International Journal of Approximate Reasoning 87 (2017) 40–66

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Generalizing type-2 fuzzy ontologies and type-2 fuzzy description logics ☆

Fernando Bobillo [a,*], Umberto Straccia [b]

[a] Aragon Institute of Engineering Research (I3A), University of Zaragoza, Spain
[b] Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche (ISTI-CNR), Pisa, Italy

A R T I C L E I N F O

Article history:
Received 22 February 2016
Received in revised form 21 April 2017
Accepted 30 April 2017
Available online 5 May 2017

Keywords:
Fuzzy ontologies
Fuzzy description logics
Type-2 fuzzy sets
Logic for the semantic web

A B S T R A C T

In the last years, we are witnessing an increase of real-world applications of fuzzy ontologies. Most fuzzy ontologies are based on type-1 fuzzy logic, and type-2 fuzzy ontologies have not yet received such attention so far. Furthermore, there exists an important gap between type-2 knowledge representation formalisms (type-2 Description Logics) and type-2 fuzzy ontology applications. In this paper, we propose a formal framework for type-2 fuzzy ontologies taking into account the needs of existing applications. Essentially, our approach makes it possible to manage some uncertainty in the fuzzy membership functions used in the fuzzy datatypes and in the degrees of truth of the axioms. We define a type-2 Description Logic, a reasoning algorithm, and give a Fuzzy OWL 2 specification of it.

1. Introduction

In the last decade, *OWL ontologies* have become standard for knowledge representation. An ontology is an explicit and formal specification of the concepts, individuals and relationships that exist in some area of interest, created by defining axioms that describe the properties of these entities [52]. Ontologies can provide semantics to data, making knowledge maintenance, information integration, and reuse of components easier.

The theoretical underpinnings of ontologies are strongly based on *Description Logics* (DLs) [4]. DLs are a family of logics for representing structured knowledge that play a key role in the design of ontologies. Notably, DLs are essential in the design of OWL 2 (Web Ontology Language) [26], the current standard language to represent ontologies. As a matter of fact, OWL 2 is almost equivalent to the DL $\mathcal{SROIQ}(\mathbf{D})$ [33].

Despite the undisputed success of ontologies, it has been widely pointed out that classical ontologies are not appropriate to deal with imprecise and vague knowledge, inherent to several real world domains [56]. Fuzzy set theory and fuzzy logic [60] have proved to be suitable formalisms to handle these types of knowledge. Therefore *fuzzy ontologies* emerge as useful in several applications, such as [7,28,29,43,50,57].

Most of the times, fuzzy ontologies are based on the notion of type-1 fuzzy set. That is, while in classical set theory elements either belong to a set or not, elements can belong to a type-1 fuzzy set to some degree. Thus, fuzzy sets can be characterized by a membership function assigning to every element a degree of truth.

☆ This paper is part of the virtual special issue on Advances in Weighted Logics for Artificial Intelligence, edited by Marcelo Finger, Lluis Godo, Henri Prade and Guilin Qi.

* Corresponding author.
E-mail addresses: fbobillo@unizar.es (F. Bobillo), straccia@isti.cnr.it (U. Straccia).

http://dx.doi.org/10.1016/j.ijar.2017.04.012
0888-613X/

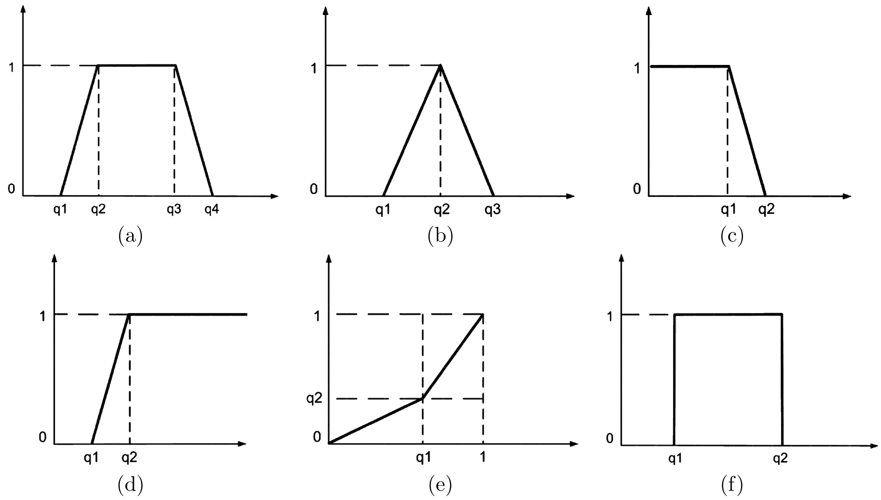

Fig. 1. (a) Trapezoidal function; (b) Triangular function; (c) Left-shoulder function; (d) Right shoulder function; (e) Linear function; (f) Crisp interval.

A limitation of these type-1 fuzzy sets is that they are based on the existence of well-known membership functions, with no uncertainty associated to them. However, in practice, it is often difficult to precisely know the membership function of a fuzzy set. A possible solution is to consider membership functions that are themselves fuzzy sets, yielding the so-called type-2 fuzzy sets [61]. It has been demonstrated in practice that type-2 fuzzy sets usually handle the imprecision related to the definition of fuzzy memberships in a better way.

Some type-2 fuzzy ontologies and type-2 fuzzy DLs have been proposed in the literature (see Section 3 for details), but current works are at a preliminary stage. On the one hand, the previously proposed type-2 fuzzy DLs are not expressive enough to cover the needs of type-2 fuzzy ontology applications presented so far. Moreover, current type-2 fuzzy ontology applications are not based on fuzzy DLs, so it is not possible to take advantage of automatic reasoning services, and developers must implement their own inference strategies.

In summary, in this paper a formal framework for type-2 fuzzy ontologies is proposed by taking into account the needs of existing applications. In particular, we define a type-2 fuzzy DL and provide a reasoning algorithm. Furthermore, we will discuss how to integrate this extension into existing type-1 fuzzy ontology languages and clarify the relation between type-1 fuzzy DLs and interval fuzzy DLs.

The remainder of this paper is organized as follows. Section 2 starts by providing some background on type-1 and type-2 fuzzy sets that will be needed to follow this paper. Section 3 performs a critical overview of the related work on type-2 ontologies and DLs. Then, we introduce type-1 and type-2 DLs in Section 4. Next, in Section 5 we state some results showing the lack of expressivity of previous works on type-2 DLs. After that, Section 6 discusses how to represent type-2 ontologies using fuzzy ontology languages and Section 7 provides a reasoning algorithm for general type-2 DLs. Finally, Section 8 sets out some conclusions and ideas for future research.

2. Preliminaries

Type-1 fuzzy logic. Fuzzy set theory and fuzzy logic were proposed by L.A. Zadeh [60] to manage imprecise and vague knowledge. While in classical set theory elements either belong to a set or not, in fuzzy set theory elements can belong to some degree. More formally, let X be a set of elements called the reference set. A *type-1 fuzzy subset* A of X is defined by a membership function $\mu_A(x)$, or simply $A(x)$, which assigns to every $x \in X$ a degree of truth, measured as a value in a truth space $\mathcal{N}$. The truth space is usually $\mathcal{N} = [0, 1]$, but other choices are possible. Indeed, $\mathcal{N}$ does not need to be a total order, nor does it need to be infinite. As in the classical case, 0 means no-membership and 1 full membership, but now a value between 0 and 1 represents the extent to which x can be considered as an element of the fuzzy set A.

Some popular membership functions, commonly used to define fuzzy sets, are the trapezoidal (Fig. 1(a)), the triangular (Fig. 1(b)), the left-shoulder (Fig. 1(c)), the right-shoulder (Fig. 1(d)), and the linear function (Fig. 1(e)).

Fuzzy logics provide compositional calculi of degrees of truth. The conjunction, disjunction, complement and implication operations are performed in the fuzzy case by a *t-norm* function $\otimes$, a *t-conorm* function $\oplus$, a *negation* function $\ominus$ and an *implication* function $\Rightarrow$, respectively. For a formal definition of these functions we refer the reader to [31,36].

A quadruple composed by a t-norm, a t-conorm, an implication function and a negation function determines a *fuzzy logic*. One usually distinguishes three fuzzy logics, namely Łukasiewicz, Gödel, and Product [31], due to the fact that any continuous t-norm can be obtained as a combination of Łukasiewicz, Gödel, and Product t-norms [49]. It is also usual to consider Zadeh logic, including the conjunction, disjunction, and negation originally proposed by Zadeh [60] together with Kleene–Dienes implication defined as $\alpha \Rightarrow_{KD} \beta = \max(1 - \alpha, \beta)$. The name of Zadeh fuzzy logic is used following the

Table 1
Combination functions of various fuzzy logics.

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

tradition in the setting of fuzzy DLs, even if it might lead to confusion because the logic does not include the sometimes called *Zadeh implication* (because it corresponds to Zadeh's fuzzy set inclusion) or Rescher implication, denoted $\Rightarrow_Z$ and defined as:

$$\alpha \Rightarrow_Z \beta = \begin{cases} 1 & \text{if } \alpha \leq \beta \\ 0 & \text{if } \alpha > \beta \,. \end{cases}$$

Table 1 summarizes the fuzzy operators in the main four fuzzy logics. We will often use the subscripts Ł, G, Π and Z to indicate that an operator belongs to Łukasiewicz, Gödel, Product, and Zadeh fuzzy logics, respectively.

An *involutive* negation verifies $\ominus(\ominus\alpha) = \alpha$ for every $\alpha \in [0, 1]$. For example, Łukasiewicz negation is involutive, while Gödel negation is not.

It is worth to introduce two important families of implications. An *S-implication* is an implication function obtained from a negation $\ominus$ and a t-conorm $\oplus$, i.e., $\alpha \Rightarrow \beta = \neg\alpha \oplus \beta$. An *R-implication* is an implication function obtained as the residuum of a continuous t-norm $\otimes$, i.e., $\alpha \Rightarrow \beta = \sup\{\gamma \mid \alpha \otimes \gamma \leq \beta\}$. Łukasiewicz, Gödel and Product implications are R-implications, while Kleene–Dienes and Rescher implications are not.

An interesting property of some fuzzy implications is the *ordering property* (OP), imposing an ordering on the underlying set of degrees of truth:

$$\alpha \Rightarrow \beta = 1 \Leftrightarrow \alpha \leq \beta, \ \forall \alpha, \beta \in [0, 1] \,. \tag{1}$$

Rescher implication and every R-implication satisfy (OP), but Kleene–Dienes implication does not [5].

We will also consider the non-filling natural negation property (NFNP) of some fuzzy implications:

$$\text{If } \alpha \in (0, 1], \text{ then } \alpha \Rightarrow 0 < 1 \tag{2}$$

Proposition 1. *Rescher, Kleene–Dienes, and R-implications satisfy (NFNP).*

Proof. In Rescher implication, $\alpha > 0$ implies $\alpha \Rightarrow 0 = 0$ by definition.

In Kleene–Dienes implication, $\alpha > 0 \Leftrightarrow -\alpha < 0 \Leftrightarrow 1 - \alpha < 1 \Leftrightarrow \max\{1 - \alpha, 0\} < 1 \Leftrightarrow \alpha \Rightarrow 0 < 1$.

Let us consider now the case of R-implications. Ad absurdum, assume that $\alpha \Rightarrow 0 = 1$. Since $\Rightarrow$ is an R-implication, it is the residuum of a left-continuous t-norm, so $\alpha \Rightarrow 0 = \sup\{\gamma \mid \alpha \otimes \gamma \leq 0\} = 1$. Thus, $\alpha \otimes 1 = \alpha \leq 0$, which contradicts the assumption that $\alpha \in (0, 1]$. Hence, $\alpha \Rightarrow 0 < 1$. □

Type-2 fuzzy logic. L. A. Zadeh proposed in 1975 an extension of type-1 fuzzy sets, called *type-2 fuzzy sets* [61]. In a type-2 fuzzy set, the membership function gives actually a fuzzy set for every element of the domain, so it can be seen as a fuzzy-fuzzy set [44]. This makes it possible to manage some uncertainty regarding the shape of the fuzzy set. Formally, a type-2 fuzzy subset A of a reference set X is characterized by a membership function $\mu_A : X \to \mathscr{P}(\mathcal{N})$, where $\mathscr{P}(\mathcal{N})$ is the power set of $\mathcal{N}$.

Although type-2 fuzzy sets have been shown to be useful in several applications, they are difficult to understand and manage in practice [45]. As a solution, it is common to restrict to *interval type-2 fuzzy sets*, where the membership function assigns to each element of the reference set an interval of the possible degrees of truth for the membership of the element to the fuzzy set. Formally, in this case, a type-2 fuzzy subset A of a reference set X is characterized by a membership function $\mu_A : X \to J$, where J is a set of subintervals of the truth space $\mathcal{N}$. Interval type-2 fuzzy sets are much easier to understand and the set operations are simplified [46].

To use interval type-2 fuzzy sets in practice, it is common to consider a pair of type-1 membership functions to define them: a lower type-1 membership function μ_L denoting the minimal values of the membership function of the type-2 fuzzy set, and an upper type-1 membership function μ_U denoting the maximal values of the membership function of the type-2 fuzzy set. The assumption here is that for all $x \in X$ $\mu_L(x) \leq \mu_U(x)$ holds. An example of such type-2 fuzzy set is shown in Fig. 2. For example, given the element x of the domain of discourse, its membership to the represented type-2 fuzzy set is given by the interval $[\mu_L(x), \mu_U(x)]$.

Fig. 2. Interval type-2 set described by an upper (green) and a lower (red) membership function. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

3. Related work

In this section, we will overview some closely related work. In particular, we will revise previous approaches using type-2 fuzzy ontologies in applications, discussing type-2 fuzzy ontology languages, or proposing type-2 fuzzy DLs.

Type-2 fuzzy ontologies. In a series of papers, Y.-G. Kim et al. discuss the use of type-2 fuzzy ontologies in different applications, namely in automatic hotel reservation [1], collision avoidance of autonomous underwater vehicles [2], and automatic air ticket booking [22]. In all these papers, the only difference with respect to classical ontologies are type-2 fuzzy datatypes, namely Fuzzy OWL 2 type-1 fuzzy datatypes extended to the type-2 case. For example, positive polarity and strong polarity are represented using pairs of triangular and right-shoulder functions, respectively.

C.-S. Lee et al. used type-2 fuzzy ontologies in different applications, such as diet assessment [40], diabetic-diet recommendation [38], malware behavior analysis [34], scheduling organizational meetings [37], and evaluating the human performance at playing the game of go [39]. All the developed type-2 fuzzy ontologies contain type-2 fuzzy datatypes as ranges of fuzzy data properties (they are called linguistic terms/labels and variables, respectively, by the authors). For example, the food ontology for diet assessment contains a fuzzy datatype representing a "low planned percentage of carbohydrates". Most of the ontologies are implemented in the type-2 Fuzzy Markup Language, which allows to define the upper membership function (UMF) and the lower membership function (LMF) of a term of type-2 fuzzy datatype.

J. Mezei et al. have considered interval fuzzy ontologies [24,47,48,59]. The particularity is that fuzzy datatypes can have two fuzzy membership functions defining the lower bound of an interval and the upper bound of an interval. The authors also report an application in the field of wines [24]. For example, a wine might have low alcohol with a degree in $[\alpha_1, \alpha_2]$. Such fuzzy membership functions could be obtained by combining the information provided by some experts using aggregation operators extended to interval-valued fuzzy numbers [48]. The authors have developed a wine recommender system and the main reasoning task is the maximum satisfiability degree of a fuzzy concept (an aggregation of criteria) given some individual (a wine) [58]. Type-2 fuzzy ontologies have also been used for intrusion detection in financial institutions [59]. The intrusion types (such as denial-of-service attack, malware and viruses) are modeled using interval fuzzy datatypes such as "Critical number of logins". Then the authors use similarity measures over interval-valued fuzzy sets to identify the most probable intrusion types. Type-2 fuzzy ontologies have also been applied to industrial paper machine maintenance [48]. Although no details are given in the publication for confidentiality reasons, the authors also relate individuals with interval-valued fuzzy datatypes.

Although these works illustrate the application of type-2 fuzzy ontologies to real-world problem solving, there are some limitations because of the lack of formal representation languages that could be problematic in other similar scenarios. They show the importance of establishing a link between the existing research on type-2 fuzzy ontologies and formal type-2 fuzzy DLs.

- Some of the design choices could be problematic if one wants to use a fuzzy ontology reasoner to infer new meaningful information. For example, in [1] the price of a hotel is represented as a subclass of room, which could clearly lead to confusing inferences.
- Similarly, it is mentioned in [47] that interval fuzzy numbers could be assigned to *"indicate different type of relationship between individuals and concepts (e.g.,* `partOf`*,* `isA`*)"*. When using a formal fuzzy DL language, this statement becomes misleading and needs to be clarified. In fuzzy DLs, the `is-a` relation links two classes and not an individual with a class. Furthermore, the membership degree of an individual to a concept should be expressed using an assertion axiom and not a relation. Finally, if a fuzzy set has a numerical domain, as it happens with the alcoholic degree of a wine, the common approach in fuzzy DLs is to represent it using a fuzzy datatype, and to relate an abstract individual, such as a particular type of wine, to the fuzzy datatype. Instead, the authors of [47] use a concept assertion to, for example, state that a wine belongs to the fuzzy set `HighAlcohol`.
- Some references have argued that conventional fuzzy ontology reasoners can be used. The authors of [1,2,22] claim that DeLorean reasoner [11] can be used to translate fuzzy type-2 ontologies into classical ontologies and to obtain

type-2 fuzzy inference results, and it is claimed in [58] that fuzzyDL reasoner [18] can be used as part of a type-2 wine recommender system. Unfortunately, this is not the case: DeLorean and fuzzyDL cannot currently represent or manage type-2 fuzzy ontologies. In order to use them, some additional processing (not explained in the literature) is needed.
- Finally, none of the previous approaches support type-2 axioms, that is, uncertainty regarding the degree of truth of an axiom.

Type-2 fuzzy languages. Nilavu Devadoss and Sivakumar Ramakrishnan discuss the representation of type-2 fuzzy rough ontologies using OWL 2 annotations [27], in a similar way to Fuzzy OWL 2 [14]. The authors explain that it is possible to represent type-2 elements using two properties `LowerDegree` and `UpperDegree`. However, the rest of the paper claims to provide a type-2 fuzzy rough OWL 2 syntax which is actually a type-1 fuzzy OWL 2 syntax. For instance, [27, Example 7] intends to define a type-2 fuzzy datatype `Tall` which is actually a type-1 fuzzy datatype because there is only one membership function.

Type-2 fuzzy Description Logics. There have also been proposals of type-2 fuzzy DLs. To the best of our knowledge, all of the existing proposals by other authors are restricted to interval fuzzy DLs.

- L. Huo et al. proposed an interval fuzzy extension of the DL $\mathcal{ALCN}$ [35]. The novelty is the existence of concept and role assertions of the form $\langle \mathfrak{a}, [\alpha_1, \alpha_2] \rangle$, where $\mathfrak{a}$ is a crisp assertion. The semantics is restricted to an interval Zadeh fuzzy logic. A tableau algorithm checking consistency is provided.
- A. Bahri et al. proposed an interval fuzzy extension of the DL $\mathcal{EL}^{++}$ [6]. The particularity is the existence of General Concept Inclusion (GCI) axioms of the form $\langle C_1 \sqsubseteq C_2, [\alpha_1, \alpha_2] \rangle$. Role inclusion axioms are not extended to the type-2 fuzzy case. The semantics is defined in terms of general type-2 operators, but the reasoning algorithm (to solve the subsumption algorithm) is restricted to interval Łukasiewicz fuzzy logic. Adding an upper bound to a GCI axiom is controversial because it does not have an equivalent counterpart in the crisp case, so the type-2 fuzzy logic is not a sound extension of classical $\mathcal{EL}^{++}$. Furthermore, S. Borgwardt et al. have recently shown that Łukasiewicz $\mathcal{EL}^{++}$ has an exponential complexity [20], so the algorithm in [6] with a polynomial complexity cannot be correct.
- R. Li et al. proposed an interval fuzzy extension of the DL $\mathcal{ALC}$ [41]. The novelty of this logic is that intervals are attached to concept and role expressions, so it is possible to express, for example, the concept $\forall \mathtt{hasFriend}_{[0.4,0.5]}.\mathtt{Tall}_{[0.6,0.7]}$. The axioms have the same syntax as in classical DLs. Adding an upper bound to a role expression in logics without role negation (such as $\mathcal{ALC}$) is controversial because it does not have an equivalent counterpart in the crisp case, so the type-2 fuzzy logic is not a sound extension of classical $\mathcal{ALC}$. The semantics is based on any interval t-norm, its dual t-conorm with respect to the standard interval negation and its corresponding interval S-implication. A type-2 fuzzy extension of the language OWL 1 is also proposed, where concept expressions can have a lower and a upper degree of truth. The authors also define a tableau algorithm to solve the subsumption problem as a reduction to concept unsatisfiability. The problem is that this reduction is valid if the semantics has an S-implication, as it happens in Zadeh and Łukasiewicz fuzzy logics, but not in general.

