On the Relationship between P-log and LP$^{\text{MLN}}$*

Evgenii Balai and Michael Gelfond

Texas Tech University, Lubbock Texas
{evgenii.balai, michael.gelfond}@ttu.edu

Abstract. The paper investigates the relationship between knowledge representation languages P-log [2] and LP$^{\text{MLN}}$ [11] designed for representing and reasoning with logic and probability. We give a translation from an important subset of LP$^{\text{MLN}}$ to P-log which preserves probabilistic functions defined by LP$^{\text{MLN}}$ programs and complements recent research [12] by the authors of LP$^{\text{MLN}}$ where they give a translation from a subset of P-log to their language. This work sheds light on the different ways to treat inconsistency in both languages.

1 Introduction

Combining logic and probability has been one of the most important directions of artificial intelligence research in recent years. Many different languages and formalisms have been developed to represent and reason about both probabilistic and logical arguments, such as ProbLog [5, 6], PRISM [16, 17], LPADs [19], CP-Logic [18], MLN [15], and others.

In this paper we focus on two such languages, P-log and LP$^{\text{MLN}}$. They are distinguished from other mentioned alternatives by their common logic base, Answer Set Prolog (ASP) [8], a logical formalism modeling beliefs of a rational agent. ASP is powerful enough to naturally represent defaults, non-monotonically update the knowledge base with new information, define relations recursively, reason about causal effects of actions, etc. The language serves as the foundation of the so called Answer Set Programming paradigm [13, 14] and has been used in a large number of applications [4].

An agent associated with an ASP knowledge base reasons about three degrees of belief – he can believe that p is $true$, believe that p is $false$, or remain uncommitted about his belief in p. In the latter case the truth of p remains $unknown$.

An extension of ASP, called P-log, allows the reasoner to express and reason with finer, numerically expressed, gradation of the strength of his beliefs. In other words, it preserves the power of ASP and, in addition, allows an agent to do sophisticated probabilistic reasoning.

The main goal of the P-log designers was to provide the language and reasoning mechanism which can be used for clear and transparent modeling of knowledge involving logical and probabilistic arguments. There are a number of non-trivial scenarios formalized in [3]. More computationally challenging scenarios, including probabilistic

*This work was sponsored by NASA under the grant CertWare ABSA: Argument Based Safety Assurance, Grant #: NRA NNH11ZEA001N-SSAT2

planning and diagnosis, can be found in [20], where their P-log representations and performance analysis are given.

A new version of P-log , introduced in [9], replaces ASP by its extension CR-Prolog [1], which expands logical power of ASP (and hence the original P-log) by allowing so called consistency restoring rules (cr-rules) used for restoring consistency of the program by a certain form of abductive reasoning.

Despite the presence of cr-rules, the underlying philosophy of P-log requires the corresponding knowledge base to be consistent. A rational reasoner is assumed to trust its rules and refuses to deal with a knowledge base containing statements p and $\neg p$. Possible means to ensure consistency of the program should be supplied by a knowledge engineer. This is natural from a theoretical goal of the authors but is also important in many practical applications, for example, where inconsistency of the knowledge base may be a sign of some errors in its design and therefore should be addressed by making the necessary changes.

The language LP$^{\text{MLN}}$, introduced in [11], is based on a different philosophy. Its first goal seems to be similar to that of P-log – it is supposed to provide means for combining ASP based reasoning with reasoning about probability. But, in addition, the new language is aimed at providing a powerful (though somewhat less predictable) way of resolving inconsistencies which may appear in LP$^{\text{MLN}}$ programs due to mechanical combination of different knowledge bases, designer mistakes, or some other reasons. The design of the language was influenced by Markov Logic Networks [15] and seems to be practically independent from P-log. As a result, the relationship between these two languages with seemingly similar goals remains unclear. This paper is a step in remedying this situation. In particular, we give a translation from an important subset of LP$^{\text{MLN}}$ to P-log which preserves probabilistic functions defined by LP$^{\text{MLN}}$ programs. The work complements resent research[12] by Lee and Wang in which the authors give a translation from a subset of P-log to LP$^{\text{MLN}}$.

The rest of this paper is organized as follows. In section 2 we define the subset of P-log used in this paper. In section 3 we briefly describe the syntax and semantics of LP$^{\text{MLN}}$. In section 4 we describe a translation from LP$^{\text{MLN}}$ to P-log and define the correspondence between LP$^{\text{MLN}}$ programs and their P-log translations precisely. Section 5 uses the results from sections 4 to describe certain properties of probabilities defined by LP$^{\text{MLN}}$ programs. Section 6 concludes the paper by summarizing the obtained results and future work.

2 Language P-log

In this section we introduce a simplified version of P-log with consistency restoring rules from [9] which is sufficient to define the translation from LP$^{\text{MLN}}$. We do so by considering a simple domain consisting of two adjacent rooms, r_1 and r_2, and a robot initially located in r_1. We present a P-log program, Π_0, modeling direct effects of the action $move$ of the robot attempting to enter the second room. The program is presented to illustrate the language constructs so we ignore concerns about its generality, elaboration tolerance, etc.

We start with declarations of objects and functions of the domain. In P-log such functions are usually referred to as *attributes*, while expressions of the form $f(\bar{x})$, where f is an attribute, are called *attribute terms*. We need the sort

$$step = \{0, 1\}$$

where 0 and 1 denote time-steps before and after the execution of the action respectively. We also need the sort

$$room = \{r_1, r_2\}$$

and attributes

$$move, broken : boolean$$

$$loc \ : \ step \rightarrow room$$

Here $move$ is true iff at step 0 the robot has attempted to move to room r_2; $broken$ holds if the robot has been broken and hence may exhibit some non-deterministic behavior; loc gives the location of the robot at a given step. Declarations of P-log are followed by the program rules. In our case we will have rules

$$loc(0) = r_1 \tag{1}$$

$$move \tag{2}$$

indicating that at step 0 the robot located in room r_1 attempted to move to room r_2. Here $move$ is a shorthand for $move = true$. We use this convention for all boolean functions: $f(\bar{x}) = true$ and $f(\bar{x}) = false$ are written as $f(\bar{x})$ and $\neg f(\bar{x})$ respectively.

As expected the effect of action $move$ for the well-functioning robot is given by the rule:

$$loc(1) = r_2 \leftarrow not\ broken, move. \tag{3}$$

If the robot is malfunctioning however we need to state that the effect of $move$ is random – the robot can still successfully move to room r_2 or to stay in room r_1. In P-log this is expressed by the following *random selection rule* r

$$[r]\ random(loc(1)) \leftarrow broken, move \tag{4}$$

which says that if the malfunctioning robot will attempt to move to room r_2 then, in the resulting state, attribute term $loc(1)$ will randomly take a value from the range of loc. The rules of the program described so far can be easily translated into regular ASP rules – we simply need to replace $random(loc(1))$ in the last rule by $(loc(1) = r_1\ or\ loc(1) = r_2)$, replace atoms of the form $f(\bar{x}) = y$ by $f(\bar{x}, y)$ and, for every term $f(\bar{x})$, add a constraint $\{\leftarrow f(\bar{x}) = y_1, f(\bar{x}) = y_2, y_1 \neq y_2\}$. In general, an atom of the form $random(f(\bar{x}))$ is replaced by the disjunction $f(\bar{x}) = y_1\ or\ \ldots\ or\ f(\bar{x}) = y_k$ where $\{y_1, \ldots, y_k\}$ is the range of f.

Answer sets of the translation of a P-log program Π into ASP are viewed as *possible worlds* of the probabilistic model defined by Π. It is easy to see that program Π^0 consisting of rules (1)–(4) has one such possible world $W_0 = \{move, loc(0) =$

$r_1, loc(1) = r_2\}$[1] . Program $\Pi^1 = \Pi^0 \cup \{broken\}$ will have two possible worlds, $W_1 = \{broken, move, loc(0) = r_1, loc(1) = r_2\}$ and $W_2 = \{broken, move, loc(0) = r_1, loc(1) = r_1\}$. In the case of multiple possible worlds we need some device allowing to specify the numeric probabilities of possible values of random attributes. This is expressed in P-log through *causal probability statements*, or, simply, *pr-atoms*. A pr-atom takes the form

$$pr_r(f(\bar{x}) = y) = v$$

where $f(\bar{x})$ is a random attribute, y is a value from the range of f, and $v \in [0, 1]$ is the probability of y to be selected as the value of $f(\bar{x})$ as the result of firing random selection rule r. In case of Π^1 such pr-atoms may look as, say,

$$pr_r(loc(1) = r_1) = 0.3$$

and

$$pr_r(loc(1) = r_2) = 0.7$$

Unnormalized probabilistic measure of a possible world W is defined as the product of probabilities of the random atoms $f_1(\bar{x}_1) = y_1, \ldots, f_k(\bar{x}_k) = y_k$ from W. These probabilities are obtained from the corresponding pr-atoms. Normalized probabilistic measures and probability function on the sets of possible worlds and on the literals of the language are defined as usual. Let P_0 and P_1 be the probability functions defined by Π^0 and Π^1 respectively. W_0 has no random atoms, the empty product is 1, and hence the probabilistic measure of W_0 and $P_0(loc(1) = r_1)$ are both equal to 1. The probabilistic measures of W_1 and W_2 are 0.7 and 0.3 respectively and hence $P_1(loc(1) = r_1) = 0.3$.

As mentioned in the introduction, a P-log program can be inconsistent. For instance, program $\Pi^2 = \Pi^0 \cup \{loc(1) = r_1\}$ has no possible worlds. To avoid this particular inconsistency the program designer can expand Π^0 by a cr-rule:

$$broken \stackrel{+}{\leftarrow} . \tag{5}$$

which allows to restore inconsistency of Π^2 by assuming that the robot is broken. Since the original program Π^0 is consistent, the resulting program, Π^0_{new} will define the same probabilistic model as Π^0. The program Π^2_{new}, consisting of Π^0_{new} and the fact $\{loc(1) = r_1\}$, unlike the program Π^2, will be consistent and have one possible world, W_2. The extension of Π^0 by a new information changed the probability of the robot being in room r_2 after execution of $move$ from 1 to 0.

3 Language LP$^{\text{MLN}}$

In this section we give a brief summary of LP$^{\text{MLN}}$ ([11]). We limit out attention to ground programs whose rules contain no double default negation $not\ not$ and no disjunction. To the best of our knowledge, no example in the literature demonstrating the use of

[1]for convenience we will often identify original P-log literals with corresponding ASP ones (e.g, we will sometimes write $loc(1) = r_1$ in place of $loc(1, r_1)$)

LP$^{\text{MLN}}$ for formalization of knowledge uses these constructs. As usual, we may use rules with variables viewed as shorthands for the sets of their ground instances. A program of the language is a finite set of LP$^{\text{MLN}}$ *rules* – ground ASP rules preceded by a *weight*: symbol α or a real number. Rules of the first type are called *hard* while rules of the second are referred to as *soft*. Despite their name the hard rules are not really "hard". Their behavior is reminiscent of that of defaults. According to the semantics of the language the reasoner associated with a program constructs possible worlds with non-zero probability by trying to satisfy as many hard rules as possible. The satisfiability requirement for the soft rules and the use of their weights for assigning the probability measure to possible worlds of M are more subtle. In what follows we give the necessary definitions and illustrate them by an example of LP$^{\text{MLN}}$ program. Sometimes we abuse the notation and identify an LP$^{\text{MLN}}$ program M with its *ASP counterpart* obtained from M by dropping the weights of its rules. Stable models of such a counterpart will be referred to as *ASP models* of M. By M_I we denote the set of rules of M which are satised by an interpretation I of M. An interpretation W is a *possible world* of M if it is a ASP model of M_W. We will say that a possible world W *is supported by* M_W. As usual, by Ω_M we denote the set of all possible worlds of M. *Unnormalized measure* of a possible world $W \in \Omega_M$ (denoted by $w_M(W)$) is exp^γ where γ is the sum of weights of all rules of M satisfied by W. Note that, in case M contains rules with α-weights satisfied by W, $w_M(W)$ is not a numerical value and should be understood as a symbolic expression. The *probability function*, P_M, *defined by program* M is

$$P_M(W) = \lim_{\alpha \to \infty} \frac{w_M(W)}{\sum_{V \in \Omega_M} w_M(V)}$$

It is easy to check that P_M maps possible worlds of M into the interval $[0, 1]$ and satisfies standard axioms of probability.

As expected, *the probabilistic model* defined by M consists of Ω_M and P_M.

