Description Logic Programs Under Probabilistic Uncertainty and Fuzzy Vagueness

Thomas Lukasiewicz[1,2] and Umberto Straccia[3]

[1] DIS, Sapienza Università di Roma, Via Ariosto 25, I-00185 Rome, Italy
lukasiewicz@dis.uniroma1.it
[2] Institut für Informationssysteme, Technische Universität Wien, Favoritenstraße 9-11, A-1040 Vienna, Austria
lukasiewicz@kr.tuwien.ac.at
[3] ISTI-CNR, Via G. Moruzzi 1, I-56124 Pisa, Italy
straccia@isti.cnr.it

Abstract. This paper is directed towards an infrastructure for handling both uncertainty and vagueness in the Rules, Logic, and Proof layers of the Semantic Web. More concretely, we present probabilistic fuzzy description logic programs, which combine fuzzy description logics, fuzzy logic programs (with stratified nonmonotonic negation), and probabilistic uncertainty in a uniform framework for the Semantic Web. We define 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. Furthermore, we describe a shopping agent example, which gives evidence of the usefulness of probabilistic fuzzy description logic programs in realistic web applications. In the extended report, we also provide algorithms for query processing in probabilistic fuzzy description logic programs, and we delineate a special case where query processing can be done in polynomial time in the data complexity.

1 Introduction

The *Semantic Web* [1,4] 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 it are to add a machine-readable 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 consists of several hierarchical layers, where the *Ontology layer*, in form of the *OWL Web Ontology Language* [19], is currently the highest layer of sufficient maturity. OWL 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 with an RDF syntax. As shown in [8], ontology entailment in OWL Lite (resp., OWL DL) reduces to knowledge base (un)satisfiability in the description logic $\mathcal{SHIF}(\mathbf{D})$ (resp., $\mathcal{SHOIN}(\mathbf{D})$). On top of the Ontology layer, sophisticated representation and reasoning capabilities for the *Rules*, *Logic*, and *Proof layers* of the Semantic Web are being developed next.

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 187–198, 2007.

In particular, a key requirement of the layered architecture of the Semantic Web is to integrate the Rules and the Ontology layer. Here, it is crucial to allow for building rules on top of ontologies, that is, for rule-based systems that use vocabulary from ontology 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 under the loose coupling, called *(loosely coupled) description logic programs* (or simply *dl-programs*), which have the form $KB = (L, P)$, where L is a description logic knowledge base, and P is a finite set of rules involving queries to L [3].

Other research efforts are directed towards *handling uncertainty and vagueness in the Semantic Web*, which are motivated by important web and semantic web applications. In particular, formalisms for handling uncertainty are used in data integration, ontology mapping, and information retrieval, while dealing with vagueness is motivated by multimedia information processing / retrieval and natural language interfaces to the Web. There are several extensions of description logics and web ontology languages by probabilistic uncertainty and fuzzy vagueness. Similarly, there are also extensions of description logic programs by probabilistic uncertainty [9] and fuzzy vagueness [16,10].

Clearly, since uncertainty and vagueness are semantically quite different, it is important to have a unifying formalism for the Semantic Web, which allows for dealing with both uncertainty and vagueness. But even though there has been some important work in the fuzzy logic community in this direction [5], to date there are no approaches to description logic programs that allow for handling both uncertainty and vagueness.

In this paper, we try to fill this gap. We present a novel approach to description logic programs, where probabilistic rules are defined on top of fuzzy rules, which are in turn defined on top of fuzzy description logics. This approach allows for handling both probabilistic uncertainty and fuzzy vagueness. Intuitively, it allows 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). The main contributions are as follows:

- We present probabilistic fuzzy description logic programs, which combine (i) fuzzy description logics, (ii) fuzzy logic programs (with stratified default negation), and (iii) probabilistic uncertainty in a uniform framework for the Semantic Web.
- Such programs allow for handling both probabilistic uncertainty (especially for probabilistic ontology mapping and probabilistic data integration) and fuzzy vagueness (especially for dealing with vague concepts). We define 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.
- We describe a shopping agent example, which gives evidence of the usefulness of probabilistic fuzzy description logic programs in realistic web applications.
- In the extended report [11], we also give algorithms for query processing in probabilistic fuzzy description logic programs, and we delineate a special case where query processing is data tractable (under suitable assumptions about the underlying fuzzy description logics), which is an important feature for the Web.

The rest of this paper is organized as follows. Section 2 gives a motivating example. In Section 3, we recall combination strategies and fuzzy description logics. Section 4

defines fuzzy dl-programs on top of fuzzy description logics. In Section 5, we then define probabilistic fuzzy dl-programs. Section 6 summarizes our main results and gives an outlook on future research. Note that algorithms for query processing and data tractability results as well as further technical details are given in the extended report [11].

2 Motivating Example

In this section, we 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 2.1 (Shopping Agent). 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. [2,6]). 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 [17,18,12].
- *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

Table 1. Combination strategies of various fuzzy logics

	Łukasiewicz Logic	Gödel Logic	Product Logic	Zadeh Logic
$a \otimes b$	$\max(a+b-1,0)$	$\min(a,b)$	$a \cdot b$	$\min(a,b)$
$a \oplus b$	$\min(a+b,1)$	$\max(a,b)$	$a+b-a \cdot b$	$\max(a,b)$
$a \rhd b$	$\min(1-a+b,1)$	$\begin{cases} 1 & \text{if } a \leqslant b \\ b & \text{otherwise} \end{cases}$	$\min(1,b/a)$	$\max(1-a,b)$
$\ominus a$	$1-a$	$\begin{cases} 1 & \text{if } a=0 \\ 0 & \text{otherwise} \end{cases}$	$\begin{cases} 1 & \text{if } a=0 \\ 0 & \text{otherwise} \end{cases}$	$1-a$

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 (see e.g. [14]) by considering the involved matching (or truth) degrees (vagueness) and probability degrees (uncertainty) and show the top-n items to the buyer.

