I F I G
RESEARCH
REPORT

INSTITUT FÜR INFORMATIK

PROBABILISTIC LOGIC PROGRAMMING UNDER MAXIMUM ENTROPY

Thomas Lukasiewicz and Gabriele Kern-Isberner

IFIG RESEARCH REPORT 9903
FEBRUARY 1999

Institut für Informatik
JLU Gießen
Arndtstraße 2
D-35392 Giessen, Germany
Tel: +49-641-99-32141
Fax: +49-641-99-32149
mail@informatik.uni-giessen.de
www.informatik.uni-giessen.de

JUSTUS-LIEBIG-
UNIVERSITÄT
GIESSEN

IFIG Research Report
IFIG Research Report 9903, February 1999

Probabilistic Logic Programming under Maximum Entropy

Thomas Lukasiewicz* and Gabriele Kern-Isberner†

Abstract. In this paper, we focus on the combination of probabilistic logic programming with the principle of maximum entropy. We start by defining probabilistic queries to probabilistic logic programs and their answer substitutions under maximum entropy. We then present an efficient linear programming characterization for the problem of deciding whether a probabilistic logic program is satisfiable. Finally, and as a main result of this paper, we introduce an efficient technique for approximative probabilistic logic programming under maximum entropy. This technique reduces the original entropy maximization task to solving a modified and relatively small optimization problem.

CR Subject Classification (1998): I.2.3, I.2.4

*Institut für Informatik, Universität Gießen, Arndtstraße 2, D-35392 Gießen, Germany
E-mail: lukasiewicz@informatik.uni-giessen.de
†Institut für Informatik, FernUniversität Hagen, P.O. Box 940, D-58084 Hagen, Germany
E-mail: gabriele.kern-isberner@fernuni-hagen.de

1 Introduction

Probabilistic propositional logics and their various dialects are thoroughly studied in the literature (see especially [27] and [9]; see also [21] and [22]). Their extensions to probabilistic first-order logics can be classified into first-order logics in which probabilities are defined over the domain and those in which probabilities are given over a set of possible worlds (see especially [2], [3], and [13]). The first ones are suitable for describing statistical knowledge, while the latter are appropriate for representing degrees of belief. The same classification holds for existing approaches to *probabilistic logic programming*: Ng [23] concentrates on probabilities over the domain. Subrahmanian and his group (see especially [24], [25], [26], and [7]) focus on annotation-based approaches to degrees of belief. Poole [32], Haddawy [12], and Jaeger [14] discuss approaches to degrees of belief close to Bayesian networks [31]. Finally, another approach to probabilistic logic programming with degrees of belief, which is especially directed towards efficient implementations, has recently been introduced in [20].

Usually, the available probabilistic knowledge does not suffice to specify completely a distribution. In this case, applying the *principle of maximum entropy* is a well-appreciated means of probabilistic inference, both from a statistical and from a logical point of view. Entropy is an information-theoretical measure [36] reflecting the indeterminateness inherent to a distribution. Given some consistent probabilistic knowledge, the principle of maximum entropy chooses as the most appropriate representation the one distribution among all distributions satisfying that knowledge which has maximum entropy (ME). Within a rich statistical first-order language, Grove, Halpern and Koller [11] show that this ME-distribution may be taken to compute degrees of belief of formulas. Paris and Vencovskà [29] investigate the foundations of consistent probabilistic inference and set up postulates that characterize ME-inference uniquely within that framework. A similar result was stated in [37], based on optimization theory. Jaynes [16] regarded the ME-principle as a special case of a more general principle for translating information into a probability assignment.

The main idea of this paper is to combine probabilistic logic programming with the principle of maximum entropy. We thus follow an old idea already stated in the pioneering work by Nilsson [27], however, lifted to the first-order framework of probabilistic logic programs. At first sight, this project might seem an intractable task, since already probabilistic propositional logics under maximum entropy suffer from efficiency problems (which are due to an exponential number of possible worlds in the number of propositional variables). In this paper, however, we will see that this is not the case. More precisely, we will show that the efficient approach to probabilistic logic programming in [20], combined with new ideas, can be extended to an efficient approach to probabilistic logic programming under maximum entropy. Roughly speaking, the probabilistic logic programs presented in [20] generally carry an additional structure that can successfully be exploited in both classical probabilistic query processing and probabilistic query processing under maximum entropy.

The main contributions of this paper can be summarized as follows:

- We define probabilistic queries to probabilistic logic programs and their correct and tight answer substitutions under maximum entropy.
- We present an efficient linear programming characterization for the problem of deciding whether a probabilistic logic program is satisfiable.

- We introduce an efficient technique for approximative probabilistic logic programming under maximum entropy. In detail, this technique reduces the original entropy maximizations to relatively small optimization problems, which can easily be solved by existing ME-technology.

The rest of this paper is organized as follows. Section 2 introduces the technical background. In Section 3, we focus on deciding the satisfiability of probabilistic logic programs. In Section 4, we discuss probabilistic logic programming under maximum entropy itself. Section 5 finally summarizes the main results and gives an outlook on future research.

2 Technical Preliminaries

In this section, we introduce the technical background.

2.1 Probabilistic Logic Programs

We define the syntax and the semantics of probabilistic logic programs [20]:

Let Φ be a first-order vocabulary that contains a finite and nonempty set of predicate symbols and a finite and nonempty set of constant symbols (that is, we do not consider function symbols in this framework). Let $\mathcal{X}$ be a set of *object variables* and *bound variables*. Object variables represent elements of a certain domain, while bound variables describe real numbers in the interval $[0,1]$.

An *object term* is a constant symbol from Φ or an object variable from $\mathcal{X}$. An *atomic formula* is an expression of the kind $p(t_1, \ldots, t_k)$ with a predicate symbol p of arity $k \geq 0$ from Φ and object terms $t_1, \ldots, t_k$. A *conjunctive formula* is the *false formula* $\bot$, the *true formula* $\top$, or the conjunction $A_1 \wedge \cdots \wedge A_l$ of atomic formulas $A_1, \ldots, A_l$ with $l > 0$. A *probabilistic clause* is an expression of the form $(H|B)[c_1, c_2]$ with real numbers $c_1, c_2 \in [0,1]$ and conjunctive formulas H and B different from $\bot$. A *probabilistic program clause* is a probabilistic clause $(H|B)[c_1, c_2]$ with $c_1 \leq c_2$. We call H its *head* and B its *body*. A *probabilistic logic program* $\mathcal{P}$ is a finite set of probabilistic program clauses.

Probabilistic program clauses can be classified into facts, rules, and constraints as follows: *facts* are probabilistic program clauses of the form $(H|\top)[c_1, c_2]$ with $c_2 > 0$, *rules* are of the form $(H|B)[c_1, c_2]$ with $B \neq \top$ and $c_2 > 0$, and *constraints* are of the kind $(H|B)[0,0]$. Probabilistic program clauses can also be divided into logical and purely probabilistic program clauses: *logical program clauses* are probabilistic program clauses of the kind $(H|B)[1,1]$ or $(H|B)[0,0]$, while *purely probabilistic program clauses* are of the form $(H|B)[c_1, c_2]$ with $c_1 < 1$ and $c_2 > 0$. We abbreviate the logical program clauses $(H|B)[1,1]$ and $(H|B)[0,0]$ by $H \leftarrow B$ and $\bot \leftarrow H \wedge B$, respectively.

The semantics of probabilistic clauses is defined by a possible worlds semantics in which each possible world is identified with a Herbrand interpretation of the classical first-order language for Φ and $\mathcal{X}$ (that is, with a subset of the Herbrand base over Φ). Hence, the *set of possible worlds* $\mathcal{I}_\Phi$ is the set of all subsets of the Herbrand base HB_Φ. A *variable assignment* maps each object variable to an element of the Herbrand universe HU_Φ and each bound variable to a real number from $[0,1]$. For Herbrand interpretations I, conjunctive formulas C, and variable assignments σ, we write $I \models_\sigma C$ to denote that C is true in I under σ.