Let us now use LP$^{\text{MLN}}$ to formalize the stories from the previous section. Program M^0 will capture the first such story corresponding to P-log program Π^0. It clearly should contain rules (1) – (3) of Π^0. In addition, for every attribute f it must include a constraint

$$\leftarrow f(X) = Y_1, f(X) = Y_2, Y_1 \neq Y_2 \tag{6}$$

which is hidden in P-log semantics of Π^0. All these rules, however, should be supplied with some weights. Since we strongly believe that the rules are correct, we would like to preserve as many of them as possible. Hence, we view them as hard. LP$^{\text{MLN}}$ does not have a direct analog of rule (4) but, it seems natural to represent it by two rules:

$$ln(0.3) \ : \ loc(1) = r_1 \leftarrow broken, move \tag{7}$$

and

$$ln(0.7) \ : \ loc(1) = r_2 \leftarrow broken, move \tag{8}$$

where the logarithms are added to the probabilities to cancel the exponentiation from the definition of unnormalized measure w_M. In addition, the hard rule

$$\alpha : \ \leftarrow not\ loc(1) = r_1, not\ loc(1) = r_2 \tag{9}$$

is added to force $loc(1)$ to take a value (in P-log this is guaranteed by the semantics of disjunction). This concludes construction of M^0. It is worth noting that M_0 is similar to the program obtained from Π_0 by a general translation from P-log to LP$^{\text{MLN}}$ described in [12].

We will show that there is a simple relationship between probabilistic models defined by Π^0 and M^0. The possible worlds of Π^0 correspond to the possible worlds M^0 with non-zero probability (also called *probabilistic stable models* of M^0 in [11]). Moreover, probability functions P_{M^0} and P_{Π^0} coincide on probabilistic stable models of M^0.

Let us first notice that $W_0 = \{move, loc(0) = r_1, loc(1) = r_2\}$ is an ASP model of $M^0_{W_0}$ and hence is a possible world of M^0. The probability of W_0 is 1. Clearly, W_0 is the only possible world of M^0 satisfying all its hard rules. M^0 however has other possible worlds. For instance, $V = \{move\}$ satisfies all the rules of $M^0 \setminus \{(1), (3), (9)\}$ and is the stable model of this program. Therefore, it is a possible world of M^0. It is easy to check however that V is not a probabilistic stable model of M^0. In fact, this is a consequence of a general result in [11] which says that if there is a possible world of LP$^{\text{MLN}}$ program M which satisfies all its hard rules then every probabilistic stable model of M also satisfies them. The result clearly implies that W_0 is the only probabilistic stable model of M^0.

The program $M^1 = M^0 \cup \{\alpha\ :\ broken\}$ is again similar to Π^1. It has two probabilistic stable models, W_1 and W_2 with probabilities equal to 0.7 and 0.3 respectively. As before, there are other possible worlds but none of them satisfies all the hard rules of M^1 and hence they have probability 0.

A more serious difference can be observed however between the program Π^2 and the new program M^2 obtained from M^0 by adding the rule

$$\alpha : loc(1) = r_1 \tag{10}$$

Since the rules of Π^2 are strict and, therefore, should be satisfied by possible worlds, the program Π^2 is inconsistent. M^2 however does not have such a restriction. It will have three probabilistic stable models. The first one is an ASP model of $M^2 \setminus \{(2)\}$. It resolves contradiction by assuming that the robot failed to move. The second is an ASP model of $M^2 \setminus \{(3)\}$. The contradiction is removed by abandoning the causal law (3). Another possible explanation may given by an ASP model of $M^2 \setminus \{(10)\}$ which assumes that our observation of the robot being in r_1 after the execution of $move$ is incorrect. This seems to be a reasonable answer.

M^2 also has other possible worlds, for instance, $W^* = \{loc(1) = r_1, loc(1) = r_2, move, loc(0) = r_1\}$. W^* however does not satisfy two ground instances of the rule (6):

$$\alpha :\leftarrow loc(1) = r_1, loc(1) = r_2, r_1 \neq r_2$$

$$\alpha :\leftarrow loc(1) = r_2, loc(1) = r_1, r_2 \neq r_1$$

and therefore, has probability 0. This follows from a simple generalization of the result from ([11]) which says that if there is a possible world of LP$^{\text{MLN}}$ program M which satisfies n hard rules then every possible world satisfying less than n hard rules has probability 0. Note, however, that if the rule (6) in M^2 were replaced by a seemingly equivalent rule

$$\alpha :\leftarrow f(X) = Y_1, f(X) = Y_2, Y_1 < Y_2 \tag{11}$$

(with a lexicographic meaning of $<$), the resulting program M^* would have an additional probabilistic stable model W^*. It is an ASP model of the program obtained from M^* by removing the following ground instance of (11):

$$\alpha :\leftarrow loc(1) = r_1, loc(1) = r_2, r_1 < r_2 \tag{12}$$

Though technically correct the result looks somewhat counterintuitive since the robot cannot occupy both rooms at the same time. However, we can extend M^* with another copy of (12) to increase our confidence in it. The new program will have the same probabilistic stable models as M^2.

Finally, to model the effect of consistency restoring rule (5), we extend M^0 with the rule

$$w : broken \tag{13}$$

where w is a very large negative weight. The resulting program M^0_{new} will have 3 probabilistic stable models, in two of which the robot is believed to be broken, however the probabilities of both of them are very low. This behavior is quite different (and, in some sense, less elegant) from the similar case in P-log where the cr-rule (5) was not used since the program Π^0 is consistent. Similarly to Π^2_{new}, the program M^2_{new} consisting of M^0_{new} and the fact $\{loc(1) = r_1\}$ has exactly one probabilistic stable model W_2 (which satisfies all the hard rules of the program). As in P-log, the semantics of LP^{MLN} allow updating of the probability of the robot to be in room r_2 at step 1 from 0 to 1 by adding new information. However, unlike in P-log, extending the original program M^0 with a soft counterpart (13) of the cr-rule (5) leads to introducing new probabilistic stable models of the program with negligible probabilities.

4 From LP^{MLN} to P-log

In this section we state the main result of this paper: establishing a relationship between LP^{MLN} and P-log programs.

First we need a definition. Let M be an LP^{MLN} program and $At(M)$ be the set of atoms in M.

Definition 1 (Counterpart). *A P-log program Π is called a counterpart of M if there exists a bijection ϕ from the set of probabilistic stable models of M to the set of possible worlds of Π such that*

1. *for every probabilistic stable model W of M , if P_M and P_Π are probability functions defined by M and Π respectively, then $P_M(W) = P_\Pi(\phi(W))$*
2. *for every probabilistic stable model W of M, $W \equiv_{At(M)} \phi(W)$, that is, W and $\phi(W)$ coincide on the atoms of M.*

Main Theorem. *For every LP^{MLN} program M (as defined in section 3) there exists its P-log counterpart, $\tau(M)$, which is linear-time constructible.* □

The previous theorem immediately implies the following important corollary:

Corollary 1. *If A is atom of an LP^MLN program M, then*

$$P_M(A) = P_{\tau(M)}(A)$$

where the probabilities $P_M(A)$ and $P_{\tau(M)}(A)$ are defined as the sum of probabilities of possible worlds which contain A of the corresponding program. □

To prove the theorem we need the following Lemma which gives an alternative characterization of probabilistic stable models of M.

Lemma 1. *Let W be an interpretation satisfying n hard rules of M. W is a probabilistic stable model of M if and only if*

1. *W is a possible world of M supported by some $M^0 \subseteq M$;*
2. *no possible world of M supported by some $M^1 \subseteq M$ satisfies more than n hard rules of M.*

□

Proof of the Main Theorem: Here we only provide a construction of $\tau(M)$ given a program M and define a map ϕ from definition 1 (the complete proof of the theorem, including a proof of lemma 1 can be found in Appendix B. A short outline of the proof is given in Appendix A).

We will assume that all atoms in Π are of the form p, where p is an identifier (in the general case, the atoms of the form $p(t_1, \ldots, t_n)$ can be translated into unique identifiers).

In what follows we will construct a P-log program which chooses a subprogram M^0 of M and computes ASP models of M supported by M^0 such that no possible world of M is supported by a program containing more hard rules than M_0. By Lemma 1, they will be probabilistic stable models of M. Appropriate probability atoms will ensure that the corresponding probabilities match.

Let $r_1, \ldots, r_n$ be the enumeration of rules of M. We will refer to i as the *label* of r_i.

The translation $\tau(M)$ is defined as follows:

1. $\tau(M)$ contains
   (a) declarations of the sorts $hard$ and $soft$ – sets of labels of hard and soft rules of M respectively.
   (b) declaration $a : boolean$ for each atom a in the signature of Π.
   (c) declarations of the auxiliary attributes

$$h, b, selected, sat : soft \to boolean$$

$$ab : hard \to boolean$$

   whose meaning will be explained later.

We refer to this part of the translation as the *declaration part* of τ.

2. For every hard rule r_i of the form

$$\alpha : head \leftarrow body \tag{14}$$

$\tau(M)$ contains the rules:

$$head \leftarrow body, not\ ab(i) \tag{15}$$

$$ab(i) \stackrel{+}{\leftarrow} . \tag{16}$$

The auxiliary relation $ab(i)$ says that "rule r_i is abnormal (or not-applicable))". The addition of $not\ ab(i)$ turns the translation (15) of Π's rule (14) into a default rule of P-log. The cr-rule (16), called Contingency Axiom [9], says that, the reasoner may possibly believe $ab(i)$. This possibility, however, may be used only if there is no way to obtain a consistent set of beliefs by using only regular rules of the program. It is commonly used to capture indirect exceptions to defaults [7]. Together, these rules allow to stop the application of a minimal number of the hard rules of M thus avoiding possible inconsistency and conforming to the semantics of such rules in LP$^{\text{MLN}}$.

This completes the translation for programs consisting of hard rules only.

3. For every soft rule r_i of the form

$$w : head \leftarrow body \tag{17}$$

$\tau(M)$ contains the rules:

$$head \leftarrow body, selected(i) \tag{18}$$

$$random(selected(i)) \tag{19}$$

$$\leftarrow \neg selected(i), sat(i) \tag{20}$$

The auxiliary relation $selected(i)$ says that "the rule with label i is selected"; relation $sat(i)$ stands for 'the rule with label i is satisfied'. The addition of $selected(i)$ to the body of the translation (18) of M's rule (17) together with random selection rule (19) allows a reasoner to select soft rules of a candidate subprogram M_0 of M. Constraint (20) is used to ensure that computed models of M_0 satisfy condition 1 from the Lemma 1. Of course, to make this work we need the definition of sat which is given by the following rules:

$$sat(i) \leftarrow b(i), h(i) \tag{21}$$

$$sat(i) \leftarrow not\ b(i) \tag{22}$$

$$b(i) \leftarrow B \tag{23}$$

where B is the body of soft rule r_i, and

$$h(i) \leftarrow l \tag{24}$$

for every literal l in the head of soft rule r_i. As expected, $b(i)$ stands for 'the body of r_i is satisfied' and $h(i)$ for 'the head of r_i is satisfied'.

4. Finally, for every $selected(i)$, $\tau(M)$ contains probability atom:

$$pr(selected(i)) = \frac{e^{w_i}}{1 + e^{w_i}} \tag{25}$$

which says 'the soft rule r_i with weight w_i is selected (that is, added to M^0) with probability $\frac{e^{w_i}}{1+e^{w_i}}$.

It is easy to see that the size of $\tau(M)$ is linear in terms of the size of M. Moreover, $\tau(M)$ is modular, that is, it can be easily extended if new rules are added to M.
The map ϕ is defined is follows:

$$\begin{aligned}\phi(W) = &W \cup \{ab(i) \mid i \in hard, r_i \text{ is not satisfied by } W\}\\ &\cup \{sat(i) \mid i \in soft, r_i \text{ is satisfied by } W\}\\ &\cup \{selected(i) \mid i \in soft, r_i \text{ is not satisfied by } W\}\\ &\cup \{\neg selected(i) \mid i \in soft, r_i \text{ is satisfied by } W\}\\ &\cup \{b(i) \mid i \in soft, \text{ the body of } r_i \text{ is satisfied by W}\}\\ &\cup \{h(i) \mid i \in soft, \text{ the head of } r_i \text{ is satisfied by W}\}\end{aligned}$$

The rest of the proof can be outlined as follows. We first need to show that for every probabilistic stable model W, $\phi(W)$ is a possible world of $\tau(M)$. This can be done by using standard techniques suitable for CR-Prolog programs. After that, we show the bijectivity of ϕ. This step can be split into two parts. Firstly, the surjectivity of $\phi(W)$ follows from the fact that a probabilistic stable model V of M obtained from a possible world W of $\tau(\Pi)$ by dropping all newly introduced literals from W satisfies $\phi(V) = W$. Secondly, the injectivity follows trivially from the definition of ϕ. Finally, the required probability equality from definition 1 follows from the definition of probabilistic functions in both languages. □

The following is an example of the translation.

