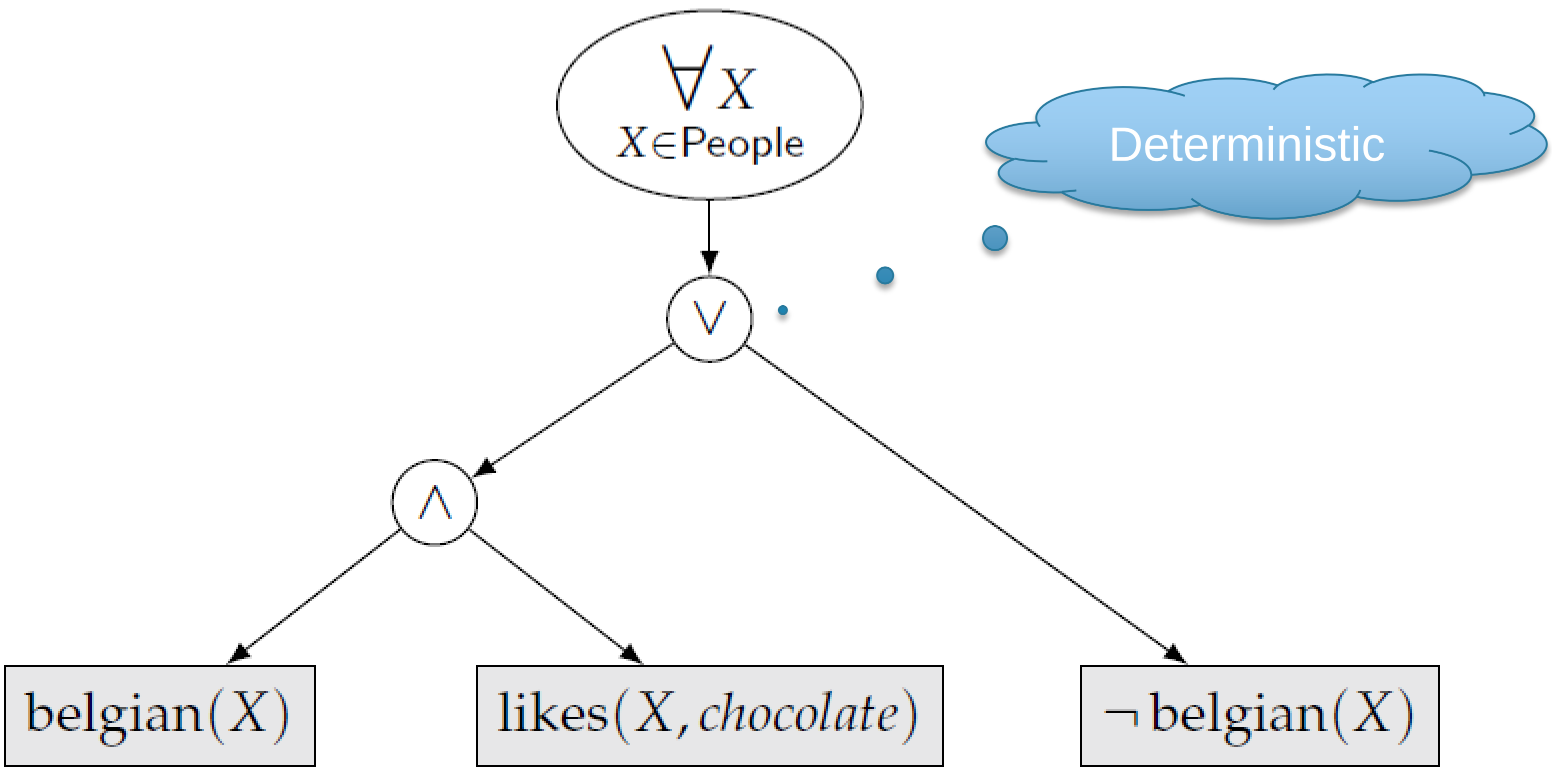
$$\forall X, X \in \text{People} : \text{belgian}(X) \Rightarrow \text{likes}(X, \mathit{chocolate})$$



[Van den Broeck 2013]

