On the Relationship between Hybrid Probabilistic Logic Programs and Stochastic Satisfiability

Emad Saad
College of Engineering and Computer Science
Abu Dhabi University
Abu Dhabi, UAE
emad.saad@adu.ac.ae

Abstract

In this paper we study the relationship between Stochastic Satisfiability (SSAT) (Papadimitriou 1985; Littman, Majercik, & Pitassi 2001) and Extended Hybrid Probabilistic Logic Programs (EHPP) with probabilistic answer set semantics (Saad 2006). We show that any instance of SSAT can be modularly translated into an EHPP program with probabilistic answer set semantics. In addition, we show that there is no modular mapping from EHPP to SSAT. This shows that EHPP is more expressive than SSAT from the knowledge representation point of view. Moreover, we show that the translation in the other way around from a program in EHPP to SSAT is more involved. We show that not every program in EHPP can be translated into an SSAT instance, rather a restricted class of EHPP can be translated into SSAT.

1 Introduction

Hybrid Probabilistic Logic Programs (HPP) (Saad & Pontelli 2006) modifies the original Hybrid Probabilistic Logic Programming framework of (Dekhtyar & Subrahmanian 2000) and generalizes and modifies the *probabilistic annotated logic programming framework*, originally proposed in (Ng & Subrahmanian 1992) and further extended in (Ng & Subrahmanian 1993; 1994). Probabilities in (Saad & Pontelli 2006) are presented in form of intervals where a probability interval represents the bounds on the degree of belief a rational agent has about the truth of an event. The semantics of HPP (Saad & Pontelli 2006), intuitively, captures the probabilistic reasoning according to how likely are the various events to occur. It was shown that the HPP (Saad & Pontelli 2006) framework is more suitable for reasoning and decision making tasks, including those arising from planning under probabilistic uncertainty (Saad 2007). In addition, it subsumes Lakshmanan and Sadri's (Lakshmanan & Sadri 2001) probabilistic implication-based framework as well as it is a natural extension of classical logic programming with answer set semantics. As a step towards enhancing its reasoning capabilities, the framework of HPP was extended to cope with non-monotonic negation (Saad & Pontelli 2005) by introducing the notion of Normal Hybrid Probabilistic Logic Programs (NHPP) and providing two different semantics namely; stable probabilistic model semantics and well-founded probabilistic model semantics. Furthermore, NHPP was extended to Extended Hybrid Probabilistic Logic Programs (EHPP) (Saad 2006) to cope directly with classical negation as well as non-monotonic negation to allow reasoning in the presence of incomplete knowledge. It was shown that Baral et al's probabilistic logic programming approach for reasoning with causal Bayes networks (P-log) (Baral, Gelfond, & Rushton 2004) is naturally subsumed by EHPP (Saad 2006). In addition, the semantics of EHPP is a natural extension to the answer set semantics of extended logic programs (Gelfond & Lifschitz 1991).

Stochastic Satisfiability (SSAT) was first introduced in (Papadimitriou 1985) as an extension to SAT with random quantifiers, in addition to the existential quantifiers. The introduction of randomized quantifiers in SSAT brings uncertainty into the question of whether there is a satisfying assignment to a propositional formula. In (Littman, Majercik, & Pitassi 2001), SSAT has been extended to allow existential, randomized, and universal quantifiers. Moreover, SSAT solver has been presented (Littman, Majercik, & Pitassi 2001) that extends Davis-Putnam-Lognmann-Loveland (DPLL) algorithm (Davis, Logemann, & Loveland 1962) to solve SSAT instances. The extended DPLL algorithm (Littman, Majercik, & Pitassi 2001) has been built by exploiting the existing work to solve SAT as efficiently as possible.

In this paper we study the relationship between Extended Hybrid Probabilistic Logic Programs (EHHP) and Stochastic Satisfiability (SSAT). We show that any SSAT formula can be easily reduced to an EHPP program, with probabilistic answer set semantics, using a local modular mapping. The importance of that is the application of SSAT to probabilistic planning, contingent probabilistic planning, the most probable explanation in belief networks, the most likely trajectory in probabilistic planning, and belief inference (Majercik & Littman 1998; 2003; Littman, Majercik, & Pitassi 2001) carry over to EHPP. This shows that EHPP is applicable to a variety of *fundamental* probabilistic reasoning tasks including those solved by SSAT. Moreover, we show that there is no similar local and modular mapping from EHPP to SSAT implying that EHPP is more expressive than SSAT from the knowledge representation point of view.

Moreover, we show that, in general, any EHPP program cannot be translated into SSAT. However, there is a class of EHPP that can be translated into SSAT, namely

$EHPP_{SSAT}$. This class of EHPP is expressive enough to represent and reason with a *variety* of probabilistic reasoning tasks such as probabilistic planning and Bayes networks. The importance of this translation from $EHPP_{SSAT}$ to SSAT is that it provides a foundation for an implementation for computing the probabilistic answer sets of EHPP by exploiting the existing work on SSAT with a selection from a variety of SSAT solvers.

This paper is organized as follows. Section 2 describes the syntax and the probabilistic answer set semantics of $EHPP_{SSAT}$. Section 3 reviews SSAT. Section 4 provides the translation from SSAT to $EHPP_{SSAT}$. In section 5, we introduce the translation from a restricted class of $EHPP_{SSAT}$ to SSAT. Conclusions and related work are presented in section 6.

2 Extended Hybrid Probabilistic Logic Programs ($EHPP_{SSAT}$)

In this section we define the syntax, declarative semantics, and the probabilistic answer sets semantics of $EHPP_{SSAT}$. $EHPP_{SSAT}$ is a class of EHPP (Saad 2006) that is sufficient to represent any instance of SSAT. The syntax and semantics of the full version of EHPP is described in (Saad 2006).

2.1 Language Syntax

Let $C[0,1]$ denotes the set of all closed intervals in $[0,1]$. In the context of $EHPP_{SSAT}$, probabilities are assigned to events (literals) as intervals in $C[0,1]$. Let $[\alpha_1,\beta_1],[\alpha_2,\beta_2] \in C[0,1]$. Then the *truth order* asserts that $[\alpha_1,\beta_1] \leq_t [\alpha_2,\beta_2]$ iff $\alpha_1 \leq \alpha_2$ and $\beta_1 \leq \beta_2$. Let $\mathcal{L}$ be an arbitrary first-order language with finitely many predicate symbols, constants, and infinitely many variables. Function symbols are disallowed. The Herbrand base of $\mathcal{L}$ is denoted by $B_{\mathcal{L}}$. A literal is either an atom a or the negation of an atom $\neg a$, where $\neg$ is the classical negation. We denote the set of all literals in $\mathcal{L}$ by Lit. An *annotation* denotes a probability interval in $C[0,1]$. An *annotated literal* is an expression of the form $l : \mu$, where l is a literal and μ is an annotation. An extended probabilistic rule (E-rule) is an expression of the form

$$\begin{array}{lll} l : \mu & \leftarrow & l_1 : \mu_1, \ldots, l_m : \mu_m, \\ & & not\ (l_{m+1} : \mu_{m+1}), \ldots, not\ (l_n : \mu_n) \end{array}$$

where l, l_i $(1 \leq i \leq n)$ are literals, and μ, μ_i $(1 \leq i \leq n)$ are annotations. The intuitive meaning of an E-rule is that, if for each $l_i : \mu_i$ $(1 \leq i \leq m)$, l_i is true with probability interval at least μ_i and for each $not\ (l_j : \mu_j)$ $(m+1 \leq j \leq n)$, it is not *known* that l_j is true with probability interval at least μ_j, then l is true with probability interval μ. An extended probabilistic logic program (*E-program*) is a pair $P = \langle R, \tau \rangle$, where R is a finite set of E-rules and τ is a mapping $\tau : Lit \rightarrow c_{pcd}$. c_{pcd} is the disjunctive positive correlation probabilistic composition function defined as $c_{pcd}([\alpha_1,\beta_1],[\alpha_2,\beta_2]) = [\max(\alpha_1,\alpha_2), \max(\beta_1,\beta_2)]$. The mapping τ in the above definition associates to each literal l the disjunctive positive correlation probabilistic composition function, c_{pcd}, that will be used to combine the probability intervals obtained from different E-rules having l in their heads. An E-program is ground if no variables appear in any of its rules.

2.2 Satisfaction and Models

A probabilistic interpretation (p-interpretation) is a mapping $h : Lit \rightarrow C[0,1]$. We say a set C, a subset of Lit, is a consistent set of literals if there is no pair of complementary literals a and $\neg a$ belonging to C. A partial or total p-interpretation h is a mapping from a consistent set of literals C to $C[0,1]$.

Definition 1 (Probabilistic Satisfaction) *Let $P = \langle R, \tau \rangle$ be a ground E-program, h be a p-interpretation, and*

$$\begin{array}{lll} r \equiv l : \mu & \leftarrow & l_1 : \mu_1, \ldots, l_m : \mu_m, \\ & & not\ (l_{m+1} : \mu_{m+1}), \ldots, not\ (l_n : \mu_n). \end{array}$$

Then

1. *h satisfies $l_i : \mu_i$ (denoted by $h \models l_i : \mu_i$) iff $l_i \in dom(h)$ and $\mu_i \leq_t h(l_i)$.*
2. *h satisfies $not\ (l_j : \mu_j)$ (denoted by $h \models not\ (l_j : \mu_j)$) iff $l_j \in dom(h)$ and $\mu_j \not\leq_t h(l_j)$ or $l_j \notin dom(h)$.*
3. *h satisfies $Body \equiv l_1 : \mu_1, \ldots, l_m : \mu_m, not\ (l_{m+1} : \mu_{m+1}), \ldots, not\ (l_n : \mu_n)$ (denoted by $h \models Body$) iff $\forall (1 \leq i \leq m), h \models l_i : \mu_i$ and $\forall (m+1 \leq j \leq n), h \models not\ (l_j : \mu_j)$.*
4. *h satisfies $l : \mu \leftarrow Body$ iff $h \models l : \mu$ or h does not satisfy $Body$.*
5. *h satisfies P iff h satisfies every E-rule in R and for every literal $l \in dom(h)$, $c_{pcd}\{\!\{\mu | l : \mu \leftarrow Body \in R \text{ and } h \models Body\}\!\} \leq_t h(l)$.*

Definition 2 (Models) *Let P be an E-program. A probabilistic model (*p-model*) of P is a p-interpretation h of P that satisfies P.*

Given the p-models h_1 and h_2, we say $h_1 \leq_o h_2$ if $dom(h_1) \subseteq dom(h_2)$ and $\forall l \in dom(h_1),\ h_1(l) \leq_t h_2(l)$. We say that h is a minimal p-model of P if there is no p-model h' of P such that $h' <_o h$.

