Online appendix for the paper

Computing LP$^{\text{MLN}}$ *Using ASP and MLN Solvers*

published in Theory and Practice of Logic Programming

Joohyung Lee, Samidh Talsania, and Yi Wang
School of Computing, Informatics and Decision Systems Engineering
Arizona State University, Tempe, AZ, USA

Appendix A Bayesian Network in LP$^{\text{MLN}}$

It is easy to represent Bayesian networks in LP$^{\text{MLN}}$ similar to the way Bayesian networks are represented by weighted Boolean formulas (Sang et al. 2005).

We assume all random variables are Boolean. Each conditional probability table associated with the nodes can be represented by a set of probabilistic facts. For each CPT entry $P(V = \mathbf{t} \mid V_1 = S_1, \dots, V_n = S_n) = p$ where $S_1, \dots, S_n \in \{\mathbf{t}, \mathbf{f}\}$, we include a set of weighted facts

- $ln(p/(1-p)):\ \ PF(V, S_1, \dots, S_n)$ if $0 < p < 1$;
- $\alpha:\ \ PF(V, S_1, \dots, S_n)$ if $p = 1$;
- $\alpha:\ \leftarrow$ *not* $PF(V, S_1, \dots, S_n)$ if $p = 0$.

For each node V whose parents are $V_1, \dots, V_n$, the directed edges can be represented by rules

$$\alpha:\ V \leftarrow V_1^{S_1}, \dots, V_n^{S_n}, PF(V, S_1, \dots, S_n) \qquad (S_1, \dots, S_n \in \{\mathbf{t}, \mathbf{f}\})$$

where $V_i^{S_i}$ is V_i if S_i is **t**, and *not* V_i otherwise.

For example, in the firing example in Figure A 1, the conditional probability table for the node "alarm" can be represented by

```
@log(0.5/0.5)      pf(a,t1f1).          @log(0.99/0.01)      pf(a,t0f1).
@log(0.85/0.15)    pf(a,t1f0).          @log(0.0001/0.0009)  pf(a,t0f0).
```

The directed edges can be represented by hard rules as follows:

```
tampering :- pf(t).

fire :- pf(f).

alarm :- tampering, fire, pf(a,t1f1).
alarm :- tampering, not fire, pf(a,t1f0).
alarm :- not tampering, fire, pf(a,t0f1).
alarm :- not tampering, not fire, pf(a,t0f0).

smoke :- fire, pf(s,f1).
smoke :- not fire, pf(s,f0).

leaving :- alarm, pf(l,a1).
leaving :- not alarm, pf(l,a0).

report :- leaving, pf(r,l1).
report :- not leaving, pf(r,l0).
```

Theorem 4

For any Bayesian network whose random variables are Boolean and any interpretation I, the probability of I according to the Bayesian network semantics coincides with the probability of I for the translated LP$^{\text{MLN}}$ program.

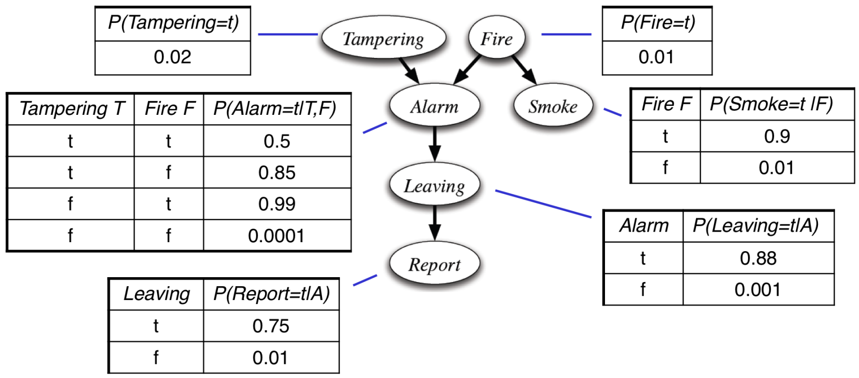

Fig. A 1. Bayes Net Example

Since Bayesian networks are represented by directed acyclic graphs, LP$^{\text{MLN}}$ programs that represent them are always tight. So both LPMLN2ASP and LPMLN2MLN can be used to compute Bayesian networks.

- *Diagnostic Inference* is to compute the probability of the cause given the effect. For example, to compute $P(fire = \mathbf{t} \mid leaving = \mathbf{t})$, the user can invoke

  ```
  lpmln2asp -i fire-bayes.lpmln -e evid.db -q fire
  ```

  where `evid.db` contains the line

  ```
  :- not leaving.
  ```

  This outputs

  ```
  fire 0.352151116689
  ```

- *Predictive Inference* is to compute the probability of the effect given the cause. For example, to compute $P(leaving = \mathbf{t} \mid fire = \mathbf{t})$, the user can invoke

  ```
  lpmln2asp -i fire-bayes.lpmln -e evid.db  -q leaving
  ```

  where `evid.db` contains the line

  ```
  :- not fire.
  ```

  This outputs

  ```
  leaving 0.862603541626
  ```

- *Mixed Inference* is to combine *predictive* and *diagnostic* inference. For example, to compute $P(alarm = \mathbf{t} \mid fire = \mathbf{f}, leaving = \mathbf{t})$, the user can invoke

  ```
  lpmln2asp -i fire-bayes.lpmln -e evid.db  -q alarm
  ```

