Dealing with Incompleteness and Vagueness in Inductive Logic Programming

Francesca A. Lisi[1] and Umberto Straccia[2]

[1] Dipartimento di Informatica, Università degli Studi di Bari "Aldo Moro", Italy
[2] ISTI - CNR, Pisa, Italy

Abstract. Incompleteness and vagueness are inherent properties of knowledge in several real world domains and are particularly pervading in those domains where entities could be better described in natural language. In order to deal with incomplete and vague structured knowledge, several fuzzy extensions of Description Logics (DLs) have been proposed in the literature. In this paper, we address the issues raised by incomplete and vague knowledge in Inductive Logic Programming (ILP). We present a novel ILP method for inducing fuzzy DL inclusion axioms from crisp DL knowledge bases and discuss the results obtained in comparison with related works.

1 Introduction

Incompleteness and vagueness are inherent properties of knowledge in several real world domains and are particularly pervading in those domains where entities could be better described in natural language. The issues raised by incomplete and vague knowledge have been traditionally addressed in the field of Knowledge Representation (KR).

Incomplete knowledge. The *Open World Assumption* (OWA) is used in KR to codify the informal notion that in general no single agent or observer has complete knowledge. The OWA limits the kinds of inference and deductions an agent can make to those that follow from statements that are known to the agent to be true. In contrast, the *Closed World Assumption* (CWA) allows an agent to infer, from its lack of knowledge of a statement being true, anything that follows from that statement being false. Heuristically, the OWA applies when we represent knowledge within a system as we discover it, and where we cannot guarantee that we have discovered or will discover complete information. In the OWA, statements about knowledge that are not included in or inferred from the knowledge explicitly recorded in the system may be considered unknown, rather than wrong or false. Description Logics (DLs) are KR formalisms compliant with the OWA, thus turning out to be particularly suitable for representing *incomplete* knowledge [1].

Vague knowledge. It is well known that "classical" DLs are not appropriate to deal with *vague* knowledge [20]. We recall for the inexpert reader that 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 [5]. Specifically, under *uncertainty theory* fall all those approaches in which statements 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/interpretation, but we are "uncertain" about which world to consider as the right one, and thus we speak about, *e.g.*, a probability distribution or a possibility distribution over the worlds. On the other hand, under *fuzzy theory* fall all those approaches in which statements (for example, "the car is long") are true to some *degree*, which is taken from a truth space (usually $[0, 1]$). That is, an interpretation maps a statement to a truth degree, since we are unable to establish whether a statement is entirely true or false due to the involvement of vague concepts, such as "long car" (the degree to which the sentence is true depends on the length of the car). Here, we shall focus on fuzzy logic only.

Learning in fuzzy DLs. Although a relatively important amount of work has been carried out in the last years concerning the use of fuzzy DLs as ontology languages [20] and the use of DLs as representation formalisms in Inductive Logic Programming (ILP) [13], the problem of automatically managing the evolution of fuzzy ontologies by applying ILP algorithms still remains relatively unaddressed. Konstantopoulos and Charalambidis [9] propose an ad-hoc translation of fuzzy Łukasiewicz $\mathcal{ALC}$ DL constructs into LP in order to apply a conventional ILP method for rule learning. However, the method is not sound as it has been recently shown that the mapping from fuzzy DLs to LP is incomplete [17] and entailment in Łukasiewicz $\mathcal{ALC}$ is undecidable [4]. Iglesias and Lehmann [7] propose an extension of DL-Learner [10] with some of the most up-to-date fuzzy ontology tools, e.g. the *fuzzyDL* reasoner [2]. Notably, the resulting system can learn fuzzy OWL DL [3] equivalence axioms from FuzzyOWL 2 ontologies. [4] However, it has been tested only on a toy problem with crisp training examples and does not build automatically fuzzy concrete domains. Lisi and Straccia [14] present *Soft*FOIL, a logic-based method for learning fuzzy $\mathcal{EL}$ inclusion axioms from fuzzy DL-Lite ontologies (also, *Soft*FOIL has not been implemented and tested).

Contribution of this paper. In this paper, we describe a novel method, named FOIL-$\mathcal{DL}$, for learning fuzzy $\mathcal{EL}(\mathbf{D})$ inclusion axioms from any crisp DL knowledge base. [5] Similarly to *Soft*FOIL, it adapts the popular rule induction method FOIL [18]. However, FOIL-$\mathcal{DL}$ differs from *Soft*FOIL mainly by the fact that the latter learns fuzzy $\mathcal{EL}$ inclusion axioms from fuzzy DL-Lite ontologies, while the former learns fuzzy $\mathcal{EL}(\mathbf{D})$ inclusion axioms from any crisp DL ontology.

Structure of the paper. The paper is structured as follows. For the sake of self-containment, Section 2 introduces some basic definitions we rely on. Section 3 describes the learning problem and the solution strategy of FOIL-$\mathcal{DL}$. Section 4 illustrates some results obtained in a comparative study between FOIL-$\mathcal{DL}$ and

[3] `http://www.w3.org/TR/2009/REC-owl2-overview-20091027/`
[4] `http://www.straccia.info/software/FuzzyOWL`
[5] $\mathcal{DL}$ stands for any DL.

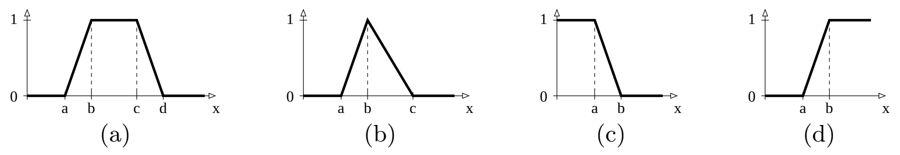


Fig. 1. (a) Trapezoidal function $trz(a, b, c, d)$, (b) triangular function $tri(a, b, c)$, (c) left-shoulder function $ls(a, b)$, and (d) right-shoulder function $rs(a, b)$.

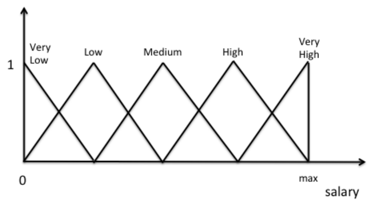
DL-Learner on the popular ILP problem of Michalski's trains. Section 5 concludes the paper by discussing limits of the current work, related work and possible directions of future work.