2.3 The Probabilistic Answer Set Semantics of E-programs

An E-program without non-monotonic negation is simpler and has exactly one minimal p-model (*probabilistic answer set*) (Saad 2006). The *probabilistic answer sets* of E-programs is defined in two steps. First, we guess a probabilistic answer set h for a certain E-program P, then we define the notion of the probabilistic reduct of P with respect to h. The probabilistic reduct is an E-program without non-monotonic negation which has a unique probabilistic answer set. Second, we determine whether h is a probabilistic answer set for P. This is verified by determining whether h is the probabilistic answer set of the probabilistic reduct of P w.r.t. h.

Definition 3 (Probabilistic Reduct) *Let $P = \langle R, \tau \rangle$ be a ground E-program and h be a p-interpretation. The probabilistic reduct P^h of P w.r.t. h is $P^h = \langle R^h, \tau \rangle$ where:*

$$R^h = \left\{ \begin{array}{l} l : \mu \leftarrow l_1 : \mu_1, \ldots, l_m : \mu_m \mid \\ \quad l : \mu \leftarrow l_1 : \mu_1, \ldots, l_m : \mu_m, \\ \quad not\ (l_{m+1} : \mu_{m+1}), \ldots, not\ (l_n : \mu_n) \in R\ and \\ \quad \forall (m+1 \leq j \leq n),\ \mu_j \not\leq_t h(l_j)\ or\ l_j \notin dom(h) \end{array} \right\}$$

The probabilistic reduct P^h is an E-program without non-monotonic negation. Therefore, its probabilistic answer set is well-defined. For any $not\ (l_j : \mu_j)$ in the body of $r \in R$ with $\mu_j \not\leq_t h(l_j)$ means that it is not known that the probability interval of l_j is at least μ_j given the available knowledge, and $not\ (l_j : \mu_j)$ is removed from the body of r. In addition, if $l_j \notin dom(h)$, i.e., l_j is undefined in h, then it is completely *not known (undecidable)* that the probability interval of l_j is at least μ_j. In this case, $not\ (l_j : \mu_j)$ is also removed from the body of r.

Definition 4 *A p-interpretation h is a probabilistic answer set of an E-program P if h is the probabilistic answer set of P^h.*

3 Stochastic Satisfiability

In this section we review the definition of stochastic satisfiability presented in (Papadimitriou 1985; Littman, Majercik, & Pitassi 2001). Stochastic satisfiability (SSAT) (Papadimitriou 1985) extends deterministic satisfiability with random quantifiers. Let $\mathbf{x} = \{x_1, \ldots, x_n\}$ be a set of n propositional variables (1 for true and 0 for false) and $\phi(\mathbf{x})$ be a k-CNF propositional formula on the variables in $\mathbf{x}$, with the underlying ordering $x_1, \ldots, x_n$. An assignment $\mathbf{A}$ of propositional variables to values from $\{true, false\}$ is said to be a satisfying assignment (model) to a formula $\phi(\mathbf{x})$ if $\phi(\mathbf{A})$ evaluates to true, otherwise, $\mathbf{A}$ is said to be unsatisfying. Formally, an SSAT formula contains both existential and randomized quantifiers and takes the form

$$\exists x_1, \text{Ɐ} y_1, \ldots, \exists x_n, \text{Ɐ} y_n\ (E[\phi(\mathbf{x})] \geq \theta).$$

The SSAT decision problem determines that, given a formula $\phi(\mathbf{x})$, if there exists a value for x_1 such that for random values (true or false with equal probability) of $y_1, \ldots,$ there exists a value for x_n such that for random values of y_n, such that the expected probability of satisfying the formula $\phi(\mathbf{x})$ is at least a probability threshold θ, where $0 \leq \theta \leq 1$. An SSAT formula (Littman, Majercik, & Pitassi 2001) can be represented as a triple $\langle \phi, \theta, Q \rangle$, where ϕ is a CNF formula over the variables $x_1, \ldots, x_n$, $0 \leq \theta \leq 1$, and Q is the mapping $Q : \mathbf{x} \rightarrow \{\exists, \text{Ɐ}\}$. The evaluation of an SSAT formula, $\langle \phi, \theta, Q \rangle$, is inductively defined on the number of quantifiers to determine the expected probability of satisfying the formula ϕ. Assume x_1 is the variable associated with the leftmost quantifier. The expected probability of satisfying ϕ, under Q, denoted by $val(\phi, Q)$, is inductively defined as:

- $val(\phi, Q) = 0.0$ if ϕ contains an empty clause.
- $val(\phi, Q) = 1.0$ if ϕ does not contain clauses.
- $val(\phi, Q) = \max(val(\phi\lceil_{x_1=0}, Q), val(\phi\lceil_{x_1=1}, Q))$ if $Q(x_1) = \exists$.
- $val(\phi, Q) = (val(\phi\lceil_{x_1=0}, Q) + val(\phi\lceil_{x_1=1}, Q))/2$ if $Q(x_1) = \text{Ɐ}$.

where $\phi\lceil_{x_i=b}$ is the (n-1)-variable CNF formula produced from the n-variable formula ϕ after assigning the variable x_i the value $b \in \{true, false\}$ and simplifying the outcome, in addition to, making any required variable renumbering. Given, an SSAT formula, $\langle \phi, \theta, Q \rangle$, we say $\exists x_1, \text{Ɐ} y_1, \ldots, \exists x_n, \text{Ɐ} y_n\ (E[\phi(\mathbf{x})] \geq \theta)$ is true (satisfied) if and only if $val(\phi, Q) \geq \theta$.

If $Q(x_1) = \text{Ɐ}$, then the probability that x_1 evaluates to true leads to a satisfying formula ϕ is equally likely to the probability that x_1 evaluates to false leads to a satisfying ϕ, i.e., both have probability equal to 0.5. However, this is not necessary. A randomly quantified variable can take the value true or false with different probabilities. $\text{Ɐ}^p x_1$ is used to represent that the random variable x_1 is true with probability p, which implies that the probability that x_1 is false is $1 - p$. Consequently, if $Q(x_1) = \text{Ɐ}^p$, $val(\phi, Q)$ becomes $val(\phi, Q) = val(\phi\lceil_{x_1=0}, Q) \times (1-p) + val(\phi\lceil_{x_1=1}, Q)) \times p$.

As pointed in (Littman, Majercik, & Pitassi 2001), many decision problems can be reduced to special cases of SSAT. The satisfiability problem (SAT), can be expressed as an instance of SSAT by allowing only existential quantifiers and setting $\theta = 1$ as: $\exists x_1, \ldots, \exists x_n\ (E[\phi(\mathbf{x})] = \mathbf{1})$. Another problem, MAJSAT, asks if the satisfying assignments of a CNF formula $\phi(\mathbf{x})$ is at least half of the possible assignments to $\phi(\mathbf{x})$. MAJSAT can be represented as an instance of SSAT of the form $\text{Ɐ} x_1, \ldots, \text{Ɐ} x_n\ (E[\phi(\mathbf{x})] \geq \mathbf{\frac{1}{2}})$. SAT and MAJSAT can be combined together to form E-MAJSAT (Littman, Majercik, & Pitassi 2001) which takes the form $\exists x_1, \ldots, \exists x_m, \text{Ɐ} x_{m+1}, \ldots, \text{Ɐ} x_n\ (E[\phi(\mathbf{x})] \geq \theta)$. E-MAJSAT asks wether there is an assignment to $x_1, \ldots, x_m$ so that the combined probability of a satisfying assignment of $\phi(\mathbf{x})$ with random variables $x_{m+1}, \ldots, x_n$ is at least θ.

4 Stochastic Satisfiability as $EHPP_{SSAT}$

In this section we show that any SSAT formula, $\langle \phi(\mathbf{x}), \theta, \mathbf{Q} \rangle$, can be modularly translated into an E-program in $EHPP_{SSAT}$ whose probabilistic answer sets correspond to the models of $\phi(\mathbf{x})$. Moreover, we show that SAT, MAJSAT, and E-MAJSAT, which are instances of SSAT, can be mapped to $EHPP_{SSAT}$. These translations are mainly adapted from (Niemela 1999).

4.1 SAT as $EHPP_{SSAT}$

Any SAT formula, $\exists x_1, \ldots, \exists x_n\ (E[\phi(\mathbf{x})] = \mathbf{1})$, can be translated into an E-program, $P = \langle R, \tau \rangle$, where R is a set of E-rules consist of only atoms of the form

$$\begin{array}{lll} A : [1,1] & \leftarrow & A_1 : [1,1], \ldots, A_m : [1,1], \\ & & not\ (A_{m+1} : [1,1]), \ldots, not\ (A_n : [1,1]) \end{array}$$

where $A, A_1, \ldots, A_n$ are atoms and $[1,1]$ represents the truth value *true*. The translation proceeds as follows:

1. For each existentially quantified variable x that appears in $\phi(\mathbf{x})$, we provide two atoms x and $\overline{x}$ and include in R the E-rules

$$x : [1,1] \leftarrow not(\overline{x} : [1,1]) \qquad \overline{x} : [1,1] \leftarrow not(x : [1,1])$$

where $x : [1,1]$ corresponds to the fact that x is true, however, $\overline{x} : [1,1]$ means that the negation of x ($\neg x$) is true or x is false.