  where `evid.db` contains two lines

  ```
  :- fire.
  :- not leaving.
  ```

  This outputs

  ```
  alarm 0.938679679707
  ```

- *Intercausal Inference* is to compute the probability of a cause given an effect common to multiple causes. For example, to compute $P(tampering = \mathbf{t} \mid fire = \mathbf{t}, alarm = \mathbf{t})$, the user can invoke

```
lpmln2asp -i fire-bayes.lpmln -e evid.db  -q tampering
```

where `evid.db` contains two lines

```
:- not fire.
:- not alarm.
```

This outputs

```
tampering 0.0102021964693
```

- *Explaining away*: Suppose we know that $alarm$ rang. Then we can use *Diagnostic Inference* to calculate $P(tampering = \mathbf{t} \mid alarm = \mathbf{t})$. But what happens if we now know that there was a $fire$ as well? In this case $P(tampering = \mathbf{t} \mid alarm = \mathbf{t})$ will change to $P(tampering = \mathbf{t} \mid fire = \mathbf{t}, alarm = \mathbf{t})$. In this case, knowing that there was a $fire$ explains away $alarm$, and hence affecting the probability of $tampering$. For example, to compute $P(tampering = \mathbf{t} \mid alarm = \mathbf{t})$, the user can invoke

```
lpmln2asp -i fire-bayes.lpmln -e evid.db  -q tampering
```

where `evid.db` contains line

```
:- not alarm.
```

This outputs

```
tampering 0.633397289908
```

If we compare this result with the result of *Intercausal Inference*, we see that $P(tampering = \mathbf{t} \mid alarm = \mathbf{t}) > P(tampering = \mathbf{t} \mid fire = \mathbf{t}, alarm = \mathbf{t})$. Observing the value of $fire$ explains away the $tampering$ i.e., the probability of $tampering$ decreases.

Appendix B Proof of Theorem 1

Theorem 1 For any LP$^{\text{MLN}}$ program Π and any interpretation I,

$$W_\Pi(I) \propto W_\Pi^{\text{pnt}}(I) \quad \text{and} \quad P_\Pi(I) = P_\Pi^{\text{pnt}}(I).$$

Proof. Let

$$TW_\Pi = exp\left(\sum_{w:F\in\Pi} w\right).$$

We first show that $W_\Pi(I) = TW_\Pi \cdot W_\Pi^{\text{pnt}}(I)$. This is obvious when $I \notin \text{SM}[\Pi]$.

When $I \in \mathrm{SM}[\Pi]$, we have

$$\begin{aligned}
W_\Pi(I) &= exp\Bigg(\sum_{w\,:\,F\,\in\,\Pi \text{ and } I\,\models\,F} w\Bigg)\\
&= exp\Bigg(\sum_{w:F\in\Pi} w - \sum_{w\,:\,F\,\in\,\Pi \text{ and } I\,\not\models\,F} w)\\
&= exp\Bigg(\sum_{w:F\in\Pi} w\Bigg)\cdot exp\Bigg(-\sum_{w\,:\,F\,\in\,\Pi \text{ and } I\,\not\models\,F} w\Bigg)\\
&= TW_\Pi \cdot exp\Bigg(-\sum_{w\,:\,F\,\in\,\Pi \text{ and } I\,\not\models\,F} w\Bigg)\\
&= TW_\Pi \cdot W^{\rm pnt}_\Pi(I).
\end{aligned}$$

Consequently,

$$\begin{aligned}
P_\Pi(I) &= \frac{W_\Pi(I)}{\sum_J W_\Pi(J)}\\
&= \frac{TW_\Pi \cdot W^{\rm pnt}_\Pi(I)}{\sum_J TW_\Pi \cdot W^{\rm pnt}_\Pi(J)}\\
&= \frac{W^{\rm pnt}_\Pi(I)}{\sum_J W^{\rm pnt}_\Pi(J)} \cdot \frac{TW_\Pi}{TW_\Pi}\\
&= \frac{W^{\rm pnt}_\Pi(I)}{\sum_J W^{\rm pnt}_\Pi(J)}\\
&= P^{\rm pnt}_\Pi(I).
\end{aligned}$$

□

Appendix C Proof of Theorem 2

We divide the ground program obtained from $\mathsf{lpmln2asp}^{\mathsf{rwd}}(\Pi)$ into three parts:

$$SAT(\Pi) \cup ORIGIN(\Pi) \cup WC(\Pi)$$

where

$$\begin{aligned}
SAT(\Pi) = &\{\mathtt{sat}(i, w_i, \mathbf{c}) \leftarrow \mathit{Head}_i(\mathbf{c}) \mid w_i :\ \mathit{Head}_i(\mathbf{c}) \leftarrow \mathit{Body}_i(\mathbf{c}) \in \mathit{Gr}(\Pi)\} \cup\\
&\{\mathtt{sat}(i, w_i, \mathbf{c}) \leftarrow \mathtt{not}\ \mathit{Body}_i(\mathbf{c}) \mid w_i :\ \mathit{Head}_i(\mathbf{c}) \leftarrow \mathit{Body}_i(\mathbf{c}) \in \mathit{Gr}(\Pi)\}
\end{aligned}$$

$$ORIGIN(\Pi) = \{\mathit{Head}_i(\mathbf{c}) \leftarrow \mathit{Body}_i(\mathbf{c}), \mathtt{not}\ \mathtt{not}\ \mathtt{sat}(i, w_i, \mathbf{c}) \mid w_i :\ \mathit{Head}_i(\mathbf{c}) \leftarrow \mathit{Body}_i(\mathbf{c}) \in \mathit{Gr}(\Pi)\}$$

and

$$WC(\Pi) = \{{:}{\sim}\ \mathtt{sat}(i, w_i, \mathbf{c}).\ \ [-w_i@l, i, \mathbf{c}] \mid w_i :\ \mathit{Head}_i(\mathbf{c}) \leftarrow \mathit{Body}_i(\mathbf{c}) \in \mathit{Gr}(\Pi)\}$$