2 Preliminaries

Mathematical Fuzzy Logic. Fuzzy Logic is the logic of fuzzy sets. A *fuzzy set* A over a countable crisp set X is a function $A\colon X \to [0, 1]$. Let A and B be two fuzzy sets. The standard fuzzy set operations conform to $(A \cap B)(x) = \min(A(x), B(x))$, $(A \cup B)(x) = \max(A(x), B(x))$ and $\bar{A}(x) = 1 - A(x)$, while the *inclusion degree* between A and B is defined typically as

$$deg(A, B) = \frac{\sum_{x \in X}(A \cap B)(x)}{\sum_{x \in X} A(x)} \ . \tag{1}$$

The trapezoidal (Fig. 1 (a)), the triangular (Fig. 1 (b)), the left-shoulder function, Fig. 1 (c)), and the right-shoulder function, Fig. 1 (d)) are frequently used to specify *membership functions* of fuzzy sets. Although fuzzy sets have a greater expressive power than classical crisp sets, their usefulness depend critically on the capability to construct appropriate membership functions for various given concepts in different contexts. The problem of constructing meaningful membership functions is a difficult one (see, *e.g.*, [8, Chapter 10]). However, one easy and typically satisfactory method to define the membership functions is to uniformly partition the range of values into 5 or 7 fuzzy sets using either trapezoidal functions, or triangular functions. The latter is the more used one, as it has less parameters and is also the approach we adopt. For instance, the figure below illustrates salary values (bounded by a minimum and maximum value), partitioned uniformly into 5 fuzzy sets.



In *Mathematical Fuzzy Logic* [6], the convention prescribing that a statement is either true or false is changed and is a matter of degree measured on an ordered scale that is no longer $\{0, 1\}$, but *e.g.* $[0, 1]$. This degree is called *degree of truth*

Table 1. Syntax and semantics of constructs for the $\mathcal{ALC}$ DL.

bottom (resp. top) concept	$\bot$ (resp. $\top$)	$\emptyset$ (resp. $\Delta^{\mathcal{I}}$)
atomic concept	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
role	R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
individual	a	$a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
concept negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
concept intersection	$C_1 \sqcap C_2$	$C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$
concept union	$C_1 \sqcup C_2$	$C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
value restriction	$\forall R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \forall y\ (x,y) \in R^{\mathcal{I}} \to y \in C^{\mathcal{I}}\}$
existential restriction	$\exists R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y\ (x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
general concept inclusion	$C_1 \sqsubseteq C_2$	$C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$
concept assertion	$a : C$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
role assertion	$(a,b) : R$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$

of the logical statement ϕ in the interpretation $\mathcal{I}$. For us, *fuzzy statements* have the form $\langle \phi, \alpha \rangle$, where $\alpha \in (0,1]$ and ϕ is a statement, encoding that the degree of truth of ϕ is *greater or equal* α.

A *fuzzy interpretation* $\mathcal{I}$ maps each atomic statement p_i into $[0,1]$ and is then extended inductively to all statements: $\mathcal{I}(\phi \wedge \psi) = \mathcal{I}(\phi) \otimes \mathcal{I}(\psi)$, $\mathcal{I}(\phi \vee \psi) = \mathcal{I}(\phi) \oplus \mathcal{I}(\psi)$, $\mathcal{I}(\phi \to \psi) = \mathcal{I}(\phi) \Rightarrow \mathcal{I}(\psi)$, $\mathcal{I}(\neg \phi) = \ominus \mathcal{I}(\phi)$, $\mathcal{I}(\exists x.\phi(x)) = \sup_{y \in \Delta^{\mathcal{I}}} \mathcal{I}(\phi(y))$, $\mathcal{I}(\forall x.\phi(x)) = \inf_{y \in \Delta^{\mathcal{I}}} \mathcal{I}(\phi(y))$, where $\Delta^{\mathcal{I}}$ is the domain of $\mathcal{I}$, and $\otimes$, $\oplus$, $\Rightarrow$, and $\ominus$ are so-called *t-norms*, *t-conorms*, *implication functions*, and *negation functions*, respectively, which extend the Boolean conjunction, disjunction, implication, and negation, respectively, to the fuzzy case. One usually distinguishes three different logics, namely Łukasiewicz, Gödel, and Product logics [6]. Any other continuous t-norm can be obtained from them. The combination functions in Gödel logic are defined as follows:

$$a \otimes b = \min(a,b), a \oplus b = \max(a,b), a \Rightarrow b = \begin{cases} 1 & \text{if } a \leq b \\ b & \text{otherwise} \end{cases}, \ominus a = \begin{cases} 1 & \text{if } a = 0 \\ 0 & \text{otherwise} \end{cases}. \quad (2)$$

The notions of satisfiability and logical consequence are defined in the standard way, where a fuzzy interpretation $\mathcal{I}$ *satisfies* a fuzzy statement $\langle \phi, \alpha \rangle$ or $\mathcal{I}$ is a *model* of $\langle \phi, \alpha \rangle$, denoted as $\mathcal{I} \models \langle \phi, \alpha \rangle$, iff $\mathcal{I}(\phi) \geq \alpha$.

Fuzzy Description Logics. Description Logics (DLs) are a family of decidable First Order Logic (FOL) fragments that allow for the specification of structured knowledge in terms of classes (*concepts*), instances (*individuals*), and binary relations between instances (*roles*) [1]. Complex concepts (denoted with C) can be defined from atomic concepts (A) and roles (R) by means of the constructors available for the DL in hand. The set of constructors for the $\mathcal{ALC}$ DL is reported in Table 1. A DL *Knowledge Base* (KB) $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is a pair where $\mathcal{T}$ is the so-called *Terminological Box* (TBox) and $\mathcal{A}$ is the so-called *Assertional Box* (ABox). The TBox is a finite set of *General Concept Inclusion* (GCI) axioms which represent is-a relations between concepts, whereas the ABox is a finite set of *assertions* (or *facts*) that represent instance-of relations between individuals (resp. couples of individuals) and concepts (resp. roles). Thus, when a DL-based ontology language is adopted, an ontology is nothing else than a TBox, and a populated ontology corresponds to a whole KB (*i.e.*, encompassing also an ABox).