2. For each clause c in $\phi(\mathbf{x})$ and for each variable l in c, if $l = x$, then $c : [1,1] \leftarrow x : [1,1]$ is included in R. Otherwise, if $l = \neg x$, then R includes
$$c : [1,1] \leftarrow \overline{x} : [1,1].$$
3. For each clause c in $\phi(\mathbf{x})$, we include in R
$$inconsistent : [1,1] \quad \leftarrow \quad not(inconsistent : [1,1]), \\ not(c : [1,1])$$
where $inconsistent$ is a special atom that does not appear in $\phi(\mathbf{x})$.

Proposition 1 *Let $\mathcal{S}$ be a SAT formula and $P = \langle R, \tau \rangle$ be the E-program translation of $\mathcal{S}$. Then, $\mathcal{S}$ has a model iff P has a probabilistic answer set.*

Example 1 *Let $\mathcal{S}$ be a SAT formula of the form $\exists x, \exists y (E[(x \vee \neg y) \wedge (\neg x \vee y)] = 1)$. The E-program translation, $P = \langle R, \tau \rangle$, of $\mathcal{S}$ consists of the following E-rules, R,*

$$\begin{array}{ll} x : [1,1] \leftarrow not\ (\overline{x} : [1,1]) & \overline{x} : [1,1] \leftarrow not\ (x : [1,1]) \\ y : [1,1] \leftarrow not\ (\overline{y} : [1,1]) & \overline{y} : [1,1] \leftarrow not\ (y : [1,1]) \\ c_1 : [1,1] \leftarrow x : [1,1] & c_1 : [1,1] \leftarrow \overline{y} : [1,1] \\ c_2 : [1,1] \leftarrow \overline{x} : [1,1] & c_2 : [1,1] \leftarrow y : [1,1] \end{array}$$

$$inconsistent : [1,1] \quad \leftarrow \quad not(inconsistent : [1,1]), \\ not(c_i : [1,1])$$

where $1 \leq i \leq 2$. P has two probabilistic answer sets h_1 and h_2, where $h_1(\overline{x}) = [1,1]$, $h_1(\overline{y}) = [1,1]$, $h_1(c_1) = [1,1]$, $h_1(c_2) = [1,1]$, and $h_2(x) = [1,1]$, $h_2(y) = [1,1]$, $h_2(c_1) = [1,1]$, $h_2(c_2) = [1,1]$.

h_1 implies that $\neg x$ and $\neg y$, as well as, the clauses c_1 and c_2 are true in h_1. Furthermore, h_2 means that x, y, c_1, c_2 are true in h_2. Notice that $\mathcal{S}$ has two models $s_1 = \{\neg x, \neg y\}$, which implies that x and y are false in s_1, and $s_2 = \{x, y\}$, which means that x and y are true in s_2. This implies that there is a one-to-one correspondence between the probabilistic answer sets of P and the models of $\mathcal{S}$, since, s_1 corresponds to h_1 and s_2 corresponds to h_2.

4.2 MAJSAT as $EHPP_{SSAT}$

Let $\mathcal{S}$ be a MAJSAT formula of the form $\text{Ɐ}^{p_1} x_1, \ldots, \text{Ɐ}^{p_n} x_n \quad (E[\phi(\mathbf{x})] \geq \frac{1}{2})$, where all variables appear in $\phi(\mathbf{x})$ are randomly quantified. We say $\mathcal{S}$ is satisfied iff $val(\phi, Q) \geq \frac{1}{2}$. $\mathcal{S}$ can be translated into an E-program, $P = \langle R, \tau \rangle$, where R is a set of E-rules consist of only atoms. The translation proceeds as follows:

1. For each randomly quantified variable x that appears in $\phi(\mathbf{x})$, with $Q(x) = \text{Ɐ}^p$, we provide two atoms x and $\overline{x}$ and include in R the E-rules
$$\begin{array}{lll} x : [p,p] & \leftarrow & not(\overline{x} : [1-p, 1-p]) \\ \overline{x} : [1-p, 1-p] & \leftarrow & not(x : [p,p]) \end{array}$$
where $x : [p,p]$ encodes the probability of x being true is p and $\overline{x} : [1-p, 1-p]$ represents the probability of x being false is $1-p$. Obviously, if events are equally likely, then $p = 0.5$.

2. For each clause c in $\phi(\mathbf{x})$ and for each variable l in c, if $l = x$, then $c : [1,1] \leftarrow x : [p,p]$ is included in R. Otherwise, if $l = \neg x$, then R includes
$$c : [1,1] \leftarrow \overline{x} : [1-p, 1-p].$$
3. For each clause c in $\phi(\mathbf{x})$, we include in R
$$inconsistent : [1,1] \quad \leftarrow \quad not(inconsistent : [1,1]), \\ not(c : [1,1])$$
where $inconsistent$ is a special atom that does not appear in $\phi(\mathbf{x})$.

Theorem 1 *Let $\mathcal{S} = \langle \phi(\mathbf{x}), \frac{1}{2}, \mathbf{Q} \rangle$ be a MAJSAT formula, $P = \langle R, \tau \rangle$ be the E-program translation of $\mathcal{S}$, and Ans be the set of all probabilistic answer sets of P. Then, $\phi(\mathbf{x})$ has a model iff P has a probabilistic answer set, and $\mathcal{S}$ is satisfied iff $\sum_{h \in Ans} \prod_{x_i \in dom(h)} h(x_i) = val(\phi(\mathbf{x}), \mathbf{Q}) \geq \frac{1}{2}$.*

Example 2 *Let $\mathcal{S}$ be a MAJSAT formula of the form $\text{Ɐ} x, \text{Ɐ} y (E[(x \vee \neg y) \wedge (\neg x \vee y)] \geq \frac{1}{2})$. The E-program translation, $P = \langle R, \tau \rangle$, of $\mathcal{S}$ consists of the following E-rules, R,*

$$\begin{array}{ll} x : \nu \leftarrow not\ (\overline{x} : \nu) & \overline{x} : \nu \leftarrow not\ (x : \nu) \\ y : \nu \leftarrow not\ (\overline{y} : \nu) & \overline{y} : \nu \leftarrow not\ (y : \nu) \\ c_1 : [1,1] \leftarrow x : \nu & c_1 : [1,1] \leftarrow \overline{y} : \nu \\ c_2 : [1,1] \leftarrow \overline{x} : \nu & c_2 : [1,1] \leftarrow y : \nu \end{array}$$

$$inconsistent : [1,1] \quad \leftarrow \quad not(inconsistent : [1,1]), \\ not(c_i : [1,1])$$

where $1 \leq i \leq 2$ and $\nu \equiv [0.5, 0.5]$. Clearly, $\mathcal{S}$ is satisfied, since $val(((x \vee \neg y) \wedge (\neg x \vee y)), Q) = \frac{1}{2} \geq \frac{1}{2}$. On the other hand, P has two probabilistic answer sets h_1 and h_2, where $h_1(\overline{x}) = [0.5, 0.5]$, $h_1(\overline{y}) = [0.5, 0.5]$, $h_1(c_1) = [1,1]$, $h_1(c_2) = [1,1]$, and $h_2(x) = [0.5, 0.5]$, $h_2(y) = [0.5, 0.5]$, $h_2(c_1) = [1,1]$, $h_2(c_2) = [1,1]$, and hence, $\sum_{h \in Ans} \prod_{x_i \in dom(h)} h(x_i) = h_1(\overline{x}) \times h_1(\overline{y}) + h_2(x) \times h_2(y) = 0.5 = val(\phi(\mathbf{x}), \mathbf{Q}) \geq \frac{1}{2}$.

Moreover, $((x \vee \neg y) \wedge (\neg x \vee y))$ has two models $s_1 = \{\neg x, \neg y\}$ and $s_2 = \{x, y\}$. This implies that there is a one-to-one correspondence between the probabilistic answer sets of P and the models of $((x \vee \neg y) \wedge (\neg x \vee y))$, since s_1 corresponds to h_1 and s_2 corresponds to h_2.

4.3 E-MAJSAT as $EHPP_{SSAT}$

Let $\mathcal{S}$ be an E-MAJSAT formula of the form $\exists x_1, \ldots, \exists x_n, \text{Ɐ}^{p_1} y_1, \ldots, \text{Ɐ}^{p_n} y_n \quad (E[\phi(\mathbf{x})] \geq \theta)$, where a sequence of existentially quantified variables, x_i ($1 \leq i \leq n$), are followed by a sequence of randomly quantified variables, y_i ($1 \leq i \leq n$). Similarly, we say that an E-MAJSAT formula $\mathcal{S}$ is satisfied iff $val(\phi, Q) \geq \theta$. Since E-MAJSAT combines both SAT and MAJSAT together, a translation form E-MAJSAT to an E-program combines the SAT and MAJSAT translations to E-programs together. $\mathcal{S}$ can be translated into an E-program, $P = \langle R, \tau \rangle$, where R is a set of E-rules consist of only atoms. The translation proceeds as follows:

1. For each existentially quantified variable x that appears in $\phi(\mathbf{x})$, we provide two atoms x and $\overline{x}$ and include in R the E-rules

$$x : [1,1] \leftarrow not(\overline{x} : [1,1]) \qquad \overline{x} : [1,1] \leftarrow not(x : [1,1])$$

2. For each randomly quantified variable y that appears in $\phi(\mathbf{x})$, with $Q(y) = \forall^p$, we provide two atoms y and $\overline{y}$ and include in R the E-rules

$$\begin{array}{lcl} y : [p,p] & \leftarrow & not(\overline{y} : [1-p, 1-p]) \\ \overline{y} : [1-p, 1-p] & \leftarrow & not(y : [p,p]) \end{array}$$

3. For each clause c in $\phi(\mathbf{x})$ and for each variable l in c, if $l = x$, with $Q(x) = \exists$, then $c : [1,1] \leftarrow x : [1,1]$ is included in R. Otherwise, if $l = \neg x$, then R includes $c : [1,1] \leftarrow \overline{x} : [1,1]$.

