Fundamenta Informaticae ??? (20xx) 1–26
DOI 10.3233/FI-2015-0000
IOS Press

A Granular Computing Method for OWL Ontologies*

Francesca A. Lisi [C], **Corrado Mencar**
Department of Informatics
CILA (Centro Interdipartimentale di Logica e Applicazioni)
Università degli Studi di Bari "Aldo Moro"
{*francescaalessandra.lisi, corrado.mencar*}*@uniba.it*

Abstract. We propose a method to extract and integrate fuzzy information granules from a populated OWL ontology. The purpose of this approach is to represent imprecise knowledge within an OWL ontology, as motivated by the fact that the Semantic Web is full of imprecise and uncertain information coming from perceptual data, incomplete data, data with errors, etc. In particular, we focus on Fuzzy Set Theory as a means for representing and processing information granules corresponding to imprecise concepts usually expressed by linguistic terms. The method applies to numerical data properties. The values of a property are first clustered to form a collection of fuzzy sets. Then, for each fuzzy set, the relative σ-count is computed and compared with a number of predefined fuzzy quantifiers, which are therefore used to define new assertions that are added to the original ontology. In this way, the extended ontology provides both a punctual view and a granular view of individuals w.r.t. the selected property. We use a real-world ontology concerning hotels and populated with data of the Italian city of Pisa, to illustrate the method and to test its implementation. We show that it is possible to extract granular properties that can be described in natural language and smoothly integrated in the original ontology by means of annotated assertions.

Keywords: Ontologies, OWL, Granular Computing, Fuzzy Set Theory.

*This work was partially funded by the *Università degli Studi di Bari "Aldo Moro"* under the IDEA Giovani Ricercatori 2011 grant "Dealing with Vague Knowledge in Ontology Refinement".
[C]Corresponding author
Address for correspondence: Dipartimento di Informatica, Università degli Studi di Bari "A. Moro", v. E. Orabona 4, 70125 Bari, Italy

1. Introduction

The World Wide Web, originally designed as a universal container of information, has witnessed a technological revolution aimed at opening the interchange of knowledge among agents, being them either humans or machines. This revolution is referred as the *Semantic Web*, and its realization, still in progress, is possible through a rich equipment of standards, languages and technologies. Among them, formal ontologies play a prominent role, and the *Web Ontology Language* (OWL)[1] is the W3C standard language for representing and processing ontologies for the Semantic Web. An OWL ontology is mainly defined by *classes*, related among them through *properties*, and embracing *individuals*. Classes enable a structured representation of knowledge, both at intensional level – through the use of set-theoretic operators and restrictions – and at extensional level, by associating individuals to classes. The logical foundations of OWL come from the knowledge representation formalisms collectively known as *Description Logics* (DLs) [1]. In particular, the very expressive DL $\mathcal{SROIQ}$ [2] is the logical counterpart of OWL 2. [2]

Endowing OWL ontologies with capabilities of representing and processing imprecise knowledge is a highly desirable feature, since the Semantic Web is full of imprecise and uncertain information coming from perceptual data (*i.e.*, data coming from subjective judgments), incomplete data, data with errors, etc. [3]. Moreover, even in the case that precise information is available, imprecise knowledge could be advantageous: tolerance to imprecision may lead to concrete benefits such as compact knowledge representation, efficient and robust reasoning, etc. [4]. Additionally, humans continually acquire, manipulate and communicate imprecise knowledge: therefore any ontology capable of expressing imprecise knowledge, when a precise alternative leads to an exceedingly complex representation, could be more *interpretable* by human users, i.e. easier to read and understand [5].

A number of mathematical theories are available to deal with imprecision and uncertainty in knowledge representation. The choice of the right theory depends on the type of imprecision. In particular, imprecision due to the lack of boundaries in concepts (usually of perceptual nature, such as "coldness" in the domain of indoor temperatures, "interestingness" of movies, etc.) are well modeled through fuzzy set theory [6]. In essence, fuzzy sets define collections of objects whose membership can be partial, thus quantifying how much the concept is applicable to the object. Fuzziness pervades human reasoning and allows humans to intelligently act in complex environments: since fuzzy sets make possible a computational representation of concepts with no sharp boundaries, they enable machines to carry out human-centered information processing and reasoning [7].

The integration of fuzzy sets in ontologies for the Semantic Web can be achieved in different ways and for different pursuits, such as information retrieval, matchmaking, ontologies alignment, etc. [8]. In most cases, existing ontologies are equipped with fuzzy sets representing expert knowledge in order to enrich the knowledge base with additional perceptual knowledge. However, the definition of such fuzzy sets could be hard if they must represent some hidden properties of individuals. In this paper, we take a data-driven approach, where fuzzy sets are automatically derived from the available individuals in the ontology through a fuzzy clustering process. The derived fuzzy sets are more compliant to represent similarities among individuals w.r.t. a manual definition by experts.

This approach promotes a granular view of individuals that can be exploited to further enrich the knowledge base of an ontology by applying the *Granular Computing* (GC) paradigm. According to GC, *information granules* (such as fuzzy sets) are elementary units of information [9]. In this capacity, infor-

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

mation granules can be represented as individuals in an ontology, eventually belonging to a generic class (e.g. the "Granule" class), and thus endowed with properties that are specific to information granules and do not pertain to the original individuals from which the granules have been derived.

An example of granule-specific properties is the granularity of an information granule [9], which can be numerically quantified and then related to fuzzy sets representing linguistic quantifiers, usually expressed with terms such as "most", "few", etc. [10, 11]. Thence, a new level of knowledge emerges from information granules, which can be profitably included in the original ontology to express knowledge not pertaining single individuals, but on (fuzzy) collections of them, being each collection defined by individuals kept together by their similarity.

The paper is structured as follows. After some preliminaries on GC and DLs (Sect. 2), we describe the proposed granulation approach in some basic cases in Sect. 3, starting from the simplest case of individuals belonging to a class with a numerical data property, then moving to hierarchical class structures and to ontology design patterns corresponding to a ternary relation. Some experimental results are reported in Sect. 4. We report some final considerations and perspectives of future work in Sect. 5.

2. Preliminaries

2.1. Granular Computing

2.1.1. Fuzzy Set Theory

Fuzzy Set Theory (FST) is a mathematical theory enabling the representation and processing of sets without sharp boundaries [6, 12]. FST is based on the notion of fuzzy set, which is mathematically characterized[3] by a membership function on a domain or Universe of Discourse X:

$$F\colon X \mapsto [0,1]$$

This definition generalizes the usual characteristic function of classical ("crisp") sets. Consequently, the classical set operations are generalized in order to comply with partial membership. 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 crisp inclusion is defined by the relation

$$A \subseteq B \Leftrightarrow \forall x : A\,(x) \leq B\,(x)$$

In fuzzy modeling, membership functions are usually defined as belonging to a specific family that shows some useful properties. In particular, in this study we adopt fuzzy sets with trapezoidal membership functions, which are characterized by four parameters and defined as

[3]More general definitions are possible; however, we stick on a more widespread definition that is enough for the present study.

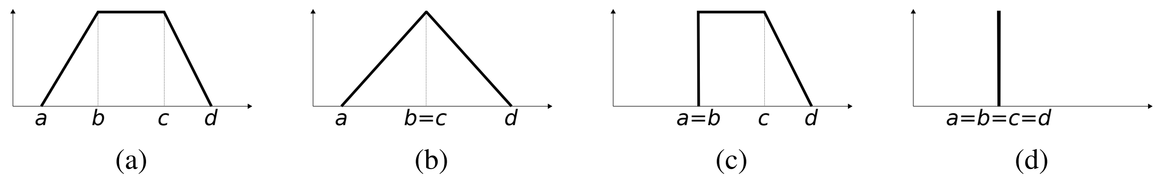


Figure 1. Four notable shapes of membership functions: (a) trapezoidal, (b) triangular, (c) left-shoulder, and (d) singleton.

$$trz[a,b,c,d](x) = \begin{cases} 0 & \text{if } x < a \\ (x-a)/(b-a) & \text{if } x \in [a,b) \\ 1 & \text{if } x \in [b,c] \\ (d-x)/(d-c) & \text{if } x \in (c,d] \\ 0 & \text{if } x > d\,. \end{cases} \tag{1}$$

Trapezoidal fuzzy sets have several attractive properties, which make them widely used in fuzzy modeling. Furthermore, they generalize other families of fuzzy sets, including the triangular fuzzy sets (when $b = c$), left/right shoulders (when $a = b$ or $c = d$) and singletons (when $a = b = c = d$) (see fig. 1 for some examples).

2.1.2. Fuzzy sets as information granules

From the modeling viewpoint, FST is very useful to represent concepts characterized by both *granularity* (*i.e.* concepts that refer to a multiplicity of objects) and *graduality* (*i.e.* the reference of concepts to objects is a matter of degree). This makes FST a valid candidate to represent perceptive concepts, i.e. concepts that are formed by an act of perception and are usually designated by terms drawn from natural language [13, 14, 15]. Fuzzy sets can be used to represent *information granules*, *i.e.* collections of objects kept together due to their similarity, proximity, etc. [9]. They are the building blocks of *Granular Computing* (GC), a paradigm for information processing where information granules are treated as elementary carriers of information [16]. Differently from classical symbolic processing systems, GC systems take into account the contents of information granules for processing; in other words, GC relies on the semantics of information granules.

