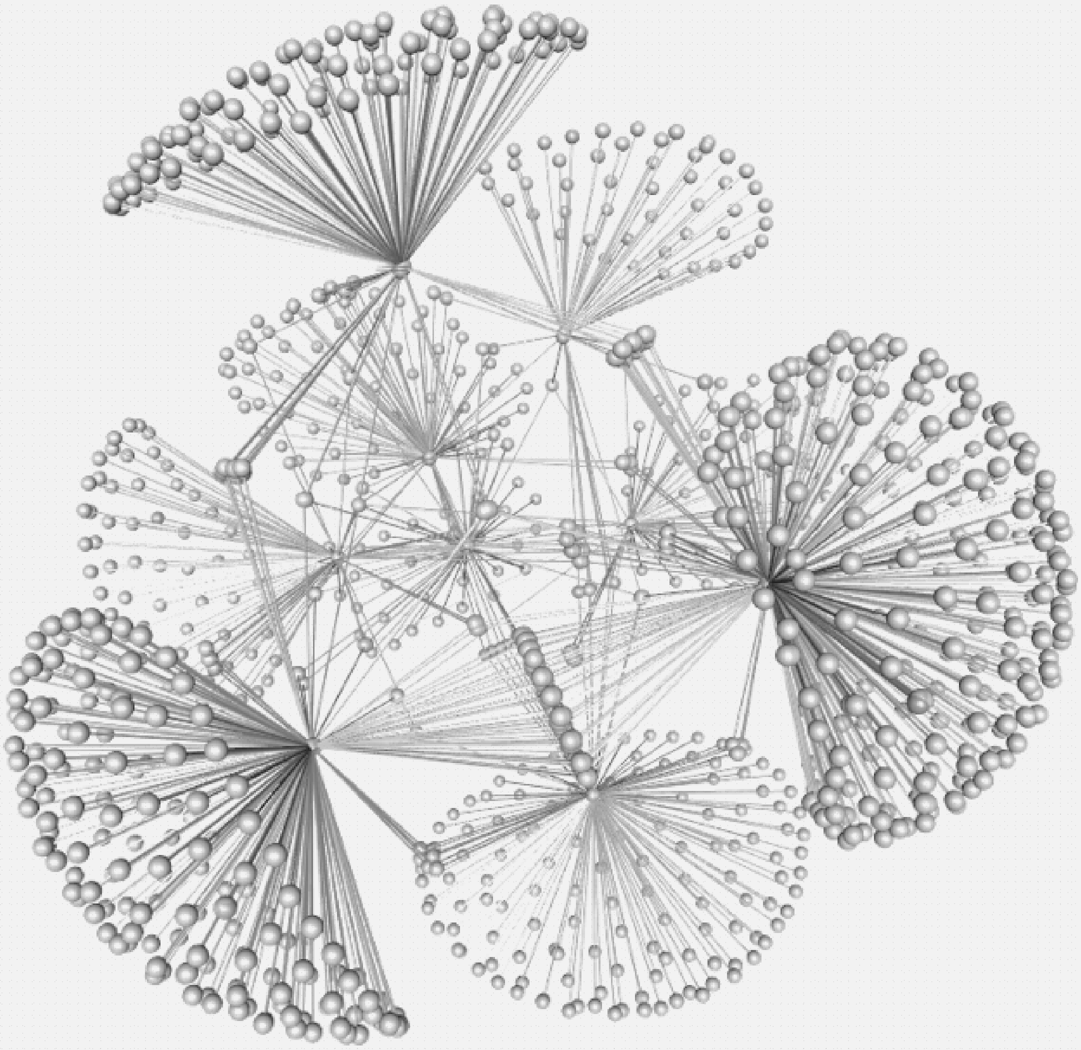
Probabilistic Programming

Michiel Janssen & Bruno Vandekerkhove



ACADEMISCH JAAR 2020

H05N0A: CAPITA SELECTA : ARTIFICIAL INTELLIGENCE

Contents

Below's our solution for the given challenges. The questions in each section of the original assignment are answered in a section having the same title.

Probabilistic Inference Using Weighted Model Counting

SRL to CNF

First the program is grounded. This is a matter of collecting all atoms involved in all proofs of the query.

```
0.2::stress(a).
0.2::stress(b).
0.2::stress(c).

0.1::friends(a,a).
0.1::friends(a,b).
0.1::friends(a,c).

0.1::friends(b,a).
0.1::friends(b,b).
0.1::friends(b,c).

0.1::friends(c,a).
0.1::friends(c,b).
0.1::friends(c,c).

0.3::smokes(a) :- stress(a).
0.3::smokes(b) :- stress(b).
0.3::smokes(c) :- stress(c).
0.4::smokes(a) :- friends(a,a), smokes(a).
0.4::smokes(a) :- friends(a,b), smokes(b).
0.4::smokes(a) :- friends(a,c), smokes(c).
0.4::smokes(b) :- friends(b,a), smokes(a).
0.4::smokes(b) :- friends(b,b), smokes(b).
0.4::smokes(b) :- friends(b,c), smokes(c).

0.4::smokes(c) :- friends(c,a), smokes(a).
0.4::smokes(c) :- friends(c,b), smokes(b).
0.4::smokes(c) :- friends(c,c), smokes(c).
```

Code snippet 1: Relevant ground program.

The proofs of the query make for a trie as shown in figure 1, where colourings indicate the presence of cycles. Any proof involving an atom `friends(X,X)` or `friends(Y,a)` (with $Y \in \{b, c\}$) is non-minimal and doesn't affect the final probability. These atoms are disregarded. For the remaining cycles (involving `friends(b,c)` and `friends(c,b)`) auxiliary variables can be used to obtain a cycle-free program without intensional probabilistic facts :

```
0.2::stress(a).
0.2::stress(b).
0.2::stress(c).

0.1::friends(a,b).
0.1::friends(a,c).
0.1::friends(b,c).
0.1::friends(c,b).

0.3::p(a).
0.3::p(b).
0.3::p(c).
```

```

0.4::p(a,b).
0.4::p(a,c).
0.4::p(b,c).
0.4::p(c,b).

smokes(a) :- stress(a), p(a).
smokes(b) :- stress(b), p(b).
smokes(c) :- stress(c), p(c).

smokes(a) :-
    friends(a,b), smokes(b), p(a,b).
smokes(a) :-
    friends(a,c), smokes(c), p(a,c).
smokes(b) :-
    friends(b,c), stress(c), p(c), p(b,c).
smokes(c) :-
    friends(c,b), stress(b), p(b), p(c,b).

query(smokes(a)).
```

Code snippet 2: Relevant ground program without cycles.

The above logic program is equivalent to the following propositional formula :

$$
\begin{aligned}
&(smokes(a) \leftrightarrow \quad (stress(a) \land p(a)) \\
&\quad \lor (friends(a,b) \land smokes(b) \land p(a,b)) \\
&\quad \lor (friends(a,c) \land smokes(c) \land p(a,c))) \\
&\qquad\qquad \land \\
&(smokes(b) \leftrightarrow \quad (stress(b) \land p(b)) \\
&\lor (friends(b,c) \land stress(c) \land p(c) \land p(b,c))) \\
&\qquad\qquad \land \\
&(smokes(c) \leftrightarrow \quad (stress(c) \land p(c)) \\
&\lor (friends(c,b) \land stress(b) \land p(b) \land p(c,b)))
\end{aligned}
$$