4. For each clause c in $\phi(\mathbf{x})$ and for each variable l in c, if $l = y$, with $Q(y) = \forall^p$, then $c : [1,1] \leftarrow y : [p,p]$ is included in R. Otherwise, if $l = \neg y$, then R includes $c : [1,1] \leftarrow \overline{y} : [1-p, 1-p]$.

5. For each clause c in $\phi(\mathbf{x})$, we include in R

$$\begin{array}{lcl} inconsistent : [1,1] & \leftarrow & not(inconsistent : [1,1]), \\ & & not(c : [1,1]) \end{array}$$

where $inconsistent$ is a special atom that does not appear in $\phi(\mathbf{x})$.

Theorem 2 *Let $\mathcal{S} = \langle \phi(\mathbf{x}), \theta, \mathbf{Q} \rangle$ be an E-MAJSAT formula, $P = \langle R, \tau \rangle$ be the E-program translation of $\mathcal{S}$, Ans be the set of all probabilistic answer sets of P, and $h, h' \in Ans$ be probabilistic answer sets of P. Then, $\phi(\mathbf{x})$ has a model iff P has a probabilistic answer set, and $\mathcal{S}$ is satisfied iff*

$$\max_{h \models \widetilde{x}_1 : [1,1], \ldots, \widetilde{x}_n : [1,1]} \left[h(\widetilde{x}_n) \sum_{h' \models D} \prod_{i=1}^{n} h'(\widetilde{y}_i) \right] \geq \theta.$$

where $D \equiv \widetilde{x}_1 : [1,1], \ldots, \widetilde{x}_n : [1,1], \widetilde{y}_1 : [p_1, p_1], \ldots, \widetilde{y}_n : [p_n, p_n]$ and $\widetilde{x}_i = x_i$ or $\widetilde{x}_i = \neg x_i$ and $\widetilde{y}_i = y_i$ or $\widetilde{y}_i = \neg y_i$.

Intuitively, in the expression of Theorem 2, the maximum is taken over all the possible assignments to the existentially quantified variables. For a given assignment to the existentially quantified variables, $\widetilde{x}_1, \ldots, \widetilde{x}_n$, a summation is taken over the product of probabilities associated with all randomly quantified variables in each satisfying assignment to $\phi(\mathbf{x})$, of the form $\widetilde{x}_1, \ldots, \widetilde{x}_n, \widetilde{y}_1, \ldots, \widetilde{y}_n$, that contains $\widetilde{x}_1, \ldots, \widetilde{x}_n$. This satisfying assignment corresponds to a probabilistic answer set h' of P.

Example 3 *Let $\mathcal{S}$ be an E-MAJSAT formula of the form*

$$\exists x, \forall y (E[(x \vee \neg y) \wedge (\neg x \vee y)] \geq 0.75).$$

The E-program, $P = \langle R, \tau \rangle$, translation of $\mathcal{S}$ consists of the following E-rules, R,

$$\begin{array}{ll} x : [1,1] \leftarrow not\ (\overline{x} : [1,1]) & \overline{x} : [1,1] \leftarrow not\ (x : [1,1]) \\ y : \nu \leftarrow not\ (\overline{y} : \nu) & \overline{y} : \nu \leftarrow not\ (y : \nu) \\ c_1 : [1,1] \leftarrow x : [1,1] & c_1 : [1,1] \leftarrow \overline{y} : \nu \\ c_2 : [1,1] \leftarrow \overline{x} : [1,1] & c_2 : [1,1] \leftarrow y : \nu \end{array}$$

$$\begin{array}{lcl} inconsistent : [1,1] & \leftarrow & not(inconsistent : [1,1]), \\ & & not(c_i : [1,1]) \end{array}$$

where $1 \leq i \leq 2$ and $\nu \equiv [0.5, 0.5]$. It can be easily verified that $\mathcal{S}$ is unsatisfied, since $val(((x \vee \neg y) \wedge (\neg x \vee y)), Q) = 0.5 \not\geq 0.75$. On the other hand, P has two probabilistic answer sets h_1 and h_2, where $h_1(\overline{x}) = [1,1]$, $h_1(\overline{y}) = [0.5, 0.5]$, $h_1(c_1) = [1,1]$, $h_1(c_2) = [1,1]$, and $h_2(x) = [1,1]$, $h_2(y) = [0.5, 0.5]$, $h_2(c_1) = [1,1]$, $h_2(c_2) = [1,1]$, and hence, $\max_{h_1 \models \overline{x}:[1,1], h_2 \models x:[1,1]} [h_1(\overline{x}) \times h_1(y), h_2(x) \times h_2(\overline{y})] = 0.5 = val(\phi(\mathbf{x}), Q) \not\geq 0.75$. Moreover, $((x \vee \neg y) \wedge (\neg x \vee y))$ has two models $s_1 = \{\neg x, \neg y\}$ and $s_2 = \{x, y\}$. This implies that there is a one-to-one correspondence between the probabilistic answer sets of P and the models of $((x \vee \neg y) \wedge (\neg x \vee y))$, since s_1 corresponds to h_1 and s_2 corresponds to h_2.

The translation from a general SSAT formula, where existentially quantified variables alternating with randomly quantified variables, is the same as the translation from an E-MAJSAT formula to an E-program. Then, the following proposition directly follows.

Proposition 2 *Let $\mathcal{S}$ be an SSAT formula of the form $\exists x_1, \forall y_1, \ldots, \exists x_n, \forall y_n \ (E[\phi(\mathbf{x})] \geq \theta)$ and $P = \langle R, \tau \rangle$ be the E-program translation of $\mathcal{S}$. Then, $\phi(\mathbf{x})$ has a model iff P has a probabilistic answer set.*

Example 4 *Consider the following SSAT formula, $\mathcal{S}$, from (Littman, Majercik, & Pitassi 2001), where $\mathcal{S}$ of the form*

$$\forall x, \exists y, \forall z (E[(x \vee y) \wedge (y \vee \neg z) \wedge (\neg x \vee \neg y)] \geq 0.5).$$

The E-program, $P = \langle R, \tau \rangle$, translation of $\mathcal{S}$ consists of the following E-rules, R,

$$\begin{array}{ll} x : \nu \leftarrow not\ (\overline{x} : \nu) & \overline{x} : \nu \leftarrow not\ (x : \nu) \\ y : [1,1] \leftarrow not\ (\overline{y} : [1,1]) & \overline{y} : [1,1] \leftarrow not\ (y : [1,1]) \\ z : \nu \leftarrow not\ (\overline{z} : \nu) & \overline{z} : \nu \leftarrow not\ (z : \nu) \\ c_1 : [1,1] \leftarrow x : \nu & c_1 : [1,1] \leftarrow y : [1,1] \\ c_2 : [1,1] \leftarrow y : [1,1] & c_2 : [1,1] \leftarrow \overline{z} : \nu \\ c_3 : [1,1] \leftarrow \overline{x} : \nu & c_3 : [1,1] \leftarrow \overline{y} : [1,1] \end{array}$$

$$\begin{array}{lcl} inconsistent : [1,1] & \leftarrow & not(inconsistent : [1,1]), \\ & & not(c_i : [1,1]) \end{array}$$

where $1 \leq i \leq 3$ and $\nu \equiv [0.5, 0.5]$. It can be easily verified that $\mathcal{S}$ is satisfied, since $val(((x \vee y) \wedge (y \vee \neg z) \wedge (\neg x \vee \neg y)), Q) = 0.75 \geq 0.5$. On the other hand, P has three probabilistic answer sets h_1, h_2, and h_3, where $h_1(x) = [0.5, 0.5]$, $h_1(\overline{y}) = [1,1]$, $h_1(\overline{z}) = [0.5, 0.5]$, $h_1(c_1) = [1,1]$, $h_1(c_2) = [1,1]$, $h_1(c_3) = [1,1]$. But, $h_2(\overline{x}) = [0.5, 0.5]$, $h_2(y) = [1,1]$, $h_2(z) = [0.5, 0.5]$, $h_2(c_1) = [1,1]$, $h_2(c_2) = [1,1]$, $h_2(c_3) = [1,1]$. Finally, $h_3(\overline{x}) = [0.5, 0.5]$, $h_3(y) = [1,1]$, $h_3(\overline{z}) = [0.5, 0.5]$, $h_3(c_1) = [1,1]$, $h_3(c_2) = [1,1]$, $h_3(c_3) = [1,1]$. Moreover, $((x \vee y) \wedge (y \vee \neg z) \wedge (\neg x \vee \neg y))$ has three models $s_1 = \{x, \neg y, \neg z\}$, $s_2 = \{\neg x, y, z\}$, and $s_3 = \{\neg x, y, \neg z\}$. This implies that there is a one-to-one correspondence between the probabilistic answer sets of P and the models of $((x \vee y) \wedge (y \vee \neg z) \wedge (\neg x \vee \neg y))$, since s_1, s_2, and s_3 correspond to h_1, h_2, and h_3 respectively.

