On the Relationships Among Probabilistic Extensions of Answer Set Semantics

by

Zhun Yang

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2017 by the
Graduate Supervisory Committee:

Joohyung Lee, Chair
Chitta Baral
Baoxin Li

ARIZONA STATE UNIVERSITY

May 2017

ABSTRACT

Answer Set Programming (ASP) is one of the main formalisms in Knowledge Representation (KR) that is being widely applied in a large number of applications. While ASP is effective on Boolean decision problems, it has difficulty in expressing quantitative uncertainty and probability in a natural way.

Logic Programs under the answer set semantics and Markov Logic Network (LP$^{\text{MLN}}$) is a recent extension of answer set programs to overcome the limitation of the deterministic nature of ASP by adopting the log-linear weight scheme of Markov Logic. This thesis investigates the relationships between LP$^{\text{MLN}}$ and two other extensions of ASP: weak constraints to express a quantitative preference among answer sets, and P-log to incorporate probabilistic uncertainty. The studied relationships show how different extensions of answer set programs are related to each other, and how they are related to formalisms in Statistical Relational Learning, such as Problog and MLN, which have shown to be closely related to LP$^{\text{MLN}}$. The studied relationships compare the properties of the involved languages and provide ways to compute one language using an implementation of another language.

This thesis first presents a translation of LP$^{\text{MLN}}$ into programs with weak constraints. The translation allows for computing the most probable stable models (i.e., MAP estimates) or probability distribution in LP$^{\text{MLN}}$ programs using standard ASP solvers so that the well-developed techniques in ASP can be utilized. This result can be extended to other formalisms, such as Markov Logic, ProbLog, and Pearl's Causal Models, that are shown to be translatable into LP$^{\text{MLN}}$.

This thesis also presents a translation of P-log into LP$^{\text{MLN}}$. The translation tells how probabilistic nonmonotonicity (the ability of the reasoner to change his probabilistic model as a result of new information) of P-log can be represented in LP$^{\text{MLN}}$, which yields a way to compute P-log using standard ASP solvers or MLN solvers.

TABLE OF CONTENTS

LIST OF TABLES

Chapter 1

INTRODUCTION

Knowledge Representation and Reasoning (KRR) is an area in Artificial Intelligence (AI) dedicated to representing information in the real world as well as the ways we reason about this information in a form that a computer can utilize. Answer set programming (ASP) (Gelfond and Lifschitz, 1988) is one of such form that is based on the stable model (answer set) semantics of logic programming, where reasoners draw tentative conclusions and can retract their conclusions as a result of new information. ASP is a declarative paradigm that is simple but expressive, and is widely applied in a large number of applications. A lot of techniques have been developed for ASP solvers to make them highly efficient for deterministic inference.

However, ASP lacks an important construct in knowledge representation — quantitative uncertainty. To overcome the deterministic nature of ASP and express uncertainty, a few extensions of answer set semantics have been proposed.

One simple but practically very useful extension is weak constraints (Buccafurri *et al.*, 2000), which assign a quantitative preference over the stable models of an ASP program. The additional weak constraint rules cannot be used for deriving stable models but only for calculating the penalty (a number to be ranked) of each stable model that violates the weak constraints. Weak constraints enable an answer set program to find its optimal stable models, but they do not have a built-in notion of probability.

In contrast to weak constraints, another probabilistic extension of answer set programs, P-log (Baral *et al.*, 2009), directly assigns a probability to each possible world (namely, stable model in ASP) by clear and transparent modeling of logical and probabilistic knowledge. P-log is highly structured and has quite a sophisticated semantics. One of its distinct features is *probabilistic nonmonotonicity* (the ability of the reasoner to change his probabilistic model as a result of new information) whereas, in most other probabilistic logic languages, new information can only cause the reasoner to abandon some of his possible worlds, making the effect of an update *monotonic*. However, the current P-log implementation (`http://www.depts.ttu.edu/cs/research/krlab/plog.php`) does not have a well developed prototype.

Rather than directly assigning the probabilities to each stable model as P-log does, LP$^{\text{MLN}}$ (Lee and Wang, 2016), a recently introduced probabilistic extension of answer set programs, enables ASP to calculate probabilities by the concept of weighted rules, whose weight scheme is adopted from that of Markov Logic (Richardson and Domingos, 2006). The idea of LP$^{\text{MLN}}$ is to assign a weight to each ASP rule — the more rules a stable model satisfies, the more weights it gets, and the probability of this stable model is calculated by normalizing its weight among all stable models. It is shown in (Lee and Wang, 2016; Lee *et al.*, 2015) that LP$^{\text{MLN}}$ is expressive enough to embed Markov Logic and several other probabilistic logic languages, such as ProbLog (De Raedt *et al.*, 2007), Pearls' Causal Models (Pearl, 2000), and a fragment of P-log (Baral *et al.*, 2009). Although LP$^{\text{MLN}}$ is very expressive, it is difficult for LP$^{\text{MLN}}$ to capture the feature of probabilistic nonmonotonicity in P-log.

Comparing these three languages, weak constraints have a good performance in practice and is easy to debug, but it does not have a built-in notion of probability. P-log is a clear and transparent modeling language that is easy to use and is capable of expressing sophisticated logical and probabilistic knowledge together. If P-log can be reduced to LP$^{\text{MLN}}$, and LP$^{\text{MLN}}$ can be reduced to weak constraints, then the sophisticated representation in P-log can be captured by logic programs with weak constraints and solved by ASP solvers. This thesis will show these two translations.

This thesis first investigates a translation from LP$^{\text{MLN}}$ to programs with weak constraints, which complements the inverse translation shown in (Lee and Wang, 2016). The translation is simple so it can be easily applied in practice.

- It shows that an LP$^{\text{MLN}}$ program can be turned into a usual ASP program with weak constraints so that the most probable stable models of the LP$^{\text{MLN}}$ program are exactly the optimal stable models of the program with weak constraints.

- The translation allows us to compute MAP estimates of LP$^{\text{MLN}}$ programs using standard ASP solvers so that the well-developed techniques in ASP can be utilized. This result can be extended to other formalisms, such as Markov Logic, ProbLog, and Pearl's Causal Models, that are shown to be translatable into LP$^{\text{MLN}}$.

Further, this thesis shows how P-log can be completely characterized in LP$^{\text{MLN}}$. Unlike the translation in (Lee and Wang, 2016), which was limited to a subset of P-log that does not allow dynamic default probability, we provide a novel translation, which applies to full P-log and complements the recent translation from LP$^{\text{MLN}}$ into P-log in (Balai and Gelfond, 2016).

- The translation helps us understand how P-log is related to LP$^{\text{MLN}}$, and furthermore how P-log is related to formalisms in Statistical Relational Learning, such as Problog and MLN, which have shown to be closely related to LP$^{\text{MLN}}$.

- In conjunction with the first translation, we show another two means of computing P-log: turning P-log through LP$^{\text{MLN}}$ into weak constraints and computing it using ASP solvers, or turning P-log through LP$^{\text{MLN}}$ into MLN and computing it using MLN solvers.

This thesis is organized as follows. Chapter 2 reviews the syntax and semantics of LP$^{\text{MLN}}$, weak constraints, and P-log. Chapter 3 and Chapter 4 show a translation from an LP$^{\mathrm{MLN}}$ program to an ASP program with weak constraints and a translation from a P-log program to an LP$^{\mathrm{MLN}}$ program. More examples of both translations are listed in Chapter 5. The comparison of run time for a P-log program, its translated LP$^{\mathrm{MLN}}$ program, and its further translated ASP program with weak constraints on various benchmarks can be found in Chapter 6. We comment on our contributions in Chapter 7.

Chapter 2

BACKGROUND

In this chapter, we review the syntax and semantics of three extension of answer set programs: LP$^{\text{MLN}}$, weak constraints, and P-log.

2.1 Review of LP$^{\text{MLN}}$

Syntax of LP$^{\text{MLN}}$

We review the definition of LP$^{\text{MLN}}$ from Lee and Wang (2016). In fact, we consider a more general syntax of programs than the one from Lee and Wang (2016), but this is not an essential extension. We follow the view of Ferraris *et al.* (2011) by identifying logic program rules as a special case of first-order formulas under the stable model semantics. For example, rule

$$r(x) \leftarrow p(x), \textit{not } q(x)$$

is identified with first-order formula

$$\forall x(p(x) \wedge \neg q(x) \rightarrow r(x)).$$

An LP$^{\text{MLN}}$ program is a finite set of weighted first-order formulas

$$w : F$$

where F is a first-order formula, w is

- a real number, in which case the weighted formula is called *soft*, or
- α for denoting the infinite weight, in which case it is called *hard*.

An LP$^{\rm MLN}$ program is called *ground* if its formulas contain no variables. We assume a finite Herbrand Universe. Any LP$^{\rm MLN}$ program can be turned into a ground program by replacing the quantifiers with multiple conjunctions and disjunctions over the Herbrand Universe. Each of the ground instances of a formula with free variables receives the same weight as the original formula.

Semantics of LP$^{\rm MLN}$

For any ground LP$^{\rm MLN}$ program Π and any interpretation I, $\overline{\Pi}$ denotes the unweighted formula obtained from Π, and Π_I denotes the set of $w : F$ in Π such that $I \models F$, and SM[Π] denotes the set $\{I \mid I$ is a stable model of $\overline{\Pi_I}\}$ (We refer the reader to the stable model semantics of first-order formulas in Ferraris *et al.* (2011)). The *unnormalized weight* of an interpretation I under Π is defined as

$$W_\Pi(I) = \begin{cases} exp\left(\sum\limits_{w:F\,\in\,\Pi_I} w\right) & \text{if } I \in \mathrm{SM}[\Pi]; \\ 0 & \text{otherwise.} \end{cases}$$

The *normalized weight* (a.k.a. *probability*) of an interpretation I under Π is defined as

$$P_\Pi(I) = \lim_{\alpha\to\infty} \frac{W_\Pi(I)}{\sum\limits_{J\in \mathrm{SM}[\Pi]} W_\Pi(J)}.$$

I is called a *(probabilistic) stable model* of Π if $P_\Pi(I) \neq 0$.

For any first-order formula F, the probability of F under Π is defined as

$$P_\Pi(F) = \sum_{I:I\models F} P_\Pi(I).$$

We say F is satisfiable in Π if $P_\Pi(F) \neq 0$.

Conditional probability under Π is defined as usual. For first-order formulas F_1 and F_2, if F_2 is satisfiable in Π, the probability of F_1 given F_2 under Π is defined as

$$P_\Pi(F_1 \mid F_2) = \frac{P_\Pi(F_1 \wedge F_2)}{P_\Pi(F_2)}.$$

Example 1 *Consider the* LP$^{\text{MLN}}$ *program* Π*:*

$$\begin{array}{rcll} 10 & : & p \rightarrow q & (r_1) \\ 1 & : & p \rightarrow r & (r_2) \\ 5 & : & p & (r_3) \\ -20 & : & \neg r \rightarrow \bot & (r_4) \end{array}$$

There are 8 interpretations of Π*. For each interpretation* I*, we list the rules that are satisfied by* I *(*Π_I*), weight of* I *under* Π *(*$W_\Pi(I)$*), and probability of* I *under* Π *(*$P_\Pi(I)$*) in the following table.*

I	Π_I	$W_\Pi(I)$	$P_\Pi(I)$
$\emptyset$	$\{r_1, r_2\}$	e^{10+1}	$\frac{e^{11}}{e^{11}+e^{5}+e^{15}+e^{-14}+e^{-4}}$
$\{p\}$	$\{r_3\}$	e^5	$\frac{e^{5}}{e^{11}+e^{5}+e^{15}+e^{-14}+e^{-4}}$
$\{q\}$	$\{r_1, r_2\}$	0	0
$\{r\}$	$\{r_1, r_2, r_4\}$	0	0
$\{p, q\}$	$\{r_1, r_3\}$	e^{10+5}	$\frac{e^{15}}{e^{11}+e^{5}+e^{15}+e^{-14}+e^{-4}}$
$\{p, r\}$	$\{r_2, r_3, r_4\}$	e^{5+1-20}	$\frac{e^{-14}}{e^{11}+e^{5}+e^{15}+e^{-14}+e^{-4}}$
$\{q, r\}$	$\{r_1, r_2, r_4\}$	0	0
$\{p, q, r\}$	$\{r_1, r_2, r_3, r_4\}$	$e^{10+5+1-20}$	$\frac{e^{-4}}{e^{11}+e^{5}+e^{15}+e^{-14}+e^{-4}}$

By definition, SM[Π] *has* 5 *elements:*

$$\emptyset, \{p\}, \{p, q\}, \{p, r\}, \{p, q, r\}.$$

Since their probabilities are all greater than 0, all of them are probabilistic stable models of Π*, among which* $\{p, q\}$ *is the most probable stable model, whose weight is* e^{15}*, while* $\{p, q, r\}$ *is a probabilistic stable model whose weight is* e^{-4}*.*

Further, if we want to calculate the probability of q *given* r*, we have*

$$P_\Pi(q \mid r) = \frac{P_\Pi(q \wedge r)}{P_\Pi(r)} = \frac{P_\Pi(\{q, r\}) + P_\Pi(\{p, q, r\})}{P_\Pi(\{r\}) + P_\Pi(\{p, r\}) + P_\Pi(\{q, r\}) + P_\Pi(\{p, q, r\})} = \frac{e^{-4}}{e^{-14} + e^{-4}}.$$

The following example illustrates an LP$^{\text{MLN}}$ program Π where some of its weighted rules are hard rules (rules with infinite weight α). We will see some interpretations of Π belong to SM[Π], but are not probabilistic stable models of Π because of the utilization of hard rules.

Example 2 *Consider the* LP$^{\text{MLN}}$ *program* Π *in Example 1 from Lee and Wang (2016).*

$$
\begin{array}{lll}
\alpha : & \mathit{Bird}(\mathit{Jo}) \leftarrow \mathit{ResidentBird}(\mathit{Jo}) & (r_1)\\
\alpha : & \mathit{Bird}(\mathit{Jo}) \leftarrow \mathit{MigratoryBird}(\mathit{Jo}) & (r_2)\\
\alpha : & \bot \leftarrow \mathit{ResidentBird}(\mathit{Jo}), \mathit{MigratoryBird}(\mathit{Jo}) & (r_3)\\
2 : & \mathit{ResidentBird}(\mathit{Jo}) & (r_4)\\
1 : & \mathit{MigratoryBird}(\mathit{Jo}) & (r_5)
\end{array}
$$

The following table shows the satisfied rules, weight, and probability of each interpretation of Π.

I	Π_I	$W_\Pi(I)$	$P_\Pi(I)$
$\emptyset$	$\{r_1, r_2, r_3\}$	$e^{3\alpha}$	$\frac{e^0}{e^2+e^1+e^0}$
$\{R(Jo)\}$	$\{r_2, r_3, r_4\}$	$e^{2\alpha+2}$	0
$\{M(Jo)\}$	$\{r_1, r_3, r_5\}$	$e^{2\alpha+1}$	0
$\{B(Jo)\}$	$\{r_1, r_2, r_3\}$	0	0
$\{R(Jo), B(Jo)\}$	$\{r_1, r_2, r_3, r_4\}$	$e^{3\alpha+2}$	$\frac{e^2}{e^2+e^1+e^0}$
$\{M(Jo), B(Jo)\}$	$\{r_1, r_2, r_3, r_5\}$	$e^{3\alpha+1}$	$\frac{e^1}{e^2+e^1+e^0}$
$\{R(Jo), M(Jo)\}$	$\{r_4, r_5\}$	e^3	0
$\{R(Jo), M(Jo), B(Jo)\}$	$\{r_1, r_2, r_4, r_5\}$	$e^{2\alpha+3}$	0

By definition, SM[Π] *has 7 elements:* $\emptyset$, $\{R(Jo)\}$, $\{M(Jo)\}$, $\{R(Jo), B(Jo)\}$, $\{M(Jo), B(Jo)\}$, $\{R(Jo), M(Jo)\}$, *and* $\{R(Jo), M(Jo), B(Jo)\}$. *Since an interpretation* I *is a probabilistic stable model of* Π *iff* $P_\Pi(I) \neq 0$, *only three elements in*

$\mathrm{SM}[\Pi]$,

$$\emptyset, \{ResidentBird(Jo), Bird(Jo)\},\ \ and\ \{MigratoryBird(Jo), Bird(Jo)\},$$

are probabilistic stable models of Π. *Among them,*

$$\{ResidentBird(Jo), Bird(Jo)\}$$

has the highest weight (and probability), and thus is the most probable stable model of Π.

As we can see from the table, an element I *in* $\mathrm{SM}[\Pi]$ *is a probabilistic stable model of* Π *only when* I *satisfies the maximal number of hard rules in* Π, *i.e., there is no interpretation* J *of* Π *such that* $J \in \mathrm{SM}[\Pi]$ *and* J *satisfies more hard rules than* I. *Thus* $\{R(Jo), M(Jo), B(Jo)\}$ *is not a probabilistic stable model of* Π *even though it satisfies four rules in* Π.

2.2 Review of Weak Constraints

Syntax of Weak Constraints

A *weak constraint* has the form

$$:\sim F \quad [\mathit{Weight} \ @ \ \mathit{Level}].$$

where F is a ground formula, *Weight* is a real number and *Level* is a nonnegative integer. Note that the syntax is more general than the one from the literature Buccafurri *et al.* (2000); Calimeri *et al.* (2013), where F was restricted to conjunctions of literals. [1] We will see the generalization is more convenient for stating our result, but will also present translations that conform to the restrictions imposed on the input language of ASP solvers.

Semantics of Weak Constraints

Let Π be a program $\Pi_1 \cup \Pi_2$, where Π_1 is a set of ground formulas and Π_2 is a set of weak constraints. We call I a stable model of Π if it is a stable model of Π_1 (in the sense of Ferraris *et al.* (2011)). For every stable model I of Π and any nonnegative integer l, the *penalty* of I at level l, denoted by $\mathit{Penalty}_\Pi(I, l)$, is defined as

$$\sum_{\substack{:\sim \ F[w@l] \in \Pi_2, \\ I \models F}} w.$$

For any two stable models I and I' of Π, we say I is *dominated* by I' if

- there is some nonnegative integer l such that $\mathit{Penalty}_\Pi(I', l) < \mathit{Penalty}_\Pi(I, l)$ and
- for all integers $k > l$, $\mathit{Penalty}_\Pi(I', k) = \mathit{Penalty}_\Pi(I, k)$.

[1] A literal is either an atom p or its negation *not* p.

A stable model of Π is called *optimal* if it is not dominated by another stable model of Π.

Example 3 *Consider the ASP program with weak constraints Π in Example 3.16 from (Gebser* et al.*, 2015). We want to choose one hotel from five candidates which are identified by numbers assigned in descending order of stars. For each of the hotels, we know its cost. We know hotel 4 is located on a main street thus we expect its rooms to be noisy. (These information are represented by rules 1-7 in the following program Π)*

$$
\begin{array}{lr}
\{hotel(1..5)\} = 1 & (r_1)\\
star(1,5).\ \ cost(1,170). & (r_2)\\
star(2,4).\ \ cost(2,140). & (r_3)\\
star(3,3).\ \ cost(3,90). & (r_4)\\
star(4,3).\ \ cost(4,75).\ \ main\_street(4) & (r_5)\\
star(5,2).\ \ cost(5,60). & (r_6)\\
noisy \leftarrow hotel(X), main\_street(X) & (r_7)\\
\#maximize\{Y@1, X : hotel(X), star(X,Y)\}. & (r_8)\\
\#minimize\{Y/Z@2, X : hotel(X), cost(X,Y), star(X,Z)\}. & (r_9)\\
:\sim noisy. [1@3] & (r_{10})
\end{array}
$$

Rule 10 says that at level 3 (highest priority), we want to avoid noise. Rule 9 says that at level 2 (medium priority), we want to minimize the cost per star. And rule 8 says that at level 1 (lowest priority), we want to maximize the number of stars among hotels that are otherwise indistinguishable.

The stable models of Π are stable models of rules 1-7:

$$\{hotel(1), ...\}, \{hotel(2), ...\}, \{hotel(3), ...\}, \{hotel(4), noisy, ...\}, \{hotel(5), ...\}$$

where the omitted atoms are atoms in rules 2-6. Among these stable models, the *optimal stable model is*

$$\{hotel(3), ...\}$$

because at level 3, it does not satisfy noisy; at level 2, hotel 3 and hotel 5 have lowest value (of Y/Z) 30; and at level 1, hotel 3 has one more star than hotel 5.

2.3 Review of P-log

Syntax of P-log

A *sort* is a set of symbols. A *constant* c maps an element in the *domain* $s_1 \times \cdots \times s_n$ to an element in the *range* s_0 (denoted by $Range(c)$), where each of $s_0, \dots, s_n$ is a sort. A *sorted propositional signature* is a special case of propositional signatures constructed from a set of constants and their associated sorts, consisting of all propositional atoms $c(\vec{u})\!=\!v$ where $c : s_1 \times \cdots \times s_n \to s_0$, and $\vec{u} \in s_1 \times \cdots \times s_n$, and $v \in s_0$. [2] Symbol $c(\vec{u})$ is called an *attribute* and v is called its *value*. If the range s_0 of c is $\{\mathbf{f}, \mathbf{t}\}$ then c is called *Boolean*, and $c(\vec{u})\!=\!\mathbf{t}$ can be abbreviated as $c(\vec{u})$ and $c(\vec{u})\!=\!\mathbf{f}$ as $\sim\!c(\vec{u})$.

The signature of a P-log program is the union of two propositional signatures σ_1 and σ_2, where σ_1 is a sorted propositional signature, and σ_2 is a usual propositional signature consisting of atoms $Do(c(\vec{u}) = v)$, $Obs(c(\vec{u}) = v)$ and $Obs(c(\vec{u}) \neq v)$ for all atoms $c(\vec{u})\!=\!v$ in σ_1.

A P-log program Π of signature $\sigma_1 \cup \sigma_2$ is a tuple

$$\Pi = \langle \mathbf{R}, \mathbf{S}, \mathbf{P}, \mathbf{Obs}, \mathbf{Act} \rangle \tag{2.1}$$

where the signature of each of $\mathbf{R}$, $\mathbf{S}$, and $\mathbf{P}$ is σ_1 and the signature of each of $\mathbf{Obs}$ and $\mathbf{Act}$ is σ_2 such that

- $\mathbf{R}$ is a set of *normal rules* of the form

  $$A \leftarrow B_1, \dots, B_m, not\ B_{m+1}, \dots, not\ B_n$$

  where $A, B_1, \dots, B_n$ are atoms in σ_1 $(0 \le m \le n)$.

- $\mathbf{S}$ is a set of *random selection rules* of the form

  $$[r]\ \ random(c(\vec{u}) : \{x : p(x)\}) \leftarrow Body \tag{2.2}$$

[2]Note that here "=" is just a part of the symbol for propositional atoms, and is not equality in first-order logic.

where r is a unique identifier, p is a boolean constant with a unary argument, and *Body* is a set of literals. x is a schematic variable ranging over the argument sort of p. Rule (2.2) is called a *random selection* rule for $c(\vec{u})$. Intuitively, rule (2.2) says that if *Body* is true, the value of $c(\vec{u})$ is selected at random from the set $\mathit{Range}(c) \cap \{x : p(x)\}$ unless this value is fixed by a deliberate action, i.e., $\mathit{Do}(c(\vec{u})=v)$ for some value v.

- **P** is a set of so-called *probability atoms (pr-atoms)* of the form

$$pr_r(c(\vec{u})=v \mid C) = p \tag{2.3}$$

where r is the identifier of some random selection rule for $c(\vec{u})$ in **S**; $c(\vec{u})=v \in \sigma_1$; C is a set of literals; and p is a real number in $[0,1]$. We say pr-atom (2.3) is *associated* with the random selection rule whose identifier is r.

- **Obs** is a set of atomic facts of the form

$$\mathit{Obs}(c(\vec{u})=v)$$
$$\mathit{Obs}(c(\vec{u}) \neq v)$$

where $c(\vec{u})$ is an attribute occurring in **S**, $v \in \mathit{Range}(c)$. $\mathit{Obs}(c(\vec{u}) = v)$ represents an observation of $c(\vec{u}) = v$ to be true, and $\mathit{Obs}(c(\vec{u}) \neq v)$ represents an observation of $c(\vec{u}) \neq v$ to be true.

- **Act** is a set of atomic facts of the form

$$\mathit{Do}(c(\vec{u})=v)$$

where $c(\vec{u})$ is an attribute occurring in **S**, $v \in \mathit{Range}(c)$. $\mathit{Do}(c(\vec{u})=v)$ represents a deliberate action to make $c(\vec{u})=v$ to be true.

Semantics of P-log

Let Π be a P-log program (2.1) of signature $\sigma_1 \cup \sigma_2$. The *possible worlds* of Π, denoted by $\omega(\Pi)$, are the stable models of $\tau(\Pi)$, a (standard) ASP program with the propositional signature

$$\sigma_1 \cup \sigma_2 \cup \{\mathit{Intervene}(c(\vec{u})) \mid c(\vec{u}) \text{ is an attribute occurring in } \mathbf{S}\}$$

that accounts for the logical part of P-log and is constructed as follows:

Definition of $\tau(\Pi)$

- $\tau(\Pi)$ contains all rules in $\mathbf{R}$.
- For each attribute $c(\vec{u})$ in σ_1, for $v_1, v_2 \in \mathit{Range}(c)$, $\tau(\Pi)$ contains the following rule:

$$\leftarrow c(\vec{u}) = v_1, c(\vec{u}) = v_2, v_1 \neq v_2$$

- For each random selection rule (2.2) in $\mathbf{S}$ with $\mathit{Range}(c) = \{v_1, \dots, v_n\}$, $\tau(\Pi)$ contains the following rules:

$$c(\vec{u}) = v_1; \dots; c(\vec{u}) = v_n \leftarrow \mathit{Body}, \mathit{not}\ \mathit{Intervene}(c(\vec{u}))$$
$$\leftarrow c(\vec{u}) = v, \mathit{not}\ p(v), \mathit{Body}, \mathit{not}\ \mathit{Intervene}(c(\vec{u}))$$

  where $\mathit{Intervene}(c(\vec{u}))$ means that the randomness of $c(\vec{u})$ is intervened (by an atomic fact $\mathit{Do}(c(\vec{u}) = v)$).

- For each atomic fact $\mathit{Obs}(c(\vec{u}) = v)$ in $\mathbf{Obs}$, $\tau(\Pi)$ contains the following rules:

$$\mathit{Obs}(c(\vec{u}) = v)$$
$$\leftarrow \mathit{Obs}(c(\vec{u}) = v), \mathit{not}\ c(\vec{u}) = v$$

- For each atomic fact $\mathit{Obs}(c(\vec{u}) \neq v)$ in $\mathbf{Obs}$, $\tau(\Pi)$ contains the following rules:

$$\mathit{Obs}(c(\vec{u}) \neq v)$$
$$\leftarrow \mathit{Obs}(c(\vec{u}) \neq v), c(\vec{u}) = v$$

- For each atomic fact $Do(c(\vec{u}) = v)$ in **Act**, $\tau(\Pi)$ contains the following rules:

$$\begin{aligned}
&Do(c(\vec{u}) = v)\\
&c(\vec{u}) = v \leftarrow Do(c(\vec{u}) = v)\\
&Intervene(c(\vec{u})) \leftarrow Do(c(\vec{u}) = v)
\end{aligned}$$

Conditions for Π

As in Baral *et al.* (2009), we assume that all P-log programs Π satisfy the following conditions:

- **Condition 1 [Unique random selection rule]:** If a P-log program Π contains two random selection rules for $c(\vec{u})$:

  $$\begin{aligned}
  &[r_1]\ \ random(c(\vec{u}) : \{x : p_1(x)\}) \leftarrow Body_1,\\
  &[r_2]\ \ random(c(\vec{u}) : \{x : p_2(x)\}) \leftarrow Body_2,
  \end{aligned}$$

  then no possible world of Π satisfies both $Body_1$ and $Body_2$.

- **Condition 2 [Unique probability assignment]:** If a P-log program Π contains a random selection rule for $c(\vec{u})$:

  $$[r]\ \ random(c(\vec{u}) : \{x : p(x)\}) \leftarrow Body$$

  along with two different pr-atoms:

  $$\begin{aligned}
  &pr_r(c(\vec{u}) = v \mid C_1) = p_1,\\
  &pr_r(c(\vec{u}) = v \mid C_2) = p_2,
  \end{aligned}$$

  then no possible world of Π satisfies $Body$, C_1, and C_2 together.

An atom $c(\vec{u})\!=\!v$ is called *possible* in a possible world W due to a random selection rule (2.2) if Π contains (2.2) such that $W \models \mathit{Body} \wedge p(v) \wedge \neg \mathit{Intervene}(c(\vec{u}))$. [3] Pr-atom (2.3) is *applied* in W if $c(\vec{u})\!=\!v$ is possible in W due to r and $W \models C$.

Given a P-log program Π, a possible world $W \in \omega(\Pi)$, and an atom $c(\vec{u}) = v$ possible in W, by **Condition 1**, it follows that there is exactly one random selection rule (2.2) such that $W \models \mathit{Body}$. Let $r_{W,c(\vec{u})}$ denote this random selection rule, and let $\mathit{AV}_W(c(\vec{u})) = \{v' \mid$ there exists a pr-atom $pr_{r_{W,c(\vec{u})}}(c(\vec{u}) = v' \mid C) = p$ that is applied in W for some C and $p\}$. We then define the following notations:

- If $v \in \mathit{AV}_W(c(\vec{u}))$, there exists a pr-atom $pr_{r_{W,c(\vec{u})}}(c(\vec{u})\!=\!v \mid C) = p$ in Π for some C and p such that $W \models C$. By **Condition 2**, for any other $pr_{r_{W,c(\vec{u})}}(c(\vec{u})\!=\!v \mid C') = p'$ in Π, it follows that $W \not\models C'$. So there is only one pr-atom that is applied in W for $c(\vec{u})\!=\!v$, and we define

$$\mathit{PossWithAssPr}(W, c(\vec{u})\!=\!v) = p.$$

(“$c(\vec{u})\!=\!v$ is possible in W with assigned probability p.”)

- If $v \notin \mathit{AV}_W(c(\vec{u}))$, we define

$$\mathit{PossWithDefPr}(W, c(\vec{u})\!=\!v) = \max\big(p, 0\big),$$

where p is

$$\frac{1 - \sum_{v' \in \mathit{AV}_W(c(\vec{u}))} \mathit{PossWithAssPr}(W, c(\vec{u})\!=\!v')}{|\{v'' \mid c(\vec{u})\!=\!v'' \text{ is possible in } W \text{ and } v'' \notin \mathit{AV}_W(c(\vec{u}))\}|}. \qquad (2.4)$$

(“$c(\vec{u})\!=\!v$ is possible in W with the default probability.”)

The max function is used to ensure that the default probability is nonnegative. [4]

[3] Note that this is slightly different from the original definition of P-log from Baral *et al.* (2009), according to which, if $\mathit{Intervene}(c(\vec{u}))$ is true, the probability of $c(\vec{u})\!=\!v$ is determined by the default probability, which is a bit unintuitive.

[4] In Baral *et al.* (2009), a stronger condition of “unitariness” is imposed to prevent (2.4) from being negative.

For each possible world $W \in \omega(\Pi)$, and each atom $c(\vec{u})=v$ possible in W, the probability of $c(\vec{u})=v$ to *happen* in W is defined as:

$$P(W, c(\vec{u})=v) = \begin{cases} PossWithAssPr(W, c(\vec{u})=v) & \text{if } v \in AV_W(c(\vec{u})); \\ PossWithDefPr(W, c(\vec{u})=v) & \text{otherwise.} \end{cases}$$

The *unnormalized probability* of a possible world W is defined as

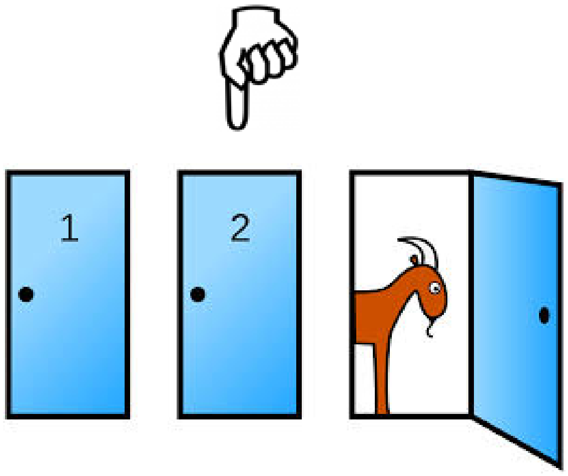
$$\hat{\mu}_\Pi(W) = \prod_{\substack{c(\vec{u})=v \in W \text{ and} \\ c(\vec{u})=v \text{ is possible in } W}} P(W, c(\vec{u})=v),$$

and, assuming Π has at least one possible world with nonzero unnormalized probability, the *normalized probability* of W is defined as

$$\mu_\Pi(W) = \frac{\hat{\mu}_\Pi(W)}{\sum_{W_i \in \omega(\Pi)} \hat{\mu}_\Pi(W_i)}.$$

We say Π is *consistent* if Π has at least one possible world with a non-zero probability.

Example 4 *The following P-log program Π describes Monty Hall Problem. A player*



is given the opportunity to select one of three closed doors, behind which are two goats and one car. Once the player has made a selection, Monty, who is the show host, *is obligated to open one of the remaining closed two doors which does not contain the car. He then asks the player if he would like to switch his selection to the other unopened door, or stay with his original choice. Let's denote the door that is selected by the player as door 2, the door that Monty opens as door 3, and the remaining as door 1.*

$$\sim CanOpen(d) \leftarrow Selected = d. \qquad (d \in \{1, 2, 3\})$$
$$\sim CanOpen(d) \leftarrow Prize = d.$$
$$CanOpen(d) \leftarrow not\ \ \sim CanOpen(d).$$

$$random(Prize).$$
$$random(Selected).$$
$$random(Open : \{X : CanOpen(X)\}).$$

$$Obs(Selected = 2).$$
$$Obs(Open = 3).$$
$$Obs(Prize \neq 3).$$

By definition, $\tau(\Pi)$ *is as follows:*

$\sim CanOpen(d) \leftarrow Selected = d$

$\sim CanOpen(d) \leftarrow Prize = d$

$CanOpen(d) \leftarrow not \sim CanOpen(d)$

$\leftarrow CanOpen(d), \sim CanOpen(d)$

$\leftarrow Prize = d_1, Prize = d_2, d_1 \neq d_2$

$\leftarrow Selected = d_1, Selected = d_2, d_1 \neq d_2$

$\leftarrow Open = d_1, Open = d_2, d_1 \neq d_2$

$Prize = 1; Prize = 2; Prize = 3 \leftarrow not\ Intervene(Prize)$

$Selected = 1; Selected = 2; Selected = 3 \leftarrow not\ Intervene(Selected)$

$Open = 1; Open = 2; Open = 3 \leftarrow not\ Intervene(Open)$

$\leftarrow Open = d, not\ CanOpen(d), not\ Intervene(Open)$

$Obs(Selected = 2)$

$\leftarrow Obs(Selected = 2), not\ Selected = 2$

$Obs(Open = 3)$

$\leftarrow Obs(Open = 3), not\ Open = 3$

$Obs(Prize \neq 3)$

$\leftarrow Obs(Prize \neq 3), Prize = 3$

There are two possible worlds W_1 and W_2 of Π (which are two stable models of $\tau(\Pi)$):

$$W_1 = \{Prize = 1, CanOpen(1) = \mathbf{f}\} \cup U,$$

$$W_2 = \{Prize = 2, CanOpen(1) = \mathbf{t}\} \cup U,$$

where U denotes the intersection of them, and

$$U = \{Obs(Selected = 2), Obs(Open = 3), Obs(Prize \neq 3),$$
$$Selected = 2, Open = 3, CanOpen(2) = \mathbf{f}, CanOpen(3) = \mathbf{t}\}.$$

Note that, by definition, $Prize = 1$, $Prize = 2$, $Prize = 3$, $Selected = 1$, $Selected = 2$, $Selected = 3$ are all possible in W_1 and W_2. Then the unnormalized weight $\hat{\mu}_\Pi(W_i)$ of each possible world W_i are shown below:

$$\begin{aligned}
\hat{\mu}_\Pi(W_1) &= P(W_1, Prize = 1) \times P(W_1, Selected = 2) \times P(W_1, Open = 3) \\
&= PossWithDefPr(W_1, Prize = 1) \times PossWithDefPr(W_1, Selected = 2) \times \\
&\quad PossWithDefPr(W_1, Open = 3) \\
&= \tfrac{1}{3} \times \tfrac{1}{3} \times \tfrac{1}{1} = \tfrac{1}{9}
\end{aligned}$$

$$\begin{aligned}
\hat{\mu}_\Pi(W_2) &= P(W_2, Prize = 2) \times P(W_2, Selected = 2) \times P(W_2, Open = 3) \\
&= PossWithDefPr(W_2, Prize = 2) \times PossWithDefPr(W_2, Selected = 2) \times \\
&\quad PossWithDefPr(W_2, Open = 3) \\
&= \tfrac{1}{3} \times \tfrac{1}{3} \times \tfrac{1}{2} = \tfrac{1}{18}
\end{aligned}$$

The normalized weight (probability) $\mu_\Pi(W_i)$ of each possible world W_i is as follows:

$$\mu_\Pi(W_1) = \tfrac{2}{3}; \quad \mu_\Pi(W_2) = \tfrac{1}{3}.$$

As we can see, switching the choice doubles the chance to win the car.

