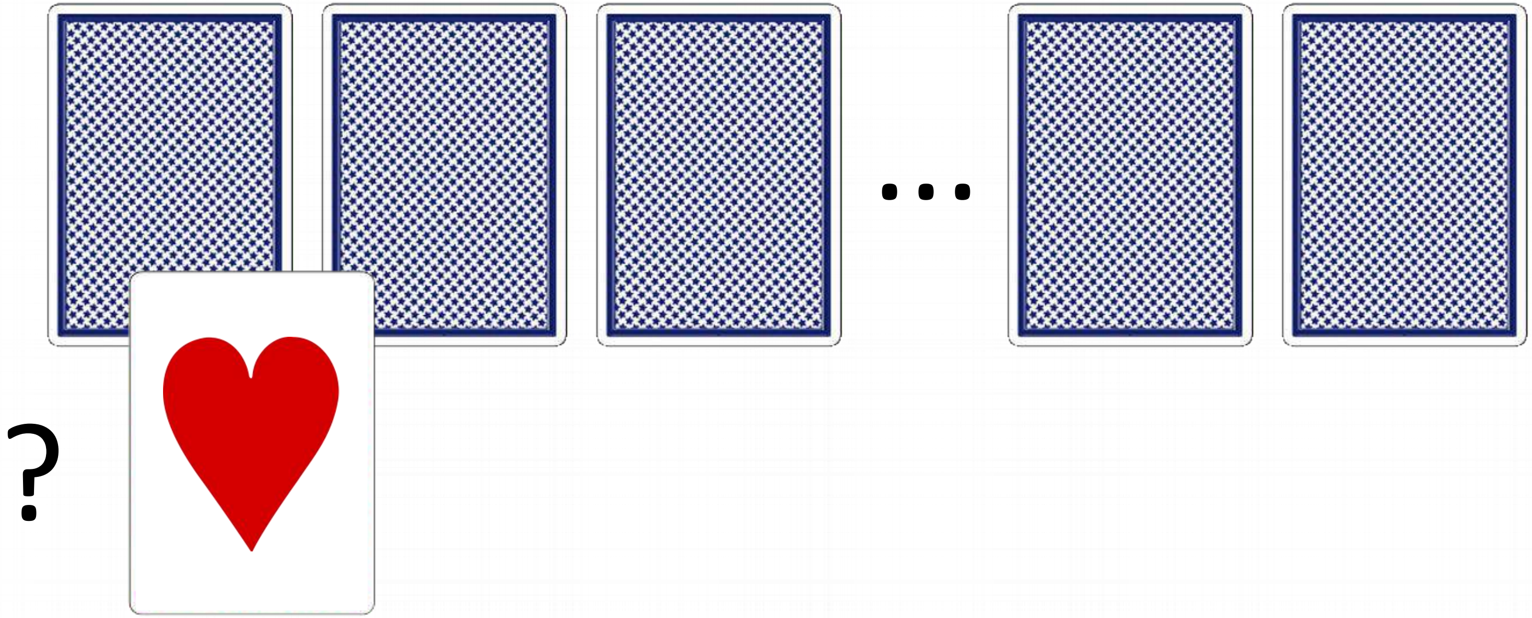
First-Order Knowledge Compilation for Probabilistic Reasoning

Guy Van den Broeck

based on joint work with Adnan Darwiche, Dan Suciu, and many others

MOTIVATION 1

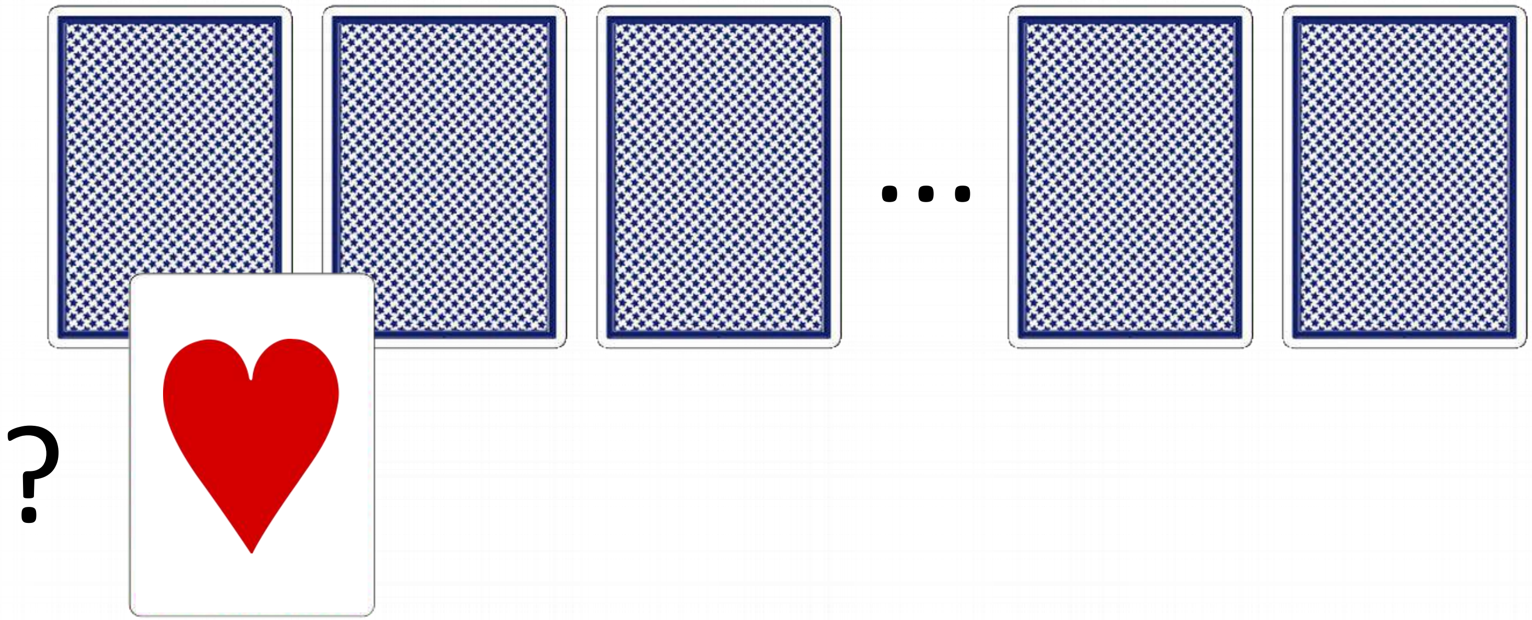
A Simple Reasoning Problem



Probability that Card1 is Hearts?

[Van den Broeck; AAAI-KRR'15]

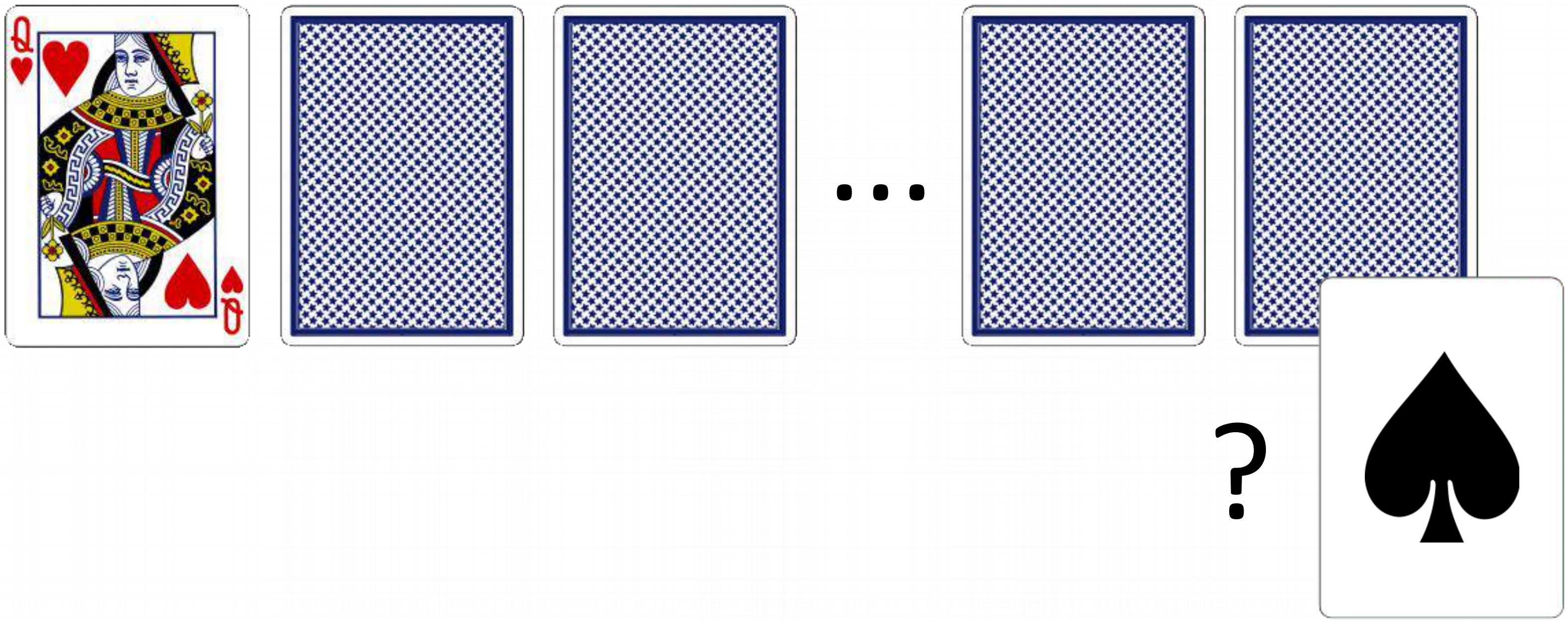
A Simple Reasoning Problem



Probability that Card1 is Hearts? 1/4

[Van den Broeck; AAAI-KRR’15]

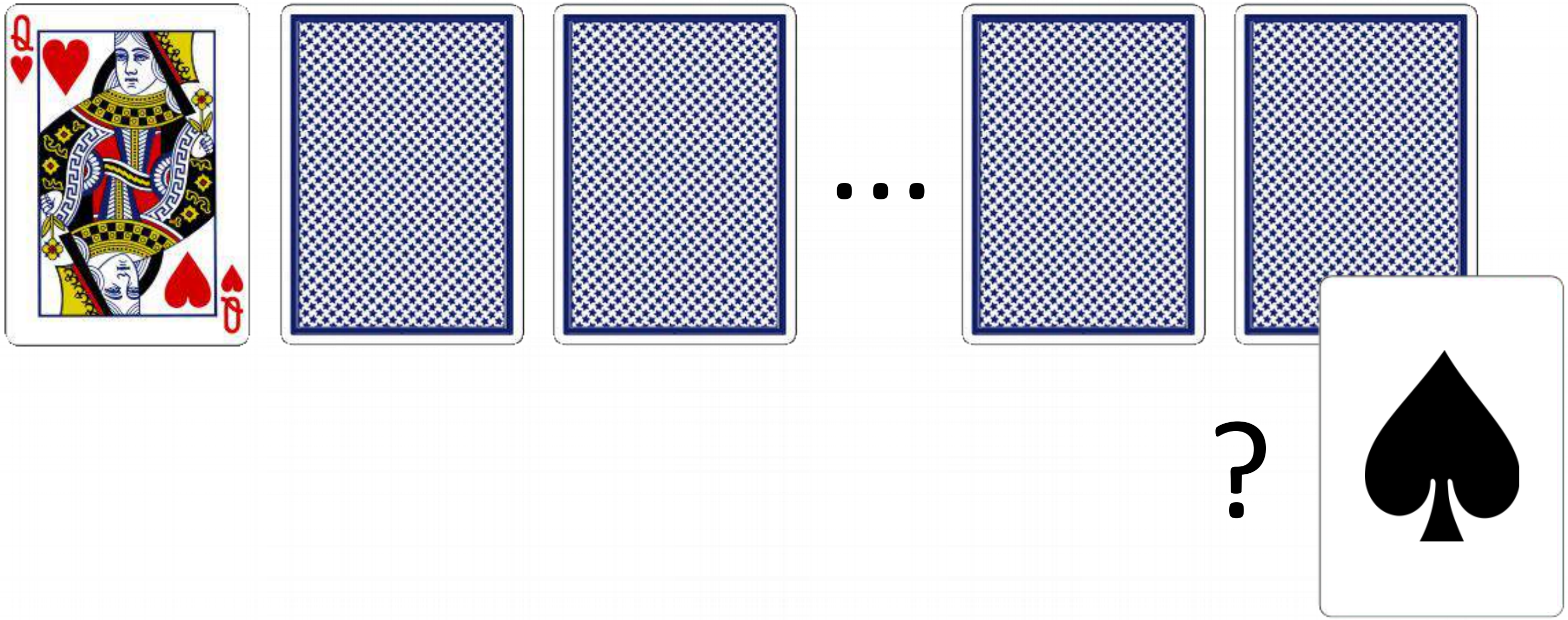
A Simple Reasoning Problem



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. CNF encoding for deck of cards
2. Compile to tractable knowledge base (e.g., d-DNNF)
3. Condition on observations/questions

   "Card1 is hearts"

4. Model counting

Automated Reasoning

Let us automate this:

1. CNF encoding for deck of cards
2. Compile to tractable knowledge base (e.g., d-DNNF)
3. Condition on observations/questions

   "Card1 is hearts"
4. Model counting

A typical BeyondNP pipeline!

