Managing Uncertainty and Vagueness in Description Logics, Logic Programs and Description Logic Programs

Umberto Straccia

ISTI-CNR, Via G. Moruzzi 1, I-56124 Pisa, Italy
straccia@isti.cnr.it

Abstract. Managing uncertainty and/or vagueness is starting to play an important role in Semantic Web representation languages. Our aim is to overview basic concepts on representing uncertain and vague knowledge in current Semantic Web ontology and rule languages (and their combination).

1 Introduction

The management of uncertainty and/or vagueness is an important issue whenever the real world information to be represented is of imperfect nature, which likely occurs in Semantic Web tasks. In this work we overview the relevant work in the context of Description Logics [6], Logic Programs [141] and their combination. This work should act as a reference/citation guide to the relevant literature, and, thus, we keep the formal level to a minimum.

2 Uncertainty and Vagueness Basics

There has been a long-lasting misunderstanding in the literature of artificial intelligence and uncertainty modelling, regarding the role of probability/possibility theory and vague/fuzzy theory. A clarifying paper is [63]. We recall here salient notes, which may clarify the role of these theories for the inexpert reader.

A standard example that points out the difference between degrees of uncertainty and degrees of truth is that of a bottle [63]. In terms of binary truth values, a bottle is viewed as full or empty. But if one accounts for the quantity of liquid in the bottle, one may e.g. say that the bottle is "half-full". Under this way of speaking, "full" becomes a fuzzy predicate [287] and the degree of truth of "the bottle is full" reflects the amount of liquid in the bottle. The situation is quite different when expressing our ignorance about whether the bottle is either full or empty (given that we know that only one of the two situations is the true one). Saying that the probability that the bottle is full is 0.5 does not mean that the bottle is half full.

We recall that under *uncertainty theory* fall all those approaches in which statements rather than being either true or false, are true or false to some *probability* or *possibility* (for example, "it will rain tomorrow"). That is, a statement is true or false in any world, but we are "uncertain" about which world to consider as the right one, and thus we

C. Baroglio et al. (Eds.): Reasoning Web 2008, LNCS 5224, pp. 54–103, 2008.

speak about e.g. a probability distribution or a possibility distribution over the worlds. For example, we cannot exactly establish whether it will rain tomorrow or not, due to our *incomplete* knowledge about our world, but we can estimate to which degree this is probable, possible, and necessary.

As for the main differences between probability and possibility theory, the probability of an event is the sum of the probabilities of all worlds that satisfy this event, whereas the possibility of an event is the maximum of the possibilities of all worlds that satisfy the event. Intuitively, the probability of an event aggregates the probabilities of all worlds that satisfy this event, whereas the possibility of an event is simply the possibility of the "most optimistic" world that satisfies the event. Hence, although both probability and possibility theory allow for quantifying degrees of uncertainty, they are conceptually quite different from each other. That is, probability and possibility theory represent different facets of uncertainty.

On the other hand, under *vagueness/fuzziness theory* fall all those approaches in which statements (for example, "the tomato is ripe") are true to some degree, which is taken from a truth space. That is, an interpretation maps a statement to a truth degree, since we are unable to establish whether a statement is completely true or false due to the involvement of vague concepts, such as "ripe", which only have an *imprecise* definition. For example, we cannot exactly say whether a tomato is ripe or not, but rather can only say that the tomato is ripe to some degree. Usually, such statements involve so-called *vague/fuzzy predicates* [287].

Note that all vague/fuzzy statements are truth-functional, that is, the degree of truth of every statement can be calculated from the degrees of truth of its constituents, while uncertain statements cannot be a function of the uncertainties of their constituents [62]. More concretely, in probability theory, only negation is truth-functional (see Eq. 1), while in possibility theory, only disjunction resp. conjunction is truth-functional in possibilities resp. necessities of events (see Eq. 4). Furthermore, mathematical fuzzy logics are based on truly many-valued logical operators, while uncertainty logics are defined on top of standard binary logical operators.

In the following, we illustrate a typical formalization of uncertain statements and vague statements. In the former case, we consider a basic probabilistic/possibilistic logic, while in the latter, we consider a basic many-valued logic.

2.1 Probabilistic Logic

Probabilistic logic has its origin in philosophy and logic. Its roots can be traced back to Boole in 1854 [17]. There is a wide spectrum of formal languages that have been explored in probabilistic logic, ranging from constraints for unconditional and conditional events to rich languages that specify linear inequalities over events (see especially the work by Nilsson [207], Fagin et al. [74], Dubois and Prade et al. [5,60,64,65], Frisch and Haddawy [81], and the first author [154,157,161]; see also the survey on sentential probability logic by Hailperin [94]). Recently, nonmonotonic generalizations of probabilistic logic have been developed and explored; see especially [165] for an overview. In this section, for illustrative purposes, we recall only the simple probabilistic logic described in [207].

We first define probabilistic formulas and probabilistic knowledge bases. We assume a set of *basic events* $\Phi = \{p_1, \ldots, p_n\}$ with $n \geqslant 1$. We use $\bot$ and $\top$ to denote *false* and *true*, respectively. We define *events* by induction as follows. Every element of $\Phi \cup \{\bot, \top\}$ is an event. If ϕ and ψ are events, then also $\neg\phi$, $(\phi \wedge \psi)$, $(\phi \vee \psi)$, and $(\phi \rightarrow \psi)$ are events. We use $(\phi \leftrightarrow \psi)$ as a shortcut for $(\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$. We adopt the usual conventions to eliminate parentheses. A *probabilistic formula* is an expression of the form $\phi \geqslant l$, where ϕ is an event, and l is a real number from the unit interval $[0,1]$. Informally, $\phi \geqslant l$ says that ϕ is true with a probability of at least l. For example, $\mathit{rain\_tomorrow} \geqslant 0.7$ may express that it will rain tomorrow with a probability of at least 0.7. Notice also that $\neg\phi \geqslant 1 - u$ encodes that ϕ is true with a probability of at most u. Also, we use $\phi = l$ as a shortcut for having both $\phi \geqslant l$ and $\neg\phi \geqslant 1 - l$. A *probabilistic knowledge base* KB is a finite set of probabilistic formulas.

We next define worlds and probabilistic interpretations. A *world* I associates with every basic event in Φ a binary truth value. We extend I by induction to all events as usual. We denote by $\mathcal{I}_\Phi$ the (finite) set of all worlds for Φ. A world I *satisfies* an event ϕ, or I is a *model* of ϕ, denoted $I \models \phi$, iff $I(\phi) = \mathbf{true}$. A *probabilistic interpretation* Pr is a probability function on $\mathcal{I}_\Phi$ (that is, a mapping $\mathit{Pr}\colon \mathcal{I}_\Phi \rightarrow [0,1]$ such that all $\mathit{Pr}(I)$ with $I \in \mathcal{I}_\Phi$ sum up to 1). Intuitively, $\mathit{Pr}(I)$ is the degree to which the world $I \in \mathcal{I}_\Phi$ is probable, that is, the probability function Pr encodes our "uncertainty" about which world is the right one. The *probability* of an event ϕ in Pr, denoted $\mathit{Pr}(\phi)$, is the sum of all $\mathit{Pr}(I)$ such that $I \in \mathcal{I}_\Phi$ and $I \models \phi$. The following equations are an immediate consequence of the above definitions: for all probabilistic interpretations Pr and events ϕ and ψ, the following relationships hold:

$$\begin{array}{ll}
\mathit{Pr}(\phi \wedge \psi) & = \mathit{Pr}(\phi) + \mathit{Pr}(\psi) - \mathit{Pr}(\phi \vee \psi)\,; \\
\mathit{Pr}(\phi \wedge \psi) & \leqslant \min(\mathit{Pr}(\phi), \mathit{Pr}(\psi))\,; \\
\mathit{Pr}(\phi \wedge \psi) & \geqslant \max(0, \mathit{Pr}(\phi) + \mathit{Pr}(\psi) - 1)\,; \\
\mathit{Pr}(\phi \vee \psi) & = \mathit{Pr}(\phi) + \mathit{Pr}(\psi) - \mathit{Pr}(\phi \wedge \psi)\,; \\
\mathit{Pr}(\phi \vee \psi) & \leqslant \min(1, \mathit{Pr}(\phi) + \mathit{Pr}(\psi))\,; \\
\mathit{Pr}(\phi \vee \psi) & \geqslant \max(\mathit{Pr}(\phi), \mathit{Pr}(\psi))\,; \\
\mathit{Pr}(\neg\phi) & = 1 - \mathit{Pr}(\phi)\,; \\
\mathit{Pr}(\bot) & = 0\,; \\
\mathit{Pr}(\top) & = 1\,.
\end{array} \tag{1}$$

A probabilistic interpretation Pr *satisfies* a probabilistic formula $\phi \geqslant l$, or Pr is a *model* of $\phi \geqslant l$, denoted $\mathit{Pr} \models \phi \geqslant l$, iff $\mathit{Pr}(\phi) \geqslant l$. We say Pr *satisfies* a probabilistic knowledge base KB, or Pr is a model of KB, iff Pr satisfies all $F \in \mathit{KB}$. We say KB is *satisfiable* iff a model of KB exists. A probabilistic formula F is a *logical consequence* of KB, denoted $\mathit{KB} \models F$, iff every model of KB satisfies F. We say $\phi \geqslant l$ is a *tight logical consequence* of KB iff l is the infimum of $\mathit{Pr}(\phi)$ subject to all models Pr of KB. Notice that the latter is equivalent to $l = \sup\,\{r \mid \mathit{KB} \models \phi \geqslant r\}$.

Note that often also *conditional events* of the form $\phi \mid \psi$ are allowed, which may then be used in *conditional probabilistic formulae* of the form $\phi \mid \psi \geqslant l$, where ϕ and ψ are events. These statements intuitively encode that the conditional probability of ϕ

given ψ is equal or greater than l. For instance, $flies \mid bird \geqslant 0.8$ dictates that at least 80% of birds fly. From a semantics point of view, we define

$$Pr(\phi \mid \psi) = \begin{cases} \frac{Pr(\phi \wedge \psi)}{Pr(\psi)} & \text{if} Pr(\psi) \neq 0 \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

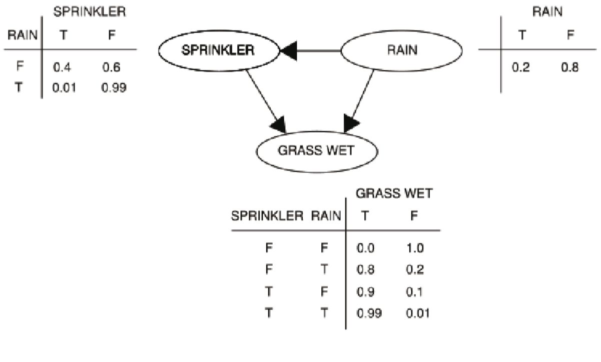
and, thus, $Pr \models \phi \mid \psi \geqslant l$, iff $Pr(\phi \mid \psi) \geqslant l$.

The main decision and optimization problems in probabilistic logic are deciding the satisfiability of probabilistic knowledge bases and logical consequences from probabilistic knowledge bases, as well as computing tight logical consequences from probabilistic knowledge bases, which can be done by deciding the solvability of a system of linear inequalities and by solving a linear optimization problem, respectively. In particular, column generation techniques from operations research have been successfully used to solve large problem instances in probabilistic logic; see especially the work by Jaumard et al. [114] and Hansen et al. [98].

Bayesian Network. We recall here also some basics of *Bayesian Networks* (BN), as they play an important role in many probabilistic logic formalisms in the sense that BNs can be expressed in these logics (see, e.g. [29, 125, 215, 285]).

A Bayesian network is a directed acyclic graph whose nodes represent random variables, and whose arcs encode conditional independencies between the variables. If there is an arc from node b to another node a, b is called a parent of a, and a is a child of b. The set of parent nodes of a node a_i is denoted by $parents(a_i)$. If nodes $b_1, \ldots, b_n$ are parents of a node a, then we have an associated conditional probability table $Pr(a \mid b_1, \ldots, b_n)$. If node a_i has no parents, its local probability distribution is said to be unconditional, otherwise it is conditional. If the value of a node is observed, then the node is said to be an evidence node. It is required that the joint distribution of the node values can be written as the product of the local distributions of each node and its parents: that is,

$$Pr(a_1, \ldots, a_n) = \Pi_{i=1}^{n} Pr(a_i \mid parents(a_i)) .$$



	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99

RAIN	
T	F
0.2	0.8

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

Fig. 1. A Bayesian network

We may also encode a BN in a probabilistic propositional logic using conditional events, as shown in the following example.

Example 2.1. Suppose that there are two events, which could cause grass to be wet: either the sprinkler is on or it's raining. Also, suppose that the rain has a direct effect on the use of the sprinkler (namely that when it rains, the sprinkler is usually not turned on). Then the situation can be modelled with a Bayesian network, as shown in Fig. 1. All three variables (Rain, $\mathit{Sprinkler}$ and $\mathit{GrassWet}$) have two possible values T (for true) and F (for false). Rain has an unconditional probability distribution: $\mathit{Pr}(\mathit{Rain} = T) = 0.2$, while $\mathit{Pr}(\mathit{Rain} = F) = 0.8$. The conditional probability table associated to the node $\mathit{Sprinkler}$ provides the conditional probabilities $\mathit{Pr}(\mathit{Sprinkler} = X \mid \mathit{Rain} = Y)$ for any $X, Y \in \{T, F\}$, while the conditional probability table associated to the node $\mathit{GrassWet}$ provides the conditional probabilities $\mathit{Pr}(\mathit{GrassWet} = X \mid \mathit{Sprinkler} = Y_1, \mathit{Rain} = Y_2)$ for any $X, Y_i \in \{T, F\}$. The joint probability function is:

$$\begin{aligned}\mathit{Pr}(\mathit{GrassWet}, \mathit{Sprinkler}, \mathit{Rain}) = {} & \mathit{Pr}(\mathit{GrassWet} \mid \mathit{Sprinkler}, \mathit{Rain}) \\ & \cdot \mathit{Pr}(\mathit{Sprinkler} \mid \mathit{Rain}) \cdot \mathit{Pr}(\mathit{Rain})\ .\end{aligned} \tag{3}$$

The model can answer questions like "What is the probability that it is raining, given the grass is wet?" using Eq. 3:

$$\begin{aligned}\mathit{Pr}(\mathit{Rain} = T \mid \mathit{GrassWet} = T) &= \frac{\mathit{Pr}(\mathit{Rain} = T, \mathit{GrassWet} = T)}{\mathit{Pr}(\mathit{GrassWet} = T)} \\ &= \frac{\sum_{Y \in \{T,F\}} \mathit{Pr}(\mathit{Rain} = T, \mathit{GrassWet} = T, \mathit{Sprinkler} = Y)}{\sum_{Y_1, Y_2 \in \{T,F\}} \mathit{Pr}(\mathit{GrassWet} = T, (\mathit{Rain} = Y_1, \mathit{Sprinkler} = Y_2))} \\ &= \frac{0.99 \cdot 0.01 \cdot 0.2 + 0.8 \cdot 0.99 \cdot 0.2}{0.99 \cdot 0.01 \cdot 0.2 + 0.9 \cdot 0.4 \cdot 0.8 + 0.8 \cdot 0.99 \cdot 0.2 + 0 \cdot 0.6 \cdot 0.8} \\ &\approx 0.3577\ .\end{aligned}$$

We may encode the BN in a probabilistic propositional logic using conditional events. Indeed, for every node a, we use a propositional letters $a(T), a(F)$, where the former encodes the event "a is true" and the latter encodes the event "a is false". Of course, we have to consider also $(a(T) \leftrightarrow \neg a(F)) = 1$. If a node a has no parents then we can easily encode its associated probability table with the formula $a(T) = p$. Hence, we have the formula $\mathit{Rain}(T) = 0.2$. If a node has parents, we encode its associated conditional probability table using conditional probability formulae. In particular, we will have the conditional probabilistic formulae

$$\begin{aligned}(\mathit{Sprinkler}(T) \mid \mathit{Rain}(F)) &= 0.4 \\ (\mathit{Sprinkler}(T) \mid \mathit{Rain}(T)) &= 0.01\end{aligned}$$

$$\begin{aligned}(\mathit{GrassWet}(T) \mid \mathit{Sprinkler}(F) \wedge \mathit{Rain}(F)) &= 0.0 \\ (\mathit{GrassWet}(T) \mid \mathit{Sprinkler}(F) \wedge \mathit{Rain}(T)) &= 0.8 \\ (\mathit{GrassWet}(T) \mid \mathit{Sprinkler}(T) \wedge \mathit{Rain}(F)) &= 0.9 \\ (\mathit{GrassWet}(T) \mid \mathit{Sprinkler}(T) \wedge \mathit{Rain}(T)) &= 0.0\ .\end{aligned}$$

□

2.2 Possibilistic Logic

We next recall possibilistic logic; see especially [59]. The main syntactic and semantic differences to probabilistic logic can be summarized as follows. Syntactically, rather than using probabilistic formulas to constrain the probabilities of propositional events, we now use possibilistic formulas to constrain the necessities and possibilities of propositional events. Semantically, rather than having probability distributions on worlds, each of which associates with every event a unique probability, we now have possibility distributions on worlds, each of which associates with every event a unique possibility and a unique necessity. Differently from the probability of an event, which is the sum of the probabilities of all worlds that satisfy that event, the possibility of an event is the maximum of the possibilities of all worlds that satisfy the event. As a consequence, probabilities and possibilities of events behave quite differently from each other (see Eqs. 1 and 4). These fundamental semantic differences between probabilities and possibilities can also be used as the main criteria for using either probabilistic logic or possibilistic logic in a given application involving uncertainty. In addition, possibilistic logic may especially be used for encoding user preferences, since possibility measures can actually be viewed as rankings (on worlds or also objects) along an ordinal scale.

The semantic differences between probabilities and possibilities are also reflected in the computational properties of possibilistic and probabilistic logic, since reasoning in probabilistic logic generally requires to solve linear optimization problems, while reasoning in possibilistic logic does not, and thus can generally be done with less computational effort. Note that although possibility measures can be viewed as sets of upper probability measures [61], and possibility and probability measures can be translated into each other [56], no translations are known between possibilistic and probabilistic knowledge bases as described here.

We first define possibilistic formulas and knowledge bases. *Possibilistic formulas* have the form $\mathsf{P}\,\phi \geqslant l$ or $\mathsf{N}\,\phi \geqslant l$, where ϕ is an event, and l is a real number from $[0,1]$. Informally, such formulas encode to what extent ϕ is *possibly* resp. *necessarily* true. For example, $\mathsf{P}\ \mathit{rain\_tomorrow} \geqslant 0.7$ encodes that it will rain tomorrow is possible to degree 0.7, while $\mathsf{N}\,\mathit{father} \rightarrow \mathit{man} \geqslant 1$ says that a father is necessarily a man. A *possibilistic knowledge base* KB is a finite set of possibilistic formulas.

A *possibilistic interpretation* is a mapping $\pi\colon \mathcal{I}_\Phi \rightarrow [0,1]$. Intuitively, $\pi(I)$ is the degree to which the world I is *possible*. In particular, every world I such that $\pi(I) = 0$ is *impossible*, while every world I such that $\pi(I) = 1$ is *totally possible*. We say π is *normalized* iff $\pi(I) = 1$ for some $I \in \mathcal{I}_\Phi$. Intuitively, this guarantees that there exists at least one world, which could be considered as the real one.

The *possibility* of an event ϕ in a possibilistic interpretation π, denoted $Poss(\phi)$, is then defined by $Poss(\phi) = \max\{\pi(I) \mid I \in \mathcal{I}_\Phi, I \models \phi\}$ (where $\max \emptyset = 0$). Intuitively, the possibility of ϕ is evaluated in the most possible world where ϕ is true. The dual notion to the possibility of an event ϕ is the *necessity* of ϕ, denoted $Nec(\phi)$, which is defined by $Nec(\phi) = 1 - Poss(\neg\phi)$. It reflects the lack of possibility of $\neg\phi$, that is, $Nec(\phi)$ evaluates to what extent ϕ is certainly true. The following properties follows immediately from the above definitions.

For all possibilistic interpretations π and events ϕ and ψ, the following relationships hold:

$$\begin{array}{ll}
Poss(\phi \wedge \psi) & \leqslant \min(Poss(\phi), Poss(\psi))\,; \\
Poss(\phi \vee \psi) & = \max(Poss(\phi), Poss(\psi))\,; \\
Poss(\neg\phi) & = 1 - Nec(\phi)\,; \\
Poss(\bot) & = 0\,; \\
Poss(\top) & = 1 \quad \text{(in the normalized case);} \\
 & \\
Nec(\phi \wedge \psi) & = \min(Nec(\phi), Nec(\psi))\,; \\
Nec(\phi \vee \psi) & \geqslant \max(Nec(\phi), Nec(\psi))\,; \\
Nec(\neg\phi) & = 1 - Poss(\phi)\,; \\
Nec(\bot) & = 0 \quad \text{(in the normalized case);} \\
Nec(\top) & = 1\,.
\end{array} \tag{4}$$

A possibilistic interpretation π *satisfies* a possibilistic formula $\mathsf{P}\,\phi \geqslant l$ (resp., $\mathsf{N}\,\phi \geqslant l$), or π is a *model* of $\mathsf{P}\,\phi \geqslant l$ (resp., $\mathsf{N}\,\phi \geqslant l$), denoted $\pi \models \mathsf{P}\,\phi \geqslant l$ (resp., $\pi \models \mathsf{N}\,\phi \geqslant l$), iff $Poss(\phi) \geqslant l$ (resp., $Nec(\phi) \geqslant l$). The notions of satisfiability, logical consequence, and tight logical consequence for possibilistic knowledge bases are then defined as usual (in the same way as in the probabilistic case). We refer the reader to [59,107] for algorithms for possibilistic logic.

