INFSYS
RESEARCH
REPORT

INSTITUT FÜR INFORMATIONSSYSTEME

ARBEITSBEREICH WISSENSBASIERTE SYSTEME

PROBABILISTIC DESCRIPTION LOGICS FOR THE SEMANTIC WEB

THOMAS LUKASIEWICZ

INFSYS RESEARCH REPORT 1843-06-05

MARCH 2007

Institut für Informationssysteme
AB 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

INFSYS RESEARCH REPORT

INFSYS RESEARCH REPORT 1843-06-05, MARCH 2007

PROBABILISTIC DESCRIPTION LOGICS FOR THE SEMANTIC WEB

MARCH 15, 2007

Thomas Lukasiewicz [1]

Abstract. The work in this paper is directed towards sophisticated formalisms for reasoning under probabilistic uncertainty in ontologies in the Semantic Web. Ontologies play a central role in the development of the Semantic Web, since they provide a precise definition of shared terms in web resources. They are expressed in the standardized web ontology language OWL, which consists of the three increasingly expressive sublanguages OWL Lite, OWL DL, and OWL Full. The sublanguages OWL Lite and OWL DL have a formal semantics and a reasoning support through a mapping to the expressive description logics $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$, respectively. In this paper, we present the expressive probabilistic description logics P-$\mathcal{SHIF}(\mathbf{D})$ and P-$\mathcal{SHOIN}(\mathbf{D})$, which are probabilistic extensions of these description logics. They allow for expressing rich terminological probabilistic knowledge about concepts and roles as well as assertional probabilistic knowledge about instances of concepts and roles. They are semantically based on the notion of probabilistic lexicographic entailment from probabilistic default reasoning, which naturally interprets this terminological and assertional probabilistic knowledge as knowledge about random and concrete instances, respectively. As an important additional feature, they also allow for expressing terminological default knowledge, which is semantically interpreted as in Lehmann's lexicographic entailment in default reasoning from conditional knowledge bases. We then present sound and complete algorithms for the main reasoning problems in the new probabilistic description logics, which are based on reductions to reasoning in their classical counterparts, and to solving linear optimization problems. In particular, this shows the important result that reasoning in the new probabilistic description logics is decidable / computable. Furthermore, we also analyze the computational complexity of the main reasoning problems in the new probabilistic description logics in the general as well as restricted cases.

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

Acknowledgements: This paper is a significantly extended and revised version of a paper that appeared in: *Proceedings JELIA-2002*, volume 2424 of *LNCS*, pp. 86–97. Springer, 2002 [30].

This work was supported by a Heisenberg Professorship of the German Research Foundation (DFG). I am very grateful to Rosalba Giugno for her contributions to the JELIA-2002 abstract of this paper.

Contents

1 Introduction

The *Semantic Web* initiative [8, 23] aims at an extension of the current World Wide Web by standards and technologies that help machines to understand the information on the Web so that they can support richer discovery, data integration, navigation, and automation of tasks. The main ideas behind the Semantic Web are to add a machine-readable meaning to Web pages to use ontologies for a precise definition of shared terms in Web resources, to make use of knowledge representation technology for automated reasoning from Web resources, and to apply cooperative agent technology for processing the information of the Web.

The Semantic Web consists of several hierarchical layers, where the *Ontology layer*, in the form of the *OWL Web Ontology Language* [75] (recommended by the W3C), is currently the highest layer of sufficient maturity. The language OWL consists of the three increasingly expressive sublanguages *OWL Lite*, *OWL DL*, and *OWL Full*, where OWL Lite and OWL DL are essentially very expressive description logics with an RDF syntax. As shown in [40], ontology entailment in OWL Lite and OWL DL reduces to knowledge base (un)satisfiability in the expressive description logics $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$, respectively.

Intuitively, description logics model a domain of interest in terms of concepts and roles, which represent classes of individuals resp. binary relations on classes of individuals. A description logic knowledge base (or ontology) encodes in particular subset relationships between classes of individuals, subset relationships between binary relations on classes of individuals, the membership of individuals to classes, and the membership of pairs of individuals to binary relations on classes. Other important ingredients of the description logics $\mathcal{SHIF}(\mathbf{D})$ (resp., $\mathcal{SHOIN}(\mathbf{D})$) are datatypes (resp., datatypes and individuals) in concept expressions.

A next crucial step in the construction of the Semantic Web is especially the development of sophisticated representation and reasoning capabilities for the *Rules*, *Logic*, and *Proof layers* of the Semantic Web. Several recent research efforts are going in this direction. In particular, a large body of work focuses on integrating rules with description logics / ontologies, which is a key requirement of the layered architecture of the Semantic Web. Another large body of work concentrates on handling uncertainty in the Semantic Web, which aims in particular at adding probabilistic uncertainty to description logics / ontologies (see Section 7.1) and to integrations of rules with description logics / ontologies [59]. An important recent forum for approaches to uncertainty in the Semantic Web is the annual *Workshop on Uncertainty Reasoning for the Semantic Web (URSW)*. There exists also a W3C Incubator Group on *Uncertainty Reasoning for the World Wide Web*.

In this paper, we present a novel combination of description logics with probabilistic uncertainty, which is especially directed towards sophisticated formalisms for reasoning under probabilistic uncertainty in ontologies in the Semantic Web. More concretely, we present probabilistic extensions of $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$.

Intuitively, probabilistic description logic knowledge bases / ontologies extend classical description logic knowledge bases / ontologies in particular by probabilistic knowledge about concepts and roles (also called *terminological probabilistic knowledge*) as well as probabilistic knowledge about the instances of concepts and roles (also called *assertional probabilistic knowledge*). The former is probabilistic knowledge about randomly chosen (or generic) instances of concepts and roles, while the latter is probabilistic knowledge about concrete instances of concepts and roles. A detailed introduction to probabilistic ontologies is given in [11, 12].

As noted in [11, 12], there is a plethora of applications with an urgent need for handling uncertain knowledge in formal domain ontologies, especially in areas like medicine, biology, defense, and astronomy. Furthermore, there are strong arguments for the critical need of dealing with probabilistic uncertainty in ontologies in the Semantic Web, some of which are briefly summarized as follows:

- In addition to being logically related, the concepts of an ontology are generally also probabilistically related. For example, two concepts either may be logically related via a subset or disjointness relationship, or they may show a certain degree of overlap. Probabilistic ontologies allow for quantifying these degrees of overlap, reasoning about them, and using them in semantic-web applications. In particular, probabilistic ontologies are successfully used in information retrieval for an increased recall [43, 42] (see also Section 7.1). The degrees of concept overlap may also be exploited in personalization and recommender systems.

- Rather than consisting of one standardized overall ontology, the Semantic Web will consist of a huge collection of different ontologies. Hence, in semantic-web applications such as automated reasoning and information retrieval, one has to align the concepts of different ontologies, which is called *ontology matching*. In general, the concepts of two different ontologies do not match exactly, and we have to deal with degrees of concept overlap as above, which can again be quantified and handled via probabilistic ontologies (see also Section 7.1).

- Like the current Web, the Semantic Web will necessarily contain ambiguous and controversial pieces of information in different web sources. This can be handled via probabilistic data integration by associating with every web source a probability value describing its degree of reliability [74, 36]. As resulting pieces of data, such a probabilistic data integration process necessarily produces probabilistic facts, that is, probabilistic knowledge at the instance level.

As underlying probabilistic reasoning formalism, we use the notion of lexicographic entailment from probabilistic default reasoning [54, 56], which is a probabilistic generalization of Lehmann's lexicographic entailment [51] in default reasoning from conditional knowledge bases. It is a formalism for reasoning from probabilistic knowledge about random and concrete objects with very nice features: In particular, it shows a similar behavior as reference-class reasoning in a number of uncontroversial examples. But it also avoids many drawbacks of reference-class reasoning: It can handle complex scenarios and even purely probabilistic subjective knowledge as input, and conclusions are drawn in a global way from all the available knowledge as a whole. Furthermore, it also has very nice nonmonotonic properties, which are essentially inherited from Lehmann's lexicographic entailment: In particular, it realizes an inheritance of properties along subclass relationships, where more specific properties override less specific properties, without showing the problem of inheritance blocking (where properties are not inherited to subclasses that are exceptional relative to some other properties). As for general nonmonotonic properties, it satisfies (probabilistic versions of) the rationality postulates by Kraus, Lehmann, and Magidor [50], the property of rational monotonicity, and some irrelevance, conditioning, and inclusion properties. All these quite appealing features carry over to our new probabilistic description logics in this paper.

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

- We present the description logics P-$\mathcal{SHIF}(\mathbf{D})$ and P-$\mathcal{SHOIN}(\mathbf{D})$, which are probabilistic generalizations of the expressive description logics $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$ behind OWL Lite resp. OWL DL. They allow for expressing rich terminological and assertional probabilistic knowledge in addition to terminological and assertional classical knowledge in $\mathcal{SHIF}(\mathbf{D})$ resp. $\mathcal{SHOIN}(\mathbf{D})$. To my knowledge, this is the first work extending $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$ by probabilistic uncertainty, and the first work on probabilistic description logics that explicitly allow for both terminological probabilistic knowledge about concepts and roles, as well as assertional probabilistic knowledge about concepts and roles (see Section 7.1 for a more detailed comparison to related work).

- Semantically, the new probabilistic description logics are based on probabilistic lexicographic entailment from probabilistic default reasoning [54, 56] (and thus inherit all its nice features), which naturally interprets terminological and assertional probabilistic knowledge as probabilistic knowledge about random and concrete instances of concepts and roles, respectively, and which allows for deriving probabilistic knowledge about both random and concrete instances.

- As an important additional feature, the new probabilistic description logics also allow for expressing default knowledge about concepts and roles (which is a special type of terminological probabilistic knowledge). This knowledge is semantically interpreted as in the sophisticated notion of lexicographic default entailment by Lehmann [51]. To my knowledge, this is the first work combining description logics with default reasoning from conditional knowledge bases.

- We present sound and complete algorithms for solving the main reasoning problems in the description logics P-$\mathcal{SHIF}(\mathbf{D})$ and P-$\mathcal{SHOIN}(\mathbf{D})$. These algorithms are based on reductions to classical reasoning problems in $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$, respectively, and to solving linear optimization problems. In particular, this shows the important result that the main reasoning problems in the new probabilistic description logics are decidable / computable.

- We analyze the complexity of the main reasoning problems in the new probabilistic description logics in the general as well as restricted cases. In particular, the problems of deciding consistency in P-$\mathcal{SHIF}(\mathbf{D})$ and P-$\mathcal{SHOIN}(\mathbf{D})$ have the same complexity (complete for EXP and NEXP) as deciding knowledge base satisfiability in $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$, respectively, while computing tight probability intervals under lexicographic entailment can be done with only slightly higher effort (complete for FEXP and in FP^{NEXP}, respectively).

- Finally, we also analyze the complexity of the main reasoning tasks in a probabilistic extension of the description logic *DL-Lite* [10]. In this special case, deciding consistency and computing tight probability intervals under lexicographic entailment are shown to have the same complexity (complete for NP and FP^{NP}) as deciding consistency and computing tight intervals under lexicographic entailment in probabilistic default reasoning, respectively.

The rest of this paper is organized as follows. In Section 2, we provide motivating examples from the medical domain and from information retrieval. Section 3 recalls the expressive description logics $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$. In Section 4, we present the probabilistic description logics P-$\mathcal{SHIF}(\mathbf{D})$ and P-$\mathcal{SHOIN}(\mathbf{D})$. Sections 5 and 6 provide algorithms for the main computational problems in the new probabilistic description logics and analyze their complexity, respectively. In Section 7, we discuss related work. Section 8 summarizes the main contributions of this paper and gives an outlook on future research. Note that detailed proofs of all results in the body of this paper are given in Appendices A to C.

2 Motivating Examples

To illustrate probabilistic ontologies, consider some medical knowledge about patients. In such knowledge, we often encounter terminological probabilistic and terminological default knowledge about classes of individuals, as well as assertional probabilistic knowledge about individuals. Such knowledge may e.g. be used by a medical company in an advertising campaign for a new product.

Example 2.1 *(Medical Example)* Consider patient records related to cardiological illnesses. We distinguish between heart patients (who have any kind of cardiological illness), pacemaker patients, male pacemaker patients, and female pacemaker patients, who all are associated with illnesses, illness statuses, symptoms of illnesses, and health insurances. Furthermore, we have the patients Tom, John, and Maria, where Tom is a heart patient, while John and Maria are male and female pacemaker patients, respectively, and John has the symptoms arrhythmia (abnormal heart beat), chest pain, and breathing difficulties, and the illness status advanced.

Then, *terminological default knowledge* is of the form "generally (or typically / in nearly all cases), heart patients suffer from high blood pressure" and "generally, pacemaker patients do not suffer from high blood pressure", while *terminological probabilistic knowledge* has the form "generally, pacemaker patients are male with a probability of at least 0.4" (that is, "generally, a randomly chosen pacemaker patient is male with a probability of at least 0.4"), "generally, heart patients have a private insurance with a probability of at least 0.9", and "generally, pacemaker patients have the symptoms arrhythmia, chest pain, and breathing difficulties with probabilities of at least 0.98, 0.9, and 0.6, respectively". Finally, *assertional probabilistic knowledge* is of the form "Tom is a pacemaker patient with a probability of at least 0.8", "Maria has the symptom breathing difficulties with a probability of at least 0.6", "Maria has the symptom chest pain with a probability of at least 0.9", and "Maria's illness status is final with a probability between 0.2 and 0.8".

Suppose now that a medical company wants to carry out a targeted advertising campaign about a new pacemaker product. The company may then first collect all potential addressees of such a campaign (e.g., pharmacies, hospitals, doctors, and heart patients) by probabilistic data integration from different web and data sources (e.g., web listings of pharmacies, hospitals, and doctors along with their product portfolio resp. fields of expertise; and available online databases with data of clients and their shopping histories). The result of this process is a collection of individuals with probabilistic memberships to a collection of concepts in a medical ontology as the one above. The terminological probabilistic and terminological default knowledge of this ontology can then be used to derive probabilistic concept memberships that are relevant for a potential addressee of the advertising campaign. For example, for persons that are known to be heart patients with certain probabilities, we may derive the probabilities with which they are also pacemaker patients.

The next example illustrates the use of probabilistic ontologies in information retrieval (which has especially been explored in [43, 42]; see also Section 7.1).

Example 2.2 *(Literature Search)* Suppose that we want to obtain a list of research papers in the area of "logic programming". Then, we should not only collect those papers that are classified as "logic programming" papers, but we should also search for papers in closely related areas, such as "rule-based systems" or "deductive databases", as well as in more general areas, such as "knowledge representation and reasoning" or "artificial intelligence" (since a paper may very well belong to the area of "logic programming", but is classified only with a closely related or a more general area). This expansion of the search can be done automatically using a probabilistic ontology, which has the papers as individuals, the areas as concepts, and the explicit paper classifications as concept memberships. The probabilistic degrees of overlap between the concepts in such a probabilistic ontology then provide a means of deriving a probabilistic membership to the concept "logic programming" and so a probabilistic estimation for the relevance to our search query.

3 The Description Logics $\mathcal{SHIF}(\mathrm{D})$ and $\mathcal{SHOIN}(\mathrm{D})$

In this section, we recall the description logics $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$, which stand behind the web ontology languages OWL Lite and OWL DL, respectively. See especially [40] for further details and background.

3.1 Syntax

We now recall the syntax of $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$. We first describe the syntax of the latter, which has the following datatypes and elementary ingredients. We assume a set of *elementary datatypes* and a set of *data values*. A *datatype* is an elementary datatype or a set of data values (called *datatype oneOf*). A *datatype theory* $\mathbf{D} = (\Delta^{\mathbf{D}}, \cdot^{\mathbf{D}})$ consists of a *datatype domain* $\Delta^{\mathbf{D}}$ and a mapping $\cdot^{\mathbf{D}}$ that assigns to each elementary datatype a subset of $\Delta^{\mathbf{D}}$ and to each data value an element of $\Delta^{\mathbf{D}}$. We extend $\cdot^{\mathbf{D}}$ to all datatypes by $\{v_1, \ldots\}^{\mathbf{D}} = \{v_1^{\mathbf{D}}, \ldots\}$. Let $\mathbf{A}$, $\mathbf{R}_A$, $\mathbf{R}_D$, and $\mathbf{I}$ be pairwise disjoint finite nonempty sets of *atomic concepts*, *abstract roles*, *datatype roles*, and *individuals*, respectively. We denote by $\mathbf{R}_A^-$ the set of inverses R^- of all $R \in \mathbf{R}_A$.

Roles and concepts are defined as follows. A *role* is any element of $\mathbf{R}_A \cup \mathbf{R}_A^- \cup \mathbf{R}_D$. *Concepts* are inductively defined as follows. Each $\phi \in \mathbf{A}$ is a concept, and if $o_1, \ldots, o_n \in \mathbf{I}$, then $\{o_1, \ldots, o_n\}$ is a concept (called *oneOf*). If ϕ, ϕ_1, and ϕ_2 are concepts and if $R \in \mathbf{R}_A \cup \mathbf{R}_A^-$, then also $\neg\phi$, $(\phi_1 \sqcap \phi_2)$, and $(\phi_1 \sqcup \phi_2)$ are concepts (called *negation*, *conjunction*, and *disjunction*, respectively), as well as $\exists R.\phi$, $\forall R.\phi$, $\geq nR$, and $\leq nR$ (called *exists*, *value*, *atleast*, and *atmost restriction*, respectively) for an integer $n \geq 0$. If D is a datatype and $U \in \mathbf{R}_D$, then $\exists U.D$, $\forall U.D$, $\geq nU$, and $\leq nU$ are concepts (called *datatype exists*, *value*, *atleast*, and *atmost restriction*, respectively) for an integer $n \geq 0$. We use $\top$ (resp., $\bot$) to abbreviate $\phi \sqcup \neg\phi$ (resp., $\phi \sqcap \neg\phi$), and eliminate parentheses as usual.

