A FOIL-Like Method for Learning under Incompleteness and Vagueness

Francesca A. Lisi[1] (✉) and Umberto Straccia[2]

[1] Dipartimento di Informatica,
Università degli Studi di Bari "Aldo Moro", Bari, Italy
francesca.lisi@uniba.it
[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 present a novel Foil-like method for inducing fuzzy DL inclusion axioms from crisp DL knowledge bases and discuss the results obtained on a real-world case study in the tourism application domain also in comparison with related works.

1 Introduction

Motivation of the paper. 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).

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 [2]. Thanks to the OWA-compliancy, DLs have been considered the ideal starting point for the

G. Zaverucha et al. (Eds.): ILP 2013, LNAI 8812, pp. 123–139, 2014.
DOI: 10.1007/978-3-662-44923-3_9

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 *vague* knowledge [24]. 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 [8]. 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.

Contribution of the paper. Although a relatively important amount of work has been carried out in the last years concerning the use of fuzzy DLs as ontology languages [26], the problem of automatically managing the evolution of fuzzy ontologies by applying machine learning algorithms still remains relatively unaddressed [11,13,17]. In this paper, we present a novel method, named FOIL-$\mathcal{DL}$, for learning fuzzy DL inclusion axioms from any crisp DL knowledge base. The popular rule induction method FOIL [19] has been chosen as a starting point in our proposal for its simplicity and efficiency. The distinguishing feature of FOIL-$\mathcal{DL}$ w.r.t. previous work in DL learning (see, *e.g.*, [9,15,16]) is the treatment of numerical concrete domains with fuzzification techniques so that the induced axioms may contain fuzzy concepts.

Structure of the paper. For the sake of self-containment, Sect. 2 introduces some basic definitions we rely on. Section 3 provides a formal statement of the learning problem solved by FOIL-$\mathcal{DL}$ and details of the distinguishing features of FOIL-$\mathcal{DL}$ w.r.t. FOIL. Section 4 discusses relevant literature. Section 5 illustrates some experimental results obtained on a real-world case study in the tourism application domain. Section 6 concludes the paper with final remarks on the current work and possible directions of future work.

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

2 Preliminaries

Description Logics. For the sake of illustrative purposes, we present here a salient representative of the DL family, namely $\mathcal{ALC}$ [21], 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 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 $\mathcal{ALC}(\mathbf{D})$.

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	$\forall R.C$	$\{x \in \Delta^\mathcal{I} \mid \forall y\ (x, y) \in R^\mathcal{I} \to y \in C^\mathcal{I}\}$
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	$(a, b) : R$	$(a^\mathcal{I}, b^\mathcal{I}) \in R^\mathcal{I}$

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 $C^\mathcal{I} : \Delta^\mathcal{I} \to \{0, 1\}$ (either an individual belongs to the extension of C or does not belong to it).

Under the *Unique Names Assumption* (UNA) [20], 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 is *instance checking*, i.e., to check whether an ABox assertion is a logical consequence of $\mathcal{K}$. A more sophisticated version of instance checking, called *instance retrieval*, retrieves 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 [27]. A *fuzzy set* A over a countable crisp set X is characterised by a function $A\colon X \to [0,1]$. Unlike crisp sets that can be characterised by a function $A\colon X \to \{0,1\}$, 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 $(A \cap B)(x) = \min(A(x), B(x))$, $(A \cup B)(x) = \max(A(x), B(x))$ and $\bar{A}(x) = 1 - A(x)$, while the *inclusion degree* between A and B is typically defined as

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

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, Chap. 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*

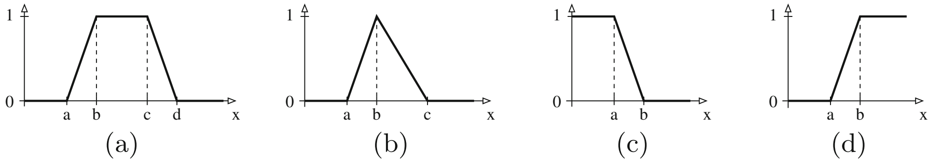


Fig. 1. (a) Trapezoidal function $trz(a, b, c, d)$, (b) triangular function $tri(a, b, c)$, (c) left-shoulder function $ls(a, b)$, and (d) right-shoulder function $rs(a, b)$.