A *probabilistic interpretation* Pr is a mapping from $\mathcal{I}_\Phi$ to $[0,1]$ such that all $Pr(I)$ with $I \in \mathcal{I}_\Phi$ sum up to 1. Pr is extended to conjunctive formulas C as follows. The *probability* of C in the interpretation Pr under a variable assignment σ, denoted $Pr_\sigma(C)$, is defined by (we write $Pr(C)$ if C is variable-free):

$$Pr_\sigma(C) \;=\; \sum_{I \in \mathcal{I}_\Phi,\, I \models_\sigma C} Pr(I)\,.$$

Pr is extended to probabilistic clauses $(H|B)[c_1, c_2]$ as follows: $(H|B)[c_1, c_2]$ *is true* in the probabilistic interpretation Pr under a variable assignment σ, denoted $Pr \models_\sigma (H|B)[c_1, c_2]$, iff $c_1 \cdot Pr_\sigma(B) \leq Pr_\sigma(H \wedge B) \leq c_2 \cdot Pr_\sigma(B)$. A probabilistic clause $(H|B)[c_1, c_2]$ *is true* in Pr, denoted $Pr \models (H|B)[c_1, c_2]$, iff $Pr \models_\sigma (H|B)[c_1, c_2]$ for all variable assignments σ.

The notions of models and of satisfiability for probabilistic clauses are defined as usual. A probabilistic interpretation Pr is a *model* of a probabilistic clause F iff $Pr \models F$. It is a *model* of a set of probabilistic clauses $\mathcal{F}$, denoted $Pr \models \mathcal{F}$, iff Pr is a model of all probabilistic clauses in $\mathcal{F}$. A set of probabilistic clauses $\mathcal{F}$ is *satisfiable* iff a model of $\mathcal{F}$ exists.

Object terms, conjunctive formulas, and probabilistic clauses are *ground* iff they do not contain any variables. The notions of substitutions, ground substitutions, and ground instances of probabilistic clauses are canonically defined. Given a probabilistic logic program $\mathcal{P}$, we use $ground(\mathcal{P})$ to denote the set of all ground instances of probabilistic program clauses in $\mathcal{P}$. Moreover, we identify Φ with the vocabulary of all predicate and constant symbols that occur in $\mathcal{P}$.

2.2 Maximum Entropy

We now introduce maximum entropy models of sets of probabilistic clauses. The application of the principle of maximum entropy (see especially [36] and [15]) to probabilistic reasoning in the artificial intelligence context has a long history (see, for example, [5], [28], [30], and [33]). Recently, the principle of maximum entropy has been proved to be the most appropriate principle for dealing with conditionals [18] (that is, using the notions of the present paper, ground probabilistic clauses of the form $(H|B)[c_1, c_2]$ with $c_1 = c_2$).

The *maximum entropy model (ME-model)* of a satisfiable set of probabilistic clauses $\mathcal{F}$, denoted $ME[\mathcal{F}]$, is the unique probabilistic interpretation Pr that is a model of $\mathcal{F}$ and that has the greatest entropy among all the models of $\mathcal{F}$, where the entropy of Pr, denoted $H(Pr)$, is defined by:

$$H(Pr) \;=\; -\sum_{I \in \mathcal{I}_\Phi} Pr(I) \cdot \log Pr(I)\,.$$

2.3 Probabilistic Logic Programs under Maximum Entropy

We now define the notions of ME-consequence, tight ME-consequence, probabilistic queries, correct and tight ME-answer substitutions, and ME-answers.

A probabilistic clause F is a *maximum entropy consequence (ME-consequence)* of a set of probabilistic clauses $\mathcal{F}$, denoted $\mathcal{F} \models^\star F$, iff $ME[\mathcal{F}] \models F$. A probabilistic clause $(H|B)[c_1, c_2]$ is a *tight maximum entropy consequence (tight ME-consequence)* of a set of probabilistic clauses $\mathcal{F}$, denoted $\mathcal{F} \models^\star_{tight} (H|B)[c_1, c_2]$, iff c_1 is the minimum and c_2 is the maximum of all $ME_\sigma[\mathcal{F}](H \wedge B) / ME_\sigma[\mathcal{F}](B)$ with $ME_\sigma[\mathcal{F}](B) > 0$ and variable assignments σ.

A *probabilistic query* is an expression of the form $\exists(H|B)[c_1, c_2]$ or of the form $\exists(H|B)[x_1, x_2]$ with real numbers $c_1, c_2 \in [0, 1]$ such that $c_1 \leq c_2$, two different bound variables $x_1, x_2 \in \mathcal{X}$, and conjunctive formulas H and B different from $\perp$. A probabilistic query $\exists(H|B)[t_1, t_2]$ is *object-ground* iff H and B are ground.

Given a probabilistic query $\exists(H|B)[c_1, c_2]$ with $c_1, c_2 \in [0,1]$ to a probabilistic logic program $\mathcal{P}$, we are interested in its *correct maximum entropy answer substitutions (correct ME-answer substitutions)*, which are substitutions θ such that $\mathcal{P} \models^\star (H\theta|B\theta)[c_1, c_2]$ and that θ acts only on variables in $\exists(H|B)[c_1, c_2]$. Its *ME-answer* is Yes if a correct ME-answer substitution exists and No otherwise. Whereas, given a probabilistic query $\exists(H|B)[x_1, x_2]$ with $x_1, x_2 \in \mathcal{X}$ to a probabilistic logic program $\mathcal{P}$, we are interested in its *tight maximum entropy answer substitutions (tight ME-answer substitutions)*, which are substitutions θ such that $\mathcal{P} \models^\star_{tight} (H\theta|B\theta)[x_1\theta, x_2\theta]$, that θ acts only on variables in $\exists(H|B)[x_1, x_2]$, and that $x_1\theta, x_2\theta \in [0,1]$. Note that for a probabilistic query $\exists(H|B)[x_1, x_2]$ with $x_1, x_2 \in \mathcal{X}$, there always exists a tight ME-answer substitution.

2.4 Example

We give an example adapted from [20]. Let us assume that John wants to pick up Mary after she stopped working. To do so, he must drive from his home to her office. However, he left quite late. So, he is wondering if he can still reach her in time. Unfortunately, since it is rush hour, it is very probable that he runs into a traffic jam. Now, John has the following knowledge at hand: given a road (*ro*) in the south (*so*) of the town, he knows that the probability that he can reach (*re*) S through R without running into a traffic jam is 90% (1). A friend just called him and gave him advice (*ad*) about some roads without any significant traffic (2). He also clearly knows that if he can reach S through T and T through R, both without running into a traffic jam, then he can also reach S through R without running into a traffic jam (3). This knowledge can be expressed by the following probabilistic rules (R, S, and T are object variables):

(1) $$(re(R,S)\,|\,ro(R,S) \wedge so(R,S))[0.9, 0.9]$$

(2) $$(re(R,S)\,|\,ro(R,S) \wedge ad(R,S))[1, 1]$$

(3) $$(re(R,S)\,|\,re(R,T) \wedge re(T,S))[1, 1]\,.$$

Some self-explaining probabilistic facts are given as follows (h, a, b, and o are constant symbols; the fourth clause describes the fact that John is not sure anymore whether or not his friend was talking about the road from a to b):

$$(ro(h,a)\,|\,\top)[1,1],\ (ad(h,a)\,|\,\top)[1,1]$$
$$(ro(a,b)\,|\,\top)[1,1],\ (ad(a,b)\,|\,\top)[0.8,0.8]$$
$$(ro(b,o)\,|\,\top)[1,1],\ (so(b,o)\,|\,\top)[1,1]\,.$$

John is wondering whether he can reach Mary's office from his home, such that the probability of him running into a traffic jam is smaller than 1%. This can be expressed by the probabilistic query $\exists(re(h,o))[.99, 1]$. His wondering about the probability of reaching the office, without running into a traffic jam, can be expressed by $\exists(re(h,o))[X_1, X_2]$, where X_1 and X_2 are bound variables.