As we will show in Section 5, the type-2 extension of all these approaches does not add enough expressivity. Indeed, interval type-2 axioms in [6,35] can be simulated using type-1 fuzzy DLs, and type-2 concept expressions in [41] can be represented using threshold concepts.[1]

4. Type-0, type-1, interval and type 2 fuzzy description logics

In this section, we define the logics that we will use in this paper. Firstly, we will introduce different types of datatypes and truth-types: type-0, type-1, and type-2. Then, we will start with a classical (type-0) language and next we will consider several fuzzy DLs. After a type-1 DL, usually called just “fuzzy DL” in the literature, we will consider two extensions: an interval DL language and a type-2 one.

It is worth to anticipate that there is not a one-to-one correspondence between type-i ($i \in \{0, 1, 2\}$) datatypes, truth-types, and DLs. For instance, an axiom with a type-1 truth-type will produce a type-2 DL.

Interval DLs have already been presented in the literature by other authors (see Section 3), although they were called type-2 DLs. The contribution of this section is the generalization of these approaches to real type-2 DLs because, as we will see in Section 5, interval axioms can be represented using type-1 axioms.

For simplicity, we will consider variants of the well-known logic $\mathcal{ALC}$ to illustrate our results, although the results can be generalized to more expressive logics. For ease of presentation, we will consider a truth space $\mathcal{N} \subseteq [0, 1]$ with $\mathcal{N}$ containing $\{0, 1\}$. Our approach can be generalized to other sets of degrees of truth.

[1] Except for the case of thresholded roles.

4.1. Type-0, type-1 and type-2 fuzzy datatypes

A *(type-0) datatype theory* [42] $\mathbf{D} = \langle \Delta^{\mathbf{D}}, \cdot^{\mathbf{D}} \rangle$ 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 unary datatype $\mathbf{d}$ a unary function $\mathbf{d}^{\mathbf{D}}$ from $\Delta^{\mathbf{D}}$ to $\{0, 1\}$. An example of datatype domain is the set of integers and an example of datatype $\mathbf{d}$ over the integers is $\leq_5$ denoting the set of integers less or equal than 5.

A *fuzzy (type-1) datatype theory* [55] $\mathbf{D} = \langle \Delta^{\mathbf{D}}, \cdot^{\mathbf{D}} \rangle$ 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 unary fuzzy datatype $\mathbf{d}$ a unary fuzzy membership function $\mathbf{d}^{\mathbf{D}}$ from $\Delta^{\mathbf{D}}$ to the truth space $\mathcal{N}$. Some examples of type-1 datatypes can be built using the functions in Fig. 1.[2] For example, one can define the type-1 datatype `about15` as $\mathtt{tri}_{10,15,20}$.

Finally, a *fuzzy type-2 datatype* consists of a pair $\mathbf{d} = (\mathbf{d}_L, \mathbf{d}_U)$ of unary type-1 datatypes $\mathbf{d}_L$ and $\mathbf{d}_U$ (e.g., such as those in Fig. 2) indicating a lower and an upper membership function, respectively. A *fuzzy type-2 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 fuzzy type-2 datatype $\mathbf{d} = (\mathbf{d}_L, \mathbf{d}_U)$ a fuzzy membership function $\mathbf{d}^{\mathbf{D}}$ from $\Delta^{\mathbf{D}}$ to $\mathcal{N}$ such that for all $v \in \Delta^{\mathbf{D}}$, (i) ${\mathbf{d}_L}^{\mathbf{D}}(v) \leq {\mathbf{d}_U}^{\mathbf{D}}(v)$[3]; and (ii)

$$\mathbf{d}^{\mathbf{D}}(v) \in [{\mathbf{d}_L}^{\mathbf{D}}(v), {\mathbf{d}_U}^{\mathbf{D}}(v)] \tag{3}$$

hold. Note that we assign to each type-2 set a single value and not an interval or an element in $\mathscr{P}(\mathcal{N})$. Note also that $\mathbf{d}_L$ and $\mathbf{d}_U$ do not need to be of the same shape (e.g., $\mathbf{d}_L$ can be a triangular function and $\mathbf{d}_U$ a trapezoidal function).

Remark 1. A fuzzy type-1 datatype $\mathbf{d}$ can be expressed as a fuzzy type-2 datatype $(\mathbf{d}, \mathbf{d})$. For ease of presentation, a type-2 datatype of the form $(\mathbf{d}, \mathbf{d})$, will be denoted simply by $\mathbf{d}$ if no ambiguity arises.

4.2. Type-0, type-1 and type-2 truth-types

Like for fuzzy datatypes, we may define *truth-types*, which are essentially like fuzzy datatypes, but the domain of them is restricted to be the truth space $\mathcal{N}$. This will allow us to express statements such as "it is *more or less true* that Bob is tall".

Formally, a *(type-0) truth-type theory* $\mathbf{T} = \langle \mathcal{N}, \cdot^{\mathbf{T}} \rangle$ consists of the truth space $\mathcal{N}$ and a mapping $\cdot^{\mathbf{T}}$ such that $\alpha^{\mathbf{T}} = \alpha$, for each truth-type $\alpha \in \mathcal{N}$. α is called a *type-0 truth-type*.

A *fuzzy (type-1) truth-type theory* $\mathbf{T} = \langle \mathcal{N}, \cdot^{\mathbf{T}} \rangle$ consists of the truth space $\mathcal{N}$ and a mapping $\cdot^{\mathbf{T}}$ such that $\alpha^{\mathbf{T}} = \alpha$ and that assigns to every unary *type-1 truth-type* $\mathbf{t}$ a unary fuzzy membership function $\mathbf{t}^{\mathbf{T}} : \mathcal{N} \to \mathcal{N}$ (see, e.g., Fig. 1).

Finally, a *type-2 truth-type* consists of a pair $\mathbf{t} = (\mathbf{t}_L, \mathbf{t}_U)$ of unary type-1 truth-types $\mathbf{t}_L$ and $\mathbf{t}_U$ (e.g., such as those in Fig. 2) indicating a lower and an upper membership function, respectively. A *type-2 truth-type theory* $\mathbf{T} = \langle \mathcal{N}, \cdot^{\mathbf{T}} \rangle$ consists of the truth space $\mathcal{N}$ and a mapping $\cdot^{\mathbf{T}}$ such that $\alpha^{\mathbf{T}} = \alpha$ and that assigns and to every type-2 truth-type $\mathbf{t} = (\mathbf{t}_L, \mathbf{t}_U)$ a fuzzy membership function $\mathbf{t}^{\mathbf{T}} : \mathcal{N} \to \mathcal{N}$ such that for all $\alpha \in \mathcal{N}$, (i) ${\mathbf{t}_L}^{\mathbf{T}}(\alpha) \leq {\mathbf{t}_U}^{\mathbf{T}}(\alpha)$; and (ii)

$$\mathbf{t}^{\mathbf{T}}(\alpha) \in [{\mathbf{t}_L}^{\mathbf{T}}(\alpha), {\mathbf{t}_U}^{\mathbf{T}}(\alpha)] \, . \tag{4}$$

Remark 2. Note again that, as for type-2 datatypes, we assign to each type-2 truth-type a single value and not an interval or an element in $\mathscr{P}(\mathcal{N})$, and that $\mathbf{t}_L$ and $\mathbf{t}_U$ do not need to be of the same shape.

Furthermore, a type-0 truth-type α can be expressed as a type-1 truth-type $\mathtt{crisp}_{\alpha,\alpha}$, a type-1 truth-type $\mathbf{t}$ can be expressed as a type-2 truth-type $(\mathbf{t}, \mathbf{t})$ and for ease of presentation a type-2 truth-type of the form $(\mathbf{t}, \mathbf{t})$ will be denoted simply by $\mathbf{t}$ if no ambiguity arises.

4.3. Syntax

The main ingredients of fuzzy DLs are *fuzzy concepts* (or classes), which denote unary predicates, *fuzzy properties* (or roles), which denote binary predicates, *individuals* (or instances), and *fuzzy datatypes* (defined using fuzzy membership functions). *Axioms* are formal statements involving these elements, like a recipe that defines how to combine the previous ingredients to represent the knowledge of some particular domain. A *fuzzy knowledge base* (or *fuzzy ontology*) $\mathcal{O}$ is a finite set of axioms. The syntax of a fuzzy DL is determined by the available constructors for building complex concepts and roles from simpler ones, and by the available axioms.

Individuals are split into two categories: *abstract individuals*, that are instances of the concepts, and *concrete individuals* (or data values), that belong to a *concrete domain* (or *datatype domain*) which are already structured and whose structure is already known to the machine, such as integers, reals, or strings. Similarly, roles are split into two categories: *abstract*

[2] Note that these membership functions assume that the domain of a fuzzy set is a dense total ordering, which does not need to be the case. For example, discretized version of the functions Fig. 1(a)–(d) are also possible [8].

[3] Fig. 2 illustrates a pair of membership functions satisfying this restriction.

roles (or object properties), that link two abstract individuals, and *concrete roles* (or data properties), that relate an abstract individual and a data value.

We will use the following notation: A denotes an atomic concept (or *concept name*), C is a (possibly complex) concept, R is an object property, T is a datatype property, o is an abstract individual, $\mathtt{v}$ is a concrete value, $\mathbf{d}$ is a unary datatype, $\alpha, \beta, \gamma \in \mathcal{N}$ are type-0 truth-types (with $\alpha \neq 0$ and $\beta \neq 1$ to avoid tautologies), and $\mathbf{t}$ is a type-2 truth-type.

4.3.1. Type-0 DLs

Consider a (type-0) datatype theory $\mathbf{D} = \langle \Delta^{\mathbf{D}}, \cdot^{\mathbf{D}} \rangle$. In our DL, all the properties are atomic, but complex concepts can be built inductively from simpler ones as according to the following syntactic rule:

$$C \rightarrow \top \mid \bot \mid A \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \neg C \mid \exists R.C \mid \forall R.C \mid \exists T.\mathbf{d} \mid \forall T.\mathbf{d}$$

Furthermore, a knowledge base (or ontology) can include the following axioms: concept assertions $a{:}C$, object property assertions $(a_1, a_2){:}R$, data property assertions $(a, \mathtt{v}){:}T$, General Concept Inclusions (GCI) $C_1 \sqsubseteq C_2$, and concept equivalences $C_1 \equiv C_2$. A fuzzy ontology is usually split into two parts: an Assertional Box (*ABox*), with the three former types of axioms, and a Terminological Box (*TBox*), with the two latter types of axioms.

Remark 3. In type-0 ontologies, concept equivalences are syntactic sugar: $C_1 \equiv C_2$ can be represented with the pair of axioms $C_1 \sqsubseteq C_2$ and $C_2 \sqsubseteq C_1$.

4.3.2. Type-1 DLs

Consider a fuzzy (type-1) datatype theory $\mathbf{D} = \langle \Delta^{\mathbf{D}}, \cdot^{\mathbf{D}} \rangle$. The syntax of concepts and roles in type-1 DLs usually is the same as in type-0 DLs. However, it is also possible to extend the logic by considering threshold and implication concepts, as we will discuss later.

It is also worth to note that $\mathbf{d}$ is now a type-1 datatype. For instance, we can consider the functions in Fig. 1, where $\mathtt{crisp}_{q_1,q_2}$ allows backwards compatibility with type-0 datatypes.

The most important syntactical difference in type-1 DLs is that it has a fuzzy knowledge base (or fuzzy ontology) where some axioms (namely, concept assertions, role assertions, and GCIs) are of type-1 and can include an inequality involving a numerical (type-0) degree of truth, as shown in Table 2. An axiom of the form $\langle \mathfrak{a} \geq \alpha \rangle$ (resp. $\langle \mathfrak{a} \leq \beta \rangle$) states that $\mathfrak{a}$ holds with degree greater or equal (resp. less or equal) than α (resp. β).

Type-1 DLs also have concept equivalences with the same syntax as in type-0 DLs.

Remark 4. A type-0 axiom $\mathfrak{a}$ can be seen as a type-1 axiom of the form $\langle \mathfrak{a} \geq 1 \rangle$. For ease of presentation, if clear from context, we may also write simply $\mathfrak{a}$ in place of $\langle \mathfrak{a} \geq 1 \rangle$.

Please note that axioms of the form $\langle (a, b){:}R \leq \beta \rangle$ are usually only allowed in logics with role negation, such as $\mathcal{SROIQ}(\mathbf{D})$. Otherwise, the fuzzy language is not a sound extension of its crisp counterpart. If we restrict to the degrees $\{0, 1\}$, the axiom $\langle (a, b){:}R \leq 0 \rangle$ can be represented in a classical DL as $(a, b){:}\neg R$, but it cannot be represented in a classical DL without negative role assertions. However, we will allow these role assertions because they will be used in some of the results in Section 5.

In the classical case, an implication $C_1 \rightarrow C_2$ can be represented as the disjunction between C_2 and the negation of C_1. In fuzzy ontologies, this is not true in general, so it is common to include implication concepts in type-1 languages. In fuzzy ontologies, it is also possible to introduce threshold concepts $[C \geq \alpha]$ and $[C \leq \beta]$ [18,23] specifying an upper or lower bound for the degree of membership of a concept. To support these elements, the syntax of fuzzy concepts can be extended as follows:

$$D \rightarrow C \mid D_1 \rightarrow D_2 \mid D[\geq \alpha] \mid D[\leq \beta]$$

In this paper, we will not require that our fuzzy DL language includes implication or threshold concepts, unless stated otherwise. Implication concepts will only be used in Propositions 5 (actually, in the discussion about its premises) and 6, as well as in our reasoning algorithm (Section 7), whereas threshold concepts will only be used in Proposition 6.

Before concluding this definition of a type-1 DL, it is worth to observe some syntactic sugar results.

Remark 5. If a fuzzy language $\mathcal{X}$ includes a conjunction and an implication $\rightarrow$ which corresponds to its residuum, then the constructors $\sqcap_G$ and $\sqcup_G$, called Gödel conjunction and disjunction, respectively, are representable as follows:

- $C_1 \sqcap_G C_2 := C_1 \sqcap (C_1 \rightarrow C_2)$
- $C_1 \sqcup_G C_2 := ((C_1 \rightarrow C_2) \rightarrow C_2) \sqcap_G ((C_2 \rightarrow C_1) \rightarrow C_1)$

Remark 6. If the fuzzy implication used in the semantics of GCIs satisfies (OP), (Eq. (1)), concept equivalences are syntactic sugar: $C_1 \equiv C_2$ can be represented with the pair of axioms $\langle C_1 \sqsubseteq C_2 \geq 1 \rangle$ and $\langle C_2 \sqsubseteq C_1 \geq 1 \rangle$. As we will discuss later, our reasoning algorithm will impose this assumption, so it will not need to consider concept equivalences explicitly.

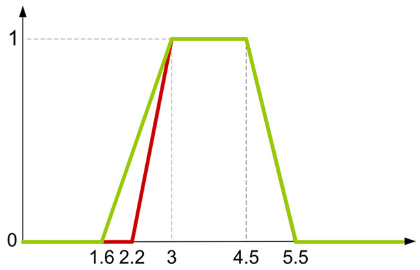

Fig. 3. Interval type-2 set described by the pair of trapezoidal membership functions $\mathtt{trap}_{2.2,3,4.5,5}$ and $\mathtt{trap}_{1.6,3,4.5,5.5}$.

4.3.3. Interval DLs

In *interval* DLs, the axioms are of the form $\langle \mathfrak{a}, [\gamma_1, \gamma_2] \rangle$, where $\mathfrak{a} \in \{a{:}C, (a_1, a_2){:}R, (a, \mathrm{v}){:}T, C_1 \sqsubseteq C_2\}$. The intuition is that the degree of satisfaction of $\mathfrak{a}$ is between γ_1 and γ_2. If γ_1 and γ_2 are omitted, 1 and 1 are assumed.

Remark 7. Type-1 axioms $\langle \mathfrak{a} \geq \alpha \rangle$ and $\langle \mathfrak{a} \leq \beta \rangle$ can be seen as interval axioms of the forms $\langle \mathfrak{a}, [\alpha, 1] \rangle$ and $\langle \mathfrak{a}, [0, \beta] \rangle$, respectively. This is the reason why the axiom considers $[\gamma_1, \gamma_2]$ instead of $[\alpha, \beta]$: we do not want to avoid the cases $\gamma_1 = 0$ and $\gamma_2 = 1$.

4.3.4. Type-2 DLs

Type-2 DLs extend the logic with type-2 datatypes, type-2 truth-types and type-2 axioms. The main characteristic of a type-2 axiom is that it includes a type-2 truth-type instead of a unique numerical or symbolic degree of truth.

So, consider a type-2 datatype theory $\mathbf{D} = \langle \Delta^{\mathbf{D}}, \cdot^{\mathbf{D}} \rangle$ and a type-2 truth-type theory $\mathbf{T} = \langle \mathcal{N}, \cdot^{\mathbf{T}} \rangle$. Type-2 DLs support axioms of the form $\langle \mathfrak{a}, \mathbf{t} \rangle$, where $\mathfrak{a} \in \{a{:}C, (a, b){:}R, (a, \mathrm{v}){:}T, C_1 \sqsubseteq C_2\}$ and $\mathbf{t}$ is a type-2 truth-type, as shown in Table 2. Note that by Remarks 1 and 2, **d** and **t** can be also of type-1.

The following examples illustrate how type-2 fuzzy membership functions can be used to restrict the range of a concrete role, as usual in DLs, and to define the truth degree of an axiom.

Example 1. Consider the type-2 datatype *MediumRank* defined in [39] and representing a medium value for the evaluation of the human performance at some activity. The type-2 datatype is represented as the pair of membership functions $\mathtt{trap}_{2.2,3,4.5,5}$ and $\mathtt{trap}_{1.6,3,4.5,5.5}$, as illustrated in Fig. 3. In a type-2 fuzzy DL, we can use the same pair to represent the type-2 fuzzy set of elements having a medium ranking as $\exists hasRanking.(\mathtt{trap}_{2.2,3,4.5,5}, \mathtt{trap}_{1.6,3,4.5,5.5})$.

Example 2. $\langle bob{:}Tall \geq 0.4 \rangle$ is a type-1 concept assertion, $\langle bob{:}Tall, [0.4, 0.9] \rangle$ is an interval concept assertion, and $\langle bob{:}Tall, tri_{0.6,0.8,1} \rangle$ is a type-2 concept assertion. In all these axioms there is some uncertainty in the sense that the membership degree of *bob* to the fuzzy concept *Tall* can not exactly be determined. The advantage of the latter axiom is that not all the possible membership degree values are equipossible. For example, in the latter assertion a membership degree 0.8 is more possible than the degree 1.

Note that the three fuzzy DLs (type-1, interval, and type-2) have the same concepts, and that only the axioms, datatypes and truth-types can be different. Note also that our previous work [13] only considers axioms of the form $\langle \mathfrak{a}, \mathbf{t} \rangle$ where $\mathbf{t}$ is a type-1 truth-type. It is also worth to stress that our syntax allows a double use of fuzzy membership functions in the same axiom, as the following example shows.

Example 3. The axiom $\langle bob{:}\exists hasAge.tri_{35,40,45}, tri_{0.6,0.8,1} \rangle$ encodes the fact that *bob* is around 40 with a degree of truth around 0.8.

The following remark shows how the membership function $\mathtt{crisp}$ is helpful to define to some useful equivalences.

Remark 8. An interval axiom $\langle \mathfrak{a}, [\gamma_1, \gamma_2] \rangle$ can be seen as a type-2 axiom of the form $\langle \mathfrak{a}, \mathtt{crisp}_{\gamma_1,\gamma_2} \rangle$. Furthermore, type-2 axioms of the form $\langle (a, \mathrm{v}){:}T, \mathbf{t} \rangle$ are syntactic sugar as they can be represented as $\langle a{:}\exists T.\mathtt{crisp}_{\mathrm{v},\mathrm{v}}, \mathbf{t} \rangle$. Hence, we will not need to consider concrete role assertions explicitly.