3 Preliminaries

In this section, we review combination strategies and fuzzy description logics, mainly through some examples; more details are given in the extended report [11].

Combination Strategies. Rather than being restricted to a binary truth value among $\mathbf{false}$ and $\mathbf{true}$, *vague propositions* may also have a truth value strictly between $\mathbf{false}$ and $\mathbf{true}$. In the sequel, we use the unit interval $[0,1]$ as the set of all possible truth values, where 0 and 1 represent the ordinary binary truth values $\mathbf{false}$ and $\mathbf{true}$, respectively. For example, the vague proposition "John is a tall man" may be more or less true, and it is thus associated with a truth value in $[0,1]$, depending on the body height of John. To combine and modify the truth values in $[0,1]$, we assume *combination strategies*, namely, *conjunction*, *disjunction*, *implication*, and *negation strategies*, denoted $\otimes$, $\oplus$, $\rhd$, and $\ominus$, respectively, which are functions $\otimes, \oplus, \rhd\colon [0,1] \times [0,1] \to [0,1]$ and $\ominus\colon [0,1] \to [0,1]$ that generalize the ordinary Boolean operators $\wedge$, $\vee$, $\rightarrow$, and $\neg$, respectively, to the set of truth values $[0,1]$. As usual, we assume that combination strategies have some natural algebraic properties. Note that conjunction and disjunction strategies are also called *triangular norms* and *triangular co-norms* [7], respectively.

Example 3.1. The combination strategies of various fuzzy logics are shown in Table 1.

Fuzzy Description Logics. Intuitively, description logics model a domain of interest in terms of concepts and roles, which represent classes of individuals resp. binary relations between classes of individuals. A knowledge base encodes in particular subset relationships between concepts, subset relationships between roles, the membership of individuals to concepts, and the membership of pairs of individuals to roles. In fuzzy description logics, these relationships and memberships then have a degree of truth in $[0,1]$.

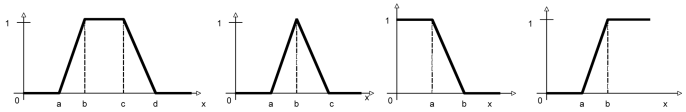

Fig. 1. (a) Tra-function, (b) Tri-function, (c) L-function, and (d) R-function

We assume fuzzy generalizations of the description logics $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$ behind OWL Lite and OWL DL, respectively. We now describe the syntax of fuzzy $\mathcal{SHIF}(\mathbf{D})$ and fuzzy $\mathcal{SHOIN}(\mathbf{D})$ (see especially [15]) and illustrate it through an example. For a formal semantics and more details see [11]; for an implementation of fuzzy $\mathcal{SHIF}(\mathbf{D})$, the *fuzzyDL* system, see http://gaia.isti.cnr.it/∼straccia.

The elementary ingredients are as follows. We assume a set of *data values*, a set of *elementary datatypes*, and a set of *datatype predicates* (each with a predefined arity $n \geqslant 1$). A *datatype* is an elementary datatype or a finite set of data values. A *fuzzy 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 data value an element of $\Delta^{\mathbf{D}}$, to each elementary datatype a subset of $\Delta^{\mathbf{D}}$, and to each datatype predicate of arity n a fuzzy relation over $\Delta^{\mathbf{D}}$ of arity n (that is, a mapping $(\Delta^{\mathbf{D}})^n \rightarrow [0,1]$). We extend $\cdot^{\mathbf{D}}$ to all datatypes by $\{v_1, \ldots, v_n\}^{\mathbf{D}} = \{v_1^{\mathbf{D}}, \ldots, v_n^{\mathbf{D}}\}$. Non-crisp predicates are usually defined by functions for specifying fuzzy set membership degrees, such as the trapezoidal, triangular, left shoulder, and right shoulder functions (see Fig. 1). Let $\mathbf{A}$, $\mathbf{R}_A$, $\mathbf{R}_D$, $\mathbf{I}$, and $\mathbf{M}$ be pairwise disjoint sets of *atomic concepts*, *abstract roles*, *datatype roles*, *individuals*, and *fuzzy modifiers*, respectively.

A *role* is any element of $\mathbf{R}_A \cup \mathbf{R}_A^- \cup \mathbf{R}_D$ (where $\mathbf{R}_A^-$ is the set of *inverses* R^- of all $R \in \mathbf{R}_A$). We define *concepts* inductively as follows. Each $A \in \mathbf{A}$ is a concept, $\bot$ and $\top$ are concepts, and if $a_1, \ldots, a_n \in \mathbf{I}$, then $\{a_1, \ldots, a_n\}$ is a concept (called *oneOf*). If C, C_1, C_2 are concepts, $R, S \in \mathbf{R}_A \cup \mathbf{R}_A^-$, and $m \in \mathbf{M}$, then $(C_1 \sqcap C_2)$, $(C_1 \sqcup C_2)$, $\neg C$, and $m(C)$ are concepts (called *conjunction*, *disjunction*, *negation*, and *fuzzy modification*, respectively), as well as $\exists R.C$, $\forall R.C$, $\geqslant nS$, and $\leqslant nS$ (called *exists*, *value*, *atleast*, and *atmost restriction*, respectively) for an integer $n \geqslant 0$. If D is a datatype and $T, T_1, \ldots, T_n \in \mathbf{R}_D$, then $\exists T_1, \ldots, T_n.D$, $\forall T_1, \ldots, T_n.D$, $\geqslant nT$, and $\leqslant nT$ are concepts (called *datatype exists*, *value*, *atleast*, and *atmost restriction*, respectively) for an integer $n \geqslant 0$. We eliminate parentheses as usual.

