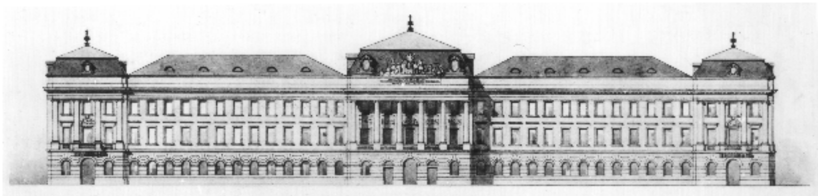
I N F S Y S

R E S E A R C H

R E P O R T



INSTITUT FÜR INFORMATIONSSYSTEME

ABTEILUNG WISSENSBASIERTE SYSTEME

COMBINING PROBABILISTIC LOGIC PROGRAMMING WITH THE POWER OF MAXIMUM ENTROPY

Gabriele KERN-ISBERNER Thomas LUKASIEWICZ

INFSYS RESEARCH REPORT 1843-02-12
OCTOBER 2002

Institut für Informationssysteme
Abtg. Wissensbasierte Systeme
Technische Universität Wien
Favoritenstraße 9-11
A-1040 Wien, Austria
Tel: +43-1-58801-18405
Fax: +43-1-58801-18493
sek@kr.tuwien.ac.at
www.kr.tuwien.ac.at

TECHNISCHE UNIVERSITÄT WIEN

INFSYS RESEARCH REPORT

INFSYS RESEARCH REPORT 1843-02-12, OCTOBER 2002

COMBINING PROBABILISTIC LOGIC PROGRAMMING WITH THE POWER OF MAXIMUM ENTROPY

Gabriele Kern-Isberner [1] and Thomas Lukasiewicz [2]

Abstract. This paper is on the combination of two powerful approaches to uncertain reasoning: logic programming in a probabilistic setting, on the one hand, and the information-theoretical principle of maximum entropy, on the other hand. More precisely, we present two approaches to probabilistic logic programming under maximum entropy. The first one is based on the usual notion of entailment under maximum entropy, and is defined for the very general case of probabilistic logic programs over Boolean events. The second one is based on a new notion of entailment under maximum entropy, where the principle of maximum entropy is coupled with the closed world assumption (CWA) from classical logic programming. It is only defined for the more restricted case of probabilistic logic programs over conjunctive events. We then analyze the nonmonotonic behavior of both approaches along benchmark examples and along general properties for default reasoning from conditional knowledge bases. It turns out that both approaches have very nice nonmonotonic features. In particular, they realize some inheritance of probabilistic knowledge along subclass relationships, without suffering from the problem of inheritance blocking and from the drowning problem. They both also satisfy the property of rational monotonicity and several irrelevance properties. We finally present algorithms for both approaches, which are based on generalizations of techniques from probabilistic logic programming under logical entailment in [45]. The algorithm for the first approach still produces quite large weighted entropy maximization problems, while the one for the second approach generates optimization problems of the same size as the ones produced by probabilistic logic programming under logical entailment in [45].

[1]Fachbereich Informatik, FernUniversität Hagen, P.O. Box 940, D-58084 Hagen, Germany; e-mail: gabriele.kern-isberner@fernuni-hagen.de.

[2]Dipartimento di Informatica e Sistemistica, Università di Roma “La Sapienza”, Via Salaria 113, 00198 Rome, Italy; e-mail: lukasiewicz@dis.uniroma1.it. Alternate address: Institut für Informationssysteme, Technische Universität Wien, Favoritenstraße 9-11, 1040 Vienna, Austria; e-mail: lukasiewicz@kr.tuwien.ac.at.

Acknowledgements: This work has been partially supported by the Austrian Science Fund under project N Z29-INF, by a DFG grant, and by a Marie Curie Individual Fellowship of the European Community (Disclaimer: The authors are solely responsible for information communicated and the European Commission is not responsible for any views or results expressed). We are very grateful to the reviewers of the ECSQARU-99 abstract of this paper [48], whose constructive comments helped to improve our work.

Contents

1 Introduction

A number of recent research efforts are directed towards integrating logic-oriented and probability-based representation and reasoning formalisms. Probabilistic propositional logics and their various dialects have been thoroughly studied in the literature (see especially the work by Nilsson [56], Fagin et al. [15], Dubois and Prade et al. [12, 11], Frisch and Haddawy [17], and the second author [41, 42]). 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 the work by Bacchus et al. [2] and Halpern [23]). 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*. In particular, Ng [52] concentrates on probabilities over the domain, while Subrahmanian and his group [53, 54, 8, 9] focus on annotation-based approaches to degrees of belief. Another approach to probabilistic logic programming with degrees of belief, which is especially directed towards efficient implementations, has been recently introduced in [40, 45]. The following shows a very simple probabilistic logic program as in [40, 45], which expresses that "all penguins are birds" and that "birds have legs with a probability of at least 0.98":

$$\begin{aligned} P \;=\; & \{(\mathit{bird}(T) \mid \mathit{penguin}(T))[1,1], \\ & (\mathit{have\_legs}(T) \mid \mathit{bird}(T))[0.98,1]\}\,. \end{aligned}$$

Nearly all the above approaches of integrating logic and probability are based on the notion of model-theoretic logical entailment. This notion, however, has often been criticized in the literature for its inferential weakness. For example, the tight probability interval that follows under model-theoretic logical entailment from the above program P for "Tweety has legs, given that Tweety is a penguin" is given by the uninformative interval $[0,1]$. For this reason, many recent approaches towards integrating logic and probabilities combine logic-based formalisms with Bayesian networks [60]. In particular, Poole's work [63, 62] describes an approach to Horn clause abduction in which probabilities are associated with hypotheses. It is implemented by a generalization of SLD resolution. Haddawy and his group [22, 55] describe an approach to query processing in first-order probabilistic knowledge bases by Bayesian network construction and inference. Jaeger's work [24, 25] goes in a similar direction. There is less closely related work on object-oriented Bayesian networks by Koller and Pfeffer [32] and by Laskey and Mahoney [36] where methods from object-oriented programming languages are used to enable flexible and large-scale knowledge representation with Bayesian networks.

Another promising way towards stronger entailment relations are the recent approaches to probabilistic default reasoning with conditional constraints in [47] and to probabilistic logic programming under inheritance with overriding in [44]. These approaches are based on new notions of entailment for reasoning with conditional constraints, which are obtained from the classical notion of logical entailment by adding inheritance with overriding, using techniques from default reasoning with conditional constraints. For example, under probabilistic default entailment, the tight probability interval that follows from the above program P for "Tweety has legs, given that Tweety is a penguin" is given by $[0.98, 1]$. The new notions of entailment have very nice nonmonotonic properties, and they can also be used in reasoning from statistical knowledge and degrees of belief [47]. A companion paper [46] explores related notions of entailment for conditioning on zero events.

A further way to overcome the inferential weakness of model-theoretic logical entailment is to use the principle of maximum entropy. In general, the available probabilistic knowledge does not suffice to completely specify a unique probability distribution. Rather, it specifies a large set of probability distributions, giving rise to an inconvenient impreciseness of the inferred probability intervals. Applying the *principle of maximum entropy* is a well-appreciated means to improve probabilistic inference, both from a statistical point of view, and for commonsense reasons [58]. Entropy is an information-theoretical measure [75] reflecting the indeterminateness inherent to a distribution. Given some satisfiable set of probabilistic facts and rules, the principle of maximum entropy chooses as the most appropriate representation the unique distribution, among all the distributions satisfying those formulas, that has maximum entropy. For example, under maximum entropy, the above program P entails the probability 0.98 for "Tweety has legs, given that Tweety is a penguin". Within a rich statistical first-order language, Grove et al. [21] show that this maximum entropy distribution may be taken to compute degrees of belief of formulas. Paris and Vencovskà [59] investigate the foundations of consistent probabilistic inference and set up postulates that characterize maximum entropy inference uniquely within that framework. A similar result was stated in [76], based on optimization theory. Jaynes [26] regarded the principle of maximum entropy as a special case of a more general principle for translating information into a probability assignment. One of the authors [29] proved it to be the most appropriate principle for dealing with conditionals (that is, using the notions of the present paper, ground conditional constraints of the form $(\psi|\phi)[c, c]$), and worked out its relevance also for qualitative approaches to uncertain reasoning [31].

The main idea of this paper is to elaborate an approach to probabilistic logic programming that is based on the inferentially powerful notion of entailment under maximum entropy. We thus follow an old idea that is already stated in the important work by Nilsson [56], however, lifted to the first-order framework of probabilistic logic programs. To our knowledge, this is the first work in the literature on *probabilistic logic programming under maximum entropy*.

Our research in this paper is directed towards an approach to probabilistic logic programming under maximum entropy, which has nice nonmonotonic properties and in the same time also nice computational features. At first sight, this project might seem a very hard task, especially from the computational point of view, since already propositional probabilistic logics under maximum entropy suffer from efficiency problems (due to an exponential number of possible worlds involved in the optimization process). In this paper, however, we will see that this is not the case. In particular, we show that the efficient approach to probabilistic logic programming in [45], refined by new ideas, and combined with a new notion of entailment under maximum entropy, can be extended to an efficient and semantically appealing approach to probabilistic logic programming under maximum entropy.

More generally, in this paper, we consider the following two approaches to probabilistic logic programming under maximum entropy:

- Our first approach combines the efficient approach to probabilistic logic programming in [45] with the classical notion of entailment under maximum entropy. This approach is defined for the very general case of probabilistic logic programs over Boolean events. We show that it has very nice nonmonotonic properties. However, it produces quite large entropy maximization problems.

- Our second approach combines the approach in [45] with a new notion of entailment under maximum entropy, where the principle of maximum entropy is coupled with the *closed world assumption (CWA)* from classical logic programming. This approach is defined for the more restricted case of probabilistic logic programs over conjunctive events. It has nearly the same nice nonmonotonic properties as our the first approach, but it has also nice computational features.

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

- We recall the syntax of probabilistic logic programs from [45] and define their new semantics through the classical notion of entailment under maximum entropy (or me-entailment) and the new notion of entailment under maximum entropy and CWA (or mc-entailment). We provide several examples that show the relevance of probabilistic logic programs in practice, and that give a comparative view on me-entailment, mc-entailment, and logical entailment. In particular, they show that there are cases where logical entailment is simply too weak.

- We explore the nonmonotonic behavior of me-entailment, mc-entailment, and logical entailment in standard benchmark examples from default reasoning from conditional knowledge bases. It turns out that me- and mc-entailment have very nice properties. Differently from logical entailment, they both inherit probabilistic knowledge along subclass relationships, and they ignore irrelevant knowledge. Moreover, they show neither the problem of inheritance blocking, nor the drowning problem. We also study the behavior of the three entailment relations in the case of conflicting and non-conflicting information.

- We explore the general nonmonotonic properties of the notions of me-entailment, mc-entailment, and logical entailment. Also here, me- and mc-entailment behave very nicely. In particular, we show that both me-entailment and logical entailment satisfy all the postulates of System P, while mc-entailment satisfies nearly all of them. Moreover, all three notions of entailment have the direct inference property. Furthermore, me- and mc-entailment both satisfy the rational monotonicity property, and they have some irrelevance and strong irrelevance properties, while logical entailment is lacking all these properties.

- As for the relationship between me-entailment, mc-entailment, and logical entailment, we show that both me- and mc-entailment are stronger than logical entailment, and that they coincide on non-probabilistic conclusions. Furthermore, all three notions of entailment are probabilistic generalizations of model-theoretic logical entailment in classical propositional logics.

- We present algorithms for the problems POSITIVE PROBABILITY (given a probabilistic logic program P and a ground event α, decide whether P has a model Pr with $Pr(\alpha) > 0$) and TIGHT 0-CONSEQUENCE (given a probabilistic logic program P and a ground conditional event $\beta|\alpha$, compute the reals $l, u \in [0, 1]$ such that $(\beta|\alpha)[l, u]$ is a tight logical consequence of P). They reduce POSITIVE PROBABILITY and TIGHT 0-CONSEQUENCE to linear optimization problems, and they make use of the following techniques for an increased efficiency: (i) making hidden classical knowledge explicit, (ii) removing vacuous conditional constraints, (iii) removing inactive conditional constraints, (iv) decomposing a probabilistic logic program, (v) exploiting classical knowledge, and (vi) clustering possible worlds. Here, (i)–(iii), (v), and (vi) are refinements of implicit techniques in [45], while (iv) is inspired by similar methods in [49, 14].

- We present algorithms for the tasks TIGHT me-CONSEQUENCE (resp., TIGHT mc-CONSEQUENCE): Given a probabilistic logic program P and a ground conditional event $\beta|\alpha$, compute the reals $l, u \in [0,1]$ such that $(\beta|\alpha)[l, u]$ is a tight me-consequence (resp., mc-consequence) of P. They reduce these problems to entropy (resp., weighted entropy) maximizations subject to a system of linear constraints, and they make use of the above techniques (i), (ii), and (iv)–(vi) (resp., the above techniques (i)–(vi)), which are shown to carry over to solving TIGHT me-CONSEQUENCE (resp., TIGHT mc-CONSEQUENCE). In particular, this shows that TIGHT mc-CONSEQUENCE and TIGHT 0-CONSEQUENCE are reduced to two optimization problems of the same size, while TIGHT me-CONSEQUENCE is reduced to a significantly larger optimization problem, since the technique (iii) of removing inactive conditional constraints cannot be applied.

A number of examples throughout the paper shed light on many interesting aspects of maximum entropy reasoning, and illustrate the presented techniques. In particular, we give an example of a small probabilistic logic program where the size of the generated system of linear constraints drops down from 120 linear constraints over $2^{64} \approx 18 \cdot 10^{18}$ (!) variables in the naive characterization to 6 linear constraints over 6 (resp., 7) variables in the characterization produced by our algorithms for POSITIVE PROBABILITY (resp., TIGHT 0- and mc-CONSEQUENCE).

The rest of this paper is organized as follows. In Section 2, we give some technical preliminaries. Section 3 introduces the entailment semantics for probabilistic logic programs that we consider in this paper. In Section 4, we give some examples that show the relevance of probabilistic logic programs in practice. Section 5 then analyzes the semantic properties of the discussed entailment semantics for probabilistic logic programs. In Section 6, we give naive algorithms for probabilistic logic programming under the entailment semantics of this paper. Sections 7–9 then present more sophisticated techniques. In Section 10, we finally summarize the main results and give an outlook on future research.

In order to not distract from the flow of reading, some technical details and proofs have been moved to Appendices A–D.

2 Preliminaries

In this section, we first describe the probabilistic background of this work. We then define the syntax of probabilistic logic programs and of probabilistic queries to probabilistic logic programs. We next define the meaning of probabilistic queries, using notions of entailment for probabilistic logic programs, and we finally describe the probabilistic logic programming tasks that we especially focus on in this paper.

2.1 Probabilistic Background

We now briefly describe how first-order logics of probability are given a semantics in which probabilities are defined over a set of possible worlds (cf. especially the work by Carnap [4], Gaifman [18], Scott and Krauss [74], and Halpern [23]). We restrict our considerations to a language of first-order Boolean combinations of conditional constraints that are implicitly universally quantified and that are interpreted by probabilities over a set of Herbrand interpretations.

Let Φ be a first-order vocabulary that contains a finite set of predicate symbols and a finite set of constant symbols (that is, we do not consider function symbols in the framework of this paper). 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 unit interval $[0, 1]$.

An *object term* is either a constant symbol from Φ or an object variable from $\mathcal{X}$. We define *events* by induction as follows. The propositional constants *false* and *true*, denoted $\perp$ and $\top$, respectively, are events. If p is a predicate symbol of arity $k \geq 0$ from Φ and $t_1, \ldots, t_k$ are object terms, then $p(t_1, \ldots, t_k)$ is an event (called *atom*). If ϕ and ψ are events, then also $\neg\phi$ and $(\phi \wedge \psi)$. A *conditional event* is an expression of the kind $\psi|\phi$ with events ψ and ϕ. A *conditional constraint* is an expression of the form $(\psi|\phi)[l, u]$ with real numbers $l, u \in [0, 1]$ and events ψ and ϕ. We call ψ its *consequent* (or *head*) and ϕ its *antecedent* (or *body*). We define *probabilistic formulas* by induction as follows. Every conditional constraint is a probabilistic formula. If F and G are probabilistic formulas, then also $\neg F$ and $(F \wedge G)$. We use $(F \vee G)$, $(F \Leftarrow G)$, and $(F \Leftrightarrow G)$ to abbreviate $\neg(\neg F \wedge \neg G)$, $\neg(\neg F \wedge G)$, and $(\neg(\neg F \wedge G) \wedge \neg(F \wedge \neg G))$, respectively, where F and G are either two events or two probabilistic formulas. We eliminate parentheses as usual. Object terms, events, conditional events, and probabilistic formulas are *ground* iff they do not contain any variables. The notions of substitutions, ground substitutions, and ground instances of conditional events and probabilistic formulas are canonically defined.

We distinguish between classical and purely probabilistic conditional constraints. *Classical conditional constraints* are of the kind $(\psi|\phi)[1,1]$ or $(\psi|\phi)[0,0]$, while *purely probabilistic conditional constraints* are of the form $(\psi|\phi)[l,u]$ with $l < 1$ and $u > 0$. We often identify the classical conditional constraint $(\psi|\phi)[1,1]$ (resp., $(\psi|\phi)[0,0]$) with the event $\psi \Leftarrow \phi$ (resp., $\bot \Leftarrow \psi \wedge \phi$).

We use HB_Φ (resp., HU_Φ) to denote the Herbrand base (resp., Herbrand universe) over Φ. In the sequel, we assume that HB_Φ is nonempty. A *possible world* I is a subset of HB_Φ. We use $\mathcal{I}_\Phi$ to denote the set of all possible worlds over Φ. A *variable assignment* σ maps each object variable to an element of HU_Φ, and each bound variable to a real number from $[0,1]$. It is extended to object terms by $\sigma(c) = c$ for all constant symbols c from Φ. The *truth* of events ϕ in I under σ, denoted $I \models_\sigma \phi$, is inductively defined as follows (we write $I \models \phi$ when ϕ is ground):

- $I \models_\sigma p(t_1, \ldots, t_k)$ iff $p(\sigma(t_1), \ldots, \sigma(t_k)) \in I$;
- $I \models_\sigma \neg\phi$ iff not $I \models_\sigma \phi$;
- $I \models_\sigma (\phi \wedge \psi)$ iff $I \models_\sigma \phi$ and $I \models_\sigma \psi$.

An event ϕ *is true* in a possible world I, or I is a *model* of ϕ, denoted $I \models \phi$, iff $I \models_\sigma \phi$ for all variable assignments σ. A possible world I is a *model* of a set of events $\mathcal{F}$ iff I is a model of all $\phi \in \mathcal{F}$. A set of events $\mathcal{F}$ is *satisfiable* iff a model of $\mathcal{F}$ exists. An event ϕ is a *logical consequence* of $\mathcal{F}$, denoted $\mathcal{F} \models \phi$, iff each model of $\mathcal{F}$ is also a model of ϕ. We use $\mathcal{F} \not\models \phi$ to denote that $\mathcal{F} \models \phi$ does not hold.

A *probabilistic interpretation* Pr is a probability function on $\mathcal{I}_\Phi$ (that is, as $\mathcal{I}_\Phi$ is finite, a mapping from $\mathcal{I}_\Phi$ to the unit interval $[0,1]$ such that all $Pr(I)$ with $I \in \mathcal{I}_\Phi$ sum up to 1). The *probability* of an event ϕ in the probabilistic interpretation Pr under a variable assignment σ, denoted $Pr_\sigma(\phi)$, is the sum of all $Pr(I)$ such that $I \in \mathcal{I}_\Phi$ and $I \models_\sigma \phi$ (we write $Pr(\phi)$ when ϕ is ground). For events ϕ and ψ with $Pr_\sigma(\phi) > 0$, we then define $Pr_\sigma(\psi|\phi) = Pr_\sigma(\psi \wedge \phi) \,/\, Pr_\sigma(\phi)$. For ground events ϕ with $Pr(\phi) > 0$, the *conditioning* of Pr on ϕ, denoted Pr_ϕ, is defined by $Pr_\phi(I) = Pr(I) \,/\, Pr(\phi)$ for all $I \in \mathcal{I}_\Phi$ with $I \models \phi$, and by $Pr_\phi(I) = 0$ for all other $I \in \mathcal{I}_\Phi$. The *truth* of a probabilistic formula F in a probabilistic interpretation Pr under a variable assignment σ, denoted $Pr \models_\sigma F$, is defined by induction as follows:

- $Pr \models_\sigma (\psi|\phi)[l,u]$ iff $Pr_\sigma(\phi) = 0$ or $Pr_\sigma(\psi|\phi) \in [l,u]$;
- $Pr \models_\sigma \neg F$ iff not $Pr \models_\sigma F$;
- $Pr \models_\sigma (F \wedge G)$ iff $Pr \models_\sigma F$ and $Pr \models_\sigma G$.

A probabilistic formula F *is true* in a probabilistic interpretation Pr, or Pr is a *model* of F, denoted $Pr \models F$, iff $Pr \models_\sigma F$ for all variable assignments σ. A probabilistic interpretation Pr is a *model* of a set of probabilistic formulas $\mathcal{F}$, denoted $Pr \models \mathcal{F}$, iff Pr is a model of all $F \in \mathcal{F}$. A set of probabilistic formulas $\mathcal{F}$ is *satisfiable* iff a model of $\mathcal{F}$ exists.

2.2 Syntax of Probabilistic Logic Programs

A *probabilistic logic program* P is a finite set of conditional constraints of the form $(\psi|\phi)[l,u]$ where $l \leq u$. We use $ground(P)$ to denote the set of all ground instances of conditional constraints in P. We use HB_P to denote the set of all ground atoms $p \in HB_\Phi$ that occur in $ground(P)$. A *probabilistic query* is an expression of the form $\exists(\beta|\alpha)[s,t]$, where α and β are two events, and s and t are either two real numbers from $[0,1]$ or two distinct bound variables from $\mathcal{X}$. A probabilistic query $\exists(\beta|\alpha)[s,t]$ is *object-ground* iff α and β are ground and $s,t \in \mathcal{X}$.

An event ϕ is *conjunctive* iff ϕ is either $\top$ or a conjunction of atoms. A conditional event $\psi|\phi$ is *conjunctive* (resp., *1-conjunctive*) iff ψ is a conjunction of atoms (resp., an atom) and ϕ is conjunctive. A conditional constraint $(\psi|\phi)[l,u]$ is *conjunctive* (resp., *1-conjunctive*) iff $\psi|\phi$ is conjunctive (resp., 1-conjunctive). A probabilistic logic program P is *conjunctive* iff all $F \in P$ are conjunctive. A probabilistic query $\exists(\beta|\alpha)[s,t]$ is *conjunctive* iff $\beta|\alpha$ is conjunctive.

Conjunctive conditional constraints $(\psi|\phi)[l,u]$ with $l \leq u$ are also called *probabilistic Horn clauses*. They are classified into *integrity clauses* and *probabilistic program clauses*, which are of the form $(\psi|\phi)[0,0]$ and $(\psi|\phi)[l,u]$, respectively, where $u > 0$. The latter are divided into *probabilistic facts* and *probabilistic rules*, which are of the form $(\psi|\top)[l,u]$ and $(\psi|\phi)[l,u]$, respectively, where $\phi \neq \top$.

A *classical Horn clause* is a classical 1-conjunctive conditional constraint of the form $(\psi|\phi)[1,1]$ or $(\psi|\phi)[0,0]$. A *classical logic program* is a finite set of classical Horn clauses. A *classical program clause* is a classical Horn clause of the kind $(\psi|\phi)[1,1]$. A *classical definite logic program* is a finite set of classical program clauses. Finally, *classical facts* and *classical rules* are classical program clauses of the form $(\psi|\top)[1,1]$ and $(\psi|\phi)[1,1]$, respectively, where $\phi \neq \top$.

2.3 Semantics of Probabilistic Logic Programs

To define the meaning of probabilistic queries to probabilistic logic programs, we first have to define a notion of entailment for probabilistic logic programs. There are several different such notions. Each notion of entailment s is associated with a consequence relation $\Vert\!\!\sim^s$ and a tight consequence relation $\Vert\!\!\sim^s_{tight}$, which relate probabilistic logic programs with their entailed conditional constraints.

In order to specify a notion of entailment s, it is sufficient to only define the consequence relation $\Vert\!\!\sim^s$ for the ground case. The tight consequence relation $\Vert\!\!\sim^s_{tight}$ for the ground case is then canonically defined by $P \Vert\!\!\sim^s_{tight} (\beta|\alpha)[l,u]$ iff l (resp., u) is the supremum (resp., infimum) of a (resp., b) subject to $P \Vert\!\!\sim^s (\beta|\alpha)[a,b]$. Furthermore, the relations $\Vert\!\!\sim^s$ and $\Vert\!\!\sim^s_{tight}$ are naturally extended to the non-ground case as follows. For all probabilistic logic programs P and all conditional constraints $(\beta|\alpha)[l,u]$, we define $P \Vert\!\!\sim^s (\beta|\alpha)[l,u]$ iff $ground(P) \Vert\!\!\sim^s (\beta'|\alpha')[l,u]$ for all ground instances $\beta'|\alpha'$ of $\beta|\alpha$. We define $P \Vert\!\!\sim^s_{tight} (\beta|\alpha)[l,u]$ iff l (resp., u) is the supremum (resp., infimum) of a (resp., b) subject to $ground(P)$ $\Vert\!\!\sim^s_{tight} (\beta'|\alpha')[a,b]$ and all ground instances $\beta'|\alpha'$ of $\beta|\alpha$.

We are now ready to define the meaning of probabilistic queries to probabilistic logic programs under some notion of entailment s. Given a probabilistic query $\exists(\beta|\alpha)[l,u]$ with $l,u \in [0,1]$ to a probabilistic logic program P, its *correct answer substitutions under* s are substitutions θ such that $P \Vert\!\!\sim^s (\beta\theta|\alpha\theta)[l,u]$ and that θ acts only on variables in $\exists(\beta|\alpha)[l,u]$. Its *correct answer under* s is Yes if such a θ exists and No otherwise. Given a probabilistic query $\exists(\beta|\alpha)[x,y]$ with $x,y \in \mathcal{X}$ to a probabilistic logic program P, its *tight answer substitutions under* s are substitutions θ such that $P \Vert\!\!\sim^s_{tight} (\beta\theta|\alpha\theta)[x\theta, y\theta]$, that θ acts only on variables in $\exists(\beta|\alpha)[x,y]$, and that $x\theta, y\theta \in [0,1]$. Note that for probabilistic queries $\exists(\beta|\alpha)[x,y]$ with $x,y \in \mathcal{X}$, there always exist tight answer substitutions (in particular, object-ground queries always have a unique tight answer substitution).

2.4 Problem Statements

In this paper, we especially concentrate on the following two important decision and optimization problems related to probabilistic logic programs:

POSITIVE PROBABILITY: Given a probabilistic logic program P and a ground event ϕ, decide whether P has a model Pr such that $Pr(\phi) > 0$.

TIGHT s-CONSEQUENCE: Given a probabilistic logic program P and an object-ground probabilistic query $Q = \exists(\beta|\alpha)[x,y]$, compute the tight answer substitution for Q to P under a fixed notion of entailment s.

Observe that POSITIVE PROBABILITY is a generalization of the problem of deciding whether a probabilistic logic program is satisfiable, since a probabilistic logic program P is satisfiable iff it has a model Pr such that $Pr(\top) > 0$.

Notice also that, differently from classical definite logic programs, conjunctive probabilistic logic programs may be unsatisfiable, because of logical inconsistencies through integrity clauses or, more generally, because of probabilistic inconsistencies in the assumed probability ranges.

3 Entailment Semantics

In this section, we describe the notions of entailment for probabilistic logic programs that we focus on in this paper. We first define the classical notion of model-theoretic logical entailment, also called 0-*entailment*. We then recall the classical notion of entailment under maximum entropy, and we finally introduce a new notion of entailment under maximum entropy, which adopts the closed world assumption (CWA) from classical logic programming. They are called me-*entailment* and mc-*entailment*, respectively, and are both stronger than 0-entailment (see Fig. 1).

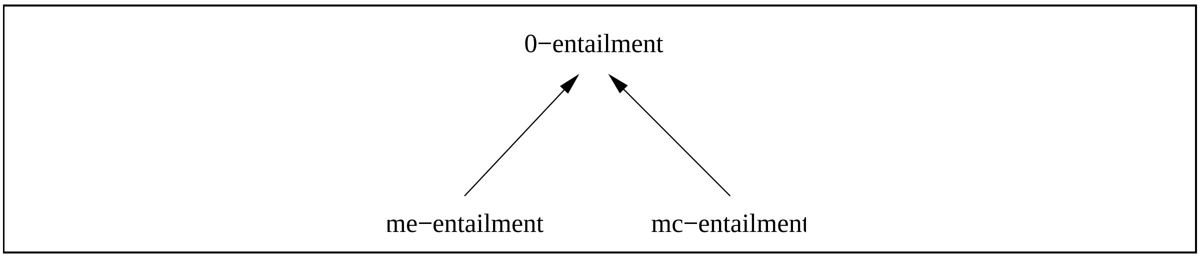


Figure 1: Entailment semantics for probabilistic logic programs.