As discussed in Remarks 4, 7, and 8, type-0, type-1 and interval axioms can be represented as type-2 axioms. Type-0, type-1 and interval axioms will be called *unqualified fuzzy axioms*, whereas type-2 axioms that cannot be represented as unqualified axioms will be called *qualified fuzzy axioms*. Therefore, in some cases, we will assume that a fuzzy ontology is partitioned into two parts, $\mathcal{O} = \mathcal{O}_Q \cup \mathcal{O}_U$, where $\mathcal{O}_Q$ contains qualified axioms only and $\mathcal{O}_U$ contains unqualified axioms only.

4.4. Semantics

Now, we will define the formal semantics of the previously introduced logics.

4.4.1. Type-0 DLs

The semantics of crisp DLs is defined using a crisp interpretation relative to a type-0 datatype theory **D**. An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ relative to **D** consists of a nonempty set $\Delta^{\mathcal{I}}$ (the *domain*) disjoint from $\Delta^{\mathbf{D}}$, and of an *interpretation function* $\cdot^{\mathcal{I}}$ that coincides with $\cdot^{\mathbf{D}}$ on every data value and datatype, and it assigns:

- to each individual a an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$;
- to each data value $\mathtt{v}$ an element $\mathtt{v}^{\mathbf{D}} \in \Delta^{\mathbf{D}}$;
- to each atomic concept A a function $A^{\mathcal{I}}\colon \Delta^{\mathcal{I}} \to \{0, 1\}$;
- to each object property R a function $R^{\mathcal{I}}\colon \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to \{0, 1\}$;
- to each data property T a function $T^{\mathcal{I}}\colon \Delta^{\mathcal{I}} \times \Delta^{\mathbf{D}} \to \{0, 1\}$.

The interpretation function is extended to *complex concepts*, and *axioms* as in Table 2 (where $x \in \Delta^{\mathcal{I}}$). $\mathcal{I} \models \tau$ denotes that $\mathcal{I}$ *satisfies* (*is a model of*) axiom τ. $\mathcal{I}$ *satisfies* (*is a model of*) an ontology $\mathcal{O}$ if $\mathcal{I}$ satisfies each axiom in $\mathcal{O}$. An ontology has a *model* (*is satisfiable*) if there is a model of $\mathcal{O}$.

4.4.2. Type-1, interval, and type-2 DLs

Let us fix the truth combination functions $\otimes, \oplus, \Rightarrow$ and $\ominus$. The semantics of fuzzy DLs is defined using a fuzzy interpretation relative to a fuzzy datatype theory $\mathbf{D} = \langle \Delta^{\mathbf{D}}, \cdot^{\mathbf{D}} \rangle$ and a fuzzy truth-type theory $\mathbf{T} = \langle \mathcal{N}, \cdot^{\mathbf{T}} \rangle$. Type-1 and interval DLs consider a type-1 datatype theory and a type-0 truth-type theory; type-2 DLs consider a type-2 datatype theory and a type-2 truth-type theory.

A *fuzzy interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ relative to **D** and **T** consists of a nonempty set $\Delta^{\mathcal{I}}$ (the *domain*) disjoint from $\Delta^{\mathbf{D}}$, and of a *fuzzy interpretation function* $\cdot^{\mathcal{I}}$ that coincides with $\cdot^{\mathbf{D}}$ on every data value $\mathtt{v}$ and fuzzy datatype **d**, coincides with $\cdot^{\mathbf{T}}$ on every truth value α and truth-type **t**, and it assigns:

- to each individual a an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$;
- to each data value $\mathtt{v}$ an element $\mathtt{v}^{\mathbf{D}} \in \Delta^{\mathbf{D}}$;
- to each fuzzy atomic concept A a function $A^{\mathcal{I}}\colon \Delta^{\mathcal{I}} \to \mathcal{N}$;
- to each fuzzy object property R a function $R^{\mathcal{I}}\colon \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to \mathcal{N}$;
- to each fuzzy data property T a function $T^{\mathcal{I}}\colon \Delta^{\mathcal{I}} \times \Delta^{\mathbf{D}} \to \mathcal{N}$;
- to each type-2 axiom τ a degree of truth $\tau^{\mathcal{I}} \in \mathcal{N}$.

The fuzzy interpretation function is extended to fuzzy *complex concepts* and *axioms* as shown in Table 2. It is worth to note that our fuzzy interpretations are type-1 in the sense that they assign to each type-2 axiom a single value in $\mathcal{N}$ instead of an element in $\mathscr{P}(\mathcal{N})$.

Remark 9. In Zadeh DLs it is common to assume Rescher implication in the semantics of GCIs because the use of Kleene–Dienes implication in the semantics of GCIs has some counter-intuitive effects [10]. Zadeh DLs assume however Kleene–Dienes implication in the semantics of universal restrictions, even if it is known to produce some counter-intuitive effects there as well [30].

Remark 10. Under Rescher implication $C_1^{\mathcal{I}}(x) \Rightarrow C_2^{\mathcal{I}}(x)$ has a value in $\{0, 1\}$, so an interval GCI does not really make sense.

Like for (classical) type-0 ontologies, for a type-1 or an interval ontology $\mathcal{O}$, with $\mathcal{I} \models \tau$ we denote the fact that $\mathcal{I}$ *satisfies* (*is a model of*) axiom τ, we say that $\mathcal{I}$ *satisfies* (*is a model of*) $\mathcal{O}$ if $\mathcal{I}$ satisfies each axiom in $\mathcal{O}$, and an ontology has a *model* (*is satisfiable*)) if there is a model of $\mathcal{O}$. In type-2 ontologies, axioms are not in general either true or false in a model, but a fuzzy interpretation $\mathcal{I}$ satisfies an axiom τ with some degree $\tau^{\mathcal{I}} \in \mathcal{N}$. This means that we must take into account the degrees of entailment of the axioms.

Example 4. Let us consider a simple fuzzy ontology with two axioms defined as $\mathcal{O} = \{\langle \mathtt{bob:Tall}, \ \mathtt{crisp}_{0.7,0.7} \rangle, \langle \mathtt{bob:Tall}, \ \mathtt{tri}_{0.6,0.8,1} \rangle\}$. According to the first axiom, $(\mathtt{bob:Tall})^{\mathcal{I}} = 0.7$ holds in every model of $\mathcal{O}$. Hence, the second axiom is satisfied with degree $\mathtt{tri}_{0.6,0.8,1}((\mathtt{bob:Tall})^{\mathcal{I}}) = \mathtt{tri}_{0.6,0.8,1}(0.7) = 0.5$.

Entailment degree. Let us consider a type-2 ontology $\mathcal{O} = \mathcal{O}_U \cup \mathcal{O}_Q$ and a fuzzy interpretation $\mathcal{I}$. The *degree of truth* of $\mathcal{O}_x$ ($x \in \{U, Q\}$) under $\mathcal{I}$, denoted $\mathcal{I}(\mathcal{O}_x)$, is defined as[4]

[4] Assuming $\mathcal{I}(\mathcal{O}') = 1$ if $\mathcal{O}' = \emptyset$ and $\min \emptyset = 1$.

Table 2
Semantics of complex concepts and axioms for fuzzy $\mathcal{ALC}(\mathbf{D})$.

Type-0 concept		
$\top^{\mathcal{I}}(x)$	$=$	1
$\bot^{\mathcal{I}}(x)$	$=$	0
$(C_1 \sqcap C_2)^{\mathcal{I}}(x)$	$=$	$\min(C_1^{\mathcal{I}}(x), C_2^{\mathcal{I}}(x))$
$(C_1 \sqcup C_2)^{\mathcal{I}}(x)$	$=$	$\max(C_1^{\mathcal{I}}(x), C_2^{\mathcal{I}}(x))$
$(\neg C)^{\mathcal{I}}(x)$	$=$	$1 - C^{\mathcal{I}}(x)$
$(\exists R.C)^{\mathcal{I}}(x)$	$=$	$\max_{y \in \Delta^{\mathcal{I}}}\{\min(R^{\mathcal{I}}(x,y), C^{\mathcal{I}}(y))\}$
$(\forall R.C)^{\mathcal{I}}(x)$	$=$	$\min_{y \in \Delta^{\mathcal{I}}}\{\max(1 - R^{\mathcal{I}}(x,y), C^{\mathcal{I}}(y))\}$
$(\exists T.\mathbf{d})^{\mathcal{I}}(x)$	$=$	$\max_{v \in \Delta^{\mathbf{D}}}\{\min(T^{\mathcal{I}}(x,v), \mathbf{d}^{\mathbf{D}}(v))\}$
$(\forall T.\mathbf{d})^{\mathcal{I}}(x)$	$=$	$\min_{v \in \Delta^{\mathbf{D}}}\{\max(1 - T^{\mathcal{I}}(x,v), \mathbf{d}^{\mathbf{D}}(v))\}$
Type-0 axiom		
$\mathcal{I} \models a : C$	iff	$C^{\mathcal{I}}(a^{\mathcal{I}}) = 1$
$\mathcal{I} \models (a_1, a_2) : R$	iff	$R^{\mathcal{I}}(a_1^{\mathcal{I}}, a_2^{\mathcal{I}}) = 1$
$\mathcal{I} \models (a, \mathrm{v}) : T$	iff	$T^{\mathcal{I}}(a^{\mathcal{I}}, \mathrm{v}^{\mathbf{D}}) = 1$
$\mathcal{I} \models C_1 \sqsubseteq C_2$	iff	$\forall x \in \Delta^{\mathcal{I}}. \max(1 - C_1^{\mathcal{I}}(x), C_2^{\mathcal{I}}(x)) = 1$
$\mathcal{I} \models C_1 \equiv C_2$	iff	$\forall x \in \Delta^{\mathcal{I}}. C_1^{\mathcal{I}}(x) = C_2^{\mathcal{I}}(x)$
Type-1, interval type-2, and type-2 concept		
$\top^{\mathcal{I}}(x)$	$=$	1
$\bot^{\mathcal{I}}(x)$	$=$	0
$(C_1 \sqcap C_2)^{\mathcal{I}}(x)$	$=$	$C_1^{\mathcal{I}}(x) \otimes C_2^{\mathcal{I}}(x)$
$(C_1 \sqcup C_2)^{\mathcal{I}}(x)$	$=$	$C_1^{\mathcal{I}}(x) \oplus C_2^{\mathcal{I}}(x)$
$(\neg C)^{\mathcal{I}}(x)$	$=$	$\ominus C^{\mathcal{I}}(x)$
$(\exists R.C)^{\mathcal{I}}(x)$	$=$	$\sup_{y \in \Delta^{\mathcal{I}}}\{R^{\mathcal{I}}(x,y) \otimes C^{\mathcal{I}}(y)\}$
$(\forall R.C)^{\mathcal{I}}(x)$	$=$	$\inf_{y \in \Delta^{\mathcal{I}}}\{R^{\mathcal{I}}(x,y) \Rightarrow C^{\mathcal{I}}(y)\}$
$(\exists T.\mathbf{d})^{\mathcal{I}}(x)$	$=$	$\sup_{v \in \Delta^{\mathbf{D}}}\{T^{\mathcal{I}}(x,v) \otimes \mathbf{d}^{\mathbf{D}}(v)\}$
$(\forall T.\mathbf{d})^{\mathcal{I}}(x)$	$=$	$\inf_{v \in \Delta^{\mathbf{D}}}\{T^{\mathcal{I}}(x,v) \Rightarrow \mathbf{d}^{\mathbf{D}}(v)\}$
$(C_1 \to C_2)^{\mathcal{I}}(x)$	$=$	$C_1{}^{\mathcal{I}}(x) \Rightarrow C_2{}^{\mathcal{I}}(x)$
$(C[\geq \alpha])^{\mathcal{I}}(x)$	$=$	$\begin{cases} C^{\mathcal{I}}(x), & \text{if } C^{\mathcal{I}}(x) \geq \alpha \\ 0, & \text{otherwise} \end{cases}$
$(C[\leq \beta])^{\mathcal{I}}(x)$	$=$	$\begin{cases} C^{\mathcal{I}}(x), & \text{if } C^{\mathcal{I}}(x) \leq \beta \\ 0, & \text{otherwise} \end{cases}$
Type-1 axiom		
$\mathcal{I} \models \langle a : C \geq \alpha \rangle$	iff	$C^{\mathcal{I}}(a^{\mathcal{I}}) \geq \alpha$
$\mathcal{I} \models \langle a : C \leq \beta \rangle$	iff	$C^{\mathcal{I}}(a^{\mathcal{I}}) \leq \beta$
$\mathcal{I} \models \langle (a_1, a_2) : R \geq \alpha \rangle$	iff	$R^{\mathcal{I}}(a_1^{\mathcal{I}}, a_2^{\mathcal{I}}) \geq \alpha$
$\mathcal{I} \models \langle (a_1, a_2) : R \leq \beta \rangle$	iff	$R^{\mathcal{I}}(a_1^{\mathcal{I}}, a_2^{\mathcal{I}}) \leq \beta$
$\mathcal{I} \models \langle (a, \mathrm{v}) : T \geq \alpha \rangle$	iff	$T^{\mathcal{I}}(a^{\mathcal{I}}, \mathrm{v}^{\mathbf{D}}) \geq \alpha$
$\mathcal{I} \models \langle (a, \mathrm{v}) : T \leq \beta \rangle$	iff	$T^{\mathcal{I}}(a^{\mathcal{I}}, \mathrm{v}^{\mathbf{D}}) \leq \beta$
$\mathcal{I} \models \langle C_1 \sqsubseteq C_2 \geq \alpha \rangle$	iff	$\inf_{x \in \Delta^{\mathcal{I}}}\{C_1^{\mathcal{I}}(x) \Rightarrow C_2^{\mathcal{I}}(x)\} \geq \alpha$
$\mathcal{I} \models C_1 \equiv C_2$	iff	$\forall x \in \Delta^{\mathcal{I}}. C_1^{\mathcal{I}}(x) = C_2^{\mathcal{I}}(x)$
Interval axiom		
$\mathcal{I} \models \langle a : C, [\gamma_1, \gamma_2] \rangle$	iff	$C^{\mathcal{I}}(a^{\mathcal{I}}) \in [\gamma_1, \gamma_2)$
$\mathcal{I} \models \langle (a_1, a_2) : R, [\gamma_1, \gamma_2] \rangle$	iff	$R^{\mathcal{I}}(a_1^{\mathcal{I}}, a_2^{\mathcal{I}}) \in [\gamma_1, \gamma_2]$
$\mathcal{I} \models \langle (a, \mathrm{v}) : T, [\gamma_1, \gamma_2] \rangle$	iff	$T^{\mathcal{I}}(a^{\mathcal{I}}, \mathrm{v}^{\mathbf{D}}) \in [\gamma_1, \gamma_2]$
$\mathcal{I} \models \langle C_1 \sqsubseteq C_2, [\gamma_1, \gamma_2] \rangle$	iff	$\inf_{x \in \Delta^{\mathcal{I}}}\{C_1^{\mathcal{I}}(x) \Rightarrow C_2^{\mathcal{I}}(x)\} \in [\gamma_1, \gamma_2]$
Type-2 axiom		
$\langle a : C, \mathbf{t} \rangle^{\mathcal{I}}$	$=$	$\mathbf{t}^{\mathbf{T}}(C^{\mathcal{I}}(a^{\mathcal{I}}))$
$\langle (a_1, a_2) : R, \mathbf{t} \rangle^{\mathcal{I}}$	$=$	$\mathbf{t}^{\mathbf{T}}(R^{\mathcal{I}}(a_1^{\mathcal{I}}, a_2^{\mathcal{I}}))$
$\langle (a, \mathrm{v}) : T, \mathbf{t} \rangle^{\mathcal{I}}$	$=$	$\mathbf{t}^{\mathbf{T}}(T^{\mathcal{I}}(a^{\mathcal{I}}, \mathrm{v}^{\mathbf{D}}))$
$\langle C_1 \sqsubseteq C_2, \mathbf{t} \rangle^{\mathcal{I}}$	$=$	$\mathbf{t}^{\mathbf{T}}(\inf_{x \in \Delta^{\mathcal{I}}}\{C_1^{\mathcal{I}}(x) \Rightarrow C_2^{\mathcal{I}}(x)\})$

$$\mathcal{I}(\mathcal{O}_X) = \min_{\tau \in \mathcal{O}_X} \tau^{\mathcal{I}} . \quad (5)$$

Note that $\mathcal{I}$ is a model of $\mathcal{O}_U$ if and only if $\mathcal{I}(\mathcal{O}_U) = 1$; otherwise $\mathcal{I}(\mathcal{O}_U) = 0$. One may wonder why in Eq. (5) we have considered the min operator in place of the $\otimes$ operator. We prefer to use min as it satisfies $\gamma = \min(\gamma, \gamma)$. This prevents that multiple occurrences of the same axiom τ in an ontology $\mathcal{O}$ may decrease the degree of truth of $\mathcal{O}$ under $\mathcal{I}$, which is a property that is unlikely to be wanted in knowledge representation.

The degree of truth of a fuzzy ontology $\mathcal{O}$ under $\mathcal{I}$, denoted $\mathcal{I}(\mathcal{O})$, is then defined as

$$\mathcal{I}(\mathcal{O}) = \min\{\mathcal{I}(\mathcal{O}_U), \mathcal{I}(\mathcal{O}_Q)\} = \min_{\tau \in \mathcal{O}} \tau^{\mathcal{I}} . \quad (6)$$

We say that $\mathcal{I}$ *satisfies* (*is a model of*) $\mathcal{O}$ if $\mathcal{I}(\mathcal{O}) > 0$. If $\mathcal{O}$ has a model then we say that $\mathcal{O}$ is *consistent*. Note that $\mathcal{O}$ is consistent if and only if there is a fuzzy interpretation $\mathcal{I}$ such that $\tau^{\mathcal{I}} > 0$ for each $\tau \in \mathcal{O}$. In particular, $\tau^{\mathcal{I}} = 1$ for each $\tau \in \mathcal{O}_U$, so $\mathcal{I}(\mathcal{O}_U) = 1$, and $\mathcal{I}(\mathcal{O}_Q) > 0$.

Proposition 2. *Let $\mathcal{O}$ be a type-2 ontology, $\mathcal{I}$ a model of $\mathcal{O}$, and τ either a type-0, a type-1, or an interval axiom. Then, $\tau^{\mathcal{I}} \in \{0, 1\}$.*

Proof. We will only prove the case of type-1 concept assertions because the other cases are similar. According to Remarks 7 and 8, a type-1 concept assertion $\langle a{:}C \geq \alpha \rangle$ can be represented as a type-2 concept assertion $\tau = \langle a{:}C, \mathtt{crisp}_{\alpha,1} \rangle$. Then, $\tau^{\mathcal{I}} = \mathtt{crisp}_{\alpha,1}(C^{\mathcal{I}}(a^{\mathcal{I}})) \in \{0, 1\}$. □

Please note that in Mathematical Fuzzy First Order Logic (FOL),[5] we can see a fuzzy ontology $\mathcal{O}$ roughly as the formula

$$\Gamma_{\mathcal{O}} = (\bigwedge_{\tau \in \mathcal{O}_U} \tau) \wedge (\bigwedge_{\tau \in \mathcal{O}_Q} \tau) , \tag{7}$$

where $\wedge$ is interpreted as Gödel t-norm and fuzzy DL axioms are mapped into fuzzy FOL similarly as for the crisp case [4]. In Appendix A we provide the details of such a mapping. We will use $\Gamma_{\mathcal{O}_U}$ to denote the left conjunct and $\Gamma_{\mathcal{O}_Q}$ to denote the right conjunct in Eq. (7), respectively, and, thus, $\Gamma_{\mathcal{O}} = \Gamma_{\mathcal{O}_U} \wedge \Gamma_{\mathcal{O}_Q}$. We will also omit the reference to $\mathcal{O}$ if clear from context.

The *entailment degree* between a fuzzy ontology $\mathcal{O}$ and an axiom τ under fuzzy interpretation $\mathcal{I}$, denoted $ed(\mathcal{O}, \tau, \mathcal{I})$, is defined as follows, where $\Rightarrow$ is a fuzzy implication:

$$ed(\mathcal{O}, \tau, \mathcal{I}) = \mathcal{I}(\mathcal{O}) \Rightarrow \tau^{\mathcal{I}} . \tag{8}$$

Usually, $\Rightarrow$ is the same fuzzy implication used in the semantics of the GCIs, but this is not really necessary. Our reasoning algorithm in Section 7 will assume that $\Rightarrow$ satisfies (OP) and (NFNP). Of course, if $\mathcal{I} \not\models \mathcal{O}_U$ then $\mathcal{I}(\mathcal{O}_U) = 0$, so $ed(\mathcal{O}, \tau, \mathcal{I}) = 0 \Rightarrow \tau^{\mathcal{I}} = 1$, while if $\mathcal{I} \models \mathcal{O}_U$ then $\mathcal{I}(\mathcal{O}_U) = 1$, so $ed(\mathcal{O}, \tau, \mathcal{I}) = ed(\mathcal{O}_Q, \tau, \mathcal{I})$. Note that for type-1 and interval ontologies and axioms, $ed(\mathcal{O}, \tau, \mathcal{I}) \in \{0, 1\}$.

