LP^{MLN}, Weak Constraints, and P-log

Joohyung Lee and Zhun Yang
School of Computing, Informatics and Decision Systems Engineering
Arizona State University, Tempe, USA
{joolee, zyang90}@asu.edu

Abstract

LP^{MLN} is a recently introduced formalism that extends answer set programs by adopting the log-linear weight scheme of Markov Logic. This paper investigates the relationships between LP^{MLN} and two other extensions of answer set programs: weak constraints to express a quantitative preference among answer sets, and P-log to incorporate probabilistic uncertainty. We present a translation of LP^{MLN} into programs with weak constraints and a translation of P-log into LP^{MLN}, which complement the existing translations in the opposite directions. The first translation allows us to compute the most probable stable models (i.e., MAP estimates) of LP^{MLN} programs using standard ASP solvers. 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^{MLN}. The second translation tells us 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^{MLN}, which yields a way to compute P-log using standard ASP solvers and MLN solvers.

Introduction

LP^{MLN} (Lee and Wang 2016) is a recently introduced probabilistic logic programming language that extends answer set programs (Gelfond and Lifschitz 1988) with the concept of weighted rules, whose weight scheme is adopted from that of Markov Logic (Richardson and Domingos 2006). It is shown in (Lee and Wang 2016; Lee, Meng, and Wang 2015) that LP^{MLN} is expressive enough to embed Markov Logic and several other probabilistic logic languages, such as ProbLog (De Raedt, Kimmig, and Toivonen 2007), Pearls’ Causal Models (Pearl 2000), and a fragment of P-log (Baral, Gelfond, and Rushton 2009).

Among several extensions of answer set programs to overcome the deterministic nature of the stable model semantics, LP^{MLN} is one of the few languages that incorporate the concept of weights into the semantics. Another one is weak constraints (Buccafurri, Leone, and Rullo 2000), which are to assign a quantitative preference over the stable models of non-weak constraint rules: weak constraints cannot be used for deriving stable models. It is relatively a simple extension of the stable model semantics but has turned out to be useful in many practical applications. Weak constraints are part of the ASP Core 2 language (Calimeri et al. 2013), and are implemented in standard ASP solvers, such as CLINGO and DLV.

P-log is a probabilistic extension of answer set programs. In contrast to weak constraints, it 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.

This paper reveals interesting relationships between LP^{MLN} and these two extensions of answer set programs. It shows how different weight schemes of LP^{MLN} and weak constraints are related, and how the probabilistic reasoning in P-log can be expressed in LP^{MLN}. The result helps us understand these languages better as well as other related languages, and also provides new, effective computational methods based on the translations.

It is shown in (Lee and Wang 2016) that programs with weak constraints can be easily viewed as a special case of LP^{MLN} programs. In the first part of this paper, we show that an inverse translation is also possible under certain conditions, i.e., an LP^{MLN} program can be turned into a usual ASP program with weak constraints so that the most probable stable models of the LP^{MLN} program are exactly the optimal stable models of the program with weak constraints. The result allows for using ASP solvers for computing Maximum A Posteriori probability (MAP) estimates of LP^{MLN} programs. Interestingly, the translation is quite simple so it can be easily applied in practice. Further, the result implies that MAP inference in 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^{MLN}, thereby allowing us to apply combinatorial optimization in standard ASP solvers to MAP inference in these languages.

In the second part of the paper, we show how P-log can be completely characterized in LP^{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, our translation applies to full P-log and complements the recent translation from LP^MLN into P-log in (Balai and Gelfond 2016). In conjunction with the embedding of LP^MLN in answer set programs with weak constraints, our work shows how MAP estimates of P-log can be computed by standard ASP solvers, which provides a highly efficient way to compute P-log.

Preliminaries

Review: LP^MLN

We review the definition of LP^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, Lee, and Lifschitz 2011) by identifying logic program rules as a special case of first-order formulas under the stable model semantics. For example, rule \( r(x) \leftarrow p(x), \lnot q(x) \) is identified with formula \( \forall x (p(x) \land \lnot q(x) \rightarrow r(x)) \). An LP^MLN program is a finite set of weighted first-order formulas \( w : F \) where \( w \) is a real number (in which case the weighted formula is called soft) or \( \alpha \) for denoting the infinite weight (in which case it is called hard). An LP^MLN program is called ground if its formulas contain no variables. We assume a finite Herbrand Universe. Any LP^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.

For any ground LP^MLN program \( \Pi \) and any interpretation \( I \), \( \Pi[I] \) denotes the set of formulas obtained from \( \Pi \), and \( \Pi[I] \) denotes the set of \( w : F \) in \( I \) such that \( I \models F \), and \( \Pi[I] \) denotes the set \( \{ I \mid I \text{ is a stable model of } \Pi[I] \} \) (We refer the reader to the stable model semantics of first-order formulas in (Ferraris, Lee, and Lifschitz 2011)). The unnormalized weight of an interpretation \( I \) under \( \Pi \) is defined as

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

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

\[
P_{\Pi}(I) = \lim_{\alpha \to \infty} \frac{W_{\Pi}(I)}{\sum_{J \in \text{SM}[\Pi]} W_{\Pi}(J)}.
\]

\( I \) is called a (probabilistic) stable model of \( \Pi \) if \( P_{\Pi}(I) \neq 0 \).

Review: Weak Constraints

A weak constraint has the form

\[
\lnot F \ [\text{Weight @ Level}],
\]

where \( F \) is a ground formula, Weight is a real number and Level is a nonnegative integer. Note that syntax is more general than the one from the literature (Buccafurri, Leone, and Rullo 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.

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

\[
\sum_{\forall F \in \Pi_2, \ l \models F} w.
\]

For any two stable models \( I \) and \( I' \) of \( \Pi \), we say \( I \) is dominated by \( I' \) if

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

A stable model of \( \Pi \) is called optimal if it is not dominated by another stable model of \( \Pi \).

Turning LP^MLN into Programs with Weak Constraints

General Translation

We define a translation that turns an LP^MLN program into a program with weak constraints. For any ground LP^MLN program \( \Pi \), the translation \( \text{lpmln2wc}(\Pi) \) is simply defined as follows. We turn each weighted formula \( w : F \) in \( \Pi \) into \( \{ F \}^\text{ch} \), where \( \{ F \}^\text{ch} \) is a choice formula, standing for \( F \lor \lnot F \). (Ferraris, Lee, and Lifschitz 2011). Further, we add

\[
\lnot F \ [\text{Weight @ Level}],
\]

for any \( w \in \{ 0, 1 \} \), and

\[
\lnot F \ [\text{Weight @ Level}],
\]

otherwise.

Intuitively, choice formula \( \{ F \}^\text{ch} \) allows \( F \) to be either included or not in deriving a stable model.\(^2\) 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” \( e^\alpha \) or \( e^w \) that it receives under the LP^MLN semantics.

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

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

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

---

\(^1\)A literal is either an atom \( p \) or its negation \( \lnot p \).

\(^2\)This view of choice formulas was used in PrASP (Nickles and Mileo 2014) in defining spanning programs.
Theorem 1 For any LP_{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 For program Π:

\[
\begin{align*}
10: & \quad p \rightarrow q \\
1: & \quad p \rightarrow r
\end{align*}
\]

SM[Π] has 5 elements: \{∅, \{p\}, \{p, q\}, \{p, r\}, \{p, q, r\}\}. Among them, \{p, q\} is the most probable stable model, whose weight is \(e^{-10}\), while \{p, q, r\} is a probabilistic stable model whose weight is \(e^{-4}\). The translation yields

\[
\begin{align*}
\{p \rightarrow q\}^\text{ch} & : \quad p \rightarrow q \quad [−10 \circ 0] \\
\{p \rightarrow r\}^\text{ch} & : \quad p \rightarrow r \quad [−1 \circ 0] \\
\{p\}^\text{ch} & : \quad p \quad [−5 \circ 0] \\
\{r \rightarrow \bot\}^\text{ch} & : \quad r \rightarrow \bot \quad [20 \circ 0]
\end{align*}
\]

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 Consider the LP_{MLN} program Π in Example 1 from (Lee and Wang 2016):

\[
\begin{align*}
\alpha : & \quad \text{Bird}(Jo) \leftarrow \text{ResidentBird}(Jo) \quad (1) \\
\alpha : & \quad \text{Bird}(Jo) \leftarrow \text{MigratoryBird}(Jo) \quad (2) \\
\alpha : & \quad \bot \leftarrow \text{ResidentBird}(Jo), \text{MigratoryBird}(Jo) \quad (3) \\
2 : & \quad \text{ResidentBird}(Jo) \\
1 & \quad \text{MigratoryBird}(Jo)
\end{align*}
\]

The translation lpmln2wc(Π) is \(^3\)

\[
\begin{align*}
\{\text{Bird}(Jo) \leftarrow \text{ResidentBird}(Jo)\}^\text{ch} & : \quad \text{Bird}(Jo) \leftarrow \text{ResidentBird}(Jo) \quad [−10 \circ 0] \\
\{\text{Bird}(Jo) \leftarrow \text{MigratoryBird}(Jo)\}^\text{ch} & : \quad \text{Bird}(Jo) \leftarrow \text{MigratoryBird}(Jo) \quad [−10 \circ 0] \\
\{\bot \leftarrow \text{ResidentBird}(Jo), \text{MigratoryBird}(Jo)\}^\text{ch} & : \quad \bot \leftarrow \text{ResidentBird}(Jo), \text{MigratoryBird}(Jo) \quad [−10 \circ 0] \\
\{\text{ResidentBird}(Jo)\}^\text{ch} & : \quad \text{ResidentBird}(Jo) \quad [−20 \circ 0] \\
\{\text{MigratoryBird}(Jo)\}^\text{ch} & : \quad \text{MigratoryBird}(Jo) \quad [−10 \circ 0]
\end{align*}
\]

The three probabilistic stable models of Π, \{∅, \{Bird(Jo), ResidentBird(Jo)\}, and \{Bird(Jo), MigratoryBird(Jo)\}\}, have the same penalty −3 at level 1. Among them, \{Bird(Jo), ResidentBird(Jo)\} has the least penalty at level 0, and thus an optimal stable model of lpmln2wc(Π).

In some applications, one may not want any hard rules to be violated assuming that hard rules encode definite knowledge. For that, lpmln2wc(Π) can be modified by simply turning hard rules into the usual ASP rules. Then the stable models of lpmln2wc(Π) satisfy all hard rules. For example, the program in Example 2 can be translated into programs with weak constraints as follows.

\[
\begin{align*}
\text{Bird}(Jo) & \leftarrow \text{ResidentBird}(Jo) \\
\text{Bird}(Jo) & \leftarrow \text{MigratoryBird}(Jo) \\
\bot & \leftarrow \text{ResidentBird}(Jo), \text{MigratoryBird}(Jo) \\
\text{ResidentBird}(Jo) & \leftarrow \text{MigratoryBird}(Jo) \\
\text{MigratoryBird}(Jo) & \leftarrow \text{MigratoryBird}(Jo)
\end{align*}
\]

Also 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.

In view of Theorem 2 from (Lee and Wang 2016), which tells us how to embed Markov Logic into LP_{MLN}, it follows from Theorem 1 that MAP inference in MLN can also 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\}^\text{ch} for all atoms A.

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_{MLN} (Theorem 4 from (Lee and Wang 2016)) and the reduction of Causal Models to LP_{MLN} (Theorem 4 from (Lee, Meng, and Wang 2015)) thereby allowing us to apply combinatorial optimization methods in standard ASP solvers to these languages.

Alternative Translations

Instead of aggregating the weights of satisfied formulas, we may aggregate the weights of formulas that are not satisfied. Let lpmln2wc_{psl}(Π) be a modification of lpmln2wc(Π) by replacing (1) with

\[
\sim \bot \leftarrow \text{ResidentBird}(Jo) \quad [−1 \circ 1]
\]

and (2) with

\[
\sim F \quad [w @ 0].
\]

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 1 Theorem 1 remains true when lpmln2wc(Π) is replaced by lpmln2wc_{psl}(Π).

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

\(^3\)Recall that we identify the rules with the corresponding first-order formulas.
view will lead to a slight advantage when we further turn the translation into the input language of ASP solvers (See Footnote 6).

The current ASP solvers do not allow arbitrary formulas to appear in weak constraints. For computation using the ASP solvers, let $lpmln2we^{pnt,rule}(\Pi)$ be the translation by turning each weighted formula $w_i : F_i$ in $\Pi$ into

$$\neg F_i \rightarrow unsat(i)$$
$$\neg unsat(i) \rightarrow F_i$$
$$\therefore unsat(i) \quad [w_i \oplus l]$$

where $unsat(i)$ is a new atom, and $l = 1$ if $w_i$ is $\alpha$, and $l = 0$ otherwise.

**Corollary 2** Let $\Pi$ be an LP$^{MLN}$ program. There is a 1-1 correspondence $\phi$ between the most probable stable models of $\Pi$ and the optimal stable models of $lpmln2we^{pnt,rule}(\Pi)$, where $\phi(I) = I \cup \{unsat(i) \mid w_i : F_i \in \Pi, I \not\models F_i\}$.

The corollary allows us to compute the most probable stable models (MAP estimates) of the LP$^{MLN}$ program using the combination of f2lp ⁴ and CLINGO ⁵ (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.

If the unweighted LP$^{MLN}$ program is already in the rule form $Head \leftarrow Body$ that is allowed in the input languages of CLINGO and DLV, we may avoid the use of f2lp by slightly modifying the translation $lpmln2we^{pnt,rule}$ by turning each weighted rule

$w_i : Head_i \leftarrow Body_i$

instead into

$$unsat(i) \leftarrow Body_i, \text{not} Head_i$$
$$Head_i \leftarrow Body_i, \text{not} unsat(i)$$
$$\therefore unsat(i) \quad [w_i \oplus l]$$

where $l = 1$ if $w_i$ is $\alpha$, and $l = 0$ otherwise.

In the case when $Head_i$ is $\bot$, the translation can be further simplified: we simply turn $w_i : \bot \leftarrow Body_i$ into

$$\therefore Body_i \quad [w_i \oplus l].$$

**Example 1 continued:** For program (3), the simplified translation $lpmln2we^{pnt,rule}$ yields

$$\begin{array}{c|c|c}
unsat(1) & p, \text{not} q & \therefore unsat(1) \quad [1000] \\
unsat(2) & p, \text{not} r & \therefore unsat(2) \quad [100] \\
unsat(3) & \text{not} p & \therefore unsat(3) \quad [500] \\
\end{array}$$

Turning P-log into LP$^{MLN}$

**Review: P-log**

**Syntax** 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 $s_0, \ldots, s_n$ 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 $c : s_1 \times \cdots \times s_n \rightarrow s_0$, and $\vec{u} \in s_1 \times \cdots \times s_n$, and $v \in s_0$. Symbol $c(\vec{u})$ is called an attribute and $v$ is called its value. If the range $s_0$ of $c$ is $\{f, t\}$ then $c$ is called Boolean, and $c(\vec{u}) = t$ can be abbreviated as $c(\vec{u})$ and $c(\vec{u}) = f$ as $\neg c(\vec{u})$.

