Generating Random Logic Programs Using Constraint Programming

Paulius Dilkas[1]   Vaishak Belle[1,2]

[1]University of Edinburgh, Edinburgh, UK

[2]Alan Turing Institute, London, UK

FMAI 2021

Probabilistic Logic Programs (PROBLOG)

"Smokers" (Domingos et al. 2008; Fierens et al. 2015)

```
0.2 :: stress(P) :- person(P).
0.3 :: influences(P_1, P_2) :- friend(P_1, P_2).
0.1 :: cancer_spont(P) :- person(P).
0.3 :: cancer_smoke(P) :- person(P).
      smokes(X) :- stress(X).
      smokes(X) :- smokes(Y), influences(Y, X).
      cancer(P) :- cancer_spont(P).
      cancer(P) :- smokes(P), cancer_smoke(P).
      person(michelle).
      person(timothy).
      friend(timothy, michelle).
```

Applications
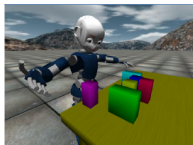
Moldovan et al. 2012

Q1: In a group of 10 people, 60 percent have brown eyes. Two people are to be selected at random from the group. What is the probability that neither person selected will have brown eyes?

Q2: Mike has a bag of marbles with 4 white, 8 blue, and 6 red marbles. He pulls out one marble from the bag and it is red. What is the probability that the second marble he pulls out of the bag is white?

Dries et al. 2017

```
is_malignant(Case):-
        biopsyProcedure(Case,usCore),
        changes_Sizeinc(Case,missing),
        feature_shape(Case).
is_malignant(Case):-
        assoFinding(Case,asymmetry),
        breastDensity(Case,scatteredFDensities),
        vacuumAssisted(Case,yes).
is_malignant(Case):-
        needleGauge(Case,9),
        offset(Case,14),
        vacuumAssisted(Case,yes).
```

Côrte-Real, Dutra, and Rocha 2017
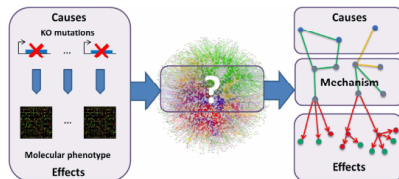
De Maeyer et al. 2013

How Many Programs Are Used to Test Algorithms?

How Many Programs Are Used to Test Algorithms?

Anytime Inference in Probabilistic Logic Programs with T_P-Compilation

Jonas Vlasselaer, Guy Van den Broeck, Angelika Kimmig, Wannes Meert, Luc De Raedt
Department of Computer Science
KU Leuven, Belgium
firstname.lastname@cs.kuleuven.be

4

How Many Programs Are Used to Test Algorithms?

Anytime Inference in Probabilistic Logic Programs with T_P-Compilation

Jonas Vlasselaer, Guy Van den Broeck, Angelika Kimmig, Wannes Meert, Luc De Raedt

Department of Computer Science
KU Leuven, Belgium
firstname.lastname@cs.kuleuven.be

Inference and learning in probabilistic logic programs using weighted Boolean formulas

DAAN FIERENS, GUY VAN DEN BROECK, JORIS RENKENS,
DIMITAR SHTERIONOV, BERND GUTMANN, INGO THON,
GERDA JANSSENS and LUC DE RAEDT

Department of Computer Science, KU Leuven, Celestijnenlaan 200A, 3001 Heverlee, Belgium
(e-mail: FirstName.LastName@cs.kuleuven.be)

Introduction
○○○●○

The Constraint Model
○○○○○

Inference
○○○

Summary
○

How Many Programs Are Used to Test Algorithms?

Anytime Inference in Probabilistic Logic Programs with T_P-Compilation

Jonas Vlasselaer, Guy Van den Broeck, Angelika Kimmig, Wannes Meert, Luc De Raedt
Department of Computer Science
KU Leuven, Belgium
firstname.lastname@cs.kuleuven.be

4

Inference and learning in probabilistic logic programs using weighted Boolean formulas

DAAN FIERENS, GUY VAN DEN BROECK, JORIS RENKENS,
DIMITAR SHTERIONOV, BERND GUTMANN, INGO THON,
GERDA JANSSENS and LUC DE RAEDT
Department of Computer Science, KU Leuven, Celestijnenlaan 200A, 3001 Heverlee, Belgium
(e-mail: FirstName.LastName@cs.kuleuven.be)

3

k-Optimal: a novel approximate inference algorithm for ProbLog

Joris Renkens · Guy Van den Broeck · Siegfried Nijssen

1

How Many Programs Are Used to Test Algorithms?