Note that in this example, the unnormalized weight of each possible world is not exactly its probability, but $\frac{1}{6}$ of it. The reason is that this example contains Obs, which prunes out the other 10 possible worlds [5] *that contain $selected = 1$, $selected = 3$, $open = 1$, $open = 2$, or $prize = 3$. We still assign the probability to these pruned out possible worlds based on the intuition that "observations don't affect the probability distribution".*

[5]These possible worlds could be added here if necessary.

Chapter 3

TURNING LP$^{\text{MLN}}$ INTO PROGRAMS WITH WEAK CONSTRAINTS

In this chapter, we show how an LP$^{\text{MLN}}$ program can be turned into an answer set program with weak constraints, so that the probabilistic stable models of the LP$^{\text{MLN}}$ program are exactly the stable models of the translated answer set program with weak constraints, while the probability distribution is also preserved.

3.1 General Translation

3.1.1 Translation lpmln2wc

We define a translation that turns an LP$^{\text{MLN}}$ program into a program with weak constraints. For any ground LP$^{\text{MLN}}$ program Π, the translation lpmln2wc(Π) is simply defined as follows. We turn each weighted formula

$$w : F$$

in Π into

$$\{F\}^{\text{ch}},$$

where $\{F\}^{\text{ch}}$ is a choice formula, standing for $F \vee \neg F$ (Ferraris *et al.*, 2011). Further, we add

$$:\sim F \quad [-1@1] \tag{3.1}$$

if w is α, and

$$:\sim F \quad [-w@0] \tag{3.2}$$

otherwise.

Intuitively, choice formula $\{F\}^{\rm ch}$ allows F to be either included or not in deriving a stable model. [1] When F is included, the stable model gets the (negative) penalty -1 at level 1 or $-w$ at level 0 depending on the weight of the formula, which corresponds to the (positive) "reward" e^{α} or e^{w} that it receives under the LP$^{\rm MLN}$ semantics.

The following proposition tells us that choice formulas can be used for generating the members of SM[Π].

Proposition 1 *For any* LP$^{\rm MLN}$ *program* Π*, the set* SM[Π] *is exactly the set of the stable models of* lpmln2wc(Π)*.*

The following theorem follows from Proposition 1. As the probability of a stable model of an LP$^{\rm MLN}$ program Π increases, the penalty of the corresponding stable model of lpmln2wc(Π) decreases, and the distinction between hard rules and soft rules can be simulated by the different levels of weak constraints, and vice versa.

Theorem 1 *For any* LP$^{\rm MLN}$ *program* Π*, the most probable stable models (i.e., the stable models with the highest probability) of* Π *are precisely the optimal stable models of the program with weak constraints* lpmln2wc(Π)*.*

Example 1 Continued *For program* Π*:*

$$\begin{array}{rcll} 10 & : & p \rightarrow q & (r_1) \\ 1 & : & p \rightarrow r & (r_2) \\ 5 & : & p & (r_3) \\ -20 & : & \neg r \rightarrow \bot & (r_4) \end{array}$$

SM[Π] *has* 5 *elements:* $\emptyset$*,* $\{p\}$*,* $\{p,q\}$*,* $\{p,r\}$*,* $\{p,q,r\}$*. Among them,* $\{p,q\}$ *is the most probable stable model, whose weight is* e^{15}*, while* $\{p,q,r\}$ *is a probabilistic*

[1]This view of choice formulas was used in PrASP (Nickles and Mileo, 2014) in defining *spanning* programs.

stable model whose weight is e^{-4}. *The translation* lpmln2wc *yields*

$$\begin{array}{c|rcr}
\{p \rightarrow q\}^{\rm ch} & :\sim & p \rightarrow q & [-10 \;@\; 0] \\
\{p \rightarrow r\}^{\rm ch} & :\sim & p \rightarrow r & [-1 \;@\; 0] \\
\{p\}^{\rm ch} & :\sim & p & [-5 \;@\; 0] \\
\{\neg r \rightarrow \bot\}^{\rm ch} & :\sim & \neg r \rightarrow \bot & [20 \;@\; 0]
\end{array}$$

whose optimal stable model is $\{p, q\}$ *with the penalty at level* 0 *being* -15, *while* $\{p, q, r\}$ *is a stable model whose penalty at level* 0 *is* 4.

The following example illustrates how the translation accounts for the difference between hard rules and soft rules by assigning different levels.

Example 2 continued: *For program* Π *in Example 2, the translation* lpmln2wc(Π) *is* [2]

$$\begin{array}{l}
\{Bird(Jo) \rightarrow ResidentBird(Jo)\}^{\rm ch} \\
\{Bird(Jo) \rightarrow MigratoryBird(Jo)\}^{\rm ch} \\
\{\bot \rightarrow ResidentBird(Jo), MigratoryBird(Jo)\}^{\rm ch} \\
\{ResidentBird(Jo)\}^{\rm ch} \\
\{MigratoryBird(Jo)\}^{\rm ch}
\end{array}$$

$$\begin{array}{ll}
:\sim\ Bird(Jo) \rightarrow ResidentBird(Jo) & [-1@1] \\
:\sim\ Bird(Jo) \rightarrow MigratoryBird(Jo) & [-1@1] \\
:\sim\ \bot \rightarrow ResidentBird(Jo), MigratoryBird(Jo)\} & [-1@1] \\
:\sim\ ResidentBird(Jo) & [-2@0] \\
:\sim\ MigratoryBird(Jo) & [-1@0]
\end{array}$$

The three probabilistic stable models of Π,

$$\emptyset, \{Bird(Jo), ResidentBird(Jo)\}, \text{and } \{Bird(Jo), MigratoryBird(Jo)\},$$

[2]Recall that we identify the rules with the corresponding first-order formulas.

have the same penalty -3 *at level* 1*. Among them,*

$$\{Bird(Jo), ResidentBird(Jo)\}$$

has the least penalty at level 0*, and thus is an optimal stable model of* lpmln2wc(Π).

Note that while the most probable stable models of Π and the optimal stable models of lpmln2wc(Π) coincide, their weights and penalties are not proportional to each other. The former is defined by an exponential function whose exponent is the sum of the weights of the satisfied formulas, while the latter simply adds up the penalties of the satisfied formulas. On the other hand, they are monotonically increasing/decreasing as more formulas are satisfied, which yields a way to simulate the exponential function of LP$^{\text{MLN}}$ in weak constraints.

Let Π be a logic program with weak constraints whose $Level \in \{0,1\}$, and I be any interpretation of Π. We define a *penalty weight* of I under Π as

$$W(\Pi, I) = \begin{cases} exp\big(-Penalty_{\Pi}(I,1) \times \alpha - Penalty_{\Pi}(I,0)\big) & \text{if } I \text{ is a stable model of } \Pi, \\ 0 & \text{otherwise;} \end{cases}$$

and define a *penalty probability* of I under Π by normalizing its penalty weight as

$$P(\Pi, I) = \lim_{\alpha \to \infty} \frac{W(\Pi, I)}{\sum\limits_{J \text{ is a stable model of } \Pi} W(\Pi, J)}.$$

The following corollary, following from Proposition 1 and Theorem 1, shows that for any LP$^{\text{MLN}}$ program Π and any interpretation I of Π, its weight and probability are preserved in the translated logic program with weak constraints lpmln2wc(Π).

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

- $W_{\Pi}(I) = W(\text{lpmln2wc}(\Pi), I)$*, and*
- $P_{\Pi}(I) = P(\text{lpmln2wc}(\Pi), I)$*.*

By Corollary 1, we can calculate the probability of I by simply analyzing the penalties it received in lpmln2wc(Π), which yields a way to compute the probability distribution of an LP$^{\text{MLN}}$ program using the output of standard ASP solvers, such as CLINGO.

Example 1 Continued *Consider the translation* lpmln2wc(Π), *denoted by* Π'*:*

$$\begin{array}{c|cc r}
\{p \rightarrow q\}^{\text{ch}} & :\sim & p \rightarrow q & [-10 \ @ \ 0] \\
\{p \rightarrow r\}^{\text{ch}} & :\sim & p \rightarrow r & [-1 \ @ \ 0] \\
\{p\}^{\text{ch}} & :\sim & p & [-5 \ @ \ 0] \\
\{\neg r \rightarrow \bot\}^{\text{ch}} & :\sim & \neg r \rightarrow \bot & [20 \ @ \ 0].
\end{array}$$

For any interpretation I of Π', the following table shows the penalty of I at level 0, its penalty weight, and its penalty probability under Π'. Note that when I is not a stable model of Π', its penalty is not defined (denoted by N/A).

I	*Penalty*$(I, 0)$	$W(\Pi', I)$	$P(\Pi', I)$
$\emptyset$	$-10-1$	e^{10+1}	$\frac{e^{11}}{e^{11}+e^{5}+e^{15}+e^{-14}+e^{-4}}$
$\{p\}$	-5	e^{5}	$\frac{e^{5}}{e^{11}+e^{5}+e^{15}+e^{-14}+e^{-4}}$
$\{q\}$	N/A	0	0
$\{r\}$	N/A	0	0
$\{p, q\}$	$-10-5$	e^{10+5}	$\frac{e^{15}}{e^{11}+e^{5}+e^{15}+e^{-14}+e^{-4}}$
$\{p, r\}$	$-1-5+20$	e^{1+5-20}	$\frac{e^{-14}}{e^{11}+e^{5}+e^{15}+e^{-14}+e^{-4}}$
$\{q, r\}$	N/A	0	0
$\{p, q, r\}$	$-10-1-5+20$	$e^{10+1+5-20}$	$\frac{e^{-4}}{e^{11}+e^{5}+e^{15}+e^{-14}+e^{-4}}$

As we see, $W(\Pi', I)$ and $P(\Pi', I)$ are exactly the same as $W_\Pi(I)$ and $P_\Pi(I)$ that are defined in the semantics of LP$^{\text{MLN}}$.

Given an LP$^{\text{MLN}}$ program Π, to know the probability of an interpretation I of Π, we can either directly calculate it based on the semantics of LP$^{\text{MLN}}$, or compute the penalty probability of I under lpmln2wc(Π). No matter what procedure we are using, we need to accumulate the weights or penalty weights of all stable models of lpmln2wc(Π) as the normalization factor, which is computationally expensive. In this case, computing a conditional probability can be more effective because it decreases the number of stable models that we need to consider.

Let Π be a logic program with weak constraints. We define a *total weight* Z for Π by summarizing the penalty weights of all its stable models as

$$Z(\Pi) = \sum_{I \text{ is a stable model of } \Pi} W(\Pi, I).$$

For any first-order formula F, let F^{constr} be the ASP rule "$\leftarrow$ *not* F".

The following corollary shows that we can compute conditional probabilities under an LP$^{\text{MLN}}$ program Π by simply comparing the outputs of two logic programs with weak constraints.

Corollary 2 *Let* Π *be an* LP$^{\text{MLN}}$ *program,* Π' *be the translated logic program with weak constraints* lpmln2wc(Π). *If* F_2 *is satisfiable in* Π*, the probability of* F_1 *given* F_2 *under* Π *is:*

$$P_{\Pi}(F_1 \mid F_2) = \lim_{\alpha \to \infty} \frac{Z(\Pi' \wedge F_1^{constr} \wedge F_2^{constr})}{Z(\Pi' \wedge F_2^{constr})}.$$

By Corollary 2, we only need to consider a subset of the stable models of Π' that satisfy F_2. Note that calculating $P_{\Pi}(I)$ is a special case of computing the conditional probability $P_{\Pi}(F_1 \mid F_2)$, where F_1 is a conjunction of literals in I (including atoms that are false) and F_2 is $\top$.

Example 1 Continued *Consider the translation* lpmln2wc(Π), *denoted by* Π′*:*

$$\begin{array}{c|cccr} \{p \rightarrow q\}^{\rm ch} & :\sim & p \rightarrow q & [-10 \;@\; 0] \\ \{p \rightarrow r\}^{\rm ch} & :\sim & p \rightarrow r & [-1 \;@\; 0] \\ \{p\}^{\rm ch} & :\sim & p & [-5 \;@\; 0] \\ \{\neg r \rightarrow \bot\}^{\rm ch} & :\sim & \neg r \rightarrow \bot & [20 \;@\; 0]. \end{array}$$

$\Pi' \wedge r^{constr}$ *has a stable model* $\{p, r\}$ *with penalty weight* e^{-14} *and a stable model* $\{p, q, r\}$ *with penalty weight* e^{-4}*, while the latter one is the only stable model of* $\Pi' \wedge q^{constr} \wedge r^{constr}$*. Thus the probability of* p *given* q *under* Π *is*

$$P_\Pi(q \mid r) = \frac{Z(\Pi' \wedge q^{constr} \wedge r^{constr})}{Z(\Pi' \wedge r^{constr})} = \frac{e^{-4}}{e^{-14} + e^{-4}},$$

which is exactly the same as the conditional probability we calculated under the semantics of $\mathrm{LP}^{\mathrm{MLN}}$.

3.1.2 Translation mln2wc

In view of Theorem 2 from (Lee and Wang, 2016), which tells us how to embed Markov Logic into $\mathrm{LP}^{\mathrm{MLN}}$, it follows from **Proposition 1** that the models finding in MLN can also be reduced to the stable models finding in answer set programs. Besides, it follows from **Theorem 1** that MAP inference in MLN can be reduced to the optimal stable model finding of programs with weak constraints.

For any Markov Logic Network Π, let mln2wc(Π) be the union of lpmln2wc(Π) plus choice rules $\{A\}^{\rm ch}$ for all atoms A in Π.

Proposition 2 *For any Markov Logic Network program* Π*, the models of* Π *are precisely the stable models of the program with weak constraints* mln2wc(Π)*.*

Theorem 2 *For any Markov Logic Network* Π*, the most probable models of* Π *are precisely the optimal stable models of the program with weak constraints* mln2wc(Π)*.*

Similarly, MAP inference in ProbLog and Pearl's Causal Models can be reduced to finding an optimal stable model of a program with weak constraints in view of the reduction of ProbLog to LP$^{\rm MLN}$ (Theorem 4 from (Lee and Wang, 2016)) and the reduction of Causal Models to LP$^{\rm MLN}$ (Theorem 4 from (Lee *et al.*, 2015)) thereby allowing us to apply combinatorial optimization methods in standard ASP solvers to these languages.

Furthermore, the probability distribution in a Markov Logic Network Π is also preserved in mln2wc(Π), and we can compute the conditional probability in MLN by using standard ASP solvers. The following corollaries follow from **Corollary 1** and **Corollary 2**.

Recall that the weight and probability of an interpretation I of a Markov Logic Network Π is defined as

$$\begin{aligned} W^{\rm mln}_{\Pi}(I) = {} & exp(\sum_{w:F\in\Pi \text{ and } I\models F} w) \\ P^{\rm mln}_{\Pi}(I) = {} & \lim_{\alpha\to\infty} \frac{W^{\rm mln}_{\Pi}(I)}{\sum\limits_{J} W^{\rm mln}_{\Pi}(J)}. \end{aligned}$$

We say I is satisfiable in Π if $P^{\rm mln}_{\Pi}(I) \neq 0$. The probability of a first-order formula F and the conditional probability of F_1 given F_2 (where F_2 is satisfiable in Π) are defined as (same as the definition in LP$^{\rm MLN}$):

$$\begin{aligned} P^{\rm mln}_{\Pi}(F) = {} & \sum_{I:I\models F} P^{\rm mln}_{\Pi}(I) \\ P^{\rm mln}_{\Pi}(F_1 \mid F_2) = {} & \frac{P^{\rm mln}_{\Pi}(F_1\wedge F_2)}{P^{\rm mln}_{\Pi}(F_2)}. \end{aligned}$$

Corollary 3 *For any Markov Logic Network* Π *and any interpretation* I *of* Π*,*

- $W^{\rm mln}_{\Pi}(I) = W(\text{mln2wc}(\Pi), I)$*, and*

- $P^{\rm mln}_{\Pi}(I) = P(\text{mln2wc}(\Pi), I)$*.*

Corollary 4 *Let* Π *be a Markov Logic Network,* Π' *be the translated logic program with weak constraints* mln2wc(Π)*. If* F_2 *is satisfiable in* Π*, the probability of* F_1 *given* F_2 *under* Π *is:*

$$P^{\rm mln}_{\Pi}(F_1 \mid F_2) = \lim_{\alpha\to\infty} \frac{Z(\Pi' \wedge F_1^{constr} \wedge F_2^{constr})}{Z(\Pi' \wedge F_2^{constr})}.$$

Example 5 *Consider the MLN program* Π*:*

$$\begin{array}{rcll} 10 & : & p \rightarrow q & (r_1) \\ 1 & : & p \rightarrow r & (r_2) \\ 5 & : & p & (r_3) \\ -20 & : & \neg r \rightarrow \bot & (r_4) \end{array}$$

There are $2^3 = 8$ *models of* Π*, among which*

$$I = \{p, q\}$$

is the most probable model and $W^{\rm mln}_{\Pi}(I) = e^{15}$*. The translation* mln2wc *yields*

$$\begin{array}{c|c|ccr} \{p \rightarrow q\}^{\rm ch} & \{p\}^{\rm ch} & :\sim & p \rightarrow q & [-10 \;@\; 0] \\ \{p \rightarrow r\}^{\rm ch} & \{q\}^{\rm ch} & :\sim & p \rightarrow r & [-1 \;@\; 0] \\ \{p\}^{\rm ch} & \{r\}^{\rm ch} & :\sim & p & [-5 \;@\; 0] \\ \{\neg r \rightarrow \bot\}^{\rm ch} & & :\sim & \neg r \rightarrow \bot & [20 \;@\; 0] \end{array}$$

whose optimal stable model is also $\{p, q\}$ *with the penalty at level* 0 *being* -15*, while* $\{p, q, r\}$ *is a stable model whose penalty at level* 0 *is* 4*. Let* Π' *denote* mln2wc(Π)*,* $W(\Pi', I) = exp(-Penalty(I, 0)) = e^{15}$*, which is equivalent to* $W^{\rm mln}_{\Pi}(I)$*.*

3.2 Alternative Translations

3.2.1 *Translation* lpmln2wc$^{\rm pnt}$

Instead of aggregating the weights of satisfied formulas, we may aggregate the weights of formulas that are not satisfied. Let lpmln2wc$^{\rm pnt}$(Π) be a modification of lpmln2wc(Π) by replacing (3.1) with

$$:\sim \quad \neg F \quad [1@1]$$

and (3.2) with

$$:\sim \quad \neg F \quad [w@0].$$

Intuitively, when F is not satisfied, the stable model gets the penalty 1 at level 1, or w at level 0 depending on whether F is a hard or soft formula.

Corollary 5 *For any* $\mathrm{LP}^{\mathrm{MLN}}$ *program* Π*, the set* $\mathrm{SM}[\Pi]$ *is exactly the set of the stable models of the program with weak constraints* $\mathrm{lpmln2wc}^{\mathrm{pnt}}(\Pi)$*.*

Corollary 6 *For any* $\mathrm{LP}^{\mathrm{MLN}}$ *program* Π *and any interpretation* I *of* Π*,*

- $W_\Pi(I) \propto W(\mathrm{lpmln2wc}^{\mathrm{pnt}}(\Pi), I)$*, and*

- $P_\Pi(I) = P(\mathrm{lpmln2wc}^{\mathrm{pnt}}(\Pi), I)$*.*

Corollary 7 *Let* Π *be an* $\mathrm{LP}^{\mathrm{MLN}}$ *program,* Π' *be the translated logic program with weak constraints* $\mathrm{lpmln2wc}^{\mathrm{pnt}}(\Pi)$*. If* F_2 *is satisfiable in* Π*, the probability of* F_1 *given* F_2 *under* Π *is:*

$$P_\Pi(F_1 \mid F_2) = \lim_{\alpha\to\infty} \frac{Z(\Pi' \wedge F_1^{constr} \wedge F_2^{constr})}{Z(\Pi' \wedge F_2^{constr})}.$$

This alternative view of assigning weights to stable models, in fact, originates from Probabilistic Soft Logic (PSL) (Bach *et al.*, 2015), where the probability density function of an interpretation is obtained from the sum over the “penalty” from the formulas that are “distant” from satisfaction. This view will lead to a slight advantage when we further turn the translation into the input language of ASP solvers (See Footnote 6).

Example 1 Continued *For* LP$^{\rm MLN}$ *program* Π*:*

$$\begin{array}{rcll} 10 & : & p \rightarrow q & (r_1) \\ 1 & : & p \rightarrow r & (r_2) \\ 5 & : & p & (r_3) \\ -20 & : & \neg r \rightarrow \bot & (r_4) \end{array}$$

The translation lpmln2wc$^{\rm pnt}$ *yields*

$$\begin{array}{c|crr} \{p \rightarrow q\}^{\rm ch} & :\sim & p, not\ q & [10\ @\ 0] \\ \{p \rightarrow r\}^{\rm ch} & :\sim & p, not\ r & [1\ @\ 0] \\ \{p\}^{\rm ch} & :\sim & not\ p & [5\ @\ 0] \\ \{\neg r \rightarrow \bot\}^{\rm ch} & :\sim & not\ r & [-20\ @\ 0] \end{array}$$

whose optimal stable model is $I_1 = \{p, q\}$ *with the penalty at level* 0 *being* -19*, while* $I_2 = \{p, q, r\}$ *is a stable model whose penalty at level* 0 *is* 0*. Let* Π' *denote* lpmln2wc$^{\rm pnt}(\Pi)$*,* k *denote the summation of the weights of all soft rules in* Π*. Then* $k = -4$*, and* $W(\Pi', I_1) = exp(-Penalty(I_1, 0)) = e^{19} = W_\Pi(I_1) \times e^{-k}$*, and* $W(\Pi', I_2) = exp(-Penalty(I_2, 0)) = e^0 = W_\Pi(I_2) \times e^{-k}$*.*

3.2.2 Translation lpmln2wc$^{\rm pnt,rule}$

The current ASP solvers do not allow arbitrary formulas to appear in choice rules or weak constraints. To omit the use of choice rules and random formulas F in the weak constraints, we introduce a new atom $unsat(i)$ for each weighted formula $w_i :\ F_i \in \Pi$ to the construction of lpmln2wc$^{\rm pnt}(\Pi)$.

Let lpmln2wc$^{\rm pnt,rule}(\Pi)$ be the translation by turning each weighted formula $w_i : F_i$ in Π into

$$\begin{array}{rcll} \neg F_i & \rightarrow & unsat(i) & \\ \neg unsat(i) & \rightarrow & F_i & \\ & :\sim & unsat(i) & [1@1]. \end{array}$$

if w_i is α, or

$$\begin{array}{rcll} \neg F_i & \rightarrow & unsat(i) & \\ \neg unsat(i) & \rightarrow & F_i & \\ & :\sim & unsat(i) & [w_i@0]. \end{array}$$

otherwise, where $unsat(i)$ is a new atom denoting the formula F_i is unsatisfied.

Corollary 8 *Let Π be an* LP$^{\rm MLN}$ *program. There is a 1-1 correspondence ϕ between the set* SM[Π] *and the stable models of* lpmln2wc$^{\rm pnt,rule}(\Pi)$*, where $\phi(I) = I \cup \{unsat(i) \mid w_i : F_i \in \Pi,\ I \not\models F_i\}$.*

Corollary 9 *For any* LP$^{\rm MLN}$ *program Π and any interpretation I of Π,*

- $W_\Pi(I) \propto W(\text{lpmln2wc}^{\rm pnt,rule}(\Pi), \phi(I))$*, and*
- $P_\Pi(I) = P(\text{lpmln2wc}^{\rm pnt,rule}(\Pi), \phi(I))$*,*

where $\phi(I) = I \cup \{unsat(i) \mid w_i : F_i \in \Pi,\ I \not\models F_i\}$.

Corollary 10 *Let Π be an* LP$^{\rm MLN}$ *program, Π' be the translated logic program with weak constraints* lpmln2wc$^{\rm pnt,rule}(\Pi)$*. If F_2 is satisfiable in Π, the probability of F_1 given F_2 under Π is:*

$$P_\Pi(F_1 \mid F_2) = \lim_{\alpha\to\infty} \frac{Z(\Pi' \wedge F_1^{constr} \wedge F_2^{constr})}{Z(\Pi' \wedge F_2^{constr})}.$$

The corollaries allow us to compute the most probable stable models (MAP estimates) and the probability distribution of the LP$^{\rm MLN}$ program using the combination of F2LP [3] and CLINGO [4] (assuming that the weights are approximated to integers). System F2LP turns this program with formulas into the input language of CLINGO, so CLINGO can be used to compute the theory.

[3] `http://reasoning.eas.asu.edu/f2lp/`

[4] `http://potassco.sourceforge.net`

Example 1 Continued *For* LP$^{\text{MLN}}$ *program* Π*:*

$$\begin{array}{rcll} 10 & : & p \rightarrow q & (r_1) \\ 1 & : & p \rightarrow r & (r_2) \\ 5 & : & p & (r_3) \\ -20 & : & \neg r \rightarrow \bot & (r_4) \end{array}$$

The translation lpmln2wc$^{\text{pnt,rule}}$ *yields*

$$\begin{array}{c|c|lll} \neg(p \rightarrow q) \rightarrow \mathit{unsat}(1) & \neg \mathit{unsat}(1) \rightarrow (p \rightarrow q) & :\sim & \mathit{unsat}(1) & [10 \ @ \ 0] \\ \neg(p \rightarrow r) \rightarrow \mathit{unsat}(2) & \neg \mathit{unsat}(2) \rightarrow (p \rightarrow r) & :\sim & \mathit{unsat}(2) & [1 \ @ \ 0] \\ \neg p \rightarrow \mathit{unsat}(3) & \neg \mathit{unsat}(3) \rightarrow p & :\sim & \mathit{unsat}(3) & [5 \ @ \ 0] \\ \neg(\neg r \rightarrow \bot) \rightarrow \mathit{unsat}(4) & \neg \mathit{unsat}(4) \rightarrow (\neg r \rightarrow \bot) & :\sim & \mathit{unsat}(4) & [-20 \ @ \ 0] \end{array}$$

whose optimal stable model is $\{p, q, \mathit{unsat}(2), \mathit{unsat}(4)\}$ *with the penalty at level* 0 *being* -19*, while* $\{p, q, r\}$ *is a stable model whose penalty at level* 0 *is* 0*. Let* Π' *denote* lpmln2wc$^{\text{pnt,rule}}(\Pi)$*,* k *denote the summation of the weights of all soft rules in* Π*. Then* $k = -4$*, and* $W(\Pi', I_1) = exp(-Penalty(I_1, 0)) = e^{19} = W_{\Pi}(I_1) \times e^{-k}$*, and* $W(\Pi', I_2) = exp(-Penalty(I_2, 0)) = e^{0} = W_{\Pi}(I_2) \times e^{-k}$*.*

3.2.3 Translation lpmln2wc$^{\text{pnt,clingo}}$

Until now, we provided three translations [5] from general LP$^{\text{MLN}}$ program to ASP with weak constraints. One drawback of these translations is that they are not in the input language of CLINGO because of the arbitrary formulas F in a general LP$^{\text{MLN}}$ program. This motivates us to introduce the following fourth translation lpmln2wc$^{\text{pnt,clingo}}$.

Recall: The *rule form* that is allowed in the input languages of CLINGO and DLV is

$$A_1; \ldots; A_k \leftarrow B_1, \ldots, B_m, \mathit{not}\ B_{m+1}, \ldots, \mathit{not}\ B_n, \mathit{not\ not}\ B_{n+1}, \ldots, \mathit{not\ not}\ B_p$$

[5] lpmln2wc, lpmln2wc$^{\text{pnt}}$, and lpmln2wc$^{\text{pnt,rule}}$.

where $A_1, \dots, A_k, B_1, \dots, B_p$ are atoms; k, m, n, p are non-negative integers; and $0 \leq m \leq n \leq p$. We also abbreviate the rule form as $\mathit{Head} \leftarrow \mathit{Body}$.

If an unweighted LP$^{\text{MLN}}$ program is already in the rule form $\mathit{Head} \leftarrow \mathit{Body}$ that is allowed in the input languages of CLINGO and DLV, a new translation lpmln2wc$^{\text{pnt,clingo}}$ may avoid the use of F2LP by slightly modifying the translation lpmln2wc$^{\text{pnt,rule}}$ by turning each weighted rule

$$w_i : \mathit{Head}_i \leftarrow \mathit{Body}_i$$

instead into

$$\begin{array}{rcl} \mathit{unsat}(i) & \leftarrow & \mathit{Body}_i, \mathit{not}\ \mathit{Head}_i \\ \mathit{Head}_i & \leftarrow & \mathit{Body}_i, \mathit{not}\ \mathit{unsat}(i) \\ & :\sim & \mathit{unsat}(i) \quad [1@1] \end{array}$$

if w_i is α, or

$$\begin{array}{rcl} \mathit{unsat}(i) & \leftarrow & \mathit{Body}_i, \mathit{not}\ \mathit{Head}_i \\ \mathit{Head}_i & \leftarrow & \mathit{Body}_i, \mathit{not}\ \mathit{unsat}(i) \\ & :\sim & \mathit{unsat}(i) \quad [w_i@0] \end{array}$$

otherwise.

Corollary 11 *Let* Π *be an* LP$^{\text{MLN}}$ *program such that all unweighted rules of* Π *are in the rule form Head* $\leftarrow$ *Body. There is a 1-1 correspondence* ϕ *between the set* SM[Π] *and the stable models of* lpmln2wc$^{\text{pnt,clingo}}$(Π), *where* $\phi(I) = I \cup \{\mathit{unsat}(i) \mid w_i : \mathit{Head}_i \leftarrow \mathit{Body}_i \in \Pi, I \models \mathit{Body}_i \wedge \neg \mathit{Head}_i\}$.

Corollary 12 *For any* LP$^{\text{MLN}}$ *program* Π *and any interpretation* I *of* Π,

- $W_\Pi(I) \propto W(\text{lpmln2wc}^{\text{pnt,clingo}}(\Pi), \phi(I))$, *and*
- $P_\Pi(I) = P(\text{lpmln2wc}^{\text{pnt,clingo}}(\Pi), \phi(I))$,

where $\phi(I) = I \cup \{\mathit{unsat}(i) \mid w_i : \mathit{Head}_i \leftarrow \mathit{Body}_i \in \Pi, I \models \mathit{Body}_i \wedge \neg \mathit{Head}_i\}$.

Corollary 13 *Let* Π *be an* LP^{MLN} *program,* Π' *be the translated logic program with weak constraints* $\text{lpmln2wc}^{\text{pnt,clingo}}(\Pi)$*. If* F_2 *is satisfiable in* Π*, the probability of* F_1 *given* F_2 *under* Π *is:*

$$P_\Pi(F_1 \mid F_2) = \lim_{\alpha\to\infty} \frac{Z(\Pi' \wedge F_1^{constr} \wedge F_2^{constr})}{Z(\Pi' \wedge F_2^{constr})}.$$

Example 1 continued: *For* LP^{MLN} *program* Π*:*

$$\begin{array}{rcll} 10 & : & p \rightarrow q & (r_1) \\ 1 & : & p \rightarrow r & (r_2) \\ 5 & : & p & (r_3) \\ -20 & : & \neg r \rightarrow \bot & (r_4) \end{array}$$

The translation $\text{lpmln2wc}^{\text{pnt,clingo}}$ *yields*

$unsat(1) \leftarrow p, not\ q$	$q \leftarrow p, not\ unsat(1)$	$:\sim unsat(1)$	$[10@0]$
$unsat(2) \leftarrow p, not\ r$	$r \leftarrow p, not\ unsat(2)$	$:\sim unsat(2)$	$[1@0]$
$unsat(3) \leftarrow not\ p$	$p \leftarrow not\ unsat(3)$	$:\sim unsat(3)$	$[5@0]$
$unsat(4) \leftarrow not\ r$	$\bot \leftarrow not\ r, unsat(4)$	$:\sim unsat(4)$	$[-20@0]$

3.2.4 Translation $\text{lpmln2wc}^{\text{pnt,clingo}}_{\text{simp}}$

Translation $\text{lpmln2wc}^{\text{pnt,clingo}}$ can be further simplified when $Head_i$ is $\bot$.

For any LP^{MLN} program Π such that all unweighted rules of Π are in the rule form $Head \leftarrow Body$, let $\text{lpmln2wc}^{\text{pnt,clingo}}_{\text{simp}}(\Pi)$ be the translation by turning each weighted rule $w_i : Head \leftarrow Body$ in Π into

$$:\sim \quad Body_i \quad [w_i'@l]$$

if $Head_i$ is $\bot$, or

$$\begin{array}{rcl} unsat(i) & \leftarrow & Body_i, not\ Head_i \\ Head_i & \leftarrow & Body_i, not\ unsat(i) \\ & :\sim & unsat(i) \quad [w'_i@l] \end{array}$$

otherwise, where "$w'_i@l$" is "1@1" if w_i is α, or "$w_i@0$" if w_i is a real number.

Corollary 14 *Let* Π *be an* LP^{MLN} *program such that all unweighted rules of* Π *are in the rule form Head* $\leftarrow$ *Body.There is a 1-1 correspondence* ϕ *between the set* $\text{SM}[\Pi]$ *and the stable models of* $\text{lpmln2wc}^{\text{pnt,clingo}}_{\text{simp}}(\Pi)$*, where* $\phi(I) = I \cup \{unsat(i) \mid w_i : Head_i \leftarrow Body_i \in \Pi, Head_i \text{ is not } \bot, I \models Body_i \wedge \neg Head_i\}$.

Corollary 14 asserts that we can simply turn an LP^{MLN} rule of the form

$$w_i : \bot \leftarrow Body_i$$

into

$$:\sim Body_i \quad [w_i@l]$$

without introducing a new atom $unsat(i)$.

Corollary 15 *For any* LP^{MLN} *program* Π *and any interpretation* I *of* Π*,*

- $W_\Pi(I) \propto W(\text{lpmln2wc}^{\text{pnt,clingo}}_{\text{simp}}(\Pi), \phi(I))$*, and*
- $P_\Pi(I) = P(\text{lpmln2wc}^{\text{pnt,clingo}}_{\text{simp}}(\Pi), \phi(I))$*,*

where $\phi(I) = I \cup \{unsat(i) \mid w_i : Head_i \leftarrow Body_i \in \Pi, Head_i \text{ is not } \bot, I \models Body_i \wedge \neg Head_i\}$.

Corollary 16 *Let* Π *be an* LP^{MLN} *program,* Π' *be the translated logic program with weak constraints* $\text{lpmln2wc}^{\text{pnt,clingo}}_{\text{simp}}(\Pi)$*. If* F_2 *is satisfiable in* Π*, the probability of* F_1 *given* F_2 *under* Π *is:*

$$P_\Pi(F_1 \mid F_2) = \lim_{\alpha \to \infty} \frac{Z(\Pi' \wedge F_1^{constr} \wedge F_2^{constr})}{Z(\Pi' \wedge F_2^{constr})}.$$

Example 1 continued: *For program* Π *in Example 1, the simplified translation* $\text{lpmln2wc}^{\text{pnt,clingo}}_{\text{simp}}(\Pi)$ *is*

$$
\begin{array}{l|l|ll}
unsat(1) \leftarrow p, not\ q & q \leftarrow p, not\ unsat(1) & :\sim unsat(1) & [10@0] \\
unsat(2) \leftarrow p, not\ r & r \leftarrow p, not\ unsat(2) & :\sim unsat(2) & [1@0] \\
unsat(3) \leftarrow not\ p & p \leftarrow not\ unsat(3) & :\sim unsat(3) & [5@0] \\
 & & :\sim not\ r & [-20@0]
\end{array}
$$

Example 2 continued: *For program* Π *in Example 2, the simplified translation* $\text{lpmln2wc}^{\text{pnt,clingo}}_{\text{simp}}(\Pi)$ *is*

$$
\begin{array}{ll}
unsat(1) \leftarrow ResidentBird(Jo), not\ Bird(Jo) & \\
Bird(Jo) \leftarrow ResidentBird(Jo), not\ unsat(1) & \\
:\sim unsat(1) & [1@1] \\
 & \\
unsat(2) \leftarrow MigratoryBird(Jo), not\ Bird(Jo) & \\
Bird(Jo) \leftarrow MigratoryBird(Jo), not\ unsat(2) & \\
:\sim unsat(2) & [1@1] \\
 & \\
:\sim ResidentBird(Jo), MigratoryBird(Jo) & [1@1] \\
 & \\
unsat(4) \leftarrow not\ ResidentBird(Jo) & \\
ResidentBird(Jo) \leftarrow not\ unsat(4) & \\
:\sim unsat(4) & [2@0] \\
 & \\
unsat(5) \leftarrow not\ MigratoryBird(Jo) & \\
MigratoryBird(Jo) \leftarrow not\ unsat(5) & \\
:\sim unsat(5) & [1@0]
\end{array}
$$

3.3 Simplification for Hard Rules

In some applications, one may not want any hard rules to be violated assuming that hard rules encode definite knowledge. For that, the translations lpmln2wc$^{\text{pnt,clingo}}$ and lpmln2wc$^{\text{pnt,clingo}}_{\text{simp}}$ can be modified by simply turning hard rules

$$\alpha : \mathit{Head} \leftarrow \mathit{Body}$$

instead into the usual ASP rules

$$\mathit{Head} \leftarrow \mathit{Body}.$$

Example 2 continued: *The program in Example 2 can be translated into programs with weak constraints as follows.*

$$\begin{array}{ll}
\mathit{Bird}(\mathit{Jo}) \leftarrow \mathit{ResidentBird}(\mathit{Jo}) & \\
\mathit{Bird}(\mathit{Jo}) \leftarrow \mathit{MigratoryBird}(\mathit{Jo}) & \\
\bot \leftarrow \mathit{ResidentBird}(\mathit{Jo}), \mathit{MigratoryBird}(\mathit{Jo}) & \\
 & \\
\mathit{unsat}(4) \leftarrow \mathit{not}\ \mathit{ResidentBird}(\mathit{Jo}) & \\
\mathit{ResidentBird}(\mathit{Jo}) \leftarrow \mathit{not}\ \mathit{unsat}(4) & \\
:\sim \mathit{unsat}(4) & [2@0] \\
 & \\
\mathit{unsat}(5) \leftarrow \mathit{not}\ \mathit{MigratoryBird}(\mathit{Jo}) & \\
\mathit{MigratoryBird}(\mathit{Jo}) \leftarrow \mathit{not}\ \mathit{unsat}(5) & \\
:\sim \mathit{unsat}(5) & [1@0]
\end{array}$$

However, for any LP$^{\text{MLN}}$ program Π, this modification will make its translated ASP program (with weak constraints) unsatisfiable if there is no element in SM[Π] satisfying all hard rules in Π.