The semantics of DLs can be defined directly with set-theoretic formalizations (as shown in Table 1 for the case of $\mathcal{ALC}$) or through a mapping to FOL (as shown in [3]). An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ for a DL KB consists of a domain $\Delta^{\mathcal{I}}$ and a mapping function $\cdot^{\mathcal{I}}$. For instance, $\mathcal{I}$ maps a concept C into a set of individuals $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, *i.e.* $\mathcal{I}$ maps C into a function $C^{\mathcal{I}} : \Delta^{\mathcal{I}} \to \{0,1\}$ (either an individual belongs to the extension of C or does not belong to it). Under the *Unique Names Assumption* (UNA) [19], individuals are mapped to elements of $\Delta^{\mathcal{I}}$ such that $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ if $a \neq b$. However UNA does not hold by default in DLs. An interpretation $\mathcal{I}$ is a *model* of a KB $\mathcal{K}$ iff it satisfies all axioms and assertions in $\mathcal{T}$ and $\mathcal{A}$. In DLs a KB represents many different interpretations, *i.e.* all its models. This is coherent with the OWA that holds in FOL semantics. A DL KB is *satisfiable* if it has at least one model.

The main reasoning task for a DL KB $\mathcal{K}$ is the *consistency check* which tries to prove the satisfiability of $\mathcal{K}$. Another well known reasoning service in DLs is *instance check*, i.e., the check of whether an ABox assertion is a logical implication of a DL KB. A more sophisticated version of instance check, called *instance retrieval*, retrieves, for a DL KB $\mathcal{K}$, all (ABox) individuals that are instances of the given (possibly complex) concept expression C, i.e., all those individuals a such that $\mathcal{K}$ entails that a is an instance of C.

Concerning fuzzy DLs, several fuzzy extensions of DLs have been proposed (see the survey in [15]). We recap here the fuzzy variant of the DL $\mathcal{ALC}(\mathbf{D})$ [21].

A *fuzzy concrete domain* or *fuzzy datatype theory* $\mathbf{D} = \langle \Delta^{\mathbf{D}}, \cdot^{\mathbf{D}} \rangle$ consists of a datatype domain $\Delta^{\mathbf{D}}$ and a mapping $\cdot^{\mathbf{D}}$ that assigns to each data value an element of $\Delta^{\mathbf{D}}$, and to every n-ary datatype predicate d an n-ary fuzzy relation over $\Delta^{\mathbf{D}}$. We will restrict to unary datatypes as usual in fuzzy DLs. Therefore, $\cdot^{\mathbf{D}}$ maps indeed each datatype predicate into a function from $\Delta^{\mathbf{D}}$ to $[0,1]$. Typical examples of datatype predicates $\mathbf{d}$ are the well known membership functions

$$\mathbf{d} := ls(a,b) \mid rs(a,b) \mid tri(a,b,c) \mid trz(a,b,c,d) \mid \geq_v \mid \leq_v \mid =_v ,$$

where *e.g.* $ls(a,b)$ is the left-shoulder membership function and $\geq_v$ corresponds to the crisp set of data values that are greater or equal than the value v.

In $\mathcal{ALC}(\mathbf{D})$, each role is either an *object property* (denoted with R) or a *datatype property* (denoted with T). Complex concepts are built according to the following syntactic rules:

$$C \to \top \mid \bot \mid A \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \neg C \mid C_1 \to C_2 \mid \exists R.C \mid \forall R.C \mid \exists T.\mathbf{d} \mid \forall T.\mathbf{d} \ . \quad (3)$$

Axioms in a fuzzy $\mathcal{ALC}(\mathbf{D})$ KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ are graded, *e.g.* a GCI is of the form $\langle C_1 \sqsubseteq C_2, \alpha \rangle$ (*i.e.* C_1 is a sub-concept of C_2 to degree at least α). We may omit the truth degree α of an axiom; in this case $\alpha = 1$ is assumed.

Concerning the semantics, let us fix a fuzzy logic. In fuzzy DLs, $\mathcal{I}$ maps C into a function $C^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0,1]$ and, thus, an individual belongs to the extension of C to some degree in $[0,1]$, *i.e.* $C^{\mathcal{I}}$ is a fuzzy set. Specifically, a *fuzzy* *interpretation* is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consisting of a nonempty (crisp) set $\Delta^{\mathcal{I}}$ (the *domain*) and of a *fuzzy interpretation function* $\cdot^{\mathcal{I}}$ that assigns: *(i)* to each atomic concept A a function $A^{\mathcal{I}}\colon \Delta^{\mathcal{I}} \to [0,1]$; *(ii)* to each object property R a function $R^{\mathcal{I}}\colon \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0,1]$; *(iii)* to each data type property T a function $T^{\mathcal{I}}\colon \Delta^{\mathcal{I}} \times \Delta^{\mathbf{D}} \to [0,1]$; *(iv)* to each individual a an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$; and *(v)* to each concrete value v an element $v^{\mathcal{I}} \in \Delta^{\mathbf{D}}$.

Now, $\cdot^{\mathcal{I}}$ is extended to concepts as specified below (where $x \in \Delta^{\mathcal{I}}$):

$$\begin{aligned}
&\bot^{\mathcal{I}}(x) = 0, \quad \top^{\mathcal{I}}(x) = 1,\\
&(C \sqcap D)^{\mathcal{I}}(x) = C^{\mathcal{I}}(x) \otimes D^{\mathcal{I}}(x), \quad (C \sqcup D)^{\mathcal{I}}(x) = C^{\mathcal{I}}(x) \oplus D^{\mathcal{I}}(x),\\
&(\neg C)^{\mathcal{I}}(x) = \ominus C^{\mathcal{I}}(x), \quad (C \to D)^{\mathcal{I}}(x) = C^{\mathcal{I}}(x) \Rightarrow D^{\mathcal{I}}(x),\\
&(\forall R.C)^{\mathcal{I}}(x) = \inf_{y \in \Delta^{\mathcal{I}}}\{R^{\mathcal{I}}(x,y) \Rightarrow C^{\mathcal{I}}(y)\}, \quad (\exists R.C)^{\mathcal{I}}(x) = \sup_{y \in \Delta^{\mathcal{I}}}\{R^{\mathcal{I}}(x,y) \otimes C^{\mathcal{I}}(y)\},\\
&(\forall T.\mathbf{d})^{\mathcal{I}}(x) = \inf_{y \in \Delta^{\mathbf{D}}}\{T^{\mathcal{I}}(x,y) \Rightarrow \mathbf{d}^{\mathbf{D}}(y)\}, \quad (\exists T.\mathbf{d})^{\mathcal{I}}(x) = \sup_{y \in \Delta^{\mathbf{D}}}\{T^{\mathcal{I}}(x,y) \otimes \mathbf{d}^{\mathbf{D}}(y)\} .
\end{aligned}$$

Hence, for every concept C we get a function $C^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0,1]$.