The content of an information granule is therefore crucial for a meaningful definition. In the case of *fuzzy* information granules (*i.e.* information granules defined in terms of fuzzy sets), when they are used to represent perceptual concepts, usually identified by linguistic terms, some additional constraints must be ensured in order to label them with linguistic terms [5]. A simple yet effective way to define such fuzzy information granules, when the domain is an interval on the real line, *i.e.* $X = [m_X, M_X] \subset \mathbb{R}$, is through a so-called *Strong Fuzzy Partition* (SFP) [17]. A SFP is a finite collection $\mathbf{F} = \{F_1, F_2, \ldots, F_n\}$ of fuzzy sets on a numerical domain such that:

1. each fuzzy set $F \in \mathbf{F}$ is *normal*, *i.e.* at least on element of the domain has full membership:

$$\exists x \in X \text{ s.t. } F(x) = 1$$

2. each fuzzy set is *convex*:

$$\forall x, y, z \in X : x \leq y \leq z \rightarrow F(y) \geq \min\{F(x), F(z)\}$$

3. each fuzzy set is *continuous*;

4. the sum of membership degrees of each element of the domain, over all the fuzzy sets of the partition, is one, *i.e.*:

$$\forall x \in X : \sum_{F \in \mathbf{F}} F(x) = 1$$

Moreover, the number of fuzzy sets in a SFP is usually small, often in the range 5 ± 2 because this is the typical range of information chunks a human being is capable to keep in her short-term memory [18].

It is not difficult to build a SFP made of trapezoidal fuzzy sets: if $[a_i, b_i, c_i, d_i]$ is the quadruple of parameters defining the i-th fuzzy set in the SFP as in (1), it is enough to ensure that $a_i = c_{i-1}$ and $b_i = d_{i-1}$ for $i = 2, 3, \ldots, n$ (also, $a_1 = b_1 = m_X$ and $c_n = d_n = M_X$) . Furthermore, it is immediate to form a uniform SFP made of triangular fuzzy sets with the additional constraint:

$$b_i = c_i = m_X + (i-1)\frac{M_X - m_X}{n-1}$$

However, a uniform SFP does not take into account data, when available. In this case, a uniform SFP fails to express a meaningful granular representation of a domain. When numerical data are available, clustering is an effective way to define meaningful information granules.

A widespread algorithm for *fuzzy clustering* is Fuzzy C-Means (FCM) [19], an extension of the well-known K-Means that accommodates partial memberships of data to clusters. FCM requires the specification of the number $c > 1$ of clusters and a "fuzzification parameter" $m > 1$ (usually, $m = 2$), as well as a set of N d-dimensional numerical data $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N \in \mathbb{R}^d$, and returns a sequence of c prototype vectors $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_c \in \mathbb{R}^d$ representing cluster centers, as well as a partition matrix $U = [u_{ij}]$ for $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, c$ such that u_{ij} represents the degree of membership of the i-th data sample to the j-th cluster. The prototypes and the partition matrix are computed by minimizing the functional

$$J = \sum_{i=1}^{N} \sum_{j=1}^{c} u_{ij}^m ||\mathbf{x}_i - \mathbf{p}_j||^2$$

constrained to

$$\forall i = 1, 2, \ldots, N : \sum_{j=1}^{c} u_{ij} = 1$$

and

$$\forall j = 1, 2, \ldots, c : 0 < \sum_{i=1}^{N} u_{ij} < N$$

The functional is usually minimized by alternate optimization on the set of prototypes and on the partition matrix. We adopted the Euclidean norm to define the distance between samples and prototypes; however, other metrics are possible.

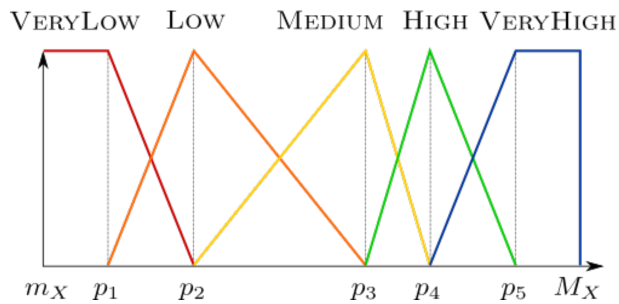


Figure 2. Example of SFP consisting of five fuzzy sets with variable granularity.

FCM, applied to one-dimensional numerical data (*i.e.* when $d = 1$), can be used to derive a set of c clusters characterized by prototypes $p_1, p_2, \ldots, p_c$, with $p_j \in \mathbb{R}$ and $p_j < p_{j+1}$. These prototypes, along with the range of data, provide enough information to define a SFP with two trapezoidal fuzzy sets and $c - 2$ triangular fuzzy sets according to the following rules:

$$F_j = \begin{cases} trz\,[m_X, m_X, p_1, p_2] & \text{if } j = 1 \\ trz\,[p_{j-1}, p_j, p_j, p_{j+1}] & \text{if } 1 < j < c \\ trz\,[p_{c-1}, p_c, M_X, M_X] & \text{if } j = c. \end{cases} \tag{2}$$

where $[m_X, M_X]$ is the range of data. In fig. 2 an example of SFP, consisting of five fuzzy sets with variable granularity obtainable from FCM prototypes, is depicted; the fuzzy sets are well-distinguished so they can be easily tagged by linguistic terms drawn from natural language.

2.1.3. Fuzzy quantifiers

Information granules are first-class citizens in GC, therefore a number of properties and relations can be defined by considering information granules as elementary units. In the case of fuzzy information granules, these properties can be directly derived from functions defined on fuzzy sets.

Fuzzy sets, like crisp sets, can be quantified in terms of their *cardinality*. Several definitions of cardinality of fuzzy sets have been proposed [20], although in this paper we consider only a relative scalar cardinality, the *relative σ-count*, defined for a finite set D in the Universe of Discourse as

$$\sigma(F) = \frac{\sum_{x \in D} F(x)}{|D|} \in [0, 1] \tag{3}$$

where, obviously, $\sigma(\emptyset) = 0$ and $\sigma(D) = 1$.

Since the range of σ is always the unitary interval, a number of fuzzy sets can be defined to represent granular concepts about cardinalities, such as MANY, MOST, etc. These concepts are called *fuzzy quantifiers* [11, 21, 22]. As usual, they can be defined so as to form a SFP; in this way linguistic labels can be easily attached, as depicted in the example in fig. 3. It is interesting to observe that the usual existential quantifier ($\exists$) and universal quantifier ($\forall$) can be represented as special cases of fuzzy quantifiers (see Fig. 4): $Q_\exists(x) = 1$ iff $x > 0$, 0 otherwise; $Q_\forall(x) = 1$ iff $x = 1$, 0 otherwise.

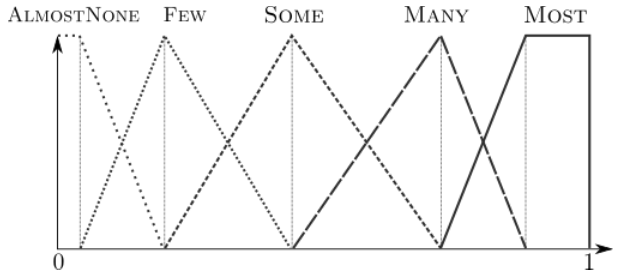


Figure 3. Examples of fuzzy quantifiers.

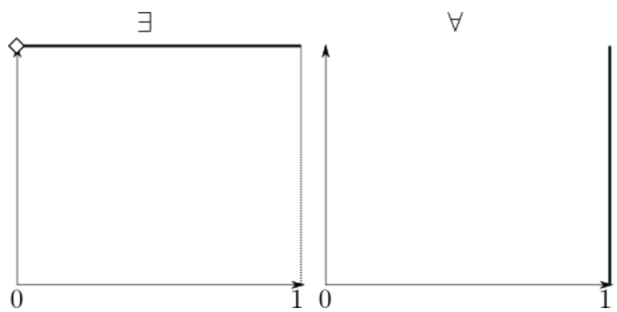


Figure 4. Granular representation of the existential and universal quantifiers.

Fuzzy quantifiers can be used to express imprecise properties on fuzzy information granules. More specifically, given a quantifier Q labeled with `Q` and a fuzzy set F labeled with `F`, the membership degree $Q(\sigma(F))$ quantifies the truth degree of the proposition

$$\texttt{Q x are F}$$

For example, if $\texttt{Q} \equiv \texttt{Many}$ and $\texttt{F} \equiv \texttt{Low}$, the fuzzy proposition

$$\texttt{Many x are Low}$$

asserts that many data points have a low value; the truth degree of this proposition is quantified by

$$Q_{\texttt{Many}}(\sigma(F_{\texttt{Low}}))$$

By a proper formal representation, these fuzzy propositions can be embodied within an ontology by introducing new individuals corresponding to the information granules.

2.2. Description Logics

Description Logics (DLs) are a family of decidable First Order Logic (FOL) fragments that allow for the specification of structured knowledge in terms of classes (*concepts*), instances (*individuals*), and binary relations between instances (*roles*) [1]. Complex concepts (denoted with C) can be defined from

Table 1. Syntax and semantics of constructs for $\mathcal{ALC}(\mathbf{D})$.