We next define axioms and knowledge bases. An *axiom* is an expression of one of the following forms: (1) $\phi \sqsubseteq \psi$ (called *concept inclusion axiom*), where ϕ and ψ are concepts; (2) $R \sqsubseteq S$ (called *role inclusion axiom*), where either $R, S \in \mathbf{R}_A$ or $R, S \in \mathbf{R}_D$; (3) $\mathrm{Trans}(R)$ (called *transitivity axiom*), where $R \in \mathbf{R}_A$; (4) $\phi(a)$ (called *concept membership axiom*), where ϕ is a concept and $a \in \mathbf{I}$; (5) $R(a, b)$ (resp., $U(a, v)$) (called *role membership axiom*), where $R \in \mathbf{R}_A$ (resp., $U \in \mathbf{R}_D$) and $a, b \in \mathbf{I}$ (resp., $a \in \mathbf{I}$ and v is a data value); and (6) $a = b$ (resp., $a \neq b$) (*equality* (resp., *inequality*) *axiom*), where $a, b \in \mathbf{I}$. A *classical (description logic) knowledge base* KB is a finite set of axioms.

For abstract roles $R \in \mathbf{R}_A$, we define $\mathrm{Inv}(R) = R^-$ and $\mathrm{Inv}(R^-) = R$. Let the transitive and reflexive closure of $\sqsubseteq$ on abstract roles relative to KB, denoted $\sqsubseteq^\star$, be defined as follows. For two abstract roles R and S in KB, let $S \sqsubseteq^\star R$ relative to KB iff either (a) $S = R$, or (b) $S \sqsubseteq R \in KB$, or (c) $\mathrm{Inv}(S) \sqsubseteq \mathrm{Inv}(R) \in KB$, or (d) some abstract role Q exists such that $S \sqsubseteq^\star Q$ and $Q \sqsubseteq^\star R$ relative to KB. An abstract role R is *simple* relative to KB iff for each abstract role S such that $S \sqsubseteq^\star R$ relative to KB, it holds (i) $\mathrm{Trans}(S) \notin KB$ and (ii) $\mathrm{Trans}(\mathrm{Inv}(S)) \notin KB$. For decidability, number restrictions in KB are restricted to simple abstract roles.

In $\mathcal{SHOIN}(\mathbf{D})$, concept and role membership axioms can also be expressed in terms of concept inclusion axioms, since $\phi(a)$ can be expressed by $\{a\} \sqsubseteq \phi$, while $R(a, b)$ (resp., $U(a, v)$) can be expressed by $\{a\} \sqsubseteq \exists R.\{b\}$ (resp., $\{a\} \sqsubseteq \exists U.\{v\}$).

The syntax of $\mathcal{SHIF}(\mathbf{D})$ is as the above syntax of $\mathcal{SHOIN}(\mathbf{D})$, but without the oneOf constructor and with the atleast and atmost constructors limited to 0 and 1.

Example 3.1 *(Medical Example cont'd)* Some atomic concepts in the Medical Example are *HeartPatient*, *PacemakerPatient*, *MalePacemakerPatient*, *FemalePacemakerPatient*, *Illness*, *IllnessStatus*, *IllnessSymptom*, and *HealthInsurance*. Some abstract roles are *HasIllness*, *HasIllnessStatus*, *HasIllnessSymptom*, and

HasHealthInsurance. Some individuals are *Tom*, *John*, *Maria*, *Arrhythmia*, *ChestPain*, *BreathingDifficulties*, *Advanced*, and *Final*. The knowledge that (1) all male and female pacemaker patients are pacemaker patients, (2) no pacemaker patient can be in the same time male and female, (3) all pacemaker patients are heart patients, (4) the role *HasIllnessSymptom* relates only heart patients with symptoms of illnesses, (5) Tom is a heart patient, (6) John is a male pacemaker patient, (7) Maria is a female pacemaker patient, (8) John has the symptoms arrhythmia, chest pain, and breathing difficulties, and the illness status advanced can be expressed by the following description logic knowledge base KB (note that other natural range restrictions on roles can be expressed by additional concept inclusion axioms):

(1) $\mathit{MalePacemakerPatient} \sqsubseteq \mathit{PacemakerPatient}$,
$\mathit{FemalePacemakerPatient} \sqsubseteq \mathit{PacemakerPatient}$,
(2) $\mathit{MalePacemakerPatient} \sqsubseteq \neg \mathit{FemalePacemakerPatient}$,
(3) $\mathit{PacemakerPatient} \sqsubseteq \mathit{HeartPatient}$,
(4) $\exists\, \mathit{HasIllnessSymptom}^{-}.\top \sqsubseteq \mathit{HeartPatient}$,
$\exists\, \mathit{HasIllnessSymptom}.\top \sqsubseteq \mathit{IllnessSymptom}$,
(5) $\mathit{HeartPatient}(\mathit{Tom})$,
(6) $\mathit{MalePacemakerPatient}(\mathit{John})$,
(7) $\mathit{FemalePacemakerPatient}(\mathit{Maria})$,
(8) $\mathit{HasIllnessSymptom}(\mathit{John}, \mathit{Arrhythmia})$,
$\mathit{HasIllnessSymptom}(\mathit{John}, \mathit{ChestPain})$,
$\mathit{HasIllnessSymptom}(\mathit{John}, \mathit{BreathingDifficulties})$,
$\mathit{HasIllnessStatus}(\mathit{John}, \mathit{Advanced})$.

3.2 Semantics

We now define the semantics of $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$ in terms of general first-order interpretations, as usual, and we recall some important reasoning problems in description logics.

An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ relative to a datatype theory $\mathbf{D} = (\Delta^{\mathbf{D}}, \cdot^{\mathbf{D}})$ consists of a nonempty (*abstract*) *domain* $\Delta^{\mathcal{I}}$ disjoint from $\Delta^{\mathbf{D}}$, and a mapping $\cdot^{\mathcal{I}}$ that assigns to each atomic concept $\phi \in \mathbf{A}$ a subset of $\Delta^{\mathcal{I}}$, to each individual $o \in \mathbf{I}$ an element of $\Delta^{\mathcal{I}}$, to each abstract role $R \in \mathbf{R}_A$ a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and to each datatype role $U \in \mathbf{R}_D$ a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathbf{D}}$. We extend $\cdot^{\mathcal{I}}$ to all roles and concepts as usual (where $\#S$ denotes the cardinality of a set S):

- $(R^{-})^{\mathcal{I}} = \{(y,x) \mid (x,y) \in R^{\mathcal{I}}\}$;
- $\{o_1, \ldots, o_n\}^{\mathcal{I}} = \{o_1^{\mathcal{I}}, \ldots, o_n^{\mathcal{I}}\}$; $(\neg \phi)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus \phi^{\mathcal{I}}$;
- $(\phi_1 \sqcap \phi_2)^{\mathcal{I}} = \phi_1^{\mathcal{I}} \cap \phi_2^{\mathcal{I}}$; $(\phi_1 \sqcup \phi_2)^{\mathcal{I}} = \phi_1^{\mathcal{I}} \cup \phi_2^{\mathcal{I}}$;
- $(\exists R.\phi)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y \colon (x,y) \in R^{\mathcal{I}} \wedge y \in \phi^{\mathcal{I}}\}$;
- $(\forall R.\phi)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \forall y \colon (x,y) \in R^{\mathcal{I}} \rightarrow y \in \phi^{\mathcal{I}}\}$;
- $({\geq} nR)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \#(\{y \mid (x,y) \in R^{\mathcal{I}}\}) \geq n\}$;
- $({\leq} nR)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \#(\{y \mid (x,y) \in R^{\mathcal{I}}\}) \leq n\}$;

- $(\exists U.D)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y\colon (x,y) \in U^{\mathcal{I}} \wedge y \in D^{\mathbf{D}}\}$;
- $(\forall U.D)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \forall y\colon (x,y) \in U^{\mathcal{I}} \rightarrow y \in D^{\mathbf{D}}\}$;
- $(\geq nU)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \#(\{y \mid (x,y) \in U^{\mathcal{I}}\}) \geq n\}$;
- $(\leq nU)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \#(\{y \mid (x,y) \in U^{\mathcal{I}}\}) \leq n\}$.

The *satisfaction* of an axiom F in an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ relative to a datatype theory $\mathbf{D} = (\Delta^{\mathbf{D}}, \cdot^{\mathbf{D}})$, denoted $\mathcal{I} \models F$, is defined as follows: (1) $\mathcal{I} \models \phi \sqsubseteq \psi$ iff $\phi^{\mathcal{I}} \subseteq \psi^{\mathcal{I}}$; (2) $\mathcal{I} \models R \sqsubseteq S$ iff $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$; (3) $\mathcal{I} \models \mathrm{Trans}(R)$ iff $R^{\mathcal{I}}$ is transitive; (4) $\mathcal{I} \models \phi(a)$ iff $a^{\mathcal{I}} \in \phi^{\mathcal{I}}$; (5) $\mathcal{I} \models R(a,b)$ iff $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$; (6) $\mathcal{I} \models U(a,v)$ iff $(a^{\mathcal{I}}, v^{\mathbf{D}}) \in U^{\mathcal{I}}$; (7) $\mathcal{I} \models a = b$ iff $a^{\mathcal{I}} = b^{\mathcal{I}}$; and (8) $\mathcal{I} \models a \neq b$ iff $a^{\mathcal{I}} \neq b^{\mathcal{I}}$. The interpretation $\mathcal{I}$ *satisfies* the axiom F, or $\mathcal{I}$ is a *model* of F, iff $\mathcal{I} \models F$. We say that $\mathcal{I}$ *satisfies* a knowledge base KB, or $\mathcal{I}$ is a *model* of KB, denoted $\mathcal{I} \models KB$, iff $\mathcal{I} \models F$ for all $F \in KB$. We say that KB is *satisfiable* (resp., *unsatisfiable*) iff KB has a (resp., no) model. An axiom F is a *logical consequence* of KB, denoted $KB \models F$, iff each model of KB satisfies F.

Some important reasoning problems in description logics are summarized as follows: (KBSAT) given a knowledge base KB, decide whether KB is satisfiable; (CSAT) given a knowledge base KB and a concept ϕ, decide whether $KB \not\models \phi \sqsubseteq \bot$; (CSUB) given a knowledge base KB and concepts ϕ and ψ, decide whether $KB \models \phi \sqsubseteq \psi$; (CMEM) given a knowledge base KB, an individual $o \in \mathbf{I}$, and a concept ϕ, decide whether $KB \models \phi(o)$; (RMEM) given a knowledge base KB, individuals $o, o' \in \mathbf{I}$ (resp., an individual $o \in \mathbf{I}$ and a value v), and a role $R \in \mathbf{R}_A$ (resp., $U \in \mathbf{R}_D$), decide whether $KB \models R(o,o')$ (resp., $KB \models U(o,v)$). Observe that KBSAT is a special case of CSAT, since KB is satisfiable iff $KB \not\models \top \sqsubseteq \bot$. Furthermore, for $\mathcal{SHOIN}(\mathbf{D})$, the problems CMEM and RMEM are special cases of CSUB, since $KB \models \phi(o)$ iff $KB \models \{o\} \sqsubseteq \phi$, and $KB \models R(o,o')$ (resp., $KB \models U(o,v)$) iff $KB \models \{o\} \sqsubseteq \exists R.\{o'\}$ (resp., $KB \models \{o\} \sqsubseteq \exists U.\{v\}$). Notice also that CSAT and CSUB can be reduced to each other, since $KB \models \phi \sqcap \neg\psi \sqsubseteq \bot$ iff $KB \models \phi \sqsubseteq \psi$. CSAT and CSUB are both decidable in $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$, if all atmost and atleast restrictions in KB are restricted to simple abstract roles relative to KB.

Example 3.2 *(Medical Example cont'd)* The description logic knowledge base KB of Example 3.1 is satisfiable and logically implies the concept inclusion axiom *FemalePacemakerPatient* $\sqsubseteq$ *HeartPatient* and the concept membership axioms *HeartPatient*(*John*) and *IllnessSymptom*(*Arrhythmia*).

4 The Probabilistic Description Logics P-$\mathcal{SHIF}(\mathbf{D})$ and P-$\mathcal{SHOIN}(\mathbf{D})$

In this section, we present the description logics P-$\mathcal{SHIF}(\mathbf{D})$ and P-$\mathcal{SHOIN}(\mathbf{D})$, which are probabilistic generalizations of $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$, respectively. We first define their syntax using the concept of a conditional constraint from [53] to express probabilistic knowledge in addition to the axioms of $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$, respectively. We then define their semantics using the notion of lexicographic entailment in probabilistic default reasoning [54, 56], which is a probabilistic generalization of the sophisticated notion of lexicographic entailment by Lehmann [51] in default reasoning from conditional knowledge bases. This semantics allows for expressing both terminological probabilistic knowledge about concepts and roles, and also assertional probabilistic knowledge about instances of concepts and roles. It naturally interprets terminological and assertional probabilistic knowledge as probabilistic knowledge about random and concrete instances of concepts and roles, respectively, and allows for deriving probabilistic knowledge about both random and concrete instances. As an important additional feature, it also allows for expressing default knowledge about concepts (as a special case of terminological probabilistic knowledge), which is semantically interpreted in the same way as in Lehmann's lexicographic default entailment [51]. See especially [56] for further details and background on probabilistic lexicographic entailment.

Informally, the main idea behind P-$\mathcal{SHIF}(\mathbf{D})$ and P-$\mathcal{SHOIN}(\mathbf{D})$ is to use a designated set of concepts from $\mathcal{SHIF}(\mathbf{D})$ resp. $\mathcal{SHOIN}(\mathbf{D})$ as basic events for lexicographic entailment in probabilistic default reasoning. Observe that this combination technique can be applied to other classical description logics as well, and is not limited to $\mathcal{SHIF}(\mathbf{D})$ resp. $\mathcal{SHOIN}(\mathbf{D})$.

4.1 Syntax

We now introduce the notion of a (description logic) probabilistic knowledge base. It is based on the language of conditional constraints [53], which encode interval restrictions for conditional probabilities over concepts. Informally, every probabilistic knowledge base consists of (i) a PTBox, which is a classical (description logic) knowledge base along with probabilistic terminological knowledge, and (ii) a collection of PABoxes, which encode probabilistic assertional knowledge about a certain set of individuals. To this end, we divide the set of individuals $\mathbf{I}$ of the classical description logic $\mathcal{SHIF}(\mathbf{D})$ (resp., $\mathcal{SHOIN}(\mathbf{D})$) into the set of *classical individuals* $\mathbf{I}_C$ and the set of *probabilistic individuals* $\mathbf{I}_P$, and we associate with every probabilistic individual a PABox. In the extreme cases, we have either only classical individuals (that is, $\mathbf{I} = \mathbf{I}_C$) or only probabilistic individuals (that is, $\mathbf{I} = \mathbf{I}_P$). Intuitively, the classical individuals here play the same role as in $\mathcal{SHIF}(\mathbf{D})$ (resp., $\mathcal{SHOIN}(\mathbf{D})$), while probabilistic individuals are those individuals in $\mathbf{I}$ for which we store some probabilistic assertional knowledge in a PABox.

Example 4.1 *(Medical Example cont'd)* Suppose we want to store some probabilistic knowledge about the individuals Tom, John, and Maria, such as "Tom is a pacemaker patient with a probability of at least 0.8". Then, the set of probabilistic individuals is given by $\mathbf{I}_P = \{\mathit{Tom}, \mathit{John}, \mathit{Maria}\}$.

We define the language of conditional constraints as follows. We assume a finite nonempty set $\mathcal{C}$ of *basic classification concepts* (or *basic c-concepts* for short), which are (not necessarily atomic) concepts in $\mathcal{SHIF}(\mathbf{D})$ (resp., $\mathcal{SHOIN}(\mathbf{D})$) that are free of individuals from $\mathbf{I}_P$. Informally, they are the relevant description logic concepts for defining probabilistic relationships. The set of *classification concepts* (or *c-concepts*) is inductively defined as follows. Every basic c-concept $\phi \in \mathcal{C}$ is a c-concept. If ϕ and ψ are c-concepts, then $\neg\phi$ and $(\phi \sqcap \psi)$ are also c-concepts. We often write $(\phi \sqcup \psi)$ to abbreviate $\neg(\neg\phi \sqcap \neg\psi)$, as usual. A *conditional constraint* is an expression of the form $(\psi|\phi)[l, u]$, where ϕ and ψ are c-concepts, and l and u are reals from $[0, 1]$. Informally, $(\psi|\phi)[l, u]$ encodes that the probability of ψ given ϕ lies between l and u.