2.3 Many-Valued Logics

In the setting of many-valued logics, the convention prescribing that a proposition is either true or false is changed. A more refined range is used for the function that represents the meaning of a proposition. This is usual in natural language when words are modelled by fuzzy sets. For example, the compatibility of "tall" in the phrase "a tall man" with some individual of a given height is often graded: The man can be judged not quite tall, somewhat tall, rather tall, very tall, etc. Changing the usual true/false convention leads to a new concept of proposition, whose compatibility with a given state of facts is a matter of degree and can be measured on an ordered scale $\mathcal{S}$ that is no longer $\{0,1\}$, but e.g. the unit interval $[0,1]$. This leads to identifying a "fuzzy proposition" ϕ with a fuzzy set of possible states of affairs; the degree of membership of a state of affairs to this fuzzy set evaluates the degree of fit between the proposition and the state of facts it refers to. This degree of fit is called *degree of truth* of the proposition ϕ in the interpretation $\mathcal{I}$ (state of affairs). Many-valued logics provide compositional calculi of degrees of truth, including degrees between "true" and "false". A sentence is now not true or false only, but may have a truth degree taken from a *truth space* $\mathcal{S}$, usually $[0,1]$ (in that case we speak bout *Mathematical Fuzzy Logic* [95]) or $\{\frac{0}{n}, \frac{1}{n}, \ldots, \frac{n}{n}\}$ for an integer $n \geqslant 1$. Often $\mathcal{S}$ may be also a complete lattice or a bilattice [85,79] (often used in logic programming [80]). In the sequel, we assume $\mathcal{S} = [0,1]$.

In the many-valued logic that we consider here, *many-valued formulas* have the form $\phi \geqslant l$ or $\phi \leqslant u$, where $l, u \in [0,1]$ [93,95], which encode that the degree of truth of ϕ is *at least* l resp. *at most* u. For example, $\mathit{ripe\_tomato} \geqslant 0.9$ says that we have a rather ripe tomato (the degree of truth of $\mathit{ripe\_tomato}$ is at least 0.9).

Semantically, a *many-valued interpretation* $\mathcal{I}$ maps each basic proposition p_i into $[0,1]$ and is then extended inductively to all propositions as follows:

$$\begin{aligned} \mathcal{I}(\phi \wedge \psi) &= \mathcal{I}(\phi) \otimes \mathcal{I}(\psi)\,; \\ \mathcal{I}(\phi \vee \psi) &= \mathcal{I}(\phi) \oplus \mathcal{I}(\psi)\,; \\ \mathcal{I}(\phi \rightarrow \psi) &= \mathcal{I}(\phi) \Rightarrow \mathcal{I}(\psi)\,; \\ \mathcal{I}(\neg\phi) &= \ominus\mathcal{I}(\phi)\,, \end{aligned} \tag{5}$$

where $\otimes$, $\oplus$, $\Rightarrow$, and $\ominus$ are so-called *combination functions*, namely, *triangular norms* (or *t-norms*), *triangular co-norms* (or *s-norms*), *implication functions*, and *negation functions*, respectively, which extend the classical Boolean conjunction, disjunction, implication, and negation, respectively, to the many-valued case.

Several t-norms, s-norms, implication functions, and negation functions have been given in the literature. An important aspect of such functions is that they satisfy some properties that one expects to hold for the connectives; see Tables 1 and 2. Note that in Table 1, the two properties Tautology and Contradiction follow from Identity, Commutativity, and Monotonicity. Usually, the implication function $\Rightarrow$ is defined as *r-implication*, that is, $a \Rightarrow b = \sup\{c \mid a \otimes c \leqslant b\}$.

Some t-norms, s-norms, implication functions, and negation functions of various fuzzy logics are shown in Table 3 [95]. In fuzzy logic, one usually distinguishes three different logics, namely, Łukasiewicz, Gödel, and Product logic; the popular Zadeh logic is a sublogic of Łukasiewicz logic. Some salient properties of these logics are shown in Table 4. For more properties, see especially [95, 209]. Note also, that a many-valued logic having all properties shown in Table 4, collapses to boolean logic, that is the truth-set can be $\{0,1\}$ only.

The implication $x \Rightarrow y = \max(1-x, y)$ is called *Kleene-Dienes implication* in the fuzzy logic literature. Note that we have the following inferences: Let $a \geqslant n$ and $a \Rightarrow b \geqslant m$. Then, under Kleene-Dienes implication, we infer that if $n > 1-m$ then $b \geqslant m$. Under r-implication relative to a t-norm $\otimes$, we infer that $b \geqslant n \otimes m$.

Note that implication functions and t-norms are also used to define the degree of subsumption between fuzzy sets and the composition of two (binary) fuzzy relations. A *fuzzy set* R over a countable crisp set X is a function $R\colon X \rightarrow [0,1]$. The *degree of subsumption* between two fuzzy sets A and B, denoted $A \sqsubseteq B$, is defined as $\inf_{x \in X} A(x) \Rightarrow B(x)$, where $\Rightarrow$ is an implication function. Note that if $A(x) \leqslant B(x)$, for all $x \in [0,1]$, then $A \sqsubseteq B$ evaluates to 1. Of course, $A \sqsubseteq B$ may evaluate to a value $v \in (0,1)$ as well. A (binary) *fuzzy relation* R over two countable crisp sets X and Y is a function $R\colon X \times Y \rightarrow [0,1]$. The *inverse* of R is the function $R^{-1}\colon Y \times X \rightarrow [0,1]$ with membership function $R^{-1}(y,x) = R(x,y)$, for every $x \in X$ and $y \in Y$. The *composition* of two fuzzy relations $R_1\colon X \times Y \rightarrow [0,1]$ and $R_2\colon Y \times Z \rightarrow [0,1]$ is defined as $(R_1 \circ R_2)(x,z) = \sup_{y \in Y} R_1(x,y) \otimes R_2(y,z)$. A fuzzy relation R is *transitive* iff $R(x,z) \geqslant (R \circ R)(x,z)$.

A many-valued interpretation $\mathcal{I}$ *satisfies* a many-valued formula $\phi \geqslant l$ (resp., $\phi \leqslant u$) or $\mathcal{I}$ is a *model* of $\phi \geqslant l$ (resp., $\phi \leqslant u$), denoted $\mathcal{I} \models \phi \geqslant l$ (resp., $\mathcal{I} \models \phi \leqslant u$), iff $\mathcal{I}(\phi) \geqslant l$ (resp., $\mathcal{I}(\phi) \leqslant u$). The notions of satisfiability, logical consequence, and tight logical consequence for many-valued knowledge bases are then defined in the standard way (in the same way as in the probabilistic case). We refer the reader to [92, 93, 95] for algorithms for many-valued logics.

Table 1. Properties for t-norms and s-norms

Axiom Name	T-norm	S-norm
Tautology / Contradiction	$a \otimes 0 = 0$	$a \oplus 1 = 1$
Identity	$a \otimes 1 = a$	$a \oplus 0 = a$
Commutativity	$a \otimes b = b \otimes a$	$a \oplus b = b \oplus a$
Associativity	$(a \otimes b) \otimes c = a \otimes (b \otimes c)$	$(a \oplus b) \oplus c = a \oplus (b \oplus c)$
Monotonicity	if $b \leqslant c$, then $a \otimes b \leqslant a \otimes c$	if $b \leqslant c$, then $a \oplus b \leqslant a \oplus c$

Table 2. Properties for implication and negation functions

Axiom Name	Implication Function	Negation Function
Tautology / Contradiction	$0 \Rightarrow b = 1,\ a \Rightarrow 1 = 1,\ 1 \Rightarrow 0 = 0$	$\ominus 0 = 1,\ \ominus 1 = 0$
Antitonicity	if $a \leqslant b$, then $a \Rightarrow c \geqslant b \Rightarrow c$	if $a \leqslant b$, then $\ominus a \geqslant \ominus b$
Monotonicity	if $b \leqslant c$, then $a \Rightarrow b \leqslant a \Rightarrow c$	

Table 3. Combination functions of various fuzzy logics

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

Table 4. Some additional properties of combination functions of various fuzzy logics

Property	Łukasiewicz Logic	Gödel Logic	Product Logic	Zadeh Logic
$x \otimes \ominus x = 0$	+	+	+	−
$x \oplus \ominus x = 1$	+	−	−	−
$x \otimes x = x$	−	+	−	+
$x \oplus x = x$	−	+	−	+
$\ominus \ominus x = x$	+	−	−	+
$x \Rightarrow y = \ominus x \oplus y$	+	−	−	+
$\ominus (x \Rightarrow y) = x \otimes \ominus y$	+	−	−	+
$\ominus (x \otimes y) = \ominus x \oplus \ominus y$	+	+	+	+
$\ominus (x \oplus y) = \ominus x \otimes \ominus y$	+	+	+	+

3 Managing Imperfect Knowledge in Semantic Web Languages

3.1 The Case of Description Logics

Probabilistic Uncertainty and Description Logics. Although there are several previous approaches to probabilistic description logics without semantic web background, P-$\mathcal{SHOIN}(\mathbf{D})$ [86, 167, 171] (see also [175]) is the most expressive probabilistic description logic, both in terms of the generalized classical description logic and in terms of the supported forms of terminological and assertional probabilistic knowledge. The syntax of the probabilistic description logic P-$\mathcal{SHOIN}(\mathbf{D})$ uses the notion of a conditional constraint from [157] 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 [159, 163], which is a probabilistic generalization of the sophisticated notion of lexicographic entailment by Lehmann [132] 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 [132].

Roughly, 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 partition the set of individuals $\mathbf{I}$ 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. That is, probabilistic individuals are those individuals in $\mathbf{I}$ for which we explicitly store some probabilistic assertional knowledge in a PABox.

We first define 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 $\mathcal{SHOIN}(\mathbf{D})$ 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.

A PTBox, a PABox, and a probabilistic knowledge bases are defined as follows: (i) A *PTBox* $PT = (T, P)$ consists of a classical (description logic) knowledge base T and a finite set of conditional constraints P; (ii) A *PABox* P is a finite set of conditional constraints; and (iii) a *probabilistic 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 $o \in \mathbf{I}_P$. The meaning of a conditional constraint $(\psi|\phi)[l, u]$ depends on whether it belongs to P or to P_o for some probabilistic individual $o \in \mathbf{I}_P$:

- Each $(\psi|\phi)[l, u]$ in P informally encodes that "generally, if an object belongs to ϕ, then it belongs to ψ with a probability in $[l, u]$". For example, $(\exists R.\{o\}|\phi)[l, u]$ in P, where $o \in \mathbf{I}_C$ and $R \in \mathbf{R}_A$, encodes that "generally, if an object belongs to ϕ, then it is related to o by R with a probability in $[l, u]$".

- Each $(\psi|\phi)[l,u]$ in P_o, where $o \in \mathbf{I}_P$, informally encodes that "if o belongs to ϕ, then o belongs to ψ with a probability in $[l,u]$". For example, $(\exists R.\{o'\}|\phi)[l,u]$ in P_o, where $o \in \mathbf{I}_P$, $o' \in \mathbf{I}_C$, and $R \in \mathbf{R}_A$, expresses that "if o belongs to ϕ, then o is related to o' by R with a probability in $[l,u]$".

So, a probabilistic knowledge base $KB = (T, P, (P_o)_{o \in \mathbf{I}_P})$ extends a classical knowledge base T by probabilistic terminological knowledge P and probabilistic assertional knowledge P_o about every $o \in \mathbf{I}_P$. That is, P represents our statistical knowledge about concepts, while every P_o represents our degrees of belief about o.

Observe that the axioms in T and the conditional constraints in every P_o with $o \in \mathbf{I}_P$ are strict (that is, they must always hold), while the conditional constraints in P are defeasible (that is, they may have exceptions and thus do not always have to hold), since $T \cup P$ may not always be satisfiable as a whole in combination with our degrees of belief (and then we ignore some elements of P).

Consequently, a conditional constraint $(\psi|\phi)[1,1]$ in P encodes "generally, if an object belongs to ϕ, then it also belongs to ψ", while $(\psi|\phi)[1,1]$ in P_o encodes "if o belongs to ϕ, then o also belongs to ψ". The latter is equivalent to the implication $o : \phi \Rightarrow o : \psi$, while the former is in general not equivalent to $\phi \sqsubseteq \psi$.

Semantics. Now we define the semantics of 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 [159, 163].

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

A *(possible) object* o is a set of basic c-concepts $\phi \in \mathcal{C}$ such that $\{i : \phi \mid \phi \in o\} \cup \{i : \neg\phi \mid \phi \in \mathcal{C} \setminus o\}$ is satisfiable, where i is a new individual. Informally, every object o represents an individual i that is fully specified on $\mathcal{C}$ in the sense that o belongs (resp., does not belong) to every c-concept $\phi \in o$ (resp., $\phi \in \mathcal{C} \setminus o$). We denote by $\mathcal{O}_{\mathcal{C}}$ the set of all objects relative to $\mathcal{C}$. An object o *satisfies* a classical knowledge base T, or o is a *model* of T, denoted $o \models T$, iff $T \cup \{i : \phi \mid \phi \in o\} \cup \{i : \neg\phi \mid \phi \in \mathcal{C} \setminus o\}$ is satisfiable, where i is a new individual. An object o *satisfies* a basic c-concept $\phi \in \mathcal{C}$, or o is a *model* of ϕ, denoted $o \models \phi$, iff $\phi \in o$. The satisfaction of c-concepts by objects is inductively extended to all c-concepts, as usual, by (i) $o \models \neg\phi$ iff $o \models \phi$ does not hold, and (ii) $o \models \phi \sqcap \psi$ iff $o \models \phi$ and $o \models \psi$. It is not difficult to verify that a classical knowledge base T is satisfiable iff an object $o \in \mathcal{O}_{\mathcal{C}}$ exists that satisfies T.

A *probabilistic interpretation* Pr is a probability function on $\mathcal{O}_{\mathcal{C}}$ (that is, a mapping $Pr \colon \mathcal{O}_{\mathcal{C}} \to [0,1]$ such that all $Pr(o)$ with $o \in \mathcal{O}_{\mathcal{C}}$ 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 $o \models T$ for every $o \in \mathcal{O}_{\mathcal{C}}$ such that $Pr(o) > 0$. 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 $Pr(o)$ such that $o \models \phi$. 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 $F \in \mathcal{F}$. It is not difficult to verify that a classical knowledge base T is satisfiable iff there exists a probabilistic interpretation that satisfies T.

The notion of consistency for PTBoxes and probabilistic knowledge bases is based on the notion of consistency in probabilistic default reasoning [159, 163]. We first give some preparative definitions. A probabilistic interpretation Pr *verifies* a conditional constraint $(\psi|\phi)[l,u]$ iff $Pr(\phi) = 1$ and $Pr(\psi) \in [l,u]$, that is, iff $Pr(\phi) = 1$ and $Pr \models (\psi|\phi)[l,u]$. We say 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 F under a classical knowledge base T iff $T \cup \mathcal{F}$ has a model that verifies F.

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, condition (ii) means that P has a natural ordered partition into collections of conditional constraints of increasing specificities such that every collection is locally consistent. That is, any inconsistencies can be naturally resolved by preferring more specific pieces of knowledge to less specific ones. For example, the inconsistency between $(\neg\exists\, \mathit{HasColor}.\{\mathit{red}\} \mid \mathit{Car})[1,1]$ and $(\exists\, \mathit{HasColor}.\{\mathit{red}\} \mid \mathit{SportsCar})[1,1]$ when reasoning about sports cars is naturally resolved by preferring the latter to the former. We call the above (unique) 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 (i) $PT = (T,P)$ is consistent and (ii) $T \cup P_o$ is satisfiable for every probabilistic individual $o \in \mathbf{I}_P$. Informally, (ii) says that the strict knowledge in T must be compatible with the strict degrees of belief in P_o, for every probabilistic individual o. Observe that (i) involves T and P, while (ii) involves T and P_o, for every probabilistic individual o. This separate treatment of P and the P_o's is due to the fact that P represents probabilistic terminological knowledge, while each P_o represents probabilistic assertional knowledge (about o).

The notion of lexicographic entailment for probabilistic knowledge bases is based on lexicographic entailment in probabilistic default reasoning [159, 163]. 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 statistical knowledge and degrees of belief about probabilistic objects from PTBoxes and probabilistic knowledge bases, respectively.

We use the (unique) z-partition $(P_0,\ldots,P_k)$ of (T,P) to define a lexicographic preference relation on 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 $i < j \leqslant k$. Roughly speaking, this preference relation implements the idea of preferring more specific pieces of knowledge to less specific ones in the case of local inconsistencies. It can thus be used for ignoring the latter when drawing conclusions in the case of local 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.

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 $T \cup \mathcal{F} \cup \{(\phi|\top)[1,1]\}$. Note that $[l,u] = [1,0]$ (where $[1,0]$ represents the empty interval) when no such model Pr exists. 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 statistical knowledge and degrees of belief follow under lexicographic entailment from PTBoxes PT and probabilistic knowledge bases $KB = (T, P, (P_o)_{o \in \mathbf{I}_P})$, respectively. A conditional constraint F is a *lex-consequence* of 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 KB, denoted $KB \,\|\!\sim^{lex} F$, iff $P_o \,\|\!\sim^{lex} F$ under $PT = (T,P)$. We say F is a *tight lex-consequence* of KB, denoted $KB \,\|\!\sim^{lex}_{tight} F$, iff $P_o \,\|\!\sim^{lex}_{tight} F$ under $PT = (T,P)$.

The main reasoning problems in P-$\mathcal{SHOIN}(\mathbf{D})$ are summarized by the following decision and computation problems (where 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): Given a PTBox $PT = (T,P)$, decide whether PT 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 $\mathcal{F} \,\|\!\sim^{lex}_{tight} (\psi|\phi)[l,u]$ 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 the rational numbers $l, u \in [0,1]$ such that $PT \,\|\!\sim^{lex}_{tight} (\psi|\phi)[l,u]$; (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 the rational numbers $l, u \in [0,1]$ such that $PT \,\|\!\sim^{lex}_{tight} (\exists R.\{o\}|\phi)[l,u]$; (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 $KB \,\|\!\sim^{lex}_{tight} (\psi|\top)[l,u]$ 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 $KB \,\|\!\sim^{lex}_{tight} (\exists R.\{o'\}|\top)[l,u]$ for o.

Another important decision problem in P-$\mathcal{SHOIN}(\mathbf{D})$ is PROBABILISTIC CONCEPT SATISFIABILITY (PCSAT): Given a consistent PTBox PT and a c-concept ϕ, decide whether $PT \not\Vdash^{lex} (\phi|\top)[0,0]$. This problem is reducible to CSAT (classical, non-probabilistic concept satisfiability), since $(T, P) \not\Vdash^{lex} (\phi|\top)[0,0]$ iff $T \not\models \phi \sqsubseteq \bot$.

There exists an algorithm for deciding whether a PTBox (resp., probabilistic knowledge base) in P-$\mathcal{SHOIN}(\mathbf{D})$ is consistent, which is based on a reduction to deciding whether a classical knowledge base in $\mathcal{SHOIN}(\mathbf{D})$ is satisfiable and to deciding whether a system of linear constraints is solvable. More specifically, one has to solve a sequence of solvability problems of systems of linear constraints, whose variables are computed by deciding classical knowledge base satisfiability in $\mathcal{SHOIN}(\mathbf{D})$ (see [167] for further details). This shows that the two consistency problems in P-$\mathcal{SHOIN}(\mathbf{D})$ are both decidable. Furthermore, there is a similar algorithm for computing tight intervals under lexicographic entailment in P-$\mathcal{SHOIN}(\mathbf{D})$, which is based on a reduction to deciding classical knowledge base satisfiability in $\mathcal{SHOIN}(\mathbf{D})$ and to solving linear optimization problems (see [167]). Thus, also lexicographic entailment in P-$\mathcal{SHOIN}(\mathbf{D})$ is computable. As for the computational complexity, deciding the two consistency problems in P-$\mathcal{SHOIN}(\mathbf{D})$ is complete for the complexity class NEXP, while computing tight intervals under lexicographic entailment in P-$\mathcal{SHOIN}(\mathbf{D})$ belongs to FP$^{\text{NEXP}}$ [167].

Note that if the chosen classical description logic allows for decidable knowledge base satisfiability, then also the main reasoning tasks in the probabilistic extension are all decidable. (see [167, 171] for further details).

There are already implementations of its predecessor P-$\mathcal{SHOQ}(\mathbf{D})$ (see [200]) and of a probabilistic description logic based on probabilistic default reasoning as in [159, 163]. Recently, the Pronto system [1], claims to have implemented P-$\mathcal{SHOIN}(\mathbf{D})$.

Example 3.1. Suppose we have the following KB, KB, where T contains $Eagle \sqsubseteq Bird$ and $Penguin \sqsubseteq Bird$, while P contains $(Fly \mid Bird)[0.95, 1]$ and $(Fly \mid Penguin)[0, 0.05]$. Then we can infer the tightest bounds $KB \models (Fly \mid Eagle)[0.95, 1]$ and $KB \models (Fly \mid Penguin)[0, 0.05]$. □

Other approaches. Other approaches to probabilistic description logics can be classified according to the generalized classical description logics, the supported forms of probabilistic knowledge, the underlying probabilistic semantics, and the reasoning techniques.

One of the earliest approaches to probabilistic description logics is due to Heinsohn [99], who presents a probabilistic extension of the description logic $\mathcal{ALC}$, which allows to represent terminological probabilistic knowledge about concepts and roles, and which is based on the notion of logical entailment in probabilistic logics, similar to [5, 81, 157, 207]. Heinsohn [99], however, does not allow for assertional (classical or probabilistic) knowledge about concept and role instances. The main reasoning problems are deciding the consistency of probabilistic terminological knowledge bases and computing logically entailed tight probability intervals. Heinsohn proposes a sound and complete global reasoning technique based on classical reasoning in $\mathcal{ALC}$ and linear programming, as well as a sound but incomplete local reasoning technique based on the iterative application of local inference rules.

