IEEE TRANSACTIONS ON

FUZZY SYSTEMS

A PUBLICATION OF THE IEEE COMPUTATIONAL INTELLIGENCE SOCIETY
http://ieeecis-tfs.nctu.edu.tw/
www.ieee-cis.org/pubs/tfs

IEEE Computational Intelligence Society
MIMICKING NATURE FOR PROBLEM SOLVING

AUGUST 2013 VOLUME 21 NUMBER 4 IEFSEV (ISSN 1063-6706)

REGULAR PAPERS

SHORT PAPERS

IEEE

[39] S. Shirdhonkar and D. Jacobs, "Approximate earth mover's distance in linear time," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[40] H. Timm, C. Borgelt, C. Doring, and R. Kruse, "An extension to possibilistic fuzzy cluster analysis," *Fuzzy Sets Syst.*, vol. 147, no. 1, pp. 3–16, Oct. 2004.
[41] M. de Cáceres, F. Oliva, and X. Font, "On relational possibilistic clustering," *Pattern Recognit.*, vol. 39, no. 11, pp. 2010–2024, 2006.
[42] M.-S. Yang and C.-Y. Lai, "A robust automatic merging possibilistic clustering method," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 1, pp. 26–41, Feb. 2011.

Parallel Algorithms for Fuzzy Ontology Reasoning

Fernando Bobillo, Miguel Delgado, and José César Sánchez–Sánchez

***Abstract*—The need to deal with imprecise and vague information in ontologies is rising in importance, as required by several real-world application domains. As a consequence, there is a growing interest in fuzzy ontologies, which combine ontologies and fuzzy logic theory. In fuzzy ontologies, some reasoning tasks usually become harder to solve, such as the concept subsumption problem and the computation of the Best Degree Bound (BDB) of an axiom. In fact, the current existing algorithms to solve these problems usually require performing some simpler tests several times. In this paper, we present a parallelization of these algorithms, implemented in the `DeLorean` reasoner, and discuss the encouraging results of an empirical evaluation.**

***Index Terms*—Fuzzy description logics (DLs), fuzzy ontologies, logic for the semantic web, parallel algorithms.**

I. Introduction

In the last decade, ontologies have become the state-of-the-art knowledge representation formalism. An ontology is as an explicit and formal specification of a shared conceptualization, which means that ontologies represent the concepts and the relationships in a domain promoting interrelation with other models and automatic processing. The current standard ontology language is the Web Ontology Language (OWL 2) [1].

Description logics (DLs for short) [2] are a family of logics for representing structured knowledge. Each logic is denoted by using a string of capital letters which identify the constructors of the logic and therefore its complexity. DLs have proved to be very useful as ontology languages. For instance, OWL 2 is equivalent to $\mathcal{SROIQ}(\mathbf{D})$ [3]. The real power of DLs is the possibility of automatically discovering implicit knowledge by using several reasoning tasks. For instance, it is possible to check if the information contained in some ontology implies that one concept is more general than (i.e., subsumes) another concept, or if the ontology entails (i.e., implies) some axiom to hold.

In spite of 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. Fuzzy logic is a suitable formalism to handle these types of knowledge. In the setting of fuzzy logics, the convention that prescribes that a statement is either true or false is changed. A more refined range is used in such a way that every fuzzy statement has a degree of truth $\alpha \in [0, 1]$ associated with it [4].

Several fuzzy extensions of DLs can be found in the literature (see [5] for a survey) as the theoretical basis of fuzzy ontologies. In the fuzzy case, concepts and relations are fuzzy. Consequently, the axioms are not in general either true or false, but they may hold to some degree of truth. For instance, the fact that the individual John is tall with degree at least 0.75 can be represented by the axiom $\langle$*john: Tall* $\geq 0.75\rangle$. Fuzzy ontologies have proved to be useful in several applications, including information retrieval [6], image interpretation [7], [8], the Semantic Web and the Internet [9], [10], summarization [11], recommendation [12], or classification [13].

Whereas in classical ontologies an axiom is either entailed by an ontology or not, in fuzzy ontologies, the satisfiability of an axiom is a matter of degree, and hence, an important new reasoning task naturally appears: The computation of the maximum degree α such that some axiom τ holds with degree at least α given the facts in the fuzzy ontology. This value is called the *Best Degree Bound* (BDB).

Two algorithms to find the BDB problem can be found in the literature: a binary search over the possible degrees of truth [14], and a reduction to a minimization problem [15]. The first algorithm requires that the set of degrees of truth $\mathcal{N}$ is finite and known. This is the case in Zadeh fuzzy logic, but it introduces a limitation in other logics. The algorithm performs a binary search among these values by successive entailment tests (the number of tests is $\mathcal{O}(\log_2 |\mathcal{N}|)$). The second algorithm to compute the BDB of an axiom τ minimizes the $[0, 1]$-valued variable x subject to the inequality $\tau \leq x$. To this end, it combines DL tableau rules that generate some constraints to hold in order to respect the semantics of the DL constructors and an optimization problem that checks that all the constraints are satisfied. Regarding the concept subsumption problem, there are also two reasoning algorithms: a reduction to several satisfiability tests and, again, a combination of a tableau algorithm and optimization problems.

There also exists a family of algorithms to reason with a fuzzy ontology which compute an equivalent nonfuzzy ontology (called here *crisp representation algorithms*), which makes it possible to reason using currently existing reasoners for OWL 2 [16]–[18]. These algorithms are interesting because they make it possible to reuse current standard ontology languages; they also allow reasoning with very expressive fuzzy DLs (e.g., fuzzy $\mathcal{SROIQ}(\mathbf{D})$) for which no other reasoning algorithm is known yet.

In crisp representation algorithms, the equivalent nonfuzzy ontologies have a larger size than the original fuzzy ontologies, and hence reasoning with fuzzy ontologies (actually, reasoning with their equivalent nonfuzzy ontologies) is harder than reasoning with classical ontologies [19]. For this reason, the need to compute several entailment tests to solve the BDB problem, or several satisfiability tests to solve the concept subsumption problem, is especially problematic in practice.

In this paper, we present new algorithms for the BDB and the concept subsumption problems, based on a parallelization of the binary search algorithm. Some of these algorithms are specially designed to operate in the setting of crisp representation algorithms. Furthermore, we implement our algorithms as part of the fuzzy ontology reasoner `DeLorean` [19].

Manuscript received December 21, 2011; revised May 4, 2012; September 14, 2012; accepted November 13, 2012. Date of publication November 29, 2012; date of current version July 31, 2013.

F. Bobillo is with the Department of Computer Science and Systems Engineering, University of Zaragoza, Saragossa 12 50009, Spain (e-mail: fbobillo@unizar.es).

M. Delgado and J. C. Sánchez–Sánchez are with the Department of Computer Science and Artificial Intelligence, University of Granada, Granada 0 18010, Spain (e-mail: mdelgado@ugr.es; jcesarsanchez@gmail.com).