The *satisfiability of axioms* is then defined by the following conditions: *(i)* $\mathcal{I}$ satisfies an axiom $\langle a{:}C, \alpha\rangle$ if $C^{\mathcal{I}}(a^{\mathcal{I}}) \geq \alpha$; *(ii)* $\mathcal{I}$ satisfies an axiom $\langle (a,b){:}R, \alpha\rangle$ if $R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) \geq \alpha$; *(iii)* $\mathcal{I}$ satisfies an axiom $\langle C \sqsubseteq D, \alpha\rangle$ if $(C \sqsubseteq D)^{\mathcal{I}} \geq \alpha$ where $(C \sqsubseteq D)^{\mathcal{I}} = \inf_{x \in \Delta^{\mathcal{I}}}\{C^{\mathcal{I}}(x) \Rightarrow D^{\mathcal{I}}(x)\}$. $\mathcal{I}$ is a model of $\mathcal{K}$ iff $\mathcal{I}$ satisfies each axiom in $\mathcal{K}$. We say that $\mathcal{K}$ *entails* an axiom $\langle \tau, \alpha\rangle$, denoted $\mathcal{K} \models \langle \tau, \alpha\rangle$, if any model of $\mathcal{K}$ satisfies $\langle \tau, \alpha\rangle$. The *best entailment degree* of τ w.r.t. $\mathcal{K}$, denoted $bed(\mathcal{K}, \tau)$, is defined as

$$bed(\mathcal{K}, \tau) = \sup\{\alpha \mid \mathcal{K} \models \langle \tau, \alpha\rangle\} . \tag{4}$$

3 Learning fuzzy $\mathcal{EL}$(D) axioms with Foil-$\mathcal{DL}$

3.1 The problem statement

The problem considered in this paper concerns the automated induction of fuzzy $\mathcal{EL}(\mathbf{D})$ [6] GCI axioms providing a sufficient condition for a given atomic concept H. It can be cast as a rule learning problem, provided that positive and negative examples of H are available. This problem can be formalized as follows.

Given:

- a consistent crisp $\mathcal{DL}$ KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ (the *background theory*);
- an atomic concept H (the *target concept*);
- a set $\mathcal{E} = \mathcal{E}^+ \cup \mathcal{E}^-$ of crisp concept assertions labelled as either positive or negative examples for H (the *training set*);
- a set $\mathcal{L}_{\mathcal{H}}$ of fuzzy $\mathcal{EL}(\mathbf{D})$ GCI axioms (the *language of hypotheses*)

the goal is to find a set $\mathcal{H} \subset \mathcal{L}_{\mathcal{H}}$ (a *hypothesis*) such that: $\forall e \in \mathcal{E}^+, \mathcal{K} \cup \mathcal{H} \models e$ (completeness), and $\forall e \in \mathcal{E}^-, \mathcal{K} \cup \mathcal{H} \not\models e$ (consistency).

Here we assume that $\mathcal{K} \cap \mathcal{E} = \emptyset$. Also, the language $\mathcal{L}_{\mathcal{H}}$ is given implicitly by means of syntactic restrictions over a given alphabet. In particular, the alphabet underlying $\mathcal{L}_{\mathcal{H}}$ is a subset of the alphabet for the language $\mathcal{L}_{\mathcal{K}}$ of the background

[6] $\mathcal{EL}(\mathbf{D})$ is a fragment of $\mathcal{ALC}(\mathbf{D})$.

theory. However, $\mathcal{L}_{\mathcal{H}}$ differs from $\mathcal{L}_{\mathcal{K}}$ as for the form of axioms. Please note that we do not make any specific assumption about the DL the background theory refers to. Two further restrictions hold naturally. One is that $\mathcal{K} \not\models \mathcal{E}^+$ since, in such a case, $\mathcal{H}$ would not be necessary to explain $\mathcal{E}^+$. The other is that $\mathcal{K} \cup \mathcal{H} \not\models \bot$, which means that $\mathcal{K} \cup \mathcal{H}$ is a consistent theory, *i.e.* has a model. An axiom $\phi \in \mathcal{L}_{\mathcal{H}}$ *covers* an example $e \in \mathcal{E}$ iff $\mathcal{K} \cup \{\phi\} \models e$.

The training examples. Given the target concept H, the training set $\mathcal{E}$ consists of concept assertions of the form $a{:}H$, where a is an individual occurring in $\mathcal{K}$. Note that both $\mathcal{K}$ and $\mathcal{E}$ is crisp. Also, $\mathcal{E}$ is split into $\mathcal{E}^+$ and $\mathcal{E}^-$. Note that, under OWA, $\mathcal{E}^-$ consists of all those individuals which can be proved to be instance of $\neg H$. On the other hand, under CWA, $\mathcal{E}^-$ is the collection of individuals, which cannot be proved to be instance of H.

The language of hypotheses. Given the target concept H, the hypotheses to be induced are fuzzy GCIs of the form

$$B \sqsubseteq H \ , \tag{5}$$

where the left-hand side is defined according to the following $\mathcal{EL}(\mathbf{D})$ syntax

$$B \longrightarrow \top \mid A \mid \exists R.B \mid \exists T.\mathbf{d} \mid B_1 \sqcap B_2 \ . \tag{6}$$

The language $\mathcal{L}_{\mathcal{H}}$ generated by this syntax is potentially infinite due, *e.g.*, to the nesting of existential restrictions yielding to complex concept expressions such as $\exists R_1.(\exists R_2 \ldots .(\exists R_n.(C)) \ldots)$. $\mathcal{L}_{\mathcal{H}}$ is made finite by imposing further restrictions on the generation process such as the maximal number of conjuncts and the depth of existential nesting allowed in the left-hand side. Also, note that the learnable GCIs do not have an explicit truth degree. However, as we shall see later on, once we have learned a fuzzy GCI of the form (5), we attach to it a confidence degree that is obtained by means of the cf function (see Eq. (8)). Finally, note that the syntactic restrictions of Eq. (6) w.r.t. Eq. (3) allow for a straightforward translation of the inducible axioms into rules of the kind “if x is a C_1 and ... and x is a C_n then x is an H”, which corresponds to the usual pattern in fuzzy rule induction (in our case, $B \sqsubseteq H$ is seen as a rule “if B then H”) .

3.2 The solution strategy

The solution proposed for the learning problem defined in Section 3.1 is inspired by Foil. Foil is a popular ILP algorithm for learning sets of rules which performs a greedy search in order to maximise a gain function [18].

