Lifted Probabilistic Inference by First-Order Knowledge Compilation

Guy Van den Broeck
Nima Taghipour
Wannes Meert
Jesse Davis
Luc De Raedt

Lifted Inference in Probabilistic Logical Models - Tutorial - IJCAI11
18/07/11

Outline

- Overview Approach
- First-Order d-DNNF Circuits
- First-Order Knowledge Compilation
- Experiments
- Conclusions

Outline

- Overview Approach
- First-Order d-DNNF Circuits
- First-Order Knowledge Compilation
- Experiments
- Conclusions

Context

	Variable Elimination	Belief Propagation	Knowledge Compilation
Ground	[Zhang94]	[Pearl82]	[Darwiche03]
Lifted	[Poole03]	[Singla08]	Our approach

Advantages of Knowledge Compilation

- Compile once, then run polytime inference for multiple queries and evidence
- Efficient data structures
- Principled logical approach
- Exploits context-specific independences
- State of the art for exact inference in
  - Bayesian networks
  - Statistical relational learning
- Used in many domains, not just probabilistic reasoning

Question?

- Can we lift knowledge compilation to a first-order setting?
- First step taken: first-order d-DNNFs for
  - weighted first-order model counting
  - lifted probabilistic inference
- Many open questions remaining!

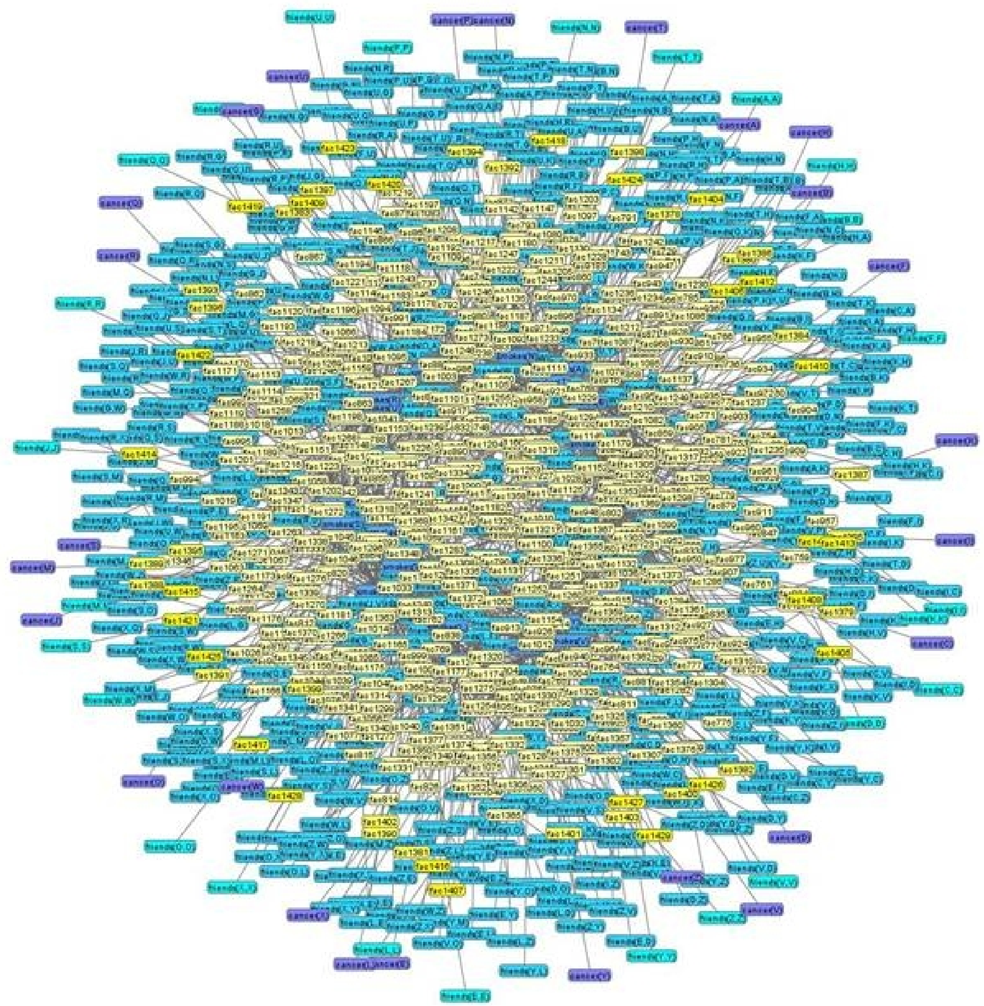
What is Lifted Inference?

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

What is Lifted Inference?

$$2 \;\text{friends}(X,Y) \land \text{smokes}(X) \Rightarrow \text{smokes}(Y)$$

- Variables X,Y range over domain People
- Represents propositional model for given domain (50 people)
- Propositional inference in factor graph is expensive
- However: symmetries

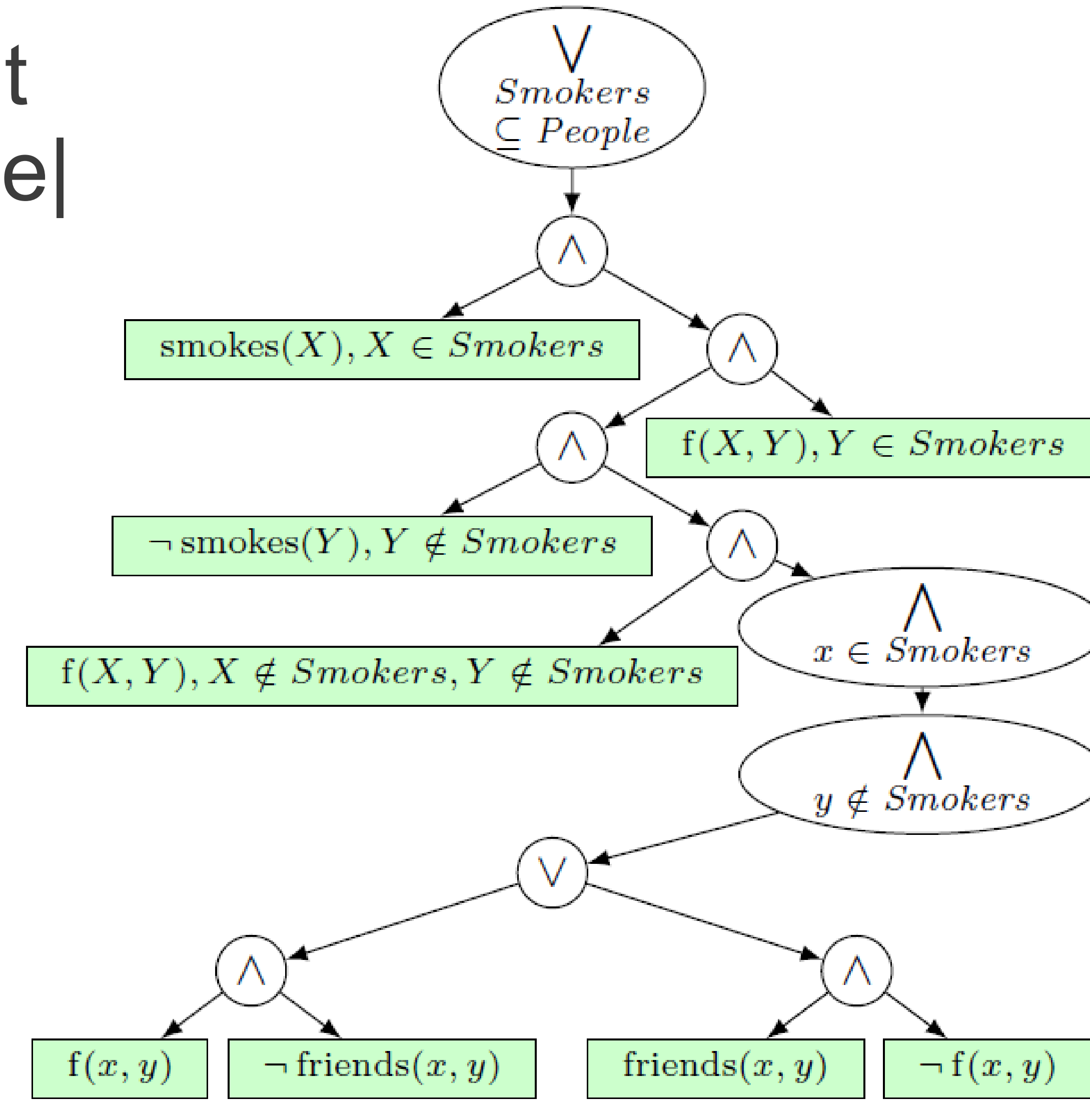


What is Lifted Inference?

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

- We compile to a circuit independent of |People|
- Inference linear in |People|

→ Lifted Inference



Knowledge Compilation

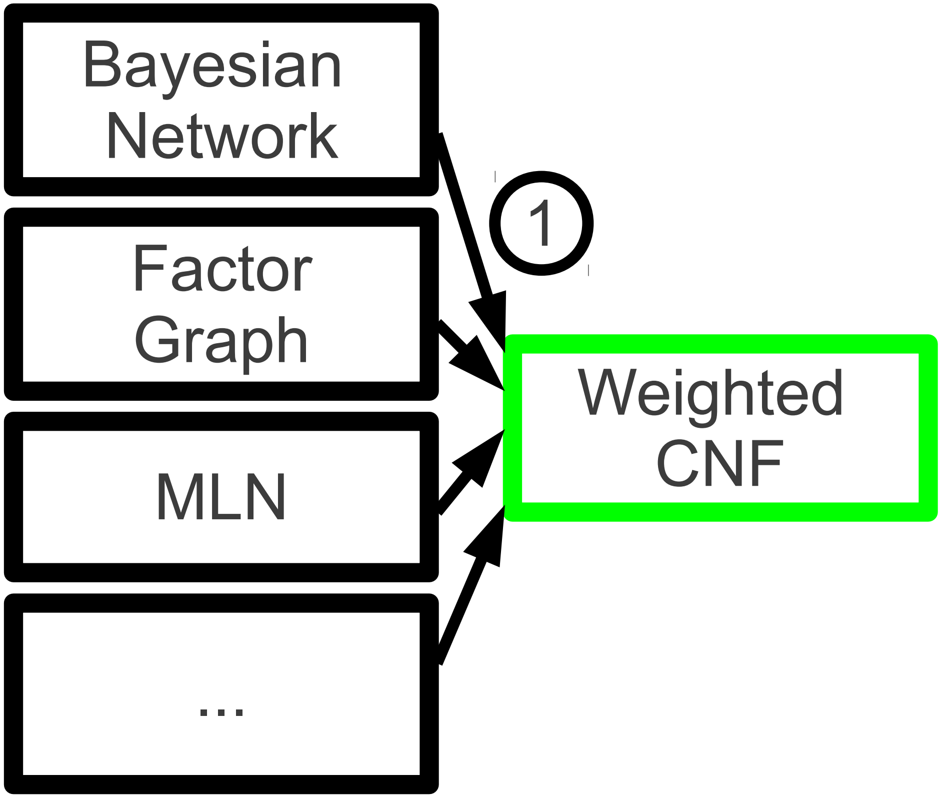
Bayesian Network

Factor Graph

MLN

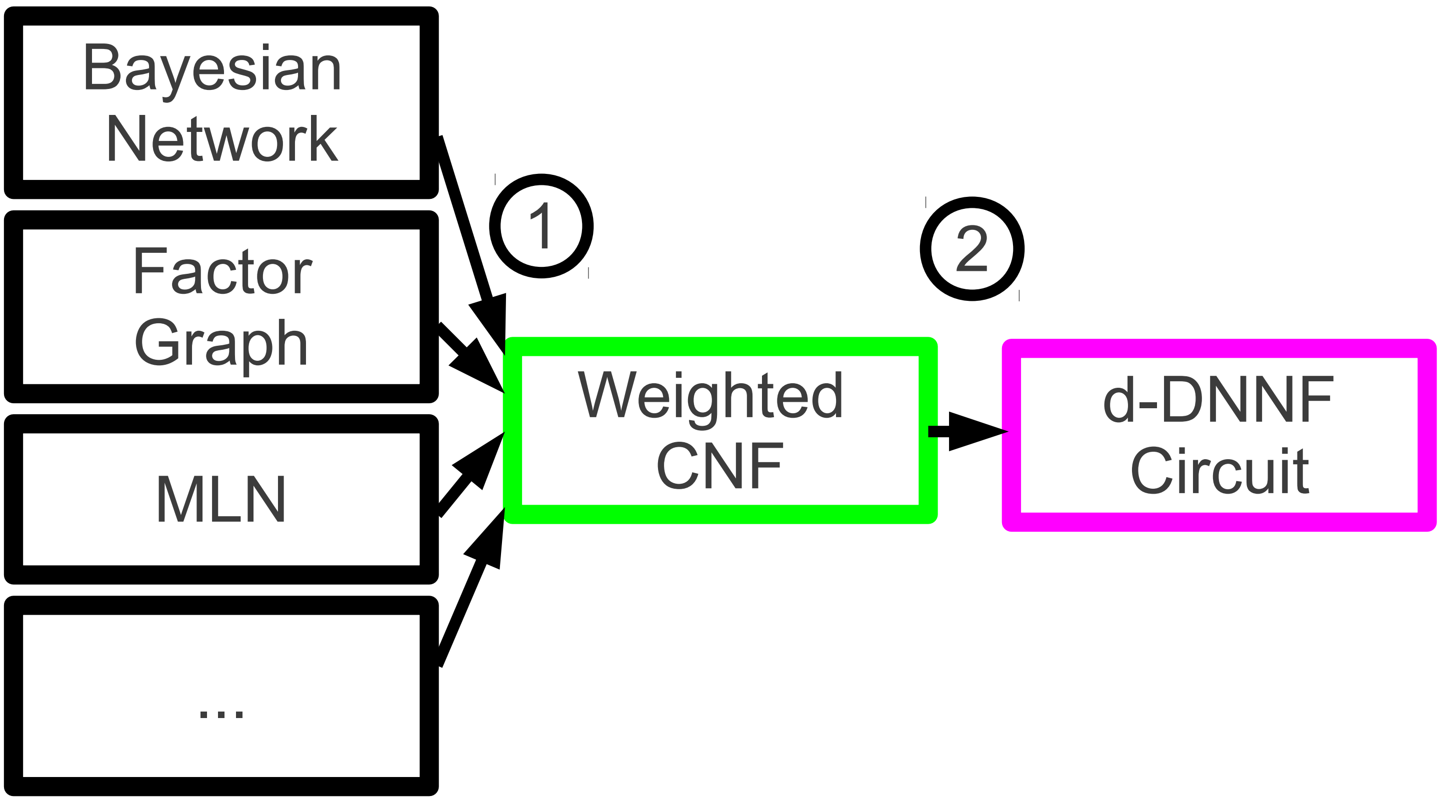
...

Knowledge Compilation



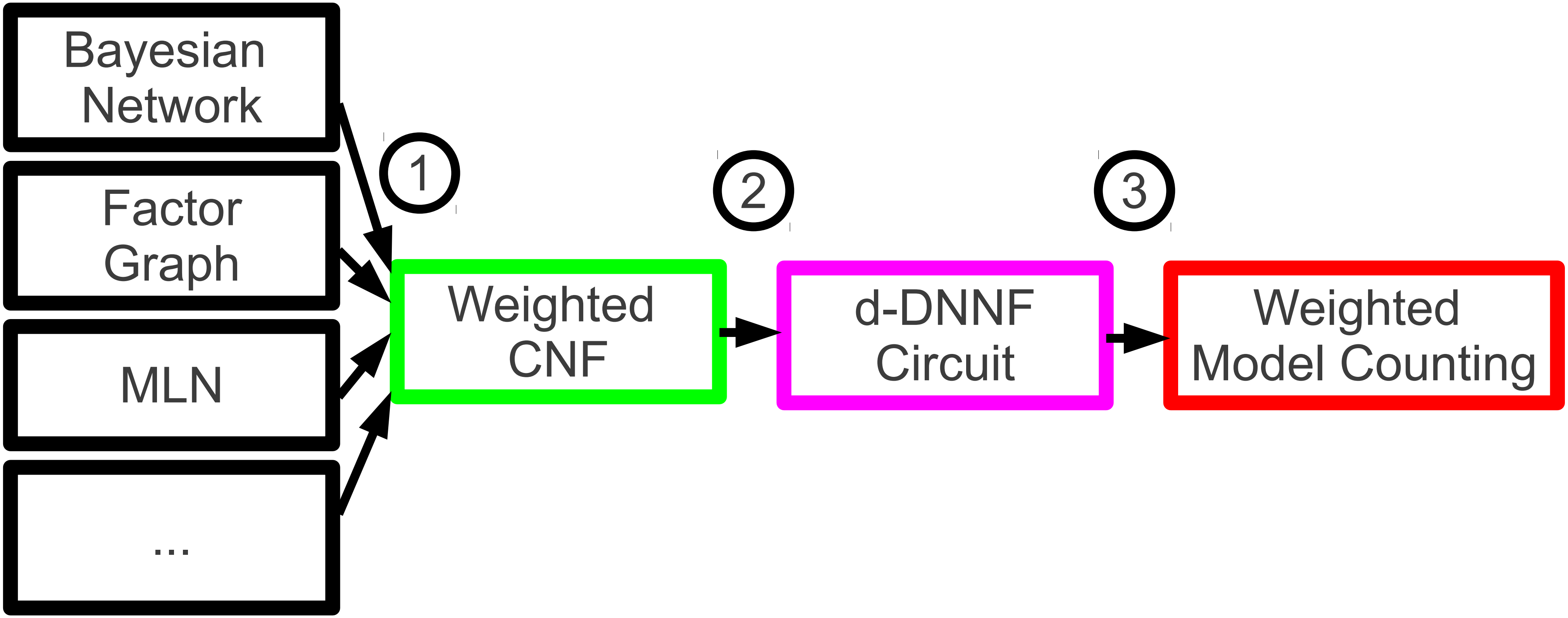
- Step ① : Convert model to weighted CNF

Knowledge Compilation



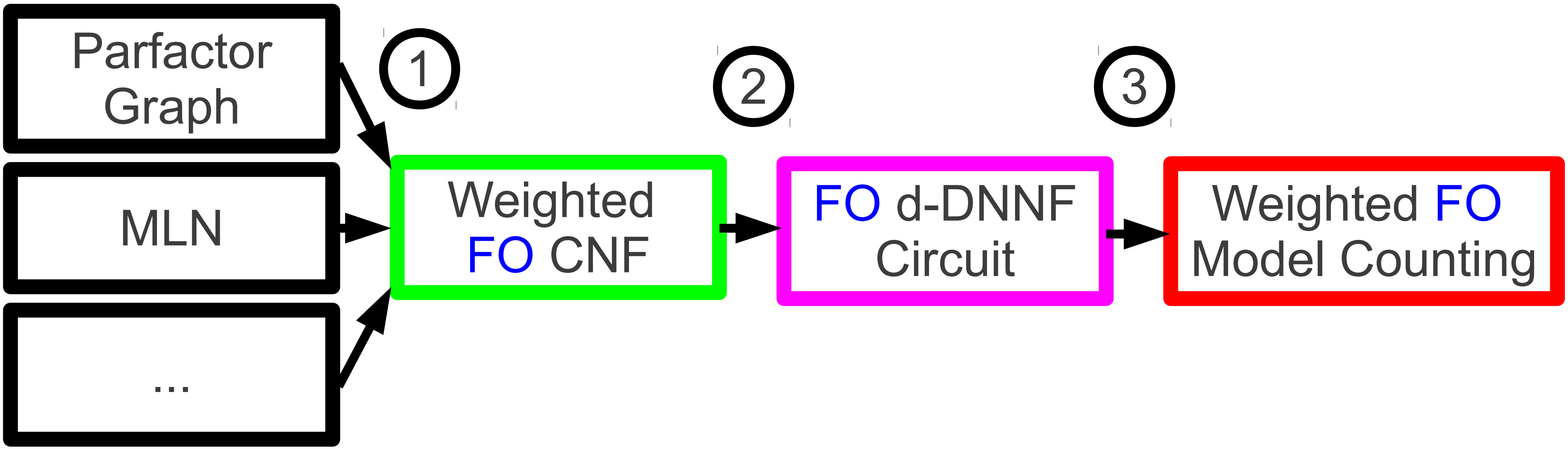
- Step 1: Convert model to weighted CNF
- Step 2: Convert CNF to d-DNNF circuit