Digital Object Identifier 10.1109/TFUZZ.2012.2230266

TABLE I
SYNTAX AND SEMANTICS OF THE FUZZY DL $\mathcal{ALCH}$

Concept	Syntax (C)	Semantics ($C^{\mathcal{I}}(x)$)
(C1)	A	$A^{\mathcal{I}}(x)$
(C2)	$\top$	γ_p
(C3)	$\bot$	γ_0
(C4)	$C \sqcap D$	$C^{\mathcal{I}}(x) \otimes D^{\mathcal{I}}(x)$
(C5)	$C \sqcup D$	$C^{\mathcal{I}}(x) \oplus D^{\mathcal{I}}(x)$
(C6)	$\neg C$	$\ominus C^{\mathcal{I}}(x)$
(C7)	$\forall R.C$	$\inf_{y \in \Delta^{\mathcal{I}}}\{R^{\mathcal{I}}(x,y) \Rightarrow C^{\mathcal{I}}(y)\}$
(C8)	$\exists R.C$	$\sup_{y \in \Delta^{\mathcal{I}}}\{R^{\mathcal{I}}(x,y) \otimes C^{\mathcal{I}}(y)\}$

Axiom	Syntax (τ)	Semantics ($\mathcal{I}$ satisfies τ if ...)
(A1)	$\langle a{:}C \bowtie \alpha\rangle$	$C^{\mathcal{I}}(a^{\mathcal{I}}) \bowtie \alpha$
(A2)	$\langle (a,b){:}R \geq \alpha\rangle$	$R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) \geq \alpha$
(A3)	$\langle C_1 \sqsubseteq C_2 \geq \alpha\rangle$	$\inf_{x \in \Delta^{\mathcal{I}}}\{C_1^{\mathcal{I}}(x) \Rightarrow C_2^{\mathcal{I}}(x)\} \geq \alpha$
(A4)	$\langle R_1 \sqsubseteq R_2 \geq \alpha\rangle$	$\inf_{x,y \in \Delta^{\mathcal{I}}}\{R_1^{\mathcal{I}}(x,y) \Rightarrow R_2^{\mathcal{I}}(x,y)\} \geq \alpha$

The remainder of the paper is organized as follows. Section II provides some background on fuzzy DLs and their reasoning algorithms. Sections III and IV contain the first part of our contributions: new parallel reasoning algorithms for the BDB and the fuzzy concept subsumption problems, respectively. Section V discusses the results of an empirical evaluation. Finally, Section VI sets out some conclusions and ideas for future research.

II. Fuzzy Description Logics

In this section, we recall the definition of the fuzzy DL $\mathcal{ALCH}$ and the reasoning algorithm based on computing an equivalent nonfuzzy ontology. We consider a relatively simple fuzzy DL for the sake of clarity, but the ideas included in this paper work for more expressive fuzzy DLs such as $\mathcal{SROIQ}(\mathbf{D})$. In fuzzy ontologies, concepts denote fuzzy sets of individuals and roles denote fuzzy binary relations. Axioms are also extended to the fuzzy case, and some of them hold with some degree.

We will assume that the degrees of truth are taken from a *finite chain*, i.e., a totally ordered set $\mathcal{N} = \{0 = \gamma_0 < \gamma_1 < \cdots < \gamma_p = 1\}$, with $p \geq 1$; the minimum value γ_0 represents absolute falsity, and the maximum value γ_p represents absolute truth [20]–[22]. According to this, t-norms, t-conorms, and negation and implication functions will be operators defined over $\mathcal{N}$ instead of $[0,1]$.

A. Syntax

Fuzzy DLs assume three alphabets of symbols, for *fuzzy concepts*, *fuzzy roles,* and *individuals*. Fuzzy concepts can be atomic or complex, inductively formed from other concepts. The syntax of fuzzy concepts is shown in Table I, where C, D are (possibly complex) fuzzy concepts, A is an atomic fuzzy concept, and R is a fuzzy role.

A *fuzzy Knowledge Base* (KB) contains a finite number of axioms. The axioms that are allowed in fuzzy $\mathcal{ALCH}$ are shown in Table I, where a, b are individuals, $\bowtie \in \{\geq, \leq\}$, and $\alpha \in \mathcal{N}$ is a degree of truth. Axioms (A1) and (A2) are called fuzzy concept assertions and fuzzy role assertions, respectively. Note that axioms $\langle \tau > \gamma_i\rangle$ and $\langle \tau < \gamma_j\rangle$ $(i \neq p, j \neq 0)$ can be represented as $\langle \tau \geq \gamma_{i+1}\rangle$ and $\langle \tau \leq \gamma_{j-1}\rangle$, respectively.

Example 2.1: Let $\mathcal{N} = \{\gamma_0, \gamma_1, \gamma_2, \gamma_3, \gamma_4\}$ be a set of degrees of truth. A fuzzy KB can be defined as $\mathcal{K} = \{\langle(\mathsf{a},\mathsf{b}) : \mathsf{R} \geq \gamma_4\rangle, \langle \mathsf{a} : \forall \mathsf{R}.\mathsf{C} \geq \gamma_3\rangle\}$, where the first axiom states that a and b are related via role R, and the second axiom imposes a universal restriction to every individual related to the individual a via the role R that they must also belong to C.

B. Semantics

The semantics is given by a fuzzy interpretation $\mathcal{I}$, which is a pair $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ that consists of a nonempty set $\Delta^{\mathcal{I}}$ (the interpretation domain) and a fuzzy interpretation function $\cdot^{\mathcal{I}}$ mapping: *1*) an *individual* a to an element $a^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$; *2*) a fuzzy *concept* C to a function $C^{\mathcal{I}} : \Delta^{\mathcal{I}} \rightarrow \mathcal{N}$; and *3*) a fuzzy *role* R to a function $R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \rightarrow \mathcal{N}$.

$C^{\mathcal{I}}$ (respectively, $R^{\mathcal{I}}$) denotes the membership function of the fuzzy concept C (respectively fuzzy role R) w.r.t. $\mathcal{I}$. $C^{\mathcal{I}}(a^{\mathcal{I}})$ (respectively $R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}})$) gives us to what extent the individual a can be considered as an element of the fuzzy concept C (respectively to what extent (a,b) can be considered as an element of the fuzzy role R) under the fuzzy interpretation $\mathcal{I}$.

The fuzzy interpretation function is defined for fuzzy concepts, roles, and axioms as shown in Table I, where $\otimes, \ominus, \oplus$, and $\Rightarrow$ are a t-norm, a t-conorm, a negation, and an implication function defined over $\mathcal{N}$, respectively. We say that a fuzzy interpretation $\mathcal{I}$ satisfies a fuzzy KB $\mathcal{K}$ iff $\mathcal{I}$ satisfies each element in $\mathcal{K}$.

C. Reasoning Tasks

There are several reasoning tasks in fuzzy DLs.