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	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
concrete role	T	$T^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathbf{D}}$
individual	a	$a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
concrete value	v	$v^{\mathcal{I}} \in \Delta^{\mathbf{D}}$
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 abstract 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 abstract role restriction	$\exists R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y\ (x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
universal concrete role restriction	$\forall T.\mathbf{d}$	$\{x \in \Delta^{\mathcal{I}} \mid \forall z\ (x,z) \in T^{\mathcal{I}} \rightarrow z \in \mathbf{d}^{\mathbf{D}}\}$
existential concrete role restriction	$\exists T.\mathbf{d}$	$\{x \in \Delta^{\mathcal{I}} \mid \exists z\ (x,z) \in T^{\mathcal{I}} \wedge z \in \mathbf{d}^{\mathbf{D}}\}$
general concept inclusion	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
concept assertion	$a : C$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
abstract role assertion	$(a,b) : R$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$
concrete role assertion	$(a,v) : T$	$(a^{\mathcal{I}}, v^{\mathcal{I}}) \in T^{\mathcal{I}}$

atomic concepts (A) and roles (R) by means of the constructors available for the DL in hand. The members of the DL family differ from each other as for the set of constructors, thus for the complexity of concept expressions they can generate. For the sake of illustrative purposes, we present here a salient representative of the DL family, namely $\mathcal{ALC}$ [23], which is often considered to illustrate some new notions related to DLs. 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).

The semantics of DLs can be defined directly with set-theoretic formalizations or through a mapping to FOL (as shown in [24]). 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) [25], 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 Open World Assumption (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 *consistency check*, which tries to prove the satisfiability of a DL KB $\mathcal{K}$, is the main reasoning task in DLs. It is performed by applying decision procedures mostly based on tableau calculus. All other reasoning tasks can be reformulated as consistency checks.

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 a so-called *concrete domain* (or *datatype theory*) $\mathbf{D} = \langle \Delta^{\mathbf{D}}, \cdot^{\mathbf{D}} \rangle$, which consists of a datatype domain $\Delta^{\mathbf{D}}$ (*e.g.*, the set of real numbers in double precision) 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 (typically, $n = 1$) relation over $\Delta^{\mathbf{D}}$ [26]. In DLs extended with concrete domains, each role is therefore either *abstract* (denoted with R) or *concrete* (denoted with T). The set of constructors for $\mathcal{ALC}(\mathbf{D})$ is reported in Table 1.

2.2.1. Relationship to OWL

The main building blocks of OWL are very similar to those of DLs, with the main difference that concepts are called *classes* and roles are called *properties*. It is therefore not surprising that DLs have had a major influence on the development of OWL and the expressive features that it provides. Historically, however, OWL has also been conceived as an extension to RDF, a Web data modelling language whose expressivity is comparable to DL ABoxes [27]. Therefore, it has been decided to specify both styles of formal semantics for OWL: the Direct Semantics based on DLs and the RDF-based Semantics. Of course, here we are mainly interested in the Direct Semantics of OWL. This semantics is only defined for OWL ontologies under the restriction that the OWL axioms can be read as $\mathcal{SROIQ}$ axioms. This syntactic fragment of OWL is called OWL DL. Under the Direct Semantics, large parts of OWL DL can indeed be considered as a syntactic variant of $\mathcal{SROIQ}$. For example, the axiom

$$\texttt{Hotel\_3\_Stars} \equiv \texttt{Hotel} \sqcap \exists \texttt{hasRank.3\_stars}$$

would be written as follows in OWL:

```
EquivalentClasses
( Hotel_3_Stars ObjectIntersectionOf( Hotel ObjectHasValue(hasRank 3_stars)) )
```

where the symbols `Hotel_3_Stars`, `Hotel`, `hasRank` and `3_stars` would be identifier strings that conform to the OWL specification. The above example illustrates the close relationship between the syntax of $\mathcal{SROIQ}$ and that of OWL in the so-called Functional-Style Syntax. However, the most prominent among the syntactic forms provided by the OWL standard is the *RDF/XML serialisation* since it is the only format that all conforming OWL tools need to understand.

It is interesting to note that there are still a few differences between OWL DL under the Direct Semantics and $\mathcal{SROIQ}$. On a syntactic level, OWL provides a lot more operators that, though logically redundant, can be convenient as shortcuts for compound DL axioms. For example, OWL has special constructs for specifying domain and range of a property, even though these could equally well be expressed in $\mathcal{SROIQ}$. Most notably, this includes support for *datatypes* and datatype literals. Both DLs and OWL in this case strictly distinguish roles/properties that relate to abstract individuals from those that relate to values from some datatype. In OWL, the constructs that relate to datatypes include Data in their name while constructs that relate to abstract individuals include Object. For example, OWL distinguishes `ObjectIntersectionOf` (used above) from `DataIntersectionOf` (the intersection of datatypes). The only other logical feature that is missing in DLs are so-called *Keys* which are inspired to key constraints in databases and can be used for data integration. Besides the logical features, OWL also includes a number of other aspects that are not considered in DLs at all. For example, it includes means of naming an ontology and of importing ontological axioms from one ontology into another. Further extra-logical features include a simple form of meta-modelling called *punning*, non-logical axioms to declare *identifiers*, and the possibility to add *annotations* to arbitrary axioms and entities similar to comments in a programming language.

2.2.2. Fuzzy extensions of DLs and OWL

Several fuzzy extensions of DLs can be found in the literature (see the survey in [8]). 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 $C^{\mathcal{I}} : \Delta^{\mathcal{I}} \rightarrow [0, 1]$ 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 [28]. In particular, $\cdot^{\mathbf{D}}$ maps each concrete role into a function from $\Delta^{\mathbf{D}}$ to $[0, 1]$. Typical examples of datatype predicates are

$$\mathbf{d} \;:=\; ls(a, b) \mid rs(a, b) \mid tri(a, b, c) \mid trz(a, b, c, d) \mid \;\geq_v\; \mid \;\leq_v\; \mid \;=_v\;, \tag{4}$$

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

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 $\langle \tau, \alpha \rangle$. Further details of the reasoning procedures for fuzzy DLs can be found in [29].

Fuzzy quantifiers have been also studied in fuzzy DLs. In particular, Sanchez and Tettamanzi [30] define an extension of fuzzy $\mathcal{ALC}(\mathbf{D})$ involving fuzzy quantifiers of the absolute and relative kind, and using qualifiers. They also provide algorithms for performing two important reasoning tasks with their DL: reasoning about instances, and calculating the fuzzy satisfiability of a fuzzy concept.

Some fuzzy DL reasoners have been implemented, such as *fuzzyDL* [31, 32]. Not surprisingly, each reasoner uses its own fuzzy DL language for representing fuzzy ontologies and, thus, there is a need for a standard way to represent such information. In [33, 34], Bobillo and Straccia propose to use OWL 2 itself to represent fuzzy ontologies. More precisely, they use OWL 2 annotation properties to encode fuzzy $\mathcal{SROIQ}(\mathbf{D})$ ontologies. The use of annotation properties makes possible (i) to use current OWL 2 editors for fuzzy ontology representation, and (ii) that OWL 2 reasoners discard the fuzzy part of a fuzzy ontology, producing the same results as if would not exist. Additionally, we identify the syntactic differences that a fuzzy ontology language has to cope with, and show how to address them using OWL 2 annotation properties.

3. Fuzzy Granulation of OWL Schemas

In this Section we show our proposal of introducing a granular view of individuals within an OWL ontology. We shall proceed incrementally starting from the simplest case. For the sake of simplicity, we shall use the OWL terminology henceforth instead of the DL terminology (we remind the reader that class stands for concept, and property stands for role).

3.1. Case 1

Let C be a class and T a functional datatype property connecting instances of C to values in a numerical range $\mathbf{d}$. (See fig. 5 for a graphical representation of this construct.) This schema can be directly translated into a table (see table 2) with two columns and as many rows as the number n of individuals of C for which T holds.

The dataset in table 2 can be easily granulated in a set of $c > 1$ fuzzy sets $F_1, F_2, \ldots, F_c$ by applying, *e.g.*, the fuzzy clustering method mentioned in Sec. 2.1.1. In essence, the granulation process puts

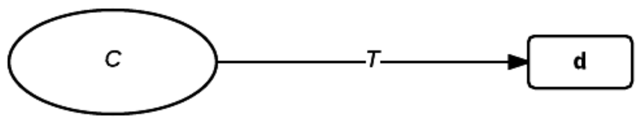


Figure 5. Graphical representation of a functional datatype property T with domain C and range over a numerical datatype $\mathbf{d}$.

Table 2. Tabular representation of the OWL schema depicted in fig. 5.

C	T
a_1	v_1
a_2	v_2
$\cdots$	$\cdots$
a_n	v_n

individuals in the same information granule if their respective values are similar. The use of fuzzy sets to define granules ensures a gradual membership degree of individuals to such granules, where the maximal membership is assigned to individuals detected as "prototypes" of each granule. Each fuzzy set represents a fuzzy concept, and can be tagged by a linguistic term, *e.g.* `Low`.

The result of granulation can be represented in a new table (see table 3), where each individual a_i is associated to a row of membership values μ_{ij}, being

$$\mu_{ij} = F_j(v_i) \tag{5}$$

For each granule F_j, the relative cardinality $\sigma(F_j)$ can be computed by means of the formula in Eq. (3). Given a fuzzy quantifier Q_k, the membership degree

$$q_{jk} = Q_k(\sigma(F_j)) \tag{6}$$

identifies the degree of truth of the fuzzy proposition "$Q_k x$ are F_j". In this way, a new table can be constructed from a collection $Q_1, Q_2, \ldots, Q_m$ of $m > 1$ fuzzy quantifiers, as shown in table 4.