Anytime Inference in Probabilistic Logic Programs with T_P-Compilation

Jonas Vlasselaer, Guy Van den Broeck, Angelika Kimmig, Wannes Meert, Luc De Raedt
Department of Computer Science
KU Leuven, Belgium
firstname.lastname@cs.kuleuven.be

4

Inference and learning in probabilistic logic programs using weighted Boolean formulas

DAAN FIERENS, GUY VAN DEN BROECK, JORIS RENKENS,
DIMITAR SHTERIONOV, BERND GUTMANN, INGO THON,
GERDA JANSSENS and LUC DE RAEDT
Department of Computer Science, KU Leuven, Celestijnenlaan 200A, 3001 Heverlee, Belgium
(e-mail: FirstName.LastName@cs.kuleuven.be)

3

**k-Optimal: a novel approximate inference algorithm
for ProbLog**

Joris Renkens · Guy Van den Broeck · Siegfried Nijssen

1

On the Efficient Execution of ProbLog Programs

Angelika Kimmig[1], Vítor Santos Costa[2], Ricardo Rocha[2], Bart Demoen[1], and
Luc De Raedt[1]

1

Introduction
○○○●○

The Constraint Model
○○○○○

Inference
○○○

Summary
○

How Many Programs Are Used to Test Algorithms?

Anytime Inference in Probabilistic Logic Programs with T_P-Compilation

Jonas Vlasselaer, Guy Van den Broeck, Angelika Kimmig, Wannes Meert, Luc De Raedt
Department of Computer Science
KU Leuven, Belgium
firstname.lastname@cs.kuleuven.be

4

On the Efficient Execution of ProbLog Programs

Angelika Kimmig[1], Vítor Santos Costa[2], Ricardo Rocha[2], Bart Demoen[1], and
Luc De Raedt[1]

1

Inference and learning in probabilistic logic programs using weighted Boolean formulas

DAAN FIERENS, GUY VAN DEN BROECK, JORIS RENKENS,
DIMITAR SHTERIONOV, BERND GUTMANN, INGO THON,
GERDA JANSSENS and LUC DE RAEDT
Department of Computer Science, KU Leuven, Celestijnenlaan 200A, 3001 Heverlee, Belgium
(e-mail: FirstName.LastName@cs.kuleuven.be)

3

On the Implementation of the Probabilistic Logic Programming Language ProbLog

Angelika Kimmig, Bart Demoen and Luc De Raedt
Departement Computerwetenschappen, K.U. Leuven
Celestijnenlaan 200A - bus 2402, B-3001 Heverlee, Belgium
(e-mail: {Angelika.Kimmig,Bart.Demoen,Luc.DeRaedt}@cs.kuleuven.be)

Vítor Santos Costa and Ricardo Rocha
CRACS & INESC-Porto LA, Faculty of Sciences, University of Porto
R. do Campo Alegre 1021/1055, 4169-007 Porto, Portugal
(e-mail: {vsc,ricroc}@dcc.fc.up.pt)

2

k-Optimal: a novel approximate inference algorithm for ProbLog

1

Joris Renkens · Guy Van den Broeck · Siegfried Nijssen

Introduction
○○●○

The Constraint Model
○○○○○

Inference
○○○

Summary
○

How Many Programs Are Used to Test Algorithms?

Anytime Inference in Probabilistic Logic Programs with T_p-Compilation

Jonas Vlasselaer, Guy Van den Broeck, Angelika Kimmig, Wannes Meert, Luc De Raedt

Department of Computer Science
KU Leuven, Belgium
firstname.lastname@cs.kuleuven.be

4

On the Efficient Execution of ProbLog Programs

Angelika Kimmig[1], Vítor Santos Costa[2], Ricardo Rocha[2], Bart Demoen[1], and Luc De Raedt[1]

1

Inference and learning in probabilistic logic programs using weighted Boolean formulas

DAAN FIERENS, GUY VAN DEN BROECK, JORIS RENKENS,
DIMITAR SHTERIONOV, BERND GUTMANN, INGO THON,
GERDA JANSSENS and LUC DE RAEDT