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:

$$
\begin{array}{l}
\mathcal{I}(\phi \wedge \psi) = \mathcal{I}(\phi) \otimes \mathcal{I}(\psi) \\
\mathcal{I}(\phi \vee \psi) = \mathcal{I}(\phi) \oplus \mathcal{I}(\psi) \\
\mathcal{I}(\neg\phi) = \ominus \mathcal{I}(\phi) \\
\mathcal{I}(\phi \rightarrow \psi) = \mathcal{I}(\phi) \Rightarrow \mathcal{I}(\psi) \\
\mathcal{I}(\exists x.\phi(x)) = \sup_{y \in \Delta^{\mathcal{I}}} \mathcal{I}(\phi(y)) \\
\mathcal{I}(\forall x.\phi(x)) = \inf_{y \in \Delta^{\mathcal{I}}} \mathcal{I}(\phi(y))
\end{array}
\quad (2)
$$

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 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 [25]. 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

Table 2. Combination functions of various fuzzy logics.

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

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	$(a^{\mathcal{I}}, v^{\mathcal{I}}) \in T^{\mathcal{I}}$

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

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

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

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

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 $\alpha > 0$.

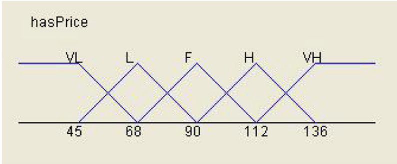


Fig. 2. Fuzzy sets derived from the concrete domain used as range of the role `hasPrice` in Example 1: `VeryLow` (VL), `Low` (L), `Fair` (F), `High` (H), and `VeryHigh` (VH).

Example 1. Let us consider the 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 Fig. 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*.*[2]

3 Learning Fuzzy $\mathcal{EL}$(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 $\mathcal{EL}(\mathbf{D})$[3] GCI axioms from a crisp $\mathcal{DL}$[4] 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 $\mathcal{L_H}$ of fuzzy $\mathcal{EL}(\mathbf{D})$ GCI axioms (the *language of hypotheses*), the goal is to find a set $\mathcal{H} \subset \mathcal{L_H}$ (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).

[2] Note that $0.18 = 0.318 \cdot 0.569$, where $0.318 = tri(90, 112, 136)(105)$.
[3] $\mathcal{EL}(\mathbf{D})$ is a fragment of $\mathcal{ALC}(\mathbf{D})$ [26].
[4] $\mathcal{DL}$ stands for any DL.

Remark 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 $\mathcal{L}_{\mathcal{K}}$. 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

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

and the concrete domain predicates are the following ones

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

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 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

```
function Foil-DL(K, A_t, E^+, E^-, L_H): H
begin
H := ∅;
D = InitialiseFuzzyConcreteDomain(K);
while E^+ ≠ ∅ do
      C := ⊤;
      φ := C ⊑ A_t;
      E^-_φ := E^-;
      while cf(φ) < θ or E^-_φ ≠ ∅ do
            C_best := C;
            maxgain := 0;
            Φ := Specialize(φ, L_H, K)
            foreach φ' ∈ Φ do
                  gain := Gain(φ', φ);
                  if gain ≥ maxgain then
                        maxgain := gain;
                        C_best := C';
                  endif
            endforeach
            φ := C_best ⊑ A_t;
            E^-_φ := {e ∈ E^- | K ∪ {φ} ⊨_+ e};
      endwhile
      H := H ∪ {φ};
      E^+_φ := {e ∈ E^+ | K ∪ {φ} ⊨_+ e};
      E^+ := E^+ \ E^+_φ;
endwhile
end
```

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

$\mathcal{E}^-$. 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 $\phi \in \mathcal{L_H}$ *covers* an example $e \in \mathcal{E}$ iff $\mathcal{K} \cup \{\phi\} \models_+ e$.