Table 3. Granulated individuals obtained from table 2.

C	F_1	F_2	$\cdots$	F_c
a_1	μ_{11}	μ_{12}	$\cdots$	μ_{1c}
a_2	μ_{21}	μ_{22}	$\cdots$	μ_{2c}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
a_n	μ_{n1}	μ_{n2}	$\cdots$	μ_{nc}

Table 4. Quantified cardinalities for the granules reported in table 3.

	Q_1	Q_2	$\cdots$	Q_m
F_1	q_{11}	q_{12}	$\cdots$	q_{1m}
F_2	q_{21}	q_{22}	$\cdots$	q_{2m}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
F_c	q_{c1}	q_{c2}	$\cdots$	q_{cm}

If $cm \ll n$, a sensible reduction of data can be achieved to represent the original property through a granulated view. (To further reduce data, a threshold τ can be set, so that all q_{jk} less than τ are set to zero.) The new granulated view can be integrated in the ontology as follows. The fuzzy sets F_i are the starting point for the definition of new subclasses of C defined as

$$D_i \equiv C \sqcap \exists T.F_i$$

Also, a new class G is defined, with individuals $g_1, g_2, \ldots, g_c$, where each individual g_i is an information granule corresponding to F_i. Each individual in D_i is then mapped to g_i by means of an object property `mapsTo`. Moreover, for each fuzzy quantifier Q_k, a new class is introduced, which models one of the fuzzy sets over the cardinality of G. The connection between the class G and each class Q_k is established through an object property with a conventional name like `hasCardinality`, with degrees identified as in table 4.

Example 3.1. For illustrative purposes, we refer to an OWL ontology in the tourism domain, *Hotel*,[4] which encompasses the datatype property `hasPrice` with the class `Hotel` as domain and range in the datatype domain `xsd:double`. Let us suppose that the room price for Hotel Verdi (instance `verdi` of `Hotel`) is 105, *i.e.* the ontology contains the assertion `(verdi, 105):hasPrice`. By applying fuzzy clustering to `hasPrice`, we might obtain three fuzzy sets (with labels `Low`, `Medium`, `High`) from which the following classes are derived:

`LowPriceHotel` $\equiv$ `Hotel` $\sqcap$ $\exists$`hasPrice.Low`
`MidPriceHotel` $\equiv$ `Hotel` $\sqcap$ $\exists$`hasPrice.Medium`
`HighPriceHotel` $\equiv$ `Hotel` $\sqcap$ $\exists$`hasPrice.High`,

With respect to these classes, `verdi` shows different degrees of membership; *e.g.* `verdi` is a low-price hotel at degree 0.8 and a mid-price hotel at degree 0.2 (see fig. 6 for a graphical representation).

Subsequently, we might be interested in obtaining aggregated information about hotels. Quantified cardinalities allow us, for instance, to represent the fact that "Many hotels are low-priced" with the fuzzy assertion

`lph` : $\exists$`hasCardinality.Many`

with truth degree 0.8, where `lph` is an instance of the class `Granule` corresponding to `LowPriceHotel` and `Many` is one of the fuzzy sets representing a fuzzy quantifier.

[4] `http://www.umbertostraccia.it/cs/software/FuzzyDL-Learner/download/FOIL-DL/examples/Hotel/Hotel.owl`

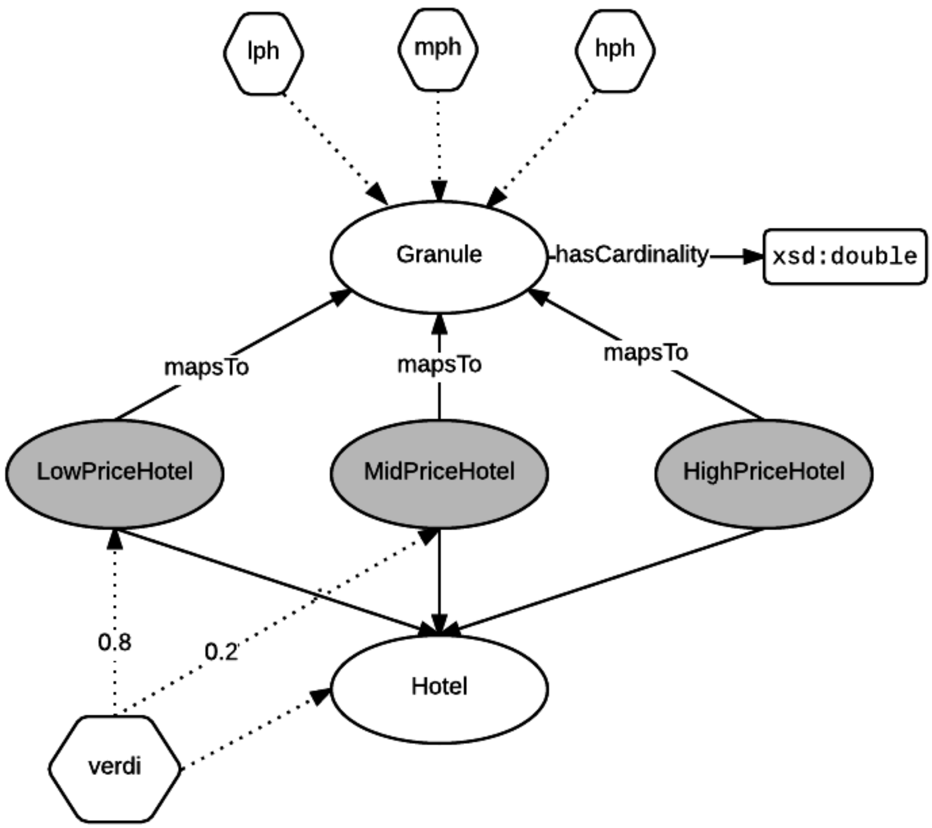


Figure 6. Graphical representation of the output of the fuzzy granulation process on the OWL schema described in fig. 5 and instantiated with concepts reported in Example 3.1. Fuzzy classes are depicted in gray.

3.2. Case 2

A natural extension of the proposed granulation method follows when the class C is specialized in subclasses, as in fig. 7. In this case, there are as many tables with the same structure of table 2 as the number of subclasses.

Analogously, for each subclass $SubC_j$ a structure of fuzzy information granules $F_{j1}, F_{j2}, \ldots, F_{jc}$ is produced and quantified according to the usual fuzzy quantifiers $Q_1, Q_2, \ldots, Q_m$. (The quantifiers do not depend on the subclass as their definition is fixed for all information granules.)

Example 3.2. Following Example 3.1, one may think of having a subsumption hierarchy with the class `Accommodation` as the root and `Hotel` and `B&B` as subclasses (see fig. 8). Hotels are granulated in three fuzzy subclasses (`LowPriceHotel`, `MidPriceHotel` and `HighPriceHotel`) while B&Bs are granulated in two fuzzy subclasses (`CheapB&B` and `ExpensiveB&B`). These fuzzy classes are related to the classes representing fuzzy quantifiers via `Granule` analogously to Example 3.1.

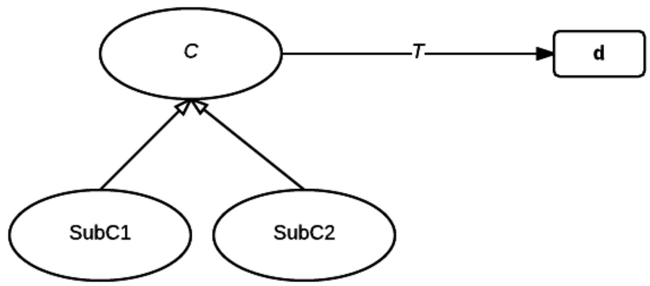


Figure 7. Variant of the OWL schema shown in Fig. 5 for the case of C having subclasses.

Table 5. Tabular representation of the OWL schema depicted in fig. 9.

C	D	T
a_1	b_1	v_1
$\cdots$	$\cdots$	$\cdots$
a_i	b_j	v_k
$\cdots$	$\cdots$	$\cdots$
a_n	b_m	v_l

3.3. Case 3

A case of particular interest is given by OWL schemes representing *ternary relations*. A ternary relation is a subset of the Cartesian product involving three domains $C \times D \times N$ (for our purposes, we will assume N a numerical domain). Because of DL restrictions, however, ternary relations are not directly representable in OWL, yet they can be indirectly represented through an auxiliary class E, two object properties R_1 and R_2, and one datatype property T, as depicted in fig. 9.

The structure in fig. 9 corresponds to a tabular representation with three columns, and as many rows as the number of elements of the relation, as in table 5. By removing one of the two columns in table 5, the resulting table is in accordance with table 2, which was the starting point of the granulation process. In particular, as in the previous cases, a number of fuzzy sets $F_1, F_2, \ldots, F_c$ can be derived starting from the dataset represented in table 5, where one column has been dropped. (We henceforth assume to drop column C.)

In order to connect information granules with classes, we proceed as follows. For each information granule F_j it is possible to compute the σ-count

$$\sigma_j = \frac{\sum_{i=1}^{n} F_j\left(v_i\right)}{n}$$

representing the relative cardinality of the fuzzy set F_j over all tuples of the relation as inn table 5. Such cardinality can be quantified according to the fuzzy quantifiers $Q_1, \ldots, Q_m$.