[1] `http://clarkparsia.com/weblog/2007/09/27/introducing-pronto/`

Another early approach to probabilistic description logics is due to Jaeger [112], who also proposes a probabilistic extension of the description logic $\mathcal{ALC}$, which allows for terminological probabilistic knowledge about concepts and roles, and assertional probabilistic knowledge about concept instances, but does not support assertional probabilistic knowledge about role instances (but he mentions a possible extension in this direction). The entailment of terminological probabilistic knowledge from terminological probabilistic knowledge is based on the notion of logical entailment in probabilistic logic, while the entailment of assertional probabilistic knowledge from terminological and assertional probabilistic knowledge is based on a cross-entropy minimization relative to terminological probabilistic knowledge. The main reasoning problems are terminological probabilistic consistency and inference, which are solved by linear programming, and assertional probabilistic consistency and inference, which are solved by an approximation algorithm.

The recent work by Dürig and Studer [66] presents a further probabilistic extension of $\mathcal{ALC}$, which is based on a model-theoretic semantics as in probabilistic logics, but which only allows for assertional probabilistic knowledge about concept and role instances, and not for terminological probabilistic knowledge. The paper also explores independence assumptions for assertional probabilistic knowledge. The main reasoning problem is deciding the consistency of assertional probabilistic knowledge, but neither an algorithm nor a decidability result is given.

Jaeger's recent work [113] 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 semantics is different from the semantics of all the other probabilistic description logics in this paper, since it is based on probability distributions over the domain, and not on the more commonly used probability distributions over a set of possible worlds. The paper proposes a sound Gentzen-style sequent calculus for the logic, but it neither proves the completeness of this calculus nor decidability in general.

Koller et al.'s work [125] presents the probabilistic description logic P-CLASSIC, which is a probabilistic generalization (of a variant) of the description logic CLASSIC. Similar to Heinsohn's work [99], it allows for encoding terminological probabilistic knowledge about concepts, roles, and attributes (via so-called p-classes), but it does not support assertional (classical or probabilistic) knowledge about instances of concepts and roles. However, in contrast to [99], its probabilistic semantics is based on a reduction to Bayesian networks. The main reasoning problem is to determine the exact probabilities for conditionals between concept expressions in canonical form. This problem is solved by a reduction to inference in Bayesian networks. As an important feature of P-CLASSIC, the above problem can be solved in polynomial time, when the underlying Bayesian network is a polytree. Note that a recent implementation of P-CLASSIC is described in [115].

Closely related work by Yelland [285] proposes a probabilistic extension of a description logic close to $\mathcal{FL}$, whose probabilistic semantics is also based on a reduction to Bayesian networks, and it applies this approach to market analysis. The approach allows for encoding terminological probabilistic knowledge about concepts and roles, but it does not support assertional (classical or probabilistic) knowledge about instances of concepts and roles. Like in Koller et al.'s work [125], the main reasoning problem is to determine the exact probabilities for conditionals between concepts, which is solved by a reduction to inference in Bayesian networks.

Probabilistic Web Ontology Languages. The literature contains several probabilistic generalizations of web ontology languages. Many of these approaches focus especially on combining the web ontology language OWL with probabilistic formalisms based on Bayesian networks.

In particular, da Costa [28], da Costa and Laskey [29], and da Costa et al. [30] suggest a probabilistic generalization of OWL, called PR-OWL, whose probabilistic semantics is based on multi-entity Bayesian networks (MEBNs). The latter are a Bayesian logic that combines first-order logic with Bayesian networks. Roughly speaking, PR-OWL represents knowledge as parameterized fragments of Bayesian networks. Hence, it can encode probability distributions on the interpretations of an associated first-order theory as well as repeated structure.

In [54, 55], Ding et al. propose a probabilistic generalization of OWL, called Bayes-OWL, 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, and it also provides a method for incorporating available probability constraints when constructing the Bayesian network. The generated Bayesian network, which preserves the semantics of the original ontology and which is consistent with all the given probability constraints, supports ontology reasoning, both within and across ontologies, as Bayesian inferences. In [55, 212], Ding et al. also describe an application of the BayesOWL approach in ontology mapping.

In closely related work, Mitra et al. [194] describe an implemented technique, called OMEN, to enhancing existing ontology mappings by using a Bayesian network to represent the influences between potential concept mappings across ontologies. More concretely, OMEN is based on a simple ontology model similar to RDF Schema. It uses a set of meta-rules that capture the influence of the ontology structure and the semantics of ontology relations, and matches nodes that are neighbours of already matched nodes in the two ontologies.

Yang and Calmet [282] present an integration of the web ontology language OWL with Bayesian networks, called OntoBayes. The approach makes use of probability and dependency-annotated OWL to represent uncertain information in Bayesian networks. The work also describes an application in risk analysis for insurance and natural disaster management. Pool and Aikin [214] also provide a method for representing uncertainty in OWL ontologies, while Fukushige [83] proposes a basic framework for representing probabilistic relationships in RDF. Nottelmann and Fuhr [208] present two probabilistic extensions of variants of OWL Lite, along with a mapping to locally stratified probabilistic Datalog.

Another important work is due to Udrea et al. [272], who present a probabilistic generalization of RDF, which allows for representing terminological probabilistic knowledge about classes and assertional probabilistic knowledge about properties of individuals. They provide a technique for assertional probabilistic inference in acyclic probabilistic RDF theories, which is based on the notion of logical entailment in probabilistic logic, coupled with a local probabilistic semantics. They also provide a prototype implementation of their algorithms.

An important application for probabilistic ontologies (and thus probabilistic description logics and ontology languages) is especially information retrieval. In particular, Subrahmanian's group [109, 271] 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. [182] 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 [101, 102] shows how degrees of overlap between concepts can be modelled 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. Finally, Weikum et al. [280] and Thomas and Sheth [268] describe the use of probabilistic ontologies in information retrieval from a more general perspective.

Possibilistic Uncertainty and Description Logics. Similar to probabilistic extensions of description logics, possibilistic extensions of description logics have been developed by Hollunder [107]; Dubois et al. [58] and more recently in [217].

A *possibilistic axiom* is of the form $\mathsf{P}\,\alpha \geqslant l$ or $\mathsf{N}\,\alpha \geqslant l$, where α is a classical description logic axiom, and l is a real number from $[0,1]$. A *possibilistic RBox* (resp., *TBox*, *ABox*) is a finite set of possibilistic axioms $\mathsf{P}\,\alpha \geqslant l$ or $\mathsf{N}\,\alpha \geqslant l$, where α is an RBox (resp., TBox, ABox) axiom. A *possibilistic knowledge base* $KB = (\mathcal{R}, \mathcal{T}, \mathcal{A})$ consists of a possibilistic RBox $\mathcal{R}$, a possibilistic TBox $\mathcal{T}$, and a possibilistic ABox $\mathcal{A}$. The semantics is a straightforward extension from the propositional case to the FOL case.

The main reasoning problems related to possibilistic description logics are deciding whether a possibilistic knowledge base is satisfiable, deciding whether a possibilistic axiom is a logical consequence of a possibilistic knowledge base, and computing the tight lower and upper bounds entailed by a possibilistic knowledge base for the necessity and the possibility of a classical description logic axiom. As shown by Hollunder [107], deciding logical consequences, and thus also deciding satisfiability and computing tight lower and upper bounds can be reduced to deciding logical consequences in classical description logics.

Example 3.2. Suppose that the KB, KB, contains

$$\begin{aligned}
\mathsf{N}(\exists owns.Porsche &\sqsubseteq CarFanatic \sqcup RichPerson) \geqslant 0.8\\
\mathsf{P}(RichPerson &\sqsubseteq Golfer) \geqslant 0.7\\
\mathsf{N}((tom, 911)\colon owns) &\geqslant 1\\
\mathsf{N}(911\colon Porsche) &\geqslant 1\\
\mathsf{N}(tom\colon \neg CarFanatic) &\geqslant 0.7\,.
\end{aligned}$$

We are interested to the question whether or not that Tom is a golfer. It can be shown that

$$KB \models \mathsf{P}(tom\colon Golfer) \geqslant 0.7\,. \qquad \Box$$

A recent implementation of reasoning in possibilistic description logics using KAON2[2] is reported in [218, 219].

We recall that Liau and Yao [139] report on an application of possibilistic description logics in information retrieval. More concretely, they define a possibilistic generalization of the description logic $\mathcal{ALC}$ and show that it can be used in typical information retrieval problems, such as query relaxation, query restriction, and exemplar-based retrieval. Possibilistic description logics can also be used for handling inconsistencies in ontologies [218, 219]. Another important application of possibilistic description logics is the representation of user preferences in the Semantic Web. For example, the recent work by Hadjali et al. [90] shows that possibilistic logic can be nicely used for encoding user preferences in the context of databases.

Vagueness and Description Logics. There are several extensions of description logics and ontology languages using the theory of fuzzy logic. They can be classified according to (a) the description logic resp. ontology language that they generalize, (b) the allowed fuzzy constructs, (c) the underlying fuzzy logics, and (d) their reasoning algorithms.

In general, fuzzy DLs allow expressions of the form $(a\colon C, n)$, stating that a is an instance of concept C with degree at least n, that is the FOL formula $C(a)$ is true to degree at least n (it is straightforward to map DL expressions into FOL formulae). Similarly, $(C_1 \sqsubseteq C_2, n)$ and $(R_1 \sqsubseteq R_2, n)$ state vague subsumption relationships. Informally, $(C_1 \sqsubseteq C_2, n)$ dictates that the FOL formula $\forall x.C_1(x) \rightarrow C_2(x)$ is always true to degree at least n (note that in mathematical fuzzy logic, the universal quantification $\forall x$ is interpreted as $\inf_x$, and similarly, $\exists x$ is interpreted as $\sup_x$ and, that not always $\neg\forall$ is the same as $\exists\neg$, –this is true only for Zadeh logic and Łukasiewicz logic).

Specifically, fuzzy DLs supports concrete data types such as reals, integers, strings and allows the definition of concepts with explicit representation of fuzzy membership functions. This is implemented by relying on so-called fuzzy data type theory. A *fuzzy data type theory* $\mathbf{D} = (\Delta_{\mathbf{D}}, \cdot_{\mathbf{D}})$ is such that $\cdot_{\mathbf{D}}$ assigns to every n-ary data type predicate d an n-ary fuzzy relation over $\Delta_{\mathbf{D}}$ [176]. For instance, the predicate $\leqslant_{18}$ may be a unary crisp predicate over the natural numbers denoting the set of integers smaller or equal to 18. Concerning non-crisp fuzzy domain predicates, we recall that in fuzzy set theory and practice, there are many functions for specifying fuzzy set membership degrees. However, the trapezoidal (Fig. 2 (a)), the triangular (Fig. 2 (b)), the L-function (left-shoulder function, Fig. 2 (c)), and the R-function (right-shoulder function, Fig. 2 (d)) are simple, but most frequently used to specify membership degrees. These functions are defined over the set of non-negative rationals $\mathbb{Q}^+ \cup \{0\}$ For instance, we may define $Young\colon \mathbb{N} \rightarrow [0,1]_D$ to be a fuzzy concrete predicate over the natural numbers denoting the degree of youngness of a person's age, as $Young(x) = ls(10, 30)$.

[2] http://kaon2.semanticweb.org/

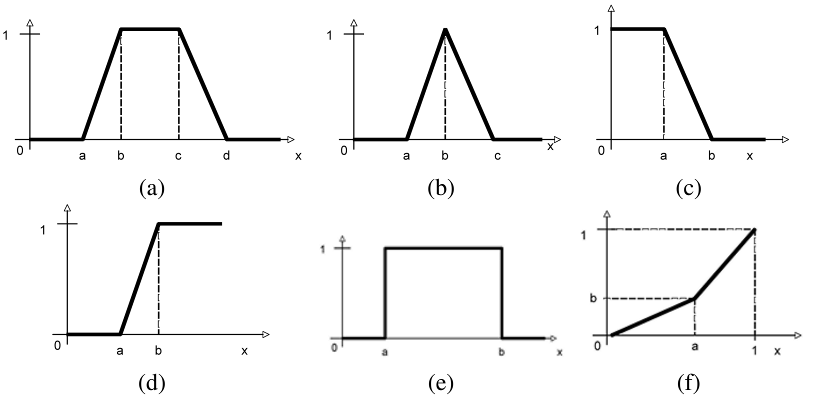


Fig. 2. (a) Trapezoidal function $trz(a,b,c,d)$; (b) Triangular function $tri(a,b,c)$; (c) L-function $ls(a,b)$; (d) R-function $rs(a,b)$; (e) Crisp interval $cr(a,b)$; (f) Linear function $ln(a,b)$

Fuzzy DLs allow fuzzy modifiers, such as $very$, $more\_or\ \_less$ and $slightly$, which apply to fuzzy sets to change their membership function. Formally, a *modifier* is a function $f_m\colon [0,1] \to [0,1]$. We will allow modifiers defined in terms of linear hedges (Figure 2 (f)) and triangular functions (Figure 2 (d)). Modifiers have also been considered in [105, 269].

Furthermore, fuzzy DLs extend crisp DLs with some specific constructs, which we define next (see, e.g. [14]). Let $\mathbf{A}$, $\mathbf{R}_A$, $\mathbf{R}_D$, $\mathbf{I}$, I_c and $\mathbf{M}$ be non-empty finite and pairwise disjoint sets of *concepts names* (denoted A), *abstract roles names* (denoted R), *concrete roles names* (denoted T), *abstract individual names* (denoted x, y), *concrete individual names* (denoted v) and *modifiers* (denoted m). Concepts may be seen as unary predicates, while roles may be seen as binary predicates. $\mathbf{R}_A$ also contains a non-empty subset $\mathbf{F}_a$ of *abstract feature names* (denoted r), while $\mathbf{R}_D$ contains a non-empty subset $\mathbf{F}_c$ of *concrete feature names* (denoted t). Features are functional roles. Besides the usual concept forming constructs, a fuzzy DL supports also constructs dealing with concrete data types, that is it has the additional concept constructs:

$$
\begin{array}{rl}
C, D & := \forall T.d \mid \exists T.d \mid DR \\
d & := cr(a,b) \mid ls(a,b) \mid rs(a,b) \mid \\
 & \quad\; tri(a,b,c) \mid trz(a,b,c,d) \\
DR & := \geqslant t\ val \mid\ \leqslant t\ val \mid\ = t\ val
\end{array}
$$

where val is an integer, a real or a string depending on the range of the concrete feature t. For instance, the expression $Human \sqcap (\leqslant hasAge\ 18)$ will denote the set of humans, which have an age less or equal than 18, while $Human \sqcap \exists hasAge.ln(10,30)$ will denote the set of young humans (their age is $L(10,30)$).

Finally, additional useful concept constructs are:

$$
\begin{array}{rl}
C, D & := C \sqcap_G D \mid C \sqcap_{Ł} D \mid C \sqcup_G D \mid C \sqcup_{Ł} D \mid \\
 & \quad\; C \to\ D \mid C \to_G D \mid C \to_{Ł} D \mid m(C) \mid \\
 & \quad\; nC \mid w_1 C_1 + \cdots + w_k C_k \mid C[\geqslant\ n] \mid C[\leqslant\ n] \\
m & := ln(a,b) \mid tri(a,b,c)
\end{array}
$$

where $n \in [0,1]_D, w_i \in [0,1]_D, \sum_{i=1}^{k} w_i = 1$. For instance, the concept $m(C)$ applies the modifier m to the concept C and, thus, e.g. $Human \sqcap \exists hasAge.ln(0.8, 0.3)(ls(10, 30))$ denotes the set of *very* young humans.

A *fuzzy knowledge base* (KB) $KB = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ consists of a fuzzy ABox $\mathcal{A}$, a fuzzy TBox $\mathcal{T}$ and a fuzzy RBox $\mathcal{R}$.

A *fuzzy ABox* $\mathcal{A}$ consists of a finite set of *fuzzy concept* and *fuzzy role assertion axioms* of the form $\langle x \colon C, \alpha\rangle$ and $\langle (x, y) \colon R, \alpha\rangle$, where $\alpha \in (0,1]_D$. Informally, from a semantical point of view, $\langle \tau, \alpha\rangle$ constrains the membership degree of τ to be at least α. Hence, $\langle jim \colon YoungPerson, 0.2\rangle$ says that jim is a $YoungPerson$ with degree at least 0.2, while $\langle (jim, tom) \colon hasFriend, 1\rangle$, states that jim and tom are friends. If the α is omitted, 1 is assumed.

A *fuzzy TBox* $\mathcal{T}$ is a finite set of *fuzzy General Concept Inclusion axioms* (GCIs) $\langle C \sqsubseteq D, \alpha\rangle$, where $\alpha \in (0,1]_D$ and C, D are concepts. Informally, $\langle C \sqsubseteq D, \alpha\rangle$ states that all instances of concept C are instances of concept D to degree α, that is, the subsumption degree between C and D is at least α. For instance, $\langle Elephant \sqsubseteq Animal, 1\rangle$ states that the class of elephants is a subclass of the class of animals. We write $C = D$ as a shorthand of the two axioms $\langle C \sqsubseteq D, 1\rangle$ and $\langle D \sqsubseteq C, 1\rangle$. For instance, $Minor = Person \sqcap (\leqslant hasAge\ 18)$ defines a person, whose age is less or equal to 18 ($hasAge$ is a concrete feature), that is a minor. If the truth value α is omitted then the value 1 is assumed.

Fuzzy DLs also allow to write $\sqsubseteq_{\Rightarrow}$ in order to specify the particular implication function to be used in the semantics of the GCI (General Concept Inclusion Axiom), e.g., Łukasiewicz or Gödel.

A *fuzzy RBox* $\mathcal{R}$ is a finite set of role axioms of the form:

- $(fun\ R)$, stating that a role R is functional, that is R is a feature.
- $(trans\ R)$, stating that a role R is transitive.
- $R_1 \sqsubseteq R_2$, meaning that role R_2 subsumes role R_1.
- $(inv\ R_1\ R_2)$, stating that role R_2 is the inverse of R_1 (and vice versa).

A simple role is a role which is neither transitive nor has a transitive subroles. An important restriction is that functional needs to be simple.

Semantics. The main idea is that concepts and roles are interpreted as fuzzy subsets of an interpretation's domain. Therefore, axioms, rather than being "classical" evaluated (being either true or false), they are "many-valued" evaluated in $[0,1]_D$.

A *fuzzy interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ relative to a fuzzy data type theory $\mathbf{D} = (\Delta_{\mathbf{D}}, \cdot_{\mathbf{D}})$ consists of a nonempty set $\Delta^{\mathcal{I}}$ (the *domain*), disjoint from $\Delta_{\mathbf{D}}$, and of a *fuzzy interpretation function* $\cdot^{\mathcal{I}}$ that coincides with $\cdot_{\mathbf{D}}$ on every data value, data type, and fuzzy data type predicate, and it assigns:

- to each abstract concept C a function $C^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \to [0,1]$;
- to each abstract role R a function $R^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0,1]$;
- to each abstract feature r a partial function $r^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0,1]$ such that for all $x \in \Delta^{\mathcal{I}}$ there is an unique $y \in \Delta^{\mathcal{I}}$ on which $r^{\mathcal{I}}(x, y)$ is defined;
- to each concrete role T a function $R^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \times \Delta_{\mathbf{D}} \to [0,1]$;

Table 5. Semantics of the complex fuzzy concepts

$$\begin{aligned}
\bot^{\mathcal{I}}(x) &= 0 \\
\top^{\mathcal{I}}(x) &= 1 \\
(\neg C)^{\mathcal{I}}(x) &= \ominus C^{\mathcal{I}}(x) \\
(C \sqcap D)^{\mathcal{I}}(x) &= C^{\mathcal{I}}(x) \otimes D^{\mathcal{I}}(x) \\
(C \sqcap_G D)^{\mathcal{I}}(x) &= C^{\mathcal{I}}(x) \otimes_G D^{\mathcal{I}}(x) \\
(C \sqcap_{\text{Ł}} D)^{\mathcal{I}}(x) &= C^{\mathcal{I}}(x) \otimes_{\text{Ł}} D^{\mathcal{I}}(x) \\
(C \sqcup D)^{\mathcal{I}}(x) &= C^{\mathcal{I}}(x) \oplus D^{\mathcal{I}}(x) \\
(C \sqcup_G D)^{\mathcal{I}}(x) &= C^{\mathcal{I}}(x) \oplus_G D^{\mathcal{I}}(x) \\
(C \sqcup_{\text{Ł}} D)^{\mathcal{I}}(x) &= C^{\mathcal{I}}(x) \oplus_{\text{Ł}} D^{\mathcal{I}}(x) \\
(C \rightarrow D)^{\mathcal{I}}(x) &= C^{\mathcal{I}}(x) \Rightarrow D^{\mathcal{I}}(x) \\
(C \rightarrow_G D)^{\mathcal{I}}(x) &= C^{\mathcal{I}}(x) \Rightarrow_G D^{\mathcal{I}}(x) \\
(= t\ val)^{\mathcal{I}}(x) &= \sup_{c \in \Delta_D} t(x,v) \otimes (v = val) \\
(C \rightarrow_{\text{Ł}} D)^{\mathcal{I}}(x) &= C^{\mathcal{I}}(x) \Rightarrow_L D^{\mathcal{I}}(x)
\end{aligned}$$

$$\begin{aligned}
(m(C))^{\mathcal{I}}(x) &= f_m(C^{\mathcal{I}}(x)) \\
(\forall R.C)^{\mathcal{I}}(x) &= \inf_{y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x,y) \Rightarrow C^{\mathcal{I}}(y) \\
(\exists R.C)^{\mathcal{I}}(x) &= \sup_{y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x,y) \otimes C^{\mathcal{I}}(y) \\
(\forall T.d)^{\mathcal{I}}(x) &= \inf_{y \in \Delta_{\mathbf{D}}} T^{\mathcal{I}}(x,v) \Rightarrow d^{\mathcal{I}}(y) \\
(\exists T.d)^{\mathcal{I}}(x) &= \sup_{y \in \Delta_{\mathbf{D}}} T^{\mathcal{I}}(x,v) \otimes d^{\mathcal{I}}(y) \\
(n\ C)^{\mathcal{I}}(x) &= n\ C^{\mathcal{I}}(x) \\
(w_1 C_1 + \cdots + w_k C_k)^{\mathcal{I}}(x) &= w_1 {C_1}^{\mathcal{I}}(x) + \cdots + w_k {C_k}^{\mathcal{I}}(x) \\
(C[\geqslant\ n])^{\mathcal{I}}(x) &= \begin{cases} C^{\mathcal{I}}(x), & \text{if } C^{\mathcal{I}}(x) \geqslant n \\ 0, & \text{otherwise} \end{cases} \\
(C[\leqslant\ n])^{\mathcal{I}}(x) &= \begin{cases} C^{\mathcal{I}}(x), & \text{if } C^{\mathcal{I}}(x) \leqslant n \\ 0, & \text{otherwise} \end{cases} \\
(\geqslant t\ val)^{\mathcal{I}}(x) &= \sup_{c \in \Delta_D} t(x,v) \otimes (v \geqslant val) \\
(\leqslant t\ val)^{\mathcal{I}}(x) &= \sup_{c \in \Delta_D} t(x,v) \otimes (v \leqslant val)
\end{aligned}$$

Table 6. Semantics of other constructs

$$\begin{aligned}
(x\colon C)^{\mathcal{I}} &= C^{\mathcal{I}}(x^{\mathcal{I}}) \\
((x,y)\colon R)^{\mathcal{I}} &= R^{\mathcal{I}}(x^{\mathcal{I}}, y^{\mathcal{I}}) \\
(C \sqsubseteq D)^{\mathcal{I}} &= \inf_{x \in \Delta^{\mathcal{I}}} C^{\mathcal{I}}(x) \Rightarrow D^{\mathcal{I}}(x) \\
(C \sqsubseteq_G D)^{\mathcal{I}} &= \inf_{x \in \Delta^{\mathcal{I}}} C^{\mathcal{I}}(x) \Rightarrow_G D^{\mathcal{I}}(x) \\
(C \sqsubseteq_{\text{Ł}} D)^{\mathcal{I}} &= \inf_{x \in \Delta^{\mathcal{I}}} C^{\mathcal{I}}(x) \Rightarrow_{\text{Ł}} D^{\mathcal{I}}(x)
\end{aligned}$$

- to each concrete feature t a partial function $t^{\mathcal{I}}\colon \Delta^{\mathcal{I}} \times \Delta_{\mathbf{D}} \rightarrow [0,1]$ such that for all $x \in \Delta^{\mathcal{I}}$ there is an unique $v \in \Delta_{\mathbf{D}}$ on which $t^{\mathcal{I}}(x,v)$ is defined;
- to each modifier m the modifier function $f_m\colon [0,1] \rightarrow [0,1]$;
- to each abstract individual x an element in $\Delta^{\mathcal{I}}$;
- to each concrete individual v an element in $\Delta_{\mathbf{D}}$.

