Proceedings of the 6th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2010)

Foreword

This volume contains the papers presented at the *6th International Workshop on Uncertainty Reasoning for the Semantic Web* (URSW 2010), held as a part of the *9th International Semantic Web Conference* (ISWC 2010) at Shanghai, China, November 5, 2010. It contains 8 technical papers and 4 position papers, which were selected in a rigorous reviewing process, where each paper was reviewed by at least four program committee members.

The International Semantic Web Conference is a major international forum for presenting visionary research on all aspects of the Semantic Web. The International Workshop on Uncertainty Reasoning for the Semantic Web is an exciting opportunity for collaboration and cross-fertilization between the uncertainty reasoning community and the Semantic Web community. Effective methods for reasoning under uncertainty are vital for realizing many aspects of the Semantic Web vision, but the ability of current-generation web technology to handle uncertainty is extremely limited. Recently, there has been a groundswell of demand for uncertainty reasoning technology among Semantic Web researchers and developers. This surge of interest creates a unique opening to bring together two communities with a clear commonality of interest but little history of interaction. By capitalizing on this opportunity, URSW could spark dramatic progress toward realizing the Semantic Web vision.

Audience: The intended audience for this workshop includes the following:

- researchers in uncertainty reasoning technologies with interest in Semantic Web and Web-related technologies;
- Semantic Web developers and researchers;
- people in the knowledge representation community with interest in the Semantic Web;
- ontology researchers and ontological engineers;
- Web services researchers and developers with interest in the Semantic Web; and
- developers of tools designed to support Semantic Web implementation, e.g., Jena, Protégé, and Protégé-OWL developers.

Topics: We intended to have an open discussion on any topic relevant to the general subject of uncertainty in the Semantic Web (including fuzzy theory, probability theory, and other approaches). Therefore, the following list should be just an initial guide:

- syntax and semantics for extensions to Semantic Web languages to enable representation of uncertainty;
- logical formalisms to support uncertainty in Semantic Web languages;

– probability theory as a means of assessing the likelihood that terms in different ontologies refer to the same or similar concepts;
– architectures for applying plausible reasoning to the problem of ontology mapping;
– using fuzzy approaches to deal with imprecise concepts within ontologies;
– the concept of a probabilistic ontology and its relevance to the Semantic Web;
– best practices for representing uncertain, incomplete, ambiguous, or controversial information in the Semantic Web;
– the role of uncertainty as it relates to Web services;
– interface protocols with support for uncertainty as a means to improve interoperability among Web services;
– uncertainty reasoning techniques applied to trust issues in the Semantic Web;
– existing implementations of uncertainty reasoning tools in the context of the Semantic Web;
– issues and techniques for integrating tools for representing and reasoning with uncertainty; and
– the future of uncertainty reasoning for the Semantic Web.

We wish to thank all authors who submitted papers and all workshop participants for fruitful discussions. We would like to thank the program committee members and external referees for their timely expertise in carefully reviewing the submissions.

November 2010

Fernando Bobillo
Rommel Carvalho
Paulo C. G. da Costa
Claudia d'Amato
Nicola Fanizzi
Kathryn B. Laskey
Kenneth J. Laskey
Thomas Lukasiewicz
Trevor Martin
Matthias Nickles
Michael Pool

Workshop Organization

Programme Chairs

Fernando Bobillo (University of Zaragoza, Spain)
Rommel Carvalho (George Mason University, USA)
Paulo C. G. da Costa (George Mason University, USA)
Claudia d'Amato (University of Bari, Italy)
Nicola Fanizzi (University of Bari, Italy)
Kathryn B. Laskey (George Mason University, USA)
Kenneth J. Laskey (MITRE Corporation, USA)
Thomas Lukasiewicz (University of Oxford, UK)
Trevor Martin (University of Bristol, UK)
Matthias Nickles (University of Bath, UK)
Michael Pool (Vertical Search Works, Inc., USA)

Programme Committee

Fernando Bobillo (University of Zaragoza, Spain)
Rommel Carvalho (George Mason University, USA)
Paulo C. G. Costa (George Mason University, USA)
Fabio Gagliardi Cozman (University of São Paulo, Brazil)
Claudia d'Amato (University of Bari, Italy)
Nicola Fanizzi (University of Bari, Italy)
Marcelo Ladeira (University of Brasilia, Brazil)
Kathryn Laskey (George Mason University, USA)
Kenneth J. Laskey (MITRE Corporation, USA)
Thomas Lukasiewicz (University of Oxford, UK)
Trevor Martin (University of Bristol, UK)
Matthias Nickles (University of Bath, UK)
Jeff Z. Pan (University of Aberdeen, UK)
Michael Pool (Vertical Search Works, Inc., USA)
Livia Predoiu (University of Mannheim, Germany)
Guilin Qi (Southeast University, China)
David Robertson (University of Edinburgh, UK)
Daniel Sánchez (University of Granada, Spain)
Sergej Sizov (University of Koblenz-Landau, Germany)
Giorgos Stoilos (University of Oxford, UK)
Umberto Straccia (ISTI-CNR, Italy)
Andreas Tolk (Old Dominion University, USA)
Johanna Voelker (University of Karlsruhe, Germany)
Peter Vojtáš (Charles University, Czech Republic)

External Reviewers

Yuan Ren
Edward Thomas
Xiaowang Zhang

Table of Contents

Technical papers

Description Logics over Multisets 1
Yuncheng Jiang

Transforming Fuzzy Description Logic $\mathcal{ALC}_{\mathcal{FL}}$ into Classical Description Logic $\mathcal{ALCH}$ 13
Yining Wu

PrOntoLearn: Unsupervised Lexico-Semantic Ontology Generation using Probabilistic Methods 25
Saminda Abeyruwan, Ubbo Visser, Vance Lemmon, Stephan Schurer

Efficient approximate SPARQL querying of Web of Linked Data 37
Kuldeep B R Reddy, P Sreenivasa Kumar

Semantic Query Extension through Probabilistic Description Logics 49
José Eduardo Ochoa-Luna, Kate Revoredo, Fabio Gagliardi Cozman

Finite Fuzzy Description Logics: A Crisp Representation for Finite Fuzzy $\mathcal{ALCH}$ 61
Fernando Bobillo, Umberto Straccia

PR-OWL 2.0 - Bridging the gap to OWL semantics 73
Rommel Carvalho, Kathryn Laskey, Paulo Costa

Learning Sentences and Assessments in Probabilistic Description Logics . . 85
Joé Eduardo Ochoa Luna, Kate Revoredo, Fabio Gagliardi Cozman

Position papers

SWRL-F - A Fuzzy-logic Extension of the Semantic Web Rule Language . 97
Tomasz Wiktor Wlodarczyk, Martin O'Connor, Chunming Rong, Mark Musen

Scalability of the Crisp Representations of Scalable Fuzzy Description Logics 101
Fernando Bobillo, Miguel Delgado

A Tractable Paraconsistent Fuzzy Description Logic 105
Henrique Viana, Thiago Alves, João Alcântara, Ana Teresa Martins

Default Logics for Plausible Reasoning with Controversial Axioms 109
Thomas Scharrenbach, Claudia d'Amato, Nicola Fanizzi, Rolf Grütter, Bettina Waldvogel, Abraham Bernstein

Description Logics over Multisets

Yuncheng Jiang[1,2]

[1] School of Computer Science, South China Normal University, Guangzhou 510631, P.R. China
[2] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, P.R. China
ycjiang@scnu.edu.cn

Abstract. Description Logics (DLs) are a family of knowledge representation languages that have gained considerable attention the last 20 years. It is well-known that the interpretation domain of classical DLs is a classical set. However, in Science and in the ordinary life the situation is not at all like this. In order to handle these types of knowledge in DLs, in this paper we present a DL framework based on multiset theory. Concretely, we present the DL over multisets $\mathcal{ALC}_{msets}$ which is a semantic extension of the classical DL $\mathcal{ALC}$. The syntax and semantics of $\mathcal{ALC}_{msets}$ are presented. Moreover, we investigate the logical properties of $\mathcal{ALC}_{msets}$ and provide a sound and terminating reasoning algorithm for satisfiability problem of $\mathcal{ALC}_{msets}$.

Keywords: Classical Description Logics; Extended Description Logics; Multisets; Satisfiability

1 Introduction

In the last 20 years a substantial amount of work has been carried out in the context of Description Logics (DLs for short) [1][20]. DLs are a family of logic-based knowledge representation formalisms that are tailored towards representing the terminological knowledge of an application domain in a structured and formally well-understood way. DLs have been applied to numerous problems in computer science such as information integration, databases, software engineering and soft sets. Recent interest in DLs has been spurred by their application in the Semantic Web [2]: the DL $\mathcal{SHOIN}$(**D**) provides the logical underpinning for the Web Ontology Language (OWL), and the DL $\mathcal{SROIQ}$(**D**) is used in OWL 2 [6][11][15][16]. A main point is that DLs are considered as to be attractive logics in knowledge based applications as they are a good compromise between expressive power and computational complexity.

From the semantics of DLs [1] we know that the interpretation domain of classical DLs is a classical set (Zermelo-Fraenkel set) [12]. That is to say, the interpretation of classical DLs is based on classical set theory from a semantics point of view. It is well-known that classical set theory states that a given element can appear only once in a set, it assumes that all mathematical objects occur without

repetition. However, in Science and in the ordinary life the situation is not at all like this. In the physical world it is observed that there is an enormous repetition [7].