Knowledge Compilation



- Step 1: Convert model to weighted CNF
- Step 2: Convert CNF to d-DNNF circuit
- Step 3: Perform weighted model counting

Our Approach: First-Order Knowledge Compilation:



- Step 1: Convert model to weighted FO CNF
- Step 2: Convert CNF to FO d-DNNF circuit
- Step 3: Perform weighted FO model counting

Step 1: Converting to Weighted FO CNF

MLN

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

Step 1: Converting to Weighted FO CNF

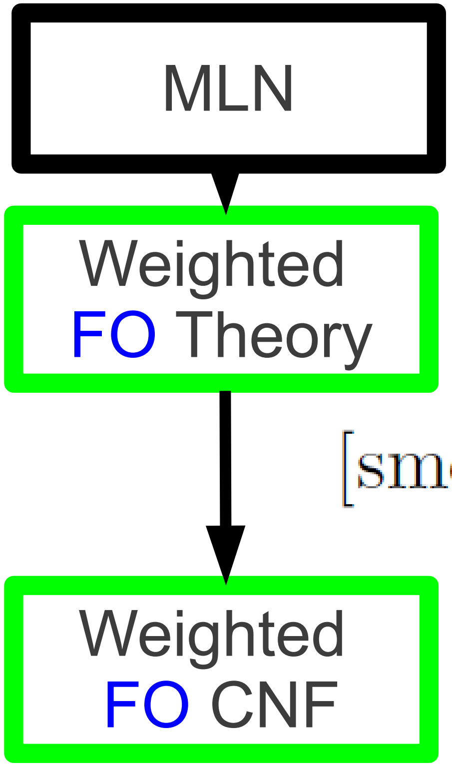
MLN

Weighted FO Theory

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

$$[\text{smokes}(X) \wedge \text{friends}(X,Y) \Rightarrow \text{smokes}(Y)] \equiv \text{f}(X,Y)$$

Step 1: Converting to Weighted FO CNF



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

$$[\text{smokes}(X) \wedge \text{friends}(X, Y) \Rightarrow \text{smokes}(Y)] \equiv \text{f}(X, Y)$$

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

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

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

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

Step 3: Weighted FO Model Counting

- Weight function on ground atoms

$$\mathrm{w}(\mathrm{smokes}(X)) = 2 \qquad \mathrm{w}(\neg\,\mathrm{smokes}(X)) = 10$$

$$\mathrm{w}(\mathrm{friends}(X, Y)) = 5 \qquad \mathrm{w}(\neg\,\mathrm{friends}(X, Y)) = 1$$

Step 3: Weighted FO Model Counting

- Weight function on ground atoms

$$\mathrm{w}(\mathrm{smokes}(X)) = 2 \qquad \mathrm{w}(\neg\,\mathrm{smokes}(X)) = 10$$

$$\mathrm{w}(\mathrm{friends}(X,Y)) = 5 \qquad \mathrm{w}(\neg\,\mathrm{friends}(X,Y)) = 1$$

- Weight of a model (possible world)

$$\underbrace{\{\underbrace{\mathrm{smokes}(alice)}_{2}, \underbrace{\neg\,\mathrm{smokes}(bob)}_{10}, \underbrace{\mathrm{friends}(alice,bob)}_{5}, \ldots\}}_{2\cdot 10\cdot 5\cdot \ldots = 100}$$

Step 3: Weighted FO Model Counting

- Weight function on ground atoms

$$\mathrm{w}(\mathrm{smokes}(X)) = 2 \qquad \mathrm{w}(\neg\,\mathrm{smokes}(X)) = 10$$

$$\mathrm{w}(\mathrm{friends}(X,Y)) = 5 \qquad \mathrm{w}(\neg\,\mathrm{friends}(X,Y)) = 1$$

- Weight of a model (possible world)

$$\{\underbrace{\underbrace{\mathrm{smokes}(alice)}_{2}, \underbrace{\neg\,\mathrm{smokes}(bob)}_{10}, \underbrace{\mathrm{friends}(alice,bob)}_{5}, \ldots}_{2\cdot 10\cdot 5\cdot \ldots = 100}\}$$

- Weight of all models is $100 + \ldots = Z$

Step 3: Weighted FO Model Counting

- Weight function on ground atoms

$$\mathrm{w}(\mathrm{smokes}(X)) = 2 \qquad \mathrm{w}(\neg\,\mathrm{smokes}(X)) = 10$$

$$\mathrm{w}(\mathrm{friends}(X,Y)) = 5 \qquad \mathrm{w}(\neg\,\mathrm{friends}(X,Y)) = 1$$

- Weight of a model (possible world)

$$\underbrace{\{\underbrace{\mathrm{smokes}(alice)}_{2}, \underbrace{\neg\,\mathrm{smokes}(bob)}_{10}, \underbrace{\mathrm{friends}(alice,bob)}_{5}, \ldots\}}_{2\cdot 10\cdot 5\cdot\ldots=100}$$

- Weight of all models is $100 + \ldots = Z$

  Weight of models where Alice smokes is $100 + \ldots = Q$

Step 3: Weighted FO Model Counting

- Weight function on ground atoms

$$\mathrm{w}(\mathrm{smokes}(X)) = 2 \qquad \mathrm{w}(\neg\,\mathrm{smokes}(X)) = 10$$

$$\mathrm{w}(\mathrm{friends}(X,Y)) = 5 \qquad \mathrm{w}(\neg\,\mathrm{friends}(X,Y)) = 1$$

- Weight of a model (possible world)

$$\underbrace{\{\underbrace{\mathrm{smokes}(alice)}_{2}, \underbrace{\neg\,\mathrm{smokes}(bob)}_{10}, \underbrace{\mathrm{friends}(alice, bob)}_{5}, \ldots\}}_{2 \cdot 10 \cdot 5 \cdot \ldots = 100}$$

- Weight of all models is $100 + \ldots = Z$

  Weight of models where Alice smokes is $100 + \ldots = Q$

- $\mathrm{P}(\mathrm{smokes}(alice)) = \dfrac{Q}{Z}$

Outline

- Overview Approach
- First-Order d-DNNF Circuits
- First-Order Knowledge Compilation
- Experiments
- Conclusions

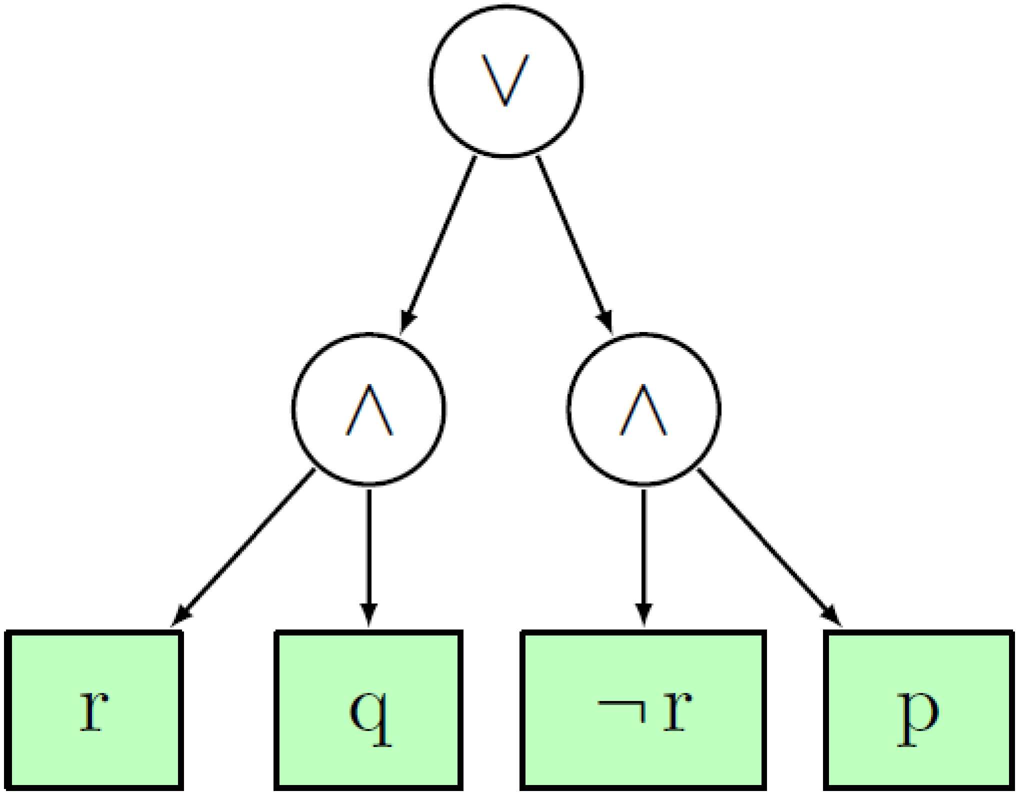
Propositional d-DNNF Circuits

[Darwiche and Marquis, 2002]

Logical theory:

$$p \vee r$$

$$q \vee \neg r$$



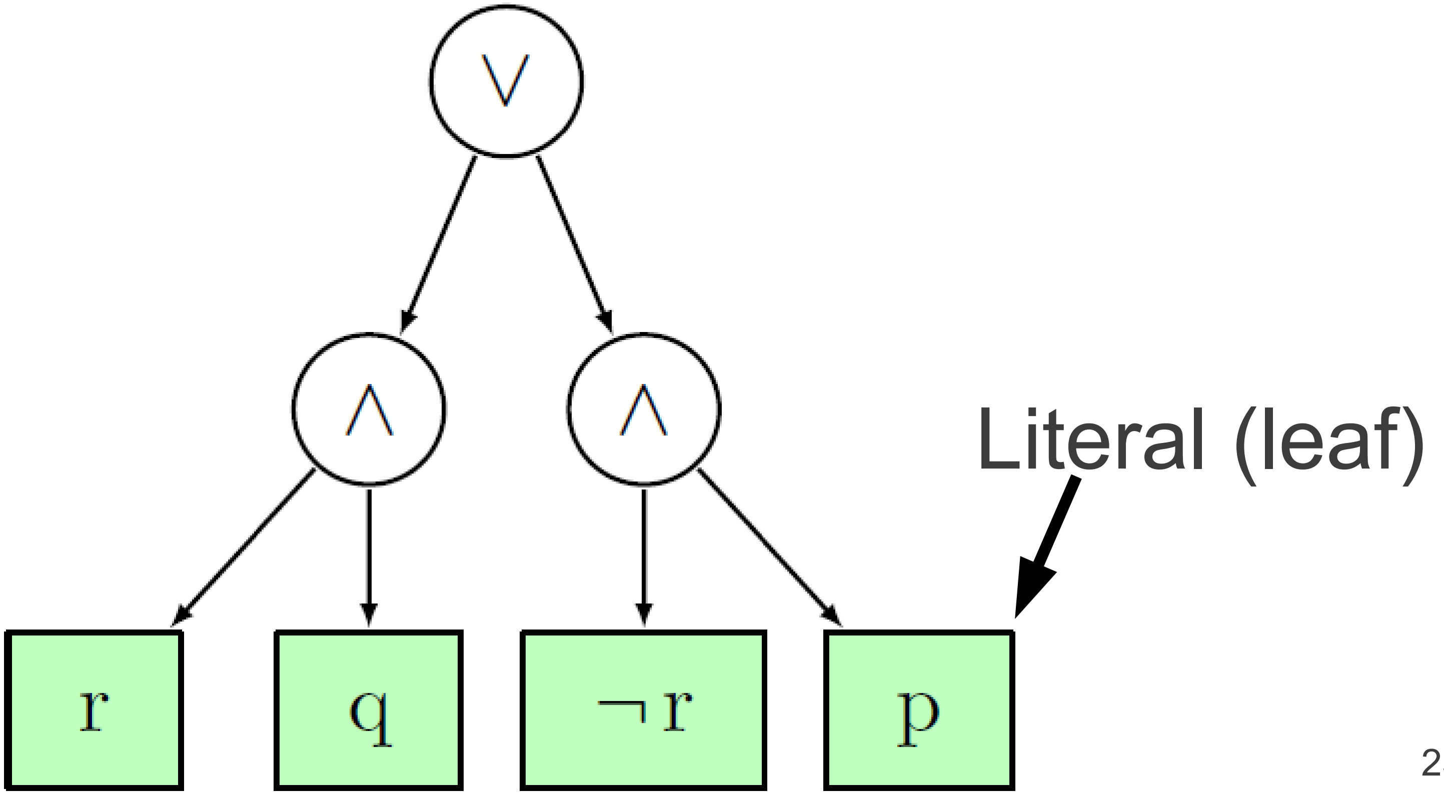
Propositional d-DNNF Circuits

[Darwiche and Marquis, 2002]

Logical theory:

$p \vee r$

$q \vee \neg r$



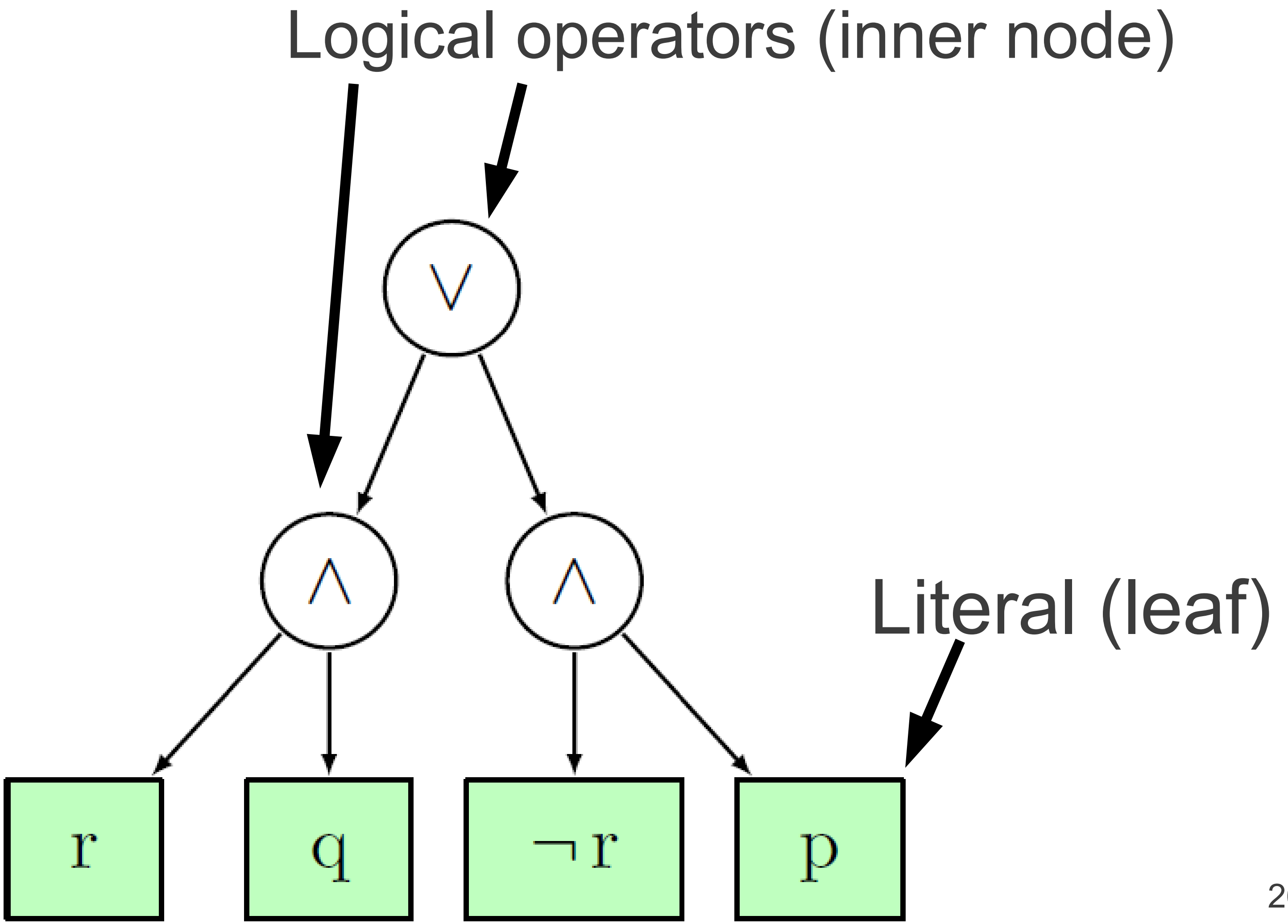
Propositional d-DNNF Circuits

[Darwiche and Marquis, 2002]

Logical theory:

$p \vee r$

$q \vee \neg r$



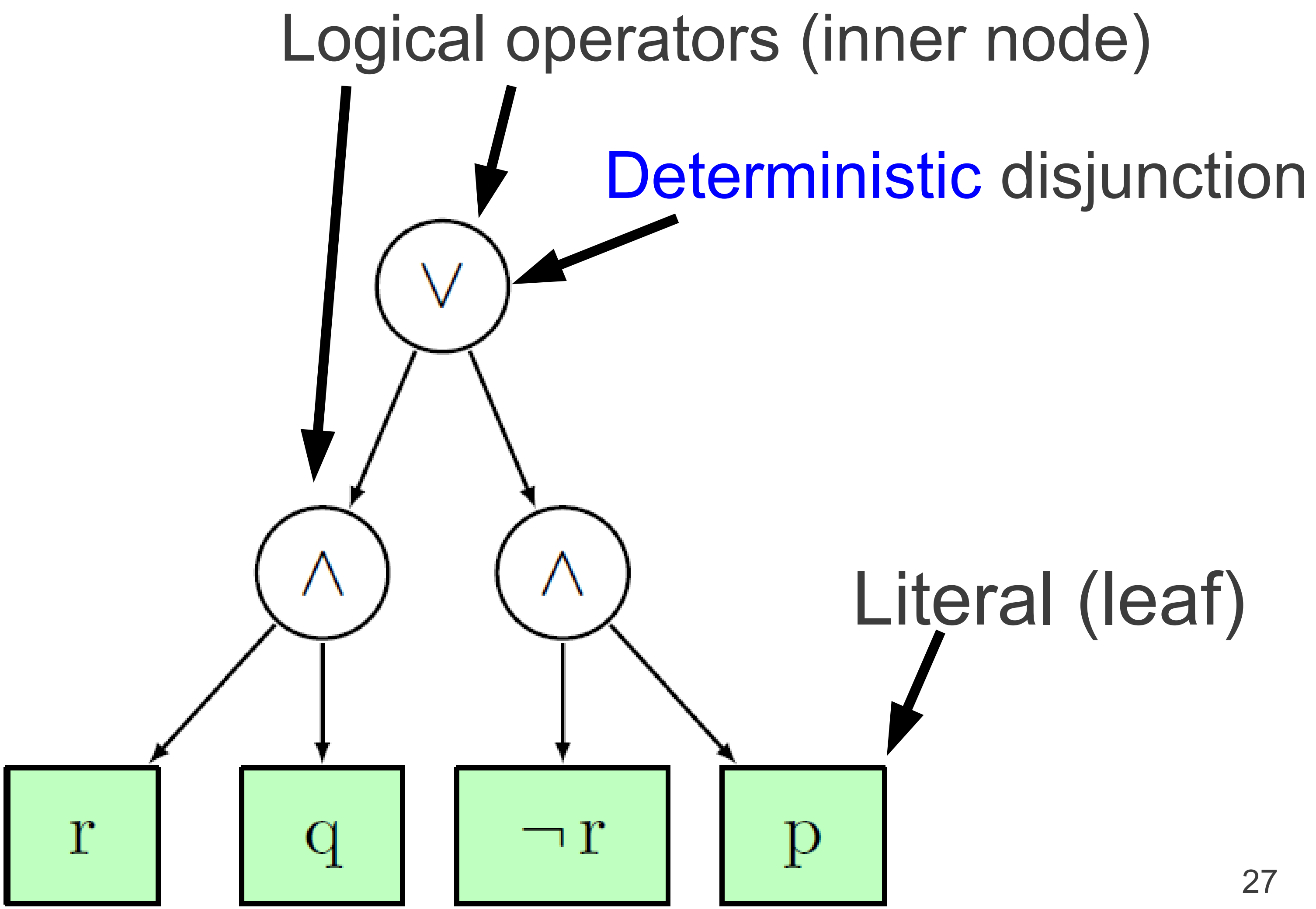
Propositional d-DNNF Circuits

[Darwiche and Marquis, 2002]