Example. Consider the following LP$^{\text{MLN}}$ program M from [11]:

$\alpha : concertBooked.$
$\alpha : longDrive \leftarrow concertBooked, not\ cancelled.$
$ln(0.2) : cancelled.$
$ln(0.8) :\leftarrow cancelled.$

The program has two probabilistic stable models (each of which satisfy both its hard rules):

1. $V_1 = \{concertBooked, cancelled\}$
2. $V_2 = \{concertBooked, longDrive\}$

with corresponding probabilities equal to 0.2 and 0.8.

The corresponding translation $\tau(M)$ looks as follows:

% declaration part:
$soft = \{3, 4\}$.
$hard = \{1, 2\}$.
$concertBooked, cancelled, longDrive : boolean.$
$b, h, selected, sat : soft \rightarrow boolean.$
$ab : hard \rightarrow boolean.$
% translation of hard rules:
$concertBooked \leftarrow not\ ab(1).$
$longDrive \leftarrow concertBooked, not\ cancelled, not\ ab(2).$
$ab(R) \overset{+}{\leftarrow} .$
% translation of soft rules:
$cancelled \leftarrow selected(3).$
$\leftarrow cancelled, selected(4).$
$random(selected(R)).$
$\leftarrow \neg selected(R), sat(R).$
% definition of satisfiability:
$sat(R) \leftarrow not\ b(R).$
$sat(R) \leftarrow b(R), h(R).$
$b(3).$
$b(4) \leftarrow cancelled.$
$h(3) \leftarrow cancelled.$
% probability atoms:
$pr(selected(3)) = 0.2/(1 + 0.2).$
$pr(selected(4)) = 0.8/(1 + 0.8).$

The translation $\tau(M)$ has two possible worlds:

1. $U_1 = \{selected(3), \neg selected(4), h(3), cancelled, b(3), b(4), sat(3), concertBooked\}$
2. $U_2 = \{\neg selected(3), selected(4), b(3), longDrive, concertBooked, sat(4)\}$

As expected, on the atoms of M, U_1 and U_2 coincide with the corresponding probabilistic stable models $\{cancelled, concertBooked\}$ and $\{concertBooked, longDrive\}$ of M (more specifically, $U_1 = \phi(V_1)$ and $U_2 = \phi(V_2)$). It can be easily checked that, as promised, $P_{\tau(M)}(U_1) = P_M(V_1) = 0.2$ and $P_{\tau(M)}(U_2) = P_M(V_2) = 0.8$.

5 Probabilities of soft rules in LP$^{\text{MLN}}$

Let M be an LP$^{\text{MLN}}$ program with at least one soft rule r_i of the form $w_i : head \leftarrow body$. The authors of LP$^{\text{MLN}}$ view r_i as an implication and define the probability $P_M(r_i)$ as follows:

$$P_M(r_i) = \sum_{W \in \Omega_M, W \models r_i} P_M(W) \tag{26}$$

Note that replacing Ω_M with the set of all probabilistic stable models of M gives an equivalent definition. We will use the result obtained in the previous section to investigate the relationship between the reasoner's confidence in r_i, i.e, its weight w_i and its probability $P_M(r_i)$.

It seems natural to assume that $P_M(r_i)$ would be proportional to w. However, this is not necessarily the case. Let us consider the following program:

$ln(3):\ a.$
$ln(3):\ \leftarrow a.$
$ln(2):\ b.$

Despite the larger weight, the first rule has smaller probability than the third one (the corresponding probabilities are equal to $1/2$ and $2/3$ respectively).

Informally speaking, this happens because the first rule is inconsistent with the second one, while the third one doesn't have such a restriction.

We next use the results from the previous section to obtain an alternative understanding of the probability $P_M(r_i)$. Let $\tau(M)$ be the counterpart of M described there and ϕ be the bijection from Definition 1. It can be easily seen that r_i is satisfied by a possible world W of M iff $\phi(W)$ contains $selected(i)$. This, together with the first clause of Definition 1 implies that:

$$P_M(r_i) = P_{\tau(M)}(selected(i)) \tag{27}$$

That is, the probability of r_i in M is equal to the probability of $selected(i)$ in P-log program $\tau(M)$. In general, this probability depends on all possible worlds of $\tau(M)$ and their probabilities. However, for some cases it can be determined uniquely by the weight of r_i. This is always the case if $\tau(M)$ belongs to the class of coherent P-log programs, where this probability is equal to the value of the pr-atom $pr(selected(i))$ in (25). This class, and the sufficient conditions for a P-log program to be in it, are given in [3].

For instance, it can be checked that the translation of the program M^3 consisting of soft facts $ln(3) : a$ and $ln(2) : b$ is coherent. Thus, the fact that probability of a is equal to $e^{ln(3)}/(1 + e^{ln(3)}) = 3/4$ can be obtained directly from the corresponding pr-atom (25) of $\tau(M^3)$. Note that, in general, to compute the probability of an atom, we may need to perform fairly complex inference (e.g, compute possible worlds of the program).

6 Conclusion and Future Work

We have defined a linear-time constructible modular translation τ from LP$^{\text{MLN}}$ programs into P-log programs. Non-zero probability possible worlds of an LP$^{\text{MLN}}$ program M coincide with possible worlds of $\tau(M)$ on atoms of M. Moreover, the probabilistic functions defined by M and $\tau(M)$ coincide on atoms M. The work allowed us to better understand both languages, including their treatment of potential inconsistencies, and opened a way to the development of LP$^{\text{MLN}}$ solvers based on P-log inference engines. We also believe that this work, together with the new complementary results from [12], will allow to use the theory developed for one language to discover properties of the other.

Our plans for future work are as follows.

1. We plan to complete the current work on the development of an efficient inference engine for P-log and use the translation τ to turn it into a solver for LP$^{\text{MLN}}$.

2. In the near future, we expect the appearance of LP^{MLN} solver based on the algorithm from Section 3.4 [11]. It will be interesting to use the translatiom from [12] to turn it into P-log solver and compare its performance with that of the one mentioned above.
3. We plan to investigate the possibility of adapting inference methods developed for MLN[10] and LP^{MLN} for improving efficiency of P-log solvers.

Acknowledgments

We would like to thank Joohyung Lee and Yi Wang for useful discussions on the topic.

Bibliography

[1] Marcello Balduccini and Michael Gelfond. Logic programs with consistency-restoring rules. In *International Symposium on Logical Formalization of Commonsense Reasoning, AAAI 2003 Spring Symposium Series*, volume 102. The AAAI Press, 2003.

[2] Chitta Baral, Michael Gelfond, and Nelson Rushton. Probabilistic reasoning with answer sets. In *Logic Programming and Nonmonotonic Reasoning*, pages 21–33. Springer, 2004.

[3] Chitta Baral, Michael Gelfond, and Nelson Rushton. Probabilistic reasoning with answer sets. *Theory and Practice of Logic Programming*, 9(01):57–144, 2009.

[4] Gerhard Brewka, Thomas Eiter, and Mirosław Truszczyński. Answer set programming at a glance. *Communications of the ACM*, 54(12):92–103, 2011.

[5] Luc De Raedt, Angelika Kimmig, and Hannu Toivonen. Problog: A probabilistic prolog and its application in link discovery. In *IJCAI*, volume 7, pages 2462–2467, 2007.

[6] Daan Fierens, Guy Van den Broeck, Joris Renkens, Dimitar Shterionov, Bernd Gutmann, Ingo Thon, Gerda Janssens, and Luc De Raedt. Inference and learning in probabilistic logic programs using weighted boolean formulas. *Theory and Practice of Logic Programming*, 15(03):358–401, 2015.

[7] Michael Gelfond and Yulia Kahl. *Knowledge representation, reasoning, and the design of intelligent agents: The answer-set programming approach*. Cambridge University Press, 2014.

[8] Michael Gelfond and Vladimir Lifschitz. Classical Negation in Logic Programs and Disjunctive Databases. *New Generation Computing*, 9(3/4):365–386, 1991.

[9] Michael Gelfond and Nelson Rushton. Causal and probabilistic reasoning in p-log. *Heuristics, Probabilities and Causality. A tribute to Judea Pearl*, pages 337–359, 2010.

[10] Vibhav Gogate and Pedro Domingos. Probabilistic theorem proving. *arXiv preprint arXiv:1202.3724*, 2012.

[11] Joohyung Lee, Yunsong Meng, and Yi Wang. Markov logc style weighted rules under the stable model semantics. In *Technical Communications of the 31st International Conference on Logic Programming*, 2015.

[12] Joohyung Lee and Yi Wang. Weighted rules under the stable model semantics. In *Proceedings of International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 2016.

[13] Victor W. Marek and Miroslaw Truszczynski. *Stable Models and an Alternative Logic Programming Paradigm*, pages 375–398. The Logic Programming Paradigm: a 25-Year Perspective. Springer Verlag, Berlin, 1999.

[14] Ilkka Niemela. Logic Programs with Stable Model Semantics as a Constraint Programming Paradigm. In *Proceedings of the Workshop on Computational Aspects of Nonmonotonic Reasoning*, pages 72–79, Jun 1998.

[15] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.

[16] Taisuke Sato. A statistical learning method for logic programs with distribution semantics. In *In Proceedings of the 12th International Conference on Logic Programming (ICLP95*. Citeseer, 1995.

[17] Taisuke Sato. Generative modeling by prism. In *Logic Programming*, pages 24–35. Springer, 2009.

[18] Joost Vennekens, Marc Denecker, and Maurice Bruynooghe. Cp-logic: A language of causal probabilistic events and its relation to logic programming. *TPLP*, 9(3):245–308, 2009.

[19] Joost Vennekens, Sofie Verbaeten, and Maurice Bruynooghe. Logic programs with annotated disjunctions. In *Logic Programming*, pages 431–445. Springer, 2004.

[20] Weijun Zhu. *Plog: Its algorithms and applications*. PhD thesis, Texas Tech University, 2012.

A Proof of the Main Theorem (Outline)

In this section we provide a sequence of lemmas which may serve as an outline of the complete proof of the theorem available in appendix B. In section 4 of the main text we have constructed the translation $\tau(M)$ from an LP$^{\text{MLN}}$ program M into a P-log program which has a size linear in terms of the size of M, as well as defined the map ϕ from probabilistic stable models of M into possible worlds of $\tau(M)$. To complete the proof, we need to show that ϕ satisfies the desired properties from definition 1. The properties follow directly from the lemmas:

Lemma 2. For every probabilistic stable model W of M, $\phi(W)$ is a possible world of $\tau(M)$. □

Proof of lemma 2, outline. We need to show that $\phi(W)$ is an answer set of the CR-Prolog program which is used to define the possible worlds of $\tau(M)$ (the translation from a P-log program $\tau(M)$ into the corresponding CR-Prolog program is described in section 2 of the main text). We denote the CR-Prolog program by $\tau^*(M)$. We can prove the lemma in the following steps:

1. Construct a subset γ of consistency restoring rules of $\tau^*(M)$ as follows:

$$\gamma = \{ab(i) \stackrel{+}{\leftarrow} | ab(i) \in \phi(W)\}$$

2. Show that $\phi(W)$ is an answer set of the program constructed from the regular rules of $\tau^*(M)$ and the regular counterparts of the rules in γ.
3. Show that γ is an abductive support of $\tau^*(M)$.

The details of step 2 can be found in sections 2.2-2.3 of the complete proof (Appendix B). Step 3 corresponds to section 2.4 of the complete proof.

□

Lemma 3. For every possible world V of $\tau(M)$, the set of atoms in V from the signature of M is a probabilistic stable model of M. □

Proof of lemma 3, outline. Let W be the set of atoms in V from the signature of M. The correctness of the lemma can be shown in the following two steps:

1. Show that W is a possible world of M. This can be done by showing that W is an ASP model of M_W (a subset of M constructed from all rules of M satisfied by W).
2. Show that W is a probabilistic stable model of M (that is, $P_M(W) > 0$). This can be shown using some properties of LP$^{\text{MLN}}$ programs, including lemma 1 from the main text.

The details are left for section 3.1 of the complete proof. □

Lemma 4. ϕ is surjective, that is, there for every possible world V of $\tau(M)$ there exists a probabilistic stable model W of M such that $\phi(W) = V$. □

Proof of lemma 4, outline. Given a possible world V of $\tau(M)$, let W be a set of atoms obtained from V by removing all atoms not belonging to the signature of M. By lemma 3, W is a probabilistic stable model of M. The fact that $\phi(W) = V$ can be proved by considering every atom l and showing that $l \in \phi(W)$ iff $l \in V$. The details of the last step can be found in section 3.2 of the complete version. □

Lemma 5. ϕ is injective, that is, for every two distinct probabilistic stable models W_1 and W_2 of M, $\phi(W_1) \neq \phi(W_2)$. □

Proof of lemma 5, outline. The correctness of the lemma follows immediately from the fact that $W \subseteq \phi(W)$ for every probabilistic stable model W of M. See section 4 of the complete proof for the details. □

Lemma 6. ϕ is bijective. □

Proof of lemma 6: the fact that ϕ is a bijective follows immediately from lemmas 4 and 5 which state that ϕ is surjective and injective and surjective respectively. □

Lemma 7. For every probabilistic stable model W of M,

$$P_M(W) = P_{\tau(M)}(\phi(W))$$

□

Proof of lemma 6, outline. The equality can be established by applying the definitions of probabilistic functions P_M and $P_{\tau(M)}$ of the corresponding programs. The key observation needed is the fact that the random attribute term $selected(i)$ is true in possible world $\phi(W)$ of $\tau(M)$ if and only if the soft rule of M labeled with i is satisfied by W. See section 5 of the complete proof for details. □

B Proof of the Main Theorem

In this section we complete the proof of the main theorem from section 4. In particular, we show that for an LP$^{\text{MLN}}$ program M, $\tau(M)$ is a counterpart of M.