As a matter of fact, in order to process the collections with repetition, multi-set theory (MST for short) has been presented and several operations as the addition, the union and the intersection of multisets have been defined and their properties investigated in several papers [3][27][28]. Intuitively, multisets (sometimes also called bags[13][28]) are set-like structures where an element can appear more than once [3]. Thus, a multiset differs from a set in that each element has a multiplicity, which is a natural number indicating (lossely speaking) how many times it is a member of the multiset [7]. We must note that the word multiset was coined by N. G. de Bruijin [18], but the first person that actually used multisets was Richard Dedekind in his well-known paper "Was sind und was sollen die Zahlen?" ("The nature and meaning of numbers") [4]. This paper was published in 1888 [27]. More concretely, a multiset is a collection of objects in which repetition of elements is significant [9]. We confront a number of situations in life when we have to deal with collections of elements in which duplicates are significant. An example may be cited to prove this point. While handling a collection of employees' ages or details of salary in a company, we need to handle entries bearing repetitions and consequently our interest may be diverted to the distribution of elements. In such situations the classical definition of set proves inadequate for the situation presented [9]. Thus, from a practical point of view multisets are very useful structures as they arise in many areas of mathematics and computer science [8][9][19][22][23][27]. A complete survey of the development of multi-set theory can be found in [3].

Naturally, a problem is raised: how we can interpret the concepts and the roles of DLs using multi-set theory? Furthermore, what are the benefits of doing so? After careful thought, we find that it is feasible to interpret the concepts and the roles of DLs using multi-set theory. Moreover, it is a more accurate interpretation for the concepts and the roles of DLs. For example, when we interpret the concept *commended-students* (students who are commended), we can say that *Zhangsan*, *Lisi* and *Wangwu* are the instances of the concept *commended-students*. More formally, we can say $commended\text{-}students^I$={*Zhangsan*, *Lisi*, *Wangwu*} in classical DLs. However, if we consider more accurate situation, e.g., *Zhangsan* is commended three times, *Lisi* is commended twice, and *Wangwu* is commended once, obviously, the classical interpretation of DLs cannot process this situation. Here we can interpret the concept *commended-students* using multi-set theory. Formally, $commended\text{-}students^{MI}$={*Zhangsan*, *Zhangsan*, *Zhangsan*, *Lisi*, *Lisi*, *Wangwu*}, where {*Zhangsan*, *Zhangsan*, *Zhangsan*, *Lisi*, *Lisi*, *Wangwu*} is a multiset.

In this paper we extend DLs allow to express that interpretation of a concept (resp., a role) is not a subset of classical set (traditional interpretation domain Δ^I) (resp., a subset of $\Delta^I \times \Delta^I$) like in classical DLs, but a subset of multisets (resp., a subset of Cartesian product of multisets). That is, we will extend the interpretation domain of DLs to multisets. More concretely, we will present the DL $\mathcal{ALC}_{msets}$, which is a semantic extension of the DL $\mathcal{ALC}$ [10][14][17][24][26] based on multiset theoretic operations presented in [5][9][13]. Moreover, we will provide a sound and incomplete reasoning algorithm for the satisfiability reasoning problem of the DL $\mathcal{ALC}_{msets}$. It is worth noting that classical set is a special case of multisets [9], hence, the DL $\mathcal{ALC}$ [10]

[14][17][24][26] is a special case of the DL $\mathcal{ALC}_{msets}$ presented in this paper from a semantics point of view.

2 Multisets

The current section provides some background on multisets.

A naive concept of multiset was formalized by Blizard [5]. It has the following properties: (i) a multiset is a collection of elements in which certain elements may occur more than once; (ii) occurrences of a particular element in a multiset are indistinguishable; (iii) each occurrence of an element in a multiset contributes to the cardinality of the multiset; (iv) the number of occurrences of a particular element in a multiset is a (finite) positive integer; (v) the number of distinguishable (distinct) elements in a multiset need not be finite; and (vi) a multiset is completely determined if we know the elements that belong to it and the number of times each element belongs to it [9]. In the following, we introduce the basic definitions and notations of multisets [5][9][13].

A collection of elements containing duplicates is called a multiset. Formally, if X is a set of elements, a multiset M drawn from the set X is represented by a function count M or C_M: $X \rightarrow N$ where N represents the set of non-negative integers. For each $x \in X$, $C_M(x)$ is the characteristic value of x in M and indicates the number of occurrences of the element x in M. A multiset M is a set if $\forall x \in X$, $C_M(x)=0$ or 1.

The word "multiset" often shortened to "mset" abbreviates the term "multiple membership set".

Let M_1 and M_2 be two msets drawn from a set X. M_1 is a sub mset of M_2 ($M_1 \subseteq M_2$) if $\forall x \in X$, $C_{M1}(x) \leq C_{M2}(x)$. M_1 is a proper sub mset of M_2 ($M_1 \subset M_2$) if $C_{M1}(x) \leq C_{M2}(x)$ $\forall x \in X$ and there exists at least one $\forall x \in X$ such that $C_{M1}(x) < C_{M2}(x)$. Two msets M_1 and M_2 are equal ($M_1=M_2$) if $M_1 \subseteq M_2$ and $M_2 \subseteq M_1$. An mset M is empty if $\forall x \in X$, $C_M(x)=0$. The cardinality of an mset M drawn from a set X is Card $M=\Sigma_{x \in X} C_M(x)$. It is also denoted by $|M|$.

The insertion of an element x into an mset M results in a new mset M' denoted by $M'=M \oplus x$ such that $C_{M'}(x)=C_M(x)+1$ and $C_{M'}(y)=C_M(y)$ $\forall y \neq x$. Addition of two msets M_1 and M_2 drawn from a set X results in a new mset $M=M_1 \oplus M_2$ such that $\forall x \in X$, $C_M(x)=C_{M1}(x)+C_{M2}(x)$. The removal of an element x from an mset M results in a new mset M' denoted by $M'=M \Theta x$ such that $C_{M'}(x)= \max\{C_M(x)-1, 0\}$ and $C_{M'}(y)=C_M(y)$ $\forall y \neq x$. Subtraction of two msets M_1 and M_2 drawn from a set X results in a new mset $M=M_1 \Theta M_2$ such that $\forall x \in X$, $C_M(x)=\max\{C_{M1}(x)-C_{M2}(x), 0\}$. The union of two msets M_1 and M_2 drawn from a set X is an mset M denoted by $M=M_1 \cup M_2$ such that $\forall x \in X$, $C_M(x)=\max\{C_{M1}(x), C_{M2}(x)\}$. The intersection of two msets M_1 and M_2 drawn from a set X is an mset M denoted by $M=M_1 \cap M_2$ such that $\forall x \in X$, $C_M(x)=\min\{C_{M1}(x), C_{M2}(x)\}$.

Let M be an mset from X with x appearing n times in M. It is denoted by $x \in^n M$. $M=\{k_1/x_1, k_2/x_2, \ldots, k_n/x_n\}$ where M is an mset with x_1 appearing k_1 times, x_2 appearing k_2 times and so on. $[M]_x$ denotes that the element x belongs to the mset M and $|[M]_x|$ denotes the cardinality of an element x in M. The entry of the form $(m/x, n/y)/k$

denotes that x is repeated m times, y is repeated n times and the pair (x, y) is repeated k times. $C_1(x, y)$ denotes the count of the first co-ordinate in the ordered pair (x, y) and $C_2(x, y)$ denotes the count of the second co-ordinate in the ordered pair (x, y).

The mset space X^n is the set of all msets whose elements are in X such that no element in the mset occurs more than n times. The set X^∞ is the set of all msets over a domain X such that there is no limit to the number of occurrences of an element in an mset. If $X=\{x_1, x_2, \ldots, x_k\}$ then $X^n= \{\{n_1/x_1, n_2/x_2, \ldots, n_n/x_n\} \mid \text{for } i=1, 2, \ldots, k; n_i \in \{0, 1, 2, \ldots, n\}\}$.

Let X be a support set and X^n be the mset space defined over X. Then for any mset $M \in X^n$, the complement M^c of M in X^n is an element of X^n such that $\forall x \in X$, $C_{MC}(x)=n-C_M(x)$.

Let M_1 and M_2 be two msets drawn from a set X, then the Cartesian product of M_1 and M_2 is defined as $M_1 \times M_2=\{(m/x, n/y)/mn \mid x \in {}^m M_1, y \in {}^n M_2\}$. The Cartesian product of three or more nonempty msets can be defined by generalizing the definition of the Cartesian product of two msets. Thus the Cartesian product $M_1 \times M_2 \times \ldots \times M_n$ of the nonempty msets $M_1, M_2, \ldots, M_n$ is the mset of all ordered n-tuples $(m_1, m_2, \ldots, m_n)$ where $m_i \in {}^{ri} M_i$, $i=1, 2, \ldots, n$ and $(m_1, m_2, \ldots, m_n) \in {}^p M_1 \times M_2 \times \ldots \times M_n$ with $p=\prod r_i$, where $r_i=C_{Mi}(m_i)$, and $i=1, 2, \ldots, n$.

A sub mset R of $M \times M$ is said to be an mset relation on M if every member $(m/x, n/y)$ of R has a count $C_1(x, y).C_2(x, y)$. We denote m/x related to n/y by $m/x R n/y$.

The domain and range of the mset relation R on M is defined as follows: Dom $R=\{x \in {}^r M \mid \exists y \in {}^s M$ such that $r/x R s/y\}$ where $C_{DomR}(x)=\sup\{C_1(x, y) \mid x \in {}^r M\}$, Ran $R=\{y \in {}^s M \mid \exists x \in {}^r M$ such that $r/x R s/y\}$ where $C_{RanR}(y)=\sup\{C_2(x, y) \mid y \in {}^s M\}$.

3 The $\mathcal{ALC}_{msets}$ DL

In the current section we will present the description logic over multisets $\mathcal{ALC}_{msets}$, which is a semantic extension of the $\mathcal{ALC}$ [1][24]. Concretely, we first define its syntax and semantics. Then, we discuss its logical properties.