Logical theory:

$$p \vee r$$

$$q \vee \neg r$$



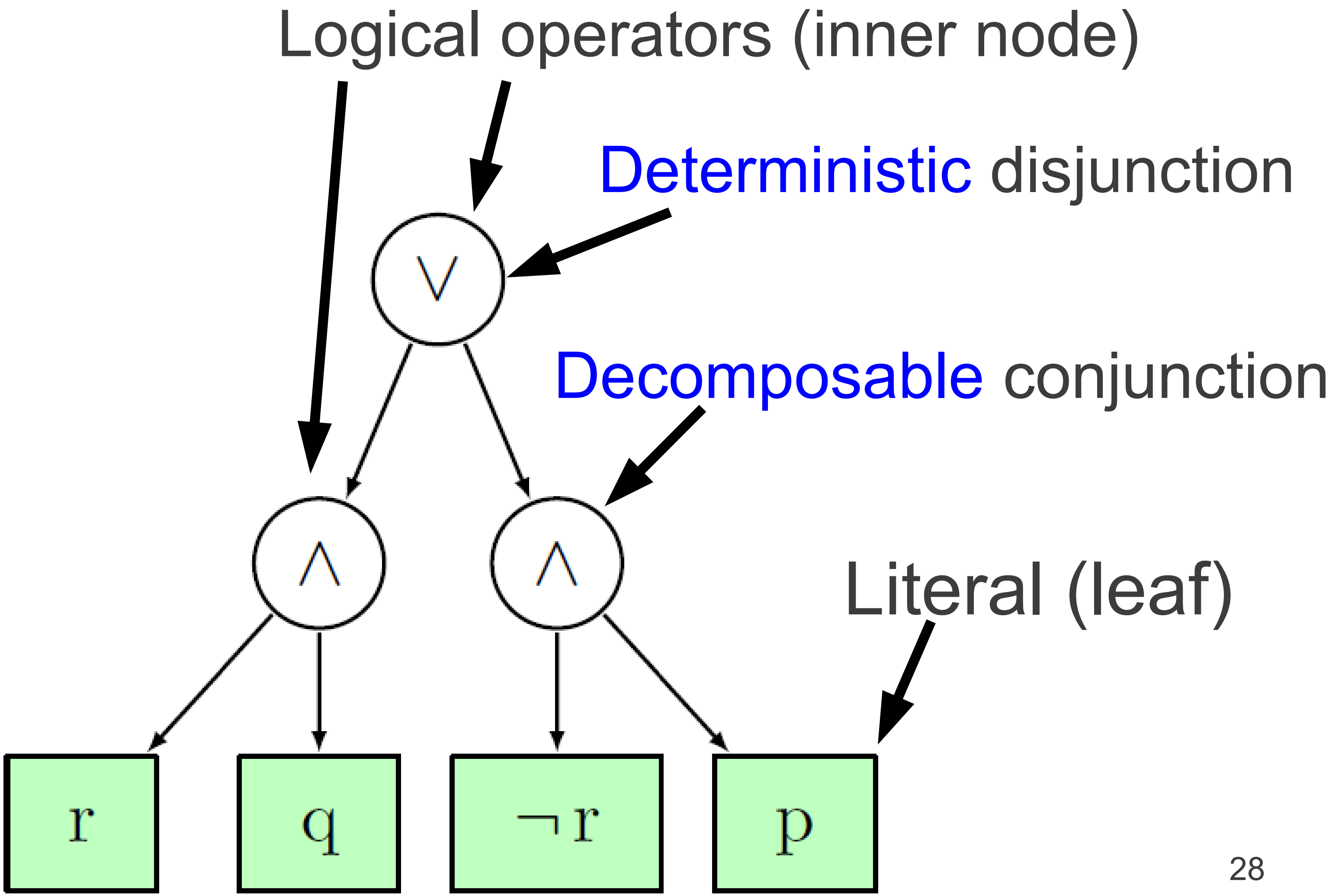
Propositional d-DNNF Circuits

[Darwiche and Marquis, 2002]

Logical theory:

$$p \vee r$$

$$q \vee \neg r$$

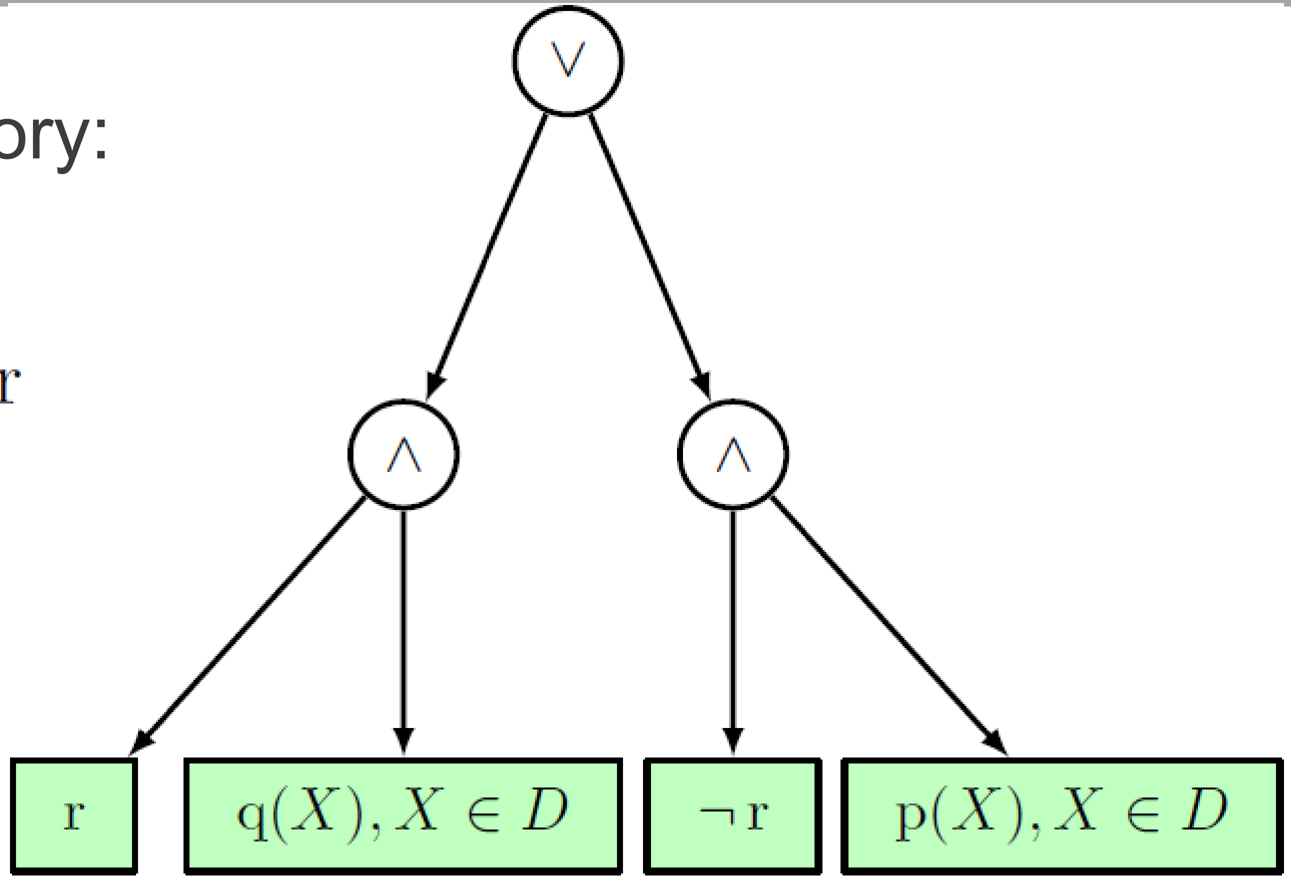


First-Order d-DNNF Circuits

Logical Theory:

$$\mathrm{p}(X) \lor \mathrm{r}$$

$$\mathrm{q}(X) \lor \neg\,\mathrm{r}$$

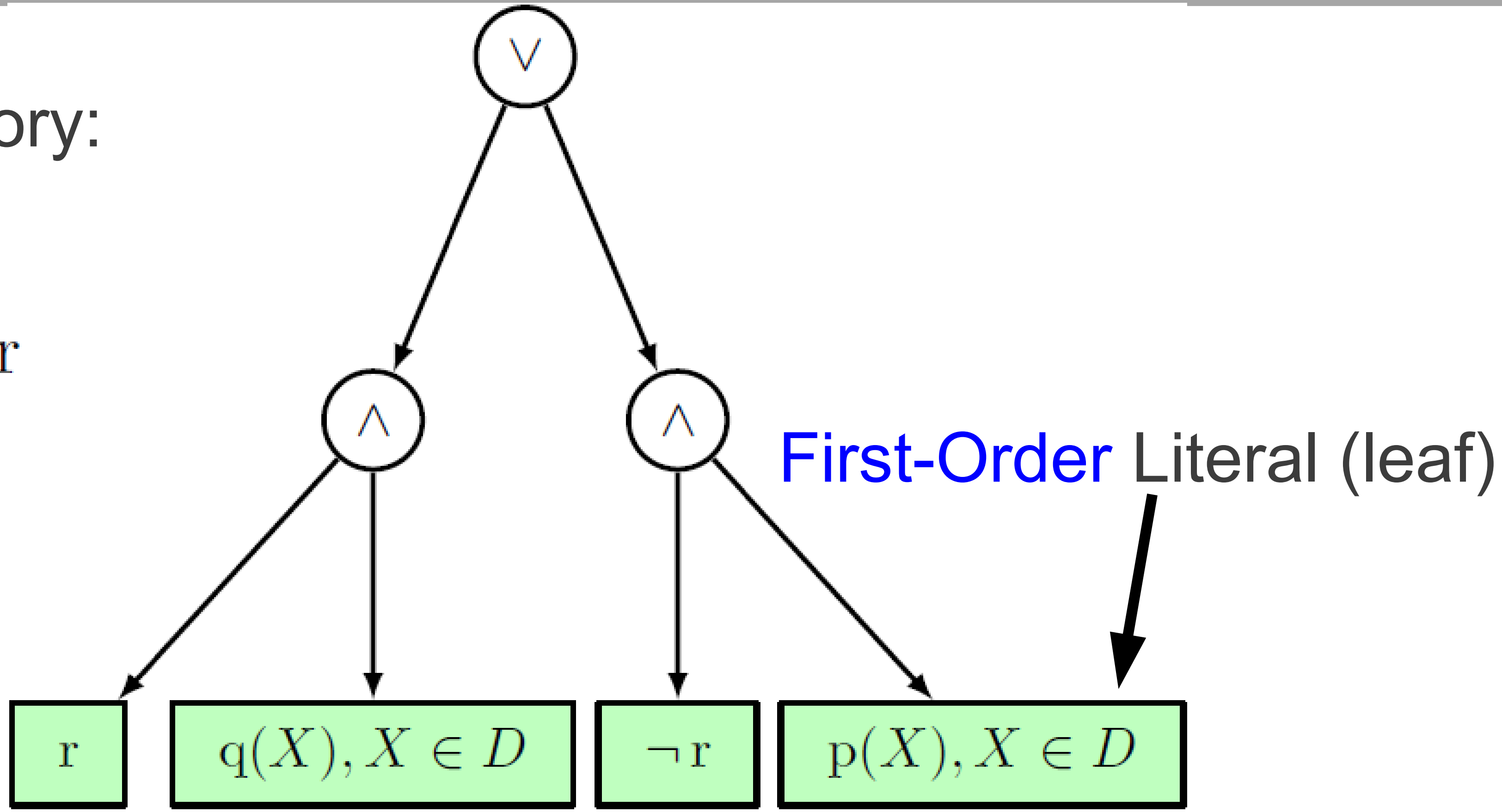


First-Order d-DNNF Circuits

Logical Theory:

$$\mathrm{p}(X) \vee \mathrm{r}$$

$$\mathrm{q}(X) \vee \neg\, \mathrm{r}$$

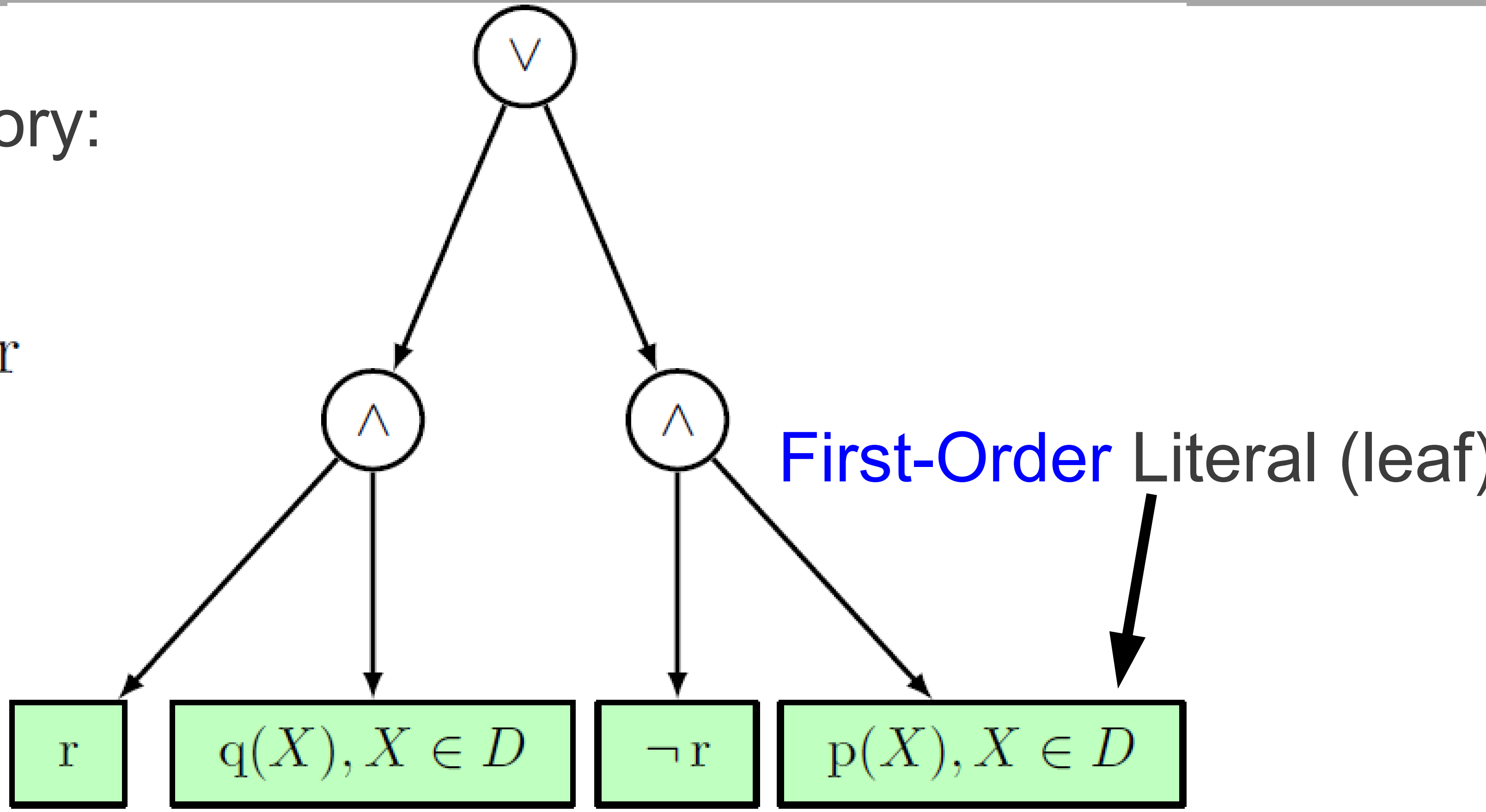


First-Order d-DNNF Circuits

Logical Theory:

$$p(X) \lor r$$

$$q(X) \lor \neg r$$

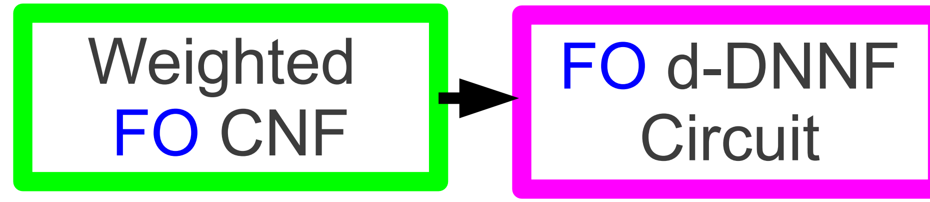


- Deterministic disjunction
- Decomposable conjunction
- 3 additional first-order operators (inner nodes)

Outline

- Overview Approach
- First-Order d-DNNF Circuits
- First-Order Knowledge Compilation
- Experiments
- Conclusions

Step 2: Our Compilation Algorithm



- Recursively apply
  - Unit Propagation
  - Independence
  - Inclusion-Exclusion (Shannon Decomposition)
  - Shattering
  - Independent Partial Grounding
  - Atom Counting
  - (Grounding)

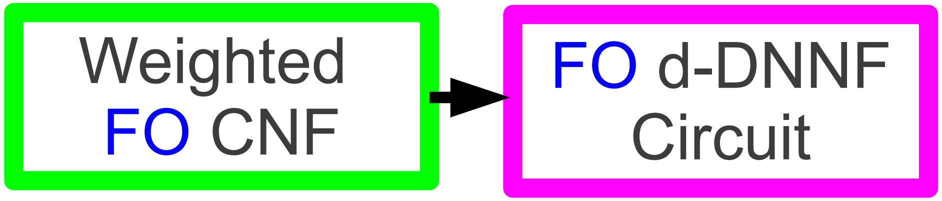
Step 2: Our Compilation Algorithm

Weighted FO CNF → FO d-DNNF Circuit

- Recursively apply
  - Unit Propagation
  - Independence
  - Inclusion-Exclusion (Shannon Decomposition)
  - Shattering
  - Independent Partial Grounding
  - Atom Counting
  - (Grounding)