First-Order Determinism

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

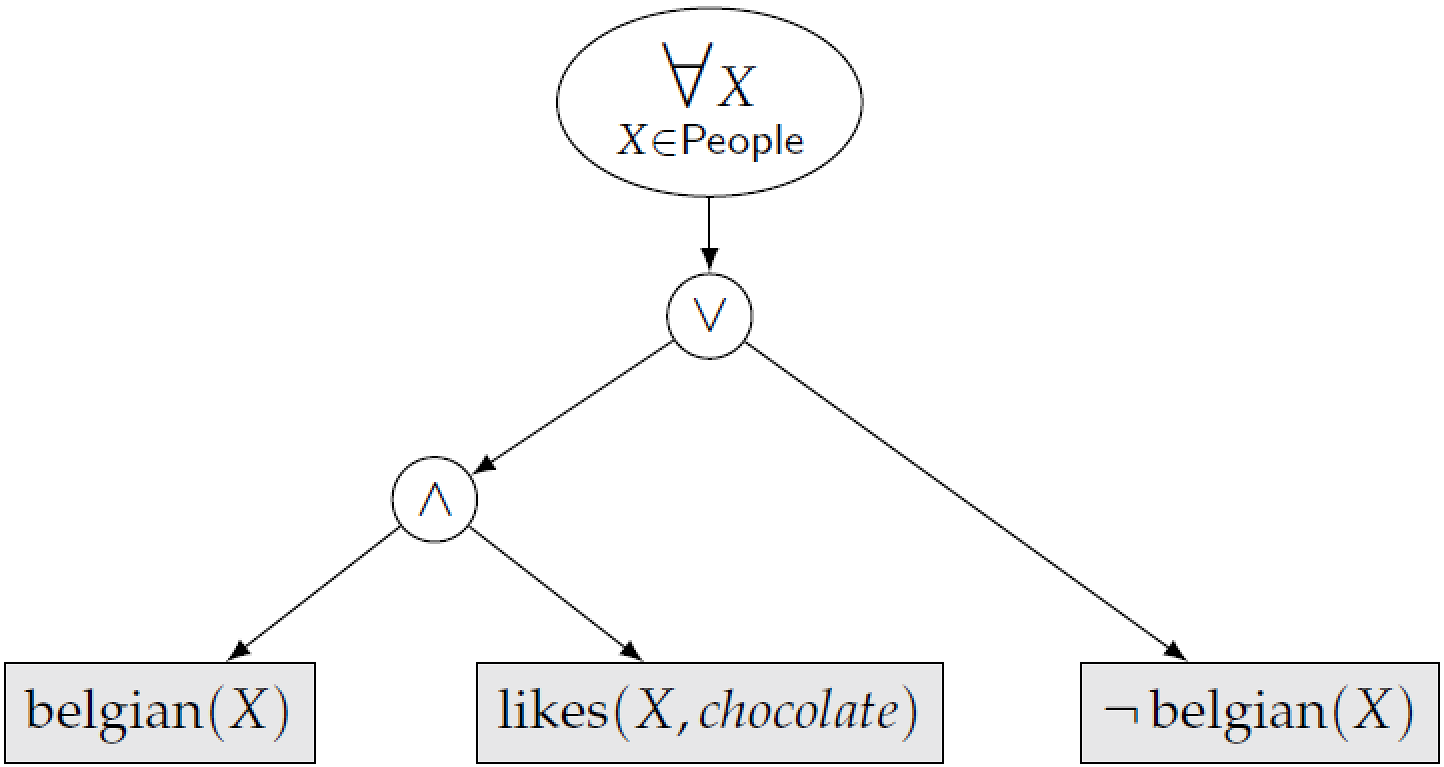


[Van den Broeck 2013]

Deterministic Decomposable FO NNF

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

Weighted Model Counting

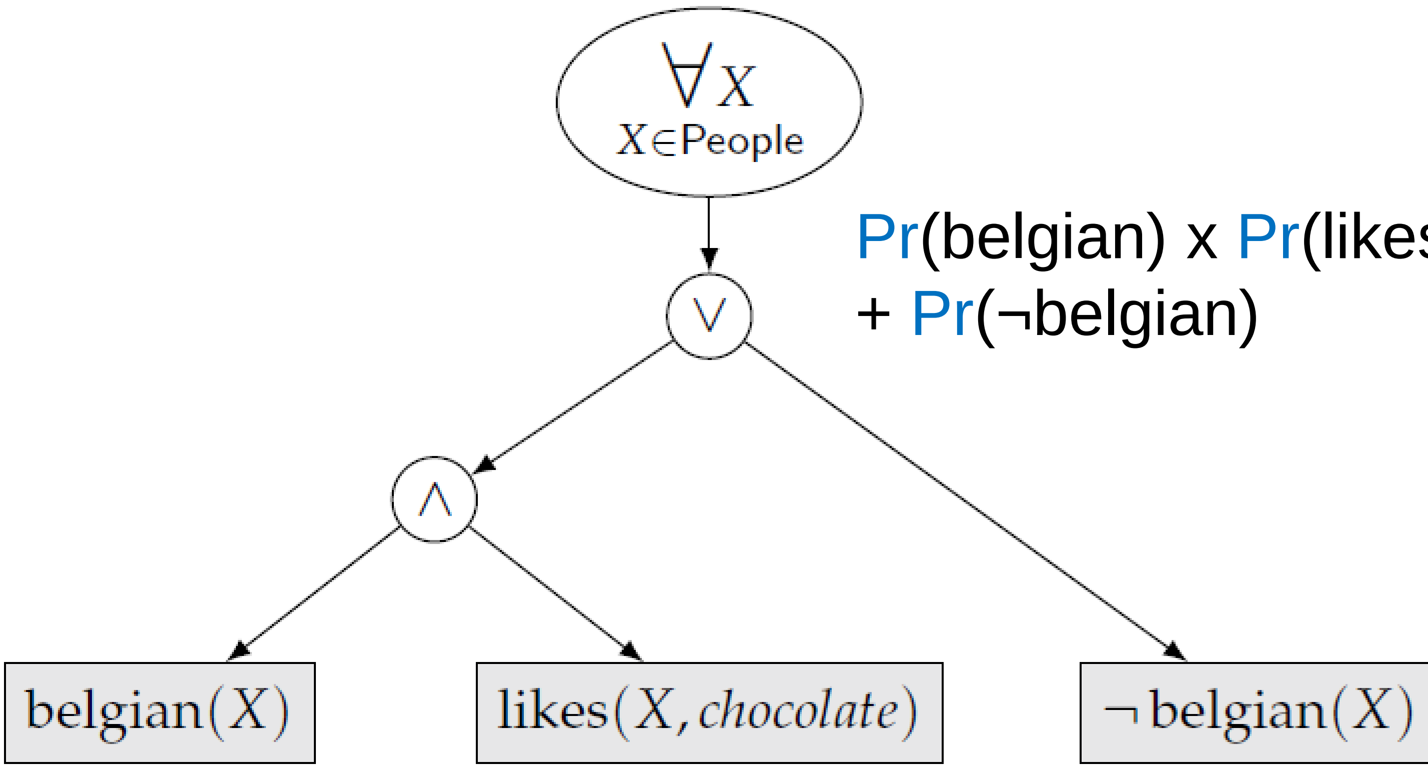


[Van den Broeck 2013]