1) *Fuzzy KB satisfiability* (or *consistency*). A fuzzy interpretation $\mathcal{I}$ *satisfies* (is a model of) a fuzzy KB $\mathcal{K}$, denoted $\mathsf{sat}(\mathcal{K})$, iff it satisfies each fuzzy axiom in $\mathcal{K}$.
2) *Entailment*. A fuzzy concept (or role) assertion τ is entailed by a fuzzy KB $\mathcal{K}$, denoted $\mathcal{K} \models \tau$, iff every model of $\mathcal{K}$ satisfies τ.
3) *Fuzzy concept satisfiability*. C is α-satisfiable w.r.t. a fuzzy KB $\mathcal{K}$ iff there exists a model $\mathcal{I}$ of $\mathcal{K}$ such that $C^{\mathcal{I}}(x) \geq \alpha$ for some $x \in \Delta^{\mathcal{I}}$.
4) *BDB*. The BDB of a concept or role assertion $\tau \in \{a : C, (a,b) : R\}$, which is denoted $\mathsf{bdb}(\mathcal{K}, \tau)$, is defined as the $\sup\{\alpha \in \mathcal{N} : \mathcal{K} \models \langle \tau \geq \alpha\rangle\}$, where $\sup \emptyset = \gamma_0$.
5) *Fuzzy concept subsumption*. D subsumes C w.r.t. a fuzzy KB $\mathcal{K}$, denoted $C \sqsubseteq_{\mathcal{K}} D$, iff every model $\mathcal{I}$ of $\mathcal{K}$ satisfies $\forall x \in \Delta^{\mathcal{I}}, C^{\mathcal{I}}(x) \leq D^{\mathcal{I}}(x)$.[1]

Example 2.2: Let us illustrate the meaning of the BDB problem. Consider the fuzzy KB defined in Example 2.1. In the Zadeh fuzzy logic, it is not hard to see that $\mathsf{bdb}(\mathcal{K}, \mathsf{b} : \mathsf{C}) = \gamma_3$. Actually, the axioms $\{\langle(\mathsf{a},\mathsf{b}) : \mathsf{R} \geq \alpha\rangle, \langle \mathsf{a} : \forall \mathsf{R}.\mathsf{C} \geq \beta\rangle\}, \alpha > \ominus\beta$ imply $\langle \mathsf{b} : \mathsf{C} \geq \beta\rangle$. In Gödel and ukasiewicz fuzzy logics, it also holds that $\mathsf{bdb}(\mathcal{K}, \mathsf{b} : \mathsf{C}) = \gamma_3$. In fact, if the implication is the residuum of the t-norm $\otimes$, then the axioms $\{\langle(\mathsf{a},\mathsf{b}) : \mathsf{R} \geq \alpha\rangle, \langle \mathsf{a} : \forall \mathsf{R}.\mathsf{C} \geq \beta\rangle\}$ imply $\langle \mathsf{b} : \mathsf{C} \geq \alpha \otimes \beta\rangle$.

By abuse of notation, we continue to write $\mathsf{sat}(\mathcal{O})$ and $\mathcal{O} \models \tau$ to denote the satisfiability of a crisp ontology $\mathcal{O}$ and the entailment of a crisp axiom τ by a crisp ontology $\mathcal{O}$, respectively.

D. Reduction to Fuzzy KB Satisfiability

In fuzzy $\mathcal{ALCH}$, these reasoning tasks are mutually reducible [14]. Let a and A be a new individual and a new atomic concept, respectively (i.e., they do not appear in the fuzzy KB) and let $\beta \in \mathcal{N} \setminus \{\gamma_0\}$. We will show here how to reduce the previous reasoning tasks to fuzzy KB satisfiability.

1) *Entailment*. $\mathcal{K} \models \langle a : C \geq \beta\rangle$ iff $\mathcal{K} \cup \{\langle a : C < \beta\rangle\}$ is unsatisfiable. $\mathcal{K} \models \langle(a,b) : R \geq \beta\rangle$ iff $\mathcal{K} \cup \{\langle b : A \geq \gamma_p\rangle\} \models \{\langle a : \exists R.A \geq \beta\rangle\}$.
2) *Fuzzy concept satisfiability*. C is β-satisfiable w.r.t. a fuzzy KB $\mathcal{K}$ iff $\mathcal{K} \cup \{a : C \geq \beta\rangle\}$ is satisfiable.

[1]This basic definition can be generalized as follows: D α-subsumes C w.r.t. $\mathcal{K}$ iff every model $\mathcal{I}$ of $\mathcal{K}$ satisfies $\forall x \in \Delta^{\mathcal{I}}, C^{\mathcal{I}}(x) \Rightarrow D^{\mathcal{I}}(x) \geq \alpha$ for some implication function $\Rightarrow$.

3) *BDB*. By definition, the BDB can be reduced to the entailment problem if the number of degrees of truth is finite and known.
4) *Fuzzy concept subsumption.* $C \sqsubseteq_{\mathcal{K}} D$ iff $\mathcal{K} \cup \{\langle a : C \geq \beta \rangle\} \cup \{\langle a : D < \beta \rangle\}$ is unsatisfiable, for every $\beta \in \mathcal{N} \setminus \{\gamma_0\}$.

The literature mainly focuses on fuzzy satisfiability, because other reasoning tasks can be reduced to it. Instead, we will focus on the BDB and the concept subsumption problems.

E. Crisp Representation Algorithms

There are several reasoning algorithms for fuzzy DLs [5]. In this study, we will focus on a family of reasoning algorithms, which consist on a reduction to reasoning in nonfuzzy ontologies. These algorithms make it possible to reuse current nonfuzzy DL reasoners. We will call this family *crisp representation algorithms*, as they compute a crisp representation of the input fuzzy ontology. There are such algorithms to reason in Zadeh fuzzy logic [16], finite Gödel fuzzy logic [17], finite Ukasiewicz fuzzy logic [18], and any finite smooth t-norm-based fuzzy logic [23]. The parallel versions proposed in this paper are independent of the particular choice.

The basic idea of the crisp representation algorithms is to create some new crisp concepts and roles, which represent the α-cuts of the fuzzy concepts and relations. Next, some new axioms are added to preserve the semantics of these new elements. Finally, every fuzzy axiom in the fuzzy ontology is represented, independently from other axioms, using these new crisp elements. The crisp representation of a fuzzy KB $\mathcal{K}$ is denoted as $\mathsf{crisp}(\mathcal{K})$. Reasoning with $\mathsf{crisp}(\mathcal{K})$ is possible because a fuzzy ontology $\mathcal{K}$ is satisfiable if and only if $\mathsf{crisp}(\mathcal{K})$ is.