3 Satisfiability

In this section, we concentrate on the problem of deciding whether a probabilistic logic program is satisfiable. Note that while classical logic programs without negation and logical constraints (see especially [19] and [1]) are always satisfiable, probabilistic logic programs may become unsatisfiable, just for logical inconsistencies through logical constraints or, more generally, for probabilistic inconsistencies in the assumed probability ranges.

3.1 Naive Linear Programming Characterization

The satisfiability of a probabilistic logic program $\mathcal{P}$ can be characterized in a straightforward way by the solvability of a system of linear constraints as follows.

Let $\mathcal{LC}_\Phi$ be the least set of linear constraints over $y_I \geq 0$ $(I \in \mathcal{I}_\Phi)$ containing:

(1) $\sum_{I \in \mathcal{I}_\Phi} y_I \;=\; 1$

(2) $c_1 \cdot \sum_{I \in \mathcal{I}_\Phi,\, I \models B} y_I \;\leq\; \sum_{I \in \mathcal{I}_\Phi,\, I \models H \wedge B} y_I \;\leq\; c_2 \cdot \sum_{I \in \mathcal{I}_\Phi,\, I \models B} y_I$
for all $(H|B)[c_1, c_2] \in \mathit{ground}(\mathcal{P})$.

It is now easy to see that $\mathcal{P}$ is satisfiable iff $\mathcal{LC}_\Phi$ is solvable. The crux with this *naive characterization* is that the number of variables and of linear constraints is linear in the cardinality of $\mathcal{I}_\Phi$ and of $\mathit{ground}(\mathcal{P})$, respectively. Thus, especially the number of variables is generally quite large, as the following example shows.

Example 3.1 Let us take the probabilistic logic program $\mathcal{P}$ that comprises all the probabilistic program clauses given in Section 2.4. If we characterize the satisfiability of $\mathcal{P}$ in the described naive way, then we get a system of linear constraints that has $2^{64} \approx 18 \cdot 10^{18}$ (!) variables and 205 linear constraints.

3.2 Reduced Linear Programming Characterization

We now present a new system of linear constraints to characterize the satisfiability of a probabilistic logic program $\mathcal{P}$. This new system generally has a much lower size than $\mathcal{LC}_\Phi$. In detail, we combine some ideas from [20] with the idea of partitioning $\mathit{ground}(\mathcal{P})$ into active and inactive ground instances, which yields another substantial increase of efficiency. We need some preparations:

Let $\underline{\mathcal{P}}$ denote the set of all logical program clauses in $\mathcal{P}$. Let $\overline{\mathcal{P}}$ denote the least set of logical program clauses that contains $H \leftarrow B$ if the program $\mathcal{P}$ contains a probabilistic program clause $(H|B)[c_1, c_2]$ with $c_2 > 0$.

We define a mapping R that maps each ground conjunctive formula C to a subset of $HB_\Phi \cup \{\bot\}$ as follows. If $C = \bot$, then $R(C)$ is $HB_\Phi \cup \{\bot\}$. If $C \neq \bot$, then $R(C)$ is the set of all ground atomic formulas that occur in C.

For a set $\mathcal{L}$ of logical program clauses, we define the operator $T_\mathcal{L}{\uparrow}\omega$ on the set of all subsets of $HB_\Phi \cup \{\bot\}$ as usual. For this task, we need the *immediate consequence operator* $T_\mathcal{L}$, which is defined as follows. For all $I \subseteq HB_\Phi \cup \{\bot\}$:

$$T_\mathcal{L}(I) \;=\; \bigcup \{R(H) \,|\, H \leftarrow B \in \mathit{ground}(\mathcal{L}) \text{ with } R(B) \subseteq I\}\,.$$

For all $I \subseteq HB_\Phi \cup \{\bot\}$, we define $T_\mathcal{L}{\uparrow}\omega(I)$ as the union of all $T_\mathcal{L}{\uparrow}n(I)$ with $n < \omega$, where $T_\mathcal{L}{\uparrow}0(I) = I$ and $T_\mathcal{L}{\uparrow}(n+1)(I) = T_\mathcal{L}(T_\mathcal{L}{\uparrow}n(I))$ for all $n < \omega$. We adopt the usual convention to abbreviate $T_\mathcal{L}{\uparrow}\alpha(\emptyset)$ by $T_\mathcal{L}{\uparrow}\alpha$.

The set $\mathit{ground}(\mathcal{P})$ is now partitioned into active and inactive ground instances as follows. A ground instance $(H|B)[c_1, c_2] \in \mathit{ground}(\mathcal{P})$ is *active* if $R(H) \cup R(B) \subseteq T_{\overline{\mathcal{P}}}{\uparrow}\omega$ and *inactive* otherwise. We use $\mathit{active}(\mathcal{P})$ to denote the set of all active ground instances of $\mathit{ground}(\mathcal{P})$.

We are now ready to define the index set $\mathcal{I}_\mathcal{P}$ of the variables in the new system of linear constraints. It is defined by $\mathcal{I}_\mathcal{P} = \mathcal{I}'_\mathcal{P} \cap \mathcal{I}_\Phi$, where $\mathcal{I}'_\mathcal{P}$ is the least set of subsets of $HB_\Phi \cup \{\bot\}$ with:

(α) $T_{\underline{\mathcal{P}}}\!\uparrow\omega \in \mathcal{I}'_{\mathcal{P}}$,

(β) $T_{\underline{\mathcal{P}}}\!\uparrow\omega(R(B)), T_{\underline{\mathcal{P}}}\!\uparrow\omega(R(H) \cup R(B)) \in \mathcal{I}'_{\mathcal{P}}$ for all purely probabilistic program clauses $(H|B)[c_1, c_2] \in \mathit{active}(\mathcal{P})$,

(γ) $T_{\underline{\mathcal{P}}}\!\uparrow\omega(I_1 \cup I_2) \in \mathcal{I}'_{\mathcal{P}}$ for all $I_1, I_2 \in \mathcal{I}'_{\mathcal{P}}$.

The index set $\mathcal{I}_{\mathcal{P}}$ just involves atomic formulas from $T_{\overline{\mathcal{P}}}\!\uparrow\omega$:

Lemma 3.2 *It holds $I \subseteq T_{\overline{\mathcal{P}}}\!\uparrow\omega$ for all $I \in \mathcal{I}_{\mathcal{P}}$.*

Proof. The proof is given in full detail in the appendix. □

The new system of linear constraints $\mathcal{LC}_{\mathcal{P}}$ itself is defined as follows: $\mathcal{LC}_{\mathcal{P}}$ is the least set of linear constraints over $y_I \geq 0$ $(I \in \mathcal{I}_{\mathcal{P}})$ that contains:

(1) $\sum_{I \in \mathcal{I}_{\mathcal{P}}} y_I \;=\; 1$

(2) $c_1 \cdot \sum_{I \in \mathcal{I}_{\mathcal{P}}, I \models B} y_I \;\leq\; \sum_{I \in \mathcal{I}_{\mathcal{P}}, I \models H \wedge B} y_I \;\leq\; c_2 \cdot \sum_{I \in \mathcal{I}_{\mathcal{P}}, I \models B} y_I$
for all purely probabilistic program clauses $(H|B)[c_1, c_2] \in \mathit{active}(\mathcal{P})$.

We now roughly describe the ideas that carry us to the new linear constraints. The first idea is to just introduce a variable for each $I \subseteq T_{\overline{\mathcal{P}}}\!\uparrow\omega$, and not for each $I \subseteq HB_{\Phi}$ anymore. This also means to introduce a linear constraint only for each member of $\mathit{active}(\mathcal{P})$, and not for each member of $\mathit{ground}(\mathcal{P})$ anymore. The second idea is to exploit all logical program clauses in $\mathcal{P}$. That is, to just introduce a variable for each $T_{\underline{\mathcal{P}}}\!\uparrow\omega(I)$ with $I \subseteq T_{\overline{\mathcal{P}}}\!\uparrow\omega$. This also means to introduce a linear constraint only for each purely probabilistic member of $\mathit{active}(\mathcal{P})$. Finally, the third idea is to exploit the structure of all purely probabilistic members of $\mathit{active}(\mathcal{P})$. That is, to just introduce a variable for each $I \in \mathcal{I}_{\mathcal{P}}$.

