Fundamenta Informaticae XXI (2001) 1001–1016 1001
DOI 10.3233/FI-2012-0000
IOS Press

Learning in Description Logics with Fuzzy Concrete Domains

Francesca A. Lisi
Dipartimento di Informatica, Università degli Studi di Bari "Aldo Moro", Italy
francesca.lisi@uniba.it

Umberto Straccia
ISTI - CNR, Pisa, Italy
umberto.straccia@isti.cnr.it

Abstract. Description Logics (DLs) are a family of logic-based Knowledge Representation (KR) formalisms, which are particularly suitable for representing incomplete yet precise structured knowledge. Several fuzzy extensions of DLs have been proposed in the KR field in order to handle imprecise knowledge which is particularly pervading in those domains where entities could be better described in natural language. Among the many approaches to fuzzification in DLs, a simple yet interesting one involves the use of fuzzy concrete domains. In this paper, we present a method for learning within the KR framework of fuzzy DLs. The method induces fuzzy DL inclusion axioms from any crisp DL knowledge base. Notably, the induced axioms may contain fuzzy concepts automatically generated from numerical concrete domains during the learning process. We discuss the results obtained on a popular learning problem in comparison with state-of-the-art DL learning algorithms.

Keywords: Fuzzy Description Logics, Ontologies, Concept Learning.

1. Introduction

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*) [2]. 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. As logic-based formalisms for Knowledge Representation (KR) compliant with the *Open World Assumption* (OWA), they are particularly suitable for representing *incomplete* knowledge. The 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. It traditionally applies in databases and related KR settings such as Logic Programming (LP) and Inductive Logic Programming (ILP). Heuristically, the OWA applies when we represent knowledge

Address for correspondence: Campus Universitario "E. Quagliariello", Via E. Orabona 4, 70125 Bari, Italy

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. Thanks to the OWA-compliancy, DLs have been considered the ideal starting point for the definition of ontology languages for the Web (an inherently open world), giving raise to the OWL 2 standard. [1]

In many applications, it is important to equip DLs with expressive means that allow to describe "concrete qualities" of real-world objects such as the length of a car. The standard approach is to augment DLs with so-called *concrete domains*, which consist of a set (say, the set of real numbers in double precision) and a set of n-ary predicates (typically, $n = 1$) with a fixed extension over this set [3]. Starting from numerical properties such as the length one may want to deduce whether, *e.g.*, a car is long or not. However, it is well known that "classical" DLs are not appropriate to deal with *imprecise* (or *vague*) knowledge, which is inherent to several real world domains and is particularly pervading in those domains where entities could be better described in natural language [27]. Vagueness is traditionally captured with *fuzzy logic*. We recall that 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). [2]

Although a relatively important amount of work has been carried out in the last years concerning the use of fuzzy DLs as ontology languages [19, 29], the problem of automatically managing the evolution of fuzzy ontologies by applying machine learning algorithms still remains relatively unaddressed [13, 11, 18]. In this paper, we present a novel method, named FOIL-$\mathcal{DL}$, for learning fuzzy DL inclusion axioms from any crisp DL knowledge base.[3] The distinguishing feature of FOIL-$\mathcal{DL}$ w.r.t. previous work in DL learning (see, *e.g.*, [9, 16, 15]) is the treatment of numerical concrete domains with fuzzification techniques so that the induced axioms may contain fuzzy concepts.

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 discusses related work and illustrates the results of a comparative evaluation with state-of-the-art DL learning algorithms on the popular ILP problem of Michalski's trains. Section 5 concludes the paper with final remarks and outlines possible directions of future work.

2. Preliminaries

Description Logics. For the sake of illustrative purposes, we present here a salient representative of the DL family, namely $\mathcal{ALC}$ [24], which is often considered to illustrate some new notions related to DLs. The set of constructors for $\mathcal{ALC}$ 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 (*i.e.*, the intensional level of knowledge), and a populated ontology corresponds to a whole KB (*i.e.*, encompassing also an ABox, that

[1] `http://www.w3.org/TR/2009/REC-owl2-overview-20091027/`

[2] For a clarification about the differences between uncertainty and vagueness see [8].

[3] The present paper deepens some of the issues treated in [17].

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

bottom (resp. top) concept	$\bot$ (resp. $\top$)	$\emptyset$ (resp. $\Delta^{\mathcal{I}}$)
atomic concept	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
(abstract) role	R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
individual	a	$a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
concept intersection	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
concept union	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
concept negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
universal role restriction	$\forall R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \forall y\ (x,y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$
existential role 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	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
concept assertion	$a : C$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
role assertion	$(a,b) : R$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$

is, the extensional level of knowledge). We also introduce two well-known DL macros, namely (i) *domain restriction*, denoted *domain*(R, A), which is a macro for the GCI $\exists R.\top \sqsubseteq A$, and states that the domain of the abstract role R is the atomic concept A; and (ii) *range restriction*, denoted *range*(R, A), which is a macro for the GCI $\top \sqsubseteq \forall R.A$, and states that the range of R is A. Finally, in $\mathcal{ALC}(\mathbf{D})$ (obtained by enriching $\mathcal{ALC}$ with concrete domains $\mathbf{D}$), each role is either *abstract* (denoted with R) or *concrete* (denoted with T). A new concept constructor is then introduced, which allows to describe constraints on concrete values using predicates from the concrete domain. We shall make further clarifications about the notion of concrete domains later on in this Section while presenting fuzzy $\mathcal{ALC}(\mathbf{D})$.