3.1 Syntax and Semantics

As usual, we consider an alphabet of distinct concept names (**C**), role names (**R**) and individual names (**I**). The abstract syntax of $\mathcal{ALC}_{msets}$-concepts and $\mathcal{ALC}_{msets}$-roles is the same as that of $\mathcal{ALC}$ [1][24]; however, their semantics is based on interpretations on multisets (msets interpretations for short) (see below). Similarly, $\mathcal{ALC}_{msets}$ keeps the same syntax of terminological axioms (concept inclusions and concept equations) as that of $\mathcal{ALC}$. Interestingly, $\mathcal{ALC}_{msets}$ extends $\mathcal{ALC}$ assertions (concept assertions and role assertions) into mset assertions, where individuals containing duplicates can appear.

In the following, we give the semantics of $\mathcal{ALC}_{msets}$-concepts and $\mathcal{ALC}_{msets}$-roles formally.

An mset interpretation is a pair $MI = (\Delta^{MI}, \bullet^{MI})$, where Δ^{MI} is a non-empty mset (the interpretation domain), and $\bullet^{MI}$ is an interpretation function that assigns each

atomic concept (concept name) $A \in \mathbf{C}$ to a set $A^{MI} \subseteq \Delta^{MI}$, each atomic role (role name) $R \in \mathbf{R}$ (note that in $\mathcal{ALC}_{msets}$ roles are always atomic) to a binary relation $R^{MI} \subseteq \Delta^{MI} \times \Delta^{MI}$, and each individual name $m/a \in \mathbf{I}$ to an element $a^{MI} \in {}^{m}\Delta^{MI}$. This interpretation function is extended to $\mathcal{ALC}_{msets}$ concept descriptions as follows:

- $\top^{MI} = \Delta^{MI}$;
- $\bot^{MI} = \phi$;
- $(\neg C)^{MI} = \Delta^{MI} \Theta C^{MI}$;
- $(C \sqcap D)^{MI} = C^{MI} \cap D^{MI}$;
- $(C \sqcup D)^{MI} = C^{MI} \cup D^{MI}$;
- $(\exists R.C)^{MI} = \{a \in {}^{m}\Delta^{MI} \mid \exists b \in {}^{n}\Delta^{MI}, (m/a, n/b) \in {}^{mn}R^{MI} \wedge b \in {}^{n}C^{MI}\}$;
- $(\forall R.C)^{MI} = \{a \in {}^{m}\Delta^{MI} \mid \forall b \in {}^{n}\Delta^{MI}, (m/a, n/b) \in {}^{mn}R^{MI} \rightarrow b \in {}^{n}C^{MI}\}$.

Note that in this paper we restrict the interpretation domain to be finite. This is not a severe limitation as it is hard to imagine an application involving infinite interpretation domains.

An $\mathcal{ALC}_{msets}$ knowledge base $\mathcal{KB}$ is also composed of a TBox and an ABox. A TBox $\mathcal{T}$ is a finite, possibly empty, set of terminological axioms that could be a combination of concept inclusions of the form $\langle C \sqsubseteq D \rangle$ and concept equations of the form $\langle C \equiv D \rangle$, where C and D are concept descriptions. An mset interpretation MI satisfies $\langle C \sqsubseteq D \rangle$ if $C^{MI} \subseteq D^{MI}$, and it satisfies $\langle C \equiv D \rangle$ if $C^{MI} = D^{MI}$ (i.e., $C^{MI} \subseteq D^{MI}$ and $D^{MI} \subseteq C^{MI}$). An mset interpretation MI satisfies a TBox $\mathcal{T}$ iff MI satisfies every axiom in $\mathcal{T}$; in this case, we say that MI is a model of $\mathcal{T}$.

An ABox $\mathcal{A}$ includes of a set of mset assertions that could be a combination of concept assertions of the form $\langle m/a:C \rangle$ and role assertions of the form $\langle (m/a, n/b):R \rangle$, where a and b are individuals, C is a concept, and R is a role. An mset interpretation MI satisfies $\langle m/a:C \rangle$ if $a^{MI} \in {}^{m}C^{MI}$, and it satisfies $\langle (m/a, n/b):R \rangle$ if $(m/a^{MI}, n/b^{MI}) \in {}^{mn}R^{MI}$. An mset interpretation MI satisfies an ABox $\mathcal{A}$ iff MI satisfies every mset assertion in $\mathcal{A}$ w.r.t. a TBox $\mathcal{T}$; in this case, we say that MI is a model of $\mathcal{A}$ w.r.t. $\mathcal{T}$.

An mset interpretation MI satisfies (or is a model of) a knowledge base $\mathcal{KB} = \langle \mathcal{T}, \mathcal{A} \rangle$ (denoted $MI \models \mathcal{KB}$), iff it satisfies both components of $\mathcal{KB}$; in this case, we say that MI is a model of $\mathcal{KB}$. The knowledge base $\mathcal{KB}$ is consistent if there exists an mset interpretation MI that satisfies $\mathcal{KB}$. We say $\mathcal{KB}$ is inconsistent otherwise.

Description logics over multisets should provide their users with reasoning capabilities that allow them to derive implicit knowledge from the one explicitly represented. In the following we will define the most important reasoning problems of the $\mathcal{ALC}_{msets}$ DL.

Let $\mathcal{T}$ be a TBox, $\mathcal{A}$ an ABox, C, D concept descriptions, and a an individual name. The definitions of the main reasoning problems of the $\mathcal{ALC}_{msets}$ DL are as follows:

- C is subsumed by D w.r.t. $\mathcal{T}$ ($\langle C \sqsubseteq_{\mathcal{T}} D \rangle$) iff $C^{MI} \subseteq D^{MI}$ for all models MI of $\mathcal{T}$;
- C is equivalent to D w.r.t. $\mathcal{T}$ ($\langle C \equiv_{\mathcal{T}} D \rangle$) iff $C^{MI} = D^{MI}$ for all models MI of $\mathcal{T}$;
- C is satisfiable w.r.t. $\mathcal{T}$ iff $C^{MI} \neq \phi$ for some model MI of $\mathcal{T}$;
- $\mathcal{A}$ is consistent w.r.t. $\mathcal{T}$ iff it has a model that is also a model of $\mathcal{T}$;

- m/a is an instance of C w.r.t. $\mathcal{A}$ and $\mathcal{T}$ ($\mathcal{A}\models_{\mathcal{T}}\langle m/a{:}C\rangle$) iff $a^{MI}\in^{m}C^{MI}$ for all models MI of $\mathcal{T}$ and $\mathcal{A}$.

One might think that, in order to realize the reasoning component of an $\mathcal{ALC}_{msets}$ system, one need to design and implement five algorithms, each solving one of the above reasoning problems. Fortunately, this is not the case since there exist some polynomial time reductions (see Section 3.2).

3.2 Logical Properties

It can be easily shown that $\mathcal{ALC}_{msets}$ is a sound extension of $\mathcal{ALC}$, in the sense that the mset interpretations coincide with the traditional interpretations if we restrict the interpretation domain Δ^{MI} to a classical set. However, since $\mathcal{ALC}_{msets}$ is based on multiset theory, some properties which are different from $\mathcal{ALC}$ are obtained. Of course, some properties are the same as that of $\mathcal{ALC}$. In the following, we will discuss these properties.

The first ones are straightforward: $\langle\neg\top\equiv\bot\rangle$, $\langle\neg\bot\equiv\top\rangle$, $\langle C\sqcap\top\equiv C\rangle$, $\langle C\sqcup\bot\equiv C\rangle$, $\langle C\sqcap\bot\equiv\bot\rangle$, $\langle C\sqcup\top\equiv\top\rangle$ and $\langle\exists R.\bot\equiv\bot\rangle$, where C is a concept, R is a role.

The following properties show that some interesting equivalences hold in $\mathcal{ALC}_{msets}$.

Proposition 1. Let C, C_1, C_2, C_3 and D be five concepts. Then

(1) $\langle\neg\neg C\equiv C\rangle$, $\langle C\sqcap C\equiv C\rangle$, $\langle C\sqcup C\equiv C\rangle$;
(2) $\langle\neg(C\sqcap D)\equiv\neg C\sqcup\neg D\rangle$, $\langle\neg(C\sqcup D)\equiv\neg C\sqcap\neg D\rangle$;
(3) $\langle C_1\sqcup(C_2\sqcap C_3)\equiv(C_1\sqcup C_2)\sqcap(C_1\sqcup C_3)\rangle$, $\langle C_1\sqcap(C_2\sqcup C_3)\equiv(C_1\sqcap C_2)\sqcup(C_1\sqcap C_3)\rangle$.

Note 1. Please note that the following properties are satisfied in $\mathcal{ALC}$, however, these properties are not satisfied in $\mathcal{ALC}_{msets}$:

$\langle(C\sqcap\neg C)\equiv\bot\rangle$, $\langle(C\sqcup\neg C)\equiv\top\rangle$, $\langle\forall R.\top\equiv\top\rangle$, $\langle\neg(\forall R.C)\equiv\exists R.\neg C\rangle$, $\langle\neg(\exists R.C)\equiv\forall R.\neg C\rangle$, $\langle(\exists R.C)\sqcup(\exists R.D)\equiv\exists R.(C\sqcup D)\rangle$, and $\langle(\forall R.C)\sqcap(\forall R.D)\equiv\forall R.(C\sqcap D)\rangle$.