Deterministic Decomposable FO NNF

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

Weighted Model Counting



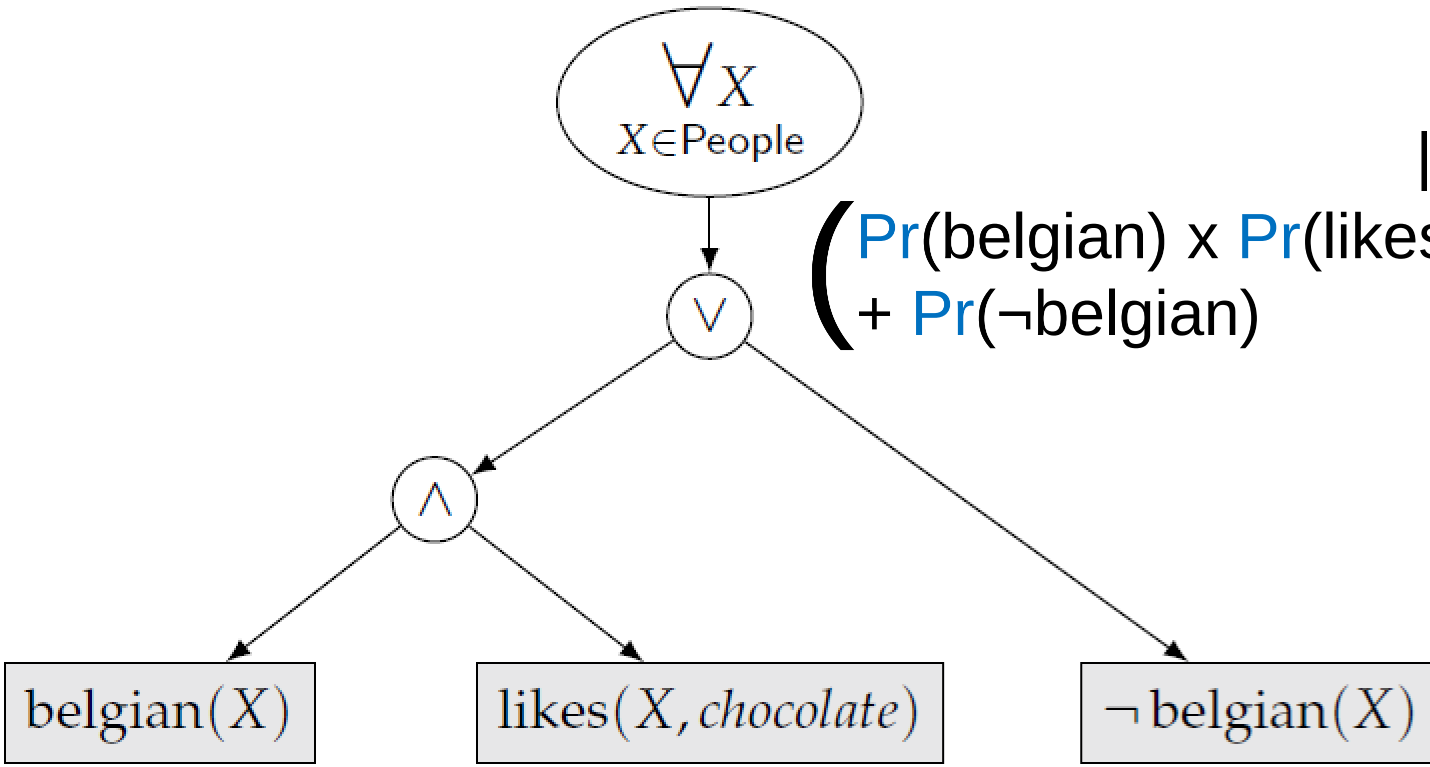
Pr(belgian) x Pr(likes)
+ Pr(¬belgian)

[Van den Broeck 2013]

Deterministic Decomposable FO NNF

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

Weighted Model Counting



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

[Van den Broeck 2013]

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

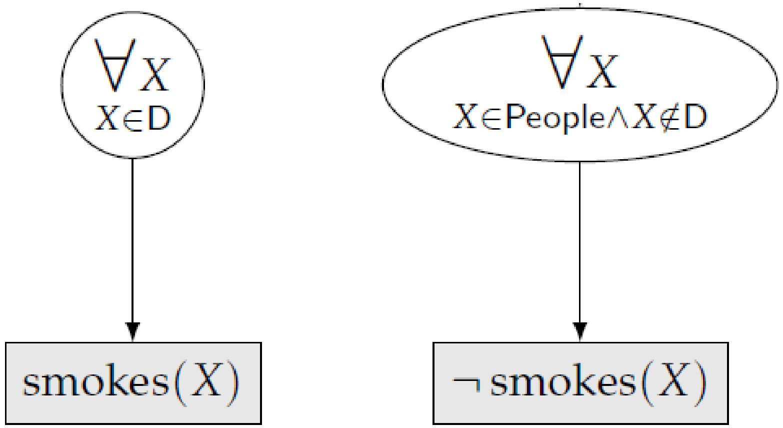
Deterministic Decomposable FO NNF

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

[Van den Broeck 2013]

Deterministic Decomposable FO NNF

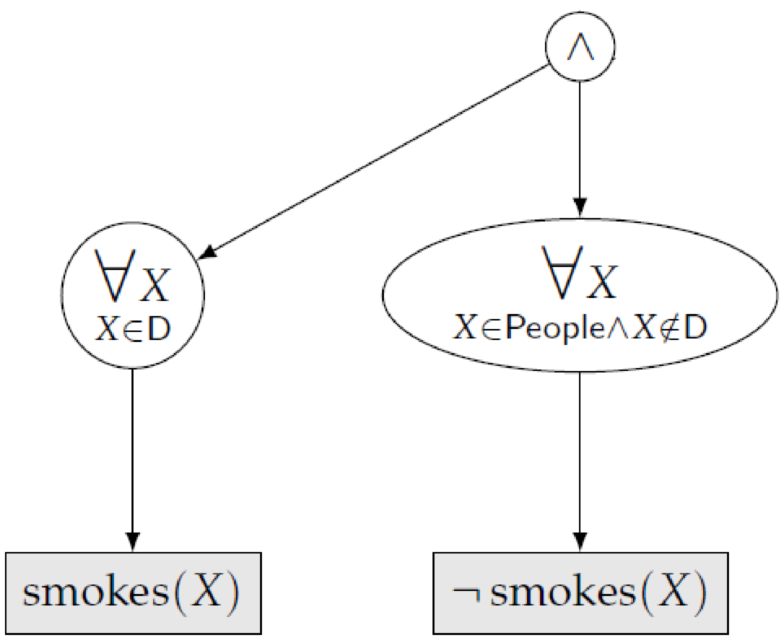
Δ = ∀x ,y ∈ **People**, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))



[Van den Broeck 2013]