3.1 Logical Entailment

We now describe the classical notion of model-theoretic logical entailment, which we also call 0-entailment. It is based on the idea of conditioning.

We define *logical entailment* (or 0-*entailment*) as follows. Given a ground probabilistic logic program P and a ground conditional constraint $(\beta|\alpha)[l,u]$, we say $(\beta|\alpha)[l,u]$ is a 0-*consequence* of P, denoted $P \,\|\!\sim^{0} (\beta|\alpha)[l,u]$, iff every model of P is also a model of $(\beta|\alpha)[l,u]$. We say $(\beta|\alpha)[l,u]$ is a *tight* 0-*consequence* of P, denoted $P \,\|\!\sim^{0}_{tight} (\beta|\alpha)[l,u]$, iff l (resp., u) is the infimum (resp., supremum) of $Pr(\beta|\alpha)$

subject to all models Pr of P with $Pr(\alpha) > 0$. Note that, canonically, $l = 1$ and $u = 0$, when $P \,\|\!\!\sim^0 \bot \Leftarrow \alpha$ (i.e., $Pr(\alpha) = 0$ for all models Pr of P).

Intuitively, we perform a conditioning of every model Pr of P on the premise α, since $P \,\|\!\!\sim^0 (\beta|\alpha)[l,u]$ expresses that $Pr_\alpha(\beta) \in [l,u]$ for all models Pr of P with $Pr(\alpha) > 0$. Moreover, $P \,\|\!\!\sim^0_{tight} (\beta|\alpha)[l,u]$ says that l (resp., u) is the infimum (resp., supremum) of $Pr_\alpha(\beta)$ subject to all models Pr of P with $Pr(\alpha) > 0$. This intuition is more formally expressed by the following lemma.

Lemma 3.1 *Let P be a ground probabilistic logic program, and let $(\beta|\alpha)[l,u]$ be a ground conditional constraint. Then, $P \,\|\!\!\sim^0 (\beta|\alpha)[l,u]$ iff $Pr_\alpha(\beta) \in [l,u]$ for all models Pr of P with $Pr(\alpha) > 0$. Moreover, $P \,\|\!\!\sim^0_{tight} (\beta|\alpha)[l,u]$ iff l (resp., u) is the infimum (resp., supremum) of $Pr_\alpha(\beta)$ subject to all models Pr of P with $Pr(\alpha) > 0$.*

3.2 Entailment under Maximum Entropy

The notion of logical entailment has often been criticized in the literature for its inferential weakness (see also Examples 4.2 and 4.3). One way to strengthen logical entailment is by using the principle of maximum entropy. This is an old idea that is already discussed by Nilsson [56]. Entailment under maximum entropy is based on selecting a single unique model (the one with maximum entropy) rather than considering all models of a probabilistic logic program.

The *maximum entropy model* (or *me-model*) of a satisfiable probabilistic logic program P, denoted $me[P]$, is the unique probabilistic interpretation Pr that is a model of P and that has the greatest entropy among all the models of P, where the *entropy* of a probabilistic interpretation Pr, denoted $H(Pr)$, is defined as follows:

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

Using Lagrange optimization techniques, we obtain the following representation of the me-model of P for all possible worlds $I \in \mathcal{I}_\Phi$:

$$me[P](I) \;=\; \alpha_0 \prod_{\substack{(\psi|\phi)[l,u] \in ground(P) \\ I \models \psi \wedge \phi}} \alpha^+_{\psi|\phi} \prod_{\substack{(\psi|\phi)[l,u] \in ground(P) \\ I \models \neg\psi \wedge \phi}} \alpha^-_{\psi|\phi}\,, \tag{1}$$

where $\alpha^+_{\psi|\phi}$ and $\alpha^-_{\psi|\phi}$ are real numbers associated with each element $(\psi|\phi)[l,u]$ of $ground(P)$ such that P is satisfied and the entropy is maximized [77]. Thus, $me[P]$ is a so-called *c-adaptation*, following the structure imposed by the conditional constraints in $ground(P)$ [29]. In particular, the following holds:

$$\begin{array}{llll} \alpha^+_{\psi|\phi} = 1 & \text{and} & \alpha^-_{\psi|\phi} = 0 & \text{for all} \quad (\psi|\phi)[1,1] \in ground(P)\,, \\ \alpha^+_{\psi|\phi} = 0 & \text{and} & \alpha^-_{\psi|\phi} = 1 & \text{for all} \quad (\psi|\phi)[0,0] \in ground(P)\,. \end{array} \tag{2}$$

We are now ready to define the notion of *entailment under maximum entropy* (or *me-entailment*) as follows. Let P be a ground probabilistic logic program, and let $(\beta|\alpha)[l,u]$ be a ground conditional constraint. We say $(\beta|\alpha)[l,u]$ is an *me-consequence* of P, denoted $P \,\|\!\!\sim^{me} (\beta|\alpha)[l,u]$, iff either (i) P is unsatisfiable, or (ii) $me[P]$ satisfies $(\beta|\alpha)[l,u]$. We say $(\phi|\psi)[l,u]$ is a *tight me-consequence* of P, denoted $P \,\|\!\!\sim^{me}_{tight} (\beta|\alpha)[l,u]$, iff either (i) P is unsatisfiable, $l = 1$, and $u = 0$, or (ii) $me[P](\alpha) = 0$, $l = 1$, and $u = 0$, or (iii) $me[P](\alpha) > 0$ and $me[P](\beta|\alpha) = l = u$.

3.3 Entailment under Maximum Entropy and CWA

We next introduce a new notion of entailment under maximum entropy, which applies to conjunctive probabilistic logic programs, and which adopts the closed world assumption (CWA) from classical logic programming. We will see that it has in particular nicer computational properties than the notion of me-entailment.

The closed world assumption for a conjunctive probabilistic logic program P is defined relative to a classical approximation of P as follows. The *classical approximation* of a conjunctive probabilistic logic program P, denoted $app(P)$, is the set of all $\psi \Leftarrow \phi$ such that (i) $(\psi|\phi)[l,u] \in ground(P)$ for some $l > 0$, and (ii) $P \not\Vdash\!\sim^0 \bot \Leftarrow \phi$. For ground conjunctive events α, the *closed world assumption* for P w.r.t. α, denoted $\mathrm{CWA}(P,\alpha)$, is defined as follows:

$$\mathrm{CWA}(P,\alpha) \;=\; \{\bot \Leftarrow p \mid p \in HB_\Phi,\; app(P) \cup \{\alpha\} \not\models p\}\,.$$

We distinguish between active and inactive ground formulas w.r.t. P and α as follows. A ground atom $p \in HB_\Phi$ is *inactive* w.r.t. P and α iff $\bot \Leftarrow p$ belongs to $\mathrm{CWA}(P,\alpha)$. A ground event γ (resp., ground conditional constraint F) is *inactive* w.r.t. P and α iff at least one ground atom in γ (resp., F) is inactive w.r.t. P and α. A ground atom (resp., ground event, ground conditional constraint) is *active* w.r.t. P and α iff it is not inactive w.r.t. P and α. In the sequel, we often omit P and α when they are clear from the context.

The following theorem shows that conclusions of the form $P \Vdash\!\sim^0 (\beta|\alpha)[l,u]$ are invariant under the closed world assumption for P w.r.t. $\beta \wedge \alpha$.

Theorem 3.2 *Let P be a conjunctive probabilistic logic program, and let $(\beta|\alpha)[l,u]$ be a ground conjunctive conditional constraint. Then,*

$$P \Vdash\!\sim^0 (\beta|\alpha)[l,u] \;\text{ iff }\; P \cup \mathrm{CWA}(P, \beta \wedge \alpha) \Vdash\!\sim^0 (\beta|\alpha)[l,u].$$

Hence, all ground atoms $p \in HB_\Phi$ that are inactive w.r.t. P and $\beta \wedge \alpha$ are actually irrelevant to conclusions of the form $P \Vdash\!\sim^0 (\beta|\alpha)[l,u]$. That is, logical entailment from P coincides with logical entailment from the "active equivalent" to P. This result is more formally expressed by the following theorem.

Theorem 3.3 *Let P be a conjunctive probabilistic logic program, and let $(\beta|\alpha)[l,u]$ be a ground conjunctive conditional constraint. Let $\widehat{P}$ denote the set of (i) all members of $ground(P)$ that are active w.r.t. P and $\beta \wedge \alpha$, and (ii) all $\bot \Leftarrow \phi$ such that (ii.a) ϕ is active w.r.t. P and $\beta \wedge \alpha$, and (ii.b) $(\psi|\phi)[r,s] \in ground(P)$ for some $r > 0$ and some ψ that is inactive w.r.t. P and $\beta \wedge \alpha$. Then,*

$$P \Vdash\!\sim^0 (\beta|\alpha)[l,u] \;\text{ iff }\; \widehat{P} \Vdash\!\sim^0 (\beta|\alpha)[l,u].$$

The notion of me-entailment, however, lacks the properties described in Theorems 3.2 and 3.3, as the following counterexample shows.

Example 3.4 Let the conjunctive probabilistic logic program P be given by:

$$P = \{(c|b)[0.9, 0.9], (b|a)[0.8, 0.8], (c|a)[0.9, 0.9]\}\,.$$

It then holds $P \Vdash\!\sim^{me} (b|c)[0.65, 0.65]$. However, since $\mathrm{CWA}(P, b \wedge c) = \{\bot \Leftarrow a\}$, we obtain $P \cup \mathrm{CWA}(P, b \wedge c) \Vdash\!\sim^{me} (b|c)[0.56, 0.56]$ and also $\widehat{P} \Vdash\!\sim^{me} (b|c)[0.56, 0.56]$, where $\widehat{P} = \{(c|b)[0.9, 0.9]\}$. □

To combine the strength of me-entailment with the straightness of logical entailment, we now define the notion of *entailment under maximum entropy and CWA* (or *mc-entailment*). Intuitively, the ground atoms that are irrelevant to logical conclusions should actually also be irrelevant to maximum entropy conclusions.

More formally, let P be a ground conjunctive probabilistic logic program, and let $(\beta|\alpha)[l,u]$ be a ground conjunctive conditional constraint. We say $(\beta|\alpha)[l,u]$ is an *mc-consequence* of P, denoted $P \Vdash\!\sim^{mc} (\beta|\alpha)[l,u]$, iff $P \cup \mathrm{CWA}(P, \beta \wedge \alpha) \Vdash\!\sim^{me} (\beta|\alpha)[l,u]$. We say $(\beta|\alpha)[l,u]$ is a *tight mc-consequence* of P, denoted $P \Vdash\!\sim^{mc}_{tight} (\beta|\alpha)[l,u]$, iff $P \cup \mathrm{CWA}(P, \beta \wedge \alpha) \Vdash\!\sim^{me}_{tight} (\beta|\alpha)[l,u]$.

The following two results then follow from the definition of mc-entailment. They show that Theorems 3.2 and 3.3 carry over to mc-entailment. The first one shows that also conclusions of the form $P \Vdash\!\sim^{mc} (\beta|\alpha)[l,u]$ are invariant under the closed world assumption for P w.r.t. $\beta \wedge \alpha$.

Theorem 3.5 *Let P be a conjunctive probabilistic logic program, and let $(\beta|\alpha)[l,u]$ be a ground conjunctive conditional constraint. Then,*

$$P \Vdash\!\sim^{mc} (\beta|\alpha)[l,u] \ \textit{ iff } \ P \cup \mathrm{CWA}(P, \beta \wedge \alpha) \Vdash\!\sim^{mc} (\beta|\alpha)[l,u].$$

The next theorem shows that all ground atoms $p \in HB_\Phi$ that are inactive w.r.t. P and $\beta \wedge \alpha$ are also irrelevant to conclusions $P \Vdash\!\sim^{mc} (\beta|\alpha)[l,u]$. It also shows that mc-entailment can be defined as me-entailment from the "active equivalent" to P.

Theorem 3.6 *Let P be a conjunctive probabilistic logic program, and let $(\beta|\alpha)[l,u]$ be a ground conjunctive conditional constraint. Let $\widehat{P}$ denote the set of (i) all members of $ground(P)$ that are active w.r.t. P and $\beta \wedge \alpha$, and (ii) all $\bot \Leftarrow \phi$ such that (ii.a) ϕ is active w.r.t. P and $\beta \wedge \alpha$, and (ii.b) $(\psi|\phi)[r,s] \in ground(P)$ for some $r > 0$ and some ψ that is inactive w.r.t. P and $\beta \wedge \alpha$. Then,*

$$P \Vdash\!\sim^{mc} (\beta|\alpha)[l,u] \ \textit{ iff } \ \widehat{P} \Vdash\!\sim^{mc} (\beta|\alpha)[l,u] \ \textit{ iff } \ \widehat{P} \Vdash\!\sim^{me} (\beta|\alpha)[l,u].$$

4 Examples

In this section, we give some examples that show the relevance of probabilistic logic programs in practice. In the sequel, predicate and constant symbols begin with lower case letters, whereas object and bound variables start with upper case letters. The correct answers and the tight answer substitutions under 0-entailment below are computed with LINOP [68], which is built on top of the public-domain linear optimization software `lp_solve`, while the correct answers and the tight answer substitutions under me- and mc-entailment are computed with SPIRIT [67] and PIT [16, 71]. Our first example concerns the problem of route planning.

Example 4.1 (*Route Planning*) Assume that John wants to pick up Mary after she stopped working. To do so, he must drive from his home to her office. But, 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*) from R to S, the probability that he can reach (*re*) S from R without running into a traffic jam is greater than 0.7. Given a road in the south (*so*) of the town, this probability is even greater than 0.9. A friend just called him and gave him advice (*ad*) about some roads without any significant traffic. Clearly, if he can reach S from T and T from R, both without running into a traffic jam, then he can also reach S from R without running into a traffic jam. Furthermore, John has some concrete knowledge about the roads, the roads in the south of the town, and the roads that his friend was talking about. For example, he knows that there is a road from his home (*h*) to the university (*u*), from the university to the airport (*a*), and from the airport to Mary's office (*o*). Moreover, John believes that his friend was talking about the road from the university to the airport with a probability between 0.8 and 0.9 (he is not completely sure about it anymore, though).

The above and some other probabilistic knowledge is expressed by the following conjunctive probabilistic logic program P:

$$\begin{aligned} P \;=\; \{ & (ro(h,u) \mid \top)[1,1], \\ & (ro(u,a) \mid \top)[1,1], \\ & (ro(a,o) \mid \top)[1,1], \\ & (ad(h,u) \mid \top)[1,1], \\ & (ad(u,a) \mid \top)[0.8,0.9], \\ & (so(a,o) \mid \top)[1,1], \\ & (re(R,S) \mid ro(R,S))[0.7,1], \\ & (re(R,S) \mid ro(R,S) \wedge so(R,S))[0.9,1], \\ & (re(R,S) \mid ro(R,S) \wedge ad(R,S))[1,1], \\ & (re(R,S) \mid re(R,T) \wedge re(T,S))[1,1]\}\,. \end{aligned}$$

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 0.01. Moreover, he is wondering about the probability of reaching the office, without running into a traffic jam. Finally, he is wondering about this probability, given that his friend was talking about the road from the university to the airport. This can be expressed by the following conjunctive probabilistic queries:

$$\begin{aligned} Q_1 &= \exists(re(h,o) \mid \top)[.99,1], \\ Q_2 &= \exists(re(h,o) \mid \top)[X,Y], \\ Q_3 &= \exists(re(h,o) \mid ad(u,a))[X,Y]\,. \end{aligned}$$

The correct answer for Q_1 to P and the tight answer substitutions for Q_2 and Q_3 to P under 0- and mc-entailment are shown in Table 4.1. □

Table 1: Correct answers and tight answer substitutions for Example 4.1.

probabilistic query	0-entailment	mc-entailment
Q_1	No	No
Q_2	$\{X/0.7000, Y/1\}$	$\{X/0.9353, Y/0.9353\}$
Q_3	$\{X/0.8750, Y/1\}$	$\{X/0.9632, Y/0.9632\}$

The following two examples are taken from the area of medical diagnosis.

Example 4.2 (*Diagnosis 1: Appendicitis***)** In a hospital, physicians have to diagnose whether patients with acute abdominal pain are suffering from appendicitis or not. Diagnosing appendicitis is a difficult task, since a lot of different symptoms (as e.g. high temperature, a high rate of leucocytes, vomiting, and various types of pains) can indicate appendicitis, but often only the joint occurrence of several of these symptoms reliably supports the diagnosis (see e.g. [57, 69]).

Here, we only consider four possible symptoms of appendicitis (*app*), namely a high rate of leucocytes (*leuco_high*) and the following three different types of pain: rectal pain (*rec_pain*), pain when released (*pain_rel*), and rebound tenderness (*reb_tender*). Thus, for the sake of intelligibility, our view on this area is a very simplified one.[1] Let our knowledge about the relationships between *app*, *leuco_high* and the three types of pain be expressed by the following conjunctive probabilistic logic program P (see also [73]):

$$\begin{aligned} P \;=\; \{&(\mathit{reb\_tender}(P) \,|\, \mathit{pain\_rel}(P))[0.70, 0.75],\\ &(\mathit{reb\_tender}(P) \,|\, \mathit{leuco\_high}(P))[0.70, 0.75],\\ &(\mathit{app}(P) \,|\, \mathit{rec\_pain}(P) \wedge \mathit{pain\_rel}(P))[0.70, 0.75],\\ &(\mathit{app}(P) \,|\, \mathit{rec\_pain}(P) \wedge \mathit{reb\_tender}(P))[0.65, 0.70],\\ &(\mathit{app}(P) \,|\, \mathit{pain\_rel}(P) \wedge \mathit{reb\_tender}(P) \wedge \mathit{leuco\_high}(P))[0.80, 0.85]\}\,. \end{aligned}$$

Suppose now that Judy is a patient showing the symptoms *leuco_high* and *pain_rel*. Which is the probability that Judy has appendicitis? Which is the probability that she has appendicitis given that she also feels rectal pain? These questions can be expressed by the following two probabilistic queries:

$$\begin{aligned} Q_1 &= \exists(\mathit{app}(\mathit{judy}) \,|\, \mathit{leuco\_high}(\mathit{judy}) \wedge \mathit{pain\_rel}(\mathit{judy}))[X, Y],\\ Q_2 &= \exists(\mathit{app}(\mathit{judy}) \,|\, \mathit{leuco\_high}(\mathit{judy}) \wedge \mathit{pain\_rel}(\mathit{judy}) \wedge \mathit{rec\_pain}(\mathit{judy}))[X, Y]\,. \end{aligned}$$

The tight answer substitutions for Q_1 and Q_2 to P under 0-, me-, and mc-entailment are shown in Table 2. Here, we observe a slight difference between the tight answer substitutions under me- and mc-entailment to Q_1, whereas the tight answer substitutions under me- and mc-entailment to Q_2 are the same. □

Table 2: Tight answer substitutions for Example 4.2.

probabilistic query	0-entailment	me-entailment	mc-entailment
Q_1	$\{X/0, Y/1\}$	$\{X/0.7375, Y/0.7375\}$	$\{X/0.7250, Y/0.7250\}$
Q_2	$\{X/0, Y/1\}$	$\{X/0.7837, Y/0.7837\}$	$\{X/0.7837, Y/0.7837\}$

Example 4.3 (*Diagnosis 2: Cold***)** We now model the dependencies between the disease cold (*cold*) and its symptoms headache (*headache*), cough (*cough*), sore throat (*sore_throat*), and fever (*fever*). Consider the following conjunctive probabilistic logic program P:

$$\begin{aligned} P \;=\; \{&(\mathit{cold}(X) \,|\, \mathit{headache}(X))[0.60, 0.70],\\ &(\mathit{cold}(X) \,|\, \mathit{cough}(X) \wedge \mathit{sore\_throat}(X))[0.90, 0.95],\\ &(\mathit{fever}(X) \,|\, \mathit{cold}(X))[0.60, 0.80]\}\,. \end{aligned}$$

[1]The system LEXMED [16] shows that maximum entropy inference is able to take into account many more symptoms for appendicitis. By accepting arbitrary combinations of them for queries, LEXMED aims at capturing adequately the complex inter-relationships between symptoms and diseases to propose a reliable diagnosis.

Suppose that Peter (*peter*), Paul (*paul*), and Mary (*mary*) are patients, each of them suspecting to have caught a cold. Peter complains about headache and a sore throat, Paul is coughing and has headache, too, and Mary shows all four symptoms. What is the probability of each of them actually suffering from a cold? This can be expressed by the following three conjunctive probabilistic queries:

$$\begin{aligned}
Q_1 &= \exists(\mathit{cold}(\mathit{peter}) \mid \mathit{headache}(\mathit{peter}) \wedge \mathit{sore_throat}(\mathit{peter}))[X, Y], \\
Q_2 &= \exists(\mathit{cold}(\mathit{paul}) \mid \mathit{cough}(\mathit{paul}) \wedge \mathit{headache}(\mathit{paul}))[X, Y], \\
Q_3 &= \exists(\mathit{cold}(\mathit{mary}) \mid \mathit{cough}(\mathit{mary}) \wedge \mathit{headache}(\mathit{mary}) \\
&\qquad\qquad \wedge\, \mathit{sore_throat}(\mathit{mary}) \wedge \mathit{fever}(\mathit{mary}))[X, Y]\,.
\end{aligned}$$

The tight answer substitutions for these queries to P under 0-, me-, and mc-entailment are shown in Table 3. Here, we see a clear difference between tight *m*e- and tight mc-entailment in the answers to the first two queries. Of course, for the third query, we obtain the same tight probability intervals under me- and mc-entailment, showing a clear accumulation of effects. □

Table 3: Tight answer substitutions for Example 4.3.

probabilistic query	0-entailment	me-entailment	mc-entailment
Q_1	$\{X/0, Y/1\}$	$\{X/0.6854, Y/0.6854\}$	$\{X/0.6000, Y/0.6000\}$
Q_2	$\{X/0, Y/1\}$	$\{X/0.6854, Y/0.6854\}$	$\{X/0.6000, Y/0.6000\}$
Q_3	$\{X/0, Y/1\}$	$\{X/0.9201, Y/0.9201\}$	$\{X/0.9201, Y/0.9201\}$

5 Nonmonotonic Properties

Nonmonotonic logics are appreciated for their closeness to commonsense reasoning, but also known as problematic as to concerns complexity and formal logical aspects. In a probabilistic environment, all the more weight must be attached to these problems, due to the richness of syntax and the abundance of models.

Within the last decade, standards have been established to judge the quality of nonmonotonic logics, and benchmark examples pointing out specific problems have been discussed vividly. That work has been done mostly in qualitative and symbolic settings. In this section, we elaborate the nonmonotonic behavior of me- and mc-entailment with respect to probabilistic versions of the above mentioned standards, to show their well-behavedness. In addition, we compare them to classical formalisms, and highlight special features by illustrative examples.

We start with delineating the nonmonotonic aspects of probabilistic reasoning under 0-, me-, and mc-entailment that we explore in this section.

5.1 Nonmonotonicity of Probabilistic Reasoning

Certain conditional constraints $(\psi|\phi)[1,1]$ and $(\psi|\phi)[0,0]$ can be understood classically as "ϕ implies ψ" and "ϕ implies $\neg\psi$", respectively. The notions of 0-, me-, and mc-entailment are all compatible with the monotonic notion of model-theoretic logical entailment in classical propositional logics for such constraints, and satisfy the following property of *inheritance of classical knowledge*:

C-INH If $\mathcal{F} \,|\!\!\sim (\psi|\phi)[c,c]$ and $\phi \Leftarrow \phi^\star$ is valid, then $\mathcal{F} \,|\!\!\sim (\psi|\phi^\star)[c,c]$,

for all ground events ψ, ϕ, and $\phi^\star$, all sets of ground conditional constraints $\mathcal{F}$, and all $c \in \{0,1\}$. That is, classical knowledge is inherited along subclass relationships.

Purely probabilistic conditional constraints $(\psi|\phi)[l,u]$, however, should be interpreted as "the conditional probability of ψ given ϕ lies between l and u". Due to this inherent uncertainty, the notions of 0-, me-, and mc-entailment generally do *not* satisfy the following property of *inheritance of purely probabilistic knowledge*:

P-INH If $\mathcal{F} \,|\!\!\sim (\psi|\phi)[l,u]$ and $\phi \Leftarrow \phi^\star$ is valid, then $\mathcal{F} \,|\!\!\sim (\psi|\phi^\star)[l,u]$.

for all ground events ψ, ϕ, and $\phi^\star$, all sets of ground conditional constraints $\mathcal{F}$, and all $l,u \in [0,1]$ with $l < 1$ and $u > 0$. Even worse, strengthening the antecedent of a purely probabilistic conditional constraint may lead to totally different probability values. This is exactly what makes probabilistic reasoning under 0-, me-, and mc-entailment an excellent candidate for default reasoning with exceptions. Therefore, no complete subclass inheritance of purely probabilistic knowledge, as expressed by **P-INH**, can be expected to hold under 0-, me-, and mc-entailment

The inheritance of purely probabilistic knowledge, however, can nevertheless be a desirable feature of probabilistic entailment, which is well-known from e.g. *reference class reasoning* [64, 34, 35, 61]. In fact, while logical entailment does not have any subclass inheritance of purely probabilistic knowledge, we will show that both notions of *entailment under maximum entropy* realize some limited form of inheritance of purely probabilistic knowledge along subclass relationships. This suggests that an appropriate form of subclass inheritance of purely probabilistic knowledge may be obtained by focusing on some preferred models, or on one "best" model, as is done by me- and mc-entailment. We will first illustrate the nonmonotonic behavior of me- and mc-entailment by studying benchmark examples, and then describe some of their general properties of nonmonotonic inference.

5.2 Benchmark Examples

We start with analyzing in more detail the nonmonotonic behavior of me- and mc-entailment with respect to the missing property **P-INH**. Our first example deals with Tweety, the non-flying penguin, which is worthwhile studying for many reasons. It is not only the most famous example for illustrating nonmonotonic behavior in general, but it also provides a well-understood setting to investigate *subclass inheritance*, the aspect of *irrelevance*, and how *exceptionality* is dealt with.

Example 5.1 (*Tweety***)** The following probabilistic logic program P describes the penguin Tweety with regard to one property that he does not share with other birds (*fly*), and one property that is common to both birds and penguins (*have_legs*):

$$
\begin{aligned}
P \;=\; \{&(\mathit{fly}(T) \mid \mathit{bird}(T))[0.9, 0.98],\\
&(\mathit{bird}(T) \mid \mathit{penguin}(T))[1,1],\\
&(\mathit{fly}(T) \mid \mathit{penguin}(T))[0, 0.05],\\
&(\mathit{have\_legs}(T) \mid \mathit{bird}(T))[0.98, 1]\}\,.
\end{aligned}
$$

So, Tweety (*tweety*) is an exceptional bird with respect to the property of being able to fly (*fly*). But is Tweety also exceptional with respect to having legs (*have_legs*)? Certainly not — it would be unintuitive to believe that Tweety does not have legs for the only reason that Tweety is a non-flying bird. Moreover, what about the ability to fly of the bird Robin (*robin*), who is red (*red*) and thus belongs to a proper subclass of birds? The atom *red* is not mentioned at all in the probabilistic rules above. Thus, it should be considered irrelevant to the conditional event $\mathit{fly}(T)|\mathit{bird}(T)$.