A *crisp axiom* has one of the following forms: (1) $C \sqsubseteq D$ (called *concept inclusion axiom*), where C and D are concepts; (2) $R \sqsubseteq S$ (called *role inclusion axiom*), where either $R, S \in \mathbf{R}_A \cup \mathbf{R}_A^-$ or $R, S \in \mathbf{R}_D$; (3) $\mathrm{Trans}(R)$ (called *transitivity axiom*), where $R \in \mathbf{R}_A$; (4) $C(a)$ (called *concept assertion axiom*), where C is a concept and $a \in \mathbf{I}$; (5) $R(a,b)$ (resp., $U(a,v)$) (called *role assertion 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 $a, b \in \mathbf{I}$. We define *fuzzy axioms* as follows: A *fuzzy concept inclusion* (resp., *fuzzy role inclusion*, *fuzzy concept assertion*, *fuzzy role assertion*) *axiom* is of the form $\alpha \, \theta \, n$, where α is a concept inclusion (resp., role inclusion, concept assertion, role assertion) axiom, $\theta \in \{\leqslant, =, \geqslant\}$, and $n \in [0,1]$. Informally, $\alpha \leqslant n$ (resp., $\alpha = n$, $\alpha \geqslant n$) encodes that the truth value of α is at most (resp., equal to,

at least) n. We often use α to abbreviate $\alpha = 1$. A *fuzzy (description logic) knowledge base* L is a finite set of fuzzy axioms, transitivity axioms, and equality and inequality axioms. For decidability, number restrictions in L are restricted to simple abstract roles. Notice that L may contain fuzzy concept inclusion axioms (between general concepts).

Fuzzy $\mathcal{SHIF}(\mathbf{D})$ has the same syntax as fuzzy $\mathcal{SHOIN}(\mathbf{D})$, but without the oneOf constructor and with the atleast and atmost constructors limited to 0 and 1.

Example 3.2 (Shopping Agent cont'd). The following axioms are an excerpt of the fuzzy description logic knowledge base L that conceptualizes the site in Example 2.1:

$$\mathit{Cars} \sqcup \mathit{Trucks} \sqcup \mathit{Vans} \sqcup \mathit{SUVs} \sqsubseteq \mathit{Vehicles}\,, \tag{1}$$

$$\mathit{PassengerCars} \sqcup \mathit{LuxuryCars} \sqsubseteq \mathit{Cars}\,, \tag{2}$$

$$\mathit{CompactCars} \sqcup \mathit{MidSizeCars} \sqcup \mathit{SportyCars} \sqsubseteq \mathit{PassengerCars}\,, \tag{3}$$

$$\begin{aligned}\mathit{Cars} \sqsubseteq\ & (\exists \mathit{hasReview}.\mathbf{Integer}) \sqcap\ (\exists \mathit{hasInvoice}.\mathbf{Integer}) \sqcap\ (\exists \mathit{hasHP}.\mathbf{Integer}) \\ & \sqcap\ (\exists \mathit{hasResellValue}.\mathbf{Integer}) \sqcap\ (\exists \mathit{hasSafetyFeatures}.\mathbf{Integer}) \sqcap\ \ldots\,,\end{aligned} \tag{4}$$

$$(\mathit{SportyCar} \sqcap (\exists \mathit{hasInvoice}.\{18883\}) \sqcap (\exists \mathit{hasHP}.\{166\}) \sqcap \ldots)(\mathit{MazdaMX5Miata})\,, \tag{5}$$

$$(\mathit{SportyCar} \sqcap (\exists \mathit{hasInvoice}.\{20341\}) \sqcap (\exists \mathit{hasHP}.\{200\}) \sqcap \ldots)(\mathit{VolkswagenGTI})\,, \tag{6}$$

$$(\mathit{SportyCar} \sqcap (\exists \mathit{hasInvoice}.\{24029\}) \sqcap (\exists \mathit{hasHP}.\{162\}) \sqcap \ldots)(\mathit{MitsubishiES})\,. \tag{7}$$

Here, axioms (1)–(3) describe the concept taxonomy of the site, while axiom (4) describes the datatype attributes of the cars sold in the site. For example, every passenger or luxury car is also a car, and every car has a resell value. Axioms (5)–(7) describe the properties of some sold cars. For example, the $\mathit{MazdaMX5Miata}$ is a sports car, costing 18 883 €. Note that $\mathbf{Integer}$ denotes the datatype of all integers.

We may now encode "costs at most about 22 000 €" and "has a power of around 150 HP" in the buyer's request through the following concepts C and D, respectively:

$$C = \exists \mathit{hasInvoice}.\mathit{LeqAbout22000} \ \text{ and } \ D = \exists \mathit{hasHP}.\mathit{Around150HP}\,,$$

where $\mathit{LeqAbout22000} = L(22000, 25000)$ and $\mathit{Around150HP} = \mathit{Tri}(125, 150, 175)$ (see Fig. 1). The latter two equations define the fuzzy concepts "at most about 22 000 €" resp. "around 150 HP". The former is modeled as a left shoulder function stating that if the price is less than 22 000, then the degree of truth (degree of buyer's satisfaction) is 1, else the truth is linearly decreasing to 0 (reached at the cost of 25 000). In fact, we are modeling a case were the buyer would like to pay less than 22 000, though may still accept a higher price (up to 25 000) to a lesser degree. Similarly, the latter models the fuzzy concept "around 150 HP" as a triangular function with vertice in 150 HP.