Lemma 1
For any $\mathrm{LP}^{\mathrm{MLN}}$ program Π,

$$\phi(I) = I \cup \{\mathtt{sat}(i, w_i, \mathbf{c}) \mid w_i : \mathit{Head}_i(\mathbf{c}) \leftarrow \mathit{Body}_i(\mathbf{c}) \in \mathit{Gr}(\Pi), I \models \mathit{Head}_i(\mathbf{c}) \leftarrow \mathit{Body}_i(\mathbf{c})\}$$

is a 1-1 correspondence between $\mathrm{SM}[\Pi]$ and the stable models of $SAT(\Pi) \cup ORIGIN(\Pi)$.

Proof. Let σ be the signature of Π, and let σ_{sat} be the set

$$\{\texttt{sat}(i, w_i, \mathbf{c}) \mid w_i : \ \mathit{Head}_i(\mathbf{c}) \leftarrow \mathit{Body}_i(\mathbf{c}) \in Gr(\Pi)\}.$$

It can be seen that

- each strongly connected component of the dependency graph of $SAT(\Pi) \cup ORIGIN(\Pi)$ w.r.t. $\sigma \cup \sigma_{sat}$ is a subset of σ or a subset of σ_{sat};
- no atom in σ_{sat} has a strictly positive occurrence in $ORIGIN(\Pi)$;
- no atom in σ has a strictly positive occurrence in $SAT(\Pi)$.

Thus, according to the splitting theorem, $\phi(I)$ is a stable model of $SAT(\Pi) \cup ORIGIN(\Pi)$ if and only if $\phi(I)$ is a stable model of $SAT(\Pi)$ w.r.t. σ_{sat} and is a stable model of $ORIGIN(\Pi)$ w.r.t. σ.

First, assuming that I belongs to SM[Π], we will prove that $\phi(I)$ is a stable model of $SAT(\Pi) \cup ORIGIN(\Pi)$. Let I be a member of SM[Π].

- **$\phi(I)$ is a stable model of $SAT(\Pi)$ w.r.t. σ_{sat}.** By the definition of ϕ, $\texttt{sat}(i, w_i, \mathbf{c}) \in \phi(I)$ if and only if $I \models \mathit{Head}_i(\mathbf{c}) \leftarrow \mathit{Body}_i(\mathbf{c})$, in which case either $I \models \mathit{Head}_i(\mathbf{c})$ or $I \not\models \mathit{Body}_i(\mathbf{c})$. This means

  $$\phi(I) \models SAT(\Pi) \cup \{\texttt{sat}(i, w_i, \mathbf{c}) \rightarrow \mathit{Head}_i(\mathbf{c}) \vee \neg \mathit{Body}_i(\mathbf{c}) \mid w_i : \mathit{Head}_i(\mathbf{c}) \leftarrow \ \mathit{Body}_i(\mathbf{c}) \in Gr(\Pi)\},$$

  which is the completion of $SAT(\Pi)$. It is obvious that $SAT(\Pi)$ is tight on σ_{sat}. So $\phi(I)$ is a stable model of $SAT(\Pi)$ w.r.t. σ_{sat}.
- **$\phi(I)$ is a stable model of $ORIGIN(\Pi)$ w.r.t. σ.** It is clear that $\phi(I)$ satisfies $ORIGIN(\Pi)$. Assume for the sake of contradiction that there is an interpretation $J \subset \phi(I)$ such that J and $\phi(I)$ agree on σ^{sat} and $J \models ORIGIN(\Pi)^{\phi(I)}$. Then

  $$J \models \mathit{Head}_i(\mathbf{c})^{\phi(I)} \leftarrow \mathit{Body}_i(\mathbf{c})^{\phi(I)}, (\texttt{not not sat}(i, w_i, \mathbf{c}))^{\phi(I)}$$

  for every rule

  $$\mathit{Head}_i(\mathbf{c}) \leftarrow \mathit{Body}_i(\mathbf{c}), \texttt{not not sat}(i, w_i, \mathbf{c})$$

  in $ORIGIN(\Pi)$. Since $\phi(I)$ satisfies $SAT(\Pi)$, it follows that for every rule $\mathit{Head}_i(\mathbf{c}) \leftarrow \mathit{Body}_i(\mathbf{c})$ satisfied by $\phi(I)$, we have $(\texttt{not not sat}(i, w_i, \mathbf{c}))^{\phi(I)} = \top$ so that $J \models \mathit{Head}_i(\mathbf{c})^{\phi(I)} \leftarrow \mathit{Body}_i(\mathbf{c})^{\phi(I)}$, or equivalently, $J \models \mathit{Head}_i(\mathbf{c})^{I} \leftarrow \mathit{Body}_i(\mathbf{c})^{I}$, which contradicts that I is a stable model of $\overline{\Pi_I}$.

Consequently, by the splitting theorem, $\phi(I)$ is a stable model of $SAT(\Pi) \cup ORIGIN(\Pi)$.

Next, assuming $\phi(I)$ is a stable model of $SAT(\Pi) \cup ORIGIN(\Pi)$, we will prove that I belongs to SM[Π].

Let $\phi(I)$ be a stable model of $SAT(\Pi) \cup ORIGIN(\Pi)$. By the splitting theorem, $\phi(I)$ is a stable model of $SAT(\Pi)$ w.r.t. σ_{sat} and $\phi(I)$ is a stable model of $ORIGIN(\Pi)$ w.r.t. σ.

