Uncertainty Reasoning for the Semantic Web

Thomas Lukasiewicz

Department of Computer Science, University of Oxford, UK
thomas.lukasiewicz@cs.ox.ac.uk

Abstract. The Semantic Web has attracted much attention, both from academia and industry. An important role in research towards the Semantic Web is played by formalisms and technologies for handling uncertainty and/or vagueness. In this paper, I first provide some motivating examples for handling uncertainty and/or vagueness in the Semantic Web. I then give an overview of some own formalisms for handling uncertainty and/or vagueness in the Semantic Web.

1 Introduction

The *Semantic Web* [1,2,3,4] aims at an extension of the current 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 it are to add a machine-understandable "meaning" to Web pages, to use ontologies for a precise definition of shared terms in Web resources, to use KR technology for automated reasoning from Web resources, and to apply cooperative agent technology for processing the information of the Web.

The Semantic Web is divided into several hierarchical layers (see Fig. 1), which include in particular the Ontology, Rules, Logic, and Proof layers. In detail, the *Ontology* layer, in the form of the *OWL Web Ontology Language* [5], consists of three increasingly expressive sublanguages, namely, *OWL Lite*, *OWL DL*, and *OWL Full*. OWL Lite and OWL DL are essentially very expressive description logics (DLs) with an RDF syntax. As shown in [6], ontology entailment in OWL Lite (resp., OWL DL) reduces to knowledge base (un)satisfiability in the description logic $\mathcal{SHIF}(\mathbf{D})$ (resp., $\mathcal{SHOIN}(\mathbf{D})$). The DL $\mathcal{SROIQ}$ [7] is one of the most expressive DLs, which is underlying OWL 2 [8], a new version of OWL. Reasoning in $\mathcal{SROIQ}$ is computationally expensive, and several more tractable languages have been proposed in the Semantic Web community. Among such languages, there are the *DL-Lite* family [9,10], $\mathcal{EL}^{++}$ [11], and DLP [12], which are underlying the OWL 2 profiles QL, EL, and RL [13], respectively. Beside and on top of the Ontology layer, there are sophisticated representation and reasoning capabilities for the *Rules*, *Logic*, and *Proof* layers of the Semantic Web.

A key requirement of the layered architecture of the Semantic Web is in particular to integrate the Rules and the Ontology layer. Here, it is crucial to allow for building rules on top of ontologies, i.e., for rule-based systems that use vocabulary from ontological knowledge bases. Another type of combination is to build ontologies on top of rules, where ontological definitions are supplemented by rules or imported from rules. Both types of integration have been realized in recent hybrid integrations of rules and ontologies, called *description logic programs* (or *dl-programs*), which are of the form

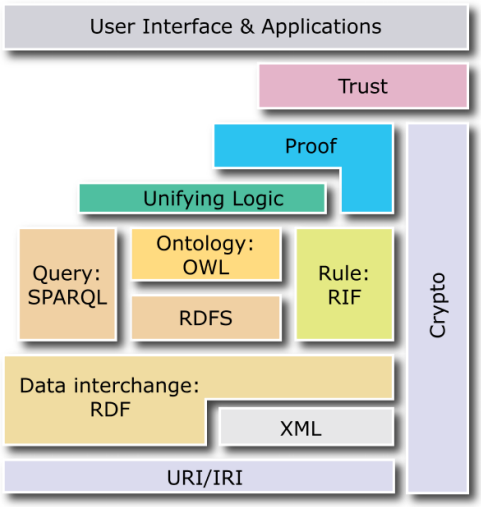

Fig. 1. Layered architecture of the Semantic Web.

$KB = (L, P)$, where L is a description logic knowledge base, and P is a finite set of rules involving either queries to L in a loose integration (see, e.g., [14,15]) or concepts and roles from L as unary resp. binary predicates in a tight integration (see, e.g., [16]).

However, classical ontology languages and description logics as well as formalisms integrating rules and ontologies are less suitable in all those domains where the information to be represented comes along with (*quantitative*) *uncertainty* and/or *vagueness* (or *imprecision*). For this reason, during the recent years, handling uncertainty and vagueness has started to play an important role in research towards the Semantic Web. A recent forum for approaches to uncertainty reasoning in the Semantic Web is the annual *International Workshop on Uncertainty Reasoning for the Semantic Web (URSW)* at the *International Semantic Web Conference (ISWC)*. There has also been a W3C Incubator Group on *Uncertainty Reasoning for the World Wide Web*. The research focuses especially on probabilistic and fuzzy extensions of description logics, ontology languages, and formalisms integrating rules and ontologies. Note that probabilistic formalisms allow to encode ambiguous information, such as "John is a student with the probability 0.7 and a teacher with the probability 0.3" (roughly, John is either a teacher or a student, but more likely a student), while fuzzy approaches allow to encode vague or imprecise information, such as "John is tall with the degree of truth 0.7" (roughly, John is quite tall). Formalisms for dealing with uncertainty and vagueness are especially applied in ontology mapping, data integration, information retrieval, and database querying. For example, some of the most prominent technologies for dealing with uncertainty are probably the ranking algorithms standing behind Web search engines. Other important applications are belief fusion and opinion pooling, recommendation systems, user preference modeling, trust and reputation modeling, and shopping agents. Vagueness

and imprecision also abound in multimedia information processing and retrieval, and are an important aspect of natural language interfaces to the Web.