Example 4.2 *(Medical Example cont'd)* The terminological probabilistic knowledge "generally, pacemaker patients are male with a probability of at least 0.4" (that is, "typically / in nearly all cases, a randomly chosen pacemaker patient is male with a probability of at least 0.4") can be expressed by the conditional constraint $(\mathit{MalePacemakerPatient} \mid \mathit{PacemakerPatient})[0.4, 1]$, while the terminological default knowledge "generally, heart patients suffer from high blood pressure" can be expressed by $(\exists\, \mathit{HasHighBloodPressure}.\{\mathit{Yes}\} \mid \mathit{HeartPatient})[1, 1]$. The assertional probabilistic knowledge "Tom is a pacemaker patient with a probability of at least 0.8" can be expressed by the conditional constraint $(\mathit{PacemakerPatient} \mid \top)[0.8, 1]$ for Tom. Here, the first two conditional constraints are default statements, while the third one is a strict statement. This different meaning is achieved by handling them differently when drawing conclusions (see Section 4.2.4).

A *PTBox* $PT = (T, P)$ consists of a classical (description logic) knowledge base T (as defined in Section 3.1) and a finite set of conditional constraints P. Informally, P encodes terminological probabilistic knowledge as well as terminological default knowledge: Every $(\psi|\phi)[l, u] \in P$ encodes that "generally, if $\phi(o)$ holds, then $\psi(o)$ holds with a probability between l and u", for every randomly chosen individual o. In particular, $(\exists R.\{o'\}|\phi)[l, u] \in P$, where o' is a classical individual from $\mathbf{I}_C$, and R is a role from $\mathbf{R}_A$, encodes that "generally, if $\phi(o)$ holds, then $R(o, o')$ holds with a probability between l and u", for every randomly chosen individual o.

Example 4.3 *(Medical Example cont'd)* The PTBox of the Medical Example contains in particular $(\mathit{MalePacemakerPatient} \,|\, \mathit{PacemakerPatient})[0.4, 1]$ and $(\exists\, \mathit{HasHighBloodPressure}.\{\mathit{Yes}\} \,|\, \mathit{HeartPatient})[1, 1]$ of Example 4.2.

A *PABox* P_o for a probabilistic individual $o \in \mathbf{I}_P$ is a finite set of conditional constraints. Informally, every $(\psi|\phi)[l, u] \in P_o$ encodes that "if $\phi(o)$ holds, then $\psi(o)$ holds with a probability between l and u". In particular, $(\psi|\top)[l, u] \in P_o$ expresses that "$\psi(o)$ holds with a probability between l and u", while $(\exists R.\{o'\} \mid \phi)[l, u] \in P_o$, where o' is a classical individual from $\mathbf{I}_C$, and R is a role from $\mathbf{R}_A$, encodes that "if $\phi(o)$ holds, then $R(o, o')$ holds with a probability between l and u". Hence, differently from the above terminological probabilistic sentences in P, the assertional probabilistic sentences in P_o refer to the concrete probabilistic individual $o \in \mathbf{I}_P$.

Example 4.4 *(Medical Example cont'd)* The PABox for the probabilistic individual Tom contains in particular $(\mathit{PacemakerPatient} \,|\, \top)[0.8, 1]$ of Example 4.2.

A *probabilistic (description logic) knowledge base* $KB = (T, P, (P_o)_{o \in \mathbf{I}_P})$ relative to $\mathbf{I}_P$ consists of a PTBox $PT = (T, P)$ and one PABox P_o for every probabilistic individual $o \in \mathbf{I}_P$. Informally, a probabilistic knowledge base extends a classical knowledge base T by terminological probabilistic knowledge P and assertional probabilistic knowledge P_o about every $o \in \mathbf{I}_P$. As for the semantics (which is formally defined in Section 4.2 below), we interpret P as probabilistic knowledge about randomly chosen individuals, while every P_o is interpreted as probabilistic knowledge about the concrete individual o. Notice also that the axioms in T and the conditional constraints in every PABox P_o with $o \in \mathbf{I}_P$ are strict statements (that is, they must always hold), while the conditional constraints in P are default statements (that is, they may have exceptions and thus do not always have to hold).

Example 4.5 *(Medical Example cont'd)* We extend the classical knowledge base of Example 3.1 by additional axioms, terminological default knowledge, terminological probabilistic knowledge, and assertional probabilistic knowledge to a probabilistic knowledge base $KB = (T, P, (P_o)_{o \in \mathbf{I}_P})$. We assume the additional atomic concept *PrivateHealthInsurance* and the additional datatype role *HasHighBloodPressure* between heart patients and the elementary datatype *Boolean* with its data values *Yes* and *No*. The following additional concept inclusion axiom in T expresses that (9) "heart patients cannot both suffer and not suffer from high blood pressure":

(9) $\exists\, \mathit{HasHighBloodPressure}.\top \sqsubseteq\ \leq 1\, \mathit{HasHighBloodPressure}$.

The terminological default knowledge (10) "generally, heart patients suffer from high blood pressure" and (11) "generally, pacemaker patients do not suffer from high blood pressure", and the terminological probabilistic knowledge (12) "generally, pacemaker patients are male with a probability of at least 0.4", (13) "generally, heart patients have a private health insurance with a probability of at least 0.9", and (14) "generally, pacemaker patients have the symptoms arrhythmia with a probability of at least 0.98, chest pain with a probability of at least 0.9, and breathing difficulties with a probability of at least 0.6" can be expressed by the following conditional constraints in P:

(10) $(\exists\, \mathit{HasHighBloodPressure}.\{\mathit{Yes}\} \,|\, \mathit{HeartPatient})[1, 1]$,

(11) $(\exists\, \mathit{HasHighBloodPressure}.\{\mathit{No}\} \,|\, \mathit{PacemakerPatient})[1, 1]$,

(12) $(\mathit{MalePacemakerPatient} \,|\, \mathit{PacemakerPatient})[0.4, 1]$,

(13) $(\exists\, \mathit{HasHealthInsurance}.\mathit{PrivateHealthInsurance} \,|\, \mathit{HeartPatient})[0.9, 1]$,

(14) $(\exists\, \mathit{HasIllnessSymptom}.\{\mathit{Arrhythmia}\} \,|\, \mathit{PacemakerPatient})[0.98, 1]$,
$(\exists\, \mathit{HasIllnessSymptom}.\{\mathit{ChestPain}\} \,|\, \mathit{PacemakerPatient})[0.9, 1]$,
$(\exists\, \mathit{HasIllnessSymptom}.\{\mathit{BreathingDifficulties}\} \,|\, \mathit{PacemakerPatient})[0.6, 1]$.

The set of probabilistic individuals is given by $\mathbf{I}_P = \{\mathit{Tom}, \mathit{John}, \mathit{Maria}\}$. The assertional probabilistic knowledge (15) "Tom is a pacemaker patient with a probability of at least 0.8" can be expressed by the following conditional constraint in P_{Tom}:

(15) $(\mathit{PacemakerPatient} \,|\, \top)[0.8, 1]$.

The assertional probabilistic knowledge (16) "Maria has the symptom breathing difficulties with a probability of at least 0.6", (17) "Maria has the symptom chest pain with a probability of at least 0.9", and (18) "Maria's illness status is final with a probability between 0.2 and 0.8" can finally be expressed by the following conditional constraints in P_{Maria}:

(16) $(\exists\, \mathit{HasIllnessSymptom}.\{\mathit{BreathingDifficulties}\} \,|\, \top)[0.6, 1]$,

(17) $(\exists\, \mathit{HasIllnessSymptom}.\{\mathit{ChestPain}\} \,|\, \top)[0.9, 1]$,

(18) $(\exists\, \mathit{HasIllnessStatus}.\{\mathit{Final}\} \,|\, \top)[0.2, 0.8]$.

4.2 Semantics

We now define the semantics of probabilistic knowledge bases in P-$\mathcal{SHIF}(\mathbf{D})$ and P-$\mathcal{SHOIN}(\mathbf{D})$. After some preliminaries, we introduce the notions of consistency and lexicographic entailment for probabilistic knowledge bases, which are based on the notions of consistency and lexicographic entailment, respectively, in probabilistic default reasoning (see [54, 56] for further details, motivation, and examples on probabilistic lexicographic entailment), which are in turn probabilistic generalizations of the notions of consistency and Lehmann's lexicographic entailment [51] in default reasoning from conditional knowledge bases (see Section 7.2 for further details on default reasoning from conditional knowledge bases).

4.2.1 Motivation and Key Ideas

There are essentially two different forms of conclusions that we want to draw from probabilistic knowledge bases $KB = (T, P, (P_o)_{o \in \mathbf{I}_P})$. First, we want to derive new terminological probabilistic knowledge from the PTBox $PT = (T, P)$. Second, given a probabilistic individual $o \in \mathbf{I}_P$, we want to derive new assertional probabilistic knowledge about o from the combination of the PTBox $PT = (T, P)$ and the PABox P_o. To carry out both these forms of conclusions, however, we may have to resolve contextual inconsistencies inside the terminological knowledge and between the terminological knowledge and the assertional knowledge about the individual o (intuitively, o may not necessarily be a typical individual).

Example 4.6 *(Medical Example cont'd)* Observe that the terminological default knowledge "generally, heart patients suffer from high blood pressure" is inconsistent with the terminological default knowledge "generally, pacemaker patients do not suffer from high blood pressure" in the context of pacemaker patients, given the terminological classical knowledge "all pacemaker patients are heart patients", since it is unclear at first sight which of the two contradicting default statements should be applied to pacemaker patients.

Such contextual inconsistencies are resolved by using the rule of maximum specificity, that is, by preferring more specific pieces of knowledge to less specific ones.

Example 4.7 *(Medical Example cont'd)* Applying the rule of specificity, the contextual inconsistency described in Example 4.6 is resolved by ignoring the terminological default knowledge "generally, heart patients suffer from high blood pressure", which is less specific than the terminological default knowledge for pacemaker patients, since all pacemaker patients are heart patients.

Hence, when drawing conclusions from probabilistic knowledge bases $KB = (T, P, (P_o)_{o \in \mathbf{I}_P})$, we first have to characterize the specificity of each conditional constraint in P. These specificities define a preference relation between all subsets of P, which in turn can be extended to a preference relation between all probabilistic interpretations. We then draw our conclusions from the preferred subsets of P under T (resp., T and P_o), or equivalently from the preferred probabilistic interpretations that satisfy T (resp., T and P_o). In some cases, the rule of maximum specificity is insufficient to resolve all contextual inconsistencies. This is when PT (resp., KB) is inconsistent.

4.2.2 Preliminaries

We now define (possible) worlds as certain sets of basic c-concepts, and probabilistic interpretations as probability functions on the set of all (possible) worlds. We also define the satisfaction of classical knowledge bases and conditional constraints in probabilistic interpretations.

A *(possible) world* I is a set of basic c-concepts $\phi \in \mathcal{C}$ such that $\{\phi(i) \mid \phi \in I\} \cup \{\neg\phi(i) \mid \phi \in \mathcal{C} \setminus I\}$ is satisfiable, where i is a new individual. Informally, every world I represents an individual i that is fully specified on $\mathcal{C}$ in the sense that I belongs (resp., does not belong) to every c-concept $\phi \in I$ (resp., $\phi \in \mathcal{C} \setminus I$). We denote by $\mathcal{I}_{\mathcal{C}}$ the set of all worlds relative to $\mathcal{C}$. Notice that $\mathcal{I}_{\mathcal{C}}$ is finite, since $\mathcal{C}$ is finite. A world I *satisfies* a classical knowledge base T, or I is a *model* of T, denoted $I \models T$, iff $T \cup \{\phi(i) \mid \phi \in I\} \cup \{\neg\phi(i) \mid \phi \in \mathcal{C} \setminus I\}$ is satisfiable, where i is a new individual. A world I *satisfies* a basic c-concept $\phi \in \mathcal{C}$, or I is a *model* of ϕ, denoted $I \models \phi$, iff $\phi \in I$. The satisfaction of c-concepts by worlds is inductively extended to all c-concepts, as usual, by (i) $I \models \neg\phi$ iff $I \models \phi$ does not hold, and (ii) $I \models \phi \sqcap \psi$ iff $I \models \phi$ and $I \models \psi$. The following proposition shows that, for classical knowledge bases T, the notion of satisfiability based on worlds I is compatible with the notion of satisfiability based on classical description logic interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$. That is, there exists a classical interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ that satisfies T iff there exists a world $I \in \mathcal{I}_{\mathcal{C}}$ that satisfies T.

Proposition 4.8 *Let $\mathcal{C} \neq \emptyset$ be a finite set of basic c-concepts, and let T be a classical knowledge base. Then, T has a model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ iff T has a model $I \in \mathcal{I}_{\mathcal{C}}$.*

A *probabilistic interpretation* Pr is a probability function on $\mathcal{I}_{\mathcal{C}}$ (that is, a mapping $Pr\colon \mathcal{I}_{\mathcal{C}} \to [0,1]$ such that all $Pr(I)$ with $I \in \mathcal{I}_{\mathcal{C}}$ sum up to 1). We say Pr *satisfies* a classical knowledge base T, or Pr is a *model* of T, denoted $Pr \models T$, iff $I \models T$ for every $I \in \mathcal{I}_{\mathcal{C}}$ such that $Pr(I) > 0$. We define the probability of a c-concept and the satisfaction of conditional constraints in probabilistic interpretations as follows. The *probability* of a c-concept ϕ in a probabilistic interpretation Pr, denoted $Pr(\phi)$, is the sum of all $Pr(I)$ such that $I \models \phi$. For c-concepts ϕ and ψ such that $Pr(\phi) > 0$, we write $Pr(\psi|\phi)$ to abbreviate $Pr(\phi \sqcap \psi)\ /\ Pr(\phi)$. We say Pr *satisfies* a conditional constraint $(\phi|\psi)[l,u]$, or Pr is a *model* of $(\psi|\phi)[l,u]$, denoted $Pr \models (\psi|\phi)[l,u]$, iff $Pr(\phi) = 0$ or $Pr(\psi|\phi) \in [l,u]$. We say Pr *satisfies* a set of conditional constraints $\mathcal{F}$, or Pr is a *model* of $\mathcal{F}$, denoted $Pr \models \mathcal{F}$, iff $Pr \models F$ for all $F \in \mathcal{F}$. The next proposition shows that, for classical knowledge bases T, the notion of satisfiability based on probabilistic interpretations Pr is compatible with the notion of satisfiability based on classical interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$. That is, T has a satisfying classical interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ iff it has a satisfying probabilistic interpretation Pr. The result follows from Proposition 4.8.

Proposition 4.9 *Let $\mathcal{C} \neq \emptyset$ be a finite set of basic c-concepts, and let T be a classical knowledge base. Then, T has a model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ iff T has a model Pr on $\mathcal{I}_{\mathcal{C}}$.*

4.2.3 Consistency

In this section, we define the notion of consistency for PTBoxes and probabilistic knowledge bases, which is based on the notion of consistency in probabilistic default reasoning [54, 56]. The latter in turn generalizes the notion of consistency in default reasoning from conditional knowledge bases [1, 32].

We first give some preparative definitions (which generalize the notions of verification, falsification, and toleration in default reasoning from conditional knowledge bases [1, 32] to the framework of probabilistic description logics). A probabilistic interpretation Pr *verifies* a conditional constraint $(\psi|\phi)[l,u]$ iff $Pr(\phi) = 1$ and $Pr(\psi) \in [l,u]$. Notice that the latter is equivalent to $Pr(\phi) = 1$ and $Pr \models (\psi|\phi)[l,u]$. We say that Pr *falsifies* $(\psi|\phi)[l,u]$ iff $Pr(\phi) = 1$ and $Pr \not\models (\psi|\phi)[l,u]$. A set of conditional constraints $\mathcal{F}$ *tolerates* a conditional constraint $(\psi|\phi)[l,u]$ under a classical knowledge base T, or $(\psi|\phi)[l,u]$ is *tolerated* under T by $\mathcal{F}$, iff $T \cup \mathcal{F}$ has a model that verifies $(\psi|\phi)[l,u]$ (which is equivalent to the existence of a model of $T \cup \mathcal{F} \cup \{(\psi|\phi)[l,u], (\phi|\top)[1,1]\}$).

A PTBox $PT = (T, P)$ is *consistent* iff (i) T is satisfiable and (ii) there exists an ordered partition $(P_0, \ldots, P_k)$ of P such that each P_i with $i \in \{0, \ldots, k\}$ is the set of all $F \in P \setminus (P_0 \cup \cdots \cup P_{i-1})$ that are tolerated under T by $P \setminus (P_0 \cup \cdots \cup P_{i-1})$. Informally, a PTBox is consistent iff all contained contextual inconsistencies can be resolved by applying the rule of maximum specificity. More concretely, (ii) means that P has a natural partition into collections $P_0, \ldots, P_k$ of conditional constraints of increasing specificities such that no collection P_i is contextually inconsistent. That is, contextual inconsistencies can only occur between two different collections P_i and P_j, but not inside a single collection P_i. Such contextual inconsistencies between two different collections can then be resolved by preferring more specific collections and their elements to less specific ones.

Example 4.10 *(Medical Example cont'd)* We partition P into two collections of conditional constraints P_0 and P_1 such that $(\exists\, \mathit{HasHighBloodPressure}.\{\mathit{Yes}\}|\mathit{HeartPatient})[1,1] \in P_0$ and $(\exists\, \mathit{HasHighBloodPressure}.\{\mathit{No}\}\,|\,\mathit{PacemakerPatient})[1,1] \in P_1$. The contextual inconsistency between the two when reasoning about pacemaker patients is then resolved by preferring the latter to the former.

We call the above ordered partition $(P_0, \ldots, P_k)$ of P the *z-partition* of PT. A probabilistic knowledge base $KB = (T, P, (P_o)_{o \in \mathbf{I}_P})$ is *consistent* iff $PT = (T, P)$ is consistent and $T \cup P_o$ is satisfiable for all probabilistic individuals $o \in \mathbf{I}_P$. Informally, the latter says that the strict knowledge in T must be compatible with the strict degrees of belief in P_o, for every probabilistic individual o.