It is clear that $I \models \overline{\Pi_I}$.

Assume for the sake of contradiction that there is an interpretation $J \subset I$ such that $J \models (\overline{\Pi_I})^I$. Take any rule

$$(\mathit{Head}_i(\mathbf{c}))^{\phi(I)} \leftarrow (\mathit{Body}_i(\mathbf{c}))^{\phi(I)}, (\texttt{not not sat}(i, w_i, \mathbf{c}))^{\phi(I)} \qquad \text{(C1)}$$

in $(ORIGIN(\Pi))^{\phi(I)}$.

Case 1: $\phi(I) \not\models \texttt{sat}(i, w_i, \mathbf{c})$. Clearly, $J \models$ (C1).

Case 2: $\phi(I) \models \mathtt{sat}(i, w_i, \mathbf{c})$. Since $Head_i(\mathbf{c})$ and $Body_i(\mathbf{c})$ do not contain sat predicates, (C1) is equivalent to

$$(Head_i(\mathbf{c}))^I \leftarrow (Body_i(\mathbf{c}))^I. \tag{C2}$$

Since $\phi(I)$ is a stable model of $SAT(\Pi)$ w.r.t. σ_{sat}, we have $\phi(I) \models Head_i(\mathbf{c}) \leftarrow Body_i(\mathbf{c})$, or equivalently, $I \models Head_i(\mathbf{c}) \leftarrow Body_i(\mathbf{c})$. So, $Head_i(\mathbf{c}) \leftarrow Body_i(\mathbf{c}) \in \overline{\Pi_I}$, and $Head_i(\mathbf{c})^I \leftarrow Body_i(\mathbf{c})^I \in (\overline{\Pi_I})^I$. Since $J \models (\overline{\Pi_I})^I$, it follows that $J \models$ (C1) as well.

Since $J \subset \phi(I)$, $\phi(I)$ is not a stable model of $ORIGIN(\Pi)$ w.r.t. σ, which contradicts the assumption that it is. Thus we conclude that I is a stable model of $\overline{\Pi_I}$, i.e., I belongs to $\mathrm{SM}[\Pi]$.
□

Theorem 2 For any $\mathrm{LP^{MLN}}$ program Π, there is a 1-1 correspondence ϕ between $\mathrm{SM}[\Pi]$ [6] and the set of stable models of $\mathsf{lpmln2asp^{rwd}}(\Pi)$, where

$$\phi(I) = I \cup \{\mathtt{sat}(i, w_i, \mathbf{c}) \mid w_i : Head_i(\mathbf{c}) \leftarrow Body_i(\mathbf{c}) \text{ in } Gr(\Pi), I \models Body_i(\mathbf{c}) \rightarrow Head_i(\mathbf{c})\}.$$

Furthermore,

$$W_\Pi(I) = exp\Bigg(\sum_{\mathtt{sat}(i,w_i,\mathbf{c}) \in \phi(I)} w_i\Bigg).$$

Also, ϕ is a 1-1 correspondence between the most probable stable models of Π and the optimal stable models of $\mathsf{lpmln2asp^{rwd}}(\Pi)$.

Proof. By Lemma 1, ϕ is a 1-1 correspondence between $\mathrm{SM}[\Pi]$ and the set of stable models of $\mathsf{lpmln2asp}^{rwd}(\Pi)$.

The fact

$$W_\Pi(I) = exp\Bigg(\sum_{\mathtt{sat}(i,w_i,\mathbf{c}) \in \phi(I)} w_i\Bigg) \tag{C3}$$

can be easily seen from the way $\phi(I)$ is defined.

It remains to show that ϕ is a 1-1 correspondence between the most probable stable models of Π and the optimal stable models of $\mathsf{lpmln2asp^{rwd}}(\Pi)$. For any interpretation I of $\mathsf{lpmln2asp^{rwd}}(\Pi)$, we use $Penalty_\Pi(I, l)$ to denote the total penalty it receives at level l defined by weak constraints:

$$Penalty_\Pi(I, l) = \sum_{\substack{:\sim \mathtt{sat}(i,w_i,\mathbf{c}).[-w_i'@l,i,\mathbf{c}] \in WC(\Pi),\\ I \models \mathtt{sat}(i,w_i,\mathbf{c})}} -w_i$$

Let $\phi(I)$ be a stable model of $\mathsf{lpmln2asp^{rwd}}(\Pi)$. By Lemma 1, $I \in \mathrm{SM}[\Pi]$. So it is sufficient to prove

$$I \in \underset{J : J \in \underset{K : K \in \mathrm{SM}[\Pi]}{\mathrm{argmax}} W_{\Pi^{\mathrm{hard}}}(K)}{\mathrm{argmax}} W_{\Pi^{\mathrm{soft}}}(J)$$

iff (C4)

$$\phi(I) \in \underset{J' : J' \in \underset{K' : K' \text{ is a stable model of } \mathsf{lpmln2asp^{rwd}}(\Pi)}{\mathrm{argmin}} Penalty_{\mathsf{lpmln2asp^{rwd}}(\Pi)}(K', 1)}{\mathrm{argmin}} Penalty_{\mathsf{lpmln2asp^{rwd}}(\Pi)}(J', 0).$$

[6] Recall the definition in Section 2.1.