Deterministic Decomposable FO NNF

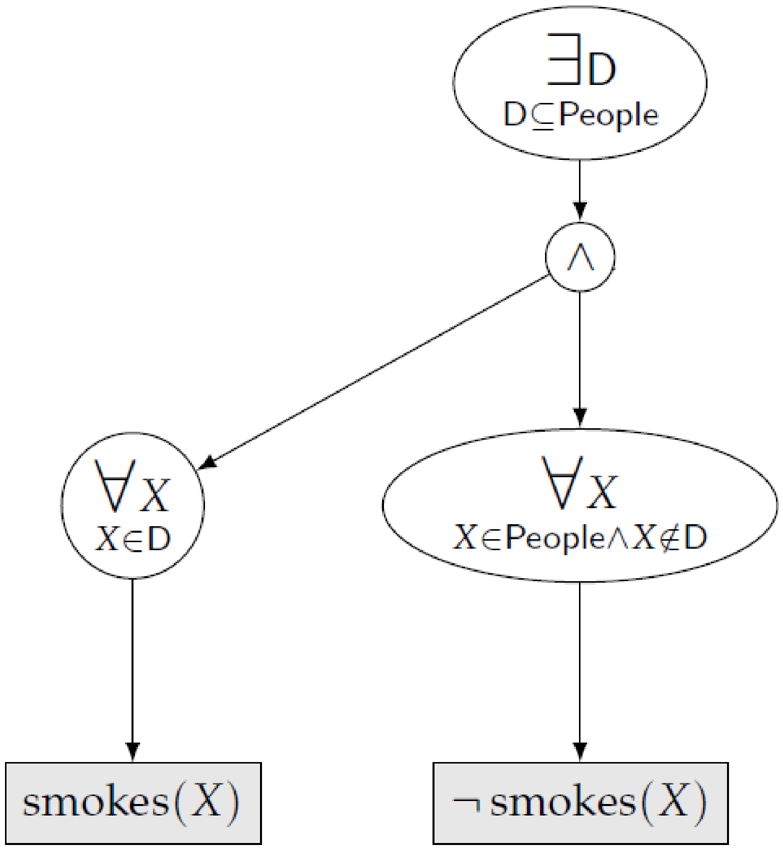
Δ = ∀x ,y ∈ **People**, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))



[Van den Broeck 2013]

Deterministic Decomposable FO NNF

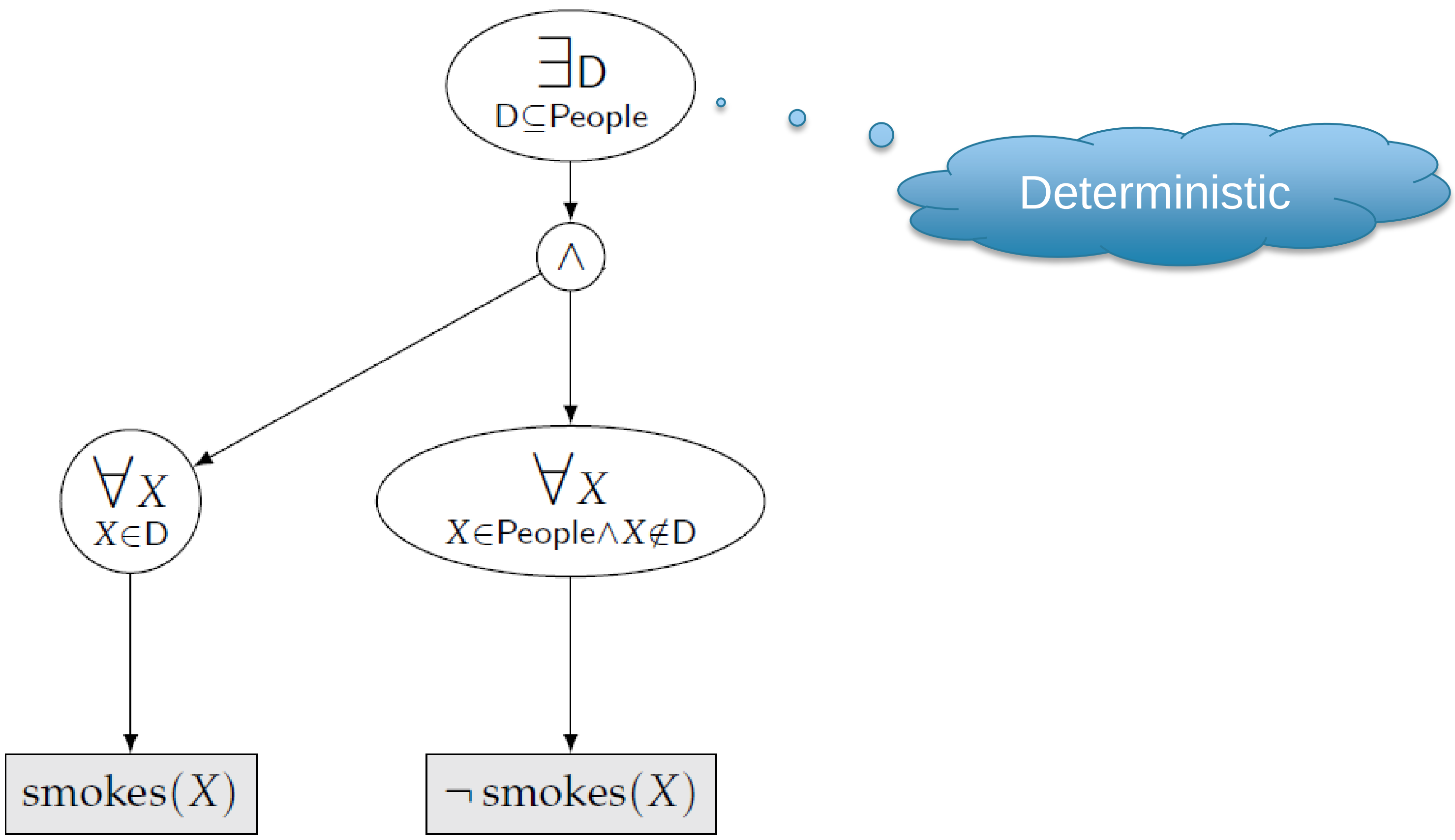
$\Delta = \forall x, y \in$ **People**, $(\text{Smokes}(x) \land \text{Friends}(x,y) \Rightarrow \text{Smokes}(y))$