Example 4.11 *(Medical Example cont'd)* The probabilistic knowledge base $KB = (T, P, (P_o)_{o \in \mathbf{I}_P})$ of Example 4.5 is consistent, since $PT = (T, P)$ is consistent, and $T \cup P_o$ is satisfiable for every $o \in \mathbf{I}_P = \{\mathit{Tom}, \mathit{John}, \mathit{Maria}\}$. Notice that the z-partition of (T, P) is given by (P_0, P_1), where $P_0 = \{(\psi|\phi)[l, u] \in P \mid \phi = \mathit{HeartPatient}\}$ and $P_1 = \{(\psi|\phi)[l, u] \in P \mid \phi = \mathit{PacemakerPatient}\}$.

The following theorem provides an alternative characterization of consistency.

Theorem 4.12 *A PTBox $PT = (T, P)$ is consistent iff (i) T is satisfiable and (ii) there exists an ordered partition $(P_0, \ldots, P_k)$ of P such that every $F \in P_i$, $i \in \{0, \ldots, k\}$, is tolerated under T by $\bigcup_{j=i}^{k} P_i$.*

4.2.4 Lexicographic Entailment

The notion of lexicographic entailment for probabilistic knowledge bases is based on lexicographic entailment in probabilistic default reasoning [54, 56], which in turn generalizes Lehmann's lexicographic entailment [51] in default reasoning from conditional knowledge bases. In the sequel, let $KB = (T, P, (P_o)_{o \in \mathbf{I}_P})$ be a consistent probabilistic knowledge base. We first define a lexicographic preference relation on probabilistic interpretations, which is then used to define the notion of lexicographic entailment for sets of conditional constraints under PTBoxes. We finally define the notion of lexicographic entailment for deriving terminological probabilistic knowledge and assertional probabilistic knowledge about probabilistic individuals from PTBoxes and probabilistic knowledge bases, respectively.

We use the z-partition $(P_0, \ldots, P_k)$ of (T, P), which partitions P into collections $P_0, \ldots, P_k$ of conditional constraints of increasing specificities, to define a lexicographic preference relation on probabilistic interpretations. For probabilistic interpretations Pr and Pr', we say Pr is *lexicographically preferable* (or *lex-preferable*) to Pr' iff some $i \in \{0, \ldots, k\}$ exists such that $|\{F \in P_i \mid Pr \models F\}| > |\{F \in P_i \mid Pr' \models F\}|$ and $|\{F \in P_j \mid Pr \models F\}| = |\{F \in P_j \mid Pr' \models F\}|$ for all $i < j \leq k$. Intuitively, this lexicographic preference relation on probabilistic interpretations implements the idea of preferring more specific collections of conditional constraints P_i and their elements to less specific ones P_j and their elements. It can thus be used for preferring more specific pieces of knowledge to less specific ones when drawing conclusions in the case of contextual inconsistencies. A model Pr of a classical knowledge base T and a set of conditional constraints $\mathcal{F}$ is a *lexicographically minimal* (or *lex-minimal*) *model* of $T \cup \mathcal{F}$ iff no model of $T \cup \mathcal{F}$ is lex-preferable to Pr.

Example 4.13 *(Medical Example cont'd)* Recall from Example 4.11 that the z-partition of the Medical Example is given by (P_0, P_1), where $P_0 = \{(\psi|\phi)[l, u] \in P \mid \phi = \mathit{HeartPatient}\}$ and $P_1 = \{(\psi|\phi)[l, u] \in P \mid \phi = \mathit{PacemakerPatient}\}$. When reasoning about pacemaker patients, every probabilistic interpretation Pr that satisfies P_1 is lex-preferable to any probabilistic interpretation Pr' that does not satisfy P_1. So, the more specific $(\exists \mathit{HasHighBloodPressure}.\{\mathit{No}\} \mid \mathit{PacemakerPatient})[1, 1]$ is preferred to the less specific $(\exists \mathit{HasHighBloodPressure}.\{\mathit{Yes}\} \mid \mathit{HeartPatient})[1, 1]$.

We define the notion of lexicographic entailment of conditional constraints from sets of conditional constraints under PTBoxes as follows. A conditional constraint $(\psi|\phi)[l, u]$ is a *lexicographic consequence* (or *lex-consequence*) of a set of conditional constraints $\mathcal{F}$ under a PTBox PT, denoted $\mathcal{F} \,\|\!\sim^{lex} (\psi|\phi)[l,u]$ under PT, iff $Pr(\psi) \in [l,u]$ for every lex-minimal model Pr of $T \cup \mathcal{F} \cup \{(\phi|\top)[1,1]\}$. We say $(\psi|\phi)[l,u]$ is a *tight lexicographic consequence* (or *tight lex-consequence*) of $\mathcal{F}$ under PT, denoted $\mathcal{F} \,\|\!\sim^{lex}_{tight} (\psi|\phi)[l,u]$ under PT, iff l (resp., u) is the infimum (resp., supremum) of $Pr(\psi)$ subject to all lex-minimal models Pr of $T \cup \mathcal{F} \cup \{(\phi|\top)[1,1]\}$. Observe that $[l,u] = [1,0]$ (where $[1,0]$ represents the empty interval) when no such model Pr exists (since $\inf \emptyset$ and $\sup \emptyset$ are formally defined as $\max([0,1]) = 1$ and $\min([0,1]) = 0$, respectively). Furthermore, for inconsistent PTBoxes PT, we define $\mathcal{F} \,\|\!\sim^{lex} (\psi|\phi)[l,u]$ and $\mathcal{F} \,\|\!\sim^{lex}_{tight} (\psi|\phi)[1,0]$ under PT for all sets of conditional constraints $\mathcal{F}$ and all conditional constraints $(\psi|\phi)[l,u]$.

We now define which terminological and assertional probabilistic knowledge is lexicographically entailed by PTBoxes resp. probabilistic knowledge bases. A conditional constraint F is a *lex-consequence* of a PTBox PT, denoted $PT \,\|\!\sim^{lex} F$, iff $\emptyset \,\|\!\sim^{lex} F$ under PT. We say F is a *tight lex-consequence* of PT, denoted $PT \,\|\!\sim^{lex}_{tight} F$, iff $\emptyset \,\|\!\sim^{lex}_{tight} F$ under PT. A conditional constraint F for a probabilistic individual $o \in \mathbf{I}_P$ is a *lex-consequence* of a probabilistic knowledge base $KB = (T, P, (P_o)_{o \in \mathbf{I}_P})$, denoted $KB \,\|\!\sim^{lex} F$, iff $P_o \,\|\!\sim^{lex} F$ under (T,P). We say F is a *tight lex-consequence* of KB, denoted $KB \,\|\!\sim^{lex}_{tight} F$, iff $P_o \,\|\!\sim^{lex}_{tight} F$ under (T,P).

Example 4.14 *(Medical Example cont'd)* Consider again the probabilistic knowledge base $KB = (T, P, (P_o)_{o \in \mathbf{I}_P})$ given in Example 4.5. The following conditional constraints are some (terminological default and terminological probabilistic) tight lex-consequences of $PT = (T,P)$:

$(\exists \mathit{HasHighBloodPressure}.\{\mathit{Yes}\} \mid \mathit{HeartPatient})[1,1]$,
$(\exists \mathit{HasHighBloodPressure}.\{\mathit{No}\} \mid \mathit{PacemakerPatient})[1,1]$,
$(\exists \mathit{HasHighBloodPressure}.\{\mathit{Yes}\} \mid \mathit{Male} \sqcap \mathit{HeartPatient})[1,1]$,
$(\mathit{MalePacemakerPatient} \mid \mathit{PacemakerPatient})[0.4,1]$,
$(\exists \mathit{HasHealthInsurance}.\mathit{PrivateHealthInsurance} \mid \mathit{HeartPatient})[0.9,1]$,
$(\exists \mathit{HasHealthInsurance}.\mathit{PrivateHealthInsurance} \mid \mathit{PacemakerPatient})[0.9,1]$,
$(\exists \mathit{HasIllnessSymptom}.\{\mathit{Arrhythmia}\} \mid \mathit{PacemakerPatient})[0.98,1]$,
$(\exists \mathit{HasIllnessSymptom}.\{\mathit{ChestPain}\} \mid \mathit{PacemakerPatient})[0.9,1]$,
$(\exists \mathit{HasIllnessSymptom}.\{\mathit{BreathingDifficulties}\} \mid \mathit{PacemakerPatient})[0.6,1]$.

So, the default property of having high blood pressure is inherited from heart patients down to male heart patients, and the probabilistic property of having a private health insurance with a probability of at least 0.9 is inherited from heart patients down to pacemaker patients. Roughly, the tight lex-consequences of $PT = (T,P)$ are given by all those conditional constraints that (a) are either in P, or (b) can be constructed by inheritance along subconcept relationships from the ones in P and are not overridden by more specific pieces of knowledge in P.

The following conditional constraints for the probabilistic individual *Tom* are some (assertional probabilistic) tight lex-consequences of $KB = (T, P, (P_o)_{o \in \mathbf{I}_P})$:

$(\mathit{PacemakerPatient} \mid \top)[0.8,1]$,
$(\exists \mathit{HasHighBloodPressure}.\{\mathit{No}\} \mid \top)[0.8,1]$,
$(\mathit{MalePacemakerPatient} \mid \top)[0.32,1]$,
$(\exists \mathit{HasHealthInsurance}.\mathit{PrivateHealthInsurance} \mid \top)[0.72,1]$,

$(\exists\, \mathit{HasIllnessSymptom}.\{\mathit{Arrhythmia}\} \mid \top)[0.78, 1]$,
$(\exists\, \mathit{HasIllnessSymptom}.\{\mathit{ChestPain}\} \mid \top)[0.72, 1]$,
$(\exists\, \mathit{HasIllnessSymptom}.\{\mathit{BreathingDifficulties}\} \mid \top)[0.48, 1]$.

We finally provide a characterization of the notion of lexicographic entailment for a set of conditional constraints under a PTBox in terms of the notions of satisfiability and logical entailment for a set of conditional constraints under a classical knowledge base, which are defined as follows. Given a classical knowledge base T and a set of conditional constraints $\mathcal{F}$, we say $T \cup \mathcal{F}$ is *satisfiable* iff a model of $T \cup \mathcal{F}$ exists. A conditional constraint $(\psi|\phi)[l, u]$ is a *logical consequence* of $T \cup \mathcal{F}$, denoted $T \cup \mathcal{F} \models (\psi|\phi)[l, u]$, iff each model of $T \cup \mathcal{F}$ is also a model of $(\psi|\phi)[l, u]$. We say $(\psi|\phi)[l, u]$ is a *tight logical consequence* of $T \cup \mathcal{F}$, denoted $T \cup \mathcal{F} \models_{tight} (\psi|\phi)[l, u]$, iff l (resp., u) is the infimum (resp., supremum) of $Pr(\psi|\phi)$ subject to all models Pr of $T \cup \mathcal{F}$ with $Pr(\phi) > 0$.

The characterization of lexicographic entailment is formally expressed by the subsequent theorem. More concretely, given a PTBox $PT = (T, P)$, a set of conditional constraints $\mathcal{F}$, and two c-concepts ϕ and ψ, the key idea behind the characterization is that a set $\mathcal{Q}$ of lexicographically minimal subsets of P exists such that $\mathcal{F} \,\|\!\sim^{lex} (\psi|\phi)[l, u]$ under PT iff $T \cup Q \cup \mathcal{F} \cup \{(\phi|\top)[1, 1]\} \models (\psi|\top)[l, u]$ for all $Q \in \mathcal{Q}$. Here, $Q \subseteq P$ is *lexicographically preferable* (or *lex-preferable*) to $Q' \subseteq P$ iff some $i \in \{0, \ldots, k\}$ exists such that $|Q \cap P_i| > |Q' \cap P_i|$ and $|Q \cap P_j| = |Q' \cap P_j|$ for all $i < j \leq k$, where $(P_0, \ldots, P_k)$ denotes the z-partition of PT. We say Q is *lexicographically minimal* (or *lex-minimal*) in a set $\mathcal{S}$ of subsets of P iff $Q \in \mathcal{S}$ and no $Q' \in \mathcal{S}$ is lex-preferable to Q.

Theorem 4.15 *Let $PT = (T, P)$ be a consistent PTBox, let $\mathcal{F}$ be a set of conditional constraints, and let ϕ and ψ be two c-concepts. Let $\mathcal{Q}$ be the set of all lex-minimal elements in the set of all $Q \subseteq P$ such that $T \cup Q \cup \mathcal{F} \cup \{(\phi|\top)[1, 1]\}$ is satisfiable. Then,*

(a) If $\mathcal{Q} = \emptyset$, then $\mathcal{F} \,\|\!\sim^{lex}_{tight} (\psi|\phi)[1, 0]$ under PT.

(b) If $\mathcal{Q} \neq \emptyset$, then $\mathcal{F} \,\|\!\sim^{lex}_{tight} (\psi|\phi)[l, u]$ under PT, where $l = \min l'$ (resp., $u = \max u'$) subject to $T \cup Q \cup \mathcal{F} \cup \{(\phi|\top)[1, 1]\} \models_{tight} (\psi|\top)[l', u']$ and $Q \in \mathcal{Q}$.

5 Algorithms

In this section, we provide algorithms for the main reasoning tasks in P-$\mathcal{SHIF}(\mathbf{D})$ (resp., P-$\mathcal{SHOIN}(\mathbf{D})$). They are based on reductions to deciding the satisfiability of classical knowledge bases in $\mathcal{SHIF}(\mathbf{D})$ (resp., $\mathcal{SHOIN}(\mathbf{D})$), deciding the solvability of systems of linear constraints, and computing the optimal value of linear programs. This shows in particular that the main reasoning tasks in P-$\mathcal{SHIF}(\mathbf{D})$ (resp., P-$\mathcal{SHOIN}(\mathbf{D})$) are decidable resp. computable.

5.1 Problem Statements

The main reasoning tasks related to PTBoxes and probabilistic knowledge bases in P-$\mathcal{SHIF}(\mathbf{D})$ and P-$\mathcal{SHOIN}(\mathbf{D})$ are summarized by the following decision and computation problems (where we assume that every lower and upper bound in the PTBox $PT = (T, P)$, the probabilistic knowledge base $KB = (T, P, (P_o)_{o \in \mathbf{I}_P})$, and the set of conditional constraints $\mathcal{F}$ is rational):

PTBOX CONSISTENCY (PTCON): Decide whether a given PTBox $PT = (T, P)$ is consistent.

PROBABILISTIC KNOWLEDGE BASE CONSISTENCY (PKBCON): Given a probabilistic knowledge base $KB = (T, P, (P_o)_{o \in \mathbf{I}_P})$, decide whether KB is consistent.

TIGHT LEXICOGRAPHIC ENTAILMENT (TLEXENT): Given a PTBox $PT = (T, P)$, a finite set of conditional constraints $\mathcal{F}$, and two c-concepts ϕ and ψ, compute the rational numbers $l, u \in [0, 1]$ such that $\mathcal{F} \Vdash\!\sim^{lex}_{tight} (\psi|\phi)[l, u]$ under PT.

Some important special cases of TLEXENT are given as follows: (PCSUB) given a consistent PTBox PT and two c-concepts ϕ and ψ, compute $l, u \in [0, 1]$ such that $PT \Vdash\!\sim^{lex}_{tight} (\psi|\phi)[l, u]$; (PCRSUB) given a consistent PTBox PT, a c-concept ϕ, a classical individual $o \in \mathbf{I}_C$, and an abstract role $R \in \mathbf{R}_A$, compute $l, u \in [0, 1]$ such that $PT \Vdash\!\sim^{lex}_{tight} (\exists R.\{o\}|\phi)[l, u]$; (PCMEM) given a consistent probabilistic knowledge base KB, a probabilistic individual $o \in \mathbf{I}_P$, and a c-concept ψ, compute $l, u \in [0, 1]$ such that $KB \Vdash\!\sim^{lex}_{tight} (\psi|\top)[l, u]$ for o; and (PRMEM) given a consistent probabilistic knowledge base KB, a classical individual $o' \in \mathbf{I}_C$, a probabilistic individual $o \in \mathbf{I}_P$, and an abstract role $R \in \mathbf{R}_A$, compute $l, u \in [0, 1]$ such that $KB \Vdash\!\sim^{lex}_{tight} (\exists R.\{o'\}|\top)[l, u]$ for o.

Another important decision problem is PROBABILISTIC CONCEPT SATISFIABILITY (PCSAT): Given a consistent PTBox PT and a c-concept ϕ, decide whether $PT \not\Vdash\!\sim^{lex} (\phi|\top)[0, 0]$. This problem is reducible to CSAT (see Section 3.2), since $(T, P) \not\Vdash\!\sim^{lex} (\phi|\top)[0, 0]$ iff $T \not\models \phi \sqsubseteq \bot$.

In Section 5.2 below, we show that the above problems PTCON, PKBCON, and TLEXENT can all be reduced to the following two decision and computation problems (where we assume that every lower and upper bound in the set of conditional constraints $\mathcal{F}$ is rational):

SATISFIABILITY (SAT): Given a classical description logic knowledge base T and a finite set of conditional constraints $\mathcal{F}$, decide whether $T \cup \mathcal{F}$ is satisfiable.

TIGHT LOGICAL ENTAILMENT (TLOGENT): Given a classical description logic knowledge base T, a finite set of conditional constraints $\mathcal{F}$, and a c-concept ψ, compute the rational numbers $l, u \in [0, 1]$ such that $T \cup \mathcal{F} \models_{tight} (\psi|\top)[l, u]$.