Department of Computer Science, KU Leuven, Celestijnenlaan 200A, 3001 Heverlee, Belgium
(e-mail: FirstName.LastName@cs.kuleuven.be)

3

On the Implementation of the Probabilistic Logic Programming Language ProbLog

Angelika Kimmig, Bart Demoen and Luc De Raedt

Departement Computerwetenschappen, K.U. Leuven
Celestijnenlaan 200A - bus 2402, B-3001 Heverlee, Belgium
(e-mail: {Angelika.Kimmig,Bart.Demoen,Luc.DeRaedt}@cs.kuleuven.be)

Vítor Santos Costa and Ricardo Rocha

CRACS & INESC-Porto LA, Faculty of Sciences, University of Porto
R. do Campo Alegre 1021/1055, 4169-007 Porto, Portugal
(e-mail: {vsc,ricroc}@dcc.fc.up.pt)

k-Optimal: a novel approximate inference algorithm for ProbLog

Joris Renkens · Guy Van den Broeck · Siegfried Nijssen

1

ProbLog Technology for Inference in a Probabilistic First Order Logic

Maurice Bruynooghe and Theofrastos Mantadelis and Angelika Kimmig and Bernd Gutmann and Joost Vennekens and Gerda Janssens and Luc De Raedt[1]

2

Outline

Introduction

The Constraint Model

Inference

Summary

What Characterises a (Probabilistic) Logic Program?

$0.2::\text{stress}(P):-\text{person}(P).$

$0.3::\text{influences}(P_1, P_2):-\text{friend}(P_1, P_2).$

$0.1::\text{cancer\_spont}(P):-\text{person}(P).$

$0.3::\text{cancer\_smoke}(P):-\text{person}(P).$

$\quad\text{smokes}(X):-\text{stress}(X).$

$\quad\text{smokes}(X):-\text{smokes}(Y), \text{influences}(Y, X).$

$\quad\text{cancer}(P):-\text{cancer\_spont}(P).$

$\quad\text{cancer}(P):-\text{smokes}(P), \text{cancer\_smoke}(P).$

$\quad\text{person}(\textit{michelle}).$

$\quad\text{person}(\textit{timothy}).$

$\quad\text{friend}(\textit{timothy}, \textit{michelle}).$

Introduction
0000

The Constraint Model
●0000

Inference
000

Summary
0

What Characterises a (Probabilistic) Logic Program?

$0.2::\text{stress}(P):-\text{person}(P).$

$0.3::\text{influences}(P_1,P_2):-\text{friend}(P_1,P_2).$

$0.1::\text{cancer\_spont}(P):-\text{person}(P).$

$0.3::\text{cancer\_smoke}(P):-\text{person}(P).$

$\quad\text{smokes}(X):-\text{stress}(X).$

$\quad\text{smokes}(X):-\text{smokes}(Y),\text{influences}(Y,X).$

$\quad\text{cancer}(P):-\text{cancer\_spont}(P).$

$\quad\text{cancer}(P):-\text{smokes}(P),\text{cancer\_smoke}(P).$

$\quad\text{person}(\textit{michelle}).$

$\quad\text{person}(\textit{timothy}).$

$\quad\text{friend}(\textit{timothy},\textit{michelle}).$

- predicates, arities

Introduction
0000

The Constraint Model
●0000

Inference
000

Summary
0

What Characterises a (Probabilistic) Logic Program?

$0.2::\text{stress}(P):-\text{person}(P).$

$0.3::\text{influences}(P_1,P_2):-\text{friend}(P_1,P_2).$

$0.1::\text{cancer\_spont}(P):-\text{person}(P).$

$0.3::\text{cancer\_smoke}(P):-\text{person}(P).$

- predicates, arities
- variables

$\quad\text{smokes}(X):-\text{stress}(X).$

$\quad\text{smokes}(X):-\text{smokes}(Y),\text{influences}(Y,X).$

$\quad\text{cancer}(P):-\text{cancer\_spont}(P).$

$\quad\text{cancer}(P):-\text{smokes}(P),\text{cancer\_smoke}(P).$

$\quad\text{person}(michelle).$

$\quad\text{person}(timothy).$

$\quad\text{friend}(timothy,michelle).$

Introduction
0000

The Constraint Model
●0000

Inference
000

Summary
0

What Characterises a (Probabilistic) Logic Program?