Example 2': *Consider Π' that is slightly modified from Π in Example 2 as follows:*

$$\begin{aligned}
\alpha :&\quad \mathit{Bird}(\mathit{Jo}) \leftarrow \mathit{ResidentBird}(\mathit{Jo})\\
\alpha :&\quad \mathit{Bird}(\mathit{Jo}) \leftarrow \mathit{MigratoryBird}(\mathit{Jo})\\
\alpha :&\quad \bot \leftarrow \mathit{ResidentBird}(\mathit{Jo}), \mathit{MigratoryBird}(\mathit{Jo})\\
\alpha :&\quad \mathit{ResidentBird}(\mathit{Jo})\\
\alpha :&\quad \mathit{MigratoryBird}(\mathit{Jo})
\end{aligned}$$

We know there exist three probabilistic stable models of Π' and no element in $\mathrm{SM}[\Pi']$ *satisfies all five hard rules in Π'. If we apply the simplification on the translation* $\mathrm{lpmln2wc}^{\mathrm{pnt,clingo}}(\Pi')$*, it yields*

$$\begin{aligned}
&\mathit{Bird}(\mathit{Jo}) \leftarrow \mathit{ResidentBird}(\mathit{Jo})\\
&\mathit{Bird}(\mathit{Jo}) \leftarrow \mathit{MigratoryBird}(\mathit{Jo})\\
&\bot \leftarrow \mathit{ResidentBird}(\mathit{Jo}), \mathit{MigratoryBird}(\mathit{Jo})\\
&\mathit{ResidentBird}(\mathit{Jo})\\
&\mathit{MigratoryBird}(\mathit{Jo})
\end{aligned}$$

which does not have a stable model. Thus we cannot apply this simplification when an $\mathrm{LP}^{\mathrm{MLN}}$ *program Π does not have a probabilistic stable model (or element in* $\mathrm{SM}[\Pi]$*) that satisfies all hard rules in Π.*

3.4 Proofs

3.4.1 Proof of Proposition 1

Proposition 1 *For any* $\mathrm{LP}^{\mathrm{MLN}}$ *program Π, the set* $\mathrm{SM}[\Pi]$ *is exactly the set of the stable models of* $\mathrm{lpmln2wc}(\Pi)$.

Proof. To prove **Proposition 1**, it is sufficient to prove

$$I \in \mathrm{SM}[\Pi] \text{ iff } I \text{ is a stable model of } \mathrm{lpmln2wc}(\Pi). \tag{3.3}$$

Since $I \in \mathrm{SM}[\Pi]$ iff I is a stable model of $\overline{\Pi_I}$, by definition, (3.3) is equivalent to saying

$$I \text{ is a minimal model of } \bigwedge_{w:F\in\Pi, I\models F} F^I \text{ iff } I \text{ is a minimal model of } \bigwedge_{w:F\in\Pi} (\{F\}^{\mathrm{ch}})^I,$$

which is true because

$$\bigwedge_{w:F\in\Pi} (\{F\}^{\mathrm{ch}})^I = \bigwedge_{w:F\in\Pi, I\models F} (\{F\}^{\mathrm{ch}})^I \wedge \bigwedge_{w:F\in\Pi, I\not\models F} (\{F\}^{\mathrm{ch}})^I = \bigwedge_{w:F\in\Pi, I\models F} F^I.$$

∎

3.4.2 Proof of Theorem 1

Let Π be an LP$^{\mathrm{MLN}}$ program. By Π^{soft} we denote the set of all soft rules in Π, by Π^{hard} we denote the set of all hard rules in Π.

For any $I \in \mathrm{SM}[\Pi]$, let

$$W_\Pi^{\mathrm{hard}}(I) = exp\Bigg(\sum_{w:F\ \in\ (\Pi^{\mathrm{hard}})_I} w\Bigg)$$

and

$$W_\Pi^{\mathrm{soft}}(I) = exp\Bigg(\sum_{w:F\ \in\ (\Pi^{\mathrm{soft}})_I} w\Bigg),$$

then I is a most probable stable model of Π iff

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

Let Π' be an ASP program with weak constraints such that $Level \in \{0,1\}$ for all weak constraints

$$:\sim F \quad [Weight\ @\ Level]$$

in Π'. I is an optimal stable model of Π' iff

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

Theorem 1 *For any* LP$^{\text{MLN}}$ *program* Π, *the most probable stable models of* Π *are precisely the optimal stable models of the program with weak constraints* lpmln2wc(Π).

Proof. Let Π' denote lpmln2wc(Π). To prove **Theorem 1**, it is sufficient to prove

I is a most probable stable model of Π iff I is an optimal stable model of Π'

which is equivalent to proving

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

iff

- $I \in \underset{J:\ J \in \underset{K:\ K \text{ is a stable model of } \Pi'}{\text{argmin}} \mathit{Penalty}_{\Pi'}(K,1)}{\text{argmin}} \mathit{Penalty}_{\Pi'}(J,0)$.

This is clear because

$$
\begin{aligned}
& \underset{J:\ J\in \underset{K:\ K\in \mathrm{SM}[\Pi]}{\mathrm{argmax}}\ W^{\text{hard}}_{\Pi}(K)}{\mathrm{argmax}} W^{\text{soft}}_{\Pi}(J) \\
= \;& \text{(by (3.3) and the definition of } W^{\text{hard}}_{\Pi}(I) \text{ and } W^{\text{soft}}_{\Pi}(I)) \\
& \underset{J:\ J\in \underset{K:\ K \text{ is a stable model of } \Pi'}{\mathrm{argmax}}\ exp\Big(\sum_{\alpha:F\ \in\ (\Pi^{\text{hard}})_K} \alpha\Big)}{\mathrm{argmax}} exp\Big(\sum_{w:F\ \in\ (\Pi^{\text{soft}})_J} w\Big) \\
= \;& \\
& \underset{J:\ J\in \underset{K:\ K \text{ is a stable model of } \Pi'}{\mathrm{argmax}}\ exp\Big(\sum_{\alpha:F\ \in\ \Pi^{\text{hard}}, K\models F} 1\Big)}{\mathrm{argmax}} exp\Big(\sum_{w:F\ \in\ \Pi^{\text{soft}}, J\models F} w\Big) \\
= \;& \\
& \underset{J:\ J\in \underset{K:\ K \text{ is a stable model of } \Pi'}{\mathrm{argmin}}\ \Big(\sum_{\alpha:F\ \in\ \Pi^{\text{hard}}, K\models F} -1\Big)}{\mathrm{argmin}} \Big(\sum_{w:F\ \in\ \Pi^{\text{soft}}, J\models F} -w\Big) \\
= \;& \\
& \underset{J:\ J\in \underset{K:\ K \text{ is a stable model of } \Pi'}{\mathrm{argmin}}\ \Big(\sum_{:\sim\ F[-1@1]\in\Pi', K\models F} -1\Big)}{\mathrm{argmin}} \Big(\sum_{:\sim\ F[-w@0]\in\Pi', J\models F} -w\Big) \\
= \;& \\
& \underset{J:\ J\in \underset{K:\ K \text{ is a stable model of } \Pi'}{\mathrm{argmin}}\ Penalty_{\Pi'}(K,1)}{\mathrm{argmin}} Penalty_{\Pi'}(J,0).
\end{aligned}
$$

▮

3.4.3 Proof of Corollary 1

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

- $W_{\Pi}(I) = W(\text{lpmln2wc}(\Pi), I)$, *and*
- $P_{\Pi}(I) = P(\text{lpmln2wc}(\Pi), I)$.

Proof. Let Π be an LP$^{\text{MLN}}$ program, and Π' be its translated logic program with weak constraints lpmln2wc(Π). For any interpretation I of Π, there are only two cases:

- $I \in \text{SM}[\Pi]$: By Proposition 1, I is a stable model of Π'.

$$
\begin{array}{ll}
& W(\Pi', I) \\
= & \text{(by definition)} \\
& exp\big(-Penalty_{\Pi'}(I,1) \times \alpha - Penalty_{\Pi'}(I,0)\big) \\
= & \text{(by the definition of } Penalty_{\Pi}(I,l)) \\
& exp\big(-(\sum_{:\sim F[-1@1]\in\Pi', I\models F} -1) \times \alpha - (\sum_{:\sim F[-w@0]\in\Pi', I\models F} -w)\big) \\
= & \\
& exp\big(-(\sum_{\alpha:F\,\in\,\Pi^{\text{hard}}, I\models F} -1) \times \alpha - (\sum_{w:F\,\in\,\Pi^{\text{soft}}, I\models F} -w)\big) \\
= & \\
& exp\big(\sum_{\alpha:F\,\in\,\Pi^{\text{hard}}, I\models F} \alpha + \sum_{w:F\,\in\,\Pi^{\text{soft}}, I\models F} w\big) \\
= & \\
& exp\big(\sum_{w:F\,\in\,\Pi_I} w\big) \\
= & \\
& W_{\Pi}(I)
\end{array}
$$

- $I \notin \text{SM}[\Pi]$: By Proposition 1, I is not a stable model of Π'. Thus $W(\Pi', I) = 0 = W_{\Pi}(I)$.

Thus for any interpretation I of Π, $W_{\Pi}(I) = W(\text{lpmln2wc}(\Pi), I)$.

It is also easy to see that

$$
\begin{aligned}
P(\Pi', I) = &\ \lim_{\alpha\to\infty} \frac{W(\Pi',I)}{\sum_{J \text{ is a stable model of } \Pi'} W(\Pi',J)} \\
= &\ \lim_{\alpha\to\infty} \frac{W_{\Pi}(I)}{\sum_{J\in\text{SM}[\Pi]} W_{\Pi}(J)} \\
= &\ P_{\Pi}(I).
\end{aligned}
$$

▮

3.4.4 Proof of Corollary 2

The proof of **Corollary 2** will use the following lemma, which is from Proposition 7 in (Ferraris, 2011).

Lemma 1 *Let Π be an ASP program, F be a propositional formula. An interpretation I is a stable model of Π such that $I \models F$ iff I is a stable model of $\Pi \wedge F^{constr}$.*

The proof of **Corollary 2** will also use the following lemma.

Lemma 2 *Let Π be an ASP program, F be a propositional formula. If I is a stable model of $\Pi \wedge F^{constr}$, then $W(\Pi, I) = W(\Pi \wedge F^{constr}, I)$.*

Proof. Let I be a stable model of $\Pi \wedge F^{constr}$. By **Lemma 1**, I is a stable model of Π. Since the weak constraints in Π and the weak constraints in $\Pi \wedge F^{constr}$ are the same, $W(\Pi, I) = W(\Pi \wedge F^{constr}, I)$.

∎

Corollary 2 *Let Π be an* LP$^{\text{MLN}}$ *program, Π' be the translated logic program with weak constraints* lpmln2wc(Π). *The probability of F_1 given F_2 (F_2 is satisfiable in Π) under Π is :*

$$P_\Pi(F_1 \mid F_2) = \lim_{\alpha \to \infty} \frac{Z(\Pi' \wedge F_1^{constr} \wedge F_2^{constr})}{Z(\Pi' \wedge F_2^{constr})}.$$

Proof. Let Π be an LP$^{\text{MLN}}$ program, and Π' be its translated logic program with weak constraints lpmln2wc(Π).

$$P_\Pi(F_1 \mid F_2) = \frac{P_\Pi(F_1 \wedge F_2)}{P_\Pi(F_2)} = \frac{\sum\limits_{I \models F_1 \wedge F_2} P_\Pi(I)}{\sum\limits_{I \models F_2} P_\Pi(I)} = \lim_{\alpha \to \infty} \frac{\sum\limits_{I \models F_1 \wedge F_2} W_\Pi(I)}{\sum\limits_{I \models F_2} W_\Pi(I)}$$

Further,

$$
\begin{aligned}
& P_{\Pi}(F_1 \mid F_2) \\
= \; & \lim_{\alpha\to\infty} \frac{\sum\limits_{I \in \mathrm{SM}[\Pi] \text{ and } I\models F_1\wedge F_2} W_{\Pi}(I)}{\sum\limits_{I \in \mathrm{SM}[\Pi] \text{ and } I\models F_2} W_{\Pi}(I)} \\
= \; & (\text{by } \textbf{Proposition 1} \text{ and } \textbf{Corollary 1}) \\
& \lim_{\alpha\to\infty} \frac{\sum\limits_{I \text{ is a stable model of } \Pi' \text{ and } I\models F_1\wedge F_2} W(\Pi',I)}{\sum\limits_{I \text{ is a stable model of } \Pi' \text{ and } I\models F_2} W(\Pi',I)} \\
= \; & (\text{By } \textbf{Lemma 1}) \\
& \lim_{\alpha\to\infty} \frac{\sum\limits_{I \text{ is a stable model of } \Pi' \wedge F_1^{constr} \wedge F_2^{constr}} W(\Pi',I)}{\sum\limits_{I \text{ is a stable model of } \Pi' \wedge F_2^{constr}} W(\Pi',I)} \\
= \; & (\text{By the definition of } Z(\Pi) \text{ and } \textbf{Lemma 2}) \\
& \lim_{\alpha\to\infty} \frac{Z(\Pi' \wedge F_1^{constr} \wedge F_2^{constr})}{Z(\Pi' \wedge F_2^{constr})}.
\end{aligned}
$$

▮

3.4.5 Proof of Proposition 2

Proposition 2 *For any Markov Logic Network program* Π*, the models of* Π *are precisely the stable models of the program with weak constraints* mln2wc(Π).

Proof. For any Markov Logic Network Π, we obtain an LP$^{\rm MLN}$ program Π' from Π by adding

$$\alpha : \{A\}^{\rm ch}$$

for every atom A in the signature of Π.

By Theorem 2 in (Lee and Wang, 2016), Π and Π' have the same probability distribution over all interpretations, and consequently, I is a model of Π iff $I \in \mathrm{SM}[\Pi']$.

By **Proposition 1**, $I \in \mathrm{SM}[\Pi']$ iff I is a stable model of lpmln2wc(Π'). Since

- I is a stable model of lpmln2wc(Π')

iff

- I is a stable model of $\text{lpmln2wc}(\Pi) \cup \{\{\{A\}^{\text{ch}}\}^{\text{ch}} \mid A \text{ is an atom in } \Pi\}$

iff (since for any interpretation I, the reduct of $\{\{A\}^{\text{ch}}\}^{\text{ch}}$ relative to I is equivalent to the reduct of $\{A\}^{\text{ch}}$ relative to I, $\text{lpmln2wc}(\Pi) \cup \{\{\{A\}^{\text{ch}}\}^{\text{ch}} \mid A \text{ is an atom in } \Pi\}$ is strongly equivalent to $\text{lpmln2wc}(\Pi) \cup \{\{A\}^{\text{ch}} \mid A \text{ is an atom in } \Pi\}$)

- I is a stable model of $\text{lpmln2wc}(\Pi) \cup \{\{A\}^{\text{ch}} \mid A \text{ is an atom in } \Pi\}$

iff

- I is a stable model of $\text{mln2wc}(\Pi)$,

then I is a model of Π iff I is a stable model of $\text{mln2wc}(\Pi)$.

$\blacksquare$

3.4.6 Proof of Theorem 2

Theorem 2 *For any Markov Logic Network* Π, *the most probable models of* Π *are precisely the optimal stable models of the program with weak constraints* $\text{mln2wc}(\Pi)$.

Proof. For any Markov Logic Network Π, we obtain an LP^{MLN} program Π' from Π by adding

$$\alpha : \{A\}^{\text{ch}}$$

for every atom A in Π. By Theorem 2 in (Lee and Wang, 2016), Π and Π' have the same probability distribution over all interpretations. Then for any interpretation I of Π,

- I is a most probable model of the MLN program Π

iff

- I is a most probable stable model of the LP$^{\text{MLN}}$ program Π'

iff (by **Theorem 1**)

- I is an optimal stable model of the ASP program with weak constraints lpmln2wc(Π')

iff (since a choice rule is always satisfied, omiting the weak constraint "$:\sim \{A\}^{\text{ch}}.\ [-1@1]$" for all atoms A in Π doesn't affect what is an optimal stable model of lpmln2wc(Π'))

- I is an optimal stable model of the ASP program with weak constraints lpmln2wc(Π) $\cup \{\{\{A\}^{\text{ch}}\}^{\text{ch}} \mid A$ is an atom in $\Pi\}$

iff (since lpmln2wc(Π) $\cup \{\{\{A\}^{\text{ch}}\}^{\text{ch}} \mid A$ is an atom in $\Pi\}$ is strongly equivalent to lpmln2wc(Π) $\cup \{\{A\}^{\text{ch}} \mid A$ is an atom in $\Pi\}$)

- I is an optimal stable model of the ASP program with weak constraints lpmln2wc(Π) $\cup \{\{A\}^{\text{ch}} \mid A$ is an atom in $\Pi\}$

Thus we proved I is a most probable model of an MLN program Π iff I is an optimal stable model of the ASP program with weak constraints mln2wc(Π).

∎

3.4.7 Proof of Corollary 3

Corollary 3 *For any Markov Logic Network Π and any interpretation I of Π,*

- $W^{\text{mln}}_{\Pi}(I) = W(\text{mln2wc}(\Pi), I)$, *and*

- $P^{\text{mln}}_{\Pi}(I) = P(\text{mln2wc}(\Pi), I)$.

Proof. Let Π be a Markov Logic Network, Π' be the translated logic program with weak constraints mln2wc(Π). By Π^{hard}, we denote all rules with infinite weight (α) in Π; and by Π^{soft}, we denote all rules whose weight is a real number in Π.

For any interpretation I of Π, I is always a model of Π, and by **Proposition 2**, I is a stable model of Π'.

For any interpretation I,

$$\begin{aligned}
& W(\Pi', I) \\
= \; & \text{(by definition)} \\
& exp\big(-Penalty_{\Pi'}(I,1) \times \alpha - Penalty_{\Pi'}(I,0)\big) \\
= \; & \text{(by the definition of } Penalty_{\Pi}(I,l)) \\
& exp\big(-(\sum_{:\sim F[-1@1]\in\Pi', I\vDash F} -1) \times \alpha - (\sum_{:\sim F[-w@0]\in\Pi', I\vDash F} -w)\big) \\
= \; & \\
& exp\big(-(\sum_{\alpha:F \,\in\, \Pi^{\text{hard}}, I\vDash F} -1) \times \alpha - (\sum_{w:F \,\in\, \Pi^{\text{soft}}, I\vDash F} -w)\big) \\
= \; & \\
& exp\big(\sum_{\alpha:F \,\in\, \Pi^{\text{hard}}, I\vDash F} \alpha + \sum_{w:F \,\in\, \Pi^{\text{soft}}, I\vDash F} w\big) \\
= \; & \\
& exp\big(\sum_{w:F \,\in\, \Pi, I\vDash F} w\big) \\
= \; & \\
& W_{\Pi}^{\text{mln}}(I)
\end{aligned}$$

Thus $W_{\Pi}^{\text{mln}}(I) = W(\text{mln2wc}(\Pi), I)$. Further, by definition, it is easy to check that $P_{\Pi}^{\text{mln}}(I) = P(\text{mln2wc}(\Pi), I)$.

▮

3.4.8 Proof of Corollary 4

Corollary 4 *Let* Π *be a Markov Logic Network,* Π' *be the translated logic program with weak constraints* mln2wc(Π)*. The probability of* F_1 *given* F_2 *(*F_2 *is satisfiable in* Π*) under* Π *is:*

$$P^{\rm mln}_{\Pi}(F_1 \mid F_2) = \lim_{\alpha\to\infty} \frac{Z(\Pi' \wedge F_1^{constr} \wedge F_2^{constr})}{Z(\Pi' \wedge F_2^{constr})}.$$

Proof. Let Π be a Markov Logic Network, and Π' be its translated logic program with weak constraints mln2wc(Π).

$$P^{\rm mln}_{\Pi}(F_1 \mid F_2) = \frac{P^{\rm mln}_{\Pi}(F_1 \wedge F_2)}{P^{\rm mln}_{\Pi}(F_2)} = \frac{\sum\limits_{I\models F_1\wedge F_2} P^{\rm mln}_{\Pi}(I)}{\sum\limits_{I\models F_2} P^{\rm mln}_{\Pi}(I)} = \lim_{\alpha\to\infty} \frac{\sum\limits_{I\models F_1\wedge F_2} W^{\rm mln}_{\Pi}(I)}{\sum\limits_{I\models F_2} W^{\rm mln}_{\Pi}(I)}$$

Further,

$$\begin{aligned}
& P^{\rm mln}_{\Pi}(F_1 \mid F_2) \\
= \ & \text{(by } \textbf{Corollary 3}\text{)} \\
& \lim_{\alpha\to\infty} \frac{\sum\limits_{I \text{ is a stable model of } \Pi' \text{ and } I\models F_1\wedge F_2} W(\Pi',I)}{\sum\limits_{I \text{ is a stable model of } \Pi' \text{ and } I\models F_2} W(\Pi',I)} \\
= \ & \text{(by } \textbf{Lemma 1}\text{)} \\
& \lim_{\alpha\to\infty} \frac{\sum\limits_{I \text{ is a stable model of } \Pi' \wedge F_1^{constr} \wedge F_2^{constr}} W(\Pi',I)}{\sum\limits_{I \text{ is a stable model of } \Pi' \wedge F_2^{constr}} W(\Pi',I)} \\
= \ & \text{(by the definition of } Z(\Pi) \text{ and } \textbf{Lemma 2}\text{)} \\
& \lim_{\alpha\to\infty} \frac{Z(\Pi' \wedge F_1^{constr} \wedge F_2^{constr})}{Z(\Pi' \wedge F_2^{constr})}.
\end{aligned}$$

▮

3.4.9 Proof of Corollary 5

Corollary 5 *For any* LP$^{\rm MLN}$ *program* Π*, the set* SM[Π] *is exactly the set of the stable models of the program with weak constraints* lpmln2wc$^{\rm pnt}$(Π).

Proof. For any LP$^{\text{MLN}}$ program Π, the logic program part (without weak constraints) of lpmln2wc$^{\text{pnt}}(\Pi)$ is exactly the same as that of lpmln2wc(Π), thus their stable models are the same. By **Proposition 1**, the set SM$[\Pi]$ is exactly the set of the stable models of the program with weak constraints lpmln2wc$^{\text{pnt}}(\Pi)$.

∎

3.4.10 Proof of Corollary 6

Corollary 6 *For any* LP$^{\text{MLN}}$ *program* Π *and any interpretation* I *of* Π,

- $W_\Pi(I) \propto W(\text{lpmln2wc}^{\text{pnt}}(\Pi), I)$, *and*
- $P_\Pi(I) = P(\text{lpmln2wc}^{\text{pnt}}(\Pi), I)$.

Proof. Let Π be an LP$^{\text{MLN}}$ program, and Π' be its translated logic program with weak constraints lpmln2wc$^{\text{pnt}}(\Pi)$. For any interpretation I of Π, there are only two cases:

- $I \in \text{SM}[\Pi]$: By **Corollary 5**, I is a stable model of Π'.

$$\begin{aligned}
& W(\Pi', I) \\
= \;& \text{(by definition)} \\
& exp\big(-\mathit{Penalty}_{\Pi'}(I,1)\times\alpha - \mathit{Penalty}_{\Pi'}(I,0)\big) \\
= \;& \text{(by the definition of } \mathit{Penalty}_{\Pi}(I,l)) \\
& exp\big(-(\sum_{:\sim\neg F[1@1]\in\Pi', I\models\neg F} 1)\times\alpha - (\sum_{:\sim\neg F[w@0]\in\Pi', I\models\neg F} w)\big) \\
= \;& \\
& exp\big(-(\sum_{\alpha:F\,\in\,\Pi^{\text{hard}}, I\not\models F} 1)\times\alpha - (\sum_{w:F\,\in\,\Pi^{\text{soft}}, I\not\models F} w)\big) \\
= \;& \\
& exp\big(\sum_{\alpha:F\,\in\,\Pi^{\text{hard}}, I\models F}\alpha + \sum_{w:F\,\in\,\Pi^{\text{soft}}, I\models F} w\big)\times exp\big(-\sum_{w:F\,\in\,\Pi^{\text{hard}}}\alpha - \sum_{w:F\,\in\,\Pi^{\text{soft}}} w\big) \\
\propto \;& \text{(since } exp\big(-\sum_{w:F\,\in\,\Pi^{\text{hard}}}\alpha - \sum_{w:F\,\in\,\Pi^{\text{soft}}} w\big) \text{ is fixed for } \Pi) \\
& exp\big(\sum_{w:F\,\in\,\Pi_I} w\big) \\
= \;& \\
& W_{\Pi}(I)
\end{aligned}$$

- $I \notin \text{SM}[\Pi]$: By **Corollary 5**, I is not a stable model of Π'. Thus $W(\Pi', I) = 0 = W_{\Pi}(I)$.

Thus for any interpretation I of Π, $W_{\Pi}(I) \propto W(\text{lpmln2wc}^{\text{pnt}}(\Pi), I)$. Further, by definition, it is easy to check that $P_{\Pi}(I) = P(\text{lpmln2wc}^{\text{pnt}}(\Pi), I)$.

▮

3.4.11 Proof of Corollary 7

Corollary 7 *Let* Π *be an* LP$^{\text{MLN}}$ *program,* Π' *be the translated logic program with weak constraints* lpmln2wc$^{\text{pnt}}(\Pi)$. *The probability of* F_1 *given* F_2 *(*F_2 *is satisfiable in*

Π) *under* Π *is:*

$$P_\Pi(F_1 \mid F_2) = \lim_{\alpha\to\infty} \frac{Z(\Pi' \wedge F_1^{constr} \wedge F_2^{constr})}{Z(\Pi' \wedge F_2^{constr})}.$$

Proof. Let Π be an LP$^{\rm MLN}$ program, and Π' be its translated logic program with weak constraints lpmln2wc$^{\rm pnt}(\Pi)$.

$$P_\Pi(F_1 \mid F_2) = \frac{P_\Pi(F_1 \wedge F_2)}{P_\Pi(F_2)} = \frac{\sum\limits_{I \models F_1 \wedge F_2} P_\Pi(I)}{\sum\limits_{I \models F_2} P_\Pi(I)} = \lim_{\alpha\to\infty} \frac{\sum\limits_{I \models F_1 \wedge F_2} W_\Pi(I)}{\sum\limits_{I \models F_2} W_\Pi(I)}$$

Further,

$$\begin{array}{rl} & P_\Pi(F_1 \mid F_2) \\ = & \\ & \lim\limits_{\alpha\to\infty} \frac{\sum\limits_{I\ \in\ \text{SM}[\Pi]\ \text{and}\ I \models F_1 \wedge F_2} W_\Pi(I)}{\sum\limits_{I\ \in\ \text{SM}[\Pi]\ \text{and}\ I \models F_2} W_\Pi(I)} \\ = & \text{(by } \textbf{Corollary 5} \text{ and } \textbf{Corollary 6}\text{)} \\ & \lim\limits_{\alpha\to\infty} \frac{\sum\limits_{I \text{ is a stable model of } \Pi' \text{ and } I \models F_1 \wedge F_2} W(\Pi', I)}{\sum\limits_{I \text{ is a stable model of } \Pi' \text{ and } I \models F_2} W(\Pi', I)} \\ = & \text{(by } \textbf{Lemma 1}\text{)} \\ & \lim\limits_{\alpha\to\infty} \frac{\sum\limits_{I \text{ is a stable model of } \Pi' \wedge F_1^{constr} \wedge F_2^{constr}} W(\Pi', I)}{\sum\limits_{I \text{ is a stable model of } \Pi' \wedge F_2^{constr}} W(\Pi', I)} \\ = & \text{(by the definition of } Z(\Pi) \text{ and } \textbf{Lemma 2}\text{)} \\ & \lim\limits_{\alpha\to\infty} \frac{Z(\Pi' \wedge F_1^{constr} \wedge F_2^{constr})}{Z(\Pi' \wedge F_2^{constr})}. \end{array}$$

∎

3.4.12 Proof of Corollary 8

Let σ and σ' be signatures such that $\sigma \subseteq \sigma'$. For any two interpretations I, J of the same signature σ', we write $I <^\sigma J$ iff

- $I|_\sigma \subsetneq J|_\sigma$, and

- I and J agree on $\sigma' \setminus \sigma$.

The proof of **Corollary 8** will use a restricted version of Theorem 1 from (Bartholomew and Lee, 2013), which is reformulated as follows:

Lemma 3 *Let F be a propositional formula. An interpretation I is a stable model of F relative to signature σ iff*

- $I \models F^I$,
- *and no interpretation J such that $J <^{\sigma} I$ satisfies F^I.*

The proof of **Corollary 8** will use a restricted version of the splitting theorem from (Ferraris *et al.*, 2009), which is reformulated as follows:

Splitting Theorem *Let Π_1, Π_2 be two finite ground programs, $\mathbf{p}$, $\mathbf{q}$ be disjoint tuples of distinct atoms. If*

- *each strongly connected component of the dependency graph of $\Pi_1 \cup \Pi_2$ w.r.t. $\mathbf{p} \cup \mathbf{q}$ is a subset of $\mathbf{p}$ or a subset of $\mathbf{q}$,*
- *no atom in $\mathbf{p}$ has a strictly positive occurrence in Π_2, and*
- *no atom in $\mathbf{q}$ has a strictly positive occurrence in Π_1,*

then an interpretation I of $\Pi_1 \cup \Pi_2$ is a stable model of $\Pi_1 \cup \Pi_2$ relative to $\mathbf{p} \cup \mathbf{q}$ if and only if I is a stable model of Π_1 relative to $\mathbf{p}$ and I is a stable model of Π_2 relative to $\mathbf{q}$.

Here and after, $w_i : F_i$ denotes the i-th rule in Π, where w_i could be α or a real number.

Corollary 8 *Let Π be an $\mathrm{LP}^{\mathrm{MLN}}$ program. There is a 1-1 correspondence ϕ between the set $\mathrm{SM}[\Pi]$ and the stable models of $\mathrm{lpmln2wc}^{\mathrm{pnt,rule}}(\Pi)$, where $\phi(I) = I \cup \{unsat(i) \mid w_i : F_i \in \Pi,\ I \not\models F_i\}$.*

Proof. Let σ be the signature of Π. We can check that the following mapping ϕ is a 1-1 correspondence:

$$\phi(I) = I \cup \{unsat(i) \mid w_i : F_i \in \Pi, I \not\models F_i\}$$

where $\phi(I)$ is of an extended signature $\sigma \cup \{unsat(i) \mid w_i : F_i \in \Pi\}$.

To prove ϕ is a 1-1 correspondence between the set SM[Π] and the set of the stable models of

$$\bigwedge_{w_i:\ F_i \in \Pi} \Big((F_i \leftarrow \neg unsat(i)) \wedge (unsat(i) \leftarrow \neg F_i) \Big), \tag{3.4}$$

it is sufficient to prove the following two bullets.

- **prove: for every interpretation $I \in$ SM[Π], $\phi(I)$ is a stable model of (3.4).**

  Assume $I \in$ SM[Π], by (3.3), I is a stable model of

  $$\bigwedge_{w_i:\ F_i \in \Pi} (F_i \leftarrow \neg\neg F_i).$$

  By **Lemma 3**, we know

  – $I \models$
  $$\bigwedge_{w_i:\ F_i \in \Pi, I \models F_i} \Big(F_i^I \Big), \tag{3.5}$$

  – and no interpretation K of signature σ such that $K <^\sigma I$ satisfies (3.5).

  To prove $\phi(I)$ is a stable model of (3.4), by the splitting theorem, it is sufficient to show

  – $\phi(I)$ is a stable model of $\bigwedge_{w_i:\ F_i \in \Pi} \Big(unsat(i) \leftarrow \neg\ F_i \Big)$ relative to $\{unsat(i) \mid w_i : F_i \in \Pi\}$, and

  – $\phi(I)$ is a stable model of $\bigwedge_{w_i:\ F_i \in \Pi} \Big(F_i \leftarrow \neg unsat(i) \Big)$ relative to σ;

which is equivalent to showing

(a) $\phi(I) \models \bigwedge_{w_i:\ F_i \in \Pi} \Big(unsat(i) \leftrightarrow \neg\, F_i\Big)$,

(b.1) $\phi(I) \models$

$$\bigwedge_{w_i:\ F_i \in \Pi, \phi(I) \models F_i} \Big(F_i \leftarrow \neg unsat(i)\Big)^{\phi(I)} \wedge \bigwedge_{w_i:\ F_i \in \Pi, \phi(I) \not\models F_i} \Big(F_i \leftarrow \neg unsat(i)\Big)^{\phi(I)}, \tag{3.6}$$

(b.2) and no interpretation L of signature $\sigma \cup \{unsat(i) \mid w_i : F_i \in \Pi\}$ such that $L <^\sigma \phi(I)$ satisfies (3.6).

It's clear that **(a)** is true by the definition of $\phi(I)$. As for **(b.1)**, since $\phi(I) \models (F_i \leftrightarrow \neg unsat(i))$ for all $w_i : F_i \in \Pi$, (3.6) is equivalent to

$$\bigwedge_{w_i:\ F_i \in \Pi, \phi(I) \models F_i} \Big(F_i^{\phi(I)} \leftarrow \top\Big) \wedge \bigwedge_{w_i:\ F_i \in \Pi, \phi(I) \not\models F_i} \Big(\bot \leftarrow \bot\Big).$$

Then **(b.1)** is equivalent to saying $\phi(I) \models$

$$\bigwedge_{w_i:\ F_i \in \Pi, \phi(I) \models F_i} \Big(F_i^{\phi(I)}\Big), \tag{3.7}$$

which is further equivalent to saying $I \models$ (3.5). As for **(b.2)**, assume for the sake of contradiction that there exists an interpretation L such that $L <^\sigma \phi(I)$ satisfies (3.6). Since (3.6) is equivalent to (3.7), $L \models \bigwedge_{w_i:\ F_i \in \Pi, \phi(I) \models F_i} \Big(F_i^{\phi(I)}\Big)$. Thus we know $L|_\sigma <^\sigma I$ and $L|_\sigma \models$ (3.5), which contradicts with "there is no interpretation K such that $K <^\sigma I$ satisfies (3.5)". So both **(b.1)** and **(b.2)** are true. Consequently, $\phi(I)$ is a stable model of (3.4).

- **prove: for every stable model J of (3.4), $J|_\sigma \in \text{SM}[\Pi]$ and $J = \phi(J|_\sigma)$.**

  Assume J is a stable model of (3.4), by the splitting theorem,

  - J is a stable model of $\bigwedge_{w_i:\ F_i \in \Pi} \Big(unsat(i) \leftarrow \neg\, F_i\Big)$ relative to $\{unsat(i) \mid w_i : F_i \in \Pi\}$, and

– J is a stable model of $\bigwedge_{w_i:\ F_i\in\Pi}\Big(F_i \leftarrow \neg unsat(i)\Big)$ relative to σ.

Thus we have

(c) $J \models unsat(i) \leftrightarrow \neg F_i$ for all $w_i : F_i \in \Pi$, which follows that $J = J|_\sigma \cup \{unsat(i) \mid w_i : F_i \in \Pi, J|_\sigma \models \neg\ F_i\}$. In other words, $J = \phi(J|_\sigma)$.

(d.1) Since $J \models (F_i \leftarrow \neg unsat(i))$ for all $w_i : F_i \in \Pi$, $J \models$

$$\bigwedge_{w_i:\ F_i\in\Pi}\Big(F_i^J \leftarrow (\neg unsat(i))^J\Big), \tag{3.8}$$

(d.2) and no interpretation L of signature $\sigma \cup \{unsat(i) \mid w_i : F_i \in \Pi\}$ such that $L <^\sigma J$ satisfies (3.8).

Since $J \models unsat(i) \leftrightarrow \neg F_i$ for all $w_i : F_i \in \Pi$, (3.8) is equivalent to

$$\bigwedge_{w_i:\ F_i\in\Pi, J|_\sigma\models F_i}\Big(F_i^J \leftarrow \top\Big) \wedge \bigwedge_{w_i:\ F_i\in\Pi, J|_\sigma\not\models F_i}\Big(\bot \leftarrow \bot\Big),$$

which is further equivalent to

$$\bigwedge_{w_i:\ F_i\in\Pi, J|_\sigma\models F_i}\Big(F_i^{J|_\sigma}\Big). \tag{3.9}$$

Thus by **(d.1)**, $J|_\sigma \models$ (3.9); and by **(d.2)**, it's easy to show that no interpretation K such that $K <^\sigma J|_\sigma$ satisfies (3.9). (Assume for the sake of contradiction, there exists an interpretation K such that $K <^\sigma J|_\sigma$ satisfies (3.9). Let $L = K \cup \{unsat(i) \mid w_i : F_i \in \Pi, J \models \neg\ F_i\}$. Then $L <^\sigma J$ and $L \models$ (3.9). Since (3.9) is equivalent to (3.8), $L \models$ (3.8), which contradicts with **(d.2)**.)