In Foil-$\mathcal{DL}$, the learning strategy of Foil (*i.e.*, the so-called *sequential covering* approach) is kept. The function Learn-Sets-of-Axioms (reported in Figure 2) carries on inducing axioms until all positive examples are covered. When an axiom is induced (step 3.), the positive examples covered by the axiom (step 5.) are removed from $\mathcal{E}$ (step 6.). In order to induce an axiom, the function

```
function Learn-Sets-of-Axioms($\mathcal{K}$, $H$, $\mathcal{E}^+$, $\mathcal{E}^-$, $\mathcal{L_H}$): $\mathcal{H}$
begin
1. $\mathcal{H} := \emptyset$;
2. while $\mathcal{E}^+ \neq \emptyset$ do
3.     $\phi :=$ Learn-One-Axiom($\mathcal{K}$, $H$, $\mathcal{E}^+$, $\mathcal{E}^-$, $\mathcal{L_H}$);
4.     $\mathcal{H} := \mathcal{H} \cup \{\phi\}$;
5.     $\mathcal{E}^+_\phi := \{e \in \mathcal{E}^+ | \mathcal{K} \cup \phi \models e\}$;
6.     $\mathcal{E}^+ := \mathcal{E}^+ \setminus \mathcal{E}^+_\phi$;
7. endwhile
8. return $\mathcal{H}$
end
```

Fig. 2. Foil-$\mathcal{DL}$: Learning a set of GCI axioms.

Learn-One-Axiom (reported in Figure 3) starts with the most general axiom (*i.e.* $\top \sqsubseteq H$) and specializes it by applying the refinement rules implemented in the function Refine (step 7.). The iterated specialization of the axiom continues until the axiom does not cover any negative example and its *confidence degree* is greater than a fixed threshold (θ). The confidence degree of axioms being generated with Refine allows for evaluating the *information gain* obtained on each refinement step by calling the function Gain (step 9.).

Due to the peculiarities of the language of hypotheses in Foil-$\mathcal{DL}$, necessary changes are made to Foil as concerns the functions Refine and Gain. Details about these novel features are provided in the next two subsections.

The refinement operator. The function Refine implements a *specialization* operator with the following refinement rules:

Add_A**:** adds an atomic concept A
$Add_{\exists R.\top}$**:** adds a complex concept $\exists R.\top$ by existential role restriction
$Add_{\exists T.\mathbf{d}}$**:** adds a complex concept $\exists T.\mathbf{d}$ by existential role restriction
$Subst_A$**:** replaces an atomic concept A with another atomic concept A' s.t. $A' \sqsubseteq A$

At each refinement step (*i.e.* at each call of Refine), the rules are applied first to the left-hand side of the axiom being specialized and then recursively to the range of all the conjuncts defined with existential role restriction. For example, let us consider that H is the target concept, A, A', B, R, R', T are concepts and properties occurring in $\mathcal{K}$, and $A' \sqsubseteq A$ holds in $\mathcal{K}$. Under these assumptions, the axiom $\exists R.B \sqsubseteq H$ is specialized into the following axioms:

- $A \sqcap \exists R.B \sqsubseteq H$, $B \sqcap \exists R.B \sqsubseteq H$, $A' \sqcap \exists R.B \sqsubseteq H$;
- $\exists R'.\top \sqcap \exists R.B \sqsubseteq H$, $\exists T.\mathbf{d} \sqcap \exists R.B \sqsubseteq H$;
- $\exists R.(B \sqcap A) \sqsubseteq H$, $\exists R.(B \sqcap A') \sqsubseteq H$;
- $\exists R.(B \sqcap \exists R.\top) \sqsubseteq H$, $\exists R.(B \sqcap \exists R'.\top) \sqsubseteq H$, $\exists R.(B \sqcap \exists T.\mathbf{d}) \sqsubseteq H$.

The application of the refinement rules is not blind. It takes the background theory into account in order to avoid the generation of redundant or useless

```
**function** LEARN-ONE-AXIOM($\mathcal{K}$, $H$, $\mathcal{E}^+$, $\mathcal{E}^-$, $\mathcal{L}_\mathcal{H}$): $\phi$
**begin**
1. $B := \top$;
2. $\phi := B \sqsubseteq H$;
3. $\mathcal{E}^-_\phi := \mathcal{E}^-$;
4. **while** $cf(\phi) < \theta$ **or** $\mathcal{E}^-_\phi \neq \emptyset$ **do**
5.      $B_{best} := B$;
6.      $maxgain := 0$;
7.      $\Phi := \text{REFINE}(\phi, \mathcal{L}_\mathcal{H})$
8.      **foreach** $\phi' \in \Phi$ **do**
9.           $gain := \text{GAIN}(\phi', \phi)$;
10.          **if** $gain \geq maxgain$ **then**
11.               $maxgain := gain$;
12.               $B_{best} := B'$;
13.          **endif**
14.     **endforeach**
15.     $\phi := B_{best} \sqsubseteq H$;
16.     $\mathcal{E}^-_\phi := \{e \in \mathcal{E}^- | \mathcal{K} \cup \phi \models e\}$;
17. **endwhile**
18. **return** $\phi$
**end**
```

Fig. 3. FOIL-$\mathcal{DL}$: Learning one GCI axiom.

hypotheses. For example, if the concept B' is the range of R' in $\mathcal{K}$, the function REFINE adds the conjunct $\exists R'.B'$ instead of $\exists R'.\top$. One such "informed" refinement operator is able to perform "cautious" big steps in the search space.

Note that a specialization operator reduces the number of examples covered by a GCI. More precisely, the aim of a refinement step is to reduce the number of covered negative examples, while still keeping some covered positive examples. Since learned GCIs cover only positive examples, $\mathcal{K}$ will remain consistent after the addition of a learned GCI.