[Van den Broeck 2013]

Deterministic Decomposable FO NNF

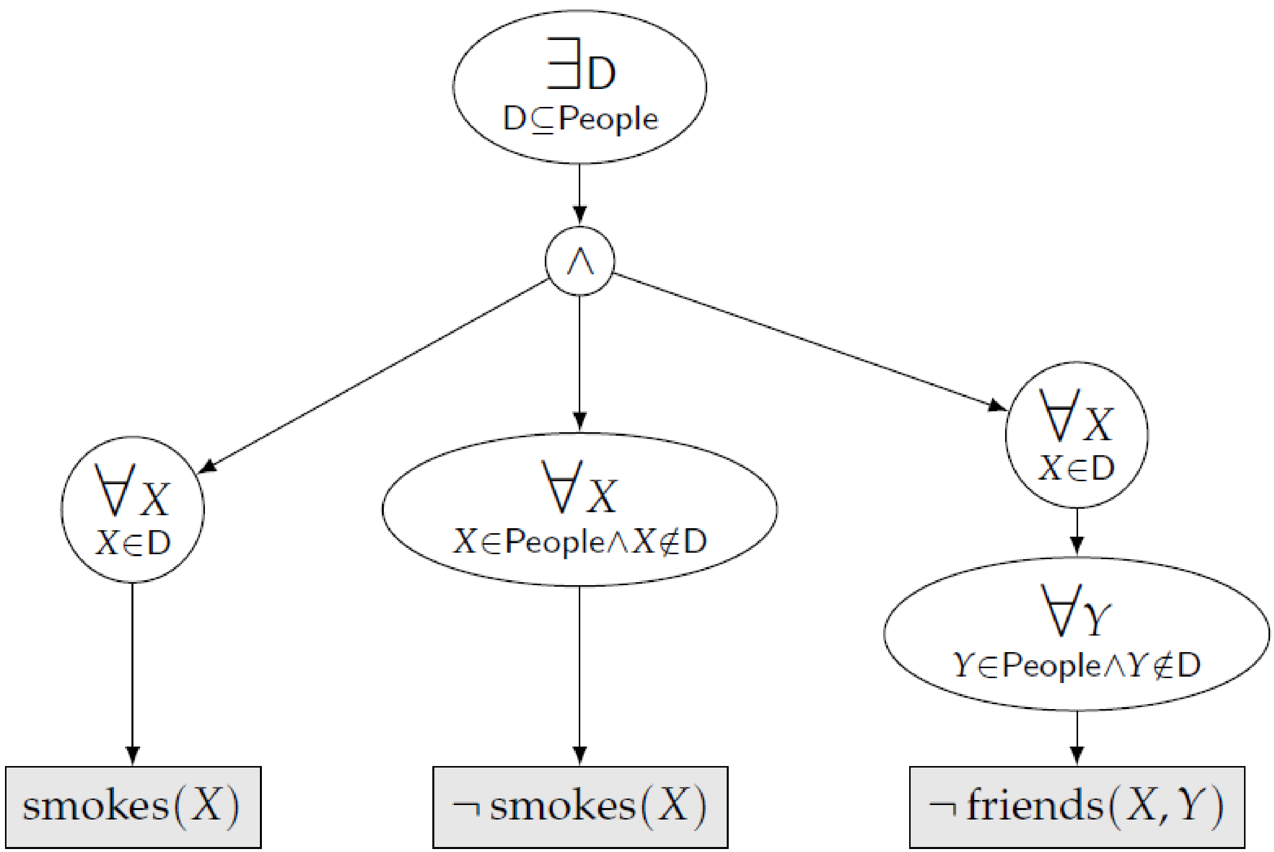
$\Delta = \forall x, y \in \textbf{People}, (\text{Smokes}(x) \wedge \text{Friends}(x,y) \Rightarrow \text{Smokes}(y))$



[Van den Broeck 2013]

Deterministic Decomposable FO NNF

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



[Van den Broeck 2013]

First-Order Model Counting: Example

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

[Van den Broeck 2015]

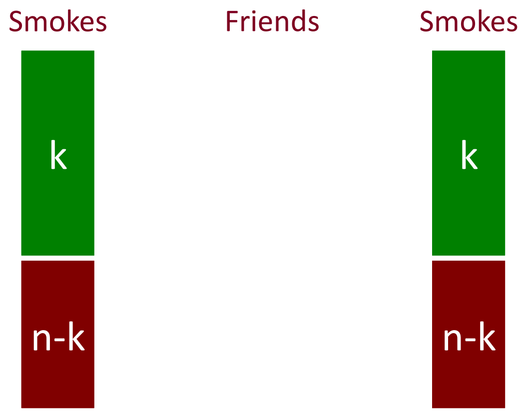
First-Order Model Counting: Example

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

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

Database:

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



[Van den Broeck 2015]

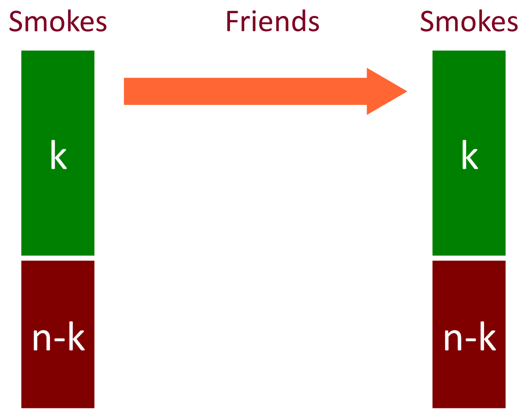
First-Order Model Counting: Example

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

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

Database:

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



[Van den Broeck 2015]

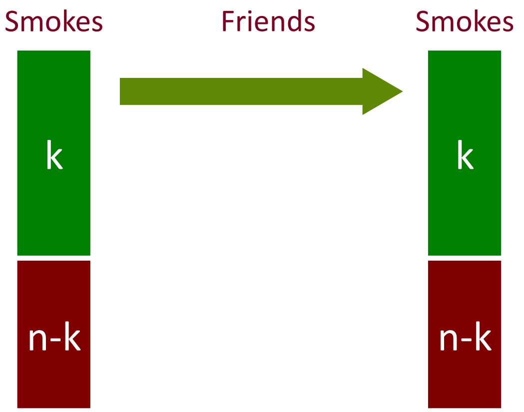
First-Order Model Counting: Example

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

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

Database:

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



[Van den Broeck 2015]

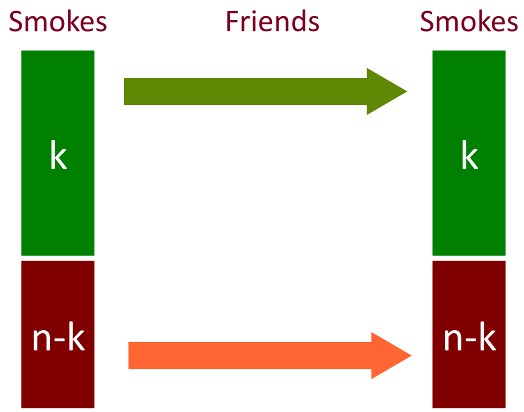
First-Order Model Counting: Example

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

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

Database:

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



[Van den Broeck 2015]

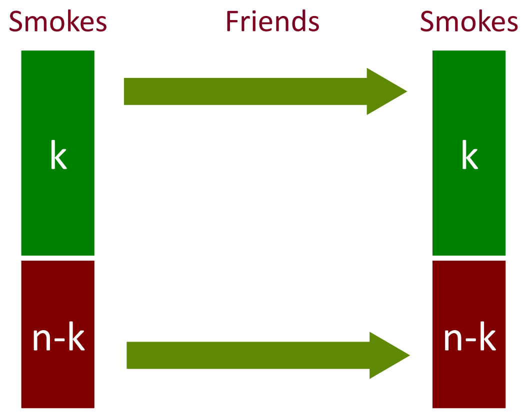
First-Order Model Counting: Example

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

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

Database:

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



[Van den Broeck 2015]

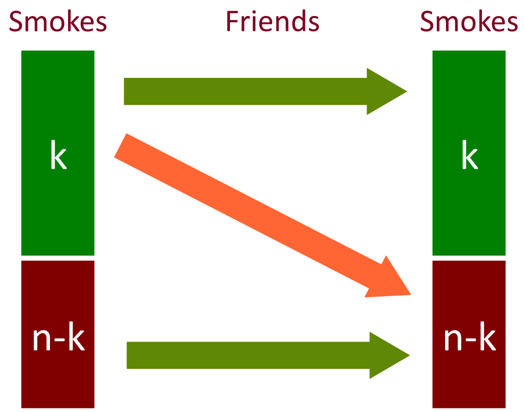
First-Order Model Counting: Example

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

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

Database:

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



[Van den Broeck 2015]

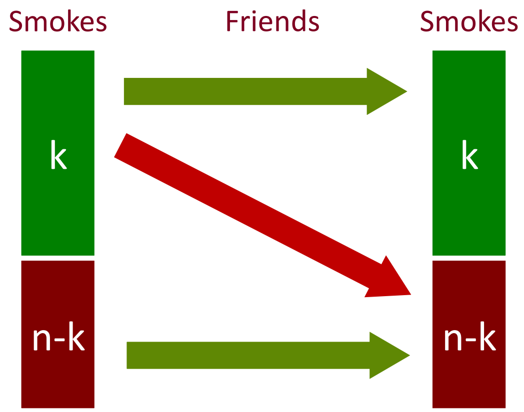
First-Order Model Counting: Example

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

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

Database:

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



[Van den Broeck 2015]

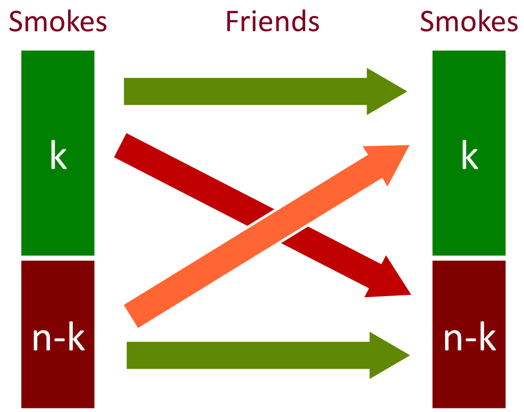
First-Order Model Counting: Example

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

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

Database:

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



[Van den Broeck 2015]

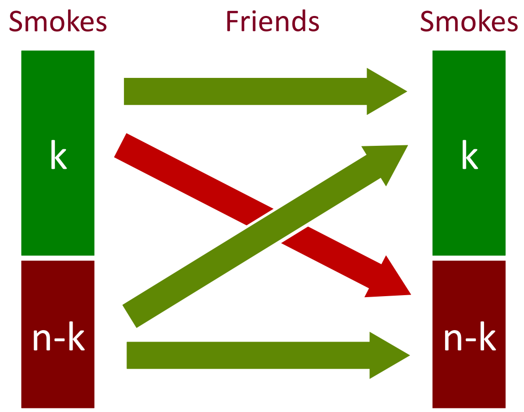
First-Order Model Counting: Example

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

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

Database:

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



[Van den Broeck 2015]

First-Order Model Counting: Example

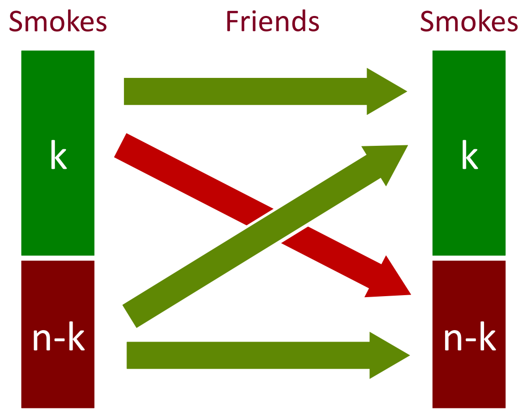
Δ = ∀x ,y ∈ **People**, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))

- If we know **D** 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



[Van den Broeck 2015]

First-Order Model Counting: Example

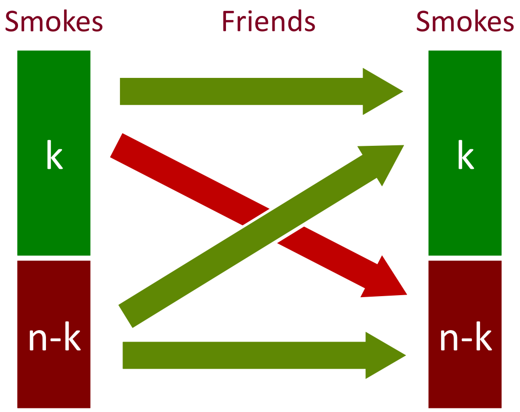
Δ = ∀x ,y ∈ **People**, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))

- If we know **D** 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?