The following fuzzy axioms are (tight) logical consequences of the above description logic knowledge base L (under the Zadeh semantics of the connectives):

$$\begin{aligned}&C(\mathit{MazdaMX5Miata}) = 1.0\,,\ C(\mathit{VolkswagenGTI}) = 1.0\,,\ C(\mathit{MitsubishiES}) = 0.32\,,\\ &D(\mathit{MazdaMX5Miata}) = 0.36\,,\ D(\mathit{VolkswagenGTI}) = 0.0\,,\ D(\mathit{MitsubishiES}) = 0.56\,.\end{aligned}$$

4 Fuzzy Description Logic Programs

In this section, we define fuzzy dl-programs, which are similar to the fuzzy dl-programs in [10], except that they are based on fuzzy description logics as in [15], and that we

consider only stratified fuzzy dl-programs here. Their canonical model associates with every ground atom a truth value, and so defines a ranking on the Herbrand base. We first introduce the syntax, and we then define the semantics of positive and stratified fuzzy dl-programs in terms of a least model resp. an iterative least model semantics.

Syntax. Informally, a normal fuzzy program is a finite collection of normal fuzzy rules, which are similar to ordinary normal rules, except that (i) they have a lower bound for their truth value, and (ii) they refer to fuzzy interpretations rather than binary interpretations, and thus every of their logical operators (that is, "$\leftarrow$", "$\wedge$", and "not") is associated with a combination strategy (that is, "$\leftarrow$" and "$\wedge$" are associated with a conjunction strategy $\otimes$, while "not" is associated with a negation strategy $\ominus$) to specify how the operator combines truth values. Formally, we assume a function-free first-order vocabulary Φ with finite nonempty sets of constant symbols (which also belong to the set $\mathbf{I}$ of all description logic individuals) and predicate symbols, and a set $\mathcal{X}$ of variables. A *term* is a constant symbol from Φ or a variable from $\mathcal{X}$. If p is a predicate symbol of arity $k \geqslant 0$ from Φ, and $t_1, \ldots, t_k$ are terms, then $p(t_1, \ldots, t_k)$ is an *atom*. A *literal* is an atom a or a default-negated atom $not\ a$. A *normal fuzzy rule* r has the form

$$\begin{aligned} a \leftarrow_{\otimes_0} b_1 \wedge_{\otimes_1} b_2 \wedge_{\otimes_2} \cdots \wedge_{\otimes_{k-1}} b_k \wedge_{\otimes_k} \\ not_{\ominus_{k+1}}\, b_{k+1} \wedge_{\otimes_{k+1}} \cdots \wedge_{\otimes_{m-1}} not_{\ominus_m}\, b_m \geqslant v\,, \end{aligned} \tag{8}$$

where $m \geqslant k \geqslant 0$, $a, b_1, \ldots, b_m$ are atoms, $\otimes_0, \ldots, \otimes_{m-1}$ are conjunction strategies, $\ominus_{k+1}, \ldots, \ominus_m$ are negation strategies, and $v \in [0, 1]$. We call a the *head* of r, denoted $H(r)$, while the conjunction $b_1 \wedge_{\otimes_1} \ldots \wedge_{\otimes_{m-1}} not_{\ominus_m}\, b_m$ is the *body* of r. We define $B(r) = B^+(r) \cup B^-(r)$, where $B^+(r) = \{b_1, \ldots, b_k\}$ and $B^-(r) = \{b_{k+1}, \ldots, b_m\}$. A *normal fuzzy program* P is a finite set of normal fuzzy rules.

Informally, a fuzzy dl-program consists of a fuzzy description logic knowledge base L and a generalized normal fuzzy program P, which may contain queries to L. In such a query, it is asked whether a concept or a role assertion logically follows from L or not (see [3] for more background and examples of such queries). Formally, a *dl-query* $Q(\mathbf{t})$ is either (a) of the form $C(t)$, where C is a concept, and t is a term, or (b) of the form $R(t_1, t_2)$, where R is a role, and t_1 and t_2 are terms. A *dl-atom* has the form $DL[S_1 \uplus p_1, \ldots, S_m \uplus p_m; Q](\mathbf{t})$, where each S_i is an atomic concept or a role, p_i is a unary resp. binary predicate symbol, $Q(\mathbf{t})$ is a dl-query, and $m \geqslant 0$. We call $p_1, \ldots, p_m$ its *input predicate symbols*. Intuitively, $S_i \uplus p_i$ encodes that the truth value of every $S_i(\mathbf{e})$ is at least the truth value of $p_i(\mathbf{e})$, where $\mathbf{e}$ is a constant (resp., pair of constants) from Φ when S_i is a concept (resp., role) (and thus p_i is a unary (resp., binary) predicate symbol). A *fuzzy dl-rule* r is of the form (8), where any b_i in the body of r may be a dl-atom. A *fuzzy dl-program* $KB = (L, P)$ consists of a satisfiable fuzzy description logic knowledge base L and a finite set of fuzzy dl-rules P. *Substitutions*, *ground substitutions*, *ground terms*, *ground atoms*, etc., are defined as usual. We denote by $ground(P)$ the set of all ground instances of fuzzy dl-rules in P relative to Φ.

