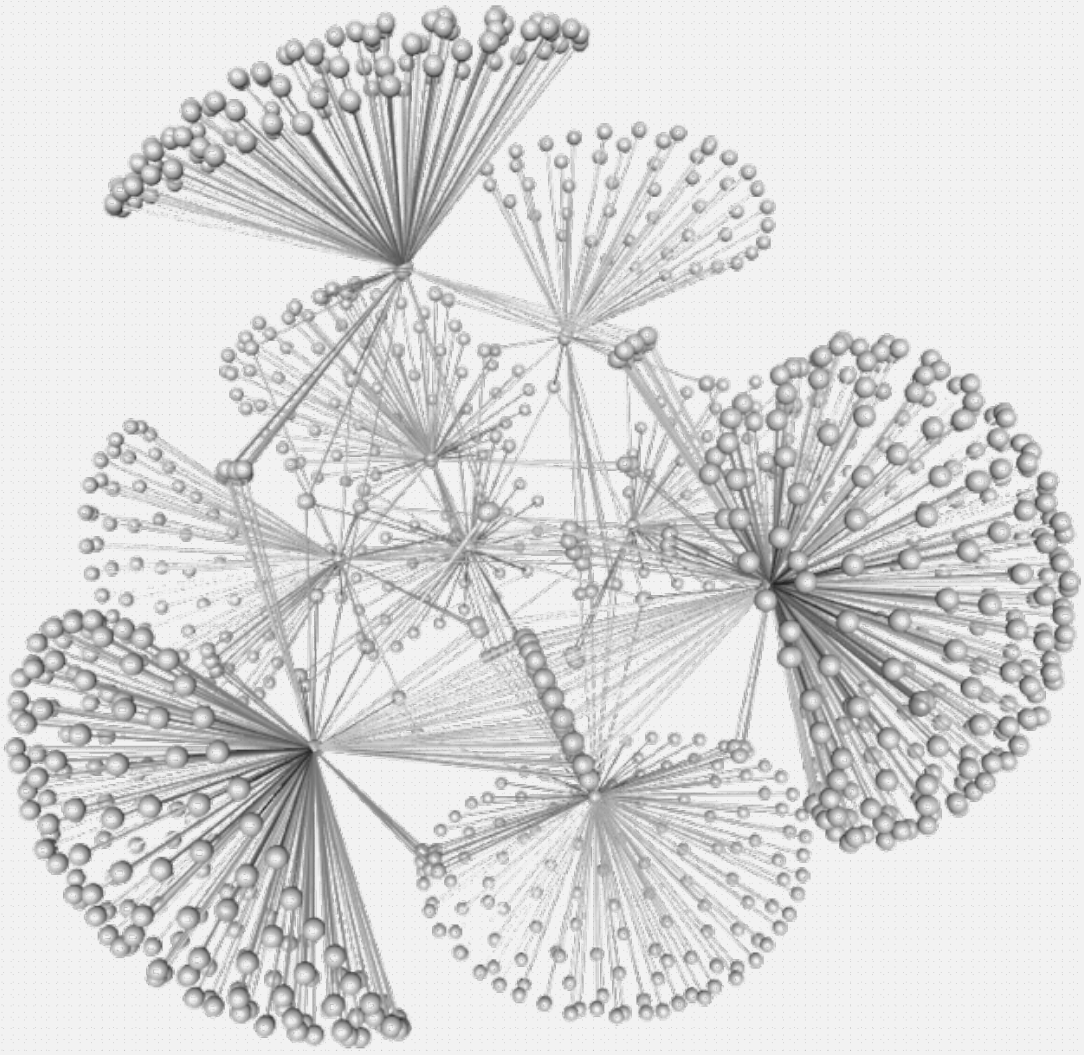
Probabilistic Programming

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.

```
person(a).
person(b).
person(c).
0.2::stress(X) :- person(X).
0.1::friends(X,Y) :- person(X), person(Y).
0.3::smokes(X) :- stress(X).
0.4::smokes(X) :- friends(X,Y), smokes(Y).
query(smokes(a)).
```

Code snippet 1: PROBLOG program used throughout the first two chapters of the report.