Consider the following probabilistic queries:

$$\begin{aligned} Q_1 &= \exists(\mathit{have\_legs}(\mathit{tweety}) \,|\, \mathit{penguin}(\mathit{tweety}))[X,Y], \\ Q_2 &= \exists(\mathit{have\_legs}(\mathit{robin}) \,|\, \mathit{bird}(\mathit{robin}))[X,Y], \\ Q_3 &= \exists(\mathit{fly}(\mathit{robin}) \,|\, \mathit{bird}(\mathit{robin}) \wedge \mathit{red}(\mathit{robin}))[X,Y], \\ Q_4 &= \exists(\mathit{fly}(\mathit{robin}) \,|\, \mathit{bird}(\mathit{robin}))[X,Y], \\ Q_5 &= \exists(\mathit{fly}(\mathit{tweety}) \,|\, \mathit{penguin}(\mathit{tweety}))[X,Y]. \end{aligned}$$

The tight answer substitutions for these queries are given in Table 4. Under me- and mc-entailment, Tweety's inability to fly has indeed no effect on the probability of him having legs – Tweety is an exceptional bird with respect to flying, but not with respect to being equipped with legs. Moreover, processing the third query Q_3 reveals that, in fact, the property of being red does not influence Robin's ability to fly. So, obviously irrelevant attributes are simply ignored by me-inference. The notion of 0-entailment, however, does not yield these desired results. □

Table 4: Tight answer substitutions for Example 5.1.

probabilistic query	0-entailment	me-entailment	mc-entailment
Q_1	$\{X/0.00,\ Y/1.00\}$	$\{X/0.98,\ Y/0.98\}$	$\{X/0.98,\ Y/0.98\}$
Q_2	$\{X/0.98,\ Y/1.00\}$	$\{X/0.98,\ Y/0.98\}$	$\{X/0.98,\ Y/0.98\}$
Q_3	$\{X/0.00,\ Y/1.00\}$	$\{X/0.90,\ Y/0.90\}$	$\{X/0.90,\ Y/0.90\}$
Q_4	$\{X/0.90,\ Y/0.98\}$	$\{X/0.90,\ Y/0.90\}$	$\{X/0.90,\ Y/0.90\}$
Q_5	$\{X/0.00,\ Y/0.05\}$	$\{X/0.05,\ Y/0.05\}$	$\{X/0.05,\ Y/0.05\}$

Probability theory provides an excellent framework to handle nonmonotonicity: Birds mostly fly, that is, $Pr(\mathit{fly}(T)|\mathit{bird}(T)) \approx 1$, but Tweety, who is a bird and a penguin, does not fly within a probabilistic environment: $Pr(\mathit{fly}(\mathit{tweety})|\mathit{bird}(\mathit{tweety}) \wedge \mathit{penguin}(\mathit{tweety})) = 0$, if only $Pr(\mathit{fly}(T)|\mathit{penguin}(T)) = 0$. The notion of logical entailment, which is based on considering *all* probabilistic models, however, is too weak in general – for instance, it allows of no systematic subclass inheritance, regarding anything as relevant in principle.

In contrast to this, the notions of me- and mc-entailment, both based on considering exactly one distinguished distribution, cope in an elegant way with obviously irrelevant information, as the following general consideration shows: If α is a ground instance of an atom not mentioned in a probabilistic logic program P but occurring in the body of a query $Q = \exists(\psi|\phi \wedge \alpha)[x, y]$, then

$$me[P](\psi|\phi \wedge \alpha) = me[P](\psi|\phi).$$

No external and explicit assumption of conditional independence is necessary here. Rather, the principle of maximum entropy treats information as irrelevant as long as there is no reason to suppose the contrary. This permits a *systematic subclass inheritance*, as in Example 5.1: birds mostly fly, and red birds do so, too.

Moreover, Example 5.1 illustrates that me-inference handles exceptionality with respect to one attribute in an adequate way, without blocking other reasonable conclusions. We showed that, given $P = P_{Tweety}$, the probability of Tweety to have legs is the same as that of any other bird, although Tweety is an exceptional bird. This is not a mere numerical coincidence – using formula (1), it is easy to verify that

$$me[P_{Tweety}](\mathit{have\_legs}(T)|\mathit{penguin}(T)) = me[P_{Tweety}](\mathit{have\_legs}(T)|\mathit{bird}(T))$$

whatever probability is assigned to $\mathit{have\_legs}(T)|\mathit{bird}(T)$.

This well-behavedness is due to an indifference property that the me-model shows for worlds that behave in the same positive, negative, or neutral way w.r.t. the involved conditional events (see Lemma 7.7), which in turn is only a superficial manifestation of a deeper *conditional indifference*: The me-principle represents conditional constraints by balancing conditional effects; for more details, see [29, 31].

In the following example, we illustrate how obviously irrelevant information is dealt with under 0-, me-, and mc-entailment.

Example 5.2 Consider the following conjunctive probabilistic logic program P:

$$\begin{aligned} P \;=\; \{&(\mathit{bird}(T) \mid \mathit{penguin}(T))[1,1], \\ &(\mathit{fly}(T) \mid \mathit{bird}(T))[0.9, 0.95], \\ &(\mathit{fly}(T) \mid \mathit{penguin}(T))[0, 0.05], \\ &(\mathit{easy\_to\_see}(T) \mid \mathit{yellow}(T))[0.95, 1]\}\,. \end{aligned}$$

We are now interested in the probabilities with which Brian, the brightly yellow penguin, is able to fly, and is easy to be seen, respectively:

$$\begin{aligned} Q_1 &= \exists(\mathit{fly}(\mathit{brian}) \mid \mathit{penguin}(\mathit{brian}) \wedge \mathit{yellow}(\mathit{brian}))[X, Y]\,, \\ Q_2 &= \exists(\mathit{easy\_to\_see}(\mathit{brian}) \mid \mathit{penguin}(\mathit{brian}) \wedge \mathit{yellow}(\mathit{brian}))[X, Y]\,. \end{aligned}$$

As expected, the tight answer substitutions for Q_1 and Q_2 under both me- and mc-entailment are given by $\{X/0.05, Y/0.05\}$ and $\{X/0.95, Y/0.95\}$, respectively. So, the maximum entropy methodologies faithfully observe explicit information, while not allowing obviously irrelevant information to have any influence. The tight answer substitutions for Q_1 and Q_2 under 0-entailment, however, are both given by $\{X/0.00, Y/1.00\}$, falling back to complete ignorance in both cases. □

The (extended) Nixon Diamond in the next example deals with conflicting evidence and the relevance of information in a more general sense.

Example 5.3 (*(Extended) Nixon Diamond***)** We consider a probabilistic generalization of the well-known Nixon Diamond "generally, quakers are pacifists" and "generally, republicans are not pacifists" extended by the default rules "generally, quakers are Americans", "generally, Americans like baseball", and "generally, quakers do not like baseball". More precisely, let the following probabilistic logic program P describe the original Nixon Diamond:

$$\begin{aligned} P \;=\; \{&(\mathit{pacifist}(T) \mid \mathit{quaker}(T))[0.8, 0.95], \\ &(\mathit{pacifist}(T) \mid \mathit{republican}(T))[0.05, 0.2]\}\,. \end{aligned}$$

Let the probabilistic logic program P' describe the extended Nixon Diamond:

$$\begin{aligned} P' \;=\; & P \cup \{(\mathit{american}(T) \,|\, \mathit{quaker}(T))[0.9, 0.99], \\ & (\mathit{like\_baseball}(T) \,|\, \mathit{american}(T))[0.85, 0.95], \\ & (\mathit{like\_baseball}(T) \,|\, \mathit{quaker}(T))[0.05, 0.3]\}\,. \end{aligned}$$

Given P and P', what is the probability of Dick, who is known to be a quaker and a republican, to be a pacifist? We get the following probabilistic query Q:

$$Q \;=\; \exists(\mathit{pacifist}(\mathit{dick}) \,|\, \mathit{quaker}(\mathit{dick}) \wedge \mathit{republican}(\mathit{dick}))[X, Y].$$

The tight answer substitutions for Q to P and P' under 0-entailment are both given by $\{X/0, Y/1\}$, while the tight answer substitutions under both me- and mc-entailment are given by $\{X/0.5,\ Y/0.5\}$ and $\{X/0.61,\ Y/0.61\}$, respectively.

That is, in this case, the tight answer substitutions under 0-entailment give a more intuitive, but less informative result, reflecting complete ignorance. The tight answer substitutions under me- and mc-entailment show that, similar to treating this problem via the rational closure approach [38] or by System Z [20] (cf. [10, p. 26]), the (numerical) answer to the query depends on which of the two programs P and P' is used. While the original Nixon diamond does not allow us to draw any conclusion about whether Dick is a pacifist or not, the extended Nixon diamond is now biased in favor of Dick being a pacifist under the maximum entropy approach. This appears a bit strange, as the new information about Americans, quakers, and baseball seems prima facie irrelevant to the relationships between quakers, pacifists, and republicans. The observed shift in probability, however, may be explained by the increased interactions between conditional constraints in P', which maximum entropy methodologies carefully follow — adding constraints may change the point of view. This, however, does not imply that being American or liking baseball is definitely relevant for Dick's loving peace: Adding $\mathit{american}(\mathit{dick}) \wedge \mathit{like\_baseball}(\mathit{dick})$ to the antecedent of the conditional event above will not alter the maximum entropy probability:

$$\begin{aligned} & me[P'](\mathit{pacifist}(\mathit{dick}) \,|\, \mathit{quaker}(\mathit{dick}) \wedge \mathit{republican}(\mathit{dick}) \\ & \qquad \wedge\, \mathit{american}(\mathit{dick}) \wedge \mathit{like\_baseball}(\mathit{dick})) \;=\; 0.61\,. \;\Box \end{aligned}$$

Once again, we illustrate how the maximum entropy approaches combine conflicting information in another penguin-example.

Example 5.4 The following probabilistic logic program P expresses knowledge about the flying capabilities of penguins, birds, and objects with metal wings:

$$\begin{aligned} P \;=\; & \{(\mathit{bird}(T) \,|\, \mathit{penguin}(T))[1, 1], \\ & (\mathit{fly}(T) \,|\, \mathit{metal\_wings}(T))[0.95, 1], \\ & (\mathit{fly}(T) \,|\, \mathit{bird}(T))[0.95, 1], \\ & (\mathit{fly}(T) \,|\, \mathit{penguin}(T))[0, 0.05]\}\,. \end{aligned}$$

What is the probability to fly of Supertweety, a penguin with metal wings,

$$\exists(\mathit{fly}(\mathit{supertweety}) \,|\, \mathit{penguin}(\mathit{supertweety}) \wedge \mathit{metal\_wings}(\mathit{supertweety}))[X, Y]\ ?$$

As tight answer substitutions for this query under both me- and mc-entailment, we obtain $\{X/0.2127,$ $Y/0.2127\}$. Although the metal wings that Supertweety has attached to his sides increase our confidence in him being able to fly, they are not convincing enough to make us believe that he can really fly. Note that the tight answer substitution under logical entailment is given by $\{X/0, Y/1\}$. $\Box$

The last example, which is taken from [2], shows how non-conflicting information is combined by the maximum entropy approaches.

Example 5.5 Let the probabilistic logic program P be given as follows:

$$\begin{aligned} P \;=\; \{ & (\mathit{bird}(T) \mid \mathit{magpie}(T))[1, 1], \\ & (\mathit{chirp}(T) \mid \mathit{bird}(T))[0.7, 0.8], \\ & (\mathit{chirp}(T) \mid \mathit{magpie}(T))[0, 0.99]\}\,. \end{aligned}$$

Consider the following probabilistic query Q:

$$Q \;=\; (\mathit{chirp}(\mathit{polly}) \mid \mathit{magpie}(\mathit{polly}))[X, Y]\,.$$

Knowing that magpies are birds, the probability to chirp of the magpie Polly is already expected to be in $[0, 0.99]$. So, the last conditional constraint of P turns out to be vacuous, and by maximizing indeterminateness, the tight answer substitutions under me- and mc-entailment are both given by $\{X/0.7, Y/0.7\}$, while the tight answer substitution under logical entailment is given by $\{X/0, Y/0.99\}$. $\Box$

5.3 General Properties

As the examples from the preceding section show, the notions of me- and mc-entailment raise the inferential power of probabilistic reasoning substantially. Instead of considering all possible models, the maximum entropy method bases its inferences on one particularly distinguished model — that with maximum entropy. However, one may wonder if this view is not too narrow. One way to explain the reasonableness of maximum entropy inference is to investigate its formal properties according to widely accepted standards for nonmonotonic inference relations, such as System P [33] and related postulates. A special focus will be on irrelevance properties; here we propose a new postulate called *strong irrelevance* aiming at preventing interactions of conditional constraints.

In the following, we assume that $\phi, \psi, \varepsilon, \varepsilon', \alpha, \beta$ are ground events, and that P is a (fixed) ground probabilistic logic program. In case of mc-entailment, we additionally assume that $\phi, \psi, \varepsilon, \varepsilon', \alpha, \beta$, and P are all conjunctive. Let $l, l', u, u', r, s \in [0,1]$.

Moreover, for the sake of representational clarity, we presuppose that P satisfies the following *explicitness condition*, which ensures that every classical relationship that is logically entailed by P is already explicitly stated in P.

Explicitness condition: For every $I \in \mathcal{I}_\Phi$, if $Pr(I) = 0$ for all models Pr of P, then there exists some $(\psi|\phi)[c, c] \in P$ such that either

$$c = 1 \text{ and } I \models \phi \wedge \neg\psi \quad \text{or} \quad c = 0 \text{ and } I \models \phi \wedge \psi\,. \tag{3}$$

We implicitly assume that all notions of entailment are naturally extended to negations of conditional constraints of the form $\neg(\beta|\alpha)[r, s]$, which are true in a probabilistic interpretation Pr iff $Pr(\alpha) > 0$ and $Pr(\beta|\alpha) \notin [r, s]$.

We first consider the postulates *Right Weakening (RW)*, *Reflexivity (Ref)*, *Left Logical Equivalence (LLE)*, *Cut*, *Cautious Monotonicity (CM)*, and *Or* proposed by Kraus, Lehmann, and Magidor [33], which are commonly regarded as being particularly desirable for any reasonable notion of nonmonotonic entailment $\Vert\!\sim$, and which are usually referred to as *System P*. We consider the following generalization for a probabilistic setting:

RW. If $(\phi|\top)[l,u] \Rightarrow (\psi|\top)[l',u']$ is logically valid and $P \mathrel{\|\!\sim} (\phi|\varepsilon)[l,u]$, then $P \mathrel{\|\!\sim} (\psi|\varepsilon)[l',u']$.

Ref. $P \mathrel{\|\!\sim} (\varepsilon|\varepsilon)[1,1]$.

LLE. If $\varepsilon \Leftrightarrow \varepsilon'$ is logically valid, then $P \mathrel{\|\!\sim} (\phi|\varepsilon)[l,u]$ iff $P \mathrel{\|\!\sim} (\phi|\varepsilon')[l,u]$.

Cut. If $P \mathrel{\|\!\sim} (\varepsilon'|\varepsilon)[1,1]$ and $P \mathrel{\|\!\sim} (\phi|\varepsilon \wedge \varepsilon')[l,u]$, then $P \mathrel{\|\!\sim} (\phi|\varepsilon)[l,u]$.

CM. If $P \mathrel{\|\!\sim} (\varepsilon'|\varepsilon)[1,1]$ and $P \mathrel{\|\!\sim} (\phi|\varepsilon)[l,u]$, then $P \mathrel{\|\!\sim} (\phi|\varepsilon \wedge \varepsilon')[l,u]$.

Or. If $P \mathrel{\|\!\sim} (\phi|\varepsilon)[1,1]$ and $P \mathrel{\|\!\sim} (\phi|\varepsilon')[1,1]$, then $P \mathrel{\|\!\sim} (\phi|\varepsilon \vee \varepsilon')[1,1]$.

The following theorem shows that most of these postulates are indeed satisfied by all described notions of entailment for conditional constraints (note that *Or* does not apply to mc-entailment).

Theorem 5.6 *$\mathrel{\|\!\sim}^0$ and $\mathrel{\|\!\sim}^{me}$ satisfy RW, Ref, LLE, Cut, CM, and Or for all probabilistic logic programs P, all ground events ε, ε', ϕ, ψ, and all $l,l',u,u' \in [0,1]$. Moreover, $\mathrel{\|\!\sim}^{mc}$ satisfies Ref, LLE, Cut, and CM for all conjunctive probabilistic logic programs P, all ground conjunctive events ε, ε', ϕ, ψ, and all $l,l',u,u' \in [0,1]$.*

It is unclear whether $\mathrel{\|\!\sim}^{mc}$ satisfies RW or not. The point is that taking new conditional constraints into account may change the resulting maximum entropy distribution so that logical dependencies may get lost. Nevertheless, there are information-theoretical, informal arguments that suggest that RW holds for $\mathrel{\|\!\sim}^{mc}$. However, neither a formal proof nor a counterexample for RW have been found so far.

Another desirable property is *Rational Monotonicity (RM)* [33]:

RM. If $P \mathrel{\|\!\sim} (\psi|\varepsilon)[l,u]$ and $P \mathrel{\|\!\not\sim} \neg(\varepsilon'|\varepsilon)[1,1]$, then $P \mathrel{\|\!\sim} (\psi|\varepsilon \wedge \varepsilon')[l,u]$.

Informally, *RM* describes a restricted form of monotony, and thus allows to ignore certain kinds of irrelevant knowledge. The following theorem shows that both me- and mc-entailment also satisfy *RM*.

Theorem 5.7 *$\mathrel{\|\!\sim}^{me}$ (resp., $\mathrel{\|\!\sim}^{mc}$) satisfies RM for all probabilistic (resp., conjunctive probabilistic) logic programs P, all ground (resp., ground conjunctive) events ε, ε', and ψ, and all $l,u \in [0,1]$.*

We next consider the property *Irrelevance (Irr)*, which is adapted from [3]:

Irr. If $P \mathrel{\|\!\sim} (\phi|\varepsilon)[l,u]$, and no atom of $ground(P)$ and $\phi|\varepsilon$ occurs in ε', then $P \mathrel{\|\!\sim} (\phi|\varepsilon \wedge \varepsilon')[l,u]$.

Informally, *Irr* says that ε' is irrelevant to a conclusion "$P \mathrel{\|\!\sim} (\psi|\varepsilon)[l,u]$" when they are defined over disjoint sets of atoms. The following result shows that both me- and mc-entailment satisfy *Irr*.

Theorem 5.8 *$\mathrel{\|\!\sim}^{me}$ (resp., $\mathrel{\|\!\sim}^{mc}$) satisfies Irr for all probabilistic (resp., conjunctive probabilistic) logic programs P, all ground (resp., ground conjunctive) events ε, ε', and ϕ, and all $l,u \in [0,1]$.*

The property *Irr* formalizes a very basic form of irrelevance to be expected from inference relations. We now propose a stronger irrelevance property, called *strong irrelevance (SI)*:

SI. If $P \mathrel{\|\!\sim} (\psi|\phi)[l,u]$ and no atom of $ground(P)$ and $\psi|\phi$ occurs in $\beta|\alpha$, then $P \cup \{(\beta|\alpha)[r,s]\} \mathrel{\|\!\sim} (\psi|\phi)[l,u]$ and $P \cup \{(\beta|\alpha)[r,s]\} \mathrel{\|\!\sim} (\psi|\phi \wedge \alpha)[l,u]$.

Strong irrelevance says that adding conditional constraints on newly occurring atoms, on the one hand, does not change previously made inferences, and, on the other hand, does not cause interferences with the other conditional constraints. The following theorem states that both me- and mc-entailment satisfy the property *SI*.

Theorem 5.9 *$\Vdash\!\sim^{me}$ (resp., $\Vdash\!\sim^{mc}$) satisfies SI for all probabilistic (resp., conjunctive probabilistic) logic programs P, all ground (resp., ground conjunctive) events α, β, ϕ, and ψ, and all $l, u, r, s \in [0, 1]$.*

We next consider the property *Direct Inference (DI)*, adapted from [2]:

DI. If $(\psi|\phi)[l, u] \in \mathit{ground}(P)$ and $\varepsilon \Leftrightarrow \phi$ is logically valid, then $P \Vdash\!\sim (\psi|\varepsilon)[l, u]$.

Informally, *DI* expresses that P should entail all its own conditional constraints. Note that the property *DI* is similar to *LLE*. In fact, *DI* is implied by *LLE* together with another nonmonotonic property called *Inclusion (Inc)*:

Inc. If $(\psi|\phi)[l, u] \in \mathit{ground}(P)$, then $P \Vdash\!\sim (\psi|\phi)[l, u]$.

Lemma 5.10 *If $\Vdash\!\sim$ satisfies LLE and Inc, then it also satisfies DI.*

The next result shows that all discussed notions of entailment satisfy *Inc* and *DI*.

Theorem 5.11 *$\Vdash\!\sim^{0}$ and $\Vdash\!\sim^{me}$ satisfy Inc and DI for all probabilistic logic programs P, all ground events ε, ϕ, and ψ, and all $l, u \in [0, 1]$. Moreover, $\Vdash\!\sim^{mc}$ satisfies Inc and DI for all conjunctive probabilistic logic programs P, all ground conjunctive events ε, ϕ, and ψ, and all $l, u \in [0, 1]$.*

5.4 Relationship between Probabilistic Formalisms

We next analyze the relationship between the discussed notions of entailment.

For entailment semantics s_1 and s_2, we say s_1 is *stronger than* s_2 iff $\Vdash\!\sim^{s_1}$ is a superset of $\Vdash\!\sim^{s_2}$. That is, $\mathcal{F} \Vdash\!\sim^{s_2} F$ implies $\mathcal{F} \Vdash\!\sim^{s_1} F$, for all sets of ground conditional constraints $\mathcal{F}$ and all ground conditional constraints F for which both s_1 and s_2 are defined. Equivalently, $\mathcal{F} \Vdash\!\sim^{s_2}_{tight} (\psi|\phi)[l_2, u_2]$ and $\mathcal{F} \Vdash\!\sim^{s_1}_{tight} (\psi|\phi)[l_1, u_1]$ implies $[l_2, u_2] \supseteq [l_1, u_1]$, for all sets of ground conditional constraints $\mathcal{F}$ and all ground conditional constraints $(\psi|\phi)[l_2, u_2]$ and $(\psi|\phi)[l_1, u_1]$ for which both s_1 and s_2 are defined. Note that this "stronger than" relation is reflexive and transitive.

The following theorem shows that the arrows in Fig. 1 actually represent "stronger than" relationships between the entailment semantics shown in Fig. 1. In fact, from Example 3.4, or 4.2, or 4.3 it can be seen that Fig. 1 draws a complete picture of the "stronger than" relationships between the shown entailment semantics.

Theorem 5.12 *Both me- and mc-entailment are stronger than logical entailment.*

The next result shows that entailment under maximum entropy and logical entailment coincide on the concluded ground classical conditional constraints.

Theorem 5.13 *Let P be a probabilistic logic program, and let $(\psi|\phi)[c, c]$ with $c \in \{0, 1\}$ be a ground classical conditional constraint. Let P and $(\psi|\phi)[c, c]$ be conjunctive for $s = mc$. Then, for every $s \in \{me, mc\}$:*

(a) $P \Vdash\!\sim^{s} (\psi|\phi)[c, c]$ *iff* $P \Vdash\!\sim^{0} (\psi|\phi)[c, c]$.

(b) $P \Vdash\!\sim^{s}_{tight} (\psi|\phi)[c, c]$ *iff* $P \Vdash\!\sim^{0}_{tight} (\psi|\phi)[c, c]$.

5.5 Relationship to Classical Formalisms

We now analyze the relationship to classical formalisms.

For classical conditional constraints F of the form $(\beta|\alpha)[0,0]$ (resp., $(\beta|\alpha)[1,1]$), we use $\gamma(F)$ to denote the events $\bot \Leftarrow \beta \wedge \alpha$ (resp., $\beta \Leftarrow \alpha$). For probabilistic logic programs P, we define $\gamma(P)$ as the set of all events $\gamma(F)$ with $F \in P$.

The following theorem shows that all three described notions of probabilistic entailment generalize logical entailment with events.

Theorem 5.14 *Let $s \in \{0, me, mc\}$. Let P be a probabilistic logic program, and let $F = (\psi|\phi)[c,c]$ with $c \in \{0,1\}$ be a ground classical conditional constraint. Let P and F be conjunctive for $s = mc$. Then,*

(a) $P \,\|\!\!\sim^{s} F$ iff $\gamma(P) \models \gamma(F)$.

(b) $P \,\|\!\!\sim^{s}_{tight} F$ iff $\gamma(P) \models \gamma(F)$ and $\gamma(P) \not\models \neg\phi$.

6 Naive Characterizations

In this section, we characterize the solutions of the problems POSITIVE PROBABILITY and TIGHT s-CONSEQUENCE, where $s \in \{0, me, mc\}$, by straightforward decision and optimization problems involving a system of linear constraints. We first describe how the models of a probabilistic logic program correspond to the solutions of a system of linear constraints. We then describe how POSITIVE PROBABILITY can be reduced to the solvability of a system of linear constraints, and how TIGHT 0-CONSEQUENCE can be reduced to POSITIVE PROBABILITY and two linear optimization problems. We next show how TIGHT me-CONSEQUENCE can be reduced to POSITIVE PROBABILITY and an entropy maximization problem. All the above results are well-known from the literature. As a new result of this paper, we finally describe how TIGHT mc-CONSEQUENCE can be reduced to POSITIVE PROBABILITY, computing the least Herbrand model of a classical definite logic program, and TIGHT me-CONSEQUENCE.

6.1 Preliminaries

The following theorem shows that the models of a probabilistic logic program P correspond to the solutions of the system of linear constraints $LC(\delta, \mathcal{F}, R)$ shown in Fig. 2, where the parameters δ, $\mathcal{F}$, and R denote a conditioning event, a finite set of ground conditional constraints, and an index set for the variables.

Theorem 6.1 *Let P be a probabilistic logic program. Let the parameters δ, $\mathcal{F}$, and R be given by $\delta = \top$, $\mathcal{F} = ground(P)$, and $R = \mathcal{I}_\Phi$, respectively. Then:*

(a) For every model Pr of P, there is a solution $(y_r)_{r \in R}$ of the system of linear constraints $LC(\delta, \mathcal{F}, R)$ such that $Pr(r) = y_r$ for all $r \in R$.

(b) For every solution $(y_r)_{r \in R}$ of the system of linear constraints $LC(\delta, \mathcal{F}, R)$, there exists a model Pr of P such that $y_r = Pr(r)$ for all $r \in R$.

The crux with this *naive* system of linear constraints is that the number of variables is given by the number of possible worlds over Φ and that the number of linear constraints is linear in the number of ground instances of conditional constraints in P. Hence, especially the number of variables is generally quite large, as the following example immediately shows.

$$
\begin{array}{rcll}
\displaystyle\sum_{r\in R,\, r\models\delta} y_r & = & 1 & \\
\displaystyle\sum_{r\in R,\, r\models\neg\psi\wedge\phi} -l\, y_r + \sum_{r\in R,\, r\models\psi\wedge\phi} (1-l)\, y_r & \geq & 0 & (\text{for all } (\psi|\phi)[l,u]\in\mathcal{F},\ l>0) \\
\displaystyle\sum_{r\in R,\, r\models\neg\psi\wedge\phi} u\, y_r + \sum_{r\in R,\, r\models\psi\wedge\phi} (u-1)\, y_r & \geq & 0 & (\text{for all } (\psi|\phi)[l,u]\in\mathcal{F},\ u<1) \\
y_r & \geq & 0 & (\text{for all } r\in R)
\end{array}
$$

Figure 2: System of linear constraints $LC(\delta,\mathcal{F},R)$ for Theorems 6.1 and 6.3–6.5.

Example 6.2 Consider the probabilistic logic program P of Example 4.1. The naive system of linear constraints of Theorem 6.1 has the quite large number of $2^{64}\approx 18\cdot 10^{18}$ (!) variables and 120 linear constraints. □

6.2 Positive Probability

The next theorem shows that the decision problem POSITIVE PROBABILITY can be reduced to the problem of deciding if the system of linear constraints $LC(\delta,\mathcal{F},R)$ in Fig. 2 is solvable. This result follows from Theorem 6.1.

Theorem 6.3 *Let P be a probabilistic logic program, and let α be a ground event. Then, P has a model Pr with $Pr(\alpha)>0$ iff the system of linear constraints $LC(\delta,\mathcal{F},R)$ is solvable, where $\delta=\alpha$, $\mathcal{F}=ground(P)$, and $R=\mathcal{I}_\Phi$.*

6.3 Tight Logical Consequence

Consider now an object-ground probabilistic query $Q=\exists(\beta|\alpha)[x,y]$ to a probabilistic logic program P. To compute the tight answer substitution σ for Q to P under logical entailment, we first decide whether $P \,\|\!\!\sim^0 \bot \Leftarrow \alpha$, which is the complement of an instance of POSITIVE PROBABILITY. If $P \,\|\!\!\sim^0 \bot \Leftarrow \alpha$, then σ is immediately given by $\{x/1,\ y/0\}$. Otherwise, we additionally solve two optimization problems in which a linear fractional objective function must be minimized and maximized subject to the system of linear constraints in Fig. 2 as follows.

