KATHOLIEKE UNIVERSITEIT LEUVEN

DTAI MACHINE LEARNING

On the Completeness of First-Order Knowledge Compilation for Lifted Probabilistic Inference

Guy Van den Broeck

NIPS11
December 13, 2011

Outline

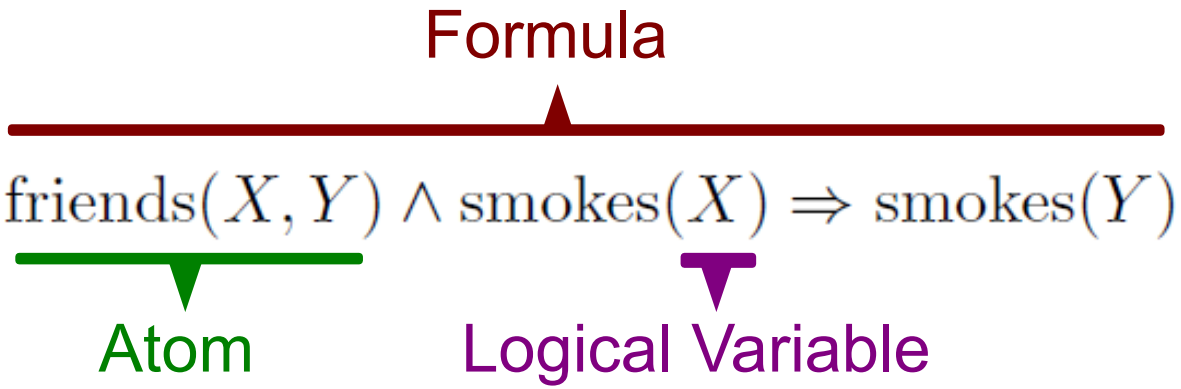
- Probabilistic Logic
- Lifted Inference
- Compilation Algorithm
- Completeness
- Conclusions

Outline

- Probabilistic Logic
- Lifted Inference
- Compilation Algorithm
- Completeness
- Conclusions

First-Order Logic

- Example: FOL



- Logical variables have domain of constants

  e.g., X,Y range over domain People = {alice,bob}

- Ground formula has no logical variables

  e.g., $\text{friends}(alice, bob) \wedge \text{smokes}(alice) \Rightarrow \text{smokes}(bob)$

Probabilistic Logic

- Example: Markov Logic Network (MLN)

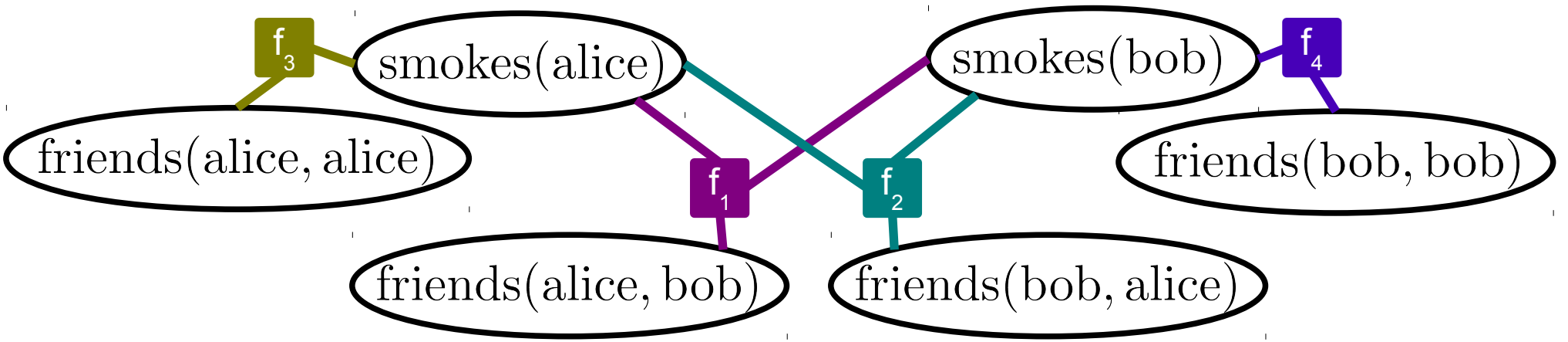
Weight~Probability Formula

$$2 \quad \text{friends}(X, Y) \wedge \text{smokes}(X) \Rightarrow \text{smokes}(Y)$$