Then by **Lemma 3**, $J|_\sigma$ is a stable model of $\bigwedge_{w_i:\ F_i\in\Pi}(F_i \leftarrow \neg\neg F_i)$. By (3.3), $J|_\sigma \in \mathrm{SM}[\Pi]$.

∎

3.4.13 Proof of Corollary 9

Corollary 9 *For any* LP$^{\text{MLN}}$ *program* Π *and any interpretation* I *of* Π,

- $W_\Pi(I) \propto W(\text{lpmln2wc}^{\text{pnt,rule}}(\Pi), \phi(I))$, *and*
- $P_\Pi(I) = P(\text{lpmln2wc}^{\text{pnt,rule}}(\Pi), \phi(I))$,

where $\phi(I) = I \cup \{unsat(i) \mid w_i : F_i \in \Pi,\ I \not\models F_i\}$.

Proof. Let Π be an LP$^{\text{MLN}}$ program, and Π' be its translated logic program with weak constraints lpmln2wc$^{\text{pnt,rule}}(\Pi)$. For any interpretation I of Π, there are only two cases:

- $I \in \text{SM}[\Pi]$: By **Corollary 8**, $\phi(I)$ is a stable model of Π'.

$$\begin{array}{cl}
 & W(\Pi', \phi(I)) \\
= & \text{(by definition)} \\
 & exp\big(-Penalty_{\Pi'}(\phi(I), 1) \times \alpha - Penalty_{\Pi'}(\phi(I), 0)\big) \\
= & \text{(by the definition of } Penalty_\Pi(I, l)) \\
 & exp\big(-(\sum\limits_{:\sim unsat(i)[1@1]\in\Pi', \phi(I)\models unsat(i)} 1) \times \alpha - (\sum\limits_{:\sim unsat(i)[w_i@0]\in\Pi', \phi(I)\models unsat(i)} w_i)\big) \\
= & \\
 & exp\big(-(\sum\limits_{\alpha:F_i\ \in\ \Pi^{\text{hard}}, I\not\models F_i} 1) \times \alpha - (\sum\limits_{w:F_i\ \in\ \Pi^{\text{soft}}, I\not\models F_i} w_i)\big) \\
\propto & \text{(from the proof of } \textbf{Corollary 6}) \\
 & W_\Pi(I)
\end{array}$$

- $I \notin \text{SM}[\Pi]$: By **Corollary 8**, $\phi(I)$ is not a stable model of Π'. Thus $W(\Pi', \phi(I)) = 0 = W_\Pi(I)$.

Thus for any interpretation I of Π, $W_\Pi(I) \propto W(\text{lpmln2wc}^{\text{pnt,rule}}(\Pi), \phi(I))$. Further, by definition, it is easy to check that $P_\Pi(I) = P(\text{lpmln2wc}^{\text{pnt,rule}}(\Pi), \phi(I))$.

∎

3.4.14 Proof of Corollary 10

Corollary 10 *Let* Π *be an* LP^{MLN} *program,* Π' *be the translated logic program with weak constraints* $\text{lpmln2wc}^{\text{pnt,rule}}(\Pi)$*. The probability of* F_1 *given* F_2 *(*F_2 *is satisfiable in* Π*) under* Π *is:*

$$P_\Pi(F_1 \mid F_2) = \lim_{\alpha\to\infty} \frac{Z(\Pi' \wedge F_1^{constr} \wedge F_2^{constr})}{Z(\Pi' \wedge F_2^{constr})}.$$

Proof. Let Π be an LP^{MLN} program, and Π' be its translated logic program with weak constraints $\text{lpmln2wc}^{\text{pnt,rule}}(\Pi)$.

$$P_\Pi(F_1 \mid F_2) = \frac{P_\Pi(F_1 \wedge F_2)}{P_\Pi(F_2)} = \frac{\sum\limits_{I\vDash F_1\wedge F_2} P_\Pi(I)}{\sum\limits_{I\vDash F_2} P_\Pi(I)} = \lim_{\alpha\to\infty} \frac{\sum\limits_{I\vDash F_1\wedge F_2} W_\Pi(I)}{\sum\limits_{I\vDash F_2} W_\Pi(I)}$$

Further,

$$\begin{aligned}
& P_\Pi(F_1 \mid F_2) \\
= \; & \lim_{\alpha\to\infty} \frac{\sum\limits_{I\,\in\,\text{SM}[\Pi] \text{ and } I\vDash F_1\wedge F_2} W_\Pi(I)}{\sum\limits_{I\,\in\,\text{SM}[\Pi] \text{ and } I\vDash F_2} W_\Pi(I)} \\
= \; & (\text{by } \textbf{Corollary 8} \text{ and } \textbf{Corollary 9}) \\
& \lim_{\alpha\to\infty} \frac{\sum\limits_{\phi(I) \text{ is a stable model of } \Pi' \text{ and } \phi(I)\vDash F_1\wedge F_2} W(\Pi',\phi(I))}{\sum\limits_{\phi(I) \text{ is a stable model of } \Pi' \text{ and } \phi(I)\vDash F_2} W(\Pi',\phi(I))} \\
= \; & (\text{by } \textbf{Lemma 1}) \\
& \lim_{\alpha\to\infty} \frac{\sum\limits_{\phi(I) \text{ is a stable model of } \Pi' \wedge F_1^{constr} \wedge F_2^{constr}} W(\Pi',\phi(I))}{\sum\limits_{\phi(I) \text{ is a stable model of } \Pi' \wedge F_2^{constr}} W(\Pi',\phi(I))} \\
= \; & (\text{by the definition of } Z(\Pi) \text{ and } \textbf{Lemma 2}) \\
& \lim_{\alpha\to\infty} \frac{Z(\Pi'\wedge F_1^{constr}\wedge F_2^{constr})}{Z(\Pi'\wedge F_2^{constr})}.
\end{aligned}$$

∎

3.4.15 Proof of Corollary 11

Corollary 11 *Let Π be an* LP$^{\rm MLN}$ *program such that all unweighted rules of* Π *are in the rule form* $Head \leftarrow Body$. *There is a 1-1 correspondence* ϕ *between the set* SM[Π] *and the stable models of* lpmln2wc$^{\rm pnt,clingo}(\Pi)$, *where* $\phi(I) = I \cup \{unsat(i) \mid w_i : Head_i \leftarrow Body_i \in \Pi, I \models Body_i \wedge \neg Head_i\}$.

Proof. Let σ denote the signature of Π. By **Corollary 8**, it suffices to prove that for any interpretation I of the signature $\sigma \cup \{unsat(i) \mid w_i : Head_i \leftarrow Body_i \in \Pi\}$,

$$\begin{aligned} &I \text{ is a stable model of } \mathrm{lpmln2wc}^{\rm pnt,clingo}(\Pi) \\ &\text{iff } I \text{ is a stable model of } \mathrm{lpmln2wc}^{\rm pnt,rule}(\Pi). \end{aligned} \tag{3.10}$$

By the splitting theorem, it is equivalent to proving

(a) I is a stable model of $\bigwedge_{w_i:\ Head_i \leftarrow Body_i \in \Pi} \Big(unsat(i) \leftarrow Body_i \wedge \neg Head_i\Big)$ relative to $\{unsat(i) \mid w_i : Head_i \leftarrow Body_i \in \Pi\}$, and

(b) I is a stable model of $\bigwedge_{w_i:\ Head_i \leftarrow Body_i \in \Pi} \Big(Head_i \leftarrow Body_i \wedge \neg unsat(i)\Big)$ relative to σ;

iff

(c) I is a stable model of $\bigwedge_{w_i:\ Head_i \leftarrow Body_i \in \Pi} \Big(unsat(i) \leftarrow \neg(Head_i \leftarrow Body_i)\Big)$ relative to $\{unsat(i) \mid w_i : Head_i \leftarrow Body_i \in \Pi\}$, and

(d) I is a stable model of $\bigwedge_{w_i:\ Head_i \leftarrow Body_i \in \Pi} \Big((Head_i \leftarrow Body_i) \leftarrow \neg unsat(i)\Big)$ relative to σ.

This is clear because

- **(a)** and **(c)** are equivalent to saying $I \models \bigwedge_{w_i:\ Head_i \leftarrow Body_i \in \Pi} \Big(unsat(i) \leftrightarrow Body_i \wedge \neg Head_i\Big)$ (by completion), and

- **(b)** and **(d)** are equivalent because $Head_i \leftarrow Body_i \wedge \neg unsat(i)$ is strongly equivalent to $(Head_i \leftarrow Body_i) \leftarrow \neg unsat(i)$. It is because for any interpretation J,

$$\big((Head_i \leftarrow Body_i) \leftarrow \neg unsat(i)\big)^J$$

$$=$$

$$\begin{cases} (Head_i \leftarrow Body_i)^J \leftarrow (\neg unsat(i))^J & \text{if } J \models Head_i \vee \neg Body_i \vee unsat(i), \\ \bot & \text{otherwise;} \end{cases}$$

$$=$$

$$\begin{cases} (Head_i^J \leftarrow Body_i^J) \leftarrow (\neg unsat(i))^J & \text{if } J \models Head_i \vee \neg Body_i, \\ \bot \leftarrow \bot & \text{if } J \not\models Head_i \vee \neg Body_i \text{ and } J \models unsat(i), \\ \bot & \text{otherwise;} \end{cases}$$

$$\Leftrightarrow$$

$$\begin{cases} Head_i^J \leftarrow \big(Body_i^J \wedge (\neg unsat(i))^J\big) & \text{if } J \models Head_i \vee \neg Body_i, \\ Head_i^J \leftarrow \big(Body_i^J \wedge (\neg unsat(i))^J\big) & \text{if } J \not\models Head_i \vee \neg Body_i \text{ and } J \models unsat(i), \\ \bot & \text{otherwise;} \end{cases}$$

$$=$$

$$\begin{cases} Head_i^J \leftarrow \big(Body_i^J \wedge (\neg unsat(i))^J\big) & \text{if } J \models Head_i \vee \neg Body_i \vee unsat(i), \\ \bot & \text{otherwise;} \end{cases}$$

$$=$$

$$\begin{cases} Head_i^J \leftarrow \big(Body_i \wedge (\neg unsat(i))\big)^J & \text{if } J \models Head_i \vee \neg Body_i \vee unsat(i), \\ \bot & \text{otherwise;} \end{cases}$$

$$=$$

$$\big(Head_i \leftarrow Body_i \wedge \neg unsat(i)\big)^J.$$

By Proposition 5 from (Ferraris, 2011), $Head_i \leftarrow Body_i \wedge \neg unsat(i)$ is strongly equivalent to $(Head_i \leftarrow Body_i) \leftarrow \neg unsat(i)$.

∎

3.4.16 Proof of Corollary 12

Corollary 12 *For any* LP^{MLN} *program* Π *and any interpretation* I *of* Π*,*

- $W_\Pi(I) \propto W(\text{lpmln2wc}^{\text{pnt,clingo}}(\Pi), \phi(I))$*, and*
- $P_\Pi(I) = P(\text{lpmln2wc}^{\text{pnt,clingo}}(\Pi), \phi(I))$*,*

where $\phi(I) = I \cup \{unsat(i) \mid w_i : Head_i \leftarrow Body_i \in \Pi, I \vDash Body_i \wedge \neg Head_i\}$.

Proof.

Let Π be an LP^{MLN} program. We know that the weak constraints of $\text{lpmln2wc}^{\text{pnt,clingo}}(\Pi)$ are exactly the same as the weak constraints of $\text{lpmln2wc}^{\text{pnt,rule}}(\Pi)$. By (3.10), for any interpretation I of Π

$$\begin{array}{rl} & W(\text{lpmln2wc}^{\text{pnt,clingo}}(\Pi), \phi(I)) \\ = & (\text{by the definition of } W(\Pi, I)) \\ & W(\text{lpmln2wc}^{\text{pnt,rule}}(\Pi), \phi(I)) \\ \propto & (\text{by } \textbf{Corollary 9}) \\ & W_\Pi(I) \end{array}$$

Further, by definition, it is easy to check that $P_\Pi(I) = P(\text{lpmln2wc}^{\text{pnt,clingo}}(\Pi), \phi(I))$.

∎

3.4.17 Proof of Corollary 13

Corollary 13 *Let* Π *be an* LP^{MLN} *program,* Π' *be the translated logic program with weak constraints* $\text{lpmln2wc}^{\text{pnt,clingo}}(\Pi)$*. The probability of* F_1 *given* F_2 *(*F_2 *is* *satisfiable in* Π*) under* Π *is:*

$$P_\Pi(F_1 \mid F_2) = \lim_{\alpha\to\infty} \frac{Z(\Pi' \wedge F_1^{constr} \wedge F_2^{constr})}{Z(\Pi' \wedge F_2^{constr})}.$$

Proof. Let Π be an LP$^{\text{MLN}}$ program, and Π' be its translated logic program with weak constraints lpmln2wc$^{\text{pnt,clingo}}(\Pi)$.

$$P_\Pi(F_1 \mid F_2) = \frac{P_\Pi(F_1 \wedge F_2)}{P_\Pi(F_2)} = \frac{\sum\limits_{I \vDash F_1 \wedge F_2} P_\Pi(I)}{\sum\limits_{I \vDash F_2} P_\Pi(I)} = \lim_{\alpha\to\infty} \frac{\sum\limits_{I \vDash F_1 \wedge F_2} W_\Pi(I)}{\sum\limits_{I \vDash F_2} W_\Pi(I)}$$

Further,

$$\begin{aligned}
& P_\Pi(F_1 \mid F_2) \\
= \; & \\
& \lim_{\alpha\to\infty} \frac{\sum\limits_{I \,\in\, \text{SM}[\Pi] \text{ and } I \vDash F_1 \wedge F_2} W_\Pi(I)}{\sum\limits_{I \,\in\, \text{SM}[\Pi] \text{ and } I \vDash F_2} W_\Pi(I)} \\
= \; & (\text{By } \textbf{Corollary 11} \text{ and } \textbf{Corollary 12}) \\
& \lim_{\alpha\to\infty} \frac{\sum\limits_{\phi(I) \text{ is a stable model of } \Pi' \text{ and } \phi(I) \vDash F_1 \wedge F_2} W(\Pi', \phi(I))}{\sum\limits_{\phi(I) \text{ is a stable model of } \Pi' \text{ and } \phi(I) \vDash F_2} W(\Pi', \phi(I))} \\
= \; & (\text{By } \textbf{Lemma 1}) \\
& \lim_{\alpha\to\infty} \frac{\sum\limits_{\phi(I) \text{ is a stable model of } \Pi' \wedge F_1^{constr} \wedge F_2^{constr}} W(\Pi', \phi(I))}{\sum\limits_{\phi(I) \text{ is a stable model of } \Pi' \wedge F_2^{constr}} W(\Pi', \phi(I))} \\
= \; & (\text{By the definition of } Z(\Pi) \text{ and } \textbf{Lemma 2}) \\
& \lim_{\alpha\to\infty} \frac{Z(\Pi' \wedge F_1^{constr} \wedge F_2^{constr})}{Z(\Pi' \wedge F_2^{constr})}.
\end{aligned}$$

∎

3.4.18 Proof of Corollary 14

Lemma 4 *For any interpretation* I *of an* LP$^{\text{MLN}}$ *program* Π*, let* Π^{constr} *denote a set of weighted rules of the form* $w : \bot \leftarrow F$*, where* w *is* α *or a real number,* F *is a first-order formula. Then* $I \in \text{SM}[\Pi \cup \Pi^{\text{constr}}]$ *iff* $I \in \text{SM}[\Pi]$*.*

Proof.

- $I \in \text{SM}[\Pi \cup \Pi^{\text{constr}}]$

iff (by definition)

- I is a stable model of $\overline{\Pi_I} \wedge \bigwedge_{\substack{w:\ \bot \leftarrow F \in \Pi^{\text{constr}} \\ I \models \bot \leftarrow F}} \Big(\bot \leftarrow F\Big)$

iff (by theorem 3 in (Ferraris *et al.*, 2011))

- I is a stable model of $\overline{\Pi_I}$ and $I \models \bigwedge_{\substack{w:\ \bot \leftarrow F \in \Pi^{\text{constr}} \\ I \models \bot \leftarrow F}} \Big(\bot \leftarrow F\Big)$

iff (since $I \models \bigwedge_{\substack{w:\ \bot \leftarrow F \in \Pi^{\text{constr}} \\ I \models \bot \leftarrow F}} \Big(\bot \leftarrow F\Big)$ is always true)

- $I \in \text{SM}[\Pi]$.

∎

Corollary 14 *Let Π be an LP^{MLN} program such that all unweighted rules of Π are in the rule form $Head \leftarrow Body$.There is a 1-1 correspondence ϕ between the set* $\text{SM}[\Pi]$ *and the stable models of* $\text{lpmln2wc}^{\text{pnt,clingo}}_{\text{simp}}(\Pi)$, *where $\phi(I) = I \cup \{unsat(i) \mid w_i : Head_i \leftarrow Body_i \in \Pi, Head_i$ is not $\bot, I \models Body_i \wedge \neg Head_i\}$.*

Proof.

We can check that the following mapping ϕ is a 1-1 correspondence:

$$\phi(I) = I \cup \{unsat(i) \mid w_i : Head_i \leftarrow Body_i \in \Pi, Head_i \text{ is not } \bot, I \models Body_i \wedge \neg Head_i\},$$

where $\phi(I)$ is of an extended signature $\sigma \cup \{unsat(i) \mid w_i : Head_i \leftarrow Body_i \in \Pi, Head_i \text{ is not } \bot\}$.

By **Lemma 4**, we know $I \in \text{SM}[\Pi]$ iff $I \in \text{SM}[\bigwedge_{\substack{w_i:\ Head_i \leftarrow Body_i \in \Pi \\ Head_i \text{ is not } \bot}} \Big(w_i : Head_i \leftarrow Body_i\Big)]$.

By **Corollary 11**, we know ϕ is a 1-1 correspondence between the set

$$\text{SM}[\bigwedge_{\substack{w_i:\ Head_i \leftarrow Body_i \in \Pi \\ Head_i \text{ is not } \bot}} \Big(w_i:\ Head_i \leftarrow Body_i\Big)]$$

and the set of the stable models of

$$\bigwedge_{\substack{w_i:\ Head_i \leftarrow Body_i \in \Pi \\ \text{and } Head_i \text{ is not } \bot}} \Big((unsat(i) \leftarrow Body_i \wedge \neg Head_i) \wedge (Head_i \leftarrow Body_i \wedge \neg unsat(i))\Big),$$

where $\phi(I) = I \cup \{unsat(i) \mid w_i : Head_i \leftarrow Body_i \in \Pi, Head_i \text{ is not } \bot, I \models Body_i \wedge \neg Head_i\}$.

Thus ϕ is a 1-1 correspondence between the set SM[Π] and the set of the stable models of $\text{lpmln2wc}^{\text{pnt,clingo}}_{\text{simp}}(\Pi)$.

▮

3.4.19 Proof of Corollary 15

Corollary 15 *For any* LP$^{\text{MLN}}$ *program* Π *and any interpretation* I *of* Π,

- $W_\Pi(I) \propto W(\text{lpmln2wc}^{\text{pnt,clingo}}_{\text{simp}}(\Pi), \phi(I))$, *and*
- $P_\Pi(I) = P(\text{lpmln2wc}^{\text{pnt,clingo}}_{\text{simp}}(\Pi), \phi(I))$,

where $\phi(I) = I \cup \{unsat(i) \mid w_i : Head_i \leftarrow Body_i \in \Pi, Head_i \textit{ is not } \bot, I \models Body_i \wedge \neg Head_i\}$.

Proof. Let Π be an LP$^{\text{MLN}}$ program, Π' be its translated logic program with weak constraints $\text{lpmln2wc}^{\text{pnt,clingo}}_{\text{simp}}(\Pi)$, Π_{c11} be $\text{lpmln2wc}^{\text{pnt,clingo}}(\Pi)$, and ϕ_{c11} be the 1-1 correspondence in **Corollary 11**.

We can prove that for any interpretation $I \in \text{SM}[\Pi]$ and $l \in \{0, 1\}$

$$Penalty_{\Pi'}(\phi(I), l) = Penalty_{\Pi_{c11}}(\phi_{c11}(I), l). \tag{3.11}$$

This is clear because

$$
\begin{array}{ll}
& \mathit{Penalty}_{\Pi'}(\phi(I),0) \\
= & \\
& \displaystyle\sum_{\substack{:\sim\, unsat(i)[w_i@0]\in\Pi' \text{ and } \phi(I)\models unsat(i) \\ \text{or } :\sim Body_i[w_i@0]\in\Pi' \text{ and } \phi(I)\models Body_i}} w_i \\
= & \text{(since, by the definition of } \phi(I)\text{, when } Head_i \text{ is not } \bot\text{,} \\
& \phi(I)\models \mathit{unsat(i)} \text{ iff } I\models Body_i\wedge\neg Head_i) \\
& \displaystyle\sum_{\substack{w_i:Head_i\leftarrow Body_i\,\in\,\Pi^{\text{soft}},\ Head_i \text{ is not } \bot,\text{ and } I\models Body_i\wedge\neg Head_i \\ \text{or } w_i:Head_i\leftarrow Body_i\,\in\,\Pi^{\text{soft}},\ Head_i \text{ is } \bot,\text{ and } I\models Body_i\wedge\neg Head_i}} w_i \\
= & \\
& \displaystyle\sum_{w_i:Head_i\leftarrow Body_i\,\in\,\Pi^{\text{soft}} \text{ and } I\models Body_i\wedge\neg Head_i} w_i \\
= & \text{(since } \phi_{c11}(I)\models unsat(i) \text{ iff } I\models Body_i\wedge\neg Head_i) \\
& \displaystyle\sum_{:\sim\, unsat(i)[w_i@0]\in\Pi_{c11} \text{ and } \phi_{c11}(I)\models unsat(i)} w_i \\
= & \\
& \mathit{Penalty}_{\Pi_{c11}}(\phi_{c11}(I),0) \qquad ;
\end{array}
$$

and similarly,

$$\mathit{Penalty}_{\Pi'}(\phi(I),1) = \mathit{Penalty}_{\Pi_{c11}}(\phi_{c11}(I),1).$$

Then for any interpretation I of Π,

$$\begin{array}{rl}
& W(\Pi', \phi(I)) \\
= & \text{(by definition)} \\
& exp\big(-Penalty_{\Pi'}(\phi(I), 1) \times \alpha - Penalty_{\Pi'}(\phi(I), 0)\big) \\
= & \text{(by (3.11))} \\
& exp\big(-Penalty_{\Pi_{c11}}(\phi_{c11}(I), 1) \times \alpha - Penalty_{\Pi_{c11}}(\phi_{c11}(I), 0)\big) \\
= & \\
& W(\Pi_{c11}, \phi_{c11}(I)) \\
\propto & \text{(by } \textbf{Corollary 12}\text{)} \\
& W_\Pi(I)
\end{array}$$

Further, by definition, it is easy to check that $P_\Pi(I) = P(\text{lpmln2wc}^{\text{pnt,clingo}}_{\text{simp}}(\Pi), \phi(I))$.

∎

3.4.20 Proof of Corollary 16

Corollary 16 *Let* Π *be an* LP^{MLN} *program,* Π' *be the translated logic program with weak constraints* $\text{lpmln2wc}^{\text{pnt,clingo}}_{\text{simp}}(\Pi)$*. The probability of* F_1 *given* F_2 *(*F_2 *is satisfiable in* Π*) under* Π *is:*

$$P_\Pi(F_1 \mid F_2) = \lim_{\alpha\to\infty} \frac{Z(\Pi' \wedge F_1^{constr} \wedge F_2^{constr})}{Z(\Pi' \wedge F_2^{constr})}.$$

Proof. Let Π be an LP^{MLN} program, and Π' be its translated logic program with weak constraints $\text{lpmln2wc}^{\text{pnt}}(\Pi)$.

$$P_\Pi(F_1 \mid F_2) = \frac{P_\Pi(F_1 \wedge F_2)}{P_\Pi(F_2)} = \frac{\sum\limits_{I \vDash F_1 \wedge F_2} P_\Pi(I)}{\sum\limits_{I \vDash F_2} P_\Pi(I)} = \lim_{\alpha\to\infty} \frac{\sum\limits_{I \vDash F_1 \wedge F_2} W_\Pi(I)}{\sum\limits_{I \vDash F_2} W_\Pi(I)}$$

Further,

$$
\begin{aligned}
& P_\Pi(F_1 \mid F_2) \\
= \; & \lim_{\alpha\to\infty} \frac{\sum\limits_{I\,\in\,\mathrm{SM}[\Pi] \text{ and } I\models F_1\wedge F_2} W_\Pi(I)}{\sum\limits_{I\,\in\,\mathrm{SM}[\Pi] \text{ and } I\models F_2} W_\Pi(I)} \\
= \; & (\text{by } \textbf{Corollary 14} \text{ and } \textbf{Corollary 15}) \\
& \lim_{\alpha\to\infty} \frac{\sum\limits_{\phi(I) \text{ is a stable model of } \Pi' \text{ and } \phi(I)\models F_1\wedge F_2} W(\Pi',\phi(I))}{\sum\limits_{\phi(I) \text{ is a stable model of } \Pi' \text{ and } \phi(I)\models F_2} W(\Pi',\phi(I))} \\
= \; & (\text{by } \textbf{Lemma 1}) \\
& \lim_{\alpha\to\infty} \frac{\sum\limits_{\phi(I) \text{ is a stable model of } \Pi' \wedge F_1^{constr} \wedge F_2^{constr}} W(\Pi',\phi(I))}{\sum\limits_{\phi(I) \text{ is a stable model of } \Pi' \wedge F_2^{constr}} W(\Pi',\phi(I))} \\
= \; & (\text{by the definition of } Z(\Pi) \text{ and } \textbf{Lemma 2}) \\
& \lim_{\alpha\to\infty} \frac{Z(\Pi'\wedge F_1^{constr}\wedge F_2^{constr})}{Z(\Pi'\wedge F_2^{constr})}.
\end{aligned}
$$

∎

Chapter 4

TURNING P-LOG INTO LP$^{\text{MLN}}$

In this chapter, we show how P-log can be completely characterized in LP$^{\text{MLN}}$.

Let Π be a P-log program of signature $\sigma_1 \cup \sigma_2$ where

$$\Pi = \langle \mathbf{R}, \mathbf{S}, \mathbf{P}, \mathbf{Obs}, \mathbf{Act} \rangle,$$

σ_1 and σ_2 are defined in Section 2.3, we define translation plog2lpmln(Π) that turns Π into an LP$^{\text{MLN}}$ program.

4.1 Signature of plog2lpmln(Π)

Before introducing the signature of plog2lpmln(Π), we need to define some notations that will be used in the signature. For any real number $p \in [0,1]$ and $b \in \{\mathbf{t}, \mathbf{f}\}$, we define $[p]^b$ as follows:

$$[p]^b = \begin{cases} p & \text{if } b = \mathbf{t}; \\ 0 & \text{if } b = \mathbf{f}. \end{cases}$$

Further, for any $c(\vec{u})$ in $\mathbf{S}$ of Π, we define the set of all possible remaining (unassigned) probabilities of $c(\vec{u})$ in Π, $\mathbf{p}_{rem}(c(\vec{u}), \Pi)$, as

$$\{p \mid p = 1 - \sum_{p_i:\ pr_r(c(\vec{u})=v_i \ \mid \ C_i)=p_i \in \Pi} [p_i]^{b_i}\ ,\ \text{where each } b_i \in \{\mathbf{t}, \mathbf{f}\}\}.$$

The signature of plog2lpmln(Π) is

$$\sigma_1 \cup \sigma_2 \cup \{Intervene(c(\vec{u})) \mid c(\vec{u}) \text{ is an attribute occurring in } \mathbf{S}\} \cup \sigma_3,$$

where σ_3 is a propositional signature constructed from Π as follows:

$$\begin{aligned}
\sigma_3 = \ & \{Poss_r(c(\vec{u})\!=\!v) \mid r \text{ is a random selection rule for } c(\vec{u}) \text{ in } \Pi \text{ and } v \in Range(c)\} \\
& \cup \{PossWithAssPr_{r,C}(c(\vec{u})\!=\!v) \mid \text{there is a pr-atom } pr_r(c(\vec{u}) = v \mid C) = p \text{ in } \Pi\} \\
& \cup \{AssPr_{r,C}(c(\vec{u})\!=\!v) \mid \text{there is a pr-atom } pr_r(c(\vec{u}) = v \mid C) = p \text{ in } \Pi\} \\
& \cup \{PossWithAssPr(c(\vec{u})\!=\!v) \mid \text{there is a random selection rule for } c(\vec{u}) \text{ in } \Pi \\
& \qquad\qquad \text{and } v \in Range(c)\} \\
& \cup \{PossWithDefPr(c(\vec{u})\!=\!v) \mid \text{there is a random selection rule for } c(\vec{u}) \text{ in } \Pi \\
& \qquad\qquad \text{and } v \in Range(c)\} \\
& \cup \{NumDefPr(c(\vec{u}), m) \mid \text{there is a random selection rule for } c(\vec{u}) \text{ in } \Pi \\
& \qquad\qquad \text{and } m \in \{1, \dots, |Range(c)|\}\} \\
& \cup \{RemPr(c(\vec{u}), k) \mid \text{there is a random selection rule for } c(\vec{u}) \text{ in } \Pi \\
& \qquad\qquad \text{and } k \in \mathbf{p}_{rem}(c(\vec{u}), \Pi)\} \\
& \cup \{TotalDefPr(c(\vec{u}), k) \mid \text{there is a random selection rule for } c(\vec{u}) \text{ in } \Pi, \\
& \qquad\qquad k \in \mathbf{p}_{rem}(c(\vec{u}), \Pi), \text{ and } k > 0\}.
\end{aligned}$$

4.2 Translation plog2lpmln

We define translation plog2lpmln(Π) that turns a P-log program Π into an LP$^{\text{MLN}}$ program in a modular way. First, every rule

$$R$$

in $\tau(\Pi)$ (used in defining the possible worlds in P-log) is turned into a hard rule

$$\alpha : R$$

in plog2lpmln(Π).

In addition, plog2lpmln(Π) contains the following rules to associate probability to each possible world of Π. Below x, y denote schematic variables, and W is a possible world of Π.

Possible Atoms: For each random selection rule (2.2) for $c(\vec{u})$ in **S** and for each $v \in \mathit{Range}(c)$, plog2lpmln(Π) includes

$$\mathit{Poss}_r(c(\vec{u}) = v) \leftarrow \mathit{Body}, p(v), \mathit{not}\ \mathit{Intervene}(c(\vec{u})) \tag{4.1}$$

Rule (4.1) expresses that $c(\vec{u}) = v$ is possible in W due to r if $W \models \mathit{Body} \wedge p(v) \wedge \neg \mathit{Intervene}(c(\vec{u}))$.

Assigned Probability: For each pr-atom (2.3) in **P**, plog2lpmln(Π) contains the following rules:

$$\alpha: \ \mathit{PossWithAssPr}_{r,C}(c(\vec{u}){=}v) \leftarrow$$
$$\mathit{Poss}_r(c(\vec{u}) = v), C \tag{4.2}$$
$$\alpha: \ \mathit{AssPr}_{r,C}(c(\vec{u}){=}v) \leftarrow$$
$$c(\vec{u}){=}v, \mathit{PossWithAssPr}_{r,C}(c(\vec{u}){=}v) \tag{4.3}$$
$$\mathit{ln}(p): \ \bot \leftarrow \mathit{not}\ \mathit{AssPr}_{r,C}(c(\vec{u}){=}v) \qquad (p > 0) \tag{4.4}$$
$$\alpha: \ \bot \leftarrow \mathit{AssPr}_{r,C}(c(\vec{u}){=}v) \qquad (p = 0)$$
$$\alpha: \ \mathit{PossWithAssPr}(c(\vec{u}){=}v) \leftarrow \mathit{PossWithAssPr}_{r,C}(c(\vec{u}){=}v).$$

Rule (4.2) expresses the condition under which pr-atom (2.3) is applied in a possible world W. Further, if $c(\vec{u}){=}v$ is true in W as well, rules (4.3) and (4.4) contribute the assigned probability $e^{\mathit{ln}(p)} = p$ to the unnormalized probability of W as a factor when $p > 0$.

Denominator for Default Probability: For each random selection rule (2.2) for $c(\vec{u})$ in **S** and for each $v \in \mathit{Range}(c)$, plog2lpmln(Π) includes

$$\begin{aligned}\alpha :\ & \mathit{PossWithDefPr}(c(\vec{u})\!=\!v) \leftarrow \\ & \quad \mathit{Poss}_r(c(\vec{u})\!=\!v), \mathit{not}\ \mathit{PossWithAssPr}(c(\vec{u})\!=\!v)\end{aligned} \tag{4.5}$$

$$\begin{aligned}\alpha :\quad & \mathit{NumDefPr}(c(\vec{u}), x) \leftarrow \\ & \quad c(\vec{u})\!=\!v, \mathit{PossWithDefPr}(c(\vec{u}) = v), \\ & \quad x = \#\mathit{count}\{y : \mathit{PossWithDefPr}(c(\vec{u})\!=\!y)\}\end{aligned} \tag{4.6}$$

$$\begin{aligned}\mathit{ln}(\tfrac{1}{m}) :\ & \bot \leftarrow \mathit{not}\ \mathit{NumDefPr}(c(\vec{u}), m) \\ & \qquad\qquad (m = 2, \dots, |\mathit{Range}(c)|)\end{aligned} \tag{4.7}$$

Rule (4.5) asserts that $c(\vec{u}) = v$ is possible in W with a default probability if it is possible in W and not possible with an assigned probability. Rule (4.6) expresses, intuitively, that $\mathit{NumDefPr}(c(\vec{u}), x)$ is true if there are exactly x different values v such that $c(\vec{u}) = v$ is possible in W with a default probability, and there is at least one of them that is also true in W. This value x is the denominator of (2.4). Then rule (4.7) contributes the factor $1/x$ to the unnormalized probability of W as a factor.

Numerator for Default Probability:

- Consider each random selection rule $[r]\ \ \mathit{random}(c(\vec{u}) : \{x : p(x)\}) \leftarrow \mathit{Body}$ for $c(\vec{u})$ in $\mathbf{S}$ along with all pr-atoms associated with it in $\mathbf{P}$:

  $$\begin{aligned}& pr_r(c(\vec{u})\!=\!v_1 \mid C_1) = p_1 \\ & \dots \\ & pr_r(c(\vec{u})\!=\!v_n \mid C_n) = p_n\end{aligned}$$

  where $n \geq 1$, and v_i and v_j $(i \neq j)$ may be equal. For each $v \in \mathit{Range}(c)$, plog2lpmln(Π) contains the following rules: [1]

[1]The sum aggregate can be represented by ground first-order formulas under the stable model semantics under the assumption that the Herbrand Universe is finite (Ferraris, 2011). In the general case, it can be represented by generalized quantifiers (Lee and Meng, 2012) or infinitary propositional formulas (Harrison *et al.*, 2014). In the input language of ASP solvers, which does not allow real number arguments, p_i can be approximated to integers of some fixed interval.

$$\begin{aligned} \alpha : \quad & \mathit{RemPr}(c(\vec{u}), 1-y) \leftarrow \mathit{Body} \\ & \quad c(\vec{u})=v, \mathit{PossWithDefPr}(c(\vec{u})=v), \\ & \quad y = \#\text{sum}\{p_1 : \mathit{PossWithAssPr}_{r,C_1}(c(\vec{u})=v_1); \\ & \qquad \ldots; p_n : \mathit{PossWithAssPr}_{r,C_n}(c(\vec{u})=v_n)\}. \end{aligned} \tag{4.8}$$

$$\alpha : \quad \mathit{TotalDefPr}(c(\vec{u}), x) \leftarrow \mathit{RemPr}(c(\vec{u}), x), x > 0 \tag{4.9}$$

$$\mathit{ln}(x) : \quad \bot \leftarrow \mathit{not}\ \ \mathit{TotalDefPr}(c(\vec{u}), x) \tag{4.10}$$

$$\alpha : \quad \bot \leftarrow \mathit{RemPr}(c(\vec{u}), x), x \leq 0. \tag{4.11}$$

In rule (4.8), y is the sum of all assigned probabilities. Rules (4.9) and (4.10) are to account for the numerator of (2.4) when $n > 0$. The variable x stands for the numerator of (2.4). Rule (4.11) is to avoid assigning a non-positive default probability to a possible world.

Note that most rules in plog2lpmln(Π) are hard rules. The soft rules (4.4), (4.7), (4.10) cannot be simplified as atomic facts, e.g., $\mathit{ln}(\frac{1}{m})\ :\ \ \mathit{NumDefPr}(c(\vec{u}), m)$ in place of (4.7), which is in contrast with the use of probabilistic choice atoms in the distribution semantics based probabilistic logic programming language, such as ProbLog. This is related to the fact that the probability of each atom to happen in a possible word in P-log is derived from assigned and default probabilities, and not from independent probabilistic choices like the other probabilistic logic programming languages. In conjunction with the embedding of ProbLog in LP$^{\text{MLN}}$ (Lee and Wang, 2016), it is interesting to note that both kinds of probabilities can be captured in LP$^{\text{MLN}}$ using different kinds of rules.

Clearly, the signature of plog2lpmln(Π) is a superset of the signature of Π. Further, plog2lpmln(Π) is linear-time constructible. The following theorem tells us that there is a 1-1 correspondence between the set of the possible worlds with non-zero probabilities of Π and the set of the probabilistic stable models of plog2lpmln(Π) such that each stable model is an extension of the possible world, and the probability of each possible world of Π coincides with the probability of the corresponding stable model of plog2lpmln(Π).