Automated Reasoning

Let us automate this:

1. **CNF encoding for deck of cards**

Card(p1,c1) v Card(p1,c2) v …
Card(p1,c1) v Card(p2,c1) v …
¬Card(p1,c1) v ¬Card(p1,c2)
¬Card(p1,c2) v ¬Card(p1,c3)
…
¬Card(p2,c1) v ¬Card(p2,c2)
…

Automated Reasoning

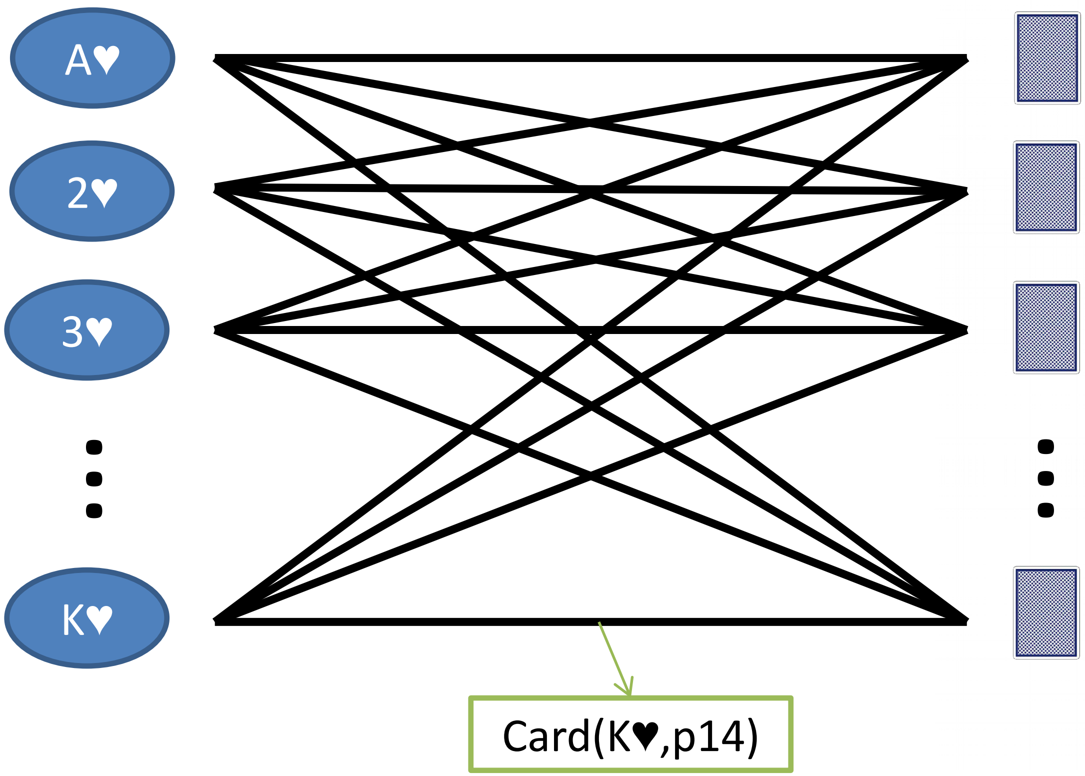
Let us automate this:

1. CNF encoding for deck of cards
2. **Compile to tractable knowledge base (e.g., d-DNNF)**
3. Condition on observations/questions

   "Card1 is hearts"
4. Model counting

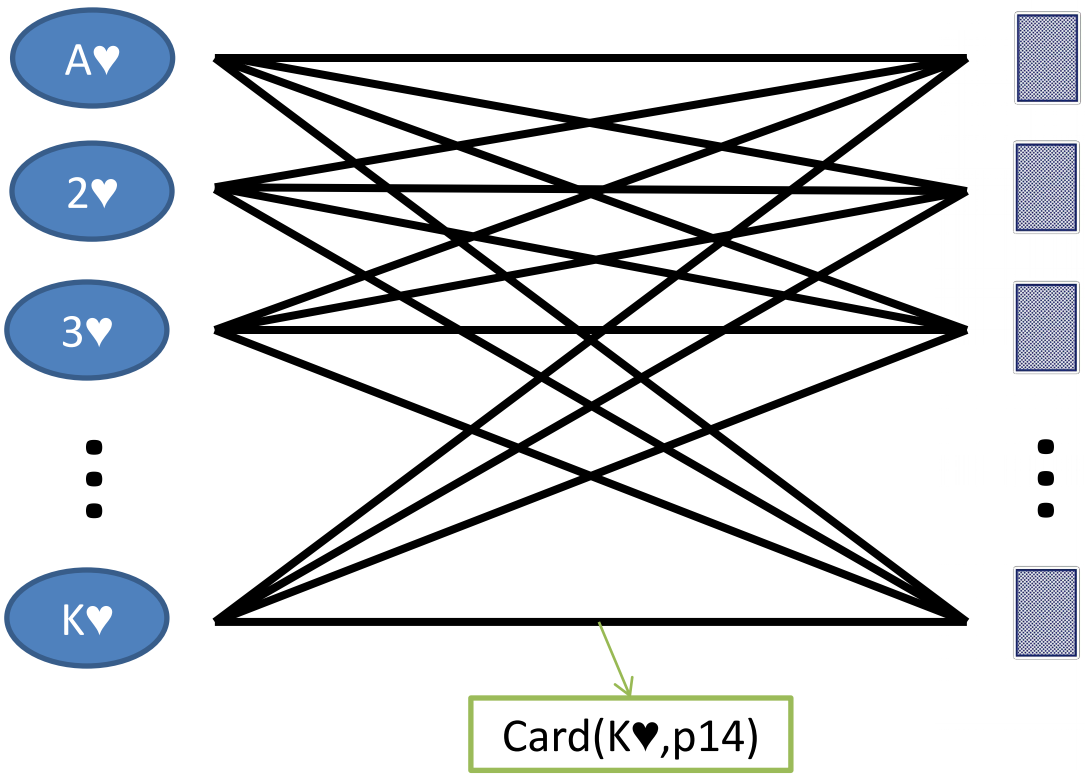
Which language to choose?
Cards problem is easy: we want to be polynomial.

Deck of Cards Graphically



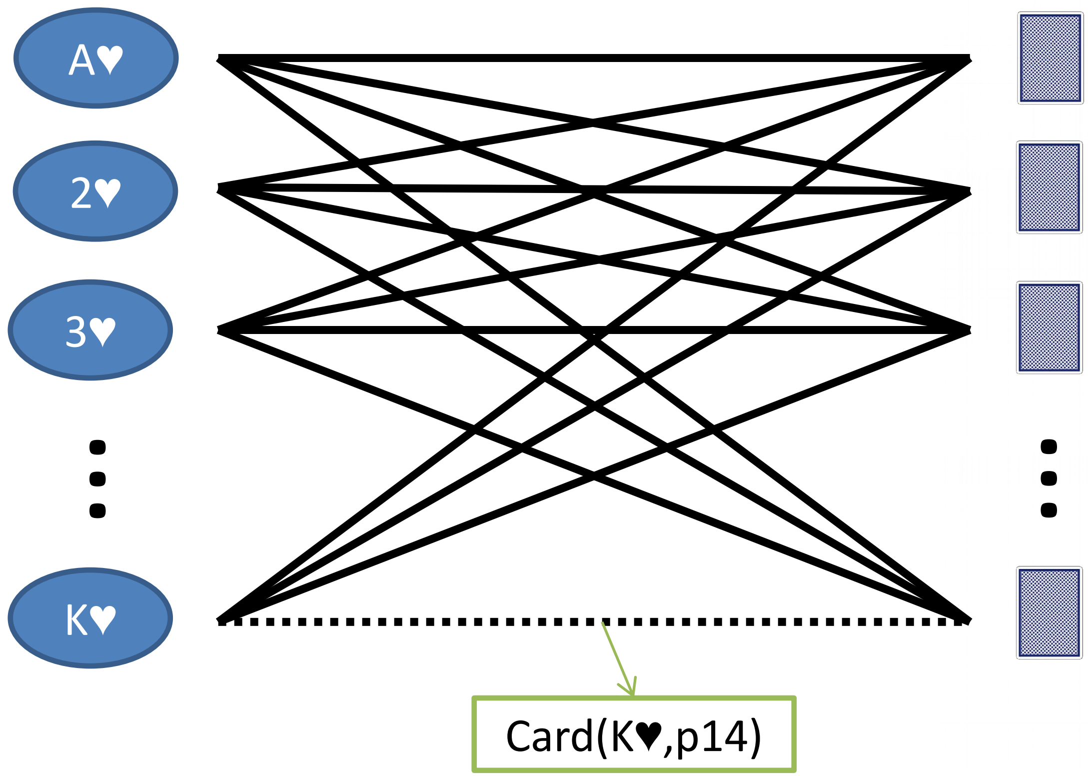
2. Compile to tractable knowledge base
3. Condition on observations/questions
4. Model counting

Deck of Cards Graphically



2. **Compile to tractable knowledge base**
3. Condition on observations/questions
4. Model counting

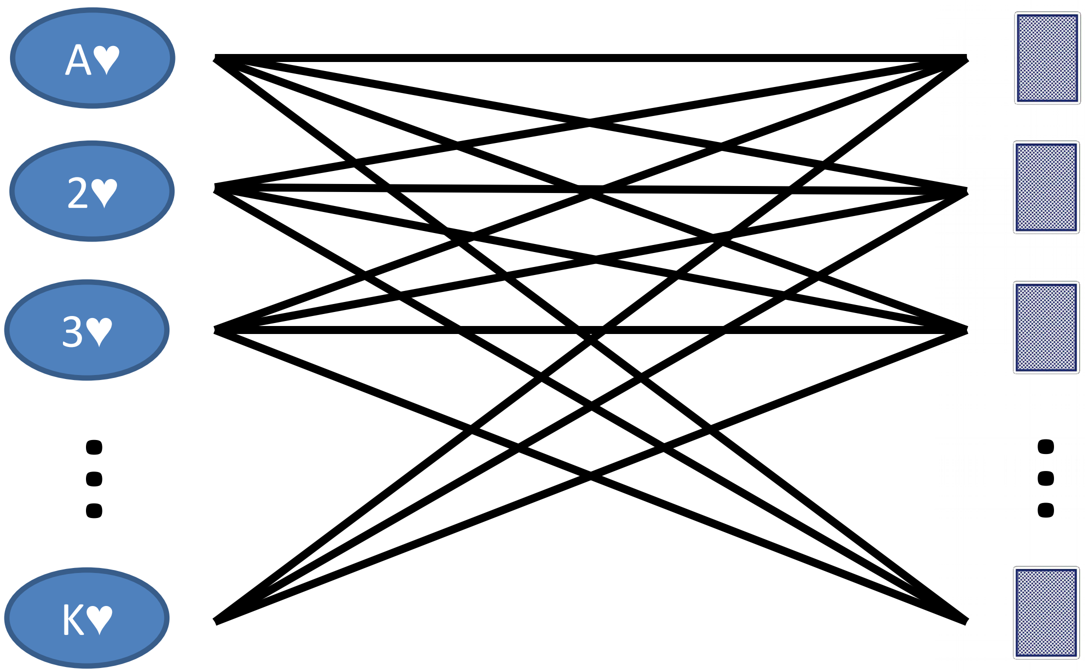
Deck of Cards Graphically



2. Compile to tractable knowledge base
3. **Condition on observations/questions**
4. Model counting

¬ Card(K♥,p14)

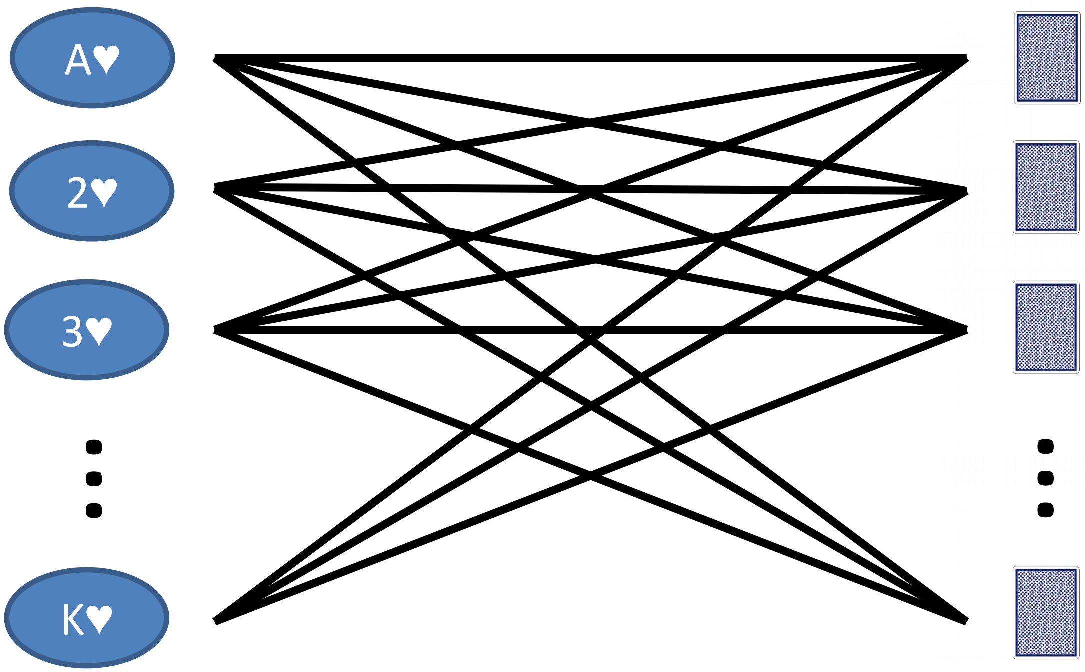
Deck of Cards Graphically



2. Compile to tractable knowledge base
3. **Condition on observations/questions**
4. Model counting

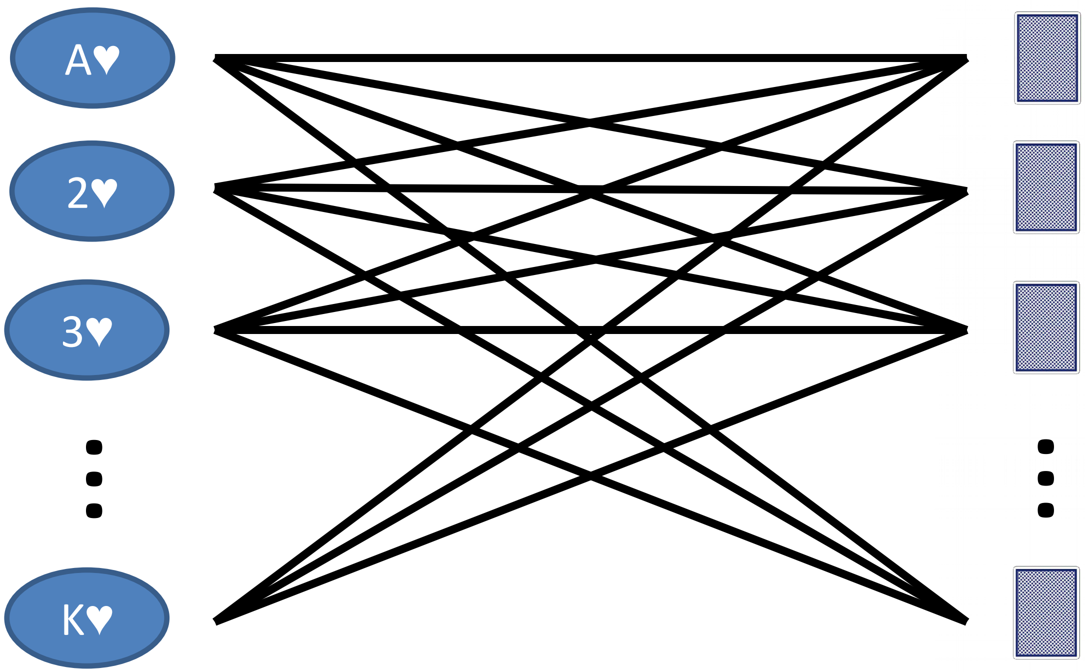
¬ Card(K♥,p14)

Deck of Cards Graphically



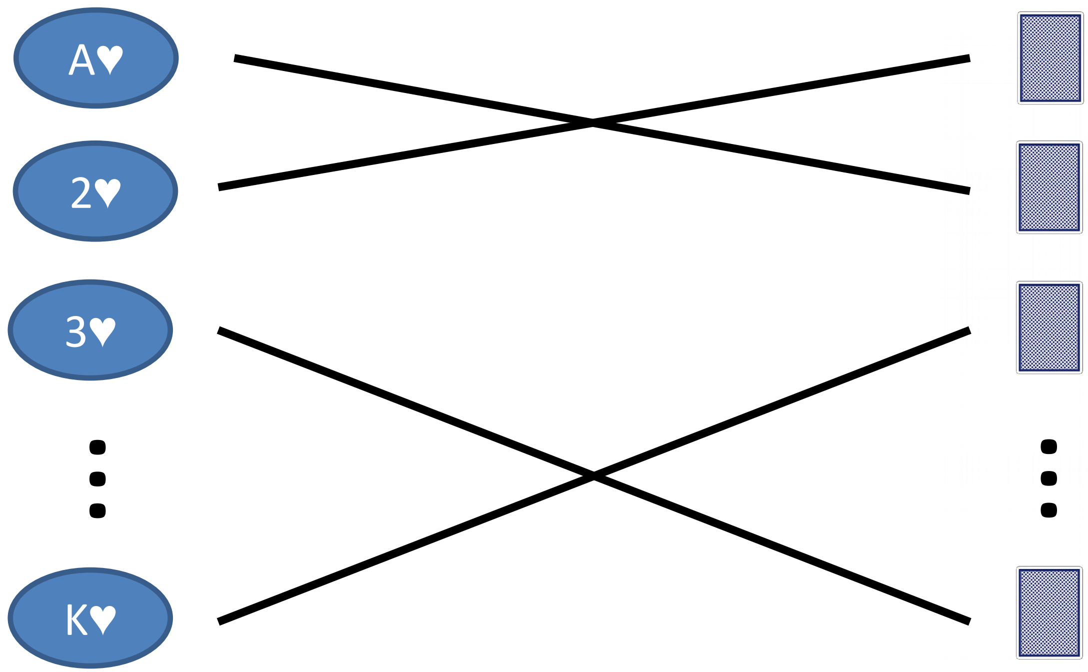
2. Compile to tractable knowledge base
3. Condition on observations/questions
4. **Model counting**