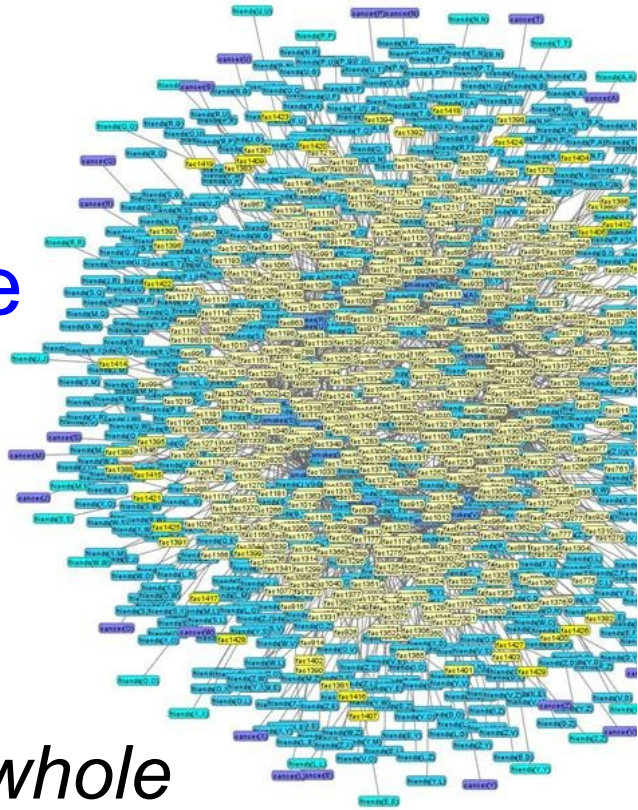
- Ground atom = random variable in {true,false}

  e.g., smokes(alice), friends(alice,bob)

- Ground formula = factor in propositional factor graph



Lifted Probabilistic Inference



- Factor graph explodes
- Propositional inference is intractable
- Solution: lifted inference

  Exploit symmetries

  Reason at first-order level

  Reason about groups of objects as a whole

  Avoid repeated computations

  Mimic resolution in theorem proving

- There is a common understanding but no formal definition of lifted inference!

Questions?

- What is commonly understood as lifted inference?
  - Contribution: A formal framework for lifted inference (definition + complexity considerations)
    ~ PAC-learnability (Valiant)
- When can a model be lifted?
  - Contribution: Extended first-order knowledge compilation
  - Contribution: Completeness result

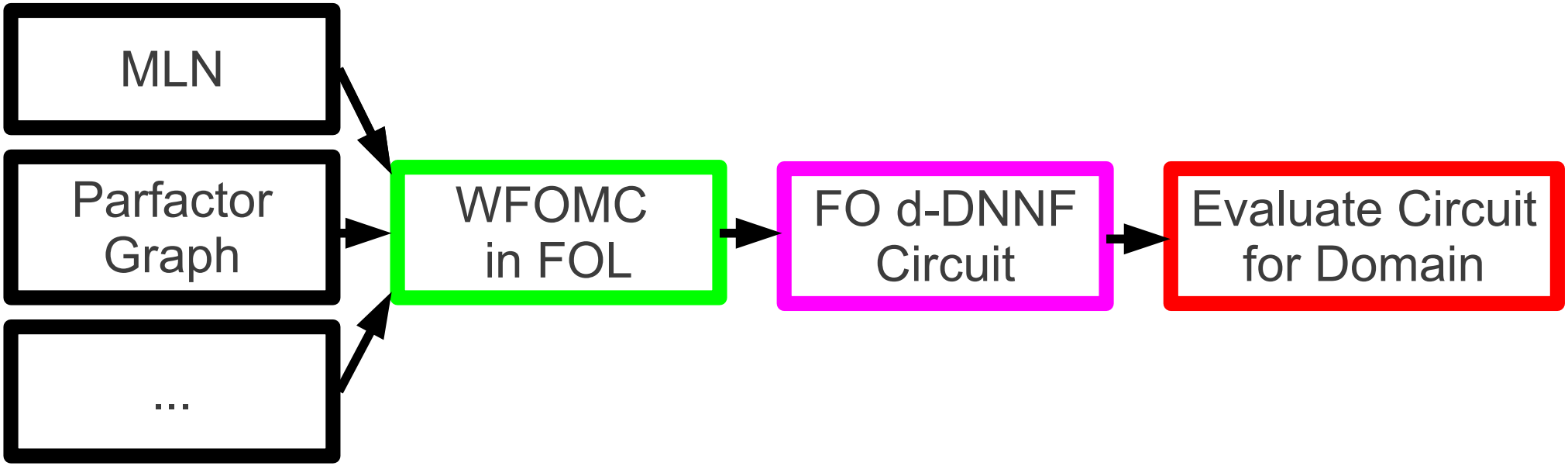
Take-away message: **Probabilistic models with 2 logical variables per formula are liftable.**

Outline

- Probabilistic Logic
- Lifted Inference
- Compilation Algorithm
- Completeness
- Conclusions

Lifted Inference by First-Order Knowledge Compilation

	Variable Elimination	Belief Propagation	Knowledge Compilation
Propositional	[Zhang94]	[Pearl82]	[Darwiche03]
Lifted	[Poole03]	[Singla08]	**[VdB11]**