The signature of a P-log program is the union of two propositional signatures $\sigma_1$ and $\sigma_2$, where $\sigma_1$ is a sorted propositional signature, and $\sigma_2$ 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 $\sigma_1$.

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

$$\Pi = (R, S, P, Obs, Act)$$

where the signature of each of $R$, $S$, and $P$ is $\sigma_1$ and the signature of each of $Obs$ and $Act$ is $\sigma_2$ such that

- $R$ is a set of normal rules of the form
  $$A \leftarrow B_1, \ldots, B_m, \text{not} B_{m+1}, \ldots, \text{not} B_n$$
  where $A, B_1, \ldots, B_n$ are atoms ($0 \leq m \leq n$).
- $S$ is a set of random selection rules of the form
  $$[r] \quad \text{random}(c(\vec{u}) : \{x : p(x)\}) \leftarrow Body$$
  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 (5) is called a random selection rule for $c(\vec{u})$. Intuitively, rule (5) says that if $Body$ is true, the value of $c(\vec{u})$ is selected at random from the set $\text{Range}(c) \cap \{x : p(x)\}$ unless this value is fixed by a deliberate action, i.e., $Do(c(\vec{u}) = v)$ for some value $v$.
- $P$ is a set of so-called probability atoms (pr-atoms) of the form
  $$pr_r(c(\vec{u}) = v \mid C) = p$$
  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 (6) is associated with the random selection rule whose identifier is $r$.
- $Obs$ is a set of atomic facts for representing “observation”: $Obs(c(\vec{u}) = v)$ and $Obs(c(\vec{u}) \neq v)$.
- $Act$ is a set of atomic facts for representing a deliberate action: $Do(c(\vec{u}) = v)$.

**Semantics** Let $\Pi$ be a P-log program (4) of signature $\sigma_1 \cup \sigma_2$. The possible worlds of $\Pi$, denoted by $w(\Pi)$, are the stable models of $\Pi$, a (standard) ASP program with the propositional signature

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

that accounts for the logical part of P-log. Due to lack of space we refer the reader to (Baral, Gelfond, and Rushton 2009) for the definition of $\Pi$.

---

¹http://reasoning.eas.asu.edu/f2lp/
²http://potassco.sourceforge.net
⁶Alternatively, we may turn it into the "reward" way, i.e., turning it into $\therefore \text{not} Body_i \quad [\neg w_i]$, but the rule may not be in the input language of CLINGO.

⁷Note that here "=" is just a part of the symbol for propositional atoms, and is not equality in first-order logic.
An atom \( c(\vec{u}) = v \) is called \emph{possible} in a possible world \( W \) due to a random selection rule (5) if \( \Pi \) contains (5) such that \( W \models Body \land p(v) \land \neg \text{Intervene}(c(\vec{u})). \) Pr-atom (6) is \emph{applied} in \( W \) if \( c(\vec{u}) = v \) is possible in \( W \) due to \( r \) and \( W \models C. \)

As in (Baral, Gelfond, and Rushton 2009), we assume that all P-log programs \( \Pi \) satisfy the following conditions:

- **Condition 1 [Unique random selection rule]**: If a P-log program \( \Pi \) contains two random selection rules for \( c(\vec{u}) \):
  \begin{align*}
  [r_1] \text{ random}(c(\vec{u}) : \{x : p_1(x)\}) & \leftarrow Body_1, \\
  [r_2] \text{ random}(c(\vec{u}) : \{x : p_2(x)\}) & \leftarrow Body_2,
  \end{align*}

  then no possible world of \( \Pi \) satisfies both \( Body_1 \) and \( Body_2 \).

- **Condition 2 [Unique probability assignment]**: If a P-log program \( \Pi \) contains a random selection rule for \( c(\vec{u}) \):
  \[ [v] \text{ random}(c(\vec{u}) : \{x : p(x)\}) \leftarrow Body \]

  along with two different pr-atoms:
  \begin{align*}
  pr_r(c(\vec{u}) = v | C_1) &= p_1, \\
  pr_r(c(\vec{u}) = v | C_2) &= p_2,
  \end{align*}

  then no possible world of \( \Pi \) satisfies \( Body, C_1, \) and \( C_2 \) together.

   Given a P-log program \( \Pi \), 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 (5) such that \( W \models Body \). Let \( r_{W,c(\vec{u})} \) denote this random selection rule, and let \( AV_W(c(\vec{u})) = \{v' : \text{there exists a pr-atom } pr_{r_{W,c(\vec{u})}}(c(\vec{u}) = v' | C) = p \text{ that is applied in } W \text{ for some } C \text{ and } p \} \). We then define the following notations:

- If \( v \in AV_W(c(\vec{u})) \), there exists a pr-atom \( pr_{r_{W,c(\vec{u})}}(c(\vec{u}) = v | C) = p \in \Pi \) for some \( C \) and \( p \) such that \( W \models C \). By **Condition 2**, for any other \( pr_{r_{W,c(\vec{u})}}(c(\vec{u}) = v | C') = p' \) in \( \Pi \), 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
  \[ \text{PossWithAssPr}(W, c(\vec{u}) = v) = p. \]

("\( c(\vec{u}) = v \) is possible in \( W \) with assigned probability \( p \).")

- If \( v \notin AV_W(c(\vec{u})) \), we define
  \[ \text{PossWithDefPr}(W, c(\vec{u}) = v) = \max \{ p, 0 \}, \]
  where \( p \) is
  \[ 1 - \sum_{v' \in AV_W(c(\vec{u}))} \text{PossWithAssPr}(W, c(\vec{u}) = v') \]
  \[ \cdot \left[ \left\{ v'' : c(\vec{u}) = v'' \text{ is possible in } W \text{ and } v'' \notin AV_W(c(\vec{u})) \right\} \right]. \]

("\( c(\vec{u}) = v \) is possible in \( W \) with the default probability.")

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

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

The \emph{unnormalized probability} of a possible world \( W \) is defined as
\[
\tilde{\mu}_\Pi(W) = \prod_{c(\vec{u}) \models W \text{ and } c(\vec{u}) \models W \text{ is possible in } W} P(W, c(\vec{u}) = v),
\]
and, assuming \( \Pi \) has at least one possible world with nonzero unnormalized probability, the \emph{normalized probability} of \( W \) is defined as
\[
\mu_\Pi(W) = \frac{\tilde{\mu}_\Pi(W)}{\sum_{W_i \in \Omega(\Pi)} \tilde{\mu}_\Pi(W_i)}.
\]

We say \( \Pi \) is \emph{consistent} if \( \Pi \) has at least one possible world with a non-zero probability.

**Example 3** Consider a variant of the Monty Hall Problem encoding in P-log from (Baral, Gelfond, and Rushton 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 \( \Pi \), using both assigned probability and default probability, as
\[
\neg \text{CanOpen}(d) \leftarrow \text{Selected} \iff d \in \{1, 2, 3, 4\},
\neg \text{CanOpen}(d) \leftarrow \text{Prize} \iff d.
\]
\[
\text{CanOpen}(d) \leftarrow \neg \neg \text{CanOpen}(d).
\]
\[
\text{random}(\text{Prize}). \quad \text{random}(\text{Selected}).
\]
\[
\text{random}(\text{Open} : \{x : \text{CanOpen}(x)\}).
\]
\[
pr(\text{Prize} = 1) = 0.3. \quad pr(\text{Prize} = 3) = 0.2.
\]
\[
\text{Obs}(\text{Selected} = 1). \quad \text{Obs}(\text{Open} = 2). \quad \text{Obs}(\text{Prize} \ne 2).
\]

The possible worlds of \( \Pi \) are as follows:

- \( W_1 = \{\text{Obs}(\text{Selected} = 1), \text{Obs}(\text{Open} = 2), \text{Obs}(\text{Prize} \ne 2), \text{Selected} = 1, \text{Open} = 2, \text{Prize} = 1, \text{CanOpen}(1) = \text{false}, \text{CanOpen}(2) = \text{true}, \text{CanOpen}(3) = \text{true}, \text{CanOpen}(4) = \text{false} \} \)
- \( W_2 = \{\text{Obs}(\text{Selected} = 1), \text{Obs}(\text{Open} = 2), \text{Obs}(\text{Prize} \ne 2), \text{Selected} = 1, \text{Open} = 2, \text{Prize} = 3, \text{CanOpen}(1) = \text{false}, \text{CanOpen}(2) = \text{false}, \text{CanOpen}(3) = \text{true}, \text{CanOpen}(4) = \text{false} \} \)
- \( W_3 = \{\text{Obs}(\text{Selected} = 1), \text{Obs}(\text{Open} = 2), \text{Obs}(\text{Prize} \ne 2), \text{Selected} = 1, \text{Open} = 2, \text{Prize} = 4, \text{CanOpen}(1) = \text{false}, \text{CanOpen}(2) = \text{false}, \text{CanOpen}(3) = \text{true}, \text{CanOpen}(4) = \text{false} \} \)

The probability of each atom to happen is
\[
P(W_1, \text{Selected} = 1) = \text{PossWithDefPr}(W, \text{Selected} = 1) = 1/4
\]
\[
P(W_1, \text{Open} = 2) = \text{PossWithDefPr}(W, \text{Open} = 2) = 1/3
\]
\[
P(W_2, \text{Open} = 2) = \text{PossWithDefPr}(W, \text{Open} = 2) = 1/2
\]
\[
P(W_3, \text{Open} = 2) = \text{PossWithDefPr}(W, \text{Open} = 2) = 1/2
\]
\[
P(W_1, \text{Prize} = 1) = \text{PossWithAssPr}(W, \text{Prize} = 1) = 0.3
\]
\[
P(W_2, \text{Prize} = 3) = \text{PossWithAssPr}(W, \text{Prize} = 3) = 0.2
\]
\[
P(W_3, \text{Prize} = 4) = \text{PossWithDefPr}(W, \text{Prize} = 4) = 0.25
\]
So,
• $\mu(W_1) = 1/4 \times 1/3 \times 0.3 = 1/40$
• $\mu(W_2) = 1/4 \times 1/2 \times 0.2 = 1/40$
• $\mu(W_3) = 1/4 \times 1/2 \times 0.25 = 1/32$.

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%.

Turning P-log into LPMLN

We define translation $\text{plog2lpmln}(\Pi)$ that turns a P-log program $\Pi$ into an LPMLN program in a modular way. First, every rule $R$ in $\tau(\Pi)$ (that is used in defining the possible worlds in P-log) is turned into a hard rule $\alpha : R$ in $\text{plog2lpmln}(\Pi)$. In addition, $\text{plog2lpmln}(\Pi)$ contains the following rules to associate probability to each possible world of $\Pi$. Below $x$, $y$ denote schematic variables, and $W$ is a possible world of $\Pi$.

Possible Atoms: For each random selection rule (5) for $c(\vec{u})$ in $S$ and for each $v \in \text{Range}(c)$, $\text{plog2lpmln}(\Pi)$ includes

$$\text{Poss}_r(c(\vec{u}) = v) \leftarrow \text{Body}, p(v), \text{not Intervene}(c(\vec{u})) \quad (8)$$

Rule (8) expresses that $c(\vec{u}) = v$ is possible in $W$ due to $r$ if $W \models \text{Body} \land p(v) \land \text{not Intervene}(c(\vec{u}))$.

Assigned Probability: For each pr-atom (6) in $P$, $\text{plog2lpmln}(\Pi)$ contains the following rules:

$$\alpha : \text{PossWithAssPr}_{r,C}(c(\vec{u}) = v) \leftarrow \text{Poss}_r(c(\vec{u}) = v), C \quad (9)$$

$$\alpha : \text{AssPr}_{r,C}(c(\vec{u}) = v) \leftarrow c(\vec{u}) = v, \text{PossWithAssPr}_{r,C}(c(\vec{u}) = v) \quad (10)$$

$$\text{ln}(p) : \bot \leftarrow \text{not AssPr}_{r,C}(c(\vec{u}) = v) \quad (p > 0) \quad (11)$$

$$\alpha : \bot \leftarrow \text{AssPr}_{r,C}(c(\vec{u}) = v) \quad (p = 0) \quad (12)$$

$$\alpha : \text{PossWithAssPr}(c(\vec{u}) = v) \leftarrow \text{PossWithAssPr}_{r,C}(c(\vec{u}) = v).$$

Rule (9) expresses the condition under which pr-atom (6) is applied in a possible world $W$. Further, if $c(\vec{u}) = v$ is true in $W$ as well, rules (10) and (11) contribute the assigned probability $e^{\text{ln}(p)} = p$ to the unnormalized probability of $W$ as a factor when $p > 0$.

Denominator for Default Probability: For each random selection rule (5) for $c(\vec{u})$ in $S$ and for each $v \in \text{Range}(c)$, $\text{plog2lpmln}(\Pi)$ includes

$$\alpha : \text{PossWithDefPr}(c(\vec{u}) = v) \leftarrow \text{Poss}_r(c(\vec{u}) = v), \text{not PossWithAssPr}(c(\vec{u}) = v) \quad (12)$$

$$\alpha : \text{NumDefPr}(c(\vec{u}), x) \leftarrow c(\vec{u}) = v, \text{PossWithDefPr}(c(\vec{u}) = v), x = \#\text{count}(y : \text{PossWithDefPr}(c(\vec{u}) = y)) \quad (m = 2, \ldots, \text{Range}(c)) \quad (13)$$

$$\text{ln}(\frac{1}{m}) : \bot \leftarrow \text{not NumDefPr}(c(\vec{u}), m) \quad \text{(m = 2, \ldots, \text{Range}(c))} \quad (14)$$

Rule (12) 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 (13) expresses, intuitively, that $\text{NumDefPr}(c(\vec{u}), x)$ 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 (7). Then rule (14) contributes the factor $1/x$ to the unnormalized probability of $W$ as a factor.

Numerator for Default Probability:

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

$$\text{pr}_r(c(\vec{u}) = v_1 \mid C_1) = p_1$$

$$\ldots$$

$$\text{pr}_r(c(\vec{u}) = v_n \mid C_n) = p_n$$

where $n \geq 1$, and $v_1$ and $v_j$ ($i \neq j$) may be equal. For each $v \in \text{Range}(c)$, $\text{plog2lpmln}(\Pi)$ contains the following rules:

$$\alpha : \text{RemPr}(c(\vec{u}), 1 - y) \leftarrow \text{Body} \quad (15)$$

$$\text{ln}(x) : \bot \leftarrow \text{not TotalDefPr}(c(\vec{u}), x), x > 0 \quad (16)$$

$$\alpha : \bot \leftarrow \text{RemPr}(c(\vec{u}), x), x \leq 0 \quad (17)$$

In rule (15), $y$ is the sum of all assigned probabilities. Rules (16) and (17) are to account for the numerator of (7) when $n > 0$. The variable $x$ stands for the numerator of (7). Rule (18) is to avoid assigning a non-positive default probability to a possible world.