[Van den Broeck 2015]

First-Order Model Counting: Example

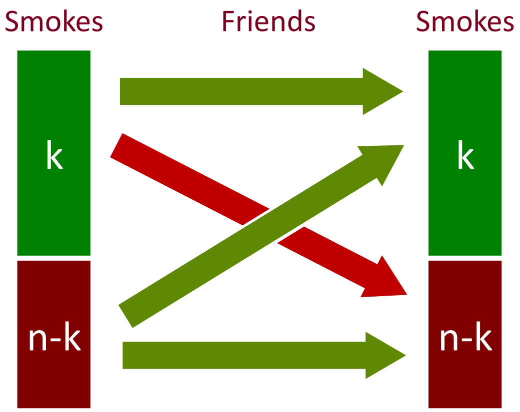
Δ = ∀x ,y ∈ **People**, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))

- If we know **D** 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

[Van den Broeck 2015]

First-Order Model Counting: Example

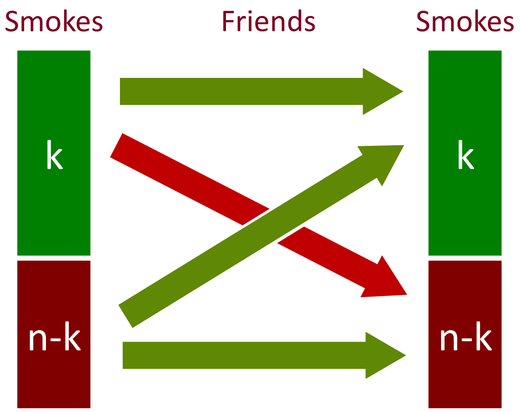
Δ = ∀x ,y ∈ **People**, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))

- If we know **D** 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…

[Van den Broeck 2015]

First-Order Model Counting: Example

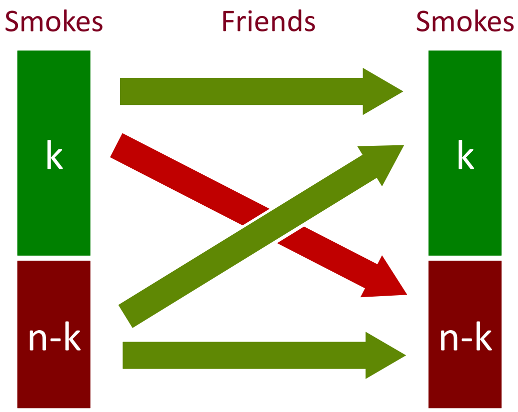
Δ = ∀x ,y ∈ **People**, (Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y))

- If we know **D** 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



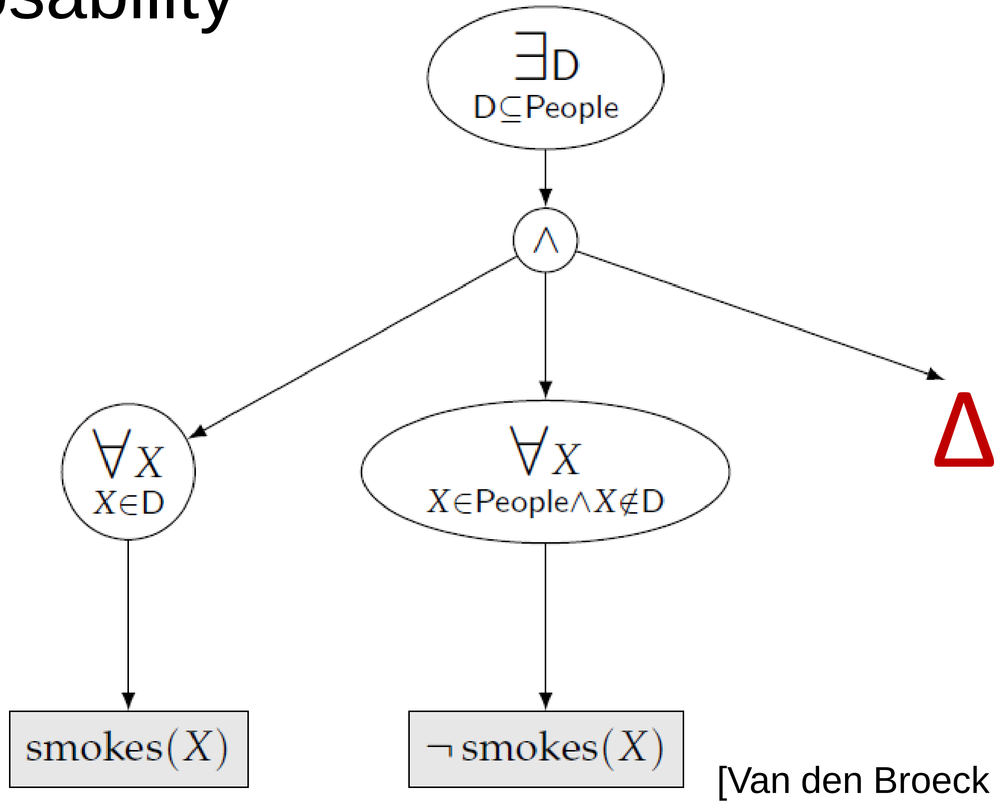
- If we know that there are *k* smokers? → $\binom{n}{k} 2^{n^2-k(n-k)}$ models

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

[Van den Broeck 2015]