[Van den Broeck, Guy; Taghipour, Nima; Meert, Wannes; Davis, Jesse; De Raedt, Luc Lifted probabilistic inference by first-order knowledge compilation, IJCAI11]

Weighted First-Order Model Counting

- A logical theory

$\text{smokes}(alice)$	$\text{smokes}(bob)$	$\text{friends}(alice, bob)$	$\text{friends}(bob, alice)$
0	0	0	0
...	...	...	...
1	0	1	0
...	...	...	...
1	1	1	1

Possible worlds
Logical interpretations

Weighted First-Order Model Counting

- A logical theory

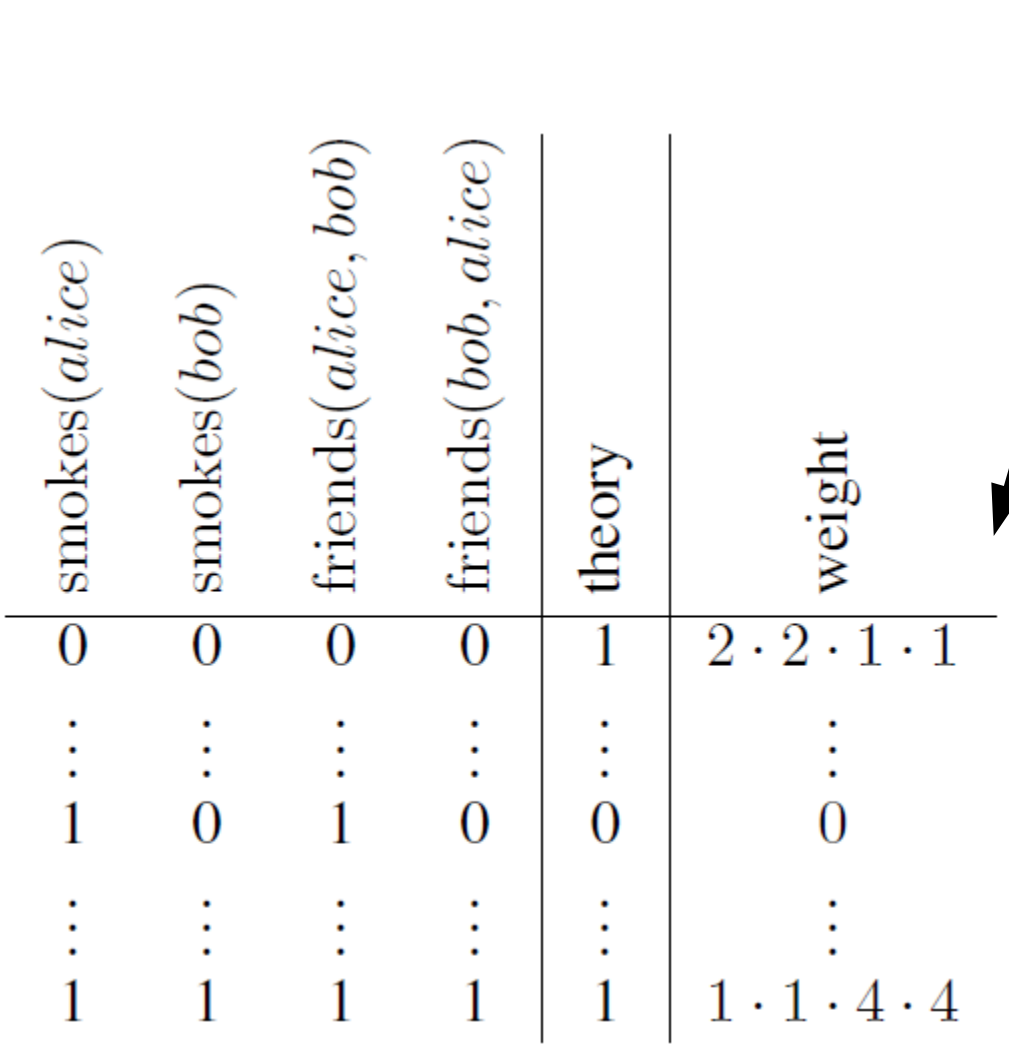
$$\text{friends}(X, Y) \wedge \text{smokes}(X) \Rightarrow \text{smokes}(Y)$$

$\text{smokes}(alice)$	$\text{smokes}(bob)$	$\text{friends}(alice, bob)$	$\text{friends}(bob, alice)$	theory
0	0	0	0	1
...	...	...	...	...
1	0	1	0	0
...	...	...	...	...
1	1	1	1	1