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 [5]). Specifically, 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 $C^{\mathcal{I}} : \Delta^{\mathcal{I}} \rightarrow \{0, 1\}$ (either an individual belongs to the extension of C or does not belong to it). Under the *Unique Names Assumption* (UNA) [23], 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. We also write $C \sqsubseteq_{\mathcal{K}} D$ if in any model $\mathcal{I}$ of $\mathcal{K}$, $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ (concept C is subsumed by concept D). Moreover we write $C \sqsubset_{\mathcal{K}} D$ if $C \sqsubseteq_{\mathcal{K}} D$ and $D \not\sqsubseteq_{\mathcal{K}} C$.

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 checking*, i.e., to check whether an ABox assertion is a logical consequence of a DL KB. A more sophisticated version of instance checking, 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, denoted $\{a \mid \mathcal{K} \models a{:}C\}$.

Mathematical Fuzzy Logic. *Fuzzy Logic* is the logic of fuzzy sets [30]. A *fuzzy set* A over a countable crisp set X is characterised by a function $A\colon X \rightarrow [0, 1]$. Unlike crisp sets that can be characterised by a function $A\colon X \rightarrow \{0, 1\}$, that is, for any $x \in X$ either $x \in A$ (*i.e.*, $A(x) = 1$) or $x \notin A$ (*i.e.*, $A(x) = 0$), for a fuzzy set A, $A(x)$ dictates that $x \in X$ belongs to the set A to a degree in $[0, 1]$. The classical set operations of intersection, union and complementation naturally extend to fuzzy sets as follows. Let A and B be two fuzzy sets. The standard fuzzy set operations are $(A \cap B)(x) = \min(A(x), B(x))$,

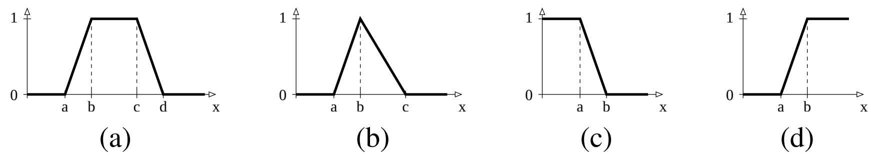

Figure 1. (a) Trapezoidal function $trz(a,b,c,d)$, (b) triangular function $tri(a,b,c)$, (c) left-shoulder function $ls(a,b)$, and (d) right-shoulder function $rs(a,b)$.

$(A \cup B)(x) = \max(A(x), B(x))$ and $\bar{A}(x) = 1 - A(x)$, while the *inclusion degree* between A and B is typically defined as

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

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 functions to specify *membership functions* of fuzzy sets. Although fuzzy sets have a greater expressive power than classical crisp sets, their usefulness depends 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.*, [12, 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.

While classical logic is based on crisp set theory, *Mathematical Fuzzy Logic* (MFL) [10] is based on generalised fuzzy set theory. Specifically, in MFL 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* 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 as follows:

$$\begin{array}{lll} \mathcal{I}(\phi \land \psi) = \mathcal{I}(\phi) \otimes \mathcal{I}(\psi) & \mathcal{I}(\phi \to \psi) = \mathcal{I}(\phi) \Rightarrow \mathcal{I}(\psi) & \mathcal{I}(\exists x.\phi(x)) = \sup_{y \in \Delta^{\mathcal{I}}} \mathcal{I}(\phi(y)) \\ \mathcal{I}(\phi \lor \psi) = \mathcal{I}(\phi) \oplus \mathcal{I}(\psi) & \mathcal{I}(\neg \phi) = \ominus \mathcal{I}(\phi) & \mathcal{I}(\forall x.\phi(x)) = \inf_{y \in \Delta^{\mathcal{I}}} \mathcal{I}(\phi(y)), \end{array} \tag{2}$$

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

Table 2. Combination functions of various fuzzy logics.

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

Table 3. Syntax and semantics of constructs for fuzzy $\mathcal{ALC}(\mathbf{D})$.

bottom (resp. top) concept	$\bot^{\mathcal{I}}(x) = 0$ (resp. $\top^{\mathcal{I}}(x) = 1$)
atomic concept	$A^{\mathcal{I}}(x) \in [0,1]$
abstract role	$R^{\mathcal{I}}(x,y) \in [0,1]$
concrete role	$T^{\mathcal{I}}(x,z) \in [0,1]$
individual	$a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
concrete value	$v^{\mathcal{I}} \in \Delta^{\mathbf{D}}$
concept intersection	$(C \sqcap D)^{\mathcal{I}}(x) = C^{\mathcal{I}}(x) \otimes D^{\mathcal{I}}(x)$
concept union	$(C \sqcup D)^{\mathcal{I}}(x) = C^{\mathcal{I}}(x) \oplus D^{\mathcal{I}}(x)$
concept negation	$(\neg C)^{\mathcal{I}}(x) = \ominus C^{\mathcal{I}}(x)$
concept implication	$(C \to D)^{\mathcal{I}}(x) = C^{\mathcal{I}}(x) \Rightarrow D^{\mathcal{I}}(x)$
universal abstract role restriction	$(\forall R.C)^{\mathcal{I}}(x) = \inf_{y \in \Delta^{\mathcal{I}}} \{R^{\mathcal{I}}(x,y) \Rightarrow C^{\mathcal{I}}(y)\}$
existential abstract role restriction	$(\exists R.C)^{\mathcal{I}}(x) = \sup_{y \in \Delta^{\mathcal{I}}} \{R^{\mathcal{I}}(x,y) \otimes C^{\mathcal{I}}(y)\}$
universal concrete role restriction	$(\forall T.\mathbf{d})^{\mathcal{I}}(x) = \inf_{z \in \Delta^{\mathbf{D}}} \{T^{\mathcal{I}}(x,z) \Rightarrow \mathbf{d}^{\mathbf{D}}(z)\}$
existential concrete role restriction	$(\exists T.\mathbf{d})^{\mathcal{I}}(x) = \sup_{z \in \Delta^{\mathbf{D}}} \{T^{\mathcal{I}}(x,z) \otimes \mathbf{d}^{\mathbf{D}}(z)\}$
general concept inclusion	$(C \sqsubseteq D)^{\mathcal{I}} = \inf_{x \in \Delta^{\mathcal{I}}} \{C^{\mathcal{I}}(x) \Rightarrow D^{\mathcal{I}}(x)\}$
concept assertion	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
abstract role assertion	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$
concrete role assertion	$(a^{\mathcal{I}}, v^{\mathcal{I}}) \in T^{\mathcal{I}}$