In this paper, I give an overview of some own recent extensions of description logics and description logic programs by probabilistic uncertainty and fuzzy vagueness. The rest of this paper is organized as follows. Section 2 provides some motivating examples. In Section 3, I describe an approach to probabilistic description logics for the Semantic Web. Sections 4 and 5 focus on approaches to probabilistic and fuzzy description logic programs for the Semantic Web, respectively, while Section 6 describes an approach to description logic programs for handling both uncertainty and vagueness in a uniform framework for the Semantic Web. For a more detailed overview of extensions of description logics for handling uncertainty and vagueness in the Semantic Web, I also refer the reader to the survey [17].

2 Motivating Examples

We now provide some examples for the use of probabilistic ontologies and of probabilistic and vague extensions of formalisms integrating rules and ontologies.

In order 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. It is often advantageous to share such medical knowledge between hospitals and/or medical centers, for example, to follow up patients, to track medical history, for case studies research, and to get information on rare diseases and/or rare cures to diseases. The need for sharing medical knowledge is also at the core of the *W3C Semantic Web Health Care and Life Sciences Interest Group*, who state that the "key to the success of Life Science Research and Health Care is the implementation of new informatics models that will unite many forms of biological and medical information across all institutions" (see http://www.w3.org/2001/sw/hcls/).

Example 2.1 (Medical Example [18]). 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 Mary, where Tom is a heart patient, while John and Mary 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" (i.e., "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", "Mary has the symptom breathing difficulties with a probability of at least 0.6", "Mary has the symptom chest pain with a probability of at least 0.9", and "Mary's illness status is final with a probability between 0.2 and 0.8".

Uncertain medical knowledge may also be collected by a medical company from own databases and public sources (e.g., client data, web pages, web inquiries, blogs, and mailing lists) and be used in an advertising campaign for a new product.

Example 2.2 (Medical Example cont'd [18]). Suppose 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 data and web sources (e.g., own databases with data of clients and their shopping histories; and web listings of pharmacies, hospitals, and doctors along with their product portfolio resp. fields of expertise). 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 for an increased recall (which has especially been explored in [19,20]).

Example 2.3 (Literature Search [18]). 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.

We finally describe a shopping agent example, where we encounter both probabilistic uncertainty (in resource selection, ontology mapping / query transformation, and data integration) and fuzzy vagueness (in query matching with vague concepts).

Example 2.4 (Shopping Agent [44,45]). Suppose a person would like to buy "a sports car that costs at most about 22 000 € and that has a power of around 150 HP".

In todays Web, the buyer has to manually (i) search for car selling sites, e.g., using Google, (ii) select the most promising sites (e.g., http://www.autos.com), (iii) browse through them, query them to see the cars that they sell, and match the cars with our requirements, (iv) select the offers in each web site that match our requirements, and (v) eventually merge all the best offers from each site and select the best ones.

It is obvious that the whole process is rather tedious and time consuming, since, e.g., (i) the buyer has to visit many sites, (ii) the browsing in each site is very time consuming, (iii) finding the right information in a site (which has to match the requirements) is not simple, and (iv) the way of browsing and querying may differ from site to site.

A shopping agent may now support us as follows, automatizing the whole selection process once it receives the request / query q from the buyer:

- *Probabilistic Resource Selection.* The agent selects some sites / resources S that it considers as promising for the buyer's request. The agent has to select a subset of some relevant resources, since it is not reasonable to assume that it will access and query all the resources known to him. The relevance of a resource S to a query is usually (automatically) estimated as the probability $Pr(q|S)$ (the probability that the information need represented by the query q is satisfied by the searching resource S; see, e.g., [21,22]). It is not difficult to see that such probabilities can be represented by probabilistic rules.
- *Probabilistic Ontology Mapping / Query Reformulation.* For the top-k selected sites, the agent has to reformulate the buyer's query using the terminology / ontology of the specific car selling site. For this task, the agent relies on so-called transformation rules, which say how to translate a concept or property of the agent's ontology into the ontology of the information resource (which is called ontology mapping in the Semantic Web). To relate a concept B of the buyer's ontology to a concept S of the seller's ontology, one often automatically estimates the probability $P(B|S)$ that an instance of S is also an instance of B, which can then be represented as a probabilistic rule [23,24].
- *Vague Query Matching.* Once the agent has translated the buyer's request for the specific site's terminology, the agent submits the query. But the buyer's request often contains many so-called vague / fuzzy concepts such as "the price is around 22 000 € or less", rather than strict conditions, and thus a car may match the buyer's condition to a degree. As a consequence, a site / resource / web service may return a ranked list of cars, where the ranks depend on the degrees to which the sold items match the buyer's requests q.
- *Probabilistic Data Integration.* Eventually, the agent has to combine the ranked lists by considering the involved matching (or truth) degrees (vagueness) and probability degrees (uncertainty) and show the top-n items to the buyer.