Interpretations that satisfy the theory
Models

Weighted First-Order Model Counting

- A logical theory and a weight function for predicates



smokes($alice$)	smokes(bob)	friends($alice$, bob)	friends(bob, $alice$)	theory	weight
0	0	0	0	1	$2 \cdot 2 \cdot 1 \cdot 1$
...	...	...	...	...	...
1	0	1	0	0	0
...	...	...	...	...	...
1	1	1	1	1	$1 \cdot 1 \cdot 4 \cdot 4$

Weighted First-Order Model Counting

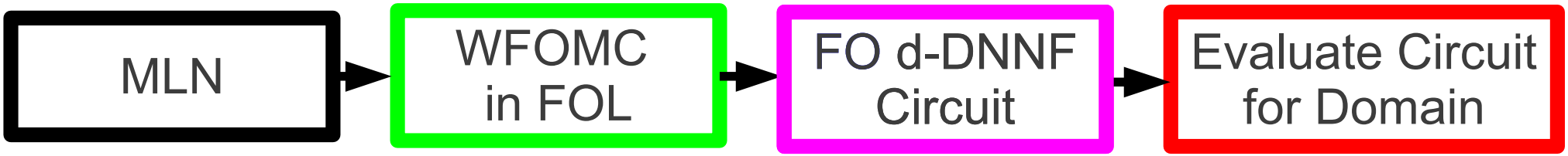
- A logical theory and a weight function for predicates

smokes($alice$)	smokes(bob)	friends($alice$, bob)	friends(bob, $alice$)	theory	weight
0	0	0	0	1	$2 \cdot 2 \cdot 1 \cdot 1$
...	...	...	...	...	...
1	0	1	0	0	0
...	...	...	...	...	...
1	1	1	1	1	$1 \cdot 1 \cdot 4 \cdot 4$

$\sum$

Weighted first-order model count
~Partition function

Lifted Inference by First-Order Knowledge Compilation

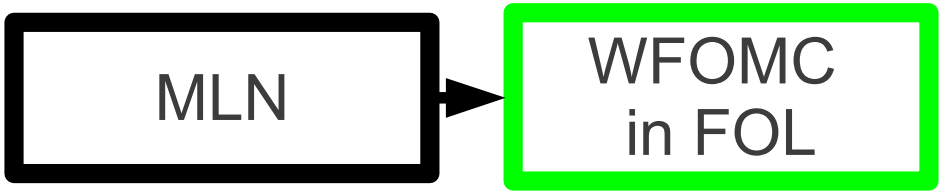


Lifted Inference by First-Order Knowledge Compilation

MLN

$$2\ \mathrm{friends}(X, Y) \wedge \mathrm{smokes}(X) \Rightarrow \mathrm{smokes}(Y)$$

Lifted Inference by First-Order Knowledge Compilation



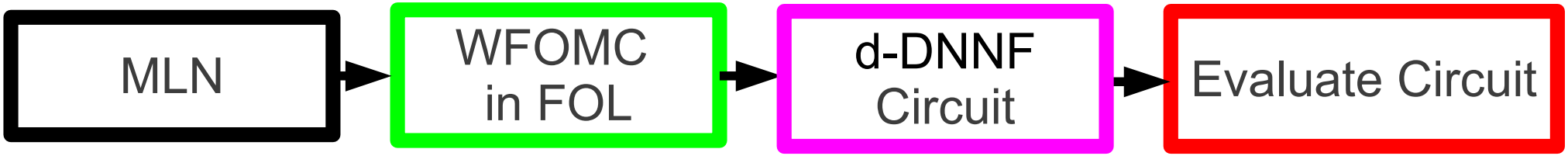
$$2\ \mathrm{friends}(X,Y) \wedge \mathrm{smokes}(X) \Rightarrow \mathrm{smokes}(Y)$$

$$\mathrm{smokes}(Y) \vee \neg\,\mathrm{smokes}(X) \vee \neg\,\mathrm{friends}(X,Y) \vee \neg\,\mathrm{f}(X,Y)$$
$$\mathrm{friends}(X,Y) \vee \mathrm{f}(X,Y)$$
$$\mathrm{smokes}(X) \vee \mathrm{f}(X,Y)$$
$$\neg\,\mathrm{smokes}(Y) \vee \mathrm{f}(X,Y).$$

Predicate	w	$\overline{\mathrm{w}}$
friends	1	1
smokes	1	1
f	e^2	1

WFOMC in FOL equivalent to partition function of MLN

Lifted Inference by First-Order Knowledge Compilation



Ground to propositional logic
Logical d-DNNF circuit
Inducing an arithmetic circuit

$$\text{smokes}(Y) \vee \neg\,\text{smokes}(X) \vee \neg\,\text{friends}(X,Y) \vee \neg\,\text{f}(X,Y)$$
$$\text{friends}(X,Y) \vee \text{f}(X,Y)$$
$$\text{smokes}(X) \vee \text{f}(X,Y)$$
$$\neg\,\text{smokes}(Y) \vee \text{f}(X,Y).$$