Example 2.3: Consider again Example 2.1 and let us show how to compute $\mathsf{crisp}(\mathcal{K})$ for the Zadeh fuzzy logic. We start by creating four new crisp atomic concepts $\mathsf{C}_{\geq\gamma_1}, \mathsf{C}_{\geq\gamma_2}, \mathsf{C}_{\geq\gamma_3}$, and $\mathsf{C}_{\geq\gamma_4}$ and four new crisp roles $\mathsf{R}_{\geq\gamma_1}, \mathsf{R}_{\geq\gamma_2}, \mathsf{R}_{\geq\gamma_3}$, and $\mathsf{R}_{\geq\gamma_4}$. Next, we add to $\mathsf{crisp}(\mathcal{K})$ some axioms to keep their semantics: $\mathsf{C}_{\geq\gamma_2} \sqsubseteq \mathsf{C}_{\geq\gamma_1}, \mathsf{C}_{\geq\gamma_3} \sqsubseteq \mathsf{C}_{\geq\gamma_2}, \mathsf{C}_{\geq\gamma_4} \sqsubseteq \mathsf{C}_{\geq\gamma_3}$, $\mathsf{R}_{\geq\gamma_2} \sqsubseteq \mathsf{R}_{\geq\gamma_1}$, $\mathsf{R}_{\geq\gamma_3} \sqsubseteq \mathsf{R}_{\geq\gamma_2}$, and $\mathsf{R}_{\geq\gamma_4} \sqsubseteq \mathsf{R}_{\geq\gamma_3}$. Finally, the first axiom is represented as $(\mathsf{a}, \mathsf{b}) : \mathsf{R}_{\geq\gamma_4}$, whereas the second one is represented as $\mathsf{a} : \forall \mathsf{R}_{\geq\gamma_2}.\mathsf{C}_{\geq\gamma_3}$.

It is worth to note that the size of $\mathsf{crisp}(\mathcal{K})$ is proportional to both the size of $\mathcal{K}$ and the number of degrees of truth [16]. In Zadeh fuzzy logic, the transformation is polynomial; therefore, the complexity class is not increased (reasoning in the classical DLs $\mathcal{ALCH}$ and $\mathcal{SROIQ}$ is N2ExpTime-complete and ExpTime-complete hard, respectively). In other fuzzy logics, the transformation may be exponential [18].

F. Optimizations

Some optimizations in the setting of crisp representation algorithms have been discussed in [19]. Among them, we will recall here *reasoning avoiding superfluous elements*. Roughly speaking, superfluous elements cannot cause a contradiction when testing KB satisfiability at a given moment, even if they could be necessary in the future, after having added new axioms to the ontology (in particular, as we have seen, reducing a reasoning task into fuzzy KB satisfiability needs to introduce additional axioms). Consequently, we can consider an optimized crisp representation $\mathsf{optimizedCrisp}(\mathcal{K})$ with no superfluous elements.

When computing the satisfiability of $\mathsf{crisp}(\mathcal{K})$, we may replace $T(\mathcal{N})$ and $R(\mathcal{N})$ with $T'(\mathcal{N})$ and $R'(\mathcal{N})$, respectively, such that they do not contain any superfluous concepts and roles, respectively. The optimized crisp representation is denoted $\mathsf{optimizeCrisp}(\mathcal{K}) = \langle \mathsf{crisp}(\mathcal{A}), T'(\mathcal{N}) \cup \mathsf{crisp}(\mathcal{T}), R'(\mathcal{N}) \cup \mathsf{crisp}(\mathcal{R}) \rangle$. The computation of $T'(\mathcal{N})$ and $R'(\mathcal{N})$ is explained in [16]. $\mathsf{crisp}(\mathcal{K})$ is satisfiable iff $\mathsf{optimizedCrisp}(\mathcal{K})$ is.

Algorithm 1 Algorithm to compute the $\mathsf{bdb}(\mathcal{K}, \tau)$ in fuzzy ontologies.

Input: $\mathcal{K}, \tau$
Output: $\mathsf{bdb}(\mathcal{K}, \tau)$

```
min ← 0
max ← p + 1
α ← γ_i for some i ∈ {min + 1, ..., max − 1}
finish ← false
while finish = false do
   if K ⊨ ⟨τ ≥ α⟩ then
      min ← α
   else
      max ← α
   end if
   if min + 1 ≠ max then
      α ← γ_i for some i ∈ {min + 1, ..., max − 1}
   else
      finish = true
   end if
end while
return min
```

Example 2.4: Consider again Example 2.3. When checking the satisfiability of $\mathsf{crisp}(\mathcal{K})$, the concepts $\mathsf{C}_{\geq\gamma_1}, \mathsf{C}_{\geq\gamma_2}$, and $\mathsf{C}_{\geq\gamma_4}$ and the roles $\mathsf{R}_{\geq\gamma_1}$ and $\mathsf{R}_{\geq\gamma_3}$ are superfluous. Hence, we can consider instead the following ontology: $\mathsf{optimizedCrisp}(\mathcal{K}) = \{\mathsf{R}_{\geq\gamma_4} \sqsubseteq \mathsf{R}_{\geq\gamma_2}, (\mathsf{a}, \mathsf{b}) : \mathsf{R}_{\geq\gamma_4}, \ \mathsf{a} : \forall \mathsf{R}_{\geq\gamma_2}.\mathsf{C}_{\geq\gamma_3}\}$. However, if we want to check whether $\mathcal{K}$ entails the axiom $\langle a : C \geq \gamma_1 \rangle$, we must use $\mathsf{crisp}(\mathcal{K})$, as $\mathsf{optimizedCrisp}(\mathcal{K})$ does not provide the correct result. More precisely, $\mathsf{crisp}(\mathcal{K}) \models \mathsf{crisp}(\langle a : C \geq \gamma_1 \rangle)$ but $\mathsf{optimizedCrisp}(\mathcal{K}) \not\models \mathsf{crisp}(\langle a : C \geq \gamma_1 \rangle)$.

III. Parallel Reasoning Algorithms for the Best Degree Bound Problem

In this section, we present new parallel algorithms to compute the BDB. To begin with, we give in Algorithm 1 a formulation of the binary search originally proposed in [14], which computes the BDB of a fuzzy assertion by performing several entailment tests.

Note that in some parts the algorithm takes some degree $\gamma_i \in \mathcal{N}$ such that $min < i < max$, but it does not specify how to choose among the candidates. As usual, if $min = \gamma_j$ and $max = \gamma_k$, the more reasonable option is to take either $n = \gamma_{\lceil (j+k)/2 \rceil}$ or $n = \gamma_{\lfloor (j+k)/2 \rfloor}$. The first option makes γ_p a best case and γ_0 a worst case, whereas the second option is the opposite case. Because it seems reasonable to assume that the number of not entailed axioms is higher than the number of entailed axioms, we have chosen the latter option. However, our experience shows that there are not significant differences in practice.

This algorithm can easily be adapted to work in a crisp representation algorithm, as shown in Algorithm 2, originally used by the fuzzy DL reasoner `DeLorean` [19].