Theorem 3 *Let Π be a consistent P-log program. There is a 1-1 correspondence ϕ between the set of the possible worlds of Π with non-zero probabilities and the set of probabilistic stable models of* plog2lpmln(Π) *such that*

- *For every possible world W of Π that has a non-zero probability, $\phi(W)$ is a probabilistic stable model of* plog2lpmln(Π)*, and $\mu_{\Pi}(W) = P_{\text{plog2lpmln}(\Pi)}(\phi(W))$.*
- *For every probabilistic stable model I of* plog2lpmln(Π)*, the restriction of I onto the signature of $\tau(\Pi)$, denoted $I|_{\sigma(\tau(\Pi))}$, is a possible world of Π and $\mu_{\Pi}(I|_{\sigma(\tau(\Pi))}) > 0$.*

The following mapping ϕ is such a 1-1 correspondence:

1. $\phi(W) \models \mathit{Poss}_r(c(\vec{u})\!=\!v)$ iff $c(\vec{u})\!=\!v$ is possible in W due to r.
2. For each pr-atom $pr_r(c(\vec{u})\!=\!v \mid C) = p$ in Π,

   $\phi(W) \models \mathit{PossWithAssPr}_{r,C}(c(\vec{u})\!=\!v)$ iff this pr-atom is applied in W.
3. For each pr-atom $pr_r(c(\vec{u})\!=\!v \mid C) = p$ in Π,

   $\phi(W) \models \mathit{AssPr}_{r,C}(c(\vec{u})\!=\!v)$ iff this pr-atom is applied in W, and $W \models c(\vec{u})\!=\!v$.
4. $\phi(W) \models \mathit{PossWithAssPr}(c(\vec{u})\!=\!v)$ iff $v \in AV_W(c(\vec{u}))$.
5. $\phi(W) \models \mathit{PossWithDefPr}(c(\vec{u}) = v)$ iff $c(\vec{u}) = v$ is possible in W and $v \notin AV_W(c(\vec{u}))$.

6. $\phi(W) \models \mathit{NumDefPr}(c(\vec{u}), m)$ iff there exist exactly m different values v such that $c(\vec{u})=v$ is possible in W; $v \notin AV_W(c(\vec{u}))$; and, for one of such v, $W \models c(\vec{u})=v$.

7. $\phi(W) \models \mathit{RemPr}(c(\vec{u}), k)$ iff there exists a value v such that $W \models c(\vec{u}) = v$; $c(\vec{u})=v$ is possible in W; $v \notin AV_W(c(\vec{u}))$; and

$$k = 1 - \sum_{v \in AV_W(c(\vec{u}))} \mathit{PossWithAssPr}(W, c(\vec{u})=v).$$

8. $\phi(W) \models \mathit{TotalDefPr}(c(\vec{u}), k)$ iff $\phi(W) \models \mathit{RemPr}(c(\vec{u}), k)$ and $k > 0$.

Example 6 *Consider a variant of the Monty Hall Problem encoding in P-log from (Baral* et al.*, 2009) to illustrate the probabilistic nonmonotonicity in the presence of assigned probabilities. There are four doors, behind which are three goats and one car. The guest picks door 1, and Monty, the show host who always opens one of the doors with a goat, opens door 2. Further, while the guest and Monty are unaware, the statistics is that in the past, with 30% chance the prize was behind door 1, and with 20% chance, the prize was behind door 3. Is it still better to switch to another door? This example can be formalized in P-log program* Π*, using both assigned probability and default probability, as*

$$\begin{array}{l}
\sim\mathit{CanOpen}(d) \leftarrow \mathit{Selected}=d. \quad (d \in \{1,2,3,4\}),\\
\sim\mathit{CanOpen}(d) \leftarrow \mathit{Prize}=d.\\
\mathit{CanOpen}(d) \leftarrow \mathit{not}\ \ \sim\mathit{CanOpen}(d).\\
\mathit{random}(\mathit{Prize}). \qquad \mathit{random}(\mathit{Selected}).\\
\mathit{random}(\mathit{Open} : \{x : \mathit{CanOpen}(x)\}).\\
\mathit{pr}(\mathit{Prize}=1) = 0.3. \ \ \mathit{pr}(\mathit{Prize}=3) = 0.2.\\
\mathit{Obs}(\mathit{Selected}=1). \ \ \mathit{Obs}(\mathit{Open}=2). \ \ \mathit{Obs}(\mathit{Prize} \neq 2).
\end{array}$$

The possible worlds of Π *are as follows:*

- $W_1 = \{Obs(Selected = 1), Obs(Open = 2), Obs(Prize \neq 2), Selected{=}1, Open{=}2, Prize{=}1, CanOpen(1) = \mathbf{f}, CanOpen(2) = \mathbf{t}, CanOpen(3) = \mathbf{t}, CanOpen(4) = \mathbf{t}\}$

- $W_2 = \{Obs(Selected = 1), Obs(Open = 2), Obs(Prize \neq 2), Selected{=}1, Open{=}2, Prize{=}3, CanOpen(1) = \mathbf{f}, CanOpen(2) = \mathbf{t}, CanOpen(3) = \mathbf{f}, CanOpen(4) = \mathbf{t}\}$

- $W_3 = \{Obs(Selected = 1), Obs(Open = 2), Obs(Prize \neq 2), Selected{=}1, Open{=}2, Prize{=}4, CanOpen(1) = \mathbf{f}, CanOpen(2) = \mathbf{t}, CanOpen(3) = \mathbf{t}, CanOpen(4) = \mathbf{f}\}$.

The probability of each atom to happen is

$$P(W_i, Selected{=}1) = PossWithDefPr(W, Selected{=}1) = 1/4$$

$$P(W_1, Open{=}2) = PossWithDefPr(W_1, Open{=}2) = 1/3$$
$$P(W_2, Open{=}2) = PossWithDefPr(W_2, Open{=}2) = 1/2$$
$$P(W_3, Open{=}2) = PossWithDefPr(W_3, Open{=}2) = 1/2$$

$$P(W_1, Prize{=}1) = PossWithAssPr(W_1, Prize{=}1) = 0.3$$
$$P(W_2, Prize{=}3) = PossWithAssPr(W_2, Prize{=}3) = 0.2$$
$$P(W_3, Prize{=}4) = PossWithDefPr(W_3, Prize{=}4) = 0.25$$

So,

- $\hat{\mu}_\Pi(W_1) = 1/4 \times 1/3 \times 0.3 = 1/40$

- $\hat{\mu}_\Pi(W_2) = 1/4 \times 1/2 \times 0.2 = 1/40$

- $\hat{\mu}_\Pi(W_3) = 1/4 \times 1/2 \times 0.25 = 1/32.$

Thus, in comparison with staying (W_1), switching to door 3 (W_2) does not affect the chance, but switching to door 4 (W_3) increases the chance by 25%.

After the translation, the LP^{MLN} *program* $\text{plog2lpmln}(\Pi)$ *is as follows:*

$//**** \; \tau(\Pi) \; ****$

$\alpha: \;\; {\sim}CanOpen(d) \leftarrow Selected = d$

$\alpha: \;\; {\sim}CanOpen(d) \leftarrow Prize = d$

$\alpha: \;\; CanOpen(d) \leftarrow not \; {\sim}CanOpen(d)$

$\alpha: \;\; \leftarrow CanOpen(d), {\sim}CanOpen(d)$

$\alpha: \;\; \leftarrow Prize = d_1, Prize = d_2, d_1 \neq d_2$

$\alpha: \;\; \leftarrow Selected = d_1, Selected = d_2, d_1 \neq d_2$

$\alpha: \;\; \leftarrow Open = d_1, Open = d_2, d_1 \neq d_2$

$\alpha: \;\; Prize = 1; Prize = 2; Prize = 3; Prize = 4 \leftarrow not \; Intervene(Prize)$

$\alpha: \;\; Selected = 1; Selected = 2; Selected = 3; Selected = 4 \leftarrow not \; Intervene(Selected)$

$\alpha: \;\; Open = 1; Open = 2; Open = 3; Open = 4 \leftarrow not \; Intervene(Open)$

$\alpha: \;\; \leftarrow Open = d, not \; CanOpen(d), not \; Intervene(Open)$

$\alpha: \;\; Obs(Selected = 1)$

$\alpha: \;\; \leftarrow Obs(Selected = 1), not \; Selected = 1$

$\alpha: \;\; Obs(Open = 2)$

$\alpha: \;\; \leftarrow Obs(Open = 2), not \; Open = 2$

$\alpha: \;\; Obs(Prize \neq 2)$

$\alpha: \;\; \leftarrow Obs(Prize \neq 2), Prize = 2$

$//**** \; Possible \; Atoms ****$

$\alpha: \;\; Poss(Prize = d) \leftarrow not \; Intervene(Prize)$

$\alpha: \;\; Poss(Selected = d) \leftarrow not \; Intervene(Selected)$

$\alpha: \;\; Poss(Open = d) \leftarrow CanOpen(d), not \; Intervene(Open)$

$//****\ \mathit{Assigned\ Probability} ****$

$\alpha:\ \mathit{PossWithAssPr}(\mathit{Prize}=1) \leftarrow \mathit{Poss}(\mathit{Prize}=1)$

$\alpha:\ \mathit{AssPr}(\mathit{Prize}=1) \leftarrow \mathit{Prize}=1, \mathit{PossWithAssPr}(\mathit{Prize}=1)$

$ln(0.3):\ \bot \leftarrow not\ \mathit{AssPr}(\mathit{Prize}=1)$

$\alpha:\ \mathit{PossWithAssPr}(\mathit{Prize}=3) \leftarrow \mathit{Poss}(\mathit{Prize}=3)$

$\alpha:\ \mathit{AssPr}(\mathit{Prize}=3) \leftarrow \mathit{Prize}=3, \mathit{PossWithAssPr}(\mathit{Prize}=3)$

$ln(0.2):\ \bot \leftarrow not\ \mathit{AssPr}(\mathit{Prize}=3)$

$//****\ \mathit{Denominator\ for\ Default\ Probability} ****$

$\alpha:\ \mathit{PossWithDefPr}(\mathit{Prize}=d) \leftarrow \mathit{Poss}(\mathit{Prize}=d), not\ \mathit{PossWithAssPr}(\mathit{Prize}=d)$

$\alpha:\ \mathit{PossWithDefPr}(\mathit{Selected}=d) \leftarrow \mathit{Poss}(\mathit{Selected}=d), not\ \mathit{PossWithAssPr}(\mathit{Selected}=d)$

$\alpha:\ \mathit{PossWithDefPr}(\mathit{Open}=d) \leftarrow \mathit{Poss}(\mathit{Open}=d), not\ \mathit{PossWithAssPr}(\mathit{Open}=d)$

$\alpha:\ \mathit{NumDefPr}(\mathit{Prize},x) \leftarrow \mathit{Prize}=d, \mathit{PossWithDefPr}(\mathit{Prize}=d),$
$\qquad x=\#count\{y: \mathit{PossWithDefPr}(\mathit{Prize}=y)\}$

$\alpha:\ \mathit{NumDefPr}(\mathit{Selected},x) \leftarrow \mathit{Selected}=d, \mathit{PossWithDefPr}(\mathit{Selected}=d),$
$\qquad x=\#count\{y: \mathit{PossWithDefPr}(\mathit{Selected}=y)\}$

$\alpha:\ \mathit{NumDefPr}(\mathit{Open},x) \leftarrow \mathit{Open}=d, \mathit{PossWithDefPr}(\mathit{Open}=d),$
$\qquad x=\#count\{y: \mathit{PossWithDefPr}(\mathit{Open}=y)\}$

$ln(\frac{1}{m}):\ \leftarrow not\ \mathit{NumDefPr}(c,m) \qquad //c \in \{\mathit{Prize}, \mathit{Selected}, \mathit{Open}\}, m \in \{2,3,4\}$

$//****\ \mathit{Numerator\ for\ Default\ Probability} ****$

$\alpha:\ \mathit{RemPr}(\mathit{Prize},1-x) \leftarrow \mathit{Prize}=d, \mathit{PossWithDefPr}(\mathit{Prize}=d),$
$\qquad x=\#\mathrm{sum}\{0.3: \mathit{PossWithAssPr}(\mathit{Prize}=1); 0.2: \mathit{PossWithAssPr}(\mathit{Prize}=3)\}$

$\alpha:\ \mathit{TotalDefPr}(\mathit{Prize},x) \leftarrow \mathit{RemPr}(\mathit{Prize},x), x>0$

$ln(x):\ \bot \leftarrow not\ \mathit{TotalDefPr}(\mathit{Prize},x)$

$\alpha:\ \bot \leftarrow \mathit{RemPr}(\mathit{Prize},x), x \leq 0$

Note that in part "Assigned Probability", we simplified slightly not to distinguish $\mathit{PossWithAssPr}(\cdot)$ and $\mathit{PossWithAssPr}_{r,C}(\cdot)$ because there is only one random selection rule for Prize and both pr-atoms for Prize has empty conditions.

plog2lpmln(Π) *has three probabilistic stable models I_1, I_2, and I_3, each of which is an extension of W_1, W_2, and W_3 respectively, and satisfies the following atoms:*

$$\begin{array}{ll}
\mathit{Poss}(\mathit{Prize}{=}i) & \text{for } i = 1,2,3,4; \\
\mathit{Poss}(\mathit{Selected}{=}i) & \text{for } i = 1,2,3,4; \\
\mathit{PossWithAssPr}(\mathit{Prize}{=}i) & \text{for } i = 1,3; \\
\mathit{PossWithDefPr}(\mathit{Prize}{=}i) & \text{for } i = 2,4; \\
\mathit{PossWithDefPr}(\mathit{Selected}{=}i) & \text{for } i = 1,2,3,4; \\
\mathit{NumDefPr}(\mathit{Selected},4). &
\end{array}$$

In addition,

- $I_1 \models \{\mathit{AssPr}(\mathit{Prize}{=}1),$
  $\mathit{Poss}(\mathit{Open}{=}2), \mathit{Poss}(\mathit{Open}{=}3), \mathit{Poss}(\mathit{Open}{=}4),$
  $\mathit{PossWithDefPr}(\mathit{Open}{=}2), \mathit{PossWithDefPr}(\mathit{Open}{=}3),$
  $\mathit{PossWithDefPr}(\mathit{Open}{=}4), \mathit{NumDefPr}(\mathit{Open},3)\}$

- $I_2 \models \{\mathit{AssPr}(\mathit{Prize}{=}3),$
  $\mathit{Poss}(\mathit{Open}{=}2), \mathit{Poss}(\mathit{Open}{=}4),$
  $\mathit{PossWithDefPr}(\mathit{Open}{=}2), \mathit{PossWithDefPr}(\mathit{Open}{=}4), \mathit{NumDefPr}(\mathit{Open},2)\}$

- $I_3 \models \{\mathit{NumDefPr}(\mathit{Prize},2), \mathit{RemPr}(\mathit{Prize},0.5), \mathit{TotalDefPr}(\mathit{Prize},0.5)$
  $\mathit{Poss}(\mathit{Open}{=}2), \mathit{Poss}(\mathit{Open}{=}3),$
  $\mathit{PossWithDefPr}(\mathit{Open}{=}2), \mathit{PossWithDefPr}(\mathit{Open}{=}3), \mathit{NumDefPr}(\mathit{Open},2)\}.$

The unnormalized weight $W_{\Pi'}(I_i)$ of each probabilistic stable model I_i is shown below, where $w(\mathit{AssPr}_{r,C}(c(\vec{u}) = v))$ denotes the exponentiated weight of rule (4.4),

$w(\mathit{NumDefPr}(c(\vec{u}), m))$ *denotes the exponentiated weight of rule* (4.7), *and* $w(\mathit{TotalDefPr}(c(\vec{u}), x))$ *denotes the exponentiated weight of rule* (4.10).

- $W_{\Pi'}(I_1) = w(\mathit{NumDefPr}(\mathit{Selected}, 4)) \times w(\mathit{AssPr}(\mathit{Prize}\!=\!1))$
  $\times\ w(\mathit{NumDefPr}(\mathit{Open}, 3))$
  $= \frac{1}{4} \times \frac{3}{10} \times \frac{1}{3} = \frac{1}{40}$;
- $W_{\Pi'}(I_2) = w(\mathit{NumDefPr}(\mathit{Selected}, 4)) \times w(\mathit{AssPr}(\mathit{Prize}\!=\!3))$
  $\times\ w(\mathit{NumDefPr}(\mathit{Open}, 2))$
  $= \frac{1}{4} \times \frac{2}{10} \times \frac{1}{2} = \frac{1}{40}$;
- $W_{\Pi'}(I_3) = w(\mathit{NumDefPr}(\mathit{Selected}, 4)) \times w(\mathit{NumDefPr}(\mathit{Open}, 2))$
  $\times\ w(\mathit{NumDefPr}(\mathit{Prize}, 2) \times w(\mathit{TotalDefPr}(\mathit{Prize}, 0.5)$
  $= \frac{1}{4} \times \frac{1}{2} \times \frac{1}{2} \times \frac{5}{10} = \frac{1}{32}$.

We can check that $I_i = \phi(W_i)$ *and* $\mu_\Pi(W_i) = P_{\text{plog2lpmln}(\Pi)}(\phi(W_i))$. *Consequently,* ϕ *is a 1-1 correspondence between the possible world of* Π *and the probabilistic stable models of* plog2lpmln(Π), *and* $\mu_\Pi(W_i) = P_{\text{plog2lpmln}(\Pi)}(\phi(W_i))$.

Combining the translations plog2lpmln and lpmln2wc, one can compute P-log MAP inference using standard ASP solvers.

4.3 Proofs

4.3.1 Proof of Theorem 3

Lemma 5 *(proposition 2 in Lee and Wang (2016)) If* $\mathrm{SM}'[\Pi]$ *is not empty, for every interpretation* I *of* Π, $P'_\Pi(I)$ *coincides with* $P_\Pi(I)$.

It follows from **Lemma 5** that if $\mathrm{SM}'[\Pi]$ is not empty, then

- I is a probabilistic stable model of Π iff $I \in \text{SM}'[\Pi]$,
- every probabilistic stable model of Π should satisfy all hard rules in Π.

Theorem 3 *Let Π be a consistent P-log program, σ be the signature of $\tau(\Pi)$. There is a 1-1 correspondence ϕ between the set of the possible worlds of Π with non-zero probabilities and the set of probabilistic stable models of* plog2lpmln(Π) *such that*

(a) *For every possible world W of Π that has a non-zero probability, $\phi(W)$ is a probabilistic stable model of* plog2lpmln(Π), *and $\mu_\Pi(W) = P_{\text{plog2lpmln}(\Pi)}(\phi(W))$.*

(b) *For every probabilistic stable model I of* plog2lpmln(Π), *$I|_\sigma$ is a possible world of Π, $I = \phi(I|_\sigma)$, and $\mu_\Pi(I|_\sigma) > 0$.*

Note that we make **(b)** a little bit more stronger than the statement in the the previous section by adding "$I = \phi(I|_\sigma)$" (which is already covered by "1-1 correspondence"). In this case, to prove **Theorem 3**, it is sufficient to prove **(a)** and **(b)**.

Proof. For any possible world W of a P-log program Π, we define the mapping ϕ as follows.

1. $\phi(W) \models \mathit{Poss}_r(c(\vec{u}){=}v)$ iff $c(\vec{u}){=}v$ is possible in W due to r.
2. For each pr-atom $pr_r(c(\vec{u}){=}v \mid C) = p$ in Π,

   $\phi(W) \models \mathit{PossWithAssPr}_{r,C}(c(\vec{u}){=}v)$ iff this pr-atom is applied in W.
3. For each pr-atom $pr_r(c(\vec{u}){=}v \mid C) = p$ in Π,

   $\phi(W) \models \mathit{AssPr}_{r,C}(c(\vec{u}){=}v)$ iff this pr-atom is applied in W, and $W \models c(\vec{u}){=}v$.
4. $\phi(W) \models \mathit{PossWithAssPr}(c(\vec{u}){=}v)$ iff $v \in AV_W(c(\vec{u}))$.

5. $\phi(W) \models \mathit{PossWithDefPr}(c(\vec{u}) = v)$ iff $c(\vec{u}) = v$ is possible in W and $v \notin AV_W(c(\vec{u}))$.

6. $\phi(W) \models \mathit{NumDefPr}(c(\vec{u}), m)$ iff there exist exactly m different values v such that $c(\vec{u}) = v$ is possible in W; $v \notin AV_W(c(\vec{u}))$; and, for one of such v, $W \models c(\vec{u}) = v$.

7. $\phi(W) \models \mathit{RemPr}(c(\vec{u}), k)$ iff there exists a value v such that $W \models c(\vec{u}) = v$; $c(\vec{u}) = v$ is possible in W; $v \notin AV_W(c(\vec{u}))$; and
$k = 1 - \sum_{v \in AV_W(c(\vec{u}))} \mathit{PossWithAssPr}(W, c(\vec{u}) = v)$.

8. $\phi(W) \models \mathit{TotalDefPr}(c(\vec{u}), k)$ iff $\phi(W) \models \mathit{RemPr}(c(\vec{u}), k)$ and $k > 0$.

Let's denote plog2lpmln(Π) as Π'. In the following two parts, we will prove each of the two bullets of **Theorem 3**.

(a) For every possible world W of Π with a non-zero probability, to prove $\phi(W)$ is a probabilistic stable model of Π', it is sufficient to prove $\phi(W)$ is a stable model of $\overline{\Pi'^{\text{ hard}}}$. Indeed, if we prove $\phi(W)$ is a stable model of $\overline{\Pi'^{\text{ hard}}}$, then $\phi(W)$ is a stable model of $\overline{\Pi'^{\text{ hard}}_{\phi(W)}}$ and $P_{\Pi'^{\text{ hard}}}(\phi(W))$ is always greater than 0. Consequently, $\phi(W)$ must be a probabilistic stable model of $\Pi'^{\text{ hard}}$. Since $\Pi' = \Pi'^{\text{ hard}} \cup \Pi'^{\text{ soft}}$, and $\Pi'^{\text{ soft}}$ is a set of soft rules of the form "$w : \leftarrow \mathit{not}\ A$", where A is an atom and w is a real number, by **Lemma 4** (it follows from **Lemma 4** that, if all w in Π^{constr} are real numbers, I is a probabilistic stable model of $\Pi \cup \Pi^{\text{constr}}$ iff I is a probabilistic stable model of Π), $\phi(W)$ is a probabilistic stable model of Π'.

Let σ denote the signature of $\tau(\Pi)$, $\Pi_{AUX} = \overline{\Pi'^{\text{ hard}}} \setminus \tau(\Pi)$. It can be seen that no atom in σ has a strictly positive occurrence in Π_{AUX}, and no atom in σ_3 has a strictly positive occurrence in $\tau(\Pi)$. Furthermore, the construction of Π'

guarantees that all loops of size greater than one involves atoms in σ only. So each strongly connected component of the dependency graph of $\overline{\Pi'^{\text{ hard}}}$ relative to $\sigma \cup \sigma_3$ is a subset of σ or a subset of σ_3. By the splitting theorem, it is equivalent to show that $\phi(W)$ is a stable model of $\tau(\Pi)$ relative to σ and $\phi(W)$ is a stable model of Π_{AUX} relative to σ_3.

- $\phi(W)$ **is a stable model of** $\tau(\Pi)$ **relative to** σ : Since W is a possible world of Π, W is a stable model of $\tau(\Pi)$ relative to σ. Since $\phi(W)$ is an extension of W and no atom in $\phi(W) \setminus W$ belongs to σ, $\phi(W)$ is a stable model of $\tau(\Pi)$ relative to σ.
- $\phi(W)$ **is a stable model of** Π_{AUX} **relative to** σ_3 : Since there is no loop of size greater than one in Π_{AUX}, we could apply completion on it. Let $Comp[\Pi_{AUX}; \sigma_3]$ denote the program obtained by applying completion on Π_{AUX} with respect to σ_3, which is as follows:
  - For each random selection rule (2.2) for $c(\vec{u})$, for each $v \in Range(c)$ and $x \in \{2, \dots, |Range(c)|\}$, $Comp[\Pi_{AUX}; \sigma_3]$ contains:

$$Poss_r(c(\vec{u}) = v) \leftrightarrow Body \wedge p(v) \wedge \neg Intervene(c(\vec{u})) \tag{4.12}$$

$$PossWithDefPr(c(\vec{u}) = v) \leftrightarrow \neg PossWithAssPr(c(\vec{u}) = v) \wedge \bigvee_{\substack{\text{r' :}\\ [r']\ random(c(\vec{u}):\{X:p(X)\})\leftarrow Body\in\Pi}} Poss_{r'}(c(\vec{u}) = v) \tag{4.13}$$

$$NumDefPr(c(\vec{u}), x) \leftrightarrow x = \#count\{y : PossWithDefPr(c(\vec{u})=y)\} \wedge \bigvee_{z\in Range(c)} \Big(c(\vec{u}) = z \wedge PossWithDefPr(c(\vec{u}) = z)\Big) \tag{4.14}$$

  - For each random selection rule (2.2) for $c(\vec{u})$ along with all pr-atoms associated with it in **P**:

$$\begin{aligned} & pr_r(c(\vec{u})\!=\!v_1 \mid C_1) = p_1 \\ & \dots \\ & pr_r(c(\vec{u})\!=\!v_n \mid C_n) = p_n \end{aligned} \tag{4.15}$$

where $n \geq 1$, for $i \in \{1, \dots, n\}$, $Comp[\Pi_{AUX}; \sigma_3]$ also contains:

$$PossWithAssPr_{r,C_i}(c(\vec{u}) = v_i) \leftrightarrow Poss_r(c(\vec{u}) = v_i) \wedge C_i \tag{4.16}$$

$$AssPr_{r,C_i}(c(\vec{u}) = v_i) \leftrightarrow c(\vec{u}) = v_i \wedge PossWithAssPr_{r,C_i}(c(\vec{u}) = v_i) \tag{4.17}$$

$$\neg AssPr_{r,C_i}(c(\vec{u}) = v_i) \qquad (\text{if } p_i = 0) \tag{4.18}$$

$$PossWithAssPr(c(\vec{u}) = v_i) \leftrightarrow \bigvee_{\substack{\text{r', j :}\\ pr_{r'}(c(\vec{u})=v_i|C_j)=p_j \in \Pi}} PossWithAssPr_{r',C_j}(c(\vec{u}) = v_i) \tag{4.19}$$

– For each $c(\vec{u})$ in **S** and $x \in \mathbf{p}_{rem}(c(\vec{u}), \Pi)$, $Comp[\Pi_{AUX}; \sigma_3]$ also contains:

$$\begin{aligned} RemPr(c(\vec{u}), x) \leftrightarrow & \bigvee_{v \in Range(c)} \Big(c(\vec{u}) = v \wedge PossWithDefPr(c(\vec{u}) = v)\Big) \wedge \\ & \bigvee_{\substack{\text{r' :}\\ [r']\ random(c(\vec{u}):\{X:p(X)\}) \leftarrow Body \in \Pi}} \Big(Body \wedge x = 1 - y \wedge \\ & y = \#\text{sum}\{p_1 : PossWithAssPr_{r',C_1}(c(\vec{u})\!=\!v_1); \dots; \\ & \qquad p_n : PossWithAssPr_{r',C_n}(c(\vec{u})\!=\!v_n)\}\Big) \end{aligned} \tag{4.20}$$

$$TotalDefPr(c(\vec{u}), x) \leftrightarrow RemPr(c(\vec{u}), x) \wedge x > 0 \tag{4.21}$$

$$\neg(RemPr(c(\vec{u}), x) \wedge x \leq 0) \tag{4.22}$$

First, let's expand some notations in the definition of $\phi(W)$:

- $c(\vec{u}) = v$ is possible in W

  By definition, it is equivalent to "there exists a random selection rule (2.2) such that $W \models Body \wedge p(v) \wedge \neg Intervene(c(\vec{u}))$".

- a pr-atom $pr_r(c(\vec{u}) = v_i \mid C_i) = p_i$ is applied in W

  By definition, it is equivalent to "$c(\vec{u}) = v_i$ is possible in W due to r, and $W \models C_i$".

- $v \in AV_W(c(\vec{u}))$

  By the definition of $AV_W(c(\vec{u}))$, it is equivalent to "there exists a pr-atom $pr_r(c(\vec{u}) = v \mid C_i) = p_i$ that is applied in W for some r and i ".

Then we will prove that each formula in $Comp[\Pi_{AUX}; \sigma_3]$ is satisfied by $\phi(W)$ based on the definition of $\phi(W)$:

- **Let's take formula (4.12) into account.** Consider the random selection rule $[r]\ random(c(\vec{u}) : \{X : p(X)\}) \leftarrow Body$, where formula (4.12) is obtained. By definition,

  * $\phi(W) \models Poss_r(c(\vec{u}) = v)$

  iff

  * $c(\vec{u}) = v$ is possible in W due to r

  iff

  * $W \models Body \wedge p(v) \wedge not\ Intervene(c(\vec{u}))$

  iff (since $\phi(W)$ is an extension of W)

  * $\phi(W) \models Body \wedge p(v) \wedge not\ Intervene(c(\vec{u}))$

  Thus formula (4.12) is satisfied by $\phi(W)$.

- **Let's take formula (4.13) into account.** By definition,

* $\phi(W) \vDash \mathit{PossWithDefPr}(c(\vec{u}) = v)$

iff

* $c(\vec{u}) = v$ is possible in W
* $v \notin AV_W(c(\vec{u}))$

iff (by definition)

* there exists a random selection rule r such that $c(\vec{u}) = v$ is possible in W due to r
* $\phi(W) \nvDash \mathit{PossWithAssPr}(c(\vec{u}) = v)$

iff (by definition)

* there exists a random selection rule r such that $\phi(W) \vDash \mathit{Poss}_r(c(\vec{u}) = v)$
* $\phi(W) \vDash \neg\mathit{PossWithAssPr}(c(\vec{u}) = v)$

Thus formula (4.13) is satisfied by $\phi(W)$.

– **Let's take formula (4.14) into account.** By definition,

* $\phi(W) \vDash \mathit{NumDefPr}(c(\vec{u}), x)$

iff

* there exist exactly x different v such that
  · $c(\vec{u}) = v$ is possible in W
  · $v \notin AV_W(c(\vec{u}))$
  · for one of such v, $W \models c(\vec{u}) = v$

iff (by definition and since $\phi(W)$ is an extension of W)

* there exists exactly x different v such that
  · $\phi(W) \vDash \mathit{PossWithDefPr}(c(\vec{u}) = v)$
  · for one of such v, $\phi(W) \vDash c(\vec{u}) = v \wedge \mathit{PossWithDefPr}(c(\vec{u}) = v)$

Thus formula (4.14) is satisfied by $\phi(W)$.

- **Let's take formula (4.16) into account.** Consider the pr-atom $pr_r(c(\vec{u}) = v_i \mid C_i) = p_i$ where formula (4.16) is obtained. By definition,

  * $\phi(W) \models PossWithAssPr_{r,C_i}(c(\vec{u}) = v_i)$

  iff

  * this pr-atom is applied in W

  iff

  * $c(\vec{u}) = v_i$ is possible in W due to r, and $W \models C_i$

  iff (by definition and since $\phi(W)$ is an extension of W)

  * $\phi(W) \models Poss_r(c(\vec{u}) = v_i) \wedge C_i$

  Thus formula (4.16) is satisfied by $\phi(W)$.

  Remark: By **Condition 1**, r is the only random selection rule for $c(\vec{u})$ whose "*Body*" is satisfied by W. And by **Condition 2**, there won't be another pr-atom $pr_r(c(\vec{u}) = v \mid C') = p' \in \Pi$ such that $W \models C'$. Thus for any $c(\vec{u}) = v$, $\phi(W)$ could at most satisfy one $PossWithAssPr_{r,C_i}(c(\vec{u}) = v_i)$ for any r and C_i.

- **Let's take formula (4.17) into account.** Consider the pr-atom $pr_r(c(\vec{u}) = v_i \mid C_i) = p_i$ in Π, where formula (4.17) is obtained, by definition,

  * $\phi(W) \models AssPr_{r,C_i}(c(\vec{u}) = v_i)$

  iff

  * this pr-atom is applied in W
  * $W \models c(\vec{u}) = v_i$

  iff (by definition and since $\phi(W)$ is an extension of W)

  - $\phi(W) \models PossWithAssPr_{r,C_i}(c(\vec{u}) = v_i) \wedge c(\vec{u}) = v_i$

  Thus formula (4.17) is satisfied by $\phi(W)$.

- **Let's take formula (4.18) into account.** For any pr-atom

$$pr_r(c(\vec{u}) = v_i \mid C_i) = p_i$$

  in Π such that $p_i = 0$, assume for the sake of contradiction that $\phi(W) \models AssPr_{r,C_i}(c(\vec{u}) = v_i)$. Then by definition, this pr-atom is applied and $W \models c(\vec{u}) = v_i$. In other words, $c(\vec{u}) = v_i \in W$, $c(\vec{u}) = v_i$ is possible in W, and $P(W, c(\vec{u}) = v_i) = 0$. Thus $\hat{\mu}_\Pi(W) = 0$, which contradicts that $\mu_\Pi(W) > 0$.

  Thus formula (4.18) is satisfied by $\phi(W)$.

- **Let's take formula (4.19) into account.** By definition,

  - $\phi(W) \models PossWithAssPr(c(\vec{u}) = v_i)$

  iff

  - $v_i \in AV_W(c(\vec{u}))$

  iff

  - there exist a pr-atom $pr_r(c(\vec{u}) = v_i \mid C_j) = p_j$ that is applied in W for some r and j (where i and j may be different)

  iff (by definition)

  - there exist r and j such that $\phi(W) \models PossWithAssPr_{r,C_j}(c(\vec{u}) = v_i)$

  Thus formula (4.19) is satisfied by $\phi(W)$.

- **Let's take formula (4.20) into account.** By definition,

  - $\phi(W) \models RemPr(c(\vec{u}), x)$

  iff

* there exists a v such that
  - $W \models c(\vec{u})=v$
  - $c(\vec{u})=v$ is possible in W
  - $v \notin AV_W(c(\vec{u}))$, and
* $x = 1 - \sum_{v' \in AV_W(c(\vec{u}))} PossWithAssPr(W, c(\vec{u})=v')$

iff (by definition and since $\phi(W)$ is an extension of W)

* there exists a v such that $\phi(W) \models c(\vec{u}) = v \wedge PossWithDefPr(c(\vec{u}) = v)$,
* $x = 1 - y$, and
* $y = \sum_{\phi(W) \models PossWithAssPr(c(\vec{u})=v')} PossWithAssPr(W, c(\vec{u})=v')$

iff (by formula (34) and the definition of $PossWithAssPr(W, c(\vec{u})=v)$)

* there exists a v such that $\phi(W) \models c(\vec{u}) = v \wedge PossWithDefPr(c(\vec{u}) = v)$,
* $x = 1 - y$, and
* there exists a random selection rule r (2.2) along with all pr-atoms (4.15) associated with it such that
  - $\phi(W) \models Body$
  - $y = \sum_{j:\phi(W) \models PossWithAssPr_{r,C_j}(c(\vec{u})=v_j)} p_j$

Thus formula (4.20) is satisfied by $\phi(W)$.

Remark: By **Condition 1**, there exits at most one random selection rule whose "*Body*" is satisfied by W. Thus there is no other random selection rule r' such that $\phi(W) \models PossWithAssPr_{r',C_j}(c(\vec{u})=v_j)$ for any j. Morevoer, for any $c(\vec{u}) \in \Pi$, there exits at most one $RemPr(c(\vec{u}), x)$ that can be satisfied by $\phi(W)$.

– **Let's take formula (4.21) into account.** By definition,

* $\phi(W) \models \mathit{TotalDefPr}(c(\vec{u}), x)$

  iff

  * $\phi(W) \models \mathit{RemPr}(c(\vec{u}), x) \wedge x > 0$

  Thus formula (4.21) is satisfied by $\phi(W)$.

- **Let's take formula (4.22) into account.** Consider the random selection rule (2.2) for $c(\vec{u})$ along with all pr-atoms (4.15) associated with it ($n \geq 1$), where formula (4.22) is obtained. Assume for the sake of contradiction that $\phi(W) \models (\mathit{RemPr}(c(\vec{u}), x) \wedge x \leq 0)$ for some x, which (by definition) follows that

  * there exists a v such that
    · $W \models c(\vec{u}) = v$
    · $c(\vec{u}) = v$ is possible in W
    · $v \notin AV_W(c(\vec{u}))$
  * $x = 1 - \sum_{v \in AV_W(c(\vec{u}))} \mathit{PossWithAssPr}(W, c(\vec{u}) = v)$ and $x \leq 0$

  In other words, $c(\vec{u}) = v \in W$, $c(\vec{u}) = v$ is possible in W, and $P(W, c(\vec{u}) = v) = \mathit{PossWithDefPr}(W, c(\vec{u}) = v) = 0$. Thus $\hat{\mu}_\Pi(W) = 0$, which contradicts that $\mu_\Pi(W) > 0$.

  Thus formula (4.22) is satisfied by $\phi(W)$.

Now we see the definition of $\phi(W)$ guarantees that $\phi(W)$ is a model of $\mathit{Comp}[\Pi_{AUX}; \sigma_3]$. Thus $\phi(W)$ is a stable model of Π_{AUX} relative to σ_3.

Until now we proved $\phi(W)$ is a stable model of Π'. Then, we are going to prove $\mu_\Pi(W) = P_{\Pi'}(\phi(W))$.

Recall that Π' denotes the translated LP$^{\text{MLN}}$ program plog2lpmln(Π), $W'_\Pi(I)$ denotes the unnormalized weight of I under Π with respect to soft rules only.