Predicate	w	$\overline{\text{w}}$
friends	1	1
smokes	1	1
f	e^2	1

Circuit for domain {alice}

Circuit for domain {alice, bob}

Circuit for domain {alice, bob, charlie}

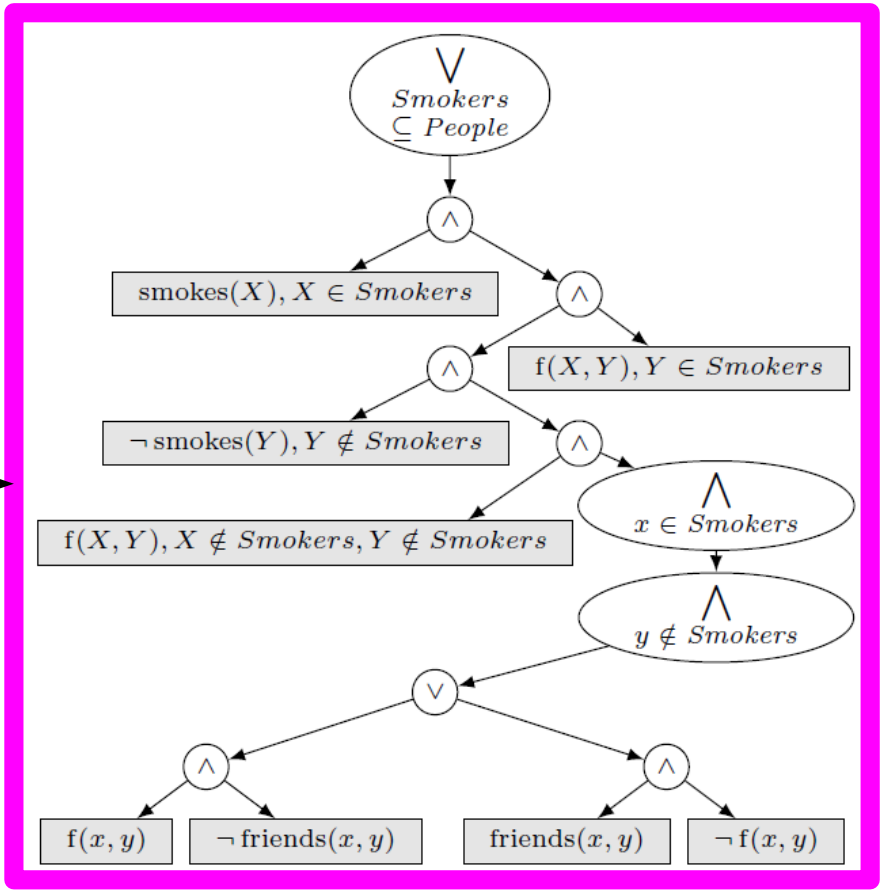
Lifted Inference by First-Order Knowledge Compilation

MLN → WFOMC in FOL → **FO** d-DNNF Circuit → Evaluate Circuit for Domain

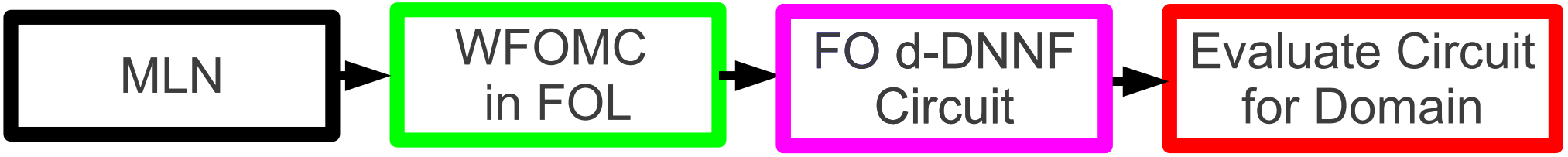
First-Order d-DNNF circuit
Independent of domain size