Łukasiewicz, Gödel, and Product logics [10], whose combination functions are reported in Table 2. Note that any other continuous t-norm can be obtained from them (see, *e.g.* [10]).

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 $\mathcal{I}(\phi) \geq \alpha$.

Description Logics with Fuzzy Concrete Domains. We recap here the syntactic features of the fuzzy DL obtained by extending $\mathcal{ALC}$ with fuzzy concrete domains [28]. 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 $\mathbf{d}$ 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 are

$$\mathbf{d} \;\; := \;\; ls(a,b) \mid rs(a,b) \mid tri(a,b,c) \mid trz(a,b,c,d) \mid \; \geq_v \; \mid \; \leq_v \; \mid \; =_v \; , \tag{3}$$

where *e.g.* $\geq_v$ corresponds to the crisp set of data values that are greater or equal than the value v.

In fuzzy DLs, an *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consist of a nonempty (crisp) set $\Delta^{\mathcal{I}}$ (the *domain*) and of a *fuzzy interpretation function* $\cdot^{\mathcal{I}}$ that, *e.g.*, maps a concept 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. The definition of $\cdot^{\mathcal{I}}$ for $\mathcal{ALC}(\mathbf{D})$ with fuzzy concrete domains is reported in Table 3 (where $x, y \in \Delta^{\mathcal{I}}$ and $z \in \Delta^{\mathbf{D}}$). Note that the truth degrees vary according to the chosen fuzzy logic, *i.e.* to its set of combination functions.

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. An interpretation $\mathcal{I}$ *satisfies* an axiom $\langle \tau, \alpha \rangle$ if $(\tau)^{\mathcal{I}} \geq \alpha$. $\mathcal{I}$ 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

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

For a crisp axiom τ, we also write $\mathcal{K} \models_+ \tau$ iff $bed(\mathcal{K}, \tau) > 0$, *i.e.* τ is entailed to some degree $\alpha > 0$.

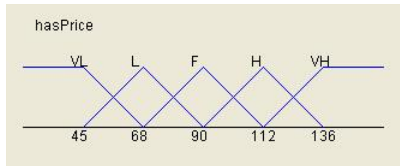

Figure 2. Fuzzy sets derived from the concrete domain used as range of the concrete role `hasPrice` in Example 2.1: `VeryLow` (VL), `Low` (L), `Fair` (F), `High` (H), and `VeryHigh` (VH).

Example 2.1. *Let us consider the following fuzzy GCI axiom* $\langle \exists \texttt{hasPrice}.\texttt{High} \sqsubseteq \texttt{GoodHotel}, 0.569\rangle$, *where* `hasPrice` *is a concrete role whose range has values measured in euros and the price concrete domain has been automatically fuzzified as follows. The partition into 5 fuzzy sets (*`VeryLow`*,* `Low`*,* `Fair`*,* `High`*, and* `VeryHigh`*) is obtained by considering the interval defined by minimal and maximal hotel prices (resp. 45 and 136), and then by splitting it into four equal subintervals on which three triangular functions, a left-shoulder and a right-shoulder function are built as illustrated in Figure 2. In particular, the membership function underlying the fuzzy set* `High` *is* $tri(90, 112, 136)$.

Now, let us suppose that the room price for hotel `verdi` *is* 105 *euro,* i.e. *the KB contains the assertion* $\texttt{verdi}{:}\exists\texttt{hasPrice}.=_{105}$. *It can be verified under Product logic that* $\mathcal{K} \models \langle \texttt{verdi}{:}\texttt{GoodHotel}, 0.18\rangle$, i.e. *hotel* `verdi` *is an instance of the class* `GoodHotel` *with a truth degree* 0.18.[4]

3. Learning fuzzy $\mathcal{EL}(\mathrm{D})$ axioms

3.1. The problem statement

The problem considered in this paper and solved by FOIL-$\mathcal{DL}$ is the automated induction of fuzzy $\mathcal{EL}(\mathbf{D})$ [5] GCI axioms from a crisp $\mathcal{DL}$ [6] KB and crisp examples, which provide a sufficient condition for a given atomic target concept A_t. It can be cast as a rule learning problem, provided that positive and negative examples of A_t are available. This problem can be formalized as follows. Given: (i) a consistent crisp $\mathcal{DL}$ KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ (the *background theory*); (ii) an atomic concept A_t (the *target concept*); (iii) a set $\mathcal{E} = \mathcal{E}^+ \cup \mathcal{E}^-$ of crisp concept assertions e labelled as either positive or negative examples for A_t (the *training set*); and (iv) 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 $\mathcal{H}$ satisfies the following conditions: $\forall e \in \mathcal{E}^+, \mathcal{K} \cup \mathcal{H} \models_+ e$ (completeness), and $\forall e \in \mathcal{E}^-, \mathcal{K} \cup \mathcal{H} \not\models_+ e$ (consistency).