The heuristic. The function GAIN implements an information-theoretic criterion for selecting the best candidate at each refinement step according to the following formula:

$$\text{GAIN}(\phi', \phi) = p * (log_2(cf(\phi')) - log_2(cf(\phi))) \ , \tag{7}$$

where p is the number of positive examples covered by the axiom ϕ that are still covered by ϕ'. Thus, the gain is positive iff ϕ' is more informative in the sense of Shannon's information theory, *i.e.* iff the confidence degree (cf) increases. If there are some refinements, which increase the confidence degree, the function GAIN tends to favour those that offer the best compromise between the confidence degree and the number of examples covered. Here, cf for an axiom ϕ of the form (5) is computed as a sort of fuzzy set inclusion degree (see Eq. (1)) between the fuzzy set represented by concept B and the (crisp) set represented by concept H. More formally:

$$cf(\phi) = cf(B \sqsubseteq H) = \frac{\sum_{a \in \mathsf{Ind}^+_H(\mathcal{A})} bed(\mathcal{K}, a{:}B)}{|\mathsf{Ind}_H(\mathcal{A})|} \qquad (8)$$

where $\mathsf{Ind}^+_H(\mathcal{A})$ (resp., $\mathsf{Ind}_H(\mathcal{A})$) is the set of individuals occurring in $\mathcal{A}$ and involved in $\mathcal{E}^+_\phi$ (resp., $\mathcal{E}^+_\phi \cup \mathcal{E}^-_\phi$) such that $bed(\mathcal{K}, a{:}B) > 0$. We remind the reader that $bed(\mathcal{K}, a{:}B)$ denotes the best entailment degree of the concept assertion $a{:}B$ w.r.t. $\mathcal{K}$ as defined in Eq. (4). Note that for individuals $a \in \mathsf{Ind}^+_H(\mathcal{A})$, $\mathcal{K} \models a{:}H$ holds and, thus, $bed(\mathcal{K}, a{:}B \sqcap H) = bed(\mathcal{K}, a{:}B)$. Also, note that, even if $\mathcal{K}$ is crisp, the possible occurrence of fuzzy concrete domains in expressions of the form $\exists T.\mathbf{d}$ in B may imply that both $bed(\mathcal{K}, B \sqsubseteq H) \notin \{0, 1\}$ and $bed(\mathcal{K}, a{:}B) \notin \{0, 1\}$.

3.3 The implementation

A variant of FOIL-$\mathcal{DL}$ has been implemented in the *fuzzyDL-Learner* [7] system and provided with two GUIs: One is a stand-alone Java application, the other is a tab widget plug-in for the ontology editor Protégé [8] (release 4.2).

Several implementation choices have been made. Notably, fuzzy GCIs in $\mathcal{L}_\mathcal{H}$ are interpreted under Gödel semantics (see Eq. (2)). However, since $\mathcal{K}$ and $\mathcal{E}$ are represented in crisp DLs, we have used a classical DL reasoner, together with a specialised code, to compute the confidence degree of fuzzy GCIs. Therefore, the system relies on the services of DL reasoners to solve all the deductive inference problems necessary to FOIL-$\mathcal{DL}$ to work, namely instance retrieval, instance check and subclasses retrieval. In particular, the sets $\mathsf{Ind}^+_H(\mathcal{A})$ and $\mathsf{Ind}_H(\mathcal{A})$ are computed by posing instance retrieval problems to the DL reasoner. Conversely, $bed(\mathcal{K}, a{:}\exists T.\mathbf{d})$ can be computed from the derived T-fillers v of a, and applying the fuzzy membership function of $\mathbf{d}$ to v. The examples covered by a GCI, and, thus, the entailment tests in LEARN-SETS-OF-AXIOMS and LEARN-ONE-AXIOM, have been determined in a similar way.

The implementation of FOIL-$\mathcal{DL}$ features several optimizations w.r.t. the solution strategy presented in Section 3.2. Notably, the search in the hypothesis space can be optimized by enabling a backtracking mode. This option allows to overcome one of the main limits of FOIL, *i.e.* the sequential covering strategy. Because it performs a greedy search, formulating a sequence of rules without backtracking, FOIL does not guarantee to find the smallest or best set of rules that explain the training examples. Also, learning rules one by one could lead to less and less interesting rules. To reduce the risk of a suboptimal choice at any search step, the greedy search can be replaced in FOIL-$\mathcal{DL}$ by a *beam search* which maintains a list of k best candidates at each step instead of a single best candidate. Additionally, to guarantee termination, we provide two parameters to limit the search space: namely, the maximal number of conjuncts and the maximal depth of existential nesting allowed in a fuzzy GCI. In fact, the computation may end without covering all positive examples.

[7] `http://straccia.info/software/FuzzyDL-Learner`
[8] `http://protege.stanford.edu/`

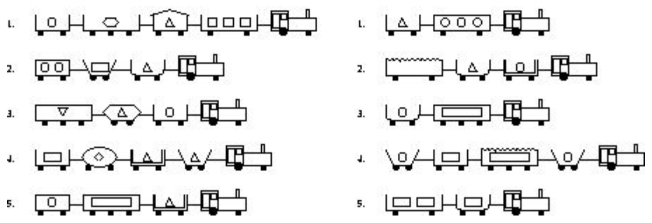


Fig. 4. Michalski's example of eastbound trains (left) and westbound trains (right) (illustration taken from [16]).

4 Comparing Foil-$\mathcal{DL}$ and DL-Learner

In this section we report the results of a comparison between FOIL-$\mathcal{DL}$ and DL-Learner on a very popular learning task in ILP proposed 20 years ago by Ryszard Michalski [16] and illustrated in Figure 4. Here, 10 trains are described, out of which 5 are eastbound and 5 are westbound. The aim of the learning problem is to find the discriminating features between these two classes.

For the purpose of this comparative study, we have considered two slightly different versions, *trains2* and *trains3*, of an ontology encoding the original Trains data set. [9] The former has been adapted from the version distributed with DL-Learner in order to be compatible with FOIL-$\mathcal{DL}$. Notably, the target classes `EastTrain` and `WestTrain` have become part of the terminology and several class assertion axioms have been added for representing positive and negative examples. The metrics for *trains2* are reported in Table 2. The ontology does not encompass any data property. Therefore, no fuzzy concept can be generated when learning GCIs from *trains2* with FOIL-$\mathcal{DL}$. However, the ontology can be slightly modified in order to test the fuzzy concept generation feature of FOIL-$\mathcal{DL}$. Note that in *trains2* cars can be classified according to the classes `LongCar` and `ShortCar`. Instead of one such crisp classification, we may want a fuzzy classification of cars. This is made possible by removing `LongCar` and `ShortCar` (together with the related class assertion axioms) from *trains2* and introducing the data property `hasLenght` with domain `Car` and range `double` (together with several data property assertions). The resulting ontology, called *trains3*, presents the metrics reported in Table 2.

DL-Learner [10] features several algorithms. Among them, the closest to FOIL-$\mathcal{DL}$ is ELTL since it implements a refinement operator for concept learning in $\mathcal{EL}$ [12]. Conversely, CELOE learns class expressions in the more expressive OWL DL [11]. Both work only under OWA and deal only with crisp DLs.

4.1 Results on the ontology *trains2*

Trial with FOIL-$\mathcal{DL}$. The settings for this experiment allow for the generation of hypotheses with up to 5 conjuncts and 2 levels of existential nestings. Under

[9] `http://archive.ics.uci.edu/ml/datasets/Trains`
[10] `http://dl-learner.org/Projects/DLLearner`

Table 2. Ontology metrics for *trains2.owl* and *trains3.owl* according to Protégé.