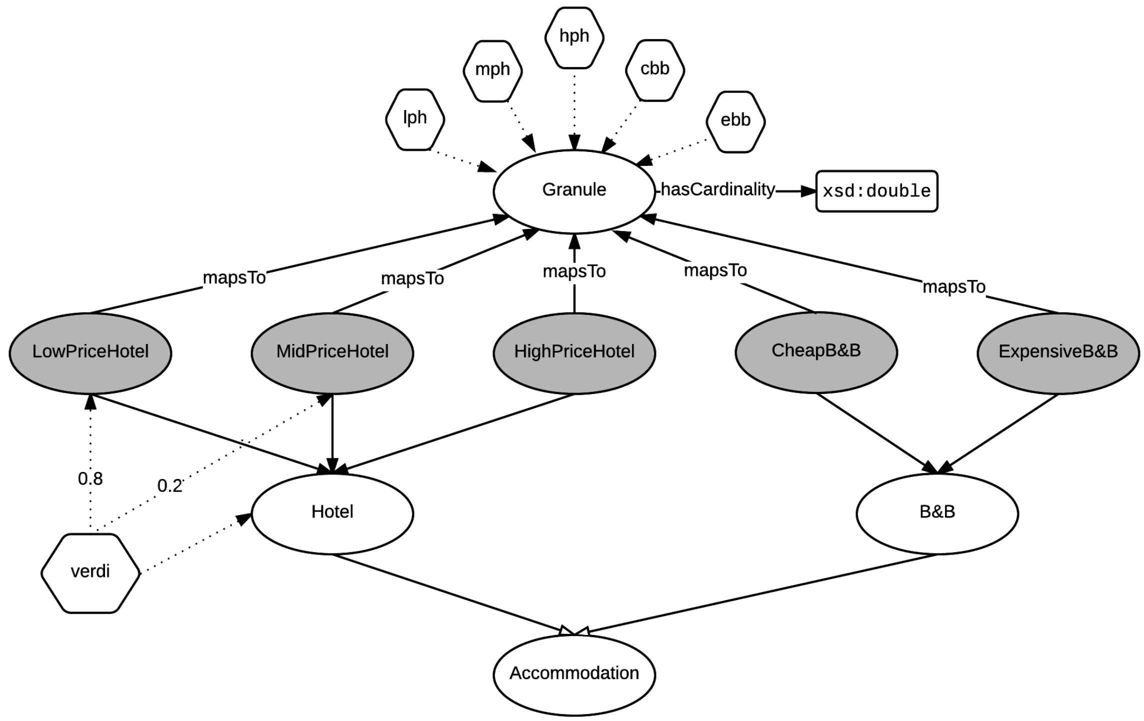


Figure 8. Graphical representation of the output of the fuzzy granulation process on the OWL schema reported in Fig. 7 and instantiated with the concepts used in Example 3.2.

The final arrangement of the information granules, connected with the individuals in C, merges the modeling of ternary relations as in fig. 9 with the granular model illustrated in case 1. The new classes, representing information granules, are connected to the auxiliary class E in order to express a granular view of the relation between classes C and D. Finally, a natural extension of this case allows the specialization of the class D in subclasses (as in case 2).

Example 3.3. With reference to *Hotel* ontology, we might also consider the distances between hotels and attractions. This is clearly a case of a ternary relation that needs to be modeled through an auxiliary class `Distance`, which is connected to the classes `Hotel` and `Attraction` by means of the object properties `hasDistance` and `isDistanceFor`, respectively, and plays the role of domain for a datatype property `hasValue` with range `xsd:double`. The knowledge that "Hotel Verdi has a distance of 100 meters from the British Museum" can be therefore modeled as follows:

$(\texttt{verdi}, \texttt{d1}) : \texttt{hasDistance}$
$(\texttt{d1}, \texttt{british\_museum}) : \texttt{isDistanceFor}$
$(\texttt{d1}, 100) : \texttt{hasValue}$

After fuzzy granulation, the imprecise sentence " Many hotels have a low distance from attractions"

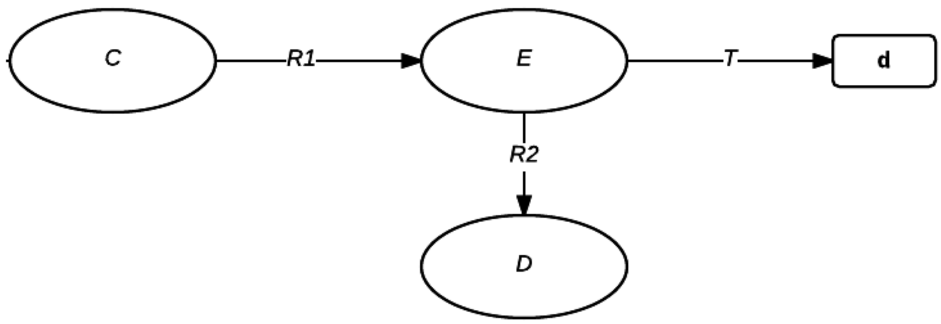


Figure 9. Graphical representation of the OWL schema modeling a ternary relation.

can be considered as a consequence of the previous and the following axioms and assertions:

$\texttt{LowDistance} \equiv \texttt{Distance} \sqcap \exists\texttt{isDistanceFor.Attraction} \sqcap \exists\texttt{hasValue.Low}$
$\texttt{d1} : \texttt{LowDistance}$ (to some degree)
$(\texttt{d1}, \texttt{ld}) : \texttt{mapsTo}$
$(\texttt{ld}, 0.5) : \texttt{hasCardinality}$
$\texttt{ld} : \exists\texttt{hasCardinality.Many}$ (to some degree)

where Many is defined as mentioned in Example 3.1.

4. Experimental results

The method presented in Sect. 3 has been implemented in Python[5]. It interfaces a Apache Jena[6] server by posing SPARQL[7] queries to an OWL ontology in order to extract data in the CSV format. The granulation process can be tuned by means of JSON[8] configuration files. The resulting granular view is then integrated in the original OWL ontology by means of additional axioms and assertions according to the syntax of Fuzzy OWL 2 as shown in the following specification in OWL/XML format:

```
<ClassAssertion>
    <Annotation>
        <AnnotationProperty IRI="#fuzzyLabel"/>
        <Literal datatypeIRI="&rdf;PlainLiteral">
            <fuzzyOwl2 fuzzyType="axiom">
                <Degree value="{degree}"/>
```

[5] https://www.python.org/
[6] https://jena.apache.org/
[7] https://www.w3.org/TR/sparql11-overview/
[8] http://www.json.org/

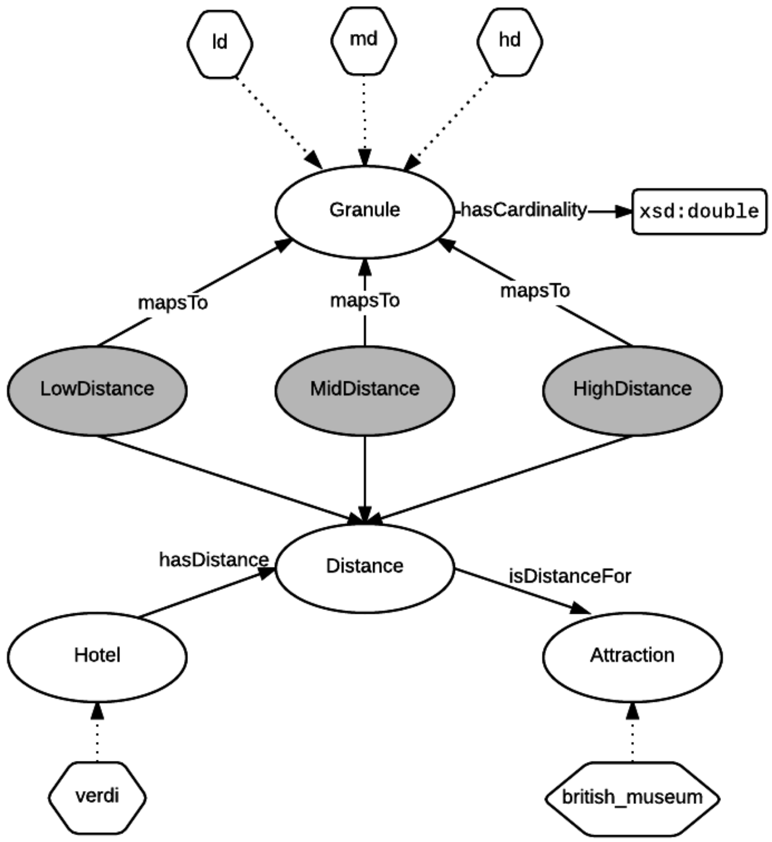


Figure 10. Graphical representation of the output of the fuzzy granulation process on the OWL schema reported in Fig. 9 and instantiated with the concepts used in Example 3.3.

```
            </fuzzyOwl2>
        </Literal>
    </Annotation>
    <ObjectSomeValuesFrom>
        <ObjectProperty IRI="#hasCardinality"/>
        <Class IRI="#{quantifier}"/>
    </ObjectSomeValuesFrom>
    <NamedIndividual IRI="#{granule}"/>
</ClassAssertion>
```

Experiments have been conducted on the OWL ontology *Hotel*, already mentioned in Example 3.1, and described in more detail in the following subsection.