The mapping $\cdot^{\mathcal{I}}$ is extended to roles and complex concepts as specified in Table 5, while the mapping $\cdot^{\mathcal{I}}$ is extended to the other constructs as specified in Table 6.

The notion of *satisfaction* of a fuzzy axiom E by a fuzzy interpretation $\mathcal{I}$, denoted $\mathcal{I} \models E$, is defined as follows:

- $\mathcal{I} \models \langle \tau, \alpha \rangle_{\geqslant}$ iff $\tau^{\mathcal{I}} \geqslant \alpha$,
- $\mathcal{I} \models (trans\ R)$ iff $\forall_{x,y \in \Delta^{\mathcal{I}}}, R^{\mathcal{I}}(x,y) \geqslant \sup_{z \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x,z) \otimes R^{\mathcal{I}}(z,y)$,
- $\mathcal{I} \models R_1 \sqsubseteq R_2$ iff $\forall x, y \in \Delta^{\mathcal{I}}. {R_1}^{\mathcal{I}}(x,y) \leqslant {R_2}^{\mathcal{I}}(x,y)$,
- $\mathcal{I} \models (inv\ R_1\ R_2)$ iff $\forall x, y \in \Delta^{\mathcal{I}}. {R_1}^{\mathcal{I}}(x,y) = {R_2}^{\mathcal{I}}(y,x)$.

We say that concept C is *satisfiable* iff there is an interpretation $\mathcal{I}$ and an individual $x \in \Delta^{\mathcal{I}}$ such that $C^{\mathcal{I}}(x) > 0$.

For a set of axioms $\mathcal{E}$, we say that $\mathcal{I}$ *satisfies* $\mathcal{E}$ iff I satisfies each element in $\mathcal{E}$. We say that $\mathcal{I}$ is a *model* of E (resp. $\mathcal{E}$) iff $\mathcal{I} \models E$ (resp. $\mathcal{I} \models \mathcal{E}$). $\mathcal{I}$ *satisfies* (is a *model* of) a fuzzy KB $KB = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$, denoted $\mathcal{I} \models KB$, iff $\mathcal{I}$ is a model of each component $\mathcal{A}$, $\mathcal{T}$ and $\mathcal{R}$, respectively.

An axiom E is a *logical consequence* of a knowledge base KB, denoted $KB \models E$ iff every model of KB satisfies E.

Given KB and a fuzzy axiom τ of the forms $\langle x\colon C, \alpha\rangle$, $\langle (x, y)\colon R, \alpha\rangle$ or $\langle C \sqsubseteq D, \alpha\rangle$, it is of interest to compute τ's best lower degree value bound. The *greatest lower bound* of τ w.r.t. KB (denoted $glb(KB, \tau)$) is $glb(KB, \tau) = \sup\{n \mid KB \models (\tau, n)\geqslant\}$, where $\sup \emptyset = 0$. Determining the glb is called the *Best Degree Bound* (BDB) problem.

Finally, another similar problem is to compute the *best satisfiability bound* of a concept C and amounts to determine $glb(KB, C) = \sup_{\mathcal{I}} \sup_{x\in\Delta^{\mathcal{I}}}\{C^{\mathcal{I}}(x) \mid \mathcal{I} \models KB\}$. Essentially, among all models $\mathcal{I}$ of the KB, we are determining the maximal degree of truth that the concept C may have over all individuals $x \in \Delta^{\mathcal{I}}$.

Example 3.3. Assume, that a car seller sells a sedan car. A buyer is looking for a second hand passenger car. Both the buyer as well as the seller have preferences (restrictions). Our aim is to find the best agreement. The preferences are as follows. Concerning the buyer:

1. He does not want to pay more than 26000 euro (buyer reservation value).
2. If there is an alarm system in the car then he is completely satisfied with paying no more than 22300 euro, but he can go up to 22750 euro to a lesser degree of satisfaction.
3. He wants a driver insurance and either a theft insurance or a fire insurance.
4. He wants air conditioning and the external colour should be either black or grey.
5. Preferably the price is no more than 22000 euro, but he can go up to 24000 euro to a lesser degree of satisfaction.
6. The kilometer warranty is preferrably at least 175000, but he may go down to 150000 to a lesser degree of satisfaction.
7. The weights of the preferences 2-6 are, $(0.1, 0.2, 0.1, 0.2, 0.4)$. The higher the value the more important is the preference.

Concerning the seller:

1. He wants to sell no less than 22000 euro (seller reservation value)
2. If there is an navigator pack system in the car then he is completely satisfied with paying no less than 22750 euro, but he can go down to 22500 euro to a lesser degree of satisfaction.
3. Preferably the buyer buys the Insurance Plus package.
4. The kilometer warranty is preferrably at most 100000, but he may go up to 125000 to a lesser degree of satisfaction.
5. The monthly warranty is preferrably at most 60, but he may go up to 72 to a lesser degree of satisfaction.
6. If the colour is black then the car has air conditioning.
7. The weights of the preferences 2-6 are, $(0.3, 0.1, 0.3, 0.1, 0.2)$. The higher the value the more important is the preference.

We have also some background theory about the domain:

1. A sedan is a passenger car.
2. A satellite alarm system is an alarm system.
3. The navigator pack is a satellite alarm system with a GPS system.
4. The Insurance Plus package is a driver insurance together with a theft insurance.
5. The car colours are black or grey.

Now, the background theory can be encoded as:

$$
\begin{array}{l}
Sedan \sqsubseteq PassengerCar \\
SatelliteAlarm \sqsubseteq AlarmSystem \\
NavigatorPack = SatelliteAlarm \sqcap GPS\_system \\
InsurancePlus = DriverInsurance \sqcap TheftInsurance \\
\top \sqsubseteq ExColorBlack \sqcup ExColorGray \\
ExColorBlack \sqcap ExColorGray \sqsubseteq \bot \\
(fun\ HasAlarmSystem) \\
(fun\ HasAirConditioning) \\
(fun\ HasExColor) \\
(fun\ HasNavigator) \\
(fun\ HasMWarranty) \\
(fun\ HasPrice) \\
(fun\ HasKMWarranty)
\end{array}
$$

The buyer's preferences can be encoded as follows:

1. $B = (PassengerCar \sqcap (\leqslant HasPrice\ 26000))$
2. $B_1 = ((\exists HasAlarmSystem.AlarmSystem) \rightarrow (\exists Has\ Price.ls(22300, 22750)))$
3. $B_2 = ((\exists HasInsurance.DriverInsurance) \sqcap ((\exists HasInsurance.TheftInsurance)$ $\sqcup (\exists HasInsurance.FireInsurance)))$
4. $B_3 = ((\exists HasAirConditioning.Airconditioning) \sqcap (\exists HasExColor.(ExColorBlack$ $\sqcup ExColorGray)))$
5. $B_4 = (\exists HasPrice.ls(22000, 24000))$
6. $B_5 = (\exists HasKMWarranty.R(15000, 175000))$
7. $Buy = (B \sqcap ((0.1B_1) + (0.2B_2) + (0.1B_3) + (0.2B_4) + (0.4B_5)))$

Please note that the concept Buy collects all the buyer's preferences together in such a way that the higher is the maximal degree of satisfiability of Buy (that is $glb(KB, Buy)$), the more the buyer is satisfied.

The seller's preferences can be encoded as follows:

1. $S = (Sedan \sqcap (\geqslant HasPrice\ 22000))$
2. $S_1 = ((\exists HasNavigator.NavigatorPack) \rightarrow (\exists Has\ Price.rs(22500, 22750))))$
3. $S_2 = (\exists HasInsurance.InsurancePlus)$
4. $S_3 = (\exists HasKMWarranty.rs(100000, 125000))$
5. $S_4 = (\exists HasMWarranty.rs(60, 72))$
6. $S_5 = ((\exists HasExColor.ExColorBlack) \rightarrow (\exists Has\ AirConditioning.AirCondi-$ $tioning))$
7. $Sell = (S \sqcap ((0.3S_1) + (0.1S_2) + (0.3S_3) + (0.1S_4) + (0.2S_5)))$

Similarly to the buyer case, the concept $Sell$ collects all the seller's preferences together in such a way that the higher is the maximal degree of satisfiability of $Sell$ (that is $glb(KB, Sell)$), the more the seller is satisfied.

Now, it is clear that the best agreement among the buyer and the seller is determined by the maximal degree of satisfiability of the conjunction $Buy \sqcap Sell$, that is we have to determine $glb(KB, Buy \sqcap Sell)$. In particular, we rely on Łukasiewicz conjunction, which guarantees that the solution is Pareto optimal [220]. In particular, we have that

$$\begin{aligned} glb(KB, Buy \sqcap_{Ł} Sell) &= 0.7 \\ HasPrice &= 22000.0 \\ HasKMWarranty &= 175000.0 \\ HasMWarranty &= 0.0\ . \end{aligned}$$

So an optimal match (the Pareto optimal degree is 0.7625) would be an agreement with a price of 22500 euro, with 100000 kilometer warranty and a 60 month warranty. □

The first work about fuzzy DLs is due to Yen [286], who proposes a fuzzy extension of a very restricted sublanguage of $\mathcal{ALC}$, called $\mathcal{FL}^-$ [18, 133]. The work includes fuzzy terminological knowledge, but no fuzzy assertional knowledge, and it is based on Zadeh logic. It already informally talks about the use of fuzzy modifiers and fuzzy concrete domains. Though, the unique reasoning facility, the subsumption test, is a crisp yes/no questioning. Tresp and Molitor [269] consider a more general extension of fuzzy $\mathcal{ALC}$. Like Yen, they also allow for fuzzy terminological knowledge along with a special form of fuzzy modifiers (which are a combination of two linear functions), but no fuzzy assertional knowledge, and they assume Zadeh logic as underlying fuzzy logic. The work also presents a sound and complete reasoning algorithm testing the subsumption relationship using a linear programming oracle.

Another fuzzy extension of $\mathcal{ALC}$ is due to Straccia [243, 245, 251, 256, 265], who allows for both fuzzy terminological and fuzzy assertional knowledge, but not for fuzzy modifiers and fuzzy concrete domains, and again assumes Zadeh logic as underlying fuzzy logic. Straccia [243, 245] also introduces the best truth value bound problem and provides a sound and complete reasoning algorithm based on completion rules. In [244], Straccia reports a four-valued variant of fuzzy $\mathcal{ALC}$. In the same spirit, Hölldobler et al. [103, 104] extend Straccia's fuzzy $\mathcal{ALC}$ with concept modifiers of the form $f_m(x) = x^\beta$, where $\beta > 0$, and present a sound and complete reasoning algorithm (based on completion rules) for the graded subsumption problem.

Straccia's works [247, 255, 261] are essentially as [245], except that now the set of possible truth values is a complete lattice rather than $[0, 1]$.

Sanchez and Tettamanzi [227, 228, 229] consider a fuzzy extension of the description logic $\mathcal{ALCQ}$ (without assertional component) under Zadeh logic, and they start addressing the issue of a fuzzy semantics of quantifiers. Essentially, fuzzy quantifiers allow to state sentences such as $\mathit{FaithfulCustomer} \sqcap (\mathit{Most})\mathit{buys}.\mathit{LowCalorie\text{-}\ Food}$ encoding "the set of all individuals that mostly buy low calorie food". An algorithm is presented, which calculates the satisfiability interval for a fuzzy concept.

Hájek [96, 97] considers a fuzzy extension of the description logic $\mathcal{ALC}$ under arbitrary t-norms. He provides in particular algorithms for deciding whether $(C \sqsubseteq D, 1){\geqslant}$ is a tautology and whether $(C \sqsubseteq D, 1){\geqslant}$ is satisfiable, which are based on a reduction to the propositional BL logic for which a Hilbert-style axiomatization exists [95] (but see also [97] for the complexity of rational Pavelka logic, and see [16] for some complexity results on reasoning in fuzzy description logics).

Straccia [246] provides a translation of fuzzy $\mathcal{ALC}$ (with general concept inclusion axioms) into classical $\mathcal{ALC}$. The translation is modular, and thus expected to be extendable to more expressive fuzzy description logics as well. The main idea is to translate a fuzzy assertion of the form $\langle a\colon C, n\rangle_{\geqslant}$ into a crisp assertion $a\colon C_n$, with the intended meaning "a is an instance of C to degree at least n". Then, concept inclusion axioms are used to correctly relate the C_n's. For example, $C_{0.7} \sqsubseteq C_{0.6}$ is used to encode that whenever an individual is an instance of C to degree at least 0.7, then it is also an instance of C to degree at least 0.6. The translation is at most quadratic in the size of the fuzzy knowledge base. Note that the translation does not yet work in the presence of fuzzy modifiers and fuzzy concrete domains. Bobillo et al. [12] extend the approach to a variant of fuzzy $\mathcal{SHOIN}$. The idea has further been considered in the works [137, 138], which essentially provide a crisp language in which expressions of, e.g., the form $a\colon \forall R_{0.8}.C_{0.9}$ are allowed, with the intended meaning "if a has an R-successor to degree at least 0.8, then this successor is also an instance of C to degree at least 0.9". The idea has further been extended to a distributed variant of fuzzy description logics in [149]. A mapping to classical DLs under Łukasiewicz semantics has been provided in [15] for the fuzzy DL $\mathcal{ALCHOI}$.

An interesting extension is due to Kang et al. [43], who extends fuzzy description logics by comparison operators, e.g., to state that "Tom is taller than Tim". Another interesting extension is proposed by Dubois et al. [58], who combine fuzzy description logics with possibility theory. Essentially, since $\langle a\colon C, n\rangle_{\geqslant}$ is Boolean (either an interpretation satisfies it or not), we can build on top of it an uncertainty logic, which is based on possibility theory in [58].

We recall that usually the semantics used for fuzzy description logics is based on Zadeh logic, but where the concept inclusion is crisp, that is, $C \sqsubseteq D$ is viewed as $\forall x.C(x) \leqslant D(x)$. In [106, 269], a calculus for fuzzy $\mathcal{ALC}$ [230] with fuzzy modifiers and simple TBoxes under Zadeh logic is reported. No indication for the BTVB problem is given. Straccia [243, 245] reports a calculus for fuzzy $\mathcal{ALC}$ and simple TBoxes under Zadeh logic and addresses the BTVB problem. How the satisfiability problem and the BTVB problem can be reduced to classical $\mathcal{ALC}$, and thus can be solved by means of tools like FaCT and RACER is shown in [246]. Results providing a tableaux calculus for fuzzy $\mathcal{SHIN}$ under Zadeh logic (but only allowing for a restricted form of concept inclusion axioms, which are called *fuzzy inclusion introductions* and *fuzzy equivalence introductions*), by adapting similar techniques as for the classical counterpart, are shown in [239, 240]. Fuzzy general concept inclusion axioms under Zadeh logic can be managed as described in [242]. Also interesting is the work [283], which provides a tableau for fuzzy $\mathcal{SHI}$ with general concept inclusion axioms. Finally, the reasoning techniques for classical $\mathcal{SHOIN}(\mathbf{D})$ [108] can be extended to [245], as [238, 239, 240, 241] already show.

On the other hand, fuzzy tableaux algorithms under Zadeh semantics do not seem to be suitable to be adapted to other semantics, such as Łukasiewicz logic. Even more problematic is the fact that they are yet unable to deal with fuzzy concrete domains [248], that is the possibility to allow an explicit representation of fuzzy membership functions. Despite these negative results, recently, [248, 249] report a calculus for fuzzy $\mathcal{ALC}(\mathbf{D})$ whenever the connectives, the modifiers, and the fuzzy datatype predicates are representable as bounded mixed integer linear programs (MILPs). For example, Łukasiewicz logic satisfies these conditions as well as the membership functions for fuzzy datatype predicates that we have presented in this paper. Additionally, modifiers should be a combination of linear functions. In that case, the calculus consists of a set of constraint propagation rules and an invocation to an oracle for MILP. The method has been extended to fuzzy $\mathcal{SHIF}(\mathbf{D})$ [262] (the description logic behind OWL Lite) and a reasoner, called *fuzzyDL* [14], has been implemented and is available at Straccia's web page. *FuzzyDL* supports more features, which we do not address here. The use of MILP for reasoning in fuzzy description logics is not surprising as their use for automated deduction in many-valued logics is well-known [92,93]. Bobillo and Straccia [13] provide a calculus for fuzzy $\mathcal{ALC}(\mathbf{D})$ under product semantics.

A very recent problem for fuzzy description logics is the top-k retrieval problem. While in classical semantics, a tuple satisfies or does not satisfy a query, in fuzzy description logics, a tuple may satisfy a query to a degree. Hence, for example, given a conjunctive query over a fuzzy description logic knowledge base, it is of interest to compute only the top-k answers. While in relational databases, this problem is a current research area (see, e.g., [73,110,135]), very few is known for the case of first-order knowledge bases in general (but see [259]) and description logics in particular. The only works that we are aware of are [254, 260, 266], which deal with the problem of finding the top-k result over knowledge bases in a fuzzy generalization of *DL-Lite* [23] (note that [210,211] is subsumed by [260], though in [210,211] the storage systems is no-longer a database, but a RDF storage system).

Fuzzy logic has numerous practical applications in general (see, e.g., [124]). Related to fuzzy description logics, we point out that they have first been proposed for *logic-based information retrieval* [192], which originated from the idea to annotate textual documents with graded description logic sentences, which goes back to [193]. The idea has been reconsidered in [240,266,288]. In particular, (i) Zhang et al. [288] describe a semantic portal that is based on fuzzy description logics; (ii) Li et al. [136] present an improved semantic search model by integrating inference and information retrieval and an implementation in the security domain; (iii) Straccia and Visco [266] report on a multimedia information retrieval system based on a fuzzy DLR-Lite description logic, which is capable to deal with hundreds of thousands of images. D'Aquin et al. [42] provide a use case in the medical domain, where fuzzy concrete domains are used to identify tumor regions in x-ray images. Agarwal and Lamparter [1] use fuzzy description logics to improve searching and comparing products in electronic markets. They provide a more expressive search mechanism that is closer to human reasoning and that aggregates multiple search criteria to a single value (ranking of an offer relative to the query), thus enabling a better selection of offers to be considered for the negotiation. Liu et al. [140] use a fuzzy description logic to model the management part in project selection tasks. Finally, [14] shows also how to use fuzzyDLs for e-Commerce Matchmaking and Semantic Fuzzy Control.

3.2 The Case of Logic Programs

In logic programming, the management of imperfect information has attracted the attention of many researchers and numerous frameworks have been proposed. Addressing all of them is almost impossible, due to both the large number of works published in this field (early works date back to early 80-ties [236]) and the different approaches proposed (see the appendix for a list of references). Like for the DL case, essentially they differ in the underlying notion of uncertainty theory and vagueness theory (probability theory, possibilistic logic, fuzzy logic and multi-valued logic) and how uncertainty/vagueness values, associated to rules and facts, are managed.

Basically [141], a logic program $\mathcal{P}$ is made out by a set of rules and a set of facts. *Facts* are *atoms* of the form $P(t_1, \ldots, t_n)$, where t_i is a *term* (usually, a constant or a variable). In most cases, facts are ground. On the other hand rules are of the form $A \leftarrow B_1, \ldots, B_n$, where each A and B_i is an atom. $B_1, \ldots, B_n$ is called *body*, while A is called head of the rule. The intended meaning of a rules is that "if all B_i are true, then also A is true". From a FOL perspective, a rule is just a FOL formula $\forall \mathbf{x}.B_1 \wedge \ldots \wedge B_n \rightarrow A$, where $\mathbf{x}$ are all the variables occurring in the rule. Such logic programs are called *positive* as no literal occurs. In case a literal occurs in the body, then we speak about *normal logic programs*. We may also have a disjunction of atoms in the head, and then we talk about *disjunctive logic programs* ([234]). In the most general setting, literals are allowed in the head as well and from a semantics point of view, the *stable model semantics* [84] is widely adopted.