Which corresponds to the following `CNF` :

$$
\begin{aligned}
&(\neg smokes(a) \lor stress(a) \lor friends(a,b) \lor friends(a,c)) \\
&\land (\neg smokes(a) \lor stress(a) \lor friends(a,b) \lor smokes(c)) \\
&\land (\neg smokes(a) \lor stress(a) \lor friends(a,b) \lor p(a,c)) \\
&\land (\neg smokes(a) \lor stress(a) \lor smokes(b) \lor friends(a,c)) \\
&\land (\neg smokes(a) \lor stress(a) \lor smokes(b) \lor smokes(c)) \\
&\land (\neg smokes(a) \lor stress(a) \lor smokes(b) \lor p(a,c)) \\
&\land (\neg smokes(a) \lor stress(a) \lor p(a,b) \lor friends(a,c)) \\
&\land (\neg smokes(a) \lor stress(a) \lor p(a,b) \lor smokes(c)) \\
&\land (\neg smokes(a) \lor stress(a) \lor p(a,b) \lor p(a,c)) \\
&\land (\neg smokes(a) \lor p(a) \lor friends(a,b) \lor friends(a,c)) \\
&\land (\neg smokes(a) \lor p(a) \lor friends(a,b) \lor smokes(c)) \\
&\land (\neg smokes(a) \lor p(a) \lor friends(a,b) \lor p(a,c)) \\
&\land (\neg smokes(a) \lor p(a) \lor smokes(b) \lor friends(a,c)) \\
&\land (\neg smokes(a) \lor p(a) \lor smokes(b) \lor smokes(c)) \\
&\land (\neg smokes(a) \lor p(a) \lor smokes(b) \lor p(a,c)) \\
&\land (\neg smokes(a) \lor p(a) \lor p(a,b) \lor friends(a,c)) \\
&\land (\neg smokes(a) \lor p(a) \lor p(a,b) \lor smokes(c)) \\
&\land (\neg smokes(a) \lor p(a) \lor p(a,b) \lor p(a,c)) \\
&\land (\neg stress(a) \lor \neg p(a) \lor smokes(a)) \\
&\land (\neg friends(a,b) \lor \neg smokes(b) \lor \neg p(a,b) \lor smokes(a)) \\
&\land (\neg friends(a,c) \lor \neg smokes(c) \lor \neg p(a,c) \lor smokes(a)) \\
&\land (\neg smokes(b) \lor stress(b) \lor friends(b,c)) \\
&\land (\neg smokes(b) \lor stress(b) \lor stress(c)) \\
&\land (\neg smokes(b) \lor stress(b) \lor p(c)) \\
&\land (\neg smokes(b) \lor stress(b) \lor p(b,c)) \\
&\land (\neg smokes(b) \lor p(b) \lor friends(b,c)) \\
&\land (\neg smokes(b) \lor p(b) \lor stress(c)) \\
&\land (\neg smokes(b) \lor p(b) \lor p(c)) \\
&\land (\neg smokes(b) \lor p(b) \lor p(b,c)) \\
&\land (\neg stress(b) \lor \neg p(b) \lor smokes(b)) \\
&\land (\neg friends(b,c) \lor \neg stress(c) \lor \neg p(c) \lor \neg p(b,c) \lor smokes(b)) \\
&\land (\neg smokes(c) \lor stress(c) \lor friends(c,b)) \\
&\land (\neg smokes(c) \lor stress(c) \lor stress(b)) \\
&\land (\neg smokes(c) \lor stress(c) \lor p(b)) \\
&\land (\neg smokes(c) \lor stress(c) \lor p(c,b)) \\
&\land (\neg smokes(c) \lor p(c) \lor friends(c,b)) \\
&\land (\neg smokes(c) \lor p(c) \lor stress(b)) \\
&\land (\neg smokes(c) \lor p(c) \lor p(b)) \\
&\land (\neg smokes(c) \lor p(c) \lor p(c,b)) \\
&\land (\neg stress(c) \lor \neg p(c) \lor smokes(c)) \\
&\land (\neg friends(c,b) \lor \neg stress(b) \lor \neg p(b) \lor \neg p(c,b) \lor smokes(c))
\end{aligned}
$$

The probabilistic literals in the `CNF` are assigned weights (derived literals get a weight of 1) :

Literal	Weight
stress(a)	0.2
¬stress(a)	0.8
stress(b)	0.2
¬stress(b)	0.8
stress(c)	0.2
¬stress(c)	0.8
friends(a,b)	0.1
¬friends(a,b)	0.9
friends(a,c)	0.1
¬friends(a,c)	0.9
friends(b,c)	0.1
¬friends(b,c)	0.9
friends(c,b)	0.1
¬friends(c,b)	0.9
p(a)	0.3
¬p(a)	0.7
p(b)	0.3
¬p(b)	0.7
p(c)	0.3
¬p(c)	0.7
p(a,b)	0.4
¬p(a,b)	0.6
p(a,c)	0.4
¬p(a,c)	0.6
p(b,c)	0.4
¬p(b,c)	0.6
p(c,b)	0.4
¬p(c,b)	0.6

SRL to PGM

A Bayesian network is shown in figure 2. The conditional probability tables (CPTs) for the nodes are given below. Note that every table represents multiple identical tables in the network (those of the *stress*(*a*), *stress*(*b*) and *stress*(*c*) nodes for example).

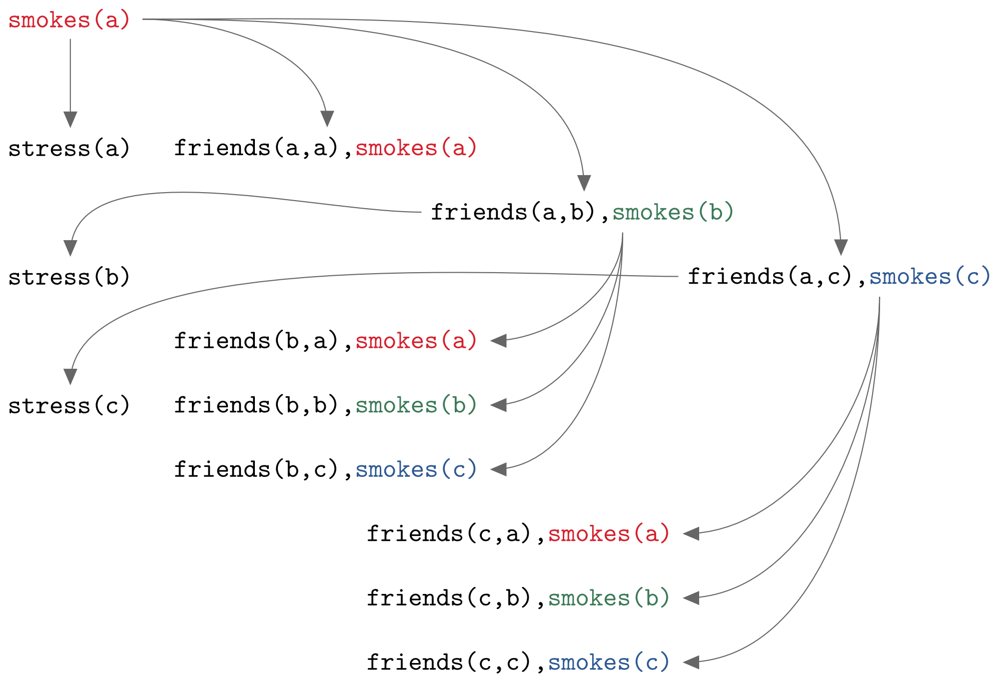


Figure 1: Trie representing proofs of the query. Coloured atoms indicate the presence of cycles.

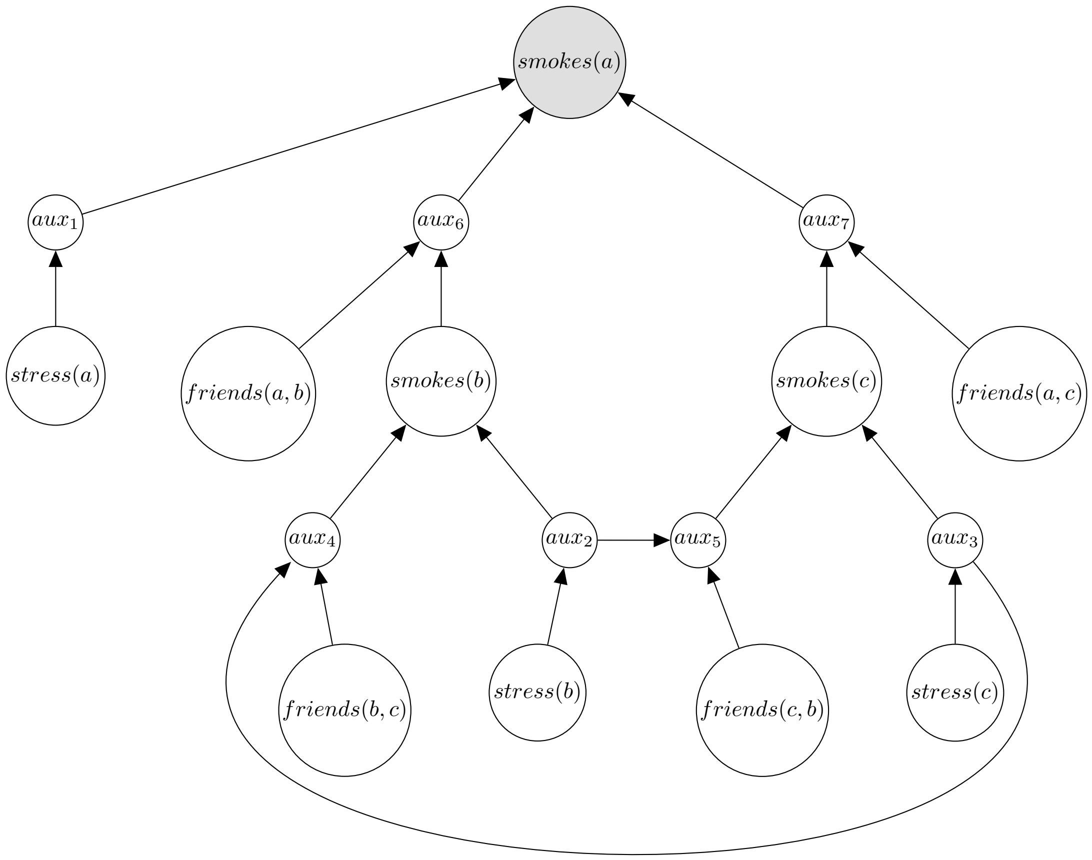


Figure 2: Bayesian network corresponding to the ground acyclic program.