Firstly we will prove $\hat{\mu}_\Pi(W) = W'_{\Pi'}(\phi(W))$. From the definition of $\hat{\mu}_\Pi(W)$ (unnormalized probability) and $P(W, c(\vec{u}) = v)$ in the semantics of P-log, we have

$$\begin{aligned}
\hat{\mu}_\Pi(W) &= \prod_{\substack{c(\vec{u}) = v : \\ c(\vec{u}) = v \text{ is possible in } W \\ \text{and } W \vDash c(\vec{u}) = v}} P(W, c(\vec{u}) = v) \\
&= \prod_{\substack{c(\vec{u}) = v : \\ c(\vec{u}) = v \text{ is possible in } W \\ W \vDash c(\vec{u}) = v \\ \text{and } v \in AV_W(c(\vec{u}))}} P(W, c(\vec{u}) = v) \times \prod_{\substack{c(\vec{u}) = v : \\ c(\vec{u}) = v \text{ is possible in } W \\ W \vDash c(\vec{u}) = v \\ \text{and } v \notin AV_W(c(\vec{u}))}} P(W, c(\vec{u}) = v) \\
&= \prod_{\substack{c(\vec{u}) = v : \\ c(\vec{u}) = v \text{ is possible in } W \\ W \vDash c(\vec{u}) = v \\ \text{and } v \in AV_W(c(\vec{u}))}} \mathit{PossWithAssPr}(W, c(\vec{u}) = v) \times \\
&\quad \prod_{\substack{c(\vec{u}) = v : \\ c(\vec{u}) = v \text{ is possible in } W \\ \text{and } W \vDash c(\vec{u}) = v \\ \text{and } v \notin AV_W(c(\vec{u}))}} \mathit{PossWithDefPr}(W, c(\vec{u}) = v)
\end{aligned}$$

Since W is a possible world of Π with a non-zero probability, $\hat{\mu}_\Pi(W) > 0$. Since the statement "$v \in AV_W(c(\vec{u}))$" is equivalent to saying "$c(\vec{u}) = v$ is possible in W, and there exists $pr_{r_{W,c(\vec{u})}}(c(\vec{u}) = v \mid C) = p \in \Pi$ for some C and p, and

$W \models C$", we have

$$\begin{aligned}
\hat{\mu}_{\Pi}(W) = & \prod_{\substack{c(\vec{u})=v: \\ c(\vec{u})=v \text{ is possible in } W \\ W \models c(\vec{u})=v \\ pr_{r_{W,c(\vec{u})}}(c(\vec{u})=v \mid C)=p \in \Pi \\ \text{and } W \models C}} p \times \\
& \prod_{\substack{c(\vec{u})=v: \\ c(\vec{u})=v \text{ is possible in } W \\ W \models c(\vec{u})=v \\ \text{and } v \notin AV_W(c(\vec{u}))}} \frac{1 - \sum_{v' \in AV_W(c(\vec{u}))} PossWithAssPr(W, c(\vec{u})=v')}{|\{v'' \mid c(\vec{u})=v'' \text{ is possible in } W \text{ and } v'' \notin AV_W(c(\vec{u}))\}|} \\
= & \prod_{\substack{c(\vec{u})=v: \\ c(\vec{u})=v \text{ is possible in } W \\ W \models c(\vec{u})=v \\ pr_{r_{W,c(\vec{u})}}(c(\vec{u})=v \mid C)=p \in \Pi \\ \text{and } W \models C}} p \times \\
& \prod_{\substack{c(\vec{u})=v: \\ c(\vec{u})=v \text{ is possible in } W \\ W \models c(\vec{u})=v \\ \text{and } v \notin AV_W(c(\vec{u}))}} \frac{1}{|\{v' \mid c(\vec{u})=v' \text{ is possible in } W \text{ and } v' \notin AV_W(c(\vec{u}))\}|} \times \\
& \prod_{\substack{c(\vec{u})=v: \\ c(\vec{u})=v \text{ is possible in } W \\ W \models c(\vec{u})=v \\ \text{and } v \notin AV_W(c(\vec{u}))}} (1 - \sum_{\substack{c(\vec{u})=v': \\ c(\vec{u})=v' \text{ is possible in } W \\ pr_{r_{W,c(\vec{u})}}(c(\vec{u})=v' \mid C)=p \in \Pi \\ \text{and } W \models C}} p)
\end{aligned}$$

Note that by **Condition 1**, the subscript $r_{W,c(\vec{u})}$ of the applied pr-atom is the only random selection rule for $c(\vec{u})$ whose body could be satisfied by W.

We then calculate $W'_{\Pi'}(\phi(W))$, the unnormalized weight of $\phi(W)$ with respect to all soft rules in Π'. From the construction of Π', it's easy to see that there are only 3 kinds of soft rules: Rule (4.4), Rule (4.7), and Rule (4.10), which are satisfied iff $\phi(W) \models AssPr_{r,C}(c(\vec{u}) = v)$, $\phi(W) \models NumDefPr(c(\vec{u}), m)$, and $\phi(W) \models TotalDefPr(c(\vec{u}), x)$, respectively. Let's denote the unnormalized weight of $\phi(W)$ with respect to each of these three rules as $W'_{\Pi'}(\phi(W))|_{4.4}$, $W'_{\Pi'}(\phi(W))|_{4.7}$, $W'_{\Pi'}(\phi(W))|_{4.10}$. It's clear that $W'_{\Pi'}(\phi(W)) = W'_{\Pi'}(\phi(W))|_{4.4} \times W'_{\Pi'}(\phi(W))|_{4.7} \times W'_{\Pi'}(\phi(W))|_{4.10}$.

Consider a $c(\vec{u}) = v$ that is possible in W and $W \vDash c(\vec{u}) = v$. Since $\hat{\mu}_\Pi(W) > 0$, if $v \in AV_W(c(\vec{u}))$, $pr_{r_{W,c(\vec{u})}}(c(\vec{u}) = v \mid C) = p \in \Pi$ and $W \vDash C$, then $P(W, c(\vec{u}) = v) = p$ and $p > 0$; if f $v \notin AV_W(c(\vec{u}))$, then $1 - \sum_{v' \in AV_W(c(\vec{u}))} PossWithAssPr(W, c(\vec{u}) = v')$ must be greater than 0. By the definition of $\phi(W)$,

$$W'_{\Pi'}(\phi(W))|_{4.4} = exp\left(\sum_{\substack{c(\vec{u}) = v: \\ pr_r(c(\vec{u}) = v \mid C) = p \in \Pi \\ \phi(W) \vDash AssPr_{r,C}(c(\vec{u}) = v)}} ln(p) \right)$$

(Note that by **Condition 1**, r must be the same as $r_{W,c(\vec{u})}$)

$$= \prod_{\substack{c(\vec{u}) = v: \\ pr_{r_{W,c(\vec{u})}}(c(\vec{u}) = v \mid C) = p \in \Pi \\ c(\vec{u}) = v \text{ is possible in } W \\ W \vDash C \\ \text{and } W \vDash c(\vec{u}) = v}} p$$

$$W'_{\Pi'}(\phi(W))|_{4.7} = exp\left(\sum_{\substack{c(\vec{u}), m: \\ m \geq 2 \\ \phi(W) \vDash NumDefPr(c(\vec{u}), m)}} ln(\frac{1}{m}) \right)$$

$$= exp\left(\sum_{\substack{c(\vec{u}), m: \\ \phi(W) \vDash NumDefPr(c(\vec{u}), m)}} ln(\frac{1}{m}) \right)$$

$$= \prod_{\substack{c(\vec{u}) = v: \\ c(\vec{u}) = v \text{ is possible in } W \\ W \vDash c(\vec{u}) = v \\ \text{and } v \notin AV_W(c(\vec{u}))}} \frac{1}{|\{v' \mid c(\vec{u}) = v' \text{ is possible in } W \text{ and } v' \notin AV_W(c(\vec{u}))\}|}$$

$$W'_{\Pi'}(\phi(W))|_{4.10} = exp\left(\sum_{\substack{c(\vec{u}),x: \\ \phi(W)\models TotalDefPr(c(\vec{u}),x)}} ln(x) \right)$$

$$= exp\left(\sum_{\substack{c(\vec{u})=v: \\ c(\vec{u})=v \text{ is possible in } W \\ v\notin AV_W(c(\vec{u})) \\ \text{and } W\models c(\vec{u})=v}} ln(1 - \sum_{v'\in AV_W(c(\vec{u}))} PossWithAssPr(W, c(\vec{u})=v') \right)$$

$$= exp\left(\sum_{\substack{c(\vec{u})=v: \\ c(\vec{u})=v \text{ is possible in } W \\ v\notin AV_W(c(\vec{u})) \\ \text{and } W\models c(\vec{u})=v}} ln(1 - \sum_{\substack{c(\vec{u})=v': \\ c(\vec{u})=v' \text{ is possible in } W \\ pr_{r_{W,c(\vec{u})}}(c(\vec{u})=v' \mid C)=p\in\Pi \\ \text{and } W\models C}} p) \right)$$

$$= \prod_{\substack{c(\vec{u})=v: \\ c(\vec{u})=v \text{ is possible in } W \\ W\models c(\vec{u})=v \\ \text{and } v\notin AV_W(c(\vec{u}))}} (1 - \sum_{\substack{c(\vec{u})=v': \\ c(\vec{u})=v' \text{ is possible in } W \\ pr_{r_{W,c(\vec{u})}}(c(\vec{u})=v' \mid C)=p\in\Pi \\ \text{and } W\models C}} p)$$

It is easy to see that $W'_{\Pi'}(\phi(W)) = W'_{\Pi'}(\phi(W))|_{4.4} \times W'_{\Pi'}(\phi(W))|_{4.7} \times W'_{\Pi'}(\phi(W))|_{4.10} = \hat{\mu}_{\Pi}(W)$. We already proved that for any possible world W of Π, $\phi(W)$ is a probabilistic stable model of Π'. Then to prove $\mu_{\Pi}(W) = P_{\Pi'}(\phi(W))$, it is sufficient to prove for any probabilistic stable model I of Π', $I|_{\sigma}$ is a possible world of Π and $I = \phi(I|_{\sigma})$ (which will be proved in the next part). Indeed, if we proved this, we know $\phi(W)$ and W are 1-1 correspondent, thus $P'_{\Pi'}(\phi(W)) = \mu_{\Pi}(W)$. Since $\phi(W) \in \text{SM}'[\Pi']$, by **Lemma 5**, $P_{\Pi'}(\phi(W)) = P'_{\Pi'}(\phi(W)) = \mu_{\Pi}(W)$.

(b) Since Π is consistent, there exists a possible world W' of Π with a non-zero probability. It's proved that $\phi(W')$ is a probabilistic stable model of Π' and $\phi(W')$ satisfies $\overline{\Pi'^{\text{ hard}}}$. So $\text{SM}'[\Pi']$ is not empty. Let I be a probabilistic stable model of Π', by **Lemma 5**, $I \models \overline{\Pi'^{\text{ hard}}}$. Besides, since $\Pi' \setminus \Pi'^{\text{ hard}}$ is a set of rules of the form $w :\leftarrow F$, by **Lemma 4**, I is a stable model of $\overline{\Pi'^{\text{ hard}}_{I}}$. Thus I is a stable model of $\overline{\Pi'^{\text{ hard}}}$.

Since (1) I is a stable model of $\tau(\Pi) \cup \Pi_{AUX}$, (2) no atom in σ has a strictly positive occurrence in Π_{AUX}, (3) no atom in σ_3 has a strictly positive occurrence in $\tau(\Pi)$, (4) each strongly connected component of the dependency graph of $\tau(\Pi) \cup \Pi_{AUX}$ relative to $\sigma \cup \sigma_3$ is a subset of σ or a subset of σ_3, by the splitting theorem

- I is a stable model of $\tau(\Pi)$ relative to σ. Thus $I|_\sigma$ is a stable model of $\tau(\Pi)$, which means $I|_\sigma$ is a possible world of Π.
- I is a stable model of Π_{AUX} relative to σ_3. So $I \models \mathit{Comp}[\Pi_{AUX}; \sigma_3]$.

Let's denote $I|_\sigma$ by W, we'll prove $I = \phi(W)$ by checking if I satisfies all conditions in the definition of $\phi(W)$.

- Let's consider condition **(1)** in the definition of ϕ. Take any random selection rule $[r]\ \mathit{random}(c(\vec{u}) : \{X : p(X)\}) \leftarrow \mathit{Body}$, since I satisfies formula (4.12),

  - $I \models \mathit{Poss}_r(c(\vec{u}) = v)$

  iff

  - $I \models \mathit{Body} \wedge p(v) \wedge \neg \mathit{Intervene}(c(\vec{u}))$

  iff (since all atoms in the above conjunction part belong to σ)

  - $W \models \mathit{Body} \wedge p(v) \wedge \neg \mathit{Intervene}(c(\vec{u}))$

  iff

  - $c(\vec{u}) = v$ is possible in W due to r.

- Let's consider condition **(2)** in the definition of ϕ. Take any pr-atom $pr_r(c(\vec{u}) = v_i \mid C_i) = p_i$ in Π, since I satisfies formula (4.16),

  - $I \models \mathit{PossWithAssPr}_{r,C_i}(c(\vec{u}) = v_i)$

iff

- $I \models Poss_r(c(\vec{u}) = v_i) \wedge C_i$

iff (from the proof of condition **(1)**, and since C_i belongs to σ)

- $c(\vec{u}) = v_i$ is possible in W due to r and $W \models C_i$

iff

- this pr-atom is applied in W

Thus condition **(2)** is satisfied by I.

- Let's consider condition **(3)** in the definition of ϕ. Take any pr-atom $pr_r(c(\vec{u}) = v_i \mid C_i) = p_i$ in Π, since I satisfies formula (4.17),

  - $I \models AssPr_{r,C_i}(c(\vec{u}) = v_i)$

  iff

  - $I \models PossWithAssPr_{r,C_i}(c(\vec{u}) = v_i) \wedge c(\vec{u}) = v_i$

  iff (from the proof of condition **(2)**, and since $c(\vec{u}) = v_i$ belongs to σ)

  - this pr-atom is applied in W
  - $W \models c(\vec{u}) = v_i$

  Thus condition **(3)** is satisfied by I.

- Let's consider condition **(4)** in the definition of ϕ. Since I satisfies formula (4.19),

  - $I \models PossWithAssPr(c(\vec{u}) = v_i)$

  iff (from the proof of condition **(2)**)

  - there exist a r and j such that $I \models PossWithAssPr_{r,C_j}(c(\vec{u}) = v_i)$

  iff

- there exist a pr-atom $pr_r(c(\vec{u}) = v_i \mid C_j) = p_j$ that is applied in W for some r and j (where i and j may be different)

iff

- $v_i \in AV_W(c(\vec{u}))$

Thus condition **(4)** is satisfied by I.

- Let's consider condition **(5)** in the definition of ϕ. Since I satisfies formula (4.13),

  - $I \models PossWithDefPr(c(\vec{u}) = v)$

  iff

  - $I \models \neg PossWithAssPr(c(\vec{u}) = v)$
  - there exists a random selection rule $[r]\ random(c(\vec{u}) : \{X : p(X)\}) \leftarrow Body$, such that $I \models Poss_r(c(\vec{u}) = v)$

  iff (by condition **(4)** and **(1)**)

  - $v \notin AV_W(c(\vec{u}))$
  - $c(\vec{u}) = v$ is possible in W

  Thus condition **(5)** is satisfied by I.

- Let's consider condition **(6)** in the definition of ϕ. Since I satisfies formula (4.14),

  - $I \models NumDefPr(c(\vec{u}), x)$

  iff

  - $x = \#count\{y : PossWithDefPr(c(\vec{u}){=}y)\}$
  - there exists a $c(\vec{u}) = z$ such that $I \models c(\vec{u}) = z \wedge PossWithDefPr(c(\vec{u}) = z)$

iff

- there exist exactly x different values v such that
    - $I \models \mathit{PossWithDefPr}(c(\vec{u}) = v)$
    - for one of such v, $I \models c(\vec{u}) = v$

iff (by condition **(5)**, and since $c(\vec{u}) = v$ belongs to σ)

- there exist exactly x different values v such that
    - $c(\vec{u}) = v$ is possible in W
    - $v \notin AV_W(c(\vec{u}))$
    - for one of such v, $W \models c(\vec{u}) = v$

Thus condition **(6)** is satisfied by I.

- Let's consider condition **(7)** in the definition of ϕ. Since I satisfies formula (4.20),

    - $I \models \mathit{RemPr}(c(\vec{u}), x)$

    iff

    - there exists a v such that $\phi(W) \models c(\vec{u}) = v \wedge \mathit{PossWithDefPr}(c(\vec{u}) = v)$
    - there exists a random selection rule (2.2) along with all pr-atoms (4.15) associated with it such that
        - $I \models \mathit{Body}$
        - $y = \sum_{pr_r(c(\vec{u})=v|C)=p \in \Pi, \phi(W) \models \mathit{PossWithAssPr}_{r,C}(c(\vec{u})=v)} p$
        - $x = 1 - y$

    iff (by condition **(5)** and **(2)**, and since $c(\vec{u}) = v$ belongs to σ)

    - there exists a v such that
        - $c(\vec{u}) = v$ is possible in W

    - $v \notin AV_W(c(\vec{u}))$
    - $W \models c(\vec{u}) = v$
  - $x = 1 - \sum_{v' \in AV_W(c(\vec{u}))} \mathit{PossWithAssPr}(W, c(\vec{u}) = v')$

  Thus condition **(7)** is satisfied by I.

- Let's consider condition **(8)** in the definition of ϕ. Since I satisfies formula (4.21),

  - $I \models \mathit{TotalDefPr}(c(\vec{u}), x)$

  iff

  - $I \models \mathit{RemPr}(c(\vec{u}), x)$
  - $x > 0$

  Thus condition **(8)** is satisfied by I.

  Now we proved that I is exactly $\phi(W)$, in other words, $I = \phi(I|_\sigma)$. Thus for every probabilistic stable model I of plog2lpmln(Π), $I|_\sigma$ is a possible world of Π and $I = \phi(I|_\sigma)$. Consequently, W and $\phi(W)$ (or $I|_\sigma$ and I) are 1-1 correspondent. Since I is a probabilistic stable model of Π', $P_{\Pi'}(I) > 0$. Then $\mu_\Pi(I|_\sigma) = P_{\Pi'}(I) > 0$.

∎

Chapter 5

MORE EXAMPLES

In this chapter, we show more examples of how we translate a P-log program into an LP$^{\text{MLN}}$ program, and further into an answer set program with weak constraints.

5.1 Variant of Monty Hall Problem (Continued)

In Example 6, we translated a P-log program Π of a varient of Monty Hall Problem into an LP$^{\text{MLN}}$ program plog2lpmln(Π). We can further translate plog2lpmln(Π) into an ASP program with weak constraints, denoted by Π', by applying the translation $\text{lpmln2wc}^{\text{pnt,clingo}}_{\text{simp}}$ and the simplification for hard rules.

Recall: The simplication for hard rules says that when all hard rules in an LP$^{\text{MLN}}$ program Π encode definite knowledge (i.e., we are sure that there exists a probabilistic stable model of Π that satisfies all hard rules of Π), we can simply translated each hard rule in Π

$$\alpha : \mathit{Head} \leftarrow \mathit{Body}$$

into the usual ASP rules

$$\mathit{Head} \leftarrow \mathit{Body}.$$

Π' is constructed as follows.

% $****$ Declaration Part $****$

$door(1..4).$

% $****$ $\tau(\Pi)$ $****$

$canOpen,(D,f) \leftarrow selected(D).$

$canOpen,(D,f) \leftarrow prize(D).$

$canOpen,(D,t) \leftarrow not\ canOpen(D,f), door(D).$

$\leftarrow canOpen,(D,t), canOpen,(D,f).$

$\leftarrow prize(D_1), prize(D_2), D_1 \neq D_2.$

$\leftarrow selected(D_1), selected(D_2), D_1 \neq D_2.$

$\leftarrow open(D_1), open(D_2), D_1 \neq D_2.$

$1\{prize(D) : door(D)\}1 \leftarrow not\ intervene(prize).$

$1\{selected(D) : door(D)\}1 \leftarrow not\ intervene(selected).$

$1\{open(D) : door(D)\}1 \leftarrow not\ intervene(open).$

$\leftarrow open(D), not\ canOpen,(D,t), not\ intervene(open).$

$obs(selected, 1).$

$\leftarrow obs(selected, 1), not\ selected(1).$

$obs(open, 2).$

$\leftarrow obs(open, 2), not\ open(2).$

$nobs(prize, 2).$

$\leftarrow nobs(prize, 2), prize(2).$

% $****$ Possible Atoms $****$

$poss(prize, D) \leftarrow not\ intervene(prize), door(D).$

$poss(selected, D) \leftarrow not\ intervene(selected), door(D).$

$poss(open, D) \leftarrow canOpen(D), not\ intervene(open), door(D).$

$\% * * * *$ Assigned Probability $* * * *$

$possWithAssPr(prize, 1) \leftarrow poss(prize, 1).$

$assPr(prize, 1) \leftarrow prize(1), possWithAssPr(prize, 1).$

$possWithAssPr(prize, 3) \leftarrow poss(prize, 3).$

$assPr(prize, 3) \leftarrow prize(3), possWithAssPr(prize, 3).$

$\% * * * *$ Denominator for Default Probability $* * * *$

$possWithDefPr(prize, D) \leftarrow poss(prize, D), not\ possWithAssPr(prize, D).$

$possWithDefPr(selected, D) \leftarrow poss(selected, D), not\ possWithAssPr(selected, D).$

$possWithDefPr(open, D) \leftarrow poss(open, D), not\ possWithAssPr(open, D).$

$numDefPr(prize, X) \leftarrow prize(D), possWithDefPr(prize, D),$

$$X = \#count\{Y : possWithDefPr(prize, Y)\}.$$

$numDefPr(selected, X) \leftarrow selected(D), possWithDefPr(selected, D),$

$$X = \#count\{Y : possWithDefPr(selected, Y)\}.$$

$numDefPr(open, X) \leftarrow open(D), possWithDefPr(open, D),$

$$X = \#count\{Y : possWithDefPr(open, Y)\}.$$

$\% * * * *$ Numerator for Default Probability $* * * *$

$remPr(prize, Y) \leftarrow prize(D), possWithDefPr(prize, D),$

$$X = \#\text{sum}\{0.3 : possWithAssPr(prize, 1); 0.2 : possWithAssPr(prize, 3)\}, Y = 1 - X.$$

$totalDefPr(prize, X) \leftarrow remPr(prize, X), X > 0.$

$\leftarrow remPr(prize, X), X \leq 0.$

$$\% **** \text{ Weak Constraints } ****$$

$$:\sim \mathit{assPr}(\mathit{prize}, 1).\ [-ln(0.3)]$$

$$:\sim \mathit{assPr}(\mathit{prize}, 3).\ [-ln(0.2)]$$

$$:\sim \mathit{numDefPr}(C, M).\ [-ln(1/M)]$$

$$:\sim \mathit{totalDefPr}(\mathit{prize}, X).\ [-ln(X)]$$

Note that, by default, the level of a weak constraint is 0. We can check all stable models of this program by simply removing "Weak Constraints" part.

5.2 Dice Problem

Example 7 *The following P-log program* Π *describes Dice Problem. There are two dices, one is owned by Mike and one is owned by John. The statistics shows that John's dice is a fair one but Mike's dice is unfair with 25% chance to roll 5 and 20% chance to roll 6.* $(d \in \{D_1, D_2\}, n \in \{1, \dots, 6\})$

$$Owner(D_1) = Mike.$$

$$Owner(D_2) = John.$$

$$Even(d) \leftarrow \mathit{Roll}(d) = n, n \ mod \ 2 = 0.$$

$$\sim Even(d) \leftarrow not \ Even(d).$$

$$[r(d)] \ \ random(\mathit{Roll}(d)).$$

$$pr_{r(d)}(\mathit{Roll}(d) = 5 \mid Owner(d) = Mike) = \tfrac{1}{4}.$$

$$pr_{r(d)}(\mathit{Roll}(d) = 6 \mid Owner(d) = Mike) = \tfrac{1}{5}.$$

The translated LP^{MLN} *encoding* plog2lpmln(Π) *is as follows, where* C_d *represents the condition "*$Owner(d) = Mike$*" in two pr-atoms.*

$//**** \; \tau(\Pi) \; ****$

$\alpha: \; Owner(D_1) = Mike$

$\alpha: \; Owner(D_2) = John$

$\alpha: \; Even(d) = \mathbf{t} \leftarrow Roll(d) = n, n \; mod \; 2 = 0$

$\alpha: \; Even(d) = \mathbf{f} \leftarrow not \; Even(d) = \mathbf{t}$

$\alpha: \; \leftarrow Roll(d) = n_1, Roll(d) = n_2, n_1 \neq n_2$

$\alpha: \; \leftarrow Even(d) = \mathbf{t}, Even(d) = \mathbf{f}$

$\alpha: \; \leftarrow Owner(d) = p_1, Owner(d) = p_2, p_1 \neq p_2$

$\alpha: \; Roll(d) = 1; Roll(d) = 2; Roll(d) = 3; Roll(d) = 4; Roll(d) = 5; Roll(d) = 6$

$\leftarrow not \; Intervene(Roll(d))$

$//**** \; Possible \; Atoms ****$

$\alpha: \; Poss_{r(d)}(Roll(d) = n) \leftarrow not \; intervene(Roll(d))$

$//**** \; Assigned \; Probability ****$

$\alpha: \; PossWithAssPr_{r(d),C_d}(Roll(d) = 5) \leftarrow Poss_{r(d)}(Roll(d) = 5), Owner(d) = Mike$

$\alpha: \; AssPr_{r(d),C_d}(Roll(d) = 5) \leftarrow Roll(d) = 5, PossWithAssPr_{r(d),C_d}(Roll(d) = 5)$

$ln(\frac{1}{4}): \; \bot \leftarrow not \; AssPr_{r(d),C_d}(Roll(d) = 5)$

$\alpha: \; PossWithAssPr(Roll(d) = 5) \leftarrow PossWithAssPr_{r(d),C_d}(Roll(d) = 5)$

$\alpha: \; PossWithAssPr_{r(d),C_d}(Roll(d) = 6) \leftarrow Poss_{r(d)}(Roll(d) = 6), Owner(d) = Mike$

$\alpha: \; AssPr_{r(d),C_d}(Roll(d) = 6) \leftarrow Roll(d) = 6, PossWithAssPr_{r(d),C_d}(Roll(d) = 6)$

$ln(\frac{1}{5}): \; \bot \leftarrow not \; AssPr_{r(d),C_d}(Roll(d) = 6)$

$\alpha: \; PossWithAssPr(Roll(d) = 6) \leftarrow PossWithAssPr_{r(d),C_d}(Roll(d) = 6)$

$// \ast\ast\ast\ast$ *Denominator for Default Probability* $\ast\ast\ast\ast$

$$\alpha: \ \mathit{PossWithDefPr}(\mathit{Roll}(d) = n) \leftarrow \mathit{Poss}_{r(d)}(\mathit{Roll}(d) = n), \quad not \ \mathit{PossWithAssPr}(\mathit{Roll}(d) = n)$$

$$\alpha: \ \mathit{NumDefPr}(\mathit{Roll}(d), x) \leftarrow \mathit{Roll}(d) = n, \mathit{PossWithDefPr}(\mathit{Roll}(d) = n), \quad x = \#count\{y : \mathit{PossWithDefPr}(\mathit{Roll}(d) = y)\}$$

$$ln(\tfrac{1}{m}): \ \bot \leftarrow not \ \mathit{NumDefPr}(\mathit{Roll}(d), m) \qquad (m \in \{2, \dots, 6\})$$

$// \ast\ast\ast\ast$ *Numerator for Default Probability* $\ast\ast\ast\ast$

$$\alpha: \ \mathit{RemPr}(\mathit{Roll}(d), 1 - y) \leftarrow \mathit{Roll}(d) = n, \mathit{PossWithDefPr}(\mathit{Roll}(d) = n), \quad y = \#sum\{\tfrac{1}{4} : \mathit{PossWithAssPr}_{r(d),C_d}(\mathit{Roll}(d) = 5); \ \tfrac{1}{5} : \mathit{PossWithAssPr}_{r(d),C_d}(\mathit{Roll}(d) = 6)\}$$

$$\alpha: \ \mathit{TotalDefPr}(\mathit{Roll}(d), x) \leftarrow \mathit{RemPr}(\mathit{Roll}(d), x), x > 0$$

$$ln(x): \ \bot \leftarrow not \ \mathit{TotalDefPr}(\mathit{Roll}(d), x)$$

$$\alpha: \ \bot \leftarrow \mathit{RemPr}(\mathit{Roll}(d), x), x \leq 0$$

Let Π' *denotes the further translated ASP program with weak constraints, which is obtained from* $\mathrm{plog2lpmln}(\Pi)$ *by applying the translation* $\mathrm{lpmln2wc}^{\mathrm{pnt,clingo}}_{\mathrm{simp}}$ *and the simplification for hard rules.* Π' *is as follows:*

$\% * * * *$ *Declaration Part* $* * * *$

$dice(d_1; d_2).$

$number(1..6).$

$\% * * * * \tau(\Pi) * * * *$

$owner(d_1, mike).$

$owner(d_2, john).$

$even(D, \mathbf{t}) \leftarrow roll(D, N), N\ mod\ 2 = 0.$

$even(D, \mathbf{f}) \leftarrow not\ even(D, \mathbf{t}), dice(D).$

$\leftarrow roll(D, N_1), roll(D, N_2), N_1 \neq N_2.$

$\leftarrow even(D, \mathbf{t}), even(D, \mathbf{f}).$

$\leftarrow owner(D, P_1), owner(D, P_2), P_1 \neq P_2.$

$1\{roll(D, N) : number(N)\}1 \leftarrow not\ intervene(roll(D)), dice(D).$

$\% * * * *$ *Possible Atoms* $* * * *$

$poss(r(D), roll(D, N)) \leftarrow not\ intervene(roll(D)), dice(D), number(N).$

$\% **** \ \textit{Assigned Probability} ****$

$possWithAssPr(r(D), owner(D, mike), roll(D, 5)) \leftarrow$

$\qquad\qquad\qquad poss(r(D), roll(D, 5)), owner(D, mike).$

$assPr(r(D), owner(D, mike), roll(D, 5)) \leftarrow$

$\qquad\qquad\qquad roll(D, 5), possWithAssPr(r(D), owner(D, mike), roll(D, 5)).$

$possWithAssPr(roll(D, 5)) \leftarrow possWithAssPr(r(D), owner(D, mike), roll(D, 5)).$

$possWithAssPr(r(D), owner(D, mike), roll(D, 6)) \leftarrow$

$\qquad\qquad\qquad poss(r(D), roll(D, 6)), owner(D, mike).$

$assPr(r(D), owner(D, mike), roll(D, 6)) \leftarrow$

$\qquad\qquad\qquad roll(D, 6), possWithAssPr(r(D), owner(D, mike), roll(D, 6)).$

$possWithAssPr(roll(D, 6)) \leftarrow possWithAssPr(r(D), owner(D, mike), roll(D, 6)).$

$\% **** \ \textit{Denominator for Default Probability} ****$

$possWithDefPr(roll(D, N)) \leftarrow poss(r(D), roll(D, N)), not\ possWithAssPr(roll(D, N)).$

$numDefPr(roll(D), X) \leftarrow roll(D, N), possWithDefPr(roll(D, N)),$

$\qquad\qquad\qquad X = \#count\{Y : possWithDefPr(roll(D, Y))\}.$

$\% **** \ \textit{Numerator for Default Probability} ****$

$remPr(roll(D), X) \leftarrow roll(D, N), possWithDefPr(roll(D, N)), X = 1 - Y,$

$\qquad\qquad\qquad Y = \#sum\{\frac{1}{4} : possWithAssPr(r(D), owner(D, mike), roll(D, 5));$

$\qquad\qquad\qquad\qquad \frac{1}{5} : possWithAssPr(r(D), owner(D, mike), roll(D, 6))\}.$

$totalDefPr(roll(D), X) \leftarrow remPr(roll(D), X), X > 0.$

$\leftarrow remPr(roll(D), X), x \leq 0.$

$\% **** \ \textit{Weak Constraints} ****$

$:\sim \ assPr(r(D), owner(D, mike), roll(D, 5)). \ [-ln(\frac{1}{4})]$

$:\sim \ assPr(r(D), owner(D, mike), roll(D, 6)). \ [-ln(\frac{1}{5})]$

$:\sim \ numDefPr(roll(D), X). \ [-ln(1/X)]$

$:\sim \ totalDefPr(roll(D), X). \ [-ln(X)]$

It is easy to check that there are $6 \times 6 = 36$ *possible worlds of the P-log program:*

$$\begin{aligned}
W_1 = \{&Roll(D_1) = 1, Roll(D_2) = 1, Even(D_1) = \mathbf{f}, Even(D_2) = \mathbf{f},\\
&Owner(D_1) = Mike, Owner(D_2) = John\}\\
W_2 = \{&Roll(D_1) = 1, Roll(D_2) = 2, Even(D_1) = \mathbf{f}, Even(D_2) = \mathbf{t},\\
&Owner(D_1) = Mike, Owner(D_2) = John\}\\
\ldots&\\
W_{36} = \{&Roll(D_1) = 6, Roll(D_2) = 6, Even(D_1) = \mathbf{t}, Even(D_2) = \mathbf{t},\\
&Owner(D_1) = Mike, Owner(D_2) = John\}
\end{aligned}$$

The probability of each atom (that is possible in W_i*) to happen is: (*$i = 1, \ldots, 36$*,* $n = 1, \ldots, 6$*)*

$$P(W_i, Roll(D_1) = n) = \begin{cases} \frac{1}{4} & \textit{if } n = 5,\\ \frac{1}{5} & \textit{if } n = 6,\\ \frac{11}{80} & \textit{otherwise;} \end{cases}$$

$$P(W_i, Roll(D_2) = n) = \frac{1}{6};$$

then

$$\begin{aligned}\hat{\mu}_\Pi(W_1) = \;& P(W_1, \mathit{Roll}(D_1)=1) \times P(W_1, \mathit{Roll}(D_2)=1)\\ = \;& \tfrac{11}{80} \times \tfrac{1}{6} = \tfrac{11}{480},\\ \dots & \\ \hat{\mu}_\Pi(W_{36}) = \;& P(W_{36}, \mathit{Roll}(D_1)=6) \times P(W_{36}, \mathit{Roll}(D_2)=6)\\ = \;& \tfrac{1}{5} \times \tfrac{1}{6} = \tfrac{1}{30}.\end{aligned}$$

As for the translated $\mathrm{LP}^{\mathrm{MLN}}$ *program* plog2lpmln(Π), *it has 36 probabilistic stable models* $I_1, \dots, I_{36}$, *each of which is an extension of* $W_1, \dots, W_{36}$ *respectively, and satisfies the following atoms:*

$\mathit{Poss}_{r(d)}(\mathit{Roll}(d)=n)$	*for* $d = D_1, D_2$ *and* $n = 1, \dots, 6$;
$\mathit{PossWithAssPr}_{r(D_1),C_{D_1}}(\mathit{Roll}(D_1)=n)$	*for* $n = 5, 6$;
$\mathit{PossWithAssPr}(\mathit{Roll}(D_1)=n)$	*for* $n = 5, 6$;
$\mathit{PossWithDefPr}(\mathit{Roll}(D_1)=n)$	*for* $n = 1, 2, 3, 4$;
$\mathit{PossWithDefPr}(\mathit{Roll}(D_2)=n)$	*for* $n = 1, \dots, 6$;
$\mathit{NumDefPr}(\mathit{Roll}(D_2), 6)$;	
$\mathit{RemPr}(\mathit{Roll}(D_2), 1)$;	
$\mathit{TotalDefPr}(\mathit{Roll}(D_2), 1)$.	

In addition, for $i = 1, \dots, 36$,

- *if* I_i *satisfies* $\mathit{Roll}(D_1) = 5$, I_i *also satisfies*

$$\{\mathit{AssPr}_{r(D_1),C_{D_1}}(\mathit{Roll}(D_1)=5)\},$$

  and $W_{\mathrm{plog2lpmln}(\Pi)}(I_i)$ *(the unnormalized weight of* I_i*) is* $\frac{1}{6} \times 1 \times \frac{1}{4} = \frac{1}{24}$*;*

- *if* I_i *satisfies* $\mathit{Roll}(D_1) = 6$, I_i *also satisfies*

$$\{\mathit{AssPr}_{r(D_1),C_{D_1}}(\mathit{Roll}(D_1)=6)\},$$

  and $W_{\mathrm{plog2lpmln}(\Pi)}(I_i) = \frac{1}{6} \times 1 \times \frac{1}{5} = \frac{1}{30}$*;*

- *otherwise,* I_i *also satisfies*

  $\{NumDefPr(Roll(D_1), 4), RemPr(Roll(D_1), 0.55), TotalDefPr(Roll(D_1), 0.55)\}$,

  and $W_{\text{plog2lpmln}(\Pi)}(I_i) = \frac{1}{6} \times 1 \times \frac{1}{4} \times 0.55 = \frac{11}{480}$.

As we see, the unnormalized weight of I_i *under* plog2lpmln(Π) *exactly equals to the unnormalized probability of* W_i *under* Π*. Thus after the translation, the probability distribution still remains the same (*$P_{\text{plog2lpmln}(\Pi)}(\phi(W_i)) = \mu_\Pi(W_i)$*).*

As for the further translated ASP program with weak constraints, Π'*, if we ignore the syntax difference between* Π' *and* plog2lpmln(Π)*, and ingore the newly added declaration atoms (*$dice(d_1; d_2)$*,* $number(1..6)$*), the stable models of* Π' *are exactly* $I_1, \dots, I_{36}$.