4.1. The OWL ontology

The ontology *Hotel* consists of 8000 axioms, 74 classes, 4 object properties, 2 data properties, and 1504 individuals. It has the expressivity of the $\mathcal{ALCHOF}(\mathbf{D})$ DL.

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.

4.2. Tests

4.2.1. Case 1

The program parameters are tuned by means of a JSON file which specifies the SPARQL Endpoint to be queried, the numerical data property to be fuzzified, the linguistic labels to be used for the fuzzy sets and the linguistic labels for the fuzzy quantifiers together with the prototype values. For instance, the following JSON file triggers the granular computing method on the data property `hasPrice` by setting the number of clusters to $n = 3$ with `Low`, `Mid`, and `High` as linguistic labels, and the number of fuzzy quantifiers to 5 with `AlmostNone`, `Few`, `Some`, `Many` and `Most` as linguistic labels and the values 0.05, 0.275, 0.5, 0.725 and 0.95 as prototypes.

```
{
    "ontologyName" : "Hotel.owl",
    "ontologyPrefix" : "<http://www.semanticweb.org/ontologies/Hotel.owl#>",
    "SPARQLEndPoint" : "http://localhost:3030/Hotel/query",

    "domainClasses" : ["Hotel"],

    "dataPropertyToFuzzify" : "hasPrice",
```

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

```
    "fuzzySetLabels" : ["Low", "Mid", "High"],

    "quantifierLabels" : ["AlmostNone","Few", "Some", "Many", "Most"],
    "quantifierPrototypes" : [0.05, 0.275, 0.5, 0.725, 0.95]
}
```

By interpreting the configuration file, the following SPARQL query is automatically generated[11], which extracts the pairs (hotel, price) that will be used for granulation:

```
PREFIX : <http://www.semantic.web.org/ontologies/Hotel.owl#>
SELECT ?hotel ?price
WHERE {
?hotel a :Hotel .
?hotel :hasPrice ?price . }
```

The lowest and highest value in the domain of hotel prices are automatically calculated; in this example they are $m = 45$ and $M = 136$, respectively.

Clusters have been automatically built around the centroids 54.821, 77.924 and 102.204. They form the SFP shown in Figure 11 which consists of a left-shoulder fuzzy set (labeled as `Low`), a triangular fuzzy set (labeled as `Medium`) and a right-shoulder fuzzy set (labeled as `High`).

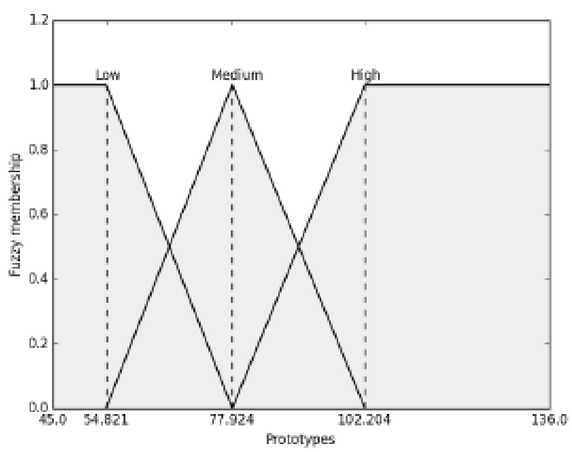
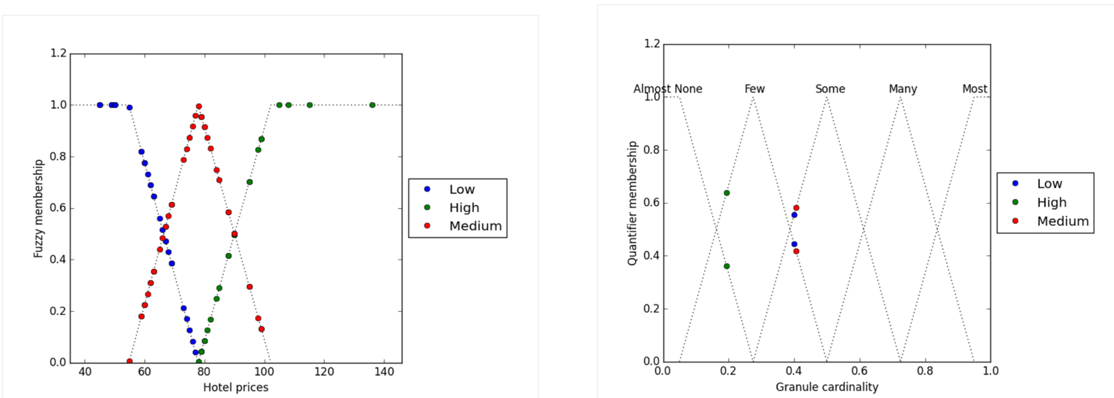


Figure 11. The generated Strong Fuzzy Partition for $n = 3$.

For each hotel price, the software computes its membership degree to the three fuzzy sets defined by the SFP. The results returned by the granulation process are graphically reported in Figure 12 (a). Hotel prices are reported along the x axis, whereas the y axis contains the membership degrees to the fuzzy sets. (For each price value, at most two membership degrees have non-zero values: these membership degrees are represented with different colors in the figure.)

Results for $n = 5$ and $n = 7$ the results of granulation are reported in Fig. 13(a), and Fig. 14(a) respectively. As expected, the higher is the number of clusters, the finer is the granulation of prices, with

[11] We use variables with readable names for the sake of intelligibility.



(a) (b)

Figure 12. Fuzzy granulation of prices according to $n = 3$ clusters (a) and fuzzy quantification of the resulting information granules (b).

fuzzy sets more concentrated in the region of the domain where date are more dense. (The number of information granules depends on the desired level of specificity for linguistically representing prices.)

For each fuzzy set in the partition, a subclass is defined a in Example 3.1. Moreover, for each individual (hotels in this case), one or two annotated assertions can be added to the ontology in order to establish the fuzzy membership of the individual to the corresponding information granules. For example, for individual `Hotel_002` the following assertions are included in the ontology:

`Hotel_002 : MediumPriceHotel` (0.709)

`Hotel_002 : HighPriceHotel` (0.291)

where membership degrees are embodied in the ontology through a Fuzzy OWL 2 annotation (fig. 15). In principle, the original data property `hasPrice` could be removed from the ontology and replaced with the new granular view of prices.

After price granulation, the resulting information granules can be quantified according to the fuzzy quantifiers specified in the JSON configuration file. The results are shown in figs. 12(b), 13(b) and 14(b). Each fuzzy information granule is represented by a new individual of the class `Granule` (e.g. `lph`, `mph` and `hph` for granules depicted in fig. 12). For these individuals, a number of axioms are generated as follows:

`lph :` $\exists$`hasCardinality.Few` (0.444)

`lph :` $\exists$`hasCardinality.Some` (0.556)

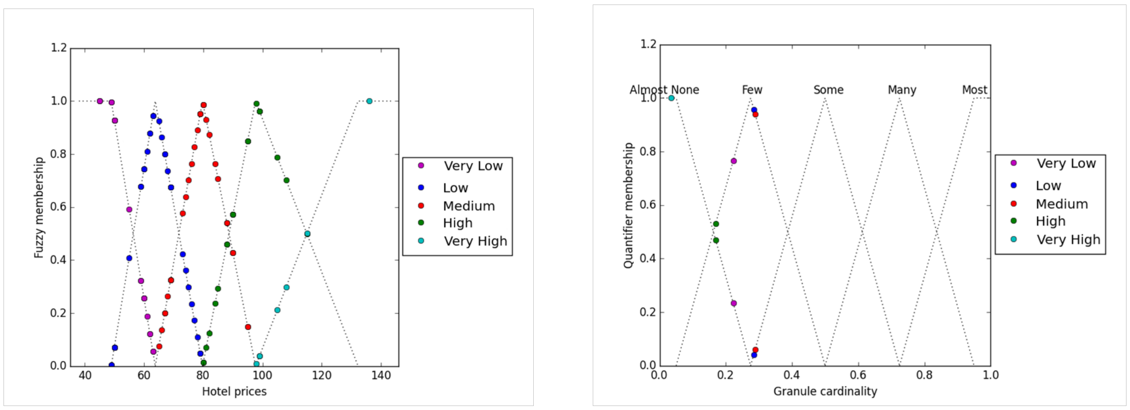
`mph :` $\exists$`hasCardinality.Few` (0.417)

`mph :` $\exists$`hasCardinality.Some` (0.583)

`hph :` $\exists$`hasCardinality.AlmostNone` (0.361)

`hph :` $\exists$`hasCardinality.Few` (0.639)

All in all, these axioms assert that there are very few hotels with high prices, some hotels with medium prices and some hotels with low prices.



(a) (b)

Figure 13. Fuzzy granulation of prices according to $n = 5$ clusters (a) and fuzzy quantification of the resulting information granules (b).

4.2.2. Case 2

The extension of the granulation process to subclasses is easily accomplished by a proper configuration of the JSON file. As an example, in order to achieve granulation of both `Hotel` and `Bed_and_Breakfast` classes, which are both subclasses of `Accommodation`, it suffices to change the `domainClasses` attribute into:

```
"domainClasses" : ["Hotel", "Bed_and_Breakfast"],
```

The results of fuzzy granulation are depicted in figs. 16 and 17, which show the difference in distribution of prices according to three information granules and five fuzzy quantifiers.