$stress(\{a,b,c\})$	
$\top$	0.2
$\bot$	0.8

$friends(\{a,b,c\},\{b,c\})$	
$\top$	0.1
$\bot$	0.9

$stress(\{a,b,c\})$	$aux_{\{1,2,3\}}$	
	$\top$	$\bot$
$\top$	0.3	0.7
$\bot$	0.0	1.0

$friends(\{b,c\},\{c,b\})$	$aux_{2,3}$	$aux_{4,5}$	
		$\top$	$\bot$
$\top$	$\top$	0.4	0.6
$\top$	$\bot$	0.0	1.0
$\bot$	$\top$	0.0	1.0
$\bot$	$\bot$	0.0	1.0

$friends(\{a\},\{b,c\})$	$smokes(\{b,c\})$	$aux_{6,7}$	
		$\top$	$\bot$
$\top$	$\top$	0.4	0.6
$\top$	$\bot$	0.0	1.0
$\bot$	$\top$	0.0	1.0
$\bot$	$\bot$	0.0	1.0

$aux_{4,5}$	$aux_{2,3}$	$smokes(\{b,c\})$	
		$\top$	$\bot$
$\top$	$\top$	1.0	0.0
$\top$	$\bot$	0.0	1.0
$\bot$	$\top$	0.0	1.0
$\bot$	$\bot$	0.0	1.0

aux_1	$smokes(b)$	$smokes(c)$	$smokes(a)$	
			$\top$	$\bot$
$\top$	$\top$	$\top$	1.0	0.0
$\top$	$\top$	$\bot$	0.0	1.0
$\top$	$\bot$	$\top$	0.0	1.0
$\top$	$\bot$	$\bot$	0.0	1.0
$\bot$	$\top$	$\top$	0.0	1.0
$\bot$	$\top$	$\bot$	0.0	1.0
$\bot$	$\bot$	$\top$	0.0	1.0
$\bot$	$\bot$	$\bot$	0.0	1.0

PGM to CNF

The Bayesian network can be encoded as a logical formula. Encodings ENC1 and ENC2 discussed by Chavira [1] both make use of the same indicator variables such as $\lambda_{stress(a)=true}$ and $\lambda_{stress(a)=false}$ (which introduces redundancy considering the fact that all network variables are boolean).

In `ENC1` each row in each CPT is encoded with equivalences. In `ENC2` an order is assumed over each variable's values. Then, each row of each CPT is encoded with implications. All indicator variables get a weight of 1, the parameter variables are assigned weights corresponding to the values in the CPTs.

For both of these a Python script was written to automate the creation of CNF files, since manual conversion to a `CNF` turned out to be cumbersome. The script also produces LaTeXoutput which is added to the appendix.

Since the noisy OR relations end up being encoded naively by considering each row of the CPT separately, a more compact encoding is obtained by replacing this part of the encoding with a one-liner. For example :

$$smokes(a) \Leftrightarrow aux_1 \vee aux_6 \vee aux_7$$

This can be converted to a `CNF` in the usual way :

$$\begin{aligned} &(smokes(a) \vee \neg aux_1 \vee \neg aux_6 \vee \neg aux_7) \\ &\wedge (\neg smokes(a) \vee aux_1) \\ &\wedge (\neg smokes(a) \vee aux_6) \\ &\wedge (\neg smokes(a) \vee aux_7) \end{aligned}$$

Weighted Model Counting

The previous paragraphs led to the construction of 5 different encodings. All of them lie in the `/src/bayesian/` folder, having the following file names :

Encoding	Filename
Conversion from ProbLog	`propositional.cnf`
`ENC1`	`enc1.cnf`
`ENC1` + noisy OR	`enc1_noisy.cnf`
`ENC2`	`enc2.cnf`
`ENC2` + noisy OR	`enc2_noisy.cnf`

Weighted model counting was performed on all of them using either `PySDD` or `miniC2D`. A modified version of `Cachet` that allows for the specification of negative weights was toyed with too, though not for all encodings as it expects a different format for specifying the weights. These were the results :

Encoding	`PySDD`	`miniC2D`
Conversion from ProbLog	XXX	XXX
`ENC1`	XXX	XXX
`ENC1` + noisy OR	XXX	XXX
`ENC2`	XXX	XXX
`ENC2` + noisy OR	XXX	XXX

The counts can be interpreted as probabilities (keeping in mind that some assumptions result from the semantics of the ProbLog program, such as independence of friendships). In this particular case since no query was specified (yet), the probability equals 1 since the count represents the probability of any possible world.

To answer a more complex query like $P(smokes(a) = \top)$ a line was added to the CNF files : $smokes(a)$. This makes sure that any model is one where person a actually smokes. A more complex query like $P(smokes(a) = \top \mid friends(a,b) = \top, friends(a,c) = \top)$ can be computed in the way Fierens proposed [3], by dividing the joint probability of these three atoms by the probability of the previous query (i.e. applying Bayes' formula).

The smallest circuit sizes that were found are the following :

Encoding	Hyperparameters	PySDD	miniC2D
Conversion from ProbLog	XXX	XXX	XXX
ENC1	XXX	XXX	XXX
ENC1 + noisy OR	XXX	XXX	XXX
ENC2	XXX	XXX	XXX
ENC2 + noisy OR	XXX	XXX	XXX

The weighted counts for the queries were :

Encoding	$P(smokes(a) = \top)$	$P(smokes(a) = \top \mid friends(a,b) = \top, friends(a,c) = \top)$
Conversion from ProbLog	XXX	XXX
ENC1	XXX	XXX
ENC1 + noisy OR	XXX	XXX
ENC2	XXX	XXX
ENC2 + noisy OR	XXX	XXX

Both `miniC2D` and `PySDD` work with sentential decision diagrams (SDDs) since they are both made at UCLA. SDDs are a particular kind of d-DNNF with useful properties [2]. The difference lies in the way these very SDDs are generated. `miniC2D` is a top-down compiler.

Lifted Inference

Calculating Probability with Probabilistic Databases

Starting from the following probabilistic database tables :

X	$stress(X)$
a	0.2
b	0.2
c	0.2

X	Y	$friends(X,Y)$
a	b	0.1
a	c	0.1
b	c	0.1
c	b	0.1

X	$p(X)$
a	0.3
b	0.3
c	0.3

X	Y	$p(X,Y)$
a	b	0.4
a	c	0.4
b	c	0.4
c	b	0.4

Table 1: Probabilistic database to be used for querying.