Remark 11. In Zadeh type-2 DLs, the use of Rescher implication implies that $ed(\mathcal{O}, \tau, \mathcal{I}) \in \{0, 1\}$.

The *entailment degree* between a fuzzy ontology $\mathcal{O}$ and axiom τ, denoted $ed(\mathcal{O}, \tau)$, is defined as[6]

$$ed(\mathcal{O}, \tau) = \inf_{\mathcal{I} \models \mathcal{O}_U, \mathcal{I}(\mathcal{O}_Q) > 0} ed(\mathcal{O}_Q, \tau, \mathcal{I}) . \tag{9}$$

Therefore, for type-1 and interval ontologies and axioms τ, $ed(\mathcal{O}, \tau) \in \{0, 1\}$ and $ed(\mathcal{O}, \tau) = 1$ iff $\mathcal{O} \models \tau$, where the latter *entailment* notion $\models$ is defined in the usual way (every model of $\mathcal{O}$ is also a model of τ).

If we see O_Q as the fuzzy FOL formula $\Gamma_{\mathcal{O}_Q}$, it is easily verified that

$$ed(\mathcal{O}, \tau) = \inf_{\mathcal{I} \models \mathcal{O}_U, \mathcal{I}(\mathcal{O}_Q) > 0} (\Gamma_{\mathcal{O}_Q} \to \tau)^{\mathcal{I}} . \tag{10}$$

where $\mathcal{I}$ has been adapted to fuzzy FOL models in the obvious way and $\to$ is interpreted as $\Rightarrow$.

The *Best Entailment Degree* (BED) between a fuzzy ontology $\mathcal{O}$ and a (classical) type-0 axiom $\mathfrak{a} \in \{a{:}C, (a_1, a_2){:}R, (a, \mathrm{v}){:}T, C_1 \sqsubseteq C_2\}$, denoted $bed(\mathcal{O}, \mathfrak{a})$, is defined as

$$bed(\mathcal{O}, \mathfrak{a}) = \sup\{\gamma \in [0, 1] \mid ed(\mathcal{O}, \langle \mathfrak{a} \geq \gamma \rangle) = 1\} . \tag{11}$$

Please note again that, for type-1 fuzzy ontologies, the above notion corresponds to the usual *best entailment degree* notion [56] defined as[7]

$$bed(\mathcal{O}, \mathfrak{a}) = \sup\{\gamma \in [0, 1] \mid \mathcal{O} \models \langle \mathfrak{a} \geq \gamma \rangle\} . \tag{12}$$

To conclude this part, we can define the *Best Satisfiability Degree* (BSD) of a concept C w.r.t. $\mathcal{O}$, denoted as $bsd(\mathcal{O}, C)$, as[8]

$$bsd(\mathcal{O}, C) = \sup_{\mathcal{I} \models \mathcal{O}_U, \mathcal{I}(\mathcal{O}_Q) > 0} \sup_{x \in \Delta^{\mathcal{I}}} \{\mathcal{I}(\mathcal{O}_Q) \Rightarrow C^{\mathcal{I}}(x)\} . \tag{13}$$

Witnessed models. As we have seen, the semantics of existential restrictions, universal restrictions, and GCIs use the operations of supremum or infimum. It could be possible to build an infinite interpretation (with infinite domain elements)

---

[5] More precisely, in an extension described in Appendix A.

[6] Assuming $\inf \emptyset = 1$.

[7] In this case, we always have that $ed(\mathcal{O}, \langle \mathfrak{a}, \gamma \rangle) \in \{0, 1\}$.

[8] Here we assume $\sup \emptyset = 1$.

such that the limit of the infinite sequence converges to some value (if the sequence is increasing, this will be supremum; otherwise it will be the infimum), but the condition is not attained at any of the elements of the domain [32]. These models are usually not interesting in knowledge representation.

As usual in fuzzy DLs, we will assume that all models are *witnessed*,[9] where whenever an sup (resp. inf) operator is involved, the supremum (resp. infimum) is attained at some point (in $\Delta^{\mathcal{I}}$ or $\Delta^{\mathbf{D}}$, depending on the domain the limit operation ranges over) called the witness of the condition. In witnessed models, we further have that the following conditions need to be satisfied:

$$(\exists R.C)^{\mathcal{I}}(x) = R^{\mathcal{I}}(x,z) \otimes C^{\mathcal{I}}(z)$$
$$(\forall R.C)^{\mathcal{I}}(x) = R^{\mathcal{I}}(x,z) \Rightarrow C^{\mathcal{I}}(z)$$
$$(\exists T.\mathbf{d})^{\mathcal{I}}(x) = T^{\mathcal{I}}(x,v) \otimes \mathbf{d}^{\mathbf{D}}(v)$$
$$(\forall T.\mathbf{d})^{\mathcal{I}}(x) = T^{\mathcal{I}}(x,v) \Rightarrow \mathbf{d}^{\mathbf{D}}(v)$$
$$\langle C_1 \sqsubseteq C_2, \mathbf{t}\rangle^{\mathcal{I}} = \mathbf{t}^{\mathbf{T}}(C_1^{\mathcal{I}}(z) \Rightarrow C_2^{\mathcal{I}}(z))$$

for some $z \in \Delta^{\mathcal{I}}$ and $v \in \Delta^{\mathbf{D}}$. Note that in finitely-valued DLs, all models are witnessed.

5. Expressivity of previous type-2 description logics

In this section we will show that the previous work on type-2 extensions to fuzzy DLs described in Section 3, which is actually restricted to interval axioms or concepts, is not expressive enough because such logics can be represented using type-1 languages. Our first results will be related with the logic in [35], with type-2 concept and role assertions.

Proposition 3. *Given a fuzzy language $\mathcal{X}$, interval concept assertions $\langle a : C, [\gamma_1, \gamma_2]\rangle$ can be reduced (by preserving models) to type-1 concept assertions if one of the following conditions hold:*

- *$\mathcal{X}$ supports axioms of the form $\langle a{:}C \leq \beta\rangle$, or*
- *$\mathcal{X}$ supports an involutive negation.*

Proof. Consider an axiom $\langle a{:}C, [\gamma_1, \gamma_2]\rangle$. Then $\mathcal{I} \models \langle a{:}C, [\gamma_1, \gamma_2]\rangle$ if $C^{\mathcal{I}}(a^{\mathcal{I}}) \in [\gamma_1, \gamma_2]$. On the one hand, if $\mathcal{X}$ contains axioms of the form $\langle a{:}C \leq \beta\rangle$, it is clear that $\langle a{:}C, [\gamma_1, \gamma_2]\rangle$ can be reduced to the pair of axioms $\langle a{:}C \geq \gamma_1\rangle$ and $\langle a{:}C \leq \gamma_2\rangle$. On the other hand, if $\mathcal{X}$ contains an involutive negation, $\langle a{:}C, [\gamma_1, \gamma_2]\rangle$ can be reduced to the pair of axioms $\langle a{:}C \geq \gamma_1\rangle$ and $\langle a{:}\neg C \geq \ominus\gamma_2\rangle$. In fact, $\mathcal{I} \models \langle a{:}C \geq \gamma_1\rangle$ if $C^{\mathcal{I}}(a^{\mathcal{I}}) \geq \gamma_1$, while $\langle a{:}\neg C \geq \ominus\gamma_2\rangle$ if $(\neg C)^{\mathcal{I}}(a^{\mathcal{I}}) = \ominus C^{\mathcal{I}}(a^{\mathcal{I}}) \geq \ominus\gamma_2$. As $\ominus$ is non-increasing, it follows that $\ominus(\ominus C^{\mathcal{I}}(a^{\mathcal{I}})) \leq \ominus(\ominus\gamma_2)$ and as $\ominus$ is involutive $C^{\mathcal{I}}(a^{\mathcal{I}}) \leq \gamma_2$ and, thus, $C^{\mathcal{I}}(a^{\mathcal{I}}) \in [\gamma_1, \gamma_2]$. □

Proposition 4. *Given a fuzzy language $\mathcal{X}$, then an interval role assertion $\langle (a_1, a_2){:}R, [\gamma_1, \gamma_2]\rangle$ can be reduced to (by preserving models) type-1 role assertions if one of the following conditions hold:*

- *$\mathcal{X}$ supports axioms of the form $\langle (a_1, a_2){:}R \leq \beta\rangle$, or*
- *$\mathcal{X}$ supports an involutive role negation.*

Proof. The proof is similar to that of Proposition 3. □

The preconditions in Propositions 3 and 4 hold in Zadeh or Łukasiewicz fuzzy DLs. In Gödel and Product fuzzy DLs, the preconditions only hold if the logic is extended with involutive negation, as proposed in [12], or if the language supports concept and role assertions of the form $\langle \mathfrak{a} \leq \alpha\rangle$. This is very likely to be the case, because it has been shown that Gödel and Product fuzzy DLs without fuzzy assertions of the form $\langle a : C \leq \alpha\rangle$ are not really fuzzy, in the sense that a fuzzy ontology has a fuzzy model only if the ontology that is obtained by replacing any axiom $\langle \mathfrak{a} \geq \alpha\rangle$ with $\mathfrak{a}$ has a model [21].

Now we will discuss some results related with the logic in [6], with type-2 GCIs.

Proposition 5. *Given a fuzzy DL language $\mathcal{X}$, interval GCIs $\langle C_1 \sqsubseteq C_2, [\gamma_1, \gamma_2]\rangle$ can be reduced to the pair of type-1 axioms $\langle C_1 \sqsubseteq C_2 \geq \gamma_1\rangle$, and $\langle a{:}C_1 \rightarrow C_2 \leq \gamma_2\rangle$ (where a is a new individual, not appearing in the fuzzy ontology) if the following conditions hold:*

- *$\mathcal{X}$ supports axioms of the form $\langle a{:}C \leq \beta\rangle$ or involutive negation, and*
- *$\mathcal{X}$ supports concept implication.*

[9] In fact, the usefulness of non-witnessed models is questionable from a knowledge representation point of view [56] and, thus, they are usually left out.

Proof. Assume $\mathcal{I} \models \langle C_1 \sqsubseteq C_2, [\gamma_1, \gamma_2] \rangle$, that is, *(i)* $(C_1^{\mathcal{I}}(x) \Rightarrow C_2^{\mathcal{I}}(x)) \geq \gamma_1$ for every $x \in \Delta^{\mathcal{I}}$; and *(ii)* there is some $y \in \Delta^{\mathcal{I}}$ such that $(C_1^{\mathcal{I}}(y) \Rightarrow C_2^{\mathcal{I}}(y)) \leq \gamma_2$.

The first part *(i)* is captured by the type-1 GCI $\langle C_1 \sqsubseteq C_2 \geq \gamma_1 \rangle$, while the second part *(ii)* is captured either by the type-1 concept assertions $\langle a{:}C_1 \rightarrow C_2 \leq \gamma_2 \rangle$ or by $\langle a{:}\neg(C_1 \rightarrow C_2) \geq \ominus\gamma_2 \rangle$, for a new individual a. □

The preconditions in Proposition 5 hold in Łukasiewicz DLs, which is the semantics considered in [6]. They also hold in Gödel and Product fuzzy DLs extended with implication concepts and involutive negation, as well as in Gödel and Product fuzzy DLs extended with implication concepts and axioms of the form $\langle a : C \leq \beta \rangle$. As discussed in Remark 10, interval GCIs do not make sense in Zadeh DLs with Rescher implication in the semantics of GCIs. Assuming Kleene–Dienes implication in the semantics of GCIs, Proposition 5 holds in Zadeh DLs as well.

Finally, we will discuss some results related with the logic in [41], with type-2 concept expressions.

Proposition 6. *Given a fuzzy DL language $\mathcal{X}$ with threshold concepts and Gödel conjunction, an interval concept expression $C_{[\alpha_1,\alpha_2]}$ can be represented as $C_{[\geq\alpha_1]} \sqcap_G C_{[\leq\alpha_2]}$, where $\sqcap_G$ denotes Gödel conjunction.*

Proof. Given an interpretation $\mathcal{I}$, $(C_{[\alpha_1,\alpha_2]})^{\mathcal{I}}$ is non-zero for those individuals $x \in \Delta^{\mathcal{I}}$ such that $C^{\mathcal{I}}(x) \in [\alpha_1, \alpha_2]$, and such individuals belong to the concept with degree $C^{\mathcal{I}}(x)$ [41]. By definition, for all $x \in \Delta^{\mathcal{I}}$

$$(C_{[\geq\alpha_1]} \sqcap_G C_{[\leq\alpha_2]})^{\mathcal{I}}(x) = \min\{C^{\mathcal{I}}_{[\geq\alpha_1]}(x), C^{\mathcal{I}}_{[\leq\alpha_2]}(x)\} \tag{14}$$

Now, if $C^{\mathcal{I}}(x) \notin [\alpha_1, \alpha_2]$ then either $C^{\mathcal{I}}_{[\geq\alpha_1]}(x)$ or $C^{\mathcal{I}}_{[\leq\alpha_2]}(x)$ is 0 and, thus, Eq. (14) evaluates to 0 as well. Otherwise, $C^{\mathcal{I}}(x) \in [\alpha_1, \alpha_2]$ and Eq. (14) evaluates to $C^{\mathcal{I}}(x)$, which concludes. □

The preconditions in Proposition 6 trivially hold in Zadeh, and Gödel fuzzy DLs with threshold concepts, because they directly have such t-norm. Furthermore, they hold in Łukasiewicz DLs with threshold concepts because the logic has an S-implication so that $C_1 \rightarrow C_2$ can be represented as $\neg C_1 \sqcup C_2$, and the minimum is representable from the implication as shown in Remark 5 (let us recall that [41] also assumes an S-implication). Finally, the preconditions hold in Product fuzzy DLs extended with threshold concepts and implication, so that Remark 5 can be applied.

However, it is unknown whether interval concept expressions over roles, such as in $\exists R_{[\alpha_1,\alpha_2]}.C$, can be reduced in a similar way as in Proposition 6 by using threshold concepts only.

Before concluding this section, it is worth to note that fuzzy DLs with involutive negation or axioms of the form $\langle \tau \leq \beta \rangle$ can be undecidable in presence of GCIs [3,21,25]. To solve it, our reasoning algorithm in Section 7 assumes some restrictions in the TBox.

6. Type-2 fuzzy OWL 2 ontology languages

The previous syntax based on classical DLs is not appropriate to represent fuzzy ontologies in practice. In such scenarios, fuzzy ontology languages are needed. This section discusses how to extend a type-1 ontology language to the type-2 case. In this section, type-1 membership functions will be restricted to those in Fig. 1, defined over [0, 1].

Type-1 Fuzzy OWL 2. Fuzzy OWL 2 [14] is one of the few fuzzy ontology languages at hand so far, which consists on extending OWL 2 ontologies with OWL 2 annotations encoding fuzzy information using a XML-like syntax. The key idea of this representation is to start with an OWL 2 ontology created as usual, with a classical ontology editor. Then, the user can annotate the elements to represent the features of the fuzzy ontology that OWL 2 cannot directly encode.

It is possible to annotate fuzzy axioms by adding a lower bound numerical degree of truth and to represent fuzzy datatypes. In more expressive fuzzy DLs, it is possible to represent fuzzy modifiers and other fuzzy concepts and roles that do not have an equivalent in the classical case, namely aggregated concepts, fuzzy nominals, and fuzzy modified roles. In order to separate the annotations including fuzzy information from other annotations, a special annotation property called `fuzzyLabel` is used, and every annotation is identified by the tag `fuzzyOwl2`. There is a Protégé plug-in making the syntax of the annotations transparent to the users.[10]

Example 5. A fuzzy concept assertion $\langle \mathit{bob{:}Tall} \geq 0.4 \rangle$ can be represented by annotating the classical axiom *bob:Tall* with the degree ≥ 0.4 as follows:

```
<ClassAssertion>
  <Class IRI='#Tall'/>
  <NamedIndividual IRI='#bob'/>
  <Annotation>
    <AnnotationProperty IRI='#fuzzyLabel'/>
    <Literal datatypeIRI='&rdf;PlainLiteral'>
```

[10] http://www.umbertostraccia.it/cs/software/FuzzyOWL.

```
      <fuzzyOwl2 fuzzyType="axiom">
        <Degree value="0.4" />
      </fuzzyOwl2>
    </Literal>
  </Annotation>
</ClassAssertion>
```

A recent extension extends the syntax of Fuzzy OWL 2 to specify an upper degree for some axioms [8], namely concept assertions and, if the equivalent crisp language supports negative property assertions, property assertions as well. The syntax is the following one:

```
<fuzzyOwl2 fuzzyType="axiom">
  <Degree maxValue="<DOUBLE>" />
</fuzzyOwl2>
```

The syntax of Fuzzy OWL 2 fuzzy datatypes is:

```
<fuzzyOwl2 fuzzyType="datatype">
  <DATATYPE>
</fuzzyOwl2>

<DATATYPE> :=
  <Datatype type="leftshoulder" a="<DOUBLE>" b="<DOUBLE>" /> |
  <Datatype type="rightshoulder" a="<DOUBLE>" b="<DOUBLE>" /> |
  <Datatype type="triangular" a="<DOUBLE>" b="<DOUBLE>" c="<DOUBLE>" /> |
  <Datatype type="trapezoidal" a="<DOUBLE>" b="<DOUBLE>" c="<DOUBLE>" d="<DOUBLE>" />
```

where $\mathtt{a} \le \mathtt{b} \le \mathtt{c} \le \mathtt{d}$, and $\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d} \in [0, 1]$. $\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}$ correspond to the parameters $\mathtt{q1}, \mathtt{q2}, \mathtt{q3}, \mathtt{q4}$ in Fig. 1, respectively.

Example 6. Let us represent the fuzzy datatype $\mathit{YoungAge} = \mathtt{left}_{20,40}$ denoting the age of a young person. This fuzzy datatype is represented using a datatype definition of base type `xsd:real` annotated as follows:

```
<AnnotationAssertion>
  <AnnotationProperty IRI='#fuzzyLabel'/>
  <IRI>#YoungAge</IRI>
  <Literal datatypeIRI='&rdf;PlainLiteral'>
    <fuzzyOwl2 fuzzyType="datatype">
      <Datatype type="leftshoulder" a="20" b="40" />
    </fuzzyOwl2>
  </Literal>
</AnnotationAssertion>
```

Type-2 Fuzzy OWL 2. Fuzzy OWL 2 has been designed to represent type-1 ontologies so far. In the rest of this section, we will discuss how to extend it to represent interval and type-2 ontologies as well.

Interval ontologies can immediately be represented in Fuzzy OWL 2 by combining `value` and `maxValue` attributes. For the sake of clarity, we propose to extend the language with the attribute `minValue` as an alias for `value`. Example 7 shows how to represent axioms in interval type-2 ontologies. The syntax is the following one, where at least one of the attributes must appear:

```
<fuzzyOwl2 fuzzyType="axiom">
  <Degree (minValue | value)="<DOUBLE>" maxValue="<DOUBLE>" />
</fuzzyOwl2>
```

Example 7. Let us represent the fuzzy axiom $\langle \mathit{bob}:\mathit{Tall}, [0.4, 0.9]\rangle$ restricting the tallness of Bob. This can be achieved by annotating the classical axiom $\mathit{bob}:\mathit{Tall}$ as follows:

```
<ClassAssertion>
  <Class IRI='#Tall'/>
  <NamedIndividual IRI='#bob'/>
  <Annotation>
    <AnnotationProperty IRI='#fuzzyLabel'/>
    <Literal datatypeIRI='&rdf;PlainLiteral'>
      <fuzzyOwl2 fuzzyType="axiom">
        <Degree minValue="0.4" maxValue="0.9" />
      </fuzzyOwl2>
    </Literal>
  </Annotation>
</ClassAssertion>
```

Now let us discuss how to represent type-2 ontologies. To start with, we need to define a syntax for type-2 datatypes, which consist of a pair of type-1 datatypes related via the `minValue` and `maxValue` attributes. To be effective (and guarantee an implementation), we restrict these datatype functions to those in Fig. 1, where the syntax of type-1 fuzzy datatypes [14] is extended with the linear function (that was not yet supported as datatype). The proposed syntax for type-2 datatypes is:

```
<fuzzyOwl2 fuzzyType="datatype">
  <TYPE1DATATYPE> | <TYPE2DATATYPE>
</fuzzyOwl2>

<TYPE1DATATYPE> :=
  <Datatype type="leftshoulder" a="<DOUBLE>" b="<DOUBLE>" /> |
  <Datatype type="rightshoulder" a="<DOUBLE>" b="<DOUBLE>" /> |
  <Datatype type="triangular" a="<DOUBLE>" b="<DOUBLE>" c="<DOUBLE>" /> |
  <Datatype type="trapezoidal" a="<DOUBLE>" b="<DOUBLE>" c="<DOUBLE>" d="<DOUBLE>" /> |
  <Datatype type="linear" a="<DOUBLE>" b="<DOUBLE>" />

<TYPE2DATATYPE> :=
  <Datatype (minValue | value)="<TYPE1DATNAME>" maxValue="<TYPE1DATNAME>" />
```

where in the linear datatypes $\mathtt{a}, \mathtt{b} \in [0, 1]$ correspond to the parameters $\mathtt{q1}, \mathtt{q2}$ in Fig. 1 (e), respectively, in type-2 datatypes both the attributes `minValue` and `maxValue` are mandatory, and `TYPE1DATNAME` refers to the name of a previously defined type-1 datatype.