There are two interesting remarks here. Firstly, in $\mathcal{ALC}$, we can assume concepts to be in negation normal form (NNF), i.e., negation signs occur immediately in front of concept names only. However, in $\mathcal{ALC}_{msets}$, we cannot do this translation due to $\langle\neg(\forall R.C)\not\equiv\exists R.\neg C\rangle$ and $\langle\neg(\exists R.C)\not\equiv\forall R.\neg C\rangle$. Secondly, in $\mathcal{ALC}$, an ABox $\mathcal{A}$ contains a clash iff $\{A(a), \neg A(a)\}\subseteq\mathcal{A}$ for some individual name a and some concept name A. However, in $\mathcal{ALC}_{msets}$, we cannot use this definition due to $\langle(C\sqcap\neg C)\not\equiv\bot\rangle$ and $\langle(C\sqcup\neg C)\not\equiv\top\rangle$. For example, let $\Delta^{MI}=\{6/a, 8/b\}$ and $\{\langle 3/a{:}C\rangle, \langle 4/b{:}C\rangle\}\subseteq\mathcal{A}$. Since $\langle 3/a{:}C\rangle$ and $\langle 4/b{:}C\rangle$, then we have $\langle 3/a{:}\neg C\rangle, \langle 4/b{:}\neg C\rangle\in\mathcal{A}$. That is, $\{\langle 3/a{:}C\rangle, \langle 3/a{:}\neg C\rangle, \langle 4/b{:}C\rangle, \langle 4/b{:}\neg C\rangle\}\subseteq\mathcal{A}$.

The properties of the polynomial time reductions for reasoning problems are as follows.

Proposition 2. Let $\mathcal{T}$ be a TBox, $\mathcal{A}$ an ABox, C, D concept descriptions, and a an individual name. Then

(1) $\langle C \sqsubseteq_{\mathcal{T}} D\rangle$ iff $\langle C \sqcap D \equiv_{\mathcal{T}} C\rangle$;

(2) $\langle C \equiv_{\mathcal{T}} D\rangle$ iff $\langle C \sqsubseteq_{\mathcal{T}} D\rangle$ and $\langle D \sqsubseteq_{\mathcal{T}} C\rangle$;

(3) C is satisfiable w.r.t. $\mathcal{T}$ iff not $\langle C \sqsubseteq_{\mathcal{T}} \bot\rangle$;

(4) C is satisfiable w.r.t. $\mathcal{T}$ iff there exist $m>0$ and individual a such that $\{\langle m/a:C\rangle\}$ is consistent w.r.t. $\mathcal{T}$;

(5) $\mathcal{A}$ is consistent w.r.t. $\mathcal{T}$ iff $\mathcal{A} \not\models_{\mathcal{T}} \langle m/a:\bot\rangle$ for any $m>0$ and individual a.

Note 2. It needs to be noted that the polynomial time reductions for instance problem to (in)consistency (i.e., $\mathcal{A} \models_{\mathcal{T}} \langle m/a:C\rangle$ iff $\mathcal{A} \cup \{\langle m/a:\neg C\rangle\}$ is inconsistent w.r.t. $\mathcal{T}$) and subsumption problem to (un)satisfiability (i.e., $\langle C \sqsubseteq_{\mathcal{T}} D\rangle$ iff $C \sqcap \neg D$ is unsatisfiable w.r.t. $\mathcal{T}$), are satisfied in $\mathcal{ALC}$, however, these reductions are not satisfied in $\mathcal{ALC}_{msets}$.

Lastly, we have to point out that in the rest of this paper we only consider unfoldable TBoxes. More concretely, a concept definition is of the form $\langle A \equiv C\rangle$ where A is a concept name and C is a concept description. Given a set $\mathcal{T}$ of concept definitions, we say that the concept name A directly uses the concept name B if $\mathcal{T}$ contains a concept definition $\langle A \equiv C\rangle$ such that B occurs in C. Let uses be the transitive closure of the relation "directly uses". We say that $\mathcal{T}$ is cyclic if there is a concept name A that uses itself, and acyclic otherwise. A TBox $\mathcal{T}$ is a finite, possibly empty, set of terminological axioms of the form $\langle A \sqsubseteq C\rangle$, called inclusion introductions, and of the form $\langle A \equiv C\rangle$, called equivalence introductions. A TBox is unfoldable if it contains no cycles and contains only unique introductions, i.e., terminological axioms with only concept names appearing on the left hand side and, for each concept name A, there is at most one axiom in $\mathcal{T}$ of which A appears on the left side.

In classical DLs [1], a knowledge base with an unfoldable TBox can be transformed into an equivalent one with an empty TBox by a transformation called unfolding or expansion [21][25]: Concept inclusion introductions $\langle A \sqsubseteq C\rangle$ are replaced by concept equivalence introductions $\langle A \equiv A' \sqcap C\rangle$, where A' is a new concept name, which stands for the qualities that distinguish the elements of A from the other elements of C. Subsequently, if C is a complex concept expression, which is defined in terms of concept names, defined in the TBox, we replace their definitions in C. It has been proved that the initial TBox with the expanded one are equivalent.

In DLs over msets such as $\mathcal{ALC}_{msets}$ presented in this paper, we also can prove that a knowledge base with an unfoldable TBox can be transformed into an equivalent one with an empty TBox.

Firstly, we can transform an $\mathcal{ALC}_{msets}$-TBox $\mathcal{T}$ into a regular $\mathcal{ALC}_{msets}$-TBox $\mathcal{T}'$, containing equivalence introductions only, such that $\mathcal{T}'$ is equivalent to $\mathcal{T}$ in a sense that will be specified below. We obtain $\mathcal{T}'$ from $\mathcal{T}$ by choosing for every concept inclusion introduction $\langle A \sqsubseteq C\rangle$ in $\mathcal{T}$ a new concept name A' and by replacing the inclusion introduction $\langle A \sqsubseteq C\rangle$ with the equivalence introduction $\langle A \equiv A' \sqcap C\rangle$. The TBox $\mathcal{T}'$ is the normalization of $\mathcal{T}$.

Now we show that $\mathcal{T}$ and $\mathcal{T}'$ are equivalent.

Proposition 3. Let $\mathcal{T}$ be a TBox and $\mathcal{T}'$ its normalization. Then

(1) Every model of $\mathcal{T}'$ is a model of $\mathcal{T}$.

(2) For every model MI of $\mathcal{T}$ there is a model MI' of $\mathcal{T}'$ that has the same domain as MI and agrees with MI on the concept names and roles in $\mathcal{T}$.

Thus, in theory, inclusion introductions do not add to the expressivity of TBoxes. However, in practice, they are a convenient means to introduce concepts into a TBox that cannot be defined completely. In fact, this case is the same as classical DLs [1].

Now we show that, if $\mathcal{T}$ is an unfoldable TBox, we can always reduce reasoning problems w.r.t. $\mathcal{T}$ to problems w.r.t. the empty TBox. Instead of saying "w.r.t. ϕ" one usually says "without a TBox", and omits the index $\mathcal{T}$ for subsumption, equivalence, and instance, i.e., writes $\equiv$, $\sqsubseteq$, $\models$ instead of $\equiv_{\mathcal{T}}$, $\sqsubseteq_{\mathcal{T}}$, and $\models_{\mathcal{T}}$. As we have seen in Proposition 3, $\mathcal{T}$ is equivalent to its expansion $\mathcal{T}'$. Recall that in the expansion every equivalence introduction $\langle A \equiv D \rangle$ such that D contains only concept names, but no concept descriptions. Now, for each concept description C we define the expansion of C w.r.t. $\mathcal{T}$ as the concept description C'' that is obtained from C by replacing each occurrence of a concept name A in C by the concept description D, where $\langle A \equiv D \rangle$ is the equivalence introduction of A in $\mathcal{T}'$, the expansion of $\mathcal{T}$.

Proposition 4. Let $\mathcal{T}$ be an unfoldable TBox, C, D concept descriptions, C'' expansion of C, and D'' expansion of D. Then

(1) $\langle C \equiv_{\mathcal{T}} C'' \rangle$;
(2) C is satisfiable w.r.t. $\mathcal{T}$ iff C'' is satisfiable;
(3) $\langle C \sqsubseteq_{\mathcal{T}} D \rangle$ iff $\langle C'' \sqsubseteq D'' \rangle$;
(4) $\langle C \equiv_{\mathcal{T}} D \rangle$ iff $\langle C'' \equiv D'' \rangle$.

4 Reasoning in $\mathcal{ALC}_{msets}$

In this section, we will provide a detailed presentation of the reasoning algorithm for the $\mathcal{ALC}_{msets}$-satisfiability problem and the properties for the termination and soundness of the procedure. There is one point we have to point out here. Since we restrict the maximal number of occurrences of an element (i.e., an individual) in a multiset (i.e., subset of interpretation domain), it is obvious to know that the satisfiability reasoning algorithm (see below) is incomplete.

In the following, we will present a tableau algorithm for testing satisfiability of an $\mathcal{ALC}_{msets}$-concept. Before we can describe the tableau-based satisfiability algorithm for $\mathcal{ALC}_{msets}$ more formally, we need to introduce some basic notions firstly.

A constraint (denoted by α) is an expression of the form $\langle m/a:C \rangle$, or $\langle (m/a, n/b):R \rangle$, where a and b are individuals, C is a concept, and R is a role. Our calculus, determining whether a finite set $\mathcal{S}$ of constraints or not, is based on a set of constraint propagation rules transforming a set $\mathcal{S}$ of constraints into "simpler" satisfiability preserving sets $\mathcal{S}_i$ until either all $\mathcal{S}_i$ contain a clash (indicating that from all the $\mathcal{S}_i$ no model of $\mathcal{S}$ can be build) or some $\mathcal{S}_i$ is completed and clash-free, that is, no rule can

further be applied to $\mathcal{S}_i$ and $\mathcal{S}_i$ contains no clash (indicating that from $\mathcal{S}_i$ a model of $\mathcal{S}$ can be build). A set of constraints $\mathcal{S}$ contains a clash iff $\{\langle m/a{:}C\rangle, \langle 0/a{:}C\rangle\} \subseteq \mathcal{S}$ for some $m>0$, individual a, and concept description C.