$0.2::\text{stress}(P):\text{-}\text{person}(P).$

$0.3::\text{influences}(P_1,P_2):\text{-}\text{friend}(P_1,P_2).$

$0.1::\text{cancer\_spont}(P):\text{-}\text{person}(P).$

$0.3::\text{cancer\_smoke}(P):\text{-}\text{person}(P).$

$\quad\text{smokes}(X):\text{-}\text{stress}(X).$

$\quad\text{smokes}(X):\text{-}\text{smokes}(Y),\text{influences}(Y,X).$

$\quad\text{cancer}(P):\text{-}\text{cancer\_spont}(P).$

$\quad\text{cancer}(P):\text{-}\text{smokes}(P),\text{cancer\_smoke}(P).$

$\quad\text{person}(\textit{michelle}).$

$\quad\text{person}(\textit{timothy}).$

$\quad\text{friend}(\textit{timothy},\textit{michelle}).$

- predicates, arities
- variables
- constants

Introduction
0000

The Constraint Model
●0000

Inference
000

Summary
0

What Characterises a (Probabilistic) Logic Program?

$0.2::$stress(P):-person(P).

$0.3::$influences(P_1, P_2):-friend(P_1, P_2).

$0.1::$cancer_spont(P):-person(P).

$0.3::$cancer_smoke(P):-person(P).

    smokes(X):-stress(X).

    smokes(X):-smokes(Y), influences(Y, X).

    cancer(P):-cancer_spont(P).

    cancer(P):-smokes(P), cancer_smoke(P).

    person(*michelle*).

    person(*timothy*).
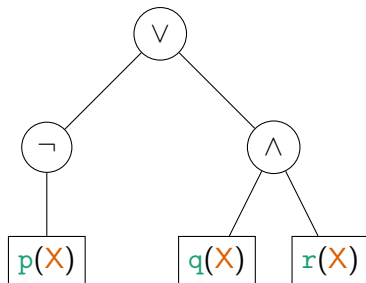
    friend(*timothy*, *michelle*).

- predicates, arities
- variables
- constants
- probabilities

Introduction
0000

The Constraint Model
●0000

Inference
○○○

Summary
○

What Characterises a (Probabilistic) Logic Program?

$0.2::\texttt{stress}(\mathsf{P}):-\texttt{person}(\mathsf{P}).$

$0.3::\texttt{influences}(\mathsf{P_1},\mathsf{P_2}):-\texttt{friend}(\mathsf{P_1},\mathsf{P_2}).$

$0.1::\texttt{cancer\_spont}(\mathsf{P}):-\texttt{person}(\mathsf{P}).$

$0.3::\texttt{cancer\_smoke}(\mathsf{P}):-\texttt{person}(\mathsf{P}).$

$\texttt{smokes}(\mathsf{X}):-\texttt{stress}(\mathsf{X}).$

$\texttt{smokes}(\mathsf{X}):-\texttt{smokes}(\mathsf{Y}),\texttt{influences}(\mathsf{Y},\mathsf{X}).$

$\texttt{cancer}(\mathsf{P}):-\texttt{cancer\_spont}(\mathsf{P}).$

$\texttt{cancer}(\mathsf{P}):-\texttt{smokes}(\mathsf{P}),\texttt{cancer\_smoke}(\mathsf{P}).$

$\texttt{person}(\textit{michelle}).$

$\texttt{person}(\textit{timothy}).$

$\texttt{friend}(\textit{timothy},\textit{michelle}).$

- predicates, arities
- variables
- constants
-
  probabilities
- length

Introduction
0000

The Constraint Model
●0000

Inference
000

Summary
0

What Characterises a (Probabilistic) Logic Program?

```
0.2::stress(P):-person(P).
0.3::influences(P₁,P₂):-friend(P₁,P₂).
0.1::cancer_spont(P):-person(P).
0.3::cancer_smoke(P):-person(P).
    smokes(X):-stress(X).
    smokes(X):-smokes(Y),influences(Y,X).
    cancer(P):-cancer_spont(P).
    cancer(P):-smokes(P),cancer_smoke(P).
    person(michelle).
    person(timothy).
    friend(timothy,michelle).
```

- predicates, arities
- variables
- constants
- probabilities
- length
- complexity