Example 4.1 (Shopping Agent cont'd). A fuzzy dl-program $KB = (L, P)$ is given by the fuzzy description logic knowledge base L in Example 3.2, and the set of fuzzydl-rules P, which contains only the following fuzzy dl-rule encoding the buyer's request, where $\otimes$ may, e.g., be the Gödel conjunction strategy (that is, $x \otimes y = \min(x, y)$):

$$query(x) \leftarrow_{\otimes} SportyCar(x) \wedge_{\otimes} hasInvoice(x, y_1) \wedge_{\otimes} hasHP(x, y_2) \wedge_{\otimes} DL[LeqAbout22000](y_1) \wedge_{\otimes} DL[Around150HP](y_2) \geqslant 1\,.$$

Models of Fuzzy DL-Programs. We first define fuzzy (Herbrand) interpretations, the semantics of dl-queries, and the truth of fuzzy dl-rules and of fuzzy dl-programs in interpretations. In the sequel, let $KB = (L, P)$ be a (fully general) fuzzy dl-program.

We denote by HB_Φ (resp., HU_Φ) the Herbrand base (resp., universe) over Φ. In the sequel, we assume that HB_Φ is nonempty. A *fuzzy interpretation* I is a mapping $I: HB_\Phi \to [0, 1]$. We denote by $\boldsymbol{HB}_\Phi$ the fuzzy interpretation I such that $I(a) = 1$ for all $a \in HB_\Phi$. For fuzzy interpretations I and J, we write $I \subseteq J$ iff $I(a) \leqslant J(a)$ for all $a \in HB_\Phi$, and we define the *intersection* of I and J, denoted $I \cap J$, by $(I \cap J)(a) = \min(I(a), J(a))$ for all $a \in HB_\Phi$. Note that $I \subseteq \boldsymbol{HB}_\Phi$ for all fuzzy interpretations I. The truth value of $a \in HB_\Phi$ in I under L, denoted $I_L(a)$, is defined as $I(a)$. The truth value of a ground dl-atom $a = DL[S_1 \uplus p_1, \ldots, S_m \uplus p_m; Q](\mathbf{c})$ in I under L, denoted $I_L(a)$, is the supremum of v subject to $L \cup \bigcup_{i=1}^{m} A_i(I) \models Q(\mathbf{c}) \geqslant v$ and $v \in [0, 1]$, where $A_i(I) = \{S_i(\mathbf{e}) \geqslant I(p_i(\mathbf{e})) \mid I(p_i(\mathbf{e})) > 0,\ p_i(\mathbf{e}) \in HB_\Phi\}$. We say I is a *model* of a ground fuzzy dl-rule r of form (8) under L, denoted $I \models_L r$, iff

$$I_L(a) \geqslant I_L(b_1) \otimes_1 I_L(b_2) \otimes_2 \cdots \otimes_{k-1} I_L(b_k) \otimes_k \ominus_{k+1} I_L(b_{k+1}) \otimes_{k+1} \cdots \otimes_{m-1} \ominus_m I_L(b_m) \otimes_0 v\,,$$

Here, we implicitly assume that $\otimes_1, \ldots, \otimes_{m-1}, \otimes_0$ are evaluated from left to right. We say I is a *model* of $KB = (L, P)$, denoted $I \models KB$, iff $I \models_L r$ for all $r \in ground(P)$.

Positive Fuzzy DL-Programs. Informally, positive fuzzy dl-programs have no default negation: A fuzzy dl-program $KB = (L, P)$ is *positive* iff P is "not"-free.

For ordinary positive programs, as well as positive dl-programs KB, the intersection of a set of models of KB is also a model of KB. A similar result holds for positive fuzzy dl-programs KB. Hence, every positive fuzzy dl-program KB has as its *canonical model* a unique least model, denoted M_{KB}, which is contained in every model of KB.

Example 4.2 (Shopping Agent cont'd). The fuzzy dl-program $KB = (L, P)$ of Example 4.1 is positive, and its minimal model M_{KB} is given as follows:

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

and all other ground instances of $query(x)$ have the truth value 0 under M_{KB}.

Stratified Fuzzy DL-Programs. We next define stratified fuzzy dl-programs, which are composed of hierarchic layers of positive fuzzy dl-programs that are linked via default negation. Like for ordinary stratified programs, as well as stratified dl-programs, a minimal model can be defined by a finite number of iterative least models, which naturally describes as the *canonical model* the semantics of stratified fuzzy dl-programs.

For any fuzzy dl-program $KB = (L, P)$, let DL_P denote the set of all ground dl-atoms that occur in $ground(P)$. An *input atom* of $a \in DL_P$ is a ground atom with an input predicate of a and constant symbols in Φ. A *stratification of* $KB = (L, P)$ (*with respect to* DL_P) is a mapping $\lambda: HB_\Phi \cup DL_P \to \{0, 1, \ldots, k\}$ such that

(i) $\lambda(H(r)) \geqslant \lambda(a)$ (resp., $\lambda(H(r)) > \lambda(a)$) for each $r \in \mathit{ground}(P)$ and $a \in B^+(r)$ (resp., $a \in B^-(r)$), and
(ii) $\lambda(a) \geqslant \lambda(a')$ for each input atom a' of each $a \in DL_P$,

where $k \geqslant 0$ is the *length* of λ. For $i \in \{0, \ldots, k\}$, we define $KB_i = (L, P_i) = (L, \{r \in \mathit{ground}(P) \mid \lambda(H(r)) = i\})$, and we define HB_{P_i} (resp., $HB^\star_{P_i}$) as the set of all $a \in HB_\Phi$ such that $\lambda(a) = i$ (resp., $\lambda(a) \leqslant i$).

A fuzzy dl-program $KB = (L, P)$ is *stratified* iff it has a stratification λ of some length $k \geqslant 0$. We define its iterative least models $M_i \subseteq \boldsymbol{HB}_\Phi$ with $i \in \{0, \ldots, k\}$ by:

(i) M_0 is the least model of KB_0;
(ii) if $i > 0$, then M_i is the least model of KB_i such that $M_i | HB^\star_{P_{i-1}} = M_{i-1} | HB^\star_{P_{i-1}}$, where $M_i | HB^\star_{P_{i-1}}$ and $M_{i-1} | HB^\star_{P_{i-1}}$ denote the restrictions of the mappings M_i and M_{i-1} to $HB^\star_{P_{i-1}}$, respectively.