Theorem 6.4 *Let P be a probabilistic logic program, and let $Q=\exists(\beta|\alpha)[x,y]$ be an object-ground probabilistic query with $P \,\|\!\!\not\sim^0 \bot \Leftarrow \alpha$. Then, the tight answer substitution for Q to P under logical entailment is given by $\sigma=\{x/l,\ y/u\}$, where l (resp., u) is the optimal value of the following linear fractional program over the variables y_r ($r\in R$), where $\delta=\top$, $\mathcal{F}=ground(P)$, and $R=\mathcal{I}_\Phi$:*

$$
\begin{array}{l}
\textit{minimize (resp., maximize)} \;\; \Big(\displaystyle\sum_{r\in R,\, r\,\models\,\beta\wedge\alpha} y_r\Big) \Big/ \Big(\sum_{r\in R,\, r\,\models\,\alpha} y_r\Big) \\
\textit{subject to } LC(\delta,\mathcal{F},R) \textit{ and } \displaystyle\sum_{r\in R,\, r\,\models\,\alpha} y_r \;>\; 0\,.
\end{array}
\qquad (4)
$$

By a standard technique going back to Charnes and Cooper [5], the linear fractional programs (4) can be transformed into two equivalent linear programs as follows.

Theorem 6.5 *Let P be a probabilistic logic program, and let $Q = \exists(\beta|\alpha)[x, y]$ be an object-ground probabilistic query with $P \not\Vdash\!\sim^0 \bot \Leftarrow \alpha$. Then, the tight answer substitution for Q to P under logical entailment is given by $\sigma = \{x/l,\ y/u\}$, where l (resp., u) is the optimal value of the following linear program over the variables y_r $(r \in R)$, where $\delta = \alpha$, $\mathcal{F} = ground(P)$, and $R = \mathcal{I}_\Phi$:*

$$\text{minimize (resp., maximize)} \sum_{r \in R,\, r \models \beta \wedge \alpha} y_r \quad \text{subject to } LC(\delta, \mathcal{F}, R). \tag{5}$$

6.4 Tight Consequence under Maximum Entropy

Consider again an object-ground probabilistic query $Q = \exists(\beta|\alpha)[x, y]$ to a probabilistic logic program P. To compute the tight answer substitution σ for Q to P under me-entailment, we first decide whether $P \Vdash\!\sim^{me} \bot \Leftarrow \alpha$. By the following well-known proposition, we can equivalently decide whether $P \Vdash\!\sim^0 \bot \Leftarrow \alpha$, which is the complement of an instance of POSITIVE PROBABILITY.

Proposition 6.6 *Let P be a probabilistic logic program, and let α be a ground event. Then, $P \Vdash\!\sim^{me} \bot \Leftarrow \alpha$ iff $P \Vdash\!\sim^0 \bot \Leftarrow \alpha$.*

If indeed $P \Vdash\!\sim^{me} \bot \Leftarrow \alpha$, then σ is immediately given by $\{x/1, y/0\}$. Otherwise, σ is given by $\{x/d, y/d\}$, where $d = me[P](\beta|\alpha) = me[P](\beta \wedge \alpha) \,/\, me[P](\alpha)$. Thus, it remains to compute the values of $\beta \wedge \alpha$ and α under the me-model of P.

More generally, the me-model of P can be computed in a straightforward way by solving the following entropy maximization problem over the variables y_r $(r \in R)$, where $\delta = \top$, $\mathcal{F} = ground(P)$, and $R = \mathcal{I}_\Phi$:

$$\max \ -\sum_{r \in R} y_r \log y_r \ \text{ subject to } LC(\delta, \mathcal{F}, R). \tag{6}$$

6.5 Tight Consequence under Maximum Entropy and CWA

To compute the tight answer substitution σ under mc-entailment for an object-ground probabilistic query $Q = \exists(\beta|\alpha)[x, y]$ to a probabilistic logic program P, we first compute the logical approximation of P, which can be done by solving some instances of POSITIVE PROBABILITY. Using classical definite logical programming techniques, we then compute the set of all active ground atoms w.r.t. P and $\beta \wedge \alpha$, which is the least Herbrand model of the logical approximation of P. We next compute the set $\widehat{P}$ of (i) all active members of $ground(P)$ w.r.t. P and $\beta \wedge \alpha$, and (ii) all active $\bot \Leftarrow \phi$ w.r.t. P and $\beta \wedge \alpha$ such that $(\psi|\phi)[r, s] \in ground(P)$ for some $r > 0$ and some inactive ψ w.r.t. P and $\beta \wedge \alpha$. By Theorem 3.6, σ is then given by the tight answer substitution for Q to $\widehat{P}$ under me-entailment.

7 Exploiting Classical Knowledge and Clustering Possible Worlds

In this section, we give some more sophisticated characterizations of the solutions of POSITIVE PROBABILITY and TIGHT s-CONSEQUENCE, where $s \in \{0, me, mc\}$, by decision and optimization problems involving a system of linear constraints. We show how classical knowledge can be exploited and how variables can be clustered into equivalence classes in order to obtain a system of linear constraints that has fewer variables and fewer linear constraints than the one in Section 6. Note that the characterizations for TIGHT me-CONSEQUENCE, and TIGHT mc-CONSEQUENCE are new results, while the characterizations for POSITIVE PROBABILITY and TIGHT 0-CONSEQUENCE are essentially taken from [45].

7.1 Preliminaries

In the sequel, let P be a probabilistic logic program. We write $P = (C, D)$ to denote that C (resp., D) is the set of all classical (resp., purely probabilistic) members of P (resp., $ground(P)$). For sets of conditional constraints $\mathcal{F}$, we use $[\![\mathcal{F}]\!]$ to denote the set of all conditional events $\psi|\phi$ such that $(\psi|\phi)[l, u] \in \mathcal{F}$ for some $l, u \in [0, 1]$.

Roughly speaking, the main ideas behind exploiting classical knowledge and clustering possible worlds can be summarized as follows:

- We introduce a variable only for every $I \in \mathcal{I}_\Phi$ that satisfies C, rather than for every $I \in \mathcal{I}_\Phi$. Moreover, we introduce up to two linear inequalities only for each member of D, rather than for each member of $ground(P)$.
- We exploit the structure of the conditional events in D, which imposes an equivalence relation on the set of all models $I \in \mathcal{I}_\Phi$ of C. We then introduce a variable only for each equivalence class of possible worlds.

We first define a characterization of the above set of equivalence classes of possible worlds. Given a set of classical conditional constraints C and a set of ground conditional events $E = \{\varepsilon_1, \ldots, \varepsilon_n\}$, denote by $R_C(E)$ the set of all mappings r that assign every $\varepsilon_i = \psi_i|\phi_i \in E \cup \{\top|\top\}$ an element of $\{\psi_i \wedge \phi_i, \neg\psi_i \wedge \phi_i, \neg\phi_i\}$ such that $C \cup \{r(\varepsilon_i) \mid \varepsilon_i \in E\}$ is satisfiable. We use $\wedge\, r$ to denote $r(\varepsilon_1) \wedge \cdots \wedge r(\varepsilon_n)$. For such mappings r and ground events ϕ, we use $r \models \phi$ to abbreviate $\wedge\, r \models \phi$.

The following lemma formulates the immediate result that $R_C(E)$ defines a partition $S_C(E)$ of the set of all possible worlds $I \in \mathcal{I}_\Phi$ that satisfy C. That is, $R_C(E)$ defines an equivalence relation on the set of all models $I \in \mathcal{I}_\Phi$ of C.

Lemma 7.1 *Let C be a set of classical conditional constraints, and let E be a finite set of ground conditional events. Then, the family $S_C(E) = \{S_r \mid r \in R_C(E)\}$, where $S_r = \{I \in \mathcal{I}_\Phi \mid I \models C \cup \{\wedge\, r\}\}$, is a partition of $\{I \in \mathcal{I}_\Phi \mid I \models C\}$.*

We next show that the set of conditional events E can be interpreted by a probability function over $S_C(E)$. That is, as far as E is concerned, we do not need the fine granulation of $\{I \in \mathcal{I}_\Phi \mid I \models C\}$. To formulate this result, we introduce the notion of *measurability* in $R_C(E)$. We say that a ground event ϕ is *measurable* in $R_C(E)$ iff for all $r \in R_C(E)$, it holds that $I \models \phi$ for *some* $I \in S_r$ iff $I \models \phi$ for *all* $I \in S_r$. A ground conditional event $\psi|\phi$ is *measurable* in $R_C(E)$ iff ϕ and $\psi \wedge \phi$ are measurable in $R_C(E)$. Intuitively, ϕ (resp., $\psi|\phi$) is measurable in $R_C(E)$ iff it can be interpreted by a probability function over $S_C(E)$. In particular, $\top$ and all $\psi|\phi \in E$ are measurable in $R_C(E)$, as the following immediate lemma shows, which also implies that all $\neg\phi$ and $\neg\psi \wedge \phi$ with $\psi|\phi \in E$ are measurable in $R_C(E)$.

Lemma 7.2 *Let C be a set of classical conditional constraints and E be a finite set of ground conditional events. Then, $\top$ and all $\psi|\phi \in E$ are measurable in $R_C(E)$.*

The next result shows that the models of a probabilistic logic program $P = (C, D)$ correspond to the solutions of a system of linear constraints in which we have one variable for each $r \in R_C([\![D]\!])$ and up to two linear constraints for each $F \in D$. This result is immediate by Theorem 6.1 and Lemmas 7.1 and 7.2.

Theorem 7.3 *Let $P = (C, D)$ be a probabilistic logic program. Let the parameters δ, $\mathcal{F}$, and R be given by $\delta = \top$, $\mathcal{F} = D$, and $R = R_C([\![D]\!])$. Then:*

(a) *For every model Pr of P, there is a solution $(y_r)_{r \in R}$ of the system of linear constraints $LC(\delta, \mathcal{F}, R)$ such that $Pr(r) = y_r$ for all $r \in R$.*

(b) *For every solution $(y_r)_{r \in R}$ of the system of linear constraints $LC(\delta, \mathcal{F}, R)$, there exists a model Pr of P such that $y_r = Pr(r)$ for all $r \in R$.*

The index set $R_C([\![D]\!])$ of the new system of linear constraints can be computed using Algorithm index_set_2 from [45]. The following proposition shows that for ground conjunctive probabilistic logic programs P, the set $R_C([\![D]\!])$ can be computed in time $O(|D|\,\|P\|\,|R_C([\![D]\!])|)$, where $|S|$ denotes the cardinality of a set S, and $\|P\|$ denotes the input size of P. This shows that the index set $R_C([\![D]\!])$ can be computed in *output-polynomial total time* (see especially [13]).

Proposition 7.4 *Given a ground conjunctive probabilistic logic program $P = (C, D)$, computing $R_C([\![D]\!])$ can be done in time $O(|D|\,\|P\|\,|R_C([\![D]\!])|)$.*

7.2 Positive Probability

The following theorem shows that POSITIVE PROBABILITY can be reduced to the solvability of a system of linear constraints similar to the one in Theorem 7.3. This result follows immediately from Theorem 6.3 and Lemmas 7.1 and 7.2.

Theorem 7.5 *Let $P = (C, D)$ be a probabilistic logic program, and let α be a ground event. Then, P has a model Pr with $Pr(\alpha) > 0$ iff the system of linear constraints $LC(\delta, \mathcal{F}, R)$ is solvable, where $\delta = \alpha$, $\mathcal{F} = D$, and $R = R_C([\![D]\!] \cup \{\alpha|\top\})$.*

7.3 Tight Logical Consequence

Let $P = (C, D)$ be a probabilistic logic program, and let $Q = \exists(\beta|\alpha)[x, y]$ be an object-ground probabilistic query where $P \not\Vdash^0 \bot \Leftarrow \alpha$. The next theorem shows that also the tight answer substitution for Q to P under logical entailment can be computed by solving two linear programs with a system of linear constraints as in Theorem 7.3. This result is immediate by Theorem 6.5 and Lemmas 7.1 and 7.2.

Theorem 7.6 *Let $P = (C, D)$ be a probabilistic logic program, and let $Q = \exists(\beta|\alpha)\ [x, y]$ be an object-ground probabilistic query such that $P \not\Vdash^0 \bot \Leftarrow \alpha$. Then, the tight answer substitution for Q to P under logical entailment is given by $\sigma = \{x/l,\, y/u\}$, where l (resp., u) is the optimal value of the following linear program over the variables y_r ($r \in R$), where $\delta = \alpha$, $\mathcal{F} = D$, and $R = R_C([\![D]\!] \cup \{\beta|\alpha\})$:*

$$\textit{minimize (resp., maximize)} \sum_{r \in R,\, r \models \beta \wedge \alpha} y_r \quad \textit{subject to } LC(\delta, \mathcal{F}, R). \tag{7}$$

7.4 Tight Consequence under Maximum Entropy

Consider again a probabilistic logic program $P = (C, D)$ and an object-ground probabilistic query $Q = \exists(\beta|\alpha)[x, y]$ where $P \not\Vdash^0 \bot \Leftarrow \alpha$. We now show that exploiting classical knowledge and clustering possible worlds can also be done when computing the tight answer substitution for Q to P under me-entailment. The following lemma states the auxiliary result that all possible worlds in the same equivalence class have the same probability under the me-model of P.

Lemma 7.7 *Let $P = (C, D)$ be a probabilistic logic program. Let $R = R_C([\![D]\!] \cup \{\beta|\alpha\})$. Then, for all $I_1, I_2 \in \mathcal{I}_\Phi$ such that $I_1, I_2 \models C \cup \{\wedge\, r\}$ for some $r \in R$, it holds that $me[P](I_1) = me[P](I_2)$.*

The following theorem shows how the tight answer substitution for Q to P under me-entailment can be characterized through the optimal solution of an optimization problem that has a system of linear constraints similar to the one in Theorem 7.3. This result can be proved using Lemmas 7.1, 7.2, and 7.7. Note that the objective function involves weights a_r $(r \in R)$, where each a_r is given by the number of all possible worlds $I \in \mathcal{I}_\Phi$ that belong to the equivalence class S_r.

Theorem 7.8 *Let $P = (C, D)$ be a probabilistic logic program, and let $Q = \exists(\beta|\alpha)\ [x, y]$ be an object-ground probabilistic query with $P \not\Vdash^0 \bot \Leftarrow \alpha$. Let $\delta = \top$, $\mathcal{F} = D$, and $R = R_C([\![D]\!] \cup \{\beta|\alpha\})$. For $r \in R$, let $a_r = |\{I \in \mathcal{I}_\Phi \mid I \models C \cup \{\wedge\, r\}\}|$. Then, the tight answer substitution for Q to P under me-entailment is given by $\sigma = \{x/d, y/d\}$, where $d = (\sum_{r \in R,\, r \models \beta \wedge \alpha} y^\star_r) \,/\, (\sum_{r \in R,\, r \models \alpha} y^\star_r)$ and $y^\star_r$ $(r \in R)$ is the optimal solution of the following optimization problem over the variables y_r $(r \in R)$:*

$$\max\ -\sum_{r \in R} y_r (\log y_r - \log a_r)\ \text{ subject to } LC(\delta, \mathcal{F}, R). \tag{8}$$

In the following two subsections, we briefly discuss the problem of computing the weights a_r $(r \in R)$ and the problem of solving the optimization problem (8).

7.5 Computing the Weights a_r

We now discuss how to compute the weights a_r $(r \in R)$ in Theorem 7.8. In the sequel, let $P = (C, D)$ be a probabilistic logic program, and let $Q = \exists(\beta|\alpha)[x, y]$ be an object-ground probabilistic query. Let $R = R_C([\![D]\!] \cup \{\beta|\alpha\})$.

In general, the weights $a_r = |\{I \in \mathcal{I}_\Phi \mid I \models C \cup \{\wedge\, r\}\}|$ with $r \in R$ can be computed by simply counting all $I \in \mathcal{I}_\Phi$ such that $I \models C \cup \{\wedge\, r\}$.

In the case of conjunctive P and Q, they can be computed more efficiently by solving one system of linear equations. Let $S = R_\emptyset([\![ground(P)]\!] \cup \{\beta|\alpha\})$. For every $s \in S$, let I_s be the set of all ground atoms $p \in HB_\Phi$ with $s \models p$. By Lemma 7.1, the set S partitions $\mathcal{I}_\Phi$ into the sets $\{I \in \mathcal{I}_\Phi \mid I \models \wedge\, s\}$. We now first compute the numbers $b_s = |\{I \in \mathcal{I}_\Phi \mid I \models \wedge\, s\}|$ with $s \in S$, which are the unique solution of the following system of linear equations (assuming that P and Q are conjunctive):

$$\sum_{s \in S,\, I_{s^\star} \subseteq I_s} b_s \;=\; 2^{|HB_\Phi| - |I_{s^\star}|} \quad \text{(for all } s^\star \in S\text{)}.$$

The numbers a_r with $r \in R$ are then given as follows:

$$a_r \;=\; \sum_{s \in S,\, I_s \models C \cup \{\wedge r\}} b_s \quad \text{(for all } r \in R\text{)}.$$

7.6 Solving the Optimization Problem

The optimization problem (8) can be solved by standard Lagrange techniques (as e.g. described in [67, 71, 77]). Here, we can build on existing maximum entropy technology. For instance, the maximum entropy system PIT [16, 71] solves entropy maximization problems subject to *indifferent* possible worlds (which are known to have the same probability in me-models, cf. Lemma 7.7). This comes down to working with the index set $R_C([\![D]\!] \cup \{\beta|\alpha\})$. Therefore, PIT can be directly used to solve the optimization problem (8). The medical system LEXMED[2] [16, 70, 71] is based on PIT and supports physicians in diagnosing appendicitis in a German hospital (cf. [69]). *SPIRIT*[3] [67, 65, 66, 50] is another system shell using the principle of maximum entropy to represent sets of probabilistic rules and to answer queries.

To compute the me-model of some finite set of ground conditional constraints P, both PIT and SPIRIT adopt the conditional constraints in P successively and iteratively. Both systems make use of tree-like structures to reduce the complexity of probabilistic interpretations: The conditional constraints are learned on adequate component (or marginal) distributions, and the changed probabilities are propagated through the tree. So, like the clique trees of Bayesian networks [37, 60, 51, 27], these trees also allow local computations and propagations of probabilities. It is worth noticing that the Lagrange factors $\alpha^+_{\psi|\phi}$ and $\alpha^-_{\psi|\phi}$ occurring in (1), which are so meaningful for the theoretical results on maximum entropy reasoning (see [29, 31]), are also of crucial importance for the efficient computation in SPIRIT.

In principle, the proper handling of inequality constraints in (8) is no problem: The me-model of P fulfills some of the constraints in (8) with equality, some with strict inequality (cf. [77, 19]). Therefore, $me[P]$ is still of the form (1), with some of the factors $\alpha^+_{\psi|\phi}$ and $\alpha^-_{\psi|\phi}$ equal to 1. Unfortunately, no method is known to decide in advance which of the constraints are essential to compute the me-model (those where equality holds) and which are irrelevant (those where inequality holds). PIT uses heuristics to solve this problem when iteratively learning the conditional constraints; for a detailed description, see [71].

8 Efficient Reductions

In this section, we present efficient reductions of instances of POSITIVE PROBABILITY and TIGHT s-CONSEQUENCE, where $s \in \{0, me, mc\}$, to smaller instances of these problems. They are to be applied before generating the sophisticated systems of linear constraints in Theorems 7.3, 7.5, 7.6, and 7.8. They all aim at reducing their number of variables, which can be done by adding further classical knowledge and by reducing the number of ground instances of purely probabilistic conditional constraints. This can be achieved by (i) making hidden classical knowledge explicit, (ii) by removing vacuous conditional constraints, (iii) by removing inactive conditional constraints, and (iv) by decomposing a probabilistic logic program. Here, (i), (ii), and (iv) apply to the general case, while (iii) applies only to the conjunctive case. We show that (i)–(iii) can be done in polynomial time in the ground conjunctive case, and that (iv) can be done in linear time in the ground case. Note that (i)–(iii) are refinements of implicit techniques in [45], while (iv) is inspired by similar methods in [49, 14].

8.1 Adding Classical Conditional Constraints

We now describe a technique, which adds to a probabilistic logic program logically entailed classical conditional constraints. Observe that any newly derived classical conditional constraint reduces the number of variables in the systems of linear constraints in Theorems 7.3, 7.5, 7.6, and 7.8.

In the sequel, let $P = (C, D)$ be a probabilistic logic program, and let K be a set of classical conditional constraints. We first define the functions $triv_K$ and $triv^\star_K$, which associate with D a set of ground classical conditional constraints that trivially follow from K and D. The function $triv_K$ assigns to D the set of all conditional constraints $\bot \Leftarrow \phi$ (that is, $(\phi|\top)[0, 0]$) such that either

[2]Homepage of LEXMED: `https://lexmed.fh-weingarten.de/`.
[3]Available at `http://www.fernuni-hagen.de/BWLOR/forsch.html`.

(i) $K \models \bot \Leftarrow \psi \wedge \phi$, $(\psi|\phi)[l,u] \in D$, and $l > 0$, or

(ii) $K \models \psi \Leftarrow \phi$, $(\psi|\phi)[l,u] \in D$, and $u < 1$, or

(iii) $K \models \psi_1 \wedge \phi_1 \Leftrightarrow \psi_2 \wedge \phi_2$, $K \models \phi_1 \Leftrightarrow \phi_2$, $(\psi_1|\phi_1)[l_1,u_1] \in D$, $(\psi_2|\phi_2)[l_2,u_2] \in D$, and $[l_1,u_1] \cap [l_2,u_2] = \emptyset$.

We define $triv^0_K(D) = \emptyset$ and $triv^{n+1}_K(D) = triv_{K \cup triv^n_K(D)}(D)$ for all $n > 0$. We then define $triv^\star_K(D) = triv^n_K(D)$, where n is the least number $n \geq 0$ such that $triv^n_K(D) = triv^{n+1}_K(D)$.

The following theorem shows that, as far as POSITIVE PROBABILITY is concerned, we can simply add the classical conditional constraints in $triv^\star_C(D)$ to P. It follows from the fact that P is logically equivalent to $P^\star = P \cup triv^\star_C(D)$.

Theorem 8.1 *Let $P = (C,D)$ be a probabilistic logic program, and let α be a ground event. Let $P^\star = P \cup triv^\star_C(D)$. Then, P has a model Pr with $Pr(\alpha) > 0$ iff $P^\star$ has a model Pr with $Pr(\alpha) > 0$.*

The following example illustrates the above $P^\star = P \cup triv^\star_C(D)$.

Example 8.2 Consider the probabilistic logic program $P = (C,D)$, where $C = \{\bot \Leftarrow f \wedge g\}$ and $D = \{(e|f)[0.4, 0.5], (f|g)[0.6, 0.8]\}$. Then, we obtain $triv^\star_C(D) = \{\bot \Leftarrow g\}$, and thus $P^\star = P \cup \{\bot \Leftarrow g\}$. □

The next theorem shows that, as far as TIGHT s-CONSEQUENCE is concerned, where $s \in \{0, me, mc\}$, we can also simply add the classical conditional constraints in $triv^\star_C(D)$ to P. This result follows from P and $P^\star = P \cup triv^\star_C(D)$ being logically equivalent and from CWA$(P, \beta \wedge \alpha) =$ CWA$(P^\star, \beta \wedge \alpha)$.

Theorem 8.3 *Let $P = (C,D)$ be a probabilistic logic program, and let $(\beta|\alpha)[l,u]$ be a ground conditional constraint. Let $P^\star = P \cup triv^\star_C(D)$. Then, for every $s \in \{0, me, mc\}$, it holds that $P \Vdash^s_{tight} (\beta|\alpha)[l,u]$ iff $P^\star \Vdash^s_{tight} (\beta|\alpha)[l,u]$.*

The following proposition shows that, in the ground conjunctive case, $triv^\star_C(D)$ can be computed in time $O(\|P\||D|^3)$, where $\|P\|$ denotes the input size of P, and $|D|$ denotes the cardinality of D, that is, in polynomial time. It follows from the well-known result that for finite sets of ground conjunctive conditional constraints K and ground conjunctive conditional events $\psi|\phi$, deciding whether $K \models \bot \Leftarrow \psi \wedge \phi$ (resp., $K \models \psi \Leftarrow \phi$) holds can be done in linear time.

Proposition 8.4 *Given a ground conjunctive probabilistic logic program $P = (C,D)$, computing $triv^\star_C(D)$ can be done in time $O(\|P\||D|^3)$.*

8.2 Removing Vacuous Conditional Constraints

Another technique towards an increased efficiency is to remove conditional constraints that are vacuous by our classical knowledge.

In the sequel, let $P = (C,D)$ be a probabilistic logic program. A conditional constraint $(\psi|\phi)[l,u] \in D$ is *vacuous* under C iff either (i) $C \models \bot \Leftarrow \phi$, or (ii) $C \models \bot \Leftarrow \psi \wedge \phi$ and $l = 0$, or (iii) $C \models \psi \Leftarrow \phi$ and $u = 1$, or (iv) $l = 0$ and $u = 1$. We use $vac_C(D)$ to denote the set of all vacuous members of D under C.

The following theorem shows that P has a model Pr with $Pr(\alpha) > 0$ iff its equivalent without vacuous conditional constraints $P^\star = C \cup (D - vac_C(D))$ has such a model. It follows from the fact that P and $P^\star$ are logically equivalent.

Theorem 8.5 *Let $P = (C, D)$ be a probabilistic logic program, and let α be a ground event. Let $P^\star = C \cup (D - vac_C(D))$. Then, P has a model Pr with $Pr(\alpha) > 0$ iff $P^\star$ has a model Pr with $Pr(\alpha) > 0$.*

We give an example to illustrate the above $P^\star = C \cup (D - vac_C(D))$.

Example 8.6 Consider the probabilistic logic program $P{=}(C, D)$, where $C{=}\{\bot \Leftarrow g\}$ and $D{=}\{(e|f)[0.4, 0.5], (f|g)[0.6, 0.8]\}$. Then, $vac_C(D) = \{(f|g)[0.6, 0.8]\}$, and thus $P^\star = \{\bot \Leftarrow g, (e|f)[0.4, 0.5]\}$. □

The next theorem shows that, for $s \in \{0, me, mc\}$, s-entailment from P coincides with s-entailment from $P^\star = C \cup (D - vac_C(D))$. This result follows from P and $P^\star$ being logically equivalent and from CWA$(P, \beta \wedge \alpha) =$ CWA$(P^\star, \beta \wedge \alpha)$.

Theorem 8.7 *Let $P = (C, D)$ be a probabilistic logic program, and let $(\beta|\alpha)[l, u]$ be a ground conditional constraint. Let $P^\star = C \cup (D - vac_C(D))$. Then, for each $s \in \{0, me, mc\}$, it holds that $P \;\|\!\!\sim^{s}_{tight} (\beta|\alpha)[l, u]$ iff $P^\star \;\|\!\!\sim^{s}_{tight} (\beta|\alpha)[l, u]$.*

The following proposition shows that in the ground conjunctive case, $vac_C(D)$ can be computed in time $O(\|P\||D|)$, that is, in polynomial time.

Proposition 8.8 *Given a ground conjunctive probabilistic logic program $P = (C, D)$, computing $vac_C(D)$ can be done in time $O(\|P\||D|)$.*

8.3 Removing Inactive Conditional Constraints

We next describe a reduction, which only applies to the conjunctive case, and which characterizes some ground purely probabilistic conditional constraints as s-inactive, and simply removes them. Note that similar techniques have also proved to be useful in default reasoning from conditional knowledge bases [14].

In the sequel, let $P = (C, D)$ be a conjunctive probabilistic logic program. We now define the strong classical approximation of P, which is a superset of the classical approximation of P. Moreover, we define the notion of s-active formulas relative to the strong classical approximation of P. More formally, the *strong classical approximation* of P, denoted $s\text{-}app(P)$, is the set of all $\psi \Leftarrow \phi$ such that (i) $(\psi|\phi)[l, u] {\in} ground(P)$ for some $l > 0$, and (ii) $C \;\|\!\!\not\sim^{0} \bot \Leftarrow \phi$. Given a ground conjunctive event α, a ground atom $p {\in} HB_\Phi$ is *s-active* w.r.t. P and α iff $s\text{-}app(P) \cup \{\alpha\} \models p$. A ground event γ (resp., ground conditional constraint F) is *s-active* w.r.t. P and α iff all ground atoms in γ (resp., F) are s-active w.r.t. P and α. A ground atom (resp., ground event, ground conditional constraint) is *s-inactive* w.r.t. P and α iff it is not s-active w.r.t. P and α. We use $act_{C,\alpha}(D)$ to denote the set of (i) all members of D that are s-active w.r.t. $C \cup D$ and α, and (ii) all $\bot \Leftarrow \phi$ such that (a) ϕ is s-active w.r.t. $C \cup D$ and α, and (b) $(\psi|\phi)[l, u] \in D$ for some $l > 0$ and some ψ that is s-inactive w.r.t. $C \cup D$ and α. The following lemma is immediate.

Lemma 8.9 *Let $P = (C, D)$ be a conjunctive probabilistic logic program, and let α be a ground conjunctive event. Then, $app(P) \subseteq s\text{-}app(P)$. Moreover, if a ground atom $p \in HB_\Phi$ is active w.r.t. P and α, then it is also s-active w.r.t. P and α.*

The following theorem shows that deciding if P has a model Pr with $Pr(\alpha) > 0$ can be reduced to deciding whether $C \cup act_{C,\alpha}(D)$ has such a model. Roughly, it follows from the result that every s-inactive ground atom w.r.t. P and α can always be assigned the probability zero under models Pr of P with $Pr(\alpha) > 0$.