Generate first-order operators in inner nodes

Step 2: Our Compilation Algorithm



- Recursively apply
  - Unit Propagation
  - Independence
  - Inclusion-Exclusion (Shannon Decomposition)
  - Shattering
  - Independent Partial Grounding
  - Atom Counting
  - (Grounding)

Unit Propagation

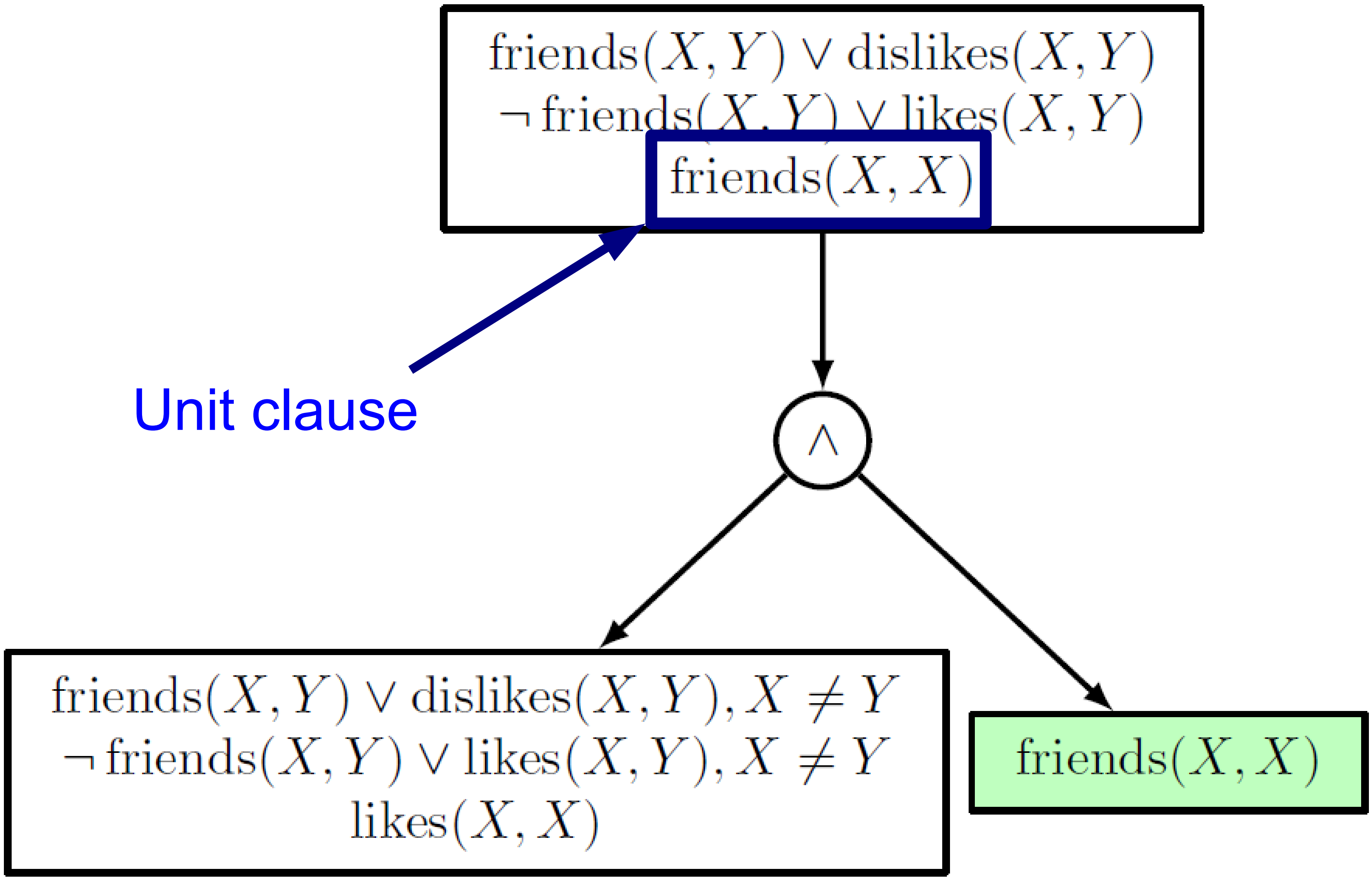
$$\text{friends}(X, Y) \vee \text{dislikes}(X, Y)$$
$$\neg\,\text{friends}(X, Y) \vee \text{likes}(X, Y)$$
$$\text{friends}(X, X)$$

Unit Propagation

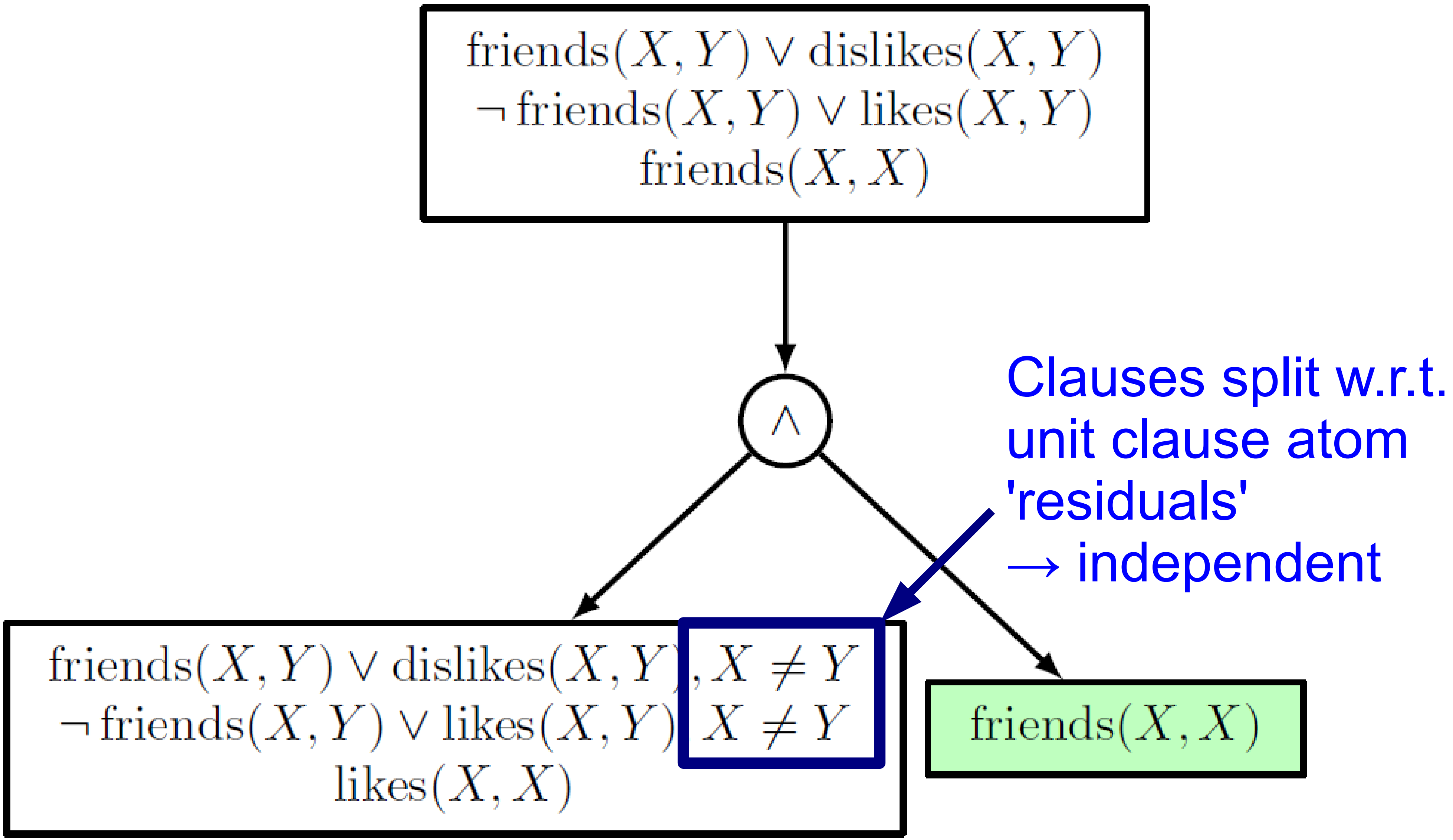
$$\text{friends}(X,Y) \vee \text{dislikes}(X,Y)$$
$$\neg\,\text{friends}(X,Y) \vee \text{likes}(X,Y)$$
$$\text{friends}(X,X)$$

Unit clause

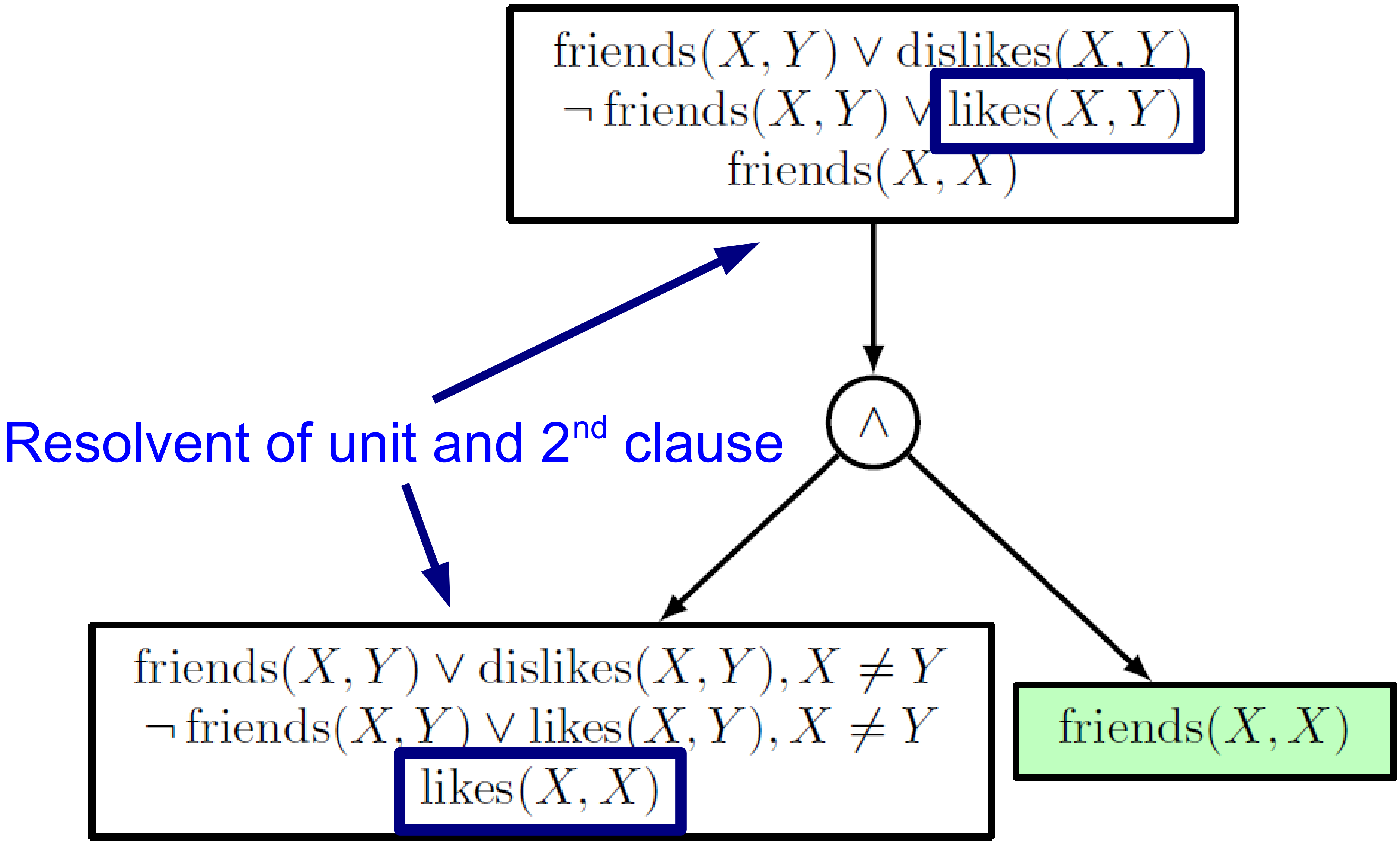
Unit Propagation



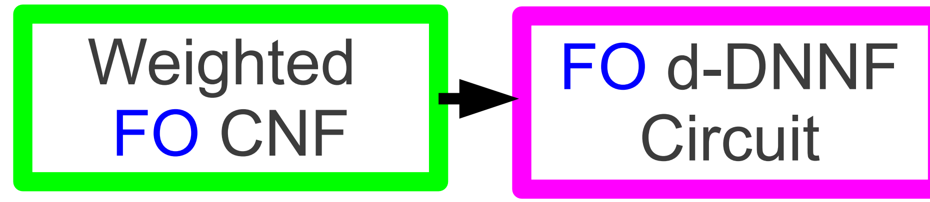
Unit Propagation



Unit Propagation



Step 2: Our Compilation Algorithm



- Recursively apply
  - Unit Propagation
  - Independence
  - Inclusion-Exclusion (Shannon Decomposition)
  - Shattering
  - Independent Partial Grounding
  - Atom Counting
  - (Grounding)

Atom Counting

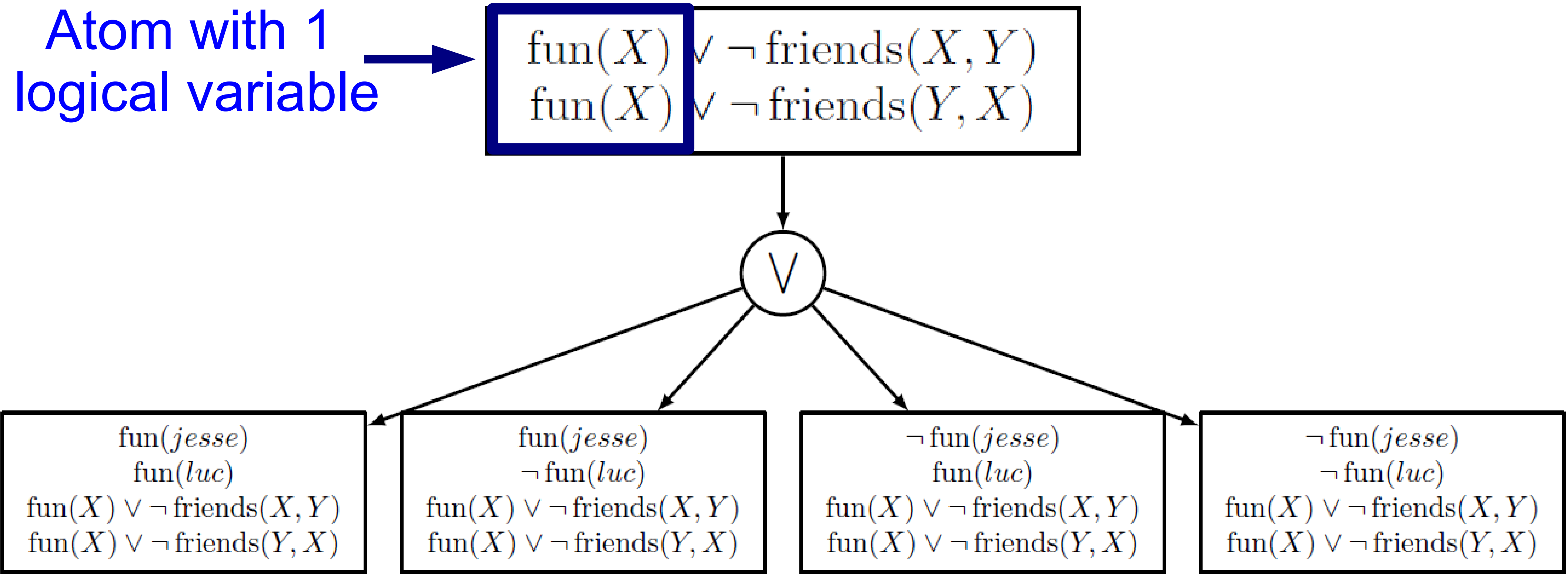
$$\mathrm{fun}(X) \lor \neg\,\mathrm{friends}(X,Y)$$
$$\mathrm{fun}(X) \lor \neg\,\mathrm{friends}(Y,X)$$

Atom Counting

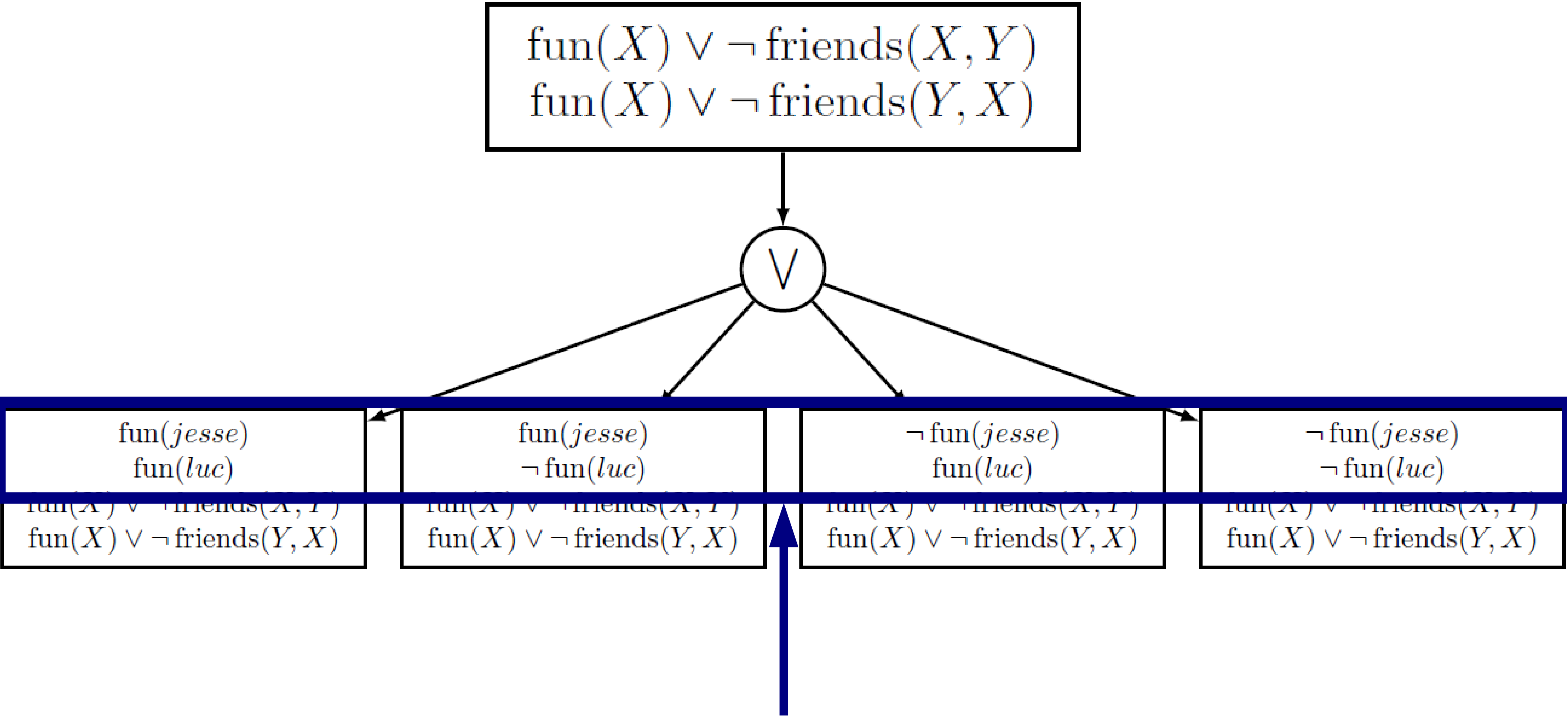
Atom with one logical variable
X ∈ {luc,jesse}

$$\mathrm{fun}(X) \lor \neg\,\mathrm{friends}(X,Y)$$
$$\mathrm{fun}(X) \lor \neg\,\mathrm{friends}(Y,X)$$