	# logical axioms	# classes	# object prop.	# data prop.	# individuals	DL expressivity
trains2	345	32	5	0	50	$\mathcal{ALCO}$
trains3	343	30	5	1	50	$\mathcal{ALCO}(\mathbf{D})$

these restrictions, the GCI axioms learned by FOIL-$\mathcal{DL}$ for the target concept `EastTrain` are:

```
Confidence Axiom
1,000      3CarTrain and hasCar some (2LoadCar) subclass of EastTrain
1,000      3CarTrain and hasCar some (3WheelsCar) subclass of EastTrain
1,000      hasCar some (ElipseShapeCar) subclass of EastTrain
1,000      hasCar some (HexagonLoadCar) subclass of EastTrain
```

whereas the following GCI axioms are returned by FOIL-$\mathcal{DL}$ for `WestTrain`:

```
Confidence Axiom
1,000      2CarTrain subclass of WestTrain
1,000      hasCar some (JaggedCar) subclass of WestTrain
```

The algorithm returns the same GCIs under both OWA and CWA. Note that an important difference between learning in DLs and standard ILP is that the former works under OWA whereas the latter under CWA. In order to complete the Trains example we would have to introduce definitions and/or assertions to model the closed world. However, the CWA holds naturally in this example, because we have complete knowledge of the world, and thus the knowledge completion was not necessary. This explains the behaviour of FOIL-$\mathcal{DL}$ which correctly induces the same hypotheses in spite of the opposite semantic assumptions.

Trial with ELTL. For the target concept `EastTrain`, the class expression learned by ELTL is the following :

```
EXISTS hasCar.(ClosedCar AND ShortCar) (accuracy: 1.0)
```

whereas the following finding has been returned for the target concept `WestTrain`:

```
EXISTS hasCar.LongCar (accuracy: 0.8)
```

The latter is not fully satisfactory as for the example coverage.

Trial with CELOE. For the target concept `EastTrain`, CELOE learns several class expressions of which the most accurate is:

```
hasCar some (ClosedCar and ShortCar) (accuracy: 1.0)
```

whereas, for the target concept `WestTrain`, the most accurate among the ones found is the following:

```
hasCar only (LongCar or OpenCar) (accuracy: 1.0)
```

Note that the former coincide with the corresponding result obtained with ELTL while the latter is a more accurate variant of the corresponding class expression returned by ELTL. The increase in example coverage is due to the augmented expressive power of the DL supported in CELOE.

```
- hasLenght_low: hasLenght, triangular(23.0,32.0,41.0)
- hasLenght_fair: hasLenght, triangular(32.0,41.0,50.0)
- hasLenght_high: hasLenght, triangular(41.0,50.0,59.0)
- hasLenght_veryhigh: hasLenght, rightShoulder(50.0,59.0)
- hasLenght_verylow: hasLenght, leftShoulder(23.0,32.0)
```

Fig. 5. Fuzzy concepts derived by FOIL-$\mathcal{DL}$ from the data property `hasLenght`.

4.2 Results on the ontology *trains3*

Trial with FOIL-$\mathcal{DL}$. The outcomes for the target concepts `EastTrain` and `WestTrain` remain unchanged when FOIL-$\mathcal{DL}$ is run on *trains3* with the same configuration of the first trial. Yet, fuzzy concepts are automatically generated by FOIL-$\mathcal{DL}$ from the data property `hasLenght` (see Figure 5). However, from the viewpoint of discriminant power, these concepts are weaker than the other crisp concepts occurring in the ontology. In order to make the fuzzy concepts emerge during the generation of hypotheses, we have appropriately biased the language of hypotheses. In particular, by enabling only the use of object and data properties in $\mathcal{L}_{\mathcal{H}}$, FOIL-$\mathcal{DL}$ returns the following axiom for `EastTrain`:

```
Confidence Axiom
1,000      hasCar some (hasLenght_fair) and hasCar some (hasLenght_veryhigh)
                and hasCar some (hasLenght_verylow) subclass of EastTrain
```

Conversely, for `WestTrain`, a lighter bias is sufficient to make fuzzy concepts appear in the learned axioms. In particular, by disabling the class `2CarTrain` in $\mathcal{L}_{\mathcal{H}}$, FOIL-$\mathcal{DL}$ returns the following axioms:

```
Confidence Axiom
1,000      hasCar some (2WheelsCar and 3LoadCar) and hasCar some (3LoadCar and CircleLoadCar)
          subclass of WestTrain
1,000      hasCar some (0LoadCar) subclass of WestTrain
1,000      hasCar some (JaggedCar) subclass of WestTrain
1,000      hasCar some (2LoadCar and hasLenght_high) subclass of WestTrain
1,000      hasCar some (ClosedCar and hasLenght_fair) subclass of WestTrain
```

Trial with ELTL. For the target class `EastTrain`, ELTL returns a class expression which leaves some positive example uncovered (incomplete hypothesis):

```
(EXISTS hasCar.TriangleLoadCar AND EXISTS hasCar.ClosedCar) (accuracy: 0.9)
```

whereas, for the target concept `WestTrain`, it returns an overly general hypothesis which covers also negative examples (inconsistent hypothesis):

```
TOP (accuracy: 0.5)
```

This bad performance of ELTL on *trains3* is due to the low expressivity of $\mathcal{EL}$ and to the fact that the classes `LongCar` and `ShortCar`, which appeared to be discriminant in the first trial, do not occur in *trains3* and thus can not be used anymore for building hypotheses.

Trial with CELOE. The most accurate class expression found by CELOE for the target concept `EastTrain` is:

```
((not 2CarTrain) and hasCar some ClosedCar) (accuracy: 1.0)
```

However, interestingly, CELOE learns also the following class expressions containing classes obtained by numerical restriction from the data property `hasLenght`:

```
hasCar some (ClosedCar and hasLenght <= 48.5) (accuracy: 1.0)
hasCar some (ClosedCar and hasLenght <= 40.5) (accuracy: 1.0)
hasCar some (ClosedCar and hasLenght <= 31.5) (accuracy: 1.0)
```

These "interval classes" are just a step back from the fuzzification which, conversely, FOIL-$\mathcal{DL}$ is able to do. It is acknowledged that using fuzzy sets in place of "intervall classes" improves the readability of the induced knowledge about the data. As for the target concept `WestTrain`, the most accurate class expression among the ones found by CELOE is:

```
(2CarTrain or hasCar some JaggedCar) (accuracy: 1.0)
```

Once again, the augmented expressivity increases the effectiveness of DL-Learner.