Deck of Cards Graphically



2. Compile to tractable knowledge base
3. Condition on observations/questions
4. **Model counting: How many *perfect matchings*?**

Deck of Cards Graphically



2. Compile to tractable knowledge base
3. Condition on observations/questions
4. **Model counting: How many *perfect matchings*?**

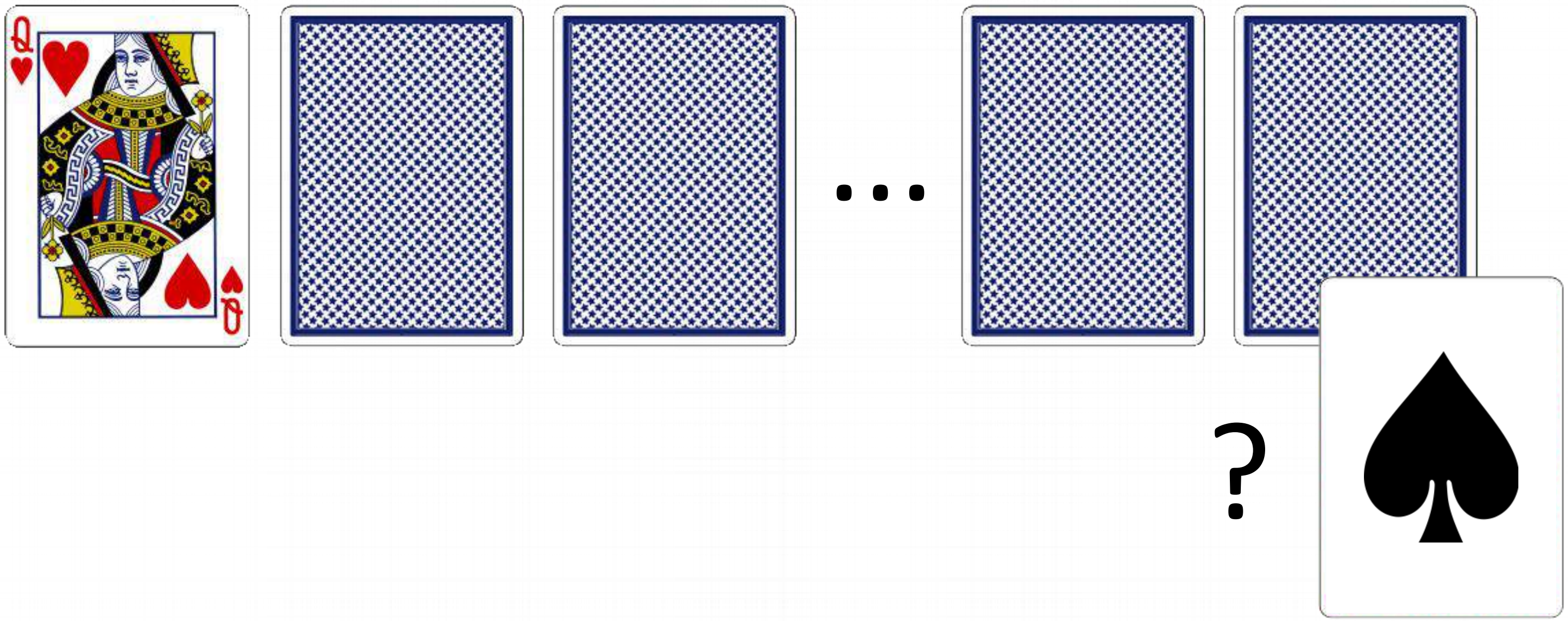
Observations

- Deck of cards = complete bigraph
- CD = removing edges in bigraph

  Encode any bigraph in cards problem
- CT = counting perfect matchings
- Problem is #P-complete!

No language with CD and CT can represent the cards problem compactly, unless P=NP.

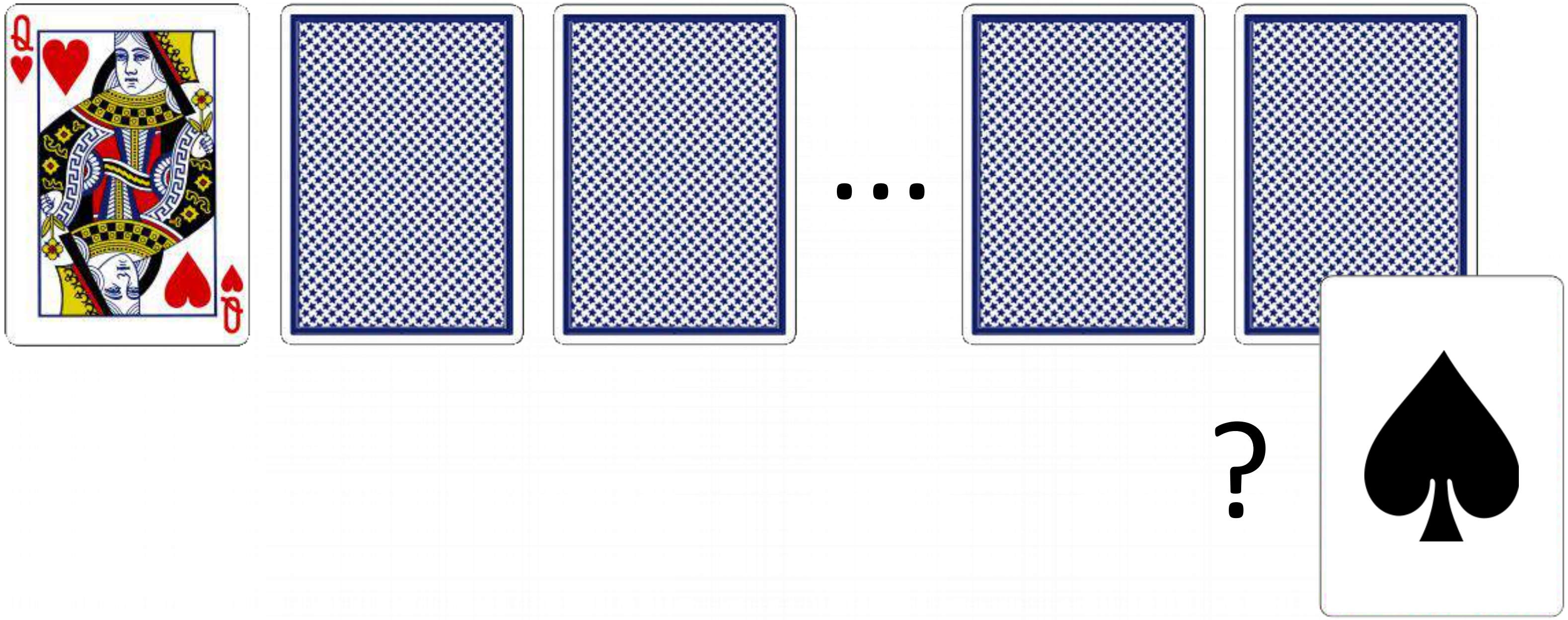
What's Going On Here?



Probability that Card52 is Spades given that Card1 is QH?

[Van den Broeck; AAAI-KRR’15]

What's Going On Here?

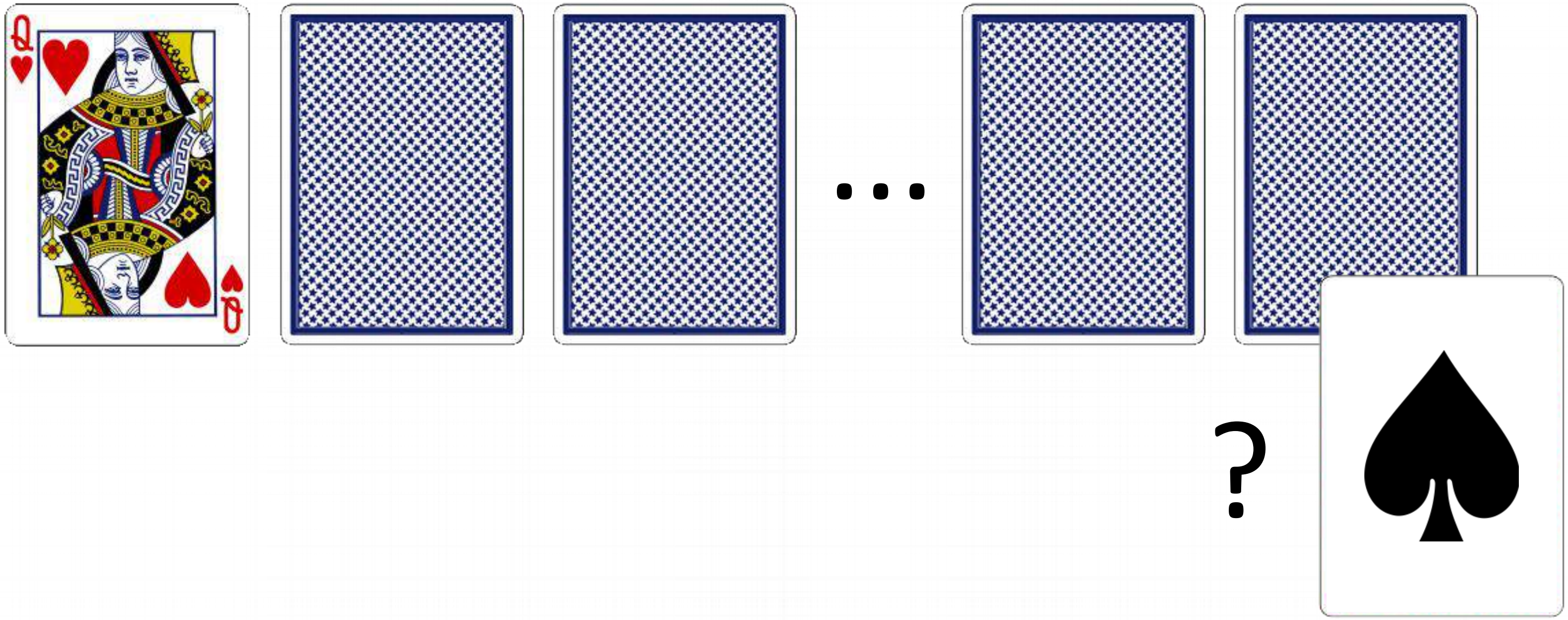


Probability that Card52 is Spades given that Card1 is QH?

13/51

[Van den Broeck; AAAI-KRR'15]

What's Going On Here?

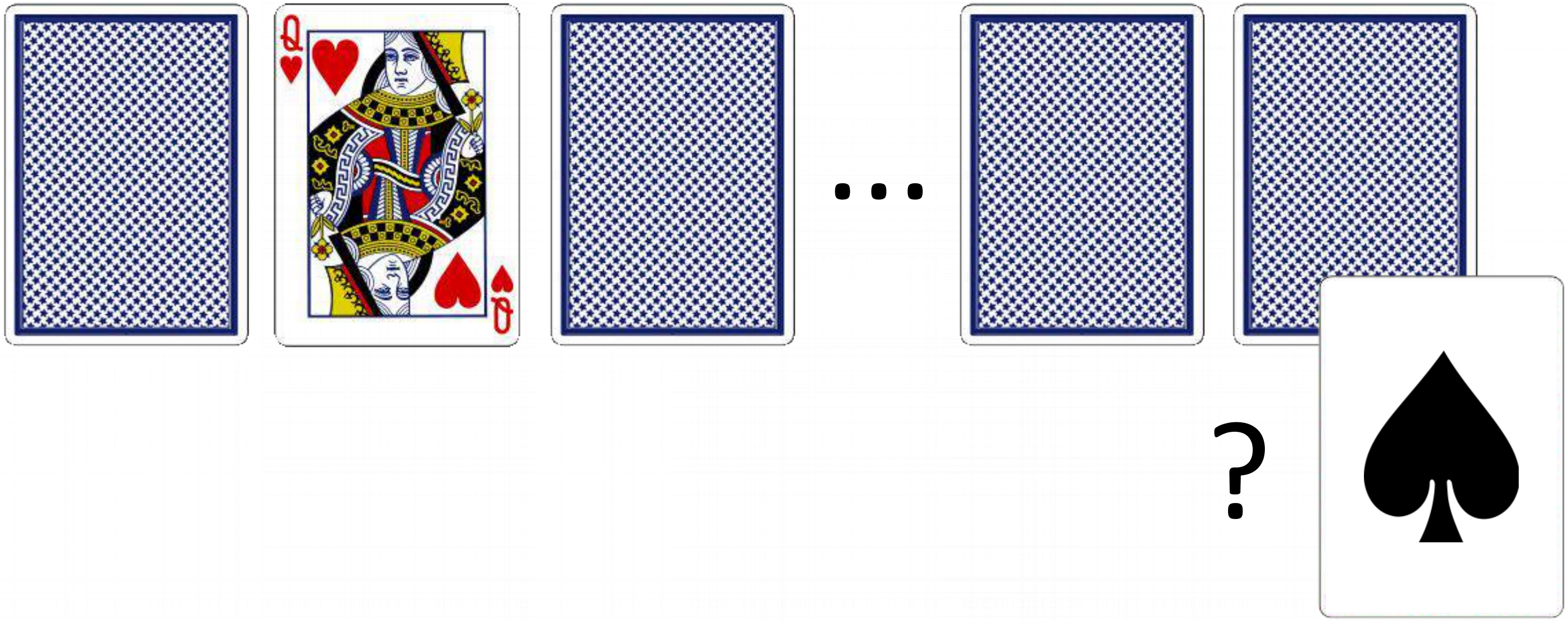


Probability that Card52 is Spades given that Card1 is QH?