Observe that the translation from SSAT to $EHPP_{SSAT}$ is modular, since small local changes in the clauses in ϕ causes small local changes in the corresponding E-program translation. However, this is not the case in the reverse direction. There is no local modular mapping from $EHPP_{SSAT}$

to SSAT. This implies that $EHPP_{SSAT}$ is more expressive than SSAT from the knowledge representation point of view. Similar to (Niemela 1999), let, e.g., $\mathcal{M}(.)$ be a modular mapping from $EHPP_{SSAT}$ to SSAT. Let $P = \langle R, \tau \rangle$ be an E-program in $EHPP_{SSAT}$ that is modularly mapped to an SSAT formula $\mathcal{S} = \langle \mathcal{M}(R), \theta, Q \rangle$, where $\mathcal{M}(R) = \phi(\mathbf{x})$. $\mathcal{M}(.)$ is said to be modular if for each set of facts $\mathcal{F}$ that is mapped to $\mathcal{M}(\mathcal{F})$, we have $P = \langle R \cup \mathcal{F}, \tau \rangle$ has a probabilistic answer set iff $\mathcal{M}(R) \cup \mathcal{M}(\mathcal{F})$ has a model. Intuitively, adding a fact to an E-program should make a local change in the translated SSAT formula, but not require translating the entire E-program.

Proposition 3 *There is no modular mapping from $EHPP_{SSAT}$ to SSAT.*

Proof. Following a proof of a corresponding proposition in (Niemela 1999), consider the E-program $P = \langle R, \tau \rangle$ where R contains the E-rule $a : [1,1] \leftarrow not\ (a : [1,1])$. Consider also that $\mathcal{M}(.)$ is a modular mapping. It can be seen that P has no probabilistic answer sets, and hence, $\mathcal{M}(R)$ is unsatisfiable. However, $\mathcal{M}(R) \cup \mathcal{M}(\{a : [1,1] \leftarrow\})$ is unsatisfiable regardless the choice of $\mathcal{M}(\{a : [1,1] \leftarrow\})$. This implies that $P = \langle R \cup \{a : [1,1] \leftarrow\}, \tau \rangle$ has no probabilistic answer sets as well. However, this is not the case, since P has a probabilistic answer set h, where $h(a) = [1,1]$. Therefore, there does not exist any modular mapping from $EHPP_{SSAT}$ to SSAT.

5 $EHPP_{SSAT}$ as SSAT

In general, it is not possible to translate any E-program in $EHPP_{SSAT}$ or EHPP (Saad 2006) to SSAT, since $EHPP_{SSAT}$ allows probability intervals while SSAT deals with point probabilities. In addition, EHPP (Saad 2006) allows conjunctions and disjunctions of literals to appear in the body of E-rules. However, we show that there is a class of $EHPP_{SSAT}$, namely *restricted* $EHPP_{SSAT}$, that can be translated into SSAT. An E-program in *restricted* $EHPP_{SSAT}$ takes the form $P = \langle R \cup R_{neg}, \tau \rangle$, where $\tau : Lit \rightarrow c_{pcd}$ and $R \cup R_{neg}$ is a set of E-rules that satisfy the following conditions:

1. All events that appear in R are atomic events, represented as positive literals (atoms) in R.
2. All probabilities that appear in any E-rule in R are point probabilities of the form $[p,p]$.
3. If the probability of an event a is $[p,p]$, then the probability of all occurrences of a in R is $[p,p]$.
4. For any event a that appears in R, we have $Pr(a) + Pr(\neg a) = 1$.
5. For each event a that appears in R with probability $[p,p] < [1,1]$, the E-rule

$$\overline{a} : [1-p, 1-p] \leftarrow not\ (a : [p,p])$$

belongs to R_{neg}. If the probability of a is $[1,1]$, then the above E-rule is simply written as

$$\overline{a} : [1,1] \leftarrow not\ (a : [1,1]).$$

This set of E-rules, R_{neg}, is not used in the translation from P to an SSAT formula. However, E-rules in R_{neg} are used to encode the default probabilities, i.e., to encode the fact that the probability of $\neg a$ is $1 - Pr(a)$.

Observe that an E-program in restricted $EHPP_{SSAT}$ contains E-rules that consist of only atoms of the form

$$r \equiv A : \mu \quad \leftarrow \quad A_1 : \mu_1, \ldots, A_m : \mu_m, \\ not\ (A_{m+1} : \mu_{m+1}), \ldots, not\ (A_n : \mu_n)$$

where $A, A_i (1 \leq i \leq n)$ are atoms. Let $Head(r) = A$, $Pos(r) = \{A_1, \ldots, A_m\}$, and $Neg(r) = \{A_{m+1}, \ldots, A_n\}$. A positive dependency graph of an E-program, $P = \langle R \cup R_{neg}, \tau \rangle$ in restricted $EHPP_{SSAT}$, is a directed graph, G_P, such that (i) vertices of G_P are atoms appearing in R and (ii) for each E-rule r in R, there is an edge from $Head(r)$ to each atom in $Pos(r)$.

Definition 5 *An E-program P in restricted $EHPP_{SSAT}$ is tight E-program if the positive dependency graph of P is acyclic.*

5.1 Tight $EHPP_{SSAT}$ as SSAT

Any tight E-program, $P = \langle R \cup R_{neg}, \tau \rangle$ in restricted $EHPP_{SSAT}$, can be translated into an SSAT formula. The resulting SSAT formula can be viewed as SAT, MAJSAT, or E-MAJSAT, depending on the probability values that appear in R, and the type of quantifiers that we associate with each distinct variable in the resulting SSAT formula. If all probabilities that appear in R are $[1,1]$, then the resulting SSAT formula, $\mathcal{S}$, is SAT with existential quantifier associated with each variable appearing in $\mathcal{S}$. But, if all probabilities that appear in R are $[p,p] \neq [1,1]$, then the resulting formula, $\mathcal{S}$, is MAJSAT with randomized quantifier associated with each variable in $\mathcal{S}$. If the probabilities appearing in R are a combination of $[p,p]$ and $[1,1]$, then the resulting formula, $\mathcal{S}$, can be viewed as E-MAJSAT or MAJSAT, depending on how we want to view the formula. If $\mathcal{S}$ is viewed as E-MAJSAT, then an atom a in R, whose associated probability is $[1,1]$, corresponds to an existentially quantified variable in $\mathcal{S}$. However, if a is associated with probability $[p,p] \neq [1,1]$ in R, then a corresponds to a randomly quantified variable in $\mathcal{S}$ (given that all existentially quantified variables are followed by the randomly quantified ones). Let $atoms(P)$ denotes the set of atoms that appearing in R. The translation from an E-program, in restricted $EHPP_{SSAT}$, to SSAT is provided by defining the notion of *probabilistic completion* of $EHPP_{SSAT}$ adapted from (Clark 1978). The probabilistic completion of an E-program, $P = \langle R \cup R_{neg}, \tau \rangle$ in restricted $EHPP_{SSAT}$, is denoted by $Comp(P) = \langle \mathcal{R}, Q \rangle$, where:

- $\mathcal{R}$ is the set of propositional formulas formed from the E-rules in R as follows:
  - For each $A \in atoms(P)$, if

$$A : \mu \quad \leftarrow \quad A_1^i : \mu_1^i, \ldots, A_m^i : \mu_m^i, \\ not\ (A_{m+1}^i : \mu_{m+1}^i), \ldots, not\ (A_n^i : \mu_n^i)$$

  for $1 \leq i \leq k$, is the set of E-rules in R whose heads contain A, then $A \equiv Body_1 \vee \cdots \vee Body_k \in \mathcal{R}$ where $Body_i = A_1^i \wedge \ldots \wedge A_m^i \wedge \neg A_{m+1}^i \wedge \ldots \wedge \neg A_n^i$. If

$k = 0$, i.e., there is no E-rule in R whose head contains A, then $\neg A \in \mathcal{R}$.

– If R contains an E-rule of the form

$$\begin{array}{rcl} inconsistent : [1,1] & \leftarrow & not(inconsistent : [1,1]), \\ & & A_1 : \mu_1, \ldots, A_m : \mu_m, \\ & & not\ (A_{m+1} : \mu_{m+1}), \ldots, \\ & & not\ (A_n : \mu_n) \end{array}$$

then, $\neg Body \in \mathcal{R}$, where

$$Body = A_1 \wedge \ldots \wedge A_m \wedge \neg A_{m+1} \wedge \ldots \wedge \neg A_n$$

- Q is a mapping, where for each atom $A \in atoms(P)$, we have $Q(A) = \text{Я}^p$, if $A : [p,p]$ appears in any E-rule r in R (either in the head of r or in its body). Similarly, $Q(A) = \exists$, if $A : [1,1]$ appears in any E-rule r in R (if the resulting SSAT is viewed as E-MAJSAT). In the mapping Q, $Q(A) = \text{Я}^p$ says that, in the resulting SSAT formula, A is randomly quantified variable with the probability of A being true is p.

Theorem 3 *Let $P = \langle R \cup R_{neg}, \tau \rangle$ be a tight E-program in restricted $EHPP_{SSAT}$ and $Comp(P) = \langle \mathcal{R}, Q \rangle$ be the probabilistic completion of P. Then, $\mathcal{R}$ has a model iff P has a probabilistic answer set.*

Theorem 4 *Let $P = \langle R \cup R_{neg}, \tau \rangle$ be a tight E-program in restricted $EHPP_{SSAT}$ and $Comp(P) = \langle \mathcal{R}, Q \rangle$ be the probabilistic completion of P. Let Ans be the set of all probabilistic answer sets of P and $h, h' \in Ans$. Then, $\mathcal{S} = \langle \mathcal{R}, \theta, Q \rangle$ is satisfied iff $\sum_{h \in Ans} \prod_{A_i \in dom(h)} h(A_i) = val(\mathcal{R}, Q) \geq \theta$, viewing the SSAT formula, $\mathcal{S}$, as MAJSAT, where $\theta = \frac{1}{2}$, and*

$$\max_{h \models \widetilde{x}_1 : [1,1], \ldots, \widetilde{x}_n : [1,1]} \left[h(\widetilde{x}_n) \sum_{h' \models \mathcal{D}} \prod_{i=1}^{n} h'(\widetilde{y}_i) \right] \geq \theta.$$

where $\mathcal{D} \equiv \widetilde{x}_1 : [1,1], \ldots, \widetilde{x}_n : [1,1], \widetilde{y}_1 : [p_1,p_1], \ldots, \widetilde{y}_n : [p_n,p_n]$ and $\widetilde{x}_i = x_i$ or $\widetilde{x}_i = \neg x_i$ and $\widetilde{y}_i = y_i$ or $\widetilde{y}_i = \neg y_i$, viewing the SSAT formula, $\mathcal{S}$, as E-MAJSAT.