Note that most rules in $\text{plog2lpmln}(\Pi)$ are hard rules. The soft rules (11), (14), (17) cannot be simplified as atomic facts, e.g., $\text{ln}(\frac{1}{m}) : \text{NumDefPr}(c(\vec{u}), m)$ in place of (14), 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 world 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 LPMLN (Lee and Wang 2016), it is interesting to note that both kinds of probabilities can be captured in LPMLN using different kinds of rules.

Example 3 Continued For the program $\Pi$ in Example 3, $\text{plog2lpmln}(\Pi)$ consists of the rules $\alpha : R$ for each rule $R$ in $\tau(\Pi)$ and the following rules.

Possible Atoms:

$$\alpha : \text{Poss}(\text{Price} = d) \leftarrow \text{not Intervene(Price)}$$

$$\alpha : \text{Poss}(\text{Selected} = d) \leftarrow \text{not Intervene(Selected)}$$

$$\alpha : \text{Poss}(\text{Open} = d) \leftarrow \text{CanOpen(d)}, \text{not Intervene(Open)}$$

\[10\]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, Lifschitz, and Yang 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.
Assigned Probability:

\[
\alpha : \text{PossWithAssPr}(\text{Prize} = 1) \leftrightarrow \text{Poss}(\text{Prize} = 1)
\]
\[
\alpha : \text{AssPr}(\text{Prize} = 1) \leftrightarrow \text{Prize} = 1, \text{PossWithAssPr}(\text{Prize} = 1)
\]
\[
\ln(0.3) : \bot \not\leftrightarrow \text{not AssPr}(\text{Prize} = 1)
\]
\[
\alpha : \text{PossWithAssPr}(\text{Prize} = 3) \leftrightarrow \text{Poss}(\text{Prize} = 3)
\]
\[
\alpha : \text{AssPr}(\text{Prize} = 3) \leftrightarrow \text{Prize} = 3, \text{PossWithAssPr}(\text{Prize} = 3)
\]
\[
\ln(0.2) : \bot \not\leftrightarrow \text{not AssPr}(\text{Prize} = 3)
\]

(We simplified slightly not to distinguish PossWithAssPr(\cdot) and PossWithAssPr_{r,C}(\cdot) because there is only one random selection rule for Prize and both pr-atoms for Prize has empty conditions.)

Denominator for Default Probability:

\[
\alpha : \text{PossWithDefPr}(\text{Prize} = d) \leftrightarrow \text{Poss}(\text{Prize} = d), \text{not PossWithAssPr}(\text{Prize} = d)
\]
\[
\alpha : \text{PossWithDefPr}(\text{Selected} = d) \leftrightarrow \text{Poss}(\text{Selected} = d), \text{not PossWithAssPr}(\text{Selected} = d)
\]
\[
\alpha : \text{PossWithDefPr}(\text{Open} = d) \leftrightarrow \text{Poss}(\text{Open} = d), \text{not PossWithAssPr}(\text{Open} = d)
\]
\[
\alpha : \text{NumDefPr}(\text{Prize}, x) \leftrightarrow \text{Prize} = d, \text{PossWithDefPr}(\text{Prize} = d),
\]
\[
x = \#\text{cnt}(y : \text{PossWithDefPr}(\text{Prize} = y))
\]
\[
\alpha : \text{NumDefPr}(\text{Selected}, x) \leftrightarrow \text{Selected} = d, \text{PossWithDefPr}(\text{Selected} = d),
\]
\[
x = \#\text{cnt}(y : \text{PossWithDefPr}(\text{Selected} = y))
\]
\[
\alpha : \text{NumDefPr}(\text{Open}, x) \leftrightarrow \text{Open} = d, \text{PossWithDefPr}(\text{Open} = d),
\]
\[
x = \#\text{cnt}(y : \text{PossWithDefPr}(\text{Open} = y))
\]
\[
\ln(\frac{1}{m}) : \bot \not\leftrightarrow \text{not NumDefPr}(c, m)
\]
\[
(c \in \{\text{Selected}, \text{Open}\}, m \in \{2, 3, 4\})
\]

Numerator for Default Probability:

\[
\alpha : \text{RemPr}(\text{Prize}, 1-x) \leftrightarrow \text{Prize} = d, \text{PossWithDefPr}(\text{Prize} = d),
\]
\[
x = \#\text{sum}(0.3 : \text{PossWithDefPr}(\text{Prize} = 1))
\]
\[
\alpha : \text{TotalDefPr}(\text{Prize}, x) \leftrightarrow \text{RemPr}(\text{Prize}, x), x > 0
\]
\[
\ln(x) : \bot \not\leftrightarrow \text{not TotalDefPr}(\text{Prize}, x)
\]
\[
\alpha : \bot \not\leftrightarrow \text{RemDefPr}(\text{Prize}, x), x \leq 0
\]

Clearly, the signature of plog2pmln(\Pi) is a superset of the signature of \Pi. Further, plog2pmln(\Pi) 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 \Pi and the set of the stable models of plog2pmln(\Pi) such that each stable model is an extension of the possible world, and the probability of each possible world of \Pi coincides with the probability of the corresponding stable model of plog2pmln(\Pi).

**Theorem 3** Let \Pi be a consistent P-log program. There is a 1-1 correspondence \phi between the set of the possible worlds of \Pi with non-zero probabilities and the set of probabilistic stable models of plog2pmln(\Pi) such that

- For every possible world \text{W} of \Pi that has a non-zero probability, \phi(\text{W}) is a probabilistic stable model of plog2pmln(\Pi), and \mu_{\Pi}(\text{W}) = P_{\text{plog2pmln}(\Pi)}(\phi(\text{W})).

- For every probabilistic stable model I of plog2pmln(\Pi), the restriction of I onto the signature of \tau(\Pi), denoted I_{\tau(\Pi)}, is a possible world of \Pi and \mu_{\Pi}(I_{\tau(\Pi)}) > 0.

**Proof.** (Sketch) We can check that the following mapping \phi is the 1-1 correspondence.

1. \phi(\text{W}) := \text{Poss}_{r}(c(\bar{u}) = v) iff \text{c(\bar{u}) = v is possible in W due to r.}

2. For each pr-atom \text{pr}_{r}(c(\bar{u}) = v | C) = p in \Pi,
\phi(\text{W}) := \text{PossWithAssPr}_{r,C}(c(\bar{u}) = v) iff this pr-atom is applied in W.

3. For each pr-atom \text{pr}_{r}(c(\bar{u}) = v | C) = p in \Pi,
\phi(\text{W}) := \text{AssPr}_{r,C}(c(\bar{u}) = v) iff this pr-atom is applied in W, and \text{W} \models c(\bar{u}) = v.

4. \phi(\text{W}) := \text{PossWithDefPr}(c(\bar{u}) = v) iff \text{v \in AV_{W}(c(\bar{u}))}.

5. \phi(\text{W}) := \text{PossWithDefPr}(c(\bar{u}) = v) iff \text{c(\bar{u}) = v is possible in W and v \not\in AV_{W}(c(\bar{u}))}.

6. \phi(\text{W}) := \text{NumDefPr}(c(\bar{u}), m) iff there exist exactly m different values \text{v such that c(\bar{u}) = v is possible in W; v \not\in AV_{W}(c(\bar{u}))}; and, for one of such \text{v, W} \models c(\bar{u}) = v.

7. \phi(\text{W}) := \text{RemPr}(c(\bar{u}), k) iff there exists a value \text{v such that W \models c(\bar{u}) = v; c(\bar{u}) = v is possible in W; v \not\in AV_{W}(c(\bar{u})}); and
\[k = 1 - \sum_{v \in AV_{W}(c(\bar{u}))} \text{PossWithAssPr}(W, c(\bar{u}) = v).
\]

8. \phi(\text{W}) := \text{TotalDefPr}(c(\bar{u}), k) iff \phi(\text{W}) \models \text{RemPr}(c(\bar{u}), k) and \text{K > 0}.

To check that \mu_{\Pi}(\text{W}) = P_{\text{plog2pmln}(\Pi)}(\phi(\text{W})) note first that the weight of \phi(\text{W}) is computed by multiplying \text{c} to the power of the weights of rules (11), (14), (17) that are satisfied by \phi(\text{W}). Let’s call this product \text{TW}.

Consider a possible world \text{W} with a non-zero probability of \text{II} and \text{c(\bar{u}) = v} is satisfied by \text{W}.

If \text{c(\bar{u}) = v is possible in W and pr-atom \text{pr}_{r}(c(\bar{u}) = v | C) = p is applied in W (i.e., v \in AV_{W}(c(\bar{u}))}, then the assigned probability is applied: \text{P}(W, c\bar{u} = v) = p. On the other hand, by condition 3, \text{\phi(\text{W}) \models AssPr}_{r,C}(c(\bar{u}) = v), so that from (11), the same \text{p is a factor of TW}.

If \text{c(\bar{u}) = v is possible in W and v \not\in AV_{W}(c(\bar{u}))}, the default probability is applied: \text{P}(W, c\bar{u} = v) = p is computed by (7). By Condition 8, \text{\phi(\text{W}) \models TotalDefPr}(c\bar{u}, x) where x = 1 - \sum_{v \in AV_{W}(c(\bar{u}))} \text{PossWithAssPr}(W, c\bar{u} = v').

Since \text{\phi(\text{W}) \models (17)}, it is a factor of \text{TW}, which is the same as the numerator of (7). Furthermore, by Condition 6, \text{\phi(\text{W}) \models NumDefPr}(c\bar{u}, m), where m is the denominator of (7). Since \text{\phi(\text{W}) \models (14)}, \frac{1}{m} is a factor of \text{TW}. ■

**Example 3 Continued** For the P-log program \Pi for the Monty Hall problem, \Pi = plog2pmln(\Pi) 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: Poss(Prize = i) for i = 1, 2, 3, 4; Poss(Selected = i) for i = 1, 2, 3, 4; PossWithAssPr(Prize =
i) for $i = 1, 3$: $\text{PossWithDefPr}(\text{Prize} = i)$ for $i = 2, 4$; $\text{PossWithDefPr}(\text{Selected} = i)$ for $i = 1, 2, 3, 4$; $\text{NumDefPr}(\text{Selected}, 4)$. In addition,

- $I_1 \models \{\text{AssPr}(\text{Prize} = 1), \text{Poss}(\text{Open} = 2), \text{Poss}(\text{Open} = 3), \text{Poss}(\text{Open} = 4), \text{PossWithDefPr}(\text{Open} = 2), \text{PossWithDefPr}(\text{Open} = 3), \text{PossWithDefPr}(\text{Open} = 4), \text{NumDefPr}(\text{Open}, 3)\}$
- $I_2 \models \{\text{AssPr}(\text{Prize} = 3), \text{Poss}(\text{Open} = 2), \text{Poss}(\text{Open} = 4), \text{PossWithDefPr}(\text{Open} = 2), \text{PossWithDefPr}(\text{Open} = 4), \text{NumDefPr}(\text{Open}, 2)\}$
- $I_3 \models \{\text{Poss}(\text{Open} = 2), \text{Poss}(\text{Open} = 3), \text{PossWithDefPr}(\text{Open} = 2), \text{PossWithDefPr}(\text{Open} = 3), \text{NumDefPr}(\text{Open}, 2), \text{NumDefPr}(\text{Prize}, 2), \text{RemPr}(\text{Prize}, 0.5), \text{TotalDefPr}(\text{Prize}, 0.5)\}$.

The unnormalized weight $W_{I_i}$ of each probabilistic stable model $I_i$ is shown below. $w(\text{AssPr}, c_{i}(\vec{u}) = v))$ denotes the exponentiated weight of rule (11); $w(\text{NumDefPr}(c_{i}(\vec{u}), m))$ denotes the exponentiated weight of rule (14); $w(\text{TotalDefPr}(c_{i}(\vec{u}), x))$ denotes the exponentiated weight of rule (17).

- $W_{I_1} = w(\text{NumDefPr}(\text{Selected}, 4)) \times w(\text{AssPr}(\text{Prize} = 1)) \times w(\text{NumDefPr}(\text{Open}, 3)) = \frac{1}{4} \times \frac{2}{15} \times \frac{5}{10} = \frac{1}{30}$.
- $W_{I_2} = w(\text{NumDefPr}(\text{Selected}, 4)) \times w(\text{AssPr}(\text{Prize} = 3)) \times w(\text{NumDefPr}(\text{Open}, 2)) = \frac{1}{4} \times \frac{2}{15} \times \frac{1}{5} = \frac{1}{75}$.
- $W_{I_3} = w(\text{NumDefPr}(\text{Selected}, 4)) \times w(\text{TotalDefPr}(\text{Open}, 2)) \times w(\text{NumDefPr}(\text{Prize}, 2)) \times w(\text{TotalDefPr}(\text{Prize}, 0.5)) = \frac{1}{4} \times \frac{2}{15} \times \frac{5}{15} = \frac{1}{90}$.

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

**Conclusion**

In this paper, we show how LP$^{MLN}$ is related to 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$^{MLN}$ is shown to be a good middle ground language that clarifies the relationships. We expect the relationships will help us to apply the mathematical and computational results developed for one language to another language.

**Acknowledgments** We are grateful to Evgenii Balai, Yi Wang and anonymous referees for their useful comments on the draft of this paper. This work was partially supported by the National Science Foundation under Grants IIS-1319794 and IIS-1526301.

**References**


Appendix to “LPMLN, Weak Constraints, and P-log”

The appendix contains

• Proofs in order of Proposition 1, Theorem 1, Theorem 2, Corollary 1, Corollary 2, Corollary 3, Corollary 4, and Theorem 3: (Corollary 3 and Corollary 4 are corollaries of Corollary 2 when lpmln2wc is simplified)

• The full LPMLN encoding and full ASP with weak constraints encoding of the variant Monty Hall problem.

Proof of Proposition 1

Proposition 1 For any LPMLN program $\Pi$, the set $SM[\Pi]$ is exactly the set of the stable models of $lpmln2wc(\Pi)$.