13/51

[Van den Broeck; AAAI-KRR'15]

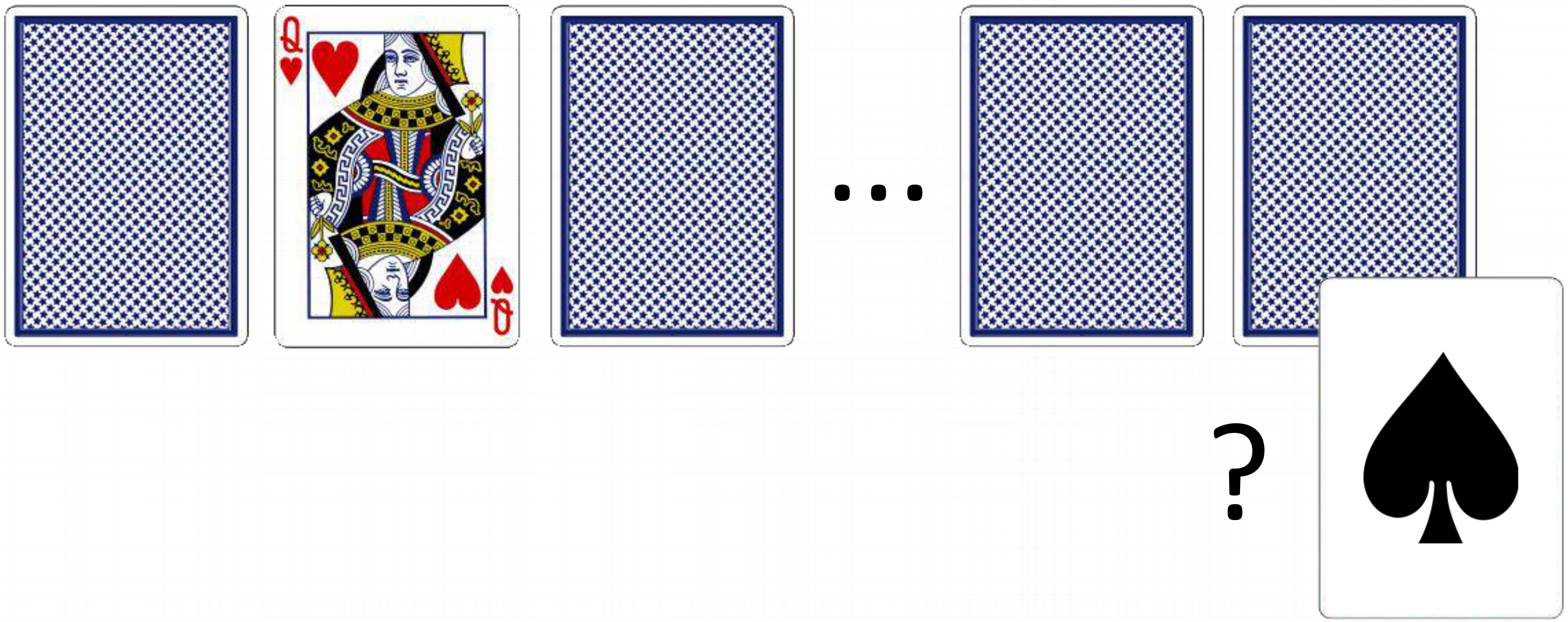
What's Going On Here?



Probability that Card52 is Spades given that Card2 is QH?

[Van den Broeck; AAAI-KRR'15]

What's Going On Here?

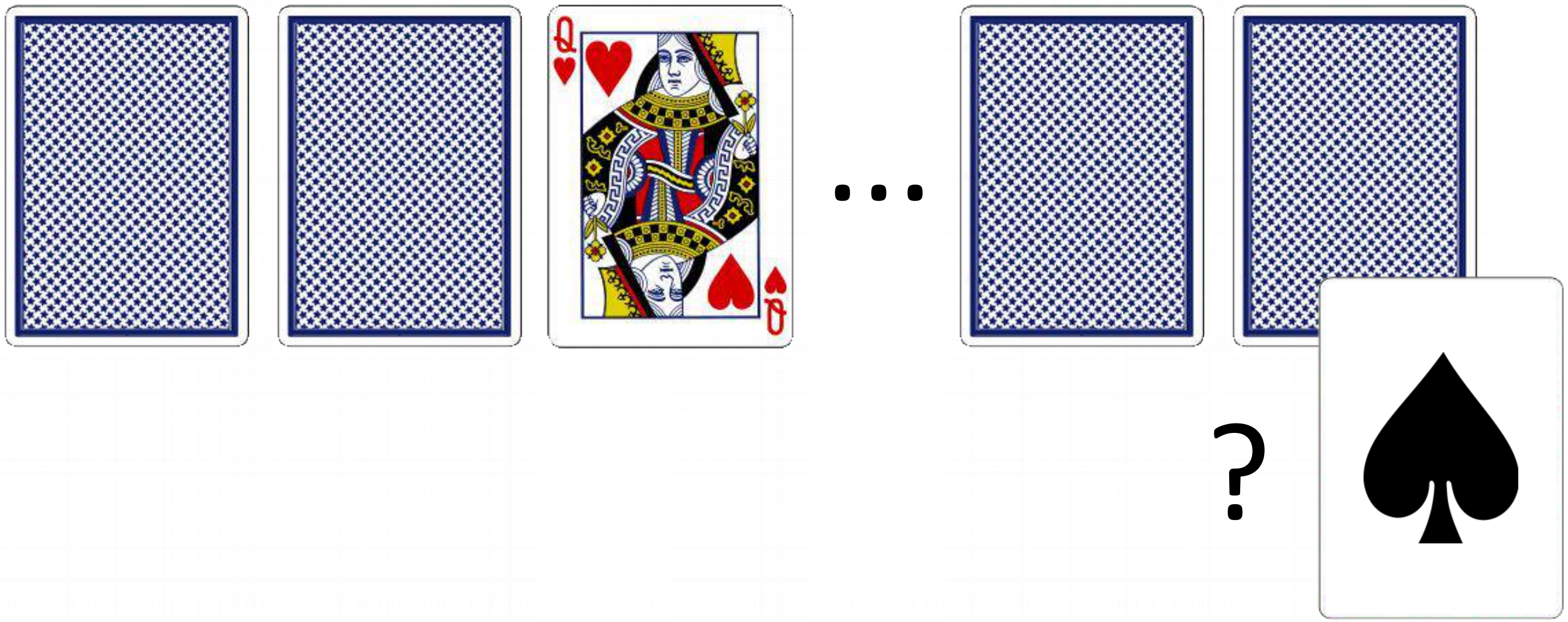


Probability that Card52 is Spades given that Card2 is QH?

13/51

[Van den Broeck; AAAI-KRR'15]

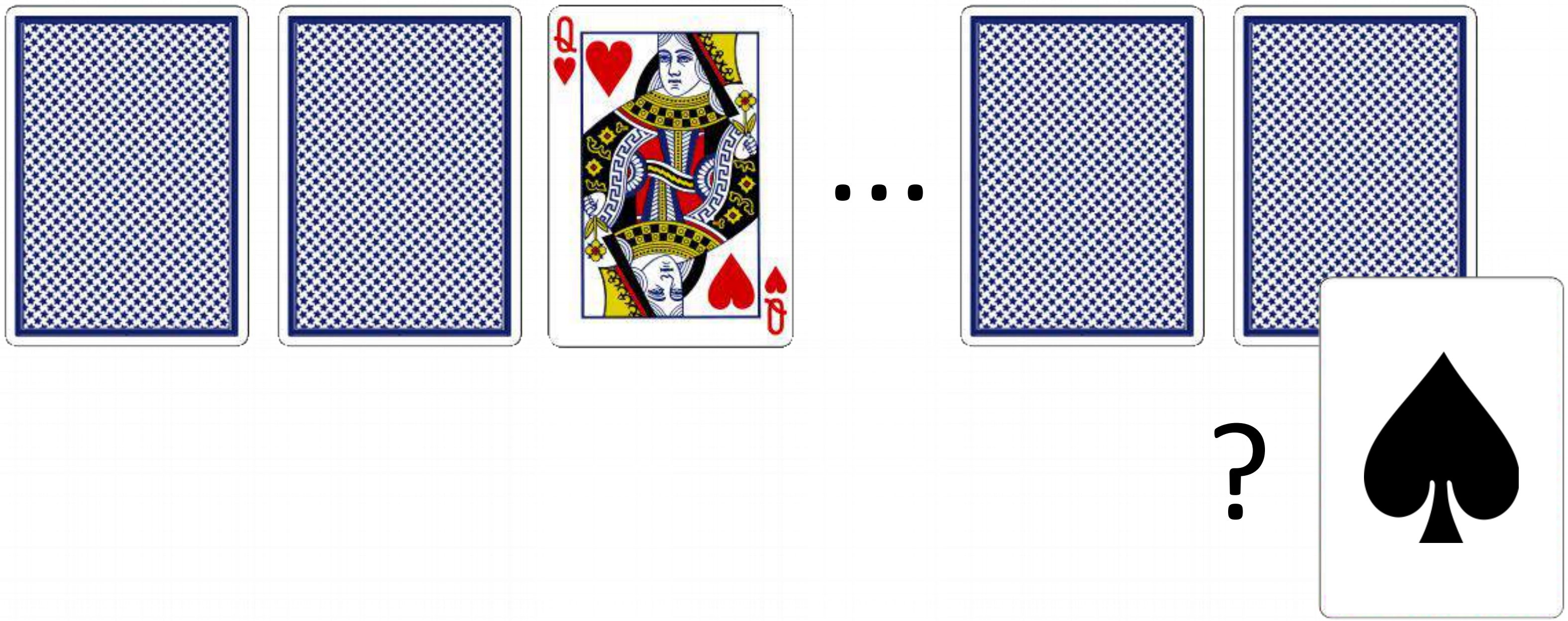
What's Going On Here?



Probability that Card52 is Spades given that Card3 is QH?

[Van den Broeck; AAAI-KRR’15]

What's Going On Here?

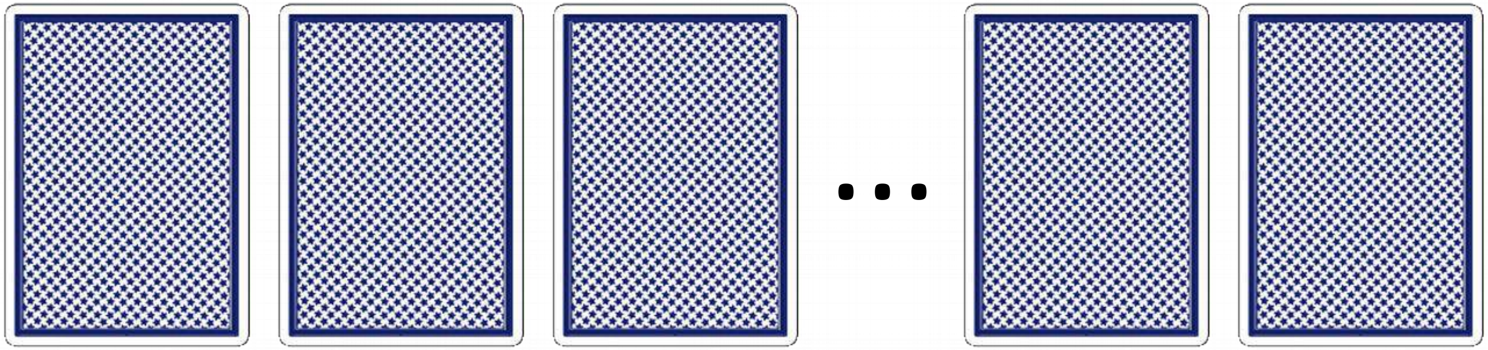


Probability that Card52 is Spades given that Card3 is QH?

13/51

[Van den Broeck; AAAI-KRR'15]

Tractable Reasoning

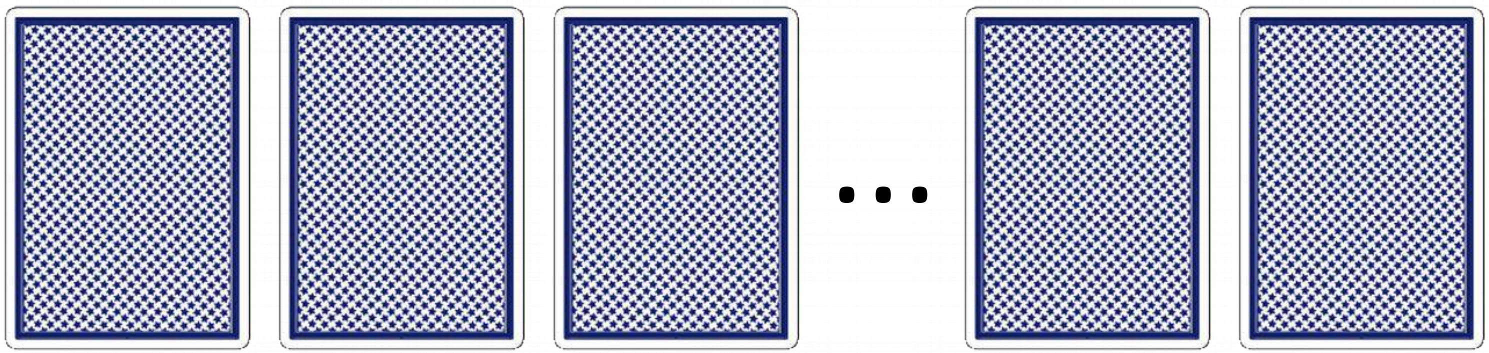


What's going on here?

Which property makes reasoning tractable?

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

Tractable Reasoning



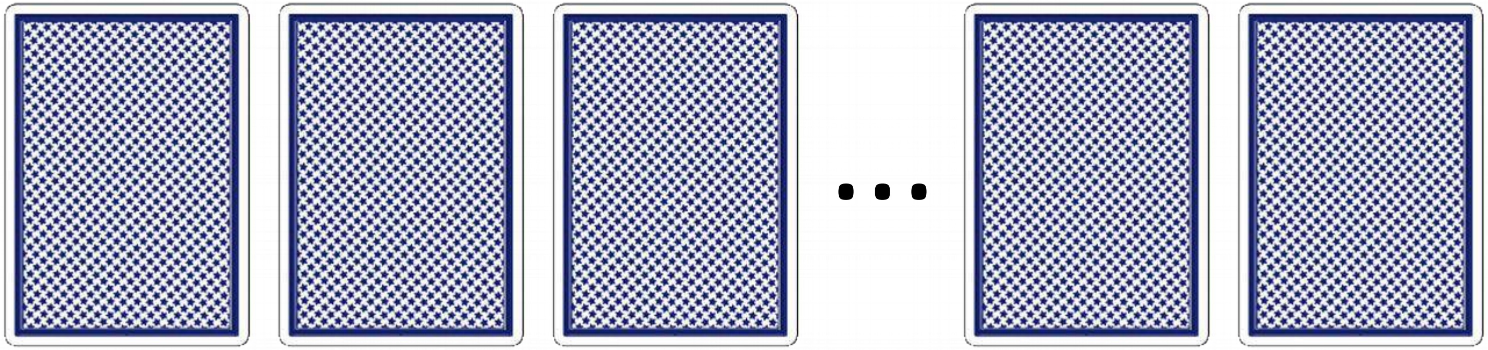
What's going on here?

Which property makes reasoning tractable?