3.2 The Solution Strategy

In Foil-$\mathcal{DL}$, the *sequential covering* approach of Foil is kept as shown in Fig. 3. However, 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. A pre-processing phase is also required in order to generate the fuzzy datatypes to be used during the candidate generation phase. These novel features impact the definition of the functions InitialiseFuzzyConcreteDomain, Specialize and Gain as detailed in the remainder of this section.

Pre-processing. 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: $VeryLow_T = ls(min_T, min_T + k)$, $Low_T = tri(min_T, min_T + k, min_T + 2k)$, $Fair_T = tri(min_T + k, min_T + 2k, min_T + 3k)$, $High_T = tri(min_T + 2k, min_T + 3k, max_T)$ 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 1.

Candidate generation. 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:

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

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):

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

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

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

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

Example 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.

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

$$\textsc{Gain}(\phi', \phi) = p * (log_2(cf(\phi')) - log_2(cf(\phi))) \ , \tag{11}$$

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 $bed(\mathcal{K}, a{:}C) \notin \{0,1\}$.

4 Related Work

Several extensions of Foil to the management of vague knowledge are reported in the literature [7,22,23] but they are not conceived for DL ontologies. In DL learning, DL-Foil [9] adapts Foil to learn OWL DL equivalence axioms. 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]. All these algorithms 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 [18] 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.[5] 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 [17] 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.

5 Towards an Application in Tourism

A variant of Foil-$\mathcal{DL}$ has been implemented in the *fuzzyDL-Learner*[6] system. Notably, 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 DL reasoners such as Pellet[7] to solve all the deductive inference problems necessary to Foil-$\mathcal{DL}$ to work, namely instance retrieval, instance check and subclasses retrieval. The system can be configured to work under both CWA and OWA.

In order to demonstrate the potential usefulness of Foil-$\mathcal{DL}$ on a real-world application, we have considered a case study in the tourism application domain. More precisely, we have focused on the task of hotel finding because it can be reformulated as a classification problem solvable with Foil-like algorithms. Our goal is to use Foil-$\mathcal{DL}$ to find axioms of the form (5) for discriminating good hotels from bad ones. To the purpose, we have built an ontology, named *Hotel*,[8] which models the meaningful entities of the domain in hand.

[5] http://straccia.info/software/FuzzyOWL
[6] http://straccia.info/software/FuzzyDL-Learner/
[7] http://clarkparsia.com/pellet/
[8] http://straccia.info/software/FuzzyDL-Learner/download/FOIL-DL/examples/Hotel/Hotel.owl/

The ontology. The ontology *Hotel* consists of 8000 axioms, 74 classes, 4 object properties, 2 data properties, and 1504 individuals.

The main concepts forming the terminology of *Hotel* model the sites of interest (class `Site`), and the distances between sites (class `Distance`). Sites include accommodations (class `Accommodation`) such as hotels, attractions (class `Attraction`) such as parks, stations (class `Station`) such as airports, and civic facilities (class `Civic`) such as hospitals. The terminology encompasses also the amenities (class `Amenity`) offered by hotels (class `Hotel`) and the official 5-star classification system for hotel ranking (class `Rank`). The object properties `hasDistance` and `isDistanceFor` model the relationship between a site and a distance, and between a distance and the two sites, respectively. The data properties `hasPrice` and `hasValue` represent the average price of a room and the numerical value of a distance, respectively. Note that the latter would be better modeled as attribute of a ternary relation. However, since only binary relations can be represented in OWL, one such ternary relation is simulated with the class `Distance` and the properties `hasDistance`, `isDistanceFor` and `hasValue`.

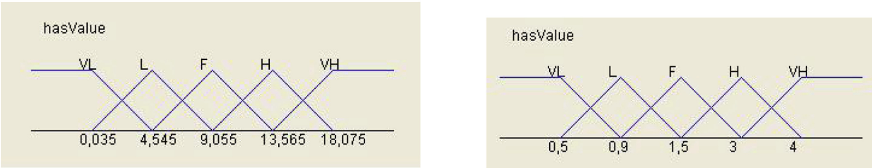
The 1504 individuals occurring in *Hotel* refer to the case of Pisa, Italy. In particular, 59 instances of the class `Hotel` have been automatically extracted from the web site of TripAdvisor.[9] Information about the rank, the amenities and the average room price has been added in the ontology for each of these instances. Further 24 instances have been created for the class `Site` and distributed among the classes under `Attraction`, `Civic` and `Station`. Finally, 1416 distances (instances of `Distance`) between the accommodations and the sites of interest have been measured in km and computed by means of Google Maps[10] API.