Remark 3.1. *In the above problem statement we assume that* $\mathcal{K} \cap \mathcal{E} = \emptyset$. *Please observe that 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* $\mathcal{K} \cup \mathcal{H} \not\models_+ a{:}\bot$, *which means that* $\mathcal{K} \cup \mathcal{H}$ *is a consistent theory,* i.e. *has a model, that is, adding the learned axioms to the KB keeps the KB consistent.*

The background theory. A DL KB allows for the specification of very rich background knowledge in the form of axioms, *e.g.* defining the range of roles or the concept subsumption hierarchy. We do not make any specific assumption about the DL which the language $\mathcal{L}_{\mathcal{K}}$ of the background theory is based on, except that $\mathcal{K}$ is a crisp KB. However, since $\mathcal{H}$ is a set of fuzzy GCI axioms, $\mathcal{K} \cup \mathcal{H}$ is fuzzy as well.

The language of hypotheses. The language $\mathcal{L}_{\mathcal{H}}$ is given implicitly by means of syntactic restrictions over a given alphabet, as usual in ILP. In particular, the alphabet underlying $\mathcal{L}_{\mathcal{H}}$ is a subset of the alphabet for

[4] Note that $0.18 = 0.318 \cdot 0.569$, where $0.318 = tri(90, 112, 136)(105)$.
[5] $\mathcal{EL}(\mathbf{D})$ is a fragment of $\mathcal{ALC}(\mathbf{D})$ [29].
[6] $\mathcal{DL}$ stands for any DL.

function LEARN-SETS-OF-AXIOMS($\mathcal{K}$, A_t, $\mathcal{E}^+$, $\mathcal{E}^-$, $\mathcal{L}_\mathcal{H}$): $\mathcal{H}$
begin
1. $\mathcal{H} := \emptyset$; $\mathbf{D}$ = INITIALISEFUZZYCONCRETEDOMAIN($\mathcal{K}$);
2. **while** $\mathcal{E}^+ \neq \emptyset$ **do**
3. $\phi :=$ LEARN-ONE-AXIOM($\mathcal{K}$, A_t, $\mathcal{E}^+$, $\mathcal{E}^-$, $\mathcal{L}_\mathcal{H}$);
4. $\mathcal{H} := \mathcal{H} \cup \{\phi\}$;
5. $\mathcal{E}^+_\phi := \{e \in \mathcal{E}^+ \mid \mathcal{K} \cup \{\phi\} \models_+ e\}$;
6. $\mathcal{E}^+ := \mathcal{E}^+ \setminus \mathcal{E}^+_\phi$;
7. **endwhile**
8. **return** $\mathcal{H}$
end

Figure 3. FOIL-$\mathcal{DL}$: Learning a set of GCI axioms.

$\mathcal{L}_\mathcal{K}$. However, $\mathcal{L}_\mathcal{H}$ differs from $\mathcal{L}_\mathcal{K}$ as for the form of axioms. More precisely, given the target concept A_t, the hypotheses to be induced are fuzzy GCIs of the form

$$C \sqsubseteq A_t \ , \tag{5}$$

where the left-hand side is defined according to the following syntax

$$C \quad \longrightarrow \quad \top \mid A \mid C_1 \sqcap C_2 \mid \exists R.C \mid \exists T.\mathbf{d} \ . \tag{6}$$

and the concrete domain predicates are the following ones

$$\mathbf{d} \quad := \quad ls(a,b) \mid rs(a,b) \mid tri(a,b,c) \ . \tag{7}$$

Note that the syntactic restrictions of Eq. (6) w.r.t. fuzzy $\mathcal{ALC}(\mathbf{D})$ (see Table 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 A_t", which corresponds to the usual pattern in fuzzy rule induction (in our case, $C \sqsubseteq A_t$ is seen as a rule "if C then A_t"). Also, the restriction of Eq. (7) w.r.t. Eq. (3) is due to the fact that we build fuzzy concrete domain predicates out of numerical data as illustrated in Example 2.1.

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 cf that is obtained by means of the function in Eq. (12).

The training examples. Given the target concept A_t, the training set $\mathcal{E}$ consists of concept assertions of the form $a{:}A_t$, where a is an individual occurring in $\mathcal{K}$. Note that $\mathcal{E}$ 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 $\neg A_t$. On the other hand, under CWA, $\mathcal{E}^-$ is the collection of individuals, which cannot be proved to be instance of A_t. We say that an axiom $\phi \in \mathcal{L}_\mathcal{H}$ *covers* an example $e \in \mathcal{E}$ iff $\mathcal{K} \cup \{\phi\} \models_+ e$.