5.2 Consistency and Tight Lexicographic Entailment

We now present algorithms for solving PTCON, PKBCON, and TLEXENT. These algorithms are all based on reductions to the problems SAT and TLOGENT.

Algorithm *pt-consistency* in Fig. 1 decides whether a given PTBox $PT = (T, P)$ is consistent. More precisely, it returns the z-partition of PT, if PT is consistent, and *nil*, otherwise. In lines 1 and 2, the algorithm handles the case where T is unsatisfiable or P is empty. In lines 3–11, it computes and returns the z-partition of PT (if it exists). Algorithm *pt-consistency* is a generalization of an algorithm for deciding consistency in probabilistic default reasoning with conditional constraints [56], which in turn is a generalization of an algorithm for deciding ε-consistency in default reasoning from conditional knowledge bases [32].

Algorithm *pkb-consistency* in Fig. 2 decides whether a given probabilistic knowledge base $KB = (T, P, (P_o)_{o \in \mathbf{I}_P})$ is consistent. In line 1, it decides whether (T, P) is consistent, and in lines 2 and 3, whether every $T \cup P_o$ with $o \in \mathbf{I}_P$ is satisfiable.

Finally, given a PTBox $PT = (T, P)$, a finite set of conditional constraints $\mathcal{F}$, and two c-concepts ϕ and ψ, Algorithm *tight-lex-entailment* in Fig. 3 computes $l, u \in [0, 1]$ such that $\mathcal{F} \Vdash\!\sim^{lex}_{tight} (\psi|\phi)[l, u]$ under PT. It is based on Theorem 4.15. In lines 1–3, it handles the case where PT is inconsistent or $T \cup \mathcal{F} \cup \{(\phi|\top)[1, 1]\}$

Algorithm *pt-consistency*

Input: PTBox $PT = (T, P)$.
Output: z-partition of PT, if PT is consistent; *nil*, otherwise.

if T is unsatisfiable **then return** *nil*;
if $P = \emptyset$ **then return** $()$;
$H := P$;
$i := -1$;
repeat
    $i := i + 1$;
    $P[i] := \{C \in H \mid C$ is tolerated under T by $H\}$;
    $H := H \setminus P[i]$
until $H = \emptyset$ **or** $P[i] = \emptyset$;
if $H = \emptyset$ **then return** $(P[0], \ldots, P[i])$
    else return *nil*.

Figure 1: Algorithm *pt-consistency*.

Algorithm *pkb-consistency*

Input: probabilistic knowledge base $KB = (T, P, (P_o)_{o \in \mathbf{I}_P})$.
Output: "Yes", if KB is consistent; "No", otherwise.

if (T, P) is inconsistent **then return** "No";
for each $o \in \mathbf{I}_P$ **do**
    if $T \cup P_o$ is unsatisfiable **then return** "No";
return "Yes".

Figure 2: Algorithm *pkb-consistency*.

is unsatisfiable. In lines 4–13, it computes the set $\mathcal{Q}$ of all lex-minimal elements among all $S \subseteq P$ such that $T \cup S \cup \mathcal{F} \cup \{(\phi|\top)[1,1]\}$ is satisfiable. In lines 14–19, it then computes the tight lex-entailed interval from $\mathcal{Q}$ and returns this interval.

Algorithms *pt-consistency*, *pkb-consistency*, and *tight-lex-entailment* reduce the problems PTCon, PKBCon, and TLexEnt, respectively, to the problems Sat and TLogEnt. The following theorem shows that *pt-consistency*, *pkb-consistency*, and *tight-lex-entailment* can be done by solving $O(|P|^2)$, $O(|P|^2 + |\mathbf{I}_P|)$, and $O(2^{|P|})$, respectively, instances of Sat and TLogEnt.

Theorem 5.1 *(a) Algorithm pt-consistency can be done by solving $O(|P|^2)$ instances of* Sat. *(b) Algorithm pkb-consistency can be done by solving $O(|P|^2 + |\mathbf{I}_P|)$ instances of* Sat. *(c) Algorithm tight-lex-entailment can be done by solving $O(2^{|P|})$ instances of* Sat *and* TLogEnt.

5.3 Satisfiability and Tight Logical Entailment

We now show that the problems Sat and TLogEnt can be reduced to deciding knowledge base satisfiability in $\mathcal{SHIF}(\mathbf{D})$ (resp., $\mathcal{SHOIN}(\mathbf{D})$), deciding the solvability of systems of linear constraints, and computing the optimal value of linear programs. These results are immediate by the fact that deciding satisfiability and

Algorithm *tight-lex-entailment*

Input: PTBox $PT = (T, P)$, finite set of conditional constraints $\mathcal{F}$,
and c-concepts ϕ and ψ.
Output: $(l, u) \in [0, 1]^2$ such that $\mathcal{F} \Vert\!\sim^{lex}_{tight} (\psi|\phi)[l, u]$ under PT.
Notation: $(P_0, \ldots, P_k)$ denotes the z-partition of PT.

if PT is inconsistent **then return** $(1, 0)$;
$R := T \cup \mathcal{F} \cup \{(\phi|\top)[1, 1]\}$;
if R is unsatisfiable **then return** $(1, 0)$;
$K := \{\emptyset\}$;
for $j := k$ **downto** 0 **do begin**
    $(m, n) := (0, |P_j|)$;
    while $m < n$ **do begin**
      $l := \lceil (m + n) / 2 \rceil$;
      $K' := \{G \cup H \mid G \subseteq P_j,\ |G| = l,\ H \in K,\ R \cup G \cup H$ is satisfiable$\}$;
      if $K' \neq \emptyset$ **then** $m := l$ **else** $n := l - 1$
    end;
    $K := \{G \cup H \mid G \subseteq P_j,\ |G| = m,\ H \in K,\ R \cup G \cup H$ is satisfiable$\}$
end;
$(l, u) := (1, 0)$;
for each $H \in K$ **do begin**
    compute $c, d \in [0, 1]$ such that $R \cup H \models_{tight} (\psi|\top)[c, d]$;
    $(l, u) := (\min(l, c), \max(u, d))$
end;
return (l, u).

Figure 3: Algorithm *tight-lex-entailment*.

computing tight intervals under logical entailment in probabilistic logic can be done by deciding whether a system of linear constraints is solvable and by computing the optimal values of two linear programs, respectively (see especially [35]).

The following theorem shows that the problem SAT can be reduced to deciding knowledge base satisfiability in $\mathcal{SHIF}(\mathbf{D})$ (resp., $\mathcal{SHOIN}(\mathbf{D})$) and deciding whether a system of linear constraints is solvable. Here (and in Theorem 5.3 below), deciding knowledge base satisfiability in $\mathcal{SHIF}(\mathbf{D})$ (resp., $\mathcal{SHOIN}(\mathbf{D})$) is used for computing the index set $R = \{I \in \mathcal{I}_{\mathcal{C}} \mid I \models T\}$, which can be done by deciding $|\mathcal{I}_{\mathcal{C}}|$ times whether some $T \cup \{\phi(i) \mid \phi \in I\} \cup \{\neg\phi(i) \mid \phi \in \mathcal{C} \setminus I\}$ with $I \in \mathcal{I}_{\mathcal{C}}$ is satisfiable.

Theorem 5.2 *Let T be a classical knowledge base, and let $\mathcal{F}$ be a finite set of conditional constraints. Let $R = \{I \in \mathcal{I}_{\mathcal{C}} \mid I \models T\}$. Then, $T \cup \mathcal{F}$ is satisfiable iff the system of linear constraints LC in Fig. 4 over the variables y_r ($r \in R$) is solvable.*

Finally, the following theorem shows that TLOGENT can be reduced to deciding knowledge base satisfiability in $\mathcal{SHIF}(\mathbf{D})$ (resp., $\mathcal{SHOIN}(\mathbf{D})$) and computing the optimal values of two linear programs.

Theorem 5.3 *Let T be a classical knowledge base, let $\mathcal{F}$ be a finite set of conditional constraints, and let ψ be a c-concept. Suppose that $T \cup \mathcal{F}$ is satisfiable. Let $R = \{I \in \mathcal{I}_{\mathcal{C}} \mid I \models T\}$. Then, l (resp., u) such that $T \cup \mathcal{F} \models_{tight} (\psi|\top)[l, u]$ is given by the optimal value of the following linear program over the*

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

Figure 4: System of linear constraints LC for Theorems 5.2 and 5.3.

variables y_r $(r\in R)$*:*

$$
\textit{minimize (resp., maximize)} \sum_{r\in R,\, r\models\psi} y_r \quad \textit{subject to } LC \textit{ in Fig. 4}. \tag{1}
$$

6 Complexity

In this section, we address the computational complexity of the main reasoning tasks in the probabilistic description logics P-$\mathcal{SHIF}(\mathbf{D})$ and P-$\mathcal{SHOIN}(\mathbf{D})$. We first recall some necessary complexity classes, and previous complexity results. Towards special cases of the main reasoning tasks in P-$\mathcal{SHIF}(\mathbf{D})$ and P-$\mathcal{SHOIN}(\mathbf{D})$ that have a lower computational complexity, we then introduce probabilistic knowledge bases in *DL-Lite*. We finally provide our complexity results.

6.1 Complexity Classes and Previous Results

We assume that the reader has some elementary background in complexity theory, the concepts of Turing machines and oracle calls, polynomial-time transformations among problems, and the hardness and completeness of a problem for a complexity class; see especially [47, 48, 66]. We now briefly recall the complexity classes that we encounter in our complexity results below. The class NP contains all decision problems that can be solved in polynomial time on a nondeterministic Turing machine, while the class EXP (resp., NEXP) contains all decision problems that can be solved in exponential time on a deterministic (resp., nondeterministic) Turing machine. The class P^{NP} (resp., $\mathrm{P}^{\mathrm{NEXP}}$) contains all problems that are decidable in polynomial time on a deterministic Turing machine with the help of an oracle for NP (resp., NEXP). The above complexity classes along with their inclusion relationships (all of which are currently believed to be strict) are summarized by:

$$
\mathrm{NP} \subseteq \mathrm{P}^{\mathrm{NP}} \subseteq \mathrm{EXP} \subseteq \mathrm{NEXP} \subseteq \mathrm{P}^{\mathrm{NEXP}}.
$$

For classifying problems that compute an output value, function classes similar to the classes above have been introduced [70, 47]. In particular, $\mathrm{FP}^{\mathrm{NP}}$, FEXP, and $\mathrm{FP}^{\mathrm{NEXP}}$ are the functional analogs of P^{NP}, EXP, and $\mathrm{P}^{\mathrm{NEXP}}$, respectively.

We recall that deciding if a knowledge base L in $\mathcal{SHIF}(\mathbf{D})$ (resp., $\mathcal{SHOIN}(\mathbf{D})$) is satisfiable is complete for EXP [73, 40] (resp., NEXP, assuming unary number encoding; see [40] and the NEXP-hardness proof for $\mathcal{ALCQI}$ in [73], which implies the NEXP-hardness of $\mathcal{SHOIN}(\mathbf{D})$). We also recall that deciding whether a finite set of conditional constraints in probabilistic logic is satisfiable and deciding whether a probabilistic default theory is σ-consistent are both NP-complete [55, 56]. Computing tight logically entailed intervals from a finite set of conditional constraints in probabilistic logic and computing tight lexicographically entailed intervals from a probabilistic default theory are both $\mathrm{FP}^{\mathrm{NP}}$-complete [55, 56].

6.2 The Description Logic *DL-Lite*

Inspired by classical description logic knowledge bases in *DL-Lite* [10], which are a restricted class of classical description logic knowledge bases for which deciding whether a knowledge base is satisfiable can be done in polynomial time, we now define a restricted class of probabilistic knowledge bases in *DL-Lite*.

We first recall *DL-Lite*. Let $\mathbf{A}$, $\mathbf{R}_A$, and $\mathbf{I}$ be pairwise disjoint finite nonempty sets of atomic concepts, abstract roles, and individuals, respectively. A *basic concept in DL-Lite* is either an atomic concept from $\mathbf{A}$ or an exists restriction on roles of the form $\exists R.\top$ (abbreviated as $\exists R$), where $R \in \mathbf{R}_A \cup \mathbf{R}_A^-$. *Concepts in DL-Lite* are defined by induction as follows. Every basic concept in *DL-Lite* is a concept in *DL-Lite*. If b is a basic concept in *DL-Lite*, and ϕ_1 and ϕ_2 are concepts in *DL-Lite*, then $\neg b$ and $\phi_1 \sqcap \phi_2$ are also concepts in *DL-Lite*. An *axiom in DL-Lite* is either (1) a concept inclusion axiom of the form $b \sqsubseteq \psi$, where b is a basic concept in *DL-Lite* and ϕ is a concept in *DL-Lite*, or (2) a *functionality axiom* $(\mathsf{funct}\ R)$, where $R \in \mathbf{R}_A \cup \mathbf{R}_A^-$, or (3) a concept membership axiom $b(a)$, where b is a basic concept in *DL-Lite* and $a \in \mathbf{I}$, or (4) a role membership axiom $R(a, c)$, where $R \in \mathbf{R}_A$ and $a, c \in \mathbf{I}$. A *knowledge base in DL-Lite* is a finite set of axioms in *DL-Lite*. We recall the following result from [10], which says that deciding whether a knowledge base in *DL-Lite* is satisfiable can be done in polynomial time.

Theorem 6.1 (see [10]) *Given a knowledge base in DL-Lite T, deciding whether T is satisfiable can be done in polynomial time.*

We now define a similarly restricted class of probabilistic knowledge bases. A *literal in DL-Lite* is either a basic concept in *DL-Lite* b or its negation $\neg b$. A *conjunctive concept in DL-Lite* is either $\bot$, or $\top$, or a conjunction of literals in *DL-Lite*. A set of conditional constraints $\mathcal{F}$ is defined in *DL-Lite* iff $\mathcal{F}$ is defined w.r.t. a set of basic c-concepts $\mathcal{C}$ that contains only literals in *DL-Lite*. A PTBox $PT = (T, P)$ is defined in *P-DL-Lite* iff (i) T is a knowledge base in *DL-Lite*, and (ii) P is defined in *DL-Lite*. A probabilistic knowledge base $KB = (T, P, (P_o)_{o \in \mathbf{I}_P})$ is defined in *P-DL-Lite* iff additionally every P_o with $o \in \mathbf{I}_P$ is defined in *DL-Lite*.

The following theorem shows that deciding whether a world $I \in \mathcal{I}_{\mathcal{C}}$ satisfies a knowledge base T in *DL-Lite* can be done in polynomial time when $\mathcal{C}$ contains only literals in *DL-Lite*. This result follows from Theorem 6.1 and the observation that $I \models T$ can be reduced to deciding whether a knowledge base in *DL-Lite* is satisfiable when $\mathcal{C}$ contains only literals in *DL-Lite*.

Theorem 6.2 *Let T be a knowledge base in DL-Lite, and let $\mathcal{C}$ be a finite nonempty set of basic c-concepts that contains only literals in DL-Lite. Then, given a world $I \in \mathcal{I}_{\mathcal{C}}$, deciding whether $I \models T$ holds can be done in polynomial time.*

6.3 Complexity Results

Our complexity results for the main tasks in P-$\mathcal{SHOIN}(\mathbf{D})$, P-$\mathcal{SHIF}(\mathbf{D})$, and P-*DL-Lite* are compactly summarized in Tables 1 and 2. In detail, the decision problems are all complete for NEXP, EXP, and NP

Table 1: Complexity of decision problems.

	P-*DL-Lite*	P-$\mathcal{SHIF}(\mathbf{D})$	P-$\mathcal{SHOIN}(\mathbf{D})$
SAT	NP-complete	EXP-complete	NEXP-complete
PTCON	NP-complete	EXP-complete	NEXP-complete
PKBCON	NP-complete	EXP-complete	NEXP-complete

Table 2: Complexity of computation problems.

	P-*DL-Lite*	P-$\mathcal{SHIF}(\mathbf{D})$	P-$\mathcal{SHOIN}(\mathbf{D})$
TLOGENT	FP^{NP}-complete	FEXP-complete	in FP^{NEXP}
TLEXENT	FP^{NP}-complete	FEXP-complete	in FP^{NEXP}

when they are defined in P-$\mathcal{SHOIN}(\mathbf{D})$, P-$\mathcal{SHIF}(\mathbf{D})$, and P-*DL-Lite*, respectively, while the computation problems are in FP^{NEXP}, complete for FEXP, and complete for FP^{NP}, respectively. Hence, when the reasoning tasks are defined in P-*DL-Lite*, then they have the same complexity as corresponding reasoning tasks in probabilistic logic and probabilistic default reasoning. Furthermore, when the reasoning tasks are defined in P-$\mathcal{SHIF}(\mathbf{D})$, then they have the same complexity as knowledge base satisfiability in $\mathcal{SHIF}(\mathbf{D})$. Finally, when the reasoning tasks are defined in P-$\mathcal{SHOIN}(\mathbf{D})$, then their complexity ranges from the complexity of deciding whether a knowledge base in $\mathcal{SHOIN}(\mathbf{D})$ is satisfiable to the complexity of other sophisticated reasoning techniques for the Semantic Web, such as deciding whether a description logic program relative to $\mathcal{SHOIN}(\mathbf{D})$ has a weak or strong answer set [21].