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

$$I \in SM[\Pi] \iff I \text{ is a stable model of } lpmln2wc(\Pi). \quad (19)$$

Since $I \in SM[\Pi] \iff I$ is a stable model of $\Pi$, by definition, (19) is equivalent to saying

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

which is true because

$$\bigwedge_{w: F \in \Pi} (\{F\}^{ch})^I = \bigwedge_{w: F \in \Pi, I \models F} (\{F\}^{ch})^I \land \bigwedge_{w: F \in \Pi, I \nabla F} (\{F\}^{ch})^I = \bigwedge_{w: F \in \Pi, I \models F} F^I.$$

Proof of Theorem 1

Let $\Pi$ be an LPMLN program. By $\Pi^{soft}$ we denote the set of all soft rules in $\Pi$, by $\Pi^{hard}$ we denote the set of all hard rules in $\Pi$. For any $I \in SM[\Pi]$, let $W^\Pi_{\text{hard}}(I) = \exp \left( \sum_{w: F \in (\Pi^{hard})_I} w \right)$ and $W^\Pi_{\text{soft}}(I) = \exp \left( \sum_{w: F \in (\Pi^{soft})_I} w \right)$, then $I$ is a most probable stable model of $\Pi$ iff

$$I \in \arg\max_{J: J \in \arg\max_{K: K \in SM[\Pi]} W^\Pi_{\text{hard}}(K)} W^\Pi_{\text{soft}}(J).$$

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

$$\sim F \ [\text{Weight} @ \text{Level}]$$

in $\Pi'$. $I$ is an optimal stable model of $\Pi'$ iff

$$I \in \arg\min_{J: J \in \arg\min_{K: K \text{ is a stable model of } \Pi'} \text{Penalty}_{\Pi'}(K, 0)} \text{Penalty}_{\Pi'}(J, 0).$$

Theorem 1 For any LPMLN program $\Pi$, the most probable stable models of $\Pi$ are precisely the optimal stable models of the program with weak constraints $lpmln2wc(\Pi)$.

Proof. Let $\Pi'$ denote $lpmln2wc(\Pi)$. To prove Theorem 1, it is sufficient to prove

$I$ is a most probable stable model of $\Pi$ iff $I$ is an optimal stable model of $\Pi'$

which is equivalent to proving

$$I \in \arg\max_{J: J \in \arg\max_{K: K \in SM[\Pi]} W^\Pi_{\text{hard}}(K)} W^\Pi_{\text{soft}}(J) \iff I \in \arg\min_{J: J \in \arg\min_{K: K \text{ is a stable model of } \Pi'} \text{Penalty}_{\Pi'}(K, 0)} \text{Penalty}_{\Pi'}(J, 0).$$
This is clear because

\[
J: \argmax_{J} \argmax_{K} \\text{argmax}_{\Pi} W_{\text{hard}}(J) = (\text{by (19) and the definition of } W_{\text{hard}}(I) \text{ and } W_{\text{soft}}(I)) \argmax_{K} \\text{argmax}_{\Pi} \left( \sum_{w: F \in \Pi_{\text{soft}}(J)} w \right)
\]

\[
J: \argmax_{K} \left( \sum_{w: F \in \Pi_{\text{soft}}(J)} w \right) = \argmin_{J} \left( \sum_{w: F \in \Pi_{\text{soft}}(J)} w \right)
\]

\[
J: \argmin_{K} \left( \sum_{w: F \in \Pi_{\text{soft}}(J)} w \right) = \argmin_{J} \left( \sum_{w: F \in \Pi_{\text{soft}}(J)} w \right)
\]

Proof of Theorem 2

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

**Proof.** For any Markov Logic Network \(\Pi\), we obtain an \(\text{LP}^\text{MLN}\) program \(\Pi'\) from \(\Pi\) by adding

\[
\alpha : \{A\}^\text{ch} = \text{for every atom } A \text{ in } \Pi.
\]

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

- \(I\) is a most probable model of the MLN program \(\Pi\)

iff

- \(I\) is a most probable stable model of the \(\text{LP}^\text{MLN}\) program \(\Pi'\)

iff (by Theorem 1)

- \(I\) is an optimal stable model of the ASP program with weak constraints \(\text{lpmln2wc}(\Pi')\)

iff (since a choice rule is always satisfied, omiting the weak constraint “\(\sim\{A\}^\text{ch} \cdot [\sim 1 @ 1]\)” for all atoms \(A\) in \(\Pi\) doesn’t affect what is an optimal stable model of \(\text{lpmln2wc}(\Pi')\))

- \(I\) is an optimal stable model of the ASP program with weak constraints \(\text{lpmln2wc}(\Pi) \cup \{\{A\}^\text{ch} \cdot [\sim 1 @ 1] \mid A \text{ is an atom in } \Pi\}\)

iff (since for any interpretation \(I\), the reduct of \(\{\{A\}^\text{ch} \cdot [\sim 1 @ 1] \mid A \text{ is an atom in } \Pi\}\) is strongly equivalent to \(\text{lpmln2wc}(\Pi) \cup \{\{A\}^\text{ch} \cdot [\sim 1 @ 1] \mid A \text{ is an atom in } \Pi\}\))

Thus we proved \(I\) is a most probable model of an MLN program \(\Pi\) iff \(I\) is an optimal stable model of the ASP program with weak constraints \(\text{mln2wc}(\Pi)\).
Proof of Corollary 1

**Corollary 1** For any LP\(^{MLN}\) program \(\Pi\), the most probable stable models of \(\Pi\) are precisely the optimal stable models of the program with weak constraints lpm\(\Pi\)we\(^{int}\) (\(\Pi\)).

**Proof.** Let \(\Pi'\) denote lpm\(\Pi\)we\(^{int}\) (\(\Pi\)). From (19), it’s clear that

\[
I \in \text{SM}[\Pi] \iff I \text{ is a stable model of } \Pi'.
\]

To prove

\[
I \text{ is a most probable stable model of } \Pi \iff I \text{ is an optimal stable model of lpm\(\Pi\)we\(^{int}\) (\(\Pi\)),}
\]

it is equivalent to proving

\[
I \in \argmax_{J: J \subseteq \text{SM}[\Pi]} W_\Pi^{\text{soft}}(J) \iff I \in \argmin_{J: J \subseteq \text{SM}[\Pi]} \text{Penalty}_\Pi(J, 0).
\]

This is clear because

\[
\argmax_{J: J \subseteq \text{SM}[\Pi]} W_\Pi^{\text{soft}}(J) = \argmax_{J: J \subseteq \text{SM}[\Pi]} \exp\left( \sum_{\alpha : F \in [\Pi]^{\text{hard}}} \alpha \right)
\]

(by (20) and the definition of \(W_\Pi^{\text{hard}}(I)\) and \(W_\Pi^{\text{soft}}(I)\))

\[
= \argmax_{J: J \subseteq \text{SM}[\Pi]} \exp\left( \sum_{w : F \in [\Pi]^{\text{soft}}} w \right)
\]

(\(I\) is a fixed integer that equals to the number of hard rules in \(\Pi\), and \(w : F \in [\Pi]^{\text{soft}}, J \models F\) is a fixed real number that equals to the sum of the weights of all soft rules in \(\Pi\))

\[
\argmin_{J: J \subseteq \text{SM}[\Pi]} \left( \sum_{\alpha : F \in [\Pi]^{\text{hard}}, J \models F} \right) \left( \sum_{w : F \in [\Pi]^{\text{soft}}, J \models F} w \right)
\]

\[
= \argmin_{J: J \subseteq \text{SM}[\Pi]} \text{Penalty}_\Pi(J, 0).
\]

\[
\]

Proof of Corollary 2

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

- \(I|_{\sigma} \subseteq J|_{\sigma}\), and
- \(I\) and \(J\) agree on \(\sigma' \setminus \sigma\).

The proof of Corollary 2 will use a restricted version of Theorem 1 from (Bartholomew, Michael, and Lee 2013), which is reformulated as follows:

**Lemma 1** Let \(F\) be a propositional formula. An interpretation \(I\) is a stable model of \(F\) relative to signature \(\sigma\) iff

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

The proof of Corollary 2 will use a restricted version of the splitting theorem from (Ferraris, Lee, Lifschitz, and Palla 2009), which is reformulated as follows:

**Splitting Theorem** Let \(\Pi_1, \Pi_2\) be two finite ground programs, \(p, q\) be disjoint tuples of distinct atoms. If
• each strongly connected component of the dependency graph of $\Pi_1 \cup \Pi_2$ w.r.t. $p \cup q$ is a subset of $p$ or a subset of $q$.
• no atom in $p$ has a strictly positive occurrence in $\Pi_2$, and
• no atom in $q$ has a strictly positive occurrence in $\Pi_1$.

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

The proof of Corollary 2 will also use the following lemma. Here and after, $w_i : F_i$ denotes the $i$-th rule in $\Pi$, where $w_i$ could be $\alpha$ or a real number.

**Lemma 2** Let $\Pi$ be an LPM $\text{LN}$ program. There is a 1-1 correspondence $\phi$ between the set $\text{SM}[\Pi]$ and the set of the stable models of lpm$\text{ln}$2w$\text{nt}$, rule $\Pi$, where $\phi(I) = I \cup \{\text{unsat}(i) \mid \text{w}_i : F_i \in \Pi, I \not\models F_i\}$.

**Proof.** Let $\sigma$ be the signature of $\Pi$. We can check that the following mapping $\phi$ is a 1-1 correspondence:

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

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

To prove $\phi$ is a 1-1 correspondence between the set $\text{SM}[\Pi]$ and the set of the stable models of

$$\bigwedge_{w_i : F_i \in \Pi} \left((F_i \leftarrow \neg \text{unsat}(i)) \land \text{unsat}(i) \leftarrow \neg F_i\right),$$

it is sufficient to prove the following two bullets.

• **prove: for every interpretation** $I \in \text{SM}[\Pi], \phi(I)$ **is a stable model of** $(21)$.

Assume $I \in \text{SM}[\Pi]$, by $(19)$, $I$ is a stable model of

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

By Lemma 1, we know

- $I \models$

$$\bigwedge_{w_i : F_i \in \Pi, I \not\models F_i} \left(F_i^I\right),$$

- and no interpretation $K$ of signature $\sigma$ such that $K \not\prec^\sigma I$ satisfies $(22)$.

To prove $\phi(I)$ is a stable model of $(21)$, by the splitting theorem, it is sufficient to show

- $\phi(I)$ is a stable model of $

$$\bigwedge_{w_i : F_i \in \Pi} \left(\text{unsat}(i) \leftarrow \neg F_i\right)$$

relative to $\{\text{unsat}(i) \mid \text{w}_i : F_i \in \Pi\}$, and

- $\phi(I)$ is a stable model of

$$\bigwedge_{w_i : F_i \in \Pi} \left(F_i \leftarrow \neg \text{unsat}(i)\right)$$

relative to $\sigma$;

which is equivalent to showing

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

(b.1) $\phi(I) \models \bigwedge_{w_i : F_i \in \Pi, \phi(I) \models F_i} \left(F_i \leftarrow \neg \text{unsat}(i)^{\phi(I)}\right)$ \land \bigwedge_{w_i : F_i \in \Pi, \phi(I) \not\models F_i} \left(F_i \leftarrow \neg \text{unsat}(i)^{\phi(I)}\right)$,

(b.2) and no interpretation $L$ of signature $\sigma \cup \{\text{unsat}(i) \mid \text{w}_i : F_i \in \Pi\}$ such that $L \not\prec^\sigma \phi(I)$ satisfies $(23)$.

It’s clear that (a) is true by the definition of $\phi(I)$. As for (b.1), since $\phi(I) \models (F_i \leftrightarrow \neg \text{unsat}(i))$ for all $w_i : F_i \in \Pi$, $(23)$ is equivalent to

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

Then (b.1) is equivalent to saying $\phi(I) \models \bigwedge_{w_i : F_i \in \Pi, \phi(I) \models F_i} \left(F_i^\phi(I)\right)$,

which is further equivalent to saying $I \models (22)$. As for (b.2), assume for the sake of contradiction that there exists an interpretation $L$ such that $L \not\prec^\sigma \phi(I)$ satisfies $(23)$. Since $(23)$ is equivalent to $(24)$, $L \models \bigwedge_{w_i : F_i \in \Pi, \phi(I) \not\models F_i} \left(F_i^\phi(I)\right)$.

Thus we know $L|_\sigma \not\prec^\sigma I$ and $L|_\sigma \models (22)$, which contradicts with “there is no interpretation $K$ such that $K \not\prec^\sigma I$ satisfies $(22)$”. So both (b.1) and (b.2) are true. Consequently, $\phi(I)$ is a stable model of $(21)$. 

prove: for every stable model \(J\) of (21), \(J|_\sigma \in \text{SM}[\Pi]\) and \(J = \phi(J|_\sigma)\).

Assume \(J\) is a stable model of (21), by the splitting theorem,

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

Thus we have

(c) \(J \models \text{unsat}(i) \leftrightarrow \neg F_i\) for all \(w_i : F_i \in \Pi\), which follows that \(J = J|_\sigma \cup \{\text{unsat}(i) \mid w_i : F_i \in \Pi, J|_\sigma \not\models \neg F_i\}.\) In other words, \(J = \phi(J|_\sigma)\).

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

\[
\bigwedge_{w_i:F_i \in \Pi, J|_\sigma \models F_i} (F_i^J \leftarrow (\neg \text{unsat}(i))^J),
\]

(d.2) and no interpretation \(L\) of signature \(\sigma \cup \{\text{unsat}(i) \mid w_i : F_i \in \Pi\}\) such that \(L <^\sigma J\) satisfies (25).

Since \(J \models \text{unsat}(i) \leftrightarrow \neg F_i\) for all \(w_i : F_i \in \Pi, (25)\) is equivalent to

\[
\bigwedge_{w_i:F_i \in \Pi, J|_\sigma \models F_i} (F_i^J \leftarrow \top) \land \bigwedge_{w_i:F_i \in \Pi, J|_\sigma \not\models F_i} (\bot \leftarrow \bot),
\]

which is further equivalent to

\[
\bigwedge_{w_i:F_i \in \Pi, J|_\sigma \models F_i} (F_i^J|_\sigma).
\]

Thus by (d.1), \(J|_\sigma \not\models (26)\); and by (d.2), it's easy to show that no interpretation \(K\) such that \(K <^\sigma J|_\sigma\) satisfies (26). (Assume for the sake of contradiction, there exists an interpretation \(K\) such that \(K <^\sigma J|_\sigma\) satisfies (26). Let \(L = K \cup \{\text{unsat}(i) \mid w_i : F_i \in \Pi, J \models \neg F_i\}.\) Then \(L <^\sigma J\) and \(L \models (26)\). Since (26) is equivalent to (25), \(L \models (25)\), which contradicts with (d.2)).

Then by Lemma 1, \(J|_\sigma\) is a stable model of \(\bigwedge_{w_i:F_i \in \Pi} (F_i \leftarrow \neg F_i)\). By (19), \(J|_\sigma \in \text{SM}[\Pi]\).

\[\]

Corollary 2 Let \(\Pi\) be an \(\text{LPM}\text{MLN}\) program. There is a 1-1 correspondence \(\phi\) between the most probable stable models of \(\Pi\) and the optimal stable models of \(\text{lpmln}2\text{wc}\text{bnt}\text{rule}(\Pi)\), where \(\phi(I) = I \cup \{\text{unsat}(i) \mid w_i : F_i \in \Pi, I \not\models F_i\}\).