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 2: 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 3: Relevant ground program without cycles.

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

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

Which corresponds to the following `CNF` :

$$
\begin{aligned}
&(\neg smokes(a) \vee stress(a) \vee friends(a,b) \vee friends(a,c)) \\
&\wedge (\neg smokes(a) \vee stress(a) \vee friends(a,b) \vee smokes(c)) \\
&\wedge (\neg smokes(a) \vee stress(a) \vee friends(a,b) \vee p(a,c)) \\
&\wedge (\neg smokes(a) \vee stress(a) \vee smokes(b) \vee friends(a,c)) \\
&\wedge (\neg smokes(a) \vee stress(a) \vee smokes(b) \vee smokes(c)) \\
&\wedge (\neg smokes(a) \vee stress(a) \vee smokes(b) \vee p(a,c)) \\
&\wedge (\neg smokes(a) \vee stress(a) \vee p(a,b) \vee friends(a,c)) \\
&\wedge (\neg smokes(a) \vee stress(a) \vee p(a,b) \vee smokes(c)) \\
&\wedge (\neg smokes(a) \vee stress(a) \vee p(a,b) \vee p(a,c)) \\
&\wedge (\neg smokes(a) \vee p(a) \vee friends(a,b) \vee friends(a,c)) \\
&\wedge (\neg smokes(a) \vee p(a) \vee friends(a,b) \vee smokes(c)) \\
&\wedge (\neg smokes(a) \vee p(a) \vee friends(a,b) \vee p(a,c)) \\
&\wedge (\neg smokes(a) \vee p(a) \vee smokes(b) \vee friends(a,c)) \\
&\wedge (\neg smokes(a) \vee p(a) \vee smokes(b) \vee smokes(c)) \\
&\wedge (\neg smokes(a) \vee p(a) \vee smokes(b) \vee p(a,c)) \\
&\wedge (\neg smokes(a) \vee p(a) \vee p(a,b) \vee friends(a,c)) \\
&\wedge (\neg smokes(a) \vee p(a) \vee p(a,b) \vee smokes(c)) \\
&\wedge (\neg smokes(a) \vee p(a) \vee p(a,b) \vee p(a,c)) \\
&\wedge (\neg stress(a) \vee \neg p(a) \vee smokes(a)) \\
&\wedge (\neg friends(a,b) \vee \neg smokes(b) \vee \neg p(a,b) \vee smokes(a)) \\
&\wedge (\neg friends(a,c) \vee \neg smokes(c) \vee \neg p(a,c) \vee smokes(a)) \\
&\wedge (\neg smokes(b) \vee stress(b) \vee friends(b,c)) \\
&\wedge (\neg smokes(b) \vee stress(b) \vee stress(c)) \\
&\wedge (\neg smokes(b) \vee stress(b) \vee p(c)) \\
&\wedge (\neg smokes(b) \vee stress(b) \vee p(b,c) \\
&\wedge (\neg smokes(b) \vee p(b) \vee friends(b,c)) \\
&\wedge (\neg smokes(b) \vee p(b) \vee stress(c)) \\
&\wedge (\neg smokes(b) \vee p(b) \vee p(c)) \\
&\wedge (\neg smokes(b) \vee p(b) \vee p(b,c) \\
&\wedge (\neg stress(b) \vee \neg p(b) \vee smokes(b)) \\
&\wedge (\neg friends(b,c) \vee \neg stress(c) \vee \neg p(c) \vee \neg p(b,c) \vee \\
&smokes(b)) \\
&\wedge (\neg smokes(c) \vee stress(c) \vee friends(c,b)) \\
&\wedge (\neg smokes(c) \vee stress(c) \vee stress(b)) \\
&\wedge (\neg smokes(c) \vee stress(c) \vee p(b)) \\
&\wedge (\neg smokes(c) \vee stress(c) \vee p(c,b)) \\
&\wedge (\neg smokes(c) \vee p(c) \vee friends(c,b)) \\
&\wedge (\neg smokes(c) \vee p(c) \vee stress(b)) \\
&\wedge (\neg smokes(c) \vee p(c) \vee p(b)) \\
&\wedge (\neg smokes(c) \vee p(c) \vee p(c,b)) \\
&\wedge (\neg stress(c) \vee \neg p(c) \vee smokes(c)) \\
&\wedge (\neg friends(c,b) \vee \neg stress(b) \vee \neg p(b) \vee \neg p(c,b) \vee \\
&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 (like those of the $stress(a)$, $stress(b)$ and $stress(c)$ nodes for example).