The following theorem shows that the decision problems are complete for NEXP, EXP, and NP when they are defined in P-$\mathcal{SHOIN}(\mathbf{D})$, P-$\mathcal{SHIF}(\mathbf{D})$, and P-*DL-Lite*, respectively. Here, hardness for NEXP resp. EXP is inherited from the hardness for NEXP resp. EXP of deciding whether a knowledge base in $\mathcal{SHOIN}(\mathbf{D})$ resp. $\mathcal{SHIF}(\mathbf{D})$ is satisfiable, while hardness for NP is inherited from the hardness for NP of deciding whether a finite set of conditional constraints in probabilistic logic is satisfiable and of deciding whether a probabilistic default theory is σ-consistent. Membership follows from a small-model theorem for deciding whether a finite set of conditional constraints in probabilistic logic is satisfiable [55].

Theorem 6.3 *(a)* SAT, PTCON, *and* PKBCON *are complete for* NEXP *when* $T \cup \mathcal{F}$*,* PT*, and* KB*, respectively, are defined in P-*$\mathcal{SHOIN}(\mathbf{D})$*. (b)* SAT, PTCON, *and* PKBCON *are complete for* EXP *when* $T \cup \mathcal{F}$*,* PT*, and* KB*, respectively, are defined in P-*$\mathcal{SHIF}(\mathbf{D})$*. (c)* SAT, PTCON, *and* PKBCON *are complete for* NP *when* $T \cup \mathcal{F}$*,* PT*, and* KB*, respectively, are defined in P-DL-Lite.*

The next theorem shows that the computation problems are in FP^{NEXP}, complete for FEXP, and complete for FP^{NP} when they are defined over P-$\mathcal{SHOIN}(\mathbf{D})$, P-$\mathcal{SHIF}(\mathbf{D})$, and P-*DL-Lite*, respectively. Here, hardness for FEXP is inherited from the EXP-hardness of deciding whether a description logic knowledge base in $\mathcal{SHIF}(\mathbf{D})$ is satisfiable, while FP^{NP}-hardness is inherited from the FP^{NP}-hardness of computing tight logically entailed intervals from a finite set of conditional constraints in probabilistic logic and of computing tight lexicographically entailed intervals from a probabilistic default theory. The membership results follow from a small-model theorem for computing tight logically entailed intervals from a finite set of conditional constraints in probabilistic logic [55].

Theorem 6.4 *(a)* TLogEnt *and* TLexEnt *are complete for* $\mathrm{FP}^{\mathrm{NEXP}}$ *when* $T \cup \mathcal{F}$ *and* $PT \cup \mathcal{F}$*, respectively, are defined in P-*$\mathcal{SHOIN}(\mathbf{D})$*. (b)* TLogEnt *and* TLexEnt *are complete for* FEXP *when* $T \cup \mathcal{F}$ *and* $PT \cup \mathcal{F}$*, respectively, are defined in P-*$\mathcal{SHIF}(\mathbf{D})$*. (c)* TLogEnt *and* TLexEnt *are complete for* $\mathrm{FP}^{\mathrm{NP}}$ *when* $T \cup \mathcal{F}$ *and* $PT \cup \mathcal{F}$*, respectively, are defined in P-DL-Lite.*

7 Related Work

In this section, we give a brief overview on related approaches to (i) probabilistic description logics and probabilistic web ontology languages, (ii) default reasoning from conditional knowledge bases and defaults in description logics, and (iii) possibilistic and fuzzy description logics.

7.1 Probabilistic Description Logics and Web Ontology Languages

There are several related approaches to probabilistic description logics in the literature [37, 44, 46, 49], which can be classified according to the generalized description logics, the supported forms of probabilistic knowledge, and the underlying probabilistic reasoning formalism. Heinsohn [37] presents a probabilistic extension of the description logic $\mathcal{ALC}$, which allows to represent terminological probabilistic knowledge about concepts and roles, and which is essentially based on probabilistic reasoning in probabilistic logics, similar to [63, 2, 24, 53]. Heinsohn [37], however, does not allow for assertional knowledge about concept and role instances. Jaeger's work [44] (which is perhaps the one closest in spirit to the new probabilistic description logics of this paper) proposes another probabilistic extension of the description logic $\mathcal{ALC}$, which allows for terminological and assertional probabilistic knowledge about concepts / roles and about concept instances, respectively, but does not support assertional probabilistic knowledge about role instances (but he mentions a possible extension in this direction). The uncertain reasoning formalism in [44] is essentially based on probabilistic reasoning in probabilistic logics, as the one in [37], but coupled with cross-entropy minimization to combine terminological probabilistic knowledge with assertional probabilistic knowledge. Jaeger's recent work [46] is less closely related, as it focuses on interpreting probabilistic concept subsumption and probabilistic role quantification through statistical sampling distributions, and develops a probabilistic version of the guarded fragment of first-order logic. The work by Koller et al. [49] gives a probabilistic generalization of the Classic description logic. Like Heinsohn's work [37], it allows for terminological probabilistic knowledge about concepts and roles, but does not support assertional knowledge about instances of concepts and roles. However, differently from [37], it is based on inference in Bayesian networks as underlying probabilistic reasoning formalism. Closely related work by Yelland [77] combines a restricted description logic close to $\mathcal{FL}$ with Bayesian networks. It also allows for terminological probabilistic knowledge about concepts and roles, but does not support assertional knowledge about instances of concepts and roles.

The novel probabilistic description logics in this paper differ from the ones in [37, 44, 46, 49] in several ways. First, they are probabilistic extensions of the expressive description logics $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$, which stand behind OWL Lite and OWL DL, respectively, towards sophisticated formalisms for reasoning under probabilistic uncertainty in the Semantic Web. Second, they allow for expressing both terminological probabilistic knowledge about concepts and roles, and also assertional probabilistic knowledge about instances of concepts and roles. Third, they are based on probabilistic lexicographic entailment from probabilistic default reasoning [54, 56] as underlying probabilistic reasoning formalism, which treats terminological and assertional probabilistic knowledge in a semantically very appealing way as probabilistic knowledge about random resp. concrete instances.

Related works on probabilistic web ontology languages focus especially on combining the web ontology language OWL with probabilistic formalisms based on Bayesian networks. In particular, da Costa [11], da Costa and Laskey [12], and da Costa et al. [13] suggest a probabilistic generalization of OWL, called PR-OWL, which is based on multi-entity Bayesian networks. The latter are a Bayesian logic that combines first-order logic with Bayesian probabilities. Ding et al. [16, 17] propose a probabilistic generalization of OWL, called BayesOWL, which is based on standard Bayesian networks. BayesOWL provides a set of rules and procedures for the direct translation of an OWL ontology into a Bayesian network that supports ontology reasoning, both within and across ontologies, as Bayesian inferences. Ding et al. [65, 17] also describe an application of this approach in ontology mapping. In closely related work, Mitra et al. [62] introduce a technique to enhancing existing ontology mappings by using a Bayesian network to represent the influences between potential concept mappings across ontologies. Yang and Calmet [76] present an integration of the web ontology language OWL with Bayesian networks. The approach makes use of probability and dependency-annotated OWL to represent uncertain information in Bayesian networks. Pool and Aikin [68] also provide a method for representing uncertainty in OWL ontologies, while Fukushige [25] proposes a basic framework for representing probabilistic relationships in RDF. Finally, Nottelmann and Fuhr [64] present two probabilistic extensions of variants of OWL Lite, along with a mapping to locally stratified probabilistic Datalog.

An important application for probabilistic ontologies (and thus probabilistic description logics and web ontology languages) is especially information retrieval: In particular, Subrahmanian's group [43, 42] explores the use of probabilistic ontologies in relational databases. They propose to extend relations by associating with every attribute a constrained probabilistic ontology, which describes relationships between terms occurring in the domain of that attribute. An extension of the relational algebra then allows for an increased recall (which is the proportion of documents relevant to a search query in the collection of all returned documents) in information retrieval. In closely related work, Mantay et al. [61] propose a probabilistic least common subsumer operation, which is based on a probabilistic extension of the description logic $\mathcal{ALN}$. They show that applying this approach in information retrieval allows for reducing the amount of retrieved data and thus for avoiding information flood. Another closely related work by Holi and Hyvönen [38, 39] shows how degrees of overlap between concepts can be modeled and computed efficiently using Bayesian networks based on RDF(S) ontologies. Such degrees of overlap indicate how well an individual data item matches the query concept, and can thus be used for measuring the relevance in information retrieval tasks.

7.2 Conditional Knowledge Bases and Defaults in Description Logics

Conditional knowledge bases consist of a set of strict statements in classical logic and a set of defeasible rules, also called defaults. The former must always hold, while the latter are rules of the kind $\psi \leftarrow \phi$, which read as "generally, if ϕ then ψ." Such rules may have exceptions, which can be handled in different ways. The following example illustrates conditional knowledge bases.

Example 7.1 *(Penguins)* A conditional knowledge base KB may encode the *strict logical knowledge* "all penguins are birds" and the *default logical knowledge* "generally, birds fly", "generally, penguins do not fly", and "generally, birds have wings". Some desirable conclusions from KB [34] are "generally, birds fly" and "generally, birds have wings" (which both belong to KB), "generally, penguins have wings" (since the set of all penguins is a subclass of the set of all birds, and thus penguins should inherit all properties of birds), "generally, penguins do not fly" (since properties of more specific classes should override inherited properties of less specific classes), and "generally, red birds fly" (since "red" is not mentioned at all in KB and thus should be considered irrelevant to the ability to fly of birds).

The literature contains several different proposals for default reasoning from conditional knowledge bases and extensive work on its desired properties. The core of these properties are the rationality postulates of System P by Kraus et al. [50], which constitute a sound and complete axiom system for several classical model-theoretic entailment relations under uncertainty measures on worlds. They characterize classical model-theoretic entailment under preferential structures [50], infinitesimal probabilities [1], possibility measures [19], and world rankings [71, 33]. They also characterize an entailment relation based on conditional objects [20]. A survey of all these relationships is given in [6, 26]. Mainly to solve problems with irrelevant information, the notion of rational closure as a more adventurous notion of entailment was introduced by Lehmann [52]. It is in particular equivalent to entailment in System Z by Pearl [67] and to the least specific possibility entailment by Benferhat et al. [5]. Mainly in order to solve problems with property inheritance from classes to exceptional subclasses, the maximum entropy approach to default entailment was proposed by Goldszmidt et al. [31], the notion of lexicographic entailment was introduced by Lehmann [51] and Benferhat et al. [4], the notion of conditional entailment was proposed by Geffner [28, 29], and an infinitesimal belief function approach was suggested by Benferhat et al. [7].

To my knowledge, the expressive probabilistic description logics P-$\mathcal{SHIF}(\mathbf{D})$ and P-$\mathcal{SHOIN}(\mathbf{D})$ are the first extensions of description logics by defaults as in conditional knowledge bases. Note that extensions of description logics by defaults as in Reiter's default logic (see e.g. [3, 9]) are less closely related.

7.3 Possibilistic and Fuzzy Description Logics

Similarly to probabilistic description logics for the Semantic Web, also possibilistic and fuzzy description logics for the Semantic Web have been developed (see especially the works by Dubois et al. [18] and by Straccia [72], respectively), which adapt and generalize previous approaches to possibilistic description logics and to fuzzy description logics to the needs of the Semantic Web.

Semantically, however, possibilistic and fuzzy description logics are very different from probabilistic description logics, since they are based on possibility measures and on many-valued interpretations under compositional truth functions, respectively, rather than probability measures. As a consequence, possibilistic description logics encode especially rankings and preferences (such as e.g. "John prefers an ice cream to a beer"), while fuzzy description logics allow for expressing forms of vagueness and imprecision (such as e.g. "John is tall"). Probabilistic description logics, in contrast, encode quantified ambiguous information (such as e.g. "John is a student with the probability 0.7 and a teacher with the probability 0.3").

8 Conclusion

Towards sophisticated formalisms for reasoning under probabilistic uncertainty in the Semantic Web, we have presented the probabilistic description logics P-$\mathcal{SHIF}(\mathbf{D})$ and P-$\mathcal{SHOIN}(\mathbf{D})$, which are extensions of the expressive description logics $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$ behind OWL Lite and OWL DL, respectively. The new probabilistic description logics allow for expressing rich terminological and assertional probabilistic knowledge. They are semantically based on the notion of probabilistic lexicographic entailment from probabilistic default reasoning, which naturally interprets terminological and assertional probabilistic knowledge as probabilistic knowledge about random resp. concrete instances. As an important additional feature, the new probabilistic description logics also allow for expressing terminological default knowledge, which is semantically interpreted as in Lehmann's lexicographic entailment in default reasoning from conditional knowledge bases. We have presented sound and complete algorithms for the main reasoning problems in P-$\mathcal{SHIF}(\mathbf{D})$ and P-$\mathcal{SHOIN}(\mathbf{D})$, which are based on reductions to classical reasoning in $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$, respectively, and to solving linear optimization problems. In particular, they show the important result that the main reasoning problems in both P-$\mathcal{SHIF}(\mathbf{D})$ and P-$\mathcal{SHOIN}(\mathbf{D})$ are decidable / computable. Furthermore, we have analyzed the computational complexity of the main reasoning problems in the general as well as restricted cases.

Note that the extension by probabilistic uncertainty can be applied to other description logics as well. The semantics of such an extension and the algorithms for the main reasoning tasks can then be defined in the same way as for P-$\mathcal{SHIF}(\mathbf{D})$ and P-$\mathcal{SHOIN}(\mathbf{D})$. The complexity results for P-$\mathcal{SHIF}(\mathbf{D})$ resp. P-$\mathcal{SHOIN}(\mathbf{D})$ also carry over to such an extension as long as the extended classical description logic has the same complexity characterization as $\mathcal{SHIF}(\mathbf{D})$ resp. $\mathcal{SHOIN}(\mathbf{D})$.

An implementation of the new probabilistic description logics can be developed on top of the system NMPROBLOG (which implements different notions of nonmonotonic probabilistic entailment, including probabilistic lexicographic entailment in probabilistic default reasoning; see [58]) by essentially replacing its component for classical reasoning in propositional logics by a component for classical reasoning in the expressive description logics $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$.

Current work concerns the development and implementation of probabilistic generalizations of OWL Lite and OWL DL that are based on the two novel probabilistic description logics P-$\mathcal{SHIF}(\mathbf{D})$ resp. P-$\mathcal{SHOIN}(\mathbf{D})$. An interesting topic of future research is to explore an application of the new probabilistic description logics for matchmaking and ranking objects in ontologies (e.g., along the lines of [15, 69, 60]). Another issue for future work is to investigate an integration of the new probabilistic description logics with description logic programs and probabilistic description logic programs. Finally, it would also be interesting to allow for more complex data types and more complex probabilistic query languages on top of the expressive probabilistic description logics P-$\mathcal{SHIF}(\mathbf{D})$ and P-$\mathcal{SHOIN}(\mathbf{D})$.

Appendix A: Proofs for Section 4

Proof of Proposition 4.8. ($\Rightarrow$) Suppose that T is satisfiable. That is, there exists an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ with $\Delta^{\mathcal{I}} \neq \emptyset$ relative to a datatype theory $\mathbf{D} = (\Delta^{\mathbf{D}}, \cdot^{\mathbf{D}})$ such that $\mathcal{I}$ satisfies T. Let i be any member of $\Delta^{\mathcal{I}}$, and let $I = \{\phi \in \mathcal{C} \mid i \in \phi^{\mathcal{I}}\}$. Then, I is a world from $\mathcal{I}_{\mathcal{C}}$ such that $T \cup \{\phi(j) \mid \phi \in I\} \cup \{\neg\phi(j) \mid \phi \in \mathcal{C} \setminus I\}$ is satisfiable. That is, I is a model of T.

($\Leftarrow$) Suppose that T has a model $I \in \mathcal{I}_{\mathcal{C}}$. That is, $T \cup \{\phi(i) \mid \phi \in I\} \cup \{\neg\phi(i) \mid \phi \in \mathcal{C} \setminus I\}$ is satisfiable, and thus also T is satisfiable. □

Proof of Proposition 4.9. ($\Rightarrow$) Suppose that T is satisfiable. By Proposition 4.8, T has a model $I \in \mathcal{I}_{\mathcal{C}}$. Let the probabilistic interpretation Pr on $\mathcal{I}_{\mathcal{C}}$ be defined by $Pr(I) = 1$ and $Pr(I') = 0$ for all other $I' \in \mathcal{I}_{\mathcal{C}}$. Then, Pr is a model of T on $\mathcal{I}_{\mathcal{C}}$.

($\Leftarrow$) Suppose that T has a model Pr on $\mathcal{I}_{\mathcal{C}}$. Then, there exists some $I \in \mathcal{I}_{\mathcal{C}}$ such that $Pr(I) > 0$. Since Pr is a model of T, also I is a model of T. By Proposition 4.8, T is satisfiable. □

Proof of Theorem 4.12. ($\Rightarrow$) Suppose that $PT = (T, P)$ is consistent. That is, (i) T is satisfiable and (ii) there exists an ordered partition $(P_0, \ldots, P_k)$ of P such that each P_i with $i \in \{0, \ldots, k\}$ is the set of all

$F \in P_i \cup \cdots \cup P_k$ that are tolerated under T by $P_i \cup \cdots \cup P_k$. The latter implies that every $F \in P_i$ with $i \in \{0, \ldots, k\}$ is tolerated under T by $P_i \cup \cdots \cup P_k$.