Theorem 8.10 *Let $P=(C,D)$ be a conjunctive probabilistic logic program, and let α be a ground conjunctive event. Let $P^\star = C \cup act_{C,\alpha}(D)$. Then, P has a model Pr with $Pr(\alpha)>0$ iff $P^\star$ has a model Pr with $Pr(\alpha)>0$.*

The following example illustrates the above result.

Example 8.11 Consider the conjunctive probabilistic logic program $P=(C,D)$, where $C=\{e \Leftarrow f\}$ and $D=\{(f|e)[0.1,0.2],(g|f)[0.1,0.2],(h|g)[0,0.2]\}$. Does P have a model Pr with $Pr(e)>0$? Then, since

$$s\text{-}app(P) \cup \{e\} = \{e \Leftarrow f, f \Leftarrow e, g \Leftarrow f, e\},$$

the ground atoms e, f, and g are all s-active w.r.t. P and e, while h is not s-active w.r.t. P and e. Thus, $act_{C,e}(D)=\{(f|e)[0.1,0.2],(g|f)[0.1,0.2]\}$. By Theorem 8.10, P has a model Pr with $Pr(e)>0$ iff $C \cup act_{C,e}(D)$ has such a model. □

The next result shows that, for $s \in \{0, mc\}$, s-entailment of ground conjunctive conditional constraints $(\beta|\alpha)[l,u]$ from P coincides with s-entailment of $(\beta|\alpha)[l,u]$ from $C \cup act_{C,\beta\wedge\alpha}(D)$. Note that this result does not carry over to me-entailment.

Theorem 8.12 *Let $P=(C,D)$ be a conjunctive probabilistic logic program and $(\beta|\alpha)[l,u]$ be a ground conjunctive conditional constraint. Let $P^\star = C \cup act_{C,\beta\wedge\alpha}(D)$. Then, for every $s \in \{0,mc\}$, it holds that $P \Vdash\!\sim^{s}_{tight} (\beta|\alpha)[l,u]$ iff $P^\star \Vdash\!\sim^{s}_{tight} (\beta|\alpha)[l,u]$.*

The following proposition shows that, in the ground conjunctive case, computing the set $act_{C,\alpha}(D)$ can be done in time $O(\|P\||P| + \|\alpha\|)$, where $\|\alpha\|$ denotes the input size of α, that is, in polynomial time.

Proposition 8.13 *Given a ground conjunctive probabilistic logic program $P=(C,D)$ and a ground conjunctive event α, computing $act_{C,\alpha}(D)$ can be done in time $O(\|P\||P|+\|\alpha\|)$.*

8.4 Decomposition

We now describe a reduction, which is based on the decomposition of the set of all purely probabilistic ground instances of a probabilistic logic program.

In the sequel, let $P=(C,D)$ be a probabilistic logic program, and let α be a ground event. We use $At(\alpha)$ to denote the set of all ground atoms $p \in HB_\Phi$ that occur in α. We use $HB_{P,\alpha}$ to denote $HB_P \cup At(\alpha)$. The *decomposition* of $HB_{P,\alpha}$ w.r.t. P and α is the unique partition $\{H_1,\ldots,H_k\}$ of $HB_{P,\alpha}$ such that (i) each member of $ground(P)$ is defined over some H_i with $i \in \{1,\ldots,k\}$, (ii) α is defined over some H_i with $i \in \{1,\ldots,k\}$, and (iii) $k \geq 1$ is maximal. For $i \in \{1,\ldots,k\}$, denote by D_i the set of all members of D that are defined over H_i. We call $\{D_1,\ldots,D_k\}$ the *decomposition* of D w.r.t. C and α, denoted $dec_{C,\alpha}(D)$. We call the unique D_i such that (i) $At(\alpha) \subseteq H_i$, and (ii) $i \in \{1,\ldots,k\}$ is minimal, the *relevant subset* of D w.r.t. C and α, denoted $rel_{C,\alpha}(D)$.

The following result shows that P has a model Pr with $Pr(\alpha)>0$ iff $C \cup rel_{C,\alpha}(D)$ has such a model and all the other $C \cup D_i$'s are satisfiable. Here, the "⇒"-part is immediate. The "⇐"-part follows from the fact that a model Pr of P can be constructed from models Pr_i of the $C \cup D_i$'s by assuming probabilistic independence.

Theorem 8.14 *Let $P=(C,D)$ be a probabilistic logic program, and let α be a ground event. Suppose that $dec_{C,\alpha}(D)=\{D_1,\ldots,D_k\}$ and $rel_{C,\alpha}(D)=D_1$. Then, P has a model Pr with $Pr(\alpha)>0$ iff*

(i) $C \cup D_1$ *has a model* Pr *with* $Pr(\alpha) > 0$, *and*

(ii) *for every* $i \in \{2, \ldots, k\}$, *it holds that* $C \cup D_i$ *is satisfiable.*

The following example illustrates the above result.

Example 8.15 Let the probabilistic logic program $P = (C, D)$, which is defined over the set of ground atoms $HB_P = \{e, f, g, h, k, l\}$, be given as follows:

$$P = (\{e \Leftarrow f\}, \{(f|e \wedge g)[\tfrac{3}{4}, 1], (\neg e \vee f|g)[\tfrac{3}{5}, 1], (h|k \wedge l)[\tfrac{2}{3}, 1], (k|h \wedge l)[\tfrac{5}{8}, 1]\}) .$$

Then, the decomposition of D w.r.t. C and e is given by $dec_{C,e}(D) = \{D_1, D_2\}$, where D_1 and D_2 over $H_1 = \{e, f, g\}$ and $H_2 = \{h, k, l\}$, respectively, are given by:

$$\begin{aligned} D_1 &= \{(f|e \wedge g)[\tfrac{3}{4}, 1], (\neg e \vee f|g)[\tfrac{3}{5}, 1]\} , \\ D_2 &= \{(h|k \wedge l)[\tfrac{2}{3}, 1], (k|h \wedge l)[\tfrac{5}{8}, 1]\} . \end{aligned}$$

By Theorem 8.14, P has a model Pr with $Pr(e) > 0$ iff (i) $C \cup D_1$ has a model Pr with $Pr(e) > 0$, and (ii) $C \cup D_2$ is satisfiable. □

The following theorem shows that computing tight answer substitutions for object-ground queries Q to P under 0-, me-, and mc-entailment can be reduced to computing tight answer substitutions for Q to $C \cup rel_{C,\,\beta \wedge \alpha}(D)$ under 0-, me-, and mc-entailment, respectively, and to checking satisfiability of the other $C \cup D_j$.

Theorem 8.16 *Let* $P = (C, D)$ *be a probabilistic logic program, and let* $(\beta|\alpha)[l, u]$ *be a ground conditional constraint. Let* $dec_{C,\,\beta \wedge \alpha}(D) = \{D_1, \ldots, D_k\}$ *and* $rel_{C,\,\beta \wedge \alpha}(D) = D_1$. *Then, for every* $s \in \{0, me, mc\}$*:*

(a) *If every* $C \cup D_i$ *with* $i \in \{2, \ldots, k\}$ *is satisfiable,*
then $P \mathrel{\|\!\sim}^{s}_{tight} (\beta|\alpha)[l, u]$ *iff* $C \cup D_1 \mathrel{\|\!\sim}^{s}_{tight} (\beta|\alpha)[l, u]$.

(b) *Otherwise,* $P \mathrel{\|\!\sim}^{s}_{tight} (\beta|\alpha)[1, 0]$.

The following result shows that, in the ground case, computing the decomposition and the relevant subset can be done in time $O(\|P\| + \|\alpha\|)$, that is, in linear time. It follows from a reduction to the problem of computing the connected components of a hypergraph, which can be done in linear time.

Proposition 8.17 *Given a ground probabilistic logic program* $P = (C, D)$ *and a ground event* α, $dec_{C,\alpha}(D)$ *and* $rel_{C,\alpha}(D)$ *can be computed in time* $O(\|P\| + \|\alpha\|)$.

9 Reduction-Based Algorithms

In this section, we present algorithms for solving the problems POSITIVE PROBABILITY and TIGHT s-CONSEQUENCE, where $s \in \{0, me, mc\}$, which are based on the techniques of exploiting classical knowledge and clustering possible worlds of Section 7 and on the efficient reductions described in Section 8.

Algorithm Positive_Probability

Input: Probabilistic logic program $P = (C, D)$ and ground event α.

Output: "No", if $P \,\|\!\!\sim^0 \bot \Leftarrow \alpha$; "Yes", otherwise.

1. $C := C \cup triv^{\star}_{C}(D)$;
2. **if** $C \,\|\!\!\sim^0 \bot \Leftarrow \alpha$ **then return** "No";
3. $D := D - vac_C(D)$;
4. **if** $C \cup D$ and α are conjunctive **then** $D := act_{C,\alpha}(D)$;
5. $D_i := rel_{C,\alpha}(D)$;
6. $R := R_C([\![D_i]\!] \cup \{\alpha|\top\})$;
7. **if** $LC(\alpha, D_i, R)$ is unsolvable **then return** "No";
8. **for each** $D_i \in dec_{C,\alpha}(D) - \{rel_{C,\alpha}(D)\}$ **do begin**
9. $\quad R := R_C([\![D_i]\!])$;
10. $\quad$ **if** $LC(\top, D_i, R)$ is unsolvable **then return** "No"
11. **end**;
12. **return** "Yes".

Figure 3: Algorithm Positive_Probability.

9.1 Positive Probability

Algorithm Positive_Probability (see Fig. 3) decides, given a probabilistic logic program $P = (C, D)$ and a ground event α, whether P has a model Pr such that $Pr(\alpha) > 0$. In step 1, we add trivially entailed classical knowledge to C. In step 2, we then check whether already C logically entails $\bot \Leftarrow \alpha$. In step 3, we then remove all vacuous conditional constraints from D, while in step 4, in the conjunctive case, we remove all s-inactive conditional constraints from D. In steps 5-7, we decide whether $C \cup rel_{C,\alpha}(D)$ has a model Pr with $Pr(\alpha) > 0$, while in steps 8-11, we decide whether all the other $C \cup D_i$ where $D_i \in dec_{C,\alpha}(D)$ are satisfiable.

The following theorem shows that Algorithm Positive_Probability is correct. It follows immediately from Theorems 7.5, 8.1, 8.5, 8.10, and 8.14.

Theorem 9.1 *Let P be a probabilistic logic program, and let α be a ground event. Then,* Positive_Probability(P, α) *is "No", if $P \,\|\!\!\sim^0 \bot \Leftarrow \alpha$, and "Yes", otherwise.*

The following example illustrates Algorithm Positive_Probability.

Example 9.2 Let $P = (C, D)$ be the probabilistic logic program given in Example 4.1, and let $\alpha = ad(u, a)$. To decide whether there exists a model Pr of P such that $Pr(\alpha) > 0$, Algorithm Positive_Probability generates a system of *six* linear constraints over *six* variables, as the following construction shows.

The sets C and D are given as follows:

$$\begin{aligned}
C \;=\;& \{(ro(h,u) \mid \top)[1,1], (ro(u,a) \mid \top)[1,1], (ro(a,o) \mid \top)[1,1],\\
& (ad(h,u) \mid \top)[1,1], (re(R,S) \mid ro(R,S) \wedge ad(R,S))[1,1],\\
& (so(a,o) \mid \top)[1,1], (re(R,S) \mid re(R,T) \wedge re(T,S))[1,1]\}\,,\\
D \;=\;& ground(\{(ad(u,a) \mid \top)[0.8,0.9], (re(R,S) \mid ro(R,S))[0.7,1],\\
& \qquad (re(R,S) \mid ro(R,S) \wedge so(R,S))[0.9,1]\})\,.
\end{aligned}$$

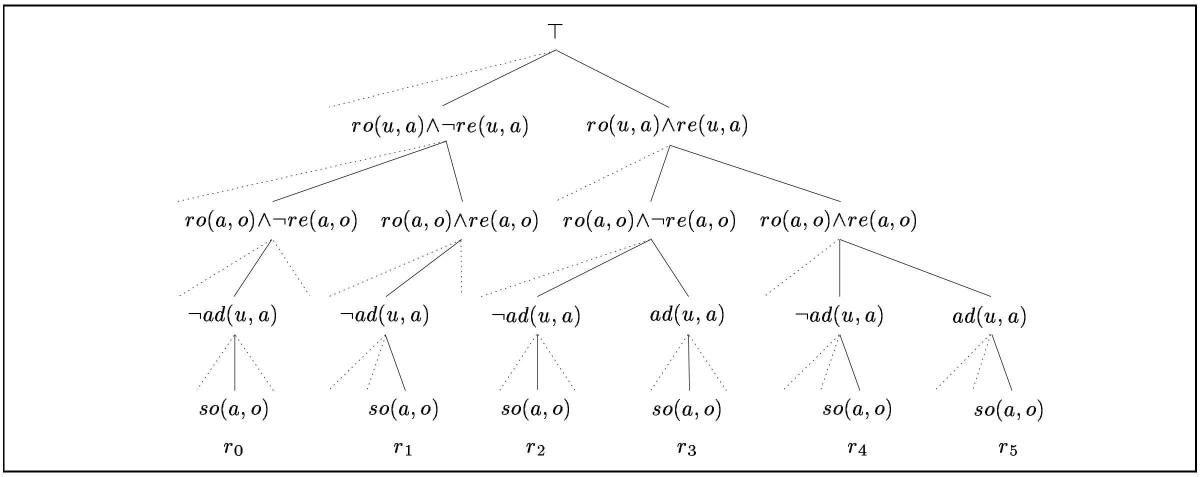


Figure 4: Semantic tree for Example 9.2.

Algorithm Positive_Probability now runs through steps 1-3 without changing C and D. In particular, it does not stop at step 2, as C does not logically entail $\bot \Leftarrow \alpha$. Since $C \cup D$ and α are conjunctive, we compute $act_{C,\alpha}(D)$ in step 4. For this, we first compute the strong classical approximation of P, which is given by:

$$\begin{aligned} s\text{-}app(P) \;=\;& ground(\{(ro(h,u) \mid \top)[1,1], (ro(u,a) \mid \top)[1,1], (ro(a,o) \mid \top)[1,1], \\ & (ad(h,u) \mid \top)[1,1], (re(R,S) \mid ro(R,S) \wedge ad(R,S))[1,1], \\ & (so(a,o) \mid \top)[1,1], (re(R,S) \mid re(R,T) \wedge re(T,S))[1,1], \\ & (ad(u,a) \mid \top)[1,1], (re(R,S) \mid ro(R,S))[1,1], \\ & (re(R,S) \mid ro(R,S) \wedge so(R,S))[1,1]\})\,. \end{aligned}$$

The set of all ground atoms $p \in HB_\Phi$ that are s-active w.r.t. P and α is given by the least Herbrand model of $s\text{-}app(P) \cup \{ad(u,a)\}$, and thus given as follows:

$$\begin{aligned} &\{ro(h,u), ro(u,a), ro(a,o), ad(h,u), ad(u,a), so(a,o), \\ &\; re(h,u), re(u,a), re(a,o), re(h,a), re(u,o), re(h,o)\}\,. \end{aligned}$$

The set $act_{C,\alpha}(D)$ is then given as follows:

$$\begin{aligned} act_{C,\alpha}(D) \;=\;& \{(re(u,a) \mid ro(u,a))[0.7,1], (re(a,o) \mid ro(a,o))[0.7,1], \\ & (ad(u,a) \mid \top)[0.8,0.9], (re(a,o) \mid ro(a,o) \wedge so(a,o))[0.9,1]\}\,. \end{aligned}$$

It is easy to verify that in steps 5 and 8, it holds that $D_i = rel_{C,\alpha}(D) = act_{C,\alpha}(D)$ and $dec_{C,\alpha}(D) = \{act_{C,\alpha}(D)\}$, respectively, and thus we only have to decide whether the system of linear constraints $LC(\alpha, act_{C,\alpha}(D), R)$ in step 7 is solvable. Here, we have the index set $R = \{r_i \mid i \in \{0,\ldots,5\}\}$, where the r_i's correspond to the leaves of the directed tree shown in Fig. 4. More precisely, every $\wedge r_i$ is logically equivalent to the conjunction of all labels along the path from the root labeled $\top$ to the leaf associated with r_i. For example, $\wedge r_5$ is logically equivalent to $ro(u,a) \wedge re(u,a) \wedge ro(a,o) \wedge re(a,o) \wedge ad(u,a) \wedge so(a,o)$.

Algorithm Tight_0_Consequence

Input: Probabilistic logic program $P = (C, D)$ and object-ground probabilistic query $Q = \exists(\beta|\alpha)[x, y]$.

Output: Tight answer substitution σ for Q to P under logical entailment.

1. **if** $P \Vdash\!\sim^0 \bot \Leftarrow \alpha$ **then return** $\sigma = \{x/1, y/0\}$
2. **else if** $P \Vdash\!\sim^0 \bot \Leftarrow \alpha \wedge \beta$ **then return** $\sigma = \{x/0, y/0\}$
3. **else if** $P \Vdash\!\sim^0 \bot \Leftarrow \alpha \wedge \neg\beta$ **then return** $\sigma = \{x/1, y/1\}$;
4. $C := C \cup \{\bot \Leftarrow \phi \mid \exists\, (\psi|\phi)[l, u] \in D \colon P \Vdash\!\sim^0 \bot \Leftarrow \phi\}$;
5. $D := D - vac_C(D)$;
6. **if** $C \cup D$ and $\alpha \wedge \beta$ are conjunctive **then** $D := act_{C,\beta\wedge\alpha}(D)$;
7. $D := rel_{C,\beta\wedge\alpha}(D)$;
8. $R := R_C([\![D]\!] \cup \{\beta|\alpha\})$;
9. l (resp., u) := min (resp., max) $\sum_{r \in R,\, r \models \beta\wedge\alpha} y_r$ subject to $LC(\alpha, D, R)$;
10. **return** $\sigma = \{x/l, y/u\}$.

Figure 5: Algorithm Tight_0_Consequence.

The system of linear constraints $LC(\alpha, act_{C,\alpha}(D), R)$ is then given as follows (where each variable y_i with $i \in \{0, \ldots, 5\}$ corresponds to r_i):

$$\begin{aligned}
y_3 + y_5 &= 1 \\
-0.7 \cdot (y_0 + y_1) + 0.3 \cdot (y_2 + y_3 + y_4 + y_5) &\geq 0 \\
-0.7 \cdot (y_0 + y_2 + y_3) + 0.3 \cdot (y_1 + y_4 + y_5) &\geq 0 \\
-0.8 \cdot (y_0 + y_1 + y_2 + y_4) + 0.2 \cdot (y_3 + y_5) &\geq 0 \\
0.9 \cdot (y_0 + y_1 + y_2 + y_4) - 0.1 \cdot (y_3 + y_5) &\geq 0 \\
-0.9 \cdot (y_0 + y_2 + y_3) + 0.1 \cdot (y_1 + y_4 + y_5) &\geq 0 \\
y_i &\geq 0 \quad \text{(for all } i \in \{0, \ldots, 5\}).
\end{aligned}$$

This system is solvable, and thus Algorithm Positive_Probability returns "Yes" in step 12. By Theorem 9.1, P has a model Pr such that $Pr(\alpha) > 0$. □

9.2 Tight Logical Consequence

Algorithm Tight_0_Consequence (see Fig. 5) computes, given a probabilistic logic program $P{=}(C, D)$ and an object-ground probabilistic query $Q{=}\exists(\beta|\alpha)[x,y]$, the tight answer substitution for Q to P under logical entailment. In steps 1-3, we first check whether P logically entails either $\bot \Leftarrow \alpha$, or $\bot \Leftarrow \beta \wedge \alpha$, or $\beta \Leftarrow \alpha$, which can be done using Algorithm Positive_Probability. If this is the case, then we immediately return either $\{x/1, y/0\}$, or $\{x/0, y/0\}$, or $\{x/1, y/1\}$, respectively. Otherwise, in step 4, we add to C some classical knowledge that is entailed by P. Here, the set of all added classical conditional constraints is a superset of $triv^{\star}_C(D)$ and can be computed using Algorithm Positive_Probability. In steps 5-6, we then remove all vacuous and s-inactive conditional constraints. In steps 7-9, we finally compute the interval $[l, u]$ such that $C \cup rel_{C,\,\beta\wedge\alpha}(D) \Vdash\!\sim^0_{tight} (\beta|\alpha)[l, u]$.

The following theorem shows that Algorithm Tight_0_Consequence is correct. This result follows immediately from Theorem 7.6, a slight generalization of Theorem 8.3, and Theorems 8.7, 8.12, and 8.16.

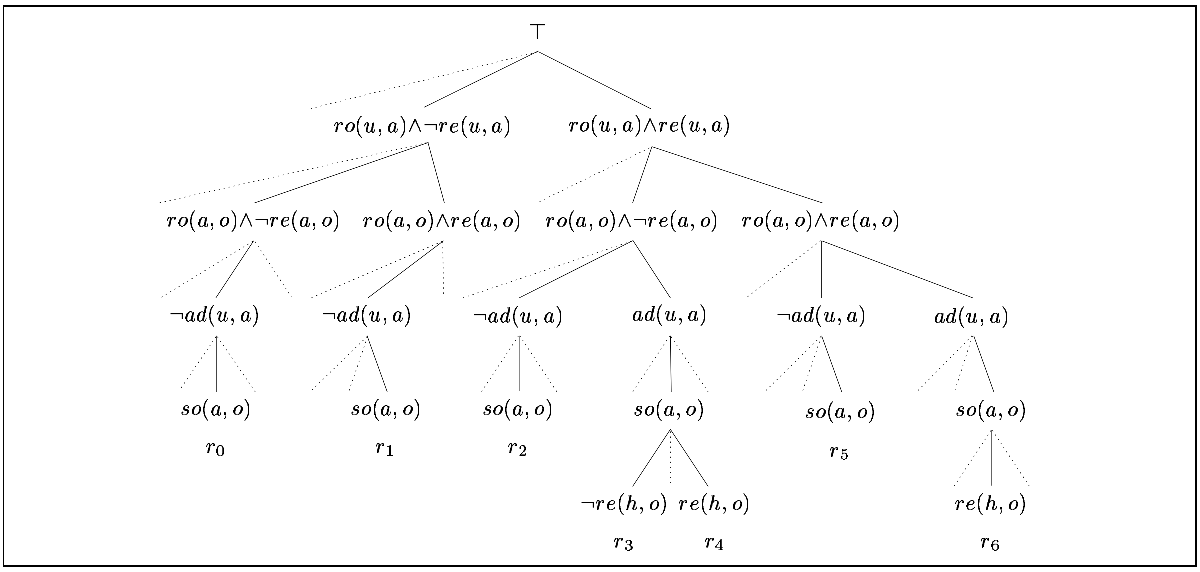


Figure 6: Semantic tree for Example 9.4.

Theorem 9.3 *Let P be a probabilistic logic program, and let Q be an object-ground probabilistic query. Then,* Tight_0_Consequence(P, Q) *is the tight answer substitution for Q to P under logical entailment.*

We give an example to illustrate Algorithm Tight_0_Consequence.

Example 9.4 Consider the probabilistic logic program P given in Example 4.1 and the object-ground probabilistic query $Q = \exists(re(h,o)|ad(u,a))[X,Y]$. In order to compute the tight answer substitution for Q to P under logical entailment, Algorithm Tight_0_Consequence generates two linear programs, each of which consists of only *six* linear constraints over *seven* variables.

In detail, we have $\beta = re(h,o)$ and $\alpha = ad(u,a)$. The system of linear constraints $LC(\alpha, D, R)$ that we use in step 9 is then given through $D = act_{C,\beta\wedge\alpha}(D)$, which coincides with $act_{C,\alpha}(D)$ of Example 9.2, and $R = \{r_i \mid i \in \{0,\ldots,6\}\}$, where the r_i's correspond to the leaves of the directed tree in Fig. 6. The two linear programs are then given as follows (where each y_i with $i \in \{0,\ldots,6\}$ corresponds to r_i):

minimize (resp., maximize) $y_4 + y_6$

subject to

$$\begin{aligned}
y_3 + y_4 + y_6 &= 1 \\
-0.7 \cdot (y_0 + y_1) + 0.3 \cdot (y_2 + y_3 + y_4 + y_5 + y_6) &\geq 0 \\
-0.7 \cdot (y_0 + y_2 + y_3 + y_4) + 0.3 \cdot (y_1 + y_5 + y_6) &\geq 0 \\
-0.8 \cdot (y_0 + y_1 + y_2 + y_5) + 0.2 \cdot (y_3 + y_4 + y_6) &\geq 0 \\
0.9 \cdot (y_0 + y_1 + y_2 + y_5) - 0.1 \cdot (y_3 + y_4 + y_6) &\geq 0 \\
-0.9 \cdot (y_0 + y_2 + y_3 + y_4) + 0.1 \cdot (y_1 + y_5 + y_6) &\geq 0 \\
y_i &\geq 0 \quad \text{(for all } i \in \{0,\ldots,6\}\text{).}
\end{aligned}$$

Algorithm Tight_me_Consequence

Input: Probabilistic logic program $P = (C, D)$ and object-ground probabilistic query $Q = \exists(\beta|\alpha)[x, y]$.

Output: Tight answer substitution σ for Q to P under me-entailment.

1. **if** $P \,\|\!\!\sim^{0} \bot \Leftarrow \alpha$ **then return** $\sigma = \{x/1, y/0\}$
2. **else if** $P \,\|\!\!\sim^{0} \bot \Leftarrow \alpha \wedge \beta$ **then return** $\sigma = \{x/0, y/0\}$
3. **else if** $P \,\|\!\!\sim^{0} \bot \Leftarrow \alpha \wedge \neg\beta$ **then return** $\sigma = \{x/1, y/1\}$;
4. $C := C \cup \{\bot \Leftarrow \phi \mid \exists\,(\psi|\phi)[l, u] \in D \colon P \,\|\!\!\sim^{0} \bot \Leftarrow \phi\}$;
5. $D := D - vac_C(D)$;
6. $D := rel_{C,\beta\wedge\alpha}(D)$;
7. $R := R_C([\![D]\!] \cup \{\beta|\alpha\})$;
8. compute the weights a_r $(r \in R)$;
9. compute the optimal solution $y^\star_r$ $(r \in R)$ of the optimization problem
10. $\max\; -\sum_{r \in R} y_r(\log y_r - \log a_r)$ subject to $LC(\top, D, R)$;
11. $d := (\sum_{r \in R,\, r \models \beta\wedge\alpha} y^\star_r) \,/\, (\sum_{r \in R,\, r \models \alpha} y^\star_r)$;
12. **return** $\sigma = \{x/d, y/d\}$.

Figure 7: Algorithm Tight_me_Consequence.

The optimal values are 0.875 and 1, respectively. Thus, by Theorem 9.3, the tight answer substitution for Q to P under logical entailment is $\{X/0.875,\ Y/1\}$. □

9.3 Tight Consequence under Maximum Entropy

Tight answer substitutions under me-entailment can be computed with Algorithm Tight_me_Consequence (see Fig. 7), which is very similar to Tight_0_Consequence. The only differences are that, in Tight_me_Consequence, we cannot remove anymore s-inactive conditional constraints (step 6 of Tight_0_Consequence), and we perform an entropy maximization in steps 7–11 rather than solving two linear programs (steps 7–9 of Tight_0_Consequence).

The next result shows that Algorithm Tight_me_Consequence is correct. It is immediate by a slight generalization of Theorem 8.3 and Theorems 7.8, 8.7, and 8.16.

Theorem 9.5 *Let P be a probabilistic logic program, and let Q be an object-ground probabilistic query. Then,* Tight_me_Consequence(P, Q) *is the tight answer substitution for Q to P under me-entailment.*

9.4 Tight Consequence under Maximum Entropy and CWA

Given a conjunctive probabilistic logic program $P = (C, D)$ and a conjunctive object-ground probabilistic query $Q = \exists(\beta|\alpha)[x, y]$, the tight answer substitution for Q to P under mc-entailment can be computed with Algorithm Tight_mc_Consequence (see Fig. 8), which is nearly identical to Tight_me_Consequence, except that we now also remove s-inactive conditional constraints in step 6.

The following theorem shows that Algorithm Tight_mc_Consequence is correct. This result follows immediately from Theorem 7.8, a slight generalization of Theorem 8.3, and Theorems 8.7, 8.12, and 8.16.

Algorithm Tight_mc_Consequence

Input: Conjunctive probabilistic logic program $P = (C, D)$ and conjunctive object-ground probabilistic query $Q = \exists(\beta|\alpha)[x, y]$.

Output: Tight answer substitution σ for Q to P under mc-entailment.