Probabilistic Uncertainty and Logic Programs. The variety of proposals of logic programming under probability theory is huge and an description of most of them is out of the scope of this work. We describe here some groups of works.

In probabilistic generalizations of (annotated) logic programs (see [122]) based on probabilistic logic fall works such as [44,45,46,47,48,201,202], where rules have the form of annotated logic programming rules. Facts are expressions of the form $A : \mu$, where μ is an interval in $[0, 1]$. The intended meaning of an expression $A : [m, n]$ is "the probability of the event corresponding to A to occur (have occurred) lies in the interval $[m, n]$". Rules have the form $A : \mu \leftarrow B_1 : \mu_1, \ldots, B_n : \mu_2$, where μ, μ_i are intervals in $[0, 1]$.

In probabilistic generalizations of logic programs based on Bayesian networks / causal models fall works such as [11,117,118,204,215,216]. Interesting is Poole's Independent Choice Logic (ICL) approach. It is based on acyclic logic programs $\mathcal{P}$ under different "choices". Each choice along with $\mathcal{P}$ produces a first-order model. By placing a probability distribution over the different choices, one then obtains a distribution over the set of first-order models. Roughly, rules and facts are as for classical logic programs. Additionally, there is a set C of choices of the form $\{(A_1 : \alpha_1), ..., (A_n : \alpha_n)\}$, where A_i is an atom and the α_i sum-up to 1. A *total choice* T_C is a set of atoms such that from each choice $C_j \in C$ there is exactly one atom $A_i^i \in C_j$ in T_C. The probability of a query q w.r.t. to $\mathcal{P}$ is the sum of the probabilities p_C of total choices T_C such that $\mathcal{P} \cup T_C \models q$, where p_C is the product of the α_i^j, for $C_i^j \in T_C$. It is worth to note that the ICL approach generalizes Bayesian networks, influence diagrams, Markov decision processes, and normal form games.

In the third group fall first-order generalization of probabilistic knowledge bases in probabilistic logic (based on logical entailment, lexicographic entailment, and maximum entropy entailment) and comprises works such as [153,160,162]. In these works, similarly to P-$\mathcal{SHOIN}(\mathbf{D})$, expressions are of the form $(\psi|\phi)[l, u]$, but now ψ, ϕ are formulae rather than concepts. The development of the semantics parallels to the case of P-$\mathcal{SHOIN}(\mathbf{D})$.

For the sake of a concrete example, let us here formally introduce Poole's ICL-based approach. Let us denote with HB_{Φ} (resp., HU_{Φ}) the Herbrand base (resp., universe) over Φ, where Φ is a function-free first-order vocabulary Φ with finite nonempty sets of constant symbols and predicate symbols.

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 $\mu\colon \bigcup C \rightarrow [0,1]$ 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 logic program* $KB = (P, C, \mu)$ consists of a logic program P, and 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 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 logic program compactly represents a probability distribution on a finite set of logic programs. A *probabilistic query* to KB is defined as follows. A formula is inductively defined as (i) and atom; (ii) if ϕ, ψ are formulae, so are $\phi \vee \psi, \phi \wedge \psi, \neg\psi, \phi \rightarrow \psi$. If ϕ is a formula and $l, u \in [0,1]$ then $\exists\phi[l,u]$, is a probabilistic query.

Semantics. A *world* I is an interpretation over HB_{Φ}. We denote by $\mathcal{I}_{\Phi}$ the set of all worlds over Φ. A *variable assignment* σ maps each variable x to some $t \in HU_{\Phi}$. It is extended to all terms by $\sigma(c) = c$ for all constant symbols c from Φ. A world I under σ is a model of an atom A, denoted $I \models_{\sigma} A$, iff $A\sigma \in I$. The extension of I under σ is a model of a formula ϕ is as usual. A world I under σ is a model of a rule $A \leftarrow B_1, \ldots, B_n$ iff $I \models_{\sigma} B_1 \wedge \ldots \wedge B_n \rightarrow A$.

A *probabilistic interpretation* Pr is a probability function on $\mathcal{I}_{\Phi}$ (that is, a mapping $Pr\colon \mathcal{I}_{\Phi} \rightarrow [0,1]$ 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 a formula ϕ in Pr under a variable assignment σ, denoted $Pr_{\sigma}(\phi)$ (or $Pr(\phi)$ when ϕ is ground), is the sum of all $Pr(I)$ such that $I \in \mathcal{I}_{\Phi}$ and $I \models_{\sigma} \phi$.

A probabilistic interpretation Pr is a *model* of a query $\exists\phi[l,u]$ iff $Pr_{\sigma}(\phi) \in [l,u]$ for every variable assignment σ. We say Pr is the *canonical model* of a probabilistic logic program $KB = (P, C, \mu)$ iff every world $I \in \mathcal{I}_{\Phi}$ with $Pr(I) > 0$ is the minimal model of $P \cup \{p \leftarrow | \, p \in B\}$ for some total choice B of C with $Pr(I) = \mu(B)$. Notice that every KB has a unique canonical model Pr. We say that a query $\exists\phi[l,u]$ is a *consequence* of KB, denoted $KB \mathrel{\|\!\sim} \exists\phi[l,u]$, iff the canonical model of KB is also a model of $\exists\phi[l,u]$. A query $\exists\phi[l,u]$ is a *tight consequence* of KB, denoted $KB \mathrel{\|\!\sim}_{tight} \exists\phi[l,u]$, iff l (resp., u) is the infimum (resp., supremum) of $Pr_{\sigma}(\phi)$ subject to the canonical model Pr of KB and all σ. A *correct answer* to $\exists\phi[l,u]$ is a substitution σ such that

$\exists\phi\sigma[l,u]$ is a consequence of KB. A *tight answer* to $\exists\phi[l,u]$ is a substitution σ such that $\exists\phi\sigma[l,u]$ is a tight consequence of KB.

As in Section 2.1, we introduce *conditional formulae* of the form $\phi \mid \psi$, where ϕ and ψ are formulae, and *conditional probabilistic queries* of the form $\exists(\phi \mid \psi)[l,u]$. A probabilistic interpretation Pr is a *model* of a conditional probabilistic query $\exists(\phi \mid \psi)[l,u]$ iff $Pr_\sigma(\phi \mid \psi) \in [l,u]$ for every variable assignment σ, where $Pr_\sigma(\phi \mid \psi)$ is defined similarly as in Eq. 2:

$$Pr_\sigma(\phi \mid \psi) = \begin{cases} \frac{Pr_\sigma(\phi \wedge \psi)}{Pr_\sigma(\psi)} & \text{if} Pr_\sigma(\psi) \neq 0 \\ 1 & \text{otherwise .} \end{cases}$$

Example 3.4. Let us show how we may encode the BN in Example 2.1 into probabilistic logic programs. For each variable a we consider an unary predicate $a(x)$, where the variable x will take either the value T or F. If a node a has no parents then we can encode its associated probability table as follows: we consider the rule

$$a(x) \leftarrow h_a(x)$$

and we consider the alternative C_a in the choice space C,

$$C_a = \{h_a(T), h_a(F)\}$$

with $\mu(h_a(T)) = Pr(a(T)), \mu(h_a(F)) = Pr(a(F)) = 1 - Pr(a(T))$. For instance, related to Fig. 1, we will have

$$\begin{aligned} Rain(x) &\leftarrow h_{Rain}(x) \\ C_{Rain} &= \{h_{Rain}(T), h_{Rain}(F)\} \\ \mu(h_{Rain}(T)) &= 0.2 \\ \mu(h_{Rain}(F)) &= 0.8\ . \end{aligned}$$

If a node a has parents $b_1, \ldots, b_n$, we encode its associated conditional probability table using a rule and an alternative in the choice space:

$$a(x) \leftarrow b_1(x_1), \ldots, b_n(x_n), h_a(x, x_1, \ldots, x_n)$$

and we consider the alternative C_a in the choice space C,

$$C_a = \{h_a(v, v_1, \ldots, v_n) \mid v, v_i \in \{T, F\}\}\ .$$

with

$$\mu(h_a(v, v_1, \ldots, v_n)) = Pr(a = v \mid b_1 = v_1, \ldots, b_n = v_n)\ .$$

For instance, related to Fig. 1, we will have

$$\begin{aligned} Sprinkler(x) &\leftarrow Rain(x_1), h_{Sprinkler}(x, x_1) \\ C_{Sprinkler} &= \{h_{Sprinkler}(T,F), h_{Sprinkler}(T,T), \\ &\quad\ h_{Sprinkler}(F,F), h_{Sprinkler}(F,T)\} \\ \mu(h_{Sprinkler}(T,F)) &= 0.4 \\ \mu(h_{Sprinkler}(T,T)) &= 0.01 \\ \mu(h_{Sprinkler}(F,F)) &= 0.6 \\ \mu(h_{Sprinkler}(F,T)) &= 0.99\ . \end{aligned}$$

The encoding for the node $\mathit{GrassWet}$ is similar.

Then, we may infer that

$$\begin{aligned}
&KB \;\Vert\!\!\sim_{tight}\; \mathit{GrassWet}(T)[0.4484, 0.4484]\\
&KB \;\Vert\!\!\sim_{tight}\; (\mathit{Rain}(T) \land \mathit{GrassWet}(T))[0.1604, 0.1604]\\
&KB \;\Vert\!\!\sim_{tight}\; (\mathit{Rain}(T) \mid \mathit{GrassWet}(T))[0.3577, 0.3577]\,,
\end{aligned}$$

Note that $0.3577 = 0.1604/0.4484$. □

Possibilistic Uncertainty and Logic Programs. In possibilistic logic programs [57], facts are of the form $(P(t_1, \ldots, t_n), \mathsf{N}\, l)$, while rules are of the form $(A \leftarrow B_1, \ldots, B_n, \mathsf{N}\, l)$. The meaning of them is given directly by the possibilistic FOL formulae, $\mathsf{N}\, P\, (t_1, \ldots, t_n),\ \geqslant l$ and $\mathsf{N}\, (\forall \mathbf{x}.B_1 \land \ldots \land B_n \to A) \geqslant l$, respectively (the necessity of the formula is greater or equal than l). This basic form has been extended in [206] (which describes also an implementation) to the case of disjunctive logic programming under the stable model semantics, while [2, 3, 4, 25] allow explicitly to deal with fuzzy sets in the language.

Vagueness and Logic Programs. While there is a large literature related to the management of vagueness in logic programs, there are rule forms that are general enough to cover a large amount of them (see e.g., [174, 250, 276]). Roughly, rules are of the form $A \leftarrow f(B_1, ..., B_n)$, where A, B_i are atoms and f is a total function $f : \mathcal{S}^n \to \mathcal{S}$ over a truth space $\mathcal{S}$. Computationally, given an assignment/interpretation I of values to the B_i, the value of A is computed by stating that A is at least as true as $f(I(B_1), ..., I(B_n))$. The form of the rules is sufficiently expressive to encompass many approaches to many-valued logic programming. [174] provides an even more general setting as the function f may also depend on the variables occurring in the rule body. On the other hand there are also some extensions to many-valued disjunctive logic programs [186, 187, 253]. In some cases, e.g. [130] there is also a function g, which dictates how to aggregate the truth values in case an atom is head of several rules.

Most works deal with logic programs without negation and some may provide some technique to answer queries in a top-down manner, as e.g. [35, 122, 130, 252, 276]. On the other hand, there are very few works dealing with normal logic programs [38, 78, 80, 142, 143, 144, 145, 146, 147, 148, 173, 186, 250, 253, 258, 263], and little is know about top-down query answering procedures. The only exceptions are [250, 258, 263].

Another rising problem is the problem to compute the top-k ranked answers to a query, without computing the score of all answers. This allows to answer queries such as "find the top-k closest hotels to the conference location". Solutions to this problem can be found in [174, 259, 264].

For illustrative purposes, we formally present a quite general logic programming formalism dealing with vagueness.

The truth space that we consider here is the finite set $[0, 1]_m = \{\frac{0}{m}, \frac{1}{m}, \ldots, \frac{m-1}{m}, \frac{m}{m}\}$ (for a natural number $m > 0$), which is pretty common in fuzzy logic. Throughout the paper, we assume $m = 100$ in the examples with usual decimal rounding (e.g., 0.375 becomes 0.38, while 0.374 becomes 0.37).

ID	MODEL	TYPE	PRICE	KM	COLOR	AIRBAG	INTERIOR TYPE	AIR COND	ENGINE FUEL
455	MAZDA 3	Sedan	12500	10000	Red	0	VelvetSeats	1	Gasoline
34	ALFA 156	Sedan	12000	15000	Black	1	LeatherSeats	0	Diesel
1812	FORD FOCUS	StationVagon	11000	16000	Gray	1	LeatherSeats	1	Gasoline

Fig. 3. The car table

A *knowledge base KB* consists of a *facts component F* and an *LP component P*, which are defined below.

Facts Component. F is a finite set of expressions of the form

$$\langle R(c_1, \ldots, c_n), s\rangle ,$$

where R is an n-ary relation, every c_i is a constant, and s is a degree of truth (or simply *score*) in $[0,1]_m$. For each R, we may represent the facts $\langle R(c_1, \ldots, c_n), s\rangle$ in F by means of a relational $n+1$-ary table T_R, containing the records $(c_1, \ldots, c_n, s)$. We assume that there cannot be two records $(c_1, \ldots, c_n, s_1)$ and $(c_1, \ldots, c_n, s_2)$ in T_R with $s_1 \neq s_2$ (if there are, then we remove the one with the lower score). Each table is sorted in descending order with respect to the scores. For ease, we may omit the score component and in such cases the value 1 is assumed.

Example 3.5 ([221]). Suppose we have a car selling site, and we would like to buy a car. The cars belong to the relation $\mathit{CarTable}$ shown in Fig. 3. Here, the score is implicitly assumed to be 1 in each record. For instance, the first record corresponds to the fact

$$\langle \mathit{CarTable}(\mathit{455}, \mathit{MAZDA3}, \mathit{Sedan}, \mathit{12500}, \mathit{10000}, \mathit{Red}, \mathit{0}, \mathit{VelvetSeats}, \mathit{1}, \mathit{Gasoline}), 1\rangle .$$

□

LP Component. P is a finite set of *vague rules* of the form (an example of a rule is shown in Example 3.7 below.)

$$R(\mathbf{x}) \leftarrow \exists \mathbf{y}.f(R_1(\mathbf{z}_1), \ldots, R_l(\mathbf{z}_l), p_1(\mathbf{z}'_1), \ldots, p_h(\mathbf{z}'_h)) ,$$

where

1. R is an n-ary relation, every R_i is an n_i-ary relation,
2. $\mathbf{x}$ are the *distinguished variables*;
3. $\mathbf{y}$ are existentially quantified variables called the *non-distinguished variables*;
4. $\mathbf{z_i}, \mathbf{z'_j}$ are tuples of constants or variables in $\mathbf{x}$ or $\mathbf{y}$;
5. p_j is an n_j-ary *fuzzy predicate* assigning to each n_j-ary tuple $\mathbf{c}_j$ a *score* $p_j(\mathbf{c}_j) \in [0,1]_m$. Such predicates are called *expensive predicates* in [26] as the score is not pre-computed off-line, but is computed on query execution. We require that an n-ary fuzzy predicate p is *safe*, that is, there is not an m-ary fuzzy predicate p' such that $m < n$ and $p = p'$. Informally, all parameters are needed in the definition of p;
6. f is a *scoring* function $f\colon ([0,1]_m)^{l+h} \to [0,1]_m$, which combines the scores of the l relations $R_i(\mathbf{c}'_i)$ and the n fuzzy predicates $p_j(\mathbf{c}''_j)$ into an overall *score*

$\mathit{Vehicles}(x) \leftarrow \mathit{Cars}(x)$
$\mathit{Vehicles}(x) \leftarrow \mathit{Trucks}(x)$
$\mathit{Vehicles}(x) \leftarrow \mathit{Vans}(x)$
$\mathit{Cars}(x) \leftarrow \mathit{LuxuryCars}(x)$
$\mathit{Cars}(x) \leftarrow \mathit{PassengerCars}(x)$
$\mathit{Cars}(x_1) \leftarrow \mathit{CarTable}(x_1, \ldots, x_{10})$
$\mathit{Cars}(x) \leftarrow \mathit{Sedan}(x)$
$\mathit{Cars}(x) \leftarrow \mathit{StationWagon}(x)$
$\mathit{Seats}(x_8) \leftarrow \mathit{CarTable}(x_1, \ldots, x_{10})$

$\mathit{Seats}(x) \leftarrow \mathit{LeatherSeats}(x)$
$\mathit{Seats}(x) \leftarrow \mathit{VelvetSeats}(x)$
$\mathit{PassengerCars}(x) \leftarrow \mathit{MidSizeCars}(x)$
$\mathit{PassengerCars}(x) \leftarrow \mathit{SportyCars}(x)$
$\mathit{PassengerCars}(x) \leftarrow \mathit{CompactCars}(x)$
$\mathit{hasPrice}(x_1, x_4) \leftarrow \mathit{CarTable}(x_1, \ldots, x_{10})$
$\mathit{hasKm}(x_1, x_5) \leftarrow \mathit{CarTable}(x_1, \ldots, x_{10})$
$\mathit{FuelType}(x_1, x_{10}) \leftarrow \mathit{CarTable}(x_1, \ldots, x_{10})$

Fig. 4. Excerpt of a car selling ontology

to be assigned to the rule head $R(\mathbf{c})$. We assume that f is *monotone*, that is, for each $\mathbf{v}, \mathbf{v}' \in ([0,1]_m)^{l+h}$ such that $\mathbf{v} \leqslant \mathbf{v}'$, it holds $f(\mathbf{v}) \leqslant f(\mathbf{v}')$, where $(v_1, \ldots, v_{l+h}) \leqslant (v'_1, \ldots, v'_{l+h})$ iff $v_i \leqslant v'_i$ for all i. We also assume that the computational cost of f and all fuzzy predicates p_i is bounded by a constant.

We call $R(\mathbf{x})$ the *head* and $\exists \mathbf{y}.f(R_1(\mathbf{z}_1), \ldots, R_l(\mathbf{z}_l), p_1(\mathbf{z}'_1), \ldots, p_h(\mathbf{z}'_h))$ the *body* of the rule. We assume that relations occurring in F do not occur in the head of rules (so, we do not allow that the fact relations occurring in F can be redefined by P). As usual in deductive databases, the relations in F are called *extensional* relations, while the others are *intensional* relations.

Example 3.6. Consider again Example 3.5. An excerpt of the domain ontology is described in Fig. 4 and partially encodes the web directory behind the car selling site `www.autos.com`. □

Semantically, the notion of Herbrand universe HU_{KB} and Herbrand base HB_{KB} are defined as usual as the set of individual constants occurring in KB and the set of ground atoms that can be formed using constants in HU_{KB} and atoms occurring in KB, respectively. An *interpretation* $\mathcal{I}$ maps every n-ary relation R to a partial function $R^{\mathcal{I}}\colon HU^n_{KB} \to [0,1]_m$ and every constant to an element of HU_{KB} such that $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ if $a \neq b$ (unique name assumption). Note that, since $R^{\mathcal{I}}$ may be a partial function, some tuples may not have a score. Alternatively, we may assume $R^{\mathcal{I}}$ to be a total function. We use the former formulation to distinguish the case where a tuple $\mathbf{c}$ may be retrieved, even though the score is 0, from the case where a tuple is not retrieved, since it does not satisfy the query. In particular, if a tuple does not belong to an extensional relation, then its score is assumed to be undefined, while if $R^{\mathcal{I}}$ is total, then the score of this tuple would be 0.

An interpretation $\mathcal{I}$ is a *model* of (or *satisfies*) a fact $\langle R(c_1, \ldots, c_n), s\rangle$, denoted $\mathcal{I} \models \langle R(c_1, \ldots, c_n), s\rangle$, iff $R^{\mathcal{I}}(c_1, \ldots, c_n) \geqslant s$ whenever $R^{\mathcal{I}}(c_1, \ldots, c_n)$ is defined.

An interpretation $\mathcal{I}$ is a *model* of a rule r of the form $R(\mathbf{x}) \leftarrow \exists \mathbf{y}.\phi(\mathbf{x}, \mathbf{y})$, where $\phi(\mathbf{x}, \mathbf{y}) = \exists \mathbf{y}.f(R_1(\mathbf{z}_1), \ldots, R_l(\mathbf{z}_l), p_1(\mathbf{z}'_1), \ldots, p_h(\mathbf{z}'_h))$, denoted $\mathcal{I} \models r$, iff for all $\mathbf{c} \in HU^n_{KB}$ such that $R^{\mathcal{I}}(\mathbf{c})$ is defined, the following holds (where $\phi^{\mathcal{I}}(\mathbf{c}, \mathbf{c}')$ is obtained from $\phi(\mathbf{c}, \mathbf{c}')$ by replacing every R_i by $R^{\mathcal{I}}_i$ and every constant c by $c^{\mathcal{I}}$):

$$R^{\mathcal{I}}(\mathbf{c}) \geqslant \sup_{\mathbf{c}' \in HU_{KB} \times \cdots \times HU_{KB},\ \phi^{\mathcal{I}}(\mathbf{c}, \mathbf{c}') \text{ is defined}} \phi^{\mathcal{I}}(\mathbf{c}, \mathbf{c}')\ .$$

$$\begin{array}{lll} \mathit{Pref1}(x,p) & \leftarrow \min(\mathit{Cars}(x), \mathit{hasPrice}(x,p), \mathit{ls}(10000, 14000)(p)) \; ; & (1) \\ \mathit{Pref2}(x,k) & \leftarrow \min(\mathit{Cars}(x), \mathit{hasKM}(x,k), \mathit{ls}(13000, 17000)(k)) \; ; & (2) \\ \mathit{Buy}(x,p,k) & \leftarrow 0.7 \cdot \mathit{Pref1}(x,p) + 0.3 \cdot \mathit{Pref2}(x,k) \; . & (3) \end{array}$$

Fig. 5. The car buying rules

We say $\mathcal{I}$ is a *model* of a knowledge base KB, denoted $\mathcal{I} \models KB$, iff $\mathcal{I}$ is a model of each expression $E \in F \cup P$. We say KB *entails* $R(\mathbf{c})$ to degree s, denoted $KB \models \langle R(\mathbf{c}), s \rangle$, iff for each model $\mathcal{I}$ of KB, it is true that $R^{\mathcal{I}}(\mathbf{c}) \geqslant s$ whenever $R^{\mathcal{I}}(\mathbf{c})$ is defined. The *greatest lower bound* of $R(\mathbf{c})$ relative to KB is $\mathit{glb}(KB, R(\mathbf{c})) = \sup\{s \mid KB \models \langle R(\mathbf{c}), s \rangle\}$.