Example 8. Consider again Example 1. The type-2 datatype can be represented by annotating three OWL 2 datatypes *LowerMediumRank*, *UpperMediumRank*, and *MediumRank* as follows:

```
<fuzzyOwl2 fuzzyType="LowerMediumRank">
  <Datatype type="trapezoidal" a="2.2" b="3" c="4.5" d="5" />
</fuzzyOwl2>

<fuzzyOwl2 fuzzyType="UpperMediumRank">
  <Datatype type="trapezoidal" a="1.6" b="3" c="4.5" d="5.5" />
</fuzzyOwl2>

<fuzzyOwl2 fuzzyType="MediumRank">
  <Datatype minValue="LowerMediumRank" maxValue="UpperMediumRank" />
</fuzzyOwl2>
```

Then, we can define the concept $\exists \mathtt{hasRanking}.\mathtt{MediumRank}$ to represent a medium value for the evaluation.

It only remains to show how to represent type-2 axioms. This can be achieved similarly as for type-2 datatypes. Firstly, we annotate OWL 2 datatypes to represent type-1 and type-2 truth-types, using the following syntax:

```
<fuzzyOwl2 fuzzyType="degree">
  <TYPE1TRUTHTYPE> | <TYPE2TRUTHTYPE>
</fuzzyOwl2>

<TYPE1TRUTHTYPE> :=
  <Degree type="leftshoulder" a="<DOUBLE>" b="<DOUBLE>" /> |
  <Degree type="rightshoulder" a="<DOUBLE>" b="<DOUBLE>" /> |
  <Degree type="triangular" a="<DOUBLE>" b="<DOUBLE>" c="<DOUBLE>" /> |
  <Degree type="trapezoidal" a="<DOUBLE>" b="<DOUBLE>" c="<DOUBLE>" d="<DOUBLE>" /> |
  <Degree type="linear" a="<DOUBLE>" b="<DOUBLE>" />

<TYPE2TRUTHTYPE> :=
 <Degree (minValue | value)="<TYPE1TTNAME>" maxValue="<TYPE1TTNAME>" />
```

where parameters $a, b, c, d \in [0, 1]$, `TYPE1TTNAME` refers to the name of previously defined type-1 truth-type, and the two attributes of a type-2 truth type are mandatory.

Finally, axioms can be annotated as follows:

```
<fuzzyOwl2 fuzzyType="axiom">
  ( <Degree (minValue | value)="<DOUBLE> | <TYPE1TTNAME>"
           maxValue="<DOUBLE> | <TYPE1TTNAME>" /> )  |
  <Degree (minValue | value)="<TYPE2TTNAME>"
</fuzzyOwl2>
```

where `TYPE1TTNAME` and `TYPE2TTNAME` refer to the names of previously defined type-1 and type-2 truth-types, respectively, and at least one of the attributes of type-0 and type-1 truth types must appear. This way, for type-0 and type-1 axioms it is possible to specify two truth types (a lower one and an upper one) or only one, while for type-2 axioms it is only possible to specify one.

7. Reasoning with type-2 description logics

In this section we discuss how to reason with type-2 DLs. Firstly, we recap some interesting reasoning tasks and discuss their mutual reductions. Then, we discuss how we extend some type-1 reasoning algorithms in order to adapt them to the new logic.

7.1. Main type-2 reasoning tasks

This section defines the main reasoning tasks in type-2 DLs we are interested in. As we will see, there are some differences with other fuzzy DLs due to the fact that qualified type-2 axioms hold to some degree of truth.

Consistency problem. Determine whether a type-2 ontology $\mathcal{O}$ is consistent.

Entailment problem. Given a type-2 ontology $\mathcal{O}$ and a type-1 axiom $\tau = \langle \mathfrak{a} \geq \alpha \rangle$, determine whether $ed(\mathcal{O}, \tau) = 1$ or, equivalently, $\mathcal{O} \models \tau$.

Best entailment degree problem. Given a type-2 ontology $\mathcal{O}$ and a (classical) type-0 axiom $\mathfrak{a} \in \{a{:}C, (a_1, a_2){:}R, (a, \mathrm{v}){:}T, C_1 \sqsubseteq C_2\}$, determine $bed(\mathcal{O}, \mathfrak{a})$.

Best satisfiability degree problem. Given a type-2 ontology $\mathcal{O}$ and a concept C, determine $bsd(\mathcal{O}, C)$.

In our opinion, the entailment and the BED do not make sense for other types of axioms.

7.2. Overview of the reasoning algorithm

Now we will provide a reasoning algorithm for some type-2 DLs. In particular, we will discuss how we may extend existing reasoning algorithms for type-1 DLs based on a combination of a tableau algorithm with an optimization problem [12,15,53,55] to deal with type-2 DLs. The method extends the preliminary results in [13], that are restricted to Łukasiewicz $\mathcal{ALC}$ and do not support type-2 datatypes or type-2 truth-types.

We recap here the idea of these optimization-based algorithms (for more details, see e.g., [9,56]). Each reasoning task is reduced to a *variable minimization/maximization problem with respect to a fuzzy ontology*. Then, the algorithm applies several tableau rules that decompose complex concept expressions into simpler ones, as usual in tableau algorithms, but also generates a system of inequation constraints. Such inequations need to hold in order to respect the semantics of the fuzzy DL constructors. These rules preserve both the consistency of the fuzzy ontology and the value of the variable that is being optimized. After all rules have been applied, an optimization problem must be solved before obtaining the final result. The tableau rules are deterministic and only one optimization problem is obtained. This problem has a solution iff the fuzzy ontology is consistent: if the problem has a solution, one can easily obtain from it the solution to the original reasoning task; otherwise, the fuzzy ontology is inconsistent and hence the result is 1 (for minimization, 0 for maximization) because an inconsistent ontology entails everything. Such an algorithm has been implemented into e.g., the fuzzyDL reasoner [18], arguably one of the most evolved fuzzy DL reasoners so far.

In order to make our method applicable some specific restrictions must hold.

- R1 The language supports Łukasiewicz negation or axioms of the form $\tau \leq \beta$.
- R2 The semantics is restricted to the truth space $[0, 1]$ or to a finite subset of it. As finite set of degrees of truth, one can take, for example, the well-known space $\mathcal{N}_n = \{0, \frac{1}{n}, \dots, \frac{n-1}{n}, 1\}$ for some natural $n \geq 2$, or the machine-representable set of degrees of truth. In some cases, one could preprocess the ontology to identify the finite set of truth values we may restrict on, in the style of [54] for type-1 Zadeh DLs.
- R3 The TBox is acyclic (or it can be converted into an equivalent acyclic TBox) [19] or empty. This restriction does not apply for Zadeh DLs, or if the truth space is finite.
- R4 The semantics is restricted to witnessed models.
- R5 The semantics of the entailment degree assumes an implication that satisfies (OP) and (NFNP).
- R6 There is a reasoning algorithm for type-1 DLs based on a combination of a tableau algorithm with an optimization problem $\mathcal{P}$ generated during the inference process such that the BED/BSD problems can be reduced to a minimization/-maximization problem w.r.t. operational research equations and that the following conditions hold:

$$bed(\mathcal{O}, \mathfrak{a}) = \min x. \text{ s.t. } \mathcal{O} \cup \{\langle \mathfrak{a} \leq x \rangle\} \text{ is satisfiable} \quad (15)$$

$$bsd(\mathcal{O}, \mathfrak{a}) = \max x. \text{ s.t. } \mathcal{O} \cup \{\langle \mathfrak{a} \geq x \rangle\} \text{ is satisfiable} . \quad (16)$$

- R7 Both the fuzzy logic operators and the fuzzy membership functions need to be representable as inequations in $\mathcal{P}$.
- R8 Inequations of the form $exp > 0$ need to be representable, where exp is an expression in $\mathcal{P}$. For ease, we assume that a sufficiently small $\epsilon > 0$ can be determined such that the condition $exp > 0$ can be encoded as $exp \geq \epsilon$. Note that this is always possible if the truth space is finite, but a priori finiteness may not always be necessary: it might be possible to take the ϵ after analyzing the degrees of truth in the ontology and/or the numerical values in the generated optimization problem. We do not address this further: for the sake of our purpose, it suffices to assume that $exp > 0$ is representable.

These restrictions might seem restrictive, though some significant type-1 fuzzy DLs with concrete domains that satisfy the restrictions enumerated above are Zadeh $\mathcal{ALC}(\mathbf{D})$ [55], Łukasiewicz $\mathcal{ALC}(\mathbf{D})$ assuming a restricted TBox [55],[11] and general $\mathcal{ALC}(\mathbf{D})$ extended with an involutive negation assuming a restricted TBox [12].[12]

In the following we comment on the few restrictions we proposed before:

Remark 12. The reason to assume that the TBox is acyclic is that it has been shown that reasoning is undecidable for several fuzzy DLs in the presence of GCIs. This is the case e.g., in Łukasiewicz [25] and Product fuzzy DLs [3,9,21]. However, it is worth to stress that non-acyclic TBoxes can usually be (totally or partially) transformed into equivalent acyclic TBoxes using an absorption algorithm, as experimentally shown in [19].

Remark 13. A side effect of restriction R6 above is that the minima (resp. maxima) in Eq. (15) (resp. Eq. (16)) exist and can be determined via operation research methods. Also, e.g., in the case of the BED, as $\mathcal{O} \cup \{\langle \mathfrak{a} \leq x \rangle\}$ has to be satisfiable, from the computed solution to the optimization problem we may materialize a model $\mathcal{I}$ of $\mathcal{O} \cup \{\langle \mathfrak{a} \leq \alpha \rangle\}$, where $bed(\mathcal{O}, \mathfrak{a}) = \alpha$ and, specifically, $\mathfrak{a}^{\mathcal{I}} = \alpha$ (as α is a minimum) holds. A similar argument applies for the BSD.

This is captured by means of some *witnessing conditions* for BED and BSD. These witnessing conditions are generalized to type-2 fuzzy DLs, which in the following we will assume to hold: for $\mathfrak{a} \in \{a{:}C, (a_1, a_2){:}R, (a, v){:}T, C_1 \sqsubseteq C_2\}$ and $\mathcal{O}$ consistent,

$$\text{if } bed(\mathcal{O}, \mathfrak{a}) = \alpha \ \text{ then } \ (\mathcal{I}(\mathcal{O}_Q) \Rightarrow (\mathfrak{a}^{\mathcal{I}} = \alpha)) = 1\,; \tag{17}$$

$$\text{if } bsd(\mathcal{O}, C) = \alpha \ \text{ then } \exists x \in \Delta^{\mathcal{I}} \text{ s.t. } (\mathcal{I}(\mathcal{O}_Q) \Rightarrow C^{\mathcal{I}}(x)) = \alpha \tag{18}$$

for some witnessed model $\mathcal{I}$ of $\mathcal{O}$ (i.e., $\mathcal{I} \models \mathcal{O}_U$ and $\mathcal{I}(\mathcal{O}_Q) > 0$). Note that Eqs. (15)–(18) hold if the truth space is finite.

Remark 14. For some type-1 DLs the BED cannot be computed using a minimization problem as in Eq. (15) because the minima is not guaranteed to exist (similarly for the maxima in the BSD problem). An example is Gödel $\mathcal{ALC}(\mathbf{D})$ over $[0, 1]$ extended with concept implication and Łukasiewicz negation or axioms of the form $\tau \leq \beta$. For instance, consider the following (strictly speaking propositional) ontology $\mathcal{O}$, with two axioms

$$\langle a{:}A \rightarrow B \leq 0.3\rangle$$
$$\langle a{:}B \geq 0.3\rangle\ .$$

It is easy to see that in any model $\mathcal{I}$ of $\mathcal{O}$, $B^{\mathcal{I}}(a^{\mathcal{I}}) = 0.3$ and $A^{\mathcal{I}}(a^{\mathcal{I}}) > 0.3$. Therefore, according to Eq. (12), $bed(\mathcal{O}, a{:}A) = 0.3$. However, in this case the BED is not witnessed, that is, the minimal x such that $\mathcal{O} \cup \{a{:}A \leq x\}$ is satisfiable actually does not exist, and indeed $\mathcal{O} \cup \{a{:}A = 0.3\}$ is inconsistent. This motivates why existing reasoning algorithms for type-1 DLs based on a combination of a tableau algorithm with an optimization problem need to be restricted to witnessed BED and BSD [12].

A possible solution is to restrict to a finite truth space, as Gödel DLs do not satisfy the witnessing model property anyway. For instance, assuming $\mathcal{N}_{1000}$, the fuzzyDL reasoner determines $bed(\mathcal{O}, a{:}A) = 0.301$, as expected.

Remark 15. One might wonder what to do if we have to deal with a fuzzy DL for which e.g., the BED witnessing condition does not hold.While this problem is beyond the scope of this work because it also affects type-1 DLs, let us nevertheless recall a partial solution to it. We know that $\mathcal{O} \models \langle \mathfrak{a} \geq \alpha \rangle$ iff $\mathcal{O} \cup \{\langle \mathfrak{a} < \alpha \rangle\}$ is not satisfiable [54]. Therefore,

$$bed(\mathcal{O}, \mathfrak{a}) = \sup x. \text{ s.t. } \mathcal{O} \cup \{\langle \mathfrak{a} < x \rangle\} \text{ is not satisfiable}\ . \tag{19}$$

While operational research methods seem not well-suited to solve a maximization problem over a set of operational research constraints that should *not* be satisfied, we may still use them somewhat (or any alternative satisfiability checking procedure) in a similar way as suggested in [54]: select some α and test whether $\mathcal{O} \cup \{\langle \mathfrak{a} < \alpha \rangle\}$ is satisfiable and then, by a binary search over the allowed values for α, search for the maximal α such that the previous satisfiability test fails. However, this method is not guaranteed to converge in finite time and may find only an approximate solution. Of course, it converges in finite time when α may take a finite number of possible values, but then the BED is witnessed.

Now, we will show how, under the restrictions assumed before, we may reason within type-2 fuzzy DLs.

At first, we have assumed that the fuzzy logic operators and the fuzzy predicates are $\mathcal{P}$ representable. In particular, we require the following constraints to be $\mathcal{P}$ representable:

- $x_1 \Rightarrow x_2 = z$,
- $\bigwedge_i x_i = z$,

[11] To illustrate the importance of these two logics, *fuzzyDL* reasoner supports a more expressive language w.r.t. machine-representable set of degrees of truth [18].

[12] This language extended with concept implication does not satisfy R6; see Remarks 13–14.

- $\mathbf{d}(x_i) \geq z$,
- $\mathbf{d}(x_i) \leq z$,
- $\mathbf{t}(x_i) \geq z$, and
- $\mathbf{t}(x_i) \leq z$,

where x_i, z are [0, 1]-variables, $\mathbf{d}$ is a type-1 datatype and $\mathbf{t}$ is a type-1 truth-type. For the sake of completeness, Appendix C illustrates how to represent those restrictions in some popular cases. Note that the four latter constraints trivially imply that $\mathbf{d}(x) = z$ and $\mathbf{t}(x) = z$ are $\mathcal{P}$ representable as well.

In general, $\mathcal{P}$ can be a Mixed Integer Non-Linear Programming (MINLP) problem, but in some particular cases we can have an easier problem. In Zadeh, Łukasiewicz, and Gödel DLs, $\mathcal{P}$ is a Mixed Integer Linear Programming (MILP) problem [51], while for the Product t-norm $\mathcal{P}$ is a Mixed Integer Quadratically Constrained Programming (MIQCP) problem. Note that all the operators in Fig. 1 are MILP representable.

In Appendix B we include an example of reasoning algorithm for general type-1 $\mathcal{ALC}(\mathbf{D})$ extended with Łukasiewicz negation and an implication. Such algorithm can be applied directly to an ontology $\mathcal{O}$ only if $\mathcal{O}_Q = \emptyset$. Our extended algorithm will be able to deal with axioms of the form $\langle \mathfrak{a} \geq x_{\mathfrak{a}} \rangle$, which can be handled similarly as $\langle \mathfrak{a} \geq \alpha \rangle$, but replacing the value α with the variable $x_{\mathfrak{a}}$ and observing that in the initialization step, it is not necessary to add constraints of the form $x_{\mathfrak{a}} \geq x_{\mathfrak{a}}$ that Lines 7, 13, and 19 of Algorithm 1 would generate.

7.3. Computing the BED and the BSD in type-2 ontologies

At first, let us note that the following result holds straightforwardly.

Proposition 7. *Let $\mathcal{O}$ be a type-2 ontology and $\Rightarrow$ an implication defining the entailment degree of $\mathcal{O}$ satisfying (OP) and (NFNP). Then, $\mathcal{O}$ is consistent iff $bed(\mathcal{O}, a{:}\bot) = 0$ iff $bsd(\mathcal{O}, \bot) < 1$, where a is a new individual.*

Proof. Let us firstly prove the equivalence between consistency and the BED. If $\mathcal{O}$ is consistent, there is a model $\mathcal{I}$ such that $\mathcal{I}(\mathcal{O}_U) = 1$ and $\mathcal{I}(\mathcal{O}_Q) > 0$. In every model $\mathcal{I}$, $\bot^{\mathcal{I}}(x) = 0$ for every $x \in \Delta^{\mathcal{I}}$, so in particular $\bot^{\mathcal{I}}(a^{\mathcal{I}}) = 0$. Thus, if $\gamma = 0$, then $(a{:}\bot \geq \gamma)^{\mathcal{I}} = 1$, while if $\gamma \in (0, 1]$ then $(a{:}\bot \geq \gamma)^{\mathcal{I}} = 0$. By definition, $bed(\mathcal{O}, a{:}\bot) = \sup\{\gamma \in [0, 1] \mid ed(\mathcal{O}, \langle a{:}\bot, \geq \gamma \rangle) = 1\}$. Thus, we are interested in those $\gamma \in [0, 1]$ such that $ed(\mathcal{O}, \langle a{:}\bot, \geq \gamma \rangle) = \inf_{\mathcal{I} \models \mathcal{O}_U, \mathcal{I}(\mathcal{O}_Q) > 0} \mathcal{I}(\mathcal{O}_Q) \Rightarrow (a{:}\bot \geq \gamma)^{\mathcal{I}} = 1$. Since $\Rightarrow$ verifies (OP), in every $\mathcal{I}$ we must have that $\mathcal{I}(\mathcal{O}_Q) \leq (a{:}\bot \geq \gamma)^{\mathcal{I}}$. Since $\mathcal{I}(\mathcal{O}_Q) > 0$ and $(a{:}\bot \geq \gamma)^{\mathcal{I}} \in \{0, 1\}$, it follows that $(a{:}\bot \geq \gamma)^{\mathcal{I}} = 1$, so $\gamma = 0$ and $bed(\mathcal{O}, a{:}\bot) = \sup\{0\} = 0$.

To prove the other direction, assume that $bed(\mathcal{O}, a{:}\bot) = 0$. If $\mathcal{O}$ is inconsistent, it has no models, so $bed(\mathcal{O}, a{:}\bot) = \inf \emptyset = 1$. Since this is in contradiction with the assumption that the BED is 0, $\mathcal{O}$ must be consistent.