3.2. The solution strategy

The popular rule induction method FOIL [22] has been chosen as a starting point in our proposal for its simplicity and efficiency. 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 3) 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 LEARN-ONE-AXIOM (reported in Figure 4) starts with the most general axiom (*i.e.* $\top \sqsubseteq A_t$) and

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

Figure 4. FOIL-$\mathcal{DL}$: Learning one GCI axiom.

refines it by calling the function SPECIALIZE (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 SPECIALIZE 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 both candidate generation and evaluation. These novel features impact the definition of the functions SPECIALIZE and GAIN as detailed in Section 3.2.2 and Section 3.2.3, respectively. A pre-processing phase (see the function INITIALISEFUZZYCONCRETEDOMAIN called at step 1. of LEARN-SETS-OF-AXIOMS) is also required in order to generate the fuzzy datatypes to be used during the candidate generation phase. The fuzzification method is shortly described in the next subsection.

3.2.1. The function INITIALISEFUZZYCONCRETEDOMAIN

Given a crisp $\mathcal{DL}$ KB $\mathcal{K}$, the function INITIALISEFUZZYCONCRETEDOMAIN behaves as follows: For each concrete role T occurring in $\mathcal{K}$,

1. determine the minimal and maximal value that T entails according to $\mathcal{K}$, that is $min_T = \min\{v \mid \mathcal{K} \models a{:}\exists T.\leq_v\}$ and $max_T = \max\{v \mid \mathcal{K} \models a{:}\exists T.\geq_v\}$;

2. partition the interval $[min_T, max_T]$ into four uniform subintervals and, for $k = (max_T - min_T)/4$, build the fuzzy concrete domain predicates (note that $max_T = min_T + 4k$): $VeryLow_T = ls(min_T, min_T + k)$, $Low_T = tri(min_T, min_T + k, min_T + 2k)$, $Fair_T = tri(min_T + k, min_T + 2k, min_T + 3k)$, $High_T = tri(min_T + 2k, min_T + 3k, max_T)$ and $VeryHigh_T = rs(min_T + 3k, max_T)$.

Eventually, the function returns the set of all built fuzzy datatype predicates

$$\mathbf{D} = \bigcup_{T \text{ concrete role occurring in } \mathcal{K}} \{VeryLow_T, Low_T, Fair_T, High_T, VeryHigh_T\}$$

The method has been illustrated in Example 2.1.

3.2.2. The function SPECIALIZE

In line with the tradition in ILP and in conformance with the search direction in FOIL-$\mathcal{DL}$, the function SPECIALIZE implements a *downward refinement* operator $\rho_\mathcal{K}$ which actually exploits the background theory $\mathcal{K}$ in order to avoid the generation of redundant or useless hypotheses:

$$\textsc{Specialize}(\phi, \mathcal{L}_\mathcal{H}, \mathcal{K}) = \{\phi' \in \mathcal{L}_\mathcal{H} \mid \phi' \in \rho_\mathcal{K}(\phi)\} \ . \tag{8}$$

The refinement operator $\rho_\mathcal{K}$ acts only upon the left-hand-side of a GCI:

$$\rho_\mathcal{K}(\phi) = \rho_\mathcal{K}(C \sqsubseteq A_t) = \{C' \sqsubseteq A_t \mid C' \in \rho^\mathcal{C}_\mathcal{K}(C)\} \tag{9}$$

by either adding a new conjunct or replacing an already existing conjunct with a more specific one. More formally, the refinement rules for $\mathcal{EL}(\mathbf{D})$ concepts are defined as follows (here $\mathbf{d}_T$ is one of the fuzzy datatypes for concrete role T build by means of the INITIALISEFUZZYCONCRETEDOMAIN function, while A, B, D and E are atomic concepts, R is an abstract role):

$$\rho^\mathcal{C}_\mathcal{K}(C) = \begin{cases} \{A\} \cup \{\exists R.\top\} \cup \{\exists R.B \mid \mathit{range}(R,B) \in \mathcal{T}\} \cup \{\exists T.\mathbf{d}_T\} & \text{if } C = \top \\ \{A \sqcap D \mid D \in \rho^\mathcal{C}_\mathcal{K}(\top)\} \cup \{B \mid B \sqsubset_\mathcal{K} A\} & \text{if } C = A \\ \{\exists R.E \mid E \in \rho^\mathcal{C}_\mathcal{K}(D)\} \cup \{\exists R.(D \sqcap E) \mid E \in \rho^\mathcal{C}_\mathcal{K}(\top)\} & \text{if } C = \exists R.D \\ \{\exists T.\mathbf{d} \sqcap D \mid D \in \rho^\mathcal{C}_\mathcal{K}(\top)\} & \text{if } C = \exists T.\mathbf{d}_T \\ \{C_1 \sqcap \cdots \sqcap C_{i-1} \sqcap D \sqcap C_{i+1} \sqcap \cdots \sqcap C_n \mid D \in \rho^\mathcal{C}_\mathcal{K}(C_i), 1 \leq i \leq n\} & \text{if } C =_{i=1}^n C_i \end{cases} \tag{10}$$

Note that the use of relevant knowledge from $\mathcal{K}$ such as range axioms and concept subsumption axioms makes $\rho^\mathcal{C}_\mathcal{K}$ an "informed" refinement operator. Indeed, its refinement rules combine the syntactic manipulation with the semantic one. Also, this allows the operator to perform "cautious" big steps in the search space. More precisely, the less blind the rules are, the bigger the steps. $\rho^\mathcal{C}_\mathcal{K}$ also incorporates, in our implementation, a series of simplifications of the concepts built such as

$$\begin{array}{rcl} C \sqcap C & \mapsto & C \\ C \sqcap D \text{ and } D \sqsubseteq_\mathcal{K} C & \mapsto & D \\ C \sqcap D \text{ and } C \sqcap D \sqsubseteq_\mathcal{K} \bot & \mapsto & \bot \text{ (in this case we drop the refinement)} \end{array}$$

to reduce the search space. We are not going to detail them here.