We start from stating and proving some propositions about LP$^{\text{MLN}}$ which extend the results from [11].

Proposition 1. *If W_1 is a possible world of an LP^{MLN} program M and there exists a possible world of M which satisfies more hard rules than W_1, then $P_M(W_1) = 0$.*

Proof: Let W_2 be a possible world of M which satisfies more hard rules of M than W_1. Let q and r be the number of hard rules satisfied by W_1 and W_2 respectively. From the definition of unnormalized measure w_M it is to see that there exist two positive real numbers y_1 and y_2 such that:

$$w_M(W_1) = y_1 \cdot e^{q\alpha} \tag{28}$$

and

$$w_M(W_2) = y_2 \cdot e^{r\alpha} \tag{29}$$

By definition of probability function P_M we have:

$$\begin{aligned} P_M(W_1) &= \lim_{\alpha\to\infty} \tfrac{w_M(W_1)}{\sum_{J\in\Omega_M} w_M(J)} \text{ (by definition)} \\ &= \lim_{\alpha\to\infty} \tfrac{y_1\cdot e^{r\alpha}}{\sum_{J\in\Omega_M} w_M(J)} \text{ (by (28))} \\ &= \lim_{\alpha\to\infty} \tfrac{y_1\cdot e^{q\alpha}}{w_M(W_2)+\sum_{J\in\Omega_M\setminus\{W_2\}} w_M(J)} \\ &= \lim_{\alpha\to\infty} \tfrac{y_1\cdot e^{q\alpha}}{y_2\cdot e^{r\alpha}+\sum_{J\in\Omega_M\setminus\{W_2\}} w_M(J)} \text{ (by (29))} \end{aligned} \tag{30}$$

To compute the limit in (36), we introduce three functions $f_1(\alpha), f_2(\alpha), f_3(\alpha)$ of a real argument α defined as follows:

$$f_1(\alpha) = 0 \tag{31}$$

$$f_2(\alpha) = \frac{y_1 \cdot e^{q\alpha}}{y_2 \cdot e^{r\alpha} + \sum_{J\in\Omega_\Pi\setminus\{W_2\}} w_\Pi(J)} \tag{32}$$

$$f_3(\alpha) = \frac{e^{q\alpha}\cdot y_1}{e^{r\alpha}\cdot y_2} \tag{33}$$

Then we have

$$\begin{aligned} P_M(W_1) &= \lim_{\alpha\to\infty} \tfrac{y_1\cdot e^{q\alpha}}{y_2\cdot e^{r\alpha}+\sum_{J\in\Omega_M\setminus\{W_2\}} w_M(J)} \text{ (from (30))} \\ &= \lim_{\alpha\to\infty} f_2(\alpha) \text{ (by (32))} \end{aligned} \tag{34}$$

It is easy to see that for every real number α:

$$0 = f_1(\alpha) \leq f_2(\alpha) \leq f_3(\alpha) \tag{35}$$

Therefore, to show that $P_M(W_1) = \lim_{\alpha\to\infty} f_2(\alpha) = 0$, it is sufficient to show $\lim_{\alpha\to\infty} f_3(\alpha) = 0$:

$$\begin{aligned} \lim_{\alpha\to\infty} f_3(\alpha) &= \lim_{\alpha\to\infty} \tfrac{e^{q\alpha}\cdot y_1}{e^{r\alpha}\cdot y_2} \\ &= \lim_{\alpha\to\infty} \tfrac{y_1}{e^{(r-q)\alpha}\cdot y_2} \\ &= 0 \text{ (since } r > q) \end{aligned} \tag{36}$$

□

Proposition 2. *Let M be an LP^{MLN} program and W_1 be a possible world of M satisfying r hard rules of M. Every probabilistic stable model of M satisfies at least r hard rules of M.*

Proof:

Let W_2 be a probabilistic stable model of M. For the sake of contradiction, suppose it satisfies $q < r$ hard rules of M. Then by Proposition 1 we have that $P_M(W_2) = 0$, which contradicts the definition of probabilistic stable model of M.

□

Proposition 3. *Let M be an LP^{MLN} program. Let W be a possible world of Π satisfying q hard rules of M. W is a probabilistic stable model of M if and only if every possible world of M satisfies at most q hard rules of M.*

Proof:

=> Suppose W is a probabilistic stable model of M. For the sake of contradiction, suppose there exists a possible world of M which satisfies $r > q$ hard rules of M. By Proposition 2, W has to satisfy at least r hard rules of M, which is a contradiction, since we know W satisfies $q < r$ hard rules of M.

<= Suppose every possible world of M satisfies at most q rules of M. We show

$$W \text{ is a probabilistic stable model of } \Pi \tag{37}$$

Let $W_1, \ldots, W_n$ be all possible worlds of M. Without loss of generality we will assume $W = W_1$. Let i be an integer in $\{1..n\}$. We have

$$w_M(W_i) = e^{\sum_{w:R \in M, W_i \models R}(w)}$$

Let $\mathcal{A}_i$ be the set of all hard rules of M satisfied by W_i and $\mathcal{B}_i$ be the set of all soft rules of M satisfied by W_i. Then we have

$$w_M(W_i) = e^{|\mathcal{A}_i| * \alpha} \cdot e^{\sum_{w:R \in \mathcal{B}_i}(w)}$$

We denote the sum $\sum_{w:R \in \mathcal{B}_i}(w)$ by y_i and $|\mathcal{A}_i|$ (the number of hard rules of M satisfied by W_i) by h_i. Then we have

$$w_M(W_i) = y_i \cdot e^{h_i * \alpha}$$

Note that y_i is a positive real number.
We have

$$\begin{aligned} P_M(W) &= P_\Pi(W_1) \\ &= \lim_{\alpha \to \infty} \frac{w_M(W_1)}{\sum_{i=1}^n w_M(W_i)} \\ &= \lim_{\alpha \to \infty} \frac{y_1 \cdot e^{h_1 \cdot \alpha}}{\sum_{i=1}^n y_i * e^{h_i \cdot \alpha}} \end{aligned} \tag{38}$$

Since every possible world of M satisfies at most q rules of M, $h_i \le h_1 = q$. Let I be the largest subset of $\{1, \ldots, n\}$ such that for each $j \in I$, W_j satisfies exactly h_1 hard rules (that is, $h_j = h_1$). Then we have

$$\text{for every } j \in \{1, \ldots, n\} \setminus I,\ h_j < h_1 \tag{39}$$

and

$$
\begin{aligned}
P_M(W) &= \lim_{\alpha\to\infty} \frac{y_1 \cdot e^{h_1\cdot\alpha}}{\sum_{i=1}^{n} y_i * e^{h_i\cdot\alpha}} \\
&= \lim_{\alpha\to\infty} \frac{y_1 \cdot e^{h_1\cdot\alpha}}{\sum_{i\in I} y_i * e^{h_1\cdot\alpha} + \sum_{i\in\{1,\ldots,n\}\setminus I} y_i * e^{h_i\cdot\alpha}} \\
&= \lim_{\alpha\to\infty} \frac{y_1 \cdot e^{h_1\cdot\alpha}}{\sum_{i\in I} y_i * e^{h_1\cdot\alpha} + \sum_{i\in\{1,\ldots,n\}\setminus I} y_i * e^{h_i\cdot\alpha}} \\
&= \lim_{\alpha\to\infty} \frac{1}{\sum_{i\in I} \frac{y_i * e^{h_1\cdot\alpha}}{y_1 * e^{h_1\cdot\alpha}} + \sum_{i\in\{1,\ldots,n\}\setminus I} \frac{y_i * e^{h_i\cdot\alpha}}{y_1 * e^{h_1\cdot\alpha}}}
\end{aligned} \tag{40}
$$

To compute the last limit in 40, we introduce a function $f(\alpha)$ of a real argument α:

$$
f(\alpha) = \sum_{i\in I} \frac{y_i * e^{h_1\cdot\alpha}}{y_1 * e^{h_1\cdot\alpha}} + \sum_{i\in\{1,\ldots,n\}\setminus I} \frac{y_i * e^{h_i\cdot\alpha}}{y_1 * e^{h_1\cdot\alpha}} \tag{41}
$$

and compute the limit:

$$
\begin{aligned}
lim_{\alpha\to\infty} f(\alpha) &= lim_{\alpha\to\infty} \left(\sum_{i\in I} \frac{y_i * e^{h_1\cdot\alpha}}{y_1 * e^{h_1\cdot\alpha}} + \sum_{i\in\{1,\ldots,n\}\setminus I} \frac{y_i * e^{h_i\cdot\alpha}}{y_1 * e^{h_1\cdot\alpha}} \right) \\
&= lim_{\alpha\to\infty} \left(\sum_{i\in I} \frac{y_i * e^{h_1\cdot\alpha}}{y_1 * e^{h_1\cdot\alpha}} \right) + lim_{\alpha\to\infty} \left(\sum_{i\in\{1,\ldots,n\}\setminus I} \frac{y_i * e^{h_i\cdot\alpha}}{y_1 * e^{h_1\cdot\alpha}} \right) \\
&= lim_{\alpha\to\infty} \left(\sum_{i\in I} \frac{y_i * e^{h_1\cdot\alpha}}{y_1 * e^{h_1\cdot\alpha}} \right) (\text{since } \forall i \in \{1,\ldots,n\} \setminus I\text{: } h_1 > h_i) \\
&= \sum_{i\in I} \frac{y_i}{y_1}
\end{aligned} \tag{42}
$$

Therefore, from (40) we have:

$$
\begin{aligned}
P_M(W) &= \lim_{\alpha\to\infty} \frac{1}{\sum_{i\in I} \frac{y_i * e^{h_1\cdot\alpha}}{y_1 * e^{h_1\cdot\alpha}} + \sum_{i\in\{1,\ldots,n\}\setminus I} \frac{y_i * e^{h_i\cdot\alpha}}{y_1 * e^{h_1\cdot\alpha}}} \\
&= \lim_{\alpha\to\infty} \frac{1}{f(\alpha)} \\
&= \frac{1}{\lim_{\alpha\to\infty} f(\alpha)} \text{ (we know from (42) that } \lim_{\alpha\to\infty} f(\alpha) \text{ exists)} \\
&= \frac{1}{\sum_{i\in I} \frac{y_i}{y_1}} \\
&= \sum_{i\in I} \frac{y_1}{y_i} > 0 \text{ (since } y_j > 0 \text{ for every } j \in \{1,\ldots,n\})
\end{aligned} \tag{43}
$$

From the fact that W is a possible world of Π and (43) we have (37).

□

It is worth noticing that Proposition 3 is equivalent to Lemma 1 from the main text of the paper.