Introduction
0000

The Constraint Model
0●000

Inference
000

Summary
0

Formulas As Trees

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

Introduction
0000

The Constraint Model
0●000

Inference
000

Summary
0

Formulas As Trees

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

Introduction
0000

The Constraint Model
0●000

Inference
000

Summary
0

Formulas As Trees

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

Introduction
0000

The Constraint Model
00●00

Inference
000

Summary
0
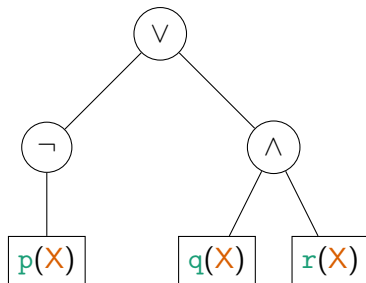
Predicate Dependency Graph

Introduction
oooo

The Constraint Model
ooo●o

Inference
ooo

Summary
o
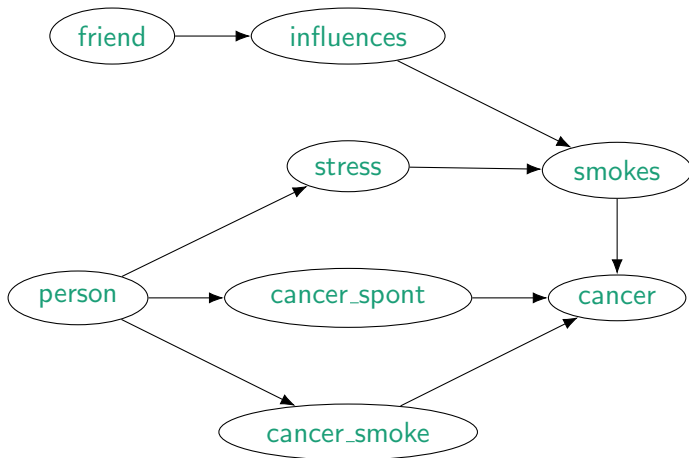
Stratification and Negative Cycles

$$0.1 :: \texttt{friend}(X, Y) :- \backslash + \texttt{smokes}(Y).$$
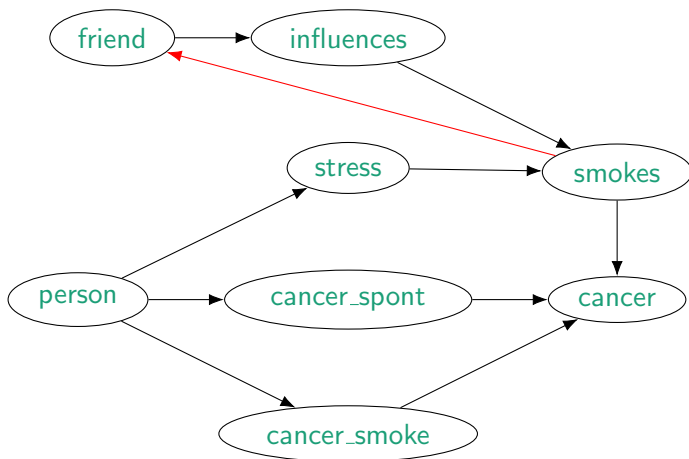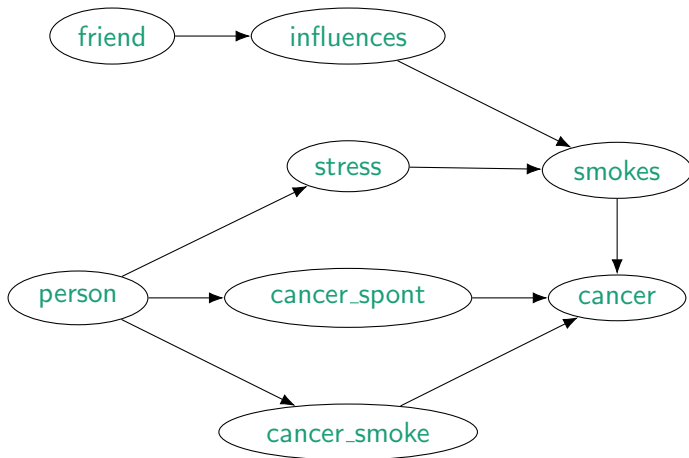
Introduction
○○○○