The following important theorem shows the correctness of these ideas.

Theorem 3.3 *$\mathcal{P}$ is satisfiable iff $\mathcal{LC}_{\mathcal{P}}$ is solvable.*

Proof. The proof is given in full detail in the appendix. □

We give an example to illustrate the new system of linear constraints $\mathcal{LC}_{\mathcal{P}}$.

Example 3.4 Let us take again the probabilistic logic program $\mathcal{P}$ that comprises all the probabilistic program clauses given in Section 2.4. The system $\mathcal{LC}_{\mathcal{P}}$ then consists of *five* linear constraints over *four* variables $y_i \geq 0$ $(i \in [0\!:\!3])$:

$$\begin{aligned}
y_0 + y_1 + y_2 + y_3 &= 1 \\
0.9 \cdot (y_0 + y_1 + y_2 + y_3) &\leq y_1 + y_3 \\
0.9 \cdot (y_0 + y_1 + y_2 + y_3) &\geq y_1 + y_3 \\
0.8 \cdot (y_0 + y_1 + y_2 + y_3) &\leq y_2 + y_3 \\
0.8 \cdot (y_0 + y_1 + y_2 + y_3) &\geq y_2 + y_3
\end{aligned}$$

More precisely, the variables y_i $(i \in [0\!:\!3])$ correspond as follows to the members of $\mathcal{I}_{\mathcal{P}}$ (written in binary as subsets of $T_{\overline{\mathcal{P}}}\!\uparrow\omega = \{ro(h,a), ro(a,b), ro(b,o), ad(h,a), ad(a,b), so(b,o), re(h,a), re(a,b), re(b,o), re(h,b), re(a,o), re(h,o)\}$):

$$y_0 \mathrel{\hat=} 111101100000,\; y_1 \mathrel{\hat=} 111101101000$$
$$y_2 \mathrel{\hat=} 111111110100,\; y_3 \mathrel{\hat=} 111111111111.$$

Moreover, the four linear inequalities correspond to the following two active ground instances of purely probabilistic program clauses in $\mathcal{P}$:

$$(re(b,o) \mid ro(b,o) \wedge so(b,o))[0.9, 0.9],\ (ad(a,b) \mid \top)[0.8, 0.8]\,.$$

4 Probabilistic Logic Programming under ME

In this section, we concentrate on the problem of computing tight ME-answer substitutions for probabilistic queries to probabilistic logic programs. Since every general probabilistic query can be reduced to a finite number of object-ground probabilistic queries, we restrict our attention to object-ground queries.

In the sequel, let $\mathcal{P}$ be a satisfiable probabilistic logic program and let $Q = \exists(G|A)[x_1, x_2]$ be an object-ground query with $x_1, x_2 \in \mathcal{X}$. To provide the tight ME-answer substitution for Q, we now need $ME[\mathcal{P}](A)$ and $ME[\mathcal{P}](G \wedge A)$.

4.1 Exact ME-Models

The ME-model of $\mathcal{P}$ can be computed in a straightforward way by solving the following entropy maximization problem over the variables $y_I \geq 0$ $(I \in \mathcal{I}_\Phi)$:

$$\text{(4)} \qquad \max \ -\sum_{I \in \mathcal{I}_\Phi} y_I \log y_I \ \text{ subject to } \mathcal{LC}_\Phi.$$

The crux with this optimization problem (4) is that especially the number of variables is generally quite large (see also Section 3.1):

Example 4.1 Let us take again the probabilistic logic program $\mathcal{P}$ from Section 2.4. The entropy maximization (4) is done subject to a system of 205 linear constraints over $2^{64} \approx 18 \cdot 10^{18}$ variables.

4.2 Approximative ME-Models

We now introduce approximative ME-models, which are characterized by optimization problems that generally have a much smaller size than (4).

Like the linear programs in Section 3.1, the optimization problems (4) suffer especially from a large number of variables. It is thus natural to wonder whether the reduction technique of Section 3.2 also applies to (4).

This is indeed the case, if we make the following two assumptions:

(1) All ground atomic formulas in Q belong to $T_{\overline{\mathcal{P}}}\!\uparrow\omega$.

(2) Instead of computing the ME-model of $\mathcal{P}$, we compute the ME-model of *active*($\mathcal{P}$) (that is, we *approximate* $ME[\mathcal{P}]$ by $ME[\mathit{active}(\mathcal{P})]$).

Note that both assumptions (1) and (2) are just small restrictions, if we consider that the logical approximation $\overline{\mathcal{P}}$ of the probabilistic logic program $\mathcal{P}$ does not logically entail any other ground atomic formulas than those in $T_{\overline{\mathcal{P}}}\!\uparrow\omega$.

We now have to adapt the technical notions of Section 3.2 as follows. The index set $\mathcal{I}_\mathcal{P}$ must be adapted by also incorporating the structure of the query Q into its definition. More precisely, the new index set $\mathcal{I}_{\mathcal{P},Q}$ is defined by $\mathcal{I}_{\mathcal{P},Q} = \mathcal{I}'_{\mathcal{P},Q} \cap \mathcal{I}_\Phi$, where $\mathcal{I}'_{\mathcal{P},Q}$ is the least set of subsets of $HB_\Phi \cup \{\bot\}$ with:

(α) $T_{\underline{\mathcal{P}}}\!\uparrow\omega, T_{\underline{\mathcal{P}}}\!\uparrow\omega(R(A)), T_{\underline{\mathcal{P}}}\!\uparrow\omega(R(G)\cup R(A)) \in \mathcal{I}'_{\mathcal{P},Q}$,

(β) $T_{\underline{\mathcal{P}}}\!\uparrow\omega(R(B)), T_{\underline{\mathcal{P}}}\!\uparrow\omega(R(H)\cup R(B)) \in \mathcal{I}'_{\mathcal{P},Q}$ for all purely probabilistic program clauses $(H|B)[c_1,c_2] \in \mathit{active}(\mathcal{P})$,

(γ) $T_{\underline{\mathcal{P}}}\!\uparrow\omega(I_1\cup I_2) \in \mathcal{I}'_{\mathcal{P},Q}$ for all $I_1, I_2 \in \mathcal{I}'_{\mathcal{P},Q}$.

Also the new index set $\mathcal{I}_{\mathcal{P},Q}$ just involves atomic formulas from $T_{\overline{\mathcal{P}}}\!\uparrow\omega$:

Lemma 4.2 *It holds $I \subseteq T_{\overline{\mathcal{P}}}\!\uparrow\omega$ for all $I \in \mathcal{I}_{\mathcal{P},Q}$.*

Proof. The proof is given in full detail in the appendix. □

The system of linear constraints $\mathcal{LC}_{\mathcal{P}}$ must be adapted to $\mathcal{LC}_{\mathcal{P},Q}$, which is the least set of linear constraints over $y_I \geq 0$ ($I \in \mathcal{I}_{\mathcal{P},Q}$) that contains:

(1) $\sum_{I\in\mathcal{I}_{\mathcal{P},Q}} y_I \;=\; 1$

(2) $c_1 \cdot \sum_{I\in\mathcal{I}_{\mathcal{P},Q},\, I\models B}\, y_I \;\leq\; \sum_{I\in\mathcal{I}_{\mathcal{P},Q},\, I\models H\wedge B}\, y_I \;\leq\; c_2 \cdot \sum_{I\in\mathcal{I}_{\mathcal{P},Q},\, I\models B}\, y_I$
for all purely probabilistic program clauses $(H|B)[c_1,c_2] \in \mathit{active}(\mathcal{P})$.