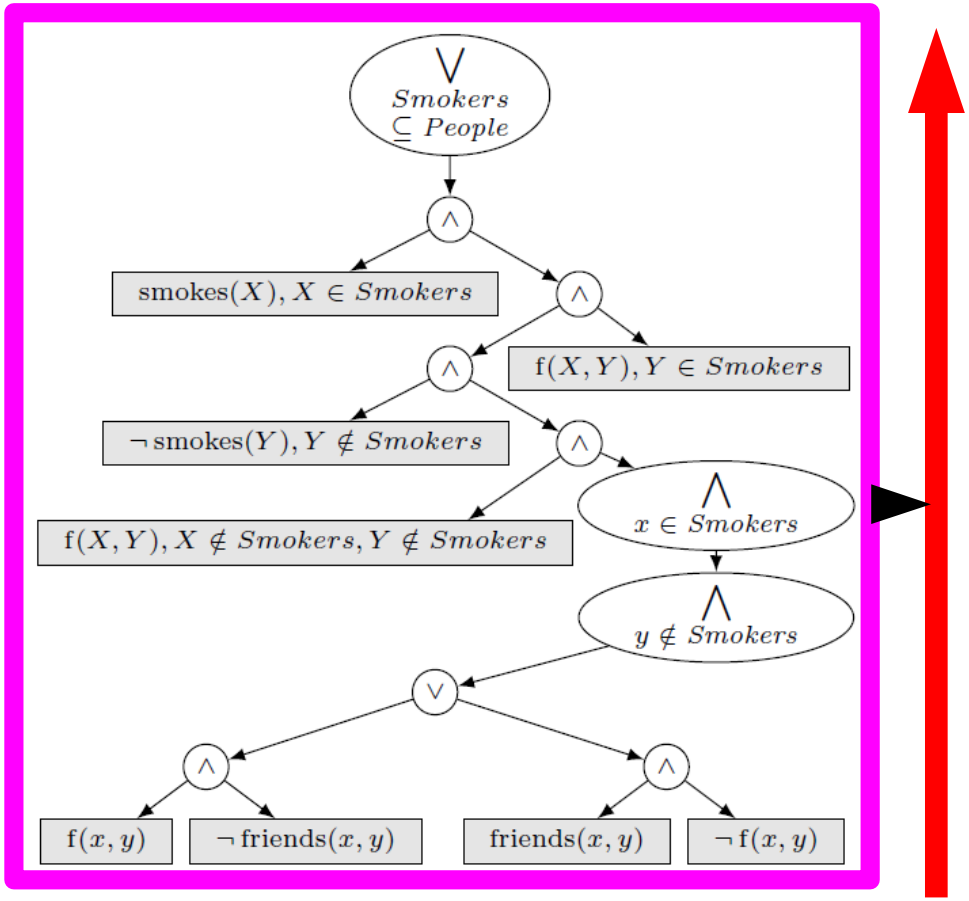
$$\text{smokes}(Y) \vee \neg\,\text{smokes}(X) \vee \neg\,\text{friends}(X,Y) \vee \neg\,\text{f}(X,Y)$$
$$\text{friends}(X,Y) \vee \text{f}(X,Y)$$
$$\text{smokes}(X) \vee \text{f}(X,Y)$$
$$\neg\,\text{smokes}(Y) \vee \text{f}(X,Y).$$

Predicate	w	$\overline{\text{w}}$
friends	1	1
smokes	1	1
f	e^2	1



Lifted Inference by First-Order Knowledge Compilation

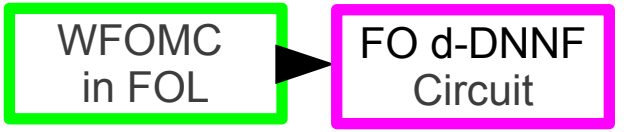




Outline

- Probabilistic Logic
- Lifted Inference
- Compilation Algorithm
- Completeness
- Conclusions

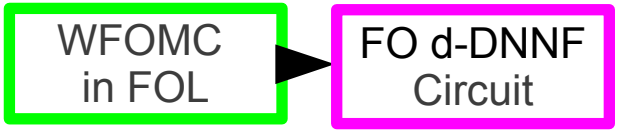
Compilation Algorithm CR1 [VdB11]