This is true because (we abbreviate $Head_i(\mathbf{c}) \leftarrow Body_i(\mathbf{c})$ as $F_i(\mathbf{c})$)

$$I \in \underset{J:\ J \in \underset{K:\ K \in \mathrm{SM}[\Pi]}{\mathrm{argmax}} W_{\Pi^{\mathrm{hard}}}(K)}{\mathrm{argmax}} W_{\Pi^{\mathrm{soft}}}(J)$$

iff

$$I \in \underset{J:\ J \in \underset{K:\ K \in \mathrm{SM}[\Pi]}{\mathrm{argmax}} exp\big(\sum_{\alpha:F_i(\mathbf{c})\ \in\ (\Pi^{\mathrm{hard}})_K} \alpha\big)}{\mathrm{argmax}} exp\Big(\sum_{w_i:F_i(\mathbf{c})\ \in\ (\Pi^{\mathrm{soft}})_J} w_i\Big)$$

iff

$$I \in \underset{J:\ J \in \underset{K:\ K \in \mathrm{SM}[\Pi]}{\mathrm{argmax}} exp\big(\sum_{\alpha:F_i(\mathbf{c})\ \in\ \Pi^{\mathrm{hard}}, K \models F_i(\mathbf{c})} 1\big)}{\mathrm{argmax}} exp\Big(\sum_{w_i:F_i(\mathbf{c})\ \in\ \Pi^{\mathrm{soft}}, J \models F_i(\mathbf{c})} w_i\Big)$$

iff

$$I \in \underset{J:\ J \in \underset{K:\ K \in \mathrm{SM}[\Pi]}{\mathrm{argmin}} \big(\sum_{\alpha:F_i(\mathbf{c})\ \in\ \Pi^{\mathrm{hard}}, K \models F_i(\mathbf{c})} -1\big)}{\mathrm{argmin}} \Big(\sum_{w_i:F_i(\mathbf{c})\ \in\ \Pi^{\mathrm{soft}}, J \models F_i(\mathbf{c})} -w_i\Big)$$

iff (by Lemma 1 and by definition of $\phi(I)$)

$$\phi(I) \in \underset{J':\ J' \in \underset{K':\ K' \text{ is a stable model of } \mathsf{lpmln2asp}^{\mathrm{rwd}}(\Pi)}{\mathrm{argmin}} \Big(\sum_{\substack{:\sim \mathsf{sat}(i,w_i,\mathbf{c}).[-1@1,i,\mathbf{c}] \\ \in\ \mathsf{lpmln2asp}^{\mathrm{rwd}}(\Pi), \\ K' \models \mathsf{sat}(i,w_i,\mathbf{c})}} -1\Big)}{\mathrm{argmin}} \Big(\sum_{\substack{:\sim \mathsf{sat}(i,w_i,\mathbf{c}).[-w_i@0,i,\mathbf{c}] \\ \in\ \mathsf{lpmln2asp}^{\mathrm{rwd}}(\Pi), \\ J' \models \mathsf{sat}(i,w_i,\mathbf{c})}} -w_i\Big)$$

iff

$$\phi(I) \in \underset{J':\ J' \in \underset{K':\ K' \text{ is a stable model of } \mathsf{lpmln2asp}^{\mathrm{rwd}}(\Pi)}{\mathrm{argmin}} Penalty_{\mathsf{lpmln2asp}^{\mathrm{rwd}}(\Pi)}(K',1)}{\mathrm{argmin}} Penalty_{\mathsf{lpmln2asp}^{\mathrm{rwd}}(\Pi)}(J',0).$$

□

Appendix D Proof of Proposition 1

For any MLN $\mathbb{L}$ and any interpretation I, we define

$$W'_{\mathbb{L}}(I) = \begin{cases} exp(\sum_{w:F \in \mathbb{L}^{soft}, I \models F)} w) & \text{if } I \models \overline{\mathbb{L}^{hard}} \\ 0 & \text{otherwise.} \end{cases}$$

Lemma 2

For any MLN $\mathbb{L}$ such that $\overline{\mathbb{L}^{hard}}$ has at least one model, we have

$$P_{\mathbb{L}}(I) = \frac{W'_{\mathbb{L}}(I)}{\sum_J W'_{\mathbb{L}}(J)}$$

for any interpretation I.

Proof. **Case 1**: Suppose $I \models \overline{\mathbb{L}^{hard}}$.

$$
\begin{aligned}
P_{\mathbb{L}}(I) &= \lim_{\alpha\to\infty} \frac{W_{\mathbb{L}}(I)}{\sum_J W_{\mathbb{L}}(J)} \\
&= \lim_{\alpha\to\infty} \frac{exp(\sum_{w:F\in\mathbb{L}, I\models F} w)}{\sum_J exp(\sum_{w:F\in\mathbb{L}, J\models F} w)} \\
&= \lim_{\alpha\to\infty} \frac{exp(|\overline{\mathbb{L}^{hard}}|\alpha)\cdot exp(\sum_{w:F\in\mathbb{L}\setminus\mathbb{L}^{hard}, I\models F} w)}{\sum_{J\models\overline{\mathbb{L}^{hard}}} exp(|\overline{\mathbb{L}^{hard}}|\alpha)\cdot exp(\sum_{w:F\in\mathbb{L}\setminus\mathbb{L}^{hard}, I\models F} w) + \sum_{J\not\models\overline{\mathbb{L}^{hard}}} exp(\sum_{w:F\in\mathbb{L}, I\models F} w)} \\
&= \lim_{\alpha\to\infty} \frac{exp(\sum_{w:F\in\mathbb{L}\setminus\mathbb{L}^{hard}, I\models F} w)}{\sum_{J\models\overline{\mathbb{L}^{hard}}} exp(\sum_{w:F\in\mathbb{L}\setminus\mathbb{L}^{hard}, I\models F} w) + \frac{1}{exp(|\overline{\mathbb{L}^{hard}}|\alpha)}\sum_{J\not\models\overline{\mathbb{L}^{hard}}} exp(\sum_{w:F\in\mathbb{L}, I\models F} w)}.
\end{aligned}
$$