Proof. Let \(\sigma\) be the signature of \(\Pi\). We can check that the following mapping \(\phi\) is a 1-1 correspondence:

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

where \(\phi(I)\) is of an extended signature \(\sigma \cup \{\text{unsat}(i) \mid w_i : F_i \in \Pi\}\). By Lemma 2, we know \(\phi\) is a 1-1 correspondence between the set \(\text{SM}[\Pi]\) and the set of the stable models of \(\text{lpmln}2\text{wc}\text{bnt}\text{rule}(\Pi)\). Let \(\Pi'\) denote \(\text{lpmln}2\text{wc}\text{bnt}\text{rule}(\Pi)\), \(J'\) and \(K'\) denote interpretations of signature \(\sigma \cup \{\text{unsat}(i) \mid w_i : F_i \in \Pi\}\). To prove

\(I\) is a most probable stable model of \(\Pi\) iff \(\phi(I)\) is an optimal stable model of \(\text{lpmln}2\text{wc}\text{bnt}\text{rule}(\Pi)\),

it is equivalent to proving

\[
I \in \arg\max_{J : J \in \text{SM}[\Pi]} W_{\Pi}^{\text{soft}}(J) \text{ iff } \phi(I) \in \arg\min_{K' : K' \in \text{SM}[\Pi]} \text{Penalty}_{\Pi'}(J', 0),
\]

which is further equivalent to proving

\[
I \in \arg\max_{J : J \in \text{SM}[\Pi]} W_{\Pi}^{\text{soft}}(J) \text{ iff } I \in \arg\min_{K : K \in \text{SM}[\Pi]} \text{Penalty}_{\Pi'}(\phi(K), 0).
\]
(by the definition of $W^\text{hard}_\Pi(I)$ and $W^\text{soft}_\Pi(J)$)
\[
\arg\max_{J \in \mathscr{K}} \exp \left( \sum_{\alpha \in \mathcal{F}_i \in \Pi^\text{hard}, K \models F_i} \alpha \right) \left( \sum_{\alpha \in \mathcal{F}_i \in \Pi^\text{soft}, J \not\models F_i} \alpha \right) \left( \sum_{\alpha \in \mathcal{F}_i \in \Pi^\text{hard}, K \not\models F_i} \alpha \right)
\]

(since for any interpretation $K \in \text{SM}[\Pi]$, $\phi(K) \models \text{unsat}(i)$ iff $K \not\models F_i$)
\[
\arg\min_{J \in \mathscr{K}} \exp \left( \sum_{\alpha \in \mathcal{F}_i \in \Pi^\text{soft}, J \models F_i} \alpha \right) \left( \sum_{\alpha \in \mathcal{F}_i \in \Pi^\text{hard}, K \not\models F_i} \alpha \right)
\]

Proof of Corollary 3

For any $\text{LP}^{\text{MLN}}$ program $\Pi$ such that all unweighted rules of $\Pi$ are in the rule form $\text{Head} \leftarrow \text{Body}$, let $\text{lpmln2wc}^{\text{pnt,clingo}}(\Pi)$ be the translation by turning each weighted rule $w_i : \text{Head}_i \leftarrow \text{Body}_i$ in $\Pi$ into
\[
\text{unsat}(i) \leftarrow \text{Body}_i, \text{not Head}_i
\]
\[
\text{Head}_i \leftarrow \text{Body}_i, \text{not unsat}(i)
\]
\[
\models \text{unsat}(i) \leftarrow [w_i \text{ if } l = 1]
\]

where $l = 1$ if $w_i$ is $\alpha$ and $l = 0$ otherwise.

**Corollary 3** Let $\Pi$ be an $\text{LP}^{\text{MLN}}$ program such that all unweighted rules of $\Pi$ are in the rule form $\text{Head} \leftarrow \text{Body}$. There is a 1-1 correspondence $\phi$ between the most probable stable models of $\Pi$ and the optimal stable models of $\text{lpmln2wc}^{\text{pnt,clingo}}(\Pi)$, where $\phi(I) = I \cup \{\text{unsat}(i) \mid w_i : \text{Head}_i \leftarrow \text{Body}_i \in \Pi, I \models \text{Body}_i \wedge \neg \text{Head}_i\}$.

**Proof.** Let $\sigma$ denote the signature of $\Pi$. Since 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 **Corollary 2**, it suffices to prove that for any interpretation $I$ of the signature $\sigma \cup \{\text{unsat}(i) \mid w_i : \text{Head}_i \leftarrow \text{Body}_i \in \Pi\}$,
\[
I \text{ is a stable model of } \text{lpmln2wc}^{\text{pnt,clingo}}(\Pi) \text{ iff } I \text{ is a stable model of } \text{lpmln2wc}^{\text{pnt,rule}}(\Pi).
\]

By the splitting theorem, it is equivalent to proving

(a) $I$ is a stable model of $\bigwedge_{w_i : \text{Head}_i \leftarrow \text{Body}_i \in \Pi} (\text{unsat}(i) \leftarrow \text{Body}_i \wedge \neg \text{Head}_i)$ relative to $\{\text{unsat}(i) \mid w_i : \text{Head}_i \leftarrow \text{Body}_i \in \Pi\}$, and

(b) $I$ is a stable model of $\bigwedge_{w_i : \text{Head}_i \leftarrow \text{Body}_i \in \Pi} (\text{Head}_i \leftarrow \text{Body}_i \wedge \neg \text{unsat}(i))$ relative to $\sigma$;

(c) $I$ is a stable model of $\bigwedge_{w_i : \text{Head}_i \leftarrow \text{Body}_i \in \Pi} (\text{unsat}(i) \leftarrow \neg (\text{Head}_i \leftarrow \text{Body}_i))$ relative to $\{\text{unsat}(i) \mid w_i : \text{Head}_i \leftarrow \text{Body}_i \in \Pi\}$, and

(d) $I$ is a stable model of $\bigwedge_{w_i : \text{Head}_i \leftarrow \text{Body}_i \in \Pi} ((\text{Head}_i \leftarrow \text{Body}_i) \leftarrow \neg \text{unsat}(i))$ relative to $\sigma$. 
This is clear because

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

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

$$
\begin{align*}
\left( \left( Head_i \leftarrow Body_i \right) \right) & \leftarrow \neg \text{unsat}(i) \\
\end{align*}
$$

$$
\begin{align*}
&= \begin{cases} \\
\left( Head_i \leftarrow Body_i \right) \leftarrow \neg \text{unsat}(i) & \text{if } J \not\models Head_i \lor \neg Body_i \lor \text{unsat}(i), \\
\bot & \text{otherwise}; \\
\end{cases} \\
&= \begin{cases} \\
\left( Head_i \leftarrow Body_i \right) \leftarrow \neg \text{unsat}(i) & \text{if } J \not\models Head_i \lor \neg Body_i, \\
\bot & \text{otherwise}; \\
\end{cases} \\
&\Leftrightarrow \begin{cases} \\
\left( Head_i \leftarrow Body_i \right) \leftarrow \neg \text{unsat}(i) & \text{if } J \not\models Head_i \lor \neg Body_i, \\
\bot & \text{otherwise}; \\
\end{cases} \\
&= \begin{cases} \\
\left( Head_i \leftarrow Body_i \right) \leftarrow \neg \text{unsat}(i) & \text{if } J \not\models Head_i \lor \neg Body_i \lor \text{unsat}(i), \\
\bot & \text{otherwise}; \\
\end{cases} \\
&= \begin{cases} \\
\left( Head_i \leftarrow Body_i \right) \leftarrow \neg \text{unsat}(i) & \text{if } J \not\models Head_i \lor \neg Body_i \lor \text{unsat}(i), \\
\bot & \text{otherwise}; \\
\end{cases} \\
&= \left( Head_i \leftarrow Body_i \land \neg \text{unsat}(i) \right).
\end{align*}
$$

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

**Proof of Corollary 4**

For any LP<sup>MLN</sup> program $\Pi$ such that all unweighted rules of $\Pi$ are in the rule form $Head \leftarrow Body$, let $lpm\ln2w_{\text{pnt,clingo}}$ (II) be the translation by turning each weighted rule $w_i : Head \leftarrow Body$ in II into (where $l = 1$ if $w_i$ is $\alpha$ and $l = 0$ otherwise)

$$
\begin{align*}
\therefore & \quad Body_i \quad [w_i @ l] \\
\end{align*}
$$

If $Head_i$ is $\bot$, or

$$
\begin{align*}
\text{unsat}(i) & \leftarrow Body_i, \neg Head_i \\
Head_i & \leftarrow Body_i, \neg \text{unsat}(i) \\
\therefore & \quad \text{unsat}(i) \quad [w_i @ l]
\end{align*}
$$

otherwise.

**Corollary 4** Let $\Pi$ be an LP<sup>MLN</sup> program such that all unweighted rules of $\Pi$ are in the rule form $Head \leftarrow Body$. There is a 1-1 correspondence $\phi$ between the most probable stable models of $\Pi$ and the optimal stable models of $lpm\ln2w_{\text{pnt,clingo}}$ (II), where $\phi(I) = I \cup \{ \text{unsat}(i) \mid w_i : Head_i \leftarrow Body_i \in \Pi, Head_i$ is not $\bot, I \not\models Body_i \land \neg Head_i \}$.

The proof of **Corollary 4** will use the following lemma:

**Lemma 3** For any interpretation $I$ of an LP<sup>MLN</sup> program $\Pi$, let $\Pi^{\text{constr}}$ denote a set of weighted rules of the form $w : \leftarrow F$, where $w$ is $\alpha$ 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 \( \Pi_I \wedge \bigwedge_{w : \bot \leftarrow F \in \Pi_{\text{constr}}} (\bot \leftarrow F) \)

iff (by theorem 3 in (Ferraris, Lee, and Lifschitz 2011))

• \( I \) is a stable model of \( \Pi_I \) and \( I \models \bigwedge_{w : \bot \leftarrow F \in \Pi_{\text{constr}}} (\bot \leftarrow F) \)

iff (since \( I \models \bigwedge_{w : \bot \leftarrow F \in \Pi_{\text{constr}}} (\bot \leftarrow F) \) is always true)

• \( I \in \text{SM}[\Pi] \).

Proof of Corollary 4. We can check that the following mapping \( \phi \) is a 1-1 correspondence:

\[
\phi(I) = I \cup \{ \text{unsat}(i) \mid w_i : \text{Head}_i \leftarrow \text{Body}_i \in \Pi, \text{Head}_i \text{ is not } \bot, I \models \text{Body}_i \wedge \neg \text{Head}_i \},
\]

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

By Lemma 3, we know \( I \in \text{SM}[\Pi] \) iff \( I \in \text{SM}[\Pi] \)

By Corollary 3, we know \( \phi \) is a 1-1 correspondence between the set SM\[
\bigwedge_{w_i : \text{Head}_i \leftarrow \text{Body}_i \in \Pi, \text{Head}_i \text{ is not } \bot} (w_i : \text{Head}_i \leftarrow \text{Body}_i) \]

and the set of the stable models of

\[
\bigwedge_{w_i : \text{Head}_i \leftarrow \text{Body}_i \in \Pi, \text{Head}_i \text{ is not } \bot} (\text{unsat}(i) \leftarrow \text{Body}_i \wedge \neg \text{Head}_i) \wedge (\text{Head}_i \leftarrow \text{Body}_i \wedge \neg \text{unsat}(i)),
\]

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

Thus \( \phi \) is a 1-1 correspondence between the set SM[\( \Pi \)] and the set of the stable models of \( \text{lpmn2wc}_\text{clingo}^\text{pnt}(\Pi) \).

Let \( \Pi_{\text{c4}} \) denote \( \text{lpmn2wc}_\text{clingo}^\text{pnt}(\Pi) \), \( \Pi_{\text{c3}} \) denote \( \text{lpmn2wc}_\text{clingo}^\text{pnt}(\Pi) \), and \( \phi_{\text{c3}} \) denote the 1-1 correspondence in Corollary 3. By Corollary 3, it is suffices to prove that for any interpretation \( I \in \text{SM}[\Pi] \) and \( l \in \{0, 1\} \)

\[
\text{Penalty}_{\Pi_{\text{c4}}} (\phi(I), 0) = \text{Penalty}_{\Pi_{\text{c3}}} (\phi_{\text{c3}}(I), 0)
\]

and similarly,

\[
\text{Penalty}_{\Pi_{\text{c4}}} (\phi(I), 1) = \text{Penalty}_{\Pi_{\text{c3}}} (\phi_{\text{c3}}(I), 1).
\]
Proof of Theorem 3

**Definition of \( \tau(\Pi) \)**

Given a P-log program \( \Pi \) of the form (4) of signature \( \sigma_1 \cup \sigma_2 \), a (standard) ASP program \( \tau(\Pi) \) with the propositional signature

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

where \( S \) is the set of random selection rules of \( \Pi \), is constructed as follows:

- \( \tau(\Pi) \) contains all rules in \( R \).
- For each attribute \( c(\vec{u}) \) in \( \sigma_1 \), for \( v_1, v_2 \in \text{Range}(c) \), \( \tau(\Pi) \) 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 (5) in \( S \) with \( \text{Range}(c) = \{v_1, \ldots, v_n\} \), \( \tau(\Pi) \) contains the following rules:
  \[ c(\vec{u}) = v_1; \ldots; c(\vec{u}) = v_n \leftarrow \text{Body, not Intervene}(c(\vec{u})) \]
  \[ \leftarrow c(\vec{u}) = v, \text{not } p(v), \text{Body, not Intervene}(c(\vec{u})) \]

where \( \text{Intervene}(c(\vec{u})) \) means that the randomness of \( c(\vec{u}) \) is intervened (by an atomic fact \( \text{Do}(c(\vec{u}) = v) \)).
- For each atomic fact \( \text{Obs}(c(\vec{u}) = v) \) in \( \text{Obs} \), \( \tau(\Pi) \) contains the following rules:
  \[ \text{Obs}(c(\vec{u}) = v) \leftarrow \text{Obs}(c(\vec{u}) = v), \text{not } c(\vec{u}) = v \]
- For each atomic fact \( \text{Obs}(c(\vec{u}) \neq v) \) in \( \text{Obs} \), \( \tau(\Pi) \) contains the following rules:
  \[ \text{Obs}(c(\vec{u}) \neq v) \leftarrow \text{Obs}(c(\vec{u}) \neq v), c(\vec{u}) = v \]
- For each atomic fact \( \text{Do}(c(\vec{u}) = v) \) in \( \text{Act} \), \( \tau(\Pi) \) contains the following rules:
  \[ \text{Do}(c(\vec{u}) = v) \leftarrow \text{Do}(c(\vec{u}) = v) \]
  \[ c(\vec{u}) = v \leftarrow \text{Do}(c(\vec{u}) = v) \]
  \[ \text{Intervene}(c(\vec{u})) \leftarrow \text{Do}(c(\vec{u}) = v) \]