The equivalence between consistency and the BSD is similar. If $\mathcal{O}$ is consistent, there is a model $\mathcal{I}$ such that $\mathcal{I}(\mathcal{O}_U) = 1$ and $\mathcal{I}(\mathcal{O}_Q) > 0$. In every model $\mathcal{I}$, $\bot^{\mathcal{I}}(x) = 0$ for every $x \in \Delta^{\mathcal{I}}$, so $bsd(\mathcal{O}, \bot) = \sup_{\mathcal{I} \models \mathcal{O}_U, \mathcal{I}(\mathcal{O}_Q) > 0} \sup_{x \in \Delta^{\mathcal{I}}} \{\mathcal{I}(\mathcal{O}_Q) \Rightarrow \bot^{\mathcal{I}}(x)\} = \sup_{\mathcal{I} \models \mathcal{O}_U, \mathcal{I}(\mathcal{O}_Q) > 0} \{\mathcal{I}(\mathcal{O}_Q) \Rightarrow 0\}$. Since $\mathcal{I}(\mathcal{O}_Q) > 0$ and $\Rightarrow$ satisfies (NFNP), it follows that $\mathcal{I}(\mathcal{O}_Q) \Rightarrow 0 < 1$, so $bsd(\mathcal{O}, \bot) < 1$.

To prove the other direction, assume that $bsd(\mathcal{O}, \bot) < 1$. If $\mathcal{O}$ is inconsistent, it has no models, so $bsd(\mathcal{O}, \top) = \sup \emptyset = 1$. Since this is in contradiction with the assumption that the BSD is smaller than 1, $\mathcal{O}$ must be consistent. □

The entailment of a type-1 axiom $\tau = \langle \mathfrak{a} \geq \alpha \rangle$, with $\alpha \in (0, 1]$, is also not of particular interest as it suffices to compute $\gamma = bed(\mathcal{O}, \mathfrak{a})$ and check whether $\gamma \geq \alpha$.

Our goal now is to provide procedures determining $bed(\mathcal{O}, \mathfrak{a})$ when $\mathcal{O}_Q \neq \emptyset$, where $\mathfrak{a} \in \{a{:}C, (a, b){:}R, C_1 \sqsubseteq C_2\}$, and $bsd(\mathcal{O}, C)$ for a concept C.

Now, the following result can be shown.

Proposition 8. *Let $\mathcal{O} = \mathcal{O}_U \cup \mathcal{O}_Q$ be a consistent type-2 ontology with a witnessed model, and assume the BED/BSD witnessing conditions in Eqs. (17) and (18) hold. Then*

$$bed(\mathcal{O}, \mathfrak{a}) = \min x. \textit{ such that } \mathcal{O}_U \cup \{\Gamma_Q > 0, (\Gamma_Q \to (\mathfrak{a} \leq x)) = 1\} \textit{ is satisfiable} \tag{20}$$

$$bsd(\mathcal{O}, C) = \max x. \textit{ such that } \mathcal{O}_U \cup \{\Gamma_Q > 0, (\Gamma_Q \to a{:}C) \geq x\} \textit{ is satisfiable} \tag{21}$$

where x is a [0, 1]*-valued variable, $\mathfrak{a} \in \{a{:}C, (a_1, a_2){:}R, (a, v){:}T, C_1 \sqsubseteq C_2\}$, a is new individual, and C is a fuzzy concept.*

Proof. Let us prove the case of Eq. (20); the case of Eq. (21) can be shown similarly.

Assume $bed(\mathcal{O}, \mathfrak{a}) = \alpha$ and consider $\mathcal{C} = \mathcal{O}_U \cup \{\Gamma_Q > 0, (\Gamma_Q \to (\mathfrak{a} \leq x)) = 1\}$. As the witnessing condition 17 is satisfied and $\mathcal{O}$ is consistent, for some witnessed model $\mathcal{I}$ of $\mathcal{O}$ (i.e., $\mathcal{I} \models \mathcal{O}_U$ and $\mathcal{I}(\mathcal{O}_Q) > 0$) we have that $\mathcal{I}(\mathcal{O}_Q) \Rightarrow \langle \mathfrak{a} = \alpha \rangle^{\mathcal{I}} = 1$ and, thus, $(\Gamma_Q > 0)^{\mathcal{I}} = 1$ and $(\Gamma_Q \to (\mathfrak{a} \leq \alpha))^{\mathcal{I}} = 1$. As a consequence, $\mathcal{C}$ is satisfiable with solution $x = \alpha$.

Now, assume to the contrary that there is another model $\mathcal{I}'$ for $\mathcal{C}$ with $x = \alpha' < \alpha$, i.e., α is not minimal. So, $\mathcal{I}' \models \mathcal{O}_U, \mathcal{I}'(\mathcal{O}_Q) > 0$ and $(\Gamma_Q \to (\mathfrak{a} \leq \alpha'))^{\mathcal{I}'} = 1$. As $\mathcal{I}'(\mathcal{O}_Q) > 0$, $(\mathfrak{a} \leq \alpha')^{\mathcal{I}'} = 1$ follows. Since $bed(\mathcal{O}, \mathfrak{a}) = \alpha$, by Eq. (11) we know that $ed(\mathcal{O}, \langle \mathfrak{a} \geq \alpha \rangle) = 1$. That is, for all interpretations $\mathcal{J}$ with $\mathcal{J} \models \mathcal{O}_U$ and $\mathcal{J}(\mathcal{O}_Q) > 0$, $(\Gamma_{\mathcal{O}_Q} \to (\mathfrak{a} \geq \alpha))^{\mathcal{J}} = 1$ has to hold, and, thus $(\Gamma_{\mathcal{O}_Q} \to (\mathfrak{a} \geq \alpha))^{\mathcal{I}'} = 1$. As $\mathcal{I}'(\mathcal{O}_Q) > 0$, $(\mathfrak{a} \geq \alpha)^{\mathcal{I}'} = 1$ follows, contradicting $(\mathfrak{a} \leq \alpha')^{\mathcal{I}'} = 1$ because $\alpha' < \alpha$. So, $x = \alpha$ is also the minimal solution for $\mathcal{C}$, which concludes. □

We next show that both optimization problems of Proposition 8 can be encoded into a type-1 fuzzy DL optimization problem, for which an algorithm is assumed to exist. Hence, let us determine both $bed(\mathcal{O}, \mathfrak{a})$, where $\mathfrak{a}$ is of the form $a{:}C$, $(a, b){:}R$, $(a, \mathrm{v}){:}T$ or $C_1 \sqsubseteq C_2$, and $bsd(\mathcal{O}, C)$.

At first, for any expression $\tau' = \langle \mathfrak{a}', \mathbf{t}' \rangle$, with $\mathbf{t}'$ being a type-1 truth-type that occurs in the optimization problems of Proposition 8, we introduce a new $[0, 1]$-variable $x_{\tau'}$ that will hold the degree of truth of τ'. In order to connect the value of the variable $x_{\tau'}$ to the degree of truth of the axiom τ', we replace τ' with $\langle \mathfrak{a}' \geq x_{\mathfrak{a}'} \rangle$, and consider the constraints:

$$\begin{array}{lll} \mathbf{t}'(x_{\mathfrak{a}'}) & \leq & x_{\tau'} \\ \mathbf{t}'(x_{\mathfrak{a}'}) & \geq & x_{\tau'} \, , \end{array} \tag{22}$$

which are by assumption $\mathcal{P}$ representable.

If we have $\tau' = \langle \mathfrak{a}', \mathbf{t}' \rangle$, with $\mathbf{t}' = (\mathbf{t}'_L, \mathbf{t}'_U)$ being a type-2 truth-type, we proceed similarly but replacing Eq. (22) with:

$$\begin{array}{lll} \mathbf{t}'_L(x_{\mathfrak{a}'}) & \leq & x_{\tau'} \\ \mathbf{t}'_U(x_{\mathfrak{a}'}) & \geq & x_{\tau'} \, , \end{array} \tag{23}$$

which are again by assumption $\mathcal{P}$ representable.

Finally, in case a type-2 datatype $\mathbf{d} = (\mathbf{d}_L, \mathbf{d}_U)$ is involved in concept expressions, there is also an implicit restriction on $\mathbf{d}$ as dictated by Eq. (3). This has to be reflected also in all the inference rules. For instance, if we consider the reasoning rules in Table B.3, this idea must be used to extend the rules $(\exists_{\mathbf{d}})$, $(\forall_{\mathbf{d}})$, $(\neg_4)$, and $(\neg_6)$. The required extension is detailed in Appendix D, but we will summarize it here.

In the type-1 rules where a datatype $\mathbf{d}$ occurs, there is a variable of the form $x_{w:\mathbf{d}}$. In the type-2 reasoning algorithm, this variable $x_{w:\mathbf{d}}$ is replaced with the expression $x_{w:\mathbf{d}} \otimes y$, where $y = (x_{w:\mathbf{d}} \in [z_1, z_2])$. Next, we must add the constraints

$$\begin{array}{lll} z_1 & \geq & \mathbf{d}_L(x_{\mathfrak{a}'}) \\ z_2 & \leq & \mathbf{d}_U(x_{\mathfrak{a}'}) \\ z_1, z_2 & \in & [0, 1] \, . \end{array} \tag{24}$$

Finally, we encode the restriction $y = (x_{w:\mathbf{d}} \in [z_1, z_2])$ by means of the following constraints (note that restriction R8 is assumed):

$$\begin{array}{lll} y + 1 & = & y_1 + y_2 \\ (1 - y) + x_{w:\mathbf{d}} & \geq & z_1 \\ x_{w:\mathbf{d}} & \leq & z_2 + (1 - y) \\ x_{w:\mathbf{d}} - z_1 - 2y_1 & < & 0 \\ 2y_2 + x_{w:\mathbf{d}} - z_2 & > & 0 \\ y, y_1, y_2 & \in & \{0, 1\} \, , \end{array} \tag{25}$$

The intuitive idea is the following. The first constraint restricts to three possible cases, since the variables y, y_1, y_2 are binary.

- If $y = 1, y_1 = 1, y_2 = 1$, it follows that $x_{w:\mathbf{d}} \in [z_1, z_2]$.
- If $y = 0, y_1 = 0, y_2 = 1$, we have that $x_{w:\mathbf{d}} < z_1$.
- If $y = 0, y_1 = 1, y_2 = 0$, we have that $x_{w:\mathbf{d}} > z_2$.

Hence, $y = (x_{w:\mathbf{d}} \in [z_1, z_2])$ is correctly encoded.

We are ready now to address the BED and BSD problems.

The case of $bed(\mathcal{O}, \mathfrak{a})$. Let us consider Eq. (20) in Proposition 8. Please note that $bed(\mathcal{O}, (a_1, a_2){:}R) = bed(\mathcal{O} \cup \{\langle a_2{:}A \geq 1 \rangle\}, a_1{:}\exists R.A)$ where A is a new atomic concept, and that $bed(\mathcal{O}, (a, \mathrm{v}){:}T)$ is equivalent to $bed(\mathcal{O}, a{:}\exists T.\mathtt{crisp}_{\mathrm{v},\mathrm{v}})$ as shown in Remark 8. So, we do not need to address further these cases. How to encode $\mathcal{O}_U$ is already known (see, e.g., [56]), so we do not address it below.

Let us consider first $\mathfrak{a} = a{:}C$. We need to encode

$$(\Gamma_Q \to (a{:}C \leq x)) = 1 \, .$$

To this end, let us add $\langle a{:}C \leq x \rangle$ to $\mathcal{O}_U$. Now, informally, $\tau = \langle a{:}C \leq x \rangle$ is either true or false. If true then $(\Gamma_Q \to 1) = 1$ always holds. Otherwise, $(\Gamma_Q \to 0) = 1$ requires that $\Gamma_Q = 0$ has to hold, which contradicts the assumption that $\Gamma_Q > 0$. Thus, we consider a variable x_{Γ_Q} that will hold the degree of truth of Γ_Q and assess that $x_{\Gamma_Q} > 0$ (restriction R8 is assumed). Then, $(\Gamma_Q \to (a{:}C \leq x)) = 1$ and $\Gamma_Q > 0$ can be encoded as

$$\begin{array}{ll} x_{\Gamma_Q} & > 0 \\ x_{a:C} & \leq x \\ \bigwedge_{\tau_j \in \mathcal{O}_Q} x_{\tau_j} & = x_{\Gamma_Q} \\ x_{\Gamma_Q}, x_{\tau_j}, x_{a:C} & \in [0,1] \,. \end{array} \tag{26}$$

Consider now the case $\mathfrak{a} = C_1 \sqsubseteq C_2$. If the fuzzy DL includes concept implication, we can proceed by taking into account that $bed(\mathcal{O}, C_1 \sqsubseteq C_2) = \min x$. such that $\mathcal{O}_U \cup \{(\Gamma_Q \to (a{:}\neg_{Ł}(C_1 \to C_2) \geq 1 - x)) = 1\}$ is satisfiable. However, we will give an encoding that does not need to assume that concept implication is in the language:

1. add $\langle a{:}C_1 \leq x_1 \rangle$ and $\langle a{:}C_2 \geq x_2 \rangle$ to $\mathcal{O}_U$, for new $[0,1]$-variables x_1 and x_2;
2. replace Eq. (26) with

$$\begin{array}{ll} x_{\Gamma_Q} & > 0 \\ x_1 \Rightarrow x_2 & = x_{imp} \\ x_{imp} & \leq x \\ \bigotimes_{\tau_j \in \mathcal{O}_Q} x_{\tau_j} & = x_{\Gamma_Q} \\ x_{\Gamma_Q}, x_{\tau_j}, x_1, x_2, x_{imp} & \in [0,1] \,. \end{array} \tag{27}$$

The following can easily be shown.

Proposition 9. *If a type-2 ontology $\mathcal{O} = \mathcal{O}_Q \cup \mathcal{O}_U$ has a witnessed model, all assumptions R1–R8 hold, and the BED satisfies the witnessing condition in Eq. (17), then for $\mathfrak{a} \in \{a{:}C, (a_1, a_2){:}R, (a, v){:}T, C_1 \sqsubseteq C_2\}$ the extended algorithm terminates and computes $bed(\mathcal{O}, \mathfrak{a})$.*

The case of $bsd(\mathcal{O}, C)$. This case is pretty similar to the case before. At first, we add $\langle a{:}C \geq x_{a:C} \rangle$ to $\mathcal{O}_U$, where a is a new individual.

Now, we introduce a variable x_{QC} holding the degree of truth of $(\Gamma_Q \to a{:}C)$. To encode $(\Gamma_Q \to a{:}C) \geq x$ and $\Gamma_Q > 0$, we consider the following constraints:

$$\begin{array}{ll} x_{\Gamma_Q} & > 0 \\ x_{QC} & \geq x \\ x_{\Gamma_Q} \Rightarrow x_{n:C} & = x_{QC} \\ \bigotimes_{\tau_j \in \mathcal{O}_Q} x_{\tau_j} & = x_{\Gamma_Q} \\ x_{QC}, x_{\Gamma_Q}, x_{\tau_j} & \in [0,1] \,. \end{array} \tag{28}$$

Proposition 10. *If a type-2 ontology $\mathcal{O} = \mathcal{O}_Q \cup \mathcal{O}_U$ has a witnessed model, all assumptions R1–R8 hold, and the BSD satisfies the witnessing condition in Eq. (18), then the extended algorithm terminates and computes $bsd(\mathcal{O}, C)$.*

Remark 16. Our encodings are designed to simplify Appendix C but are not optimal: we can easily save some variables and constraints. For example, in Eq. (27) we can omit x_{imp} and replace the second and the third constraints with $x_1 \Rightarrow x_2 \leq x$. Similarly, in Eq. (28) we can omit x_{QC} and replace the second and the third constraints with $x_{\Gamma_Q} \Rightarrow x_{a:C} \geq x$.

8. Conclusions and future work

This paper addressed the existing gap between type-2 fuzzy ontology applications and type-2 fuzzy ontology formalisms, guided by the so far considered applications of type-2 fuzzy ontologies. In particular, we have introduced a novel syntax and semantics of type-2 fuzzy DLs. Our approach makes the management of some uncertainty in the fuzzy membership functions possible.

The main feature of the proposed logic is the existence of type-2 datatypes and type-2 axioms. The supported type-2 datatypes are the main requirement of type-2 fuzzy ontology applications. The supported type-2 axioms are satisfied to some degree by the models of a type-2 ontology, as opposed to traditional fuzzy ontology axioms.

We have also discussed the relation with type-0, type-1, and interval fuzzy DLs. In particular, we have shown that previous type-2 fuzzy DLs are generally subsumed by our proposal.

This paper also provides the basics to provide ontology developers with some tools to develop type-2 fuzzy ontology applications. In this regard, we have proposed an extension of the language Fuzzy OWL 2 to support type-2 fuzzy ontologies, and we have also discussed how to extend existing type-1 DL reasoning algorithms to support type-2 DLs under some restrictions.

Future work will include the implementation of the extensions of the Fuzzy OWL 2 language and the fuzzyDL reasoning system [18] to support type-2 fuzzy ontologies. To this end, it will also be necessary to investigate whether existing optimizations for type-1 ontologies, such as [16,17,19], are still applicable in type-2 ontologies and to propose new optimization techniques: in fact, we have witnessed that optimizations make a huge difference in terms of computation time in practice.

Finally, it is worth to stress that our approach to manage type-2 fuzzy DLs is not unique and alternative definitions could also be studied.

Acknowledgements

This research work has been partially supported by the Spanish Ministry of Economy and Competitiveness (projects TIN2013-46238-C4-4-R and TIN2016-78011-C4-3-R) and DGA/Feder (research group T79-SID). We would like to thank María Ros for some helpful discussions and the anonymous reviewers for their valuable comments on an earlier version of this paper, particularly for the example in Remark 14.

Appendix A. Mapping fuzzy DLs to an extended fuzzy FOL

We show here how to map a fuzzy ontology into a fuzzy FOL formula. To ease the mapping of fuzzy datatypes and truth-types to FOL we will also slightly extend the fuzzy FOL language by allowing the so-called *built-in* predicates and the so-called *fuzzy modifiers*. Both have a fixed interpretation, and the former will be used to accommodate datatypes and truth-values, while the latter will be used to accommodate truth-types.

To start with, assume that the ontology $\mathcal{O}$ is of type-1 and does not have datatypes. Then the following inductively defined mapping σ, maps a type-1 axiom to a fuzzy FOL formula supporting type-0 truth-types[13]:

$$
\begin{aligned}
\sigma(\langle \mathfrak{a} \geq \alpha \rangle) &\mapsto \sigma(\mathfrak{a}) \geq \alpha \\
\sigma(\langle \mathfrak{a} \leq \alpha \rangle) &\mapsto \sigma(\mathfrak{a}) \leq \alpha \\
\sigma(a{:}C) &\mapsto \sigma(C, a) \\
\sigma((a_1, a_2){:}R) &\mapsto R(a_1, a_2) \\
\sigma((a, \mathrm{v}){:}T) &\mapsto T(a, \mathrm{v}) \\
\sigma(C_1 \sqsubseteq C_2) &\mapsto \forall x.\sigma(C_1, x) \to \sigma(C_2, x) \\
\sigma(C, z) &\mapsto 1 \text{ if } C = \top \\
\sigma(C, z) &\mapsto 0 \text{ if } C = \bot \\
\sigma(C, z) &\mapsto A(z) \text{ if } C = A \\
\sigma(C, z) &\mapsto \sigma(C_1, z) \wedge \sigma(C_2, z) \text{ if } C = C_1 \sqcap C_2 \\
\sigma(C, z) &\mapsto \sigma(C_1, z) \vee \sigma(C_2, z) \text{ if } C = C_1 \sqcup C_2 \\
\sigma(C, z) &\mapsto \neg\sigma(D, z) \text{ if } C = \neg D \\
\sigma(C, z) &\mapsto \exists y.R(z, y) \wedge \sigma(D, y) \text{ if } C = \exists R.D \\
\sigma(C, z) &\mapsto \forall y.R(z, y) \to \sigma(D, y) \text{ if } C = \forall R.D \ .
\end{aligned}
$$

We next accommodate fuzzy datatypes. So, consider a type-1 datatype theory $\mathbf{D} = \langle \Delta^{\mathbf{D}}, \cdot^{\mathbf{D}} \rangle$. Then we define

$$
\begin{aligned}
&\sigma(C, z) \mapsto \exists y.T(z, y) \wedge \mathbf{d}(y) \text{ if } C = \exists T.\mathbf{d} \\
&\sigma(C, z) \mapsto \forall y.T(z, y) \to \mathbf{d}(y) \text{ if } C = \forall T.\mathbf{d} \ ,
\end{aligned}
$$

where fuzzy FOL has been extended by allowing unary built-in predicates $\mathbf{d}$ to occur with fixed interpretation $\mathbf{d}^{\mathcal{I}} = \mathbf{d}^{\mathbf{D}}$, for a fuzzy FOL interpretation $\mathcal{I}$. The case of a type-2 datatype theory $\mathbf{D} = \langle \Delta^{\mathbf{D}}, \cdot^{\mathbf{D}} \rangle$ is more complicated. In that case,

$$
\begin{aligned}
&\sigma(C, z) \mapsto \exists y \exists r.\mathbb{Q}(r) \wedge T(z, y) \wedge r \wedge (\mathbf{d}_L(y) \leq r) \wedge (r \leq \mathbf{d}_U(y)) \text{ if } C = \exists T.(\mathbf{d}_L, \mathbf{d}_U) \\
&\sigma(C, z) \mapsto \forall y \exists r.(\mathbb{Q}(r) \wedge T(z, y)) \to (r \wedge (\mathbf{d}_L(y) \leq r) \wedge (r \leq d_U(y))) \text{ if } C = \forall T.(\mathbf{d}_L, \mathbf{d}_U) \ ,
\end{aligned}
$$

where the unary built-in predicate $\mathbb{Q}$ has the fixed interpretation of rational numbers.