We are ready to prove that $\tau(M)$ is a counterpart of M. In step 1 of the proof we define a mapping ϕ from probabilistic stable models of M to possible worlds of $\tau(M)$. In 2 we prove for every probabilistic stable model W of M, $\phi(W)$ is a possible world of

$\tau(M)$. In 3-4 we prove that ϕ is a bijection (to do this, we show that ϕ is surjective and injective in 3 and 4 respectively). In 5 we show $P_M(W) = P_{\tau(M)}(\phi(W))$. 1-5 together imply that $\tau(M)$ is a counterpart of M

1. Suppose
$$W \text{ is a probabilistic stable model of } M \tag{44}$$
We define $\phi(W)$ as follows[2]
$$\begin{aligned}\phi(W) := W &\cup \{ab(i) \mid i \in hard, r_i \text{ is not satisfied by } W\}\\ &\cup \{sat(i) \mid i \in soft, r_i \text{ is satisfied by } W\}\\ &\cup \{selected(i) \mid i \in soft, r_i \text{ is not satisfied by } W\}\\ &\cup \{\neg selected(i) \mid i \in soft, r_i \text{ is satisfied by } W\}\\ &\cup \{b(i) \mid i \in soft, \text{ the body of } r_i \text{ is satisfied by W}\}\\ &\cup \{h(i) \mid i \in soft, \text{ the head of } r_i \text{ is satisfied by W}\}\end{aligned} \tag{45}$$
Note that $\phi(W)$ is consistent by construction.
2. Let W be a probabilistic stable model of M. Let τ' be the mapping from P-log programs to ASP program defined in section 4 and τ^* be the composition $\tau' \circ \tau$. We need to show that $\phi(W)$ is an answer set of $\tau^*(M)$.
As in [7], by $\alpha(r)$ we denote a regular rule obtained from a consistency restoring rule r by replacing $\stackrel{+}{\leftarrow}$ with $\leftarrow$; $\alpha(r)$ is expanded in a standard way to a set R of cr-rules, i.e. $\alpha(R) = \{\alpha(r) : r \in R\}$.
In 2.1 we construct a subset γ of consistency restoring rules of $\tau^*(M)$.
Let R be the set of regular rules of $\tau^*(M)$. In 2.2-2.3 we show that $\phi(W)$ is an answer set of $R \cup \alpha(\gamma)$ (in particular, in 2.2 we show that $\phi(W)$ satisfies the rules of $R \cup \alpha(\gamma)$ and in 2.3 we prove that there there does not exist a proper subset of $\phi(W)$ satisfying $(R \cup \alpha(\gamma))^{\phi(W)}$).
In 2.4 we show that γ is an *abductive support* of $\tau^*(M)$. Since from 2.3 it follows that $R \cup \alpha(\gamma)$ is consistent, it is sufficient to show there does not exist a subset γ' of consistency restoring rules of $\tau^*(M)$ such that
(a) $|\gamma'| < \gamma$
(b) the program $R \cup \alpha(\gamma')$ is consistent.
From 2.1 – 2.4 it follows $\phi(W)$ is an answer set of CR-Prolog program $\tau^*(M)$, and therefore a possible world of $\tau(M)$.
2.1 We construct a subset γ of consistency restoring rules of $\tau^*(M)$ as follows:
$$\gamma = \{ab(i) \stackrel{+}{\leftarrow} |ab(i) \in \phi(W)\} \tag{46}$$
2.2 We will prove that $\phi(W)$ satisfies the rules of $R\ \cup \alpha(\gamma)$. In 2.2.1 we prove $\phi(W)$ satisfies the rules in R and In 2.2.2 we prove $\phi(S)$ satisfies the rules in $\alpha(\gamma)$.

[2]Note that in what follows we, as previously, identify a P-log literal of the form $f(\bar{t})$ (which is a shorthand for $f(\bar{t}) = true$) with the literal of the form $f(\bar{t}, true)$ and the literal of the form $\neg f(\bar{t})$ with $f(\bar{t}, false)$. For example, we view $ab(i)$ and $ab(i, true)$ as identical literals. Similarly $\neg selected(i)$ is the same as $selected(i, false)$

2.2.1 We will prove $\phi(W)$ satisfies the rules of R. Let r be a rule in R. In what follows we will consider all possible forms of the rule r and show that r is satisfied. We first notice that, since the rules of M do not contain new literals introduced in $\tau(M)$, for every rule r of M we have:

$$r \text{ is satisfied by } W \text{ iff } r \text{ is satisfied by } \phi(W) \tag{47}$$

if $head \leftarrow body$ is a rule of M

2.2.1.1 Suppose r is of the form

$$head \leftarrow body, not\ ab(i) \tag{48}$$

where $head \leftarrow body$ is a rule of M.
If $\phi(W)$ contains $ab(i)$, the rule is satisfied. Otherwise, by construction of $\phi(W)$, $head \leftarrow body$ is satisfied by W, and, therefore, by (47), the rule (48) is satisfied by $\phi(W)$.

2.2.1.2.2 Suppose r is of the form:

$$head \leftarrow body, selected(i) \tag{49}$$

where $head \leftarrow body$ is a rule of M. If $\phi(W)$ does not contain $selected(i)$, the rule is satisfied. Otherwise, similarly to the previous case, the rule is satisfied.

2.2.1.2.3 Suppose r is of the form:

$$sat(i) \leftarrow b(i), h(i) \tag{50}$$

If $\phi(W)$ satisfies both $b(i)$ and $h(i)$, then the head and the body of r_i are satisfied by W. Therefore, r_i is satisfied by W, and, by construction of $\phi(W)$, $sat(i) \in \phi(W)$. Therefore, $\phi(W)$ satisfies the rule (50).

2.2.1.2.4 Suppose r is of the form:

$$sat(i) \leftarrow not\ b(i) \tag{51}$$

Similarly to the case 2.2.1.2.3 we can show W satisfies r_i, $sat(i) \in \phi(W)$ and, therefore, $\phi(W)$ satisfies the rule (51).

2.2.1.2.5 Suppose r is of the form:

$$b(i) \leftarrow B \tag{52}$$

where B is the body of the rule r_i of M. Suppose $\phi(W)$ satisfies B. Then, by construction of $\phi(W)$, $\phi(W)$ satisfies $b(i)$, and, therefore, $\phi(W)$ satisfies (52).

2.2.1.2.6 Suppose r is of the form:

$$h(i) \leftarrow l \tag{53}$$

By the reasoning similar to the one from case 2.2.1.2.5, we can show that if $\phi(W)$ satisfies l, it also satisfies $h(i)$. Therefore, $\phi(W)$ satisfies (53).

2.2.1.2.7 Suppose r is of the form:

$$\leftarrow \neg selected(i), sat(i) \quad (54)$$

Suppose $\phi(W)$ satisfies $\neg selected(i)$. Thus, by construction of $\phi(W)$, r_i is not satisfied by W and $\phi(W)$ does not contain $sat(i)$. Therefore the rule (54) is satisfied by $\phi(W)$.
If $\neg selected(i)$ is not satisfied by $\phi(W)$, the rule (54) is satisfied by $\phi(W)$.

2.2.1.2.8 Suppose r is of the form:

$$selected(i) \text{ } or \text{ } \neg selected(i). \quad (55)$$

where $i \in soft$. In this case, depending on whether or not r_i is satisfied by W, by construction $\phi(W)$ contains either $selected(i)$ or $\neg selected(i)$ correspondingly. Therefore, the rule (55) is satisfied by $\phi(W)$.

2.2.2 By construction of $\alpha(\gamma)$, it is easy to see that $\alpha(\gamma)$ consists of facts of the form $ab(i)$, where $ab(i) \in \phi(W)$. Therefore, $\phi(W)$ satisfies all facts in $\alpha(\gamma)$.

2.3 We show that every subset of $\phi(W)$ satisfying the reduct $(R \cup \alpha(\gamma))^{\phi(W)}$ is equal to $\phi(W)$.
For the sake of contradiction, suppose there exists a subset W' of $\phi(W)$ such that

$$W' \subsetneq \phi(W) \quad (56)$$

that is, W' is a proper subset of $\phi(W)$, and

$$W' \text{ satisfies } (R \cup \alpha(\gamma))^{\phi(W)} \quad (57)$$

We introduce some notation. By O (read "original") we denote the set of atoms of M. By N (read "newly added") we denote the set of atoms of $\tau^*(M)$ excluding atoms of M. For a set X of atoms of $\tau^*(M)$, by X_O we denote the set of atoms $X \cap O$. Finally, by X_N we denote the set of atoms $X \cap N$.
We derive a contradiction by showing that $W' = \phi(W)$, that is, by showing that (56) does not hold. In 2.3.1 we will prove the implication

$$(W'_O = \phi(W)_O) \implies (W'_N = \phi(W)_N) \quad (58)$$

That is, if W' and $\phi(W)$ coincide on the atoms of M, they must also coincide on other atoms of $\tau^*(M)$.
In 2.3.2 we will show

$$W'_O \subsetneq W \quad (59)$$

In what follows, whenever convenient, we ignore the weights of M_W and treat it as an ASP program. For instance, $(M_W)^W$ will denote the ASP reduct of ASP program M_W with respect to W. In 2.3.3 we will show

$$W'_O \text{ satisfies the rules of } (M_W)^W \quad (60)$$

(60) together with (59) contradicts the fact that W is an ASP model of M_W, and, therefore, the fact that W is a probabilistic stable model of M which was our original supposition (44).

2.3.1 We prove (58). Suppose

$$W'_O = \phi(W)_O \quad (61)$$

We will show

$$W'_N = \phi(W)_N \quad (62)$$

Since $A \subsetneq B$ implies that for an arbitrary set U, $A \cap U$ is a subset of $B \cap U$, (56) implies:

$$W'_N \subseteq \phi(W)_N \quad (63)$$

Therefore, to show (62) it is sufficient to show

$$\phi(W)_N \subseteq W'_N \quad (64)$$

Let l be an atom such that

$$l \in \phi(W)_N \quad (65)$$

In what follows we consider all possible forms of l and show that $l \in W'_N$. In doing that we find useful the following observation. By definition of ϕ,

$$\phi(W)_O = W \quad (66)$$

From (66) and (61) we have

$$W'_O = W \quad (67)$$

2.3.1.1 Suppose $l = ab(i)$.
In this case, by construction of γ,

$$ab(i) \overset{+}{\leftarrow} \text{ belongs to } \gamma \quad (68)$$

Therefore, by definition of α

$$ab(i) \text{ belongs to } \alpha(\gamma) \quad (69)$$

Hence, $ab(i)$ belongs to $(R \cup \alpha(\gamma))^{\phi(W)}$. Therefore, by (57), $ab(i) \in W'$, and, since $ab(i) \in N$, $ab(i) \in W'_N$.

2.3.1.2 Suppose $l = b(i)$. Let B_i be the body of the rule r_i of M. Since $b(i)$ belongs to $\phi(W)$, by its construction B_i is satisfied by W.
From (67) we have

$$B_i \text{ is satisfied by } W' \quad (70)$$

Let B_i' be the set of literals obtained from B_i by removing all extended literals containing not. The reduct $R^{\phi(W)}$ contains the rule

$$b(i) \leftarrow B_i' \tag{71}$$

From (70) we have

$$B_i' \text{ is satisfied by } W' \tag{72}$$

By (57), W' satisfies all the rules in $R^{\phi(W)}$, including the rule (71). Therefore, from (72) we have $b(i) \in W'$, and, since $b(i) \in N$, $b(i) \in W_N'$.

2.3.1.3 Suppose $l = h(i)$. Since, by (65), $h(i)$ belongs to $\phi(W)$, by construction of $\phi(W)$, the head of r_i is satisfied by W. That is, there exists a literal l_i belonging to the head of r_i such that

$$l_i \in W \tag{73}$$

From (73) and (67) we have

$$l_i \in W' \tag{74}$$

By construction, $\tau^*(M)$, and therefore the reduct $(R \cup \alpha(\gamma))^{\phi(W)}$ contain the rule

$$h(i) \leftarrow l_i \tag{75}$$

By (57), W' satisfies all the rules of $(R \cup \alpha(\gamma))^{\phi(W)}$, including (75). Therefore, from (74) we have $h(i) \in W'$, and, since $h(i) \in N$, $h(i) \in W_N'$.

2.3.1.4 Suppose $l = sat(i)$.
In this case, by construction of $\phi(W)$ we have that the rule r_i of M is satisfied by W. There are two possible cases considered in 2.3.1.2.1 and 2.3.1.2.2 below.

2.3.1.4.1 The body and the head of r_i are satisfied by W. By construction of $\phi(W)$ we have

$$b(i) \in \phi(W) \tag{76}$$

and

$$h(i) \in \phi(W) \tag{77}$$

In 2.3.1.2 and 2.3.1.3 we have shown that from (76) and (77) it follows

$$b(i) \in W' \tag{78}$$

and

$$h(i) \in W' \tag{79}$$

respectively.
Since the rule

$$sat(i) \leftarrow b(i), h(i)$$

belongs to the reduct $(R \cup \alpha(\gamma))^{\phi(W)}$, by (57) we have W' satisfies this rule and , from (78) and (79) we have $sat(i) \in W'$, and, since $sat(i) \in N$, we have $sat(i) \in W'_N$.

2.3.1.4.2 The body of the rule r_i is not satisfied by W. In this case, by construction of $\phi(W)$, $b(i) \notin \phi(W)$. In this case the fact

$$sat(i) \tag{80}$$

obtained from the rule

$$sat(i) \leftarrow not\ b(i) \tag{81}$$

of $\tau^*(M)$ belongs to the reduct $(R \cup \alpha(\gamma))^{\phi(W)}$. Since, by (57), W' satisfies $(R \cup \alpha(\gamma))^{\phi(W)}$, we have $sat(i) \in W'$, and, since $sat(i) \in N$, we have $sat(i) \in W'_N$.