Finally, we need the following definitions. Let $\mathcal{I}_{\overline{\mathcal{P}}} = \{L \mid L \subseteq T_{\overline{\mathcal{P}}}\!\uparrow\omega\}$ and let a_I ($I \in \mathcal{I}_{\mathcal{P},Q}$) be the number of all possible worlds $J \in T_{\underline{\mathcal{P}}}\!\uparrow\omega(\mathcal{I}_{\overline{\mathcal{P}}}) \cap \mathcal{I}_{\overline{\mathcal{P}}}$ that are a superset of I and that are not a superset of any $K \in \mathcal{I}_{\mathcal{P},Q}$ that properly includes I.

Roughly speaking, $\mathcal{I}_{\mathcal{P},Q}$ defines a partition $\{\mathcal{S}_I \mid I \in \mathcal{I}_{\mathcal{P},Q}\}$ of $T_{\underline{\mathcal{P}}}\!\uparrow\omega(\mathcal{I}_{\overline{\mathcal{P}}}) \cap \mathcal{I}_{\overline{\mathcal{P}}}$ and each a_I with $I \in \mathcal{I}_{\mathcal{P},Q}$ denotes the cardinality of $\mathcal{S}_I$. Note especially that $a_I > 0$ for all $I \in \mathcal{I}_{\mathcal{P},Q}$, since $\mathcal{I}_{\mathcal{P},Q} \subseteq T_{\underline{\mathcal{P}}}\!\uparrow\omega(\mathcal{I}_{\overline{\mathcal{P}}}) \cap \mathcal{I}_{\overline{\mathcal{P}}}$.

We are now ready to characterize the ME-model of $\mathit{active}(\mathcal{P})$ by the optimal solution of a reduced optimization problem.

Theorem 4.3 *For all ground conjunctive formulas C with $T_{\underline{\mathcal{P}}}\!\uparrow\omega(R(C)) \in \mathcal{I}_{\mathcal{P},Q}$:*

$$\mathit{ME}[\mathit{active}(\mathcal{P})](C) \;=\; \sum_{I\in\mathcal{I}_{\mathcal{P},Q},\, I\models C} y_I^\star\,,$$

where $y_I^\star$ with $I \in \mathcal{I}_{\mathcal{P},Q}$ is the optimal solution of the following optimization problem over the variables $y_I \geq 0$ with $I \in \mathcal{I}_{\mathcal{P},Q}$:

$$\text{(5)} \qquad \max\; -\sum_{I\in\mathcal{I}_{\mathcal{P},Q}} y_I\,(\log y_I - \log a_I) \;\textit{ subject to } \mathcal{LC}_{\mathcal{P},Q}.$$

Proof. The proof is given in full detail in the appendix. □

The tight ME-answer substitution for the probabilistic query Q to the ground probabilistic logic program $\mathit{active}(\mathcal{P})$ is more precisely given as follows.

Corollary 4.4 *Let $y_I^\star$ with $I \in \mathcal{I}_{\mathcal{P},Q}$ be the optimal solution of (5).*

a) If $y_I^\star = 0$ for all $I \in \mathcal{I}_{\mathcal{P},Q}$ with $I \models A$, then the tight ME-answer substitution for the query $\exists(G|A)[x_1,x_2]$ to $\mathit{active}(\mathcal{P})$ is given by $\{x_1/1, x_2/0\}$.

b) If $y_I^\star > 0$ for some $I \in \mathcal{I}_{\mathcal{P},Q}$ with $I \models A$, then the tight ME-answer substitution for the query $\exists(G|A)[x_1,x_2]$ to $\mathit{active}(\mathcal{P})$ is given by $\{x_1/d, x_2/d\}$, where

$$d \;=\; \sum_{I\in\mathcal{I}_{\mathcal{P},Q},\, I\models G\wedge A} y_I^\star \;/\; \sum_{I\in\mathcal{I}_{\mathcal{P},Q},\, I\models A} y_I^\star\,.$$

We give an example to illustrate the optimization problem (5).

Example 4.5 Let us take again the probabilistic logic program $\mathcal{P}$ from Section 2.4. The tight ME-answer substitution for the query $\exists(re(h,o))[X_1, X_2]$ to $active(\mathcal{P})$ is given by $\{X_1/.9353, X_2/.9353\}$, since $ME[active(\mathcal{P})](re(h,o)) = y_3^\star + y_4^\star + y_5^\star + y_6^\star = .9353$, where $y_i^\star$ $(i \in [0:6])$ is the optimal solution of the following optimization problem over the variables $y_i \geq 0$ $(i \in [0\!:\!6])$:

$$\max \quad -\sum_{i=0}^{6} y_i\,(\log y_i - \log a_i) \text{ subject to } \mathcal{LC}_{\mathcal{P},Q},$$

where $(a_0, a_1, a_2, a_3, a_4, a_5, a_6)$ is given by $(3,1,1,1,6,5,2)$ and $\mathcal{LC}_{\mathcal{P},Q}$ consists of the following *five* linear constraints over the *seven* variables $y_i \geq 0$ $(i \in [0\!:\!6])$:

$$\begin{array}{rcl} y_0+y_1+y_2+y_3+y_4+y_5+y_6 &=& 1 \\ 0.9\cdot(y_0+y_1+y_2+y_3+y_4+y_5+y_6) &\leq& y_1+y_3+y_5 \\ 0.9\cdot(y_0+y_1+y_2+y_3+y_4+y_5+y_6) &\geq& y_1+y_3+y_5 \\ 0.8\cdot(y_0+y_1+y_2+y_3+y_4+y_5+y_6) &\leq& y_2+y_3+y_6 \\ 0.8\cdot(y_0+y_1+y_2+y_3+y_4+y_5+y_6) &\geq& y_2+y_3+y_6 \end{array}$$

More precisely, the variables y_i $(i \in [0\!:\!6])$ correspond as follows to the members of $\mathcal{I}_{\mathcal{P},Q}$ (written in binary as subsets of $T_{\overline{\mathcal{P}}}\!\uparrow\omega = \{ro(h,a), ro(a,b), ro(b,o), ad(h,a), ad(a,b), so(b,o), re(h,a), re(a,b), re(b,o), re(h,b), re(a,o), re(h,o)\}$):

$$\begin{array}{l} y_0 \mathrel{\hat{=}} 111101100000,\; y_1 \mathrel{\hat{=}} 111101101000,\; y_2 \mathrel{\hat{=}} 111111110100 \\ y_3 \mathrel{\hat{=}} 111111111111,\; y_4 \mathrel{\hat{=}} 111101100001,\; y_5 \mathrel{\hat{=}} 111101101001 \\ y_6 \mathrel{\hat{=}} 111111110101 \end{array}$$

Furthermore, the variables y_i $(i \in [0\!:\!6])$ correspond as follows to the members of $T_{\underline{\mathcal{P}}}\!\uparrow\omega(\mathcal{I}_{\overline{\mathcal{P}}}) \cap \mathcal{I}_{\overline{\mathcal{P}}}$ (written in binary as subsets of $T_{\overline{\mathcal{P}}}\!\uparrow\omega$). Note that a_i with $i \in [0:6]$ is given by the number of members associated with y_i.

$$\begin{array}{l} y_0 \mathrel{\hat{=}} \langle 111101100000, 111101100100, 111101110100\rangle \\ y_1 \mathrel{\hat{=}} \langle 111101101000\rangle \\ y_2 \mathrel{\hat{=}} \langle 111111110100\rangle \\ y_3 \mathrel{\hat{=}} \langle 111111111111\rangle \\ y_4 \mathrel{\hat{=}} \langle 111101100001, 111101100011, 111101100101, \\ \qquad 111101100111, 111101110101, 111101110111\rangle \\ y_5 \mathrel{\hat{=}} \langle 111101101001, 111101101011, 111101101101, \\ \qquad 111101101111, 111101111111\rangle \\ y_6 \mathrel{\hat{=}} \langle 111111110101, 111111110111\rangle \end{array}$$

Finally, the four linear inequalities correspond to the following two active ground instances of purely probabilistic program clauses in $\mathcal{P}$:

$$(re(b,o) \mid ro(b,o) \wedge so(b,o))[0.9, 0.9],\; (ad(a,b) \mid \top)[0.8, 0.8]\,.$$

Note that we used the ME-system shell SPIRIT (see especially [33] and [34]) to compute the ME-model of $active(\mathcal{P})$.