**Signature of \( \text{plog2lpmln}(\Pi) \)**

For any real number \( p \in [0, 1] \) and \( b \in \{t, f\} \), we define \([p]^b\) as follows: \([p]^b = p \) if \( b = t \), and \([p]^b = 0 \) if \( b = f \). Further, for any P-log program \( \Pi \) and any \( c(\vec{u}) \) in \( S \) of \( \Pi \), we define the set of all possible remaining (unassigned) probabilities of \( c(\vec{u}) \) in \( \Pi \), \( p_{\text{rem}}(c(\vec{u}), \Pi) \), as

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

Given a P-log program \( \Pi \) of the form (4) of signature \( \sigma_1 \cup \sigma_2 \), the signature of \( \text{plog2lpmln}(\Pi) \) is

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

where \( \sigma_3 \) is a propositional signature constructed from \( \Pi \) as follows:

\[ \sigma_3 = \{ \text{Poss}_r(c(\vec{u}) = v) \mid r \text{ is a random selection rule for } c(\vec{u}) \text{ in } \Pi \text{ and } v \in \text{Range}(c) \} \]
\[ \cup \{ \text{PossWithAssPr}_{r,c}(c(\vec{u}) = v) \mid \text{there is a pr-atom } pr_r(c(\vec{u}) = v \mid C) = p \text{ in } \Pi \} \]
\[ \cup \{ \text{AssPr}_{r,c}(c(\vec{u}) = v) \mid \text{there is a pr-atom } pr_r(c(\vec{u}) = v \mid C) = p \text{ in } \Pi \} \]
\[ \cup \{ \text{PossWithAssPr}(c(\vec{u}) = v) \mid \text{there is a random selection rule for } c(\vec{u}) \text{ in } \Pi \text{ and } v \in \text{Range}(c) \} \]
\[ \cup \{ \text{PossWithDefPr}(c(\vec{u}) = v) \mid \text{there is a random selection rule for } c(\vec{u}) \text{ in } \Pi \text{ and } v \in \text{Range}(c) \} \]
\[ \cup \{ \text{NumDefPr}(c(\vec{u}), m) \mid \text{there is a random selection rule for } c(\vec{u}) \text{ in } \Pi \text{ and } m \in \{1, \ldots, |\text{Range}(c)|\} \] \]
\[ \cup \{ \text{RemPr}(c(\vec{u}), k) \mid \text{there is a random selection rule for } c(\vec{u}) \text{ in } \Pi \text{ and } k \in p_{\text{rem}}(c(\vec{u}), \Pi) \} \]
\[ \cup \{ \text{TotalDefPr}(c(\vec{u}), k) \mid \text{there is a random selection rule for } c(\vec{u}) \text{ in } \Pi, k \in p_{\text{rem}}(c(\vec{u}), \Pi), \text{ and } k > 0 \}. \]

Let \( SM'[\Pi] \) be the set

\[ \{ I \mid I \text{ is a stable model of } \Pi_I \text{ that satisfy } \Pi_{\text{hard}} \}. \]
The unnormalized weight of \( I \) under \( \Pi \) with respect to soft rules only is defined as
\[
W^I_\Pi(I) = \begin{cases} \exp \left( \sum_{w: R \in (\Pi^{\text{non}})_I} w \right) & \text{if } I \in \text{SM}'[\Pi]; \\ 0 & \text{otherwise}. \end{cases}
\]

The normalized weight (a.k.a. probability) of \( I \) under \( \Pi \) with respect to soft rules only is defined as
\[
P^I_\Pi(I) = \frac{W^I_\Pi(I)}{\sum_{J \in \text{SM}'[\Pi]} W^J_\Pi(J)}.
\]

The proof of Theorem 3 will use the following lemma:

Lemma 4 (proposition 2 in (Lee and Wang 2016)) If \( \text{SM}'[\Pi] \) is not empty, for every interpretation \( I \) of \( \Pi \), \( P^I_\Pi(I) \) coincides with \( P_\Pi(I) \).

It follows from Lemma 4 that if \( \text{SM}'[\Pi] \) is not empty, then

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

Theorem 3 Let \( \Pi \) be a consistent P-log program, \( \sigma \) be the signature of \( \tau(\Pi) \). There is a 1-1 correspondence \( \phi \) between the set of the possible worlds of \( \Pi \) with non-zero probabilities and the set of probabilistic stable models of \( \text{plog2pmln}(\Pi) \) such that

(a) For every possible world \( W \) of \( \Pi \) that has a non-zero probability, \( \phi(W) \) is a probabilistic stable model of \( \text{plog2pmln}(\Pi) \), and \( \mu_\Pi(W) = \text{plog2pmln}(\Pi)(\phi(W)) \).

(b) For every probabilistic stable model \( I \) of \( \text{plog2pmln}(\Pi) \), \( I|_\sigma \) is a possible world of \( \Pi \), \( I = \phi(I|_\sigma) \), and \( \mu_\Pi(I|_\sigma) > 0 \).

Note that we make (b) a little bit more stronger than the statement in the main body (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 \( \Pi \), we define the mapping \( \phi \) as follows.

1. \( \phi(W) \models \text{Poss}_r(c(\bar{i}) = v) \) if \( c(\bar{i}) = v \) is possible in \( W \) due to \( r \).
2. For each pr-atom \( pr_r(c(\bar{i}) = v \mid C) = p \) in \( \Pi \), \( \phi(W) \models \text{PossWithAssPr}_{r,C}(c(\bar{i}) = v) \) if this pr-atom is applied in \( W \).
3. For each pr-atom \( pr_r(c(\bar{i}) = v \mid C) = p \) in \( \Pi \), \( \phi(W) \models \text{AssPr}_{r,C}(c(\bar{i}) = v) \) if this pr-atom is applied in \( W \), and \( W \models c(\bar{i}) = v \).
4. \( \phi(W) \models \text{PossWithAssPr}(c(\bar{i}) = v) \) if \( v \in \text{AV}_{W}(c(\bar{i})) \).
5. \( \phi(W) \models \text{PossWithDefPr}(c(\bar{i}) = v) \) if \( c(\bar{i}) = v \) is possible in \( W \) and \( v \notin \text{AV}_{W}(c(\bar{i})) \).
6. \( \phi(W) \models \text{NumDefPr}(c(\bar{i}), m) \) if there exist exactly \( m \) different values \( v \) such that \( c(\bar{i}) = v \) is possible in \( W \); \( v \notin \text{AV}_{W}(c(\bar{i})) \); and, for one of such \( v, W \models c(\bar{i}) = v \).
7. \( \phi(W) \models \text{RemPr}(c(\bar{i}), k) \) if there exists a value \( v \) such that \( W \models c(\bar{i}) = v; c(\bar{i}) = v \) is possible in \( W \); \( v \notin \text{AV}_{W}(c(\bar{i})) \); and \( k = 1 - \sum_{v \in \text{AV}_{W}(c(\bar{i}))} \text{PossWithAssPr}(W, c(\bar{i}) = v) \).
8. \( \phi(W) \models \text{TotalDefPr}(c(\bar{i}), k) \) if \( \phi(W) \models \text{RemPr}(c(\bar{i}), k) \) and \( k > 0 \).

Let’s denote \( \text{plog2pmln}(\Pi) \) as \( \Pi' \). In the following two parts, we will prove each of the two bullets of Theorem 3.

(a) For every possible world \( W \) of \( \Pi \) with a non-zero probability, to prove \( \phi(W) \) is a probabilistic stable model of \( \Pi' \), it is sufficient to prove \( \phi(W) \) is a stable model of \( \Pi'|_{\text{hard}} \). Indeed, if we prove \( \phi(W) \) is a stable model of \( \Pi'|_{\text{hard}} \), then \( \phi(W) \) is a stable model of \( \Pi' \), and \( P_{\Pi'|_{\text{hard}}}(\phi(W)) \) is always greater than 0. Consequently, \( \phi(W) \) must be a probabilistic stable model of \( \Pi' \). Since \( \Pi' = \Pi'|_{\text{hard}} \cup \Pi'|_{\text{soft}} \), and \( \Pi'|_{\text{soft}} \) is a set of soft rules of the form “\( w \leftarrow \text{not } A \)”, where \( A \) is an atom and \( w \) is a real number, by Lemma 3 (it follows from Lemma 3 that, if all \( w \) in \( \Pi'|_{\text{constr}} \) are real numbers, \( I \) is a probabilistic stable model of \( \Pi \cup \Pi'|_{\text{constr}} \) iff \( I \) is a probabilistic stable model of \( \Pi \), \( \phi(W) \) is a probabilistic stable model of \( \Pi' \).

Let \( \sigma \) denote the signature of \( \tau(\Pi), \Pi_{\text{AX}} = \Pi'|_{\text{hard}} \setminus \tau(\Pi) \). It can be seen that no atom in \( \sigma \) has a strictly positive occurrence in \( \Pi_{\text{AX}}, \) and no atom in \( \sigma_3 \) has a strictly positive occurrence in \( \tau(\Pi) \). Furthermore, the construction of \( \Pi' \) guarantees that all loops of size greater than one involves atoms in \( \sigma \) only. So each strongly connected component of the dependency graph of \( \Pi'|_{\text{hard}} \) relative to \( \sigma \cup \sigma_3 \) is a subset of \( \sigma \) or a subset of \( \sigma_3 \). By the splitting theorem, it is equivalent to show that \( \phi(W) \) is a stable model of \( \tau(\Pi) \) relative to \( \sigma \) and \( \phi(W) \) is a stable model of \( \Pi_{\text{AX}} \) relative to \( \sigma_3 \).
• **\( \phi(W) \) is a stable model of \( \tau(\Pi) \) relative to \( \sigma \):** Since \( W \) is a possible world of \( \Pi \), \( W \) is a stable model of \( \tau(\Pi) \) relative to \( \sigma \). Since \( \phi(W) \) is an extension of \( W \) and no atom in \( \phi(W) \setminus W \) belongs to \( \sigma \), \( \phi(W) \) is a stable model of \( \tau(\Pi) \) relative to \( \sigma \).

• **\( \phi(W) \) is a stable model of \( \Pi_{AUX} \) relative to \( \sigma_3 \):** Since there is no loop of size greater than one in \( \Pi_{AUX} \), we could apply completion on it. Let \( \text{Comp}[\Pi_{AUX}; \sigma_3] \) denote the program obtained by applying completion on \( \Pi_{AUX} \) with respect to \( \sigma_3 \), which is as follows:

  – For each random selection rule (5) for \( c(\vec{u}) \), for each \( v \in \text{Range}(c) \) and \( x \in \{2, \ldots, |\text{Range}(c)|\} \), \( \text{Comp}[\Pi_{AUX}; \sigma_3] \) contains:

    \[
    \text{Poss}_{r,c}(c(\vec{u}) = v) \iff \text{Body} \land p(v) \land \neg \text{Intervene}(c(\vec{u}))
    \]  \hspace{1cm} (27)

    \[
    \text{PossWithDefPr}(c(\vec{u}) = v) \iff \neg \text{PossWithAssPr}(c(\vec{u}) = v) \land \\
    \bigvee_{r' \mid \text{random}(c(\vec{u}) : \{X : p(X)\}) \leftarrow \text{Body} \in \Pi} \text{Poss}_{r',c}(c(\vec{u}) = v)
    \]  \hspace{1cm} (28)

    \[
    \text{NumDefPr}(c(\vec{u}), x) \iff x = \#\text{count}\{y : \text{PossWithDefPr}(c(\vec{u}) = y)\} \land \\
    \bigvee_{c(\vec{u}) \in \text{Range}(c)} (c(\vec{u}) = z \land \text{PossWithDefPr}(c(\vec{u}) = z))
    \]  \hspace{1cm} (29)

  – For each random selection rule (5) for \( c(\vec{u}) \) along with all pr-atoms associated with it in \( P \):

    \[
    \text{pr}_{r,c}(c(\vec{u}) = v_1 \mid C_1) = p_1 \land \\
    \ldots \land \\
    \text{pr}_{r,c}(c(\vec{u}) = v_n \mid C_n) = p_n
    \]  \hspace{1cm} (30)

    where \( n \geq 1 \), for \( i \in \{1, \ldots, n\} \), \( \text{Comp}[\Pi_{AUX}; \sigma_3] \) also contains:

    \[
    \text{PossWithAssPr}_{r,c_i}(c(\vec{u}) = v_1) \iff \text{Poss}_{r,c}(c(\vec{u}) = v_1) \land C_i
    \]  \hspace{1cm} (31)

    \[
    \text{AssPr}_{r,c_i}(c(\vec{u}) = v_1) \iff c(\vec{u}) = v_1 \land \text{PossWithAssPr}_{r,c_i}(c(\vec{u}) = v_i)
    \]  \hspace{1cm} (32)

    \[
    \text{\neg AssPr}_{r,c_i}(c(\vec{u}) = v_1) \iff (\text{if } p_i = 0)
    \]  \hspace{1cm} (33)

    \[
    \text{PossWithAssPr}(c(\vec{u}) = v_1) \iff \bigvee_{r' \mid \text{random}(c(\vec{u}) : \{X : p(X)\}) \leftarrow \text{Body} \in \Pi} \text{PossWithAssPr}_{r',c_j}(c(\vec{u}) = v_1)
    \]  \hspace{1cm} (34)

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

    \[
    \text{RemPr}(c(\vec{u}), x) \iff \bigvee_{v \in \text{Range}(c)} (c(\vec{u}) = v \land \text{PossWithDefPr}(c(\vec{u}) = v)) \land \\
    \bigvee_{r' \mid \text{random}(c(\vec{u}) : \{X : p(X)\}) \leftarrow \text{Body} \in \Pi} (\text{Body} \land x = 1 - y \land \\
    y = \#\text{sum}\{p_i : \text{PossWithAssPr}_{r',c_j}(c(\vec{u}) = v_1) ; \ldots p_n : \text{PossWithAssPr}_{r',c_n}(c(\vec{u}) = v_n)\})
    \]  \hspace{1cm} (35)

    \[
    \text{TotalDefPr}(c(\vec{u}), x) \iff \text{RemPr}(c(\vec{u}), x) \land x \geq 0
    \]  \hspace{1cm} (36)

    \[
    \neg (\text{RemPr}(c(\vec{u}), x) \land x \leq 0)
    \]  \hspace{1cm} (37)

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 (5) such that \( W \models \text{Body} \land p(v) \land \neg \text{Intervene}(c(\vec{u})) \)”.

  – a pr-atom \( \text{pr}_{r,c}(c(\vec{u}) = v_1 \mid C_i) = p_i \) is applied in \( W \)

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

  – \( v \in \text{AV}_W(c(\vec{u})) \)

    By the definition of \( \text{AV}_W(c(\vec{u})) \), it is equivalent to “there exists a pr-atom \( \text{pr}_{r,c}(c(\vec{u}) = v \mid C_i) = p_i \) that is applied in \( W \) for some \( r \) and \( i \)”.