Atom Counting



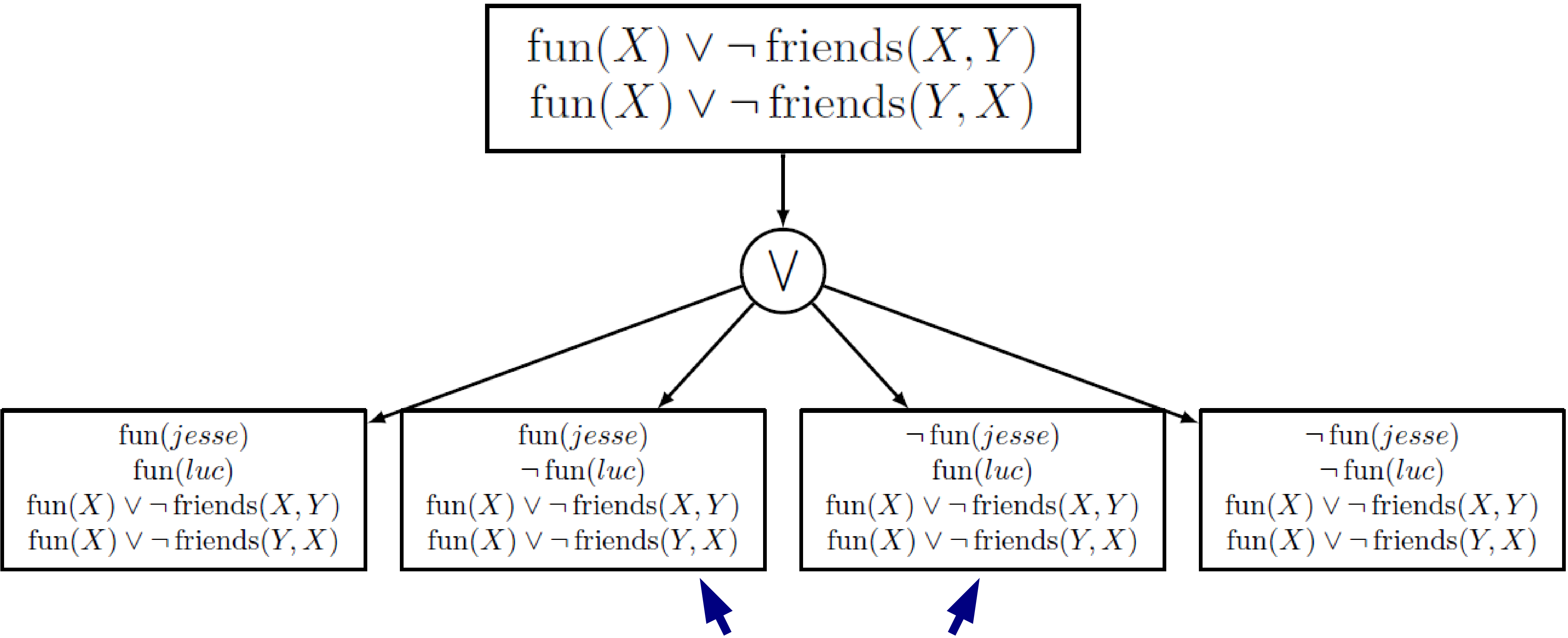
Atom Counting



All partial interpretations for fun(X)

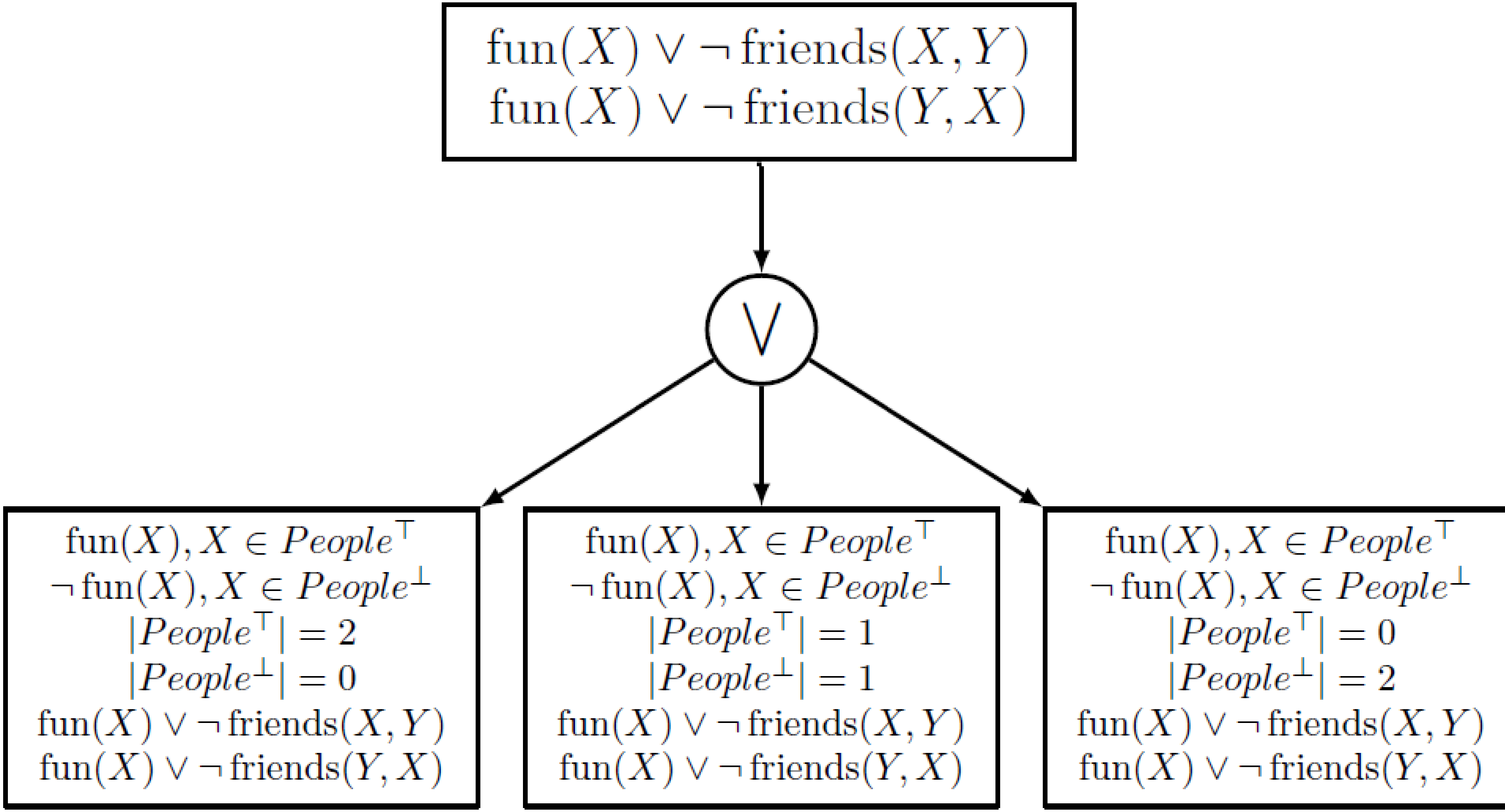
- deterministic
- $2^{|People|}$

Atom Counting



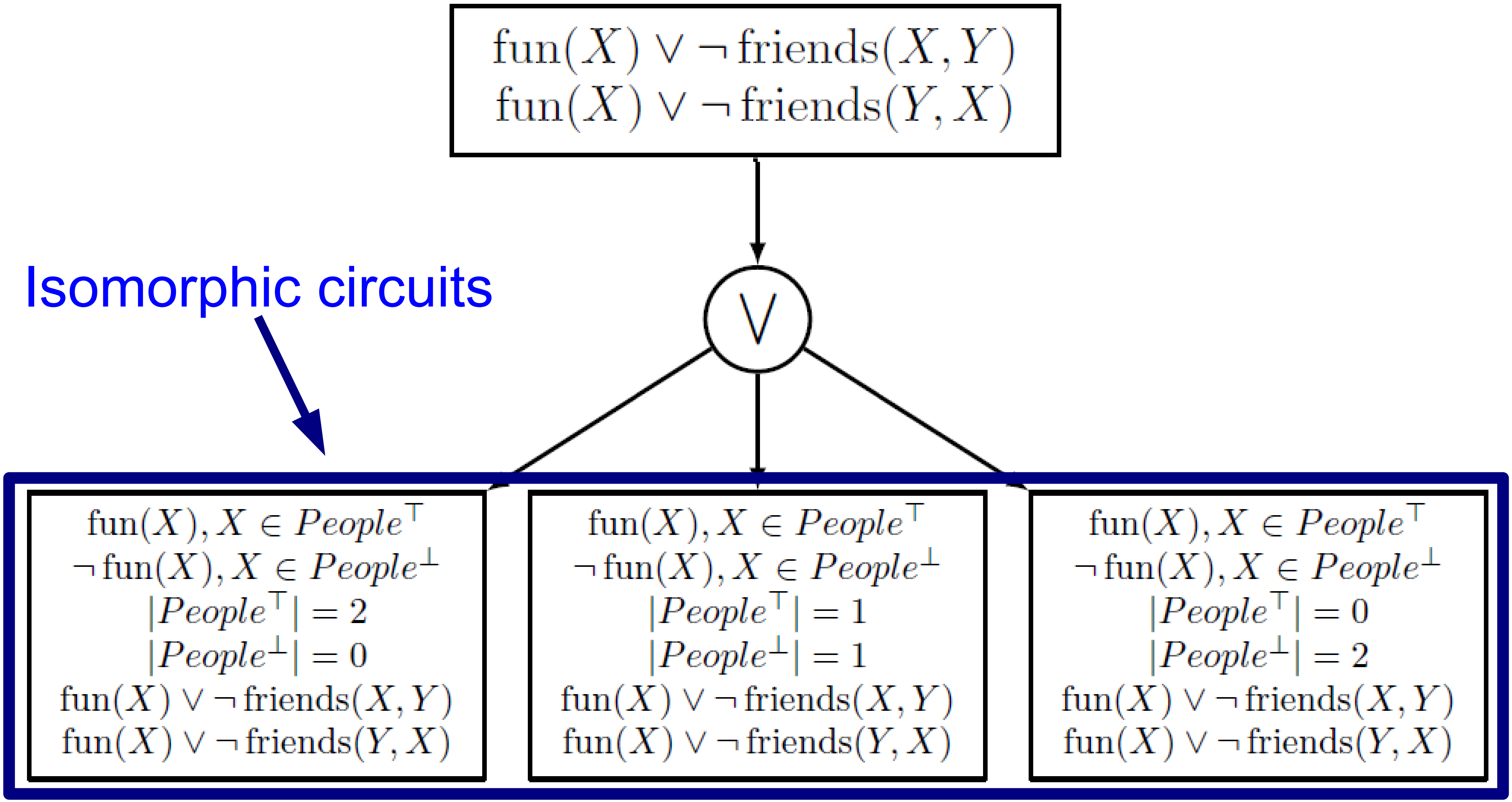
Same weighted model count

Atom Counting

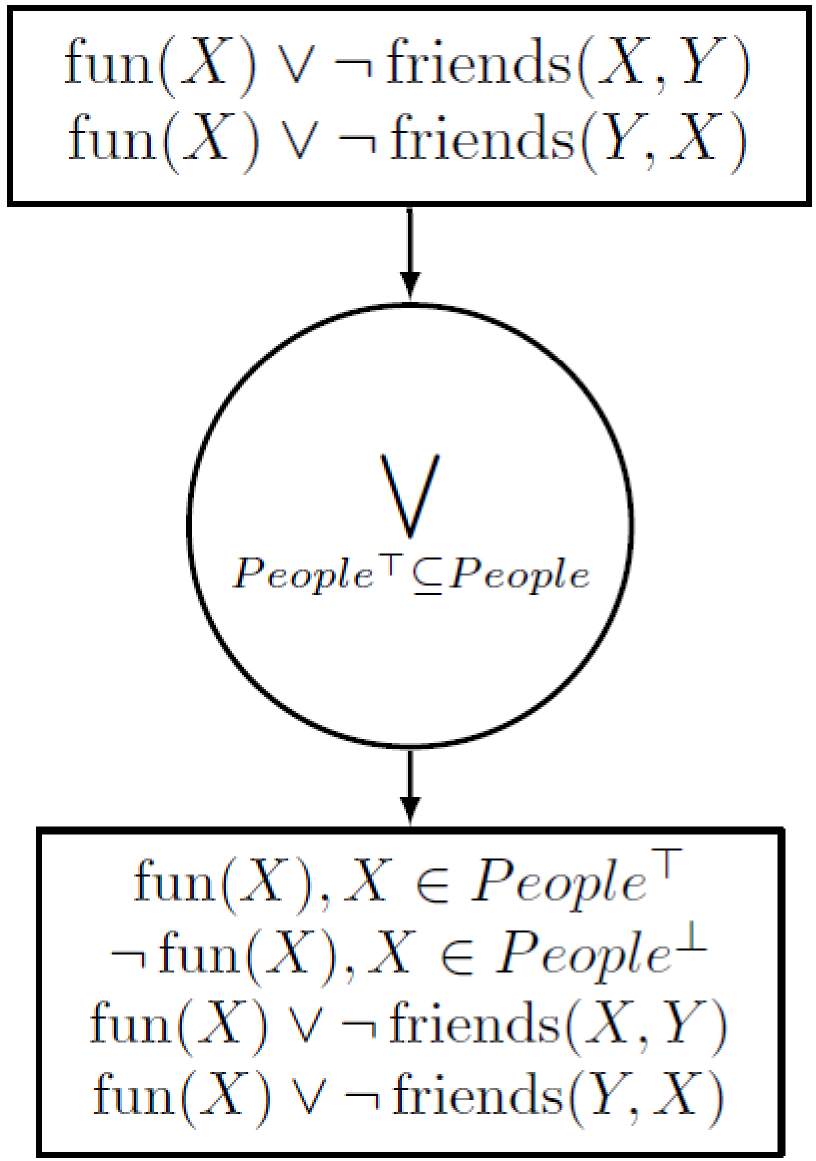


$2^{|People|} \rightarrow |People|+1$

Atom Counting

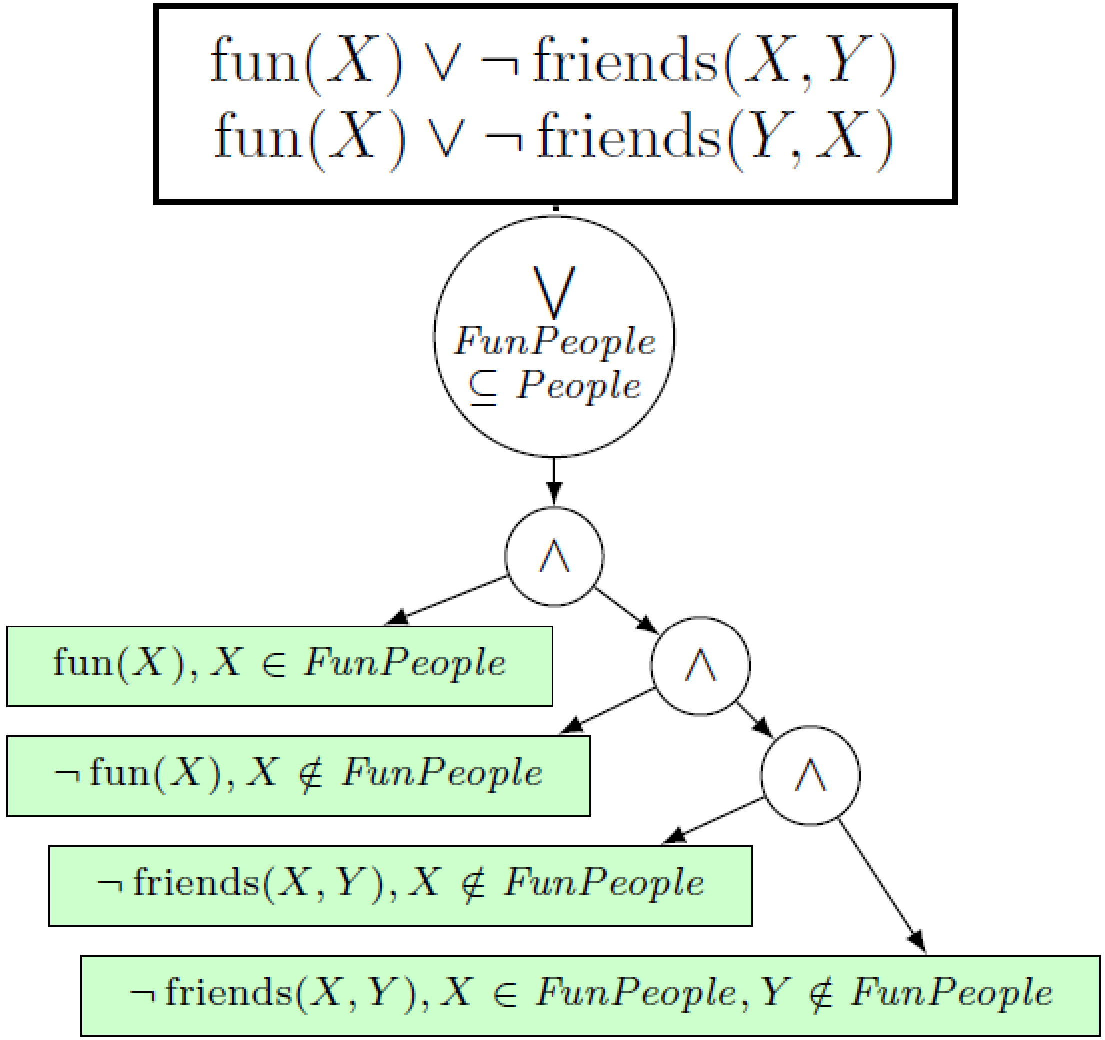


Atom Counting



|People|+1 → 1

Atom Counting



Outline

- Overview Approach
- First-Order d-DNNF Circuits
- First-Order Knowledge Compilation
- Experiments
- Conclusions

Evaluated Models

- Sick Death [de Salvo Braz 2005]
- WebKB [Lowd 2007]
- Competing Workshops [Milch 2008]
- Workshop Attributes [Milch 2008]
- Friends Smoker [Singla 2008]
- Friends Smoker Drinker