3 Probabilistic Description Logics

In this section, we briefly describe the probabilistic description logic P-$\mathcal{SHOIN}(\mathbf{D})$, which is a probabilistic generalization of the description logic $\mathcal{SHOIN}(\mathbf{D})$ (behind OWL DL), directed towards sophisticated formalisms for reasoning under probabilistic uncertainty in the Semantic Web [18]. Closely related probabilistic generalizations of

the *DL-Lite* family of tractable description logics (which lies between the Semantic Web languages RDFS and OWL Lite) and the description logics $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOQ}(\mathbf{D})$ (which stand behind OWL Lite and DAML+OIL, respectively) have been introduced in [18,25]. A closely related paper [26] combines *DL-Lite* with Bayesian networks.

Probabilistic description logics allow for representing probabilistic ontologies and for reasoning about them. There is a plethora of applications with an urgent need for handling probabilistic knowledge in ontologies, especially in areas like medicine, biology, defense, and astronomy. Moreover, probabilistic ontologies allow for quantifying the degrees of overlap between the ontological concepts in the Semantic Web, reasoning about them, and using them in Semantic Web applications and systems, such as information retrieval, personalization tasks, and recommender systems. Furthermore, probabilistic ontologies can be used to align the concepts of different ontologies (called ontology mapping) and for handling inconsistencies in Semantic Web data.

The syntax of P-$\mathcal{SHOIN}(\mathbf{D})$ uses the notion of a conditional constraint from [27] to express probabilistic knowledge in addition to the axioms of $\mathcal{SHOIN}(\mathbf{D})$. Its semantics is based on the notion of lexicographic entailment in probabilistic default reasoning [28,29], which is a probabilistic generalization of the sophisticated notion of lexicographic entailment by Lehmann [30] in default reasoning from conditional knowledge bases. Due to this semantics, P-$\mathcal{SHOIN}(\mathbf{D})$ 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 statistical knowledge about concepts and roles, and as degrees of belief about instances of concepts and roles, respectively, and allows for deriving both statistical knowledge and degrees of belief. 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 as in Lehmann's lexicographic default entailment [30].

Example 3.1. Suppose a classical description logic knowledge base T is used to encode knowledge about cars and their properties (e.g., that sports cars and roadsters are cars). A probabilistic knowledge base $KB = (T, P, (P_o)_{o \in \mathbf{I}_P})$ in P-$\mathcal{SHOIN}(\mathbf{D})$ then extends T by terminological default and terminological probabilistic knowledge in P as well as by assertional probabilistic knowledge in P_o for certain objects $o \in \mathbf{I}_P$. For example, the terminological default knowledge (1) "generally, cars do not have a red color" and (2) "generally, sports cars have a red color", and the terminological probabilistic knowledge (3) "cars have four wheels with a probability of at least 0.9", can be expressed by the following conditional constraints in P:

$$\begin{array}{ll} (1) & (\neg\exists\, \mathit{HasColor}.\{\mathit{red}\} \mid \mathit{Car})[1,1], \\ (2) & (\exists\, \mathit{HasColor}.\{\mathit{red}\} \mid \mathit{SportsCar})[1,1], \\ (3) & (\mathit{HasFourWheels} \mid \mathit{Car})[0.9,1]\,. \end{array}$$

Suppose we want to encode some probabilistic information about John's car (which we have not seen so far). Then, the set of probabilistic individuals $\mathbf{I}_P$ contains the individual *John's car*, and the assertional probabilistic knowledge (4) "John's car is a sports car with a probability of at least 0.8" (we know that John likes sports cars) can

be expressed by the following conditional constraint in $P_{\textit{John's car}}$:

$$(4)\ \ (\textit{SportsCar} \mid \top)[0.8, 1]\,.$$

Then, the following are some (terminological default and terminological probabilistic) tight lexicographic consequences of $PT = (T, P)$:

$$\begin{array}{l}(\neg\exists\, \textit{HasColor}.\{\textit{red}\} \mid \textit{Car})[1, 1],\\ (\exists\, \textit{HasColor}.\{\textit{red}\} \mid \textit{SportsCar})[1, 1],\\ (\textit{HasFourWheels} \mid \textit{Car})[0.9, 1],\\ (\neg\exists\, \textit{HasColor}.\{\textit{red}\} \mid \textit{Roadster})[1, 1],\\ (\textit{HasFourWheels} \mid \textit{SportsCar})[0.9, 1],\\ (\textit{HasFourWheels} \mid \textit{Roadster})[0.9, 1]\,.\end{array}$$

Hence, in addition to the sentences (1) to (3) directly encoded in P, we also conclude "generally, roadsters do not have a red color", "sports cars have four wheels with a probability of at least 0.9", and "roadsters have four wheels with a probability of at least 0.9". Observe here that the default property of not having a red color and the probabilistic property of having four wheels with a probability of at least 0.9 are inherited from cars down to roadsters. Roughly, the tight lexicographic 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 *John's car* are some (assertional probabilistic) tight lexicographic consequences of KB, which informally say that John's car is a sports car, has a red color, and has four wheels with probabilities of at least 0.8, 0.8, and 0.72, respectively:

$$\begin{array}{l}(\textit{SportsCar} \mid \top)[0.8, 1],\\ (\exists\, \textit{HasColor}.\{\textit{red}\} \mid \top)[0.8, 1],\\ (\textit{HasFourWheels} \mid \top)[0.72, 1]\,.\end{array}$$

4 Probabilistic Description Logic Programs

We now summarize the main ideas behind loosely and tightly coupled probabilistic dl-programs, introduced in [31,32,33,34] and [35,36,37,38,39], respectively. For further details on the syntax and semantics of these programs, their background, and their semantic and computational properties, we refer to the above works.