- 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/FO** model

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

- **First-Order Knowledge Compilation**

MOTIVATION 2

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

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)

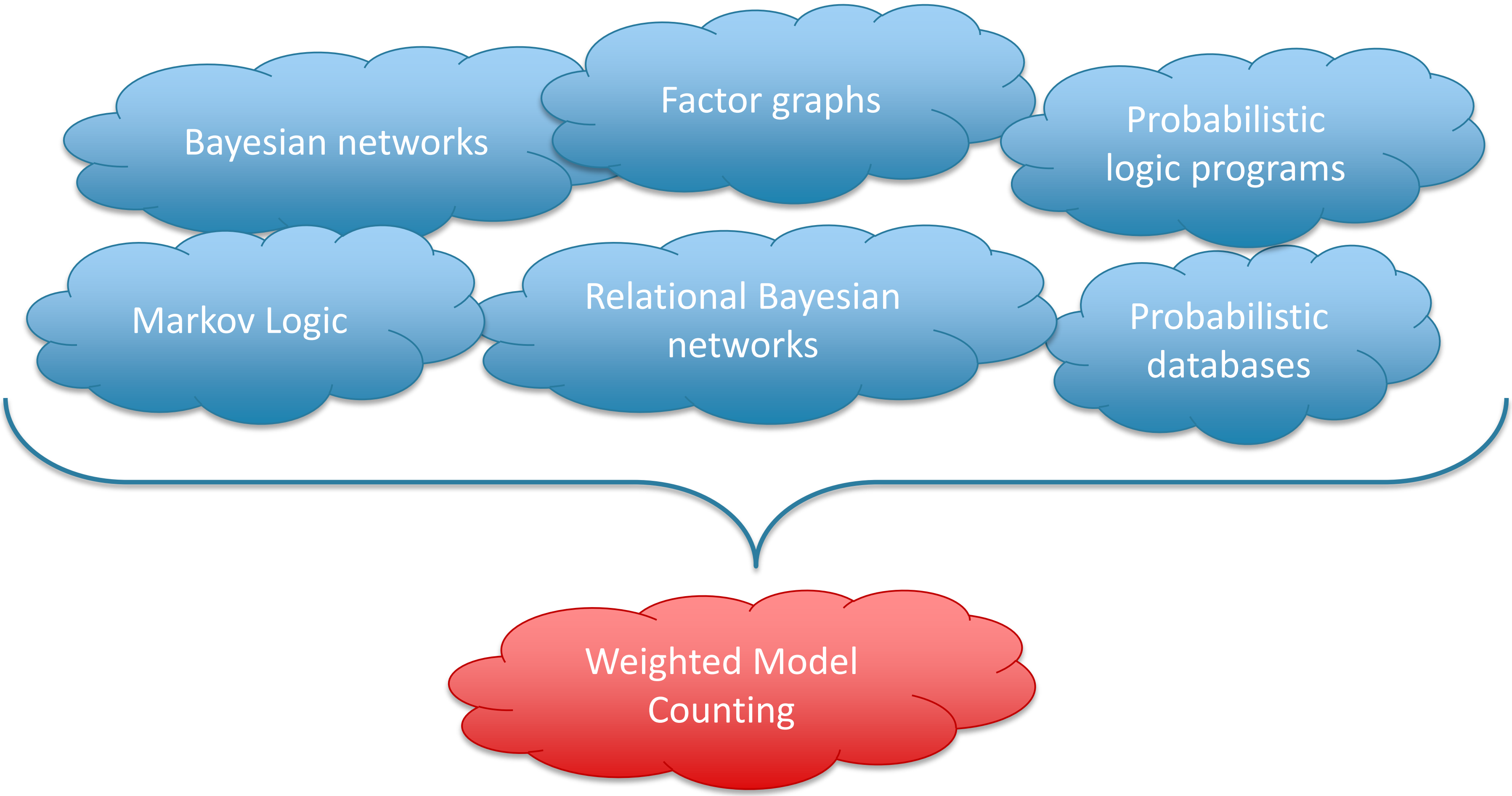
w(R)=1
w(¬R)=2
w(C)=3
w(¬C)=5

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



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

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

$\Delta = \forall d\ (Rain(d) \Rightarrow Cloudy(d))$

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**}

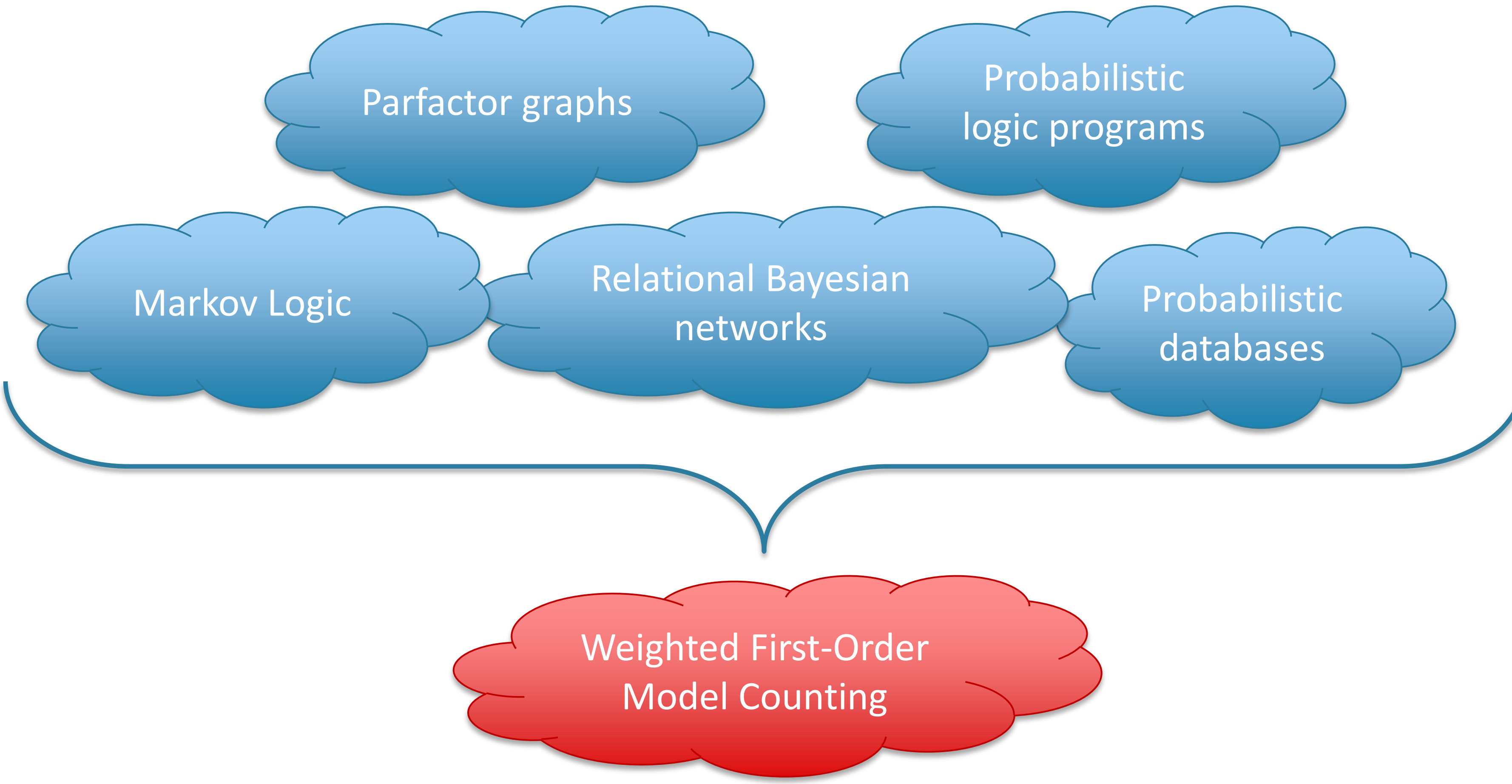
w(R)=1
w(¬R)=2
w(C)=3
w(¬C)=5

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

Assembly language for high-level probabilistic reasoning and learning



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

Statistical Relational Learning

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, Γ(**x**))
- **Weight of a world** = product of w, for all rules (w, Γ(**x**)) and groundings Γ(**a**) that hold in the world

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

Applications: large probabilistic KBs

FO NNF SYNTAX

First-Order Knowledge Compilation

- Input: Sentence in FOL
- Output: Representation tractable for some class of queries.
- In this work:
  - Function-free FOL
  - Model counting in NNF tradition
- Some pre-KC-map work:
  - FO Horn clauses
  - FO BDDs

Alphabet

- FOL
  - Predicates/relations: Friends
  - Object names: x, y, z
  - Object variables: X, Y, Z
  - Symbols classical FOL (∀, ∃, ∧, ∨, ¬,…)
- Group logic
  - Group variables: **X, Y, Z**
  - Symbols from basic set theory (e.g., ∪, ∩, ∈, ⊆, {, }, complement).

Syntax

- Object terms: X, alice, bob
- Group terms : **X,** {alice,bob}, **X** ∪ **Y**
- Atom: Friends(alice,X)
- Formulas:
  - (α), ¬α, α ∨ β, and α ∧ β
  - ∀X ∈ **G**, α and ∃X ∈ **G**, α
  - ∀**X** ⊆ **G**, α and ∃**X** ⊆ **G**, α
- Group logic syntactic sugar:
  - P(**G**) is ∀X ∈ **G,** P (X)
  - $\overline{\text{P}}$(**G**) is ∀X ∈ **G,** ¬P (X)

Examples:

- $\forall X \in \mathbf{G}, Y \in \{alice, bob\},$
  $\text{Enemies}(X, Y)$
  $\Rightarrow \neg\text{Friends}(X, Y) \wedge \neg\text{Friends}(Y, X)$

- $\forall X \in \mathbf{G}, Y \in \mathbf{G},$
  $\text{Smokes}(X) \wedge \text{Friends}(X, Y) \Rightarrow \text{Smokes}(Y)$

- $\exists \mathbf{G} \subseteq \{alice, bob\}, \text{Smokes}(\mathbf{G}) \wedge \overline{\text{H}}\text{ealthy}(\mathbf{G})$

Semantics

- Template language for propositional logic
- Grounding a sentence: gr(α)
  - Replace ∀ by ∧
  - Replace ∃ by ∨
  - End result: ground sentence = propositional logic
- Grounding is polynomial in group sizes when no $\forall \mathbf{X} \subseteq \mathbf{G}$ or $\exists \mathbf{X} \subseteq \mathbf{G}$

  Important for polytime reduction to NNF circuits

Decomposability

- Conjunction: $\alpha(X,\mathbf{G}) \land \beta(X,\mathbf{G})$

  For any substitution X=c and **G**=g, we have that $gr(\alpha(c,g)) \land gr(\beta(c,g))$ is decomposable

  Meaning: α and β can never talk about the same ground atoms

- Quantifier: $\forall Y \in \mathbf{G}, \alpha(Y)$

  For any two a,b ∈ **G,** we have that $gr(\alpha(a)) \land gr(\alpha(b))$ is decomposable

Determinism

- Disjunction: $\alpha(X,\mathbf{G}) \vee \beta(X,\mathbf{G})$

  For any substitution $X=c$ and $\mathbf{G}=g$, we have that $gr(\alpha(c,g)) \vee gr(\beta(c,g))$ is deterministic

  Meaning: $\alpha \wedge \beta$ is UNSAT

- Quantifier: $\exists Y \in \mathbf{G}, \alpha(Y)$

  For any two $a,b \in \mathbf{G}$, we have that
  $gr(\alpha(a)) \vee gr(\alpha(b))$ is decomposable

Group Quantifiers

- Decomposability: $\forall \mathbf{X} \subseteq \mathbf{G}, \alpha(\mathbf{X})$

  For any two $\mathbf{A},\mathbf{B} \subseteq \mathbf{G}$, we have that $gr(\alpha(\mathbf{A})) \vee gr(\alpha(\mathbf{B}))$ is decomposable

- Determinism: $\exists \mathbf{X} \subseteq \mathbf{G}, \alpha(\mathbf{X})$

  For any two $\mathbf{A},\mathbf{B} \subseteq \mathbf{G}$, we have that $gr(\alpha(\mathbf{A})) \vee gr(\alpha(\mathbf{B}))$ is deterministic

Automorphism

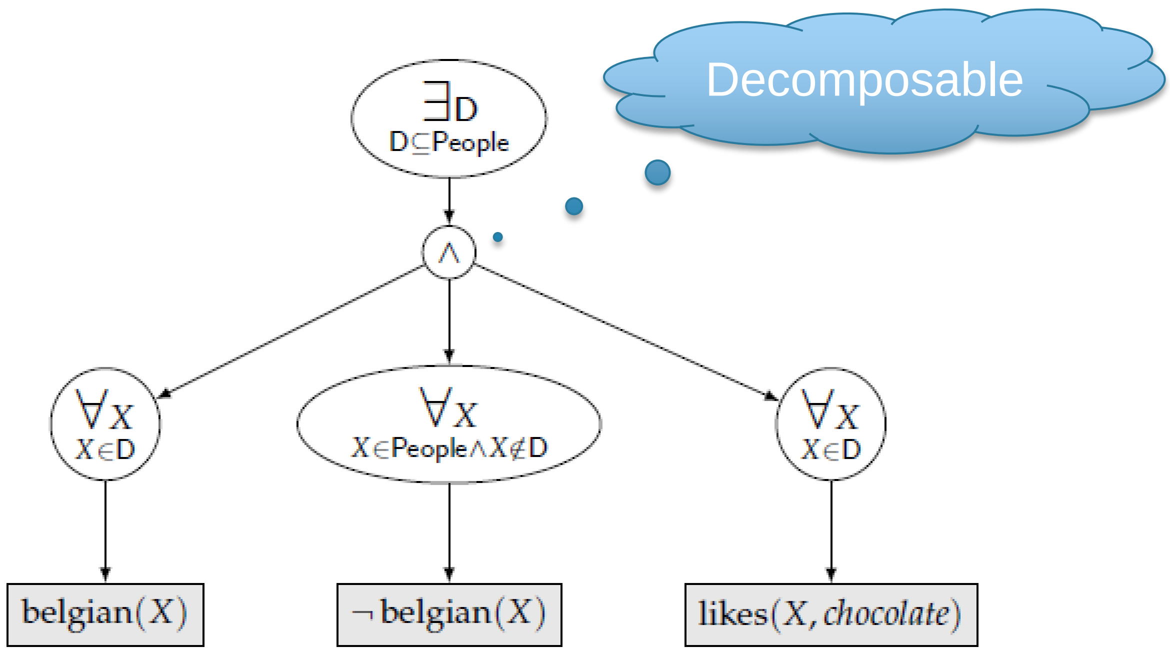
- Object permutation $\sigma : D \rightarrow D$ is a one-to-one mapping from objects to objects.
- Permuting α using σ replaces o in α by σ(o).
- Sentences α and β are p-equivalent iff α is equivalent to an object permutation of β.