The $\rightarrow_{\neg}$-rule

Condition: $\mathcal{S}_i$ contains $\langle m/a{:}\neg C\rangle$, but it does not contain $\langle 1/a{:}C\rangle$, $\langle 2/a{:}C\rangle$, …, or $\langle nmax-m/a{:}C\rangle$.

Action: $\mathcal{S}_{i,1}=\mathcal{S}_i\cup\{\langle 1/a{:}C\rangle\}$, $\mathcal{S}_{i,2}=\mathcal{S}_i\cup\{\langle 2/a{:}C\rangle\}$, …, $\mathcal{S}_{i,nmax-m}=\mathcal{S}_i\cup\{\langle nmax-m/a{:}C\rangle\}$.

The $\rightarrow_{\sqcap}$-rule

Condition: $\mathcal{S}_i$ contains $\langle m/a{:}C_1\sqcap C_2\rangle$, but neither $\{\langle m/a{:}C_1\rangle, \langle j/a{:}C_2\rangle\}$ nor $\{\langle m/a{:}C_2\rangle, \langle j/a{:}C_1\rangle\}$, where $m\leq j\leq nmax$.

Action: $\mathcal{S}_{i,1}'=\mathcal{S}_i\cup\{\langle m/a{:}C_1\rangle, \langle m/a{:}C_2\rangle\}$, $\mathcal{S}_{i,2}'=\mathcal{S}_i\cup\{\langle m/a{:}C_1\rangle, \langle m+1/a{:}C_2\rangle\}$, ..., $\mathcal{S}_{i,nmax+1}'=\mathcal{S}_i\cup\{\langle m/a{:}C_1\rangle, \langle nmax/a{:}C_2\rangle\}$, $\mathcal{S}_{i,1}''=\mathcal{S}_i\cup\{\langle m/a{:}C_2\rangle, \langle m/a{:}C_1\rangle\}$, $\mathcal{S}_{i,2}''=\mathcal{S}_i\cup\{\langle m/a{:}C_2\rangle, \langle m+1/a{:}C_1\rangle\}$, ..., $\mathcal{S}_{i,nmax+1}''=\mathcal{S}_i\cup\{\langle m/a{:}C_2\rangle, \langle nmax/a{:}C_1\rangle\}$.

The $\rightarrow_{\sqcup}$-rule

Condition: $\mathcal{S}_i$ contains $\langle m/a{:}C_1\sqcup C_2\rangle$, but neither $\{\langle m/a{:}C_1\rangle, \langle j/a{:}C_2\rangle\}$ nor $\{\langle m/a{:}C_2\rangle, \langle j/a{:}C_1\rangle\}$, where $1\leq j\leq m$.

Action: $\mathcal{S}_{i,1}'=\mathcal{S}_i\cup\{\langle m/a{:}C_1\rangle, \langle m/a{:}C_2\rangle\}$, $\mathcal{S}_{i,2}'=\mathcal{S}_i\cup\{\langle m/a{:}C_1\rangle, \langle m-1/a{:}C_2\rangle\}$, ..., $\mathcal{S}_{i,m}'=\mathcal{S}_i\cup\{\langle m/a{:}C_1\rangle, \langle 1/a{:}C_2\rangle\}$, $\mathcal{S}_{i,1}''=\mathcal{S}_i\cup\{\langle m/a{:}C_2\rangle, \langle m/a{:}C_1\rangle\}$, $\mathcal{S}_{i,2}''=\mathcal{S}_i\cup\{\langle m/a{:}C_2\rangle, \langle m-1/a{:}C_1\rangle\}$, ..., $\mathcal{S}_{i,m}''=\mathcal{S}_i\cup\{\langle m/a{:}C_2\rangle, \langle 1/a{:}C_1\rangle\}$.

The $\rightarrow_{\exists}$-rule

Condition: $\mathcal{S}_i$ contains $\langle m/a{:}\exists R.C\rangle$, but there are no individuals $1/b$, $2/b$, …, $nmax/b$ such that $\langle 1/b{:}C\rangle$ and $\langle (m/a,\ 1/b){:}R\rangle$, $\langle 2/b{:}C\rangle$ and $\langle (m/a,\ 2/b){:}R\rangle$, …, or $\langle nmax/b{:}C\rangle$ and $\langle (m/a,\ nmax/b){:}R\rangle$ are in $\mathcal{S}_i$.

Action: $\mathcal{S}_{i,1}=\mathcal{S}_i\cup\{\langle 1/b{:}C\rangle, \langle (m/a,\ 1/b){:}R\rangle\}$, $\mathcal{S}_{i,2}=\mathcal{S}_i\cup\{\langle 2/b{:}C\rangle, \langle (m/a,\ 2/b){:}R\rangle\}$, …, $\mathcal{S}_{i,nmax}=\mathcal{S}_i\cup\{\langle nmax/b{:}C\rangle, \langle (m/a,\ nmax/b){:}R\rangle\}$, where $1/b$, $2/b$, ..., $nmax/b$ are individuals not occurring in $\mathcal{S}_i$.

The $\rightarrow_{\forall}$-rule

Condition: $\mathcal{S}_i$ contains $\langle m/a{:}\forall R.C\rangle$ and $\langle (m/a,\ n/b){:}R\rangle$, but it does not contain $\langle n/b{:}C\rangle$.

Action: $\mathcal{S}_i'=\mathcal{S}_i\cup\{\langle n/b{:}C\rangle\}$.

Fig. 1. Transformation rules of the satisfiability algorithm

The tableau-based satisfiability algorithm for $\mathcal{ALC}_{msets}$ works as follows. Let C by an $\mathcal{ALC}_{msets}$-concept. In order to test satisfiability of C, the algorithm starts with a finite set of constraints $\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_{\text{nmax}}\}$, and applies satisfiability preserving transformation rules (see Figure 1) (in arbitrary order) to the set of constraints $\mathcal{S}_i$ ($1\leq i\leq nmax$) until no more rules apply, where $\mathcal{S}_1=\{\langle 1/a{:}C\rangle\}$, $\mathcal{S}_2=\{\langle 2/a{:}C\rangle\}$, …, $\mathcal{S}_{nmax}=\{nmax/a{:}C\}$. If the "complete" constraint obtained this way does not contain an

obvious contradiction (called clash), then $\mathcal{S}$ is consistent (and thus C is satisfiable), and inconsistent (unsatisfiable) otherwise. The transformation rules that handle negation, conjunction, disjunction, and exists restrictions are non-deterministic in the sense that a given set of constraints is transformed into finitely many new sets of constraints such that the original set of constraints is consistent iff one of the new sets of constraints is so. For this reason we will consider finite sets of constraints $\mathcal{S}=\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k\}$ instead of the original set of constraints $\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_{nmax}\}$, where $k \geq nmax$. Such a set is consistent iff there is some i, $1 \leq i \leq k$, such that $\mathcal{S}_i$ is consistent. A rule of Figure 1 is applied to a given finite set of constraints $\mathcal{S}$ as follows: it takes an element $\mathcal{S}_i$ of $\mathcal{S}$, and replaces it by one set of constraints $\mathcal{S}_i'$, by two sets of constraints $\mathcal{S}_i'$ and $\mathcal{S}_i''$, or by finitely many sets of constraints $\mathcal{S}_{i,j}$.

Termination and soundness of the procedures can be shown.

Proposition 5 (Termination)**.** Let C be an $\mathcal{ALC}_{msets}$-concept. There cannot be some infinite sequences of rule applications

$$\{\langle j/a:C\rangle\} \rightarrow \mathcal{S}_1 \rightarrow \mathcal{S}_2 \rightarrow \ldots, \text{ where } 1 \leq j \leq nmax.$$

Proof. The main reasons for this proposition to hold are the following.

(1) The original sets of constraints $\{\langle j/a:C\rangle\}$ is finite. Namely, there exist $nmax$ original sets of constraints $\{\langle 1/a:C\rangle\}$, $\{\langle 2/a:C\rangle\}$, …, $\{\langle nmax/a:C\rangle\}$.

(2) Without loss of generality, we consider the original set of constraints $\{\langle j/a:C\rangle\}$. Let $\mathcal{S}'$ be a set of constraints contained in $\mathcal{S}_i$ for some $i \geq 1$. For every individual $m/b \neq j/a$ occurring in $\mathcal{S}'$, there is a unique sequence $R_1, \ldots, R_k$ ($k \geq 1$) of role names and a unique sequence of individuals of the form $1/b_1$, $1/b_2$, …, $1/b_{k-1}$, or $1/b_1$, $1/b_2$, …, $2/b_{k-1}$, …, or $1/b_1$, $1/b_2$, …, $nmax/b_{k-1}$, …, or $nmax/b_1$, $nmax/b_2$, …, $nmax/b_{k-1}$, such that $\{\langle (j/a, 1/b_1):R_1\rangle, \langle (1/b_1, 1/b_2):R_2\rangle, \ldots, \langle (1/b_{k-1}, m/b):R_k\rangle\} \subseteq \mathcal{S}'$, $\{\langle (j/a, 1/b_1):R_1\rangle, \langle (1/b_1, 1/b_2):R_2\rangle, \ldots, \langle (2/b_{k-1}, m/b):R_k\rangle\} \subseteq \mathcal{S}'$, …, or $\{\langle (j/a, nmax/b_1):R_1\rangle, \langle (nmax/b_1, nmax/b_2):R_2\rangle, \ldots, \langle (nmax/b_{k-1}, m/b):R_k\rangle\} \subseteq \mathcal{S}'$. In this case, we say that m/b occurs on the level k in $\mathcal{S}'$.

(3) If $\langle m/b:C'\rangle \in \mathcal{S}'$ for an individual m/b on level k, then the maximal role depth of C' (i.e., the maximal nesting of constructors involving roles) is bounded by the maximal role depth of C minus k. Consequently, the level of any individual in $\mathcal{S}'$ is bounded by the maximal role depth of C.