Finally, let us accommodate type-2 axioms. Hence, consider a type-2 truth-type theory $\mathbf{T} = \langle \mathcal{N}, \cdot^{\mathbf{T}} \rangle$ and let $p_{\mathfrak{a}}$ be a new propositional letter for $\mathfrak{a} \in \{a{:}C, (a_1, a_2){:}R, (a, \mathrm{v}){:}T, C_1 \sqsubseteq C_2\}$. We define:

[13] $\wedge, \vee, \neg$, and $\to$ denote a t-norm, t-conorm, negation and an implication function, respectively, at syntactic level, z is either a variable or an individual constant. Rationals and type-0 truth-types may occur as atoms in formulae. The interpretation of rationals and type-0 truth-types is fixed in the obvious way. For formulae φ and ψ, $\varphi \leq \psi$ and $\varphi \geq \psi$ are shorthands for $1 \to (\varphi \to \psi)$ and $1 \to (\psi \to \varphi)$, respectively, while $\varphi \leftrightarrow \psi$ is a shorthand for $(\varphi \to \psi) \wedge (\psi \to \varphi)$.

$$\sigma(\langle \mathfrak{a}, \mathbf{t} \rangle) \mapsto ((p_{\mathfrak{a}} \leftrightarrow \sigma(\mathfrak{a})) \geq 1) \wedge \mathbf{t}(p_{\mathfrak{a}})$$

$$\sigma(\langle \mathfrak{a}, (\mathbf{t}_L, \mathbf{t}_U) \rangle) \mapsto ((p_{\mathfrak{a}} \leftrightarrow \sigma(\mathfrak{a})) \geq 1) \wedge \exists r.\mathbb{Q}(r) \wedge r \wedge (\mathbf{t}_L(p_{\mathfrak{a}}) \leq r) \wedge (r \leq \mathbf{t}_U(p_{\mathfrak{a}}))\,,$$

where the fuzzy FOL has been extended by allowing *fuzzy modifiers* of the form $\mathbf{t}(p)$ to occur, for propositional letter p, and $\mathbf{t}$ has a fixed interpretation $\mathbf{t}^{\mathcal{I}} = \mathbf{t}^{\mathbf{T}}$. Note that the formula $(p_{\mathfrak{a}} \leftrightarrow \sigma(\mathfrak{a})) \geq 1)$ encodes the fact that $p_{\mathfrak{a}}$ and $\sigma(\mathfrak{a})$ should have the same degree of truth in any model of the formula.

At last, given a type-2 ontology $\mathcal{O}$ we define

$$\Gamma_{\mathcal{O}} = \Big(\bigwedge_{\tau \in \mathcal{O}_U} \sigma(\tau)\Big) \wedge \Big(\bigwedge_{\tau \in \mathcal{O}_Q} \sigma(\tau)\Big) \qquad \text{(A.1)}$$

where the top-level $\wedge$ are interpreted as Gödel t-norm. $\Gamma_{\mathcal{O}_U}$ is used to denote the left conjunct and $\Gamma_{\mathcal{O}_Q}$ is used to denote the right conjunct in Eq. (A.1), respectively, and, thus, $\Gamma_{\mathcal{O}} = \Gamma_{\mathcal{O}_U} \wedge \Gamma_{\mathcal{O}_Q}$. We will also omit the reference to $\mathcal{O}$ if clear from context.

Appendix B. A reasoning algorithm for type-1 $\mathcal{ALC}(\mathbf{D})$

To make this paper somewhat self-contained, we present here some essential details about the reasoning algorithm we rely on. This algorithm works for type-1 $\mathcal{ALC}(\mathbf{D})$ extended with Łukasiewicz negation and concept implication. The semantics is general, as it considers any left-continuous t-norm and its residuum. For ease of presentation, we restrict to the datatypes in Fig. 1 and assume that all the axioms are of the form $\langle \tau \geq \alpha \rangle$, but recall that $\langle a : C \leq \alpha \rangle$ can be represented as $\langle a : \neg_L C \geq 1 - \alpha \rangle$.

So, let $\mathcal{O} = \langle \mathcal{A}, \mathcal{T} \rangle$ be a type-1 ontology with $\mathcal{T}$ acyclic. $bed(\mathcal{O}, a{:}C)$ can be computed as $\min x$. such that $\mathcal{O} \cup \{a{:}\neg_Ł C \geq 1 - x\rangle\}$ is consistent. This is shown in Algorithm 1, which is based on the algorithm in [12], but does not use 'lazy unfolding' for simplicity (see, e.g., [56]) and considers fuzzy datatypes [55].[14] The set of rules the algorithm is based on is illustrated in Table B.3.

Algorithm 1 Reasoning algorithm for type-1 ontologies.

Input: A concept assertion $a{:}C$, a fuzzy KB $\mathcal{O}$
Output: $\min x$. such that $\mathcal{O} \cup \{\langle a{:}\neg_Ł C \geq 1 - x \rangle\}$ is consistent.

$\mathcal{O} \leftarrow \mathcal{O} \cup \{\langle a{:}\neg_Ł C \geq 1 - x \rangle\}$
create an empty forest $\mathcal{F}$
$\mathcal{C}_{\mathcal{F}} \leftarrow \emptyset$
for each concept assertion $\langle a{:}C \geq \alpha \rangle \in \mathcal{O}$ **do**
    create a node v_a in $\mathcal{F}$ if it does not exist
    $\mathcal{L}(v_a) \leftarrow \mathcal{L}(v_a) \cup \{C\}$
    $\mathcal{C}_{\mathcal{F}} \leftarrow \mathcal{C}_{\mathcal{F}} \cup \{x_{v_a:C} \geq \alpha\}$
end for
for each abstract role assertion $\langle (a_1, a_2){:}R \geq \alpha \rangle \in \mathcal{O}$ **do**
    create nodes v_{a_1}, v_{a_2} in $\mathcal{F}$ if they do not exist
    create an edge $\langle v_{a_1}, v_{a_2} \rangle$ in $\mathcal{F}$ if it does not exist
    $\mathcal{L}(\langle v_{a_1}, v_{a_2} \rangle) \leftarrow \mathcal{L}(\langle v_{a_1}, v_{a_2} \rangle) \cup \{R\}$
    $\mathcal{C}_{\mathcal{F}} \leftarrow \mathcal{C}_{\mathcal{F}} \cup \{x_{(v_{a_1}, v_{a_2}):R} \geq \alpha\}$
end for
for each concrete role assertion $\langle (a, \mathrm{v}){:}T \geq \alpha \rangle \in \mathcal{O}$ **do**
    create nodes v_a, w_{v} in $\mathcal{F}$ if they do not exist
    create an edge $\langle v_a, w_{\mathrm{v}} \rangle$ in $\mathcal{F}$ if it does not exist
    $\mathcal{L}(\langle v_a, w_{\mathrm{v}} \rangle) \leftarrow \mathcal{L}(\langle v_a, w_{\mathrm{v}} \rangle) \cup \{T\}$
    $\mathcal{C}_{\mathcal{F}} \leftarrow \mathcal{C}_{\mathcal{F}} \cup \{x_{(v_a, w_{\mathrm{v}}):T} \geq \alpha\}$
end for
while some tableaux rule is applicable to some node in $\mathcal{F}$ **do**
    apply one of the rules in Table B.3 to a node v
    mark the applied rule as not applicable to node v
end while
if $\mathcal{C}_{\mathcal{F}}$ has a solution **then**
    solve the optimization problem minimizing x
    return x
else
    return 1 // $\mathcal{O}$ is inconsistent
end if

The algorithm uses a *completion-forest* $\mathcal{F}$, which is a collection of trees whose distinguished roots (the individuals of the ontology) are arbitrarily connected by edges. The forest has associated a set $\mathcal{C}_{\mathcal{F}}$ of $\mathcal{P}$ constraints on the variables

[14] Please note that in fuzzy DLs, it is common to consider data properties as crisp and functional, as discussed in [12], but we are not imposing this assumption here.

Table B.3
Rules of the tableaux algorithm combined with an optimization problem.

Rule	Preconditions	Actions
$(\bot)$	$\bot \in \mathcal{L}(v)$	$\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{v:\bot} = 0\}$
$(\top)$	$\top \in \mathcal{L}(v)$	$\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{v:\top} = 1\}$
$(\sqcap)$	$C_1 \sqcap C_2 \in \mathcal{L}(v)$	$\mathcal{L}(v) = \mathcal{L}(v) \cup \{C_1, C_2\}$ $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{v:C_1} \otimes x_{v:C_2} = x_{v:C_1 \sqcap C_2}\}$
$(\sqcup)$	$C_1 \sqcup C_2 \in \mathcal{L}(v)$	$\mathcal{L}(v) = \mathcal{L}(v) \cup \{C_1, C_2\}$ $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{v:C_1} \oplus x_{v:C_2} = x_{v:C_1 \sqcup C_2}\}$
$(\rightarrow)$	$C_1 \rightarrow C_2 \in \mathcal{L}(v)$	$\mathcal{L}(v) = \mathcal{L}(v) \cup \{\neg_{Ł} C_1, C_2\}$ $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{v:C_1} \Rightarrow x_{v:C_2} = x_{v:C_1 \rightarrow C_2}\}$
$(\forall)$	$\forall R.C \in \mathcal{L}(v_1)$ $R \in \mathcal{L}(\langle v_1, v_2 \rangle)$	$\mathcal{L}(v_2) = \mathcal{L}(v_2) \cup \{C\}$ $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{z = x_{v_1:\forall R.C} \otimes x_{(v_1,v_2):R},$ $x_{v_2:C} \geq z, z \in [0,1]\}$
$(\forall_{\mathbf{d}})$	$\forall T.\mathbf{d} \in \mathcal{L}(v)$ $T \in \mathcal{L}(\langle v, w \rangle)$	$\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{z = x_{v:\forall T.\mathbf{d}} \otimes x_{(v,w):T},$ $\mathbf{d}(x_w) = x_{w:\mathbf{d}}, x_{w:\mathbf{d}} \geq z, x_{w:\mathbf{d}} \in [0,1], z \in [0,1]\}$
$(\exists)$	$\exists R.C \in \mathcal{L}(v_1)$	create a new node v_2 $\mathcal{L}(v_2) = \mathcal{L}(v_2) \cup \{C\}$ $\mathcal{L}(\langle v_1, v_2 \rangle) = \mathcal{L}(\langle v_1, v_2 \rangle) \cup \{R\}$ $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{(v_1,v_2):R} \otimes x_{v_2:C} = z,$ $z \geq x_{v_1:\exists R.C}, z \in [0,1]\}$
$(\exists_{\mathbf{d}})$	$\exists T.\mathbf{d} \in \mathcal{L}(v)$	create a new node w $\mathcal{L}(\langle v, w \rangle) = \mathcal{L}(\langle v, w \rangle) \cup \{T\}$ $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{(v,w):T} \otimes x_{w:\mathbf{d}} = z,$ $x_{w:\mathbf{d}} = \mathbf{d}(x_w), z \geq x_{v:\exists T.\mathbf{d}}, x_{w:\mathbf{d}} \in [0,1], z \in [0,1]\}$
$(\neg_1)$	$\neg C \in \mathcal{L}(v)$ $C \in \{\bot, \top, A, \neg C,$ $C_1 \sqcap C_2, C_1 \sqcup C_2\}$	$\mathcal{L}(v) = \mathcal{L}(v) \cup \{C\}$ $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{v:\neg C} = \ominus x_{v:C}\}$
$(\neg_2)$	$\neg(C_1 \rightarrow C_2) \in \mathcal{L}(v)$	$\mathcal{L}(v) = \mathcal{L}(v) \cup \{C_1, \neg_{Ł} C_2\}$ $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{v:\neg(C_1 \rightarrow C_2)} = \ominus x_{v:C_1 \rightarrow C_2}\}$ $\cup \{x_{v:C_1} \Rightarrow x_{v:C_2} = x_{v:C_1 \rightarrow C_2}\}$
$(\neg_3)$	$\neg \exists R.C \in \mathcal{L}(v)$ $R \in \mathcal{L}(\langle v_1, v_2 \rangle)$	$\mathcal{L}(v_2) = \mathcal{L}(v_2) \cup \{C\}$ $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{v:\neg \exists R.C} = \ominus x_{v:\exists R.C},$ $x_{(v_1,v_2):R} \otimes x_{v_2:C} = z, z \leq x_{v_1:\exists R.C}, z \in [0,1]\}$
$(\neg_4)$	$\neg \exists T.\mathbf{d} \in \mathcal{L}(v)$ $T \in \mathcal{L}(\langle v, w \rangle)$	$\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{v:\neg \exists T.\mathbf{d}} = \ominus x_{v:\exists T.\mathbf{d}}, x_{w:\mathbf{d}} = \mathbf{d}(x_w),$ $x_{(v,w):T} \otimes x_{w:\mathbf{d}} = z, z \leq x_{v:\exists T.\mathbf{d}}, x_{w:\mathbf{d}} \in [0,1], z \in [0,1]\}$
$(\neg_5)$	$\neg \forall R.C \in \mathcal{L}(v)$	create a new node v_2 $\mathcal{L}(v_2) = \mathcal{L}(v_2) \cup \{C\}$ $\mathcal{L}(\langle v_1, v_2 \rangle) = \mathcal{L}(\langle v_1, v_2 \rangle) \cup \{R\}$ $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{v_1:\neg \forall R.C} = \ominus x_{v_1:\forall R.C}, z \in [0,1],$ $x_{(v_1,v_2):R} \Rightarrow x_{v_2:C} = z, z \leq x_{v_1:\forall R.C}\}$
$(\neg_6)$	$\neg \forall T.\mathbf{d} \in \mathcal{L}(v)$	create a new node w $\mathcal{L}(\langle v, w \rangle) = \mathcal{L}(\langle v, w \rangle) \cup \{T\}$ $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{v:\neg \forall T.\mathbf{d}} = \ominus x_{v:\forall T.\mathbf{d}}, x_{w:\mathbf{d}} = \mathbf{d}(x_w),$ $x_{(v,w):T} \Rightarrow x_{w:\mathbf{d}} = z, z \leq x_{v:\forall T.\mathbf{d}}, x_{w:\mathbf{d}} \in [0,1], z \in [0,1]\}$
$(\sqsubseteq)$	$\langle C_1 \sqsubseteq C_2 \geq \alpha \rangle \in \mathcal{T}$ Rule not applied to v	add $C_1 \rightarrow C_2$ to $\mathcal{L}(v)$ $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{v:C_1} \Rightarrow x_{v:C_2} \geq \alpha\}$

occurring in the node labels and edge labels. Those nodes representing abstract individuals will be denoted as v, and those representing concrete individuals will be denoted as w. Each node v is labeled with a set $\mathcal{L}(v)$ of concepts C. If $C \in \mathcal{L}(v)$ then we consider a variable $x_{v:C}$. The intuition here is that v is an instance of C to degree equal or greater than of the value of the variable $x_{v:C}$ in a minimal solution. Essentially, $x_{v:C}$ will hold the degree of truth of $v{:}C$. Each edge $\langle v_1, v_2 \rangle$ is labeled with a set $\mathcal{L}(\langle v_1, v_2 \rangle)$ of abstract roles R. If $R \in \mathcal{L}(\langle v_1, v_2 \rangle)$ then we consider a variable $x_{(v_1,v_2):R}$ holding the degree of truth of $(v_1, v_2){:}R$. Similarly, each edge $\langle v, w \rangle$ is labeled with a set $\mathcal{L}(\langle v, w \rangle)$ of concrete roles T and if $T \in \mathcal{L}(\langle v, w \rangle)$ then we consider a variable $x_{(v,w):T}$. We will assume that there is a bijection between assertions $\mathfrak{a}$ and variables $x_{\mathfrak{a}}$. For each node w representing a concrete individual, we will consider a variable x_w holding the value of the concrete individual (recall that we restrict to numerical datatypes). Note that we do not need a notion of *blocking* as $\mathcal{T}$ is acyclic.

Remark 17. It is worth to note a mistake in the reasoning rules in [12]. There, both the rules $(\rightarrow)$ and $(\neg\rightarrow)$ add C_1 and C_2 to $\mathcal{L}(v)$. Actually, $(\rightarrow)$ should add $\neg C_1$ instead of C_1 because fuzzy implications are antitone in the first argument [5]. Similarly, $(\neg\rightarrow)$ should add $\neg C_2$ instead of C_2.

Appendix C. Some examples of encodings of the restrictions

This appendix shows how to represent the restrictions that appear in our reasoning algorithms in some popular cases: the main fuzzy logics, and the fuzzy membership functions in Fig. 1. Our objective is to offer an easy to understand encoding to make the paper self-contained (for more details, see [56]). The reader interested in more optimized representations is referred to [17].

Table C.4 shows how to represent the restrictions $x_1 \otimes x_2 = z$, $x_1 \Rightarrow x_2 = z$, $x_1 \oplus x_2 = z$, and $\ominus x = z$ in Łukasiewicz, Gödel, Product, and Zadeh logics. Observe that $\bigotimes_i x_i = z$ can be represented using associativity and restrictions of the form $x_1 \otimes x_2 = z$. In the table, y denotes a new variable and $\epsilon > 0$.

Table C.4
Encoding of some popular fuzzy logic operators.

Restriction	Encoding
$x_1 \otimes_G x_2 = z$	$\{z \le x_1, z \le x_2, x_1 \le z + y, x_2 \le z + (1-y), y \in \{0,1\}\}$
$x_1 \otimes_Ł x_2 = z$	$\{x_1 + x_2 - 1 \le z, x_1 + x_2 - 1 \ge z - y, z \le 1 - y, y \in \{0,1\}\}$
$x_1 \otimes_\Pi x_2 = z$	$\{x_1 \cdot x_2 = z\}$
$x_1 \Rightarrow_G x_2 = z$	$\{2y + x_1 \ge x_2 + \epsilon, x_1 \le x_2 + (1-y), y + x_2 \ge z, x_2 \le z + y,$ $z \ge y, y \in \{0,1\}\}$
$x_1 \Rightarrow_Ł x_2 = z$	$\{1 - x_1 + x_2 \le z + y, y \le z, 1 - x_1 + x_2 \ge z, y \in \{0,1\}\}$
$x_1 \Rightarrow_\Pi x_2 = z$	$\{2y + x_1 \ge x_2 + \epsilon, x_1 \le x_2 + (1-y), 2y + x_1 \cdot z \ge x_2,$ $x_1 \cdot z \le x_2 + 2y, z \ge y, y \in \{0,1\}\}$
$x_1 \Rightarrow_Z x_2 = z$	$\{2y + x_1 \ge x_2 + \epsilon, z = y, x_1 \le x_2 + (1-y), y \in \{0,1\}\}$
$x_1 \Rightarrow_{KD} x_2 = z$	$\{z \ge 1 - x_1, z \ge x_2, (1-x_1) + y \ge z, x_2 + (1-y) \ge z, y \in \{0,1\}\}$
$x_1 \oplus_G x_2 = z$	$\{z \ge x_1, z \ge x_2, x_1 + y \ge z, x_2 + (1-y) \ge z, y \in \{0,1\}\}$
$x_1 \oplus_Ł x_2 = z$	$\{x_1 + x_2 \le z + y, y \le z, x_1 + x_2 \ge z, y \in \{0,1\}\}$
$x_1 \oplus_\Pi x_2 = z$	$\{x_1 + x_2 - x_1 \cdot x_2 = z\}$
$\ominus_G x = \ominus_\Pi x = z$	$\{z \le 1 - x, x + z \ge \epsilon, z \in \{0,1\}\}$
$\ominus_Ł x = z$	$\{1 - x = z\}$
$z > 0$	$\{z \ge \epsilon\}$

Table C.5
Encoding of some popular fuzzy membership functions.