2.3.1.5 Suppose $l = selected(i)$. Since $selected(i)$ belongs to $\phi(W)$, by construction, $\neg selected(i)$ does not belong to $\phi(W)$. By (56), $W' \subseteq \phi(W)$. Therefore,

$$\neg selected(i) \notin W' \tag{82}$$

Since

$$selected(i)\ or\ \neg selected(i)$$

belonging to the reduct $(R \cup \alpha(\gamma))^{\phi(W)}$ is satisfied by W' (by (57)), from (82) we have $selected(i) \in W'$. Since $selected(i) \in N$, we have $selected(i) \in W'_N$.

2.3.1.6 Suppose l is $\neg selected(i)$. This case is very similar to 2.3.1.5.

From 2.3.1.1 - 2.3.1.6 we have (64). From (64) and (63) we have (62). Therefore, we proved (62) assuming (61), and the implication (58) holds.

2.3.2 We prove that $W'_O \subsetneq W$.
From the definitions of W'_O and W'_N:

$$W' = W'_O \cup W'_N \tag{83}$$

and

$$\phi(W) = \phi(W)_O \cup \phi(W)_N \tag{84}$$

From (58), (83), (84) we have

$$(W'_O = \phi(W)_O) \implies (W' = \phi(W)) \tag{85}$$

From (85) and (56) we have

$$W'_O \neq \phi(W)_O \tag{86}$$

From (86) and (56) we have

$$W'_O \subsetneq \phi(W)_O \tag{87}$$

By construction of ϕ, we have

$$\phi(W)_O = W \tag{88}$$

From (88) and (87) we have (59).

2.3.3 We prove (60) which says that W'_O satisfies all the rules of the reduct $(M_W)^W$.
We divide the rules of $(M_W)^W$ into two categories: the ones obtained from hard and soft rules of M_W respectively. We will show in 2.3.3.1. and 2.3.3.2 respectively that the rules of both types are satisfied by W'_O.

2.3.3.1 Let r be a rule of $(M_W)^W$ of the form $head \leftarrow body$ such that

$$r \text{ is obtained from a hard rule } r_i \text{ of } M_W \tag{89}$$

That is, r_i is a rule of M such that

$$r_i \text{ is satisfied by } W \tag{90}$$

We will prove that

$$W'_O \text{ satisfies } r \tag{91}$$

by showing that the rule r belongs to the reduct $(R \cup \alpha(\gamma))^{\phi(W)}$ and, therefore, is satisfied by W' by (57). Since r only contains atoms from O, this will immediately imply (91).
Let r_i be of the form

$$\alpha : head \leftarrow body'$$

We will show

$$body' \setminus body \text{ is satisfied by } \phi(W) \tag{92}$$

For the sake of contradiction, suppose

$$body' \setminus body \text{ is not satisfied by } \phi(W) \tag{93}$$

In this case there exists $not\ l' \in body'$ such that

$$l' \in \phi(W) \tag{94}$$

But, since l' occurs in a rule of M, and therefore belongs to O, from (94) we have

$$not\ l' \text{ is not satisfied by } W \tag{95}$$

But then we have $not\ l'$ in $body'$ which is not satisfied by W, therefore, the rule $head \leftarrow body$ of $(M_W)^W$ cannot be obtained from the hard rule $\alpha : head \leftarrow body'$ of M_W, which is a contradiction to (89). Therefore, (92) holds. From (92), the construction of $\phi(W)$, and the fact that $body$ and $body'$ are constructed from the literals of M, we have

$$body' \setminus body \text{ is satisfied by } W \tag{96}$$

By construction of $\tau^*(M)$, the rule

$$head \leftarrow body', not\ ab(i) \tag{97}$$

belongs to R, the set of all regular rules of $\tau^*(M)$.
By (57) W' satisfies all the rules of $(R \cup \alpha(\gamma))^{\phi(W)}$, therefore

$$W' \text{ satisfies the reduct } \{head \leftarrow body', not\ ab(i)\}^{\phi(W)} \tag{98}$$

By (90), r_i is satisfied by W. Therefore, by construction of $\phi(W)$ we have

$$ab(i) \notin \phi(W) \tag{99}$$

Since W' is a subset of $\phi(W)$ (by (56)), we have

$$ab(i) \notin W' \tag{100}$$

Since $ab(i)$ does not belong to $\phi(W)$, from (98) we have

$$W' \text{ satisfies the reduct } \{head \leftarrow body'\}^{\phi(W)} \tag{101}$$

Since all the literals in $head$ and $body'$ are atoms from O, the reduct of $head \leftarrow body'$ with respect to $\phi(W)$ is the same as the reduct of $head \leftarrow body'$ with respect to W. Therefore,

$$W' \text{ satisfies the reduct } \{head \leftarrow body'\}^{W} \tag{102}$$

Since all the literals in $head$ and $body'$ are atoms from O, possibly preceded by default negation, from (102) we have

$$W'_O \text{ satisfies the reduct } \{head \leftarrow body'\}^{W} \tag{103}$$

By (96), the reduct of $head \leftarrow body'$ with respect to W is $head \leftarrow body$. Thus, from (103) we have (91).

2.3.3.2 Let r be a rule of $(M_W)^W$ of the form $head \leftarrow body$ obtained from a soft rule r_i of M_W. We will show

$$W'_O \text{ satisfies } r \tag{104}$$

Let r_i be of the form

$$w : head \leftarrow body'$$

Since r_i belongs to M_W, r_i is satisfied by W. Therefore, we have

$$selected(i) \in \phi(W) \tag{105}$$

and

$$\neg selected(i) \notin \phi(W) \tag{106}$$

By construction of R the rule

$$selected(i) \mid \neg selected(i) \tag{107}$$

belongs to R and, hence, to the reduct $R^{\phi(W)}$. From (56) and (106) we have

$$\neg selected(i) \notin W' \tag{108}$$

Since, by (57), W' satisfies all the rules of the reduct $R^{\phi(W)}$, it satisfies the rule (107), and

$$selected(i) \in W' \text{ or } \neg selected(i) \in W' \tag{109}$$

From (108) and (109) we have

$$selected(i) \in W' \tag{110}$$

The further reasoning needed to obtain (104) is similar to the case 2.3.3.1. The only difference is that we need to consider the rule

$$head \leftarrow body', selected(i) \tag{111}$$

from R where $selected(i) \in W'$ instead of the rule (97) we considered in 2.3.3.1 with $ab(i) \notin W'$.

2.4 We show that γ, constructed in 2.1, is an abductive support of $\tau^*(M)$. In 2.3 we have shown that the program $(R \cup \alpha(\gamma))$ is consistent (in particular, it has an answer set $\phi(W)$). Therefore, it is sufficient to show that there does not exist a subset γ' of consistency restoring rules of $\tau^*(M)$ such that
(a) $|\gamma'| < |\gamma|$
(b) the program $R \cup \alpha(\gamma')$ is consistent.
We prove by contradiction. Suppose there exists γ' such that (a) and (b) hold Let W' be an answer set of $R \cup \alpha(\gamma')$.
As in 2.2, we will use the notation W'_O to denote a subset of W' consisting of all literals of M and W'_N to denote the set difference $W' \setminus W'_O$.
In 2.4.1 we show that $W'_O \in \Omega_M$. In 2.4.2 we obtain a contradiction to the fact that W is a probabilistic stable model of M by establishing $Pr_M(W) = 0$. We do so by proving that W satisfies less hard rules than W'_O and using Proposition 1.

2.4.1 We show $W'_O \in \Omega_M$. By definition of a possible world of M, it is sufficient to prove there is no proper subset of W'_O satisfying the rules $(M_{W'_O})^{W'_O}$. For the sake of contradiction, suppose there exists W''_O such that

$$W''_O \subsetneq W'_O \tag{112}$$

and

$$W''_O \text{ satisfies } (M_{W'_O})^{W'_O} \tag{113}$$

Let H be the set of literals defined as:

$$H = W''_O \cup W'_N \tag{114}$$

It is easy to see that, by 112 and 114,

$$H \subsetneq W' \tag{115}$$

We prove

$$H \text{ satisfies the rules of } (R \cup \alpha(\gamma'))^{W'} \tag{116}$$

Let r be a rule in $(R \cup \alpha(\gamma'))^{W'}$. Suppose

$$\text{the body of } r \text{ is satisfied by } H \tag{117}$$

(otherwise r is satisfied by H). Since r belongs to a reduct, its body contains no default negations, and therefore from (115) and (117) we have

$$\text{the body of } r \text{ is satisfied by } W' \tag{118}$$

Since W' is an answer set of $(R \cup \alpha(\gamma'))$, it satisfies all the rules of $(R \cup \alpha(\gamma'))^{W'}$, including r. Therefore, from (118) we have

$$\text{the head of } r \text{ is satisfied by } W' \tag{119}$$

Since W' and H coincide on the atoms in N, if the head of r is in N, (119) implies r is satisfied by H. Therefore, it is sufficient to consider possible forms of r in case the head of r is constructed from the atoms in O. There are only two possible cases, 2.4.1.1 and 2.4.1.2, considered below.

2.4.1.1 Suppose r is of the form

$$head \leftarrow body$$

where the corresponding rule r^* of R from which r was obtained is of the form

$$head \leftarrow body, not\ l_1, \ldots, not\ l_k, not\ ab(i) \tag{120}$$

By construction of R, the rule

$$head \leftarrow body, not\ l_1, \ldots, not\ l_k \tag{121}$$

belongs to M. We will denote this rule by r^M.
Since r belongs to the reduct $(R \cup \alpha(\gamma'))^{W'}$, and $\{l1, \ldots, l_k\} \subseteq O$ we must have

$$ab(i) \notin W' \tag{122}$$

and

$$\{l_1, \ldots, l_k\} \cap W'_O = \emptyset \tag{123}$$

To prove r is satisfied by H, suppose *body* is satisfied by H, and, therefore, since body is constructed from the atoms in O,

$$body \subseteq W''_O \tag{124}$$

From (112) we have $W''_O \subseteq W'_O$, and therefore:

$$body \subseteq W'_O \tag{125}$$

From (125) and (123) we have:

$$W'_O \text{ satisfies the body of } r^M \tag{126}$$

From (126), (122) we have that

$$W' \text{ satisfies the body of } r^* \tag{127}$$

From the fact that W' is an answer set of $R \cup \alpha(\gamma'')$, and therefore satisfies the rules of R including r^* we have

$$W' \text{ satisfies } head \tag{128}$$

Since $head$ is constructed from the atoms in O, we have:

$$W'_O \text{ satisfies } head \tag{129}$$

From (129) we have

$$W'_O \text{ satisfies } r^M \tag{130}$$

Since r^M belongs to M, from (130) (by definition of $M_{W'_O}$) we have

$$r^M \in M_{W'_O} \tag{131}$$

From (123) and the fact that $\{l_1, \ldots, l_n\} \subseteq O$ we have:

$$\{r^M\}^{W'_O} = \{head \leftarrow body\} \subseteq (M_{W'_O})^{W'_O} \tag{132}$$

Since, by (113), W''_O satisfies the rules of $(M_{W'_O})^{W'_O}$, including $head \leftarrow body$, from (124) we have

$$W''_O \text{ satisfies } head \tag{133}$$

By (114), $W''_O \subseteq H$, and therefore

$$H \text{ satisfies } head \tag{134}$$

Therefore, H satisfies r.

2.4.1.2 Suppose r is of the form

$$head \leftarrow body, selected(i)$$

where the corresponding rule r^* of R from which r was obtained is of the form

$$head \leftarrow body, selected(i), not\ l_1, \dots, not\ l_k \qquad (135)$$

We will consider two possible cases

2.4.1.2.1 Suppose $selected(i) \notin W'$. In this case, by construction of H, $selected(i) \notin H$, and therefore r is satisfied by H.

2.4.1.2.2 Suppose $selected(i) \in W'$. This case is similar to 2.4.1.1, except we use the fact $selected(i) \in W'$ instead of (122) in all the arguments from there.

Therefore, $H \subsetneq W'$ satisfies the rules of $(R \cup \alpha(\gamma'))^{W'}$ which contradicts the fact that W' is an answer set of $R \cup \alpha(\gamma')$.

2.4.2 We show $P_M(W) = 0$. From 2.4.1 we have that

$$W'_O \in \Omega_M \qquad (136)$$

Let h be the number of hard rules in M and k be the number of hard rules of M satisfied by W. Then we have $|\gamma| = h - k$ (by construction of $\phi(W)$ and γ).

We next show

$$W'_O \text{ satisfies at least } h - |\gamma'| \text{ hard rules of } M \qquad (137)$$

Let $ab(i)$ be the head of a cr-rule of $\tau^*(M)$ not belonging to γ' (such a rule must exist since $|\gamma'| < |\gamma|$ and γ is a subset of cr-rules of $\tau^*(M)$).