1. **if** $P \,\|\!\!\sim^0 \bot \Leftarrow \alpha$ **then return** $\sigma = \{x/1, y/0\}$
2. **else if** $P \,\|\!\!\sim^0 \bot \Leftarrow \alpha \wedge \beta$ **then return** $\sigma = \{x/0, y/0\}$
3. **else if** $P \,\|\!\!\sim^0 \bot \Leftarrow \alpha \wedge \neg\beta$ **then return** $\sigma = \{x/1, y/1\}$;
4. $C := C \cup \{\bot \Leftarrow \phi \mid \exists\, (\psi|\phi)[l, u] \in D\colon P \,\|\!\!\sim^0 \bot \Leftarrow \phi\}$;
5. $D := D - vac_C(D)$;
6. $D := act_{C,\beta\wedge\alpha}(D)$;
7. $D := rel_{C,\beta\wedge\alpha}(D)$;
8. $R := R_C([\![D]\!] \cup \{\beta|\alpha\})$;
9. compute the weights a_r $(r \in R)$;
10. compute the optimal solution $y^\star_r$ $(r \in R)$ of the optimization problem
11. max $-\sum_{r\in R} y_r(\log y_r - \log a_r)$ subject to $LC(\top, D, R)$;
12. $d := (\sum_{r\in R,\, r\models\beta\wedge\alpha} y^\star_r) \,/\, (\sum_{r\in R,\, r\models\alpha} y^\star_r)$;
13. **return** $\sigma = \{x/d, y/d\}$.

Figure 8: Algorithm Tight_mc_Consequence.

Theorem 9.6 *Let P be a probabilistic logic program, and let Q be an object-ground probabilistic query. Then,* Tight_mc_Consequence(P, Q) *is the tight answer substitution for Q to P under mc-entailment.*

The following example illustrates Algorithm Tight_mc_Consequence.

Example 9.7 Consider again the probabilistic logic program $P = (C, D)$ of Example 4.1 and the object-ground probabilistic query $Q = \exists(re(h, o)|ad(u, a))[X, Y]$. In order to compute the tight answer substitution for Q to P under mc-entailment, Algorithm Tight_mc_Consequence generates an entropy maximization problem subject to a system of only *six* linear constraints over *seven* variables.

More precisely, in steps 7–11, we compute the unique optimal solution of

$$\max\ -\sum_{r\in R} y_r(\log y_r - \log a_r) \text{ subject to } LC(\top, D, R)\,,$$

where D and R are the same as in Example 9.4, and every weight a_r with $r \in R$ is given by $|\{I \in \mathcal{I}_\Phi \mid I \models C \cup \{\wedge\, r\}\}|$. Here, there are 18 possible worlds $I \in \mathcal{I}_\Phi$ such that $I \models C$, which are partitioned as follows through $R = \{r_0, \ldots, r_6\}$:

$$r_0 \mathrel{\hat=} \{J_0, \ldots, J_5\},\ \ r_1 \mathrel{\hat=} \{J_9, \ldots, J_{13}\},\ \ r_2 \mathrel{\hat=} \{J_6, \ldots, J_8\},$$
$$r_3 \mathrel{\hat=} \{J_{15}\},\ \ r_4 \mathrel{\hat=} \{J_{16}, J_{17}\},\ \ r_5 \mathrel{\hat=} \{J_{14}\},\ \ r_6 \mathrel{\hat=} \{J_{18}\}\,.$$

Thus, $(a_0, a_1, a_2, a_3, a_4, a_5, a_6) = (6, 5, 3, 1, 2, 1, 1)$, and we get the following optimization problem (where each y_i and a_i with $i \in \{0,\ldots,6\}$ corresponds to r_i):

$$\text{maximize} \quad -\sum_{i=0}^{6} y_i\,(\log y_i - \log a_i)$$

subject to

$$\begin{aligned}
y_0 + y_1 + y_2 + y_3 + y_4 + y_5 + y_6 &= 1\\
-0.7 \cdot (y_0 + y_1) + 0.3 \cdot (y_2 + y_3 + y_4 + y_5 + y_6) &\geq 0\\
-0.7 \cdot (y_0 + y_2 + y_3 + y_4) + 0.3 \cdot (y_1 + y_5 + y_6) &\geq 0\\
-0.8 \cdot (y_0 + y_1 + y_2 + y_5) + 0.2 \cdot (y_3 + y_4 + y_6) &\geq 0\\
0.9 \cdot (y_0 + y_1 + y_2 + y_5) - 0.1 \cdot (y_3 + y_4 + y_6) &\geq 0\\
-0.9 \cdot (y_0 + y_2 + y_3 + y_4) + 0.1 \cdot (y_1 + y_5 + y_6) &\geq 0\\
y_i &\geq 0 \quad (\text{for all } i \in \{0,\ldots,6\}).
\end{aligned}$$

The tight answer substitution for Q to P under mc-entailment is then $\{X/d,\, Y/d\}$, where $d = (y_4^\star + y_6^\star)\,/\,(y_3^\star + y_4^\star + y_6^\star)$ and $y_i^\star$ $(i \in \{0,\ldots,6\})$ is the optimal solution of the above optimization problem. Numerically, it is $\{X/0.9632,\ Y/0.9632\}$. □

10 Conclusion

In this paper, we presented two approaches to probabilistic logic programming under maximum entropy, which are based on the usual notion of entailment under maximum entropy (me-entailment), and a new notion of entailment under maximum entropy (mc-entailment) that couples the principle of maximum entropy with the closed world assumption (CWA) from classical logic programming. We analyzed the nonmonotonic behavior of both approaches along benchmark examples and along general properties for default reasoning from conditional knowledge bases. It turned out that both approaches have very nice nonmonotonic features. Furthermore, we presented algorithms for computing tight intervals from probabilistic logic programs under me- and mc-entailment, which are based on generalizations of techniques from [45]. In particular, computing tight intervals under mc-entailment is reduced to an optimization problem of the same size as the one produced by computing tight intervals under logical entailment in [45].

We have used the principle of maximum entropy as a way to overcome the inferential weakness of model-theoretic logical entailment. This approach has a number of advantages over the Bayesian network approaches in [63, 62, 22, 55, 24, 25]: Since the latter approaches originated from Bayesian networks, they all assume some strong structural restrictions on probabilistic knowledge bases. In particular, they all require that the grounding of a knowledge base is acyclic. Moreover, conditional probabilities are always given by a precise point value, rather than by an interval. Our work, in contrast, is free from such strong restrictions.

The me-model of a probabilistic logic program P automatically satisfies conditional independencies that are implicitly entrenched in the structure of P. Roughly, we find conditional independencies in the me-model where the available information does not justify establishing a dependency. In general, however, the me-model of a probabilistic logic program P that represents a fragment of a Bayesian network BN does not satisfy the conditional independencies implicitly encoded in BN. However, a simulation of Bayesian networks is possible through a slight modification of the usual me-approach: In order to obtain maximum entropy models that automatically encode conditional independencies as in Bayesian networks, one can use the principle of sequential maximum entropy introduced in [43].

There may be applications in practice where the imprecision that is expressed by the width of the probability intervals in a probabilistic logic program should also somehow be reflected in the concluded intervals. In such cases, the approaches to probabilistic logic programming under inheritance with overriding introduced in [44] are better suited than our approaches to probabilistic logic programming under maximum entropy. Observe, however, that the notions of entailment under inheritance with overriding in [44] are closely related to entailment under maximum entropy, as they all have very similar nonmonotonic properties, in particular, they all realize some inheritance of probabilistic knowledge. Exploring in some more detail this relationship between entailment under inheritance with overriding and entailment under maximum entropy is an exciting topic of further research.

Another interesting topic of future research is to generalize the notion of entailment under maximum entropy and CWA to a larger class of probabilistic logic programs, beyond those over conjunctive events. A closely related issue of further research is to extend the reduction of removing s-inactive conditional constraints described in Section 8.3 to a larger class of probabilistic logic programs.

A Appendix: Proofs for Section 3

Proof of Lemma 3.1. Recall that $P \,\|\!\!\sim^0 (\beta|\alpha)[l,u]$ iff every model Pr of P is also a model of $(\beta|\alpha)[l,u]$. The latter is equivalent to $Pr(\beta|\alpha) \in [l,u]$ for every model Pr of P such that $Pr(\alpha) > 0$, which in turn is equivalent to $Pr_\alpha(\beta) \in [l,u]$ for every model Pr of P such that $Pr(\alpha) > 0$. This argumentation also shows that $P \,\|\!\!\sim^0_{tight} (\beta|\alpha)[l,u]$ iff l (resp., u) is the infimum (resp., supremum) of $Pr_\alpha(\beta)$ subject to all models Pr of P such that $Pr(\alpha) > 0$. $\Box$

The following lemma will be used in the proof of Theorem 3.2.

Lemma A.1 *Let P be a conjunctive probabilistic logic program, and let α be a ground conjunctive event. Then, for every model Pr of P, there exists a model $Pr^\star$ of $P \cup \mathrm{CWA}(P,\alpha)$ such that $Pr^\star(\gamma) = Pr(\gamma)$ for every ground event γ that is active w.r.t. P and α.*

Proof of Lemma A.1. Let Pr be a model of P. We define $Pr^\star$ by $Pr^\star(I) = Pr(\varepsilon_I)$ for all $I \in \mathcal{I}_\Phi$ with $I \models \mathrm{CWA}(P,\alpha)$, where ε_I is the conjunction of all active atoms $p \in I$ and of all negations of active atoms $p \notin I$, and by $Pr^\star(I) = 0$ for all other $I \in \mathcal{I}_\Phi$. Clearly, $Pr^\star$ satisfies $\mathrm{CWA}(P,\alpha)$, and $Pr^\star(\gamma) = Pr(\gamma)$ for all active ground events γ. Hence, $Pr^\star$ satisfies all active members of $ground(P)$. We now show that $Pr^\star$ also satisfies all inactive members of $ground(P)$. Suppose the contrary. That is, some inactive $(\psi|\phi)[l,u] \in ground(P)$ exists such that $Pr^\star \not\models (\psi|\phi)[l,u]$. It then follows that $Pr^\star(\phi) > 0$, as otherwise $Pr^\star \models (\psi|\phi)[l,u]$. Hence, ϕ is active, as otherwise $Pr^\star(\phi) = 0$. Since $(\psi|\phi)[l,u]$ is inactive, it thus follows that ψ is inactive. Hence, we obtain $l > 0$, as otherwise $Pr^\star \models (\psi|\phi)[l,u]$. Furthermore, as $Pr(\phi) = Pr^\star(\phi) > 0$, it follows that $P \,\|\!\!\not\sim^0 \bot \Leftarrow \phi$. This shows in particular that $\psi \Leftarrow \phi$ belongs to $app(P)$. But this contradicts ϕ being active and ψ being inactive. This shows that $Pr^\star$ is also a model of all inactive members of $ground(P)$. $\Box$

Proof of Theorem 3.2. ($\Rightarrow$) Since the consequence relation $\|\!\!\sim^0$ is monotonic, $P \,\|\!\!\sim^0 (\beta|\alpha)[l,u]$ implies $P \cup \mathrm{CWA}(P,\beta \wedge \alpha) \,\|\!\!\sim^0 (\beta|\alpha)[l,u]$.

($\Leftarrow$) Assume that $(*)$ $P \cup \mathrm{CWA}(P,\beta \wedge \alpha) \,\|\!\!\sim^0 (\beta|\alpha)[l,u]$. Towards a contradiction, suppose now that $P \,\|\!\!\not\sim^0 (\beta|\alpha)[l,u]$. That is, there exists a model Pr of P such that $Pr \not\models (\beta|\alpha)[l,u]$. Since $(\beta|\alpha)[l,u]$ is active, by Lemma A.1, there exists a model $Pr^\star$ of $P \cup \mathrm{CWA}(P,\beta \wedge \alpha)$ such that $Pr^\star \not\models (\beta|\alpha)[l,u]$. But this then contradicts $(*)$. This shows that $P \,\|\!\!\sim^0 (\beta|\alpha)[l,u]$. $\Box$

Proof of Theorem 3.3. We first show that $P \cup \mathrm{CWA}(P, \beta \wedge \alpha)$ is logically equivalent to $\widehat{P} \cup \mathrm{CWA}(P, \beta \wedge \alpha)$. Observe that P logically entails all $\bot \Leftarrow \phi$ such that (a) ϕ is active w.r.t. P and $\beta \wedge \alpha$, and (b) $(\psi|\phi)[r,s] \in ground(P)$ for some $r > 0$ and some ψ that is inactive w.r.t. P and $\beta \wedge \alpha$. Thus, every model of $P \cup \mathrm{CWA}(P, \beta \wedge \alpha)$ is also a model of $\widehat{P} \cup \mathrm{CWA}(P, \beta \wedge \alpha)$. To prove the converse, it is sufficient to show that $\widehat{P} \cup \mathrm{CWA}(P, \beta \wedge \alpha)$ logically entails every $(\psi|\phi)[r,s] \in ground(P)$ that is inactive w.r.t. P and $\beta \wedge \alpha$. Towards a contradiction, assume that there exists a model Pr of $\widehat{P} \cup \mathrm{CWA}(P, \beta \wedge \alpha)$ and some inactive $(\psi|\phi)[r,s] \in ground(P)$ such that $Pr \not\models (\psi|\phi)[r,s]$. Hence, $Pr(\phi) > 0$, and thus ϕ is active. Hence, ψ is inactive, and thus $r > 0$. Thus, $\bot \Leftarrow \phi$ belongs to $\widehat{P}$. But this contradicts Pr being a model of $\widehat{P}$. Thus, $\widehat{P} \cup \mathrm{CWA}(P, \beta \wedge \alpha)$ logically entails every inactive $(\psi|\phi)[r,s] \in ground(P)$. Hence, every model of $\widehat{P} \cup \mathrm{CWA}(P, \beta \wedge \alpha)$ is also a model of $P \cup \mathrm{CWA}(P, \beta \wedge \alpha)$. In summary, $P \cup \mathrm{CWA}(P, \beta \wedge \alpha)$ is logically equivalent to $\widehat{P} \cup \mathrm{CWA}(P, \beta \wedge \alpha)$.

By Theorem 3.2, $P \,\|\!\!\sim^0 (\beta|\alpha)[l,u]$ iff $P \cup \mathrm{CWA}(P, \beta \wedge \alpha) \,\|\!\!\sim^0 (\beta|\alpha)[l,u]$. By the result above, the latter is equivalent to $\widehat{P} \cup \mathrm{CWA}(P, \beta \wedge \alpha) \,\|\!\!\sim^0 (\beta|\alpha)[l,u]$. As $\widehat{P}$ and $(\beta|\alpha)[l,u]$ contain only active $p \in HB_\Phi$, this is equivalent to $\widehat{P} \,\|\!\!\sim^0 (\beta|\alpha)[l,u]$. □

Proof of Theorem 3.5. Recall that $P \,\|\!\!\sim^{mc} (\beta|\alpha)[l,u]$ iff $P \cup \mathrm{CWA}(P, \beta \wedge \alpha) \,\|\!\!\sim^{me} (\beta|\alpha)[l,u]$. Since $\mathrm{CWA}(P, \beta \wedge \alpha) = \mathrm{CWA}(P \cup \mathrm{CWA}(P, \beta \wedge \alpha), \beta \wedge \alpha)$, as easily verified, the latter is equivalent to $P \cup \mathrm{CWA}(P, \beta \wedge \alpha) \cup \mathrm{CWA}(P \cup \mathrm{CWA}(P, \beta \wedge \alpha), \beta \wedge \alpha) \,\|\!\!\sim^{me} (\beta|\alpha)[l,u]$. That is, $P \cup \mathrm{CWA}(P, \beta \wedge \alpha) \,\|\!\!\sim^{mc} (\beta|\alpha)[l,u]$. □

Proof of Theorem 3.6. Recall that $P \,\|\!\!\sim^{mc} (\beta|\alpha)[l,u]$ iff $P \cup \mathrm{CWA}(P, \beta \wedge \alpha) \,\|\!\!\sim^{me} (\beta|\alpha)[l,u]$. By the proof of Theorem 3.3, $P \cup \mathrm{CWA}(P, \beta \wedge \alpha)$ is logically equivalent to $\widehat{P} \cup \mathrm{CWA}(P, \beta \wedge \alpha)$. It is then easy to verify that $\mathrm{CWA}(P, \beta \wedge \alpha) = \mathrm{CWA}(\widehat{P}, \beta \wedge \alpha)$. It thus follows that $P \cup \mathrm{CWA}(P, \beta \wedge \alpha) \,\|\!\!\sim^{me} (\beta|\alpha)[l,u]$ iff $\widehat{P} \cup \mathrm{CWA}(\widehat{P}, \beta \wedge \alpha) \,\|\!\!\sim^{me} (\beta|\alpha)[l,u]$. The latter is equivalent to $\widehat{P} \,\|\!\!\sim^{mc} (\beta|\alpha)[l,u]$ and, since $\mathrm{CWA}(P, \beta \wedge \alpha) = \mathrm{CWA}(\widehat{P}, \beta \wedge \alpha)$, also to $\widehat{P} \,\|\!\!\sim^{me} (\beta|\alpha)[l,u]$. □

B Appendix: Proofs for Section 5

We next prove Theorems 5.6–5.11. In the sequel, for ground probabilistic logic programs P, and ground events ϕ, we define $Mod_\phi(P)$ as follows:

$$Mod_\phi(P) \;\; = \;\; \{Pr_\phi \mid Pr \models P,\ Pr(\phi) > 0\}\,.$$

For the proof of Theorem 5.6 in the case $s = mc$, we need the following lemma, respectively, its (immediate) corollary.

Lemma B.1 *Let $\phi, \psi, \varepsilon, \varepsilon'$ be ground conjunctive events, and let P be a (fixed) ground conjunctive probabilistic logic program; set*

$$\overline{P_1} = \{\psi \Leftarrow \phi \mid (\psi|\phi)[1,1] \in P,\ P \,\|\!\!\not\sim^0 (\phi|\top)[0,0]\} \subseteq app(P)$$

If $P \cup \mathrm{CWA}(P, \varepsilon \wedge \varepsilon') \,\|\!\!\sim^0 (\varepsilon \wedge \neg\varepsilon'|\top)[0,0]$, then $P \,\|\!\!\sim^0 (\varepsilon|\top)[0,0]$, or $\overline{P_1} \cup \{\varepsilon\} \models \varepsilon'$.

Proof. Let $\varepsilon = p_1 \wedge \ldots \wedge p_n$, and let $s_1, \ldots, s_m$ be the other atoms which are entailed by $\overline{P_1} \cup \{\varepsilon\}$, i.e. $\overline{P_1} \cup \{\varepsilon\} \models p_1, \ldots, p_n, s_1, \ldots, s_m$. Set $I := \{p_1, \ldots, p_n, s_1, \ldots, s_m\} \in \mathcal{I}_\Phi$. In particular, $I \models \varepsilon$. Assume $I \not\models \varepsilon'$, hence $I \models \varepsilon \wedge \neg\varepsilon'$. By presupposition, $P \cup \mathrm{CWA}(P, \varepsilon \wedge \varepsilon') \models I[0,0]$. Since each s_i is entailed by

$\overline{P_1} \cup \{\varepsilon\}$, it is also entailed by $app(P) \cup \{\varepsilon \wedge \varepsilon'\}$. Hence $P \models I[0,0]$. Now, using the explicitness condition (3), there is $(\rho_2|\rho_1)[c,c] \in P$ such that either $c = 1$ and $I \models \rho_1 \wedge \neg\rho_2$, or $c = 0$ and $I \models \rho_1 \wedge \rho_2$. If $c = 1$, then $\rho_2 \Leftarrow \rho_1 \in \overline{P_1}$, and $I \models \rho_1 \wedge \neg\rho_2$ is a contradiction to the choice of $s_1, \ldots, s_m$. So $c = 0$ and $I \models \rho_1 \wedge \rho_2$, i.e. $\rho_1 \wedge \rho_2$ is a conjunction of some of the atoms $p_1, \ldots, p_n, s_1, \ldots, s_m$. But then, $P \mathrel{|\!\!\sim}^0 (\varepsilon|\top)[0,0]$, as can be seen as follows: Let $I' \in \mathcal{I}_\Phi$ with $I' \models \varepsilon$. If there is an $s_i, 1 \leq i \leq m$, such that $I' \not\models s_i$, then $P \models I'[0,0]$, according to the choice of $s_1, \ldots, s_m$. Otherwise, $I' \models s_1, \ldots, s_m$. Then also $I' \models \rho_1 \wedge \rho_2$, and hence $P \models I'[0,0]$. In either case, $P \models I'[0,0]$. □

Corollary B.2 *If $P \cup \mathrm{CWA}(P, \varepsilon \wedge \varepsilon') \mathrel{|\!\!\sim}^0 (\varepsilon \wedge \neg\varepsilon'|\top)[0,0]$, then $P \mathrel{|\!\!\sim}^0 (\varepsilon|\top)[0,0]$, or $app(P) \cup \{\varepsilon\} \models \varepsilon'$.*

Proof of Theorem 5.6. Let $\mathrel{|\!\!\sim}^s$ be one of $\mathrel{|\!\!\sim}^0$, $\mathrel{|\!\!\sim}^{me}$, $\mathrel{|\!\!\sim}^{mc}$, i.e. we consider the cases $s \in \{0, me, mc\}$ (except for RW, where we only consider $s \in \{0, me\}$.

RW. Let $s = 0$. If every model of $(\phi|\top)[l,u]$ satisfies $(\psi|\top)[l',u']$, and $(\phi|\top)[l,u]$ is true in all $Pr \in Mod_\varepsilon(P)$, then $(\psi|\top)[l',u']$ is true in all $Pr \in Mod_\varepsilon(P)$, too.
Let $s = me$. Let $Pr^{me} = me[P]$ be the me-model of P. Due to $P \mathrel{|\!\!\sim} (\phi|\varepsilon)[l,u]$, we have $Pr^{me} \models (\phi|\varepsilon)[l,u]$, that is, $Pr^{me}_\varepsilon \models (\phi|\top)[l,u]$. Then also $Pr^{me}_\varepsilon \models (\psi|\top)[l',u']$, since $(\phi|\varepsilon)[l,u] \Rightarrow (\psi|\top)[l',u']$ is logically valid. But this means that $Pr^{me} \models (\psi|\varepsilon)[l',u']$, and so $P \mathrel{|\!\!\sim}^{me} (\psi|\varepsilon)[l',u']$.

Ref. Clearly, $(\varepsilon|\varepsilon)[1,1]$ is true in all probabilistic interpretations Pr, in particular, in $me[\mathcal{F}]$ and in $me[\mathcal{F} \cup \mathrm{CWA}(\mathcal{F}, \varepsilon)]$. So, RW holds for each $s \in \{0, me, mc\}$.

LLE. If $\varepsilon \Leftrightarrow \varepsilon'$ is logically valid, then ε and ε' are propositionally equivalent. So, $Pr_\varepsilon = Pr_{\varepsilon'}$ for all probabilistic interpretations Pr. In particular, $Pr_\varepsilon \models (\phi|\top)[l,u]$ iff $Pr_{\varepsilon'} \models (\phi|\top)[l,u]$. Setting $Pr = me[P]$ and $Pr = me[P \cup \mathrm{CWA}(P, \phi \wedge \varepsilon)] = me[P \cup \mathrm{CWA}(P, \phi \wedge \varepsilon')]$, this proves the assertion for $s = me$ and $s = mc$, respectively. For $s = 0$, we have $Mod_\varepsilon(P) = Mod_{\varepsilon'}(P)$. Hence, $(\phi|\top)[l,u]$ is true in all $Pr \in Mod_\varepsilon(P)$ iff $(\phi|\top)[l,u]$ is true in all $Pr \in Mod_{\varepsilon'}(P)$.

Cut and *CM*. Let $s = 0$. If $(\varepsilon'|\top)[1,1]$ is true in all $Pr \in Mod_\varepsilon(P)$, then $Mod_{\varepsilon \wedge \varepsilon'}(P) = Mod_\varepsilon(P)$. Hence, $(\phi|\top)[l,u]$ is true in all $Pr \in Mod_{\varepsilon \wedge \varepsilon'}(P)$ iff $(\phi|\top)[l,u]$ is true in all $Pr \in Mod_\varepsilon(P)$.
Let $s = me$, and let $Pr^{me} = me[P]$. Suppose $P \mathrel{|\!\!\sim}^{me} (\varepsilon'|\varepsilon)[1,1]$, that means, $Pr^{me} \models (\varepsilon'|\varepsilon)[1,1]$. Then for all events α, $Pr^{me}(\alpha \wedge \varepsilon) = Pr^{me}(\alpha \wedge \varepsilon \wedge \varepsilon')$, due to $Pr^{me}(\varepsilon \wedge \neg\varepsilon') = 0$. Therefore, $Pr^{me} \models (\phi|\varepsilon \wedge \varepsilon')[l,u]$ iff $Pr^{me} \models (\phi|\varepsilon)[l,u]$.
Let $s = mc$. Here $P \mathrel{|\!\!\sim}^{mc} (\varepsilon'|\varepsilon)[1,1]$ means $me[P \cup \mathrm{CWA}(P, \varepsilon \wedge \varepsilon')] \models \varepsilon \wedge \neg\varepsilon'[0,0]$, and hence, due to the *open mindedness principle* (cf. [59]), $P \cup \mathrm{CWA}(P, \varepsilon \wedge \varepsilon') \mathrel{|\!\!\sim}^0 \varepsilon \wedge \neg\varepsilon'[0,0]$. By Corollary B.2, $P \mathrel{|\!\!\sim}^0 (\varepsilon|\top)[0,0]$, or $app(P) \cup \{\varepsilon\} \models \varepsilon'$. If $P \mathrel{|\!\!\sim}^0 (\varepsilon|\top)[0,0]$, then both $me[P \cup \mathrm{CWA}(P, \psi \wedge \varepsilon)]$ and $me[P \cup \mathrm{CWA}(P, \psi \wedge \varepsilon \wedge \varepsilon')]$ satisfy $(\varepsilon|\top)[0,0]$, and therefore, $P \mathrel{|\!\!\sim} (\phi|\varepsilon \wedge \varepsilon')[l,u]$ iff $P \mathrel{|\!\!\sim} (\phi|\varepsilon)[l,u]$. Otherwise, we have $app(P) \cup \{\varepsilon\} \models \varepsilon'$, and consequently, $\mathrm{CWA}(P, \psi \wedge \varepsilon \wedge \varepsilon') = \mathrm{CWA}(P, \psi \wedge \varepsilon)$, which implies $me[P \cup \mathrm{CWA}(P, \psi \wedge \varepsilon \wedge \varepsilon')] = me[P \cup \mathrm{CWA}(P, \psi \wedge \varepsilon)] =: Pr^*$. Moreover, $Pr^*(\varepsilon \wedge \neg\varepsilon') = 0$, so that $Pr^*(\psi|\varepsilon \wedge \varepsilon') = Pr^*(\psi|\varepsilon)$. This shows $P \mathrel{|\!\!\sim} (\phi|\varepsilon \wedge \varepsilon')[l,u]$ iff $P \mathrel{|\!\!\sim} (\phi|\varepsilon)[l,u]$. Therefore, *Cut* and *Cautious Monotonicity* hold for $s = mc$.

Or. Let $s = 0$. Assume that $(\phi|\top)[1,1]$ is true in all $Pr \in Mod_\varepsilon(P) \cup Mod_{\varepsilon'}(P)$. Hence, ϕ is true in all $I \in \mathcal{I}_\Phi$ such that $I \models \varepsilon \vee \varepsilon'$ and $Pr(I) > 0$ for some model Pr of P. Thus, $(\phi|\top)[1,1]$ is true in all $Pr \in Mod_{\varepsilon \vee \varepsilon'}(P)$.
Let $s = me$, and let $Pr^{me} = me[P]$. Assume $P \mathrel{|\!\!\sim} (\phi|\varepsilon)[1,1]$ and $P \mathrel{|\!\!\sim} (\phi|\varepsilon')[1,1]$, which means $Pr^{me} \models (\phi|\varepsilon)[1,1]$ and $Pr^{me} \models (\phi|\varepsilon')$. This implies $Pr^{me}(\varepsilon \wedge \neg\phi) = Pr^{me}(\varepsilon' \wedge \neg\phi) = 0$, and hence, $Pr^{me}((\varepsilon \vee \varepsilon') \wedge \neg\phi) = 0$. Therefore, $Pr^{me}((\varepsilon \vee \varepsilon') \wedge \phi) = Pr^{me}(\varepsilon \vee \varepsilon')$, i.e. $Pr^{me} \models (\phi|\varepsilon \vee \varepsilon')[1,1]$. This shows $P \mathrel{|\!\!\sim} (\phi|\varepsilon \vee \varepsilon')[1,1]$. □

Proof of Theorem 5.7. In the framework of me- and mc-entailment, $P \mathrel{|\!\not\sim} \neg(\varepsilon'|\varepsilon)[1,1]$ is equivalent to $P \mathrel{|\!\sim} (\varepsilon'|\varepsilon)[1,1]$, so Rational Monotonicity (RM) here is equivalent to Cautious Monotonicity (CM). The assertion now follows with Theorem 5.6. □

Proof of Theorem 5.8. First, we consider the case of me-entailment. Set $Pr^{me} = me[P]$. Then $P \mathrel{|\!\sim}^{me} (\phi|\varepsilon)[l,u]$ means $Pr^{me} \models (\phi|\varepsilon)[l,u]$. Let $HB_\Phi = HB_1 \dot{\cup} HB_2$ with HB_2 containing the atoms occurring in ε'. Then all atoms occurring in $ground(P), \varepsilon$ and ϕ will be in HB_1. We will show $Pr^{me}(\phi|\varepsilon) = Pr^{me}(\phi|\varepsilon \wedge \varepsilon')$, which implies $P \mathrel{|\!\sim}^{me} (\phi|\varepsilon \wedge \varepsilon')[l,u]$.