The Constraint Model
○○○●○

Inference
○○○

Summary
○

Stratification and Negative Cycles

Introduction
0000

The Constraint Model
0000●

Inference
000

Summary
0

Predicate Independence: $friend \perp\!\!\!\perp stress$

Introduction
oooo

The Constraint Model
ooooo

Inference
ooo

Summary
o

Predicate Independence: friend ⫫ stress

Introduction
0000

The Constraint Model
00000

Inference
000

Summary
0

Predicate Independence: $\text{friend} \perp\!\!\!\perp \text{stress}$

Introduction
oooo

The Constraint Model
ooooo●

Inference
ooo

Summary
o

Predicate Independence: $friend \perp\!\!\!\perp stress$

Introduction
0000

The Constraint Model
00000

Inference
000

Summary
0

Predicate Independence: friend ⫫ stress

Introduction
○○○○

The Constraint Model
○○○○●

Inference
○○○

Summary
○

Predicate Independence: friend ⊥⊥ stress

Introduction
0000

The Constraint Model
00000

Inference
●oo

Summary
o
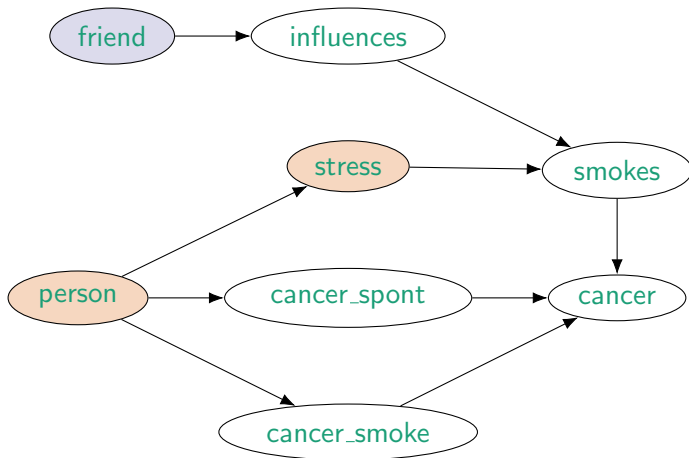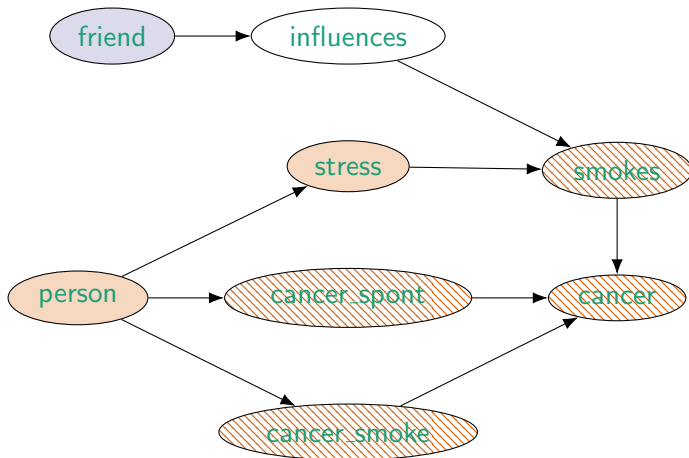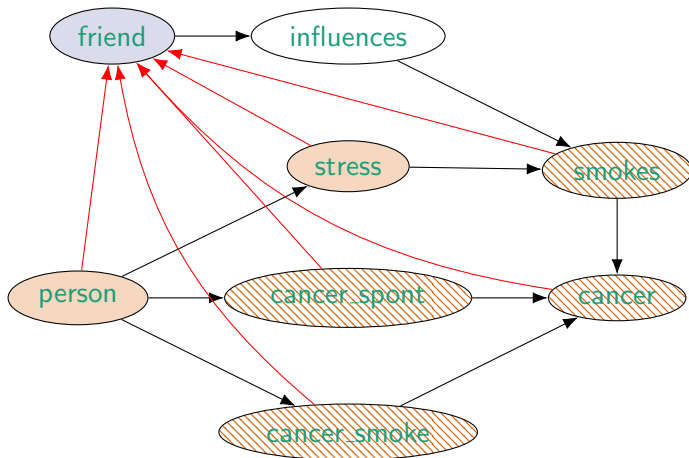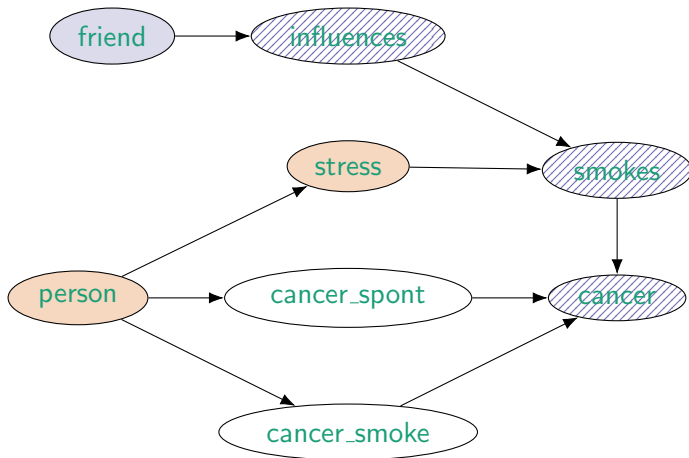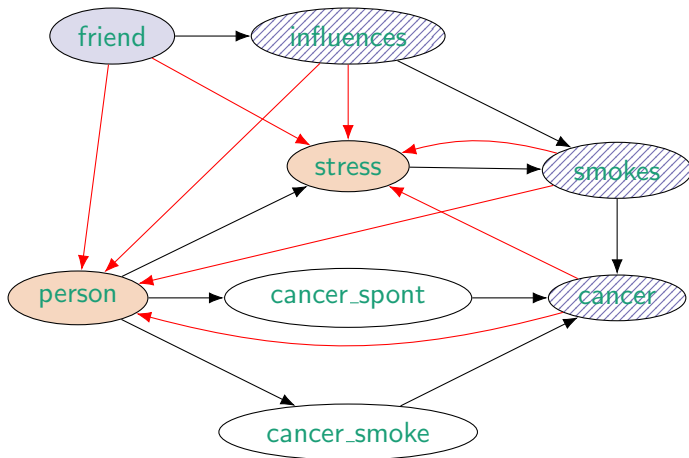
Inference and Knowledge Compilation