Then, M_{KB} denotes M_k. Note that M_{KB} is well-defined, since it does not depend on a particular stratification λ. Furthermore, M_{KB} is in fact a minimal model of KB.

5 Probabilistic Fuzzy Description Logic Programs

In this section, we introduce probabilistic fuzzy dl-programs as a combination of stratified fuzzy dl-programs with Poole's independent choice logic (ICL) [13]. This will allow us to express probabilistic rules. Poole's ICL is based on ordinary acyclic logic programs P under different "atomic choices", where each atomic choice along with P produces a first-order model, and one then obtains a probability distribution on the set of first-order models by placing a probability distribution on the different atomic choices. Here, we use stratified fuzzy dl-programs rather than ordinary acyclic logic programs, and thus we define a probability distribution on a set of fuzzy interpretations. In other words, we define a probability distribution on a set of rankings on the Herbrand base.

Syntax. We now define the syntax of probabilistic fuzzy dl-programs and probabilistic queries addressed to them. We first introduce fuzzy formulas, query constraints, and probabilistic formulas, and we define choice spaces and probabilities on choice spaces.

We define *fuzzy formulas* by induction as follows. The propositional constants *false* and *true*, denoted $\bot$ and $\top$, respectively, and all atoms $p(t_1, \ldots, t_k)$ are fuzzy formulas. If ϕ and ψ are fuzzy formulas, and $\otimes$, $\oplus$, $\rhd$, and $\ominus$ are conjunction, disjunction, implication, resp. negation strategies, then $(\phi \wedge_\otimes \psi)$, $(\phi \vee_\oplus \psi)$, $(\phi \Rightarrow_\rhd \psi)$, and $\neg_\ominus \phi$ are also fuzzy formulas. A *query constraint* has the form $(\phi\, \theta\, r)[l, u]$ or $(\mathbf{E}[\phi])[l, u]$ with $\theta \in \{\geqslant, >, <, \leqslant\}$, $r, l, u \in [0, 1]$, and fuzzy formulas ϕ. Informally, the former asks for the interval of the probability that the truth value v of ϕ satisfies $v\, \theta\, r$, while the latter asks for the interval of the expected truth value of ϕ. We define *probabilistic formulas* inductively as follows. Each query constraint is a probabilistic formula. If F and G are probabilistic formulas, then also $\neg F$ and $(F \wedge G)$. We use $(F \vee G)$ and $(F \Rightarrow G)$ to abbreviate $\neg(\neg F \wedge \neg G)$ resp. $\neg(F \wedge \neg G)$, and eliminate parentheses as usual.

A *choice space* C is a set of pairwise disjoint and nonempty sets $A \subseteq HB_\Phi$. Any $A \in C$ is an *alternative* of C and any $a \in A$ an *atomic choice* of C. Intuitively, every $A \in C$ represents a random variable and every $a \in A$ one of its possible values. A *total*

choice of C is a set $B \subseteq HB_\Phi$ such that $|B \cap A| = 1$ for all $A \in C$. Intuitively, every total choice B of C represents an assignment of values to all the random variables. A *probability* μ on a choice space C is a probability function on the set of all total choices of C. Intuitively, every μ is a probability distribution over the set of all variable assignments. Since C and all its alternatives are finite, μ can be defined by (i) a mapping $\mu\colon \bigcup C \to [0,1]$ such that $\sum_{a \in A} \mu(a) = 1$ for all $A \in C$, and (ii) $\mu(B) = \Pi_{b \in B} \mu(b)$ for all total choices B of C. Intuitively, (i) defines a probability over the values of each random variable of C, and (ii) assumes independence between the random variables.

A *probabilistic fuzzy dl-program* $KB = (L, P, C, \mu)$ consists of a stratified fuzzy dl-program (L, P), a choice space C such that (i) $\bigcup C \subseteq HB_\Phi$ and (ii) no atomic choice in C coincides with the head of any fuzzy dl-rule in $ground(P)$, and a probability μ on C. Intuitively, since the total choices of C select subsets of P, and μ is a probability distribution on the total choices of C, every probabilistic fuzzy dl-program compactly represents a probability distribution on a finite set of stratified fuzzy dl-programs. A *probabilistic query* to KB has the form $\exists F$, or $\exists(\alpha\, \theta\, r)[L, U]$, or $\exists(\mathbf{E}[\alpha])[L, U]$, where F is a probabilistic formula, α is a fuzzy formula, $r \in [0,1]$, and L, U are variables.

Example 5.1 (Shopping Agent cont'd). A probabilistic fuzzy dl-program $KB = (L, P, C, \mu)$ is given by L of Example 3.2, the following set of fuzzy dl-rules P, which model the query reformulation and retrieval steps using ontology mapping rules:

$$\begin{aligned} query(x) \leftarrow_\otimes\ & SportsCar(x) \wedge_\otimes hasPrice(x, y_1) \wedge_\otimes hasPower(x, y_2) \wedge_\otimes \\ & DL[LeqAbout22000](y_1) \wedge_\otimes DL[Around150HP](y_2) \geqslant 1\,, \qquad (9) \\ SportsCar(x) \leftarrow_\otimes\ & DL[SportyCar](x) \wedge_\otimes sc_{pos} \geqslant 0.9\,, \qquad (10) \\ hasPrice(x) \leftarrow_\otimes\ & DL[hasInvoice](x) \wedge_\otimes hi_{pos} \geqslant 0.8\,, \qquad (11) \\ hasPower(x) \leftarrow_\otimes\ & DL[hasHP](x) \wedge_\otimes hhp_{pos} \geqslant 0.8\,, \qquad (12) \end{aligned}$$