($\Leftarrow$) Suppose that (i) T is satisfiable and (ii) there exists an ordered partition $(P_0, \ldots, P_k)$ of P such that every $F \in P_i$ with $i \in \{0, \ldots, k\}$ is tolerated under T by $P_i \cup \cdots \cup P_k$. Let the pairwise disjoint subsets $P'_0, \ldots, P'_l$ of P be defined as follows: (a) every P'_i with $i \in \{0, \ldots, l\}$ is the set of all $F \in P \setminus (P'_0 \cup \cdots \cup P'_{i-1})$ that are tolerated under T by $P \setminus (P'_0 \cup \cdots \cup P'_{i-1})$, and (b) no F in $P^\star = P \setminus (P'_0 \cup \cdots \cup P'_{l-1})$ is tolerated under T by $P^\star$. We now show that $P^\star = \emptyset$. Towards a contradiction, suppose the contrary. Then, let $j \in \{0, \ldots, k\}$ be maximal such that $P^\star \subseteq P_j \cup \cdots \cup P_k$. Hence, there exists some $F \in P^\star \cap P_j$ that is not tolerated under T by $P^\star$, and thus also not tolerated under T by $P_j \cup \cdots \cup P_k$. But this contradicts every $F \in P_i$ with $i \in \{0, \ldots, k\}$ being tolerated under T by $P_i \cup \cdots \cup P_k$. This shows that $P^\star = \emptyset$. Thus, $PT = (T, P)$ is consistent. □

Proof of Theorem 4.15. (a) If $\mathcal{Q} = \emptyset$, then $T \cup Q \cup \mathcal{F} \cup \{(\phi|\top)[1,1]\}$ is not satisfiable for all $Q \subseteq P$. In particular, $T \cup \mathcal{F} \cup \{(\phi|\top)[1,1]\}$ is not satisfiable. Thus, $\mathcal{F} \,|\!\!\sim^{lex}_{tight} (\psi|\phi)[1,0]$ under PT.

(b) Suppose that $\mathcal{Q} \neq \emptyset$. Then, a probabilistic interpretation Pr is a lex-minimal model of $T \cup \mathcal{F} \cup \{(\phi \mid \top)[1,1]\}$ iff (i) Pr is a model of $T \cup \mathcal{F} \cup \{(\phi|\top)[1,1]\}$ and (ii) $\{F \in P \mid Pr \models F\}$ is a lex-minimal element in the set of all $Q \subseteq P$ such that $T \cup Q \cup \mathcal{F} \cup \{(\phi|\top)[1,1]\}$ is satisfiable. The latter is in turn equivalent to Pr being a model of $T \cup Q \cup \mathcal{F} \cup \{(\phi|\top)[1,1]\}$ for some $Q \in \mathcal{Q}$. □

Appendix B: Proofs for Section 5

Proof of Theorem 5.1. (a) We first consider Algorithm *pt-consistency*. In line 1, we first decide one time whether T is satisfiable. In line 7, we then decide $O(|P|^2)$ times whether $T \cup H \cup \{(\phi|\top)[1,1]\}$, for $C = (\psi|\phi)[l,u] \in H$, is satisfiable. In summary, *pt-consistency* can be done by solving $O(|P|^2)$ instances of SAT.

(b) We next consider Algorithm *pkb-consistency*. In line 1, we first decide whether (T, P) is consistent. By (a), this can be done by solving $O(|P|^2)$ instances of SAT. In line 3, we then decide $|\mathbf{I}_P|$ times whether $T \cup P_o$, for $o \in \mathbf{I}_P$, is satisfiable. In summary, *pkb-consistency* can be done by solving $O(|P|^2 + |\mathbf{I}_P|)$ instances of SAT.

(c) As for Algorithm *tight-lex-entailment*, in line 1, we first decide whether PT is consistent. By (a), this can be done by solving $O(|P|^2)$ instances of SAT. In line 3, we then decide one time whether $T \cup \mathcal{F} \cup \{(\phi|\top)[1,1]\}$ is satisfiable, and in lines 9 and 12, we decide $O(2^{|P|})$ times whether $T \cup \mathcal{F} \cup \{(\phi|\top)[1,1]\} \cup G \cup H$ is satisfiable, for certain $G \subseteq P_i$ and $H \subseteq P_{i+1} \cup \cdots \cup P_k$. Moreover, in line 16, we compute $O(2^{|P|})$ times the rational numbers $c, d \in [0,1]$ such that $T \cup \mathcal{F} \cup \{(\phi|\top)[1,1]\} \cup H \models_{tight} (\psi|\top)[c,d]$, for certain $H \subseteq P$. In summary, *tight-lex-consequence* can be done by solving $O(2^{|P|})$ instances of SAT and TLOGENT. □

Appendix C: Proofs for Section 6

Proof of Theorem 6.2. Recall that $I \models T$ is equivalent to $T' = T \cup \{\phi(i) \mid \phi \in I\} \cup \{\neg\phi(i) \mid \phi \in \mathcal{C} \setminus I\}$ being satisfiable, where i is a new individual. Since $\mathcal{C}$ is a finite set of literals in *DL-Lite*, deciding whether T' is satisfiable can be done by a satisfiability test on a knowledge base of the form $T'' = T \cup \{b(i) \mid b \in B^+\} \cup \{\neg b(i) \mid b \in B^-\}$, which in turn is satisfiable iff $T''' = T \cup \{b(i) \mid b \in B^+\} \cup \{b^\neg(i) \mid$

$b \in B^-\} \cup \{b^\neg \sqsubseteq \neg b \mid b \in B^-\}$ is satisfiable, where the $b^\neg$'s are new basic concepts in *DL-Lite* that do not occur in T''. Since T''' is a knowledge base in *DL-Lite*, by Theorem 6.1, deciding whether T''' is satisfiable can be done in polynomial time. In summary, deciding whether $I \models T$ holds can be done in polynomial time. $\Box$

Proof of Theorem 6.3. (a) We first show that SAT is NEXP-complete for $T \cup \mathcal{F}$ in $\mathcal{SHOIN}(\mathbf{D})$. By Theorem 5.2, $T \cup \mathcal{F}$ is satisfiable iff the system of linear constraints LC in Fig. 4 over the variables y_r $(r \in R = \{I \in \mathcal{I}_\mathcal{C} \mid I \models T\})$ is solvable. By a fundamental result from linear programming, the solvability of LC implies the existence of a solution of LC that has a polynomial size in the input size of $\mathcal{F}$ [22], that is, a solution $y^\star_r$ $(r \in R)$ of LC such that (i) the number of all $r \in R$ with $y^\star_r > 0$ and (ii) the size of each $y^\star_r$ with $r \in R$ and $y^\star_r > 0$ are polynomial in the input size of $\mathcal{F}$. Hence, guessing such a solution $y^\star_r$ $(r \in R)$ of LC can be done in nondeterministic polynomial time, and verifying that (a) $r \models T$ for all $r \in R$ with $y^\star_r > 0$ and (b) $y^\star_r$ $(r \in R)$ satisfies LC can be done in nondeterministic exponential time for T in $\mathcal{SHOIN}(\mathbf{D})$ and in polynomial time, respectively. In summary, guessing and verifying such a solution of LC, and thus deciding whether $T \cup \mathcal{F}$ is satisfiable, is in NEXP. The NEXP-hardness of SAT is immediate by a reduction from the NEXP-hard problem of deciding knowledge base satisfiability in $\mathcal{SHOIN}(\mathbf{D})$, since T has a probabilistic model Pr iff T has a classical model $\mathcal{I}$.

We next prove that PTCON is NEXP-complete for PT in $\mathcal{SHOIN}(\mathbf{D})$. By Theorem 4.12, a PTBox $PT = (T, P)$ is consistent iff (i) T is satisfiable and (ii) there exists an ordered partition $(P_0, \ldots, P_k)$ of P such that every $F \in P_i$, $i \in \{0, \ldots, k\}$, is tolerated under T by $P_i \cup \cdots \cup P_k$. Since deciding knowledge base satisfiability in $\mathcal{SHOIN}(\mathbf{D})$ is in NEXP, deciding whether (i) holds is in NEXP. Observe then that guessing a partition $(P_0, \ldots, P_k)$ of P can be done in nondeterministic polynomial time, and verifying that every $F \in P_i$, $i \in \{0, \ldots, k\}$, is tolerated under T by $P_i \cup \cdots \cup P_k$ is in NEXP, since SAT is in NEXP. In summary, guessing and verifying such a partition is in NEXP, and thus also deciding whether PT is consistent is in NEXP. The NEXP-hardness of PTCON holds by a reduction from the NEXP-hard problem of deciding knowledge base satisfiability in $\mathcal{SHOIN}(\mathbf{D})$, since $PT = (T, \emptyset)$ is consistent iff T has a classical model $\mathcal{I}$.

We finally prove that PKBCON is NEXP-complete for KB in $\mathcal{SHOIN}(\mathbf{D})$. Recall that a probabilistic knowledge base $KB = (T, P, (P_o)_{o \in \mathbf{I}_P})$ is consistent iff (T, P) is consistent and every $T \cup P_o$ with $o \in \mathbf{I}_P$ is satisfiable. As argued above, deciding whether (T, P) is consistent is in NEXP, and deciding whether every $T \cup P_o$ with $o \in \mathbf{I}_P$ is satisfiable is in NEXP as well. This already shows that PKBCON is in NEXP. The NEXP-hardness of PTCON holds by a reduction from the NEXP-hard problem of deciding knowledge base satisfiability in $\mathcal{SHOIN}(\mathbf{D})$, since $KB = (T, \emptyset, ())$ is consistent iff T has a classical model $\mathcal{I}$.

(b) We first show that SAT is EXP-complete for $T \cup \mathcal{F}$ in $\mathcal{SHIF}(\mathbf{D})$. By Theorem 5.2, $T \cup \mathcal{F}$ is satisfiable iff the system of linear constraints LC in Fig. 4 over the variables y_r $(r \in R = \{I \in \mathcal{I}_\mathcal{C} \mid I \models T\})$ is solvable. By a fundamental result from linear programming, the solvability of LC implies the existence of a solution of LC that has a polynomial size in the input size of $\mathcal{F}$ [22], that is, a solution $y^\star_r$ $(r \in R)$ of LC such that (i) the number of all $r \in R$ with $y^\star_r > 0$ is polynomial in the input size of $\mathcal{F}$ and (ii) the size of each $y^\star_r$ with $r \in R$ and $y^\star_r > 0$ is polynomial in the input size of $\mathcal{F}$. Hence, deciding whether LC is solvable can be done by generating all potential polynomial-size solutions $y^\star_r$ $(r \in R)$ of LC, and verifying that (a) $r \models T$ for all $r \in R$ with $y^\star_r > 0$ and (b) $y^\star_r$ $(r \in R)$ satisfies LC. Since (a) is in EXP and (b) can be done in polynomial time, deciding whether LC is solvable is in EXP as well. The EXP-hardness of SAT is immediate by a reduction from the EXP-hard problem of deciding knowledge base satisfiability in $\mathcal{SHIF}(\mathbf{D})$, since T has a probabilistic model Pr iff T has a classical model $\mathcal{I}$.

We next prove that PTCON is EXP-complete for PT in $\mathcal{SHIF}(\mathbf{D})$. Recall that Algorithm *pt-consis-*

tency decides whether $PT = (T, P)$ is consistent. By Theorem 5.1, Algorithm *pt-consistency* can be done by solving $O(|P|^2)$ instances of SAT. As argued above, solving one such instance of SAT is in EXP. In summary, deciding whether PT is consistent is in EXP. The EXP-hardness of PTCON holds by a reduction from the EXP-hard problem of deciding knowledge base satisfiability in $\mathcal{SHIF}(\mathbf{D})$, since $PT = (T, \emptyset)$ is consistent iff T has a classical model $\mathcal{I}$.

We finally prove that PKBCON is EXP-complete for KB in $\mathcal{SHIF}(\mathbf{D})$. Recall that a probabilistic knowledge base $KB = (T, P, (P_o)_{o \in \mathbf{I}_P})$ is consistent iff (T, P) is consistent and every $T \cup P_o$ with $o \in \mathbf{I}_P$ is satisfiable. As argued above, deciding whether (T, P) is consistent is in EXP, and deciding whether every $T \cup P_o$ with $o \in \mathbf{I}_P$ is satisfiable is in EXP as well. In summary, this shows that deciding whether KB is consistent is in EXP. The EXP-hardness of PTCON holds by a reduction from the EXP-hard problem of deciding knowledge base satisfiability in $\mathcal{SHIF}(\mathbf{D})$, since $KB = (T, \emptyset, ())$ is consistent iff T has a classical model $\mathcal{I}$.

(c) The proofs that SAT, PTCON, and PKBCON are in NP when $T \cup \mathcal{F}$, PT, and KB, respectively, are defined in *DL-Lite* are verbally the same as the proofs that SAT, PTCON, and PKBCON are in NEXP when $T \cup \mathcal{F}$, PT, and KB, respectively, are defined in $\mathcal{SHOIN}(\mathbf{D})$ in (a), except that now we use that deciding whether $r \models T$ holds is in P, rather than in NEXP, as for $\mathcal{SHOIN}(\mathbf{D})$. Furthermore, we use that SAT and PTCON for *DL-Lite* are both in NP, rather than in NEXP.

Hardness for NP of SAT holds by a reduction from the NP-complete problem of deciding whether a finite set $\mathcal{F}$ of conditional constraints over Boolean combinations of elementary propositions is satisfiable [55], since $T \cup \mathcal{F}$ with $T = \emptyset$ is satisfiable in our framework iff $\mathcal{F}$ is satisfiable in the framework of [55]. Hardness for NP of PTCON holds by a reduction from the NP-complete *graph 3-colorability problem* [27]. The proof is identical to the proof of NP-hardness of deciding whether a probabilistic default theory is σ-consistent in [56]. Finally, hardness for NP of PKBCON holds by a reduction from the NP-complete problem PTCON, since $KB = (T, P, ())$ is consistent iff (T, P) is consistent. □

Proof of Theorem 6.4. (a) We first show that TLOGENT is in FP^{NEXP} for $T \cup \mathcal{F}$ in $\mathcal{SHOIN}(\mathbf{D})$. By Theorem 5.3, l (resp., u) such that $T \cup \mathcal{F} \models_{tight} (\psi|\top)[l, u]$ is the optimal value of the linear program (1) over the variables y_r ($r \in R = \{I \in \mathcal{I}_{\mathcal{C}} \mid I \models T\}$). By a fundamental result from linear programming, the optimal value l (resp., u) of (1) has a polynomial size in the input size of $\mathcal{F}$ [55]. Thus, we can compute l (resp., u) by binary search on the set of all potential polynomial-size values s of the objective function of (1) subject to LC. For each such s, we decide whether $T \cup \mathcal{F} \cup \{(\psi|\top)[s, s]\}$ is satisfiable. The binary search can be done in polynomial time, and each satisfiability check is in NEXP, by Theorem 6.3 (a). In summary, this shows that TLOGENT is in FP^{NEXP}.

We next show that TLEXENT is in FP^{NEXP} for $PT = (T, P)$ and $\mathcal{F}$ in $\mathcal{SHOIN}(\mathbf{D})$. Let l (resp., u) be such that $\mathcal{F} \Vdash^{lex}_{tight} (\psi|\phi)[l, u]$ under PT. If $T \cup \mathcal{F} \cup \{(\phi|\top)[1, 1]\}$ is unsatisfiable, then $l = 1$ (resp., $u = 0$). Otherwise, by Theorem 4.15, l (resp., u) is given by $\min l'$ (resp., $\max u'$) subject to $T \cup Q \cup \mathcal{F} \cup \{(\phi|\top)[1, 1]\} \models_{tight} (\psi|\top)[l', u']$ and $Q \in \mathcal{Q}$, where $\mathcal{Q}$ is the set of all lex-minimal elements in the set of all $S \subseteq P$ such that $T \cup S \cup \mathcal{F} \cup \{(\phi|\top)[1, 1]\}$ is satisfiable.

Observe then that the vector $(n_0, \ldots, n_k) = (|Q \cap P_0|, \ldots, |Q \cap P_k|)$ is the same for all $Q \in \mathcal{Q}$ and in fact characterizes $\mathcal{Q}$. More concretely, $\mathcal{Q}$ is the set of all $S \subseteq P$ such that (i) $(|S \cap P_0|, \ldots, |S \cap P_k|) = (n_0, \ldots, n_k)$ and (ii) $T \cup S \cup \mathcal{F} \cup \{(\phi|\top)[1, 1]\}$ is satisfiable. Furthermore, the vector $(n_0, \ldots, n_k)$ can be computed stepwise for decreasing $i \in \{0, \ldots, k\}$ (starting with $i = k$) by deciding for every $d \in \{0, \ldots, |P_i|\}$ ($\star$) whether there exists some $S \subseteq P_i \cup \cdots \cup P_k$ such that (i) $|S \cap P_i| = d$, (ii) $|S \cap P_j| = n_j$ for all $j \in \{i+1, \ldots, k\}$, and (iii) $T \cup S \cup \mathcal{F} \cup \{(\phi|\top)[1, 1]\}$ is satisfiable. Guessing some $S \subseteq P_i \cup \cdots \cup P_k$ can be done in nondeterministic polynomial time, and deciding whether (i)–(iii) hold is in NEXP, by Theorem 6.3 (a).

Hence, $(\star)$ is in NEXP, and thus the overall stepwise computation of the vector $(n_0, \ldots, n_k)$ is in FP^{NEXP}.