  Smokes(alice) and Smokes(bob) are p-equivalent
- Group quantifiers: $\forall \mathbf{X} \subseteq \mathbf{G}, \alpha(\mathbf{X})$ or $\exists \mathbf{X} \subseteq \mathbf{G}, \alpha(\mathbf{X})$

  Are ***automorphic*** iff for any two **A**,**B** ⊆ **G** s.t. |A|=|B|, gr(α(**A**)) and gr(α(**B**)) are p-equivalent

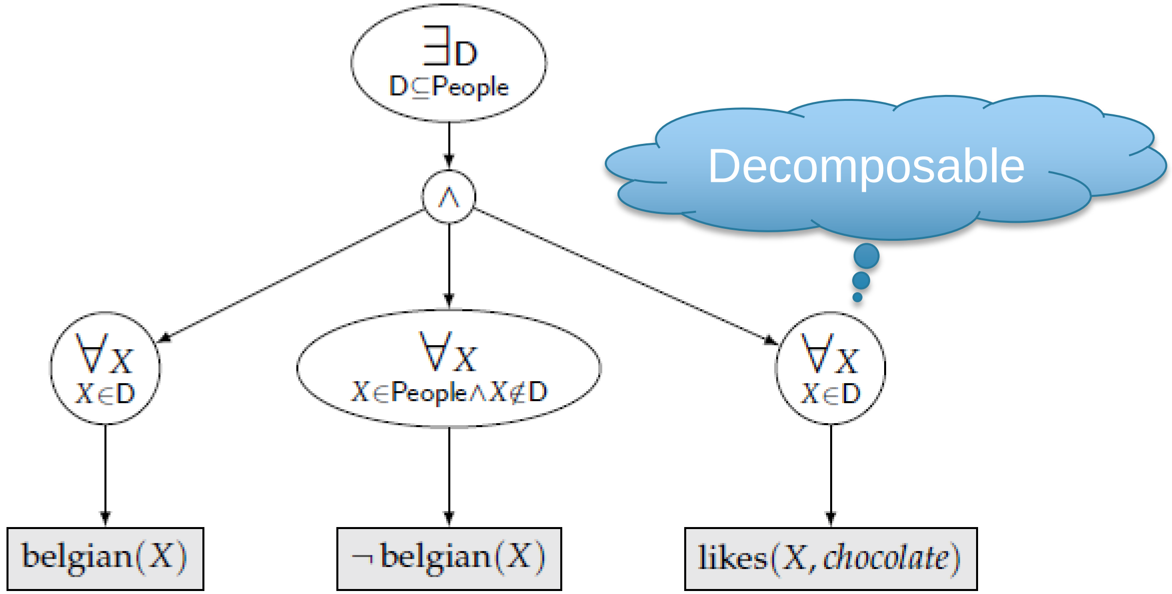
First-Order NNF

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



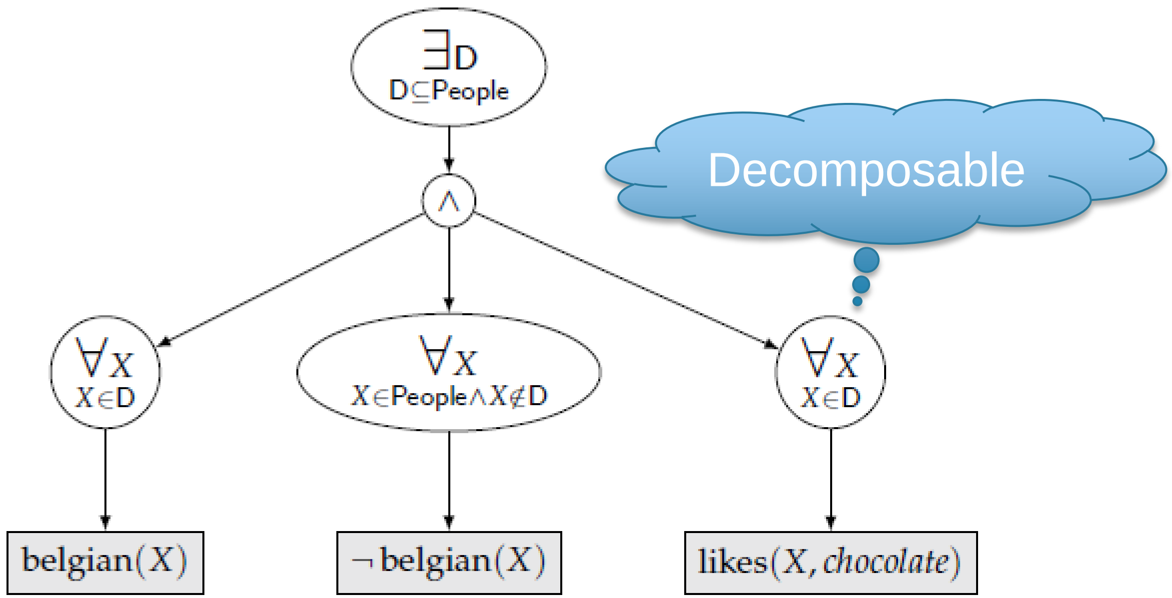
First-Order NNF

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



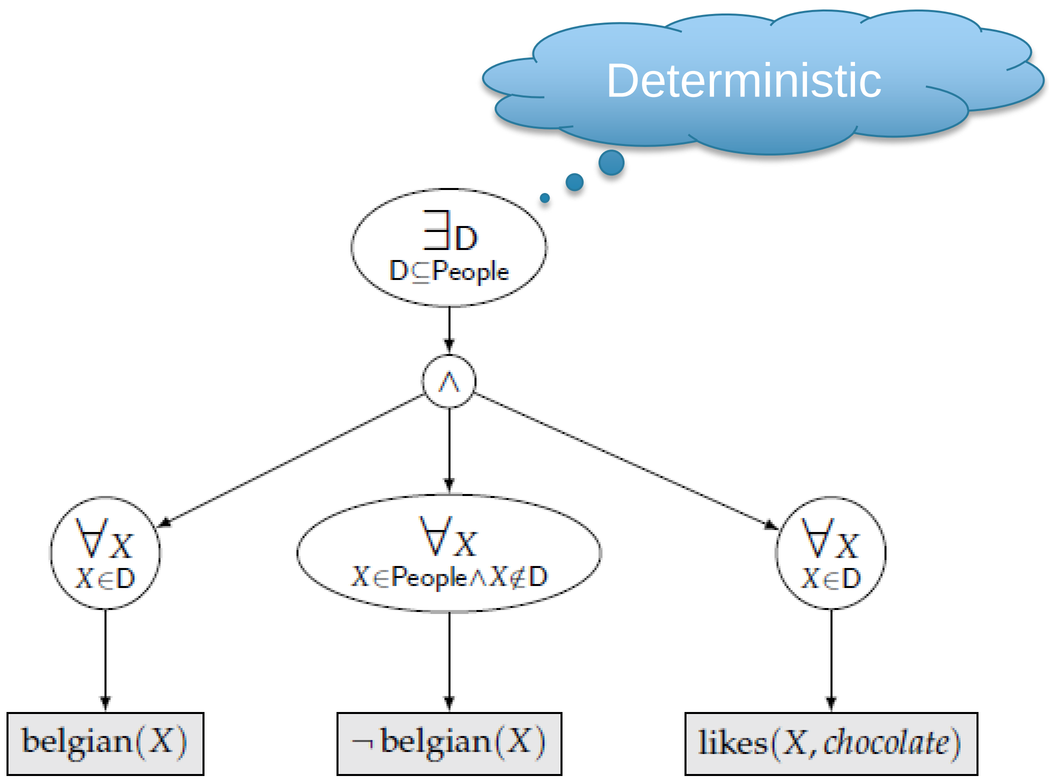
First-Order DNNF

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



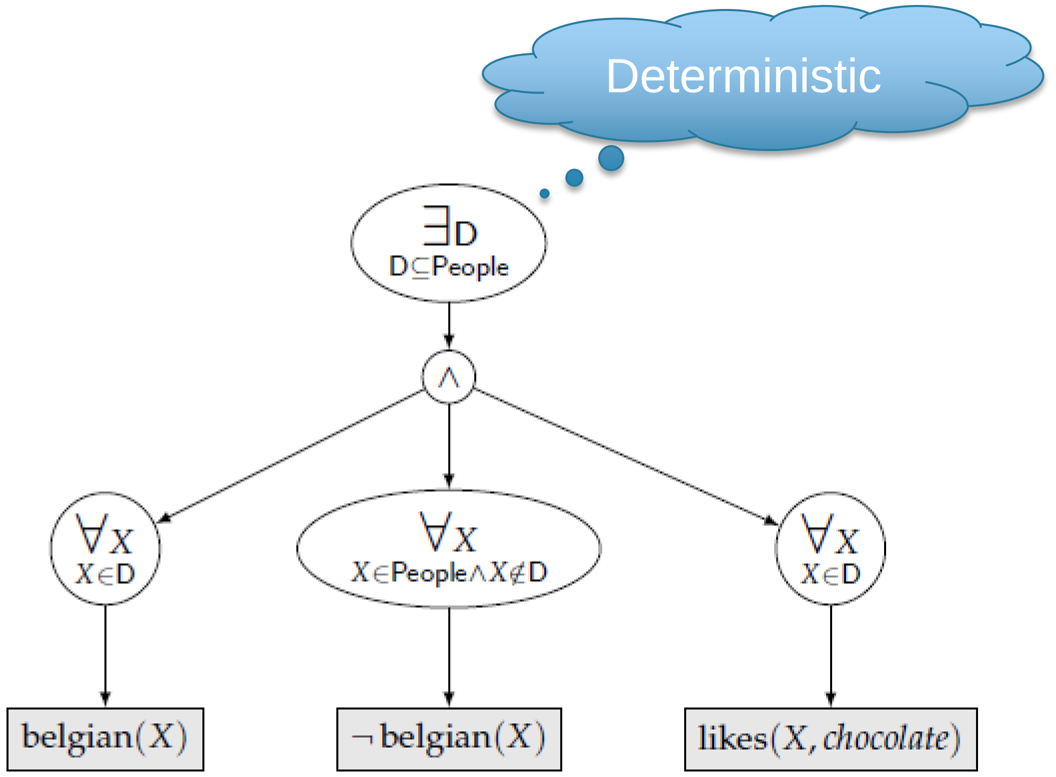
First-Order DNNF

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



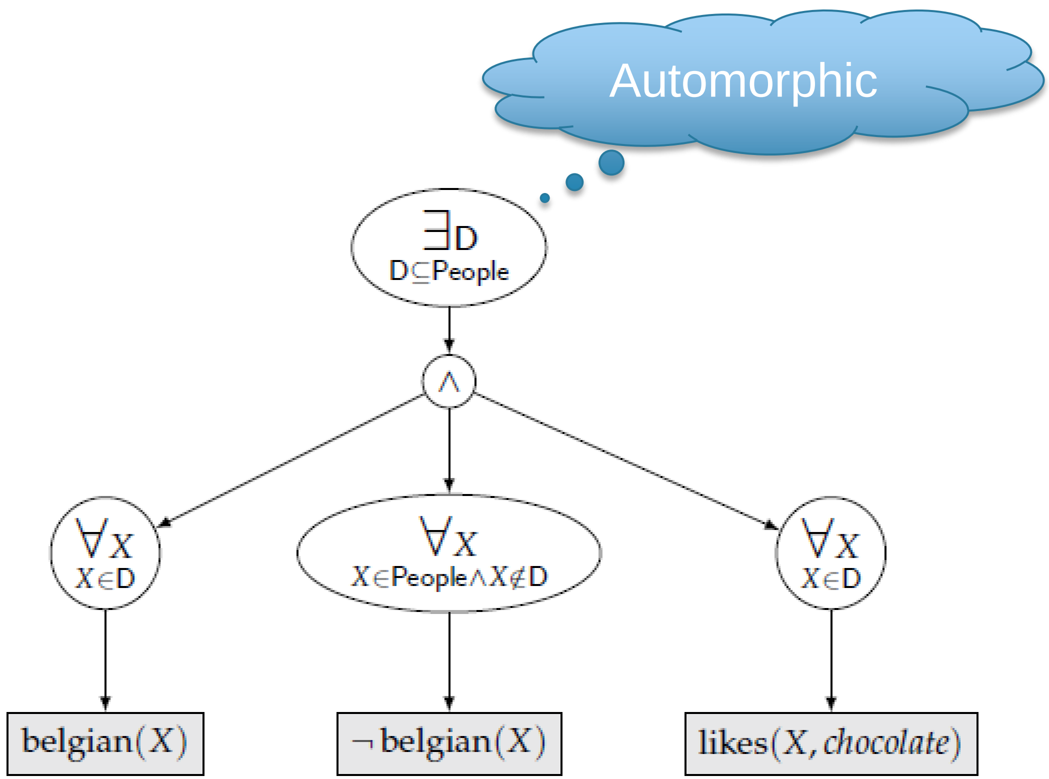
First-Order d-DNNF

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



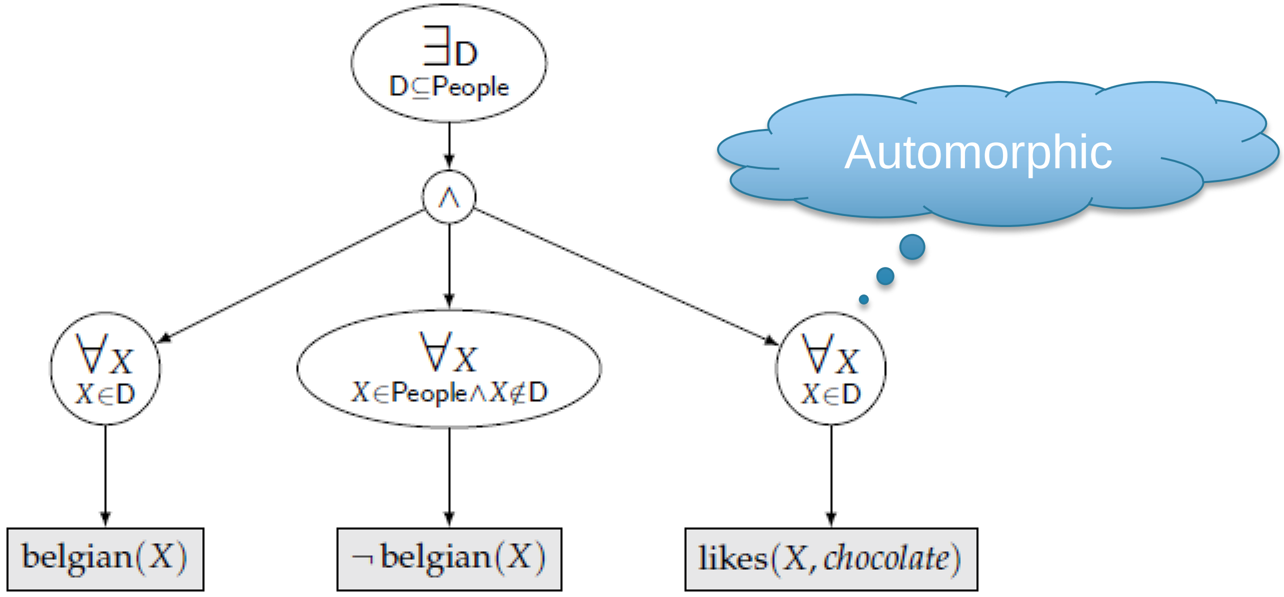
First-Order d-DNNF

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



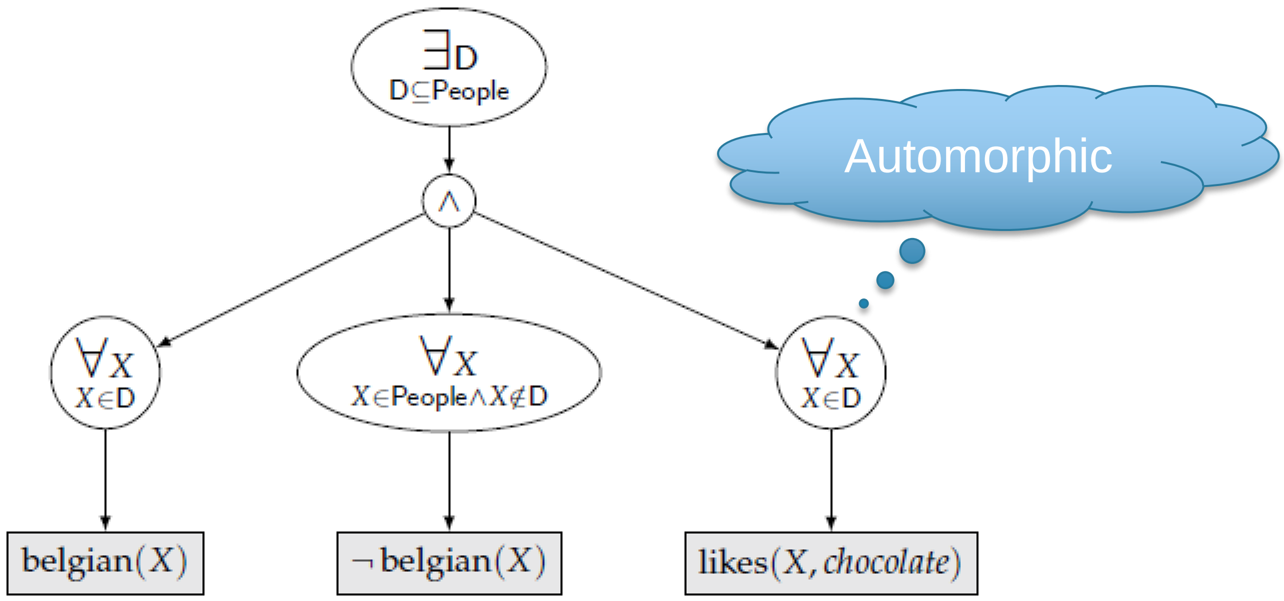
First-Order d-DNNF

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



First-Order ad-DNNF

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



FO NNF Languages

- FO NNF: group logic circuits, negation only on atoms
- FO d-DNNF: determinism and decomposability

  Grounding generates a d-DNNF
- FO DNNF

  Grounding generates a DNNF
- FO ad-DNNF: automorphic

  Powerful properties!