In the following examples, without loss of generality, we consider MAJSAT translation from E-programs.

Example 5 *Consider the E-program, $P = \langle R \cup R_{neg}, \tau \rangle$ in restricted $EHPP_{SSAT}$, where $R \cup R_{neg}$ contains the E-rules*

$$\begin{array}{rcl} a : [0.3, 0.3] & \leftarrow & not\ (b : [0.4, 0.4]) \\ b : [0.4, 0.4] & \leftarrow & not\ (a : [0.3, 0.3]) \\ \overline{a} : [0.7, 0.7] & \leftarrow & not\ (a : [0.3, 0.3]) \\ \overline{b} : [0.6, 0.6] & \leftarrow & not\ (b : [0.4, 0.4]) \end{array}$$

Obviously, the first two E-rules belong to R and the last two E-rules belong to R_{neg}. Clearly, P is tight. The probabilistic completion of P is $Comp(\mathcal{R}, Q)$, where $\mathcal{R} = \{a \equiv \neg b, b \equiv \neg a\} = \{(a \vee b), (\neg a \vee \neg b)\}$, and $Q(a) = \text{Я}^{0.3}$, $Q(b) = \text{Я}^{0.4}$. P has two probabilistic answer sets h_1 and h_2, where $h_1(a) = [0.3, 0.3]$, $h_1(\overline{b}) = [0.6, 0.6]$, and $h_2(\overline{a}) = [0.7, 0.7]$, $h_2(b) = [0.4, 0.4]$. In addition, $\mathcal{R} = \{(a \vee b), (\neg a \vee \neg b)\}$ has two models $s_1 = \{a, \neg b\}$ and $s_2 = \{\neg a, b\}$. This implies that there is a one-to-one correspondence between the probabilistic answer sets of P and the models of $\mathcal{R}$, since s_1 corresponds to h_1 and s_2 corresponds to h_2. It can be easily verified that the SSAT formula, $\mathcal{S} = \langle \mathcal{R}, 0.5, Q \rangle$, is unsatisfied, since $val(\mathcal{R}, Q) = 0.46 \not\geq 0.5$, in addition, we have $\sum_{h \in Ans} \prod_{x_i \in dom(h)} h(x_i) = h_1(a) \times h_1(\overline{b}) + h_2(\overline{a}) \times h_2(b) = 0.46 = val(\mathcal{R}, Q) \not\geq 0.5$.

Example 6 *Consider the E-program, $P = \langle R \cup R_{neg}, \tau \rangle$ in restricted $EHPP_{SSAT}$, where $R \cup R_{neg}$ contains the E-rules*

$$\begin{array}{rcl} a : [0.3, 0.3] & \leftarrow & not\ (b : [0.4, 0.4]) \\ \overline{a} : [0.7, 0.7] & \leftarrow & not\ (a : [0.3, 0.3]) \\ \overline{b} : [0.6, 0.6] & \leftarrow & not\ (b : [0.4, 0.4]) \end{array}$$

The first E-rule belongs to R and the last two E-rules belong to R_{neg}. Clearly, P is tight. The probabilistic completion of P is $Comp(\mathcal{R}, Q)$, where $\mathcal{R} = \{a \equiv \neg b, \neg b\} = \{(a \vee b), (\neg a \vee \neg b), (\neg b)\}$, and $Q(a) = \text{Я}^{0.3}$, $Q(b) = \text{Я}^{0.4}$. P has only one probabilistic answer set h, where $h(a) = [0.3, 0.3]$, $h(\overline{b}) = [0.6, 0.6]$. In addition, $\mathcal{R} = \{(a \vee b), (\neg a \vee \neg b), (\neg b)\}$ has only one model $s = \{a, \neg b\}$. This implies that there is a one-to-one correspondence between the probabilistic answer set of P and the model of $\mathcal{R}$, since s corresponds to h. It can be easily verified that the SSAT formula, $\mathcal{S} = \langle \mathcal{R}, 0.5, Q \rangle$, is unsatisfied, since $val(\mathcal{R}, Q) = 0.18 \not\geq 0.5$, in addition, we have $\sum_{h \in Ans} \prod_{x_i \in dom(h)} h(x_i) = h(a) \times h(\overline{b}) = 0.18 = val(\mathcal{R}, Q) \not\geq 0.5$.

Example 7 *Consider the E-program, $P = \langle R \cup R_{neg}, \tau \rangle$ in restricted $EHPP_{SSAT}$, where $R \cup R_{neg}$ contains the E-rules*

$$\begin{array}{rcl} a : [0.3, 0.3] & \leftarrow & not\ (a : [0.3, 0.3]) \\ \overline{a} : [0.7, 0.7] & \leftarrow & not\ (a : [0.3, 0.3]) \end{array}$$

The first E-rule belongs to R and the last E-rule belongs to R_{neg}. Clearly, P is tight. The probabilistic completion of P is $Comp(\mathcal{R}, Q)$, where $\mathcal{R} = \{a \equiv \neg a\}$, and $Q(a) = \text{Я}^{0.3}$. It can be easily verified that P has no probabilistic answer sets. In addition, $\mathcal{R}$ does not have any models either. This implies that there is a one-to-one correspondence between the probabilistic answer sets of P and the models of $\mathcal{R}$. Moreover, the SSAT formula, $\mathcal{S} = \langle \mathcal{R}, 0.5, Q \rangle$, is unsatisfied, since $val(\mathcal{R}, Q) = 0 \not\geq 0.5$, in addition, we have $\sum_{h \in Ans} \prod_{x_i \in dom(h)} h(x_i) = 0 = val(\mathcal{R}, Q) \not\geq 0.5$.

Example 8 *Consider the E-program, $P = \langle R \cup R_{neg}, \tau \rangle$ in restricted $EHPP_{SSAT}$, where $R \cup R_{neg}$ contains the E-rules*

$$\begin{array}{rcl} a : [0.9, 0.9] & \leftarrow & not\ (b : [0.2, 0.2]) \\ b : [0.2, 0.2] & \leftarrow & not\ (a : [0.9, 0.9]) \\ c : [1, 1] & \leftarrow & a : [0.9, 0.9] \\ c : [1, 1] & \leftarrow & b : [0.2, 0.2] \\ \overline{a} : [0.1, 0.1] & \leftarrow & not\ (a : [0.9, 0.9]) \\ \overline{b} : [0.8, 0.8] & \leftarrow & not\ (b : [0.2, 0.2]) \\ \overline{c} : [1, 1] & \leftarrow & not\ (c : [1, 1]) \end{array}$$

The first four E-rules belong to R and the last three E-rules belong to R_{neg}. Clearly, P is tight. The probabilistic completion of P is $Comp(\mathcal{R}, Q)$, where $\mathcal{R} = \{a \equiv$

$\neg b, b \equiv \neg a, c \equiv a \vee b\}$, *and* $Q(a) = \text{ꓤ}^{0.9}$, $Q(b) = \text{ꓤ}^{0.2}$, $Q(c) = \text{ꓤ}^{1}$. *P has two probabilistic answer sets h_1 and h_2, where $h_1(a) = [0.9, 0.9]$, $h_1(\bar{b}) = [0.8, 0.8]$, $h_1(c) = [1, 1]$, and $h_2(\overline{a}) = [0.1, 0.1]$, $h_2(b) = [0.2, 0.2]$, $h_2(c) = [1, 1]$. In addition, $\mathcal{R} = \{a \equiv \neg b, b \equiv \neg a, c \equiv a \vee b\}$ has two models $s_1 = \{a, \neg b, c\}$ and $s_2 = \{\neg a, b, c\}$. This implies that there is a one-to-one correspondence between the probabilistic answer sets of P and the models of $\mathcal{R}$, since s_1 corresponds to h_1 and s_2 corresponds to h_2. It can be easily verified that the SSAT formula, $\mathcal{S} = \langle \mathcal{R}, 0.5, Q \rangle$, is satisfied, since $val(\mathcal{R}, Q) = 0.74 \geq 0.5$, in addition, we have $\sum_{h \in Ans} \prod_{x_i \in dom(h)} h(x_i) = h_1(a) \times h_1(\bar{b}) \times h_1(c) + h_2(\overline{a}) \times h_2(b) \times h_2(c) = 0.74 = val(\mathcal{R}, Q) \geq 0.5$.*

5.2 Non-Tight $EHPP_{SSAT}$ as SSAT

Let $P = \langle R \cup R_{neg}, \tau \rangle$ be any E-program in restricted $EHPP_{SSAT}$ and $Comp(P) = \langle \mathcal{R}, Q \rangle$ be its probabilistic completion. It is possible to get a model of $\mathcal{R}$ that does not correspond to any probabilistic answer set of P, and hence, Theorems 3 and 4 do not apply for that E-program. This occurs for any E-program in restricted $EHPP_{SSAT}$ that is not tight. Consider the following E-program.