4.3 Computing Approximative ME-Models

We now briefly discuss the problem of computing the numbers a_I with $I \in \mathcal{I}_{\mathcal{P},Q}$ and the problem of solving the optimization problem (5).

As far as the numbers a_I are concerned, we just have to solve two linear equations. For this purpose, we need the new index set $\mathcal{I}^+_{\mathcal{P},Q}$ defined by $\mathcal{I}^+_{\mathcal{P},Q} = \mathcal{I}''_{\mathcal{P},Q} \cap \mathcal{I}_\Phi$, where $\mathcal{I}''_{\mathcal{P},Q}$ is the least set of subsets of $HB_\Phi \cup \{\bot\}$ with:

(α) $\emptyset, R(A), R(G) \cup R(A) \in \mathcal{I}''_{\mathcal{P},Q}$,

(β) $R(B), R(H) \cup R(B) \in \mathcal{I}''_{\mathcal{P},Q}$ for all $(H|B)[c_1, c_2] \in \mathit{active}(\mathcal{P})$,

(γ) $I_1 \cup I_2 \in \mathcal{I}''_{\mathcal{P},Q}$ for all $I_1, I_2 \in \mathcal{I}''_{\mathcal{P},Q}$.

We start by computing the numbers s_J with $J \in \mathcal{I}^+_{\mathcal{P},Q}$, which are the unique solution of the following system of linear equations:

$$\sum_{J \in \mathcal{I}^+_{\mathcal{P},Q},\, J \subseteq I} s_J \;=\; 2^{|T_{\underline{\mathcal{P}}} \uparrow \omega| - |I|} \;\text{ for all } I \in \mathcal{I}^+_{\mathcal{P},Q}\,.$$

We are now ready to compute the numbers a_J with $J \in \mathcal{I}_{\mathcal{P},Q}$, which are the unique solution of the following system of linear equations:

$$\sum_{J \in \mathcal{I}_{\mathcal{P},Q},\, J \subseteq I} a_J \;=\; \sum_{J \in \mathcal{I}^+_{\mathcal{P},Q},\, J \subseteq I,\, J \models \underline{\mathcal{P}}} s_J \;\text{ for all } I \in \mathcal{I}_{\mathcal{P},Q}\,.$$

As far as the optimization problem (5) is concerned, we can build on existing ME-technology. For example, the ME-system PIT (see [10] and [35]) solves entropy maximization problems subject to *indifferent* possible worlds (that is, certain possible worlds are assumed to have the same probability). It can thus directly be used to solve the optimization problem (5).

Note also that if the probabilistic logic program $\mathcal{P}$ contains just probabilistic program clauses of the form $(H|B)[c_1, c_2]$ with $c_1 = c_2$, then the optimization problem (5) can easily be solved by standard Lagrangean techniques (as described in [34] and [35] for entropy maximization).

5 Summary and Outlook

In this paper, we discussed the combination of probabilistic logic programming with the principle of maximum entropy. We presented an efficient linear programming characterization for the problem of deciding whether a probabilistic logic program is satisfiable. Furthermore, we especially introduced an efficient technique for approximative query processing under maximum entropy.

A very interesting topic of future research is to analyze the relationship between the ideas of this paper and the characterization of the principle of maximum entropy in the framework of conditionals given in [18].

Appendix

Proof of Lemma 3.2: The claim is proved by induction on the definition of $\mathcal{I}'_{\mathcal{P}}$ as follows. Let $I \in \mathcal{I}'_{\mathcal{P}}$ with $I \neq HB_\Phi \cup \{\bot\}$.

(α) If $I = T_{\underline{\mathcal{P}}}\!\uparrow\omega$, then $I \subseteq T_{\overline{\mathcal{P}}}\!\uparrow\omega$, since $T_{\underline{\mathcal{P}}}\!\uparrow\omega \subseteq T_{\overline{\mathcal{P}}}\!\uparrow\omega$.

(β) If $I = T_{\underline{\mathcal{P}}}\!\uparrow\omega(R(B))$ or $I = T_{\underline{\mathcal{P}}}\!\uparrow\omega(R(H) \cup R(B))$ for some purely probabilistic $(H|B)[c_1, c_2] \in$ *active*$(\mathcal{P})$, then $I \subseteq T_{\overline{\mathcal{P}}}\!\uparrow\omega$, since $R(H) \cup R(B) \subseteq T_{\overline{\mathcal{P}}}\!\uparrow\omega$.

(γ) If $I = T_{\underline{\mathcal{P}}}\!\uparrow\omega(I_1 \cup I_2)$ for some $I_1, I_2 \in \mathcal{I}'_{\mathcal{P}}$, then $I \subseteq T_{\overline{\mathcal{P}}}\!\uparrow\omega$, since $I_1, I_2 \neq HB_\Phi \cup \{\bot\}$ and thus $I_1 \cup I_2 \subseteq T_{\overline{\mathcal{P}}}\!\uparrow\omega$ by the induction hypothesis. □

Proof of Theorem 3.3: We first need some preparation as follows. We show that all purely probabilistic program clauses from *active*$(\mathcal{P})$ can be interpreted by probability functions over a partition $\{\mathcal{S}_I \mid I \in \mathcal{I}_{\mathcal{P}}\}$ of $T_{\underline{\mathcal{P}}}\!\uparrow\omega(\mathcal{I}_\Phi) \cap \mathcal{I}_\Phi$. That is, as far as *active*$(\mathcal{P})$ is concerned, we do not need the fine granulation of $\mathcal{I}_\Phi$.

For all $I \in \mathcal{I}_{\mathcal{P}}$ let $\mathcal{S}_I$ be the set of all possible worlds $J \in T_{\underline{\mathcal{P}}}\!\uparrow\omega(\mathcal{I}_\Phi) \cap \mathcal{I}_\Phi$ that are a superset of I and that are not a superset of any $K \in \mathcal{I}_{\mathcal{P}}$ that properly includes I. We now show that $\{\mathcal{S}_I \mid I \in \mathcal{I}_{\mathcal{P}}\}$ is a partition of $T_{\underline{\mathcal{P}}}\!\uparrow\omega(\mathcal{I}_\Phi) \cap \mathcal{I}_\Phi$. Assume first that there are two different $I_1, I_2 \in \mathcal{I}_{\mathcal{P}}$ and some $J \in T_{\underline{\mathcal{P}}}\!\uparrow\omega(\mathcal{I}_\Phi) \cap \mathcal{I}_\Phi$ with $J \in \mathcal{S}_{I_1} \cap \mathcal{S}_{I_2}$. Then $J \supseteq I_1 \cup I_2$ and thus $J \supseteq T_{\underline{\mathcal{P}}}\!\uparrow\omega(I_1 \cup I_2)$. Moreover, it holds $T_{\underline{\mathcal{P}}}\!\uparrow\omega(I_1 \cup I_2) \in \mathcal{I}_{\mathcal{P}}$ by (γ) and $T_{\underline{\mathcal{P}}}\!\uparrow\omega(I_1 \cup I_2) \supset I_1$ or $T_{\underline{\mathcal{P}}}\!\uparrow\omega(I_1 \cup I_2) \supset I_2$. But this contradicts the assumption $J \in \mathcal{S}_{I_1} \cap \mathcal{S}_{I_2}$. Assume next that there are some $J \in T_{\underline{\mathcal{P}}}\!\uparrow\omega(\mathcal{I}_\Phi) \cap \mathcal{I}_\Phi$ that do not belong to $\bigcup\{\mathcal{S}_I \mid I \in \mathcal{I}_{\mathcal{P}}\}$. We now construct an infinite chain $I_0 \subset I_1 \subset \cdots$ of elements of $\mathcal{I}_{\mathcal{P}}$ as follows. Let us define $I_0 = T_{\underline{\mathcal{P}}}\!\uparrow\omega$. It then holds $I_0 \in \mathcal{I}_{\mathcal{P}}$ by (α) and also $J \supseteq I_0$. But, since $J \notin \mathcal{S}_{I_0}$, there must be some $I_1 \in \mathcal{I}_{\mathcal{P}}$ with $J \supseteq I_1$ and $I_1 \supset I_0$. This argumentation can now be continued in an infinite way. However, the number of subsets of HB_Φ is finite and we are thus arrived at a contradiction.