Loosely coupled probabilistic dl-programs [31,32,33] are a combination of loosely coupled dl-programs under the answer set and the well-founded semantics with probabilistic uncertainty as in Bayesian networks. Roughly, they consist of a loosely coupled dl-program (L, P) under different "total choices" B (they are the full joint instantiations of a set of random variables, and they serve as pairwise exclusive and exhaustive possible worlds), and a probability distribution μ over the set of total choices B. One then obtains a probability distribution over Herbrand models, since every total choice B along with the loosely coupled dl-program produces a set of Herbrand models of which

the probabilities sum up to $\mu(B)$. As in the classical case, the answer set semantics of loosely coupled probabilistic dl-programs is a refinement of the well-founded semantics of loosely coupled probabilistic dl-programs. Consistency checking and tight query processing (i.e., computing the entailed tight interval for the probability of a conditional or unconditional event) in such probabilistic dl-programs under the answer set semantics can be reduced to consistency checking and query processing in loosely coupled dl-programs under the answer set semantics, while tight query processing under the well-founded semantics can be done in an anytime fashion by reduction to loosely coupled dl-programs under the well-founded semantics. For suitably restricted description logic components, the latter can be done in polynomial time in the data complexity. Query processing for stratified loosely coupled probabilistic dl-programs can be reduced to computing the canonical model of stratified loosely coupled dl-programs. Loosely coupled probabilistic dl-programs can especially be used for (database-oriented) probabilistic data integration in the Semantic Web, where probabilistic uncertainty is used to handle inconsistencies between different data sources [34].

Example 4.1. A university database may use a loosely coupled dl-program (L, P) to encode ontological and rule-based knowledge about students and exams. A probabilistic dl-program $KB = (L, P', C, \mu)$ then additionally allows for encoding probabilistic knowledge. For example, the following two probabilistic rules in P' along with a probability distribution on a set of random variables may express that if two master (resp., bachelor) students have given the same exam, then there is a probability of 0.9 (resp., 0.7) that they are friends:

$$\begin{aligned} \mathit{friends}(X,Y) \leftarrow\ & \mathit{given\_same\_exam}(X,Y), DL[\mathit{master\_student}(X)], \\ & DL[\mathit{master\_student}(Y)],\ \mathit{choice}_m\,; \\ \mathit{friends}(X,Y) \leftarrow\ & \mathit{given\_same\_exam}(X,Y), DL[\mathit{bachelor\_student}(X)], \\ & DL[\mathit{bachelor\_student}(Y)],\ \mathit{choice}_b\,. \end{aligned}$$

Here, we assume the set $C = \{V_m, V_b\}$ of value sets $V_m = \{\mathit{choice}_m, \mathit{not\_choice}_m\}$ and $V_b = \{\mathit{choice}_b, \mathit{not\_choice}_b\}$ of two random variables X_m resp. X_b and the probability distribution μ on all their joint instantiations, given by $\mu\colon\ \mathit{choice}_m, \mathit{not\_choice}_m, \mathit{choice}_b, \mathit{not\_choice}_b \mapsto 0.9, 0.1, 0.7, 0.3$ under probabilistic independence. For example, the joint instantiation $\mathit{choice}_m, \mathit{choice}_b$ is associated with the probability $0.9 \times 0.7 = 0.63$. Asking about the entailed tight interval for the probability that john and bill are friends can then be expressed by a probabilistic query $\exists(\mathit{friends}(\mathit{john}, \mathit{bill}))[R, S]$, whose answer depends on the available concrete knowledge about john and bill (namely, whether they have given the same exams, and are both master or bachelor students).

Tightly coupled probabilistic dl-programs [35,36] are a tight combination of disjunctive logic programs under the answer set semantics with description logics and Bayesian probabilities. They are a logic-based representation formalism that naturally fits into the landscape of Semantic Web languages. Tightly coupled probabilistic dl-programs can especially be used for representing mappings between ontologies [37,38], which are a common way of approaching the semantic heterogeneity problem on the Semantic Web. Here, they allow in particular for resolving inconsistencies and for merging

mappings from different matchers based on the level of confidence assigned to different rules (see below). Furthermore, tightly coupled probabilistic description logic programs also provide a natural integration of ontologies, action languages, and Bayesian probabilities towards Web Services. Consistency checking and query processing in tightly coupled probabilistic dl-programs can be reduced to consistency checking and cautious/brave reasoning, respectively, in tightly coupled disjunctive dl-programs. Under certain restrictions, these problems have a polynomial data complexity.