Since there is at least one hard formula in $\overline{\mathbb{L}^{hard}}$ not satisfied by those J that do not satisfy $\overline{\mathbb{L}^{hard}}$, we have

$$
\begin{aligned}
&\frac{1}{exp(|\overline{\mathbb{L}^{hard}}|\alpha)} \sum_{J\not\models\overline{\mathbb{L}^{hard}}} exp(\sum_{w:F\in\mathbb{L}, I\models F} w) \\
\leq &\frac{1}{exp(|\overline{\mathbb{L}^{hard}}|\alpha)} \sum_{J\not\models\overline{\mathbb{L}^{hard}}} exp((|\overline{\mathbb{L}^{hard}}|-1)\alpha + \sum_{w:F\in\mathbb{L}\setminus\mathbb{L}^{hard}, I\models F} w) \\
= &\frac{1}{exp(\alpha)} \sum_{J\not\models\overline{\mathbb{L}^{hard}}} exp(\sum_{w:F\in\mathbb{L}\setminus\mathbb{L}^{hard}, I\models F} w).
\end{aligned}
$$

This, along with the fact that $\sum_{J\not\models\overline{\mathbb{L}^{hard}}} exp(\sum_{w:F\in\mathbb{L}\setminus\mathbb{L}^{hard}, I\models F} w)$ does not contain α, we have

$$
\begin{aligned}
P_{\mathbb{L}}(I) &= \lim_{\alpha\to\infty} \frac{exp(\sum_{w:F\in\mathbb{L}\setminus\mathbb{L}^{hard}, I\models F} w)}{\sum_{J\models\overline{\mathbb{L}^{hard}}} exp(\sum_{w:F\in\mathbb{L}\setminus\mathbb{L}^{hard}, I\models F} w) + \frac{1}{exp(|\overline{\mathbb{L}^{hard}}|\alpha)}\sum_{J\not\models\overline{\mathbb{L}^{hard}}} exp(\sum_{w:F\in\mathbb{L}, I\models F} w)} \\
&= \frac{exp(\sum_{w:F\in\mathbb{L}\setminus\mathbb{L}^{hard}, I\models F} w)}{\sum_{J\models\overline{\mathbb{L}^{hard}}} exp(\sum_{w:F\in\mathbb{L}\setminus\mathbb{L}^{hard}, I\models F} w)} \\
&= \frac{exp(\sum_{w:F\in\mathbb{L}^{soft}, I\models F} w)}{\sum_{J\models\overline{\mathbb{L}^{hard}}} exp(\sum_{w:F\in\mathbb{L}^{soft}, I\models F} w)} \\
&= \frac{W'_{\mathbb{L}}(I)}{\sum_J W'_{\mathbb{L}}(J)}.
\end{aligned}
$$

Case 2: Suppose I does not satisfy $\overline{\mathbb{L}^{hard}}$. Let K be an interpretation that satisfies $\overline{\mathbb{L}^{hard}}$. We have

$$
\begin{aligned}
P_{\mathbb{L}}(I) &= \lim_{\alpha\to\infty} \frac{W_{\mathbb{L}}(I)}{\sum_J W_{\mathbb{L}}(J)} \\
&= \lim_{\alpha\to\infty} \frac{exp(\sum_{w:F\in\mathbb{L}, I\models F} w)}{\sum_J exp(\sum_{w:F\in\mathbb{L}, J\models F} w)} \\
&\leq \lim_{\alpha\to\infty} \frac{exp(\sum_{w:F\in\mathbb{L}, I\models F} w)}{exp(\sum_{w:F\in\mathbb{L}, K\models F} w)} \\
&= \lim_{\alpha\to\infty} \frac{exp(\sum_{w:F\in\mathbb{L}, I\models F} w)}{exp(|\overline{\mathbb{L}^{hard}}|\alpha)\cdot exp(\sum_{w:F\in\mathbb{L}^{soft}, K\models F} w)}.
\end{aligned}
$$

Since I does not satisfy $\overline{\mathbb{L}^{hard}}$, I satisfies at most $\overline{\mathbb{L}^{hard}} - 1$ hard formulas in $\overline{\mathbb{L}^{hard}}$. So we have

$$\begin{aligned}
P_{\mathbb{L}}(I) &\leq \lim_{\alpha\to\infty} \frac{exp(\sum_{w:F\in\mathbb{L},I\models F} w)}{exp(|\overline{\mathbb{L}^{hard}}|\alpha)\cdot exp(\sum_{w:F\in\mathbb{L}^{soft},K\models F} w)} \\
&\leq \lim_{\alpha\to\infty} \frac{(exp(|\overline{\mathbb{L}^{hard}}|-1)\alpha)\cdot exp(\sum_{w:F\in\mathbb{L}^{soft},I\models F} w)}{(exp(|\overline{\mathbb{L}^{hard}}|\alpha)\cdot exp(\sum_{w:F\in\mathbb{L}^{soft},K\models F} w)} \\
&= \lim_{\alpha\to\infty} \frac{exp(\sum_{w:F\in\mathbb{L}^{soft},I\models F} w)}{exp(\alpha)\cdot exp(\sum_{w:F\in\mathbb{L}^{soft},K\models F} w)} \\
&= 0.
\end{aligned}$$