Anectodal experiments with Foil-$\mathcal{DL}$. As an illustration of the potential of FOIL-$\mathcal{DL}$, we discuss here the results obtained in two trials concerning the aforementioned learning problem with the class `Good_Hotel` as target concept. Graded hotel ratings from TripAdvisor users have been exploited for distinguishing good hotels from bad ones. Out of the 59 hotels, 12 with a higher percentage of positive feedback have been classified as instances of `Good_Hotel` (*i.e.*, as positive examples). In both trials, FOIL-$\mathcal{DL}$ has been configured to work under OWA by using Pellet as a DL reasoner. Syntactic restrictions are imposed on the form of the learnable GCI axioms. More precisely, conjunctions can have at most 5 conjuncts and at most 2 levels of nesting are allowed in existential role restrictions. The two trials differ as for the alphabet underlying the language of hypotheses.

First trial. In the first experiment, all the classes and the properties occurring in *Hotel* are considered to be part of the alphabet of the language of hypotheses. The membership functions for fuzzy concepts derived from the data properties `hasPrice` and `hasValue` in this trial are shown in Figs. 2 and 4(a), respectively. The results obtained for this configuration of FOIL-$\mathcal{DL}$ are:

```
Confidence         Axiom
   1,000       Hostel subclass of Good_Hotel
   1,000       hasPrice_veryhigh subclass of Good_Hotel
   0,569       hasPrice_high subclass of Good_Hotel
```

[9] http://www.tripadvisor.com/
[10] http://maps.google.com/



(a) (b)

Fig. 4. Membership functions for the fuzzy concepts `hasValue_verylow`, `hasValue_low`, `hasValue_fair`, `hasValue_high`, and `hasValue_veryhigh`, derived by FOIL-$\mathcal{DL}$ from the data property `hasValue` of *Hotel* in (a) the first trial and (b) the second trial.

```
0,286     hasAmenity some (24h_Reception) and hasAmenity some (Disabled_Facilities)
              and hasPrice_low subclass of Good_Hotel
0,282     hasAmenity some (Babysitting) and hasRank some (Rank)
              and hasPrice_fair subclass of Good_Hotel
0,200     Hotel_1_Star subclass of Good_Hotel
```

These results suggest the existence of user profiles, *e.g.* families and disabled people. Note that no axiom has been induced which encompasses knowledge about the distance of the accommodation from the sites of interest.

Second trial. In the second experiment, the configuration of FOIL-$\mathcal{DL}$ remains unchanged except for the alphabet underlying the language of hypotheses and the definition of membership functions for the fuzzy concepts. More precisely, the use of the object property `hasAmenity` is forbidden. Also, the fuzzification of the data property `hasValue` is more reasonable for a foot distance. Here, a very low distance does not exceed 900 meters, an average distance is about 1500 meters, and so on, as illustrated in Fig. 4(b). The axioms learned by FOIL-$\mathcal{DL}$ are:

```
Confidence    Axiom
  1,000     Hostel subclass of Good_Hotel
  1,000     hasPrice_veryhigh subclass of Good_Hotel
  0,739     hasDistance some (isDistanceFor some (Bus_Station) and hasValue_low)
              and hasDistance some (isDistanceFor some (Town_Hall) and hasValue_fair)
              and hasRank some (Rank) and hasPrice_verylow subclass of Good_Hotel
  0,569     hasPrice_high subclass of Good_Hotel
  0,289     Hotel_3_Stars and hasDistance some (isDistanceFor some (Train_Station)
              and hasValue_verylow) and hasPrice_fair subclass of Good_Hotel
  0,198     Hotel_4_Stars and hasDistance some (isDistanceFor some (Square) and hasValue_high)
              and hasRank some (Rank) and hasPrice_fair subclass of Good_Hotel
```

Note that the knowledge concerning the distance of the hotels from the sites of interest has now emerged during the learning process. In particular, the induced axioms suggest that closeness to stations of transportation means is a desirable feature when choosing a hotel.