Example 3.2. *Let us consider that A_t is the target concept, A, A', B, R, R', T are concepts and properties occurring in $\mathcal{K}$, and $A' \sqsubseteq_\mathcal{K} A$. Under these assumptions, the axiom $\exists R.B \sqsubseteq A_t$ is specialised into the following axioms:*

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

*Note that in the above list, $\mathbf{d}_T$ has to be instantiated for any of the five candidates for concrete role T (*i.e.*, $VeryLow_T, Low_T, Fair_T, High_T, VeryHigh_T$).*

It can be verified that $\rho^\mathcal{C}_\mathcal{K}$ is correct, *i.e.* it drives the search towards more specific concepts according to $\sqsubseteq$. Please note that $\rho_\mathcal{K}$ 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 GCIs.

3.2.3. The function GAIN

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

$$\text{GAIN}(\phi', \phi) = p * (log_2(cf(\phi')) - log_2(cf(\phi))) \;, \tag{11}$$

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 C and the (crisp) set represented by concept A_t. More formally:

$$cf(\phi) = cf(C \sqsubseteq D) = \frac{\sum_{a \in \mathsf{Ind}^+_D(\mathcal{A})} bed(\mathcal{K}, a{:}C)}{|\mathsf{Ind}_D(\mathcal{A})|} \tag{12}$$

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

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 as detailed below.

Reasoning support. Fuzzy GCIs in $\mathcal{L}_\mathcal{H}$ are interpreted under Gödel semantics (see Table 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 crisp 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 $\mathsf{Ind}^+_D(\mathcal{A})$ and $\mathsf{Ind}_D(\mathcal{A})$ are computed by posing instance retrieval problems to the DL reasoner. Conversely, $bed(\mathcal{K}, a{:}\exists T.\mathbf{d}_T)$ can be computed from the derived T-fillers v of a, and applying the fuzzy membership function of $\mathbf{d}_T$ 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 system can be configured to work under both CWA and OWA.

Optimizations. 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

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

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.

Declarative bias. The language of hypotheses can be biased by imposing the use of only "direct" subclasses. Additionally, FOIL-$\mathcal{DL}$ provides two parameters to limit the search space and guarantee termination: 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.

4. A comparative study

4.1. Related work

Several extensions of FOIL to the management of vague knowledge are reported in the literature [7, 25, 26] but they are not conceived for DL ontologies. In DL learning, DL-FOIL [9] adapts FOIL to learn crisp OWL DL equivalence axioms.[9] DL-Learner [14] is a state-of-the-art system which features several algorithms, none of which however is based on FOIL. Yet, among them, the closest to FOIL-$\mathcal{DL}$ is ELTL since it implements a refinement operator for concept learning in $\mathcal{EL}$ [16]. Conversely, CELOE learns class expressions in the more expressive OWL DL [15]. Both DL-FOIL, ELTL and CELOE work only under OWA and deal only with crisp DLs. Learning in fuzzy DLs has been little investigated. Konstantopoulos and Charalambidis [13] 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 [21] and entailment in Łukasiewicz $\mathcal{ALC}$ is undecidable [6]. Iglesias and Lehmann [11] propose an extension of DL-Learner with some of the most up-to-date fuzzy ontology tools, e.g. the *fuzzyDL* reasoner [4]. Notably, the resulting system can learn fuzzy OWL DL equivalence axioms from FuzzyOWL 2 ontologies. [10] 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 [18] present *Soft*FOIL, a FOIL-like method for learning fuzzy $\mathcal{EL}$ inclusion axioms from fuzzy DL-Lite$_{core}$ ontologies (a fuzzy variant of the classical DL, DL-Lite$_{core}$ [1]). We would like to stress the fact that FOIL-$\mathcal{DL}$ provides a different solution from *Soft*FOIL not only as for the KR framework but also as for the refinement operator and the heuristic. Also, unlike *Soft*FOIL, FOIL-$\mathcal{DL}$ has been implemented and tested.

4.2. FOIL-$\mathcal{DL}$ vs DL-Learner

In this section we report the results of a comparison of FOIL-$\mathcal{DL}$ with ELTL and CELOE (available in DL-Learner [11]) on a very popular learning task in ILP proposed 20 years ago by Ryszard Michalski [20] and illustrated in Figure 5. 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. [12] 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 4. The ontology does not encompass any data property. Therefore, no fuzzy concept can be generated

[9]The implementation of DL-FOIL was not made available by the authors.

[10]`http://www.straccia.info/software/FuzzyOWL`

[11]`http://dl-learner.org/Projects/DLLearner`

[12]`http://archive.ics.uci.edu/ml/datasets/Trains`

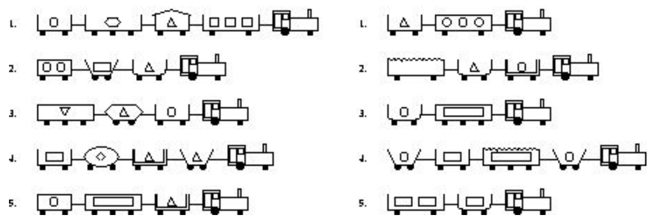

Figure 5. Michalski's example of eastbound (left) and westbound (right) trains (illustration taken from [20]).