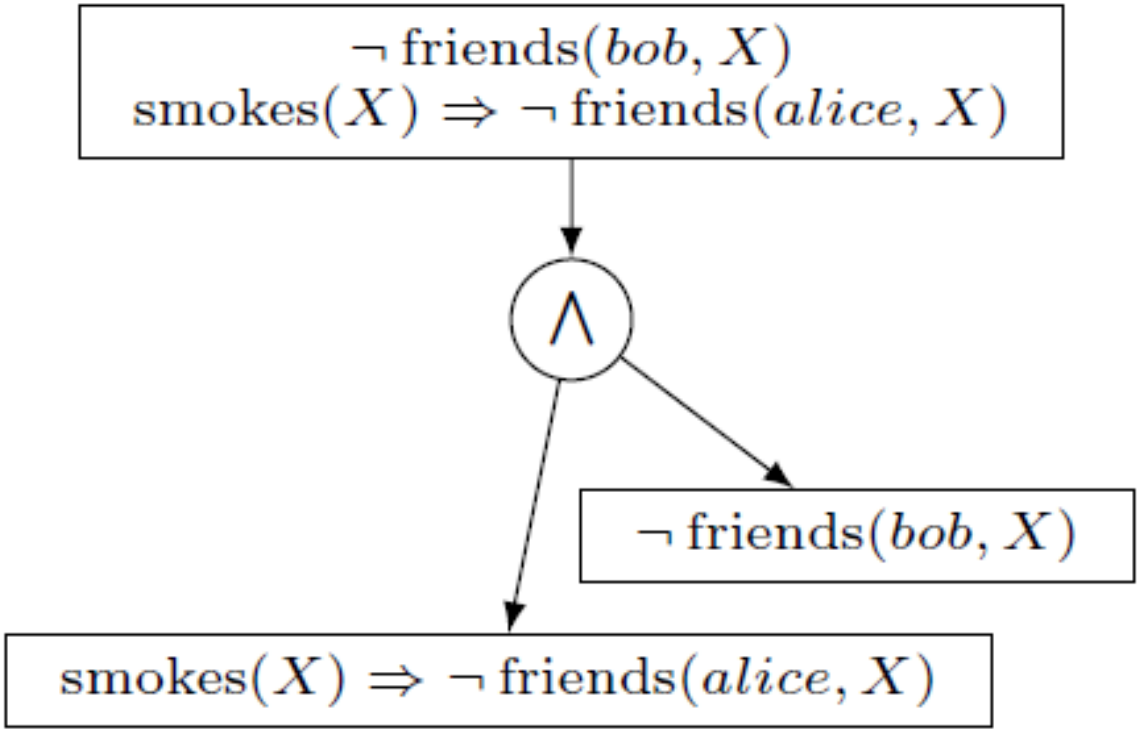
- 6 compilation rules:
  - Input: FO logic theory; Output: FO d-DNNF circuit
  - Compilation rule recursively compiles 'simpler' theories
- Example: Independence compilation rule

$$\neg\,\text{friends}(bob, X)$$
$$\text{smokes}(X) \Rightarrow \neg\,\text{friends}(alice, X)$$

Compilation Algorithm CR1 [VdB11]



- 6 compilation rules:
  - Input: FO logic theory; Output: FO d-DNNF circuit
  - Compilation rule recursively compiles 'simpler' theories
- Example: Independence compilation rule



New Rule: Domain Recursion

- Example theory: $\boxed{\text{friends}(X, Y) \Rightarrow \text{friends}(Y, X)}$
- Split up domain People into {c} U People'
- Split up theory into 3 independent subtheories

New Rule: Domain Recursion

- Example theory: $\text{friends}(X, Y) \Rightarrow \text{friends}(Y, X)$
- Split up domain People into {c} U People'
- Split up theory into 3 independent subtheories

  1) where X=c and Y=c: $\text{friends}(c, c) \Rightarrow \text{friends}(c, c)$

New Rule: Domain Recursion

- Example theory: $\text{friends}(X, Y) \Rightarrow \text{friends}(Y, X)$
- Split up domain People into {c} U People'
- Split up theory into 3 independent subtheories

  1) where X=c and Y=c: $\text{friends}(c, c) \Rightarrow \text{friends}(c, c)$

  2) where X≠c and Y≠c: $\text{friends}(X, Y) \Rightarrow \text{friends}(Y, X), X \neq c \wedge Y \neq c$

New Rule: Domain Recursion

- Example theory: $\text{friends}(X, Y) \Rightarrow \text{friends}(Y, X)$
- Split up domain People into {c} U People'
- Split up theory into 3 independent subtheories

1) where X=c and Y=c: $\text{friends}(c, c) \Rightarrow \text{friends}(c, c)$

2) where X≠c and Y≠c: $\text{friends}(X, Y) \Rightarrow \text{friends}(Y, X), X \neq c \wedge Y \neq c$