By construction, $ab(i)$ does not belong to the heads of $R \cup \alpha(\gamma')$. Hence,

$$ab(i) \notin W' \qquad (138)$$

By construction of $\tau^*(M)$, R contains the rule

$$head \leftarrow body, not\ ab(i)$$

Since W' is an answer set of $R \cup \alpha(\gamma')$, it satisfies the rules of M, and, therefore,

$$W' \text{ satisfies } head \leftarrow body, not\ ab(i) \qquad (139)$$

From (138) and (139) we have that W' satisfies $head \leftarrow body$, and, therefore, the hard rule

$$\alpha : head \leftarrow body$$

of M is satisfied by W'_O.

Therefore, since $ab(i)$ was chosen arbitrarily from the heads of cr-rules not belonging to γ', we have (137).
Since $|\gamma'| < |\gamma|$, for some positive integer m from (137) we have

$$W'_O \text{ satisfies } h - |\gamma| + m = k + m \text{ hard rules of } M \tag{140}$$

From (140) and the fact that W satisfies k hard rules of M, W satisfies less hard rules then W'_O. Since both W and W'_O are possible worlds of M, by Proposition 1, we have $P_M(W) = 0$. Therefore, γ' cannot exist, and γ is an abductive support of $\tau^*(M)$.

To summarize, from 2.2-2.3 it follows $\phi(W)$ is an answer set of $R \cup \alpha(\gamma)$, where R is the set of regular rules of $\tau^*(M)$ and γ is a subset of consistency restoring rules of $\tau^*(M)$ constructed in 2.1. In 2.4 we have shown that γ is an abductive support of $\tau^*(M)$. Together 2.1 - 2.4 imply that $\phi(W)$ is an answer set of $\tau^*(M)$ and therefore a possible world of $\tau(M)$.

3. We show ϕ is surjective. That is, for every possible world V of $\tau(M)$ there exists a probabilistic stable model U of M s.t. $\phi(U) = V$. In 3.1 we prove V_O (V_O here is the notation introduced in step 2 meaning the subset of all atoms of M in V) is a probabilistic stable model of M. In 3.2 we show $\phi(V_O) = V$.

3.1 Since V is a possible world of Π, it is, by definition, an answer set of CR-Prolog program $\tau^*(M)$, which is, in turn, an answer set of the ASP program $R \cup \alpha(\gamma^*)$, where R is the set of regular rules of $\tau^*(M)$ and γ^* is an abductive support of $\tau^*(M)$. First of all, by applying exactly the same reasoning as in 2.4.1, we get

$$V_O \in \Omega_M \tag{141}$$

Therefore, we only need to show

$$P_M(V_O) \neq 0 \tag{142}$$

Let h be the number of hard rules in M. In 3.1.1 we show that every possible world of M satisfies at most $h - |\gamma^*|$ hard rules of M, in 3.1.2 we show that V_O satisfies at least $h - |\gamma^*|$ hard rules of M, and in 3.1.3 we show $P_M(V_O) > 0$ using the results from 3.1.2 and 3.1.1 and Proposition 3.

3.1.1 We show by contradiction that every possible world of M satisfies at most $h - |\gamma^*|$ hard rules of M. Suppose there exists $X \in \Omega_M$ such that for some $m > 0$

$$X \text{ satisfies } h - |\gamma^*| + m \text{ hard rules of } M \tag{143}$$

We denote $h - |\gamma^*| + m$ by k. By the reasoning identical to the one we applied in 2.1 - 2.3 we have

$$\phi(X) \text{ is an answer set of } R \cup \alpha(\gamma^+) \tag{144}$$

for a subset γ^+ of consistency restoring of $\tau^*(M)$, such that

$$|\gamma^+| = h - k \tag{145}$$

Therefore, from (143) and (145) we have

$$|\gamma^+| = h - k \tag{146}$$
$$= h - h - |\gamma^*| + m \tag{147}$$
$$= |\gamma^*| - m \tag{148}$$

Since $m > 0$, and the program $R \cup \alpha(\gamma^+)$ is consistent (by (144)), we have a contradiction to the fact that abductive support γ^* is minimal (by cardinality) subset of consistency restoring rules of M such that the program $R \cup \alpha(\gamma^*)$ is consistent.

3.1.2 We show that V_O satisfies at least $h - |\gamma^*|$ hard rules of M. If $h = |\gamma^*|$, the claim is obviously true. Otherwise, there exists at least one consistency restoring rule $ab(i) \stackrel{+}{\leftarrow}$ such that

$$ab(i) \stackrel{+}{\leftarrow} \quad \notin \gamma^* \tag{149}$$

By construction of $\tau^*(M)$, R contains the rule

$$head \leftarrow body, not\ ab(i) \tag{150}$$

From (149) and by construction of $\tau^*(M)$, the program $R \cup \alpha(\gamma^*)$ does not contain any rules with $ab(i)$ in the head, and V does not contain $ab(i)$. Since V satisfies all rules of R, including (150), and $ab(i) \notin V$, we have the hard rule

$$\alpha : head \leftarrow body$$

is satisfied by V and, since all the atoms in the rule are in O, by V_O. Therefore, since we have $h - |\gamma^*|$ distinct literals of the form $ab(i)$ not belonging to γ^*, V_O satisfies at least $h - |\gamma^*|$ distinct hard rules of M.

3.1.3 We show

$$P_M(V_O) > 0 \tag{151}$$

Let $W_1, \ldots, W_n$ be all the members of Ω_M. Without loss of generality, we will assume $V_O = W_1$. Let h_i be the number of hard rules of M satisfied by W_i. By 3.1.1 and 3.1.2 we have $h_i \leq h_1$ Therefore, by (141) and Proposition 3, we have $W_1 = V_O$ is a probabilistic stable model of M, that is, V_O is a possible world of M such that (151) holds.

3.2 We show $\phi(V_O) = V$. We need to show that for every literal l of $\tau^*(M)$,

$$l \in V \text{ if and only if } l \in \phi(V_O) \tag{152}$$

First of all, if l is an atom in O, then, by definition of V_O and construction of ϕ, $l \in V$ if and only if $l \in \phi(V_O)$. In what follows we will consider all other possible forms of l.

3.2.1 Suppose $l = ab(i)$, where i is the label of a hard rule r_i of M of the form

$$\alpha : head \leftarrow body$$

To show (152), it is sufficient to show

$$ab(i) \notin V \text{ if and only if } ab(i) \notin \phi(V_O) \tag{153}$$

By construction of ϕ,

$$ab(i) \in \phi(V_O) \text{ iff } r_i \text{ is not satisfied by } V_O \tag{154}$$

From (154) we have:

$$ab(i) \notin \phi(V_O) \text{ iff } r_i \text{ is satisfied by } V_O \tag{155}$$

Therefore, based on (155), to show (153) (and, therefore, (152)), it is sufficient to show:

$$r_i \text{ is satisfied by } V_O \text{ iff } ab(i) \notin V \tag{156}$$

Suppose $ab(i) \notin V$. By construction of τ^*, the rule r_i^*

$$head \leftarrow body, not\ ab(i)$$

belongs to $\tau^*(M)$. Since, by definition, V satisfies $\tau^*(M)$, V satisfies r^*. Therefore, since $ab(i) \notin V$,

$$V \text{ satisfies } head \leftarrow body \tag{157}$$

Since r_i is of the form $\alpha : head \leftarrow body$, where $head$ and $body$ are constructed from atoms in O, from (157) we have

$$V_O \text{ satisfies } r_i \tag{158}$$

Now suppose r_i is satisfied by V_O. For the sake of contradiction, suppose

$$ab(i) \in V \tag{159}$$

By construction, the only rule of $\tau^*(M)$ with $ab(i)$ in the head is the consistency restoring rule $ab(i) \stackrel{+}{\leftarrow}$. Therefore, there exists an abductive support σ of $\tau^*(M)$ such that

$$ab(i) \stackrel{+}{\leftarrow} \quad \in \sigma \tag{160}$$

and

$$V \text{ is an answer set of } R \cup \alpha(\sigma) \tag{161}$$

Let us consider the set

$$V' = V \setminus \{ab(i)\} \tag{162}$$

Let σ' be the set of consistency restoring rules of $\tau^*(M)$ obtained from V' by converting every atom of the form $ab(j)$ in V' into consistency restoring rule $ab(j) \stackrel{+}{\leftarrow}$. From (159), (161) and (162) we have

$$\sigma' \subsetneq \sigma \tag{163}$$

We will show σ' is an abductive support of $\tau^*(M)$, which together with (163) gives a contradiction to the fact that σ is an abductive support of $\tau^*(M)$. To show σ' is an abductive support of $\tau^*(M)$, it is sufficient to show

$$V' \text{ is an answer set of } R \cup \alpha(\sigma') \tag{164}$$

We denote $R \cup \alpha(\sigma')$ by Π_1. Let $P = \Pi_1 \setminus \{r^*\}$. Then we have

$$\begin{aligned} \Pi_1^{V'} =& \{head \leftarrow body\}^{V'} \\ & \cup P^{V'} \end{aligned} \tag{165}$$

Since $ab(i)$ does not occur in P, by 162 we have:

$$P^{V'} = P^{V} \tag{166}$$

From (163) and (161) and (166) we have that V satisfies the rules of $P^{V'}$. Since those rules do not contain an occurrence of $ab(i)$, by construction of V' we have

$$V' \text{ satisfies the rules of } P^{V'}. \tag{167}$$

By (158), V_O satisfies

$$head \leftarrow body$$

Therefore, from (162) and the fact that $body$ and head are constructed from atoms of O, we have

$$V' \text{ satisfies } head \leftarrow body \tag{168}$$

From (165), (167) and (168) it follows that

$$V' \text{ satisfies all the rules of } \Pi_1^{V'} \tag{169}$$

We now show V' is minimal. Suppose there exists V'' such that

$$V'' \subsetneq V' \tag{170}$$

and

$$V'' \text{ satisfies the rules of } \Pi_1^{V'} \tag{171}$$

Let

$$V''' = V'' \cup \{ab(i)\} \tag{172}$$

By (162) and (159) and the fact that $V'' \subsetneq V'$ we have

$$V''' \subseteq V \tag{173}$$

Let us now show that V''' is a proper subset of V. Since $V'' \subsetneq V'$ and V' does not contain $ab(i)$ by construction, we have

$$\text{there exists an atom } l \text{ different from } ab(i) \quad \text{s.t. } l \in \{V' \setminus V''\} \tag{174}$$

By construction of V', $l \in V$. Therefore, by (172) and (174), $l \notin V'''$, and

$$V''' \subsetneq V \tag{175}$$

We will next show that

$$V''' \text{ satisfies } (R \cup \alpha(\sigma))^V \tag{176}$$

We notice

$$(R \cup \alpha(\sigma))^V = \Pi_1^V \cup \{ab(i)\}$$

In 3.2.1.1 we will prove

$$V''' \text{ satisfies } \Pi_1^V \tag{177}$$

and in 3.2.1.2 we will prove

$$V''' \text{ satisfies } ab(i) \tag{178}$$

From (177) and (178) we will have (176).

3.2.1.1 We prove Π_1^V is satisfied by V''' By definition of reduct, since $V' = V \setminus \{ab(i)\} \subsetneq V$, we have

$$\Pi_1^V \subseteq \Pi_1^{V'} \tag{179}$$

By (171),

$$V'' \text{ satisfies the rules of } \Pi_1^V \tag{180}$$

Since Π_1^V does not contain an occurrence of $ab(i)$, from (172) and (180) we have

$$V''' \text{ satisfies the rules of } \Pi_1^V \tag{181}$$

3.2.1.2 By (172), V''' satisfies $ab(i)$.

Therefore, $V''' \subsetneq V$ satisfies all the rules of $(R \cup \alpha(\sigma))^V$ which is a contradiction to the fact that V is an answer set of $(R \cup \alpha(\sigma))^V$.
Therefore, V'' satisfying conditions (171) and (170) cannot exist and V' is an answer set of $\Pi_1 = R \cup \alpha(\sigma')$, and σ' is an abductive support of $\tau^*(M)$, which, by (163) is a contradiction to the fact that σ is an abductive support of $\tau^*(M)$. Therefore, (159) does not hold, $ab(i) \notin V$ and (156) holds.