NNF negation normal form

d-DNNF deterministic decomposable negation normal form

BDD binary decision diagrams

SDD sentential decision diagrams

Introduction
oooo

The Constraint Model
ooooo

Inference
o●o

Summary
o

Example Diagrams for $C \wedge (A \vee \neg B)$
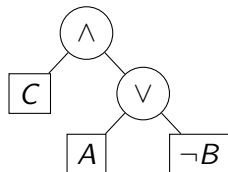
Figure: NNF
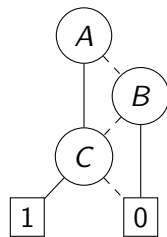
Figure: BDD

Figure: d-DNNF

Figure: SDD

Introduction
○○○○

The Constraint Model
○○○○○
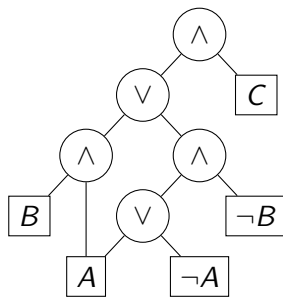
Inference
○○●

Summary
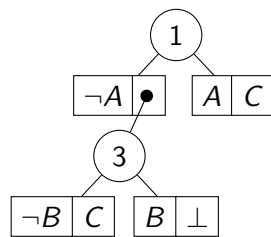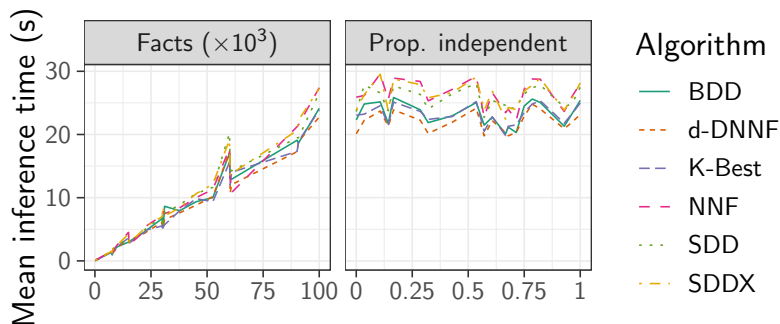○

Properties of Programs vs. Inference Algorithms

Facts: `friend(`*timothy*`, `*michelle*`)`.

Rules: `0.2::stress(`P`):-person(`P`)`.

Summary

- The model can generate (approximately) realistic instances of reasonable size.
- Open questions for future work:
    - Can we ensure uniform sampling?
    - Why do all of the algorithms behave so similarly?
    - Why does independence have no effect on inference time?