Example 4.2. The two correspondences between two ontologies O_1 and O_2 that (i) an element of $Collection$ in O_1 is an element of $Book$ in O_2 with the probability 0.62, and (ii) an element of $Proceedings$ in O_1 is an element of $Proceedings$ in O_2 with the probability 0.73 (found by the matching system hmatch) can be expressed by the following two probabilistic rules:

$$O_2 : Book(X) \leftarrow O_1 : Collection(X) \wedge hmatch_1;$$
$$O_2 : Proceedings(X) \leftarrow O_1 : Proceedings(X) \wedge hmatch_2.$$

Here, we assume the set $\mathcal{C} = \{\{hmatch_i, not\_hmatch_i\} \mid i \in \{1,2\}\}$ of values of two random variables and the probability distribution μ on all joint instantiations of these variables, given by $\mu\colon hmatch_1, not\_hmatch_1, hmatch_2, not\_hmatch_2 \mapsto 0.62, 0.38, 0.73, 0.27$ under probabilistic independence.

Similarly, two other correspondences between O_1 and O_2 (found by the matching system falcon) are expressed by the following two probabilistic rules:

$$O_2 : InCollection(X) \leftarrow O_1 : Collection(X) \wedge falcon_1;$$
$$O_2 : Proceedings(X) \leftarrow O_1 : Proceedings(X) \wedge falcon_2,$$

where we assume the set $\mathcal{C}' = \{\{falcon_i, not\_falcon_i\} \mid i \in \{1,2\}\}$ of values of two random variables and the probability distribution μ' on all joint instantiations of these variables, given by $\mu'\colon falcon_1, not\_falcon_1, falcon_2, not\_falcon_2 \mapsto 0.94, 0.06, 0.96, 0.04$ under probabilistic independence.

Using the trust probabilities 0.55 and 0.45 for hmatch and falcon, respectively, for resolving inconsistencies between rules, we can now define a merged mapping set that consists of the following probabilistic rules:

$$O_2 : Book(X) \leftarrow O_1 : Collection(X) \wedge hmatch_1 \wedge sel\_hmatch_1;$$
$$O_2 : InCollection(X) \leftarrow O_1 : Collection(X) \wedge falcon_1 \wedge sel\_falcon_1;$$
$$O_2 : Proceedings(X) \leftarrow O_1 : Proceedings(X) \wedge hmatch_2;$$
$$O_2 : Proceedings(X) \leftarrow O_1 : Proceedings(X) \wedge falcon_2.$$

Here, we assume the set $\mathcal{C}''$ of values of random variables and the probability distribution μ'' on all joint instantiations of these variables, which are obtained from $\mathcal{C} \cup \mathcal{C}'$ and $\mu \cdot \mu'$ (defined as $(\mu \cdot \mu')(B\,B') = \mu(B) \cdot \mu'(B')$, for all joint instantiations B of $\mathcal{C}$ and B' of $\mathcal{C}'$), respectively, by adding the values $\{sel\_hmatch_1, sel\_falcon_1\}$ of a new random variable, with the probabilities $sel\_hmatch_1, sel\_falcon_1 \mapsto 0.55, 0.45$ under probabilistic independence, for resolving the inconsistency between the first two rules.

A companion approach to probabilistic description logic programs [39] combines probabilistic logic programs, probabilistic default theories, and the description logics behind OWL Lite and OWL DL. It is based on new notions of entailment for reasoning with conditional constraints, which realize the principle of inheritance with overriding for both classical and purely probabilistic knowledge. They are obtained by generalizing previous formalisms for probabilistic default reasoning with conditional constraints (similarly as for P-$\mathcal{SHOIN}(\mathbf{D})$ in Section 3). In addition to dealing with probabilistic knowledge, these notions of entailment thus also allow for handling default knowledge.

5 Fuzzy Description Logic Programs

We next briefly describe loosely and tightly coupled fuzzy dl-programs, which have been introduced in [40,41] and [42,43], respectively, and extended by a top-k retrieval technique in [46]. All these fuzzy dl-programs have natural special cases where query processing can be done in polynomial time in the data complexity. For further details on their syntax and semantics, background, and properties, we refer to the above works.

Towards dealing with vagueness and imprecision in the reasoning layers of the Semantic Web, loosely coupled (normal) fuzzy dl-programs under the answer set semantics [40,41] generalize normal dl-programs under the answer set semantics by fuzzy vagueness and imprecision in both the description logic and the logic program component. This is the first approach to fuzzy dl-programs that may contain default negations in rule bodies. Query processing in such fuzzy dl-programs can be done by reduction to normal dl-programs under the answer set semantics. In the special cases of positive and stratified loosely coupled fuzzy dl-programs, the answer set semantics coincides with a canonical least model and an iterative least model semantics, respectively, and has a characterization in terms of a fixpoint and an iterative fixpoint semantics, respectively.