By Theorem 5.3, l' (resp., u') such that $T \cup Q \cup \mathcal{F} \cup \{(\phi|\top)[1,1]\} \models_{tight} (\psi|\top)[l', u']$ is the optimal value of the linear program $(1)'$ over the variables y_r $(r \in R = \{I \in \mathcal{I}_\mathcal{C} \mid I \models T\})$, where $(1)'$ is obtained from (1) by replacing $\mathcal{F}$ by $Q \cup \mathcal{F} \cup \{(\phi|\top)[1,1]\}$. As argued above, by a fundamental result from linear programming, the optimal value l' (resp., u') of $(1)'$ has a polynomial size in the input size of $Q \cup \mathcal{F} \cup \{(\phi|\top)[1,1]\}$ [55]. This shows that the desired l (resp., u) has a polynomial size in the input size of $P \cup \mathcal{F} \cup \{(\phi|\top)[1,1]\}$. Hence, we can compute l (resp., u) by binary search on the set of all potential polynomial-size values s of the objective function of $(1)'$. For each such s, we decide whether some $Q \in \mathcal{Q}$ exists such that $T \cup Q \cup \mathcal{F} \cup \{(\phi|\top)[1,1]\} \cup \{(\psi|\top)[s,s]\}$ is satisfiable. The binary search can be done in polynomial time, and guessing some $S \subseteq P$ and verifying that (i) $(|S \cap P_0|, \ldots, |S \cap P_k|) = (n_0, \ldots, n_k)$ and (ii) $T \cup S \cup \mathcal{F} \cup \{(\phi|\top)[1,1]\} \cup \{(\psi|\top)[s,s]\}$ is satisfiable is in NEXP, by Theorem 6.3 (a). In summary, computing the desired l (resp., u) is in FP^{NEXP}, once $(n_0, \ldots, n_k)$ is given.

The overall algorithm for computing l (resp., u) such that $\mathcal{F} \Vdash\!\sim^{lex}_{tight} (\psi|\phi)[l,u]$ under PT is thus given as follows. We first decide whether $T \cup \mathcal{F} \cup \{(\phi|\top)[1,1]\}$ is unsatisfiable, which is in co-NEXP, by Theorem 6.3 (a). If this is the case, then $l = 1$ (resp., $u = 0$). Otherwise, we first compute the vector $(n_0, \ldots, n_k)$, and then l (resp., u) by binary search, which is both in FP^{NEXP}. In summary, computing l (resp., u) such that $\mathcal{F} \Vdash\!\sim^{lex}_{tight} (\psi|\phi)[l,u]$ under PT is in FP^{NEXP}.

(b) We first prove that TLOGENT is FEXP-complete for $T \cup \mathcal{F}$ in $\mathcal{SHIF}(\mathbf{D})$. Membership in FEXP is proved in the same way as membership in FP^{NEXP} for $\mathcal{SHOIN}(\mathbf{D})$, except that now each satisfiability check is in EXP, by Theorem 6.3 (b), and thus TLOGENT is in FEXP. The FEXP-hardness of TLOGENT is immediate by a reduction from the EXP-hard problem of deciding knowledge base satisfiability in $\mathcal{SHIF}(\mathbf{D})$, since $T \models_{tight} (\top|\top)[1,1]$ iff T has a classical model $\mathcal{I}$.

As for the FEXP-completeness of TLEXENT for $\mathcal{SHIF}(\mathbf{D})$, membership in FEXP is proved in the same way as membership in FP^{NEXP} for $\mathcal{SHOIN}(\mathbf{D})$, except that now each satisfiability check is in EXP, by Theorem 6.3 (b), and thus TLEXENT is in FEXP. Furthermore, hardness for FEXP of TLEXENT is immediate by a reduction from the EXP-hard problem of deciding knowledge base satisfiability in $\mathcal{SHIF}(\mathbf{D})$, since $T \Vdash\!\sim^{lex}_{tight} (\top|\top)[1,1]$ iff T has a classical model $\mathcal{I}$.

(c) The proofs that TLOGENT and TLEXENT are in FP^{NP} when $T \cup \mathcal{F}$ and $PT \cup \mathcal{F}$, respectively, are defined in *DL-Lite* are the same as the proofs that TLOGENT and TLEXENT are in FP^{NEXP} when $T \cup \mathcal{F}$ and $PT \cup \mathcal{F}$, respectively, are defined in $\mathcal{SHOIN}(\mathbf{D})$ in (a), except that now we use that SAT for *DL-Lite* is in NP, rather than in NEXP, as for $\mathcal{SHOIN}(\mathbf{D})$.

Hardness for FP^{NP} of TLOGENT holds by a reduction from the FP^{NP}-complete problem of computing tight logically entailed intervals from a finite set $\mathcal{F}$ of conditional constraints over Boolean combinations of elementary propositions [55], since tight logical entailment from $T \cup \mathcal{F}$ with $T = \emptyset$ here coincides with tight logical entailment from $\mathcal{F}$ in [55]. Hardness for FP^{NP} of TLEXENT holds by a reduction from the FP^{NP}-complete *traveling salesman cost problem* [66]. The proof is identical to the proof of FP^{NP}-hardness of computing tight lex-entailed intervals from a probabilistic default theory in [56]. □

References

[1] E. W. Adams. *The Logic of Conditionals*, volume 86 of *Synthese Library*. D. Reidel, Dordrecht, Netherlands, 1975.

[2] S. Amarger, D. Dubois, and H. Prade. Constraint propagation with imprecise conditional probabilities. In *Proc. UAI-1991*, pp. 26–34. Morgan Kaufmann, 1991.

[3] F. Baader and B. Hollunder. Embedding defaults into terminological knowledge representation formalisms. *J. Autom. Reasoning*, 14(1):149–180, 1995.

[4] S. Benferhat, C. Cayrol, D. Dubois, J. Lang, and H. Prade. Inconsistency management and prioritized syntax-based entailment. In *Proceedings IJCAI-1993*, pp. 640–645. Morgan Kaufmann, 1993.

[5] S. Benferhat, D. Dubois, and H. Prade. Representing default rules in possibilistic logic. In *Proceedings KR-1992*, pp. 673–684. Morgan Kaufmann, 1992.

[6] S. Benferhat, D. Dubois, and H. Prade. Nonmonotonic reasoning, conditional objects and possibility theory. *Artif. Intell.*, 92(1–2):259–276, 1997.

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

[8] T. Berners-Lee. *Weaving the Web*. Harper, San Francisco, 1999.

[9] P. A. Bonatti, C. Lutz, and F. Wolter. Description logics with circumscription. In *Proceedings KR-2006*, pp. 400–410. AAAI Press, 2006.

[10] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. *DL-Lite*: Tractable description logics for ontologies. In *Proceedings AAAI-2005*, pp. 602–607. AAAI Press / MIT Press, 2005.

[11] P. C. G. da Costa. *Bayesian semantics for the Semantic Web*. PhD thesis, George Mason University, Fairfax, VA, USA, 2005.

[12] P. C. G. da Costa and K. B. Laskey. PR-OWL: A framework for probabilistic ontologies. In *Proceedings FOIS-2006*, pp. 237–249. IOS Press, 2006.

[13] P. C. G. da Costa, K. B. Laskey, and K. J. Laskey. PR-OWL: A Bayesian ontology language for the Semantic Web. In *Proceedings URSW-2005*, pp. 23–33, 2005.

[14] F. D. de Saint-Cyr and H. Prade. Possibilistic handling of uncertain default rules with applications to persistence modeling and fuzzy default reasoning. In *Proceedings KR-2006*, pp. 440–451. AAAI Press, 2006.

[15] T. Di Noia, E. Di Sciascio, F. M. Donini, and M. Mongiello. Abductive matchmaking using description logics. In *Proceedings IJCAI-2003*, pp. 337–342. Morgan Kaufmann, 2003.

[16] Z. Ding and Y. Peng. A probabilistic extension to ontology language OWL. In *Proceedings HICSS-2004*, 2004.

[17] Z. Ding, Y. Peng, and R. Pan. BayesOWL: Uncertainty modeling in Semantic Web ontologies. In Z. Ma, editor, *Soft Computing in Ontologies and Semantic Web*, volume 204 of *Studies in Fuzziness and Soft Computing*. Springer, 2006.

[18] D. Dubois, J. Mengin, and H. Prade. Possibilistic uncertainty and fuzzy features in description logic: A preliminary discussion. In E. Sanchez, editor, *Fuzzy Logic and the Semantic Web*, Capturing Intelligence, chapter 6, pp. 101–114. Elsevier, 2006.

[19] D. Dubois and H. Prade. Possibilistic logic, preferential models, non-monotonicity and related issues. In *Proc. IJCAI-1991*, pp. 419–424. Morgan Kaufmann, 1991.

[20] D. Dubois and H. Prade. Conditional objects as non-monotonic consequence relationships. *IEEE Trans. Systems, Man and Cybernetics*, 24:1724–1740, 1994.

[21] T. Eiter, T. Lukasiewicz, R. Schindlauer, and H. Tompits. Combining answer set programming with description logics for the Semantic Web. In *Proceedings KR-2004*, pp. 141–151. AAAI Press, 2004.

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

[23] D. Fensel, W. Wahlster, H. Lieberman, and J. Hendler, editors. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press, 2002.

[24] A. M. Frisch and P. Haddawy. Anytime deduction for probabilistic logic. *Artif. Intell.*, 69(1–2):93–122, 1994.

[25] Y. Fukushige. Representing probabilistic knowledge in the Semantic Web. In *Proceedings of the W3C Workshop on Semantic Web for Life Sciences*, Cambridge, MA, USA, 2004.

[26] D. M. Gabbay and P. Smets, editors. *Handbook on Defeasible Reasoning and Uncertainty Management Systems*. Kluwer Academic, Dordrecht, Netherlands, 1998.

[27] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.

[28] H. Geffner. *Default Reasoning: Causal and Conditional Theories*. MIT Press, Cambridge, MA, 1992.

[29] H. Geffner and J. Pearl. Conditional entailment: Bridging two approaches to default reasoning. *Artif. Intell.*, 53(2–3):209–244, 1992.

[30] R. Giugno and T. Lukasiewicz. P-$\mathcal{SHOQ}(\mathbf{D})$: A probabilistic extension of $\mathcal{SHOQ}(\mathbf{D})$ for probabilistic ontologies in the Semantic Web. In *Proceedings JELIA-2002*, volume 2424 of *LNCS*, pp. 86–97. Springer, 2002.

[31] M. Goldszmidt, P. Morris, and J. Pearl. A maximum entropy approach to nonmonotonic reasoning. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(3):220–232, 1993.

[32] M. Goldszmidt and J. Pearl. On the consistency of defeasible databases. *Artif. Intell.*, 52(2):121–149, 1991.

[33] M. Goldszmidt and J. Pearl. Rank-based systems: A simple approach to belief revision, belief update and reasoning about evidence and actions. In *Proceedings KR-1992*, pp. 661–672. Morgan Kaufmann, 1992.

[34] M. Goldszmidt and J. Pearl. Qualitative probabilities for default reasoning, belief revision, and causal modeling. *Artif. Intell.*, 84(1–2):57–112, 1996.

[35] T. Hailperin. *Boole's Logic and Probability*. North-Holland, 2nd edition, 1986.

[36] A. Halevy, M. Franklin, and D. Maier. Principles of dataspace systems. In *Proceedings PODS-2006*, pp. 1–9. ACM Press, 2006.

[37] J. Heinsohn. Probabilistic description logics. In *Proceedings UAI-1994*, pp. 311–318. Morgan Kaufmann, 1994.

[38] M. Holi and E. Hyvönen. A method for modeling uncertainty in Semantic Web taxonomies. In *Proceedings WWW-2004*, pp. 296–297. ACM Press, 2004.

[39] M. Holi and E. Hyvönen. Modeling degrees of conceptual overlap in Semantic Web ontologies. In *Proceedings URSW-2005*, pp. 98–99, 2005.

[40] I. Horrocks and P. F. Patel-Schneider. Reducing OWL entailment to description logic satisfiability. In *Proceedings ISWC-2003*, volume 2870 of *LNCS*, pp. 17–29. Springer, 2003.

[41] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. *J. Web Sem.*, 1(1):7–26, 2003.

[42] E. Hung, Y. Deng, and V. S. Subrahmanian. TOSS: An extension of TAX with ontologies and similarity queries. In *Proceedings ACM SIGMOD 2004*, pp. 719–730. ACM Press, 2004.

[43] O. Udrea, Y. Deng, E. Hung, and V. S. Subrahmanian. Probabilistic ontologies and relational databases. In *Proceedings CoopIS/DOA/ODBASE-2005*, volume 3760 of *LNCS*, pp. 1–17. Springer, 2005.

[44] M. Jaeger. Probabilistic reasoning in terminological logics. In *Proceedings KR-1994*, pp. 305–316. Morgan Kaufmann, 1994.

[45] M. Jaeger. On the complexity of inference about probabilistic relational models. *Artif. Intell.*, 117(2):297–308, 2000.

[46] M. Jaeger. Probabilistic role models and the guarded fragment. In *Proc. IPMU-2004*, pp. 235–242, 2004. Extended version in *Int. J. Uncertain. Fuzz.*, 14(1):43–60, 2006.

[47] B. Jenner and J. Toran. Computing functions with parallel queries to NP. *Theor. Comput. Sci.*, 141:175–193, 1995.

[48] D. S. Johnson. A catalog of complexity classes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A, chapter 2, pp. 67–161. MIT Press, Cambridge, MA, 1990.

[49] D. Koller, A. Levy, and A. Pfeffer. P-CLASSIC: A tractable probabilistic description logic. In *Proceedings AAAI-1997*, pp. 390–397. AAAI Press / MIT Press, 1997.

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

[51] D. Lehmann. Another perspective on default reasoning. *Ann. Math. Artif. Intell.*, 15(1):61–82, 1995.

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

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

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

[55] T. Lukasiewicz. Probabilistic logic programming with conditional constraints. *ACM Trans. Comput. Log.*, 2(3):289–339, 2001.

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

[57] T. Lukasiewicz. Weak nonmonotonic probabilistic logics. *Artif. Intell.*, 168(1–2):119–161, 2005.

[58] T. Lukasiewicz. Nonmonotonic probabilistic logics under variable-strength inheritance with overriding: Complexity, algorithms, and implementation. *Int. J. Approx. Reasoning*, 44(3):301–321, 2007.

[59] T. Lukasiewicz. Probabilistic description logic programs. *Int. J. Approx. Reasoning*, 2007. In press.

[60] T. Lukasiewicz and J. Schellhase. Variable-strength conditional preferences for matchmaking in description logics. In *Proceedings KR-2006*, pp. 164–174. AAAI Press, 2006.

[61] T. Mantay, R. Möller, and A. Kaplunova. Computing probabilistic least common subsumers in description logics. In *Proceedings KI-1999*, volume 1701 of *LNCS*, pp. 89–100. Springer, 1999.

[62] P. Mitra, N. F. Noy, and A. Jaiswal. OMEN: A probabilistic ontology mapping tool. In *Proceedings ISWC-2005*, volume 3729 of *LNCS*, pp. 537–547. Springer, 2005.

[63] N. J. Nilsson. Probabilistic logic. *Artif. Intell.*, 28(1):71–88, 1986.

[64] H. Nottelmann and N. Fuhr. Adding probabilities and rules to OWL Lite subsets based on probabilistic Datalog. *Int. J. Uncertain. Fuzz.*, 14(1):17–42, 2006.

[65] R. Pan, Z. Ding, Y. Yu, and Y. Peng. A Bayesian network approach to ontology mapping. In *Proc. ISWC-2005*, volume 3729 of *LNCS*, pp. 563–577. Springer, 2005.

[66] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, Reading, 1994.

[67] J. Pearl. System Z: A natural ordering of defaults with tractable applications to default reasoning. In *Proceedings TARK-1990*, pp. 121–135. Morgan Kaufmann, 1990.

[68] M. Pool and J. Aikin. KEEPER and Protégé: An elicitation environment for Bayesian inference tools. In *Proceedings of the Workshop on Protégé and Reasoning* held at the *7th International Protégé Conference*, 2004.

[69] C. Smyth and D. Poole. Qualitative probabilistic matching with hierarchical descriptions. In *Proceedings KR-2004*, pp. 479–487. AAAI Press, 2004.

[70] A. Selman. A taxonomy of complexity classes of functions. *J. Computer and System Sciences*, 48:357–381, 1994.

[71] W. Spohn. Ordinal conditional functions: A dynamic theory of epistemic states. In W. Harper and B. Skyrms, editors, *Causation in Decision, Belief Change, and Statistics*, volume 2, pp. 105–134. Reidel, Dordrecht, Netherlands, 1988.

[72] U. Straccia. A fuzzy description logic for the Semantic Web. In E. Sanchez, editor, *Fuzzy Logic and the Semantic Web*, Capturing Intelligence, chapter 4, pp. 73–90. Elsevier, 2006.

[73] S. Tobies. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. PhD thesis, RWTH Aachen, Germany, 2001.

[74] M. van Keulen, A. de Keijzer, and W. Alink. A probabilistic XML approach to data integration. In *Proc. ICDE-2005*, pp. 459–470. IEEE Computer Society, 2005.

[75] W3C. *OWL Web Ontology Language Overview*. 2004. W3C Recommendation (10 Feb. 2004). Available at www.w3.org/TR/2004/REC-owl-features-20040210/.

[76] Y. Yang and J. Calmet. OntoBayes: An ontology-driven uncertainty model. In *Proceedings IAWTIC-2005*, pp. 457–463. IEEE Press, 2005.

[77] P. M. Yelland. An alternative combination of Bayesian networks and description logics. In *Proceedings KR-2000*, pp. 225–234. Morgan Kaufmann, 2000.