Querying for $smokes(a)$ can be done as follows :

$$(stress(a) \wedge p(a)) \vee (\exists x : p(a,x) \wedge friends(a,x) \wedge ((stress(x) \wedge p(x)) \vee \exists y : p(x,y) \wedge friends(x,y) \wedge stress(y) \wedge p(y)))$$

Rules can be applied to this formula. Applying the inclusive or rule gives :

$$1 - (1 - p_{stress(a)\ \wedge\ p(a)}) \times (1 - p_{s_1})$$

Where s_1 refers to the subformula for the first existential operator. The probability p_{s_1} cannot simply be calculated by exponentiation as the various instantiations of s_1 are dependent. However, there are only two such instantiations (for $smokes(a)$ and $smokes(b)$) and since the existential operator generalises the logical or, the formula for exclusive or can directly be used instead :

$$p_{s_1} = p_{friends(a,b)} \times p_{p(a,b)} \times p_{smokes(b)} + p_{friends(a,c)} \times p_{p(a,c)} \times p_{smokes(c)} - p_{both}$$

Where $p_{both} = p_{friends(a,b)} \wedge p_{p(a,b)} \wedge p_{friends(a,c)} \wedge p_{p(a,c)} \wedge p_{smokes(b)} \wedge p_{smokes(c)}$. The probability of either $smokes(b)$ or $smokes(c)$ is the same (as is the probability of $friends(a,b) \wedge p(a,b)$ and $friends(a,c) \wedge p(a,c)$, so this reduces to :

$$p_{s_1} = 2 \times p_{friends(a,b)} \times p_{p(a,b)} \times p_{smokes(b)} - p_{friends(a,b)\ \wedge\ friends(a,c)\ \wedge\ smokes(b)\ \wedge\ smokes(c)}$$

The probability of $smokes(b)$ can be calculated using the same rules (an inclusive or and the and-rule) :

$$\begin{aligned} 1 - (1 - p_{stress(b)} \times p_{p(b)}) \times (1 - p_{friends(b,c)} \times p_{p(b,c)} \times p_{stress(c)} \times p_{p(c)}) \\ = (1 - (1 - 0,2 \times 0,3) \times (1 - 0,1 \times 0,4 \times 0,3 \times 0,2) \\ \approx 0.62256 \end{aligned}$$

The probability of $p_{smokes(b) \ \wedge \ smokes(c)}$ can also be calculated with these rules, for example :

$$\begin{gathered}
p_{smokes(b) \ \wedge \ smokes(c)} \\
= p_{stress(b) \ \wedge \ p(b) \ \wedge \ smokes(c)} + p_{stress(c) \ \wedge \ p(c) \ \wedge \ smokes(b)} - p_{all} \\
= 2 \times p_{stress(b)} \times p_{p(b)} \times (p_{stress(c) \ \wedge \ p(c)} + p_{friends(b,c) \ \wedge \ p(b,c)} - p_{stress(c) \ \wedge \ p(c) \ \wedge \ friends(b,c) \ \wedge \ p(b,c)}) \\
- p_{stress(b)} \times p_{p(b)} \times p_{stress(c)} \times p_{p(c)}
\end{gathered}$$

The probability of $p_{stress(a) \ \wedge \ p(a)}$ equals $p_{p(a)} \times p_{stress(a)}$. After filling in all the missing probabilities of the involved literals (by looking them up in the database) the following total probability is obtained :

$$\begin{gathered}
1-(1-0,2\times 0,3)\times(1-(2\times 0,1\times 0,4\times(1-(1-0,2\times 0,3)\times(1-0,1\times 0,4\times 0,3\times 0,2))- \\
0,1\times 0,4\times 0,1\times 0,4\times(0,2\times 0,3\times(0,2\times 0,3+0,1\times 0,4-0,2\times 0,3\times 0,1\times 0,4)+0,2\times 0,3\times \\
(0,2\times 0,3+0,1\times 0,4-0,2\times 0,3\times 0,1\times 0,4)-0,2\times 0,3\times 0,2\times 0,3))) \\
\approx 0,06466945075
\end{gathered}$$

This is the same number as the one previously found by the weighted model counters. Some symmetry was taken advantage of.

Skolemization & Noisy OR

A new encoding can be made by, instead of converting the formula $X \Leftrightarrow Y_1 \vee Y_2 \vee ... \vee Y_n$ to CNF in the usual way, applying a Tseitin transformation. This leads to :

$$\begin{gathered}
X \vee \neg Y_1 \\
X \vee \neg Y_2 \\
... \\
X \vee \neg Y_n \\
X \vee T \\
T \vee \neg Y_1 \\
T \vee \neg Y_2 \\
... \\
T \vee \neg Y_n
\end{gathered}$$

Applying this transformation to the noisy OR relations in the encoding lead to a slightly *larger* circuit though, when tested with a program in which there were more ancestors, the resulting circuits were *smaller*. The very CNF files (`reference.cnf` and `lifted.cnf`) lie in the `/src/bayesian/` directory.

Parameter Learning

Learning from interpretations can be done by compiling these interpretations together with the base program into some kind of structure such that inference becomes tractable. In the algorithm that was written, a d-DNNF is generated for each interpretation (an SDD in particular since SharpSAT had trouble with a known issue). Then, the weights of the parameters of interest are updated iteratively until convergence i.e. until new weights don't differ much from the previous ones. The resulting (local) optimum approximates the parameter's real values.

The d-DNNFs are generated from CNFs with `PySDD`. During each iteration they are used to calculate marginals based on current values of the parameters. Their structure never changes, only their weights are updated as the algorithm progresses. Either because the new estimates have to be taken into account at the end of every iteration, or because the marginals have to be calculated (which is done by toying with the weights rather than by applying Bayes' theorem which was the approach used in section 1).

The algorithm is reasonably fast and some results are shown below :

Table 2: Results of the parameter learning algorithm. Initial weights were always set to 0.5 for the sake of reproducibility (setting them randomly is a matter of commenting out a line). On a regular computer running for 1000 iterations on 1000 examples took no longer than a minute. Various tests were done such as comparing with ProbLog's own system, using the same initial values.

Appendix

Bayesian Network Encodings

Indicator Clauses

$\lambda_{stress(a)=true} \vee \lambda_{stress(a)=false}$
$\neg\lambda_{stress(a)=true} \vee \neg\lambda_{stress(a)=false}$
$\lambda_{stress(b)=true} \vee \lambda_{stress(b)=false}$
$\neg\lambda_{stress(b)=true} \vee \neg\lambda_{stress(b)=false}$
$\lambda_{stress(c)=true} \vee \lambda_{stress(c)=false}$
$\neg\lambda_{stress(c)=true} \vee \neg\lambda_{stress(c)=false}$
$\lambda_{aux1=true} \vee \lambda_{aux1=false}$
$\neg\lambda_{aux1=true} \vee \neg\lambda_{aux1=false}$
$\lambda_{aux2=true} \vee \lambda_{aux2=false}$
$\neg\lambda_{aux2=true} \vee \neg\lambda_{aux2=false}$
$\lambda_{aux3=true} \vee \lambda_{aux3=false}$
$\neg\lambda_{aux3=true} \vee \neg\lambda_{aux3=false}$
$\lambda_{friends(a,b)=true} \vee \lambda_{friends(a,b)=false}$
$\neg\lambda_{friends(a,b)=true} \vee \neg\lambda_{friends(a,b)=false}$
$\lambda_{friends(a,c)=true} \vee \lambda_{friends(a,c)=false}$
$\neg\lambda_{friends(a,c)=true} \vee \neg\lambda_{friends(a,c)=false}$
$\lambda_{friends(b,c)=true} \vee \lambda_{friends(b,c)=false}$
$\neg\lambda_{friends(b,c)=true} \vee \neg\lambda_{friends(b,c)=false}$
$\lambda_{friends(c,b)=true} \vee \lambda_{friends(c,b)=false}$
$\neg\lambda_{friends(c,b)=true} \vee \neg\lambda_{friends(c,b)=false}$
$\lambda_{aux4=true} \vee \lambda_{aux4=false}$
$\neg\lambda_{aux4=true} \vee \neg\lambda_{aux4=false}$
$\lambda_{aux5=true} \vee \lambda_{aux5=false}$
$\neg\lambda_{aux5=true} \vee \neg\lambda_{aux5=false}$
$\lambda_{smokes(b)=true} \vee \lambda_{smokes(b)=false}$
$\neg\lambda_{smokes(b)=true} \vee \neg\lambda_{smokes(b)=false}$
$\lambda_{smokes(c)=true} \vee \lambda_{smokes(c)=false}$
$\neg\lambda_{smokes(c)=true} \vee \neg\lambda_{smokes(c)=false}$
$\lambda_{aux6=true} \vee \lambda_{aux6=false}$
$\neg\lambda_{aux6=true} \vee \neg\lambda_{aux6=false}$
$\lambda_{aux7=true} \vee \lambda_{aux7=false}$
$\neg\lambda_{aux7=true} \vee \neg\lambda_{aux7=false}$
$\lambda_{smokes(a)=true} \vee \lambda_{smokes(a)=false}$
$\neg\lambda_{smokes(a)=true} \vee \neg\lambda_{smokes(a)=false}$

ENC1 (Parameter Clauses)