Clearly, the overall response time is more important than the number of entailment tests carried out. Furthermore, it is not difficult to see that the individual entailment tests can be computed in parallel. Following these ideas, we propose the parallel Algorithm 3, where the key insight is that each of the $|\mathcal{N}| - 1$ iterations of the for loop (lines 1–10) can be done in parallel. Then, when all the tests have finished, we take the maximum degree such that the result of the entailment is true. Furthermore, we also apply an additional optimization: If the result of

Algorithm 2 Algorithm to compute the $\mathsf{bdb}(\mathcal{K}, \tau)$ in crisp representation algorithms.

Input: $\mathcal{K}, \tau$
Output: $\mathsf{bdb}(\mathcal{K}, \tau)$

$min \leftarrow 0$
$max \leftarrow p + 1$
$i \leftarrow \lfloor \frac{min+max}{2} \rfloor$
$crispKB \leftarrow \mathsf{crisp}(\mathcal{K})$
while $i \neq min$ **and** $i \neq max$ **do**
  $crispAx \leftarrow \mathsf{crisp}(\langle \tau \geq \gamma_i \rangle)$
  if $crispKB \models crispAx$ **then**
    $min \leftarrow i$
  else
    $max \leftarrow i$
  end if
  $i \leftarrow \lfloor \frac{min+max}{2} \rfloor$
end while
return γ_i

Algorithm 3 Parallel algorithm to compute the $\mathsf{bdb}(\mathcal{K}, \tau)$ in fuzzy ontologies.

Input: $\mathcal{K}, \tau$
Output: $\mathsf{bdb}(\mathcal{K}, \tau)$

for $i := 1$ to $|\mathcal{N}| - 1$ **do** // Parallel for: iterations are run in parallel
  $t[i] \leftarrow$ new thread
  $v[i] \leftarrow \mathcal{K} \models \langle \tau \geq \gamma_i \rangle$, computed by $t[i]$
  if $t[i]$ has finished **and** $v[i] =$ **false then**
    for $j := i + 1$ to $|\mathcal{N}| - 1$ **do**
      stop $t[j]$
      $v[j] \leftarrow$ **false**
    end for
  end if
end for
$i \leftarrow 1$
while $(i < |\mathcal{N}|)$ **and** $(v[i]$ **= true**$)$ **do**
  $i \leftarrow i + 1$
end while
return γ_{i-1}

the ith test is false, then we stop the computation of the jth test for every $j > i$, because we know that the answer will be false as well.

The previous algorithm is independent of how the entailment tests are computed. The next step is to adapt it so we can use it in the context of a crisp reasoning algorithm. To this end, we propose Algorithm 4, where we perform satisfiability tests instead of entailment tests, and now if the result of the ith test is false, we stop the computation of the jth test for every $j < i$, because we know that the answer will be false as well.

Algorithm 4 does not assume that the crisp representation of the fuzzy ontology $\mathsf{crisp}(\mathcal{K})$ can be shared for all the parallel computations. In practice, this is the common situation, as it happens for instance in `DeLorean` reasoner. For this reason, every iteration computes a local version of $\mathsf{crisp}(\mathcal{K})$. While this requires an extra effort, it also makes it possible to apply the well-known optimization of reasoning avoiding superfluous elements. To this end, each iteration i will consider a smaller crisp ontology that contains only the axioms needed to reason with $\langle \tau \geq \gamma_i \rangle$, and does not contain superfluous elements that would be needed to reason with $\langle \tau \geq \gamma_j \rangle$, for $i \neq j$.

Algorithm 4 Parallel algorithm to compute the $\mathsf{bdb}(\mathcal{K}, \tau)$ in crisp representation algorithms.

Input: $\mathcal{K}, \tau$
Output: $\mathsf{bdb}(\mathcal{K}, \tau)$

for $i := 1$ to $|\mathcal{N}| - 1$ **do** // Parallel for: iterations are run in parallel
  $t[i] \leftarrow$ new thread
  $crispKB[i] \leftarrow \mathsf{optimizedCrisp}(\mathcal{K} \cup \{\langle \tau < \gamma_i \rangle\})$, computed by $t[i]$
  $v[i] \leftarrow \mathsf{sat}(crispKB[i])$, computed by $t[i]$
  if $t[i]$ has finished **and** $v[i] =$ **false then**
    for $j := 1$ to $i - 1$ **do**
      stop
      $v[j] \leftarrow$ **false**
    end for
  end if
end for
$i \leftarrow 1$
while $(i < |\mathcal{N}|)$ **and** $(v[i]$ **= false**$)$ **do**
  $i \leftarrow i + 1$
end while
return γ_{i-1}

Example 3.1: Consider the fuzzy KB in Example 2.1 and let us show how Algorithms 3 and 4 behave to compute the BDB of the axiom $\mathsf{b} : \mathsf{C}$. In Algorithm 3, the "for" loop of the algorithm computes the following values: v[1] = v[2] = v[3] = true, v[4] = false. Then, the "while" loop stops when $i = 4$. Consequently, the BDB is γ_3. However, in Algorithm 4 the "for" loop of the algorithm computes the following values: v[1] = v[2] = v[3] = false, v[4] = true. In fact, $\mathsf{optimizedCrisp}(\mathcal{K})$ (see its definition in Example 2.4) entails the axiom $\mathsf{b} : \mathsf{C}_{\geq \gamma_3}$; therefore, v[1], v[2], and v[3] are false. Then, the "while" loop stops when $i = 4$; therefore, the BDB is γ_3.

To end this section, we will highlight the differences between Algorithms 2 (originally implemented by the `DeLorean` reasoner) and 4 (implemented as a result of this research). In the former case, the number of entailment tests *numberOfTests* to compute is $\lfloor \log_2 |\mathcal{N}| \rfloor$ (in the best case) or $\lceil \log_2 |\mathcal{N}| \rceil$ (in the worst case). In the latter case, the number of consistency tests is $|\mathcal{N}| - 1$ in the worst case, but if the result of the entailment tests is false, we can save some of them. Hence, the number ranges in $[1, |\mathcal{N}| - 1]$. Note that the first algorithm requires the time necessary to solve *numberOfTests* tests, while the second requires the needed time to solve $\lceil$*numberOfTests*/*numberOfProcessors*$\rceil$ tests. Furthermore, the tests performed by Algorithm 2 are harder, as they involve a larger crisp representation ontology.

IV. Parallel Reasoning Algorithms for the Fuzzy Concept Subsumption

In this section, we present new parallel algorithms to perform a fuzzy concept subsumption test. To start with, we give in Algorithm 5 an algorithmic formulation of the reduction to fuzzy KB satisfiability already mentioned in Section II and originally proposed in [24].

In principle, we have to perform a satisfiability test for every possible degree of truth in $\mathcal{N}$, but if any of the results of these tests is true (i.e.,

Algorithm 5 Algorithm to compute $C \sqsubseteq_{\mathcal{K}} D$.