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

  – Let’s take formula (27) into account. Consider the random selection rule \( [r] \text{random}(c(\vec{u}) : \{X : p(X)\}) \leftarrow \text{Body} \), where formula (27) is obtained. By definition, \( \phi(W) \models \text{Poss}_{r,c}(c(\vec{u}) = v) \iff \phi(W) \models \text{Poss}_{r,c}(c(\vec{u}) = v) \).

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

* $W \models Body \land p(v) \land not \text{Intervene}(c(\vec{u}))$
  
  iff (since $\phi(W)$ is an extension of $W$)

* $\phi(W) \models Body \land p(v) \land not \text{Intervene}(c(\vec{u}))$
  
  Thus formula (29) is satisfied by $\phi(W)$.

- Let’s take formula (28) into account. By definition,
  
  * $\phi(W) \models PossWithDefPr(c(\vec{u}) = v)$
    
    iff

  * $c(\vec{u}) = v$ is possible in $W$
    
    iff (by definition)

  * there exists a random selection rule $r$ such that $c(\vec{u}) = v$ is possible in $W$ due to $r$
    
    iff (by definition)

  * $\phi(W) \not\models PossWithAssPr(c(\vec{u}) = v)$
    
    iff (by definition)

  * there exists a random selection rule $r$ such that $\phi(W) \models Poss_r(c(\vec{u}) = v)$
    
    Thus formula (31) is satisfied by $\phi(W)$.

- Let’s take formula (29) into account. By definition,
  
  * $\phi(W) \models NumDefPr(c(\vec{u}), x)$
    
    iff

  * there exist exactly $x$ different $v$ such that
    
    $c(\vec{u}) = v$ is possible in $W$
    
    $v \not\in AV_W(c(\vec{u}))$
    
    iff (by definition and since $\phi(W)$ is an extension of $W$)

  * there exist exactly $x$ different $v$ such that
    
    $\phi(W) \models PossWithDefPr(c(\vec{u}) = v)$
    
    $v$ is possible in $W$ and $\phi(W) \models c(\vec{u}) = v \land PossWithDefPr(c(\vec{u}) = v)$
    
    Thus formula (29) is satisfied by $\phi(W)$.

- Let’s take formula (31) into account. Consider the pr-atom $pr_r(c(\vec{u}) = v_i \mid C_i) = p_i$ where formula (31) 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 $W$)

  * $\phi(W) \models Poss_r(c(\vec{u}) = v_i) \land C_i$
    
    Thus formula (31) 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 $pr_r(c(\vec{u}) = v \mid C’) = p’ \in I$ 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 (32) into account. Consider the pr-atom $pr_r(c(\vec{u}) = v_i \mid C_i) = p_i$ in II, where formula (32) is obtained, by definition,

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

  * this pr-atom is applied in $W$
    
    iff

  * $W \models c(\vec{u}) = v_i$
    
    iff (by definition and since $\phi(W)$ is an extension of $W$)

  * $\phi(W) \models PossWithAssPr_{r,C_i}(c(\vec{u}) = v_i) \land c(\vec{u}) = v_i$
    
    Thus formula (32) is satisfied by $\phi(W)$.

- Let’s take formula (33) into account. For any pr-atom $pr_r(c(\vec{u}) = v_i \mid C_i) = p_i$ in II 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 \( \mu_{\Pi}(W) = 0 \), which contradicts that \( \mu_{\Pi}(W) > 0 \).

Thus formula (33) is satisfied by \( \phi(W) \).

Let’s take formula (34) 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 (34) is satisfied by \( \phi(W) \).

Let’s take formula (35) 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 \( \phi(W) \models c(\vec{u}) = v \land PossWithDefPr(c(\vec{u}) = v) \)
* \( x = 1 - y \), and
* \( y = \sum_{v' \models PossWithAssPr(W, c(\vec{u}) = v')} PossWithAssPr(W, c(\vec{u}) = v') \)
  iff (by formula (34) and the definition of \( PossWithAssPr(W, c(\vec{u}) = v) \))
* there exists a \( v \) such that \( \phi(W) \models c(\vec{u}) = v \land PossWithDefPr(c(\vec{u}) = v) \)
* \( x = 1 - y \), and
* there exists a random selection rule \( r \) (5) along with all pr-atoms (30) 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 (35) is satisfied by \( \phi(W) \).

Remark: By Condition 1, there exists at most one random selection rule whose “Body” is satisfied by \( W \). Thus there is no other random selection rule \( v' \) such that \( \phi(W) \models PossWithAssPr_r,C_j(c(\vec{u}) = v_j) \) for any \( j \). Moreover, for any \( c(\vec{u}) \in \Pi \), there exists at most one \( RemPr(c(\vec{u}), x) \) that can be satisfied by \( \phi(W) \).

Let’s take formula (36) into account. By definition,

* \( \phi(W) \models TotalDefPr(c(\vec{u}), x) \)
  iff
* \( \phi(W) \models RemPr(c(\vec{u}), x) \land x > 0 \)

Thus formula (36) is satisfied by \( \phi(W) \).

Let’s take formula (37) into account. Consider the random selection rule (5) for \( c(\vec{u}) \) along with all pr-atoms (30) associated with it (\( n \geq 1 \)), where formula (37) is obtained. Assume for the sake of contradiction that \( \phi(W) \models (RemPr(c(\vec{u}), x) \land x \leq 0) \) 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 \( W \)
  · \( v \notin AV_W(c(\vec{u})) \)
* \( x = 1 - \sum_{v' \in AV_W(c(\vec{u}))} PossWithAssPr(W, c(\vec{u}) = v') \) and \( x \leq 0 \)

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

Thus formula (37) is satisfied by \( \phi(W) \).

Now we see the definition of \( \phi(W) \) guarantees that \( \phi(W) \) is a model of \( Comp[\Pi_{AUX}; \sigma_3] \). Thus \( \phi(W) \) is a stable model of \( \Pi_{AUX} \) relative to \( \sigma_3 \).
Until now we proved $\phi(W)$ is a stable model of $\Pi'$. Then, we are going to prove $\mu_{\Pi}(W) = P_W(\phi(W))$.

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

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

$$
\mu_{\Pi}(W) = \prod_{c(\bar{u}) = v : c(\bar{u}) = v \text{ is possible in } W \text{ and } W \models c(\bar{u}) = v} P(W,c(\bar{u}) = v) \times \prod_{c(\bar{u}) = v : c(\bar{u}) = v \text{ is possible in } W \text{ and } v \in AV_W(c(\bar{u}))} \text{PossWithAssPr}(W,c(\bar{u}) = v) \times \prod_{c(\bar{u}) = v : c(\bar{u}) = v \text{ is possible in } W \text{ and } v \notin AV_W(c(\bar{u}))} \text{PossWithDefPr}(W,c(\bar{u}) = v)
$$

Since $W$ is a possible world of $\Pi$ with a non-zero probability, $\mu_{\Pi}(W) > 0$. Since the statement “$v \in AV_W(c(\bar{u}))$” is equivalent to saying “$c(\bar{u}) = v$ is possible in $W$, and there exists $pr_{r_{W,c(\bar{u})}}(c(\bar{u}) = v \mid C) = p \in \Pi$ for some $C$ and $p$, and $W \models C$”, we have

$$
\mu_{\Pi}(W) = \prod_{c(\bar{u}) = v : c(\bar{u}) = v \text{ is possible in } W \text{ and } W \models c(\bar{u}) = v} P(W,c(\bar{u}) = v) \times \prod_{c(\bar{u}) = v : c(\bar{u}) = v \text{ is possible in } W \text{ and } v \in AV_W(c(\bar{u}))} \text{PossWithAssPr}(W,c(\bar{u}) = v) \times \prod_{c(\bar{u}) = v : c(\bar{u}) = v \text{ is possible in } W \text{ and } v \notin AV_W(c(\bar{u}))} \frac{1 - \sum_{v' \in AV_W(c(\bar{u}))} \text{PossWithAssPr}(W,c(\bar{u}) = v')}{|\{v'' \mid c(\bar{u}) = v'' \text{ is possible in } W \text{ and } v'' \notin AV_W(c(\bar{u}))\}|}
$$

Note that by Condition 1, the subscript $r_{W,c(\bar{u})}$ of the applied pr-atom is the only random selection rule for $c(\bar{u})$ 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 $\Pi'$. From the construction of $\Pi'$, it’s easy to see that there are only 3 kinds of soft rules: Rule (11), Rule (14), and Rule (17), which are satisfied...
if \( \phi(W) \models \text{AssPr}_{R,C}(c(\vec{u}) = v) \), \( \phi(W) \models \text{NumDefPr}(c(\vec{u}), m) \), and \( \phi(W) \models \text{TotalDefPr}(c(\vec{u}), x) \), respectively. Let’s denote the unnormalized weight of \( \phi(W) \) with respect to each of these three rules as \( W_{\Pi}^r(\phi(W)) |_{11} \), \( W_{\Pi}^r(\phi(W)) |_{14} \), \( W_{\Pi}^r(\phi(W)) |_{17} \). It’s clear that \( W_{\Pi}^r(\phi(W)) = W_{\Pi}^r(\phi(W)) |_{11} \times W_{\Pi}^r(\phi(W)) |_{14} \times W_{\Pi}^r(\phi(W)) |_{17} \).

Consider a \( c(\vec{u}) = v \) that is possible in \( W \) and \( W \models c(\vec{u}) = v \). Since \( \mu_{\Pi}(W) > 0 \), if \( v \in \text{AV}_W(c(\vec{u})) \), then \( P(W, c(\vec{u}) = v) = p > 0 \); if \( v \notin \text{AV}_W(c(\vec{u})) \), then \( 1 - \sum_{v' \in \text{AV}_W(c(\vec{u}))} \text{PossWithAssPr}(W, c(\vec{u}) = v') \) must be greater than 0. By the definition of \( \phi(W) \),

\[
W''_{\Pi} (\phi(W)) |_{11} = \exp \left( \sum_{c(\vec{u}) = v :} \sum_{\phi(W) \models \text{AssPr}_{R,C}(c(\vec{u}) = v)} \ln \left( \frac{1}{m} \right) \right)
\]

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

\[
W''_{\Pi} (\phi(W)) |_{14} = \exp \left( \sum_{c(\vec{u}) = v :} \sum_{\phi(W) \models \text{TotalDefPr}(c(\vec{u}), x)} \ln \left( 1 - \sum_{v'' \in \text{AV}_W(c(\vec{u}))} \text{PossWithAssPr}(W, c(\vec{u}) = v') \right) \right)
\]

\[
W''_{\Pi} (\phi(W)) |_{17} = \exp \left( \sum_{c(\vec{u}) = v :} \sum_{\phi(W) \models \text{NumDefPr}(c(\vec{u}), m)} \ln \left( \frac{1}{m} \right) \right)
\]

It’s easy to see that \( W''_{\Pi} (\phi(W)) = W''_{\Pi} (\phi(W)) |_{11} \times W''_{\Pi} (\phi(W)) |_{14} \times W''_{\Pi} (\phi(W)) |_{17} = \mu_{\Pi}(W) \). We already proved that for any possible world \( W \) of \( \Pi \), \( \phi(W) \) is a probabilistic stable model of \( \Pi' \). Then to prove \( \mu_{\Pi}(W) = P_W(\phi(W)) \), it is sufficient to prove for any probabilistic stable model \( I \) of \( \Pi' \), \( I|\sigma \) is a possible world of \( \Pi \) and \( I = \phi(I|\sigma) \) (which will be
proven in the next part). Indeed, if we proved this, we know φ(W) and W are 1-1 correspondent, thus \( P_{\Pi'}^{r}(\phi(W)) = \mu_{\Pi}(W) \).

Since \( \phi(W) \in SM'[\Pi'] \), by Lemma 4, \( P_{\Pi'}^{r}(\phi(W)) = P_{\Pi'}^{r}(\phi(W)) = \mu_{\Pi}(W) \).

(b) Since \( \Pi \) is consistent, there exists a possible world \( W' \) of \( \Pi \) with a non-zero probability. It’s proved that \( \phi(W') \) is a probabilistic stable model of \( \Pi' \) and \( \phi(W') \) satisfies \( \Pi'_{\text{hard}} \). So \( SM'[\Pi'] \) is not empty. Let \( I \) be a probabilistic stable model of \( \Pi' \), by Lemma 4, \( I \models \Pi'_{\text{hard}} \). Besides, since \( \Pi' \setminus \Pi'_{\text{hard}} \) is a set of rules of the form \( w \rightharpoonup F \), by Lemma 3, \( I \) is a stable model of \( \Pi'_{\text{hard}} \). Thus \( I \) is a stable model of \( \Pi'_{\text{hard}} \).

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

- \( I \) is a stable model of \( \tau(\Pi) \) relative to \( \sigma \). Thus \( I|_{\sigma} \) is a stable model of \( \tau(\Pi) \), which means \( I|_{\sigma} \) is a possible world of \( \Pi \).
- \( I \) is a stable model of \( \Pi_{\text{AUX}} \) relative to \( \sigma_{3} \). So \( I \models \text{Comp}[\Pi_{\text{AUX}}; \sigma_{3}] \).

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 \( \phi \). Take any random selection rule \([r] \) \( \text{random}(c(\vec{u}) : \{X : p(X)\}) \leftarrow \text{Body} \), since \( I \) satisfies formula (27),
  - \( I \models \text{Poss}_{r}(c(\vec{u}) = v) \)
  - \( I \models \text{Body} \land p(v) \land \neg\text{Intervene}(c(\vec{u})) \)
  - \( W \models \text{Body} \land p(v) \land \neg\text{Intervene}(c(\vec{u})) \)
  - \( c(\vec{u}) = v \) is possible in \( W \) due to \( r \).
- Let’s consider condition (2) in the definition of \( \phi \). Take any pr-atom \( pr_{r}(c(\vec{u}) = v_{i} \mid C_{i}) = p_{i} \) in \( \Pi \), since \( I \) satisfies formula (31),
  - \( I \models \text{PossWithAssPr}_{r,C_{i}}(c(\vec{u}) = v_{i}) \)
  - \( I \models \text{Poss}_{r}(c(\vec{u}) = v_{i}) \land C_{i} \)
  - \( c(\vec{u}) = v_{i} \) is possible in \( W \) due to \( r \) and \( W \models C_{i} \)
  - this pr-atom is applied in \( W \)
  - Thus condition (2) is satisfied by \( I \).
- Let’s consider condition (3) in the definition of \( \phi \). Take any pr-atom \( pr_{r}(c(\vec{u}) = v_{i} \mid C_{i}) = p_{i} \) in \( \Pi \), since \( I \) satisfies formula (32),
  - \( I \models \text{AssPr}_{r,C_{i}}(c(\vec{u}) = v_{i}) \)
  - \( I \models \text{PossWithAssPr}_{r,C_{i}}(c(\vec{u}) = v_{i}) \land c(\vec{u}) = v_{i} \)
  - 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 \( \phi \). Since \( I \) satisfies formula (34),
  - \( I \models \text{PossWithAssPr}(c(\vec{u}) = v_{i}) \)
  - there exist a \( r \) and \( j \) such that \( I \models \text{PossWithAssPr}_{r,C_{j}}(c(\vec{u}) = v_{i}) \)
  - \( v_{i} \in AV_{W}(c(\vec{u})) \)
  - Thus condition (4) is satisfied by \( I \).
- Let’s consider condition (5) in the definition of \( \phi \). Since \( I \) satisfies formula (28),
  - \( I \models \text{PossWithDefPr}(c(\vec{u}) = v) \)
iff
- \( I \models \neg PossWithAssPr(c(\bar{u}) = v) \)
- there exists a random selection rule \([v] \text{ random}(c(\bar{u}) : \{X : p(X)\}) \leftarrow Body\), such that \( I \models Poss_r(c(\bar{u}) = v) \)
  iff (by condition (4) and (1))
- \( v \notin AV_W(c(\bar{u})) \)
- \( c(\bar{u}) = v \) is possible in \( W \)

Thus condition (5) is satisfied by \( I \).