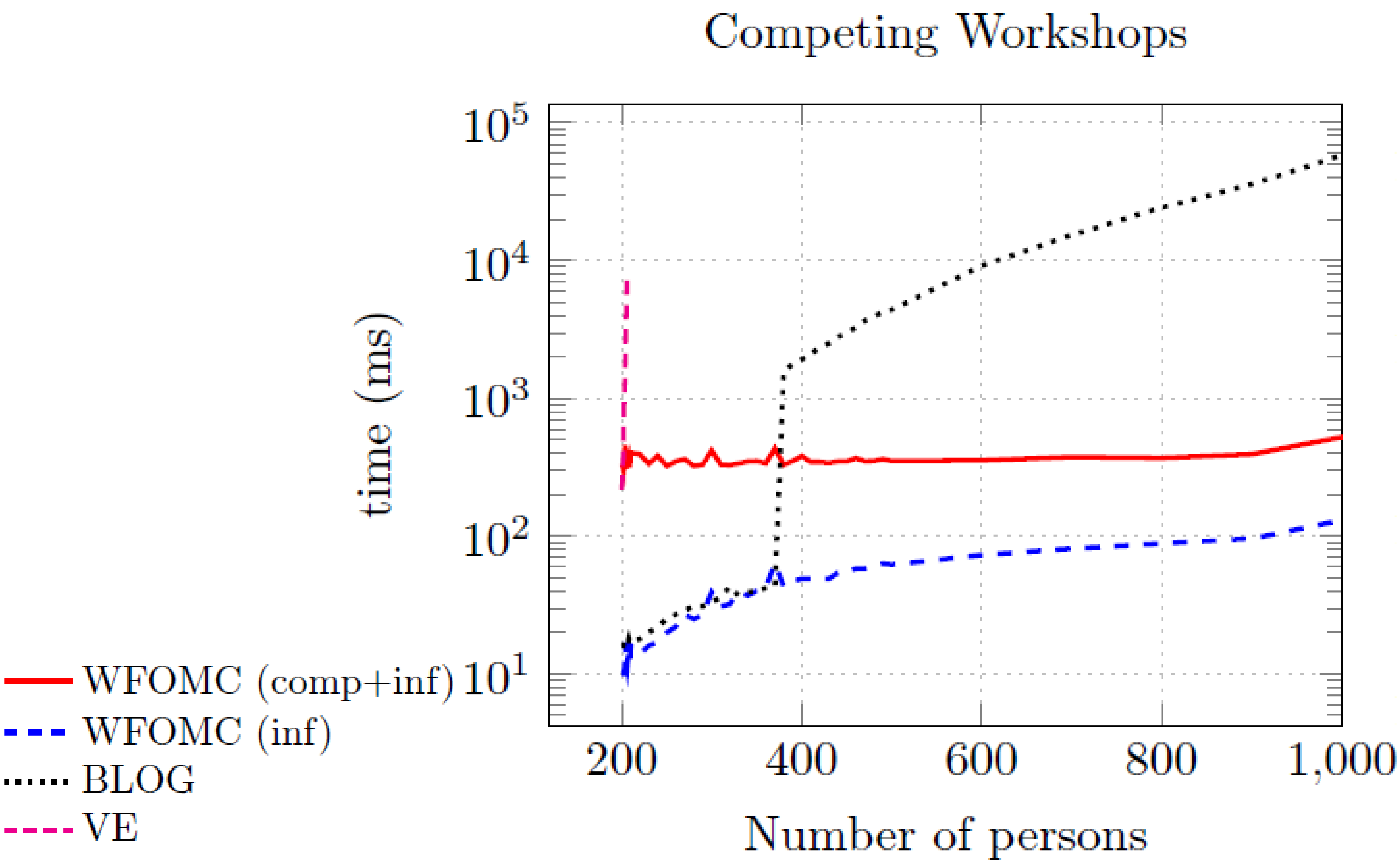
Competing Workshops [Milch 2008]

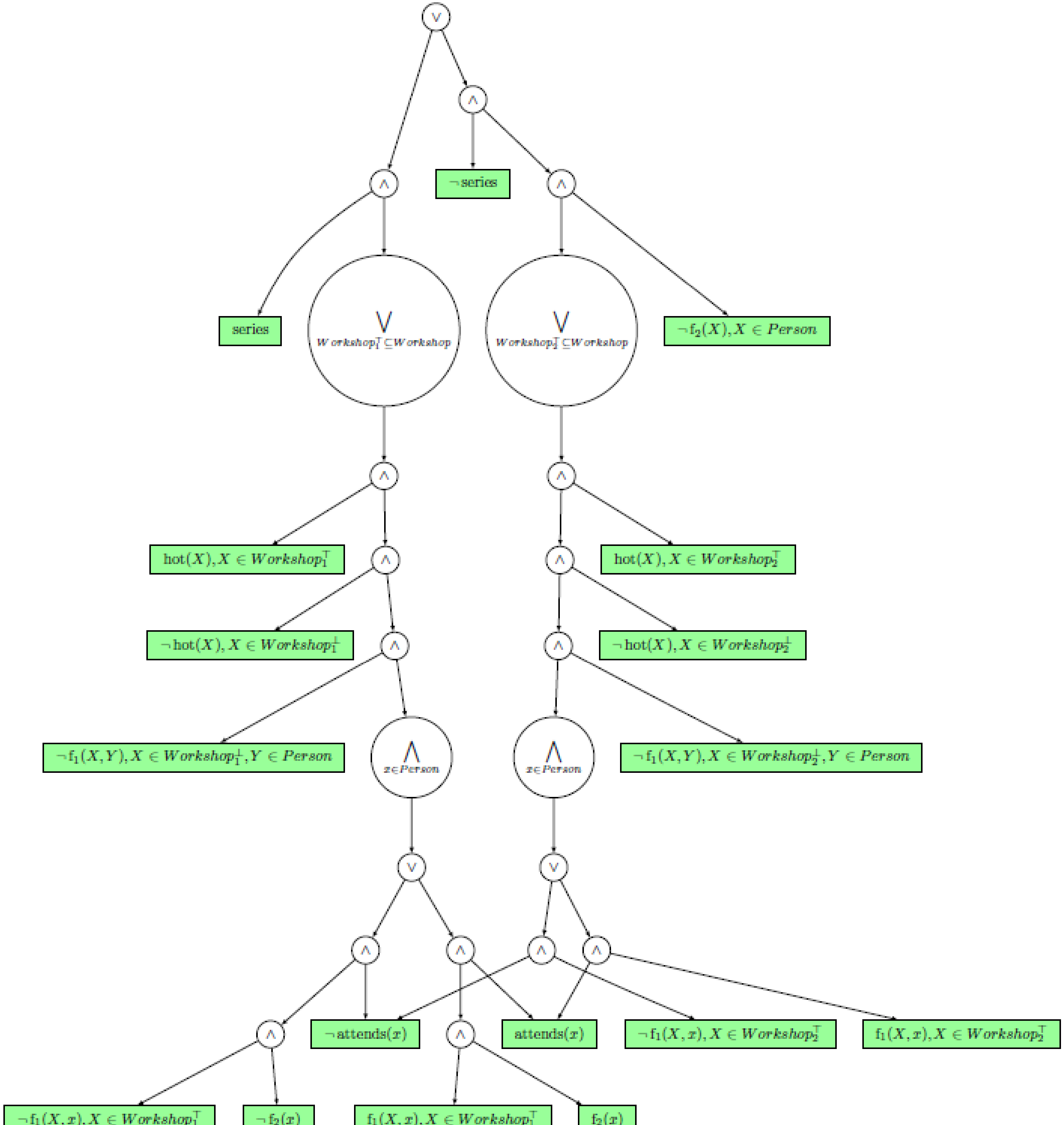
Probabilistic Model:

$$-2 : \mathrm{hot}(W) \wedge \mathrm{attends}(P)$$

$$3 : \mathrm{attends}(P) \wedge \mathrm{series}$$

Competing Workshops [Milch 2008]




series
¬series
$\neg f_2(X), X \in Person$
$\bigvee_{Workshop_1^\top \subseteq Workshop}$
$\bigvee_{Workshop_2^\top \subseteq Workshop}$
$hot(X), X \in Workshop_1^\top$
$hot(X), X \in Workshop_2^\top$
$\neg hot(X), X \in Workshop_1^\bot$
$\neg hot(X), X \in Workshop_2^\bot$
$\neg f_1(X,Y), X \in Workshop_1^\bot, Y \in Person$
$\bigwedge_{x \in Person}$
$\bigwedge_{x \in Person}$
$\neg f_1(X,Y), X \in Workshop_2^\bot, Y \in Person$
$\neg attends(x)$
$attends(x)$
$\neg f_1(X,x), X \in Workshop_2^\top$
$f_1(X,x), X \in Workshop_2^\top$
$\neg f_1(X,x), X \in Workshop_1^\top$
$\neg f_2(x)$
$f_1(X,x), X \in Workshop_1^\top$
$f_2(x)$

Friends Smoker [Singla 2008]

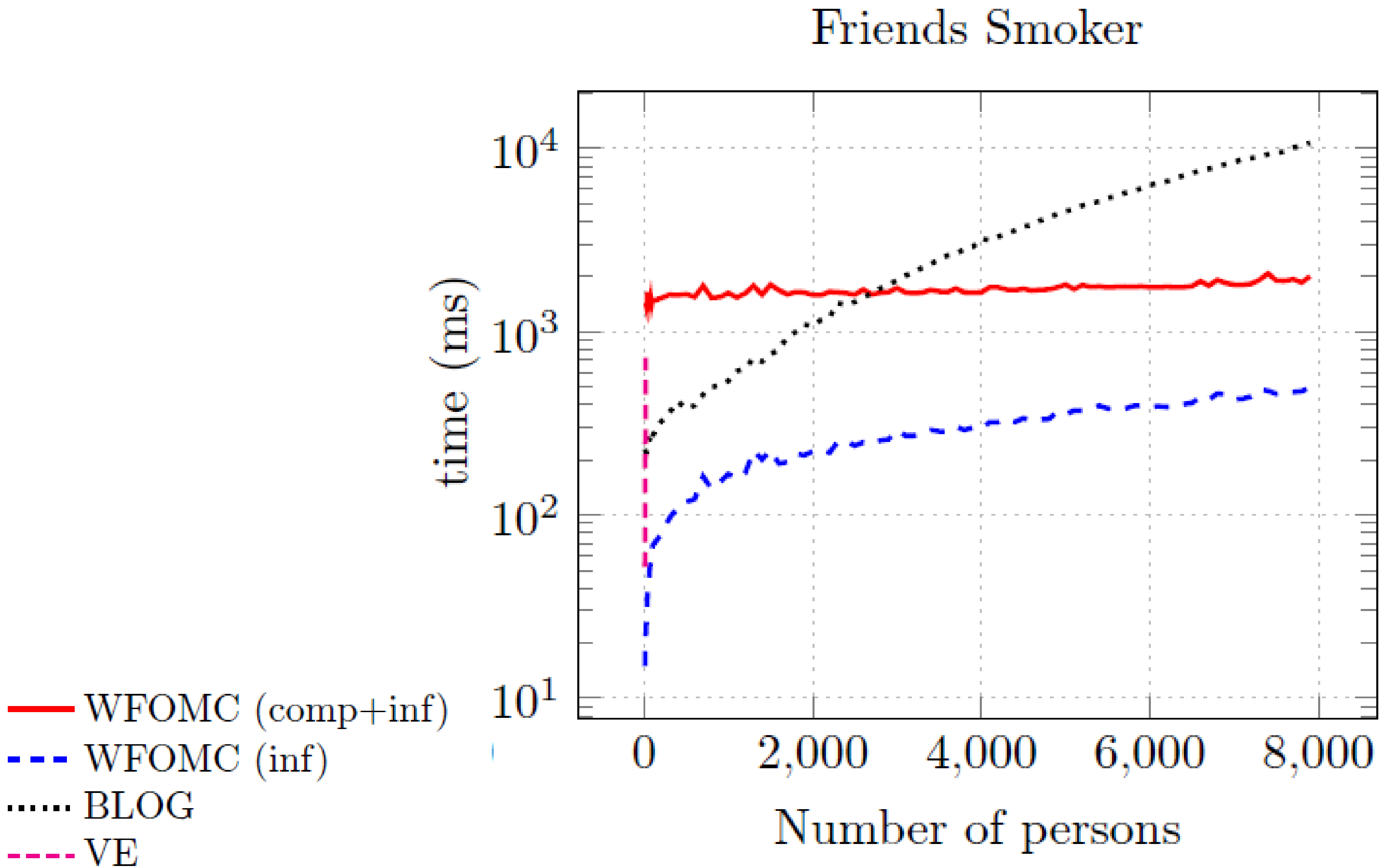
Probabilistic Model:

$$1.2 : \mathrm{smokes}(X) \wedge \mathrm{friends}(X, Y) \Rightarrow \mathrm{smokes}(Y)$$

$$1.2 : \mathrm{smokes}(X) \wedge \mathrm{friends}(Y, X) \Rightarrow \mathrm{smokes}(Y)$$

$$2 : \mathrm{smokes}(X) \Rightarrow \mathrm{cancer}(X)$$

Friends Smoker [Singla 2008]



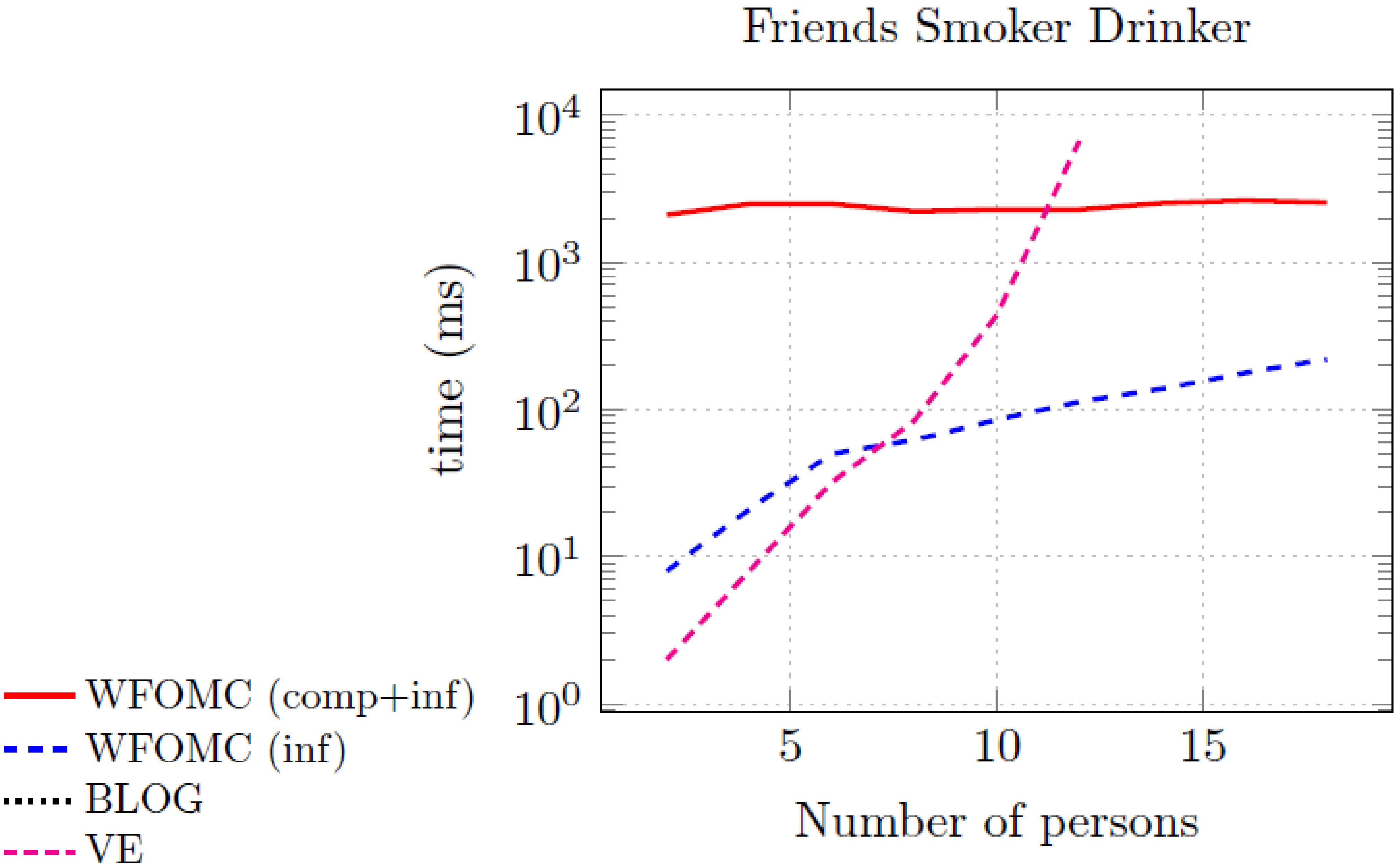
Friends Smoker Drinker

New Probabilistic Model:

$$1.2 : \text{smokes}(X) \wedge \text{friends}(X, Y) \Rightarrow \text{smokes}(Y)$$

$$1.2 : \text{drinks}(X) \wedge \text{friends}(X, Y) \Rightarrow \text{drinks}(Y)$$

Friends Smoker Drinker



Outline

- Overview Approach
- First-Order d-DNNF Circuits
- First-Order Knowledge Compilation
- Experiments
- Conclusions

Benefits of First-Order Knowledge Compilation?

- Compile once for a given set of evidence then run polytime inference
- Efficient data structure
- Principled logical approach
  - First model theoretic approach to lifted probabilistic inference
  - Uses concepts from logical inference: model counting, unit propagation, Shannon decomposition, etc.
- Exploits context-specific independences
- State of the art for exact lifted inference
  - Lifts more models than C-FOVE

Contributions

- We introduced first-order ...
  - knowledge compilation
  - d-DNNF circuits
  - weighted model counting
  - smoothing
- Algorithm to compile a first-order probabilistic model into FO d-DNNF circuits
- Closer to understanding the connection between lifted inference in first-order logic (resolution) and lifted inference in graphical models