Input: $\mathcal{K}, C, D$

Output: true if $C \sqsubseteq_{\mathcal{K}} D$, false otherwise

$a \leftarrow$ a new individual, not appearing in $\mathcal{K}$
$i \leftarrow 1$
$isSat \leftarrow$ **false**
while $i \leq p$ **and** $isSat =$ **false do**
  $isSat \leftarrow \mathsf{sat}(\mathcal{K} \cup \{\langle a : C \geq \gamma_i \rangle\} \cup \{\langle a : D < \gamma_i \rangle\})$
  $i \leftarrow i + 1$
end while
return not $isSat$

Algorithm 6 Algorithm to compute $C \sqsubseteq_{\mathcal{K}} D$ in crisp representation algorithms.

Input: $\mathcal{K}, C, D$

Output: true if $C \sqsubseteq_{\mathcal{K}} D$, false otherwise

$i \leftarrow 1$
$entails \leftarrow$ **true**
$crispKB \leftarrow \mathsf{crisp}(\mathcal{K})$
while $i \leq p$ **and** $entails =$ **true do**
  $crispAx \leftarrow \mathsf{crisp}(C \geq \gamma_i \rangle) \sqsubseteq \mathsf{crisp}(D \geq \gamma_i \rangle)$
  $entails \leftarrow crispKB \models crispAx$
  $i \leftarrow i + 1$
end while
return $entails$

if some of the tested KBs is satisfiable), we can conclude that there is no subsumption. Hence, the number of satisfiability tests ranges in $[1, |\mathcal{N}| - 1]$ and, if the answer of the subsumption test is 1, the exact number is $|\mathcal{N}| - 1$.

Algorithm 6 is a modification of Algorithm 5 to work in a crisp representation algorithm. It is the algorithm that the fuzzy DL reasoner `DeLorean` originally used to solve the problem. Note that inconsistency is transformed into entailment of a concept subsumption axiom.

Algorithm 7 provides a parallel algorithm to solve the fuzzy subsumption problem in fuzzy ontologies. The basic idea is to compute in parallel $|\mathcal{N}| - 1$ satisfiability tests and then to check whether the result of all of them is false (i.e., all the KBs are unsatisfiable). In this case, the answer is true; otherwise, the answer is false.

Algorithm 8 is the corresponding version to work in the context of a crisp reasoning algorithm, adapted to reason avoiding superfluous elements.

Similarly as in the previous section, we will highlight the differences between Algorithms 6 (originally implemented by `DeLorean`) and 8 (implemented as a result of this research). Both of them require a number of consistency tests *numberOfTests* ranging in $[1, |\mathcal{N}| - 1]$. However, the first algorithm requires the time necessary to solve *number Of Tests* tests, while the second requires the needed time to solve ⌈*number Of Tests*/*number Of Processors*⌉ tests, keeping in mind that the tests performed by Algorithm 6 are harder, as they involve a larger crisp representation ontology.

V. Evaluation

The discussions at the end of the two previous sections clearly show that our parallel algorithms (to solve the BDB and the concept subsumption problems) outperform the sequential algorithms provided that there are enough available processors, since the time required to solve the reasoning (entailment or consistency) tests decreases, as each of these individual tests can be computed in parallel, each one in a different processor.

However, one wonders what happens when the number of processors available is less than the number of tests to be done. Since parallel algorithms allow reasoning avoiding superfluous elements, and crisp representations and optimized crisp representations are very different both in size and the required time to reason with them, what will produce better results: reducing the size of the crisp representations or reducing the number of performed tests? The aim of this section is to analyze this question.

Algorithm 7 Parallel algorithm to compute $C \sqsubseteq_{\mathcal{K}} D$ in fuzzy ontologies.

Input: $\mathcal{K}, C, D$

Output: true if $C \sqsubseteq_{\mathcal{K}} D$, false otherwise

$a \leftarrow$ a new individual, not appearing in $\mathcal{K}$
for $i := 1$ to $|\mathcal{N}| - 1$ **do** // Parallel for: iterations are run in parallel
  $t[i] \leftarrow$ new thread
  $v[i] \leftarrow \mathsf{sat}(\mathcal{K} \cup \{\langle a : C \geq \gamma_i \rangle\} \cup \{\langle a : D < \gamma_i \rangle\})$, computed by $t[i]$
  if $t[i]$ has finished **and** $v[i] =$ **true then**
    for $j := 1$ to $j - 1$ **do**
      stop $t[j]$
    end for
    return false;
  end if
end for
return true;

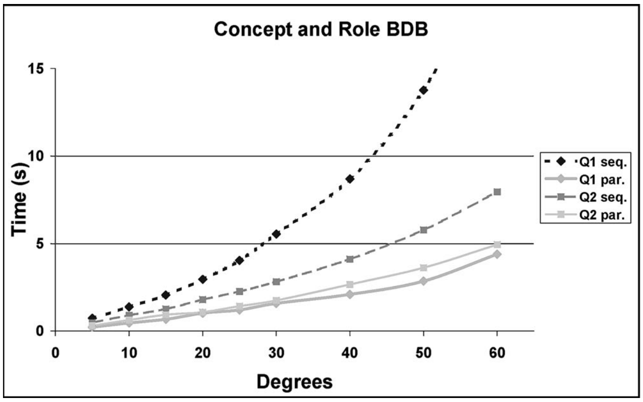
Algorithm 8 Parallel algorithm to compute $C \sqsubseteq_{\mathcal{K}} D$ in crisp representation algorithms.

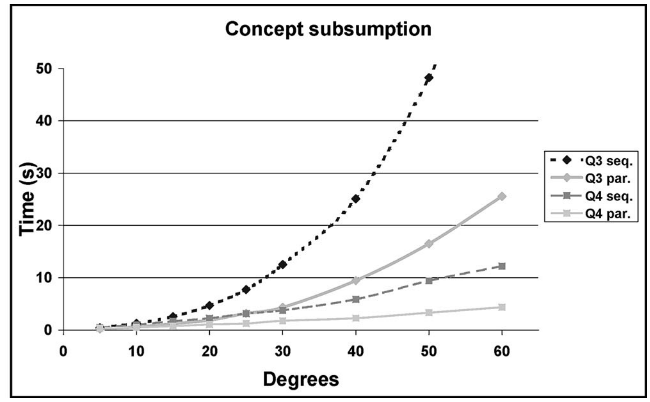
Input: $\mathcal{K}, C, D$

Output: true if $C \sqsubseteq_{\mathcal{K}} D$, false otherwise

$a \leftarrow$ a new individual, not appearing in $\mathcal{K}$
for $i := 1$ to $|\mathcal{N}| - 1$ **do** // Parallel for: iterations are run in parallel
  $t[i] \leftarrow$ new thread
  $crispKB[i] \leftarrow \mathsf{optimizedCrisp}(\mathcal{K} \cup \{\langle a : C \geq \gamma_i \rangle\} \cup \{\langle a : D < \gamma_i \rangle\})$, computed by $t[i]$
  $v[i] \leftarrow \mathsf{sat}(crispKB)$, computed by $t[i]$
  if $t[i]$ has finished **and** $v[i] =$ **true then**
    for $j := 1$ to $j - 1$ **do**
      stop $t[j]$
    end for
    return false;
  end if
end for
return true;

TABLE II
RUNNING TIME OF THE SEQUENTIAL (SEQ.) AND PARALLEL (PAR.) ALGORITHMS FOR QUERIES Q1–Q4





To this end, we have implemented the parallel algorithms as part of DELOREAN. DELOREAN[2] is an implementation of a fuzzy DL reasoner which, given a fuzzy ontology, is able to produce an equivalent OWL 2 ontology (in the sense that it is equivalent to reason with any of them). Then, it uses a classical DL reasoner to reason with the resulting ontology. For implementation details, we refer the reader to [19]. DELOREAN originally implemented Algorithm 2 for the BDB problem and Algorithm 6 for the fuzzy subsumption problem. The research described in this paper has led us to implement Algorithms 4 and 8.