(4) If $\langle m/b:C'\rangle \in \mathcal{S}'$, then C' is a subdescription of C. Consequently, the number of different concept assertions on m/b is bounded by the size of C.

(5) The number of different role successors of m/b in $\mathcal{S}'$ (i.e., individuals l/c such that $\langle (m/b, l/c):R\rangle \in \mathcal{S}'$ for a role name R) is bounded by the number of different existential restrictions in C. □

Proposition 6 (Soundness)**.** Assume that $\mathcal{S}'$ is obtained from the finite set of constraints $\mathcal{S}$ by application of a transformation rule. If $\mathcal{S}$ is consistent, then $\mathcal{S}'$ is consistent.

Proof. [Sketch] Given the termination property (see Proposition 5), it is easily verified, by case analysis, that the transformation rules of the satisfiability algorithm are sound. For example, the $\rightarrow_{\neg}$-rule: Assume that *MI* is an mset interpretation satisfying $\langle m/a{:}\neg C\rangle$, where $0<m\leq nmax$. Let us show that *MI* satisfies $\langle 1/a{:}C\rangle$, $\langle 2/a{:}C\rangle$, …, or $\langle nmax-m/a{:}C\rangle$. Since *MI* satisfies $\langle m/a{:}\neg C\rangle$, by the semantics of $\neg C$ we have that $a\in^{m}(\neg C)^{MI}=\Delta^{MI}\Theta C^{MI}$. Since the maximal number of occurrences of a in Δ^{MI} is $nmax$, thus, by the definition of subtraction of two msets, we know that the number of occurrences of a in C^{MI} is 1, 2, …, or $nmax-m$. That is, $a\in^{1}C^{MI}$, $a\in^{2}C^{MI}$, …, or $a\in^{nmax-m}C^{MI}$. Therefore, *MI* satisfies $\langle 1/a{:}C\rangle$, $\langle 2/a{:}C\rangle$, …, or $\langle nmax-m/a{:}C\rangle$. □

5 Conclusion

We present a DL framework based on multiset theory. Our main feature is that we extend classical DLs allow to express that interpretation of a concept (resp., a role) is not a subset of classical set (resp., a subset of Cartesian product of sets) like in classical DLs, but a subset of multisets (resp., a subset of Cartesian product of multisets). To the best of our knowledge, this is the first attempt in this direction.

Current research effort is to implement the reasoning algorithm and to perform an empirical evaluation in real scenarios. An interesting topic of future research is to study the complexity and optimization techniques of reasoning in DLs over multisets such as $\mathcal{ALC}_{msets}$. Furthermore, additional research effort can be focused on the reasoning algorithms for the (very) expressive DLs over multisets such as $\mathcal{SROIQ}_{msets}$.

Acknowledgments. The works described in this paper are supported by The National Natural Science Foundation of China under Grant No. 60663001; The Foundation of the State Key Laboratory of Computer Science of Chinese Academy of Sciences under Grant No. SYSKF0904; The Natural Science Foundation of Guangdong Province of China under Grant No. 10151063101000031; The Natural Science Foundation of Guangxi Province of China under Grant No. 0991100.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.,: The Description Logic Handbook: Theory, Implementation and Applications, 2nd Edition. Cambridge University Press, Cambridge (2007)
2. Berners-Lee, T., Hendler, J., Lassila, O.,: The Semantic Web. Scientific American 284, 34--43 (2001)
3. Blizard, W.D.,: The Development of Multiset Theory. Modern Logic 1, 319--352 (1991)
4. Blizard, W.D.,: Dedeking Multisets and Functions Shells. Theoretical Computer Science, 110, 79--98 (1993)
5. Blizard, W.D.,: Multiset Theory. Notre Dame Journal of Logic 30, 36--65 (1989)
6. Bobillo, F., Delgado, M., Gomez-Romero, J., Straccia, U.,: Fuzzy Description Logics under Gödel Semantics. International Journal of Approximate Reasoning 50, 494--514 (2009)
7. Casasnovas, J., Mayor, G.,: Discrete t-norms and Operations on Extended Multisets. Fuzzy Sets and Systems 159, 1165--1177 (2008)

8. Csuhaj-Varju, E., Martin-Vide, C., Mitrana, V.,: Multiset Automata. In: Calude, C.S., Paun, G., Rozenberg, G., Salomaa, A. (eds.). Multiset Processing: Mathematical, Computer Science, and Molecular Computing Points of View. LNCS, vol. 2235, pp. 69--83. Springer, Heidelberg (2001)
9. Girish, K.P., John, S.J.,: Relations and Functions in Multiset Context. Information Sciences 179, 758--768 (2009)
10. Haarslev, V., Pai, H.I., Shiri, N.,: A Formal Framework for Description Logics with Uncertainty. International Journal of Approximate Reasoning 50, 1399--1415 (2009)
11. Horrocks, I., Kutz, O., Sattler, U.,: The even more irresistible SROIQ. In: Proceedings of the 10th International Conference of Knowledge Representation and Reasoning, pp. 57--67. AAAI Press (2006)
12. Hrbacek, K., Jech, T.,: Introduction to Set Theory. Marcel Dekker Inc. (1984)
13. Jena, S.P., Ghosh, S.K., Tripathy, B.K.,: On the Theory of Bags and Lists. Information Sciences 132, 241--254 (2001)
14. Jiang, Y., Tang, Y., Wang, J., Tang, S.,: Reasoning within Intuitionistic Fuzzy Rough Description Logics. Information Sciences 179, 2362--2378 (2009)
15. Jiang, Y., Wang, J., Deng, P., Tang, S.,: Reasoning within Expressive Fuzzy Rough Description Logics. Fuzzy Sets and Systems 160, 3403--3424 (2009)
16. Jiang, Y., Tang, Y., Wang, J., Deng, P., Tang, S.,: Expressive Fuzzy Description Logics over Lattices. Knowledge-Based Systems 23, 150--161 (2010)
17. Jiang, Y., Wang, J., Tang, S., Xiao, B.,: Reasoning with Rough Description Logics: An Approximate Concepts Approach. Information Sciences 179, 600--612 (2009)
18. Knuth, D.E.,: The Art of Computer Programming, vol. 2: Seminumerical Algorithms. Addison-Wesley (1981)
19. Lamperti, G., Melchiori, M., Zanella, M.,: On Multisets in Database Systems. In: Calude, C.S., Paun, G., Rozenberg, G., Salomaa A. (eds.). Multiset Processing: Mathematical, Computer Science, and Molecular Computing Points of View. LNCS, vol. 2235, pp. 147--215. Springer, Heidelberg (2001)
20. Lutz, C.,: NEXP TIME-complete Description Logics with Concrete Domains. ACM Transactions on Computational Logic 5, 669--705 (2004)
21. Nebel, B.,: Terminological Reasoning is Inherently Intractable. Artificial Intelligence 43, 235--249 (1990)
22. Miyamoto, S.,: Information Clustering Based on Fuzzy Multisets. Information Processing and Management 39, 195--213 (2003)
23. Miyamoto, S.,: Fuzzy Multisets and Their Generalizations. In: Calude, C.S., Paun, G., Rozenberg, G., Salomaa A. (eds.). Multiset Processing: Mathematical, Computer Science, and Molecular Computing Points of View. LNCS, vol. 2235, pp. 225--235. Springer, Heidelberg (2001)
24. Schmidt-Schauβ, M., Smolka, G.,: Attributive Concept Descriptions with Complements. Artificial Intelligence 48, 1--26 (1991)
25. Stoilos, G., Stamou, G., Pan, J.Z., Tzouvaras, V., Horrocks, I.,: Reasoning with Very Expressive Fuzzy Description Logics. Journal of Artificial Intelligence Research 30, 273--320 (2007)
26. Straccia, U.,: Reasoning within Fuzzy Description Logics. Journal of Artificial Intelligence Research 14, 137--166 (2001)
27. Syropoulos, A.,: Mathematics of Multisets. In: Calude, C.S., Paun, G., Rozenberg, G., Salomaa A. (eds.). Multiset Processing: Mathematical, Computer Science, and Molecular Computing Points of View. LNCS, vol. 2235, pp. 347--358. Springer, Heidelberg (2001)
28. Yager, R.,: On the Theory of Bags. International Journal of General Systems 13, 23--37 (1986)

Transforming Fuzzy Description Logic $\mathcal{ALC}_{\mathcal{FL}}$ into Classical Description Logic $\mathcal{ALCH}$

Yining Wu

University of Luxembourg, Luxembourg

Abstract. In this paper, we present a satisfiability preserving transformation of the fuzzy Description Logic $\mathcal{ALC}_{\mathcal{FL}}$ into the classical Description Logic $\mathcal{ALCH}$. We can use the already existing DL systems to do the reasoning of $\mathcal{ALC}_{\mathcal{FL}}$ by applying the result of this paper. This work is inspired by Straccia, who has transformed the fuzzy Description Logic f$\mathcal{ALCH}$ into the classical Description Logic $\mathcal{ALCH}$.

1 Introduction

The Semantic Web is a vision for the future of the Web in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web. While as a basic component of the Semantic Web, an ontology is a collection of information and is a document or file that formally defines the relations among terms. OWL[1] is a *Web Ontology Language* and is intended to provide a language that can be used to describe the classes and relations between them that are inherent in Web documents and applications. The OWL language provides three increasingly expressive sublanguages: OWL Lite, OWL DL, OWL Full. OWL DL is so named due to its correspondence with description logics. OWL DL was designed to support the existing Description Logic business segment and has desirable computational properties for reasoning systems. According to the corresponding relation between axioms of OWL ontology and terms of Description Logic, we can represent the knowledge base contained in the ontology in syntax of DLs.