We next show that for all $I \in \mathcal{I}_{\mathcal{P}}$, all possible worlds $J \in T_{\underline{\mathcal{P}}}\!\uparrow\omega(\mathcal{I}_\Phi) \cap \mathcal{I}_\Phi$, and all ground conjunctive formulas C with $T_{\underline{\mathcal{P}}}\!\uparrow\omega(R(C)) \in \mathcal{I}_{\mathcal{P}}$, it holds $J \models C$ for *some* $J \in \mathcal{S}_I$ iff $J \models C$ for *all* $J \in \mathcal{S}_I$. Let $J \models C$ for some $J \in \mathcal{S}_I$. It then holds $J \supseteq I$, $J \supseteq R(C)$, and thus $J \supseteq T_{\underline{\mathcal{P}}}\!\uparrow\omega(R(C))$. We now show that $I \supseteq T_{\underline{\mathcal{P}}}\!\uparrow\omega(R(C))$. Assume first $I \subset T_{\underline{\mathcal{P}}}\!\uparrow\omega(R(C))$. But this contradicts $J \in \mathcal{S}_I$. Suppose next that $I \not\subseteq T_{\underline{\mathcal{P}}}\!\uparrow\omega(R(C))$ and $I \not\supseteq T_{\underline{\mathcal{P}}}\!\uparrow\omega(R(C))$. Since $J \supseteq I \cup T_{\underline{\mathcal{P}}}\!\uparrow\omega(R(C))$, we get $J \supseteq T_{\underline{\mathcal{P}}}\!\uparrow\omega(I \cup T_{\underline{\mathcal{P}}}\!\uparrow\omega(R(C)))$. Moreover, it holds $T_{\underline{\mathcal{P}}}\!\uparrow\omega(I \cup T_{\underline{\mathcal{P}}}\!\uparrow\omega(R(C))) \in \mathcal{I}_{\mathcal{P}}$ by (γ) and $T_{\underline{\mathcal{P}}}\!\uparrow\omega(I \cup T_{\underline{\mathcal{P}}}\!\uparrow\omega(R(C))) \supset I$. But this contradicts $J \in \mathcal{S}_I$. Hence, we get $I \supseteq T_{\underline{\mathcal{P}}}\!\uparrow\omega(R(C))$. Since $J \supseteq I$ for all $J \in \mathcal{S}_I$, we thus get $J \supseteq R(C)$ for all $J \in \mathcal{S}_I$. That is, $J \models C$ for all $J \in \mathcal{S}_I$. The converse trivially holds.

We are now ready to prove the theorem as follows. Let Pr be a model of $\mathcal{P}$. Let y_I ($I \in \mathcal{I}_{\mathcal{P}}$) be defined as the sum of all $Pr(J)$ with $J \in \mathcal{S}_I$. It is now easy to see that y_I ($I \in \mathcal{I}_{\mathcal{P}}$) is a solution of $\mathcal{LC}_{\mathcal{P}}$.

Conversely, let y_I ($I \in \mathcal{I}_{\mathcal{P}}$) be a solution of $\mathcal{LC}_{\mathcal{P}}$. Let the probabilistic interpretation Pr be defined by $Pr(I) = y_I$ if $I \in \mathcal{I}_{\mathcal{P}}$ and $Pr(I) = 0$ otherwise. It is easy to see that Pr is a model of all logical program clauses in $\mathcal{P}$ and of all purely probabilistic program clauses in *active*$(\mathcal{P})$. Let us now take a purely probabilistic program clause $(H|B)[c_1, c_2]$ from *ground*$(\mathcal{P}) \setminus$ *active*$(\mathcal{P})$. Assume that $R(B)$ contains some $B_i \notin T_{\overline{\mathcal{P}}}\!\uparrow\omega$. By Lemma 3.2, we then get $B_i \notin I$ for all $I \in \mathcal{I}_{\mathcal{P}}$. Hence, $Pr(B) = 0$ and thus $Pr \models (H|B)[c_1, c_2]$. Suppose now that $R(B) \subseteq T_{\overline{\mathcal{P}}}\!\uparrow\omega$ and that $R(H)$ contains some $H_i \notin T_{\overline{\mathcal{P}}}\!\uparrow\omega$. But this contradicts the assumption $c_2 > 0$. That is, Pr is a model of $\mathcal{P}$. □

Proof of Lemma 4.2: The claim can be proved like Lemma 3.2 (by induction on the definition of $\mathcal{I}_{\mathcal{P},Q}$). The proof makes use of $R(G) \cup R(A) \subseteq T_{\overline{\mathcal{P}}}\!\uparrow\omega$. □

Proof of Theorem 4.3: Since $active(\mathcal{P})$ does not involve any other atomic formulas than those in $T_{\overline{\mathcal{P}}}\!\uparrow\omega$, we can restrict our attention to probability functions over the set of possible worlds $\mathcal{I}_{\overline{\mathcal{P}}} = \{L \mid L \subseteq T_{\overline{\mathcal{P}}}\!\uparrow\omega\}$. Like in the proof of Theorem 3.3, we need some preparations as follows. We show that all purely probabilistic program clauses from $active(\mathcal{P})$ can be interpreted by probability functions over a partition $\{\mathcal{S}_I \mid I \in \mathcal{I}_{\mathcal{P},Q}\}$ of $\mathcal{I}_{\overline{\mathcal{P}}}$:

For all $I \in \mathcal{I}_{\mathcal{P},Q}$ let $\mathcal{S}_I$ be the set of all possible worlds $J \in T_{\underline{\mathcal{P}}}\!\uparrow\omega(\mathcal{I}_{\overline{\mathcal{P}}}) \cap \mathcal{I}_{\overline{\mathcal{P}}}$ that are a superset of I and that are not a superset of any $K \in \mathcal{I}_{\mathcal{P},Q}$ that properly includes I. By an argumentation like in the proof of Theorem 3.3, it can easily be shown that $\{\mathcal{S}_I \mid I \in \mathcal{I}_{\mathcal{P},Q}\}$ is a partition of $T_{\underline{\mathcal{P}}}\!\uparrow\omega(\mathcal{I}_{\overline{\mathcal{P}}}) \cap \mathcal{I}_{\overline{\mathcal{P}}}$ and that for all $I \in \mathcal{I}_{\mathcal{P},Q}$, all possible worlds $J \in T_{\underline{\mathcal{P}}}\!\uparrow\omega(\mathcal{I}_{\overline{\mathcal{P}}}) \cap \mathcal{I}_{\overline{\mathcal{P}}}$, and all ground conjunctive formulas C with $T_{\underline{\mathcal{P}}}\!\uparrow\omega(R(C)) \in \mathcal{I}_{\mathcal{P},Q}$, it holds $J \models C$ for *some* $J \in \mathcal{S}_I$ iff $J \models C$ for *all* $J \in \mathcal{S}_I$.