We built a fuzzy ontology to represent the publications of the Distributed Information Systems Research Group in the University of Zaragoza, to which one of the authors belongs to. The list of publications is publicly available in BibTeX format.[3] We used a Java BibTeX API[4] to navigate through the publications and exported them into DELOREAN syntax.

The fuzzy ontology is strongly based on the Bibliographic Ontology (BIBO).[5] Essentially, there are a class Person and a subclass of Document for every type of publication (e.g., Article). The relation creator links documents and authors, and isPartOf relates articles with journal, books, proceedings, etc. isAuthorOf is the inverse relation of creator.

The interesting thing of having a fuzzy ontology to manage the publications is that we are able to quantify their quality. For instance, we can have a fuzzy concept GoodJournal, since some journals are considered better than others. Now, we can define the fuzzy concept C of people who have published some article in a good journal as $\exists\mathsf{isAuthorOf}.(\exists\mathsf{isPartOf}.\mathsf{GoodJournal})$ (this could be easily extended to n papers that use cardinality restrictions).

We have based our quality criteria in Journal Citation Reports,[6] a ranking of scientific journals grouped in categories and ordered by its impact factor, as it is common in our country. The higher the quartile position of the journal, the higher is the membership degree to GoodJournal. We have included the data of the journal categories where some member of the research group has published at. This way, our fuzzy ontology has 62 concepts, 102 roles, and 1082 individuals. Concepts and roles come from the original BIBO ontology, whereas the individuals correspond to our publications and journal data. It could be argued that the size of the fuzzy ontology is not large enough to make our experiments representative; therefore, we have added new concepts to have a total of 1000.

We studied the behavior of the algorithms with different values, as the size of the crisp representations is proportional to the number of degrees of truth. In particular, we considered finite chains of degrees of truth with $p \in \{5, 10, 15, 20, 25, 30, 40, 50, 60\}$ (recall that the number of degrees of truth of a finite chain is $p + 1$).

After having tried several queries, we selected four representative ones: $\mathsf{bdb}(\mathsf{FernandoBobillo} : \mathsf{C})$ (Q1), $\mathsf{bdb}(\mathcal{K}, (\mathsf{FernandoBobillo}, \mathsf{paper001}) : \mathsf{isAuthorOf})$ (Q2), $\mathsf{C} \sqsubseteq_{\mathcal{K}} \mathsf{Book}$ (Q3) and $\mathsf{C} \sqsubseteq_{\mathcal{K}} \mathsf{C}$ (Q4). We found that concept subsumption produces significant differences if the answer is false (Q3) or true (Q4), but this does not happen in the BDB problem.

For each of these cases, we checked that the results are indeed correct and we calculated the execution time. Although the experiments are deterministic, we repeated them ten times to achieve greater precision when measuring the execution time. After discarding the maximum and minimum values, we calculated the average value. We considered a computer with a dual core processor (Intel Core 2 Duo), which internally has two processors.

Table II includes two figures summarizing the results. In all the cases, the parallel algorithms (denoted par.) always scale much better than the sequential ones (denoted seq.). The improvement is higher when computing a concept BDB (Q1) than in the case of a role BDB (Q2). This is probably due to the higher number of concepts, which maximizes the chances of optimizing the number of produced crisp concepts in the parallel algorithms. As expected, if the fuzzy concept subsumption test is false (Q3), the improvement is higher than in the true case (Q4). The reason is that whenever one of the subtests is false, we can stop the other subtests and conclude that the answer is false.

We tried the same fuzzy ontology with a higher number of degrees of truth, but the sequential implementation runs out of memory when $p > 60$ (note that this produces more than 60 000 crisp concepts). Similarly, we tried with a higher number of concepts (10 000), but again the sequential implementation runs out of memory when $p > 30$ and the results are similar to the case of 1000 concepts, and therefore, we will not discuss them here.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have presented some parallel algorithms to reason with fuzzy ontologies when there is a finite number of possible degrees of truth. To the best of our knowledge, this is the first attempt in this direction. In particular, we have focused on solving the BDB and the fuzzy concept subsumption problems.

First, we have proposed two general algorithms to solve these problems, namely Algorithms 3 and 7, respectively. The idea behind them is to compute in parallel several fuzzy satisfiability tests, whatever this

[2]http://webdiis.unizar.es/~fbobillo/delorean
[3]http://sid.cps.unizar.es/BiD
[4]http://code.google.com/p/java-bibtex
[5]http://bibliontology.com
[6]http://science.thomsonreuters.com/training/jcr

is done. Then, we have modified these algorithms; therefore, they can work in the setting of crisp representation algorithms, which perform different reasoning tasks by considering an equivalent nonfuzzy ontology. To this end, we proposed Algorithm 4 for the BDB problem and Algorithm 8 for the fuzzy concept subsumption problem. It is worth to remark that these algorithms are independent of the particular fuzzy DL language and of the fuzzy logic operators considered in the semantics of the logic: We just need a procedure to solve the fuzzy KB satisfiability problem for those operators.

We have also implemented Algorithms 4 and 8 as part of the fuzzy ontology reasoner DeLorean, which supports a fuzzy extension of the ontology language OWL 2. This made it possible to perform an empirical evaluation of the algorithms. Our parallel algorithms outperform the previous sequential algorithms whenever there are enough available processors, as every processor can compute in parallel the necessary individual tests. What is more surprising is that, even in a more conventional computer with two processors, our parallel algorithms scale much better (when the number of degrees of truth increase) than the sequential algorithms. This is not only an advantage of the parallel algorithms themselves but of the use of an optimized crisp representation, which could be used by the sequential algorithms as well.