So $P_{\mathbb{L}}(I) = 0$, which is equivalent to $\frac{W'_{\mathbb{L}}(I)}{\sum_J W'_{\mathbb{L}}(J)}$ as $W'_{\mathbb{L}}(I) = 0$. □

Proposition 1 For any MLN $\mathbb{L}$ of signature σ, let $F(\mathbf{x})$ be a subformula of some formula in $\mathbb{L}$ where $\mathbf{x}$ is the list of all free variables of $F(\mathbf{x})$, and let $\mathbb{L}^F_{Aux}$ be the MLN program obtained from $\mathbb{L}$ by replacing $F(\mathbf{x})$ with a new predicate $Aux(\mathbf{x})$ and adding the formula

$$\alpha \ : \ Aux(\mathbf{x}) \leftrightarrow F(\mathbf{x}).$$

For any interpretation I of $\mathbb{L}$, let I_{Aux} be the extension of I of signature $\sigma \cup \{Aux\}$ defined by $I_{Aux}(Aux(\mathbf{c})) = (F(\mathbf{c}))^I$ for every list $\mathbf{c}$ of elements in the Herbrand universe. When $\overline{\mathbb{L}^{hard}}$ has at least one model, we have

$$P_{\mathbb{L}}(I) = P_{\mathbb{L}^F_{Aux}}(I_{Aux}).$$

Proof. For any formula G, let G^F_{Aux} be the formulas obtained from G by replacing subformulas $F(\mathbf{x})$ with $Aux_F(\mathbf{x})$. According to Lemma 2, we have

$$P_{\mathbb{L}}(I) = \frac{W'_{\mathbb{L}}(I)}{\sum_J W'_{\mathbb{L}}(J)}.$$

Case 1: Suppose I satisfies $\overline{\mathbb{L}^{hard}}$. Then we have

$$P_{\mathbb{L}}(I) = \frac{exp(\sum_{w:G\in\mathbb{L}^{soft},I\models F} w)}{\sum_{J\models\overline{\mathbb{L}^{hard}}} exp(\sum_{w:G\in\mathbb{L}^{soft},J\models F} w)}.$$

From the way I_{Aux} is defined, we have

$$\begin{aligned}
P_{\mathbb{L}}(I) &= \frac{exp(\sum_{w:G^F_{Aux}\in(\mathbb{L}^F_{Aux})^{soft},I_{Aux}\models G^F_{Aux}} w)}{\sum_{J_{Aux}\models\overline{(\mathbb{L}^F_{Aux})^{hard}}} exp(\sum_{w:G^F_{Aux}\in(\mathbb{L}^F_{Aux})^{soft},J_{Aux}\models G^F_{Aux}} w)} \\
&= P_{\mathbb{L}^F_{Aux}}(I_{Aux}).
\end{aligned}$$

Case 2: Suppose I does not satisfy $G \in \overline{\mathbb{L}^{hard}}$. From the way I_{Aux} is defined, I_{Aux} does not satisfy $G^F_{Aux} \in \overline{(\mathbb{L}^F_{Aux})^{hard}}$. So $W'_{\mathbb{L}}(I) = W'_{\mathbb{L}^F_{Aux}}(I_{Aux}) = 0$ and thus $P_{\mathbb{L}}(I) = P_{\mathbb{L}^F_{Aux}}(I_{Aux}) = 0$.

□

Appendix E More Experiments

E.1 Link Prediction in Biological Networks - Another Comparison with PROBLOG2 *on a Real World Problem*

Public biological databases contain huge amounts of rich data, such as annotated sequences, proteins, genes and gene expressions, gene and protein interactions, scientific articles, and ontologies. Biomine (Eronen and Toivonen 2012) is a system that integrates cross-references from several biological databases into a graph model with multiple types of edges. Edges are weighted based on their type, reliability, and informativeness.

We use graphs extracted from the Biomine network. The graphs are extracted around genes known to be connected to the Alzheimer's disease (HGNC ids 620, 582, 983, and 8744). A typical query on such a database of biological concepts is whether a given gene is connected to a given disease. In a probabilistic graph, the importance of the connection can be measured as the probability that a path exists between the two given nodes, assuming that each edge is true with the specified probability, and that edges are mutually independent (Sevon et al. 2006). Nodes in the graph correspond to different concepts such as gene, protein, domain, phenotype, biological process, tissue, and edges connect related concepts. Such a program can be expressed in the language of PROBLOG2 as

```
p(X,Y) :- drc(X,Y).
p(X,Y) :- drc(X, Z), Z \== Y, p(Z, Y).
```

The LPMLN2ASP encoding for the same problem is

```
p(X,Y) :- drc(X,Y).
p(X,Y) :- drc(X, Z), Z != Y, p(Z, Y).
```

The evidence file contains weighted edges `drc/2` encoded as

```
0.942915444848::drc('hgnc_983','pubmed_11749053').
0.492799999825::drc('pubmed_10075692','hgnc_620').
0.434774330065::drc('hgnc_620','pubmed_10460257').
...
```

The same evidence used for PROBLOG2 is processed to work with the syntax of LPMLN2ASP as

```
0.942915444848 drc('hgnc_983','pubmed_11749053').
0.492799999825 drc('pubmed_10075692','hgnc_620').
0.434774330065 drc('hgnc_620','pubmed_10460257').
...
```

We test the systems on varying graph sizes ranging from 366 nodes, 363 edges to 5646 nodes, 64579 edges. The experiment was run on a 40 core Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz machine with 128 GB of RAM. The timeout for the experiment was set to 20 minutes.