Advertisement

- Poster

  Wednesday 10:30 UAI session

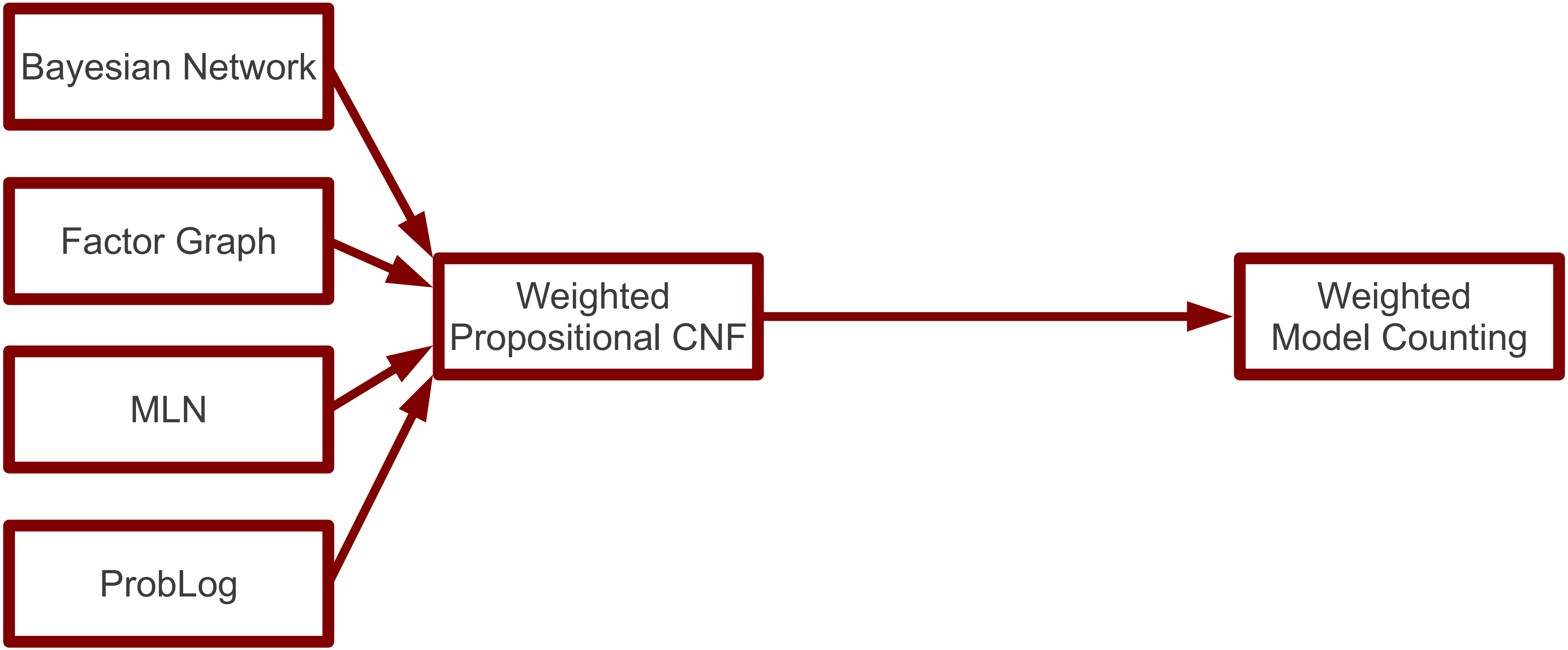
- Talk

  Thursday 10:30 UAI session

Thanks

Extra Slides

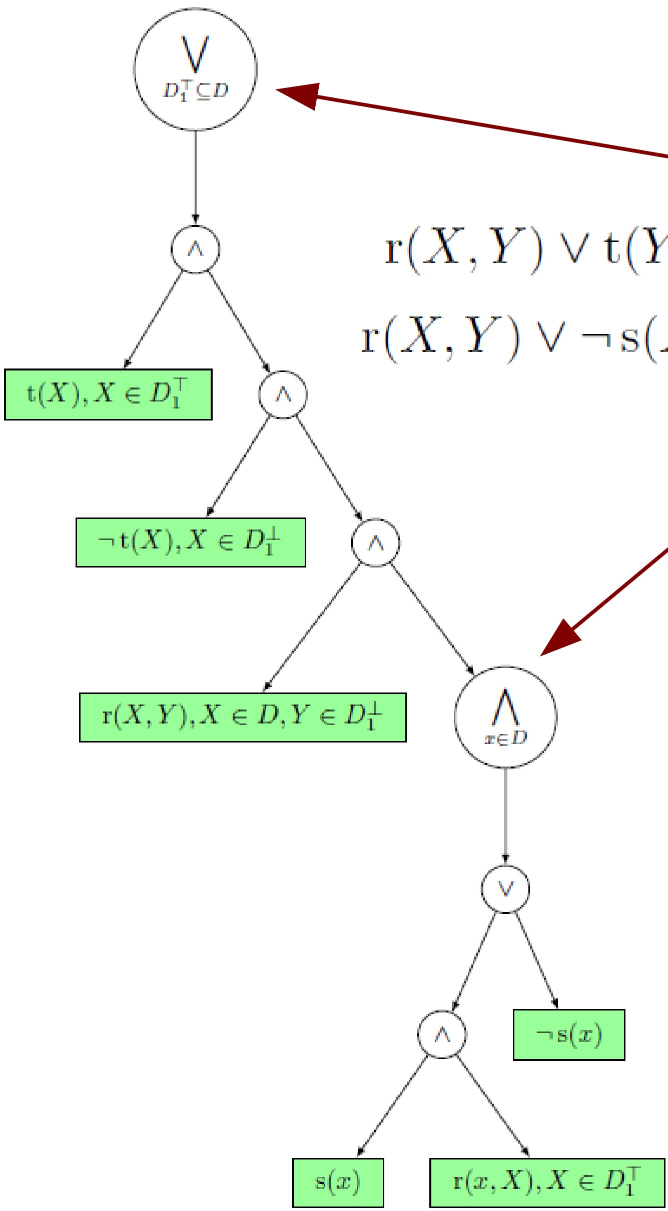
Logic-based Probabilistic Inference



- DPLL Search Weighted Model Counting [Sang 2005]
- **Knowledge Compilation** [Darwiche]

Additional Operator Nodes

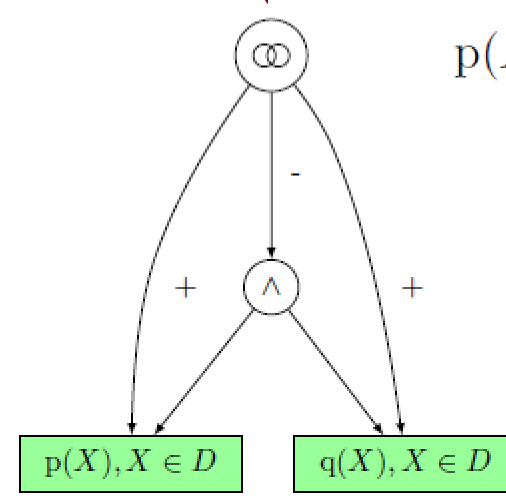
FO d-DNNF Circuit



$$r(X,Y) \vee t(Y)$$

$$r(X,Y) \vee \neg s(X)$$

- Deterministic set-disjunction
- Decomposable set-conjunction
- Inclusion-Exclusion
(non-deterministic disjunction)



$$p(X) \vee q(Y)$$

Auxiliary Operations

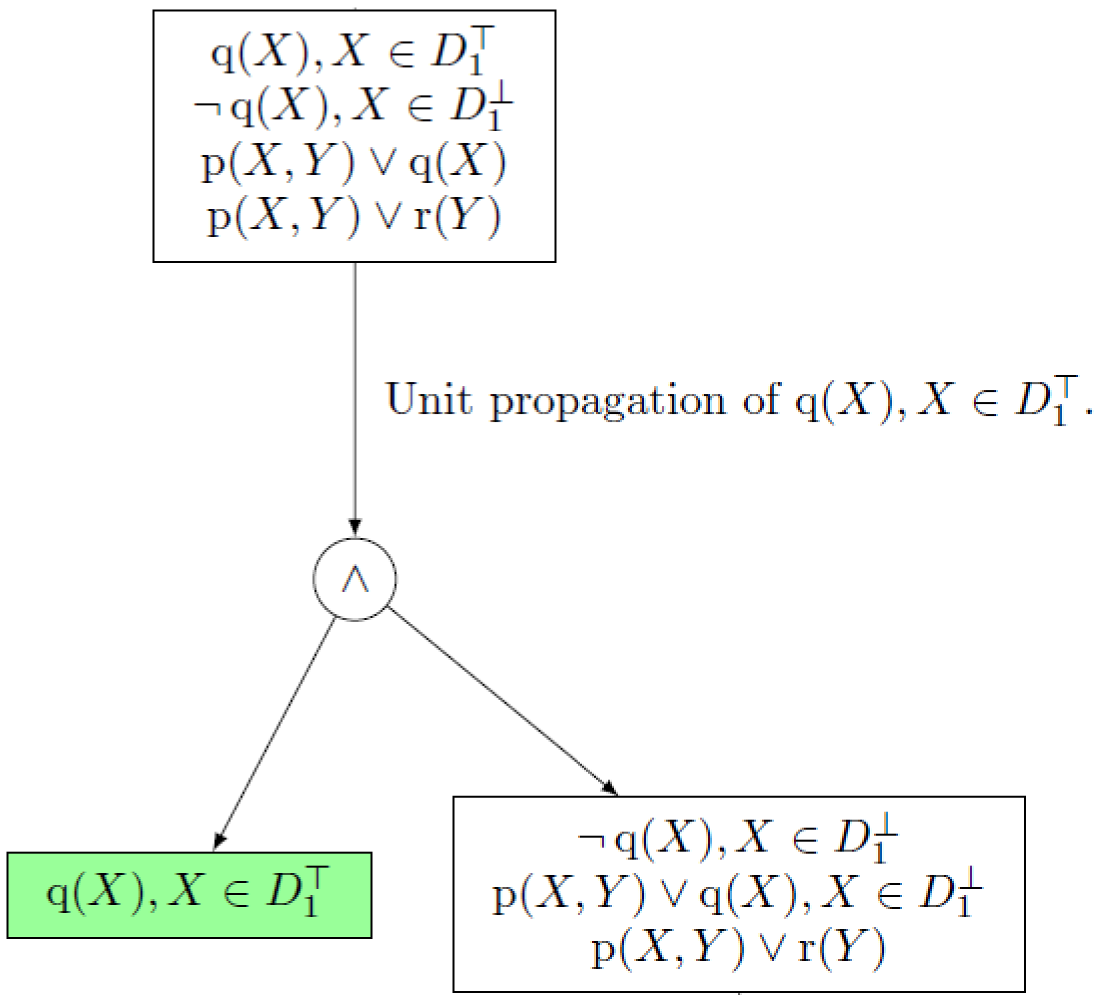
- Splitting w.r.t. an atom [Poole 2003]:
  - Similar to splitting in (C-)FOVE, but with domain constraints

Before Splitting	Atom	After Splitting
$\mathrm{p}(X,Y) \vee \mathrm{q}(X) \vee \mathrm{r}(Y)$	$\mathrm{p}(X,X)$	$\mathrm{p}(X,X) \vee \mathrm{q}(X) \vee \mathrm{r}(X)$ $\mathrm{p}(X,Y) \vee \mathrm{q}(X) \vee \mathrm{r}(Y), X \neq Y$
$\mathrm{p}(X) \vee \mathrm{q}(X),\ X \in D$	$\mathrm{p}(X),\ X \in D_1$	$\mathrm{p}(X) \vee \mathrm{q}(X),\ X \in D_1$ $\mathrm{p}(X) \vee \mathrm{q}(X),\ X \in D_2$

- Shattering [de Salvo Braz 2005]

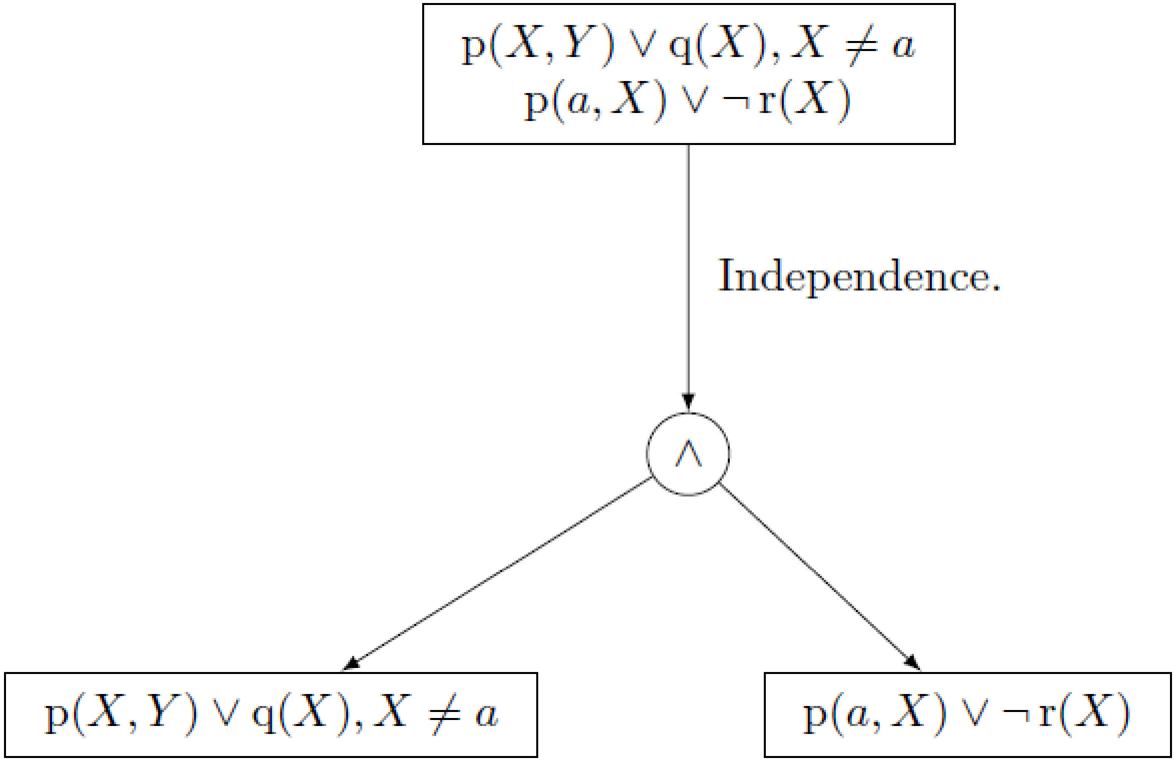
  Splitting w.r.t. any atom in theory, until convergence

Unit Propagation



Independence

- Set of clauses independent from rest
- Independence when no unifying atoms



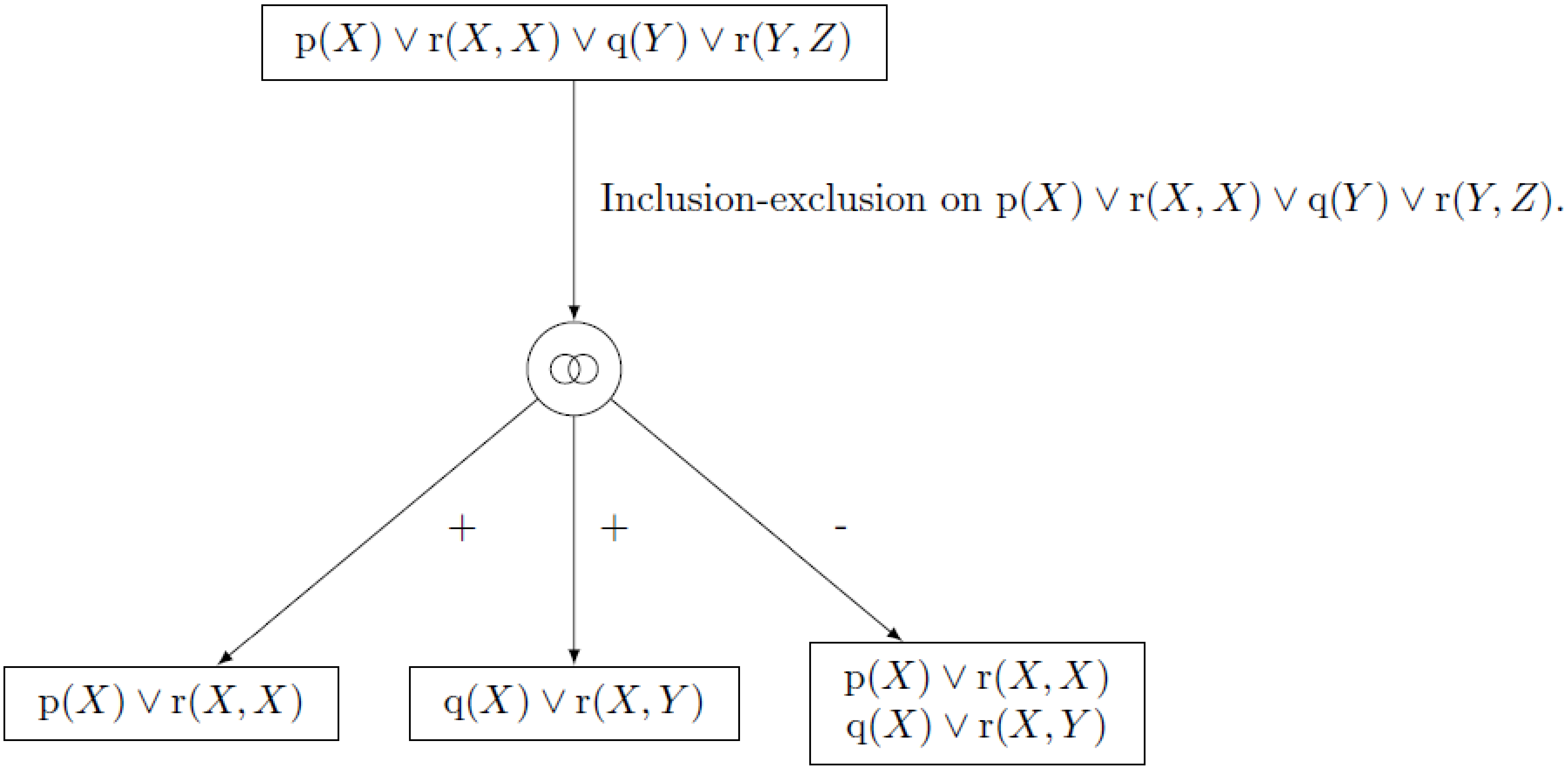
Inclusion-Exclusion

- Clause has set of literals that share no logical variables with rest
- Non-deterministic disjunction & intersection

$$\mathrm{p}(X) \vee \mathrm{r}(X, X) \vee \mathrm{q}(Y) \vee \mathrm{r}(Y, Z)$$

Inclusion-Exclusion

- Clause has set of literals that share no logical variables with rest
- Non-deterministic disjunction & intersection



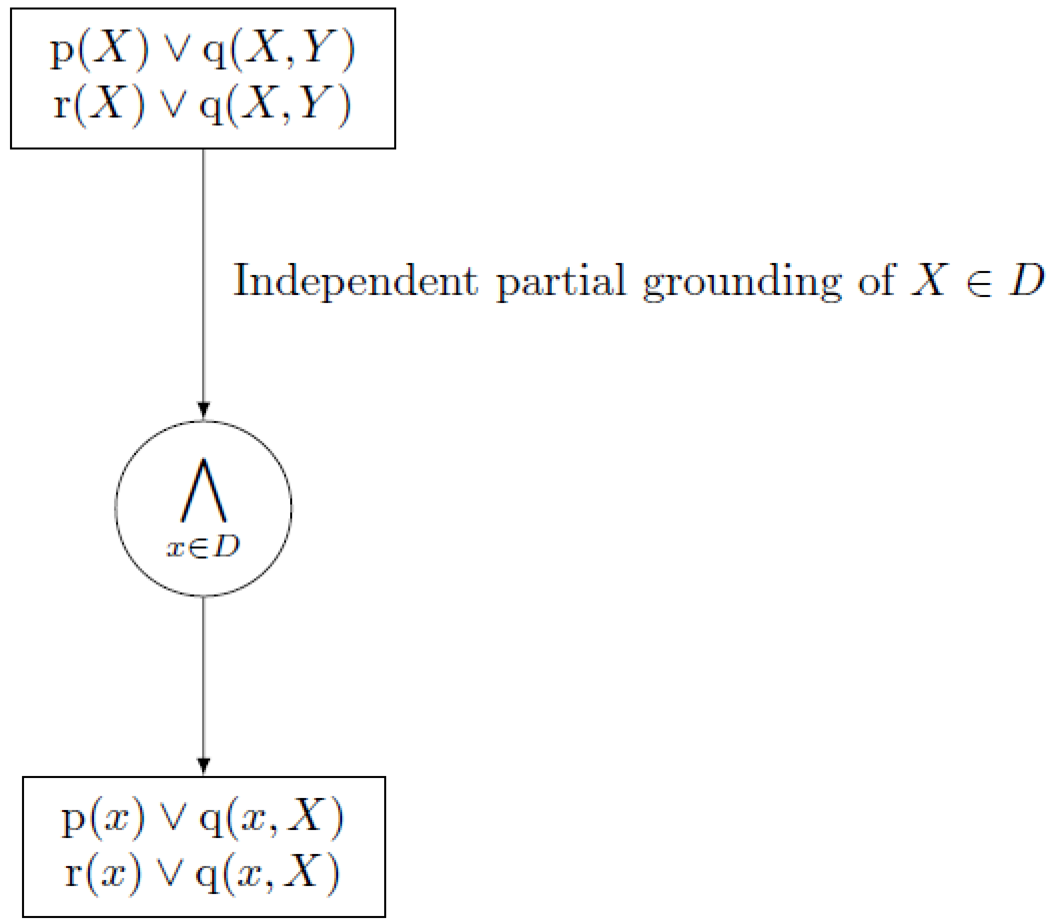
Independent Partial Grounding

- Single logical variable in every atom (position!)
- Different partial groundings are independent