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 `hasLength` with domain `Car` and range `double` (together with several data property assertions). The resulting ontology, called *trains3*, presents the metrics reported in Table 4.

Table 4. Ontology metrics for *trains2.owl* and *trains3.owl* according to Protégé.

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

Note that a fuzzy OWL 2 version of the trains' problem (ontology *fuzzytrains_v1.5.owl*) [13] has been developed by Iglesias for testing the fuzzy extension of CELOE proposed in [11]. 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.

4.2.1. Qualitative analysis of 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 these restrictions, the GCIs 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 GCIs are returned by FOIL-$\mathcal{DL}$ for `WestTrain`:

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

[13] Available at `http://wiki.aksw.org/Projects/DLLearner/fuzzyTrains`.

```
- hasLength_low: hasLength, triangular(23.0,32.0,41.0)
- hasLength_fair: hasLength, triangular(32.0,41.0,50.0)
- hasLength_high: hasLength, triangular(41.0,50.0,59.0)
- hasLength_veryhigh: hasLength, rightShoulder(50.0,59.0)
- hasLength_verylow: hasLength, leftShoulder(23.0,32.0)
```

Figure 6. Fuzzy concepts derived by FOIL-$\mathcal{DL}$ from the data property `hasLength`.

The algorithm returns the same GCIs under both OWA and CWA. Note that an important difference between DL learning 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. The class expressions learned by ELTL are the following:

```
EastTrain: hasCar some (ClosedCar and ShortCar) (accuracy: 1.0)
WestTrain: hasCar some LongCar (accuracy: 0.8)
```

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

Trial with CELOE. For each target class CELOE learns several class expressions out of which the most accurate are the following:

```
EastTrain: hasCar some (ClosedCar and ShortCar) (accuracy: 1.0)
WestTrain: 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.

4.2.2. Qualitative analysis of 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 `hasLength` (see Figure 6). 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 (hasLength_fair) and hasCar some (hasLength_veryhigh)
                and hasCar some (hasLength_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 hasLength_high) subclass of WestTrain
1,000      hasCar some (ClosedCar and hasLength_fair) subclass of WestTrain
```

Trial with ELTL. ELTL returns the following class expressions:

```
EastTrain: (hasCar some TriangleLoadCar) and (hasCar some ClosedCar) (accuracy: 0.9)
WestTrain: TOP (accuracy: 0.5)
```