Example 3.7. Consider again Example 3.6. Now, suppose that in buying a car, preferably we would like to pay around \$12000 and the car should have less than 15000 km. Of course, our constraints on price and kilometers are not crisp as we may still accept to some degree, e.g., a car's cost of \$12200 and with 16000 km. Hence, these constraints are rather *vague*. We model this by means of left-shoulder functions (see Fig. 2). We may model the vague constraint on the cost with $\mathit{ls}(10000, 14000)(x)$ dictating that we are definitely satisfied if the price is less than \$10000, but can pay up to \$14000 to a lesser degree of satisfaction. Similarly, we may model the vague constraint on the kilometers with $\mathit{ls}(13000, 17000)(x)$.[3] We also set some preference (weights) on these two vague constraints, say the weight 0.7 to the price constraint and 0.3 to the kilometers constraint, indicating that we give more priority to the price rather than to the car's kilometers. The rules encoding the above conditions are represented in Fig. 5. Rule (1) in Fig. 5 encodes the preference on the price. Here, $\mathit{ls}(10000, 14000)(p)$ is the function that given a price p returns the degree of truth provided by the left-shoulder function $\mathit{ls}(10000, 14000)(p)$ evaluated on the input p. Similarly, for rule (2). Rule (3) encodes the combination of the preferences by taking into account the weight given to each preference. The table below reports the instances of $\mathit{Buy}(x, p, k)$ together with their greatest lower bound.

ID	*PRICE*	*KM*	*s*
455	*12500*	10000	0.56
34	*12000*	15000	0.50
1812	*11000*	16000	0.60 .

□

The basic inference problem that we are interested in here is the top-k retrieval problem, which is formulated as follows.

Top-k Retrieval. Given a knowledge base KB, retrieve k tuples $(\mathbf{c}, s)$ that instantiate the query relation R with maximal scores (if k such tuples exist), and rank them in decreasing order relative to the score s, denoted

$$\mathit{ans}_k(KB, R) = \mathit{Top}_k\{(\mathbf{c}, s) \mid s = \mathit{glb}(KB, R(\mathbf{c}))\} \; .$$

[3] Recall that in our setting, all fuzzy membership functions provide a truth value in $[0, 1]_m$.

Example 3.8. It can be verified that the answer to the top-2 problem for Example 3.7 is

ID	*PRICE*	*KM*	*s*
1812	*11000*	16000	0.60
455	*12500*	10000	0.56 .

□

3.3 Description Logic Programs

Description Logic Programs [87, 134, 195, 225] are a combination of description logics with logic programming.[4] There is a large body of work on integrating rules and ontologies, which is a key requirement of the layered architecture of the Semantic Web. Significant research efforts focus on hybrid integrations of rules and ontologies, called *description logic programs* (or *dl-programs*), which are of the form $KB = (L, P)$, 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 especially [71, 72, 68, 69, 70]) or concepts and roles from L as unary resp. binary predicates in a tight integration (see especially [134, 223, 224, 168, 195, 196]). Roughly, in the loosely coupled approach, DL atoms may appear in rule bodies and act as queries to an underlying DL system, while in the tightly coupled approach the integration is more involved.

In parallel to these to approaches (loosely coupled vs. tightly coupled) there has been some works on the extension of these approaches towards the management of imperfect information: (i) under probability fall works such as [20, 21, 164, 169, 170]; (ii) under vagueness fall the works [166, 173, 174, 178, 255, 257, 261, 274]; while a combination of probability and vagueness in description logic programs can be found in the work (unique so far) [172].

References

1. Agarwal, S., Lamparter, S.: Smart: A semantic matchmaking portal for electronic markets. In: CEC 2005: Proceedings of the Seventh IEEE International Conference on E-Commerce Technology (CEC 2005), pp. 405–408. IEEE Computer Society Press, Washington (2005)
2. Alsinet, T., Godo, L.: Towards an automated deduction system for first-order possibilistic logic programming with fuzzy constants. International Journal of Intelligent Systems 17(9), 887–924 (2002)
3. Alsinet, T., Godo, L., Sandri, S.: On the semantics and automated deduction fo PLFC, a logic of possibilistic uncertainty and fuzzyness. In: Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI 1999) (1999)
4. Alsinet, T., Godo, L.G.L.: A complete calcultis for possibilistic logic programming with fuzzy propositional variables with fuzzy propositional variables. In: Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence (UAI 2000), pp. 1–10. Morgan Kaufmann, San Francisco (2000)
5. Amarger, S., Dubois, D., Prade, H.: Constraint propagation with imprecise conditional probabilities. In: Proceedings UAI 1991, pp. 26–34. Morgan Kaufmann, San Francisco (1991)

[4] The term Description Logic Programs was introduced for the specific approach of [87]. We use it here to address any kind of combination of description logics with logic programming.

6. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, Cambridge (2003)
7. Baldwin, J.F.: Evidential support of logic programming. Fuzzy Sets and Systems 24(1), 1–26 (1987)
8. Baldwin, J.F.: A theory of mass assignments for artificial intelligence. In: Driankov, D., L. Ralescu, A., Eklund, P.W. (eds.) IJCAI-WS 1991. LNCS, vol. 833, pp. 22–34. Springer, Heidelberg (1994)
9. Baldwin, J.F., Martin, T.P., Pilsworth, B.W.: Fril - Fuzzy and Evidential Reasoning in Artificial Intelligence. Research Studies Press Ltd (1995)
10. Baldwin, J.F., Martin, T.P., Pilsworth, B.W.: Applications of fuzzy computation: Knowledge based systems: Knowledge representation. In: Ruspini, E.H., Bonnissone, P., Pedrycz, W. (eds.) Handbook of Fuzzy Computing, IOP Publishing (1998)
11. Baral, C., Gelfond, M., Rushton, N.: Probabilistic reasoning with answer sets. In: Lifschitz, V., Niemelä, I. (eds.) LPNMR 2004. LNCS (LNAI), vol. 2923, pp. 21–33. Springer, Heidelberg (2003)
12. Bobillo, F., Delgado, M., Gómez-Romero, J.: A crisp representation for fuzzy $\mathcal{SHOIN}$ with fuzzy nominals and general concept inclusions. In: Proceedings of the 2th Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2006) (November 2006)
13. Bobillo, F., Straccia, U.: A fuzzy description logic with product t-norm. In: Proceedings of the IEEE International Conference on Fuzzy Systems (Fuzz-IEEE 2007), pp. 652–657. IEEE Computer Society Press, Los Alamitos (2007)
14. Bobillo, F., Straccia, U.: fuzzyDL: An expressive fuzzy description logic reasoner. In: 2008 International Conference on Fuzzy Systems (FUZZ 2008), pp. 923–930. IEEE Computer Society Press, Los Alamitos (2008)
15. Bobillo, F., Straccia, U.: Towards a crisp representation of fuzzy description logics under Łukasiewicz semantics. In: An, A., Matwin, S., Raś, Z.W., Ślęzak, D. (eds.) Foundations of Intelligent Systems. LNCS (LNAI), vol. 4994, pp. 309–318. Springer, Heidelberg (2008)
16. Bonatti, P., Tettamanzi, A.: Some complexity results on fuzzy description logics. In: Di Gesú, V., Masulli, F., Petrosino, A. (eds.) WILF 2003. LNCS (LNAI), vol. 2955. Springer, Heidelberg (2006)
17. Boole, G.: An Investigation of the Laws of Thought, on which are Founded the Mathematical Theories of Logic and Probabilities. Walton and Maberley, London (1854); (reprint: Dover Publications, New York, 1958)
18. Brachman, R.J., Levesque, H.J.: The tractability of subsumption in frame-based description languages. In: Proceedings of AAAI 1984, 4th Conference of the American Association for Artificial Intelligence, Austin, TX, pp. 34–37 (1984); [a] An extended version appears as [133]
19. Bueno, F., Cabeza, D., Carro, M., Hermenegildo, M., López-García, P., Puebla, G.: The Ciao prolog system. Reference manual. Technical Report CLIPS3/97.1, School of Computer Science, Technical University of Madrid (UPM) (1997), `http://www.cliplab.org/Software/Ciao/`
20. Calì, A., Lukasiewicz, T.: Tightly integrated probabilistic description logic programs for the semantic web. In: Dahl, V., Niemelä, I. (eds.) ICLP 2007. LNCS, vol. 4670, pp. 428–429. Springer, Heidelberg (2007)
21. Calì, A., Lukasiewicz, T., Predoiu, L., Stuckenschmidt, H.: Tightly integrated probabilistic description logic programs for representting ontology mappings. In: Hartmann, S., Kern-Isberner, G. (eds.) FoIKS 2008. LNCS, vol. 4932, pp. 178–198. Springer, Heidelberg (2008)
22. Calmet, J., Lu, J., Rodriguez, M., Schü, J.: Signed formula logic programming: operational semantics and applications. In: Michalewicz, M., Raś, Z.W. (eds.) ISMIS 1996. LNCS, vol. 1079, pp. 202–211. Springer, Berlin (1996)

23. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: DL-Lite: Tractable description logics for ontologies. In: Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI 2005). AAAI Press, Menlo Park (2005)
24. Cao, T.H.: Annotated fuzzy logic programs. Fuzzy Sets and Systems 113(2), 277–298 (2000)
25. Carlos, C., Guillermo, S., Teresa, A., Lluis, G.: A logic programming framework for possibilistic argumentation with vague knowledge. In: Proceedings of the 20th Annual Conference on Uncertainty in Artificial Intelligence (UAI 2004), Arlington, Virginia, pp. 76–84. AUAI Press (2004)
26. Chang, K.C.-C., won Hwang, S.: Minimal probing: Supporting expensive predicates for top-k queries. In: SIGMOD Conference (2002)
27. Chortaras, A., Stamou, G.B., Stafylopatis, A.: Integrated query answering with weighted fuzzy rules. In: Mellouli, K. (ed.) ECSQARU 2007. LNCS (LNAI), vol. 4724, pp. 767–778. Springer, Heidelberg (2007)
28. da Costa, P.C.G.: Bayesian semantics for the Semantic Web. PhD thesis, George Mason University, Fairfax, VA, USA (2005)
29. da Costa, P.C.G., Laskey, K.B.: PR-OWL: A framework for probabilistic ontologies. In: Proceedings FOIS 2006, pp. 237–249. IOS Press, Amsterdam (2006)
30. da Costa, P.C.G., Laskey, K.B., Laskey, K.J.: PR-OWL: A Bayesian ontology language for the Semantic Web. In: Proceedings URSW 2005, pp. 23–33 (2005)
31. Damásio, C., Medina, J., Ojeda-Aciego, M.: Termination of logic programs with imperfect information: applications and query procedure. Journal of Applied Logic (to appear 2006)
32. Damásio, C., Medina, M., Ojeda-Aciego, J.: A tabulation procedure for first-order residuated logic programs. In: Proceedings of the 11th International Conference on Information Processing and Managment of Uncertainty in Knowledge-Based Systems (IPMU 2006) (2006)
33. Damásio, C., Medina, M., Ojeda-Aciego, J.: Termination of logic programs with imperfect information: applications and query procedure. Journal of Applied Logic 7(5), 435–458 (2007)
34. Damásio, C.V., Medina, J., Ojeda Aciego, M.: Sorted multi-adjoint logic programs: Termination results and applications. In: Alferes, J.J., Leite, J.A. (eds.) JELIA 2004. LNCS (LNAI), vol. 3229, pp. 252–265. Springer, Heidelberg (2004)
35. Damásio, C.V., Medina, J., Ojeda Aciego, M.: A tabulation proof procedure for residuated logic programming. In: Proceedings of the 6th European Conference on Artificial Intelligence (ECAI 2004) (2004)
36. Damásio, C.V., Medina, J., Ojeda Aciego, M.: Termination results for sorted multi-adjoint logic programs. In: Proceedings of the 10th International Conference on Information Processing and Managment of Uncertainty in Knowledge-Based Systems (IPMU 2004), pp. 1879–1886 (2004)
37. Damásio, C.V., Pereira, L.M.: A survey of paraconsistent semantics for logic programs. In: Gabbay, D., Smets, P. (eds.) Handbook of Defeasible Reasoning and Uncertainty Management Systems, pp. 241–320. Kluwer Academic Publishers, Dordrecht (1998)
38. Damásio, C.V., Pereira, L.M.: Antitonic logic programs. In: Eiter, T., Faber, W., Truszczyński, M. (eds.) LPNMR 2001. LNCS (LNAI), vol. 2173. Springer, Heidelberg (2001)
39. Damásio, C.V., Pereira, L.M.: Monotonic and residuated logic programs. In: Benferhat, S., Besnard, P. (eds.) ECSQARU 2001. LNCS (LNAI), vol. 2143, pp. 748–759. Springer, Heidelberg (2001)
40. Damásio, C.V., Pereira, L.M.: Hybrid probabilistic logic programs as residuated logic programs. Studia Logica 72(1), 113–138 (2002)

41. Damásio, C.V., Pereira, L.M.: Sorted monotonic logic programs and their embeddings. In: Proceedings of the 10th International Conference on Information Processing and Managment of Uncertainty in Knowledge-Based Systems (IPMU 2004), pp. 807–814 (2004)
42. d'Aquin, M., Lieber, J., Napoli, A.: Towards a semantic portal for oncology using a description logic with fuzzy concrete domains. In: Sanchez, E. (ed.) Fuzzy Logic and the Semantic Web, pp. 379–393. Elsevier, Amsterdam (2006)
43. Dazhou Kang, J.L.: Baowen Xu and Y. Li. Reasoning for a fuzzy description logic with comparison expressions. In: Proceeedings of the International Workshop on Description Logics (DL 2006). CEUR (2006)
44. Dekhtyar, A., Dekhtyar, M.I.: Possible worlds semantics for probabilistic logic programs. In: Demoen, B., Lifschitz, V. (eds.) ICLP 2004. LNCS, vol. 3132, pp. 137–148. Springer, Heidelberg (2004)
45. Dekhtyar, A., Dekhtyar, M.I.: Revisiting the semantics of interval probabilistic logic programs. In: Baral, C., Greco, G., Leone, N., Terracina, G. (eds.) LPNMR 2005. LNCS (LNAI), vol. 3662, pp. 330–342. Springer, Heidelberg (2005)
46. Dekhtyar, A., Dekhtyar, M.I., Subrahmanian, V.S.: Temporal probabilistic logic programs. In: Schreye, D.D. (ed.) Logic Programming: The 1999 International Conference, Las Cruces, New Mexico, USA, November 29 - December 4, 1999, pp. 109–123 (1999)
47. Dekhtyar, A., Subrahmanian, V.: Hybrid probabilistic programs. Journal of Logic Programming 43(3), 187–250 (2000)
48. Dekhtyar, M.I., Dekhtyar, A., Subrahmanian, V.S.: Hybrid probabilistic programs: Algorithms and complexity. In: Laskey, K.B., Prade, H. (eds.) Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI 1999), January 30, 1999, pp. 160–169. Morgan Kaufmann, San Francisco (1999)
49. Denecker, M., Marek, V., Truszczyński, M.: Approximations, stable operators, well-founded fixpoints and applications in nonmonotonic reasoning. In: Minker, J. (ed.) Logic-Based Artifical Intelligence, pp. 127–144. Kluwer Academic Publishers, Dordrecht (2000)
50. Denecker, M., Marek, V.W., Truszczyński, M.: Uniform semantic treatment of default and autoepistemic logics. In: Cohn, A., Giunchiglia, F., Selman, B. (eds.) Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning, pp. 74–84. Morgan Kaufmann, San Francisco (2000)
51. Denecker, M., Marek, V.W., Truszczyński, M.: Ultimate approximations. Technical Report CW 320, Katholieke Iniversiteit Leuven (September 2001)
52. Denecker, M., Marek, V.W., Truszczyński, M.: Ultimate approximations in nonmonotonic knowledge representation systems. In: Fensel, D., Giunchiglia, F., McGuinness, D., Williams, M. (eds.) Principles of Knowledge Representation and Reasoning: Proceedings of the 8th International Conference, pp. 177–188. Morgan Kaufmann, San Francisco (2002)
53. Denecker, M., Pelov, N., Bruynooghe, M.: Ultimate well-founded and stable semantics for logic programs with aggregates. In: Codognet, P. (ed.) ICLP 2001. LNCS, vol. 2237. Springer, Heidelberg (2001)
54. Ding, Z., Peng, Y.: A probabilistic extension to ontology language OWL. In: Proceedings of the 37th Hawaii International Conference on Systems Sciences. IEEE, Los Alamitos (2004)
55. Ding, Z., Peng, Y., Pan, R.: BayesOWL: Uncertainty modeling in Semantic Web ontologies. In: Ma, Z. (ed.) Soft Computing in Ontologies and Semantic Web. Studies in Fuzziness and Soft Computing, vol. 204. Springer, Heidelberg (2006)
56. Dubois, D., Foulloy, L., Mauris, G., Prade, H.: Probability-possibility transformations, triangular fuzzy sets, and probabilistic inequalities. Reliable Computing 10(4), 273–297 (2004)
57. Dubois, D., Lang, J., Prade, H.: Towards possibilistic logic programming. In: Proc. of the 8th Int. Conf. on Logic Programming (ICLP 1991), pp. 581–595. MIT Press, Cambridge (1991)

58. Dubois, D., Mengin, J., Prade, H.: Possibilistic uncertainty and fuzzy features in description logic. a preliminary discussion. In: Sanchez, E. (ed.) Capturing Intelligence: Fuzzy Logic and the Semantic Web. Elsevier, Amsterdam (2006)
59. Dubois, D., Prade, H.: Possibilistic logic. In: Gabbay, D.M., Hogger, C.J. (eds.) Handbook of Logic in Artificial Intelligence, vol. 3, pp. 439–513. Clarendon Press, Oxford (1986)
60. Dubois, D., Prade, H.: On fuzzy syllogisms. Computational Intelligence 4(2), 171–179 (1988)
61. Dubois, D., Prade, H.: When upper probabilities are possibility measures. Fuzzy Sets and Systems 49, 65–74 (1992)
62. Dubois, D., Prade, H.: Can we enforce full compositionality in uncertainty calculi? In: Proc. of the 12th Nat. Conf. on Artificial Intelligence (AAAI 1994), pp. 149–154. AAAI Press, Menlo Park (1994)
63. Dubois, D., Prade, H.: Possibility theory, probability theory and multiple-valued logics: A clarification. Annals of Mathematics and Artificial Intelligence 32(1-4), 35–66 (2001)
64. Dubois, D., Prade, H., Godo, L., de Màntaras, R.L.: Qualitative reasoning with imprecise probabilities. J. Intell. Inf. Syst. 2, 319–363 (1993)
65. Dubois, D., Prade, H., Touscas, J.-M.: Inference with imprecise numerical quantifiers. In: Ras, Z.W., Zemankova, M. (eds.) Intelligent Systems, ch. 3, pp. 53–72. Ellis Horwood (1990)
66. Dürig, M., Studer, T.: Probabilistic ABox reasoning: Preliminary results. In: Proceedings DL 2005, pp. 104–111 (2005)
67. Ebrahim, R.: Fuzzy logic programming. Fuzzy Sets and Systems 117(2), 215–230 (2001)
68. Eiter, T., Ianni, G., Schindlauer, R., Tompits, H.: Nonmonotonic description logic programs: Implementation and experiments. In: Baader, F., Voronkov, A. (eds.) LPAR 2004. LNCS (LNAI), vol. 3452, pp. 511–527. Springer, Heidelberg (2005)
69. Eiter, T., Ianni, G., Schindlauer, R., Tompits, H.: A uniform integration of higher-order reasoning and external evaluations in answer-set programming. In: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI 2005), pp. 90–96. Professional Book Center (2005)
70. Eiter, T., Ianni, G., Schindlauer, R., Tompits, H.: Effective integration of declarative rules with external evaluations for semantic-web reasoning. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 273–287. Springer, Heidelberg (2006)
71. Eiter, T., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Combining answer set programming with description logics for the semantic web. In: Proceedings of the 9th International Conference on Principles of Knowledge Representation and Reasoning (KR 2004). AAAI Press, Menlo Park (2004)
72. Eiter, T., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Well-founded semantics for description logic programs in the semantic web. In: Antoniou, G., Boley, H. (eds.) RuleML 2004. LNCS, vol. 3323, pp. 81–97. Springer, Heidelberg (2004)
73. Fagin, R.: Combining fuzzy information: an overview. SIGMOD Rec. 31(2), 109–118 (2002)
74. Fagin, R., Halpern, J.Y., Megiddo, N.: A logic for reasoning about probabilities. Inf. Comput. 87, 78–128 (1990)
75. Fitting, M.: A Kripke-Kleene-semantics for general logic programs. Journal of Logic Programming 2, 295–312 (1985)
76. Fitting, M.: Pseudo-Boolean valued Prolog. Studia Logica XLVII(2), 85–91 (1987)
77. Fitting, M.: Bilattices and the semantics of logic programming. Journal of Logic Programming 11, 91–116 (1991)
78. Fitting, M.C.: The family of stable models. Journal of Logic Programming 17, 197–225 (1993)