$$\mathrm{p}(X) \vee \mathrm{q}(X, Y)$$
$$\mathrm{r}(X) \vee \mathrm{q}(X, Y)$$

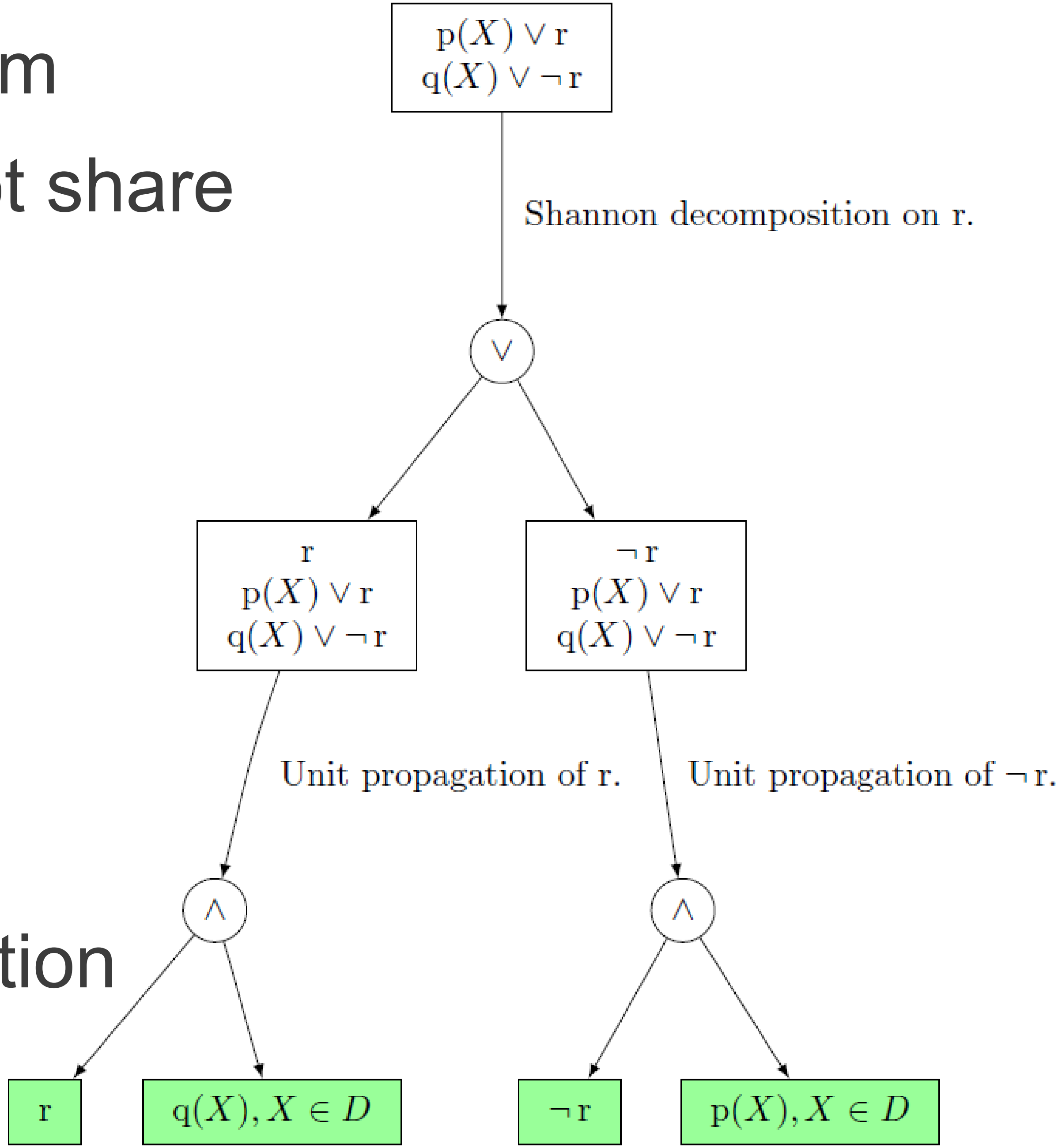
Independent Partial Grounding

- Single logical variable in every atom (position!)
- Different partial groundings are independent



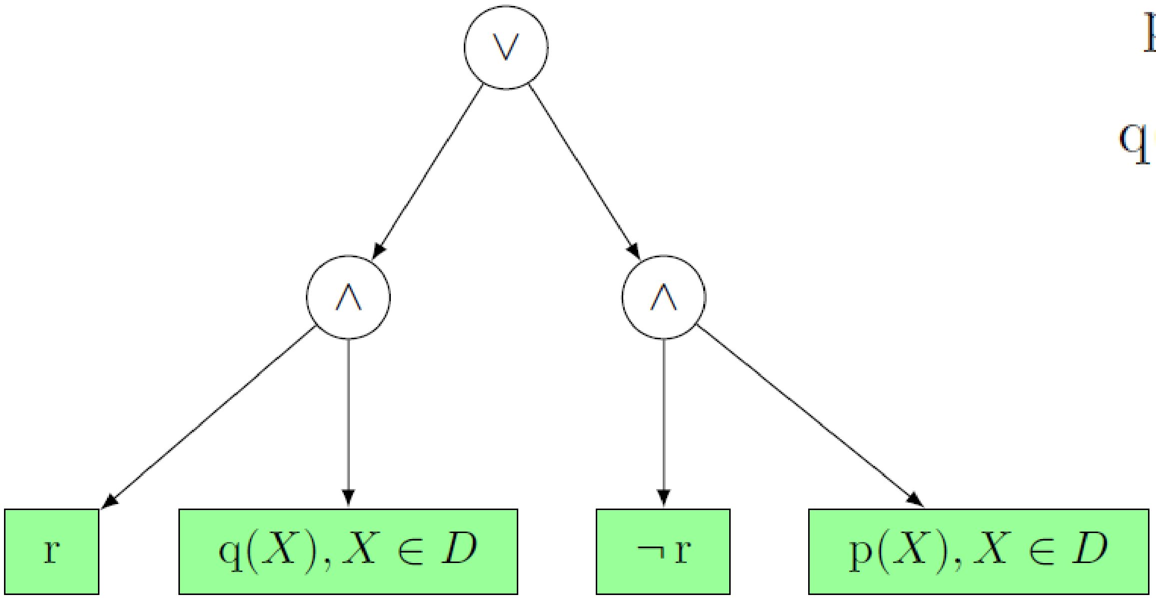
Special Inclusion Exclusion Case: Shannon Decomposition

- CNF has ground atom
- Ground atoms do not share logical variables
- We can always add clause $r \vee \neg r$
- Intersection is unsatisfiable
- Inclusion-Exclusion becomes deterministic disjunction



First-Order Smoothing

FO d-DNNF

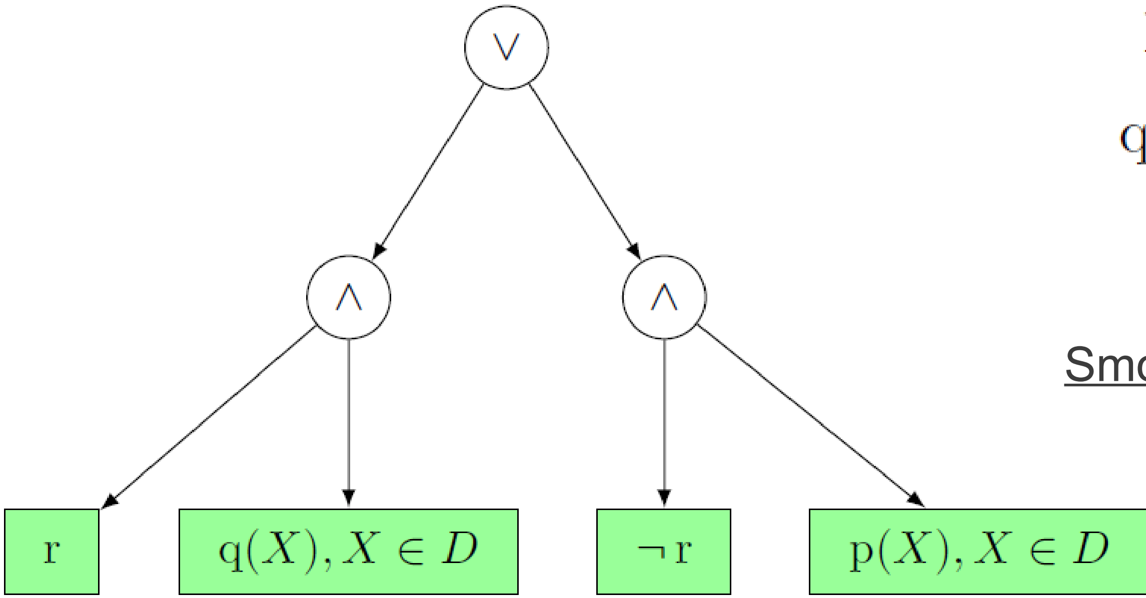


$$\mathrm{p}(X) \vee \mathrm{r}$$

$$\mathrm{q}(X) \vee \neg\, \mathrm{r}$$

First-Order Smoothing

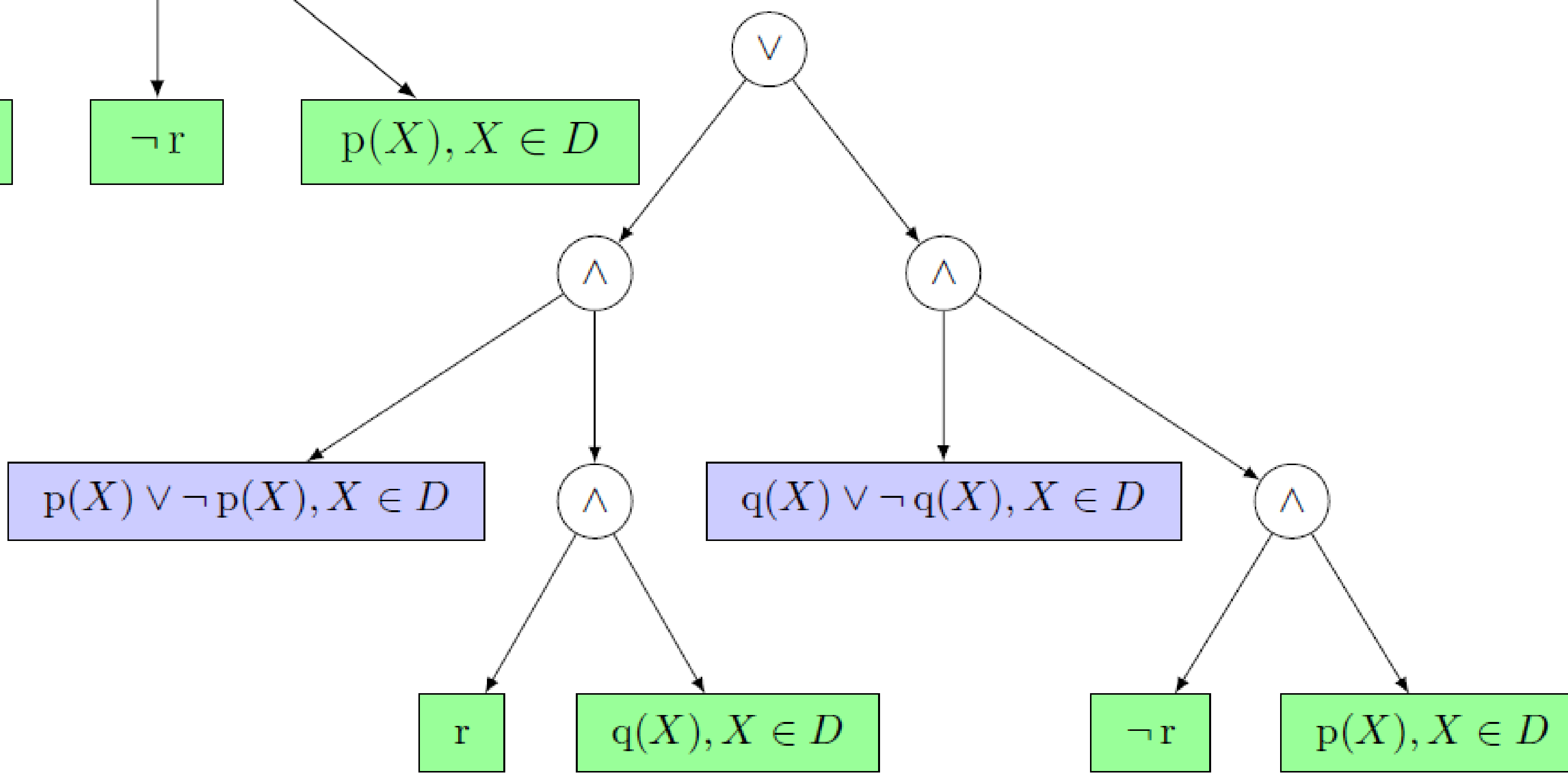
FO d-DNNF



$$\mathrm{p}(X) \vee \mathrm{r}$$

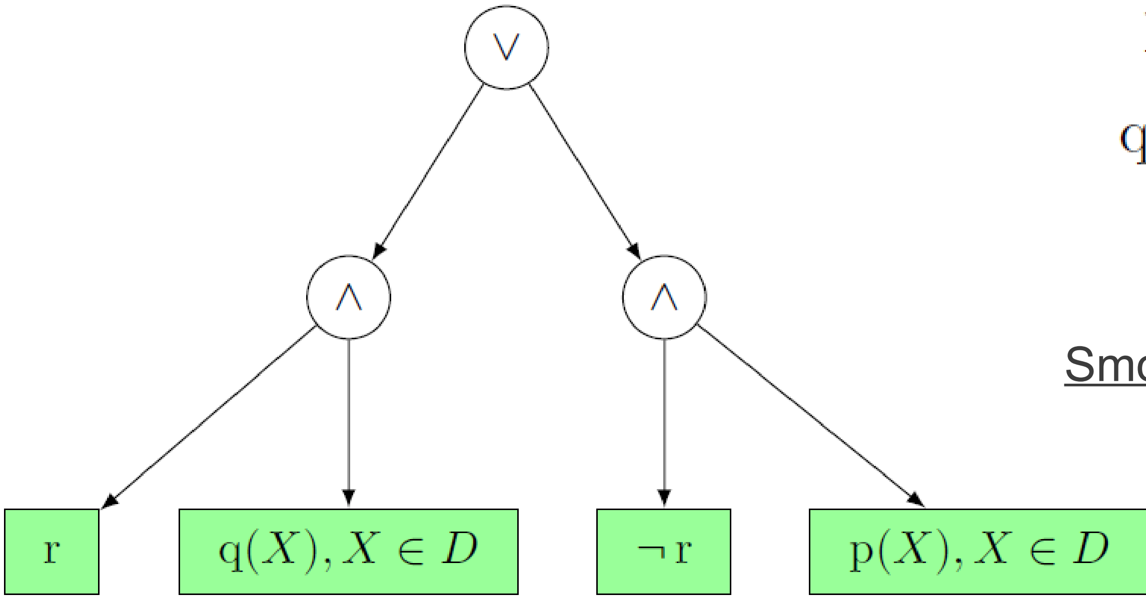
$$\mathrm{q}(X) \vee \neg\, \mathrm{r}$$

Smooth FO d-DNNF



First-Order Smoothing

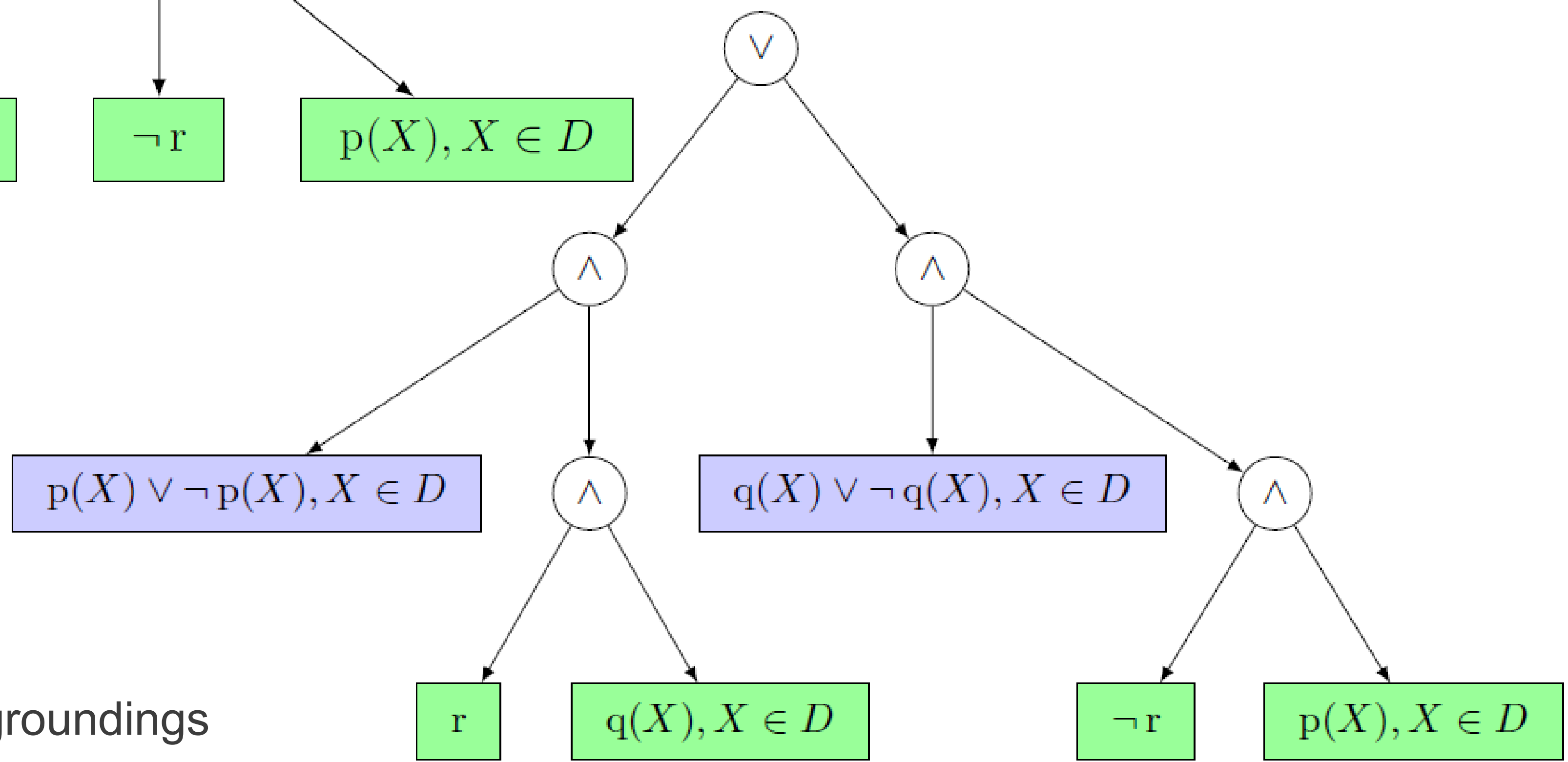
FO d-DNNF



$$p(X) \vee r$$

$$q(X) \vee \neg r$$

Smooth FO d-DNNF



Complicated rules for
- atom counting
- independent partial groundings

Circuit Evaluation

- Propagate weighted model count to root node
- Propagate
  - + for disjunction
  - * for conjunction
  - ...
  - $\sum_{s} \binom{|D|}{s} \mathrm{wmc}(c \wedge |D_1^\top| = s)$

    for atom counting
- Atom counting **linear** in domain size, others **independent** of