Given a model Pr of $active(\mathcal{P})$, we can thus define a model $Pr^\star$ of $active(\mathcal{P})$ by $Pr^\star(L) = 1 / a_I \cdot \sum_{J \in \mathcal{S}_I} Pr(J)$ if $L \in T_{\underline{\mathcal{P}}}\!\uparrow\omega(\mathcal{I}_{\overline{\mathcal{P}}}) \cap \mathcal{I}_{\overline{\mathcal{P}}}$, where $I \in \mathcal{I}_{\mathcal{P},Q}$ such that $L \in \mathcal{S}_I$, and $Pr^\star(L) = 0$ otherwise. Hence, for all $I \in \mathcal{I}_{\mathcal{P},Q}$ and all $J_1, J_2 \in \mathcal{S}_I$: $ME[active(\mathcal{P})](J_1) = ME[active(\mathcal{P})](J_2)$.

Hence, for all ground conjunctive formulas C with $T_{\underline{\mathcal{P}}}\!\uparrow\omega(R(C)) \in \mathcal{I}_{\mathcal{P},Q}$:

$$ME[active(\mathcal{P})](C) \;=\; \textstyle\sum_{I \in \mathcal{I}_{\mathcal{P},Q},\, I \models C} a_I x_I^\star\,,$$

where $x_I^\star$ with $I \in \mathcal{I}_{\mathcal{P},Q}$ is the optimal solution of the following optimization problem over the variables $x_I \geq 0$ with $I \in \mathcal{I}_{\mathcal{P},Q}$:

$$\max \quad -\sum_{I \in \mathcal{I}_{\mathcal{P},Q}} a_I x_I \log x_I \text{ subject to } \mathcal{LC}'_{\mathcal{P},Q},$$

where $\mathcal{LC}'_{\mathcal{P},Q}$ is the least set of constraints over $x_I \geq 0$ with $I \in \mathcal{I}_{\mathcal{P},Q}$ containing:

(1) $\sum_{I \in \mathcal{I}_{\mathcal{P},Q}} a_I x_I \;=\; 1$

(2) $c_1 \cdot \sum_{I \in \mathcal{I}_{\mathcal{P},Q},\, I \models B} a_I x_I \;\leq\; \sum_{I \in \mathcal{I}_{\mathcal{P},Q},\, I \models H \wedge B} a_I x_I \;\leq\; c_2 \cdot \sum_{I \in \mathcal{I}_{\mathcal{P},Q},\, I \models B} a_I x_I$

for all purely probabilistic program clauses $(H|B)[c_1, c_2] \in active(\mathcal{P})$.

Thus, we finally just have to perform the variable substitution $x_I = y_I / a_I$. □

References

[1] K. R. Apt. Logic programming. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 10, pages 493–574. MIT Press, 1990.

[2] F. Bacchus. *Representing and Reasoning with Probabilistic Knowledge: A Logical Approach to Probabilities.* MIT Press, Cambridge, USA, 1990.

[3] F. Bacchus, A. Grove, J. Y. Halpern, and D. Koller. From statistical knowledge bases to degrees of beliefs. *Artif. Intell.*, 87:75–143, 1996.

[4] R. Carnap. *Logical Foundations of Probability.* University of Chicago Press, Chicago, 1950.

[5] P. Cheeseman. A method of computing generalized Bayesian probability values for expert systems. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, pages 198–202, Karlsruhe, Germany, 1983.

[6] B. de Finetti. *Theory of Probability.* J. Wiley, New York, 1974.

[7] A. Dekhtyar and V. S. Subrahmanian. Hybrid probabilistic programs. In *Proc. of the 14th International Conference on Logic Programming*, pages 391–405, 1997.

[8] H. B. Enderton. *A Mathematical Introduction to Logic.* Academic Press, New York, 1972.

[9] R. Fagin, J. Y. Halpern, and N. Megiddo. A logic for reasoning about probabilities. *Inf. Comput.*, 87:78–128, 1990.

[10] V. Fischer and M. Schramm. tabl — a tool for efficient compilation of probabilistic constraints. Technical Report TUM-I9636, TU München, 1996.

[11] A. J. Grove, J. H. Halpern, and D. Koller. Random worlds and maximum entropy. *J. Artif. Intell. Res.*, 2:33–88, 1994.

[12] P. Haddawy. Generating Bayesian networks from probability logic knowledge bases. In *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, pages 262–269. Morgan Kaufmann, 1994.

[13] J. Y. Halpern. An analysis of first-order logics of probability. *Artif. Intell.*, 46:311–350, 1990.

[14] M. Jaeger. Relational Bayesian networks. In *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence*, pages 266–273. Morgan Kaufmann, 1997.

[15] E. T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106:620–30, 1957.

[16] E. T. Jaynes. *Papers on Probability, Statistics and Statistical Physics.* D. Reidel, Dordrecht, Holland, 1983.

[17] R. W. Johnson and J. E. Shore. Comments on and corrections to "Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy". *IEEE Trans. Inf. Theory*, IT-29(6):942–943, 1983.

[18] G. Kern-Isberner. Characterizing the principle of minimum cross-entropy within a conditional-logical framework. *Artif. Intell.*, 98:169–208, 1998.

[19] J. W. Lloyd. *Foundations of Logic Programming.* Springer, Berlin, 2nd ed., 1987.

[20] T. Lukasiewicz. Probabilistic logic programming. In *Proceedings of the 13th European Conference on Artificial Intelligence*, pages 388–392. J. Wiley & Sons, 1998.

[21] T. Lukasiewicz. Local probabilistic deduction from taxonomic and probabilistic knowledge-bases over conjunctive events. Accepted for publication in *Int. J. Approx. Reasoning*, 1999. To appear.

[22] T. Lukasiewicz. Probabilistic deduction with conditional constraints over basic events. Accepted for publication in *J. Artif. Intell. Res.*, 1999. To appear.

[23] R. T. Ng. Semantics, consistency, and query processing of empirical deductive databases. *IEEE Trans. Knowl. Data Eng.*, 9(1):32–49, 1997.

[24] R. T. Ng and V. S. Subrahmanian. Probabilistic logic programming. *Inf. Comput.*, 101:150–201, 1992.

[25] R. T. Ng and V. S. Subrahmanian. A semantical framework for supporting subjective and conditional probabilities in deductive databases. *J. Autom. Reasoning*, 10(2):191–235, 1993.

[26] R. T. Ng and V. S. Subrahmanian. Stable semantics for probabilistic deductive databases. *Inf. Comput.*, 110:42–83, 1994.

[27] N. J. Nilsson. Probabilistic logic. *Artif. Intell.*, 28:71–88, 1986.

[28] J. B. Paris and A. Vencovska. On the applicability of maximum entropy to inexact reasoning. *Int. J. Approx. Reasoning*, 3:1–34, 1989.

[29] J. B. Paris and A. Vencovska. A note on the inevitability of maximum entropy. *Int. J. Approx. Reasoning*, 14:183–223, 1990.

[30] J. B. Paris and A. Vencovska. In defense of the maximum entropy inference process. *Int. J. Approx. Reasoning*, 17:77–103, 1997.

[31] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, CA, 1988.

[32] D. Poole. Probabilistic Horn abduction and Bayesian networks. *Artif. Intell.*, 64:81–129, 1993.

[33] W. Rödder and G. Kern-Isberner. Representation and extraction of information by probabilistic logic. *Inf. Syst.*, 21(8):637–652, 1997.

[34] W. Rödder and C.-H. Meyer. Coherent knowledge processing at maximum entropy by SPIRIT. In *Proceedings 12th Conference on Uncertainty in Artificial Intelligence*, pages 470–476. Morgan Kaufmann, 1996.

[35] M. Schramm, V. Fischer, and P. Trunk. Probabilistic knowledge representation and reasoning with maximum entropy - the system PIT, Technical report (forthcoming), 1999.

[36] C. E. Shannon and W. Weaver. *A Mathematical Theory of Communication.* University of Illinois Press, Urbana, Illinois, 1949.

[37] J. E. Shore and R. W. Johnson. Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. *IEEE Trans. Inf. Theory*, IT-26:26–37, 1980.