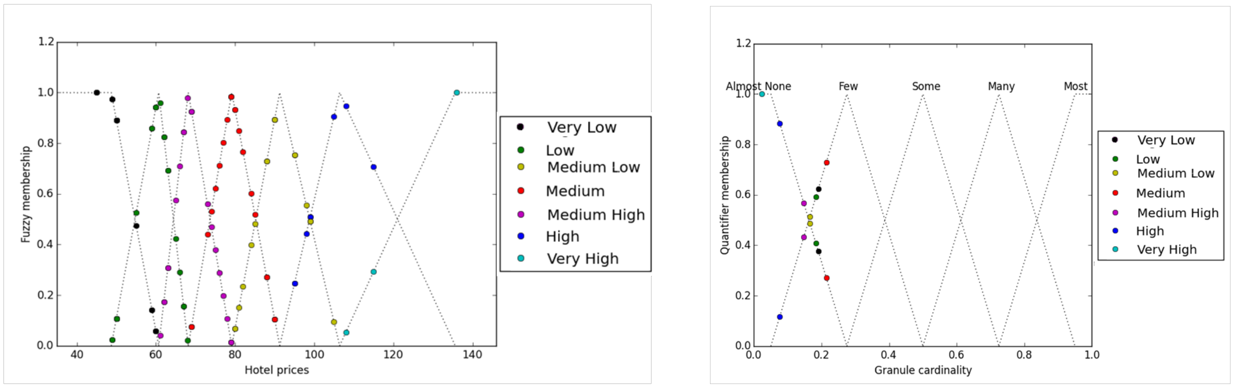
4.2.3. Case 3

The case 3 concerns ontology design patterns which represent ternary relations by means of binary relations and auxiliary classes. The JSON configuration file for the granulation of the distances between accommodations and attractions is reported below.

```
{
    "ontologyName" : "hotel_OWL.owl",
    "ontologyPrefix" : "<http://www.semanticweb.org/ontologies/Hotel.owl#>",
    "SPARQLEndPoint" : "http://localhost:3030/inf/sparql",

    "domainClasses" : ["Accomodation"],
    "rangeClasses" : ["Attraction"],

    "objectPropertyToAuxiliaryClass" : "hasDistance",
    "auxiliaryClass" : "Distance",
    "objectPropertyFromAuxiliaryClass" : "isDistanceFor",
```



(a) (b)

Figure 14. Fuzzy granulation of prices according to $n = 7$ clusters (a) and fuzzy quantification of the resulting information granules (b).

```
    "dataPropertyToFuzzify" : "hasValue",
    "fuzzySetsLabels" : ["Low", "Mid", "High"],

    "quantifiersLabels" : ["AlmostNone", "Few", "Some", "Many", "Most"],
    "quantifiersPrototypes" : [0.05, 0.275, 0.5, 0.725, 0.95]
}
```

The underlying SPARQL query extracts the quadruples of the kind (hotel, distance, attraction, value) upon which information granulation is applied.

```
PREFIX ontology:<http://www.semanticweb.org/ontologies/Hotel.owl#>
SELECT ?hotel ?distance ?attraction ?value
WHERE {
  ?hotel a ontology:Hotel.
  ?attraction a ontology:Bridge.
  ?hotel ontology:hasDistance ?distance.
  ?distance ontology:isDistanceFor ?attraction.
  ?distance ontology:hasValue ?value
}
```

Results are reported in Fig. 18. It is possible to observe a high concentration of distances in a region of the space where two information granules are mostly representative (“Mid” and “Low”). Correspondingly, the quantification of information granules synthetically represents the distribution of values: many hotels have low distance from attractions, few of them have medium distance and almost none of the hotels have high distance from attractions. This granular information can be integrated in the ontology by following the form depicted in Example 3.3. In principle, the granular view of distances can be used

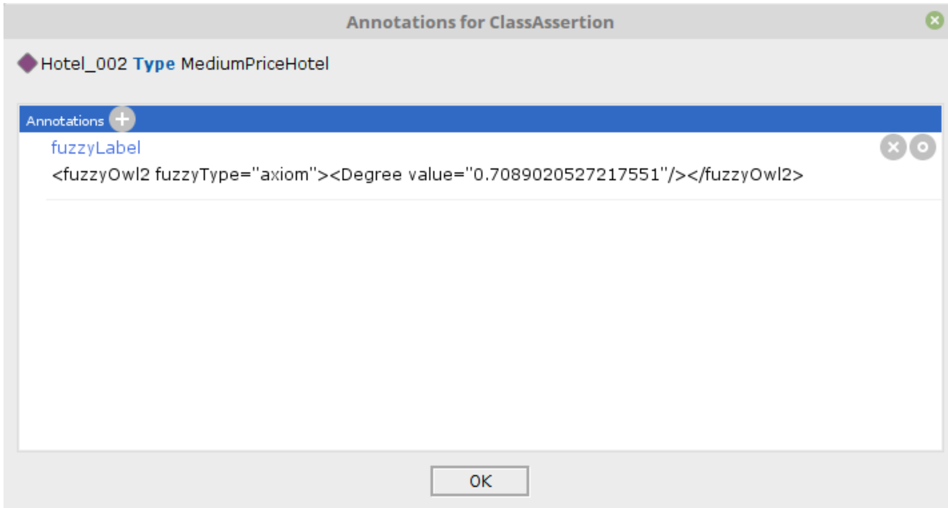


Figure 15. Annotation of membership degree of `Hotel_002` to `MediumPriceHotel` as seen in Protégé.

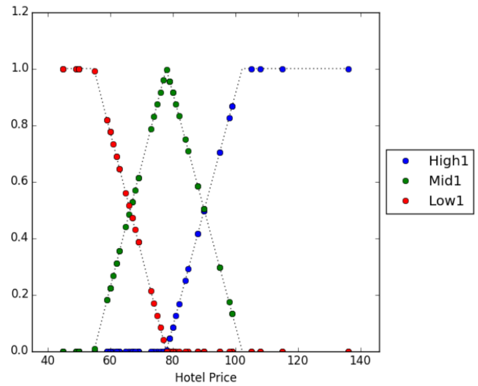
in place of their numerical representation, by removing the `hasValue` data property from the `Distance` class, therefore achieving a significant simplification of the knowledge base.

5. Conclusions

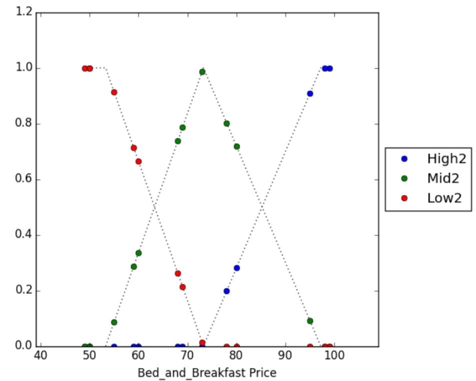
This paper presents a computational method for introducing a granular view of data within an OWL ontology. According to it, a number of individuals belonging to the ontology can be replaced by information granules, represented as fuzzy sets. One such view, obtained by applying Granular Computing techniques, is a highly desirable feature for many Semantic Web applications pervaded by imprecise and uncertain information coming from perceptual data, incomplete data, data with errors, etc. The proposed approach, though in an initial stage, paves the way for exploiting tolerance to imprecision in OWL ontologies, which may lead to concrete benefits such as compact knowledge representation, efficient reasoning and knowledge comprehensibility.

A preliminary version of the method has been already presented in [35]. However, with respect to the conference paper, we completely rewrote the sections describing the motivation and the theoretical framework. We also revised the description of the method and gave detailed experimental results on medium-size OWL ontology populated with real-world data. Notably, in our implementation of the method, we have chosen to represent the output of our fuzzy granulation method by using Fuzzy OWL 2 so that it could be easily integrated into the original ontology.

In the future we plan to verify the benefits of information granulation in the context of inductive learning algorithms, such as FOIL-$\mathcal{DL}$ [36], in terms of efficiency and effectiveness of the learning process, as well as in terms of interpretability of the learning results.



(a)



(b)

Figure 16. Fuzzy granulation of (a) hotel and (b) b&b prices, according to $n = 3$ clusters.

References

[1] Baader F, Calvanese D, McGuinness D, Nardi D, Patel-Schneider PF, editors. The Description Logic Handbook: Theory, Implementation and Applications (2nd ed.). Cambridge University Press; 2007.

[2] Horrocks I, Kutz O, Sattler U. The Even More Irresistible $\mathcal{SROIQ}$. In: Doherty P, Mylopoulos J, Welty CA, editors. Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006. AAAI Press; 2006. p. 57–67.

[3] Stoilos G, Simou N, Stamou G, Kollias S. Uncertainty and the Semantic Web. IEEE Intelligent Systems. 2006;21. doi:10.1109/MIS.2006.105.

[4] Zadeh LA. Is there a need for fuzzy logic? Information sciences. 2008;178(13):2751–2779.

[5] Alonso JM, Castiello C, Mencar C. Interpretability of Fuzzy Systems: Current Research Trends and Prospects. In: Kacprzyk J, Pedrycz W, editors. Springer Handbook of Computational Intelligence. Springer Berlin / Heidelberg; 2015. .

[6] Zadeh LA. Fuzzy Sets. Information and Control. 1965;8(3):338–353.