$\lambda_{stress(a)=true} \Leftrightarrow \theta_{stress(a)=true}$
$\lambda_{stress(a)=false} \Leftrightarrow \theta_{stress(a)=false}$
$\lambda_{stress(b)=true} \Leftrightarrow \theta_{stress(b)=true}$
$\lambda_{stress(b)=false} \Leftrightarrow \theta_{stress(b)=false}$
$\lambda_{stress(c)=true} \Leftrightarrow \theta_{stress(c)=true}$
$\lambda_{stress(c)=false} \Leftrightarrow \theta_{stress(c)=false}$
$\lambda_{stress(a)=true} \wedge \lambda_{aux1=true} \Leftrightarrow \theta_{aux1=true|stress(a)=true}$
$\lambda_{stress(a)=true} \wedge \lambda_{aux1=false} \Leftrightarrow \theta_{aux1=false|stress(a)=true}$
$\lambda_{stress(a)=false} \wedge \lambda_{aux1=true} \Leftrightarrow \theta_{aux1=true|stress(a)=false}$
$\lambda_{stress(a)=false} \wedge \lambda_{aux1=false} \Leftrightarrow \theta_{aux1=false|stress(a)=false}$
$\lambda_{stress(b)=true} \wedge \lambda_{aux2=true} \Leftrightarrow \theta_{aux2=true|stress(b)=true}$
$\lambda_{stress(b)=true} \wedge \lambda_{aux2=false} \Leftrightarrow \theta_{aux2=false|stress(b)=true}$

$\lambda_{stress(b)=false} \wedge \lambda_{aux2=true} \Leftrightarrow \theta_{aux2=true|stress(b)=false}$
$\lambda_{stress(b)=false} \wedge \lambda_{aux2=false} \Leftrightarrow \theta_{aux2=false|stress(b)=false}$
$\lambda_{stress(c)=true} \wedge \lambda_{aux3=true} \Leftrightarrow \theta_{aux3=true|stress(c)=true}$
$\lambda_{stress(c)=true} \wedge \lambda_{aux3=false} \Leftrightarrow \theta_{aux3=false|stress(c)=true}$
$\lambda_{stress(c)=false} \wedge \lambda_{aux3=true} \Leftrightarrow \theta_{aux3=true|stress(c)=false}$
$\lambda_{stress(c)=false} \wedge \lambda_{aux3=false} \Leftrightarrow \theta_{aux3=false|stress(c)=false}$
$\lambda_{friends(a,b)=true} \Leftrightarrow \theta_{friends(a,b)=true}$
$\lambda_{friends(a,b)=false} \Leftrightarrow \theta_{friends(a,b)=false}$
$\lambda_{friends(a,c)=true} \Leftrightarrow \theta_{friends(a,c)=true}$
$\lambda_{friends(a,c)=false} \Leftrightarrow \theta_{friends(a,c)=false}$
$\lambda_{friends(b,c)=true} \Leftrightarrow \theta_{friends(b,c)=true}$
$\lambda_{friends(b,c)=false} \Leftrightarrow \theta_{friends(b,c)=false}$
$\lambda_{friends(c,b)=true} \Leftrightarrow \theta_{friends(c,b)=true}$
$\lambda_{friends(c,b)=false} \Leftrightarrow \theta_{friends(c,b)=false}$
$\lambda_{friends(b,c)=true} \wedge \lambda_{aux3=true} \wedge \lambda_{aux4=true} \Leftrightarrow \theta_{aux4=true|friends(b,c)=true \wedge aux3=true}$
$\lambda_{friends(b,c)=true} \wedge \lambda_{aux3=true} \wedge \lambda_{aux4=false} \Leftrightarrow \theta_{aux4=false|friends(b,c)=true \wedge aux3=true}$
$\lambda_{friends(b,c)=true} \wedge \lambda_{aux3=false} \wedge \lambda_{aux4=true} \Leftrightarrow \theta_{aux4=true|friends(b,c)=true \wedge aux3=false}$
$\lambda_{friends(b,c)=true} \wedge \lambda_{aux3=false} \wedge \lambda_{aux4=false} \Leftrightarrow \theta_{aux4=false|friends(b,c)=true \wedge aux3=false}$
$\lambda_{friends(b,c)=false} \wedge \lambda_{aux3=true} \wedge \lambda_{aux4=true} \Leftrightarrow \theta_{aux4=true|friends(b,c)=false \wedge aux3=true}$
$\lambda_{friends(b,c)=false} \wedge \lambda_{aux3=true} \wedge \lambda_{aux4=false} \Leftrightarrow \theta_{aux4=false|friends(b,c)=false \wedge aux3=true}$
$\lambda_{friends(b,c)=false} \wedge \lambda_{aux3=false} \wedge \lambda_{aux4=true} \Leftrightarrow \theta_{aux4=true|friends(b,c)=false \wedge aux3=false}$
$\lambda_{friends(b,c)=false} \wedge \lambda_{aux3=false} \wedge \lambda_{aux4=false} \Leftrightarrow \theta_{aux4=false|friends(b,c)=false \wedge aux3=false}$
$\lambda_{friends(c,b)=true} \wedge \lambda_{aux2=true} \wedge \lambda_{aux5=true} \Leftrightarrow \theta_{aux5=true|friends(c,b)=true \wedge aux2=true}$
$\lambda_{friends(c,b)=true} \wedge \lambda_{aux2=true} \wedge \lambda_{aux5=false} \Leftrightarrow \theta_{aux5=false|friends(c,b)=true \wedge aux2=true}$
$\lambda_{friends(c,b)=true} \wedge \lambda_{aux2=false} \wedge \lambda_{aux5=true} \Leftrightarrow \theta_{aux5=true|friends(c,b)=true \wedge aux2=false}$
$\lambda_{friends(c,b)=true} \wedge \lambda_{aux2=false} \wedge \lambda_{aux5=false} \Leftrightarrow \theta_{aux5=false|friends(c,b)=true \wedge aux2=false}$
$\lambda_{friends(c,b)=false} \wedge \lambda_{aux2=true} \wedge \lambda_{aux5=true} \Leftrightarrow \theta_{aux5=true|friends(c,b)=false \wedge aux2=true}$
$\lambda_{friends(c,b)=false} \wedge \lambda_{aux2=true} \wedge \lambda_{aux5=false} \Leftrightarrow \theta_{aux5=false|friends(c,b)=false \wedge aux2=true}$
$\lambda_{friends(c,b)=false} \wedge \lambda_{aux2=false} \wedge \lambda_{aux5=true} \Leftrightarrow \theta_{aux5=true|friends(c,b)=false \wedge aux2=false}$
$\lambda_{friends(c,b)=false} \wedge \lambda_{aux2=false} \wedge \lambda_{aux5=false} \Leftrightarrow \theta_{aux5=false|friends(c,b)=false \wedge aux2=false}$
$\lambda_{aux4=true} \wedge \lambda_{aux2=true} \wedge \lambda_{smokes(b)=true} \Leftrightarrow \theta_{smokes(b)=true|aux4=true \wedge aux2=true}$
$\lambda_{aux4=true} \wedge \lambda_{aux2=true} \wedge \lambda_{smokes(b)=false} \Leftrightarrow \theta_{smokes(b)=false|aux4=true \wedge aux2=true}$
$\lambda_{aux4=true} \wedge \lambda_{aux2=false} \wedge \lambda_{smokes(b)=true} \Leftrightarrow \theta_{smokes(b)=true|aux4=true \wedge aux2=false}$
$\lambda_{aux4=true} \wedge \lambda_{aux2=false} \wedge \lambda_{smokes(b)=false} \Leftrightarrow \theta_{smokes(b)=false|aux4=true \wedge aux2=false}$
$\lambda_{aux4=false} \wedge \lambda_{aux2=true} \wedge \lambda_{smokes(b)=true} \Leftrightarrow \theta_{smokes(b)=true|aux4=false \wedge aux2=true}$
$\lambda_{aux4=false} \wedge \lambda_{aux2=true} \wedge \lambda_{smokes(b)=false} \Leftrightarrow \theta_{smokes(b)=false|aux4=false \wedge aux2=true}$
$\lambda_{aux4=false} \wedge \lambda_{aux2=false} \wedge \lambda_{smokes(b)=true} \Leftrightarrow \theta_{smokes(b)=true|aux4=false \wedge aux2=false}$
$\lambda_{aux4=false} \wedge \lambda_{aux2=false} \wedge \lambda_{smokes(b)=false} \Leftrightarrow \theta_{smokes(b)=false|aux4=false \wedge aux2=false}$
$\lambda_{aux5=true} \wedge \lambda_{aux3=true} \wedge \lambda_{smokes(c)=true} \Leftrightarrow \theta_{smokes(c)=true|aux5=true \wedge aux3=true}$
$\lambda_{aux5=true} \wedge \lambda_{aux3=true} \wedge \lambda_{smokes(c)=false} \Leftrightarrow \theta_{smokes(c)=false|aux5=true \wedge aux3=true}$
$\lambda_{aux5=true} \wedge \lambda_{aux3=false} \wedge \lambda_{smokes(c)=true} \Leftrightarrow \theta_{smokes(c)=true|aux5=true \wedge aux3=false}$
$\lambda_{aux5=true} \wedge \lambda_{aux3=false} \wedge \lambda_{smokes(c)=false} \Leftrightarrow \theta_{smokes(c)=false|aux5=true \wedge aux3=false}$
$\lambda_{aux5=false} \wedge \lambda_{aux3=true} \wedge \lambda_{smokes(c)=true} \Leftrightarrow \theta_{smokes(c)=true|aux5=false \wedge aux3=true}$
$\lambda_{aux5=false} \wedge \lambda_{aux3=true} \wedge \lambda_{smokes(c)=false} \Leftrightarrow \theta_{smokes(c)=false|aux5=false \wedge aux3=true}$
$\lambda_{aux5=false} \wedge \lambda_{aux3=false} \wedge \lambda_{smokes(c)=true} \Leftrightarrow \theta_{smokes(c)=true|aux5=false \wedge aux3=false}$
$\lambda_{aux5=false} \wedge \lambda_{aux3=false} \wedge \lambda_{smokes(c)=false} \Leftrightarrow \theta_{smokes(c)=false|aux5=false \wedge aux3=false}$
$\lambda_{friends(a,b)=true} \wedge \lambda_{smokes(b)=true} \wedge \lambda_{aux6=true} \Leftrightarrow \theta_{aux6=true|friends(a,b)=true \wedge smokes(b)=true}$
$\lambda_{friends(a,b)=true} \wedge \lambda_{smokes(b)=true} \wedge \lambda_{aux6=false} \Leftrightarrow \theta_{aux6=false|friends(a,b)=true \wedge smokes(b)=true}$
$\lambda_{friends(a,b)=true} \wedge \lambda_{smokes(b)=false} \wedge \lambda_{aux6=true} \Leftrightarrow \theta_{aux6=true|friends(a,b)=true \wedge smokes(b)=false}$
$\lambda_{friends(a,b)=true} \wedge \lambda_{smokes(b)=false} \wedge \lambda_{aux6=false} \Leftrightarrow \theta_{aux6=false|friends(a,b)=true \wedge smokes(b)=false}$
$\lambda_{friends(a,b)=false} \wedge \lambda_{smokes(b)=true} \wedge \lambda_{aux6=true} \Leftrightarrow \theta_{aux6=true|friends(a,b)=false \wedge smokes(b)=true}$
$\lambda_{friends(a,b)=false} \wedge \lambda_{smokes(b)=true} \wedge \lambda_{aux6=false} \Leftrightarrow \theta_{aux6=false|friends(a,b)=false \wedge smokes(b)=true}$
$\lambda_{friends(a,b)=false} \wedge \lambda_{smokes(b)=false} \wedge \lambda_{aux6=true} \Leftrightarrow \theta_{aux6=true|friends(a,b)=false \wedge smokes(b)=false}$
$\lambda_{friends(a,b)=false} \wedge \lambda_{smokes(b)=false} \wedge \lambda_{aux6=false} \Leftrightarrow \theta_{aux6=false|friends(a,b)=false \wedge smokes(b)=false}$
$\lambda_{friends(a,c)=true} \wedge \lambda_{smokes(c)=true} \wedge \lambda_{aux7=true} \Leftrightarrow \theta_{aux7=true|friends(a,c)=true \wedge smokes(c)=true}$
$\lambda_{friends(a,c)=true} \wedge \lambda_{smokes(c)=true} \wedge \lambda_{aux7=false} \Leftrightarrow \theta_{aux7=false|friends(a,c)=true \wedge smokes(c)=true}$
$\lambda_{friends(a,c)=true} \wedge \lambda_{smokes(c)=false} \wedge \lambda_{aux7=true} \Leftrightarrow \theta_{aux7=true|friends(a,c)=true \wedge smokes(c)=false}$
$\lambda_{friends(a,c)=true} \wedge \lambda_{smokes(c)=false} \wedge \lambda_{aux7=false} \Leftrightarrow \theta_{aux7=false|friends(a,c)=true \wedge smokes(c)=false}$
$\lambda_{friends(a,c)=false} \wedge \lambda_{smokes(c)=true} \wedge \lambda_{aux7=true} \Leftrightarrow \theta_{aux7=true|friends(a,c)=false \wedge smokes(c)=true}$
$\lambda_{friends(a,c)=false} \wedge \lambda_{smokes(c)=true} \wedge \lambda_{aux7=false} \Leftrightarrow \theta_{aux7=false|friends(a,c)=false \wedge smokes(c)=true}$