Description Logics (DLs) [1] have been studied and applied successfully in a lot of fields. The concepts in classical DLs are usually interpreted as crisp sets, i.e., an individual either belongs to the set or not. In the real world, the answers to some questions are often not only yes or no, rather we may say that an individual is an instance of a concept only to some certain degree. We often say linguistic terms such as "Very", "More or Less" etc. to distinguish, e.g. between a young person and a very young person. In 1970s, the theory of approximate reasoning based on the notions of linguistic variable and fuzzy logic was introduced and developed by Zadeh [19–21]. Adverbs as "Very", "More or Less" and "Possibly"

[1] Please visit http://www.w3.org/TR/owl-guide/ for more details.

are called hedges in fuzzy DLs. Some approaches to handling uncertainty and vagueness in DL for the Semantic Web are described in [10].

A well known feature of DLs is the emphasis on reasoning as a central service. Some reasoning procedures for fuzzy DLs have been proposed in [16]. A transformation of $\mathfrak{f}\mathcal{ALCH}$ into $\mathcal{ALCH}$ has been presented in [17]. This approach, however, only works for DLs where modifier concepts are not allowed.

In this paper we consider the fuzzy linguistic description logic $\mathcal{ALC}_{\mathcal{FL}}$ [7] which is an instance of the description logic framework $\mathcal{L}-\mathcal{ALC}$ with the certainty lattice characterized by a hedge algebra and allows the modification by hedges. Because the certainty lattice is characterized by a HA, the modification by hedges becomes more natural than that in $\mathcal{ALC}_{\mathcal{FH}}$ [8] and $\mathcal{ALC}_{\mathcal{FLH}}$ [14] which extend fuzzy $\mathcal{ALC}$ by allowing the modification by hedges of HAs. We will present a satisfiability preserving transformation of $\mathcal{ALC}_{\mathcal{FL}}$ into $\mathcal{ALCH}$ which makes the reuse of the technical results of classical Dls for $\mathcal{ALC}_{\mathcal{FL}}$ feasible.

The remaining part of this paper is organized in the following way. First we state some preliminaries on $\mathcal{ALCH}$, hedge algebra and $\mathcal{ALC}_{\mathcal{FL}}$. Then we present the transformation of $\mathcal{ALC}_{\mathcal{FL}}$ into $\mathcal{ALCH}$. Finally we discuss the main result of the paper and identify some possibilities for further work.

2 Preliminaries

$\mathcal{ALCH}$

We consider the language $\mathcal{ALCH}$ (Attributive Language with Complement and role Hierarchy). In abstract notation, we use the letters A and B for concept names, the letter R for role names, and the letters C and D for concept terms.

Definition 1. *Let N_R and N_C be disjoint sets of role names and concept names. Let $A \in N_C$ and $R \in N_R$.* Concept terms *in $\mathcal{ALCH}$ are formed according to the following syntax rule:*

$$A|\top|\bot|C \sqcap D|C \sqcup D|\neg C|\forall R.C|\exists R.C$$

The semantics of concept terms are defined formally by interpretations.

Definition 2. *An* interpretation *$\mathcal{I}$ is a pair $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a nonempty set (*interpretation domain*) and $\cdot^{\mathcal{I}}$ is an* interpretation function *which assigns to each concept name A a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and to each role name R a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation of complex concept terms is extended by the following inductive definitions:*

$$\begin{aligned}
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \\
\bot^{\mathcal{I}} &= \emptyset \\
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(\forall R.C)^{\mathcal{I}} &= \{d \in \Delta^{\mathcal{I}} \mid \forall d'.(d,d') \notin R^{\mathcal{I}} \text{ or } d' \in C^{\mathcal{I}}\} \\
(\exists R.C)^{\mathcal{I}} &= \{d \in \Delta^{\mathcal{I}} \mid \exists d'.(d,d') \in R^{\mathcal{I}} \text{ and } d' \in C^{\mathcal{I}}\}
\end{aligned}$$

A concept term C is *satisfiable* iff there exists an interpretation $\mathcal{I}$ such that $C^{\mathcal{I}} \neq \emptyset$, denoted by $\mathcal{I} \models C$. Two concept terms C and D are *equivalent* (denoted by $C \equiv D$) iff $C^{\mathcal{I}} = D^{\mathcal{I}}$ for all interpretation $\mathcal{I}$.

We have seen how we can form complex descriptions of concepts to describe classes of objects. Now, we introduce *terminological axioms*, which make statements about how concept terms and roles are related to each other respectively.

In the most general case, *terminological axiom* have the form $C \sqsubseteq D$ or $R \sqsubseteq S$, where C, D are concept terms, R, S are role names. This kind of terminological axioms are also called *inclusions*. A set of axioms of the form $R \sqsubseteq S$ is called *role hierarchy*. An interpretation $\mathcal{I}$ *satisfies* an inclusion $C \sqsubseteq D$ ($R \sqsubseteq S$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ ($R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$), denoted by $\mathcal{I} \models C \sqsubseteq D$ ($\mathcal{I} \models R \sqsubseteq S$).

A *terminology*, i.e., *TBox*, is a finite set of terminological axioms. An interpretation $\mathcal{I}$ *satisfies* (is a *model* of) a terminology $\mathcal{T}$ iff $\mathcal{I}$ *satisfies* each element in $\mathcal{T}$, denoted by $\mathcal{I} \models \mathcal{T}$.

Assertions define how individuals relate with each other and how individuals relate with concept terms. Let N_I be a set of individual names which is disjoint to N_R and N_C. An *assertion* α is an expression of the form $a : C$ or $(a, b) : R$, where $a, b \in N_I$, $R \in N_R$ and $C \in N_C$. A finite set of *assertions* is called *ABox*. An interpretation $\mathcal{I}$ *satisfies* a concept assertion $a : C$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$, denoted by $\mathcal{I} \models a : C$. $\mathcal{I}$ *satisfies* a role assertion $(a, b) : R$ iff $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$, denoted by $\mathcal{I} \models (a, b) : R$. An interpretation $\mathcal{I}$ *satisfies* (is a *model* of) an ABox $\mathcal{A}$ iff $\mathcal{I}$ *satisfies* each assertion in $\mathcal{A}$, denoted by $\mathcal{I} \models \mathcal{A}$.

A *knowledge base* is of the form $\langle \mathcal{T}, \mathcal{A} \rangle$ where $\mathcal{T}$ is a TBox and $\mathcal{A}$ is an ABox. An interpretation $\mathcal{I}$ *satisfies* (is a *model* of, denoted by $\mathcal{I} \models \mathcal{K}$) a knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ iff $\mathcal{I}$ *satisfies* both $\mathcal{T}$ and $\mathcal{A}$. We say that a knowledge base $\mathcal{K}$ *entails* an assertion α, denoted $\mathcal{K} \models \alpha$ iff each model of $\mathcal{K}$ satisfies α. Furthermore, let $\mathcal{T}$ be a TBox and let C, D be two concept terms. We say that D *subsumes* C with respect to $\mathcal{T}$ (denoted by $C \sqsubseteq_{\mathcal{T}} D$) iff for each model of $\mathcal{T}$, $\mathcal{I} \models C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

The problem of determining whether $\mathcal{K} \models \alpha$ is called *entailment problem*; the problem of determining whether $C \sqsubseteq_{\mathcal{T}} D$ is called *subsumption problem*; and the problem of determining whether $\mathcal{K}$ is satisfiable is called *satisfiability problem*. Entailment problem and subsumption problem can be reduced to satisfiability problem.

Linear symmetric Hedge Algebra

In this section, we introduce linear symmetric Hedge Algebras (HAs). For general HAs, please refer to [12, 11, 13].

Let us consider a linguistic variable $TRUTH$ with the domain $dom(TRUTH) = \{True, False, VeryTrue, VeryFalse, MoreTrue, MoreFalse, PossiblyTrue, \ldots\}$. This domain is an infinite partially ordered set, with a natural ordering $a < b$ meaning that b describes a larger degree of truth if we consider $True > False$. This set is generated from the basic elements (*generators*) $G = \{True, False\}$ by using *hedges*, i.e., unary operations from a finite set $H = \{Very, Possibly, More\}$. The $dom(TRUTH)$ which is a set of linguistic values can be represented as

$X = \{\delta c \mid c \in G, \delta \in H^*\}$ where H^* is the Kleene star of H, From the algebraic point of view, the truth domain can be described as an abstract algebra $AX = (X, G, H, >)$.

To define relations between hedges, we introduce some notations first. We define that $H(x) = \{\sigma x \mid \sigma \in H^*\}$ for all $x \in X$. Let I be the identity hedge, i.e., $\forall x \in X.Ix = x$. The identity I is the least element. Each element of H is an *ordering operation*, i.e., $\forall h \in H$, $\forall x \in X$, either $hx > x$ or $hx < x$.

Definition 3. *[12] Let $h, k \in H$ be two hedges, for all $x \in X$ we define:*

- *h, k are* converse *if $hx < x$ iff $kx > x$;*
- *h, k are* compatible *if $hx < x$ iff $kx < x$;*
- *h modifies terms stronger or equal than k, denoted by $h \geq k$ if $hx \leq kx \leq x$ or $hx \geq kx \geq x$;*
- *$h > k$ if $h \geq k$ and $h \neq k$;*
- *h is* positive wrt k *if $hkx < kx < x$ or $hkx > kx > x$;*
- *h is* negative wrt k *if $kx < hkx < x$ or $kx > hkx > x$.*

$\mathcal{ALC}_{\mathcal{FL}}$ only considers symmetric HAs, i.e., there are exactly two generators as in the example $G = \{True, False\}$. Let $G = \{c^+, c^-\}$ where $c^+ > c^-$. c^+ and c^- are called *positive* and *negative generators* respectively. Because there are only two generators, the relations presented in Definition 3 divides the set H into two subsets $H^+ = \{h \in H \mid hc^+ > c^+\}$ and $H^- = \{h \in H \mid hc^+ < c^+\}$, i.e., every operation in H^+ is converse w.r.t. any operation in H^- and vice-versa, and the operations in the same subset are compatible with each other.