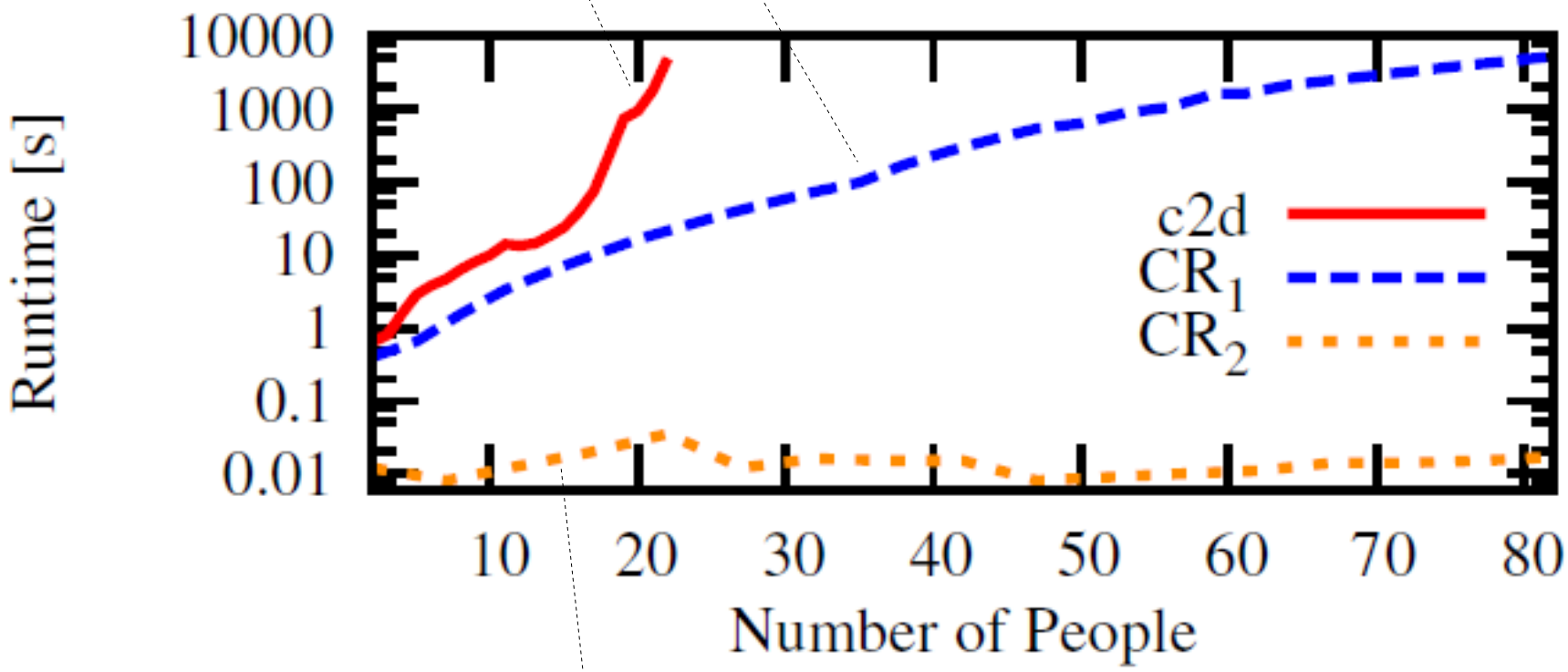
3) where (X≠c and Y=c) or (X=c and Y≠c):

$$\text{friends}(c, Y) \Rightarrow \text{friends}(Y, c), Y \neq c$$
$$\text{friends}(X, c) \Rightarrow \text{friends}(c, X), X \neq c$$

Experiments

c2d: Propositional knowledge compilation

CR1: Existing FO knowledge compilation



CR2: CR1 with domain recursion

Outline

- Probabilistic Logic
- Lifted Inference
- Compilation Algorithm
- Completeness
- Conclusions

Domain-Lifted Probabilistic Inference

Definition:
Complexity of computing P(*q*|*e*) in model *m* is polynomial time in the domain sizes of the logical variables in *q,e,m*

Possibly exponential in the size of *q,e,m*

 # predicates, # parfactors, # atoms,
 # arguments, # formulas, # constants in model

Motivation: Large domains lead to intractable propositional inference.

Completeness

A procedure that is domain-lifted for all models in a class M is called complete for M

All models in M are "liftable"

No completeness result so far for existing algorithms

If you give me a model,
I cannot say if grounding will be needed,
untill I run the inference algorithm itself.

Completeness of CR1 and CR2

- Definition:
  k-WFOMC consists of WFOMC theories with up to k logical variables per formula
- Theorem:
  CR1 is complete domain-lifted for 1-WFOMC

  ... but not for e.g., $\text{friends}(X,Y) \Rightarrow \text{friends}(Y,X)$

  $\text{parent}(X,Y) \Rightarrow \neg\,\text{parent}(Y,X), X \neq Y,$

  $\leq(\mathrm{X},\mathrm{Y}) \vee \leq(\mathrm{Y},\mathrm{X})$
- Theorem:
  CR2 is complete domain-lifted for 2-WFOMC

Importance of Completeness Results

- These are sufficient conditions for domain-lifted inference ("liftability")
- First completeness result so far for lifted probabilistic inference
- 2-WFOMC is a non-trivial class of models
  - (anti-)symmetric, total relations are useful concepts
  - CR1 could already lift more than previous methods
  - CR2 can lift even more, now all of 2-WFOMC
- Open question: other classes?

Outline

- Probabilistic Logic
- Lifted Inference
- Compilation Algorithm
- Completeness
- Conclusions

Conclusions

3 contributions:

1) A formal framework for lifted inference
(definition + complexity considerations)

2) New compilation rule for first-order knowledge compilation

3) First completeness result in lifted inference

Take-away message: **2-WFOMC is liftable.**
This is the first non-trivial class of problems.

Poster today!

Website & Implementation:
http://dtai.cs.kuleuven.be/ml/systems/wfomc