$\lambda_{friends(a,c)=false} \wedge \lambda_{smokes(c)=false} \wedge \lambda_{aux7=true} \Leftrightarrow \theta_{aux7=true|friends(a,c)=false \wedge smokes(c)=false}$
$\lambda_{friends(a,c)=false} \wedge \lambda_{smokes(c)=false} \wedge \lambda_{aux7=false} \Leftrightarrow \theta_{aux7=false|friends(a,c)=false \wedge smokes(c)=false}$
$\lambda_{aux1=true} \wedge \lambda_{aux6=true} \wedge \lambda_{aux7=true} \wedge \lambda_{smokes(a)=true} \Leftrightarrow \theta_{smokes(a)=true|aux1=true \wedge aux6=true \wedge aux7=true}$
$\lambda_{aux1=true} \wedge \lambda_{aux6=true} \wedge \lambda_{aux7=true} \wedge \lambda_{smokes(a)=false} \Leftrightarrow \theta_{smokes(a)=false|aux1=true \wedge aux6=true \wedge aux7=true}$
$\lambda_{aux1=true} \wedge \lambda_{aux6=true} \wedge \lambda_{aux7=false} \wedge \lambda_{smokes(a)=true} \Leftrightarrow \theta_{smokes(a)=true|aux1=true \wedge aux6=true \wedge aux7=false}$
$\lambda_{aux1=true} \wedge \lambda_{aux6=true} \wedge \lambda_{aux7=false} \wedge \lambda_{smokes(a)=false} \Leftrightarrow \theta_{smokes(a)=false|aux1=true \wedge aux6=true \wedge aux7=false}$
$\lambda_{aux1=true} \wedge \lambda_{aux6=false} \wedge \lambda_{aux7=true} \wedge \lambda_{smokes(a)=true} \Leftrightarrow \theta_{smokes(a)=true|aux1=true \wedge aux6=false \wedge aux7=true}$
$\lambda_{aux1=true} \wedge \lambda_{aux6=false} \wedge \lambda_{aux7=true} \wedge \lambda_{smokes(a)=false} \Leftrightarrow \theta_{smokes(a)=false|aux1=true \wedge aux6=false \wedge aux7=true}$
$\lambda_{aux1=true} \wedge \lambda_{aux6=false} \wedge \lambda_{aux7=false} \wedge \lambda_{smokes(a)=true} \Leftrightarrow \theta_{smokes(a)=true|aux1=true \wedge aux6=false \wedge aux7=false}$
$\lambda_{aux1=true} \wedge \lambda_{aux6=false} \wedge \lambda_{aux7=false} \wedge \lambda_{smokes(a)=false} \Leftrightarrow \theta_{smokes(a)=false|aux1=true \wedge aux6=false \wedge aux7=false}$
$\lambda_{aux1=false} \wedge \lambda_{aux6=true} \wedge \lambda_{aux7=true} \wedge \lambda_{smokes(a)=true} \Leftrightarrow \theta_{smokes(a)=true|aux1=false \wedge aux6=true \wedge aux7=true}$
$\lambda_{aux1=false} \wedge \lambda_{aux6=true} \wedge \lambda_{aux7=true} \wedge \lambda_{smokes(a)=false} \Leftrightarrow \theta_{smokes(a)=false|aux1=false \wedge aux6=true \wedge aux7=true}$
$\lambda_{aux1=false} \wedge \lambda_{aux6=true} \wedge \lambda_{aux7=false} \wedge \lambda_{smokes(a)=true} \Leftrightarrow \theta_{smokes(a)=true|aux1=false \wedge aux6=true \wedge aux7=false}$
$\lambda_{aux1=false} \wedge \lambda_{aux6=true} \wedge \lambda_{aux7=false} \wedge \lambda_{smokes(a)=false} \Leftrightarrow \theta_{smokes(a)=false|aux1=false \wedge aux6=true \wedge aux7=false}$
$\lambda_{aux1=false} \wedge \lambda_{aux6=false} \wedge \lambda_{aux7=true} \wedge \lambda_{smokes(a)=true} \Leftrightarrow \theta_{smokes(a)=true|aux1=false \wedge aux6=false \wedge aux7=true}$
$\lambda_{aux1=false} \wedge \lambda_{aux6=false} \wedge \lambda_{aux7=true} \wedge \lambda_{smokes(a)=false} \Leftrightarrow \theta_{smokes(a)=false|aux1=false \wedge aux6=false \wedge aux7=true}$
$\lambda_{aux1=false} \wedge \lambda_{aux6=false} \wedge \lambda_{aux7=false} \wedge \lambda_{smokes(a)=true} \Leftrightarrow \theta_{smokes(a)=true|aux1=false \wedge aux6=false \wedge aux7=false}$
$\lambda_{aux1=false} \wedge \lambda_{aux6=false} \wedge \lambda_{aux7=false} \wedge \lambda_{smokes(a)=false} \Leftrightarrow \theta_{smokes(a)=false|aux1=false \wedge aux6=false \wedge aux7=false}$