A comparison with DL-Learner. In order to better illustrate the differences between FOIL-$\mathcal{DL}$ and state-of-the-art DL learning algorithms, we report here the results obtained for the same learning problem with two of the algorithms in the suite of DL-Learner: ELTL and CELOE.[11] For the purpose of this comparison we have adapted *Hotel* to make it compatible with DL-Learner. Notably, it has been necessary to provide explicitly negative examples for `Good_Hotel`. They have been generated by exploiting once again the graded hotel ratings of TripAdvisor users. More precisely, of the 59 hotels, 11 with a lower percentage of positive feedback have been classified as negative examples for `Good_Hotel`. The two algorithms have been run with default configuration. ELTL has returned 100 class expressions, out of which the first ten are reported below:

```
1: Thing (pred. acc.: 52,17%, F-measure: 68,57%)
2: Site (pred. acc.: 52,17%, F-measure: 68,57%)
3: Hotel (pred. acc.: 52,17%, F-measure: 68,57%)
4: Accomodation (pred. acc.: 52,17%, F-measure: 68,57%)
5: hasDistance some Distance (pred. acc.: 52,17%, F-measure: 68,57%)
6: (Site and hasDistance some Distance) (pred. acc.: 52,17%, F-measure: 68,57%)
7: (Hotel and hasDistance some Distance) (pred. acc.: 52,17%, F-measure: 68,57%)
8: (Accomodation and hasDistance some Distance) (pred. acc.: 52,17%, F-measure: 68,57%)
9: hasDistance some isDistanceFor some University (pred. acc.: 52,17%, F-measure: 68,57%)
10: hasDistance some isDistanceFor some Train_Station (pred. acc.: 52,17%, F-measure: 68,57%)
```

Note that they are rather weak as for predictive accuracy and quite trivial as for the significance except for the last two ones which involve the notion of distance from a site of interest. CELOE has returned the following ten solutions, with a little improvement of the effectiveness with respect to ELTL due to the augmented expressivity of the DL employed for the language of hypotheses:

```
1: ((not Camping) and (not Hotel_2_Stars)) (pred. acc.: 60,87%, F-measure: 72,73%)
2: (not Hotel_2_Stars) (pred. acc.: 56,52%, F-measure: 70,59%)
3: (not Camping) (pred. acc.: 56,52%, F-measure: 70,59%)
4: (Rank or (not Hotel_2_Stars)) (pred. acc.: 56,52%, F-measure: 70,59%)
5: (Rank or (not Camping)) (pred. acc.: 56,52%, F-measure: 70,59%)
6: (Place or (not Hotel_2_Stars)) (pred. acc.: 56,52%, F-measure: 70,59%)
7: (Place or (not Camping)) (pred. acc.: 56,52%, F-measure: 70,59%)
8: (Distance or (not Hotel_2_Stars)) (pred. acc.: 56,52%, F-measure: 70,59%)
9: (Distance or (not Camping)) (pred. acc.: 56,52%, F-measure: 70,59%)
10: (Amenity or (not Hotel_2_Stars)) (pred. acc.: 56,52%, F-measure: 70,59%)
```

Interestingly, the most accurate defines a good hotel by saying what it can not be considered as such. However, none of the learned class expressions provides a definition on the basis of the hotel features or the closeness with sites of interest.

6 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

[11] One such comparison could not be made with DL-FOIL since the implemented algorithm was not made available by the authors.