Note that the class expression learned for `EastTrain` leaves some positive example uncovered (incomplete hypothesis) whereas the one induced for `WestTrain`, being overly general, covers also negative examples (inconsistent hypothesis). 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 `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 `hasLength`:

```
hasCar some (ClosedCar and hasLength <= 48.5) (accuracy: 1.0)
hasCar some (ClosedCar and hasLength <= 40.5) (accuracy: 1.0)
hasCar some (ClosedCar and hasLength <= 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.

4.2.3. Quantitative analysis of results

Some indicators of time performance of the three algorithms for the two learning problems on *trains2* and *trains3* are reported in Table 5 and Table 6, respectively. Notably, FOIL-$\mathcal{DL}$ outperforms CELOE in 3 out of the four cases. The bad time performance of FOIL-$\mathcal{DL}$ for the `EastTrain` learning problem on *trains3* is due to the computational overhead of satisfying the many constraints imposed on the language of hypotheses in this case.

Table 5. Time performance (in ms) on *trains2*.

	`EastTrain`			`WestTrain`		
algorithm	instance retrieval	instance checking	total	instance retrieval	instance checking	total
FOIL-$\mathcal{DL}$	73	203	649	22	53	230
ELTL	2	13	62	2	315	1,089
CELOE	2	1,584	10,000	3	3,440	10,000

Table 6. Time performance (in ms) on *trains3*.

	`EastTrain`			`WestTrain`		
algorithm	instance retrieval	instance checking	total	instance retrieval	instance checking	total
FOIL-$\mathcal{DL}$	5,000	8,000	19,093	347	1,000	3,681
ELTL	4	350	1,005	5	429	1,028
CELOE	5	2,665	10,000	5	3,130	10,000

5. Conclusions

We have described a novel method, named FOIL-$\mathcal{DL}$, which addresses the problem of learning fuzzy $\mathcal{EL}(\mathbf{D})$ GCI axioms from any crisp $\mathcal{DL}$ KB. 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 (12) 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 [14] (the same used in Protégé).

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. A distinguishing feature of $\rho_{\mathcal{K}}$ is that it exploits the TBox, *e.g.* for concepts $A_2 \sqsubset A_1$, we reach A_2 via $\top \rightsquigarrow A_1 \rightsquigarrow A_2$. This way, we can stop the search if A_1 is already too weak. The operator also uses the range of roles to reduce the search space. This is similar to mode declarations widely used in ILP. However, in DL KBs, domain and range are usually explicitly given, so there is no need to define them manually. Overall, $\rho_{\mathcal{K}}$ supports more structures, *i.e.* concrete domains, than *e.g.* [16] and tries to smartly incorporate background knowledge. Additionally, unlike CELOE, the fuzzification of concrete domains enables the invention of new concepts during the learning process, which can be considered as a special case of predicate invention.

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.

References

[1] Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M.: The DL-Lite Family and Relations, *Journal of Artificial Intelligence Research*, **36**, 2009, 1–69.

[2] 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.

[3] Baader, F., Hanschke, P.: A Scheme for Integrating Concrete Domains into Concept Languages, *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence.* (J. Mylopoulos, R. Reiter, Eds.), Morgan Kaufmann, 1991.

[4] Bobillo, F., Straccia, U.: fuzzyDL: An expressive fuzzy description logic reasoner, *Proc. of the 17th IEEE Int. Conf. on Fuzzy Systems*, IEEE Press, 2008.

[5] Borgida, A.: On the Relative Expressiveness of Description Logics and Predicate Logics, *Artificial Intelligence*, **82**(1–2), 1996, 353–367.

[6] Cerami, M., Straccia, U.: On the (un)decidability of fuzzy description logics under Łukasiewicz t-norm, *Information Sciences*, **227**, 2013, 1–21.

[7] Drobics, M., Bodenhofer, U., Klement, E.-P.: FS-FOIL: an inductive learning method for extracting interpretable fuzzy descriptions, *Int. J. Approximate Reasoning*, **32**(2-3), 2003, 131–152.

[14] `http://www.w3.org/TR/owl2-manchester-syntax/`

[8] Dubois, D., Prade, H.: Possibility Theory, Probability Theory and Multiple-Valued Logics: A Clarification, *Annals of Mathematics and Artificial Intelligence*, **32**(1-4), 2001, 35–66.

[9] Fanizzi, N., d'Amato, C., Esposito, F.: DL-FOIL Concept Learning in Description Logics, *Proc. 18th Int. Conf. on Inductive Logic Programming* (F. Zelezný, N. Lavrač, Eds.), 5194, Springer, 2008.

[10] Hájek, P.: *Metamathematics of Fuzzy Logic*, Kluwer, 1998.

[11] Iglesias, J., Lehmann, J.: Towards Integrating Fuzzy Logic Capabilities into an Ontology-based Inductive Logic Programming Framework, *Proc. of the 11th Int. Conf. on Intelligent Systems Design and Applications*, IEEE Press, 2011.

[12] Klir, G. J., Yuan, B.: *Fuzzy sets and fuzzy logic: theory and applications*, Prentice-Hall, Inc., 1995.

[13] Konstantopoulos, S., Charalambidis, A.: Formulating description logic learning as an Inductive Logic Programming task, *Proc. of the 19th IEEE Int. Conf. on Fuzzy Systems*, IEEE Press, 2010.

[14] Lehmann, J.: DL-Learner: Learning Concepts in Description Logics, *Journal of Machine Learning Research*, **10**, 2009, 2639–2642.

[15] Lehmann, J., Auer, S., Bühmann, L., Tramp, S.: Class expression learning for ontology engineering, *Journal of Web Semantics*, **9**(1), 2011, 71–81.

[16] Lehmann, J., Haase, C.: Ideal Downward Refinement in the $\mathcal{EL}$ Description Logic, *Proc. 19th Int. Conf. on Inductive Logic Programming* (L. De Raedt, Ed.), 5989, Springer, 2010.

[17] Lisi, F. A., Straccia, U.: Dealing with Incompleteness and Vagueness in Inductive Logic Programming, *Proc. 28th It. Conf. on Computational Logic* (D. Cantone, M. Nicolosi Asmundo, Eds.), 1068, CEUR-WS.org, 2013.

[18] Lisi, F. A., Straccia, U.: A Logic-based Computational Method for the Automated Induction of Fuzzy Ontology Axioms, *Fundamenta Informaticae*, **124**(4), 2013, 503–519.

[19] Lukasiewicz, T., Straccia, U.: Managing Uncertainty and Vagueness in Description Logics for the Semantic Web, *Journal of Web Semantics*, **6**, 2008, 291–308.

[20] Michalski, R.: Pattern recognition as a rule-guided inductive inference, *IEEE transactions on Pattern Analysis and Machine Intelligence*, **2**(4), 1980, 349–361.

[21] Motik, B., Rosati, R.: A Faithful Integration of Description Logics with Logic Programming, *IJCAI 2007, Proc. of the 20th Int. Joint Conf. on Artificial Intelligence* (M. Veloso, Ed.), 2007.

[22] Quinlan, J. R.: Learning Logical Definitions from Relations, *Machine Learning*, **5**, 1990, 239–266.

[23] Reiter, R.: Equality and Domain Closure in First Order Databases, *Journal of ACM*, **27**, 1980, 235–249.

[24] Schmidt-Schauss, M., Smolka, G.: Attributive Concept Descriptions with Complements, *Artificial Intelligence*, **48**(1), 1991, 1–26.

[25] Serrurier, M., Prade, H.: Improving Expressivity of Inductive Logic Programming by Learning Different Kinds of Fuzzy Rules, *Soft Computing*, **11**(5), 2007, 459–466.

[26] Shibata, D., Inuzuka, N., Kato, S., Matsui, T., Itoh, H.: An Induction Algorithm Based on Fuzzy Logic Programming, *Proc. 3rd Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining* (N. Zhong, L. Zhou, Eds.), 1574, Springer, 1999.

[27] Straccia, U.: Reasoning within Fuzzy Description Logics, *Journal of Artificial Intelligence Research*, **14**, 2001, 137–166.

[28] Straccia, U.: Description Logics with Fuzzy Concrete Domains, *Proc. of the 21st Conf. in Uncertainty in Artificial Intelligence*, AUAI Press, 2005.

[29] Straccia, U.: *Foundations of Fuzzy Logic and Semantic Web Languages*, CRC Studies in Informatics Series, Chapman & Hall, 2013.

[30] Zadeh, L. A.: Fuzzy Sets, *Information and Control*, **8**(3), 1965, 338–353.