79. Fitting, M.C.: Bilattices are nice things. In: Conference on Self-Reference, Copenhagen, Denmark (2002)
80. Fitting, M.C.: Fixpoint semantics for logic programming - a survey. Theoretical Computer Science 21(3), 25–51 (2002)
81. Frisch, A.M., Haddawy, P.: Anytime deduction for probabilistic logic. Artif. Intell. 69(1–2), 93–122 (1994)
82. Fuhr, N.: Probabilistic Datalog: Implementing logical information retrieval for advanced applications. Journal of the American Society for Information Science 51(2), 95–110 (2000)
83. Fukushige, Y.: Representing probabilistic knowledge in the Semantic Web. In: Proceedings of the W3C Workshop on Semantic Web for Life Sciences (2004)
84. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Generation Computing 9(3/4), 365–386 (1991)
85. Ginsberg, M.L.: Multi-valued logics: a uniform approach to reasoning in artificial intelligence. Computational Intelligence 4, 265–316 (1988)
86. 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: Flesca, S., Greco, S., Leone, N., Ianni, G. (eds.) JELIA 2002. LNCS (LNAI), vol. 2424, pp. 86–97. Springer, Heidelberg (2002)
87. Grosof, B.N., Horrocks, I., Volz, R., Decker, S.: Description logic programs: combining logic programs with description logic. In: Proceedings of the twelfth international conference on World Wide Web, pp. 48–57. ACM Press, New York (2003)
88. Guller, D.: Procedural semantics for fuzzy disjunctive programs. In: Baaz, M., Voronkov, A. (eds.) LPAR 2002. LNCS (LNAI), vol. 2514, pp. 247–261. Springer, Heidelberg (2002)
89. Guller, D.: Semantics for fuzzy disjunctive programs with weak similarity. In: Abraham, A., Köppen, M. (eds.) Hybrid Information Systems, Proceedings of First International Workshop on Hybrid Intelligent Systems, Adelaide, Australia, December 11-12, 2001. Advances in Soft Computing, pp. 285–299. Physica-Verlag (2002)
90. Hadjali, A., Kaci, S., Prade, H.: Database preference queries — a possibilistic logic approach with symbolic priorities. In: Hartmann, S., Kern-Isberner, G. (eds.) FoIKS 2008. LNCS, vol. 4932. Springer, Heidelberg (2008)
91. Hähnle, R.: Uniform notation of tableaux rules for multiple-valued logics. In: Proc. International Symposium on Multiple-Valued Logic, Victoria, pp. 238–245. IEEE Press, Los Alamitos (1991)
92. Hähnle, R.: Many-valued logics and mixed integer programming. Annals of Mathematics and Artificial Intelligence 3,4(12), 231–264 (1994)
93. Hähnle, R.: Advanced many-valued logics. In: Gabbay, D.M., Guenthner, F. (eds.) Handbook of Philosophical Logic, 2nd edn. Kluwer Academic Publishers, Dordrecht (2001)
94. Hailperin, T.: Sentential Probability Logic: Origins, Development, Current Status, and Technical Applications. Associated University Presses, London, UK (1996)
95. Hájek, P.: Metamathematics of Fuzzy Logic. Kluwer Academic Publishers, Dordrecht (1998)
96. Hájek, P.: Making fuzzy description logics more expressive. Fuzzy Sets and Systems 154(1), 1–15 (2005)
97. Hájek, P.: What does mathematical fuzzy logic offer to description logic. In: Sanchez, E. (ed.) Fuzzy Logic and the Semantic Web, Capturing Intelligence, ch. 5, pp. 91–100. Elsevier, Amsterdam (2006)
98. Hansen, P., Jaumard, B., Nguetsé, G.-B.D., de Aragão, M.P.: Models and algorithms for probabilistic and Bayesian logic. In: Proceedings IJCAI 1995, pp. 1862–1868. Morgan Kaufmann, San Francisco (1995)
99. Heinsohn, J.: Probabilistic description logics. In: Proceedings UAI 1994, pp. 311–318. Morgan Kaufmann, San Francisco (1994)

100. Hinde, C.: Fuzzy prolog. International Journal Man.-Machine Stud. (24), 569–595 (1986)
101. Holi, M., Hyvönen, E.: A method for modeling uncertainty in Semantic Web taxonomies. In: Proceedings WWW 2004, pp. 296–297. ACM Press, New York (2004)
102. Holi, M., Hyvönen, E.: Modeling degrees of conceptual overlap in Semantic Web ontologies. In: Proceedings URSW 2005, pp. 98–99 (2005)
103. Hölldobler, S., Khang, T.D., Störr, H.-P.: A fuzzy description logic with hedges as concept modifiers. In: Phuong, N.H., Nguyen, H.T., Ho, N.C., Santiprabhob, P. (eds.) Proceedings InTech/VJFuzzy', Hanoi, Vietnam, Institute of Information Technology, Vietnam Center for Natural Science and Technology, Science and Technics Publishing House, Hanoi, Vietnam, pp. 25–34 (2002)
104. Hölldobler, S., Nga, N.H., Khang, T.D.: The fuzzy description logic ALC_{FH}. In: Proceeedings of the International Workshop on Description Logics (DL 2005) (2005)
105. Hölldobler, S., Störr, H.-P., Khang, T.D.: The fuzzy description logic ALC_{FH} with hedge algebras as concept modifiers. Journal of Advanced Computational Intelligence (2003)
106. Hölldobler, S., Störr, H.-P., Khang, T.D.: The subsumption problem of the fuzzy description logic ALC_{FH}. In: Proceedings of the 10th International Conference on Information Processing and Managment of Uncertainty in Knowledge-Based Systems (IPMU 2004) (2004)
107. Hollunder, B.: An alternative proof method for possibilistic logic and its application to terminological logics. International Journal of Approximate Reasoning 12, 85–109 (1995)
108. Horrocks, I., Sattler, U.: A tableaux decision procedure for $\mathcal{SHOIQ}$. In: Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005) (2005)
109. Hung, E., Deng, Y., Subrahmanian, V.S.: TOSS: An extension of TAX with ontologies and similarity queries. In: Proceedings ACM SIGMOD 2004, pp. 719–730. ACM Press, New York (2004)
110. Ilyas, I.F., Aref, W.G., Elmagarmid, A.K.: Supporting top-k join queries in relational databases. In: Proceedings of 29th International Conference on Very Large Data Bases (VLDB 2003), pp. 754–765 (2003)
111. Ishizuka, M., Kanai, N.: Prolog-ELF: incorporating fuzzy logic. In: Proceedings of the 9th International Joint Conference on Artificial Intelligence (IJCAI 1985), Los Angeles, CA, pp. 701–703 (1985)
112. Jaeger, M.: Probabilistic reasoning in terminological logics. In: Proceedings KR 1994, pp. 305–316. Morgan Kaufmann, San Francisco (1994)
113. Jaeger, M.: Probabilistic role models and the guarded fragment. In: Proceedings IPMU-2004, pp. 235–242 (2004); Extended version in International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 14(1), 43–60 (2006)
114. Jaumard, B., Hansen, P., de Aragão, M.P.: Column generation methods for probabilistic logic. ORSA J. Comput. 3, 135–147 (1991)
115. Kaplunova, A., Möller, R.: Probabilistic LCS in a P-Classic Implementation. Technical report, Institute for Software Systems (STS), Hamburg University of Technology, Germany (2007), `http://www.sts.tu-harburg.de/tech-reports/papers.html`
116. Kern-Isberner, G., Lukasiewicz, T.: Combining probabilistic logic programming with the power of maximum entropy. Artificial Intelligence 157(1-2), 139–202 (2004)
117. Kersting, K., Raedt, L.D.: Bayesian logic programs. In: Cussens, J., Frisch, A.M. (eds.) ILP Work-in-progress reports, 10th International Conference on Inductive Logic Programming, CEUR Workshop Proceedings. CEUR-WS.org (2000)
118. Kersting, K., Raedt, L.D.: Bayesian logic programming: Theory and tools. In: Getoor, L., Taskar, B. (eds.) An Introduction to Statistical Relational Learning. MIT Press, Cambridge (2005)
119. Khamsi, M., Misane, D.: Disjunctive signed logic programs. Fundamenta Informaticae 32, 349–357 (1996)

120. Khamsi, M., Misane, D.: Fixed point theorems in logic programming. Annals of Mathematics and Artificial Intelligence 21, 231–243 (1997)
121. Kifer, M., Li, A.: On the semantics of rule-based expert systems with uncertainty. In: Gyssens, M., Van Gucht, D., Paredaens, J. (eds.) ICDT 1988. LNCS, vol. 326, pp. 102–117. Springer, Heidelberg (1988)
122. Kifer, M., Subrahmanian, V.: Theory of generalized annotated logic programming and its applications. Journal of Logic Programming 12, 335–367 (1992)
123. Klawonn, F., Kruse, R.: A Łukasiewicz logic based Prolog. Mathware & Soft Computing 1(1), 5–29 (1994)
124. Klir, G.J., Yuan, B.: Fuzzy sets and fuzzy logic: theory and applications. Prentice-Hall, Inc., Upper Saddle River (1995)
125. Koller, D., Levy, A., Pfeffer, A.: P-Classic: A tractable probablistic description logic. In: Proceedings AAAI 1997, pp. 390–397. AAAI Press/MIT Press (1997)
126. Kulmann, P., Sandri, S.: An annotaded logic theorem prover for an extended possibilistic logic. Fuzzy Sets and Systems 144, 67–91 (2004)
127. Lakshmanan, L.: An epistemic foundation for logic programming with uncertainty. In: Thiagarajan, P.S. (ed.) FSTTCS 1994. LNCS, vol. 880, pp. 89–100. Springer, Heidelberg (1994)
128. Lakshmanan, L.V., Sadri, F.: Uncertain deductive databases: a hybrid approach. Information Systems 22(8), 483–508 (1997)
129. Lakshmanan, L.V., Shiri, N.: Probabilistic deductive databases. In: Int'l Logic Programming Symposium, pp. 254–268 (1994)
130. Lakshmanan, L.V., Shiri, N.: A parametric approach to deductive databases with uncertainty. IEEE Transactions on Knowledge and Data Engineering 13(4), 554–570 (2001)
131. Lakshmanan, L.V.S., Sadri, F.: On a theory of probabilistic deductive databases. Theory and Practice of Logic Programming 1(1), 5–42 (2001)
132. Lehmann, D.: Another perspective on default reasoning. Ann. Math. Artif. Intell. 15(1), 61–82 (1995)
133. Levesque, H.J., Brachman, R.J.: Expressiveness and tractability in knowledge representation and reasoning. Computational Intelligence 3, 78–93 (1987)
134. Levy, A.Y., Rousset, M.-C.: Combining horn rules and description logics in CARIN. Artificial Intelligence 104, 165–209 (1998)
135. Li, C., Chang, K.C.-C., Ilyas, I.F., Song, S.: RankSQL: query algebra and optimization for relational top-k queries. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data (SIGMOD 2005), pp. 131–142. ACM Press, New York (2005)
136. Li, R., Wen, K., Lu, Z., Sun, X., Wang, Z.: An improved semantic search model based on hybrid fuzzy description logic. In: Workshop on Frontier of Computer Science and Technology (FCST 2006), pp. 139–146. IEEE Press, Los Alamitos (2006)
137. Li, Y., Xu, B., Lu, J., Kang, D., Wang, P.: Extended fuzzy description logic ALCN. In: Khosla, R., Howlett, R.J., Jain, L.C. (eds.) KES 2005. LNCS (LNAI), vol. 3684, pp. 896–902. Springer, Heidelberg (2005)
138. Li, Y., Xu, B., Lu, J., Kang, D., Wang, P.: A family of extended fuzzy description logics. In: 29th Annual International Computer Software and Applications Conference (COMPSAC 2005), pp. 221–226. IEEE Computer Society Press, Los Alamitos (2005)
139. Liau, C.-J., Yao, Y.Y.: Information retrieval by possibilistic reasoning. In: Mayr, H.C., Lazanský, J., Quirchmayr, G., Vogel, P. (eds.) DEXA 2001. LNCS, vol. 2113, pp. 52–61. Springer, Heidelberg (2001)
140. Liu, O., Tian, Q., Ma, J.: A fuzzy description logic approach to model management in R&D project selection. In: Proceedings of the 8th Pacific Asia Conference on Information Systems (PACIS 2004) (2004)

141. Lloyd, J.W.: Foundations of Logic Programming. Springer, Heidelberg (1987)
142. Loyer, Y., Straccia, U.: Uncertainty and partial non-uniform assumptions in parametric deductive databases. In: Flesca, S., Greco, S., Leone, N., Ianni, G. (eds.) JELIA 2002. LNCS (LNAI), vol. 2424, pp. 271–282. Springer, Heidelberg (2002)
143. Loyer, Y., Straccia, U.: The well-founded semantics in normal logic programs with uncertainty. In: Hu, Z., Rodríguez-Artalejo, M. (eds.) FLOPS 2002. LNCS, vol. 2441, pp. 152–166. Springer, Heidelberg (2002)
144. Loyer, Y., Straccia, U.: The approximate well-founded semantics for logic programs with uncertainty. In: Rovan, B., Vojtáš, P. (eds.) MFCS 2003. LNCS, vol. 2747, pp. 541–550. Springer, Heidelberg (2003)
145. Loyer, Y., Straccia, U.: Default knowledge in logic programs with uncertainty. In: Palamidessi, C. (ed.) ICLP 2003. LNCS, vol. 2916, pp. 466–480. Springer, Heidelberg (2003)
146. Loyer, Y., Straccia, U.: Epistemic foundation of the well-founded semantics over bilattices. In: Fiala, J., Koubek, V., Kratochvíl, J. (eds.) MFCS 2004. LNCS, vol. 3153, pp. 513–524. Springer, Heidelberg (2004)
147. Loyer, Y., Straccia, U.: Any-world assumptions in logic programming. Theoretical Computer Science 342(2-3), 351–381 (2005)
148. Loyer, Y., Straccia, U.: Epistemic foundation of stable model semantics. Journal of Theory and Practice of Logic Programming 6, 355–393 (2006)
149. Lu, J., Li, Y., Zhou, B., Kang, D., Zhang, Y.: Distributed reasoning with fuzzy description logics. In: 7th International International Conference on Computational Science, pp. 196–203 (2007)
150. Lu, J.J.: Logic programming with signs and annotations. Journal of Logic and Computation 6(6), 755–778 (1996)
151. Lu, J.J., Calmet, J., Schü, J.: Computing multiple-valued logic programs. Mathware % Soft Computing 2(4), 129–153 (1997)
152. Lukasiewicz, T.: Many-valued first-order logics with probabilistic semantics. In: Gottlob, G., Grandjean, E., Seyr, K. (eds.) CSL 1998. LNCS, vol. 1584, pp. 415–429. Springer, Heidelberg (1999)
153. Lukasiewicz, T.: Probabilistic logic programming. In: Proc. of the 13th European Conf. on Artificial Intelligence (ECAI 1998), Brighton, England, August 1998, pp. 388–392 (1998)
154. Lukasiewicz, T.: Local probabilistic deduction from taxonomic and probabilistic knowledge-bases over conjunctive events. Int. J. Approx. Reasoning 21(1), 23–61 (1999)
155. Lukasiewicz, T.: Many-valued disjunctive logic programs with probabilistic semantics. In: Gelfond, M., Leone, N., Pfeifer, G. (eds.) LPNMR 1999. LNCS (LNAI), vol. 1730, pp. 277–289. Springer, Heidelberg (1999)
156. Lukasiewicz, T.: Probabilistic and truth-functional many-valued logic programming. In: The IEEE International Symposium on Multiple-Valued Logic, pp. 236–241 (1999)
157. Lukasiewicz, T.: Probabilistic deduction with conditional constraints over basic events. J. Artif. Intell. Res. 10, 199–241 (1999)
158. Lukasiewicz, T.: Fixpoint characterizations for many-valued disjunctive logic programs with probabilistic semantics. In: Eiter, T., Faber, W., Truszczyński, M. (eds.) LPNMR 2001. LNCS (LNAI), vol. 2173, pp. 336–350. Springer, Heidelberg (2001)
159. Lukasiewicz, T.: Probabilistic logic programming under inheritance with overriding. In: Proceedings UAI 2001, pp. 329–336. Morgan Kaufmann, San Francisco (2001)
160. Lukasiewicz, T.: Probabilistic logic programming under inheritance with overriding. In: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence (UAI 2001), pp. 329–336. Morgan Kaufmann Publishers Inc., San Francisco (2001)
161. Lukasiewicz, T.: Probabilistic logic programming with conditional constraints. ACM Trans. Comput. Log. 2(3), 289–339 (2001)

162. Lukasiewicz, T.: Probabilistic logic programming with conditional constraints. ACM Transactions on Computational Logic 2(3), 289–339 (2001)
163. Lukasiewicz, T.: Probabilistic default reasoning with conditional constraints. Ann. Math. Artif. Intell. 34(1–3), 35–88 (2002)
164. Lukasiewicz, T.: Probabilistic description logic programs. In: Godo, L. (ed.) ECSQARU 2005. LNCS (LNAI), vol. 3571. Springer, Heidelberg (2005)
165. Lukasiewicz, T.: Weak nonmonotonic probabilistic logics. Artif. Intell. 168(1–2), 119–161 (2005)
166. Lukasiewicz, T.: Fuzzy description logic programs under the answer set semantics for the semanticweb. In: Second International Conference on Rules and Rule Markup Languages for the Semantic Web (RuleML 2006), pp. 89–96. IEEE Computer Society Press, Los Alamitos (2006)
167. Lukasiewicz, T.: Probabilistic description logics for the Semantic Web. Technical Report INFSYS RR-1843-06-04, Institut für Informationssysteme, Technische Universität Wien (submitted for journal publication) (June 2006)
168. Lukasiewicz, T.: A novel combination of answer set programming with description logics for the semantic web. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 384–398. Springer, Heidelberg (2007)
169. Lukasiewicz, T.: Probabilistic description logic programs. International Journal of Approximate Reasoning 42(2), 288–307 (2007)
170. Lukasiewicz, T.: Tractable probabilistic description logic programs. In: Prade, H., Subrahmanian, V.S. (eds.) SUM 2007. LNCS (LNAI), vol. 4772, pp. 143–156. Springer, Heidelberg (2007)
171. Lukasiewicz, T.: Expressive probabilistic description logics. Artificial Intelligence 172(6-7), 852–883 (2008)
172. Lukasiewicz, T., Straccia, U.: Description logic programs under probabilistic uncertainty and fuzzy vagueness. In: Mellouli, K. (ed.) ECSQARU 2007. LNCS (LNAI), vol. 4724, pp. 187–198. Springer, Heidelberg (2007)
173. Lukasiewicz, T., Straccia, U.: Tightly integrated fuzzy description logic programs under the answer semantics for the semantic web. In: Marchiori, M., Pan, J.Z. (eds.) RR 2007. LNCS, vol. 4524, pp. 289–298. Springer, Heidelberg (2007)
174. Lukasiewicz, T., Straccia, U.: Top-k retrieval in description logic programs under vagueness for the semantic web. In: Prade, H., Subrahmanian, V.S. (eds.) SUM 2007. LNCS (LNAI), vol. 4772, pp. 16–30. Springer, Heidelberg (2007)
175. Lukasiewicz, T., Straccia, U.: Uncertainty and vagueness in description logic programs for the semantic web. INFSYS RESEARCH REPORT 1843-07-02, INSTITUT FüR INFORMATIONSSYSTEME ARBEITSBEREICH WISSENSBASIERTE SYSTEME, Technische Universität Wien (2007)
176. Lutz, C.: Description logics with concrete domains—a survey. In: Advances in Modal Logics, vol. 4, King's College Publications (2003)
177. Magrez, P., Smets, P.: Fuzzy modus ponens: a new model suitable for applications in knowledge-based systems. International Journal of Intelligent Systems 4, 181–200 (1989)
178. Mailis, T.P., Stoilos, G., Stamou, G.B.: Expressive reasoning with horn rules and fuzzy description logics. In: Marchiori, M., Pan, J.Z., d Marie, C.S. (eds.) RR 2007. LNCS, vol. 4524, pp. 43–57. Springer, Heidelberg (2007)
179. Majkic, Z.: Coalgebraic semantics for logic programs. In: 18th Workshop on (Constraint) Logic Programming ((W(C)LP 2005), Ulm, Germany (2004)
180. Majkic, Z.: Many-valued intuitionistic implication and inference closure in a bilattice-based logic. In: 35th International Symposium on Multiple-Valued Logic (ISMVL 2005), pp. 214–220 (2005)