Future work will include a detailed evaluation of the performance of the new algorithms with real fuzzy ontologies, since unfortunately we are not aware of any publicly available serious fuzzy ontology. Another idea worth mentioning is the implementation of the new algorithms in other fuzzy ontology reasoners. For instance, Algorithms 3 and 7 could be implemented in conjunction with a fuzzy tableau algorithm for Zadeh fuzzy logic, which could be done as an extension of the `FiRE` reasoner.[7]

References

[1] B. Cuenca-Grau, I. Horrocks, B. Motik, B. Parsia, P. F. Patel-Schneider, and U. Sattler, "OWL 2: The next step for OWL," *J. Web Semant.*, vol. 6, no. 4, pp. 309–322, 2008.

[2] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[3] I. Horrocks, O. Kutz, and U. Sattler, "The even more irresistible $\mathcal{SROIQ}$," in *Proc. 10th Int. Conf. Knowl. Represent. Reason.*, 2006, pp. 452–457.

[4] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, pp. 338–353, 1965.

[5] T. Łukasiewicz and U. Straccia, "Managing uncertainty and vagueness in description logics for the semantic web," *J. Web Semant.*, vol. 6, no. 4, pp. 291–308, 2008.

[6] S. Calegari and E. Sanchez, "Object-fuzzy concept network: An enrichment of ontologies in semantic information retrieval," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 59, no. 13, pp. 2171–2185, 2008.

[7] C. Hudelot, J. Atif, and I. Bloch, "Fuzzy spatial relation ontology for image interpretation," *Fuzzy Sets Syst.*, vol. 159, no. 15, pp. 1929–1951, 2008.

[8] S. Dasiopoulou, I. Kompatsiaris, and M. G. Strintzis, "Investigating fuzzy DLs-based reasoning in semantic image analysis," *Multimedia Tools Appl.*, vol. 46, no. 2, pp. 331–370, 2010.

[9] P. C. G. Costa, K. B. Laskey, and T. Lukasiewicz, "Uncertainty representation and reasoning in the semantic web," in *Semantic Web Engineering in the Knowledge Society*, J. Cardoso and M. D. Lytras, Eds. Hershey, PA: IGI Global, 2008.

[10] E. Sanchez, Ed., *Fuzzy Logic and the Semantic Web*. (Capturing Intelligence Series, vol. 1). Amsterdam, The Netherlands: Elsevier Science, 2006.

[11] C. S. Lee, Z. W. Jian, and L. K. Huang, "A fuzzy ontology and its application to news summarization," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 35, no. 5, pp. 859–880, Oct. 2005.

[12] C. S. Lee, M. H. Wang, and H. Hagras, "A type-2 fuzzy ontology and its application to personal diabetic-diet recommendation," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 2, pp. 374–395, Apr. 2010.

[13] P. Tzouveli, N. Simou, G. Stamou, and S. Kollias, "Semantic classification of byzantine icons," *IEEE Intell. Syst.*, vol. 24, no. 2, pp. 35–43, Mar./Apr. 2009.

[14] U. Straccia, "Reasoning within fuzzy description logics," *J. Artif. Intell. Res.*, vol. 14, pp. 137–166, 2001.

[15] F. Bobillo and U. Straccia, "Fuzzy description logics with general t-norms and datatypes," *Fuzzy Sets Syst.*, vol. 160, no. 23, pp. 3382–3402, 2009.

[16] F. Bobillo, M. Delgado, and J. Gómez-Romero, "Crisp representations and reasoning for fuzzy ontologies," *Int. J. Uncertainty Fuzziness Knowl.-Based Syst.*, vol. 17, no. 4, pp. 501–530, 2009.

[17] F. Bobillo, M. Delgado, J. Gómez-Romero, and U. Straccia, "Fuzzy description logics under Gödel semantics," *Int. J. Approx. Reason.*, vol. 50, no. 3, pp. 494–514, 2009.

[18] F. Bobillo and U. Straccia, "Reasoning with the Łukasiewicz fuzzy description logic $\mathcal{SROIQ}$," *Inf. Sci.*, vol. 181, no. 4, pp. 758–778, 2011.

[19] F. Bobillo, M. Delgado, and J. Gómez-Romero, "DeLorean: A reasoner for fuzzy OWL 2," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 258–272, 2012.

[20] G. Mayor and J. Torrens, "On a class of operators for expert systems," *Int. J. Intell. Syst.*, vol. 8, no. 7, pp. 771–778, 1993.

[21] M. Mas, M. Monserrat, and J. Torrens, "S-implications and r-implications on a finite chain," *Kybernetika*, vol. 40, no. 1, pp. 3–20, 2004.

[22] M. Mas, M. Monserrat, and J. Torrens, "On two types of discrete implications," *Int. J. Approx. Reason.*, vol. 40, no. 3, pp. 262–279, 2005.

[23] F. Bobillo, M. Delgado, J. Gómez-Romero, and U. Straccia, "Joining goödel and Zadeh fuzzy logics in fuzzy description logics," *Int. J. Uncertainty Fuzzy Knowl.-Based Syst.*, vol. 20, no. 4, pp. 475–508, 2012.

[24] G. Stoilos, G. Stamou, and J. Z. Pan, "Fuzzy extensions of OWL: Logical properties and reduction to fuzzy description logics," *Int. J. Approx. Reason.*, vol. 51, pp. 656–679, 2010.

[7] http://www.image.ece.ntua.gr/∼nsimou/FiRE

A Matrix Approach to Latticized Linear Programming With Fuzzy-Relation Inequality Constraints

Haitao Li and Yuzhen Wang

***Abstract*—This paper investigates the latticized linear programming that is subject to the fuzzy-relation inequality (FRI) constraints with the max–min composition by using the semi-tensor product method, and proposes a matrix approach to this problem. First, the resolution of the FRI is studied, and it is proved that all the minimal solutions and the unique maximum solution are within the finite parameter set solutions. Based on this and using the semi-tensor product, solving FRIs is converted to solving a set of algebraic inequalities, and some new results on the resolution of FRIs are presented. Second, the latticized linear programming that is subject to the FRI constraints is solved by taking the following two key steps: 1) the optimal value is obtained by calculating the minimum value of the objective function among all the minimal solutions to the FRI constraints; and 2) the optimal solution set is obtained by solving the fuzzy-relation equation that is generated by letting the objective function equal to the optimal value. The study of illustrative examples shows that the new results that are obtained in this paper are very effective in solving the latticized linear programming subject to the FRI constraints.**

Manuscript received June 27, 2012; revised October 26, 2012; accepted November 26, 2012. Date of publication December 11, 2012; date of current version July 31, 2013. This work was supported by the National Natural Science Foundation of China under Grant G61174036, Grant G61074068, and Grant G61034007, the Research Fund for the Taishan Scholar Project of Shandong Province, the Natural Science Foundation of Shandong Province under Grant ZR2010FM013, the Scholarship Award for Excellent Doctoral Student granted by Ministry of Education, and the Graduate Independent Innovation Foundation of Shandong University under Grant yzc10064.

The authors are with the School of Control Science and Engineering, Shandong University, Jinan 250061, China (e-mail: haitaoli09@gmail.com; yzwang@sdu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TFUZZ.2012.2232932