5 Conclusions and future work

We have described a novel method, named FOIL-$\mathcal{DL}$, which addresses the problem of learning fuzzy $\mathcal{EL}(\mathbf{D})$ GCI axioms from crisp $\mathcal{DL}$ assertions. The method extends FOIL in a twofold direction: from crisp to fuzzy and from rules to GCIs. Notably, vagueness is captured by the definition of confidence degree reported in (8) and incompleteness is dealt with the OWA. Also, thanks to the variable-free syntax of DLs, the learnable GCIs are highly understandable by humans and translate easily into natural language sentences. In particular, FOIL-$\mathcal{DL}$ present the learned axioms according to the user-friendly presentation style of the Manchester OWL syntax [11] (the same used in Protégé).

We would like to stress the fact that FOIL-$\mathcal{DL}$ provides a different solution from *Soft*FOIL [14] as for the KR framework, the refinement operator and the heuristic. Also, unlike *Soft*FOIL, FOIL-$\mathcal{DL}$ has been implemented and tested. The experimental results are quite promising and encourage the application of FOIL-$\mathcal{DL}$ to more challenging real-world problems. Notably, in spite of the low expressivity of $\mathcal{EL}$, FOIL-$\mathcal{DL}$ has turned out to be robust mainly due to the refinement operator and to the fuzzification facilities. Note that a fuzzy OWL 2 version of the trains' problem (ontology *fuzzytrains_v1.5.owl*) [12] has been developed by Iglesias for testing the fuzzy extension of CELOE proposed in [7]. However, FOIL-$\mathcal{DL}$ can not handle fuzzy OWL 2 constructs such as fuzzy classes obtained by existential restriction of fuzzy datatypes, fuzzy concept assertions, and fuzzy role assertions. Therefore, it has been necessary to prepare an *ad-hoc* ontology (*trains3*) for comparing FOIL-$\mathcal{DL}$ and DL-Learner.

For the future, we intend to conduct a more extensive empirical evaluation of FOIL-$\mathcal{DL}$, which could suggest directions of improvement of the method towards more effective formulations of, *e.g.*, the information gain function and the refinement operator as well as of the search strategy and the halt conditions employed in LEARN-ONE-AXIOM. Also, it can be interesting to analyse the impact of the different fuzzy logics on the learning process. Eventually, we shall investigate about learning fuzzy GCI axioms from FuzzyOWL 2 ontologies, by coupling the learning algorithm to the *fuzzyDL* reasoner, instead of learning from crisp OWL 2 data by using a classical DL reasoner.

[11] `http://www.w3.org/TR/owl2-manchester-syntax/`
[12] Available at `http://wiki.aksw.org/Projects/DLLearner/fuzzyTrains`.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook: Theory, Implementation and Applications (2nd ed.). Cambridge University Press (2007)
2. Bobillo, F., Straccia, U.: fuzzyDL: An expressive fuzzy description logic reasoner. In: 2008 Int. Conf. on Fuzzy Systems. pp. 923–930. IEEE Computer Society (2008)
3. Borgida, A.: On the relative expressiveness of description logics and predicate logics. Artificial Intelligence Journal 82, 353–367 (1996)
4. Cerami, M., Straccia, U.: On the (un)decidability of fuzzy description logics under Łukasiewicz t-norm. Information Sciences 227, 1–21 (2013)
5. 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)
6. Hájek, P.: Metamathematics of Fuzzy Logic. Kluwer (1998)
7. Iglesias, J., Lehmann, J.: Towards integrating fuzzy logic capabilities into an ontology-based inductive logic programming framework. In: Proc. of the 11th Int. Conf. on Intelligent Systems Design and Applications. IEEE Press (2011)
8. Klir, G.J., Yuan, B.: Fuzzy sets and fuzzy logic: theory and applications. Prentice-Hall, Inc., Upper Saddle River, NJ, USA (1995)
9. Konstantopoulos, S., Charalambidis, A.: Formulating description logic learning as an inductive logic programming task. In: Proc. of the 19th IEEE Int. Conf. on Fuzzy Systems. pp. 1–7. IEEE Press (2010)
10. Lehmann, J.: DL-Learner: Learning Concepts in Description Logics. Journal of Machine Learning Research 10, 2639–2642 (2009)
11. Lehmann, J., Auer, S., Bühmann, L., Tramp, S.: Class expression learning for ontology engineering. Journal of Web Semantics 9(1), 71–81 (2011)
12. Lehmann, J., Haase, C.: Ideal Downward Refinement in the $\mathcal{EL}$ Description Logic. In: De Raedt, L. (ed.) ILP 2009. Revised Papers. Lecture Notes in Computer Science, vol. 5989, pp. 73–87. Springer (2010)
13. Lisi, F.A.: A formal characterization of concept learning in description logics. In: Kazakov, Y., Lembo, D., Wolter, F. (eds.) Proc. of the 2012 Int. Workshop on Description Logics. CEUR Workshop Proceedings, vol. 846. CEUR-WS.org (2012)
14. Lisi, F.A., Straccia, U.: A logic-based computational method for the automated induction of fuzzy ontology axioms. Fundamenta Informaticae 124, 1–17 (2013)
15. Lukasiewicz, T., Straccia, U.: Managing Uncertainty and Vagueness in Description Logics for the Semantic Web. Journal of Web Semantics 6(4), 291–308 (2008)
16. Michalski, R.: Pattern recognition as a rule-guided inductive inference. IEEE Transactions on Pattern Analysis and Machine Intelligence 2(4), 349–361 (1980)
17. Motik, B., Rosati, R.: A faithful integration of description logics with logic programming. In: Veloso, M. (ed.) IJCAI 2007, Proc. of the 20th Int. Joint Conf. on Artificial Intelligence. pp. 477–482 (2007)
18. Quinlan, J.R.: Learning logical definitions from relations. Machine Learning 5, 239–266 (1990)
19. Reiter, R.: Equality and domain closure in first order databases. Journal of ACM 27, 235–249 (1980)
20. Straccia, U.: Reasoning within fuzzy description logics. Journal of Artificial Intelligence Research 14, 137–166 (2001)
21. Straccia, U.: Description logics with fuzzy concrete domains. In: Bachus, F., Jaakkola, T. (eds.) 21st Conf. on Uncertainty in Artificial Intelligence. pp. 559–567. AUAI Press (2005)