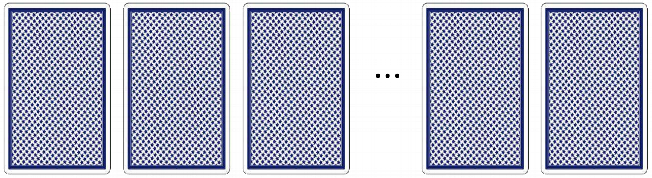
Compilation Rules

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



[Van den Broeck 2013]

Playing Cards Revisited

Let us automate this: 

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

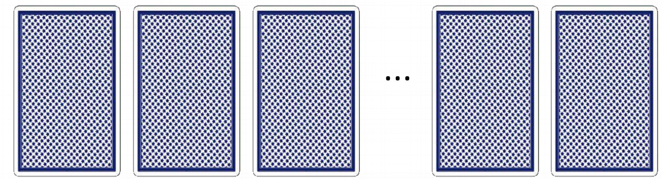
$\forall c, \exists p, Card(p,c)$

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

[Van den Broeck 2015]

Playing Cards Revisited

Let us automate this:



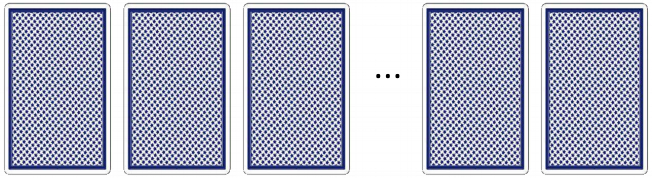
∀p, ∃c, Card(p,c)
∀c, ∃p, Card(p,c)
∀p, ∀c, ∀c', Card(p,c) ∧ Card(p,c') ⇒ c = c'

↓

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

[Van den Broeck 2015]

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'

↓

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

Computed in time polynomial in n

[Van den Broeck 2015]

Perspectives…

What I did not talk about… in KC

- Other queries and transformations
  (see Dan Olteanu poster)
- Other KC languages
  (FO-AODD)
- KC for logic programs
  (see Vlasselaer poster)

[Gogate 2010, Vlasselaer 2015]

What I did not talk about…in FOMC

- WFOMC for probabilistic databases
  (see Gribkoff poster)
- WFOMC for probabilistic programs
  (see Vlasselaer poster)
- Complexity theory (data or domain)
  - PTime domain complexity for 2-var fragment
  - $\#P_1$ domain complexity for some 3-var CNFs

[Gribkoff 2014, Vlasselaer 2015, Beame 2015]

What I did not talk about…in FO

- Very related problems
  - Lifted inference in SRL
- Very related applications
  - Approximate lifted inference in Markov Logic
  - Learn Markov logic networks
- Classical first-order reasoning
  - Answer set programming,
  - SMT,
  - Theorem proving

[Kersting 2011]

Format for First-Order BeyondNP

- DIMACS contributed to SAT success
- Goals
  - Trivial to parse
  - Captures MLNs, Prob. Programs, Prob. DBs
  - *Not* a powerful representation language
- FO-CNF format under construction
- Vibhav?

```
p fo-cnf 2 1
d people 1000
r Friends(people,people)
r Smokes(people)
-Smokes(x) -Friends(x,y) Smokes(y)
w Friends 3.5 1.2
w Smokes -0.5 4
```

Calendar

At **IJCAI** in **New York** on **July 9-11**

- StarAI 2016 (http://www.starai.org/2016/)

  Sixth International Workshop on
  Statistical Relational AI

- IJCAI Tutorial

  "*Lifted Probabilistic Inference in Relational Models*" with Dan Suciu

Conclusions

- FOMC is BeyondNP reduction target
- Existing solvers inadequate

  Exponential speedups from FO solvers
- FOKC is Elegant, more than FOMC
- Intersection of communities
  - Statistical relational learning (lifted inference)
  - Probabilistic databases
  - Automated reasoning (you!)

References

- Chavira, Mark, and Adnan Darwiche. "On probabilistic inference by weighted model counting." Artificial Intelligence 172.6 (2008): 772-799.
- Sang, Tian, Paul Beame, and Henry A. Kautz. "Performing Bayesian inference by weighted model counting." AAAI. Vol. 5. 2005.
- Fierens, Daan, et al. "Inference and learning in probabilistic logic programs using weighted boolean formulas." Theory and Practice of Logic Programming 15.03 (2015): 358-401.
- Van den Broeck, Guy, et al. "Lifted probabilistic inference by first-order knowledge compilation." Proceedings of IJCAI. AAAI Press, 2011.
- Gogate, Vibhav, and Pedro Domingos. "Probabilistic theorem proving." UAI (2012).
- Belle, Vaishak, Andrea Passerini, and Guy Van den Broeck. "Probabilistic inference in hybrid domains by weighted model integration." Proceedings of 24th International Joint Conference on Artificial Intelligence (IJCAI). 2015.
- Gribkoff, Eric, Guy Van den Broeck, and Dan Suciu. "Understanding the complexity of lifted inference and asymmetric weighted model counting." UAI (2014).
- Van den Broeck, Guy. Lifted inference and learning in statistical relational models. Diss. Ph. D. Dissertation, KU Leuven, 2013.

References

- Chavira, Mark, Adnan Darwiche, and Manfred Jaeger. "Compiling relational Bayesian networks for exact inference." International Journal of Approximate Reasoning 42.1 (2006): 4-20.
- Van den Broeck, Guy. "Towards high-level probabilistic reasoning with lifted inference." (2015).
- Niepert, Mathias, and Guy Van den Broeck. "Tractability through exchangeability: A new perspective on efficient probabilistic inference." AAAI (2014).
- Darwiche, Adnan, and Pierre Marquis. "A knowledge compilation map." Journal of Artificial Intelligence Research 17.1 (2002): 229-264.
- Gogate, Vibhav, and Pedro M. Domingos. "Exploiting Logical Structure in Lifted Probabilistic Inference." Statistical Relational Artificial Intelligence. 2010.
- Vlasselaer, Jonas, et al. "Anytime inference in probabilistic logic programs with Tp-compilation." Proceedings of IJCAI. (2015).
- Beame, Paul, et al. "Symmetric weighted first-order model counting." Proceedings of the 34th ACM Symposium on Principles of Database Systems. ACM, 2015.
- Kersting, Kristian. "Lifted Probabilistic Inference." ECAI. 2012.