	$stress(\{a,b,c\})$
$\top$	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 by a parameter clause.
In `ENC2` an order is assumed over each variable's values. Then, each row of each CPT is encoded by an equivalence.

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 :

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

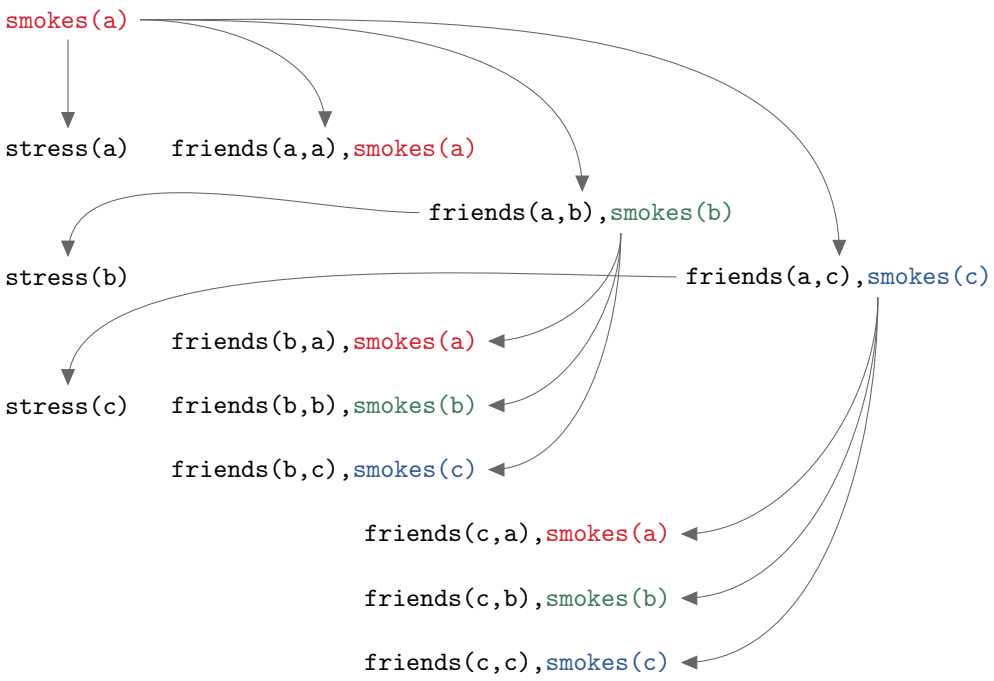


Figure 1: Trie representing proofs of the query. Coloured atoms indicate the presence of cycles.