Restriction	Encoding
$\mathtt{trap}_{q_1,q_2,q_3,q_4}(x) = z$	$\{x + (k_1 - q_1)y_2 \ge k_1, x + (k_1 - q_2)y_3 \ge k_1, x + (k_1 - q_3)y_4 \ge k_1,$ $x + (k_1 - q_4)y_5 \ge k_1, x + (k_2 - q_1)y_1 \le k_2, x + (k_2 - q_2)y_2 \le k_2,$ $x + (k_2 - q_3)y_3 \le k_2, x + (k_2 - q_4)y_4 \le k_2, z \le 1 - y_1 - y_5, z \ge y_3,$ $x + (q_1 - q_2)z + (k_2 - q_1)y2 \le k_2,$ $x + (q_1 - q_2)z + (k_1 - q_2)y2 \ge k_1 + q_1 - q_2,$ $x + (q_4 - q_3)z + (k_2 - q_3)y4 \le k_2 + q_4 - q_3,$ $x + (q_4 - q_3)z + (k_1 - q_4)y4 \ge k_1,$ $y_1 + y_2 + y_3 + y_4 + y_5 = 1, y_i \in \{0,1\}\}$
$\mathtt{tri}_{q_1,q_2,q_3}(x) = z$	$\{x + (k_1 - q_1)y_2 \ge k_1, x + (k_1 - q_2)y_3 \ge k_1, x + (k_1 - q_3)y_4 \ge k_1,$ $x + (k_2 - q_1)y_1 \le k_2, x + (k_2 - q_2)y_2 \le k_2, x + (k_2 - q_3)y_3 \le k_2,$ $z \le 1 - y_1 - y_4, x + (q_1 - q_2)z + (k_2 - q_1)y2 \le k_2,$ $x + (q_1 - q_2)z + (k_1 - q_2)y2 \ge k_1 + q_1 - q_2,$ $x + (q_3 - q_2)z + (k_2 - q_2)y3 \le k_2 + q_3 - q_2,$ $x + (q_3 - q_2)z + (k_1 - q_3)y3 \ge k_1, y_1 + y_2 + y_3 + y_4 = 1, y_i \in \{0,1\}\}$
$\mathtt{left}_{q_1,q_2}(x) = z$	$\{x + (k_1 - q_1)y_2 \ge k_1, x + (k_1 - q_2)y_3 \ge k_1, x + (k_2 - q_1)y_1 \le k_2,$ $x + (k_2 - q_2)y_2 \le k_2, z \le 1 - y_3, z \ge y_1,$ $x + (q_2 - q_1)z + (k_2 - q_1)y2 \le k_2 + q_2 - q_1,$ $x + (q_2 - q_1)z + (k_1 - q_2)y2 \ge k_1,$ $y_1 + y_2 + y_3 = 1, y_i \in \{0,1\}\}$
$\mathtt{crisp}_{q_1,q_2}(x) = z$	$\{x + (k_1 - q_1)y_2 \ge k_1, x + (k_1 - q_2 - \epsilon)y_3 \ge k_1,$ $x + (k_2 - q_1 + \epsilon)y_1 \le k_2, x + (k_2 - q_2)y_2 \le k_2,$ $z \le 1 - y_1 - y_3, z \ge y_2, y_1 + y_2 + y_3 = 1, y \in \{0,1\}\}$

(*continued on next page*)

Table C.5 (*continued*)

Restriction	Encoding
$\mathtt{lin}_{q_1,q_2}(x) = z$	$\{q_2x - (q_1 - k_1)z + (q_1 - k_1)y \geq q_2k_1,$ $q_2x - (q_1 - k_1)z - q_2(k_2 - k_1)y \leq q_2k_1,$ $(1 - q_2)x - (k_2 - q_1)z - (1 - q_2)(k_2 - k_1)y \geq q_1 - k_2 - k_1q_2 + k_1,$ $(1 - q_2)x - (k_2 - q_1)z - (q_1 - k_2)y \leq k_2 - q_2k_2,$ $x + (q_1 - k_2)y \leq q_1, x + (k_1 - q_1)y \geq k_1, y \in \{0, 1\}\}$
$\mathtt{right}_{q_1,q_2}(x) = z$	$\{x + (k_1 - q_1)y_2 \geq k_1, x + (k_1 - q_2)y_3 \geq k_1, x + (k_2 - q_1)y_1 \leq k_2$ $x + (k_2 - q_2)y_2 \leq k_2, z \leq 1 - y_1, z \geq y_3,$ $x + (q_1 - q_2)z + (k_2 - q_1)y2 \leq k_2,$ $x + (q_1 - q_2)z + (k_1 - q_2)y2 \geq k_1 + q_1 - q_2,$ $y_1 + y_2 + y_3 = 1, y_i \in \{0, 1\}\}$

Finally, Table C.5 shows how to represent e.g., $\mathbf{d}(x) = z$ in some cases, where y_i denote new variables, $[k_1, k_2]$ is the range of $\mathbf{d}$,[15] and $\epsilon > 0$. The cases $\mathbf{d}(x) \geq z$ and $\mathbf{d}(x) \leq z$ can be worked out similarly.

The encoding for $\mathbf{t}(x) \geq z$ and $\mathbf{t}(x) \leq z$ can be derived immediately from those for $\mathbf{d}(x) \geq z$ and $\mathbf{d}(x) \leq z$, by replacing $[k_1, k_2]$ with $[0, 1]$.

Appendix D. Updated reasoning rules

In this appendix, we detail how to update the rules $(\exists_{\mathbf{d}})$, $(\forall_{\mathbf{d}})$, $(\neg_4)$, and $(\neg_6)$ in Table B.3 to take into account Eq. (3). The new rules use the binary condition $y = (x_{w:\mathbf{d}} \in [z_1, z_2])$, which can be encoded as shown in Eq. (25). The rules are illustrated in Table D.6.

Note that some of the new rules introduce restrictions of the form $(x_1 \otimes x_2 \otimes y) = z$ or $x_1 \Rightarrow (x_2 \otimes y) = z$. By associativity, we can add a new constraint of the form $x_3 = x_2 \otimes y$ and transform the constraints into the forms $(x_1 \otimes x_3) = z$ or $x_1 \Rightarrow x_3 = z$, for which an encoding has been given in Appendix C.

Table D.6
Updated rules to take into account Eq. (3).

Rule	Preconditions	Actions
$(\exists_{\mathbf{d}})$	$\exists T.\mathbf{d} \in \mathcal{L}(v)$	create a new node w $\mathcal{L}(\langle v, w\rangle) = \mathcal{L}(\langle v, w\rangle) \cup \{T\}$ $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{(x_{(v,w):T} \otimes x_{w:\mathbf{d}} \otimes y) = z,$ $y = (x_{w:\mathbf{d}} \in [z_1, z_2]),$ $z_1 = \mathbf{d}_L(x_w),$ $z_2 = \mathbf{d}_U(x_w),$ $z_1 \leq z_2,$ $z \geq x_{v:\exists T.\mathbf{d}},$ $x_{w:\mathbf{d}}, z, z_1, z_2 \in [0, 1],$ $y \in \{0, 1\}\}$
$(\forall_{\mathbf{d}})$	$\forall T.\mathbf{d} \in \mathcal{L}(v)$ $T \in \mathcal{L}(\langle v, w\rangle)$	$\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{z = x_{v:\forall T.\mathbf{d}} \otimes x_{(v,w):T},$ $x_{w:\mathbf{d}} \otimes y \geq z,$ $y = (x_{w:\mathbf{d}} \in [z_1, z_2]),$ $z_1 = \mathbf{d}_L(x_w),$ $z_2 = \mathbf{d}_U(x_w),$ $z_1 \leq z_2,$ $x_{w:\mathbf{d}}, z, z_1, z_2 \in [0, 1],$ $y \in \{0, 1\}\}$
$(\neg_4)$	$\neg\exists T.\mathbf{d} \in \mathcal{L}(v)$ $T \in \mathcal{L}(\langle v, w\rangle)$	$\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{v:\neg\exists T.\mathbf{d}} =$ $\ominus x_{v:\exists T.\mathbf{d}}, x_{w:\mathbf{d}} = \mathbf{d}(x_w),$ $(x_{(v,w):T} \otimes x_{w:\mathbf{d}} \otimes y) = z,$ $y = (x_{w:\mathbf{d}} \in [z_1, z_2]),$ $z_1 = \mathbf{d}_L(x_w),$ $z_2 = \mathbf{d}_U(x_w),$ $z_1 \leq z_2,$ $z \leq x_{v:\exists T.\mathbf{d}},$ $x_{w:\mathbf{d}}, z, z_1, z_2 \in [0, 1,$ $y \in \{0, 1\}\}$

[15] The range of the fuzzy datatypes is usually $(-\infty, \infty)$ although in more expressive fuzzy DLs it is possible to add more specific ranges. In practical applications, $-\infty$ and ∞ are the minimal and the maximal representable number, respectively.

Table D.6 (*continued*)

Rule	Preconditions	Actions
$(\neg_6)$	$\neg\forall T.\mathbf{d} \in \mathcal{L}(v)$	create a new node w $\mathcal{L}(\langle v, w\rangle) = \mathcal{L}(\langle v, w\rangle) \cup \{T\}$ $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{v:\neg\forall T.\mathbf{d}} = \ominus x_{v:\forall T.\mathbf{d}},$ $(x_{(v,w):T} \Rightarrow (x_{w:\mathbf{d}} \otimes y)) = z,$ $y = (x_{w:\mathbf{d}} \in [z_1, z_2]),$ $z_1 = \mathbf{d}_L(x_w),$ $z_2 = \mathbf{d}_U(x_w),$ $z_1 \leq z_2,$ $z \leq x_{v:\forall T.\mathbf{d}},$ $x_{w:\mathbf{d}}, z, z_1, z_2 \in [0, 1],$ $y \in \{0, 1\}\}$

References

[1] F. Ali, E.K. Kim, Y. Kim, Type-2 fuzzy ontology-based opinion mining and information extraction: a proposal to automate the hotel reservation system, Appl. Intell. 3 (2015) 481–500.
[2] F. Ali, E.K. Kim, Y. Kim, Type-2 fuzzy ontology-based semantic knowledge for collision avoidance of autonomous underwater vehicles, Inf. Sci. 295 (3) (2015) 441–464.
[3] F. Baader, S. Borgwardt, R. Peñaloza, On the decidability status of fuzzy $\mathcal{ALC}$ with general concept inclusions, J. Philos. Log. 44 (2015).
[4] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P.F. Patel-Schneider (Eds.), The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge University Press, 2003.
[5] M. Baczyński, B. Jayaram, Fuzzy Implications, Studies in Fuzziness and Soft Computing, vol. 231, Springer, 2008.
[6] A. Bahri, R. Bouziz, F. Gargouri, A generalized fuzzy extension of $\mathcal{EL}$++, in: Proceedings of the Annual Meeting of the North American Fuzzy Information Processing Society, NAFIPS 2010, IEEE Press, 2010, pp. 1–6.
[7] C. Bobed, R. Yus, F. Bobillo, E. Mena, Semantic reasoning on mobile devices: do androids dream of efficient reasoners?, J. Web Semant. 35 (4) (2015) 167–183.
[8] F. Bobillo, The role of crisp elements in fuzzy ontologies: the case of fuzzy OWL 2 EL, IEEE Trans. Fuzzy Syst. 24 (5) (2016) 1193–1209.
[9] F. Bobillo, M. Cerami, F. Esteva, À. García-Cerdaña, R. Peñaloza, U. Straccia, Fuzzy description logics, in: Handbook of Mathematical Fuzzy Logic, vol. 3, in: Studies in Logic, Mathematical Logic and Foundations, vol. 58, College Publications, 2015, pp. 1105–1181, chapter 16.
[10] F. Bobillo, M. Delgado, J. Gómez-Romero, Crisp representations and reasoning for fuzzy ontologies, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 17 (4) (2009) 501–530.
[11] F. Bobillo, M. Delgado, J. Gómez-Romero, DeLorean: a reasoner for fuzzy OWL 2, Expert Syst. Appl. 39 (1) (2012) 258–272.
[12] F. Bobillo, U. Straccia, Fuzzy description logics with general t-norms and datatypes, Fuzzy Sets Syst. 160 (23) (2009) 3382–3402.
[13] F. Bobillo, U. Straccia, Fuzzy descriptions logics with fuzzy truth values, in: Proceedings of the 13th World Congress of the International Fuzzy Systems Association and 6th Conference of the European Society for Fuzzy Logic and Technology, IFSA-EUSFLAT 2009, 2009, pp. 189–194.
[14] F. Bobillo, U. Straccia, Fuzzy ontology representation using OWL 2, Int. J. Approx. Reason. 52 (7) (2011) 1073–1094.
[15] F. Bobillo, U. Straccia, Aggregation operators for fuzzy ontologies, Appl. Soft Comput. 13 (9) (2013) 3816–3830.
[16] F. Bobillo, U. Straccia, On partitioning-based optimisations in expressive fuzzy description logics, in: Proceedings of the 24th IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2015, IEEE Press, 2015.
[17] F. Bobillo, U. Straccia, Reducing the size of the optimization problems in fuzzy ontology reasoning, in: Proceedings of the 11th International Workshop on Uncertainty Reasoning for the Semantic Web, URSW 2015, 2015, pp. 54–59.
[18] F. Bobillo, U. Straccia, The fuzzy ontology reasoner fuzzyDL, Knowl.-Based Syst. 95 (2016) 12–34.
[19] F. Bobillo, U. Straccia, Optimising fuzzy description logic reasoners with general concept inclusions absorption, Fuzzy Sets Syst. 292 (2016) 98–129.
[20] S. Borgwardt, M. Cerami, R. Peñaloza, The complexity of subsumption in fuzzy $\mathcal{EL}$, in: Proceedings of the 24th International Joint Conference on Artificial Intelligence, IJCAI 2015, 2015, pp. 2812–2818.
[21] S. Borgwardt, F. Distel, R. Peñaloza, The limits of decidability in fuzzy description logics with general concept inclusions, Artif. Intell. 218 (2015) 23–55.
[22] A.C. Bukhari, Y. Kim, Integration of a secure type-2 fuzzy ontology with a multi-agent platform: a proposal to automate the personalized flight ticket booking domain, Inf. Sci. 198 (2012) 24–47.
[23] S. Calegari, D. Ciucci, Fuzzy ontology and fuzzy-OWL in the KAON project, in: Proceedings of the 16th IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2007, 2007, pp. 1415–1420.
[24] C. Carlsson, M. Brunelli, J. Mezei, Decision making with a fuzzy ontology, Soft Comput. 16 (7) (2012) 1143–1152.
[25] M. Cerami, U. Straccia, On the undecidability of fuzzy description logics with GCIs with Łukasiewicz t-norm, Inf. Sci. 227 (1) (2013) 1–21.
[26] B. Cuenca-Grau, I. Horrocks, B. Motik, B. Parsia, P.F. Patel-Schneider, U. Sattler, OWL 2: the next step for OWL, J. Web Semant. 6 (4) (2008) 309–322.
[27] N. Devadoss, S. Ramakrishnan, Knowledge representation using type-2 fuzzy rough ontologies in ontology web language, Fuzzy Inf. Eng. 7 (1) (2015) 73–99.
[28] N. Díaz-Rodríguez, M. Pegalajar-Cuéllar, J. Lilius, M. Delgado, A fuzzy ontology for semantic modelling and recognition of human behaviour, Knowl.-Based Syst. 66 (2014) 46–60.
[29] M. Eich, R. Hartanto, S. Kasperski, S. Natarajan, J. Wollenberg, Towards coordinated multirobot missions for lunar sample collection in an unknown environment, J. Field Robot. 31 (1) (2014) 35–74.
[30] P. Hájek, Making fuzzy description logic more general, Fuzzy Sets Syst. 154 (1) (2005) 1–15.
[31] P. Hájek, Metamathematics of Fuzzy Logic, Trends Log., vol. 4, Kluwer, 1998.
[32] P. Hájek, On witnessed models in fuzzy logic, Math. Log. Q. 53 (1) (2007) 66–77.
[33] I. Horrocks, O. Kutz, U. Sattler, The even more irresistible $\mathcal{SROIQ}$, in: Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning, KR 2006, AAAI Press, 2006, pp. 57–67.
[34] H. Huang, C. Lee, M. Wang, H. Kao, IT2FS-based ontology with soft-computing mechanism for malware behavior analysis, Soft Comput. 18 (2) (2014) 267–284.
[35] L. Huo, J. Ouyang, W. Sun, D. Liu, Interval-valued fuzzy description logic $\mathcal{IFALCN}$, J. Comput. Inf. Syst. 8 (7) (2012) 3027–3034.
[36] G.J. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice-Hall, Inc., 1995.
[37] C. Lee, M. Su, M. Wu, H. Hagras, A type-2 fml-based meeting scheduling support system, in: On the Power of Fuzzy Markup Language, in: Studies in Fuzziness and Soft Computing, vol. 296, Springer, 2013, pp. 169–187.

[38] C. Lee, M. Wang, H. Hagras, A type-2 fuzzy ontology and its application to personal diabetic-diet recommendation, IEEE Trans. Fuzzy Syst. 18 (2) (2010) 374–395.

[39] C. Lee, M. Wang, M. Wu, O. Teytaud, S. Yen, T2FS-based adaptive linguistic assessment system for semantic analysis and human performance evaluation on game of go, IEEE Trans. Fuzzy Syst. 23 (2) (2015) 400–420.

[40] C.-S. Lee, M.H. Wang, G. Acampora, C.Y. Hsu, H. Hagras, Diet assessment based on type-2 fuzzy ontology and fuzzy markup language, Int. J. Intell. Syst. 25 (12) (2010) 1187–1216.

[41] R. Li, K. Wen, X. Gu, Y. Li, X. Sun, B. Li, Type-2 fuzzy description logic, Front. Comput. Sci. China 5 (2) (2011) 205–215.

[42] C. Lutz, Reasoning with concrete domains, in: Proceedings of the 16th International Joint Conference on Artificial Intelligence, IJCAI 1999, Morgan Kaufmann Publishers Inc., 1999, pp. 90–95.

[43] C. Martínez-Cruz, A. van der Heide, D. Sánchez, G. Triviño, An approximation to the computational theory of perceptions using ontologies, Expert Syst. Appl. 39 (10) (2012) 9494–9503.

[44] J.M. Mendel, Type-2 fuzzy sets and systems: an overview, IEEE Comput. Intell. Mag. 2 (2007) 20–29.

[45] J.M. Mendel, R.I. John, Type-2 fuzzy sets made simple, IEEE Trans. Fuzzy Syst. 10 (2) (2002) 117–127.

[46] J.M. Mendel, R.I. John, F. Liu, Interval type-2 fuzzy logic systems made simple, IEEE Trans. Fuzzy Syst. 14 (6) (2006) 808–821.

[47] J. Mezei, R. Wikström, OWAD operators in type-2 fuzzy ontologies, in: Proceedings of the Joint IFSA World Congress and NAFIPS Annual Meeting, IFSA-NAFIPS 2013, IEEE Press, 2013, pp. 848–853.

[48] J. Mezei, R. Wikström, C. Carlsson, Aggregating linguistic expert knowledge in type-2 fuzzy ontologies, Appl. Soft Comput. 35 (2015) 911–920.

[49] P.S. Mostert, A.L. Shields, On the structure of semigroups on a compact manifold with boundary, Ann. Math. 65 (1) (1957) 117–143.

[50] J.A. Rodger, A fuzzy linguistic ontology payoff method for aerospace real options valuation, Expert Syst. Appl. 40 (8) (2013) 2828–2840.

[51] H.M. Salkin, K. Mathur, Foundations of Integer Programming, North-Holland, 1989.

[52] S. Staab, R. Studer (Eds.), Handbook on Ontologies. International Handbooks on Information Systems, Springer, 2004.

[53] G. Stoilos, G. Stamou, A framework for reasoning with expressive continuous fuzzy description logics, in: Proceedings of the 22nd International Workshop on Description Logics, DL 2009, 2009.

[54] U. Straccia, Reasoning within fuzzy description logics, J. Artif. Intell. Res. 14 (2001) 137–166.

[55] U. Straccia, Description logics with fuzzy concrete domains, in: Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence, UAI 2005, 2005, pp. 559–567.

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

[57] K. Todorov, C. Hudelot, A. Popescu, P. Geibel, Fuzzy ontology alignment using background knowledge, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 22 (1) (2014) 75–112.

[58] R. Wikström, Fuzzy Ontology for Knowledge Mobilisation and Decision Support, PhD thesis, Åbo Akademi, Turku, Finland, 2014.

[59] R. Wikström, J. Mezei, Intrusion detection with type-2 fuzzy ontologies and similarity measures, in: Intelligent Methods for Cyber Warfare, in: Studies in Computational Intelligence, vol. 563, Springer, 2015, pp. 151–172.

[60] L.A. Zadeh, Fuzzy sets, Inf. Control 8 (3) (1965) 338–353.

[61] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning I, Inf. Sci. 8 (3) (1975) 199–249.