Example 5.1. Consider the fuzzy description logic knowledge base L of a car shopping Web site, which defines especially (i) the fuzzy concepts of sports cars ($SportsCar$), "at most 22 000 €" ($LeqAbout22000$), and "around 150 horse power" ($Around150HP$), (ii) the attributes of the price and of the horse power of a car ($hasInvoice$ resp. $hasHP$), and (iii) the properties of some concrete cars (such as a $MazdaMX5Miata$ and a $MitsubishiES$). Then, a loosely coupled fuzzy dl-program $KB = (L, P)$ is given by the set of fuzzy dl-rules P, which contains only the following fuzzy dl-rule encoding the request of a buyer (asking for a sports car costing at most 22 000 € and having around 150 horse power), where $\otimes$ may be the conjunction strategy of, e.g., Gödel Logic (i.e., $x \otimes y = \min(x, y)$, for all $x, y \in [0, 1]$, is used to evaluate $\wedge$ and $\leftarrow$ on truth values):

$$\begin{aligned} query(x) \leftarrow_{\otimes} \; & DL[SportsCar](x) \wedge_{\otimes} DL[\exists hasInvoice.LeqAbout22000](x) \wedge_{\otimes} \\ & DL[\exists hasHP.Around150HP](x) \geqslant 1\,. \end{aligned}$$

The above fuzzy dl-program $KB = (L, P)$ is positive (i.e., without default negation), and has a minimal model M_{KB}, which defines the degree to which some concrete cars in the description logic knowledge base L match the buyer's request, for example,

$$M_{KB}(query(MazdaMX5Miata)) = 0.36\,, \quad M_{KB}(query(MitsubishiES)) = 0.32\,.$$

That is, the car $MazdaMX5Miata$ is ranked top with the degree 0.36, while the car $MitsubishiES$ is ranked second with the degree 0.32.

Tightly coupled fuzzy dl-programs under the answer set semantics [42,43] are a tight integration of fuzzy disjunctive logic programs under the answer set semantics with fuzzy description logics. They are also a generalization of tightly coupled disjunctive dl-programs by fuzzy vagueness in both the description logic and the logic program component. This is the first approach to fuzzy dl-programs that may contain disjunctions in rule heads. Query processing in such programs can essentially be done by a reduction to tightly coupled disjunctive dl-programs. A closely related work [46] explores the evaluation of ranked top-k queries. It shows in particular how to compute the top-k answers in data-complexity tractable tightly coupled fuzzy dl-programs.

Example 5.2. A tightly coupled fuzzy dl-program $KB = (L, P)$ is given by a suitable fuzzy description logic knowledge base L and the set of fuzzy rules P, which contains only the following fuzzy rule (where $x \otimes y = \min(x, y)$):

$$\begin{aligned} query(x) \leftarrow_{\otimes} \; & SportyCar(x) \wedge_{\otimes} hasInvoice(x, y_1) \wedge_{\otimes} hasHorsePower(x, y_2) \wedge_{\otimes} \\ & LeqAbout22000(y_1) \wedge_{\otimes} Around150(y_2) \geqslant 1 \,. \end{aligned}$$

Informally, $query$ collects all sports cars, and ranks them according to whether they cost at most around 22 000 € and have around 150 HP. Another fuzzy rule involving also a negation in its body and a disjunction in its head is given as follows (where $\ominus x = 1 - x$ and $x \oplus y = \max(x, y)$):

$$\begin{aligned} Small(x) \vee_{\oplus} Old(x) \leftarrow_{\otimes} \; & Car(x) \wedge_{\otimes} hasInvoice(x, y) \wedge_{\otimes} \\ & not_{\ominus} GeqAbout15000(y) \geqslant 0.7 \,. \end{aligned}$$

This rule says that a car costing at most around 15 000 € is either small or old. Notice here that $Small$ and Old may be two concepts in the fuzzy description logic knowledge base L. That is, the tightly coupled approach to fuzzy dl-programs under the answer set semantics also allows for using the rules in P to express relationships between the concepts and roles in L. This is not possible in the loosely coupled approach to fuzzy dl-programs under the answer set semantics in [40,41], since the dl-queries there can only occur in rule bodies, but not in rule heads.

6 Probabilistic Fuzzy Description Logic Programs

We finally describe (loosely coupled) probabilistic fuzzy dl-programs [44,45], which combine fuzzy description logics, fuzzy logic programs (with stratified default-negation), and probabilistic uncertainty in a uniform framework for the Semantic Web. Intuitively, they allow for defining several rankings on ground atoms using fuzzy vagueness, and then for merging these rankings using probabilistic uncertainty (by associating with each ranking a probabilistic weight and building the weighted sum of all rankings). Such programs also give rise to important concepts dealing with both probabilistic uncertainty and fuzzy vagueness, such as the expected truth value of a crisp sentence and the probability of a vague sentence. Probabilistic fuzzy dl-programs can be used to model a shopping agent as described in Example 2.4.