FO NNF TRACTABILITY

Symmetric WFOMC

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

- **R** = $(R_1, R_2, \ldots, R_k)$ = relational vocabulary
- **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

Complexity of FOMC / WFOMC(Q, n)?
Data/domain complexity:
fixed Q, input n / and **w**

Symmetric WFOMC on FO ad-DNNF

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

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

FOMC Query: Example

FO-Model Counting: $w(R) = w(\neg R) = 1$

FO ad-DNNF sentences

FOMC Query: Example

FO-Model Counting: $w(R) = w(\neg R) = 1$

FO ad-DNNF sentences

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

Domain = {Alice}

FOMC Query: Example

FO-Model Counting: $w(R) = w(\neg R) = 1$

FO ad-DNNF sentences

---

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

Domain = {Alice}

→ 3 models

FOMC Query: Example

FO-Model Counting: $w(R) = w(\neg R) = 1$

FO ad-DNNF sentences

---

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

Domain = {Alice}

→ 3 models

---

3. Δ = ∀x, (Stress(x) ⇒ Smokes(x))

Domain = {n people}

FOMC Query: Example

FO-Model Counting: $w(R) = w(\neg R) = 1$

FO ad-DNNF sentences

---

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

→ 3 models

---

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

→ 3^n models

FOMC Query: Example

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

Domain = {n people}

→ 3^n models

FOMC Query: 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}

FOMC Query: 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

FOMC Query: Example

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

Domain = {n people}

→ 3^n models

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

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 Query: Example

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

Domain = {n people}

→ 3^n models

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

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

→ $3^n + 4^n$ models

FOMC Query: Example

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

Domain = {n people}

→ 3^n models

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

D = {n people}

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

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

→ $3^n + 4^n$ models

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

D = {n people}

FOMC Query: 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

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

→ $3^n + 4^n$ models

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

→ $(3^n + 4^n)^n$ models

Group Quantifiers: Example

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

Domain = {n people}

- Not decomposable!
- Rewrite as FO ad-DNNF:

$\exists \mathbf{G} \subseteq \mathbf{D}$, Smokes($\mathbf{G}$) ∧ $\overline{\text{S}}$mokes($\overline{\mathbf{G}}$) ∧ $\overline{\text{F}}$riends($\mathbf{G}$, $\overline{\mathbf{G}}$)

- Not possible to ground to d-DNNF!
- How to do tractable CT?

$\sum_{i=0}^{|\tau|} \binom{|\tau|}{i} \cdot U(\beta\{\mathbf{X}/\mathbf{x}_i\})$ *when* $\alpha = \exists \mathbf{X} \subseteq \tau, \beta,$ *and* $\mathbf{x}_i$ *is any subset of* τ *such that* $|\mathbf{x}_i| = i$

Group Quantifiers: Example

$$\exists \mathbf{G} \subseteq \mathbf{D}, \text{Smokes}(\mathbf{G}) \wedge \overline{\text{S}}\text{mokes}(\overline{\mathbf{G}}) \wedge \overline{\text{F}}\text{riends}(\mathbf{G}, \overline{\mathbf{G}})$$

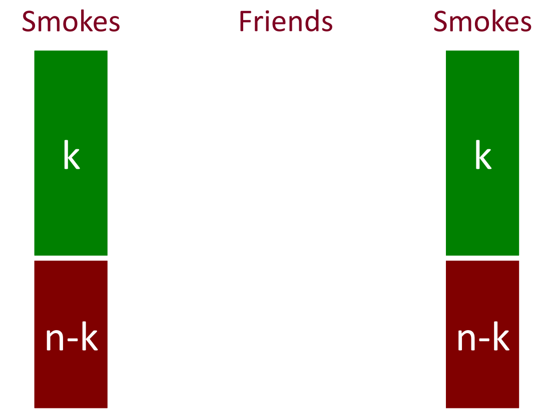
Group Quantifiers: Example

$$\exists \mathbf{G} \subseteq \mathbf{D}, \text{Smokes}(\mathbf{G}) \wedge \overline{\text{Smokes}}(\overline{\mathbf{G}}) \wedge \overline{\text{Friends}}(\mathbf{G}, \overline{\mathbf{G}})$$

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

Database:

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



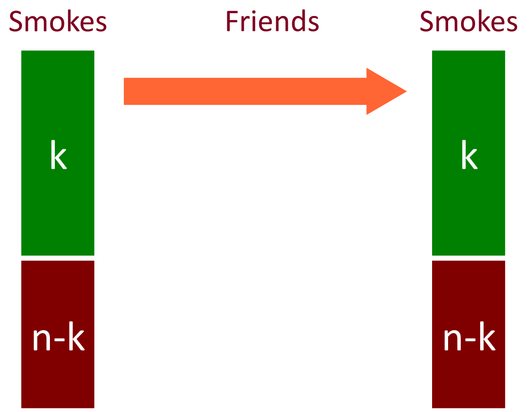
Group Quantifiers: Example

$\exists \mathbf{G} \subseteq \mathbf{D}, \text{Smokes}(\mathbf{G}) \wedge \overline{\text{Smokes}}(\overline{\mathbf{G}}) \wedge \overline{\text{Friends}}(\mathbf{G}, \overline{\mathbf{G}})$

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

Database:

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



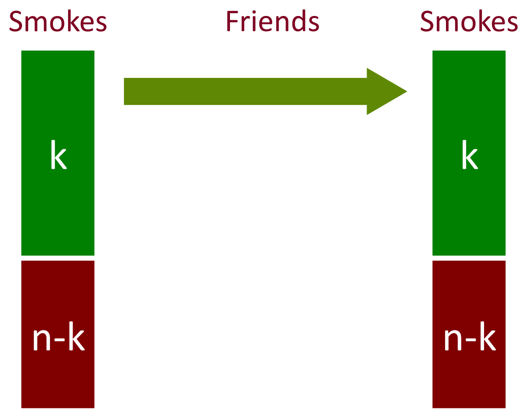
Group Quantifiers: Example

$$\exists \mathbf{G} \subseteq \mathbf{D},\ \text{Smokes}(\ \mathbf{G}\ ) \wedge \overline{\text{S}}\text{mokes}(\ \overline{\mathbf{G}}\ ) \wedge \overline{\text{F}}\text{riends}(\ \mathbf{G}\ ,\ \overline{\mathbf{G}}\ )$$

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

Database:

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



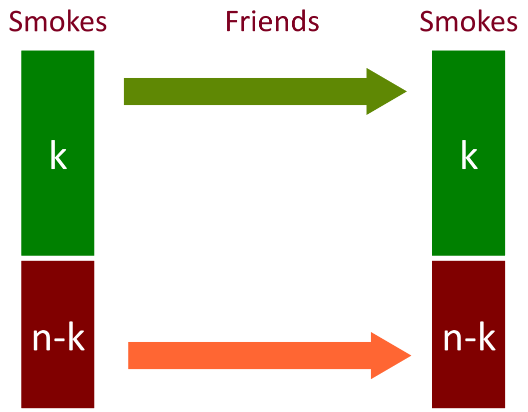
Group Quantifiers: Example

$$\exists \mathbf{G} \subseteq \mathbf{D}, \text{Smokes}(\mathbf{G}) \wedge \overline{\text{S}}\text{mokes}(\overline{\mathbf{G}}) \wedge \overline{\text{F}}\text{riends}(\mathbf{G}, \overline{\mathbf{G}})$$

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

Database:

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



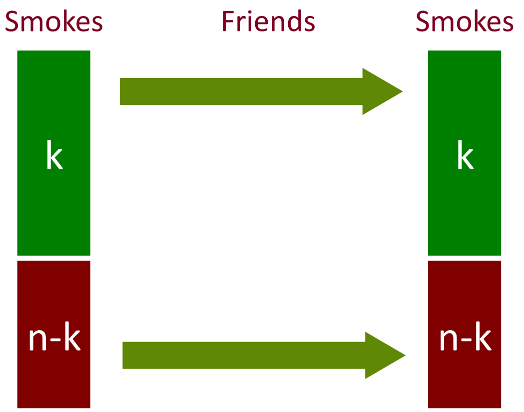
Group Quantifiers: Example

$$\exists \mathbf{G} \subseteq \mathbf{D}, \text{Smokes}(\mathbf{G}) \wedge \overline{\text{Smokes}}(\overline{\mathbf{G}}) \wedge \overline{\text{Friends}}(\mathbf{G}, \overline{\mathbf{G}})$$

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

Database:

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



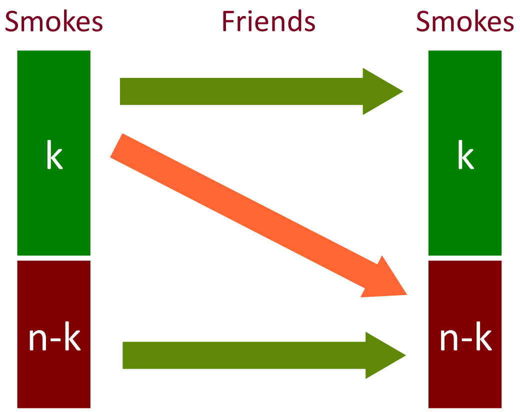
Group Quantifiers: Example

$$\exists \mathbf{G} \subseteq \mathbf{D},\ \text{Smokes}(\mathbf{G}) \wedge \overline{\text{Smokes}}(\overline{\mathbf{G}}) \wedge \overline{\text{Friends}}(\mathbf{G}, \overline{\mathbf{G}})$$

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

Database:

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



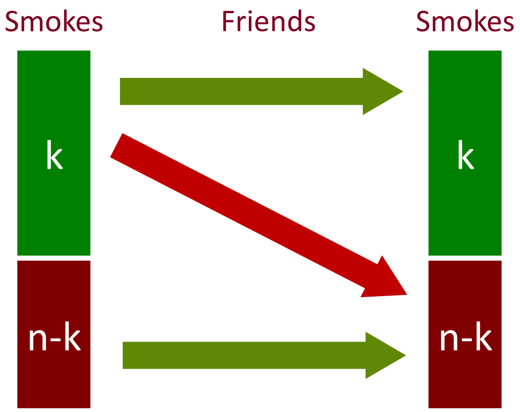
Group Quantifiers: Example

$$\exists \mathbf{G} \subseteq \mathbf{D},\ \text{Smokes}(\mathbf{G}) \wedge \overline{\text{Smokes}}(\overline{\mathbf{G}}) \wedge \overline{\text{Friends}}(\mathbf{G}, \overline{\mathbf{G}})$$

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

Database:

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



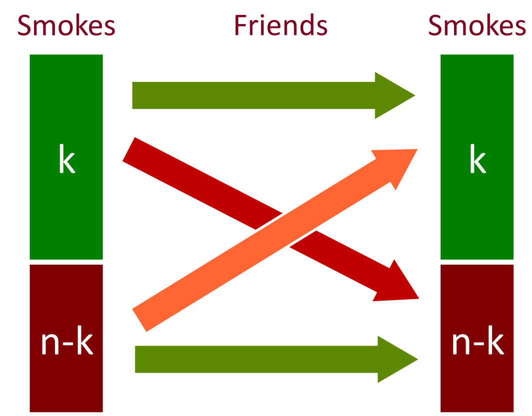
Group Quantifiers: Example

$$\exists \mathbf{G} \subseteq \mathbf{D}, \text{Smokes}(\mathbf{G}) \wedge \overline{\text{Smokes}}(\overline{\mathbf{G}}) \wedge \overline{\text{Friends}}(\mathbf{G}, \overline{\mathbf{G}})$$

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

Database:

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



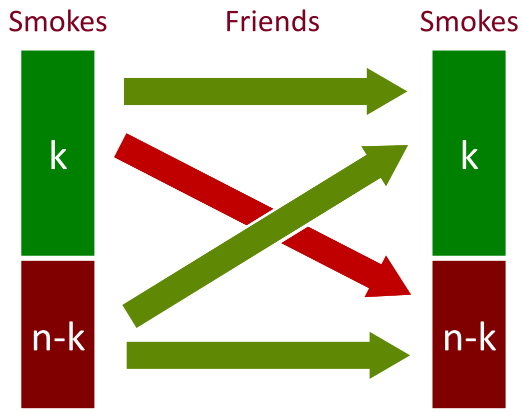
Group Quantifiers: Example

$$\exists \mathbf{G} \subseteq \mathbf{D}, \text{Smokes}(\mathbf{G}) \wedge \overline{\text{S}}\text{mokes}(\overline{\mathbf{G}}) \wedge \overline{\text{F}}\text{riends}(\mathbf{G}, \overline{\mathbf{G}})$$

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

Database:

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



Group Quantifiers: Example

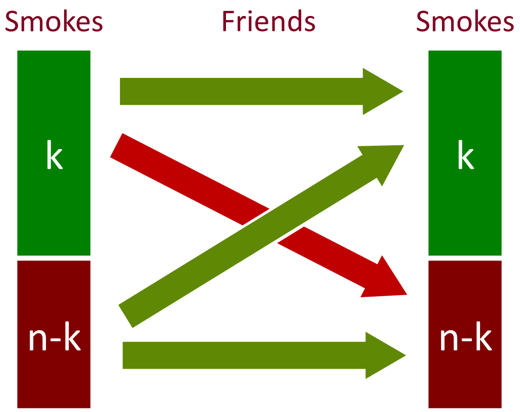
$\exists \mathbf{G} \subseteq \mathbf{D}$, Smokes($\mathbf{G}$) $\wedge$ $\overline{\text{Smokes}}$($\overline{\mathbf{G}}$) $\wedge$ $\overline{\text{Friends}}$($\mathbf{G}$, $\overline{\mathbf{G}}$)

- If we know **G** 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



Group Quantifiers: Example

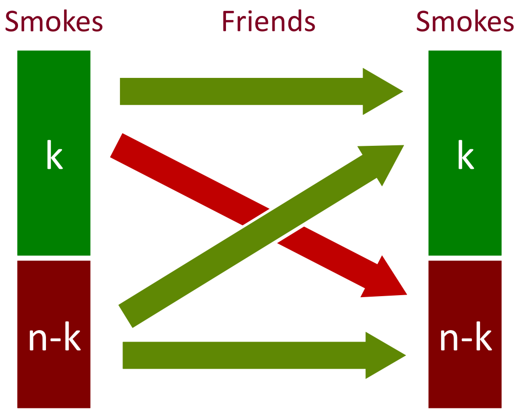
$$\exists \mathbf{G} \subseteq \mathbf{D}, \text{Smokes}(\mathbf{G}) \wedge \overline{\text{Smokes}}(\overline{\mathbf{G}}) \wedge \overline{\text{Friends}}(\mathbf{G}, \overline{\mathbf{G}})$$

- If we know **G** 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?