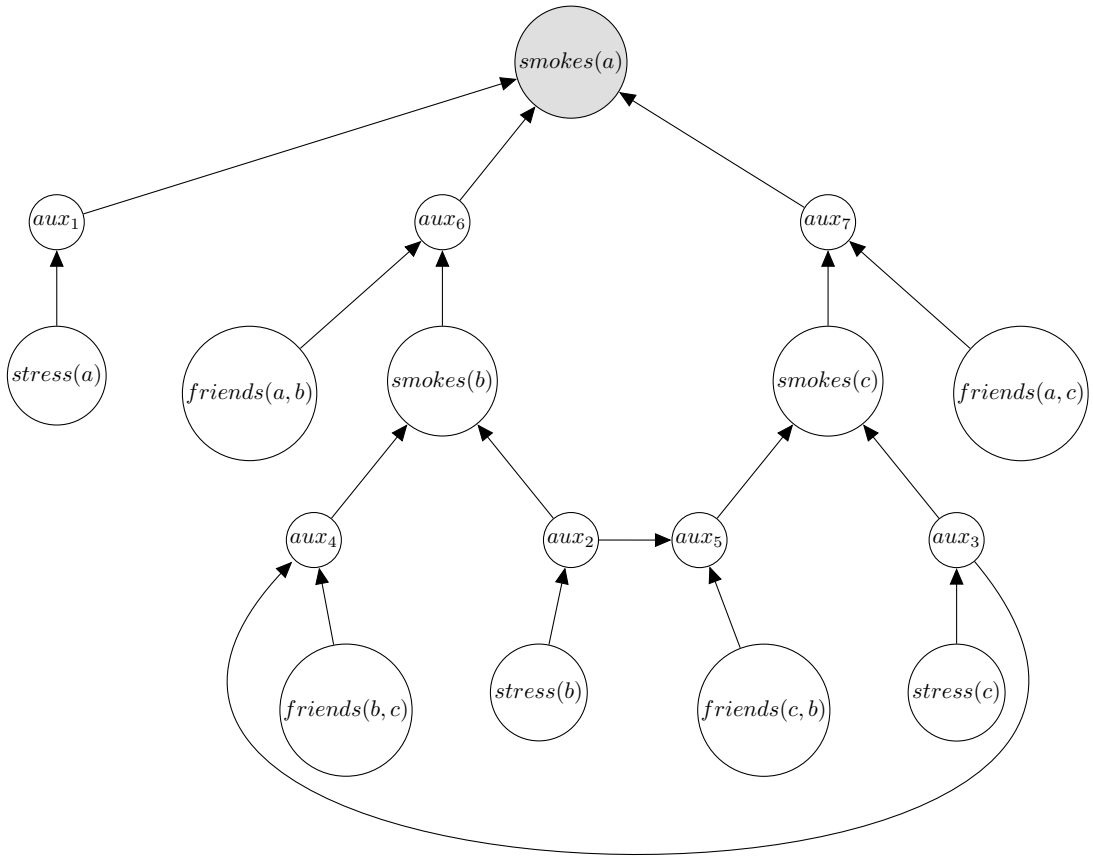


Figure 2: Bayesian network corresponding to the ground acyclic program.

Weighted Model Counting

The previous paragraphs led to the construction of 5 different encodings. 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	File location	`PySDD`	`miniC2D`
Conversion from ProbLog	`/src/bayesian/propositional.cnf`	XXX	XXX
`ENC1`	`/src/bayesian/enc1.cnf`	XXX	XXX
`ENC1` + noisy OR	`/src/bayesian/enc1_noisy.cnf`	XXX	XXX
`ENC2`	`/src/bayesian/enc2.cnf`	XXX	XXX
`ENC2` + noisy OR	`/src/bayesian/enc2_noisy.cnf`	XXX	XXX

The resulting 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.

The smallest circuit sizes were 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

To answer 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 [2], by dividing the joint probability of these three atoms by the probability of the previous query. One could also toy with the weights specified in the headers. Now for the results :

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

EXPLAIN DIFFERENCE BETWEEN MODEL COUNTERS HERE

Lifted Inference

Calculating Probability with Probabilistic Databases

Given 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 :

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

Rules can be applied to this formula though one has to keep in mind that the subformulae, in this case, are dependent. Applying the rules one by one gives the following results :

$$rules$$

The probabilities can now be looked up in the database itself, which gives :

$$gives$$

This is the same number as the one previously found by the weighted model counters.

Skolemization & Noisy OR

A new encoding can be made by considering the very first CNF that was constructed. Instead of converting the formula $X \Leftrightarrow Y_1 \vee Y_2 \vee ... \vee Y_n$ to CNF in the usual way, a Tseitin transformation is applied. 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*.

Parameter Learning

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] 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.