Pr^{me} fulfills some of the conditional constraints in P with equality (i.e. $Pr^{me}(\beta_i|\alpha_i) = l_i$, or $Pr^{me}(\beta_i|\alpha_i) = u_i$ for some $(\beta_i|\alpha_i)[l_i,u_i] \in P$), and the other constraints with inequality (i.e. $Pr^{me}(\beta_i|\alpha_i) \in (l_i,u_i)$ for the other $(\beta_i|\alpha_i)[l_i,u_i] \in P$) (see, e.g., [77]). Only the conditionals of the first type are essential for computing Pr^{me}; we assume that these are exactly $(\psi_1|\phi_1)[l_1,u_1], \ldots, (\psi_m|\phi_m)[l_m,u_m] \in P$. Then for $I \in \mathcal{I}_\Phi$, Pr^{me} can be written as

$$Pr^{me}(I) = \alpha_0 \prod_{\substack{1 \le i \le m \\ I \models \phi_i \wedge \psi_i}} \alpha_i^+ \prod_{\substack{1 \le i \le m \\ I \models \phi_i \wedge \neg\psi_i}} \alpha_i^-$$

with suitable non-negative factors $\alpha_0, \alpha_1^+, \alpha_1^-, \ldots, \alpha_m^+, \alpha_m^-$. Let ψ be an event no atom of which occurs in ε', that is, all atoms occurring in ψ lie within HB_2, as do all atoms occurring in $\phi_1, \psi_1, \ldots, \phi_m, \psi_m$. Writing each $I \in \mathcal{I}_\Phi$ in the form $I = I_1 \cup I_2$ with $I_1 \subseteq HB_1, I_2 \subseteq HB_2$, we see that $I \models \psi \wedge \varepsilon'$ iff $I_1 \models \psi$ and $I_2 \models \varepsilon'$, $I \models \phi_i \wedge \psi_i$ ($I \models \phi_i \wedge \neg\psi_i$, respectively) iff $I_1 \models \phi_i \wedge \psi_i$ ($I_1 \models \phi_i \wedge \neg\psi_i$, respectively). So we obtain

$$\begin{aligned}
Pr^{me}(\psi \wedge \varepsilon') &= \alpha_0 \sum_{\substack{I \in \mathcal{I}_\Phi \\ I \models \psi \wedge \varepsilon'}} \prod_{\substack{1 \le i \le m \\ I \models \phi_i \wedge \psi_i}} \alpha_i^+ \prod_{\substack{1 \le i \le m \\ I \models \phi_i \wedge \neg\psi_i}} \alpha_i^- \\
&= \alpha_0 \sum_{\substack{I_1 \subseteq HB_1 : I_1 \models \psi \\ I_2 \subseteq HB_2 : I_2 \models \varepsilon'}} \prod_{\substack{1 \le i \le m \\ I_1 \models \phi_i \wedge \psi_i}} \alpha_i^+ \prod_{\substack{1 \le i \le m \\ I_1 \models \phi_i \wedge \neg\psi_i}} \alpha_i^- \\
&= card(\{I_2 \subseteq HB_2 \mid I_2 \models \varepsilon'\})\ \alpha_0 \sum_{\substack{I_1 \subseteq HB_1 \\ I_1 \models \psi}} \prod_{\substack{1 \le i \le m \\ I_1 \models \phi_i \wedge \psi_i}} \alpha_i^+ \prod_{\substack{1 \le i \le m \\ I_1 \models \phi_i \wedge \neg\psi_i}} \alpha_i^-
\end{aligned}$$

Similarly,

$$Pr^{me}(\psi) = 2^{card(HB_2)}\ \alpha_0 \sum_{I_1 \subseteq HB_1 : I_1 \models \psi} \prod_{\substack{1 \le i \le m \\ I_1 \models \phi_i \wedge \psi_i}} \alpha_i^+ \prod_{\substack{1 \le i \le m \\ I_1 \models \phi_i \wedge \neg\psi_i}} \alpha_i^-$$

Therefore, $Pr^{me}(\psi \wedge \varepsilon') \propto Pr^{me}(\psi)$ for each event ψ which has no atom in common with ε'. In particular, by presupposition, ϕ and ε are such events. This implies

$$Pr^{me}(\phi|\varepsilon \wedge \varepsilon') = \frac{Pr^{me}(\phi \wedge \varepsilon \wedge \varepsilon')}{Pr^{me}(\varepsilon \wedge \varepsilon')} = \frac{Pr^{me}(\phi \wedge \varepsilon)}{Pr^{me}(\varepsilon)} = Pr^{me}(\phi|\varepsilon)$$

which was to be shown.

Now we deal with mc-entailment. $P \mathrel{|\!\sim}^{mc} (\phi|\varepsilon)[l,u]$ here means $me[P \cup \mathrm{CWA}(P, \phi \wedge \varepsilon)] \models (\phi|\varepsilon)[l,u]$. It is to be proved that $P \mathrel{|\!\sim}^{mc} (\phi|\varepsilon \wedge \varepsilon')[l,u]$, i.e. that $me[P \cup \mathrm{CWA}(P, \phi \wedge \varepsilon \wedge \varepsilon')] \models (\phi|\varepsilon \wedge \varepsilon')[l,u]$. Set

$$Pr_1^* = me[P \cup \mathrm{CWA}(P, \phi \wedge \varepsilon \wedge \varepsilon')], \quad \text{and} \quad Pr_2^* = me[P \cup \mathrm{CWA}(P, \phi \wedge \varepsilon)].$$

Moreover, let

$$\begin{aligned} HB_2 &= \{p \in HB_\Phi \mid app(P) \cup \{\phi \wedge \varepsilon\} \models p\}, \\ HB_3 &= \{p \in HB_\Phi \mid app(P) \cup \{\phi \wedge \varepsilon \wedge \varepsilon'\} \models p\}, \\ HB_4 &= \{p \in HB_\Phi \mid \varepsilon' \models p\}. \end{aligned}$$

By presupposition, no atom occurring in P, ε and ϕ also occurs in ε'. So we have $HB_2 \cap HB_4 = \emptyset$, and $HB_3 = HB_2 \cup HB_4$.

Due to the fact, that no atom of ε' occurs in P, a conditional constraint in P is essential for calculating Pr_1^* iff it is essential for calculating Pr_2^*. Let $(\psi_1|\phi_1)[l_1, u_1], \ldots, (\psi_m|\phi_m)[l_m, u_m] \in P$ be the conditional constraints in P which are essential for computing Pr_1^* and Pr_2^*. Then the distributions can be written as follows (cf. [29, 31]):

$$Pr_1^*(I) = \begin{cases} \alpha_0 \prod\limits_{\substack{1 \le i \le m \\ I \models \phi_i \wedge \psi_i}} \alpha_i^+ \prod\limits_{\substack{1 \le i \le m \\ I \models \phi_i \wedge \neg\psi_i}} \alpha_i^-, & \text{if } I \subseteq HB_3 \\ 0, & \text{otherwise} \end{cases}$$

where $\alpha_i^+ = \alpha_i^{1-x_i}$, $\alpha_i^- = \alpha_i^{-x_i}$, $x_i \in \{l_i, u_i\}$, and the α_i's, $1 \le i \le m$, being solutions to the equations

$$\alpha_i = \frac{x_i}{1-x_i} \frac{\sum_{\substack{I \subseteq HB_3 \\ I \models \phi \wedge \neg\psi}} \prod_{\substack{j \ne i \\ I \models \phi_j \wedge \psi_j}} \alpha_j^+ \prod_{\substack{j \ne i \\ I \models \phi_j \wedge \neg\psi_j}} \alpha_j^-}{\sum_{\substack{I \subseteq HB_3 \\ I \models \phi \wedge \psi}} \prod_{\substack{j \ne i \\ I \models \phi_j \wedge \psi_j}} \alpha_j^+ \prod_{\substack{j \ne i \\ I \models \phi_j \wedge \neg\psi_j}} \alpha_j^-}; \tag{9}$$

analogously,

$$Pr_2^*(I) = \begin{cases} \beta_0 \prod\limits_{\substack{1 \le i \le m \\ I \models \phi_i \wedge \psi_i}} \beta_i^+ \prod\limits_{\substack{1 \le i \le m \\ I \models \phi_i \wedge \neg\psi_i}} \beta_i^-, & \text{if } I \subseteq HB_2 \\ 0, & \text{otherwise} \end{cases}$$

where $\beta_i^+ = \beta_i^{1-x_i}$, $\beta_i^- = \beta_i^{-x_i}$, $x_i \in \{l_i, u_i\}$, and the β_i's, $1 \le i \le m$, being solutions to the equations

$$\beta_i = \frac{x_i}{1-x_i} \frac{\sum_{\substack{I \subseteq HB_2 \\ I \models \phi \wedge \neg\psi}} \prod_{\substack{j \ne i \\ I \models \phi_j \wedge \psi_j}} \beta_j^+ \prod_{\substack{j \ne i \\ I \models \phi_j \wedge \neg\psi_j}} \beta_j^-}{\sum_{\substack{I \subseteq HB_2 \\ I \models \phi \wedge \psi}} \prod_{\substack{j \ne i \\ I \models \phi_j \wedge \psi_j}} \beta_j^+ \prod_{\substack{j \ne i \\ I \models \phi_j \wedge \neg\psi_j}} \beta_j^-}; \tag{10}$$

α_0 and β_0 are obtained as normalizing constants.

First, we show that the two equational systems (9) and (10) are equivalent. For $I_1, I_2 \subseteq HB_3$, we define a relation $\sim$ by $I_1 \sim I_2$ iff $I_1 \backslash HB_4 = I_2 \backslash HB_4$. Clearly, $\sim$ is an equivalence relation, partitioning HB_3 in equivalence classes, with exactly one $J \subseteq HB_2$ in each equivalence class. For each such $J \subseteq HB_2$, and for each $I \subseteq HB_3$ with $I \sim J$, we have $J \models \phi_j \wedge \dot{\psi}_j$ iff $I \models \phi_j \wedge \dot{\psi}_j$, with $\dot{\psi}_j \in \{\psi_j, \neg\psi_j\}$, for all $1 \le j \le m$. This implies

$$\begin{aligned} &\sum_{\substack{I \subseteq HB_3 \\ I \models \phi \wedge \dot{\psi}}} \prod_{\substack{j \ne i \\ I \models \phi_j \wedge \psi_j}} \alpha_j^+ \prod_{\substack{j \ne i \\ I \models \phi_j \wedge \neg\psi_j}} \alpha_j^- \\ &\quad = \sum_{\substack{I \subseteq HB_2 \\ I \models \phi \wedge \dot{\psi}}} \sum_{J \sim I} \prod_{\substack{j \ne i \\ I \models \phi_j \wedge \psi_j}} \alpha_j^+ \prod_{\substack{j \ne i \\ I \models \phi_j \wedge \neg\psi_j}} \alpha_j^- \\ &\quad = 2^{card(HB_4)} \sum_{\substack{I \subseteq HB_2 \\ I \models \phi \wedge \dot{\psi}}} \prod_{\substack{j \ne i \\ I \models \phi_j \wedge \psi_j}} \alpha_j^+ \prod_{\substack{j \ne i \\ I \models \phi_j \wedge \neg\psi_j}} \alpha_j^- \end{aligned}$$

Thus, the factor $2^{card(HB_4)}$ is canceled in (9), which shows that in fact, (9) and (10) are equivalent. Since the me-distributions do not depend upon a particular solution to (9) and (10), respectively (see [29, 31]), we may choose $\alpha_i = \beta_i$ for $1 \leq i \leq m$; note, however, that the normalizing constants α_0 and β_0 may differ.

Observing that each $I \subseteq HB_3$ can be written as $I = I_1 \cup I_2$ with $I_1 \subseteq HB_2$ and $I_2 \subseteq HB_4$, and that $I \models \phi_i \wedge \dot{\psi}_i$ iff $I_1 \models \phi_i \wedge \dot{\psi}_i$, and $I \models \phi \wedge \varepsilon \wedge \varepsilon'$ iff $I_1 \models \phi \wedge \varepsilon$ and $I_2 \models \varepsilon'$ (i.e. $I_2 = HB_4$), we obtain

$$\begin{aligned} Pr_1^*(\phi \wedge \varepsilon \wedge \varepsilon') &= \sum_{\substack{I \subseteq HB_3 \\ I \models \phi \wedge \varepsilon \wedge \varepsilon'}} \alpha_0 \prod_{\substack{1 \leq i \leq m \\ I \models \phi_i \wedge \psi_i}} \alpha_i^+ \prod_{\substack{1 \leq i \leq m \\ I \models \phi_i \wedge \neg \psi_i}} \alpha_i^- \\ &= \sum_{\substack{I_1 \subseteq HB_2 \\ I \models \phi \wedge \varepsilon}} \alpha_0 \prod_{\substack{1 \leq i \leq m \\ I \models \phi_i \wedge \psi_i}} \alpha_i^+ \prod_{\substack{1 \leq i \leq m \\ I \models \phi_i \wedge \neg \psi_i}} \alpha_i^- \\ &= \alpha_0 \beta_0^{-1} Pr_2^*(\phi \wedge \varepsilon). \end{aligned}$$

In the same way, we calculate $Pr_1^*(\varepsilon \wedge \varepsilon') = \alpha_0 \beta_0^{-1} Pr_2^*(\varepsilon)$. This proves $Pr_1^*(\phi|\varepsilon \wedge \varepsilon') = Pr_2^*(\phi|\varepsilon)$. □

Proof of Theorem 5.9. Set $P^+ := P \cup \{(\beta|\alpha)[r,s]\}$. We first consider the case of me-entailment. Here, $P \mathrel{|\!\!\sim}^{me} (\psi|\phi)[l,u]$ means $me[P] \models (\psi|\phi)[l,u]$. We may now argue similarly as in the proof of Theorem 5.8, or we may use the property of *system independence* [76] stating that learning conditional constraints with disjoint sets of atoms yields statistical independence of these sets of atoms in the resulting me-distribution. Hence, in the present case, $me[P^+] = me[P] \cdot me[\{(\beta|\alpha)[r,s]\}]$, i.e., $P^* = P_1^* \cdot P_2^*$ with $P^* := me[P^+], P_1^* := me[P]$, and $P_2^* := me[\{(\beta|\alpha)[r,s]\}]$, all distributions on the obviously corresponding sets of atoms. This shows at once $P^*(\psi|\phi) = P_1^*(\psi|\phi)$, and $P^*(\psi|\phi \wedge \alpha) = P_1^*(\psi|\phi)$. Therefore, $P \cup \{(\beta|\alpha)[r,s]\} \mathrel{|\!\!\sim} (\psi|\phi)[l,u]$ and $P \cup \{(\beta|\alpha)[r,s]\} \mathrel{|\!\!\sim} (\psi|\phi \wedge \alpha)[l,u]$, as desired.

Let us now consider the case of mc-entailment, that is, we have to take classical approximations and the closed world assumptions into account. $P \mathrel{|\!\!\sim}^{mc} (\psi|\phi)[l,u]$ means $me[P \cup \text{CWA}(P, \phi \wedge \psi)] \models (\psi|\phi)[l,u]$. Since no atom of α, β also occurs in $ground(P), \phi, \psi$, we have $\text{CWA}(P^+, \phi \wedge \psi) = \text{CWA}(P, \phi \wedge \psi)$ which makes the new constraint $(\beta|\alpha)[x,y]$ vacuous for me-propagation, hence $me[P^+ \cup \text{CWA}(P^+, \phi \wedge \psi)] = me[P \cup \text{CWA}(P, \phi \wedge \psi)]$. This shows $P \cup \{(\beta|\alpha)[r,s]\} \mathrel{|\!\!\sim}^{mc} (\psi|\phi)[l,u]$. In order to prove the second statement, we first note that $\text{CWA}(P, \phi \wedge \psi) = \text{CWA}(P^+, \phi \wedge \psi \wedge \alpha) \cup \{\bot \Leftarrow p \mid \alpha \wedge \beta \models p\}$, where the union is disjoint. By *system independence* [76], $me[P \cup \text{CWA}(P, \phi \wedge \psi)] = me[P \cup \text{CWA}(P^+, \phi \wedge \psi \wedge \alpha)] \cdot me[\{\bot \Leftarrow p \mid \alpha \wedge \beta \models p\}]$. Set $P_1^* := me[P \cup \text{CWA}(P^+, \phi \wedge \psi \wedge \alpha)]$, taken as a distribution on all atoms except those occurring in α, β. So, $me[P \cup \text{CWA}(P, \phi \wedge \psi)] \models (\psi|\phi)[l,u]$ implies $P_1^* \models (\psi|\phi)[l,u]$. Again, by system independence, we further obtain $P^* := me[P^+ \cup \text{CWA}(P^+, \phi \wedge \psi \wedge \alpha)] = me[P \cup \text{CWA}(P^+, \phi \wedge \psi \wedge \alpha) \cup \{(\beta|\alpha)[r,s]\}] = me[P \cup \text{CWA}(P^+, \phi \wedge \psi \wedge \alpha)] \cdot me[\{(\beta|\alpha)[r,s]\}] = P_1^* \cdot me[\{(\beta|\alpha)[r,s]\}]$, hence $P^*(\psi|\phi \wedge \alpha) = P_1^*(\psi|\phi) \in [l,u]$. This proves that $P \cup \{(\beta|\alpha)[r,s]\} \mathrel{|\!\!\sim}^{mc} (\psi|\phi \wedge \alpha)[l,u]$. □

Proof of Lemma 5.10. Let $(\psi|\phi)[l,u] \in ground(P)$. Then, by *Inclusion*, $P \mathrel{|\!\!\sim} (\psi|\phi)[l,u]$. Now, if $\varepsilon \Leftrightarrow \phi$ is logically valid, then *LLE* implies $P \mathrel{|\!\!\sim} (\psi|\varepsilon)[l,u]$. □

Proof of Theorem 5.11. Since all notions of entailment considered here satisfy *LLE* (cf. Theorem 5.6), we only have to show that they also satisfy *Inc*. But this is obvious for $\mathrel{|\!\!\sim}^0$, $\mathrel{|\!\!\sim}^{me}$, and $\mathrel{|\!\!\sim}^{mc}$. □

Proof of Theorem 5.12. Let P be a (conjunctive) probabilistic logic program, and let $(\psi|\phi)[l,u]$ be a ground (conjunctive) conditional constraint. Assume $P \mathrel{|\!\!\sim}^0 (\psi|\phi)[l,u]$, that is, every model of P satisfies $(\psi|\phi)[l,u]$. Hence, as $me[P]$ and $me[P \cup \text{CWA}(P, \phi \wedge \psi)]$ are models of P, they also satisfy $(\psi|\phi)[l,u]$. That is, $P \mathrel{|\!\!\sim}^s (\psi|\phi)[l,u]$ for every $s \in \{me, mc\}$. □

Proof of Theorem 5.13. (a) Assume first that $s = me$. Assume now that $P \,\|\!\!\sim^0 (\psi|\phi)[c,c]$. That is, every model of P satisfies $(\psi|\phi)[c,c]$. Thus, in particular, $me[P]$ satisfies $(\psi|\phi)[c,c]$. That is, $P \,\|\!\!\sim^{me} (\psi|\phi)[c,c]$. Conversely, as shown by Paris and Vencovská [59], the maximum entropy inference process satisfies the *open-mindedness principle.* Applied to the framework of this paper, this principle says that $P \,\|\!\!\not\sim^0 (\psi|\phi)[c,c]$ implies $me[P] \not\models (\psi|\phi)[c,c]$, for all probabilistic logic programs P, and all ground classical conditional constraints $(\psi|\phi)[c,c]$. This shows that $P \,\|\!\!\sim^{me} (\psi|\phi)[c,c]$ implies $P \,\|\!\!\sim^0 (\psi|\phi)[c,c]$.

Assume next that $s = mc$. By Theorem 3.2, $P \,\|\!\!\sim^0 (\psi|\phi)[c,c]$ iff $P \cup \mathrm{CWA}(P, \psi \wedge \phi) \,\|\!\!\sim^0 (\psi|\phi)[c,c]$. As shown above, the latter is equivalent to $P \cup \mathrm{CWA}(P, \psi \wedge \phi) \,\|\!\!\sim^{me} (\psi|\phi)[c,c]$, that is, $P \,\|\!\!\sim^{mc} (\psi|\phi)[c,c]$.

(b) Immediate by (a). □

Proof of Theorem 5.14. (a) It is easy to verify that every model Pr of P satisfies F iff every model $I \in \mathcal{I}_\Phi$ of $\gamma(P)$ satisfies $\gamma(F)$. That is, $P \,\|\!\!\sim^0 F$ iff $\gamma(P) \models \gamma(F)$.

Assume now $s \in \{mc, me\}$. By Theorem 5.13, it holds that $P \,\|\!\!\sim^s F$ iff $P \,\|\!\!\sim^0 F$. As $P \,\|\!\!\sim^0 F$ iff $\gamma(P) \models \gamma(F)$, it thus follows that $P \,\|\!\!\sim^s F$ iff $\gamma(P) \models \gamma(F)$.

(b) For all $s \in \{0, me, mc\}$, it holds $P \,\|\!\!\sim^s_{tight} F$ iff $P \,\|\!\!\sim^s F$ and $P \,\|\!\!\not\sim^s (\phi|\top)[0,0]$. Thus, by (a), $P \,\|\!\!\sim^s_{tight} F$ is equivalent to $\gamma(P) \models \gamma(F)$ and $\gamma(P) \not\models \neg\phi$. □

C Appendix: Proofs for Section 7

Proof of Proposition 7.4. Observe first that the sets C and $[\![D]\!]$ can be computed in linear time in the size of P. Algorithm index_set_2 from [45] reduces the computation of $R_C([\![D]\!])$ to $O(|D|\,|R_C([\![D]\!])|)$ satisfiability tests on classical logic programs of size up to $\|P\|$. Since each satisfiability test can be done in time $O(\|P\|)$, it follows that $R_C([\![D]\!])$ can be computed in time $O(|D|\,\|P\|\,|R_C([\![D]\!])|)$. □

Proof of Lemma 7.7. Due to Equations (1) and (2), for all $I \in \mathcal{I}_\Phi$ with $I \models C$, the following holds:

$$me[P](I) \;=\; \alpha_0 \prod_{\substack{(\psi|\phi)[l,u]\in D\\ I\models\psi\wedge\phi}} \alpha^+_{\psi|\phi} \prod_{\substack{(\psi|\phi)[l,u]\in D\\ I\models\neg\psi\wedge\phi}} \alpha^-_{\psi|\phi}\,. \tag{11}$$

Assume now that $I_1, I_2 \in \mathcal{I}_\Phi$ with $I_1, I_2 \models C \cup \{\wedge r\}$ for some $r \in R$. Thus, for all $\psi|\phi \in E$, it holds that (i) $I_1 \models \psi \wedge \phi$ iff $I_2 \models \psi \wedge \phi$, (ii) $I_1 \models \neg\psi \wedge \phi$ iff $I_2 \models \neg\psi \wedge \phi$, and (iii) $I_1 \models \neg\phi$ iff $I_2 \models \neg\phi$. By (11), it then follows $me[P](I_1) = me[P](I_2)$. □

Proof of Theorem 7.8. Recall first that $me[P]$ is given by the optimal solution $x^\star_I$ ($I \in \mathcal{I}_\Phi$) of the following optimization problem (12) over the variables x_I ($I \in \mathcal{I}_\Phi$):

$$\begin{aligned}
&\text{maximize} \quad -\sum_{I\in\mathcal{I}_\Phi} x_I \log x_I\\
&\text{subject to}\\
&\sum_{I\in\mathcal{I}_\Phi} x_I \;=\; 1\\
&\sum_{I\in\mathcal{I}_\Phi,\, I\models\neg\psi\wedge\phi} -l\, x_I + \sum_{I\in\mathcal{I}_\Phi,\, I\models\psi\wedge\phi} (1-l)\, x_I \;\geq\; 0 \quad (\text{for all } (\psi|\phi)[l,u] \in \mathit{ground}(P),\ l > 0)\\
&\sum_{I\in\mathcal{I}_\Phi,\, I\models\neg\psi\wedge\phi} u\, x_I + \sum_{I\in\mathcal{I}_\Phi,\, I\models\psi\wedge\phi} (u-1)\, x_I \;\geq\; 0 \quad (\text{for all } (\psi|\phi)[l,u] \in \mathit{ground}(P),\ u < 1)\\
&x_I \;\geq\; 0 \quad (\text{for all } I \in \mathcal{I}_\Phi)
\end{aligned} \tag{12}$$

By Lemmas 7.1 and 7.2, it is now sufficient to show that for every $r \in R$, it holds that $me[P](\wedge r) = y_r^\star$, where $y_r^\star$ $(r \in R)$ is the optimal solution of the following optimization problem (13) over y_r $(r \in R)$:

$$
\begin{aligned}
&\text{maximize} \quad -\sum_{r \in R} y_r (\log y_r - \log a_r) \\
&\text{subject to} \\
&\qquad \sum_{r \in R} y_r \;=\; 1 \\
&\sum_{r \in R,\, r \models \neg\psi\wedge\phi} -l\, y_r + \sum_{r \in R,\, r \models \psi\wedge\phi} (1-l)\, y_r \;\geq\; 0 \quad (\text{for all } (\psi|\phi)[l,u] \in D,\ l > 0) \\
&\sum_{r \in R,\, r \models \neg\psi\wedge\phi} u\, y_r + \sum_{r \in R,\, r \models \psi\wedge\phi} (u-1)\, y_r \;\geq\; 0 \quad (\text{for all } (\psi|\phi)[l,u] \in D,\ u < 1) \\
&\qquad y_r \;\geq\; 0 \quad (\text{for all } r \in R)
\end{aligned}
\tag{13}
$$

We first show that $(*)$ for every solution x_I $(I \in \mathcal{I}_\Phi)$ of (12) such that

$$x_{I_1} \;=\; x_{I_2} \text{ for all } I_1, I_2 \in \mathcal{I}_\Phi \text{ with } I_1, I_2 \models C \cup \{\wedge r\} \text{ for some } r \in R\,, \tag{14}$$

there exists a solution y_r $(r \in R)$ of (13) such that

$$\sum_{I \in \mathcal{I}_\Phi} x_I \log x_I \;=\; \sum_{r \in R} y_r (\log y_r - \log a_r)\,. \tag{15}$$

Let x_I $(I \in \mathcal{I}_\Phi)$ be a solution of (12) that satisfies (14). For every $r \in R$, we then define $y_r = a_r\, x_I$, where $I \in \mathcal{I}_\Phi$ such that $I \models C \cup \{\wedge r\}$. Then, y_r $(r \in R)$ is a solution of (13) that satisfies (15).

We next prove that $(**)$ for every solution y_r $(r \in R)$ of (13), there exists a solution x_I $(I \in \mathcal{I}_\Phi)$ of (12) that satisfies (14) and (15). Let y_r $(r \in R)$ be a solution of (13). We define $x_I = 0$ for all $I \in \mathcal{I}_\Phi$ such that $I \not\models C$, and $x_I = y_r / a_r$ for all $I \in \mathcal{I}_\Phi$ and $r \in R$ such that $I \models C \cup \{\wedge r\}$. Then, x_I $(I \in \mathcal{I}_\Phi)$ is a solution of (12) that satisfies (14) and (15).