the choice space $C = \{\{sc_{pos}, sc_{neg}\}, \{hi_{pos}, hi_{neg}\}, \{hhp_{pos}, hhp_{neg}\}\}$, and the probability distribution μ, which is given by the following probabilities for the atomic choices sc_{pos}, sc_{neg}, hi_{pos}, hi_{neg}, hhp_{pos}, and hhp_{neg} (which are 0-ary predicate symbols), and then extended to all total choices by assuming independence:

$$\begin{aligned} &\mu(sc_{pos}) = 0.91\,, \quad \mu(sc_{neg}) = 0.09\,, \quad \mu(hi_{pos}) = 0.78\,, \\ &\mu(hi_{neg}) = 0.22\,, \quad \mu(hhp_{pos}) = 0.83\,, \quad \mu(hhp_{neg}) = 0.17\,. \end{aligned}$$

Intuitively, C encodes three probabilistically independent random variables with the binary domains $\{sc_{pos}, sc_{neg}\}$, $\{hi_{pos}, hi_{neg}\}$, and $\{hhp_{pos}, hhp_{neg}\}$. Rule (9) is the buyer's request, but in a "different" terminology than the one of the car selling site. Rules (10)–(12) are so-called ontology alignment mapping rules. For example, rule (10) states that the predicate "SportsCar" of the buyer's terminology refers to the concept "SportyCar" of the selected side, with probability 0.91. Such mapping rules can be automatically built by relying on ontology alignment tools, such as oMap [17,18], whose main purpose is to find relations among the concepts and roles of two different ontologies. oMap is particularly suited for our case, as it is based on a probabilistic model, and thus the mappings have a probabilistic reading (see also [12]).

Semantics. A *world* I is a fuzzy interpretation over HB_Φ. We denote by $\mathcal{I}_\Phi$ the set of all worlds over Φ. A *variable assignment* σ maps each $X \in \mathcal{X}$ to some $t \in HU_\Phi$.

It is extended to all terms by $\sigma(c) = c$ for all constant symbols c from Φ. The *truth value* of fuzzy formulas ϕ in I under σ, denoted $I_\sigma(\phi)$ (or $I(\phi)$ when ϕ is ground), is inductively defined by (1) $I_\sigma(\phi \wedge_\otimes \psi) = I_\sigma(\phi) \otimes I_\sigma(\psi)$, (2) $I_\sigma(\phi \vee_\oplus \psi) = I_\sigma(\phi) \oplus I_\sigma(\psi)$, (3) $I_\sigma(\phi \Rightarrow_\rhd \psi) = I_\sigma(\phi) \rhd I_\sigma(\psi)$, and (4) $I_\sigma(\neg_\ominus \phi) = \ominus I_\sigma(\phi)$.

A *probabilistic interpretation* Pr is a probability function on $\mathcal{I}_\Phi$ (that is, a mapping $Pr\colon \mathcal{I}_\Phi \to [0,1]$ such that (i) the set of all $I \in \mathcal{I}_\Phi$ with $Pr(I) > 0$ is denumerable, and (ii) all $Pr(I)$ with $I \in \mathcal{I}_\Phi$ sum up to 1). The *probability* of $\phi\, \theta\, r$ in Pr under a variable assignment σ, denoted $Pr_\sigma(\phi\, \theta\, r)$ (or $Pr(\phi\, \theta\, r)$ when ϕ is ground), is the sum of all $Pr(I)$ such that $I \in \mathcal{I}_\Phi$ and $I_\sigma(\phi)\, \theta\, r$. The *expected truth value* of ϕ under Pr and σ, denoted $\mathbf{E}_{Pr,\sigma}[\phi]$, is the sum of all $Pr(I) \cdot I_\sigma(\phi)$ with $I \in \mathcal{I}_\Phi$. Notice that in the notion of expected truth value, we combine probabilities and truth values. The *truth* of probabilistic formulas F in Pr under σ, denoted $Pr \models_\sigma F$, is inductively defined by (1) $Pr \models_\sigma (\phi\, \theta\, r)[l,u]$ iff $Pr_\sigma(\phi\, \theta\, r) \in [l,u]$, (2) $Pr \models_\sigma (\mathbf{E}[\phi])[l,u]$ iff $\mathbf{E}_{Pr,\sigma}[\phi] \in [l,u]$, (3) $Pr \models_\sigma \neg F$ iff not $Pr \models_\sigma F$, and (4) $Pr \models_\sigma (F \wedge G)$ iff $Pr \models_\sigma F$ and $Pr \models_\sigma G$.

A probabilistic interpretation Pr is a *model* of a probabilistic formula F iff $Pr \models_\sigma F$ for every variable assignment σ. We say Pr is the *canonical model* of a probabilistic fuzzy dl-program $KB = (L, P, C, \mu)$ iff every world $I \in \mathcal{I}_\Phi$ with $Pr(I) > 0$ is the canonical model of $(L,\, P \cup \{p \leftarrow \mid p \in B\})$ for some total choice B of C with $Pr(I) = \mu(B)$. Notice that every KB has a unique canonical model Pr. We say F is a *consequence* of KB, denoted $KB \mid\!\sim F$, iff the canonical model of KB is also a model of F. A query constraint $(\phi\, \theta\, r)[l,u]$ (resp., $(\mathbf{E}[\phi])[l,u]$) is a *tight consequence* of KB, denoted $KB \mid\!\sim_{tight} (\phi\, \theta r)[l,u]$ (resp., $KB \mid\!\sim_{tight} (\mathbf{E}[\phi])[l,u]$), iff l (resp., u) is the infimum (resp., supremum) of $Pr_\sigma(\phi\, \theta\, r)$ (resp., $\mathbf{E}_{Pr,\sigma}[\phi]$) subject to the canonical model Pr of KB and all σ. A *correct answer* to $\exists F$ is a substitution σ such that $F\sigma$ is a consequence of KB. A *tight answer* to $\exists(\alpha\, \theta\, r)[L,U]$ (resp., $\exists(\mathbf{E}[\alpha])[L,U]$) is a substitution σ such that $(\alpha\, \theta\, r)[L,U]\sigma$ (resp., $(\mathbf{E}[\alpha])[L,U]\sigma$) is a tight consequence of KB.