Example 6.1. A (loosely coupled) probabilistic fuzzy dl-program is given by a suitable fuzzy description logic knowledge base L and the following set of fuzzy dl-rules P, modeling some query reformulation / retrieval steps using ontology mapping rules:

$$\begin{aligned} query(x) \leftarrow_{\otimes} {} & SportyCar(x) \wedge_{\otimes} hasPrice(x, y_1) \wedge_{\otimes} hasPower(x, y_2) \wedge_{\otimes} \\ & DL[LeqAbout22000](y_1) \wedge_{\otimes} DL[Around150HP](y_2) \geqslant 1\,, \qquad (1) \\ SportyCar(x) \leftarrow_{\otimes} {} & DL[SportsCar](x) \wedge_{\otimes} sc_{pos} \geqslant 0.9\,, \qquad (2) \\ hasPrice(x, y) \leftarrow_{\otimes} {} & DL[hasInvoice](x, y) \wedge_{\otimes} hi_{pos} \geqslant 0.8\,, \qquad (3) \\ hasPower(x, y) \leftarrow_{\otimes} {} & DL[hasHP](x, y) \wedge_{\otimes} hhp_{pos} \geqslant 0.8\,, \qquad (4) \end{aligned}$$

where we assume the set $C = \{\{sc_{pos}, sc_{neg}\}, \{hi_{pos}, hi_{neg}\}, \{hhp_{pos}, hhp_{neg}\}\}$ of values of random variables and the probability distribution μ on all joint instantiations of these variables, given by $\mu\colon sc_{pos}, sc_{neg}, hi_{pos}, hi_{neg}, hhp_{pos}, hhp_{neg} \mapsto 0.91, 0.09, 0.78, 0.22, 0.83, 0.17$ under probabilistic independence. Here, rule (1) is the buyer's request, but in a "different" terminology than the one of the car selling site. Rules (2)–(4) are so-called ontology alignment mapping rules. For example, rule (2) states that the predicate "SportyCar" of the buyer's terminology refers to the concept "SportsCar" of the selected site with probability 0.91.

The following are some tight consequences of the above probabilistic fuzzy dl-program (where for ground atoms q, we use $(\mathbf{E}[q])[L, U]$ to denote that the expected truth value of q lies in the interval $[L, U]$):

$$(\mathbf{E}[query(MazdaMX5Miata)])[0.21, 0.21],\ (\mathbf{E}[query(MitsubishiES)])[0.19, 0.19]\,.$$

That is, the car $MazdaMX5Miata$ is ranked first with the degree 0.21, while the car $MitsubishiES$ is ranked second with the degree 0.19.

Acknowledgments. This work was supported the UK EPSRC grants EP/J008346/1, EP/L012138/1, EP/M025268/1, and EP/N510129/1.

References

1. Berners-Lee, T.: Weaving the Web. Harper, San Francisco (1999)
2. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Sci. Amer. **284**(5) (2001) 34–43
3. Fensel, D., Wahlster, W., Lieberman, H., Hendler, J., eds.: Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential. MIT Press (2002)
4. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. J. Web Sem. **1**(1) (2003) 7–26
5. W3C: OWL Web Ontology Language Overview (2004) W3C Recommendation (10 February 2004). Available at: http://www.w3.org/TR/2004/REC-owl-features-20040210/.
6. Horrocks, I., Patel-Schneider, P.F.: Reducing OWL entailment to description logic satisfiability. In: Proceedings ISWC-2003. *LNCS* 2870, Springer (2003) 17–29
7. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible $\mathcal{SROIQ}$. In: *Proceedings KR-2006* (2006) 57–67

8. W3C: OWL 2 Web Ontology Language Document Overview (2009) W3C Recommendation (27 October 2009). Available at: http://www.w3.org/TR/owl2-overview/.
9. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. Autom. Reasoning* **39**(3) (2007) 385–429
10. Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Linking data to ontologies. *J. Data Semantics* **10** (2008) 133–173
11. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: *Proceedings IJCAI-2005* (2005) 364–369
12. Grosof, B. N., Horrocks, I., Volz, R., Decker, S.: Description logic programs: Combining logic programs with description logic. In: *Proceedings WWW-2003* (2003) 48–57
13. W3C: OWL 2 Web Ontology Language Profiles (2009) W3C Recommendation (27 October 2009). Available at: http://www.w3.org/TR/owl2-profiles/.
14. Eiter, T., Ianni, G., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Combining answer set programming with description logics for the Semantic Web. Artif. Intell. **172**(12/13) (2008) 1495–1539
15. Eiter, T., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Well-founded semantics for description logic programs in the Semantic Web. In: Proceedings RuleML-2004. *LNCS* 3323, Springer (2004) 81–97
16. Lukasiewicz, T.: A novel combination of answer set programming with description logics for the Semantic Web. In: Proceedings ESWC-2007. *LNCS* 4519, Springer (2007) 384–398
17. Lukasiewicz, T., Straccia, U.: Managing uncertainty and vagueness in description logics for the Semantic Web. J. Web Sem. **6**(4) (2008) 291–308
18. Lukasiewicz, T.: Expressive probabilistic description logics. Artif. Intell. **172**(6/7) (2008) 852–883
19. Udrea, O., Deng, Y., Hung, E., Subrahmanian, V.S.: Probabilistic ontologies and relational databases. In: Proceedings CoopIS/DOA/ODBASE-2005. *LNCS* 3760, Springer (2005) 1–17
20. Hung, E., Deng, Y., Subrahmanian, V.S.: TOSS: An extension of TAX with ontologies and similarity queries. In: Proceedings ACM SIGMOD 2004. ACM Press (2004) 719–730
21. Callan, J.: Distributed information retrieval. In: Croft, W.B., editor, Advances in Information Retrieval. Kluwer (2000) 127–150
22. Fuhr, N.: A decision-theoretic approach to database selection in networked IR. ACM Trans. Inf. Syst. **3**(17) (1999) 229–249
23. Straccia, U., Troncy, R.: Towards distributed information retrieval in the Semantic Web. In: Proceedings ESWC-2006. *LNCS* 4011, Springer (2006) 378–392
24. Nottelmann, H., Straccia, U.: Information retrieval and machine learning for probabilistic schema matching. Inf. Process. Manage. **43**(3) (2007) 552–576
25. Giugno, R., Lukasiewicz, T.: P-$\mathcal{SHOQ}(\mathbf{D})$: A probabilistic extension of $\mathcal{SHOQ}(\mathbf{D})$ for probabilistic ontologies in the Semantic Web. In: Proceedings JELIA-2002. *LNCS* 2424, Springer (2002) 86–97
26. d'Amato, C., Fanizzi, N., Lukasiewicz, T.: Tractable reasoning with Bayesian description logics. In: Proceedings SUM-2008. *LNCS* 5291, Springer (2008) 146–159
27. Lukasiewicz, T.: Probabilistic deduction with conditional constraints over basic events. J. Artif. Intell. Res. **10** (1999) 199–241
28. Lukasiewicz, T.: Probabilistic logic programming under inheritance with overriding. In: Proceedings UAI-2001, Morgan Kaufmann (2001) 329–336
29. Lukasiewicz, T.: Probabilistic default reasoning with conditional constraints. Ann. Math. Artif. Intell. **34**(1–3) (2002) 35–88
30. Lehmann, D.: Another perspective on default reasoning. Ann. Math. Artif. Intell. **15**(1) (1995) 61–82