• Let's consider condition (6) in the definition of \( \phi \). Since \( I \) satisfies formula (29),
- \( I \models NumDefPr(c(\bar{u}), x) \)
  iff
- \( x = \#\text{count}\{y : PossWithDefPr(c(\bar{u}) = y)\} \)
- there exists a \( c(\bar{u}) = z \) such that \( I \models c(\bar{u}) = z \land PossWithDefPr(c(\bar{u}) = z) \)
  iff
- there exist exactly \( x \) different values \( v \) such that
  * \( I \models PossWithDefPr(c(\bar{u}) = v) \)
  * for one of such \( v, I \models c(\bar{u}) = v \)
  iff (by condition (5), and since \( c(\bar{u}) = v \) belongs to \( \sigma \))
- there exist exactly \( x \) different values \( v \) such that
  * \( c(\bar{u}) = v \) is possible in \( W \)
  * \( v \notin AV_W(c(\bar{u})) \)
  * for one of such \( v, W \models c(\bar{u}) = v \)

Thus condition (6) is satisfied by \( I \).

• Let's consider condition (7) in the definition of \( \phi \). Since \( I \) satisfies formula (35),
- \( I \models RemPr(c(\bar{u}), x) \)
  iff
- there exists a \( v \) such that \( \phi(W) \models c(\bar{u}) = v \land PossWithDefPr(c(\bar{u}) = v) \)
- there exists a random selection rule (5) along with all pr-atoms (30) associated with it such that
  * \( I \models Body \)
  * \( y = \sum_{pr_r(c(\bar{u}) = v) \in \text{Body}} \frac{p}{\mu} \)
  * \( x = 1 - y \)
  iff (by condition (5) and (2), and since \( c(\bar{u}) = v \) belongs to \( \sigma \))
- there exists a \( v \) such that
  * \( c(\bar{u}) = v \) is possible in \( W \)
  * \( v \notin AV_W(c(\bar{u})) \)
  * \( W \models c(\bar{u}) = v \)
  - \( x = 1 - \sum_{v' \in AV_W(c(\bar{u}))} PossWithAssPr(W, c(\bar{u}) = v') \)

Thus condition (7) is satisfied by \( I \).

• Let's consider condition (8) in the definition of \( \phi \). Since \( I \) satisfies formula (36),
- \( I \models TotalDefPr(c(\bar{u}), x) \)
  iff
- \( I \models RemPr(c(\bar{u}), x) \)
- \( x > 0 \)

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 \( plog2pmln(\Pi), I|\sigma \) is a possible world of \( \Pi \) 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 \( \Pi' \), \( P_{\Pi'}(I) > 0 \). Then \( \mu_{\Pi}(I|\sigma) = P_{\Pi'}(I) > 0 \).
**Full Translation of Monty Hall**

For the P-log program II in Example 2, we showed a part of the translated LP\textsubscript{MLN} program, plog2lpmln(Π), in Example 3. The full version of plog2lpmln(Π) is as follows: \((d \in \{1, 2, 3, 4\})\)

```
// *** τ(Π) ***
α: ¬CanOpen(d) ← Selected = d
α: ¬CanOpen(d) ← Prize = d
α: CanOpen(d) ← not ¬CanOpen(d)

α: ← CanOpen(d), ¬CanOpen(d)
α: ← Prize = d, Prize = d_1, d_1 ≠ d_2
α: ← Selected = d, Selected = d_2, d_2 ≠ d_2
α: ← Open = d, Open = d_2, d_1 ≠ d_2

α: Prize = 1; Prize = 2; Prize = 3; Prize = 4 ← not Intervene(Prize)
α: Selected = 1; Selected = 2; Selected = 3; Selected = 4 ← not Intervene(Selected)
α: Open = 1; Open = 2; Open = 3; Open = 4 ← not Intervene(Open)
α: ← Open = d, not CanOpen(d), not Intervene(Open)

α: Obs(Selected = 1)
α: ← Obs(Selected = 1), not Selected = 1
α: Obs(Open = 2)
α: ← Obs(Open = 2), not Open = 2
α: Obs(Prize ≠ 2)
α: ← Obs(Prize ≠ 2), Prize = 2

// *** Possible Atoms ***
α: Poss(Prize = d) ← not Intervene(Prize)
α: Poss(Selected = d) ← not Intervene(Selected)
α: Poss(Open = d) ← CanOpen(d), not Intervene(Open)

// *** Assigned Probability ***
α: PossWithAssPr(Prize = 1) ← Poss(Prize = 1)
α: Poss(Prize = 1) ← Prize = 1, PossWithAssPr(Prize = 1)
ln (0.3) : ⊥ ← not AssPr(Prize = 1)

α: PossWithAssPr(Prize = 3) ← Poss(Prize = 3)
α: Poss(Prize = 3) ← Prize = 3, PossWithAssPr(Prize = 3)
ln (0.2) : ⊥ ← not AssPr(Prize = 3)

// *** Denominator for Default Probability ***
α: PossWithDefPr(Prize = d) ← Poss(Prize = d), not PossWithAssPr(Prize = d)
α: PossWithDefPr(Selected = d) ← Poss(Selected = d), not PossWithAssPr(Selected = d)
α: PossWithDefPr(Open = d) ← Poss(Open = d), not PossWithAssPr(Open = d)

α: NumDefPr(Prize, x) ← Prize = d, PossWithDefPr(Prize = d), x = \#count \{ y : PossWithDefPr(Prize = y) \}
α: NumDefPr(Selected, x) ← Selected = d, PossWithDefPr(Selected = d), x = \#count \{ y : PossWithDefPr(Selected = y) \}
α: NumDefPr(Open, x) ← Open = d, PossWithDefPr(Open = d), x = \#count \{ y : PossWithDefPr(Open = y) \}
ln (\frac{m}{n}) : ← not NumDefPr(c, m) \hspace{1cm} \forall c \in \{\text{Prize, Selected, Open}\}, m \in \{2, 3, 4\}

// *** Numerator for Default Probability ***
α: RemPr(Prize, 1 - x) ← Prize = d, PossWithDefPr(Prize = d), x = \#sum \{0.3 : PossWithAssPr(Prize = 1); 0.2 : PossWithAssPr(Prize = 3)\}
α: TotalDefPr(Prize, x) ← RemPr(Prize, x), x > 0
ln (x) : ⊥ ← not TotalDefPr(Prize, x)
α: ⊥ ← RemPr(Prize, x), x ≤ 0
```
The further translated ASP with Weak Constraints (WC) encoding is as follows:

```plaintext
% *** Declaration Part ***
door(1..4).

% *** \tau(\Pi) ***
canOpen, (D, f) ← selected(D).
canOpen, (D, f) ← prize(D).
canOpen, (D, t) ← not canOpen, (D, f), door(D).

← canOpen, (D, t), canOpen, (D, f).
← prize(D_1), prize(D_2), D_1 ≠ D_2.
← selected(D_1), selected(D_2), D_1 ≠ D_2.
← open(D_1), open(D_2), D_1 ≠ D_2.

\{prize(D) : door(D)\} 1 ← not intervene(prize).
\{selected(D) : door(D)\} 1 ← not intervene(selected).
\{open(D) : door(D)\} 1 ← not intervene(open).

obs(selected, 1).
← obs(selected, 1), not selected(1).
obs(open, 2).
← obs(open, 2), not open(2).

obs(prize, 2).
← obs(prize, 2), prize(2).

% *** Possible Atoms ***
poss(prize, D) ← not intervene(prize).
poss(selected, D) ← not intervene(selected).
poss(open, D) ← canOpen(D), not intervene(open).

% *** Assigned Probability ***
possWithAssPr(prize, 1) ← poss(prize, 1).
assPr(prize, 1) ← prize(1), possWithAssPr(prize, 1).

possWithAssPr(prize, 3) ← poss(prize, 3).
assPr(prize, 3) ← prize(3), possWithAssPr(prize, 3).

% *** Denominator for Default Probability ***
possWithDefPr(prize, D) ← poss(prize, D), not possWithAssPr(prize, D).
possWithDefPr(selected, D) ← poss(selected, D), not possWithAssPr(selected, D).
possWithDefPr(open, D) ← poss(open, D), not possWithAssPr(open, D).

numDefPr(prize, X) ← prize(D), possWithDefPr(prize, D), X = \#count{Y : possWithDefPr(prize, Y)}.
numDefPr(selected, X) ← selected(D), possWithDefPr(selected, D), X = \#count{Y : possWithDefPr(selected, Y)}.
numDefPr(open, X) ← open(D), possWithDefPr(open, D), X = \#count{Y : possWithDefPr(open, Y)}.

% *** Numerator for Default Probability ***
remPr(prize, Y) ← prize(D), possWithDefPr(prize, D), X = \#sum{0.3 : possWithAssPr(prize, 1); 0.2 : possWithAssPr(prize, 3}, Y = 1 - X.
totalDefPr(prize, X) ← remPr(prize, X), X > 0.
← remPr(prize, X), X ≤ 0.

% *** Weak Constraints ***
%note that if we remove this part, we can get all stable models of this program, not just the optimal ones
:= assPr(prize, 1). [~ln(0.3)]
:= assPr(prize, 3). [~ln(0.2)]
:= numDefPr(C, M). [~ln(1/M)]
:= totalDefPr(prize, X). [~ln(X)]
```
For both translated \( \text{LP}^{\text{MLN}} \) and WC programs, there are 3 stable models that satisfy all hard rules. The intersection of 3 stable models is shown below, followed by the remaining part of these stable models:

**Intersection of 3 stable models**: (following the syntax in \( \text{LP}^{\text{MLN}} \) encoding)

\[
\{ \text{Obs}(\text{Selected} = 1), \text{Obs}(\text{Open} = 2), \text{Obs}(\text{Price} \neq 2), \\
\text{Selected} = 1, \text{Open} = 2, \\
\text{CanOpen}(1) = \text{t}, \text{CanOpen}(2) = \text{t}, \\
\text{Poss}(\text{Price} = 1), \text{Poss}(\text{Price} = 2), \text{Poss}(\text{Price} = 3), \text{Poss}(\text{Price} = 4), \\
\text{Poss}(\text{Selected} = 1), \text{Poss}(\text{Selected} = 2), \text{Poss}(\text{Selected} = 3), \text{Poss}(\text{Selected} = 4), \\
\text{Poss}(\text{Open} = 2), \\
\text{PossWithAssPr}(\text{Price} = 1), \text{PossWithAssPr}(\text{Price} = 3), \\
\text{PossWithDefPr}(\text{Price} = 2), \text{PossWithDefPr}(\text{Price} = 4), \\
\text{PossWithDefPr}(\text{Selected} = 1), \text{PossWithDefPr}(\text{Selected} = 2), \text{PossWithDefPr}(\text{Selected} = 3), \text{PossWithDefPr}(\text{Selected} = 4), \\
\text{PossWithDefPr}(\text{Open} = 2), \\
\text{NumDefPr}(\text{Selected}, 4) \}
\]

\[
I_1 = \{ \text{Price} = 1, \text{CanOpen}(3) = \text{t}, \text{CanOpen}(4) = \text{t}, \\
\text{AssPr}(\text{Price} = 1), \text{NumDefPr}(\text{Open}, 3), \\
\text{Poss}(\text{Open} = 3), \text{Poss}(\text{Open} = 4), \\
\text{PossWithDefPr}(\text{Open} = 3), \text{PossWithDefPr}(\text{Open} = 4) \}
\]

\[
I_2 = \{ \text{Price} = 3, \text{CanOpen}(3) = \text{f}, \text{CanOpen}(4) = \text{t}, \\
\text{AssPr}(\text{Price} = 3), \text{NumDefPr}(\text{Open}, 2), \\
\text{Poss}(\text{Open} = 4), \\
\text{PossWithDefPr}(\text{Open} = 4) \}
\]

\[
I_3 = \{ \text{Price} = 4, \text{CanOpen}(3) = \text{t}, \text{CanOpen}(4) = \text{f}, \\
\text{NumDefPr}(\text{Price}, 2), \text{TotalDefPr}(\text{Price}, 0.5), \text{NumDefPr}(\text{Open}, 2), \\
\text{Poss}(\text{Open} = 3), \\
\text{PossWithDefPr}(\text{Open} = 3), \\
\text{RemPr}(\text{Price}, 0.5) \}
\]

The unnormalized weight \( \omega(I_i) \) of each stable model \( I_i \) is shown below:

\[
\omega(I_1) = \omega(\text{NumDefPr}(\text{Selected}, 4)) \times \omega(\text{AssPr}(\text{Price} = 1)) \times \omega(\text{NumDefPr}(\text{Open}, 3))
\]
\[
= \frac{1}{4} \times 0.3 \times 0.3 = \frac{1}{40}
\]

\[
\omega(I_2) = \omega(\text{NumDefPr}(\text{Selected}, 4)) \times \omega(\text{AssPr}(\text{Price} = 3)) \times \omega(\text{NumDefPr}(\text{Open}, 2))
\]
\[
= \frac{1}{4} \times 0.2 \times \frac{1}{2} = \frac{1}{32}
\]

\[
\omega(I_3) = \omega(\text{NumDefPr}(\text{Selected}, 4)) \times \omega(\text{NumDefPr}(\text{Price}, 2)) \times \omega(\text{TotalDefPr}(\text{Price}, 0.5)) \times \omega(\text{NumDefPr}(\text{Open}, 2))
\]
\[
= \frac{1}{4} \times \frac{1}{2} \times 0.5 \times \frac{1}{2} = \frac{1}{32}
\]