[7] Bargiela A, Pedrycz W. Human-centric information processing through granular modelling. vol. 182. Springer Science & Business Media; 2009.

[8] Straccia U. Foundations of Fuzzy Logic and Semantic Web Languages. CRC Studies in Informatics Series. Chapman & Hall; 2013.

[9] Bargiela A, Pedrycz W. Granular computing: an introduction. Springer Science & Business Media; 2003.

[10] Delgado M, Sánchez D, Vila MA. Fuzzy cardinality based evaluation of quantified sentences. International Journal of Approximate Reasoning. 2000;23(1):23–66. doi:10.1016/S0888-613X(99)00031-6.

[11] Zadeh LA. A computational approach to fuzzy quantifiers in natural languages. Computers & Mathematics with Applications. 1983;9(1):149–184. doi:10.1016/0898-1221(83)90013-5.

[12] Zimmermann HJ. Fuzzy set theory. Wiley Interdisciplinary Reviews: Computational Statistics. 2010 may;2(3):317–332. Available from: `http://doi.wiley.com/10.1002/wics.82`. doi:10.1002/wics.82.

[13] Zadeh LA. From computing with numbers to computing with words. From manipulation of measurements to manipulation of perceptions. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications. 1999;46(1):105–119. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=739259. doi:10.1109/81.739259.

[14] Trillas E, Termini S, Moraga C. A naïve way of looking at fuzzy sets. Fuzzy Sets and Systems. 2016 jun;292:380–395. Available from: http://linkinghub.elsevier.com/retrieve/pii/S0165011414003376. doi:10.1016/j.fss.2014.07.016.

[15] Toth H. Fuzziness: From Epistemic Considerations to Terminological Clarification. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems. 1997 aug;05(04):481–503. Available from: http://www.worldscientific.com/doi/abs/10.1142/S021848859700035X. doi:10.1142/S021848859700035X.

[16] Zadeh LA. The Information Principle. Information Sciences. 2015;294:540–549. doi:10.1016/j.ins.2014.09.026.

[17] de Oliveira JV. Semantic constraints for membership function optimization. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans. 1999;29(1):128–138. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=736369. doi:10.1109/3468.736369.

[18] Saaty TL, Ozdemir MS. Why the magic number seven plus or minus two. Mathematical and Computer Modelling. 2003;38(3):233–244. doi:10.1016/S0895-7177(03)90083-5.

[19] Bezdek JC. Fuzzy Clustering. In: Ruspini EH, Bonissone PP, Pedrycz W, editors. Handbook of Fuzzy Computation. Institute of Physics Pub.; 1998. p. 2.

[20] Dubois D, Prade H. Fuzzy cardinality and the modeling of imprecise quantification. Fuzzy sets and Systems. 1985;16(3):199–230.

[21] Liu Y, Kerre EE. An overview of fuzzy quantifiers. (I). Interpretations. Fuzzy Sets and Systems. 1998;95(1):1–21. doi:10.1016/S0165-0114(97)00254-6.

[22] Delgado M, Ruiz MD, Sánchez D, Vila MA. Fuzzy quantification: a state of the art. Fuzzy Sets and Systems. 2014;242:1–30.

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

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

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

[26] Baader F, Hanschke P. A Scheme for Integrating Concrete Domains into Concept Languages. In: Mylopoulos J, Reiter R, editors. Proceedings of the 12th International Joint Conference on Artificial Intelligence. Sydney, Australia, August 24-30, 1991. Morgan Kaufmann; 1991. p. 452–457.

[27] Horrocks I, Patel-Schneider PF, van Harmelen F. From $\mathcal{SHIQ}$ and RDF to OWL: The Making of a Web Ontology Language. Journal of Web Semantics. 2003;1(1):7–26.

[28] Straccia U. Description Logics with Fuzzy Concrete Domains. In: UAI '05, Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence, Edinburgh, Scotland, July 26-29, 2005. AUAI Press; 2005. p. 559–567.

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

[30] Sanchez D, Tettamanzi AGB. Fuzzy quantification in fuzzy description logics. In: Sanchez E, editor. Fuzzy Logic and the Semantic Web. vol. 1 of Capturing Intelligence. Elsevier; 2006. p. 135 – 159. Available from: http://www.sciencedirect.com/science/article/pii/S1574957606800109.

[31] Bobillo F, Straccia U. fuzzyDL: An expressive fuzzy description logic reasoner. In: FUZZ-IEEE 2008, IEEE International Conference on Fuzzy Systems, Hong Kong, China, 1-6 June, 2008, Proceedings. IEEE; 2008. p. 923–930.

[32] Bobillo F, Straccia U. The fuzzy ontology reasoner fuzzyDL. Knowledge-Based Systems. 2016;95:12–34.

[33] Bobillo F, Straccia U. Representing fuzzy ontologies in OWL 2. In: FUZZ-IEEE 2010, IEEE International Conference on Fuzzy Systems, Barcelona, Spain, 18-23 July, 2010, Proceedings. IEEE; 2010. p. 1–6.

[34] Bobillo F, Straccia U. Fuzzy ontology representation using OWL 2. International Journal of Approximate Reasoning. 2011;52(7):1073–1094.

[35] Lisi FA, Mencar C. Towards fuzzy granulation in OWL ontologies. In: Ancona D, Maratea M, Mascardi V, editors. Proceedings of the 30th Italian Conference on Computational Logic, Genova, Italy, July 1-3, 2015.. vol. 1459 of CEUR Workshop Proceedings. CEUR-WS.org; 2015. p. 144–158. Available from: http://ceur-ws.org/Vol-1459/paper19.pdf.

[36] Lisi FA, Straccia U. Learning in Description Logics with Fuzzy Concrete Domains. Fundamenta Informaticae. 2015;140(3-4):373–391. Available from: http://dx.doi.org/10.3233/FI-2015-1259. doi:10.3233/FI-2015-1259.

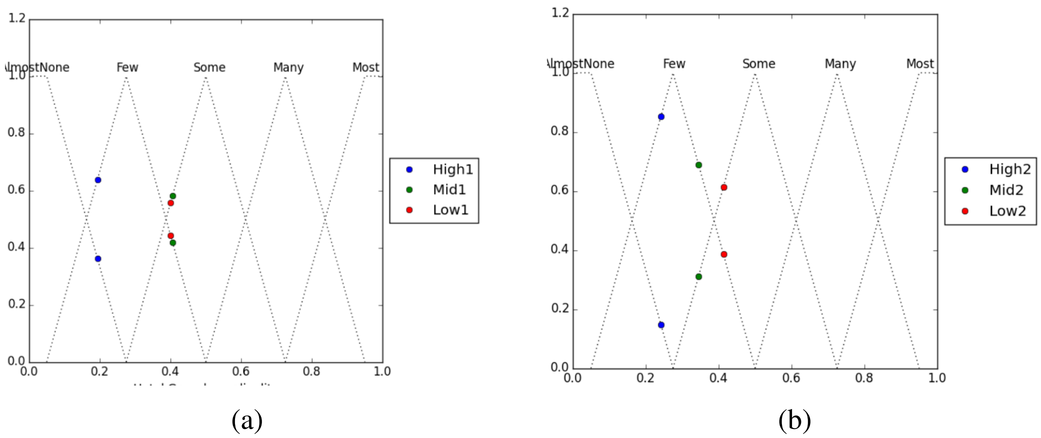


Figure 17. Fuzzy quantification of (a) hotels and (b) b&b prices, according to price.

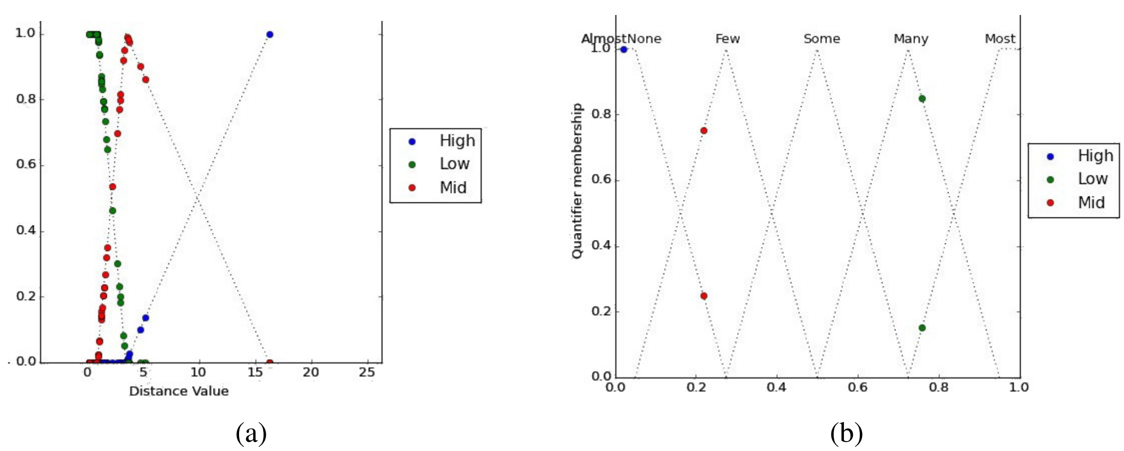


Figure 18. Fuzzy granulation of distances of hotels from bridges according to $n = 3$ clusters (a) and fuzzy quantification of the resulting information granules (b).