ENC2 (Parameter Clauses)

$\rho_{stress(a)=true} \Rightarrow \lambda_{stress(a)=true}$
$\neg\rho_{stress(a)=true} \Rightarrow \lambda_{stress(a)=false}$
$\rho_{stress(b)=true} \Rightarrow \lambda_{stress(b)=true}$
$\neg\rho_{stress(b)=true} \Rightarrow \lambda_{stress(b)=false}$
$\rho_{stress(c)=true} \Rightarrow \lambda_{stress(c)=true}$
$\neg\rho_{stress(c)=true} \Rightarrow \lambda_{stress(c)=false}$
$\lambda_{stress(a)=true} \wedge \rho_{aux1=true} \Rightarrow \lambda_{aux1=true}$
$\lambda_{stress(a)=true} \wedge \neg\rho_{aux1=true} \Rightarrow \lambda_{aux1=false}$
$\lambda_{stress(a)=false} \wedge \rho_{aux1=true} \Rightarrow \lambda_{aux1=true}$
$\lambda_{stress(a)=false} \wedge \neg\rho_{aux1=true} \Rightarrow \lambda_{aux1=false}$
$\lambda_{stress(b)=true} \wedge \rho_{aux2=true} \Rightarrow \lambda_{aux2=true}$
$\lambda_{stress(b)=true} \wedge \neg\rho_{aux2=true} \Rightarrow \lambda_{aux2=false}$
$\lambda_{stress(b)=false} \wedge \rho_{aux2=true} \Rightarrow \lambda_{aux2=true}$
$\lambda_{stress(b)=false} \wedge \neg\rho_{aux2=true} \Rightarrow \lambda_{aux2=false}$
$\lambda_{stress(c)=true} \wedge \rho_{aux3=true} \Rightarrow \lambda_{aux3=true}$
$\lambda_{stress(c)=true} \wedge \neg\rho_{aux3=true} \Rightarrow \lambda_{aux3=false}$
$\lambda_{stress(c)=false} \wedge \rho_{aux3=true} \Rightarrow \lambda_{aux3=true}$
$\lambda_{stress(c)=false} \wedge \neg\rho_{aux3=true} \Rightarrow \lambda_{aux3=false}$
$\rho_{friends(a,b)=true} \Rightarrow \lambda_{friends(a,b)=true}$
$\neg\rho_{friends(a,b)=true} \Rightarrow \lambda_{friends(a,b)=false}$
$\rho_{friends(a,c)=true} \Rightarrow \lambda_{friends(a,c)=true}$
$\neg\rho_{friends(a,c)=true} \Rightarrow \lambda_{friends(a,c)=false}$
$\rho_{friends(b,c)=true} \Rightarrow \lambda_{friends(b,c)=true}$
$\neg\rho_{friends(b,c)=true} \Rightarrow \lambda_{friends(b,c)=false}$
$\rho_{friends(c,b)=true} \Rightarrow \lambda_{friends(c,b)=true}$
$\neg\rho_{friends(c,b)=true} \Rightarrow \lambda_{friends(c,b)=false}$
$\lambda_{friends(b,c)=true} \wedge \lambda_{aux3=true} \wedge \rho_{aux4=true} \Rightarrow \lambda_{aux4=true}$
$\lambda_{friends(b,c)=true} \wedge \lambda_{aux3=true} \wedge \neg\rho_{aux4=true} \Rightarrow \lambda_{aux4=false}$
$\lambda_{friends(b,c)=true} \wedge \lambda_{aux3=false} \wedge \rho_{aux4=true} \Rightarrow \lambda_{aux4=true}$
$\lambda_{friends(b,c)=true} \wedge \lambda_{aux3=false} \wedge \neg\rho_{aux4=true} \Rightarrow \lambda_{aux4=false}$
$\lambda_{friends(b,c)=false} \wedge \lambda_{aux3=true} \wedge \rho_{aux4=true} \Rightarrow \lambda_{aux4=true}$
$\lambda_{friends(b,c)=false} \wedge \lambda_{aux3=true} \wedge \neg\rho_{aux4=true} \Rightarrow \lambda_{aux4=false}$
$\lambda_{friends(b,c)=false} \wedge \lambda_{aux3=false} \wedge \rho_{aux4=true} \Rightarrow \lambda_{aux4=true}$
$\lambda_{friends(b,c)=false} \wedge \lambda_{aux3=false} \wedge \neg\rho_{aux4=true} \Rightarrow \lambda_{aux4=false}$
$\lambda_{friends(c,b)=true} \wedge \lambda_{aux2=true} \wedge \rho_{aux5=true} \Rightarrow \lambda_{aux5=true}$
$\lambda_{friends(c,b)=true} \wedge \lambda_{aux2=true} \wedge \neg\rho_{aux5=true} \Rightarrow \lambda_{aux5=false}$
$\lambda_{friends(c,b)=true} \wedge \lambda_{aux2=false} \wedge \rho_{aux5=true} \Rightarrow \lambda_{aux5=true}$
$\lambda_{friends(c,b)=true} \wedge \lambda_{aux2=false} \wedge \neg\rho_{aux5=true} \Rightarrow \lambda_{aux5=false}$
$\lambda_{friends(c,b)=false} \wedge \lambda_{aux2=true} \wedge \rho_{aux5=true} \Rightarrow \lambda_{aux5=true}$