We perform MAP inference for comparison. Table E.1 shows the results of the experiment. Apart from the smaller graph instances where PROBLOG2 is faster than LPMLN2ASP, LPMLN2ASP significantly outperforms PROBLOG2 for medium to large graphs for MAP inference. In fact, for graphs with nodes greater than 1980 PROBLOG2 times out. For Marginal inference, to check for the probability of path between two genes, LPMLN2ASP times out with just 25 nodes and therefore it is infeasible to experiment for marginal probability on LPMLN2ASP . The sampling based approach of PROBLOG2 computes the probability of a path from `'hgnc_983'` to `'hgnc_620'`

Nodes	**Edges**	LPMLN2ASP	PROBLOG2
366	363	0.37	0.152
1677	2086	9.77	1.7406
1982	4143	14	Timeout
2291	6528	19.71	Timeout
2588	9229	25.92	Timeout
2881	12248	33.05	Timeout
3168	15583	42.21	Timeout
3435	19204	49.91	Timeout
3724	23135	59.56	Timeout
3989	27370	69.72	Timeout
4252	31891	82.04	Timeout
4501	36690	93.23	Timeout
4750	41761	105.4	Timeout
4983	47094	116.79	Timeout
5200	52673	129.27	Timeout
5431	58506	142.2	Timeout
5646	64579	157.77	Timeout

Table E 2. PROBLOG2 *vs.* LPMLN2ASP *Comparison on Biomine Network*

in 13 seconds. This experiment goes on to show that for MAP inference, our implementation far outperforms the current implementation of PROBLOG2 while being significantly slower in computing Marginal and Conditional probabilities.

E.2 Social influence of smokers - Computing MLN using LPMLN2ASP

Following Section 3.5, we compare the scalability of LPMLN2ASP for MAP inference on MLN encodings and compare with the MLN solvers ALCHEMY, TUFFY and ROCKIT used in LPMLN2MLN. We scale the example by increasing the number of people and relationships among them.

The LPMLN2ASP encoding of the example used in the experiment is

```
1.1 cancer(X) :- smokes(X).
1.5 smokes(Y) :- smokes(X), influences(X, Y).
{smokes(X)} :- person(X).
{cancer(X)} :- person(X).
```

The ALCHEMY encoding of the example is

```
smokes(node)
influences(node,node)
cancer(node)

1.1 smokes(x) => cancer(x)
1.5 smokes(x) ^ influences(x,y) => smokes(y)
```

and is run with the command line

```
infer -m -i input -e evidence -r output -q cancer -ow smokes,cancer
```

The TUFFY encoding of the example is[7]

[7] * makes the predicate closed world assumption

```
smokes(node)
*influences(node,node)
cancer(node)

1.1 smokes(x) => cancer(x)
1.5 smokes(x) , influences(x,y) => smokes(y)
```

and is run with the command line

```
java -jar tuffy.jar -i input -e evidence -r output -q cancer
```

The ROCKIT encoding of the example is

```
smokes(node)
*influences(node,node)
cancer(node)

1.1 !smokes(x) v cancer(x)
1.5 !smokes(x) v !influences(x,y) v smokes(y)
```

and is run with the command line

```
java -jar rockit.jar -input input -data evidence -output output
```

The data was generated such that for each person p, the person *smokes* with an 80% probability, and p *influences* every other person with a 60% probability. We generate evidence instances based on different number of persons ranging from 10 to 1000. We compare the performance of the solvers based on the time it takes to compute the MAP estimate. The experiment was run on a 40 core Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz machine with 128 GB of RAM. The timeout for the experiment was set to 20 minutes.

# of Persons	LPMLN2ASP **w.** CLINGO **4.5**	ALCHEMY **2.0**	TUFFY **0.3**	ROCKIT **0.5**
10	0	0.04	1.014	0.465
50	0.03	1.35	1.525	0.676
100	0.10	18.87	1.560	0.931
200	0.32	435.71	2.672	1.196
300	0.7	Timeout	4.054	1.660
400	1.070	Timeout	4.505	1.914
500	1.730	Timeout	5.935	2.380
600	2.760	Timeout	7.683	2.822
700	3.560	Timeout	10.390	3.274
800	4.72	Timeout	11.384	3.727
900	Timeout	Timeout	12.056	4.012
1000	Timeout	Timeout	12.958	4.678

Table E 3: Performance of solvers on MLN program

Table E 3 lists the computation time in seconds for each of the four solvers on instances of domains of varying size. LPMLN2ASP is the best performer for the number of people till 600.

ALCHEMY is the worst performer out of all 4 and for instances with number of people greater than 200 it times out. As expected, for ALCHEMY, grounding is the major bottleneck. For the instance with 200 persons, ALCHEMY grounds it in 422.85 seconds and only takes 9 seconds to compute the MAP estimate. TUFFY and ROCKIT have more scalable grounding times. ROCKIT has the best results amongst all the solvers. This experiment shows that for medium sized instances, our implementation is comparable to the fastest available solver for MAP inference on MLN programs.

References

SANG, T., BEAME, P., AND KAUTZ, H. 2005. Solving bayesian networks by weighted model counting. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-05)*. Vol. 1. 475–482.

ERONEN, L. AND TOIVONEN, H. 2012. Biomine: predicting links between biological entities using network models of heterogeneous databases. *BMC bioinformatics 13,* 1, 119.

SEVON, P., ERONEN, L., HINTSANEN, P., KULOVESI, K., AND TOIVONEN, H. 2006. Link discovery in graphs derived from biological databases. In *International Workshop on Data Integration in the Life Sciences*. Springer, 35–49.