181. Majkic, Z.: Truth and knowledge fixpoint semantics for many-valued logic programming. In: 19th Workshop on (Constraint) Logic Programming ((W(C)LP 2005), Ulm, Germany, pp. 76–87 (2005)
182. Mantay, T., Möller, R., Kaplunova, A.: Computing Probabilistic Least Common Subsumers in Description Logics. In: Burgard, W., Christaller, T., Cremers, A.B. (eds.) KI 1999. LNCS (LNAI), vol. 1701, pp. 89–100. Springer, Heidelberg (1999)
183. Marek, V.W., Truszczyński, M.: Logic programming with costs. Technical report, University of Kentucky (2000), `ftp://al.cs.engr.uky.edu/cs/manuscripts/lp-costs.ps`
184. Martin, T.P.: Soft computing, logic programming and the semantic web. In: Proceedings of the 10th International Conference on Information Processing and Managment of Uncertainty in Knowledge-Based Systems (IPMU 2004), pp. 815–822 (2004)
185. Martin, T.P., Baldwin, J.F., Pilsworth, B.W.: The implementation of FProlog –a fuzzy prolog interpreter. Fuzzy Sets Syst. 23(1), 119–129 (1987)
186. Mateis, C.: Extending Disjunctive Logic Programming by t-norms. In: Gelfond, M., Leone, N., Pfeifer, G. (eds.) LPNMR 1999. LNCS (LNAI), vol. 1730, pp. 290–304. Springer, Heidelberg (1999)
187. Mateis, C.: Quantitative disjunctive logic programming: Semantics and computation. AI Communications 13, 225–248 (2000)
188. Medina, J., Ojeda-Aciego, M.: Multi-adjoint logic programming. In: Proceedings of the 10th International Conference on Information Processing and Managment of Uncertainty in Knowledge-Based Systems (IPMU 2004), pp. 823–830 (2004)
189. Medina, J., Ojeda-Aciego, M., Vojtáš, P.: Multi-adjoint Logic Programming with Continuous Semantics. In: Eiter, T., Faber, W., Truszczyński, M. (eds.) LPNMR 2001. LNCS (LNAI), vol. 2173, pp. 351–364. Springer, Heidelberg (2001)
190. Medina, J., Ojeda-Aciego, M., Vojtáš, P.: A procedural semantics for multi-adjoint logic programming. In: Proceedings of the10th Portuguese Conference on Artificial Intelligence on Progress in Artificial Intelligence, Knowledge Extraction, Multi-agent Systems, Logic Programming and Constraint Solving, pp. 290–297. Springer, Heidelberg (2001)
191. Medina, J., Ojeda-Aciego, M., Vojtáš, P.: Similarity-based unification: a multi-adjoint approach. Fuzzy sets and systems 1(146), 43–62 (2004)
192. Meghini, C., Sebastiani, F., Straccia, U.: A model of multimedia information retrieval. Journal of the ACM 48(5), 909–970 (2001)
193. Meghini, C., Sebastiani, F., Straccia, U., Thanos, C.: A model of information retrieval based on a terminological logic. In: Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR 1993), Pittsburgh, USA, pp. 298–307 (1993)
194. Mitra, P., Noy, N.F., Jaiswal, A.: OMEN: A probabilistic ontology mapping tool. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 537–547. Springer, Heidelberg (2005)
195. Motik, B., Horrocks, I., Rosati, R., Sattler, U.: Can OWL and Logic Programming Live Together Happily Ever After? In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 501–514. Springer, Heidelberg (2006)
196. Motik, B., Rosati, R.: A Faithful Integration of Description Logics with Logic Programming. In: Proc. of the 20th Int. Joint Conf. on Artificial Intelligence (IJCAI 2007). Morgan Kaufmann Publishers, San Francisco (2007)
197. Muggleton, S.: Stochastic logic programs. In: De Raedt, L. (ed.) Proceedings of the 5th International Workshop on Inductive Logic Programming, p. 29. Department of Computer Science, Katholieke Universiteit Leuven (1995)

198. Mukaidono, M.: Foundations of fuzzy logic programming. In: Advances in Fuzzy Systems – Application and Theory, vol. 1. World Scientific, Singapore (1996)
199. Mukaidono, M., Shen, Z., Ding, L.: Fundamentals of fuzzy prolog. Int. J. Approx. Reasoning 3(2), 179–193, (1989)
200. Naeth, T.H.: Analysis of the average-case behavior of an inference algorithm for probabilistic description logics. Diplomarbeit, TU Hamburg-Harburg, Germany, (February 2007)
201. Ng, R., Subrahmanian, V.: Probabilistic logic programming. Information and Computation 101(2), 150–201, (1993)
202. Ng, R., Subrahmanian, V.: Stable model semantics for probabilistic deductive databases. Information and Computation 110(1), 42–83, (1994)
203. Ngo, L.: Probabilistic disjunctive logic programming. In: Uncertainty in Artificial Intelligence: Proceedings of the Twelfeth Conference (UAI 1996), pp. 397–404. Morgan Kaufmann, San Francisco (1996)
204. Ngo, L., Haddawy, P.: Probabilistic logic programming and bayesian networks. In: Kanchanasut, K., Levy, J.-J. (eds.) ACSC 1995. LNCS, vol. 1023, pp. 286–300. Springer, Heidelberg (1995)
205. Ngo, L., Haddawy, P.: Answering queries from context-sensitive probabilistic knowledge bases. Theoretical Computer Science 171(1-2), 147–177 (1997)
206. Nicolas, P., Garcia, L., Stéphan, I.: Possibilistic stable models. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005), pp. 248–253. Morgan Kaufmann, San Francisco (2005)
207. Nilsson, N.J.: Probabilistic logic. Artif. Intell. 28(1), 71–88 (1986)
208. Nottelmann, H., Fuhr, N.: Adding probabilities and rules to OWL Lite subsets based on probabilistic Datalog. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 14(1), 17–42 (2006)
209. Novák, V.: Which logic is the real fuzzy logic? Fuzzy Sets and Systems, 635–641 (2005)
210. Pan, J.Z., Stamou, G., Stoilos, G., Thomas, E.: Expressive querying over fuzzy dl-lite ontologies. In: 20th International Workshop on Description Logics, Brixen-Bressanone, Italy (2007)
211. Pan, J.Z., Stamou, G., Stoilos, G., Thomas, E., Taylor, S.: Scalable querying service over fuzzy ontologies. In: International World Wide Web Conference (WWW 2008), Beijing (2008)
212. Pan, R., Ding, Z., Yu, Y., Peng, Y.: A Bayesian network approach to ontology mapping. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 563–577. Springer, Heidelberg (2005)
213. Paulik, L.: Best possible answer is computable for fuzzy sld-resolution. In: Hajék, P. (ed.) Gödel 1996: Logical Foundations of Mathematics, Computer Science, and Physics. Lecture Notes in Logic, vol. 6, pp. 257–266. Springer, Heidelberg (1996)
214. Pool, M., Aikin, J.: 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)
215. Poole, D.: Probabilistic horn abduction and bayesian networks. Artificial Intelligence 64(1), 81–129 (1993)
216. Poole, D.: The independent choice logic for modelling multiple agents under uncertainty. Artificial Intelligence 94(1-2), 7–56 (1997)
217. Qi, G., Pan, J.Z., Ji, Q.: Extending description logics with uncertainty reasoning in possibilistic logic. In: Mellouli, K. (ed.) ECSQARU 2007. LNCS (LNAI), vol. 4724, pp. 828–839. Springer, Heidelberg (2007)
218. Qi, G., Pan, J.Z., Ji, Q.: Extending description logics with uncertainty reasoning in possibilistic logic. In: Mellouli, K. (ed.) ECSQARU 2007. LNCS (LNAI), vol. 4724, Springer, Heidelberg (2007)

219. Qi, G., Pan, J.Z., Ji, Q.: A possibilistic extension of description logics. In: Proceedings DL 2007 (2007)
220. Ragone, A., Straccia, U., Bobillo, F., Noia, T.D., Sciascio, E.D.: Fuzzy description logics for bilateral matchmaking in e-marketplaces. In: Proceedings of the 21st International Workshop on Description Logics (DL 2008) (2008)
221. Ragone, A., Straccia, U., Noia, T.D., Sciascio, E.D., Donini, F.M.: Vague knowledge bases for matchmaking in p2p e-marketplaces. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 414–428. Springer, Heidelberg (2007)
222. Rhodes, P.C., Menani, S.M.: Towards a fuzzy logic programming system: a clausal form fuzzy logic. Knowledge-Based Systems 8(4), 174–182 (1995)
223. Rosati, R.: On the decidability and complexity of integrating ontologies and rules. Journal of Web Semantics 3(1), 61–73, (2005)
224. Rosati, R.: Dl+log: Tight integration of description logics and disjunctive datalog. In: Tenth International Conference on Principles of Knowledge Representation and Reasoning (KR 2006), pp. 68–78. AAAI Press, Menlo Park (2006)
225. Rosati, R.: On the decidability and finite controllability of query processing in databases with incomplete information. In: PODS 2006: Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 356–365. ACM Press, New York (2006)
226. Rounds, W.C., Zhang, G.-Q.: Clausal logic and logic programming in algebraic domains. Information and Computation 171, 183–200 (2001)
227. Sanchez, D., Tettamanzi, A.G.: Generalizing quantification in fuzzy description logics. In: Proceedings 8th Fuzzy Days in Dortmund, (2004)
228. Sanchez, D., Tettamanzi, A.G.: Reasoning and quantification in fuzzy description logics. In: Bloch, I., Petrosino, A., Tettamanzi, A.G.B. (eds.) WILF 2005. LNCS (LNAI), vol. 3849, pp. 81–88. Springer, Heidelberg (2006)
229. Sanchez, D., Tettamanzi, A.G.: Fuzzy quantification in fuzzy description logics. In: Sanchez, E. (ed.) Capturing Intelligence: Fuzzy Logic and the Semantic Web. Elsevier, Amsterdam (2006)
230. Schmidt-Schauß, M., Smolka, G.: Attributive concept descriptions with complements. Artificial Intelligence 48, 1–26 (1991)
231. Schroeder, M., Schweimeier, R.: Fuzzy argumentation and extended logic programming. In: Proceedings of ECSQARU Workshop Adventures in Argumentation (2001)
232. Schroeder, M., Schweimeier, R.: Arguments and misunderstandings: Fuzzy unification for negotiating agents. In: Proceedings of the ICLP workshop CLIMA 2002. Elsevier, Amsterdam (2002)
233. Schroeder, M., Schweimeier, R.: Fuzzy unification and argumentation for well-founded semantics. In: Van Emde Boas, P., Pokorný, J., Bieliková, M., Štuller, J. (eds.) SOFSEM 2004. LNCS, vol. 2932, pp. 102–121. Springer, Heidelberg (2004)
234. Seipel, D., Minker, J., Ruiz, C.: Model generation and state generation for disjunctive logic programs. Journal of Logic Programming 32(1), 49–69 (1997)
235. Sessa, M.I.: Approximate reasoning by similarity-based sld resolution. Theoretical Computer Science 275, 389–426 (2002)
236. Shapiro, E.Y.: Logic programs with uncertainties: A tool for implementing rule-based systems. In: Proceedings of the 8th International Joint Conference on Artificial Intelligence (IJCAI 1983), pp. 529–532 (1983)
237. Shen, Z., Ding, L., Mukaidono, M.: A Theoretical Framework of Fuzzy Prolog Machine. In: Fuzzy Computing, pp. 89–100. Elsevier Science Publishers B.V, Amsterdam (1988)
238. Stoilos, G., Stamou, G., Pan, J.: D2.5.6: Fuzzy reasoning extensions. Knowledge Web Technical Report, (2007)

239. Stoilos, G., Stamou, G., Pan, J., Tzouvaras, V., Horrocks, I.: The fuzzy description logic f-SHIN. In: International Workshop on Uncertainty Reasoning For the Semantic (Web 2005) (2005)
240. Stoilos, G., Stamou, G., Tzouvaras, V., Pan, J.Z., Horrock, I.: A Fuzzy Description Logic for Multimedia Knowledge Representation. In: Proc. of the International Workshop on Multimedia and the Semantic Web, (2005)
241. Stoilos, G., Stamou, G.B., Pan, J.Z., Tzouvaras, V., Horrocks, I.: Reasoning with very expressive fuzzy description logics. Journal of Artificial Intelligence Research 30, 273–320 (2007)
242. Stoilos, G., Straccia, U., Stamou, G., Pan, J.: General concept inclusions in fuzzy description logics. In: Proceedings of the 17th Eureopean Conference on Artificial Intelligence (ECAI 2006), pp. 457–461. IOS Press, Amsterdam (2006)
243. Straccia, U.: A fuzzy description logic. In: Proc. of the 15th Nat. Conf. on Artificial Intelligence (AAAI 1998), pp. 594–599. AAAI Press, Menlo Park (1998)
244. Straccia, U.: A framework for the retrieval of multimedia objects based on four-valued fuzzy description logics. In: Crestani, F., Pasi, G. (eds.) Soft Computing in Information Retrieval: Techniques and Applications, pp. 332–357. Springer, Heidelberg (2000)
245. Straccia, U.: Reasoning within fuzzy description logics. Journal of Artificial Intelligence Research 14, 137–166 (2001)
246. Straccia, U.: Transforming fuzzy description logics into classical description logics. In: Alferes, J.J., Leite, J.A. (eds.) JELIA 2004. LNCS (LNAI), vol. 3229, pp. 385–399. Springer, Heidelberg (2004)
247. Straccia, U.: Uncertainty in description logics: a lattice-based approach. In: Proceedings of the 10th International Conference on Information Processing and Managment of Uncertainty in Knowledge-Based Systems (IPMU 2004), pp. 251–258 (2004)
248. Straccia, U.: Description logics with fuzzy concrete domains. In: Bachus, F., Jaakkola, T. (eds.) 21st Conference on Uncertainty in Artificial Intelligence (UAI 2005), Edinburgh, Scotland, pp. 559–567. AUAI Press (2005)
249. Straccia, U.: Fuzzy alc with fuzzy concrete domains. In: Proceeedings of the International Workshop on Description Logics (DL 2005), Edinburgh, Scotland, pp. 96–103 (2005)
250. Straccia, U.: Query answering in normal logic programs under uncertainty. In: Godo, L. (ed.) ECSQARU 2005. LNCS (LNAI), vol. 3571, pp. 687–700. Springer, Heidelberg (2005)
251. Straccia, U.: Towards a fuzzy description logic for the semantic web (preliminary report). In: Gómez-Pérez, A., Euzenat, J. (eds.) ESWC 2005. LNCS, vol. 3532, pp. 167–181. Springer, Heidelberg (2005)
252. Straccia, U.: Uncertainty management in logic programming: Simple and effective top-down query answering. In: Khosla, R., Howlett, R.J., Jain, L.C. (eds.) KES 2005. LNCS (LNAI), vol. 3682, pp. 753–760. Springer, Heidelberg (2005)
253. Straccia, U.: Annotated answer set programming. In: Proceedings of the 11th International Conference on Information Processing and Managment of Uncertainty in Knowledge-Based Systems (IPMU 2006), pp. 1212–1219. E.D.K, Paris (2006)
254. Straccia, U.: Answering vague queries in fuzzy dl-lite. In: Proceedings of the 11th International Conference on Information Processing and Managment of Uncertainty in Knowledge-Based Systems (IPMU 2006), pp. 2238–2245. E.D.K., Paris (2006)
255. Straccia, U.: Description logics over lattices. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 14(1), 1–16 (2006)
256. Straccia, U.: A fuzzy description logic for the semantic web. In: Sanchez, E. (ed.) Fuzzy Logic and the Semantic Web, Capturing Intelligence, vol. 4, pp. 73–90. Elsevier, Amsterdam (2006)

257. Straccia, U.: Fuzzy description logic programs. In: Proceedings of the 11th International Conference on Information Processing and Managment of Uncertainty in Knowledge-Based Systems (IPMU 2006), pp. 1818–1825. E.D.K., Paris (2006)
258. Straccia, U.: Query answering under the any-world assumption for normal logic programs. In: Proceedings of the 10th International Conference on Principles of Knowledge Representation (KR 2006), pp. 329–339. AAAI Press, Menlo Park (2006)
259. Straccia, U.: Towards top-k query answering in deductive databases. In: Proceedings of the 2006 IEEE International Conference on Systems, Man and Cybernetics (SMC 2006), pp. 4873–4879. IEEE, Los Alamitos (2006)
260. Straccia, U.: Towards top-k query answering in description logics: the case of DL-Lite. In: Fisher, M., van der Hoek, W., Konev, B., Lisitsa, A. (eds.) JELIA 2006. LNCS (LNAI), vol. 4160, pp. 439–451. Springer, Heidelberg (2006)
261. Straccia, U.: Uncertainty and description logic programs over lattices. In: Sanchez, E. (ed.) Fuzzy Logic and the Semantic Web, Capturing Intelligence, vol. 7, pp. 115–133. Elsevier, Amsterdam (2006)
262. Straccia, U.: Reasoning in ł-$\mathcal{SHIF}$: an expressive fuzzy description logic under łukasiewicz semantics. Technical Report TR-2007-10-18, Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, Pisa, Italy (2007)
263. Straccia, U.: A top-down query answering procedure for normal logic programs under the any-world assumption. In: Mellouli, K. (ed.) ECSQARU 2007. LNCS (LNAI), vol. 4724, pp. 115–127. Springer, Heidelberg (2007)
264. Straccia, U.: Towards vague query answering in logic programming for logic-based information retrieval. In: Melin, P., Castillo, O., Aguilar, L.T., Kacprzyk, J., Pedrycz, W. (eds.) IFSA 2007. LNCS (LNAI), vol. 4529, pp. 125–134. Springer, Heidelberg (2007)
265. Straccia, U., Bobillo, F.: Mixed integer programming, general concept inclusions and fuzzy description logics. Mathware & Soft Computing 14(3), 247–259 (2007)
266. Straccia, U., Visco, G.: DLMedia: an ontology mediated multimedia information retrieval system. In: Proceeedings of the International Workshop on Description Logics (DL 2007), Insbruck, Austria, vol. 250, CEUR (2007)
267. Subramanian, V.: On the semantics of quantitative logic programs. In: Proc. 4th IEEE Symp. on Logic Programming, pp. 173–182. IEEE Computer Society Press, Los Alamitos (1987)
268. Thomas, C., Sheth, A.: On the expressiveness of the languages for the Semantic Web — Making a case for "A little more". In: Sanchez, E. (ed.) Fuzzy Logic and the Semantic Web, Capturing Intelligence, pp. 3–20. Elsevier, Amsterdam (2006)
269. Tresp, C., Molitor, R.: A description logic for vague knowledge. In: Proc. of the 13th European Conf. on Artificial Intelligence (ECAI 1998), Brighton, England (August 1998)
270. Turner, H.: Signed logic programs. In: Bruynooghe, M. (ed.) Logic Programming: Proc. of the 1994 International Symposium, pp. 61–75. MIT Press, Cambridge (1994)
271. Udrea, O., Deng, Y., Hung, E., Subrahmanian, V.S.: Probabilistic ontologies and relational databases. In: Meersman, R., Tari, Z. (eds.) OTM 2005. LNCS, vol. 3760, pp. 1–17. Springer, Heidelberg (2005)
272. Udrea, O., Subrahmanian, V.S., Majkic, Z.: Probabilistic rdf. In: Proceedings of the 2006 IEEE International Conference on Information Reuse and Integration, IRI 2006: Heuristic Systems Engineering, pp. 172–177. IEEE Systems, Los Alamitos (2006)
273. van Emden, M.: Quantitative deduction and its fixpoint theory. Journal of Logic Programming 4(1), 37–53 (1986)
274. Venetis, T., Stoilos, G., Stamou, G., Kollias, S.: f-dlps: Extending description logic programs with fuzzy sets and fuzzy logic. In: IEEE International Conference on Fuzzy Systems (Fuzz-IEEE 2007), London (2007)

275. Vennekens, J., Verbaeten, S., Bruynooghe, M.: Logic programs with annotated disjunctions. In: Demoen, B., Lifschitz, V. (eds.) ICLP 2004. LNCS, vol. 3132, pp. 431–445. Springer, Heidelberg (2004)
276. Vojtáš, P.: Fuzzy logic programming. Fuzzy Sets and Systems 124, 361–370 (2001)
277. Vojtáš, P., Paulík, L.: Soundness and completeness of non-classical extended SLD-resolution. In: Herre, H., Dyckhoff, R., Schroeder-Heister, P. (eds.) ELP 1996. LNCS, vol. 1050, pp. 289–301. Springer, Heidelberg (1996)
278. Vojtáš, P., Vomelelová, M.: Transformation of deductive and inductive tasks between models of logic programming with imperfect information. In: Proceedings of the 10th International Conference on Information Processing and Managment of Uncertainty in Knowledge-Based Systems (IPMU 2004), pp. 839–846 (2004)
279. Wagner, G.: Negation in fuzzy and possibilistic logic programs. In: Martin, T., Arcelli, F. (eds.) Logic programming and Soft Computing. Research Studies Press (1998)
280. Weikum, G., Graupmann, J., Schenkel, R., Theobald, M.: Towards a statistically Semantic Web. In: Atzeni, P., Chu, W., Lu, H., Zhou, S., Ling, T.-W. (eds.) ER 2004. LNCS, vol. 3288, pp. 3–17. Springer, Heidelberg (2004)
281. Wüttrich, B.: Probabilistic knowledge bases. IEEE Transactions on Knowledge and Data Engineering 7(5), 691–698 (1995)
282. Yang, Y., Calmet, J.: OntoBayes: An ontology-driven uncertainty model. In: Proceedings IAWTIC 2005, pp. 457–463. IEEE Press, Los Alamitos (2005)
283. Li, J.L.Y., Xu, B., Kang, D.: Discrete tableau algorithms for shi. In: Proceeedings of the International Workshop on Description Logics (DL 2006). CEUR (2006)
284. Yasui, H., Hamada, Y., Mukaidono, M.: Fuzzy prolog based on lukasiewicz implication and bounded product. IEEE Trans. Fuzzy Systems 2, 949–954 (1995)
285. Yelland, P.M.: An alternative combination of Bayesian networks and description logics. In: Proceedings KR 2000, pp. 225–234. Morgan Kaufmann, San Francisco (2000)
286. Yen, J.: Generalizing term subsumption languages to fuzzy logic. In: Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI 1991), Sydney, Australia, pp. 472–477 (1991)
287. Zadeh, L.A.: Fuzzy sets. Information and Control 8(3), 338–353 (1965)
288. Zhang, L., Yu, Y., Zhou, J., Lin, C., Yang, Y.: An enhanced model for searching in semantic portals. In: WWW 2005: Proceedings of the 14th international conference on World Wide Web, pp. 453–462. ACM Press, New York (2005)

A Some References Related to Logic Programming, Uncertainty and Vagueness

Below a list of references and the underlying imprecision and uncertainty theory in logic programming frameworks. The list of references is by no means intended to be all-inclusive. The author apologizes both to the authors and with the readers for all the relevant works, which are not cited here.

Probability theory:[11,7,8,40,46,48,44,45,47,82,117,118,116,129,131,152,153,155] [156,158,162,160,184,197,201,202,203,205,215,216,275,281]

Possibilistic logic: [3,4,2,25,57,206]

Fuzzy set theory: [185,9,10,19,24,27,67,100,111,123,89,88,185,199,198,213] [222,235,236,237,177,267,273,277,276,278,279,284]

Multi-valued logic: [22, 34, 35, 36, 31, 32, 37, 38, 39, 41, 33, 53, 49, 52, 50, 51]
[78, 80, 75, 76, 77, 91, 119, 120, 121, 122, 126, 127, 128, 130]
[142, 143, 144, 145, 146, 147, 148, 150, 151, 179, 181, 180, 183, 186, 187, 190, 190, 188, 189, 191]
[226, 231, 232, 233, 252, 250, 253, 258, 259, 264, 263, 270]