3.2.2 Suppose $l = h(i)$, where r_i is of the form $w : head \leftarrow body$. The only rules of $\tau^*(M)$ with $h(i)$ in the head are

$$\{h(i) \leftarrow l' | l' \in head\}$$

Therefore,

$$h(i) \text{ belongs to } V \text{ iff } \exists l' \in head \text{ s.t. } l' \in V \tag{182}$$

By construction of $\phi(V_O)$ and definition of V_O,

$$\exists l' \in head \text{ s.t. } l' \in V \text{ iff } \exists l' \in head \text{ s.t } l' \in \phi(V_O) \tag{183}$$

By construction of ϕ,

$$\exists l' \in head \text{ s.t. } l' \in \phi(V_O) \text{ iff } h(i) \text{ belongs to } \phi(V_O) \tag{184}$$

From (182) - (184) we have

$$h(i) \text{ belongs to } V \text{ iff } h(i) \text{ belongs to } \phi(V_O) \tag{185}$$

3.2.3 Suppose l is of the form $b(i)$, where
r_i be of the form $w : head \leftarrow body$. The only rule of $\tau^*(M)$ defining $b(i)$ is

$$b(i) \leftarrow B$$

where B is a set of literals formed by atoms in O. Therefore,

$$b(i) \text{ belongs to } V \text{ iff } B \text{ is satisfied by } V \tag{186}$$

By construction of $\phi(V_O)$ and definition of V_O,

$$B \text{ is satisfied by } V \text{ iff } B \text{ is satisfied by } \phi(V_O) \tag{187}$$

By construction of ϕ,

$$B \text{ is satisfied by } \phi(V_O) \text{ iff } b(i) \text{ belongs to } \phi(V_O) \tag{188}$$

From (186) - (188) we have

$$b(i) \text{ belongs to } V \text{ iff } b(i) \text{ belongs to } \phi(V_O) \tag{189}$$

3.2.4 Suppose $l = sat(i)$.
By applying the reasoning identical to the one in 2.2-2.3, from (141) we have

$$\phi(V_O) \text{ is an answer set of } R \cup \alpha(\sigma_1) \tag{190}$$

for some subset σ_1 of consistency restoring rules of $\tau^*(M)$.
The only rules of $\tau^*(M)$ and $R \cup \alpha(\sigma_1)$ with $sat(i)$ in the head are :

$$sat(i) \leftarrow b(i), h(i)$$

and

$$sat(i) \leftarrow not\ b(i)$$

By 3.2.2 and 3.2.3, each of the atoms $b(i)$ and $h(i)$ belongs to V if and only if it belongs to $\phi(V_O)$, Therefore, from (190) and the fact that V is an answer set of $\tau^*(M)$ by minimality of answer sets we have that $sat(i)$ belongs to V if and only if $sat(i)$ belongs to $\phi(V_O)$.

3.2.5 Suppose $l = selected(i)$

We first prove the following:

$$selected(i) \in \phi(V_O) \text{ iff } sat(i) \in \phi(V_O) \tag{191}$$

Indeed, by construction of $\phi(V_O)$, $selected(i) \in \phi(V_O)$ iff r_i is satisfied by V_O iff $sat(i) \in \phi(V_O)$.

We next prove the following:

$$selected(i) \in V \text{ iff } sat(i) \in V \tag{192}$$

First of all, since V is an answer set of $\tau^*(M)$,

$$\text{either } \neg selected(i) \text{ or } selected(i) \text{ belongs to } V \tag{193}$$

Suppose $sat(i) \in V$. In this case $\neg selected(i) \notin V$ (or else, the constraint

$$\leftarrow \neg selected(i), sat(i)$$

is violated). Therefore, $selected(i) \in V$.

Suppose now $selected(i) \in V$. For the sake of contradiction assume

$$sat(i) \notin V \tag{194}$$

Let r_i be of the form $w:\ head \leftarrow body$. By construction of $\tau^*(M)$, $sat(i)$ is defined by the rules

$$sat(i) \leftarrow not\ b(i)$$

$$sat(i) \leftarrow b(i), h(i)$$

where the only rules of $\tau^*(M)$ defining $h(i)$ and $b(i)$ are

$$\{h(i) \leftarrow l | l \in head\}$$

and

$$b(i) \leftarrow body$$

Since V is an answer set of $\tau^*(M)$, $sat(i) \in V$ iff $head \leftarrow body$ is satisfied by V. Therefore, by (194), we have $head \leftarrow body$ is not satisfied by V. But then the rule

$$head \leftarrow body, selected(i)$$

of $\tau^*(M)$ is not satisfied by V, which is a contradiction.

Therefore, (194) does not hold and we have

$$sat(i) \in V \tag{195}$$

Therefore, (192) holds.

By (192), (191) and 3.2.4 we have $selected(i) \in V$ iff $selected(i) \in \phi(V_O)$

3.2.6 Suppose $l = \neg selected(i)$.
Since the rule

$$selected(i) \mid \neg selected(i) \tag{196}$$

belongs to $\tau^*(M)$, the disjunction $selected(i) \mid \neg selected(i)$ has to be satisfied by both V (since it's an answer set of program containing (196)) and $\phi(V_O)$ (by construction of ϕ)).

3.2.6.1 Suppose $\neg selected(i) \in V$. Since V satisfies the rules of $\tau^*(M)$, including the constraint:

$$\leftarrow \neg selected(i), sat(i) \tag{197}$$

we have

$$sat(i) \notin V \tag{198}$$

By 3.2.4 we have

$$sat(i) \notin \phi(V_O) \tag{199}$$

By construction of ϕ we have

$$selected(i) \notin \phi(V_O) \tag{200}$$

and

$$\neg selected(i) \in \phi(V_O) \tag{201}$$

3.2.6.2 Suppose $\neg selected(i) \in \phi(V_O)$. By construction of ϕ, $selected(i) \notin \phi(V_O)$. By 3.2.5, $selected(i) \notin V$. Since V satisfies the rule

$$selected(i)|\neg selected(i)$$

$\neg selected(i) \in V$.

4. We show ϕ is injective. That is, for every two distinct probabilistic stable models W_1 and W_2 of M, we will show

$$\phi(W_1) \neq \phi(W_2) \tag{202}$$

By definition of ϕ, for every possible world U of M, $\phi(U)_O = U$. Therefore,

$$\phi(W_1)_O = W_1 \tag{203}$$

and

$$\phi(W_2)_O = W_2 \tag{204}$$

From (203) and (204) and the fact that W_1 and W_2 are distinct we have

$$\phi(W_1)_O \neq \phi(W_1)_O \tag{205}$$

From (205) we immediately have (202).

5. We show

$$P_M(W) = P_{\tau(M)}(\phi(W)) \tag{206}$$

Let F^W_{SAT} (F^W_{UNSAT}) be the set of soft rules of M satisfied (unsatisfied) by W. Similarly, let H^W_{SAT} (H^W_{UNSAT}) be the set of hard rules of M satisfied (unsatisfied) by W. We first prove the following:

$$\text{All possible worlds of } M \text{ satisfy at most } |H^W_{SAT}| \text{ hard rules of } M \tag{207}$$

Suppose there exists a possible world W' of M such that it satisfies $q > |H^W_{SAT}|$ hard rules of M. We will denote $|H^W_{SAT}|$ $r = |H^W_{SAT}|$). Note that $q > r$. Then by Proposition 1 we have $Pr_M(W) = 0$ which is a contradiction to the fact that W is a probabilistic stable model of M.
We then perform a computation similar to the one we did in in the proof of Proposition 3. Let $\Omega_M = \{W_1, \ldots, W_n\}$. Without loss of generality we can assume $W = W_1$. Let I be the largest subset of $\{1, \ldots, n\}$ such that for every $i \in I$, W_i satisfies exactly $|H^W_{SAT}|$ hard rules of M, let h_i be the number of hard rules satisfied by W_i and let y_i denote $exp(\sum_{w:R\in F^{W_i}_{SAT}} w)$
Similarly to (43) we have

$$\begin{aligned} P_M(W) &= \frac{y_1}{\sum_{i\in I} y_i} \\ &= \frac{exp(\sum_{w:R\in F^W_{SAT}} w)}{\sum_{i\in I} exp(\sum_{w:R\in F^{W_i}_{SAT}} w)} \end{aligned} \tag{208}$$

We next compute $P_{\tau(M)}(\phi(W))$.

In what follows we will sometimes use the shorthand sel for $selected$. By $P(sel(j))$ and $P(\neg sel(j))$ we denote the probabilities of $sel(j)$ and $\neg sel(j)$ respectively defined by pr-atoms of $\tau(M)$ (the probabilities are $P(sel(j)) = \frac{e^{w^j}}{1+e^{w^j}}$ and $P(\neg sel(j)) = \frac{1}{1+e^{w^j}}$ respectively). We also denote the set of all probabilistic stable models of M by Ω^+_M. We then have

$$\begin{aligned} P_{\tau(M)}(\phi(W)) &= \frac{w_{\tau(M)}(\phi(W))}{\sum_{W_i\in\Omega^+_M} w_{\tau(M)}(\phi(W_i))} \\ &= \frac{\prod_{sel(j)\in\phi(W)} P(sel(j))\cdot\prod_{\neg sel(j)\in\phi(W)} P(\neg sel(j))}{\sum_{W_i\in\Omega^+_M}\left(\prod_{sel(j)\in\phi(W_i)} P(sel(j))\cdot\prod_{\neg sel(j)\in\phi(W_i)} P(\neg sel(j)\right)} \end{aligned} \tag{209}$$

We note that for a probabilistic stable model W of M, $selected(j)$ belongs to $\phi(W)$ if and only if r_j is a soft rule of M satisfied by W if and only if $sat(j)$ belongs to $\phi(W)$.

Therefore,

$$\prod_{selected(j)\in\phi(W)} P(selected(j)) = \prod_{r_j\in F^W_{SAT}} p(j) \tag{210}$$

where $p(j)$ denotes $\frac{e^{w_j}}{1+e^{w_j}}$.

Similarly, for a probabilistic stable model W of M, $\neg selected(j)$ belongs to $\phi(W)$ if and only if r_j is a soft rule of M not satisfied by W if and only if $\neg sat(j)$ belongs to $\phi(W)$. Therefore,

$$\prod_{\neg selected(j)\in\phi(W)} P(\neg selected(j)) = \prod_{r_j\in F^W_{UNSAT}} (1-p(j)) \tag{211}$$

By U we will denote the product $\prod_{w:R\in M}(1+e^w)$. From (209) - (211) we have:

$$\begin{aligned}
P_{\tau(M)}(\phi(W)) &= \frac{\prod_{r_i\in F^W_{SAT}} p(i)\cdot\prod_{r_i\in F^W_{UNSAT}}(1-p(i))}{\sum_{W_j\in\Omega^+_M}\left(\prod_{r_i\in F^{W_j}_{SAT}} p(i)\cdot\prod_{r_i\in F^{W_j}_{UNSAT}}(1-p(i))\right)}\\
&= \frac{\prod_{w:R\in F^W_{SAT}} e^w/(1+e^w)\cdot\prod_{w:R\in F^W_{UNSAT}} 1/(1+e^w)}{\sum_{W_j\in\Omega^+_M}\left(\prod_{w:R\in F^{W_j}_{SAT}} e^w/(1+e^w)\cdot\prod_{r\in F^{W_j}_{UNSAT}} 1/(1+e^w)\right)}\\
&= \frac{\left(\prod_{w:R\in F^W_{SAT}} e^w\right)/U}{\sum_{W_j\in\Omega^+_M}\left(\prod_{w:R\in F^{W_j}_{SAT}} e^w\right)/U}\\
&= \frac{\left(\prod_{w:R\in F^W_{SAT}} e^w\right)/U}{\left(\sum_{W_j\in\Omega^+_M}\left(\prod_{w:R\in F^{W_j}_{SAT}} e^w\right)\right)/U}\\
&= \frac{\prod_{w:R\in F^W_{SAT}} e^w}{\left(\sum_{W_j\in\Omega^+_M}\prod_{w:R\in F^{W_j}_{SAT}} e^w\right)}\\
&= \frac{exp(\sum_{w:R\in F^W_{SAT}} w)}{\sum_{W_j\in\Omega^+_M} exp(\sum_{w:R\in F^{W_j}_{SAT}} e)}
\end{aligned} \tag{212}$$

Let $W_I = \{W_i | i\in I\}$ We next show

$$W_I = \Omega^+_M \tag{213}$$

Let O be a member of Ω^+_M, that is, O is a probabilistic stable model of M. Let r denote $|H^W_{SAT}|$. Suppose O satisfies $r_1 < r$ hard rules of M. By Proposition 1 we have $P_M(O) = 0$, which is a contradiction. Suppose now O satisfies $r_2 > r$ hard rules of M. In this case, again, by Proposition 1 we have $P_M(W) = 0$, which contradicts the fact that W is a probabilistic stable model of M.

Therefore, O satisfies exactly r hard rules of M and

$$\Omega^+_M \subseteq W_I \tag{214}$$

Let now H be a member of W_I, that is, H is a possible world of M which satisfies exactly r hard rules of M. Clearly, there does not exist a possible world H_1 of M satisfying more than r hard rules of M (since in this case $P_M(W) = 0$ by Proposition 1). In other words, every stable model of M satisfies at most r hard rules of M. Then H is a probabilistic stable model of M by Proposition 3 and we have

$$W_I \subseteq \Omega^+_M \tag{215}$$

From (214) and (215) we have (213). From (213), (212) and (208) we have (206). □