Example 5.2 (Shopping Agent cont'd). The following are some tight consequences of the probabilistic fuzzy dl-program $KB = (L, P, C, \mu)$ in Example 5.1:

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

So, the agent ranks the $MazdaMX5Miata$ first with degree 0.21 ($= 0.36 \cdot 0.91 \cdot 0.78 \cdot 0.83$) and the $MitsubishiES$ second with degree 0.19 ($= 0.32 \cdot 0.91 \cdot 0.78 \cdot 0.83$).

6 Summary and Outlook

We have presented probabilistic fuzzy dl-programs for the Semantic Web, which allow for handling both probabilistic uncertainty (especially for probabilistic ontology mapping and probabilistic data integration) and fuzzy vagueness (especially for dealing with vague concepts) in a uniform framework. We have defined important concepts related to both probabilistic uncertainty and fuzzy vagueness. Furthermore, we have described a shopping agent example, which gives evidence of the usefulness of probabilistic fuzzy dl-programs in realistic web applications. In the extended report [11], we also provide algorithms for query processing in such programs, which can be done in polynomial time in the data complexity under suitable assumptions.

An interesting topic of future research is to generalize probabilistic fuzzy dl-programs by non-stratified default negations, classical negations, and disjunctions. Another interesting issue is to explore how to update probabilistic fuzzy dl-programs.

Acknowledgments. This work has been partially supported by the German Research Foundation (DFG) under the Heisenberg Programme.

References

1. Berners-Lee, T.: Weaving the Web. Harper, San Francisco (1999)
2. Callan, J.: Distributed information retrieval. In: Croft, W.B. (ed.) Advances in Information Retrieval, pp. 127–150. Kluwer, Dordrecht (2000)
3. Eiter, T., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Combining answer set programming with description logics for the Semantic Web. In: Proc. KR-2004, pp. 141–151 (2004)
4. Fensel, D., Wahlster, W., Lieberman, H., Hendler, J. (eds.): Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential. MIT Press, Cambridge (2002)
5. Flaminio, T., Godo, L.: A logic for reasoning about the probability of fuzzy events. Fuzzy Sets and Systems 158(6), 625–638 (2007)
6. Fuhr, N.: A decision-theoretic approach to database selection in networked IR. ACM Trans. Inf. Syst. 3(17), 229–249 (1999)
7. Hájek, P.: Metamathematics of Fuzzy Logic. Kluwer, Dordrecht (1998)
8. Horrocks, I., Patel-Schneider, P.F.: Reducing OWL entailment to description logic satisfiability. In: Fensel, D., Sycara, K.P., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, pp. 17–29. Springer, Heidelberg (2003)
9. Lukasiewicz, T.: Probabilistic description logic programs. In: Godo, L. (ed.) ECSQARU 2005. LNCS (LNAI), vol. 3571, pp. 737–749. Springer, Heidelberg (2005) (Extended version in Int. J. Approx. Reason. 45(2):288–307, 2007)
10. Lukasiewicz, T.: Fuzzy description logic programs under the answer set semantics for the Semantic Web. In: Proc. RuleML-2006, pp. 89–96 (2006) (Extended version Fundamenta Informaticae (to appear))
11. Lukasiewicz, T., Straccia, U.: Uncertainty and vagueness in description logic programs for the Semantic Web. Report 1843-07-02, Institut für Informationssysteme, TU Wien (2007)
12. Nottelmann, H., Straccia, U.: Information retrieval and machine learning for probabilistic schema matching. Inf. Process. Manage. 43(3), 552–576 (2007)
13. Poole, D.: The independent choice logic for modelling multiple agents under uncertainty. Artif. Intell. 94(1-2), 7–56 (1997)
14. Renda, M.E., Straccia, U.: Web metasearch: Rank vs. score-based rank aggregation methods. In: Proc. SAC-2003, pp. 841–846 (2003)
15. Straccia, U.: A fuzzy description logic for the Semantic Web. In: Sanchez, E. (ed.) Fuzzy Logic and the Semantic Web, Capturing Intelligence, ch. 4, pp. 73–90. Elsevier, Amsterdam (2006)
16. Straccia, U.: Fuzzy description logic programs. In: Proc. IPMU-2006, pp. 1818–1825 (2006)
17. Straccia, U., Troncy, R.: oMAP: Combining classifiers for aligning automatically OWL ontologies. In: Ngu, A.H.H., Kitsuregawa, M., Neuhold, E.J., Chung, J.-Y., Sheng, Q.Z. (eds.) WISE 2005. LNCS, vol. 3806, pp. 133–147. Springer, Heidelberg (2005)
18. Straccia, U., Troncy, R.: Towards distributed information retrieval in the Semantic Web. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 378–392. Springer, Heidelberg (2006)
19. W3C. OWL web ontology language overview 2004, W3C Recommendation (February 10, 2004),
Available at `http://www.w3.org/TR/2004/REC-owl-features-20040210/`