By $(*)$, $(**)$, and Lemma 7.7, the fact that (12) has a unique optimal solution $x_I^\star$ $(I \in \mathcal{I}_\Phi)$ implies that also (13) has a unique optimal solution $y_r^\star$ $(r \in R)$. Moreover, $x_I^\star = y_r^\star / a_r$ for all $I \in \mathcal{I}_\Phi$ and $r \in R$ with $I \models C \cup \{\wedge r\}$. Thus, for all $r \in R$:

$$me[P](\wedge r) \;=\; \sum_{I \in \mathcal{I}_\Phi,\, I \models \wedge r} x_I^\star \;=\; \sum_{I \in \mathcal{I}_\Phi,\, I \models C \cup \{\wedge r\}} x_I^\star \;=\; \sum_{I \in \mathcal{I}_\Phi,\, I \models C \cup \{\wedge r\}} y_r^\star / a_r \;=\; y_r^\star\,. \;\Box$$

D Appendix: Proofs for Section 8

Proof of Theorem 8.1. Observe first that for all $i \in \{0, \ldots, n-1\}$, where $n \geq 0$ such that $triv_C^\star(D) = triv_C^n(D)$, it holds that every model of $P \cup triv_C^i(D)$ is also a model of $P \cup triv_C^{i+1}(D)$. Hence, every model of P is also a model of $P^\star = P \cup triv_C^\star(D)$. Thus, P and $P^\star$ have exactly the same sets of models. This shows that P has a model Pr with $Pr(\alpha) > 0$ iff $P^\star$ has a model Pr with $Pr(\alpha) > 0$. $\Box$

Proof of Theorem 8.3. As argued in the proof of Theorem 8.1, P and $P^\star$ have the same sets of models. This already proves that for every $s \in \{0, me\}$, it holds that $P \Vdash\!\sim^{s}_{tight} (\beta|\alpha)[l,u]$ iff $P^\star \Vdash\!\sim^{s}_{tight} (\beta|\alpha)[l,u]$. Recall then that $P \Vdash\!\sim^{mc} (\beta|\alpha)[l,u]$ iff $P \cup \text{CWA}(P, \beta\wedge\alpha) \Vdash\!\sim^{me}_{tight} (\beta|\alpha)[l,u]$. Since $\text{CWA}(P, \beta\wedge\alpha) = \text{CWA}(P^\star, \beta\wedge\alpha)$, as easily seen, it also follows that $P \Vdash\!\sim^{mc}_{tight} (\beta|\alpha)[l,u]$ iff $P^\star \Vdash\!\sim^{mc}_{tight} (\beta|\alpha)[l,u]$. $\Box$

Proof of Proposition 8.4. Observe that $n \leq |D|$ for the number $n \geq 0$ such that $triv_C^\star(D) = triv_C^n(D)$. Thus, $triv_C^\star(D)$ is computable by an iteration of at most $|D| + 1$ steps. In iteration step i, we first compute $triv_C^i(D)$, which is the set of all $(\phi|\top)[0,0]$ such that either (i), or (ii), or (iii) holds relative to $C \cup triv_C^{i-1}(D)$, and then check whether $triv_C^i(D) = triv_C^{i-1}(D)$. In the conjunctive case, the former can be done in time $O(\|P\||D|^2)$, while the latter is possible in time $O(|D|)$. If $triv_C^i(D) = triv_C^{i-1}(D)$, then the iteration stops and $triv_C^\star(D) = triv_C^i(D)$. In summary, $triv_C^\star(D)$ can be computed in time $O(\|P\||D|^3)$. □

Proof of Theorem 8.5. Observe that each model of C is also a model of $vac_C(D)$. Hence, each model of $P^\star = C \cup (D - vac_C(D))$ is also a model of $vac_C(D)$ and thus of P. Thus, P and $P^\star$ have exactly the same sets of models. Hence, P has a model Pr with $Pr(\alpha) > 0$ iff $P^\star$ has a model Pr with $Pr(\alpha) > 0$. □

Proof of Theorem 8.7. As argued in the proof of Theorem 8.5, P and $P^\star$ have the same sets of models. This already proves that for every $s \in \{0, me\}$, it holds that $P \Vdash\!\sim^s_{tight} (\beta|\alpha)[l,u]$ iff $P^\star \Vdash\!\sim^s_{tight} (\beta|\alpha)[l,u]$. Recall then that $P \Vdash\!\sim^{mc}_{tight}(\beta|\alpha)[l,u]$ iff $P \cup \text{CWA}(P, \beta\wedge\alpha) \Vdash\!\sim^{me}_{tight}(\beta|\alpha)[l,u]$. So, to prove that $P \Vdash\!\sim^{mc}_{tight}(\beta|\alpha)[l,u]$ iff $P^\star \Vdash\!\sim^{mc}_{tight}(\beta|\alpha)[l,u]$, it is sufficient to show that $\text{CWA}(P, \beta\wedge\alpha) = \text{CWA}(P^\star, \beta\wedge\alpha)$. Observe first that $(*)$ $C \models \psi \Leftarrow \phi$ and $C \not\models \bot \Leftarrow \phi$ implies $app(C) \models \psi \Leftarrow \phi$. Clearly, every $(\psi|\phi)[l,u] \in D$ with either (i), or (ii), or (iv) can be removed from $ground(P)$ without changing $\text{CWA}(ground(P), \beta\wedge\alpha)$. By $(*)$, every $(\psi|\phi)[l,u] \in D$ with (iii) can also be removed without changing $\text{CWA}(ground(P), \beta\wedge\alpha)$, since either $C \models \bot \Leftarrow \phi$ or $app(C) \models \psi \Leftarrow \phi$. Hence, $\text{CWA}(P, \beta\wedge\alpha) = \text{CWA}(ground(P), \beta\wedge\alpha) = \text{CWA}(P^\star, \beta\wedge\alpha)$, and thus $P \Vdash\!\sim^{mc}_{tight}(\beta|\alpha)[l,u]$ iff $P^\star \Vdash\!\sim^{mc}_{tight}(\beta|\alpha)[l,u]$. □

Proof of Proposition 8.8. The set $vac_C(D)$ can be computed by checking for every $(\psi|\psi)[l,u] \in D$ whether a condition among (i)–(iv) is satisfied. In the ground conjunctive case, each such check is possible in time $O(\|P\|)$. In summary, $vac_C(D)$ can be computed in time $O(\|P\||D|)$. □

Proof of Theorem 8.10. $(\Rightarrow)$ We show that every model of P is also a model of $P^\star = C \cup act_{C,\alpha}(D)$. Let Pr be a model of P. In particular, Pr is a model of C and of all members of D that are s-active w.r.t. P and α. We now show that Pr is also a model of all $\bot \Leftarrow \phi$ such that $(\star)$ ϕ is s-active w.r.t. P and α, and $(\star\star)$ $(\psi|\phi)[l,u] \in D$ for some $l > 0$ and some ψ that is s-inactive w.r.t. P and α. Towards a contradiction, assume that Pr is not a model of some $\bot \Leftarrow \phi$ such that $(\star)$ and $(\star\star)$. Hence, $C \not\Vdash\!\sim^0 \bot \Leftarrow \phi$, and thus $\psi \Leftarrow \phi$ belongs to $s\text{-}app(P)$. But this contradicts ψ being s-inactive and ϕ being s-active w.r.t. P and α. Hence, Pr is a model of all $\bot \Leftarrow \phi$ such that $(\star)$ and $(\star\star)$. In summary, Pr is a model of $P^\star$.

$(\Leftarrow)$ We show that for every model $Pr^\star$ of $P^\star = C \cup act_{C,\alpha}(D)$, there exists a model Pr of P such that (1) $Pr(\gamma) = Pr^\star(\gamma)$ for all ground events γ that are s-active w.r.t. P and α, and (2) $Pr(p) = 0$ for all $p \in HB_\Phi$ that are s-inactive w.r.t. P and α. Let $Pr^\star$ be a model of $P^\star$. We define Pr by $Pr(I) = Pr^\star(\varepsilon_I)$ for all $I \in \mathcal{I}_\Phi$ with $I \not\models p$ for all s-inactive $p \in HB_\Phi$, where ε_I is the conjunction of all s-active ground atoms $q \in I$ and of all negations of s-active ground atoms $q \notin I$, and by $Pr(I) = 0$ for all other $I \in \mathcal{I}_\Phi$. Then, Pr satisfies (1) and (2), and thus Pr is also a model of all $F \in ground(P)$ that are s-active w.r.t. P and α. We now show that Pr is also a model of all s-inactive $F \in ground(P)$. Towards a contradiction, assume that Pr is not a model of some s-inactive $(\psi|\phi)[l,u] \in ground(P)$. Thus, ϕ is s-active, ψ is s-inactive, and $l > 0$. Hence, $C \Vdash\!\sim^0 \bot \Leftarrow \phi$. Hence, $Pr(\phi) = Pr^\star(\phi) = 0$, but this contradicts Pr not being a model of $(\psi|\phi)[l,u]$. This shows that Pr is also a model of all s-inactive $F \in ground(P)$. In summary, Pr is a model of P. □

Proof of Theorem 8.12. We define $R = \{Pr(\beta|\alpha) \mid Pr \models P,\ Pr(\alpha) > 0\}$ and $R^\star = \{Pr(\beta|\alpha) \mid Pr \models P^\star,$ $Pr(\alpha) > 0\}$. As argued in the proof of Theorem 8.10, every model of P satisfies $P^\star$, and for every model

$Pr^\star$ of $P^\star$, there exists a model Pr of P such that $Pr(\gamma) = Pr^\star(\gamma)$ for all ground events γ that are s-active w.r.t. P and $\beta \wedge \alpha$. It thus follows $R \subseteq R^\star$ and $R^\star \subseteq R$, respectively. This already shows that $P \Vdash^0_{tight} (\beta|\alpha)[l,u]$ iff $P^\star \Vdash^0_{tight} (\beta|\alpha)[l,u]$. Recall then that $P \Vdash^{mc}_{tight}(\beta|\alpha)[l,u]$ iff $P \cup \text{CWA}(P, \beta\wedge\alpha)$ $\Vdash^{me}_{tight}(\beta|\alpha)[l,u]$. We now prove that $(*)$ $\text{CWA}(P, \beta\wedge\alpha) = \text{CWA}(P^\star, \beta\wedge\alpha)$. Clearly, $\text{CWA}(P, \beta\wedge\alpha) \subseteq$ $\text{CWA}(P^\star, \beta\wedge\alpha)$, since $C \Vdash^0 \bot \Leftarrow \phi$ for all s-active ϕ w.r.t. P and $\beta \wedge \alpha$ such that $(\psi|\phi)[l,u] \in ground(P)$ for some $l > 0$ and some s-inactive ψ w.r.t. P and $\beta \wedge \alpha$. To prove the converse, consider some $(\psi|\phi)[l,u] \in$ $ground(P)$ such that $l > 0$, that $P \not\Vdash^0 \bot \Leftarrow \phi$, and that ϕ is s-inactive w.r.t. P and $\beta \wedge \alpha$. Hence, ϕ is also inactive w.r.t. P and $\beta\wedge\alpha$. Thus, $\psi \Leftarrow \phi$ in $app(P) - app(P^\star)$ does not have any influence on $\text{CWA}(P, \beta\wedge\alpha)$. Hence, $\text{CWA}(P, \beta\wedge\alpha) \supseteq \text{CWA}(P^\star, \beta\wedge\alpha)$, and thus $(*)$ holds. By the proof of Theorem 8.10, every model of P satisfies $P^\star$. Thus, $(*)$ implies that $(**)$ every model of $P \cup \text{CWA}(P, \beta\wedge\alpha)$ is also a model of $P^\star \cup$ $\text{CWA}(P^\star, \beta\wedge\alpha)$. We now show the converse, namely that $(***)$ every model of $P^\star \cup \text{CWA}(P^\star, \beta\wedge\alpha)$ is also a model of $P \cup \text{CWA}(P, \beta\wedge\alpha)$. Towards a contradiction, assume that there exists some model Pr of $P^\star \cup \text{CWA}(P^\star, \beta\wedge\alpha)$ that is not a model of some $(\psi|\phi)[l,u] \in D$. Thus, $(\psi|\phi)[l,u]$ is s-inactive and thus also inactive w.r.t. P and $\beta \wedge \alpha$. Hence, $l > 0$, ϕ is active, and ψ is inactive w.r.t. P and $\beta \wedge \alpha$. Thus, ϕ is s-active, and ψ is s-inactive w.r.t. P and $\beta \wedge \alpha$. Hence, $\bot \Leftarrow \phi$ belongs to $P^\star$, and thus $Pr(\phi) = 0$, but this contradicts Pr not satisfying $(\psi|\phi)[l,u]$. This shows that $(***)$ holds. By $(**)$ and $(***)$, it thus follows that $P \Vdash^{mc}_{tight}(\beta|\alpha)[l,u]$ iff $P^\star \Vdash^{mc}_{tight}(\beta|\alpha)[l,u]$. □

Proof of Proposition 8.13. We first compute $s\text{-}app(P)$. For each $(\psi|\phi)[l,u] \in P$, we decide whether $l > 0$ and $C \not\Vdash^0 \bot \Leftarrow \phi$. This can be done in time $O(\|P\||P|)$. We then compute the set of all ground atoms that are s-active w.r.t. P and α. This can be done in time $O(\|P\| + \|\alpha\|)$. Once we are given the set of all ground atoms that are s-active w.r.t. P and α, the set $act_{C,\alpha}(D)$ can be computed in time $O(\|P\|)$. In summary, computing $act_{C,\alpha}(D)$ can be done in time $O(\|P\||P| + \|\alpha\|)$. □

Proof of Theorem 8.14. $(\Rightarrow)$ Every model Pr of P with $Pr(\alpha) > 0$ is a model of $C \cup D_1$ with $Pr(\alpha) > 0$ and also a model of every $C \cup D_i$ with $i \in \{2, \ldots, k\}$.

$(\Leftarrow)$ For $i \in \{1, \ldots, k\}$, let C_i be the set of all $F \in ground(C)$ defined over H_i. Assume that (i) and (ii) hold. It follows that every $C_i \cup D_i$ with $i \in \{1, \ldots, k\}$ has a model Pr_i over 2^{H_i}, where $Pr_1(\alpha) > 0$. Thus, the probabilistic interpretation Pr over $2^{HB_{P,\alpha}}$ that is defined by $Pr(I) = Pr_1(I_1) \cdots Pr_k(I_k)$ for all $I \in 2^{HB_{P,\alpha}}$, where $I_i = I \cap H_i$ for all $i \in \{1, \ldots, k\}$, is a model of P with $Pr(\alpha) > 0$. □

Proof of Theorem 8.16. (a) For $i \in \{1, \ldots, k\}$, denote by C_i the set of all members of $ground(C)$ that are defined over H_i. Assume that each $C \cup D_i$ with $i \in \{2, \ldots, k\}$ is satisfiable. We define $R = \{Pr(\beta|\alpha) \mid Pr \models P,\ Pr(\alpha) > 0\}$ and $R_1 = \{Pr(\beta|\alpha) \mid Pr \models C \cup D_1,\ Pr(\alpha) > 0\}$. Since every model Pr of P with $Pr(\alpha) > 0$ is also a model of $C \cup D_1$ with $Pr(\alpha) > 0$, it follows that $R \subseteq R_1$. To show the converse, assume that Pr is a model of $C \cup D_1$ with $Pr(\alpha) > 0$. Hence, there exists a model Pr_1 of $C_1 \cup D_1$ over 2^{H_1} such that $Pr_1(\alpha) > 0$ and $Pr_1(\beta|\alpha) = Pr(\beta|\alpha)$. As each $C \cup D_i$ with $i \in \{2, \ldots, k\}$ is satisfiable, every $C_i \cup D_i$ with $i \in \{2, \ldots, k\}$ has a model Pr_i over 2^{H_i}. Thus, the probabilistic interpretation Pr' over $2^{HB_{P,\beta\wedge\alpha}}$ that is defined by $Pr'(I) = Pr_1(I_1) \cdots Pr_k(I_k)$ for all $I \in 2^{HB_{P,\beta\wedge\alpha}}$, where $I_i = I \cap H_i$ for all $i \in \{1, \ldots, k\}$, is a model of P with $Pr'(\beta|\alpha) = Pr_1(\beta|\alpha) = Pr(\beta|\alpha)$ and $Pr'(\alpha) = Pr_1(\alpha) > 0$. This shows that $R \supseteq R_1$. In summary, it holds that $R = R_1$. It thus follows that $P \Vdash^0_{tight} (\beta|\alpha)[l,u]$ iff $C \cup D_1 \Vdash^0_{tight} (\beta|\alpha)[l,u]$. By the proof of Theorem 8.14, it holds that P has a model Pr with $Pr(\alpha) > 0$ iff $C \cup D_1$ has such a model. Thus, $P \Vdash^s_{tight}(\beta|\alpha)[1,0]$ iff $C \cup D_1 \Vdash^s_{tight}(\beta|\alpha)[1,0]$ for all $s \in \{me, mc\}$. Assume now that P and $C \cup D_1$ have a model Pr with $Pr(\alpha) > 0$. Observe then that, $me[P](I) = \Pi_{i=1}^k me_i[C_i \cup D_i](\varepsilon_i)$ for all $I \in 2^{HB_{P,\beta\wedge\alpha}}$, where ε_i is the conjunction of all $p \in I \cap H_i$ and all negations of $p \notin I \cap H_i$, and me_i is the me-model over 2^{H_i}, for all $i \in \{1, \ldots, k\}$; this relationship is known as "system independence" from [76]. It then follows that $me[P](\beta|\alpha) = me[C_1 \cup D_1](\beta|\alpha)$, and also that $me[C \cup D_1](\beta|\alpha) = me[C_1 \cup D_1](\beta|\alpha)$. In summary, $me[P](\beta|\alpha) = me[C \cup D_1](\beta|\alpha)$, and thus $P \Vdash\!\sim^{me}_{tight} (\beta|\alpha)[l,u]$ iff $C \cup D_1 \Vdash\!\sim^{me}_{tight} (\beta|\alpha)[l,u]$. Recall then that $P \Vdash\!\sim^{mc} (\beta|\alpha)[l,u]$ iff $P \cup$ CWA$(P, \beta\wedge\alpha)$ $\Vdash\!\sim^{me}_{tight}$ $(\beta|\alpha)[l,u]$. The latter is equivalent to $C \cup D_1 \cup$ CWA$(C \cup D_1, \beta\wedge\alpha)$ $\Vdash\!\sim^{me}_{tight}$ $(\beta|\alpha)[l,u]$, since CWA$(P, \beta\wedge\alpha)$ and CWA$(C \cup D_1, \beta\wedge\alpha)$ coincide on their restriction to H_1. We have thus proved that $P \Vdash\!\sim^{mc}_{tight} (\beta|\alpha)[l,u]$ iff $C \cup D_1 \Vdash\!\sim^{mc}_{tight} (\beta|\alpha)[l,u]$.

(b) If some $C \cup D_i$ with $i \in \{2, \ldots, k\}$ is not satisfiable, then P is not satisfiable, and thus $P \Vdash\!\sim^{s}_{tight}$ $(\beta|\alpha)[1,0]$ for all $s \in \{0, me, mc\}$. □

Proof of Proposition 8.17. We first compute the set S of all connected components of the hypergraph $G = (V, E) = (HB_{P,\alpha}, \{At(F) \mid F \in P\} \cup \{At(\alpha)\})$, where $At(F)$ denotes the set of all ground atoms $p \in HB_P$ that occur in F. This set S can be computed in linear time using standard methods and data structures. Once S is given, $dec_{C,\alpha}(D)$ and $rel_{C,\alpha}(D)$ can be computed in linear time. In summary, $dec_{C,\alpha}(D)$ and $rel_{C,\alpha}(D)$ can be computed in linear time. □

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, A. Grove, J.Y. Halpern, and D. Koller. From statistical knowledge bases to degrees of belief. *Artif. Intell.*, 87:75–143, 1996.

[3] S. Benferhat, A. Saffiotti, and P. Smets. Belief functions and default reasoning. *Artif. Intell.*, 122(1–2):1–69, 2000.

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

[5] A. Charnes and W.W. Cooper. Programming with linear fractional functionals. *Naval Research Logistics Quaterly*, 9:181–186, 1962.

[6] V. Chvátal. *Linear Programming*. Freeman, San Francisco, 1983.

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

[8] A. Dekhtyar and V.S. Subrahmanian. Hybrid probabilistic programs. *J. Logic Program.*, 43(3):187–250, 2000.

[9] M.I. Dekhtyar, A. Dekhtyar, and V.S. Subrahmanian. Hybrid probabilistic programs: Algorithms and complexity. In *Proceedings UAI-99*, pages 160–169. Morgan Kaufmann, 1999.

[10] D. Dubois and H. Prade. Non-standard theories of uncertainty in plausible reasoning. In G. Brewka, editor, *Principles of Knowledge Representation*. CSLI Publications, Stanford, CA, 1996.

[11] D. Dubois, H. Prade, L. Godo, and R. López de Màntaras. Qualitative reasoning with imprecise probabilities. *Journal of Intelligent Information Systems*, 2:319–363, 1993.

[12] D. Dubois, H. Prade, and J.-M. Touscas. Inference with imprecise numerical quantifiers. In Z. W. Ras and M. Zemankova, editors, *Intelligent Systems*, chapter 3, pages 53–72. Ellis Horwood, 1990.

[13] T. Eiter and G. Gottlob. Identifying the minimal transversals of a hypergraph and related problems. *SIAM J. on Computing*, 24(6):1278–1304, 1995.

[14] T. Eiter and T. Lukasiewicz. Default reasoning from conditional knowledge bases: Complexity and tractable cases. *Artif. Intell.*, 124(2):169–241, 2000.

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

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

[17] A.M. Frisch and P. Haddawy. Anytime deduction for probabilistic logic. *Artif. Intell.*, 69:93–122, 1994.

[18] H. Gaifman. Concerning measures in first order calculi. *Israel J. Math.*, 2:1–18, 1964.

[19] M. Goldszmidt, P. Morris, and J. Pearl. A maximum entropy approach to nonmono- tonic reasoning. In *Proceedings AAAI-90*, pages 646–652. AAAI Press/MIT Press, 1990.

[20] M. Goldszmidt and J. Pearl. Qualitative probabilities for default reasoning, belief revision, and causal modeling. *Artif. Intell.*, 84:57–112, 1996.

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

[22] P. Haddawy. Generating Bayesian networks from probability logic knowledge bases. In *Proceedings UAI-94*, pages 262–269. Morgan Kaufmann, 1994.

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

[24] M. Jaeger. Relational Bayesian networks. In *Proceedings UAI-97*, pages 266–273. Morgan Kaufmann, 1997.

[25] M. Jaeger. On the complexity of inference about probabilistic relational models. *Artif. Intell.*, 117:297–308, 1999.

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

[27] F.V. Jensen. *Introduction to Bayesian networks*. UCL Press, London, 1996.

[28] 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.

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

[30] G. Kern-Isberner. A note on conditional logics and entropy. *Int. J. Approx. Reasoning*, 19:231–246, 1998.

[31] G. Kern-Isberner. *Conditionals in Nonmonotonic Reasoning and Belief Revision*, volume 2087 of *LNCS/LNAI*. Springer, 2001.

[32] D. Koller and A. Pfeffer. Object-oriented Bayesian networks. In *Proceedings UAI-97)*, pages 302–313. Morgan Kaufmann, 1997.

[33] S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artif. Intell.*, 14(1):167–207, 1990.

[34] H.E. Kyburg, Jr. *The Logical Foundations of Statistical Inference*. D. Reidel, Dordrecht, Netherlands, 1974.

[35] H.E. Kyburg, Jr. The reference class. *Philos. Sci.*, 50:374–397, 1983.

[36] K.B. Laskey and S.M. Mahoney. Network fragments: Representing knowledge for constructing probabilistic models. In *Proceedings UAI-97)*, pages 334–341. Morgan Kaufmann, 1997.

[37] S.L. Lauritzen and D.J. Spiegelhalter. Local computations with probabilities in graphical structures and their applications to expert systems. *Journal of the Royal Statistical Society B*, 50(2):415–448, 1988.

[38] D. Lehmann and M. Magidor. What does a conditional knowledge base entail? *Artif. Intell.*, 55:1–60, 1992.

[39] T. Lukasiewicz. Efficient global probabilistic deduction from taxonomic and probabilistic knowledge-bases over conjunctive events. In *Proceedings CIKM-97*, pages 75–82. ACM Press, 1997.

[40] T. Lukasiewicz. Probabilistic logic programming. In *Proceedings ECAI-98*, pages 388–392. Wiley & Sons, 1998.

[41] T. Lukasiewicz. Local probabilistic deduction from taxonomic and probabilistic knowledge-bases over conjunctive events. *Int. J. Approx. Reasoning*, 21(1):23–61, 1999.

[42] T. Lukasiewicz. Probabilistic deduction with conditional constraints over basic events. *J. Artif. Intell. Res.*, 10:199–241, 1999.

[43] T. Lukasiewicz. Credal networks under maximum entropy. In *Proceedings UAI-2000*, pages 363–370. Morgan Kaufmann, 2000.

[44] T. Lukasiewicz. Probabilistic logic programming under inheritance with overriding. In *Proceedings UAI-2001*, pages 329–336. Morgan Kaufmann, 2001.

[45] T. Lukasiewicz. Probabilistic logic programming with conditional constraints. *ACM Transactions on Computational Logic (TOCL)*, 2(3):264–312, 2001.

[46] T. Lukasiewicz. Nonmonotonic probabilistic logics between model-theoretic prob- abilistic logic and probabilistic logic under coherence. In *Proceedings of the 9th International Workshop on Non-Monotonic Reasoning*, pages 265–274, 2002.

[47] T. Lukasiewicz. Probabilistic default reasoning with conditional constraints. *Ann. Math. Artif. Intell.*, 34(1–3):35–88, 2002.

[48] T. Lukasiewicz and G. Kern-Isberner. Probabilistic logic programming under maximum entropy. In *Proceedings ECSQARU-99*, volume 1638 of *LNCS/LNAI*, pages 279–292. Springer, 1999.

[49] C. Luo, C. Yu, J. Lobo, G. Wang, and T. Pham. Computation of best bounds of probabilities from uncertain data. *Computational Intelligence*, 12(4):541–566, 1996.

[50] C.H. Meyer. *Korrektes Schliessen bei unvollständiger Information*. Peter Lang Verlag, 1998.

[51] R.E. Neapolitan. *Probabilistic Reasoning in Expert Systems*. Wiley, New York, 1990.

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

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

[54] 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.

[55] L. Ngo and P. Haddawy. Answering queries from context-sensitive probabilistic knowledge bases. *Theor. Comput. Sci.*, 171:147–177, 1997.

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

[57] C. Ohmann, V. Moustakis, Q. Yang, and K.Lang. Evaluation of automatic knowledge acquisition techniques in the diagnosis of acute abdominal pain. *Artificial Intelligence in Medicine*, 8:23–26, 1996.

[58] J.B. Paris. *The Uncertain Reasoner's Companion: A Mathematical Perspective*. Cambridge University Press, Cambridge, UK, 1995.

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

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

[61] J.L. Pollock. *Nomic Probabilities and the Foundations of Induction*. Oxford University Press, Oxford, 1990.

[62] D. Poole. Logic programming, abduction and probability. *New Gener. Comput.*, 11:377–400, 1993.

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

[64] H. Reichenbach. *Theory of Probability*. University of California Press, Berkeley, CA, 1949.

[65] W. Rödder and G. Kern-Isberner. Léa Sombé und Entropie-optimale Informationsver- arbeitung mit der Expertensystem-Shell SPIRIT. *OR Spektrum*, 19/3, 1997.

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

[67] W. Rödder and C.-H. Meyer. Coherent knowledge processing at maximum entropy by SPIRIT. In *Proceedings UAI-96*, pages 470–476. Morgan Kaufmann, 1996.

[68] M. Ruprecht. Implementierung und Performance-Auswertung für probabilistisches Schließen mit taxonomischem Wissen. Master's thesis, Universität Augsburg, 1996.

[69] M. Schramm and W. Ertel. Reasoning with probability and maximum entropy: The System PIT and its application in LEXMED. In *Proceedings of the Symposium on Operations Research (SOR-99)*, pages 274–280. Springer, 1999.

[70] M. Schramm and V. Fischer. Probabilistic reasoning with maximum entropy — The System PIT. In *Proceedings of the 12th Workshop on Logic Programming*, 1997.

[71] M. Schramm and B. Fronhöfer. PIT: A system for reasoning with probabilities. In *Proceedings of the KI-2001 Workshop on Uncertainty in AI*, pages 109–123, 2001.

[72] A. Schrijver. *Theory of Linear and Integer Programming*. J. Wiley, New York, 1986.

[73] M. Schweitzer. Wissensfindung in Datenbanken auf probabilistischer Basis. Master's thesis, FernUniversität Hagen, 1998.

[74] D. Scott and P. Krauss. Assigning probabilities to logical formulas. In J. Hintikka and P. Suppes, editors, *Aspects of Inductive Logic*, pages 219–264. North-Holland, Amsterdam, 1966.

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

[76] 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.

[77] J.E. Shore and R.W. Johnson. Properties of cross-entropy minimization. *IEEE Trans. Inf. Theory*, IT-27:472–482, 1981.