31. Lukasiewicz, T.: Probabilistic description logic programs. In: Proceedings ECSQARU-2005. *LNCS* 3571, Springer (2005) 737–749
32. Lukasiewicz, T.: Probabilistic description logic programs. Int. J. Approx. Reasoning **45**(2) (2007) 288–307
33. Lukasiewicz, T.: Tractable probabilistic description logic programs. In: Proceedings SUM-2007. *LNCS* 4772, Springer (2007) 143–156
34. Calì, A., Lukasiewicz, T.: An approach to probabilistic data integration for the Semantic Web. In: Uncertainty Reasoning for the Semantic Web I. *LNCS* 5327, Springer (2008) 52–65
35. Calì, A., Lukasiewicz, T.: Tightly integrated probabilistic description logic programs for the Semantic Web. In: Proceedings ICLP-2007. *LNCS* 4670, Springer (2007) 428–429
36. Calì, A., Lukasiewicz, T., Predoiu, L., Stuckenschmidt, H.: Tightly coupled probabilistic description logic programs for the Semantic Web. J. Data Sem. **12** (2009) 95–130
37. Calì, A., Lukasiewicz, T., Predoiu, L., Stuckenschmidt, H.: Rule-based approaches for representing probabilistic ontology mappings. In: Uncertainty Reasoning for the Semantic Web I. *LNCS* 5327, Springer (2008) 66–87
38. Calì, A., Lukasiewicz, T., Predoiu, L., Stuckenschmidt, H.: Tightly integrated probabilistic description logic programs for representing ontology mappings. In: Proceedings FoIKS-2008. *LNCS* 4932, Springer (2008) 178–198
39. Lukasiewicz, T.: Probabilistic description logic programs under inheritance with overriding for the Semantic Web. Int. J. Approx. Reasoning **49**(1) (2008) 18–34
40. Lukasiewicz, T.: Fuzzy description logic programs under the answer set semantics for the Semantic Web. In: Proceedings RuleML-2006, IEEE Computer Society (2006) 89–96
41. Lukasiewicz, T.: Fuzzy description logic programs under the answer set semantics for the Semantic Web. Fundam. Inform. **82**(3) (2008) 289–310
42. Lukasiewicz, T., Straccia, U.: Tightly integrated fuzzy description logic programs under the answer set semantics for the Semantic Web. In: Proceedings RR-2007. *LNCS* 4524, Springer (2007) 289–298
43. Lukasiewicz, T., Straccia, U.: Tightly coupled fuzzy description logic programs under the answer set semantics for the Semantic Web. Int. J. Semantic Web Inf. Syst. **4**(3) (2008) 68–89
44. Lukasiewicz, T., Straccia, U.: Description logic programs under probabilistic uncertainty and fuzzy vagueness. In: Proceedings ECSQARU-2007. *LNCS* 4724, Springer (2007) 187–198
45. Lukasiewicz, T., Straccia, U.: Description logic programs under probabilistic uncertainty and fuzzy vagueness. Int. J. Approx. Reasoning **50**(6) (2009) 837–853
46. Lukasiewicz, T., Straccia, U.: Top-k retrieval in description logic programs under vagueness for the Semantic Web. In: Proceedings SUM-2007. *LNCS* 4772, Springer (2007) 16–30