Example 9 *Let $P = \langle R \cup R_{neg}, \tau \rangle$ be an E-program in restricted $EHPP_{SSAT}$, where $R \cup R_{neg}$ consists of the E-rules*

$$
\begin{array}{rcl}
a : [0.5, 0.5] & \leftarrow & b : [0.3, 0.3] \\
b : [0.3, 0.3] & \leftarrow & a : [0.5, 0.5] \\
\overline{a} : [0.5, 0.5] & \leftarrow & not\ (a : [0.5, 0.5]) \\
\bar{b} : [0.7, 0.7] & \leftarrow & not\ (b : [0.3, 0.3])
\end{array}
$$

The probabilistic completion of P is $Comp(P) = \langle \mathcal{R}, Q \rangle$, where $\mathcal{R} = \{a \equiv b\}$ and $Q(a) = \text{ꓤ}^{0.5}$, $Q(b) = \text{ꓤ}^{0.3}$. This E-program, P, has only one probabilistic answer set, h, where $h(\overline{a}) = [0.5, 0.5]$ and $h(\bar{b}) = [0.7, 0.7]$ ($h(\overline{a})$ corresponds to $Pr(\neg a)$ and $h(\bar{b})$ corresponds to $Pr(\neg b)$). We have, $\sum_{h \in Ans} \prod_{A_i \in dom(h)} h(A_i) = \prod_{A_i \in dom(h)} h(A_i) = h(\overline{a}) \times h(\bar{b}) = [0.5, 0.5] \times [0.7, 0.7] = [0.35, 0.35]$. But, on the other hand, there are two models of $\mathcal{R}$ that contribute to $val(\mathcal{R}, Q)$. These models are $s_1 = \{\neg a, \neg b\}$ and $s_2 = \{a, b\}$. The probabilistic answer set h of P corresponds to the model $s_1 = \{\neg a, \neg b\}$ of $\mathcal{R}$. Given the models s_1 and s_2 of $\mathcal{R}$, it can be easily verified that $val(\mathcal{R}, Q) = [0.5, 0.5]$. This implies that $\sum_{h \in Ans} \prod_{A_i \in dom(h)} h(A_i) \neq val(\mathcal{R}, Q)$.

There is a one-to-one correspondence between the probabilistic answer sets of any tight E-program, P, in restricted $EHPP_{SSAT}$, and the models of $\mathcal{R}$ in $Comp(P) = \langle \mathcal{R}, Q \rangle$. But this is not the case for the E-program in Example 9. The reason is that this E-program, P, is not tight, since there is a cycle in the positive dependency graph of P. The set $\{a, b\}$ is a cycle (loop) in P because in the positive dependency graph of P, a depends on b from the first E-rule and b depends on a from the second E-rule. This loop does not allow us to conclude any knowledge about the probabilities of a and b using the probabilistic answer set semantics of $EHPP_{SSAT}$. However, in SSAT, assumptions can be made about the truth values and the probabilities of a and b in that loop. These loops are the reason for the existence of a model (or models) of $\mathcal{R}$ that does not correspond to any probabilistic answer set of P, and hence $\sum_{h \in Ans} \prod_{A_i \in dom(h)} h(A_i) \neq val(\mathcal{R}, Q)$. In the rest of this section, we follow the approach of (Lin & Zhao 2004) adapted to deal with $EHPP_{SSAT}$.

Definition 6 *Let $P = \langle R \cup R_{neg}, \tau \rangle$ be a (finite and non-tight) E-program in restricted $EHPP_{SSAT}$ and LP be a non-empty subset of $atoms(P)$. Then, LP is a loop of P if for any $A, B \in LP$, there exists a path of length > 0 from A to B, in the positive dependency graph of P, such that all the vertices in the path are in LP.*

Following (Lin & Zhao 2004), to allow Theorems 3 and 4 to be applied to non-tight E-programs $P = \langle R \cup R_{neg}, \tau \rangle$ in restricted $EHPP_{SSAT}$, we associate to each loop, LP, of P a formula, LF, called loop formula, and add this loop formula LF to $\mathcal{R}$ in the probabilistic completion, $Comp = \langle \mathcal{R}, Q \rangle$, of P. This obtains a one-to-one correspondence between the models of $\mathcal{R} \cup \mathcal{LF}$ and the probabilistic answer sets of P, and hence, Theorems 3 and 4 apply to non-tight E-programs (where $\mathcal{LF}$ is the set of all loop formulas of P). The loop means that non of the atoms involved in the loop can be defined in any probabilistic answer set, h, of P, and hence they do not exist in $dom(h)$. The added loop formulas associated with each loop of P to $\mathcal{R}$ in the probabilistic completion of P means that the atoms of the loops are not in any model of $\mathcal{R} \cup \mathcal{LF}$.

Definition 7 *Let $P = \langle R \cup R_{neg}, \tau \rangle$ be an E-program in restricted $EHPP_{SSAT}$ and LP be a loop in P. We define*

$$
R_P^+(LP) = \left\{ \begin{array}{l|l}
A : \mu \leftarrow A_1 : \mu_1, \ldots, A_m : \mu_m, & \\
not\ (A_{m+1} : \mu_{m+1}), \ldots, not\ (A_n : \mu_n) & \\
A : \mu \leftarrow A_1 : \mu_1, \ldots, A_m : \mu_m, & \\
not\ (A_{m+1} : \mu_{m+1}), \ldots, not\ (A_n : \mu_n) \in R, & \\
\quad and\ A \in LP, (\exists A').A' \in LP, A' \in \mathcal{B} &
\end{array} \right\}
$$

$$
R_P^-(LP) = \left\{ \begin{array}{l|l}
A : \mu \leftarrow A_1 : \mu_1, \ldots, A_m : \mu_m, & \\
not\ (A_{m+1} : \mu_{m+1}), \ldots, not\ (A_n : \mu_n) & \\
A : \mu \leftarrow A_1 : \mu_1, \ldots, A_m : \mu_m, & \\
not\ (A_{m+1} : \mu_{m+1}), \ldots, not\ (A_n : \mu_n) \in R, & \\
\quad and\ A \in LP, \neg\ (\exists A').A' \in LP, A' \in \mathcal{B} &
\end{array} \right\}
$$

where $\mathcal{B} = \{A_1, \ldots, A_m, A_{m+1}, \ldots, A_n\}$.

Intuitively, similar to (Lin & Zhao 2004), $R_P^+(LP)$ contains the E-rules in R that are involved in the loop LP. However, $R_P^-(LP)$ contains the E-rules in R that are not in the loop LP. Clearly, $R_P^+(LP)$ and $R_P^-(LP)$ are disjoint sets.

Definition 8 *Let $P = \langle R \cup R_{neg}, \tau \rangle$ be an E-program in restricted $EHPP_{SSAT}$ and LP be a loop in P. Let*

$$
\begin{array}{rcl}
A^i : \mu^i & \leftarrow & A_1^{ij} : \mu_1^{ij}, \ldots, A_m^{ij} : \mu_m^{ij}, \\
& & not\ (A_{m+1}^{ij} : \mu_{m+1}^{ij}), \ldots, not\ (A_n^{ij} : \mu_n^{ij})
\end{array}
$$

for $1 \leq j \leq k_n$, be the set of E-rules in $R_P^-(LP)$, for an atom A^i ($1 \leq i \leq n$). Then, the implication

$$
\neg[Body_{11} \vee \cdots \vee Body_{1k_1} \vee \ldots \vee Body_{n1} \vee \ldots \vee Body_{nk_n}] \supset \bigwedge_{A \in LP} \neg A
$$

is called a probabilistic loop formula, denoted by $LF(LP)$, of LP, where $Body_{ij} = A_1^{ij} \wedge \ldots \wedge A_m^{ij} \wedge \neg A_{m+1}^{ij} \wedge \ldots \wedge \neg A_n^{ij}$.

Theorem 5 *Let $P = \langle R \cup R_{neg}, \tau \rangle$ be any E-program in restricted $EHPP_{SSAT}$, $Comp(P) = \langle \mathcal{R}, Q \rangle$ be the probabilistic completion of P, Ans be the set of all probabilistic answer sets of P and $h \in Ans$. Let $\mathcal{LF}$ be the set of all probabilistic loop formulas associated with all loops of P. Then, $\mathcal{R} \cup \mathcal{LF}$ has a model iff P has a probabilistic answer set, and $\mathcal{S} = \langle \mathcal{R} \cup \mathcal{LF}, \theta, Q \rangle$ is satisfied iff $\sum_{h \in Ans} \prod_{A_i \in dom(h)} h(A_i) = val(\mathcal{R} \cup \mathcal{LF}, Q) \geq \theta$, viewing the SSAT formula, $\mathcal{S}$, as MAJSAT, where $\theta = \frac{1}{2}$, and*

$$\max_{h \models \widetilde{x}_1:[1,1],\ldots,\widetilde{x}_n:[1,1]} \left[h(\widetilde{x}_n) \sum_{h' \models \mathcal{D}} \prod_{i=1}^{n} h'(\widetilde{y}_i) \right] \geq \theta$$

where $\mathcal{D} \equiv \widetilde{x}_1 : [1,1], \ldots, \widetilde{x}_n : [1,1], \widetilde{y}_1 : [p_1, p_1], \ldots, \widetilde{y}_n : [p_n, p_n]$ and $\widetilde{x}_i = x_i$ or $\widetilde{x}_i = \neg x_i$ and $\widetilde{y}_i = y_i$ or $\widetilde{y}_i = \neg y_i$, viewing the SSAT formula, $\mathcal{S}$, as E-MAJSAT.