The penalty of each stable model of Π' *at level 0 is:*

$$\begin{aligned} Penalty_\Pi(I_1, 0) = &\ -ln(\tfrac{1}{6}) - ln(1) - ln(\tfrac{1}{4}) - ln(0.55) = -ln(\tfrac{11}{480}), \\ &\dots \\ Penalty_\Pi(I_{36}, 0) = &\ -ln(\tfrac{1}{6}) - ln(1) - ln(\tfrac{1}{5}) = -ln(\tfrac{1}{30}). \end{aligned}$$

Thus the most probable stable models of plog2lpmln(Π) *are exactly the optimal stable models of* Π'.

5.3 Simpson's Paradox

Example 8 *Consider Simpson's Paradox from (Baral* et al.*, 2009). A patient is thinking about trying an experimental drug and decides to consult a doctor. The doctor has the following statistics of the recovery rates that have been observed among males and females, taking and not taking the drug.*

$Males:$

	fraction of population	*recovery rate*
$drug$	3/8	60%
$\neg drug$	1/8	70%

$Females:$

	fraction of population	*recovery rate*
$drug$	1/8	20%
$\neg drug$	3/8	30%

The following P-log program Π *describes this problem.*

$$[r_1]\, random(Male).$$
$$[r_2]\, random(Recover).$$
$$[r_3]\, random(Drug).$$

$$pr_{r_1}(Male) = 0.5.$$
$$pr_{r_2}(Recover \mid Male, Drug) = 0.6.$$
$$pr_{r_2}(Recover \mid Male, \sim\!Drug) = 0.7.$$
$$pr_{r_2}(Recover \mid \sim\!Male, Drug) = 0.2.$$
$$pr_{r_2}(Recover \mid \sim\!Male, \sim\!Drug) = 0.3.$$
$$pr_{r_3}(Drug \mid Male) = 0.75.$$
$$pr_{r_3}(Drug \mid \sim\!Male) = 0.25.$$

The translated $\mathrm{LP}^{\mathrm{MLN}}$ *encoding* $\mathrm{plog2lpmln}(\Pi)$ *is as follows: (where* $b \in \{\mathbf{t}, \mathbf{f}\}$*, and we denote the conditions "*$Male, Drug$*", "*$Male, \sim\!Drug$*", "*$\sim\!Male, Drug$*", "*$\sim\!Male, \sim\!Drug$*", "*$Male$*", and "*$\sim\!Male$*" by* $C_1, C_2, \dots, C_6$ *respectively)*

$$// **** \ \tau(\Pi) \ ****$$
$$\alpha : \ \leftarrow Male = \mathbf{t}, Male = \mathbf{f}$$
$$\alpha : \ \leftarrow Recover = \mathbf{t}, Recover = \mathbf{f}$$
$$\alpha : \ \leftarrow Drug = \mathbf{t}, Drug = \mathbf{f}$$

$$\alpha : \ Male = \mathbf{t}; Male = \mathbf{f} \leftarrow not \ \ Intervene(Male)$$
$$\alpha : \ Recover = \mathbf{t}; Recover = \mathbf{f} \leftarrow not \ \ Intervene(Recover)$$
$$\alpha : \ Drug = \mathbf{t}; Drug = \mathbf{f} \leftarrow not \ \ Intervene(Drug)$$

$//**** \; Possible \; Atoms ****$

$\alpha: \; Poss_{r_1}(Male = b) \leftarrow not \;\; Intervene(Male)$

$\alpha: \; Poss_{r_2}(Recover = b) \leftarrow not \;\; Intervene(Recover)$

$\alpha: \; Poss_{r_3}(Drug = b) \leftarrow not \;\; Intervene(Drug)$

$//**** \; Assigned \; Probability ****$

$\alpha: \; PossWithAssPr_{r_1}(Male = \mathbf{t}) \leftarrow Poss_{r_1}(Male = \mathbf{t})$

$\alpha: \; AssPr_{r_1}(Male = \mathbf{t}) \leftarrow Male = \mathbf{t}, PossWithAssPr_{r_1}(Male = \mathbf{t})$

$ln(0.5): \; \bot \leftarrow not \;\; AssPr_{r_1}(Male = \mathbf{t})$

$\alpha: \; PossWithAssPr(Male = \mathbf{t}) \leftarrow PossWithAssPr_{r_1}(Male = \mathbf{t})$

$\alpha: \; PossWithAssPr_{r_2,C_1}(Recover = \mathbf{t}) \leftarrow Poss_{r_2}(Recover = \mathbf{t}), Male = \mathbf{t}, Drug = \mathbf{t}$

$\alpha: \; AssPr_{r_2,C_1}(Recover = \mathbf{t}) \leftarrow Recover = \mathbf{t}, PossWithAssPr_{r_2,C_1}(Recover = \mathbf{t})$

$ln(0.6): \; \bot \leftarrow not \;\; AssPr_{r_2,C_1}(Recover = \mathbf{t})$

$\alpha: \; PossWithAssPr(Recover = \mathbf{t}) \leftarrow PossWithAssPr_{r_2,C_1}(Recover = \mathbf{t})$

$\alpha: \; PossWithAssPr_{r_2,C_2}(Recover = \mathbf{t}) \leftarrow Poss_{r_2}(Recover = \mathbf{t}), Male = \mathbf{t}, Drug = \mathbf{f}$

$\alpha: \; AssPr_{r_2,C_2}(Recover = \mathbf{t}) \leftarrow Recover = \mathbf{t}, PossWithAssPr_{r_2,C_2}(Recover = \mathbf{t})$

$ln(0.7): \; \bot \leftarrow not \;\; AssPr_{r_2,C_2}(Recover = \mathbf{t})$

$\alpha: \; PossWithAssPr(Recover = \mathbf{t}) \leftarrow PossWithAssPr_{r_2,C_2}(Recover = \mathbf{t})$

$\alpha: \; PossWithAssPr_{r_2,C_3}(Recover = \mathbf{t}) \leftarrow Poss_{r_2}(Recover = \mathbf{t}), Male = \mathbf{f}, Drug = \mathbf{t}$

$\alpha: \; AssPr_{r_2,C_3}(Recover = \mathbf{t}) \leftarrow Recover = \mathbf{t}, PossWithAssPr_{r_2,C_3}(Recover = \mathbf{t})$

$ln(0.2): \; \bot \leftarrow not \;\; AssPr_{r_2,C_3}(Recover = \mathbf{t})$

$\alpha: \; PossWithAssPr(Recover = \mathbf{t}) \leftarrow PossWithAssPr_{r_2,C_3}(Recover = \mathbf{t})$

$\alpha: \; PossWithAssPr_{r_2,C_4}(Recover = \mathbf{t}) \leftarrow Poss_{r_2}(Recover = \mathbf{t}), Male = \mathbf{f}, Drug = \mathbf{f}$

$\alpha: \; AssPr_{r_2,C_4}(Recover = \mathbf{t}) \leftarrow Recover = \mathbf{t}, PossWithAssPr_{r_2,C_4}(Recover = \mathbf{t})$

$ln(0.3): \; \bot \leftarrow not \;\; AssPr_{r_2,C_4}(Recover = \mathbf{t})$

$\alpha: \; PossWithAssPr(Recover = \mathbf{t}) \leftarrow PossWithAssPr_{r_2,C_4}(Recover = \mathbf{t})$

$// **** \textit{Assigned Probability Continued} ****$

$\alpha: \; PossWithAssPr_{r_3,C_5}(Drug = \mathbf{t}) \leftarrow Poss_{r_3}(Drug = \mathbf{t}), Male = \mathbf{t}$

$\alpha: \; AssPr_{r_3,C_5}(Drug = \mathbf{t}) \leftarrow Drug = \mathbf{t}, PossWithAssPr_{r_3,C_5}(Drug = \mathbf{t})$

$ln(0.75): \; \bot \leftarrow not \;\; AssPr_{r_3,C_5}(Drug = \mathbf{t})$

$\alpha: \; PossWithAssPr(Drug = \mathbf{t}) \leftarrow PossWithAssPr_{r_3,C_5}(Drug = \mathbf{t})$

$\alpha: \; PossWithAssPr_{r_3,C_6}(Drug = \mathbf{t}) \leftarrow Poss_{r_3}(Drug = \mathbf{t}), Male = \mathbf{f}$

$\alpha: \; AssPr_{r_3,C_6}(Drug = \mathbf{t}) \leftarrow Drug = \mathbf{t}, PossWithAssPr_{r_3,C_6}(Drug = \mathbf{t})$

$ln(0.25): \; \bot \leftarrow not \;\; AssPr_{r_3,C_6}(Drug = \mathbf{t})$

$\alpha: \; PossWithAssPr(Drug = \mathbf{t}) \leftarrow PossWithAssPr_{r_3,C_6}(Drug = \mathbf{t})$

$// **** \textit{Denominator for Default Probability} ****$

$\alpha: \; PossWithDefPr(Male = b) \leftarrow Poss_{r_1}(Male = b), not \;\; PossWithAssPr(Male = b)$

$\alpha: \; NumDefPr(Male, x) \leftarrow Male = b, PossWithDefPr(Male = b),$

$$x = \#count\{y : PossWithDefPr(Male = y)\}$$

$ln(\frac{1}{m}): \; \bot \leftarrow not \;\; NumDefPr(Male, m) \qquad\qquad (m \in \{2\})$

$\alpha: \; PossWithDefPr(Recover = b) \leftarrow$

$$Poss_{r_2}(Recover = b), not \;\; PossWithAssPr(Recover = b)$$

$\alpha: \; NumDefPr(Recover, x) \leftarrow Recover = b, PossWithDefPr(Recover = b),$

$$x = \#count\{y : PossWithDefPr(Recover = y)\}$$

$ln(\frac{1}{m}): \; \bot \leftarrow not \;\; NumDefPr(Recover, m) \qquad\qquad (m \in \{2\})$

$\alpha: \; PossWithDefPr(Drug = b) \leftarrow Poss_{r_3}(Drug = b), not \;\; PossWithAssPr(Drug = b)$

$\alpha: \; NumDefPr(Drug, x) \leftarrow Drug = b, PossWithDefPr(Drug = b),$

$$x = \#count\{y : PossWithDefPr(Drug = y)\}$$

$ln(\frac{1}{m}): \; \bot \leftarrow not \;\; NumDefPr(Drug, m) \qquad\qquad (m \in \{2\})$

$//**** \textit{Numerator for Default Probability} ****$

$$\alpha: \ RemPr(Male, 1-y) \leftarrow Male = b, PossWithDefPr(Male = b),$$
$$y = \#sum\{0.5 : PossWithAssPr_{r_1}(Male = \mathbf{t})\}$$

$$\alpha: \ TotalDefPr(Male, x) \leftarrow RemPr(Male, x), x > 0$$

$$ln(x): \ \bot \leftarrow not \ \ TotalDefPr(Male, x)$$

$$\alpha: \ \bot \leftarrow RemPr(Male, x), x \leq 0$$

$$\alpha: \ RemPr(Recover, 1-y) \leftarrow Recover = b, PossWithDefPr(Recover = b),$$
$$\begin{aligned} y = \#sum\{0.6 &: PossWithAssPr_{r_2,C_1}(Recover = \mathbf{t}); \\ 0.7 &: PossWithAssPr_{r_2,C_2}(Recover = \mathbf{t}); \\ 0.2 &: PossWithAssPr_{r_2,C_3}(Recover = \mathbf{t}); \\ 0.3 &: PossWithAssPr_{r_2,C_4}(Recover = \mathbf{t})\} \end{aligned}$$

$$\alpha: \ TotalDefPr(Recover, x) \leftarrow RemPr(Recover, x), x > 0$$

$$ln(x): \ \bot \leftarrow not \ \ TotalDefPr(Recover, x)$$

$$\alpha: \ \bot \leftarrow RemPr(Recover, x), x \leq 0$$

$$\alpha: \ RemPr(Drug, 1-y) \leftarrow Drug = b, PossWithDefPr(Drug = b),$$
$$\begin{aligned} y = \#sum\{0.75 &: PossWithAssPr_{r_3,C_5}(Drug = \mathbf{t}); \\ 0.25 &: PossWithAssPr_{r_3,C_6}(Drug = \mathbf{t})\} \end{aligned}$$

$$\alpha: \ TotalDefPr(Drug, x) \leftarrow RemPr(Drug, x), x > 0$$

$$ln(x): \ \bot \leftarrow not \ \ TotalDefPr(Drug, x)$$

$$\alpha: \ \bot \leftarrow RemPr(Drug, x), x \leq 0$$

Let Π' *denotes the further translated ASP program with weak constraints, which is obtained from* $\text{plog2lpmln}(\Pi)$ *by applying the translation* $\text{lpmln2wc}^{\text{pnt,clingo}}_{\text{simp}}$ *and the simplification for hard rules.* Π' *is as follows:*

$\% **** \textit{ Declaration Part} ****$

$boolean(\mathbf{t};\mathbf{f}).$

$\% **** \ \tau(\Pi) \ ****$

$\leftarrow male(\mathbf{t}), male(\mathbf{f}).$

$\leftarrow recover(\mathbf{t}), recover(\mathbf{f}).$

$\leftarrow drug(\mathbf{t}), drug(\mathbf{f}).$

$male(\mathbf{t}); male(\mathbf{f}) \leftarrow not\ \ intervene(male).$

$recover(\mathbf{t}); recover(\mathbf{f}) \leftarrow not\ \ intervene(recover).$

$drug(\mathbf{t}); drug(\mathbf{f}) \leftarrow not\ \ intervene(drug).$

$\% **** \textit{ Possible Atoms} ****$

$poss(r(1), male(B)) \leftarrow not\ \ intervene(male), boolean(B).$

$poss(r(2), recover(B)) \leftarrow not\ \ intervene(recover), boolean(B).$

$poss(r(3), drug(B)) \leftarrow not\ \ intervene(drug), boolean(B).$

$\% **** \textit{Assigned Probability} ****$

$possWithAssPr(r(1), male(\mathbf{t})) \leftarrow poss(r(1), male(\mathbf{t})).$

$assPr(r(1), male(\mathbf{t})) \leftarrow male(\mathbf{t}), possWithAssPr(r(1), male(\mathbf{t})).$

$possWithAssPr(male(\mathbf{t})) \leftarrow possWithAssPr(r(1), male(\mathbf{t})).$

$possWithAssPr(r(2), c(1), recover(\mathbf{t})) \leftarrow poss(r(2), recover(\mathbf{t})), male(\mathbf{t}), drug(\mathbf{t}).$

$assPr(r(2), c(1), recover(\mathbf{t})) \leftarrow recover(\mathbf{t}), possWithAssPr(r(2), c(1), recover(\mathbf{t})).$

$possWithAssPr(recover(\mathbf{t})) \leftarrow possWithAssPr(r(2), c(1), recover(\mathbf{t})).$

$possWithAssPr(r(2), c(2), recover(\mathbf{t})) \leftarrow poss(r(2), recover(\mathbf{t})), male(\mathbf{t}), drug = \mathbf{f}.$

$assPr(r(2), c(2), recover(\mathbf{t})) \leftarrow recover(\mathbf{t}), possWithAssPr(r(2), c(2), recover(\mathbf{t})).$

$possWithAssPr(recover(\mathbf{t})) \leftarrow possWithAssPr(r(2), c(2), recover(\mathbf{t})).$

$possWithAssPr(r(2), c(3), recover(\mathbf{t})) \leftarrow poss(r(2), recover(\mathbf{t})), male = \mathbf{f}, drug(\mathbf{t}).$

$assPr(r(2), c(3), recover(\mathbf{t})) \leftarrow recover(\mathbf{t}), possWithAssPr(r(2), c(3), recover(\mathbf{t})).$

$possWithAssPr(recover(\mathbf{t})) \leftarrow possWithAssPr(r(2), c(3), recover(\mathbf{t})).$

$possWithAssPr(r(2), c(4), recover(\mathbf{t})) \leftarrow poss(r(2), recover(\mathbf{t})), male = \mathbf{f}, drug = \mathbf{f}.$

$assPr(r(2), c(4), recover(\mathbf{t})) \leftarrow recover(\mathbf{t}), possWithAssPr(r(2), c(4), recover(\mathbf{t})).$

$possWithAssPr(recover(\mathbf{t})) \leftarrow possWithAssPr(r(2), c(4), recover(\mathbf{t})).$

$possWithAssPr(r(3), c(5), drug(\mathbf{t})) \leftarrow poss(r(3), drug(\mathbf{t})), male(\mathbf{t}).$

$assPr(r(3), c(5), drug(\mathbf{t})) \leftarrow drug(\mathbf{t}), possWithAssPr(r(3), c(5), drug(\mathbf{t})).$

$possWithAssPr(drug(\mathbf{t})) \leftarrow possWithAssPr(r(3), c(5), drug(\mathbf{t})).$

$possWithAssPr(r(3), c(6), drug(\mathbf{t})) \leftarrow poss(r(3), drug(\mathbf{t})), male = \mathbf{f}.$

$assPr(r(3), c(6), drug(\mathbf{t})) \leftarrow drug(\mathbf{t}), possWithAssPr(r(3), c(6), drug(\mathbf{t})).$

$possWithAssPr(drug(\mathbf{t})) \leftarrow possWithAssPr(r(3), c(6), drug(\mathbf{t})).$

$\% **** \textit{ Denominator for Default Probability} ****$

$possWithDefPr(male(B)) \leftarrow poss(r(1), male(B)), not\ possWithAssPr(male(B)).$

$numDefPr(male, X) \leftarrow male(B), possWithDefPr(male(B)),$

$$X = \#count\{Y : possWithDefPr(male(Y))\}.$$

$possWithDefPr(recover(B)) \leftarrow poss(r(2), recover(B)), not\ possWithAssPr(recover(B)).$

$numDefPr(recover, X) \leftarrow recover(B), possWithDefPr(recover(B)),$

$$X = \#count\{Y : possWithDefPr(recover(Y))\}.$$

$possWithDefPr(drug(B)) \leftarrow poss(r(3), drug(B)), not\ possWithAssPr(drug(B)).$

$numDefPr(drug, X) \leftarrow drug(B), possWithDefPr(drug(B)),$

$$X = \#count\{Y : possWithDefPr(drug(Y))\}.$$

$\% **** \textit{ Numerator for Default Probability} ****$

$remPr(male, X) \leftarrow male(B), possWithDefPr(male(B)), X = 1 - Y,$

$$Y = \#sum\{0.5 : possWithAssPr(r(1), male(\mathbf{t}))\}.$$

$totalDefPr(male, X) \leftarrow remPr(male, X), X > 0.$

$\bot \leftarrow remPr(male, X), X \leq 0.$

$remPr(recover, X) \leftarrow recover(B), possWithDefPr(recover(B)), X = 1 - Y,$

$$\begin{aligned} Y = \#sum\{&0.6 : possWithAssPr(r(2), c(1), recover(\mathbf{t}));\\ &0.7 : possWithAssPr(r(2), c(2), recover(\mathbf{t}));\\ &0.2 : possWithAssPr(r(2), c(3), recover(\mathbf{t}));\\ &0.3 : possWithAssPr(r(2), c(4), recover(\mathbf{t}))\}.\end{aligned}$$

$totalDefPr(recover, X) \leftarrow remPr(recover, X), X > 0.$

$\bot \leftarrow remPr(recover, X), X \leq 0.$

$remPr(drug, X) \leftarrow drug(B), possWithDefPr(drug(B)), X = 1 - Y,$

$$\begin{aligned} Y = \#sum\{&0.75 : possWithAssPr(r(3), c(5), drug(\mathbf{t}));\\ &0.25 : possWithAssPr(r(3), c(6), drug(\mathbf{t}))\}.\end{aligned}$$

$totalDefPr(drug, X) \leftarrow remPr(drug, X), X > 0.$

$\bot \leftarrow remPr(drug, X), X \leq 0.$

$\% * * * *$ *Weak Constraints* $* * * *$

$:\sim\ assPr(r(1), male(\mathbf{t})).\ [-ln(0.5)]$

$:\sim\ assPr(r(2), c(1), recover(\mathbf{t})).\ [-ln(0.6)]$

$:\sim\ assPr(r(2), c(2), recover(\mathbf{t})).\ [-ln(0.7)]$

$:\sim\ assPr(r(2), c(3), recover(\mathbf{t})).\ [-ln(0.2)]$

$:\sim\ assPr(r(2), c(4), recover(\mathbf{t})).\ [-ln(0.3)]$

$:\sim\ assPr(r(3), c(5), drug(\mathbf{t})).\ [-ln(0.75)]$

$:\sim\ assPr(r(3), c(6), drug(\mathbf{t})).\ [-ln(0.25)]$

$:\sim\ numDefPr(A, X).\ \left[-ln(\frac{1}{X})\right] \qquad (A \in \{male, recover, drug\}, X \in \{2\})$

$:\sim\ totalDefPr(A, X).\ [-ln(X)]$

It is easy to check that there are $2^3 = 8$ *possible worlds of the P-log program* Π*:*

$$W_1 = \{Male = \mathbf{f}, Recover = \mathbf{f}, Drug = \mathbf{f}\}$$
$$W_2 = \{Male = \mathbf{f}, Recover = \mathbf{f}, Drug = \mathbf{t}\}$$
$$\dots$$
$$W_8 = \{Male = \mathbf{t}, Recover = \mathbf{t}, Drug = \mathbf{t}\},$$

and their unnormalized probabilities are as follows:

$$\begin{aligned}
\hat{\mu}_\Pi(W_1) = &\quad P(W_1, Male = \mathbf{f}) \times P(W_1, Recover = \mathbf{f}) \times P(W_1, Drug = \mathbf{f}) \\
= &\quad \tfrac{1}{2} \times \tfrac{7}{10} \times \tfrac{3}{4} = \tfrac{21}{80}, \\
\hat{\mu}_\Pi(W_2) = &\quad P(W_2, Male = \mathbf{f}) \times P(W_2, Recover = \mathbf{f}) \times P(W_2, Drug = \mathbf{t}) \\
= &\quad \tfrac{1}{2} \times \tfrac{4}{5} \times \tfrac{1}{4} = \tfrac{1}{10}, \\
\dots & \\
\hat{\mu}_\Pi(W_8) = &\quad P(W_8, Male = \mathbf{t}) \times P(W_8, Recover = \mathbf{t}) \times P(W_8, Drug = \mathbf{t}) \\
= &\quad \tfrac{1}{2} \times \tfrac{3}{5} \times \tfrac{3}{4} = \tfrac{9}{40}.
\end{aligned}$$

As for the translated LP$^{\text{MLN}}$ *program* plog2lpmln(Π)*, it has 8 probabilistic stable models* $I_1, I_2, \dots, I_8$*, each of which is an extension of* $W_1, W_2, \dots, W_8$ *respectively, and* *satisfies the following atoms:*

$$
\begin{array}{ll}
\mathit{Poss}_{r_1}(\mathit{Male} = b) & \textit{for } b \in \{\mathbf{t}, \mathbf{f}\}, \\
\mathit{Poss}_{r_2}(\mathit{Recover} = b) & \textit{for } b \in \{\mathbf{t}, \mathbf{f}\}, \\
\mathit{Poss}_{r_3}(\mathit{Drug} = b) & \textit{for } b \in \{\mathbf{t}, \mathbf{f}\}, \\
\mathit{PossWithAssPr}_{r_1}(\mathit{Male} = \mathbf{t}), & \\
\mathit{PossWithAssPr}(\mathit{Male} = \mathbf{t}), & \\
\mathit{PossWithDefPr}(\mathit{Male} = \mathbf{f}), & \\
\mathit{PossWithAssPr}(\mathit{Recover} = \mathbf{t}), & \\
\mathit{PossWithDefPr}(\mathit{Recover} = \mathbf{f}), & \\
\mathit{PossWithAssPr}(\mathit{Drug} = \mathbf{t}), & \\
\mathit{PossWithDefPr}(\mathit{Drug} = \mathbf{f}). &
\end{array}
$$

In addition,

$$
\begin{array}{rl}
I_1 \vDash \{ & \mathit{NumDefPr}(\mathit{Male}, 1), \mathit{RemPr}(\mathit{Male}, 0.5), \mathit{TotalDefPr}(\mathit{Male}, 0.5), \\
& \mathit{PossWithAssPr}_{r_2,C_4}(\mathit{Recover} = \mathbf{t}), \\
& \mathit{NumDefPr}(\mathit{Recover}, 1), \mathit{RemPr}(\mathit{Recover}, 0.7), \mathit{TotalDefPr}(\mathit{Recover}, 0.7), \\
& \mathit{PossWithAssPr}_{r_3,C_6}(\mathit{Drug} = \mathbf{t}), \\
& \mathit{NumDefPr}(\mathit{Drug}, 1), \mathit{RemPr}(\mathit{Drug}, 0.75), \mathit{TotalDefPr}(\mathit{Drug}, 0.75) \quad \}
\end{array}
$$

$$
\begin{array}{rl}
I_2 \vDash \{ & \mathit{NumDefPr}(\mathit{Male}, 1), \mathit{RemPr}(\mathit{Male}, 0.5), \mathit{TotalDefPr}(\mathit{Male}, 0.5), \\
& \mathit{PossWithAssPr}_{r_2,C_3}(\mathit{Recover} = \mathbf{t}), \\
& \mathit{NumDefPr}(\mathit{Recover}, 1), \mathit{RemPr}(\mathit{Recover}, 0.8), \mathit{TotalDefPr}(\mathit{Recover}, 0.8), \\
& \mathit{PossWithAssPr}_{r_3,C_6}(\mathit{Drug} = \mathbf{t}), \\
& \mathit{AssPr}_{r_3,C_6}(\mathit{Drug} = \mathbf{t}) \quad \}
\end{array}
$$

$$\begin{aligned}
I_3 \models \{ \quad & \mathit{NumDefPr}(Male, 1), \mathit{RemPr}(Male, 0.5), \mathit{TotalDefPr}(Male, 0.5),\\
& \mathit{PossWithAssPr}_{r_2,C_4}(\mathit{Recover} = \mathbf{t}),\\
& \mathit{AssPr}_{r_2,C_4}(\mathit{Recover} = \mathbf{t}),\\
& \mathit{PossWithAssPr}_{r_3,C_6}(\mathit{Drug} = \mathbf{t}),\\
& \mathit{NumDefPr}(\mathit{Drug}, 1), \mathit{RemPr}(\mathit{Drug}, 0.75), \mathit{TotalDefPr}(\mathit{Drug}, 0.75) \quad \}
\end{aligned}$$

$$\begin{aligned}
I_4 \models \{ \quad & \mathit{NumDefPr}(Male, 1), \mathit{RemPr}(Male, 0.5), \mathit{TotalDefPr}(Male, 0.5),\\
& \mathit{PossWithAssPr}_{r_2,C_3}(\mathit{Recover} = \mathbf{t}),\\
& \mathit{AssPr}_{r_2,C_3}(\mathit{Recover} = \mathbf{t}),\\
& \mathit{PossWithAssPr}_{r_3,C_6}(\mathit{Drug} = \mathbf{t}),\\
& \mathit{AssPr}_{r_3,C_6}(\mathit{Drug} = \mathbf{t}) \quad \}
\end{aligned}$$

$$\begin{aligned}
I_5 \models \{ \quad & \mathit{AssPr}_{r_1}(Male = \mathbf{t}),\\
& \mathit{PossWithAssPr}_{r_2,C_2}(\mathit{Recover} = \mathbf{t}),\\
& \mathit{NumDefPr}(\mathit{Recover}, 1), \mathit{RemPr}(\mathit{Recover}, 0.3), \mathit{TotalDefPr}(\mathit{Recover}, 0.3),\\
& \mathit{PossWithAssPr}_{r_3,C_5}(\mathit{Drug} = \mathbf{t}),\\
& \mathit{NumDefPr}(\mathit{Drug}, 1), \mathit{RemPr}(\mathit{Drug}, 0.25), \mathit{TotalDefPr}(\mathit{Drug}, 0.25) \quad \}
\end{aligned}$$

$$\begin{aligned}
I_6 \models \{ \quad & \mathit{AssPr}_{r_1}(Male = \mathbf{t}),\\
& \mathit{PossWithAssPr}_{r_2,C_1}(\mathit{Recover} = \mathbf{t}),\\
& \mathit{NumDefPr}(\mathit{Recover}, 1), \mathit{RemPr}(\mathit{Recover}, 0.4), \mathit{TotalDefPr}(\mathit{Recover}, 0.4),\\
& \mathit{PossWithAssPr}_{r_3,C_5}(\mathit{Drug} = \mathbf{t}),\\
& \mathit{AssPr}_{r_3,C_5}(\mathit{Drug} = \mathbf{t}) \quad \}
\end{aligned}$$

$$\begin{aligned}
I_7 \models \{ \quad & \mathit{AssPr}_{r_1}(Male = \mathbf{t}),\\
& \mathit{PossWithAssPr}_{r_2,C_2}(\mathit{Recover} = \mathbf{t}),\\
& \mathit{AssPr}_{r_2,C_2}(\mathit{Recover} = \mathbf{t}),\\
& \mathit{PossWithAssPr}_{r_3,C_5}(\mathit{Drug} = \mathbf{t}),\\
& \mathit{NumDefPr}(\mathit{Drug}, 1), \mathit{RemPr}(\mathit{Drug}, 0.25), \mathit{TotalDefPr}(\mathit{Drug}, 0.25) \quad \}
\end{aligned}$$

$$
\begin{aligned}
I_8 \vDash \{ \quad & \mathit{AssPr}_{r_1}(\mathit{Male} = \mathbf{t}), \\
& \mathit{PossWithAssPr}_{r_2,C_1}(\mathit{Recover} = \mathbf{t}), \\
& \mathit{AssPr}_{r_2,C_1}(\mathit{Recover} = \mathbf{t}), \\
& \mathit{PossWithAssPr}_{r_3,C_5}(\mathit{Drug} = \mathbf{t}), \\
& \mathit{AssPr}_{r_3,C_5}(\mathit{Drug} = \mathbf{t}) \quad \}
\end{aligned}
$$

It is easy to check that $I_i = \phi(W_i)$ for $i \in \{1, \dots, 8\}$. The unnormalized weight $W_{\text{plog2lpmln}(\Pi)}(I_i)$ of each probabilistic stable model I_i is shown below:

$$
\begin{aligned}
W_{\text{plog2lpmln}(\Pi)}(I_1) = \quad & w(\mathit{TotalDefPr}(\mathit{Male}, 0.5)) \times w(\mathit{TotalDefPr}(\mathit{Recover}, 0.7)) \times \\
& w(\mathit{TotalDefPr}(\mathit{Drug}, 0.75)) \\
= \quad & \tfrac{1}{2} \times \tfrac{7}{10} \times \tfrac{3}{4} = \tfrac{21}{80} \\
W_{\text{plog2lpmln}(\Pi)}(I_2) = \quad & w(\mathit{TotalDefPr}(\mathit{Male}, 0.5)) \times w(\mathit{TotalDefPr}(\mathit{Recover}, 0.8)) \times \\
& w(\mathit{AssPr}_{r_3,C_6}(\mathit{Drug} = \mathbf{t})) \\
= \quad & \tfrac{1}{2} \times \tfrac{4}{5} \times \tfrac{1}{4} = \tfrac{1}{10} \\
W_{\text{plog2lpmln}(\Pi)}(I_3) = \quad & w(\mathit{TotalDefPr}(\mathit{Male}, 0.5)) \times w(\mathit{AssPr}_{r_2,C_4}(\mathit{Recover} = \mathbf{t})) \times \\
& w(\mathit{TotalDefPr}(\mathit{Drug}, 0.75)) \\
= \quad & \tfrac{1}{2} \times \tfrac{3}{10} \times \tfrac{3}{4} = \tfrac{9}{80} \\
W_{\text{plog2lpmln}(\Pi)}(I_4) = \quad & w(\mathit{TotalDefPr}(\mathit{Male}, 0.5)) \times w(\mathit{AssPr}_{r_2,C_3}(\mathit{Recover} = \mathbf{t})) \times \\
& w(\mathit{AssPr}_{r_3,C_6}(\mathit{Drug} = \mathbf{t})) \\
= \quad & \tfrac{1}{2} \times \tfrac{1}{5} \times \tfrac{1}{4} = \tfrac{1}{40} \\
W_{\text{plog2lpmln}(\Pi)}(I_5) = \quad & w(\mathit{AssPr}_{r_1}(\mathit{Male} = \mathbf{t})) \times w(\mathit{TotalDefPr}(\mathit{Recover}, 0.3)) \times \\
& w(\mathit{TotalDefPr}(\mathit{Drug}, 0.25)) \\
= \quad & \tfrac{1}{2} \times \tfrac{3}{10} \times \tfrac{1}{4} = \tfrac{3}{80}
\end{aligned}
$$

$$\begin{aligned}
W_{\text{plog2lpmln}(\Pi)}(I_6) = \;& w(AssPr_{r_1}(Male = \mathbf{t})) \times w(TotalDefPr(Recover, 0.4)) \times \\
& w(AssPr_{r_3,C_5}(Drug = \mathbf{t})) \\
= \;& \tfrac{1}{2} \times \tfrac{2}{5} \times \tfrac{3}{4} = \tfrac{3}{20} \\
W_{\text{plog2lpmln}(\Pi)}(I_7) = \;& w(AssPr_{r_1}(Male = \mathbf{t})) \times w(AssPr_{r_2,C_2}(Recover = \mathbf{t})) \times \\
& w(TotalDefPr(Drug, 0.25)) \\
= \;& \tfrac{1}{2} \times \tfrac{7}{10} \times \tfrac{1}{4} = \tfrac{7}{80} \\
W_{\text{plog2lpmln}(\Pi)}(I_8) = \;& w(AssPr_{r_1}(Male = \mathbf{t})) \times w(AssPr_{r_2,C_1}(Recover = \mathbf{t})) \times \\
& w(AssPr_{r_3,C_5}(Drug = \mathbf{t})) \\
= \;& \tfrac{1}{2} \times \tfrac{3}{5} \times \tfrac{3}{4} = \tfrac{9}{40}
\end{aligned}$$

As we see, the unnormalized weight $W_{\text{plog2lpmln}(\Pi)}(I_i)$ *of the translated* LP$^{\text{MLN}}$ *program coincide with the unnormalized probability* $\hat{\mu}_\Pi(W_i)$ *of the original P-log program* ($i \in \{1, \ldots, 8\}$)*. The probabilities of recover when a person takes drug or not are as follows: (*$W_{\text{plog2lpmln}(\Pi)}(I_i)$ *is denoted by* $\omega(I_i)$*)*

$$\begin{aligned}
P(recover \mid \sim drug) &= \tfrac{\omega(I_3)+\omega(I_7)}{\omega(I_1)+\omega(I_3)+\omega(I_5)+\omega(I_7)} = 0.4 \\
P(recover \mid drug) &= \tfrac{\omega(I_4)+\omega(I_8)}{\omega(I_2)+\omega(I_4)+\omega(I_6)+\omega(I_8)} = 0.5
\end{aligned}$$ [1]

The result above coincide with our understanding of this Simpson's Paradox: the drug is beneficial to patients of unknown gender — though this drug is harmful to the patients of known gender, whether they are male or female.

[1] Note that these two probabilities calculated in (Baral *et al.*, 2009) are wrong where some mistakes were made when calculating $P(W, drug = \mathbf{t})$ and $P(W, drug = \mathbf{f})$.