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-$\mathcal{DL}$ adopts the user-friendly presentation style of the Manchester OWL syntax.[12]

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. 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. J. Artif. Intell. Res. **36**, 1–69 (2009)
2. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook: Theory Implementation and Applications, 2nd edn. Cambridge University Press, Cambridge (2007)
3. Baader, F., Hanschke, P.: A scheme for integrating concrete domains into concept languages. In: Mylopoulos, J., Reiter, R. (eds.) Proceedings of the 12th International Joint Conference on Artificial Intelligence, pp. 452–457. Morgan Kaufmann (1991)
4. Bobillo, F., Straccia, U.: fuzzyDL: An expressive fuzzy description logic reasoner. In: IEEE International Conference on Fuzzy Systems, pp. 923–930. IEEE (2008)
5. Borgida, A.: On the relative expressiveness of description logics and predicate logics. Artif. Intell. **82**(1–2), 353–367 (1996)
6. Cerami, M., Straccia, U.: On the (un)decidability of fuzzy description logics under Łukasiewicz t-norm. Inf. Sci. **227**, 1–21 (2013)

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

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), 131–152 (2003)
8. Dubois, D., Prade, H.: Possibility theory, probability theory and multiple-valued logics: a clarification. Ann. Math. Artif. Intell. **32**(1–4), 35–66 (2001)
9. Fanizzi, N., d'Amato, C., Esposito, F.: DL-FOIL concept learning in description logics. In: Železný, F., Lavrač, N. (eds.) ILP 2008. LNCS (LNAI), vol. 5194, pp. 107–121. Springer, Heidelberg (2008)
10. Hájek, P.: Metamathematics of Fuzzy Logic. Kluwer, Dordrecht (1998)
11. Iglesias, J., Lehmann, J.: Towards integrating fuzzy logic capabilities into an ontology-based inductive logic programming framework. In: Proceedings of the 11th International Conference 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., Englewood Cliffs (1995)
13. Konstantopoulos, S., Charalambidis, A.: Formulating description logic learning as an inductive logic programming task. In: Proceedings of the 19th IEEE International Conference on Fuzzy Systems, pp. 1–7. IEEE Press (2010)
14. Lehmann, J.: DL-Learner: learning concepts in description logics. J. Mach. Learn. Res. **10**, 2639–2642 (2009)
15. Lehmann, J., Auer, S., Bühmann, L., Tramp, S.: Class expression learning for ontology engineering. J. Web Seman. **9**(1), 71–81 (2011)
16. Lehmann, J., Haase, C.: Ideal Downward Refinement in the $\mathcal{EL}$ Description Logic. In: De Raedt, L. (ed.) ILP 2009. LNCS, vol. 5989, pp. 73–87. Springer, Heidelberg (2010)
17. Lisi, F.A., Straccia, U.: A logic-based computational method for the automated induction of fuzzy ontology axioms. Fundamenta Informaticae **124**(4), 503–519 (2013)
18. Motik, B., Rosati, R.: A faithful integration of description logics with logic programming. In: Veloso, M. (ed.) Proceedings of the 20th International Joint Conference on Artificial Intelligence, pp. 477–482 (2007)
19. Quinlan, J.R.: Learning logical definitions from relations. Mach. Learn. **5**, 239–266 (1990)
20. Reiter, R.: Equality and domain closure in first order databases. J. ACM **27**, 235–249 (1980)
21. Schmidt-Schauss, M., Smolka, G.: Attributive concept descriptions with complements. Artif. Intell. **48**(1), 1–26 (1991)
22. Serrurier, M., Prade, H.: Improving expressivity of inductive logic programming by learning different kinds of fuzzy rules. Soft. Comput. **11**(5), 459–466 (2007)
23. Shibata, D., Inuzuka, N., Kato, S., Matsui, T., Itoh, H.: An induction algorithm based on fuzzy logic programming. In: Zhong, N., Zhou, L. (eds.) PAKDD 1999. LNCS (LNAI), vol. 1574, pp. 268–274. Springer, Heidelberg (1999)
24. Straccia, U.: Reasoning within fuzzy description logics. J. Artif. Intell. Res. **14**, 137–166 (2001)
25. Straccia, U.: Description logics with fuzzy concrete domains. In: Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence, pp. 559–567. AUAI Press (2005)
26. Straccia, U.: Foundations of Fuzzy Logic and Semantic Web Languages. CRC Studies in Informatics Series. Chapman & Hall, London (2013)
27. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**(3), 338–353 (1965)