$\lambda_{friends(c,b)=false} \wedge \lambda_{aux2=true} \wedge \neg\rho_{aux5=true} \Rightarrow \lambda_{aux5=false}$
$\lambda_{friends(c,b)=false} \wedge \lambda_{aux2=false} \wedge \rho_{aux5=true} \Rightarrow \lambda_{aux5=true}$
$\lambda_{friends(c,b)=false} \wedge \lambda_{aux2=false} \wedge \neg\rho_{aux5=true} \Rightarrow \lambda_{aux5=false}$
$\lambda_{aux4=true} \wedge \lambda_{aux2=true} \wedge \rho_{smokes(b)=true} \Rightarrow \lambda_{smokes(b)=true}$
$\lambda_{aux4=true} \wedge \lambda_{aux2=true} \wedge \neg\rho_{smokes(b)=true} \Rightarrow \lambda_{smokes(b)=false}$
$\lambda_{aux4=true} \wedge \lambda_{aux2=false} \wedge \rho_{smokes(b)=true} \Rightarrow \lambda_{smokes(b)=true}$
$\lambda_{aux4=true} \wedge \lambda_{aux2=false} \wedge \neg\rho_{smokes(b)=true} \Rightarrow \lambda_{smokes(b)=false}$
$\lambda_{aux4=false} \wedge \lambda_{aux2=true} \wedge \rho_{smokes(b)=true} \Rightarrow \lambda_{smokes(b)=true}$
$\lambda_{aux4=false} \wedge \lambda_{aux2=true} \wedge \neg\rho_{smokes(b)=true} \Rightarrow \lambda_{smokes(b)=false}$
$\lambda_{aux4=false} \wedge \lambda_{aux2=false} \wedge \rho_{smokes(b)=true} \Rightarrow \lambda_{smokes(b)=true}$
$\lambda_{aux4=false} \wedge \lambda_{aux2=false} \wedge \neg\rho_{smokes(b)=true} \Rightarrow \lambda_{smokes(b)=false}$
$\lambda_{aux5=true} \wedge \lambda_{aux3=true} \wedge \rho_{smokes(c)=true} \Rightarrow \lambda_{smokes(c)=true}$
$\lambda_{aux5=true} \wedge \lambda_{aux3=true} \wedge \neg\rho_{smokes(c)=true} \Rightarrow \lambda_{smokes(c)=false}$
$\lambda_{aux5=true} \wedge \lambda_{aux3=false} \wedge \rho_{smokes(c)=true} \Rightarrow \lambda_{smokes(c)=true}$
$\lambda_{aux5=true} \wedge \lambda_{aux3=false} \wedge \neg\rho_{smokes(c)=true} \Rightarrow \lambda_{smokes(c)=false}$
$\lambda_{aux5=false} \wedge \lambda_{aux3=true} \wedge \rho_{smokes(c)=true} \Rightarrow \lambda_{smokes(c)=true}$
$\lambda_{aux5=false} \wedge \lambda_{aux3=true} \wedge \neg\rho_{smokes(c)=true} \Rightarrow \lambda_{smokes(c)=false}$
$\lambda_{aux5=false} \wedge \lambda_{aux3=false} \wedge \rho_{smokes(c)=true} \Rightarrow \lambda_{smokes(c)=true}$
$\lambda_{aux5=false} \wedge \lambda_{aux3=false} \wedge \neg\rho_{smokes(c)=true} \Rightarrow \lambda_{smokes(c)=false}$
$\lambda_{friends(a,b)=true} \wedge \lambda_{smokes(b)=true} \wedge \rho_{aux6=true} \Rightarrow \lambda_{aux6=true}$
$\lambda_{friends(a,b)=true} \wedge \lambda_{smokes(b)=true} \wedge \neg\rho_{aux6=true} \Rightarrow \lambda_{aux6=false}$
$\lambda_{friends(a,b)=true} \wedge \lambda_{smokes(b)=false} \wedge \rho_{aux6=true} \Rightarrow \lambda_{aux6=true}$
$\lambda_{friends(a,b)=true} \wedge \lambda_{smokes(b)=false} \wedge \neg\rho_{aux6=true} \Rightarrow \lambda_{aux6=false}$
$\lambda_{friends(a,b)=false} \wedge \lambda_{smokes(b)=true} \wedge \rho_{aux6=true} \Rightarrow \lambda_{aux6=true}$
$\lambda_{friends(a,b)=false} \wedge \lambda_{smokes(b)=true} \wedge \neg\rho_{aux6=true} \Rightarrow \lambda_{aux6=false}$
$\lambda_{friends(a,b)=false} \wedge \lambda_{smokes(b)=false} \wedge \rho_{aux6=true} \Rightarrow \lambda_{aux6=true}$
$\lambda_{friends(a,b)=false} \wedge \lambda_{smokes(b)=false} \wedge \neg\rho_{aux6=true} \Rightarrow \lambda_{aux6=false}$
$\lambda_{friends(a,c)=true} \wedge \lambda_{smokes(c)=true} \wedge \rho_{aux7=true} \Rightarrow \lambda_{aux7=true}$
$\lambda_{friends(a,c)=true} \wedge \lambda_{smokes(c)=true} \wedge \neg\rho_{aux7=true} \Rightarrow \lambda_{aux7=false}$
$\lambda_{friends(a,c)=true} \wedge \lambda_{smokes(c)=false} \wedge \rho_{aux7=true} \Rightarrow \lambda_{aux7=true}$
$\lambda_{friends(a,c)=true} \wedge \lambda_{smokes(c)=false} \wedge \neg\rho_{aux7=true} \Rightarrow \lambda_{aux7=false}$
$\lambda_{friends(a,c)=false} \wedge \lambda_{smokes(c)=true} \wedge \rho_{aux7=true} \Rightarrow \lambda_{aux7=true}$
$\lambda_{friends(a,c)=false} \wedge \lambda_{smokes(c)=true} \wedge \neg\rho_{aux7=true} \Rightarrow \lambda_{aux7=false}$
$\lambda_{friends(a,c)=false} \wedge \lambda_{smokes(c)=false} \wedge \rho_{aux7=true} \Rightarrow \lambda_{aux7=true}$
$\lambda_{friends(a,c)=false} \wedge \lambda_{smokes(c)=false} \wedge \neg\rho_{aux7=true} \Rightarrow \lambda_{aux7=false}$
$\lambda_{aux1=true} \wedge \lambda_{aux6=true} \wedge \lambda_{aux7=true} \wedge \rho_{smokes(a)=true} \Rightarrow \lambda_{smokes(a)=true}$
$\lambda_{aux1=true} \wedge \lambda_{aux6=true} \wedge \lambda_{aux7=true} \wedge \neg\rho_{smokes(a)=true} \Rightarrow \lambda_{smokes(a)=false}$
$\lambda_{aux1=true} \wedge \lambda_{aux6=true} \wedge \lambda_{aux7=false} \wedge \rho_{smokes(a)=true} \Rightarrow \lambda_{smokes(a)=true}$
$\lambda_{aux1=true} \wedge \lambda_{aux6=true} \wedge \lambda_{aux7=false} \wedge \neg\rho_{smokes(a)=true} \Rightarrow \lambda_{smokes(a)=false}$
$\lambda_{aux1=true} \wedge \lambda_{aux6=false} \wedge \lambda_{aux7=true} \wedge \rho_{smokes(a)=true} \Rightarrow \lambda_{smokes(a)=true}$
$\lambda_{aux1=true} \wedge \lambda_{aux6=false} \wedge \lambda_{aux7=true} \wedge \neg\rho_{smokes(a)=true} \Rightarrow \lambda_{smokes(a)=false}$
$\lambda_{aux1=true} \wedge \lambda_{aux6=false} \wedge \lambda_{aux7=false} \wedge \rho_{smokes(a)=true} \Rightarrow \lambda_{smokes(a)=true}$
$\lambda_{aux1=true} \wedge \lambda_{aux6=false} \wedge \lambda_{aux7=false} \wedge \neg\rho_{smokes(a)=true} \Rightarrow \lambda_{smokes(a)=false}$
$\lambda_{aux1=false} \wedge \lambda_{aux6=true} \wedge \lambda_{aux7=true} \wedge \rho_{smokes(a)=true} \Rightarrow \lambda_{smokes(a)=true}$
$\lambda_{aux1=false} \wedge \lambda_{aux6=true} \wedge \lambda_{aux7=true} \wedge \neg\rho_{smokes(a)=true} \Rightarrow \lambda_{smokes(a)=false}$
$\lambda_{aux1=false} \wedge \lambda_{aux6=true} \wedge \lambda_{aux7=false} \wedge \rho_{smokes(a)=true} \Rightarrow \lambda_{smokes(a)=true}$
$\lambda_{aux1=false} \wedge \lambda_{aux6=true} \wedge \lambda_{aux7=false} \wedge \neg\rho_{smokes(a)=true} \Rightarrow \lambda_{smokes(a)=false}$
$\lambda_{aux1=false} \wedge \lambda_{aux6=false} \wedge \lambda_{aux7=true} \wedge \rho_{smokes(a)=true} \Rightarrow \lambda_{smokes(a)=true}$
$\lambda_{aux1=false} \wedge \lambda_{aux6=false} \wedge \lambda_{aux7=true} \wedge \neg\rho_{smokes(a)=true} \Rightarrow \lambda_{smokes(a)=false}$
$\lambda_{aux1=false} \wedge \lambda_{aux6=false} \wedge \lambda_{aux7=false} \wedge \rho_{smokes(a)=true} \Rightarrow \lambda_{smokes(a)=true}$
$\lambda_{aux1=false} \wedge \lambda_{aux6=false} \wedge \lambda_{aux7=false} \wedge \neg\rho_{smokes(a)=true} \Rightarrow \lambda_{smokes(a)=false}$

References

[1] Mark Chavira and Adnan Darwiche. On probabilistic inference by weighted model counting. *Artificial Intelligence*, 172(6):772 – 799, 2008.

[2] Adnan Darwiche. Sdd: A new canonical representation of propositional knowledge bases.

[3] Daan Fierens, Guy Van Den Broeck, Joris Renkens, Dimitar Shterionov, Bernd Gutmann, Ingo Thon, Gerda Janssens, and Luc De Raedt. Inference and learning in probabilistic logic programs using weighted boolean formulas. *Theory and Practice of Logic Programming*, 15(3):358–401, April 2014.