5.4 Moving Robot

Example 9 *Let's consider Moving Robot Problem from (Baral* et al.*, 2009). There are three rooms, R_1, R_2, and R_3, each of which is either open or closed. A robot cannot open a door and we navigate it to go into an open room, say R_1. However, a possible malfunction can cause the robot to abnormally go into any one of the open rooms, while there is 50% chance for the robot to go into the navigated room. This problem is represented by the following P-log program* Π. *($r \in \{R_1, R_2, R_3\}$):*

$$
\begin{array}{l}
Open(r) \leftarrow not \sim\!Open(r). \\
In = r \leftarrow GoIn = r, not\ Ab. \\
Ab \leftarrow Break. \\
GoIn = R_1. \\
Break. \\
[1]\, random(In : \{X : Open(X)\}) \leftarrow GoIn = r, Break. \\
pr_1(In = r | GoIn = r, Break) = 1/2.
\end{array}
$$

The translated LP^{MLN} *encoding* plog2lpmln(Π) *is as follows, where we denote the condition "GoIn(r), Break" by C_r.*

$//**** \; \tau(\Pi) \; ****$

$\alpha: \; Open(r) = \mathbf{t} \leftarrow not \; Open(r) = \mathbf{f}$

$\alpha: \; In = r \leftarrow GoIn = r, not \; Ab = \mathbf{t}$

$\alpha: \; Ab = \mathbf{t} \leftarrow Break = \mathbf{t}$

$\alpha: \; GoIn = R_1$

$\alpha: \; Break = \mathbf{t}$

$\alpha: \; \leftarrow Open(r) = \mathbf{t}, Open(r) = \mathbf{f}$

$\alpha: \; \leftarrow In = r_1, In = r_2, r_1 \neq r_2$

$\alpha: \; \leftarrow GoIn = r_1, GoIn = r_2, r_1 \neq r_2$

$\alpha: \; \leftarrow Ab = \mathbf{t}, Ab = \mathbf{f}$

$\alpha: \; \leftarrow Break = \mathbf{t}, Break = \mathbf{f}$

$\alpha: \; In = R_1; In = R_2; In = R_3 \leftarrow GoIn = r, Break = \mathbf{t}, not \; Intervene(In)$

$\alpha: \; \leftarrow In = r', not \; Open(r') = \mathbf{t}, GoIn = r, Break = \mathbf{t}, not \; Intervene(In)$

$//**** \; Possible \; Atoms ****$

$\alpha: \; Poss_1(In = r') \leftarrow GoIn = r, Break = \mathbf{t}, Open(r') = \mathbf{t}, not \; Intervene(In)$

$//**** \; Assigned \; Probability ****$

$\alpha: \; PossWithAssPr_{1,C_1}(In = r) \leftarrow Poss_1(In = r), GoIn = r, Break = \mathbf{t}$

$\alpha: \; AssPr_{1,C_1}(In = r) \leftarrow In = r, PrAtomApplied_{1,C_1}(In = r)$

$ln(\frac{1}{2}): \quad \leftarrow not \; AssPr_{1,C_1}(In = r)$

$\alpha: \; PossWithAssPr(In = r) \leftarrow PossWithAssPr_{1,C_1}(In = r)$

$//**** \textit{Denominator for Default Probability} ****$

$\alpha: \ \ PossWithDefPr(In=r) \leftarrow Poss_1(In=r), not\ \ PossWithAssPr(In=r)$

$\alpha: \ \ NumDefPr(In,x) \leftarrow In=r, PossWithDefPr(In=r),$

$\qquad\qquad\qquad\qquad x=\#count\{y: PossWithDefPr(In=y)\}$

$ln(\frac{1}{m}): \ \ \bot \leftarrow not\ \ NumDefPr(In,m) \qquad\qquad (m\in\{2,3\})$

$//**** \textit{Numerator for Default Probability} ****$

$\alpha: \ \ RemPr(In,1-y) \leftarrow In=r, PossWithDefPr(In=r),$

$\qquad\qquad\qquad\qquad y=\#sum\{0.5: PossWithAssPr_{1,C_1}(In=r')\}$

$\alpha: \ \ TotalDefPr(In,x) \leftarrow RemPr(In,x), x>0$

$ln(x): \ \ \bot \leftarrow not\ \ TotalDefPr(In,x)$

$\alpha: \ \ \bot \leftarrow RemPr(In,x), x\leq 0$

Let Π' *denotes the further translated ASP program with weak constraints, which is obtained from* plog2lpmln(Π) *by applying the translation* $\text{lpmln2wc}^{\text{pnt,clingo}}_{\text{simp}}$ *and the simplification for hard rules.* Π' *is as follows:*

$\% **** \; Declaration \; Part ****$

$room(r_1; r_2; r_3).$

$\% **** \; \tau(\Pi) \; ****$

$open(R, \mathbf{t}) \leftarrow not \; open(R, \mathbf{f}), room(R).$

$in(R) \leftarrow goIn(R), not \; ab(\mathbf{t}).$

$ab(\mathbf{t}) \leftarrow break(\mathbf{t}).$

$goIn(r_1).$

$break(\mathbf{t}).$

$\leftarrow open(R, \mathbf{t}), open(R, \mathbf{f}).$

$\leftarrow in(R), in(R_1), R \neq R_1.$

$\leftarrow goIn(R), goIn(R_1), R \neq R_1.$

$\leftarrow ab(\mathbf{t}), ab(\mathbf{f}).$

$\leftarrow break(\mathbf{t}), break(\mathbf{f}).$

$in(r_1); in(r_2); in(r_3) \leftarrow goIn(R), break(\mathbf{t}), not \; intervene(in).$

$\leftarrow in(R), not \; open(R, \mathbf{t}), goIn(R_1), break(\mathbf{t}), not \; intervene(in).$

$\% **** \; possible \; Atoms ****$

$poss(1, in(R)) \leftarrow goIn(R_1), break(\mathbf{t}), open(R, \mathbf{t}), not \; intervene(in).$

$\% **** \; Assigned \; Probability ****$

$possWithAssPr(1, c(1), in(R)) \leftarrow poss(1, in(R)), goIn(R), break(\mathbf{t}).$

$assPr(1, c(1), in(R)) \leftarrow in(R), possWithAssPr(1, c(1), in(R)).$

$possWithAssPr(in(R)) \leftarrow possWithAssPr(1, c(1), in(R)).$

$\% **** \; Denominator\ for\ Default\ Probability \; ****$

$possWithDefPr(in(R)) \leftarrow poss(1, in(R)), not\ possWithAssPr(in(R)).$

$numDefPr(in, X) \leftarrow in(R), possWithDefPr(in(R)),$

$\qquad\qquad X = \#count\{Y : possWithDefPr(in(Y))\}.$

$\% **** \; Numerator\ for\ Default\ Probability \; ****$

$remPr(in, X) \leftarrow in(R), possWithDefPr(in(R)), X = 1 - Y,$

$\qquad\qquad Y = \#sum\{0.5 : possWithAssPr(1, c(1), in(R_1))\}.$

$totalDefPr(in, X) \leftarrow remPr(in, X), X > 0.$

$\bot \leftarrow remPr(in, X), X \leq 0.$

$\% **** \; Weak\ Constraints \; ****$

$:\sim\ assPr(1, c(1), in(R)).\ [-ln(0.5)]$

$:\sim\ numDefPr(in, X).\ \left[-ln(\frac{1}{X})\right]$

$:\sim\ totalDefPr(in, X).\ [-ln(X)]$

It is easy to check that there are 3 *possible worlds of the P-log program* Π*:*

$$W_1 = \{In = R_1, GoIn = R_1, Break, Ab, Open(R_1), Open(R_2), Open(R_3)\}$$
$$W_2 = \{In = R_2, GoIn = R_1, Break, Ab, Open(R_1), Open(R_2), Open(R_3)\}$$
$$W_3 = \{In = R_3, GoIn = R_1, Break, Ab, Open(R_1), Open(R_2), Open(R_3)\},$$

and their unnormalized probabilities are as follows:

$$\hat{\mu}_\Pi(W_1) = P(W_1, In = R_1) = 0.5$$
$$\hat{\mu}_\Pi(W_2) = P(W_2, In = R_2) = \tfrac{1-0.5}{2} = 0.25$$
$$\hat{\mu}_\Pi(W_3) = P(W_3, In = R_3) = \tfrac{1-0.5}{2} = 0.25$$

As for the translated LP^{MLN} *program* $\text{plog2lpmln}(\Pi)$*, it has 3 probabilistic stable models* I_1, I_2, I_3*, each of which is an extension of* W_1, W_2, W_3 *respectively, and satisfies* *the following atoms:*

$$\begin{aligned}
&Poss_1(In = R_1), Poss_1(In = R_2), Poss_1(In = R_3),\\
&PossWithAssPr_{1,C_1}(In = R_1),\\
&PossWithAssPr(In = R_1),\\
&PossWithDefPr(In = R_2), PossWithDefPr(In = R_3).
\end{aligned}$$

In addition,

$$\begin{aligned}
I_1 \vDash\ & \{AssPr_{1,C_1}(In = R_1)\},\\
I_2 \vDash\ & \{NumDefPr(In, 2), RemPr(In, 0.5), TotalDefPr(In, 0.5)\},\\
I_3 \vDash\ & \{NumDefPr(In, 2), RemPr(In, 0.5), TotalDefPr(In, 0.5)\}.
\end{aligned}$$

It is easy to check that $I_i = \phi(W_i)$ *for* $i \in \{1, 2, 3\}$. *The unnormalized weight* $W_{\rm plog2lpmln(\Pi)}(I_i)$ *of each probabilistic stable model* I_i *is shown below:*

$$\begin{aligned}
W_{\rm plog2lpmln(\Pi)}(I_1) =\ & w(AssPr_{1,C_1}(In = R_1)) = \tfrac{1}{2}\\
W_{\rm plog2lpmln(\Pi)}(I_2) =\ & w(NumDefPr(In, 2)) \times w(TotalDefPr(In, 0.5)) = \tfrac{1}{2} \times \tfrac{1}{2} = \tfrac{1}{4}\\
W_{\rm plog2lpmln(\Pi)}(I_3) =\ & w(NumDefPr(In, 2)) \times w(TotalDefPr(In, 0.5)) = \tfrac{1}{2} \times \tfrac{1}{2} = \tfrac{1}{4}
\end{aligned}$$

As we see, the unnormalized weight $W_{\rm plog2lpmln(\Pi)}(I_i)$ *of the translated* LP^{MLN} *program coincide with the unnormalized probability* $\hat{\mu}_\Pi(W_i)$ *of the original P-log program (*$i \in \{1, 2, 3\}$*).*

5.5 Bayesian squirrel

Example 10 *The following P-log program* Π *describes Bayesian Squirrel from (Baral* et al.*, 2009). The squirrel has hidden its acorns in one of two patches, say* P_1 *and* P_2*, but cannot remember which. The squirrel is 80% certain the food is hidden in* P_1*. Also, it knows there is a 20% chance of finding food per day when it looking in the right patch (and, of course, a 0% probability if it is looking in the wrong patch).(*$p \in \{P_1, P_2\}, d \in \{1,2,3,4,5\}$*):*

$$\sim Found(p,d) \leftarrow not\ Found(p,d).$$
$$[r_1]\ \ random(HiddenIn).$$
$$[r_2]\ \ random(Found(p,d)) \leftarrow HiddenIn = p, Look(d) = p.$$
$$pr_{r_1}(HiddenIn = P_1) = 0.8.$$
$$pr_{r_2}(Found(p,d)) = 0.2.$$

$$Do(Look(1) = P_1).$$

$Found(p,d)$ *represents that the squirrel found food at patch* p *at day* d*, and* $Look(d) = p$ *represents that the squirrel looked in patch* p *at day* d*. The translated* $\mathrm{LP}^{\mathrm{MLN}}$ *encoding* plog2lpmln(Π) *is as follows:*

$// **** \tau(\Pi) \ ****$

$\alpha : \ Found(p,d) = \mathbf{f} \leftarrow not \ Found(p,d) = \mathbf{t}$

$\alpha : \ \leftarrow Found(p,d) = \mathbf{t}, Found(p,d) = \mathbf{f}$

$\alpha : \ \leftarrow HiddenIn = P_1, HiddenIn = P_2$

$\alpha : \ \leftarrow Look(d) = P_1, Look(d) = P_2$

$\alpha : \ HiddenIn = P_1; HiddenIn = P_2 \leftarrow not \ Intervene(HiddenIn)$

$\alpha : \ Found(p,d) = \mathbf{t}; Found(p,d) = \mathbf{f} \leftarrow$

$HiddenIn = p, Look(d) = p, not \ Intervene(Found(p,d))$

$\alpha : \ Do(Look(1) = P_1)$

$\alpha : \ Look(1) = P_1 \leftarrow Do(Look(1) = P_1)$

$\alpha : \ Intervene(Look(1)) \leftarrow Do(Look(1) = P_1)$

$// **** \ Possible \ Atoms ****$

$\alpha : \ Poss_{r_1}(HiddenIn = p) \leftarrow not \ Intervene(HiddenIn)$

$\alpha : \ Poss_{r_2}(Found(p,d) = b) \leftarrow HiddenIn = p, Look(d) = p, not \ Intervene(Found(p,d))$

$// **** \ Assigned \ Probability ****$

$\alpha : \ PossWithAssPr_{r_1}(HiddenIn = P_1) \leftarrow Poss_{r_1}(HiddenIn = P_1)$

$\alpha : \ AssPr_{r_1}(HiddenIn = P_1) \leftarrow HiddenIn = P_1, PossWithAssPr_{r_1}(HiddenIn = P_1)$

$ln(0.8) : \ \bot \leftarrow not \ AssPr_{r_1}(HiddenIn = P_1)$

$\alpha : \ PossWithAssPr(HiddenIn = P_1) \leftarrow PossWithAssPr_{r_1}(HiddenIn = P_1)$

$\alpha : \ PossWithAssPr_{r_2}(Found(p,d) = \mathbf{t}) \leftarrow Poss_{r_2}(Found(p,d) = \mathbf{t})$

$\alpha : \ AssPr_{r_2}(Found(p,d) = \mathbf{t}) \leftarrow Found(p,d) = \mathbf{t}, PossWithAssPr_{r_2}(Found(p,d) = \mathbf{t})$

$ln(0.2) : \ \bot \leftarrow not \ AssPr_{r_2}(Found(p,d) = \mathbf{t})$

$\alpha : \ PossWithAssPr(Found(p,d) = \mathbf{t}) \leftarrow PossWithAssPr_{r_2}(Found(p,d) = \mathbf{t})$

$//**** \textit{Denominator for Default Probability} ****$

$$\alpha: \ PossWithDefPr(HiddenIn = p) \leftarrow Poss_{r_1}(HiddenIn = p),$$
$$not \ \ PossWithAssPr(HiddenIn = p)$$

$$\alpha: \ NumDefPr(HiddenIn, x) \leftarrow HiddenIn = p, PossWithDefPr(HiddenIn = p),$$
$$x = \#count\{y : PossWithDefPr(HiddenIn = y)\}$$

$$ln(\tfrac{1}{m}): \ \bot \leftarrow not \ \ NumDefPr(HiddenIn, m) \qquad (m \in \{2\})$$

$$\alpha: \ PossWithDefPr(Found(p, d) = b) \leftarrow Poss_{r_2}(Found(p, d) = b),$$
$$not \ \ PossWithAssPr(Found(p, d) = b)$$

$$\alpha: \ NumDefPr(Found(p, d), x) \leftarrow Found(p, d) = b, PossWithDefPr(Found(p, d) = b),$$
$$x = \#count\{y : PossWithDefPr(Found(p, d) = y)\}$$

$$ln(\tfrac{1}{m}): \ \bot \leftarrow not \ \ NumDefPr(Found(p, d), m) \qquad (m \in \{2\})$$

$//**** \textit{Numerator for Default Probability} ****$

$$\alpha: \ RemPr(HiddenIn, 1 - y) \leftarrow HiddenIn = p, PossWithDefPr(HiddenIn = p),$$
$$y = \#sum\{0.8 : PossWithAssPr_{r_1}(HiddenIn = P_1)\}$$

$$\alpha: \ TotalDefPr(HiddenIn, x) \leftarrow RemPr(HiddenIn, x), x > 0$$

$$ln(x): \ \bot \leftarrow not \ \ TotalDefPr(HiddenIn, x)$$

$$\alpha: \ \bot \leftarrow RemPr(HiddenIn, x), x \leq 0$$

$$\alpha: \ RemPr(Found(p, d), 1 - y) \leftarrow HiddenIn = p, Look(d) = p,$$
$$Found(p, d) = b, PossWithDefPr(Found(p, d) = b),$$
$$y = \#sum\{0.2 : PossWithAssPr_{r_2}(Found(p, d) = \mathbf{t})\}$$

$$\alpha: \ TotalDefPr(Found(p, d), x) \leftarrow RemPr(Found(p, d), x), x > 0$$

$$ln(x): \ \bot \leftarrow not \ \ TotalDefPr(Found(p, d), x)$$

$$\alpha: \ \bot \leftarrow RemPr(Found(p, d), x), x \leq 0$$

Let Π' *denotes the further translated ASP program with weak constraints, which is obtained from* $\text{plog2lpmln}(\Pi)$ *by applying the translation* $\text{lpmln2wc}^{\text{pnt,clingo}}_{\text{simp}}$ *and the simplification for hard rules.* Π' *is as follows:*

$\% **** \; Declaration\ Part ****$

$patch(p_1; p_2).$

$day(1..5).$

$boolean(\mathbf{t}; \mathbf{f}).$

$\% **** \; \tau(\Pi) \; ****$

$found(P, D, \mathbf{f}) \leftarrow not\ found(P, D, \mathbf{t}), patch(P), day(D).$

$\leftarrow found(P, D, \mathbf{t}), found(P, D, \mathbf{f}).$

$\leftarrow hiddenIn(p_1), hiddenIn(p_2).$

$\leftarrow look(D, p_1), look(D, p_2).$

$hiddenIn(p_1); hiddenIn(p_2) \leftarrow not\ intervene(hiddenIn).$

$found(P, D, \mathbf{t}); found(P, D, \mathbf{f}) \leftarrow hiddenIn(P), look(D, P), not\ intervene(found(P, D)).$

$do(look(1, p_1)).$

$look(1, p_1) \leftarrow do(look(1, p_1)).$

$intervene(look(1)) \leftarrow do(look(1, p_1)).$

$\% **** \; possible\ Atoms ****$

$poss_{r_1}(hiddenIn(P)) \leftarrow not\ intervene(hiddenIn), patch(P).$

$$poss_{r_2}(found(P, D, B)) \leftarrow hiddenIn(P), look(D, P), \\ not\ intervene(found(P, D)), boolean(B).$$

$\% **** \textit{ Assigned Probability } ****$

$possWithAssPr_{r_1}(hiddenIn(p_1)) \leftarrow poss_{r_1}(hiddenIn(p_1)).$

$assPr_{r_1}(hiddenIn(p_1)) \leftarrow hiddenIn(p_1), possWithAssPr_{r_1}(hiddenIn(p_1)).$

$possWithAssPr(hiddenIn(p_1)) \leftarrow possWithAssPr_{r_1}(hiddenIn(p_1)).$

$possWithAssPr_{r_2}(found(P, D, \mathbf{t})) \leftarrow poss_{r_2}(found(P, D, \mathbf{t})).$

$assPr_{r_2}(found(P, D, \mathbf{t})) \leftarrow found(P, D, \mathbf{t}), possWithAssPr_{r_2}(found(P, D, \mathbf{t})).$

$possWithAssPr(found(P, D, \mathbf{t})) \leftarrow possWithAssPr_{r_2}(found(P, D, \mathbf{t})).$

$\% **** \textit{ Denominator for Default Probability } ****$

$$\begin{aligned} possWithDefPr(hiddenIn(P)) \leftarrow\ & poss_{r_1}(hiddenIn(P)), \\ & not\ possWithAssPr(hiddenIn(P)). \end{aligned}$$

$$\begin{aligned} numDefPr(hiddenIn, X) \leftarrow\ & hiddenIn(P), possWithDefPr(hiddenIn(P)), \\ & X = \#count\{Y : possWithDefPr(hiddenIn(Y))\}. \end{aligned}$$

$$\begin{aligned} possWithDefPr(found(P, D, B)) \leftarrow\ & poss_{r_2}(found(P, D, B)), \\ & not\ possWithAssPr(found(P, D, B)). \end{aligned}$$

$$\begin{aligned} numDefPr(found(P, D), X) \leftarrow\ & found(P, D, B), possWithDefPr(found(P, D, B)), \\ & X = \#count\{Y : possWithDefPr(found(P, D, Y))\}. \end{aligned}$$

$\% **** \textit{ Numerator for Default Probability} ****$

$$remPr(hiddenIn, X) \leftarrow hiddenIn(P), possWithDefPr(hiddenIn(P)), X = 1 - Y,$$
$$Y = \#sum\{0.8 : possWithAssPr_{r_1}(hiddenIn(p_1))\}.$$

$$totalDefPr(hiddenIn, X) \leftarrow remPr(hiddenIn, X), X > 0.$$

$$\bot \leftarrow remPr(hiddenIn, X), X \leq 0.$$

$$remPr(found(P, D), X) \leftarrow hiddenIn(P), look(D, P), X = 1 - Y,$$
$$found(P, D, B), possWithDefPr(found(P, D, B)),$$
$$Y = \#sum\{0.2 : possWithAssPr_{r_2}(found(P, D, \mathbf{t}))\}.$$

$$totalDefPr(found(P, D), X) \leftarrow remPr(found(P, D), X), X > 0.$$

$$\bot \leftarrow remPr(found(P, D), X), X \leq 0.$$

$\% **** \textit{ Weak Constraints} ****$

$$:\sim assPr_{r_1}(hiddenIn(p_1)).\ [-ln(0.8)]$$

$$:\sim assPr_{r_2}(found(P, D, \mathbf{t})).\ [-ln(0.2)]$$

$$:\sim numDefPr(hiddenIn, X).\ \left[-ln(\tfrac{1}{X})\right]$$

$$:\sim numDefPr(found(P, D), X).\ \left[-ln(\tfrac{1}{X})\right]$$

$$:\sim totalDefPr(hiddenIn, X).\ [-ln(X)]$$

$$:\sim totalDefPr(found(P, D), X).\ [-ln(X)]$$

It is easy to check that there are 3 *possible worlds of the P-log program* Π*:*

$$W_1 = \{HiddenIn = P_1, Found(P_1, 1) = \mathbf{t}, Look(1) = P_1, \dots\}$$

$$W_2 = \{HiddenIn = P_1, Found(P_1, 1) = \mathbf{f}, Look(1) = P_1, \dots\}$$

$$W_3 = \{HiddenIn = P_2, Found(P_2, 1) = \mathbf{f}, Look(1) = P_1, \dots\},$$

and their unnormalized probabilities are as follows:

$$\hat{\mu}_\Pi(W_1) = P(W_1, HiddenIn = P_1) \times P(W_1, Found(P_1, 1) = \mathbf{t}) = \tfrac{4}{5} \times \tfrac{1}{5} = \tfrac{4}{25}$$
$$\hat{\mu}_\Pi(W_2) = P(W_2, HiddenIn = P_1) \times P(W_1, Found(P_1, 1) = \mathbf{f}) = \tfrac{4}{5} \times (1 - \tfrac{1}{5}) = \tfrac{16}{25}$$
$$\hat{\mu}_\Pi(W_3) = P(W_3, HiddenIn = P_2) = 1 - \tfrac{4}{5} = \tfrac{1}{5}$$

As for the translated LP$^{\text{MLN}}$ *program* plog2lpmln(Π)*, it has 3 probabilistic stable models I_1, I_2, I_3, each of which is an extension of W_1, W_2, W_3 respectively, and satisfies the following atoms:*

$$\begin{aligned}
&Poss_{r_1}(HiddenIn = P_1), Poss_{r_1}(HiddenIn = P_2),\\
&PossWithAssPr_{r_1}(HiddenIn = P_1),\\
&PossWithAssPr(HiddenIn = P_1), PossWithDefPr(HiddenIn = P_2)
\end{aligned}$$

In addition,

$$\begin{aligned}
I_1 \vDash\ & \{AssPr_{r_1}(HiddenIn = P_1),\\
& Poss_{r_2}(Found(P_1, 1) = \mathbf{t}), Poss_{r_2}(Found(P_1, 1) = \mathbf{f}),\\
& PossWithAssPr(Found(P_1, 1) = \mathbf{t}), PossWithDefPr(Found(P_1, 1) = \mathbf{f}),\\
& PossWithAssPr_{r_2}(Found(P_1, 1) = \mathbf{t}), AssPr_{r_2}(Found(P_1, 1) = \mathbf{t})\}\\
I_2 \vDash\ & \{AssPr_{r_1}(HiddenIn = P_1)\\
& Poss_{r_2}(Found(P_1, 1) = \mathbf{t}), Poss_{r_2}(Found(P_1, 1) = \mathbf{f}),\\
& PossWithAssPr(Found(P_1, 1) = \mathbf{t}), PossWithDefPr(Found(P_1, 1) = \mathbf{f}),\\
& PossWithAssPr_{r_2}(Found(P_1, 1) = \mathbf{t}),\\
& NumDefPr(Found(P_1, 1), 1),\\
& RemPr(Found(P_1, 1), 0.8), TotalDefPr(Found(P_1, 1), 0.8)\}\\
I_3 \vDash\ & \{NumDefPr(HiddenIn, 1), RemPr(HiddenIn, 0.2), TotalDefPr(HiddenIn, 0.2)\}.
\end{aligned}$$

It is easy to check that $I_i = \phi(W_i)$ for $i \in \{1, 2, 3\}$. The unnormalized weight $W_{\text{plog2lpmln}(\Pi)}(I_i)$ of each probabilistic stable model I_i is shown below:

$$\begin{aligned}
W_{\text{plog2lpmln}(\Pi)}(I_1) = &\quad w(AssPr_{r_1}(HiddenIn = P_1)) \times w(AssPr_{r_2}(Found(P_1, 1)) \\
= &\quad \tfrac{4}{5} \times \tfrac{1}{5} = \tfrac{4}{25} \\
W_{\text{plog2lpmln}(\Pi)}(I_2) = &\quad w(AssPr_{r_1}(HiddenIn = P_1)) \times w(\mathit{TotalDefPr}(Found(P_1, 1), 0.8)) \\
= &\quad \tfrac{4}{5} \times \tfrac{4}{5} = \tfrac{16}{25} \\
W_{\text{plog2lpmln}(\Pi)}(I_3) = &\quad w(\mathit{TotalDefPr}(HiddenIn, 0.2)) = \tfrac{1}{5}
\end{aligned}$$

As we see, for $i \in \{1, 2, 3\}$, the unnormalized weight $W_{\text{plog2lpmln}(\Pi)}(I_i)$ of the translated LP$^{\text{MLN}}$ *program coincide with the unnormalized probability $\hat{\mu}_\Pi(W_i)$ of the original P-log program. Consequently, the probability $P_{\text{plog2lpmln}(\Pi)}(I_i)$ of the translated* LP$^{\text{MLN}}$ *program also coincide with the probability $\mu_\Pi(W_i)$ of the original P-log program.*

5.6 When a P-log program is not consistent

Example 11 ***An Example for Theorem 3 if*** Π ***is not consistent***

The following P-log program describes Coin Problem, where we deliberately make the coin to be head but at the same time observe that the coin is tail. It's obvious that this P-log program is not consistent.($v \in \{H, T\}$*):*

$$[r]\ \ random(Flip).$$

$$Do(Flip = H).$$
$$Obs(Flip = T).$$

It is easy to check that Π *has no possible world. The translated* LP^{MLN} *encoding* plog2lpmln(Π) *is as follows:*

$//****\ \tau(\Pi)\ \ ****$

$\alpha:\ \ \leftarrow Flip = H, Flip = T$

$\alpha:\ \ Flip = H; Flip = T \leftarrow not\ \ Intervene(Flip)$

$\alpha:\ \ Do(Flip = H)$ ***(1)***

$\alpha:\ \ Flip = H \leftarrow Do(Flip = H)$

$\alpha:\ \ Intervene(Flip) \leftarrow Do(Flip = H)$

$\alpha:\ \ Obs(Flip = T)$ ***(2)***

$\alpha:\ \ \leftarrow Obs(Flip = T), not\ \ Flip = T$ ***(3)***

$//****\ Possible\ Atoms****$

$\alpha:\ \ Poss_r(Flip = v) \leftarrow not\ \ Intervene(Flip)$

$// **** \textit{Denominator for Default Probability} ****$

$\alpha: \ \mathit{PossWithDefPr}(\mathit{Flip}=v) \leftarrow \mathit{Poss}_r(\mathit{Flip}=v), \mathit{not} \ \mathit{PossWithAssPr}(\mathit{Flip}=v)$

$\alpha: \ \mathit{NumDefPr}(\mathit{Flip}, x) \leftarrow \mathit{Flip}=v, \mathit{PossWithDefPr}(\mathit{Flip}=v),$

$\qquad\qquad\qquad x = \#\mathit{count}\{y : \mathit{PossWithDefPr}(\mathit{Flip}=y)\}$

$ln(\frac{1}{2}): \quad \leftarrow \mathit{not} \ \mathit{NumDefPr}(\mathit{Flip}, 2)$

Although Π *has no possible world, its translated* LP$^{\text{MLN}}$ *program above has three stable models below, each of which doesn't satisfy one hard rule (marked in the program with the same number as the subscript of each interpretation):*

$$I_1 = \{\mathit{Obs}(\mathit{Flip}=T), \mathit{Flip}=T, \mathit{Poss}_r(\mathit{Flip}=H), \mathit{Poss}_r(\mathit{Flip}=T),$$
$$\mathit{PossWithDefPr}(\mathit{Flip}=H), \mathit{PossWithDefPr}(\mathit{Flip}=T), \mathit{NumDefPr}(\mathit{Flip}, 2)\}$$
$$I_2 = \{\mathit{Do}(\mathit{Flip}=H), \mathit{Intervene}(\mathit{Flip}), \mathit{Flip}=H\}$$
$$I_3 = \{\mathit{Do}(\mathit{Flip}=H), \mathit{Intervene}(\mathit{Flip}), \mathit{Flip}=H, \mathit{Obs}(\mathit{Flip}=T)\}$$

The reason of this is that LP$^{\text{MLN}}$ *tries to find a sub-optimal solution when the program is inconsistent, i.e.,* LP$^{\text{MLN}}$ *considers some hard rules incorrect when it is impossible to satisfy all hard rules. This intuition is different from that in P-log, where all hard rules represent definite knowledge. Thus Theorem 3 is only suitable for consistent P-log programs.*

Chapter 6

EXPERIMENTS

In this chapter, we present the results of benchmarking our translations.

The existing solvers of P-log (P-LOG 1, using Smodels based engine, and P-LOG 2, using Partial Grounding Algorithm based engine) and a solver of ASP programs with weak constraints (CLINGO) have different functionalities. With a pre-declared query (a set of atoms), P-LOG 1 and P-LOG 2 show the probability of one queried atom that has the highest marginal probability. However, CLINGO shows a stable model that maximum a posteriori probability. Besides, CLINGO is capable of showing all stable models along with their quantitative penalties, which can be turned into probabilities by normalization. In other words, CLINGO is good at finding an optimal stable model or the probability of any stable model, while P-LOG 1 and P-LOG 2 are good at finding an optimal choice of a single random event or the probability of a single atom.

We examine our translation by testing all consistent P-log examples in this thesis on P-LOG 1 and P-LOG 2, and also testing their translated ASP programs with weak constraints on CLINGO. [1] The outputs of P-LOG 1, P-LOG 2, and CLINGO coincide for each domain. For example, consider the domain "Variant of Monty Hall", the only stable model that contains "prize=4" (whose unnormalized probability is $\frac{1}{32}$) has the highest probability 0.3846. P-LOG 1 and P-LOG 2 output "prize = 4 with probability 0.3846", while CLINGO outputs an optimal stable model containing "prize(4)" with penalty 3465.

[1] These encodings can be found at `https://github.com/zhunyoung/MS-Thesis`.

Note that, we can further obtain the unnormalized probability of this stable model by $e^{-3.465} = \frac{1}{32}$. [2]

We record their run time in **Table 6.1**.

Domain	**P-log 1** (Smodels)	**P-log 2** (Partial Grounding)	**Clingo 4.5**
Variant of Monty Hall	0.000s + 0.000385s [a]	0.00191s + 0.000386s [b]	0.016s
Dice Problem	0.000s + 0.003114s	0.001333s + 0.000298s	0.015s
Simpson's Paradox	0.000s + 0.000666s	0.001381s + 0.000194s	0.019s
Moving Robot	0.000s + 0.000146s	0.001795s + 0.000085s	0.013s
Bayesian Squirrel	0.000s + 0.000478s	core dumped	0.017s

Table 6.1: Runtime of Examples on P-LOG 1, P-LOG 2, and CLINGO

[a]Smodels answer set finding time + probability computing time

[b]partial grounding time + probability computing time

Since the domains above are too small, we show the difference of computing the original P-log program on the existing P-log solver and computing the translated answer set program with weak constraints on CLINGO by benchmarking domains "Variant of Monty Hall" and "Dice Problem".

To test the efficiency of P-log 1, P-log 2, and Clingo under different domain size, we benchmarked the Varient of Monty Hall Problem where the number of doors is increased from 4 to 1000 with a fixed number of pr-atoms as 2. We record their run time in **Table 6.2**.

As we can see, P-log 1 and P-log 2 have much better performance over Clingo when we only increase the domain size. And P-log 2 has a even better performance over P-log 1 because the former one uses sampling method while the latter one uses exact inference. However, when the number of doors is increased to 1000, P-log 1 showed a random output after about 84 seconds. Thus for exact inference under large domain size, using clingo conjunction with our translations is a better means of

[2] CLINGO restricts all numbers to be integers, thus in practice, we scale up all numbers by 1000.

#doors + #pr-atoms	**P-log 1** (Smodels)	**P-log 2** (Partial Grounding)	**Clingo 4.5**
4 + 2	0.000s + 0.000224s [a]	0.001228s + 0.000221s [b]	0.014s
10 + 2	0.000s + 0.001265s	0.003765s + 0.00142s	0.020s
100 + 2	0.086s + 0.076525s	0.032208s + 0.09779s	1.400s
200 + 2	0.633s + 0.316585s	0.094238s + 0.585789s	10.258s
300 + 2	2.242s + 0.724783s	0.193164s + 1.77677s	34.456s
400 + 2	5.063s + 1.31343s	0.335731s + 3.99656s	80.680s
500 + 2	10.021s + 2.14988s	0.505037 + 7.59064	153.535s
1000 + 2	? + 0.709586s(84s)	2.05518s + 55.6697s	1216.697s

Table 6.2: Benchmarks of Varient Monty Hall Problem

[a] Smodels answer set finding time + probability computing time

[b] partial grounding time + probability computing time

computing P-log.

Chapter 7

CONCLUSION

Combining logic and probability has received attention in artificial intellegence. Some formalisms have been proposed by extending one of the most logic-expressive paradigm, Answer Set Programming, to overcome the deterministic nature of ASP and incorporate probability into its syntax or semantics. In this thesis, we focus on three such extensions: LP$^{\text{MLN}}$, weak constraints, and P-log. Weak constraints are a relatively simple extension to ASP programs, while P-log is highly structured but a more complex extension. LP$^{\text{MLN}}$ is shown to be a good middle ground language that clarifies the relationships.

In this thesis, we proposed a translation of LP$^{\text{MLN}}$ into programs with weak constraints and a translation of P-log into LP$^{\text{MLN}}$, which complement the existing translations in the opposite directions. It shows how different weight schemes of LP$^{\text{MLN}}$ and weak constraints are related, and how the probabilistic reasoning in P-log can be expressed in LP$^{\text{MLN}}$.

The results imply that MAP inference in P-log and other probabilistic logic languages, such as Markov Logic, ProbLog, and Pearl's Causal Models, can be computed by standard ASP solvers because they are known to be embeddable in LP$^{\text{MLN}}$, thereby allowing us to apply combinatorial optimization in standard ASP solvers to MAP inference in these languages to increase the efficiency.

Moreover, P-log has been viewed distant from Statistical Relational Learning. In this thesis, we show how P-log is reduced to LP$^{\text{MLN}}$. Since LP$^{\text{MLN}}$ is shown to be translatable into MLN, it yields a way to compute P-log using MLN solvers and allows us to learn the probabilities in P-log by turning it into MLN and applying SRL methods.

We expect the relationships will help us to apply the mathematical and computational results developed for one language to another language.

This work has been published within

- Lee and Yang (2017), "LP$^{\text{MLN}}$, Weak Constraints, and P-log", in "Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI 2017)".

REFERENCES

Bach, S. H., M. Broecheler, B. Huang and L. Getoor, “Hinge-loss markov random fields and probabilistic soft logic”, **arXiv:1505.04406 [cs.LG]** (2015).

Balai, E. and M. Gelfond, “On the relationship between P-log and LP$^{\mathrm{MLN}}$”, in “Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)”, (2016).

Baral, C., M. Gelfond and J. N. Rushton, “Probabilistic reasoning with answer sets”, TPLP **9**, 1, 57–144 (2009).

Bartholomew, M. and J. Lee, “On the stable model semantics for intensional functions”, Theory and Practice of Logic Programming **13**, 4-5, 863–876 (2013).

Buccafurri, F., N. Leone and P. Rullo, “Enhancing disjunctive datalog by constraints”, Knowledge and Data Engineering, IEEE Transactions on **12**, 5, 845–860 (2000).

Calimeri, F., W. Faber, M. Gebser, G. Ianni, R. Kaminski, T. Krennwallner, N. Leone, F. Ricca and T. Schaub, “Asp-core-2 input language format”, (2013).

De Raedt, L., A. Kimmig and H. Toivonen, “ProbLog: A probabilistic prolog and its application in link discovery.”, in “IJCAI”, vol. 7, pp. 2462–2467 (2007).

Ferraris, P., “Logic programs with propositional connectives and aggregates”, ACM Transactions on Computational Logic (TOCL) **12**, 4, 25 (2011).

Ferraris, P., J. Lee and V. Lifschitz, “Stable models and circumscription”, Artificial Intelligence **175**, 236–263 (2011).

Ferraris, P., J. Lee, V. Lifschitz and R. Palla, “Symmetric splitting in the general theory of stable models”, in “Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)”, pp. 797–803 (2009).

Gebser, M., R. Kaminski, B. Kaufmann, M. Lindauer, M. Ostrowski, J. Romero, T. Schaub and S. Thiele, “Potassco user guide”, (2015).

Gelfond, M. and V. Lifschitz, “The stable model semantics for logic programming”, in “Proceedings of International Logic Programming Conference and Symposium”, edited by R. Kowalski and K. Bowen, pp. 1070–1080 (MIT Press, 1988).

Harrison, A. J., V. Lifschitz and F. Yang, “The semantics of gringo and infinitary propositional formulas”, in “Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014”, (2014).

Lee, J. and Y. Meng, “Stable models of formulas with generalized quantifiers (preliminary report)”, in “Technical Communications of the 28th International Conference on Logic Programming”, pp. 61–71 (2012).

Lee, J., Y. Meng and Y. Wang, "Markov logic style weighted rules under the stable model semantics", In Technical Communications of the 31st International Conference on Logic Programming (2015).

Lee, J. and Y. Wang, "Weighted rules under the stable model semantics", in "Proceedings of International Conference on Principles of Knowledge Representation and Reasoning (KR)", (2016).

Nickles, M. and A. Mileo, "Probabilistic inductive logic programming based on answer set programming", in "15th International Workshop on Non-Monotonic Reasoning (NMR 2014)", (2014).

Pearl, J., *Causality: models, reasoning and inference*, vol. 29 (Cambridge Univ Press, 2000).

Richardson, M. and P. Domingos, "Markov logic networks", Machine Learning **62**, 1-2, 107–136 (2006).