Group Quantifiers: Example

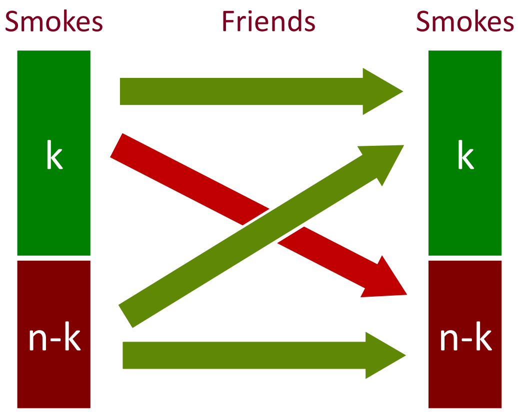
$$\exists \mathbf{G} \subseteq \mathbf{D}, \text{Smokes}(\mathbf{G}) \wedge \overline{\text{S}}\text{mokes}(\overline{\mathbf{G}}) \wedge \overline{\text{F}}\text{riends}(\mathbf{G}, \overline{\mathbf{G}})$$

- If we know **G** 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

Group Quantifiers: Example

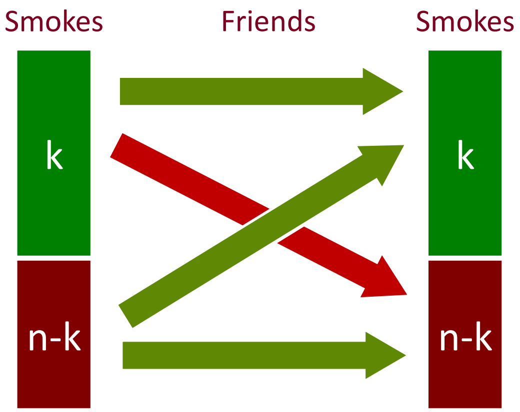
$\exists \mathbf{G} \subseteq \mathbf{D}$, Smokes(**G**) ∧ $\overline{\text{Smokes}}$($\overline{\mathbf{G}}$) ∧ $\overline{\text{Friends}}$(**G** , $\overline{\mathbf{G}}$)

- If we know **G** 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…

Group Quantifiers: Example

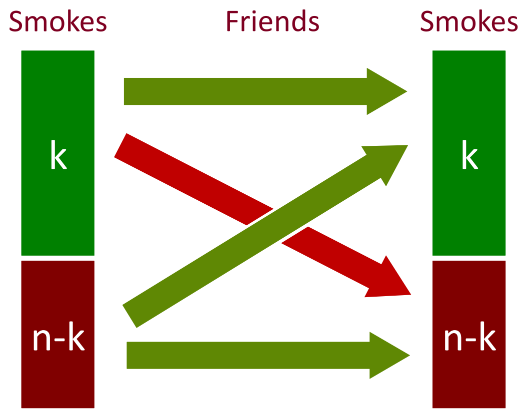
$\exists \mathbf{G} \subseteq \mathbf{D}, \text{Smokes}(\mathbf{G}) \wedge \overline{\text{Smokes}}(\overline{\mathbf{G}}) \wedge \overline{\text{Friends}}(\mathbf{G}, \overline{\mathbf{G}})$

- If we know **G** 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

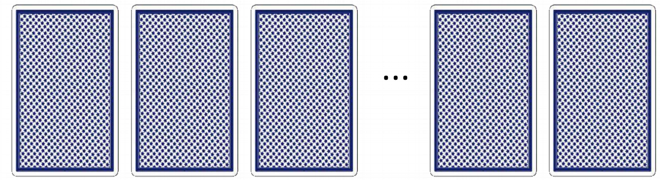


- If we know that there are *k* smokers? → $\binom{n}{k} 2^{n^2-k(n-k)}$ models

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

Playing Cards Revisited

Let us automate this:

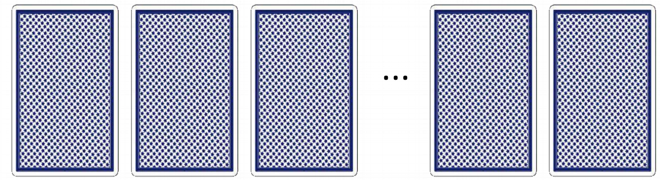


$$\forall p, \exists c, Card(p,c)$$
$$\forall c, \exists p, Card(p,c)$$
$$\forall p, \forall c, \forall c', Card(p,c) \land Card(p,c') \Rightarrow c = c'$$

[Van den Broeck.; 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’

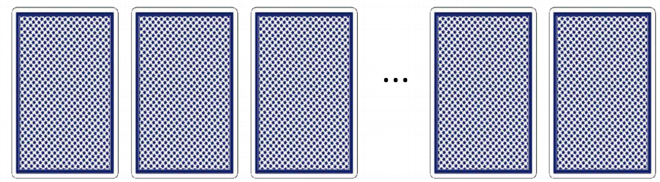
↓

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

[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]

FO COMPILATION

Compilation Rules

- Lots of preprocessing
- Shannon decomposition/Boole's expansion
- Detect propositional decomposability
- FO Shannon decomposition:

$$\exists \mathbf{X} \subseteq \tau, P(\mathbf{X}) \wedge \overline{P}(\overline{\mathbf{X}}) \wedge \beta$$

  Simplify β (remove atoms subsumed by P(**X**))

  Always deterministic! Ensure automorphic ∃

- Detect FO decomposability

FO NNF EXPRESSIVENESS

Main Positive Result: FO^2

- FO^2 = FO restricted to two variables
- "The graph has a path of length 10":

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

- Theorem: Compilation algorithm to FO ad-DNNF is complete for FO^2
- Model counting for FO^2 in PTIME domain complexity

Main Negative Results

Domain 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

Therefore, no FO ad-DNNF can exist ☹

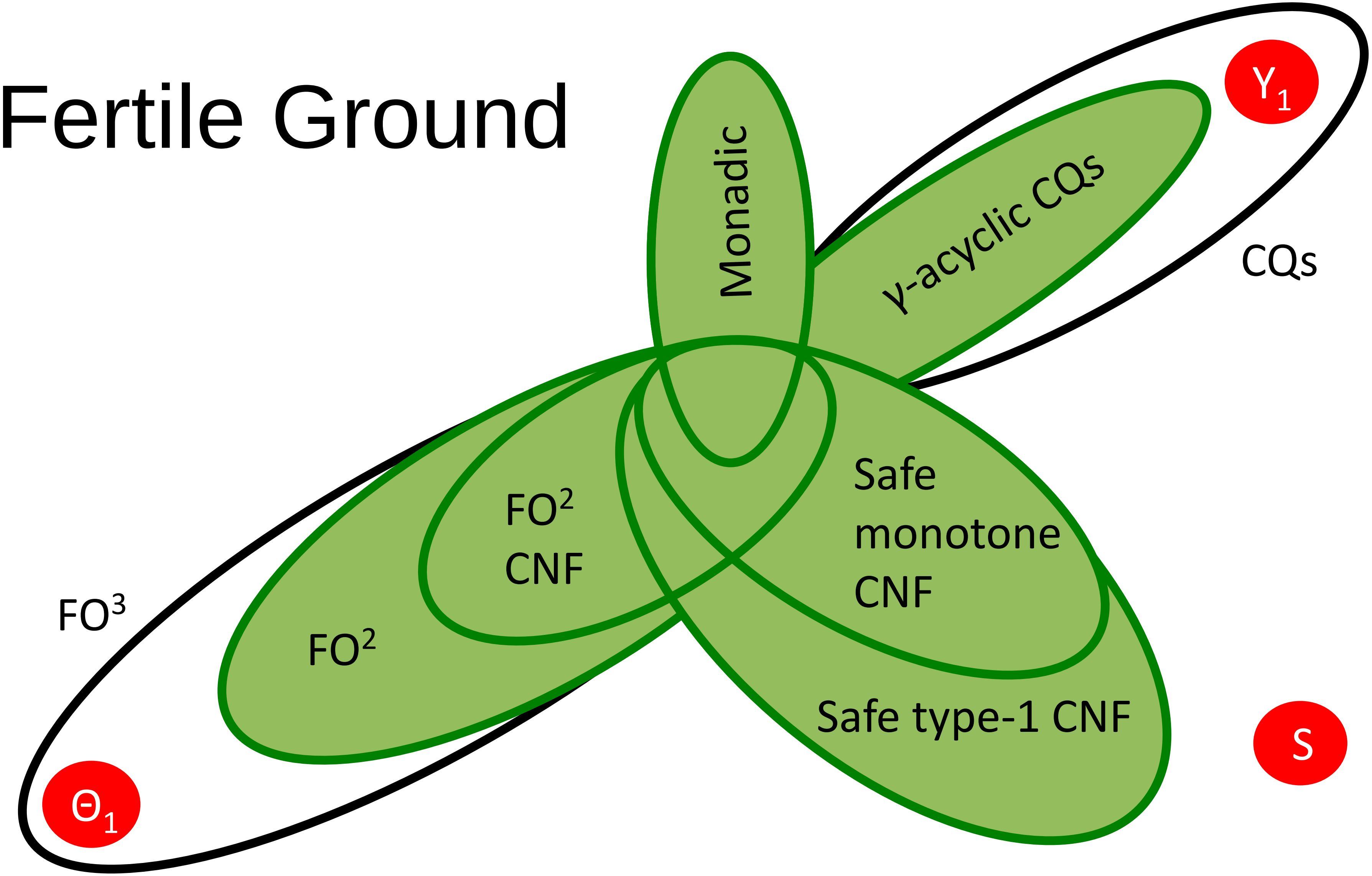
Proof

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

Proof

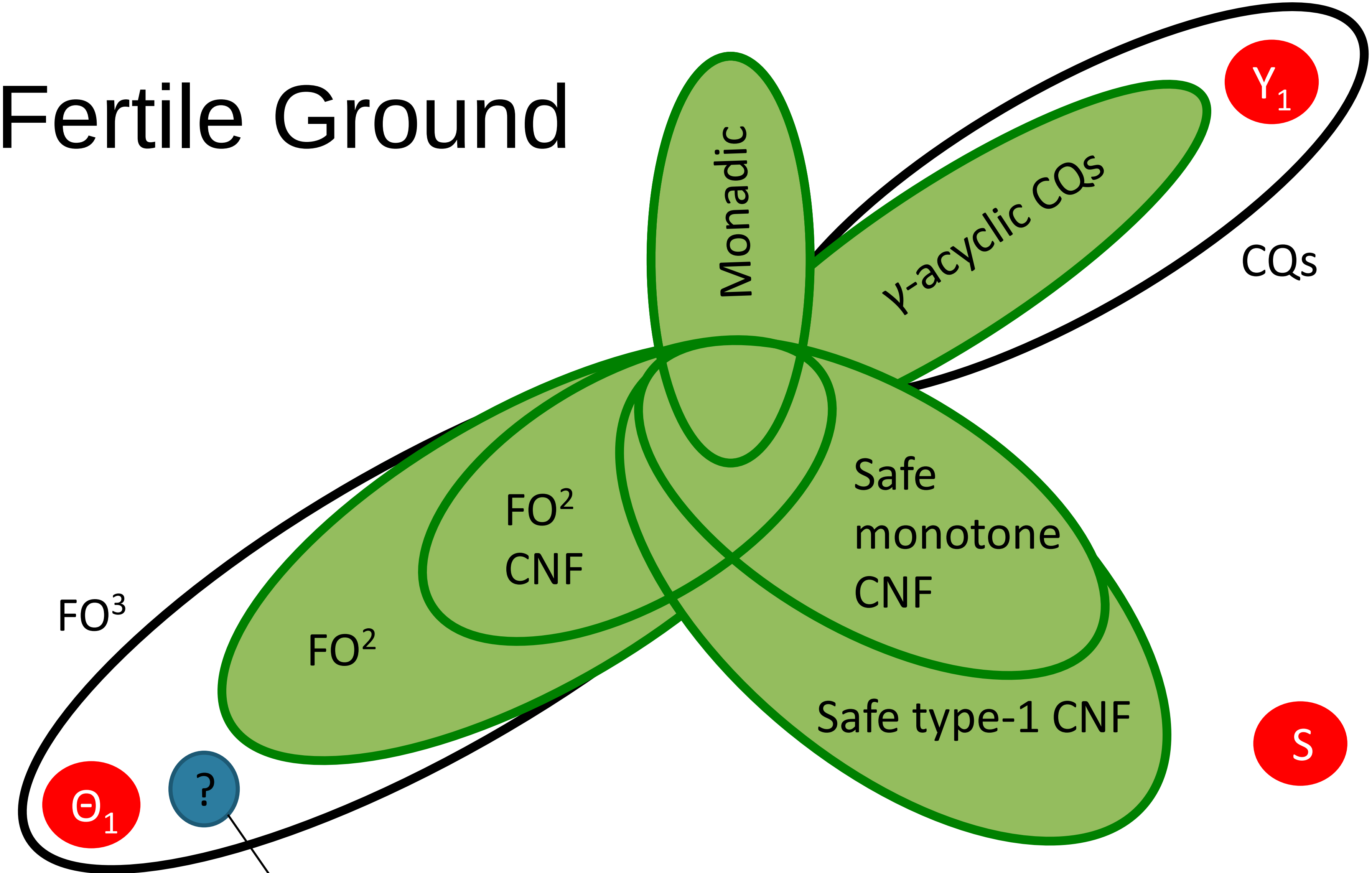
- Step 1. Construct a Turing Machine U s.t.
  - U is in $\#P_1$ and runs in linear time in n
  - U computes a $\#P_1$ –hard function
- Step 2. Construct an FO^3 sentence Q s.t. FOMC(Q,n) / n! = U(n)

Fertile Ground



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

Fertile Ground



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

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

Other Queries and Transformations

- What if all ground atoms have different weights? Asymmetric WFOMC
- FO d-DNNF complete for all monotone FO CNFs that support efficient **CT**
- No clausal entailment
- No conditioning

Conclusions

- Very powerful already!
- We need to solve this!

THANKS

References

- Cards Example:
  Guy Van den Broeck. Towards High-Level Probabilistic Reasoning with Lifted Inference, In Proceedings of KRR, 2015.
- First-Order Knowledge Compilation:
  Guy Van den Broeck. Lifted Inference and Learning in Statistical Relational Models, PhD thesis, KU Leuven, 2013.
- Expressiveness:
  Paul Beame, Guy Van den Broeck, Eric Gribkoff, Dan Suciu. Symmetric Weighted First-Order Model Counting, In Proceedings of PODS, 2015.