Example 10 *Consider again the non-tight E-program, P, from Example 9. This E-program belongs to restricted $EHPP_{SSAT}$ and has one loop $LP = \{a, b\}$, where*

$$\begin{array}{rcl} R^+(LP) & = & \{a : [0.5, 0.5] \leftarrow b : [0.3, 0.3], \\ & & b : [0.3, 0.3] \leftarrow a : [0.5, 0.5]\} \\ R^-(LP) & = & \emptyset. \end{array}$$

Thus, the loop formula $LF(LP)$ is $\neg false \supset (\neg a \wedge \neg b)$, which is equivalent to $(\neg a \wedge \neg b)$. Adding $LF(LP)$ to $\mathcal{R}$ outcomes the propositional formula $\mathcal{R} \cup LF(LP) = \{a \equiv b, (\neg a \wedge \neg b)\}$, which has only one model $\{\neg a, \neg b\}$. It can be easily verified that $val(\mathcal{R} \cup \{(\neg a \wedge \neg b)\}, Q) = 0.35$. This implies that the only probabilistic answer set h, where $h(\bar{a}) = [0.5, 0.5], h(\bar{b}) = [0.7, 0.7]$, of P corresponds to the only model of $\mathcal{R} \cup LF(LP)$. Moreover, the SSAT formula, $\mathcal{S} = \langle \mathcal{R} \cup \{(\neg a \wedge \neg b)\}, 0.5, Q \rangle$, is unsatisfied, since $\sum_{h \in Ans} \prod_{A_i \in dom(h)} h(A_i) = 0.35 = val(\mathcal{R} \cup \{(\neg a \wedge \neg b)\}, Q) \not\geq 0.5$.

6 Related Work and Conclusions

We studied the relationship between Extended Hybrid Probabilistic Logic Programs and Stochastic Satisfiability. We presented a modular translation from SSAT to a class of EHPP with probabilistic answer set semantics. The translation is based on a corresponding local translation from SAT to normal logic programs described in (Niemela 1999). This translation shows that the *fundamental* probabilistic reasoning tasks that can be encoded by SSAT (Majercik & Littman 1998; 2003; Littman, Majercik, & Pitassi 2001) — such as probabilistic planning, contingent probabilistic planning, the most probable explanation in a belief network, the most likely trajectory in probabilistic planning, and belief inference — can be also encoded and reasoned about using EHPP. Moreover, we have shown that there is no modular mapping from EHPP to SSAT. This shows that EHPP is more expressive than SSAT from the knowledge representation point of view.

In addition, we presented a translation from $EHPP_{SSAT}$ to SSAT that relies on a corresponding translation from normal logic programs to SAT described in (Clark 1978; Lin & Zhao 2004). $EHPP_{SSAT}$ is a restricted class of EHPP, since not every program in EHPP can be translated into SSAT. This is because EHPP allows probability intervals and conjunctions and disjunctions of literals to appear in the body of rules. However, SSAT allows point probabilities. Two classes of $EHPP_{SSAT}$ are identified; tight and non-tight $EHPP_{SSAT}$. The translation form tight $EHPP_{SSAT}$ to SSAT is based on the translation from tight normal logic programs (Fages 1994) to SAT, using Clark's completion (Clark 1978), by employing the notion of probabilistic completion that is a probabilistic extension to Clark's completion (Clark 1978). In addition, the translation form non tight $EHPP_{SSAT}$ to SSAT relies on the translation from non tight normal logic programs to SAT, using loop formulas (Lin & Zhao 2004), by employing the notion of probabilistic loop formulas that is a probabilistic extension to loop formulas of (Lin & Zhao 2004). The translation from $EHPP_{SSAT}$ to SSAT provides a foundation for an implementation for computing the probabilistic answer set semantics of EHPP by exploiting the existing work on SSAT with a selection from a variety of SSAT solvers.

A similar relationship between SSAT and other probabilistic logic programming frameworks, e.g., (Ng & Subrahmanian 1992; 1993; 1994; Dekhtyar & Subrahmanian 2000; Kern-Isberner & Lukasiewicz 2004; Lukasiewicz 1998; Baral, Gelfond, & Rushton 2004; Kersting & Raedt 2000; Lakshmanan & Sadri 2001; Poole 1997; Vennekens, Verbaeten, & Bruynooghe 2004), has not been studied. However, the relationship between the probabilistic logic programming frameworks (Ng & Subrahmanian 1992; 1993; 1994; Dekhtyar & Subrahmanian 2000; Kern-Isberner & Lukasiewicz 2004; Lukasiewicz 1998) and a different extension to SAT, namely, Probabilistic SAT (PSAT) (Boole 1854) has been studied. Given an assignment of probabilities to a collection of propositional formulas, PSAT asks if this assignment is consistent. The solution to PSAT is based on the possible world semantics. The possible world semantics solution to PSAT is achieved by compiling a linear program from the given probability assignments to a collection of propositional formulas, PSAT, and if this linear program has a solution, implies that the probability assignments to the set of propositional formulas is consistent. The relationship between PSAT and the probabilistic logic programming frameworks presented in (Ng & Subrahmanian 1992; 1993; 1994; Dekhtyar & Subrahmanian 2000; Kern-Isberner & Lukasiewicz 2004; Lukasiewicz 1998) has been studied. This relationship has been investigated by translating a probabilistic logic program, P, in (Ng & Subrahmanian 1992; 1993; 1994; Dekhtyar & Subrahmanian 2000; Kern-Isberner & Lukasiewicz 2004; Lukasiewicz 1998), to PSAT, by compiling a linear program, $\mathcal{LP}$, that is equivalent to PSAT from P. A solution to $\mathcal{LP}$ implies that P is consistent. This corresponds to translating a probabilistic logic program in (Ng & Subrahmanian 1992; 1993; 1994; Dekhtyar & Subrahmanian 2000; Kern-Isberner & Lukasiewicz 2004; Lukasiewicz 1998) to PSAT. However, it is not clear how to translate PSAT to a probabilistic logic program in (Ng & Subrahmanian 1992; 1993; 1994; Dekhtyar & Subrahmanian 2000; Kern-Isberner & Lukasiewicz 2004; Lukasiewicz 1998). The probabilistic logic programming frameworks of (Baral, Gelfond, & Rushton 2004; Kersting & Raedt 2000; Poole 1997; Vennekens, Verbaeten, & Bruynooghe 2004) relate probabilistic logic programming to Bayesian networks, which is different from SSAT and PSAT.

References

Baral, C.; Gelfond, M.; and Rushton, N. 2004. Probabilistic reasoning with answer sets. In *7th International Conference on Logic Programming and Nonmonotonic Reasoning*. Springer Verlag.

Boole, G. 1854. *The laws of thought*. Macmillan, London.

Clark, K. 1978. Negation as failure. In Gallaire, H., and Minker, J., eds., *Logic and Data Bases*. Plenum Press, New York. 293–322.

Davis, M.; Logemann, G.; and Loveland, D. 1962. A machine program for theorem-proving. *Communications of the ACM* 5(7):394–397.

Dekhtyar, A., and Subrahmanian, V. 2000. Hybrid probabilistic program. *Journal of Logic Programming* 43(3):187–250.

Fages, F. 1994. Consistency of clark's completion and existence of stable models. *Methods of Logic in Computer Science* 1:51–60.

Gelfond, M., and Lifschitz, V. 1991. Classical negation in logic programs and disjunctive databases. *New Generation Computing* 9(3-4):363–385.

Kern-Isberner, G., and Lukasiewicz, T. 2004. Combining probabilistic logic programming with the power of maximum entropy. *Artificial Intelligence* 157(1-2):139–202.

Kersting, K., and Raedt, L. D. 2000. Bayesian logic programs. In *Inductive Logic Programming*.

Lakshmanan, L., and Sadri, F. 2001. On a theory of probabilistic deductive databases. *Journal of Theory and Practice of Logic Programming* 1(1):5–42.

Lin, F., and Zhao, Y. 2004. Assat: Computing answer sets of a logic program by sat solvers. *Artificial Intelligence* 157(1-2):115–137.

Littman, M.; Majercik, S.; and Pitassi, T. 2001. Stochastic boolean satisfiability. *Journal of Automated Reasoning* 27(3):251–296.

Lukasiewicz, T. 1998. Probabilistic logic programming. In *13th European Conference on Artificial Intelligence*, 388–392.

Majercik, S., and Littman, M. 1998. Maxplan: A new approach to probabilistic planning. In *Fourth International Conference on Artificial Intelligence Planning*, 86–93.

Majercik, S., and Littman, M. 2003. Contingent planning under uncertainty via stochastic satisfiability. *Artificial Intelligence* 147(1-2):119–162.

Ng, R., and Subrahmanian, V. 1992. Probabilistic logic programming. *Information & Computation* 101(2).

Ng, R., and Subrahmanian, V. 1993. A semantical framework for supporting subjective and conditional probabilities in deductive databases. *Journal of Automated Reasoning* 10(2).

Ng, R., and Subrahmanian, V. 1994. Stable semantics for probabilistic deductive databases. *Information & Computation* 110(1).

Niemela, I. 1999. Logic programs with stable model semantics as a constraint programming paradigm. *Annals of Mathematics and Artificial Intelligence* 25(3-4):241–273.

Papadimitriou, C. H. 1985. Games against nature. *Journal of Computer and System Sciences* 31(2):288–301.

Poole, D. 1997. The independent choice logic for modelling multiple agents under uncertainty. *Artificial Intelligence* 94(1-2):7–56.

Saad, E., and Pontelli, E. 2005. Hybrid probabilistic logic programs with non-monotonic negation. In *International Conference on Logic Programming*.

Saad, E., and Pontelli, E. 2006. A new approach to hybrid probabilistic logic programs. *Annals of Mathematics and Artificial Intelligence* 48(3-4):187–243.

Saad, E. 2006. Incomplete knowledge in hybrid probabilistic logic programs. In *Tenth European Conference on Logics in Artificial Intelligence*. Springer Verlag.

Saad, E. 2007. Probabilistic planning in hybrid probabilistic logic programs. In *First International Conference on Scalable Uncertainty Management*. Springer Verlag.

Vennekens, J.; Verbaeten, S.; and Bruynooghe, M. 2004. Logic programs with annotated disjunctions. In *International Conference on Logic Programming*, 431–445.