Definition 4. *[7] An abstract algebra $AX = (X, G, H, >)$, where $H \neq \emptyset, G = \{c^+, c^-\}$ and $X = \{\sigma c \mid c \in G, \sigma \in H^*\}$ is called a* linear symmetric hedge algebra *if it satisfies the properties* (A1)-(A5).

(A1) Every hedge in H^+ is a converse operation of all operations in H^-.
(A2) Each hedge operation is either positive or negative w.r.t. the others, including itself.
(A3) The sets $H^+ \cup \{I\}$ and $H^- \cup \{I\}$ are linearly ordered with the I.
(A4) If $h \neq k$ and $hx < kx$ then $h'hx < k'kx$, for all $h, k, h', k' \in H$ and $x \in X$.
(A5) If $u \notin H(v)$ and $u \leq v$ $(u \geq v)$ then $u \leq hv$ $(u \geq hv)$, for any hedge h and $u, v \in X$.

Let $AX = (X, G, H, >)$ be a linear symmetric hedge algebra and $c \in G$. We define that, $\bar{c} = c^+$ if $c = c^-$ and $\bar{c} = c^-$ if $c = c^+$. Let $x \in X$ and $x = \sigma c$, where $\sigma \in H^*$. The *contradictory element* to x is $y = \sigma\bar{c}$ written $y = -x$.

[12] gave us the following proposition to compare elements in X.

Proposition 5 *Let $AX = (X, G, H, >)$ be a linear symmetric HA, $x = h_n \cdots h_1 u$ and $y = k_m \cdots k_1 u$ are two elements of X where $u \in X$. Then there exists an index $j \leq \min\{n, m\} + 1$ such that $h_i = k_i$ for all $i < j$, and*

(i) *$x < y$ iff $h_j x_j < k_j x_j$, where $x_j = h_{j-1} \cdots h_1 u$;*

(ii) $x = y$ *iff* $n = m = j$ *and* $h_j x_j = k_j x_j$.

In order to define the semantics of the hedge modification, we only consider monotonic HAs defined in [7] which also extended the order relation on $H^+ \cup \{I\}$ and $H^- \cup \{I\}$ to one on $H \cup \{I\}$. We will use "hedge algebra" instead of "linear symmetric hedge algebra" in the rest of this paper.

Inverse mapping of hedges

Fuzzy description logics represent the assessment "It is true that Tom is very old" by

$$(VeryOld)^{\mathcal{I}}(Tom)^{\mathcal{I}} = True. \tag{1}$$

In a fuzzy linguistic logic [19–21], the assessment "It is true that Tom is very old" and the assessment "It is very true that Tom is old" are equivalent, which means

$$(Old)^{\mathcal{I}}(Tom)^{\mathcal{I}} = VeryTrue, \tag{2}$$

and (1) has the same meaning. This signifies that the modifier can be moved from concept term to truth value and vice versa. For any $h \in H$ and for any $\sigma \in H^*$, the rules of moving hedges [11] are as follows,

$$RT1 : (hC)^{\mathcal{I}}(d) = \sigma c \rightarrow (C)^{\mathcal{I}}(d) = \sigma hc$$
$$RT2 : (C)^{\mathcal{I}}(d) = \sigma hc \rightarrow (hC)^{\mathcal{I}}(d) = \sigma c.$$

where C is a concept term and $d \in \Delta^{\mathcal{I}}$.

Definition 6. *[7] Consider a monotonic HA* $AX = (X, \{c^+, c^-\}, H, >)$ *and a* $h \in H$. *A mapping* $h^- : X \rightarrow X$ *is called an* inverse mapping *of* h *iff it satisfies the following two properties,*

1. $h^-(\sigma hc) = \sigma c$.
2. $\sigma_1 c_1 > \sigma_2 c_2 \Leftrightarrow h^-(\sigma_1 c_1) > h^-(\sigma_2 c_2)$.

where $c, c_1, c_2 \in G$, $h \in H$ *and* $\sigma_1, \sigma_2 \in H^*$.

$\mathcal{ALC}_{\mathcal{FL}}$

$\mathcal{ALC}_{\mathcal{FL}}$ is a Description Logic in which the truth domain of interpretations is represented by a hedge algebra. The syntax of $\mathcal{ALC}_{\mathcal{FL}}$ is similar to that of $\mathcal{ALCH}$ except that $\mathcal{ALC}_{\mathcal{FL}}$ allows concept modifiers and does not include role hierarchy.

Definition 7. *Let* H *be a set of hedges. Let* A *be a concept name and* R *a role, complex concept terms denoted by* C, D *in* $\mathcal{ALC}_{\mathcal{FL}}$ *are formed according to the following syntax rule:*

$$A|\top|\bot|C \sqcap D|C \sqcup D|\neg C|\delta C|\forall R.C|\exists R.C$$

where $\delta \in H^*$.

In [13], HAs are extended by adding two artificial hedges inf and sup defined as $\inf(x) = \text{ infimum}(H(x))$, $\sup(x) = \text{supremum}(H(x))$. If $H = \emptyset$, $H(c^+)$ and $H(c^-)$ are infinite, according to [13] $\inf(c^+) = \sup(c^-)$. Let $W = \inf(True) = \sup(False)$ and let $\sup(True)$ and $\inf(False)$ be the greatest and the least elements of X respectively.

The semantics is based on the notion of interpretations.

Definition 8. *Let AX be a monotonic HA such that $AX = (X, \{True, False\}, H, >)$. A* fuzzy interpretation (ɟ-interpretation) *$\mathcal{I}$ for $\mathcal{ALC}_{\mathcal{FL}}$ is a pair $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a nonempty set and $\cdot^{\mathcal{I}}$ is an* interpretation function *mapping:*

- *individuals to elements in $\Delta^{\mathcal{I}}$;*
- *a concept C into a function $C^{\mathcal{I}} : \Delta^{\mathcal{I}} \to X$;*
- *a role R into a function $R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to X$.*

For all $d \in \Delta^{\mathcal{I}}$ the interpretation function satisfies the following equations

$$\begin{aligned}
\top^{\mathcal{I}}(d) &= \sup(True), \\
\bot^{\mathcal{I}}(d) &= \inf(False), \\
(\neg C)^{\mathcal{I}}(d) &= -C^{\mathcal{I}}(d), \\
(C \sqcap D)^{\mathcal{I}}(d) &= \min(C^{\mathcal{I}}(d), D^{\mathcal{I}}(d)), \\
(C \sqcup D)^{\mathcal{I}}(d) &= \max(C^{\mathcal{I}}(d), D^{\mathcal{I}}(d)), \\
(\delta C)^{\mathcal{I}}(d) &= \delta^{-}(C^{\mathcal{I}}(d)), \\
(\forall R.C)^{\mathcal{I}}(d) &= \inf_{d' \in \Delta^{\mathcal{I}}}\{\max(-R^{\mathcal{I}}(d, d'), C^{\mathcal{I}}(d'))\}, \\
(\exists R.C)^{\mathcal{I}}(d) &= \sup_{d' \in \Delta^{\mathcal{I}}}\{\min(R^{\mathcal{I}}(d, d'), C^{\mathcal{I}}(d'))\},
\end{aligned}$$

where $-x$ is the contradictory element of x, and δ^- is the inverse of the hedge chain δ.

Definition 9. *A* fuzzy assertion (ɟassertion) *is an expression of the form $\langle \alpha \bowtie \sigma c \rangle$ where α is of the form $a : C$ or $(a, b) : R$, $\bowtie \in \{\geq, >, \leq, <\}$ and $\sigma c \in X$.*

Formally, an ɟ-interpretation $\mathcal{I}$ *satisfies* a fuzzy assertion $\langle a : C \geq \sigma c \rangle$ (respectively $\langle (a, b) : R \geq \sigma c \rangle$) iff $C^{\mathcal{I}}(a^{\mathcal{I}}) \geq \sigma c$ (respectively $R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) \geq \sigma c$). An ɟ-interpretation $\mathcal{I}$ *satisfies* a fuzzy assertion $\langle a : C \leq \sigma c \rangle$ (respectively $\langle (a, b) : R \leq \sigma c \rangle$) iff $C^{\mathcal{I}}(a^{\mathcal{I}}) \leq \sigma c$ (respectively $R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) \leq \sigma c$). Similarly for $>$ and $<$.

Concerning terminological axioms, an $\mathcal{ALC}_{\mathcal{FL}}$ terminology axiom is of the form $C \sqsubseteq D$, where C and D are $\mathcal{ALC}_{\mathcal{FL}}$ concept terms. From a semantics point of view, a ɟ-interpretation $\mathcal{I}$ *satisfies* a fuzzy concept inclusion $C \sqsubseteq D$ iff $\forall d \in \Delta^{\mathcal{I}}. C^{\mathcal{I}}(d) \leq D^{\mathcal{I}}(d)$. Two concept terms C, D are said to be *equivalent*, denoted by $C \equiv D$ iff $C^{\mathcal{I}} = D^{\mathcal{I}}$ for all ɟ-interpretations $\mathcal{I}$. Some properties concerning the hedge modification are showed in the following proposition [7].

Proposition 10 *We have the following semantical equivalence:*

$$\begin{aligned}
\delta(C \sqcap D) &\equiv \delta(C) \sqcap \delta(D) \\
\delta(C \sqcup D) &\equiv \delta(C) \sqcup \delta(D) \\
\delta_1(\delta_2 C) &\equiv (\delta_1 \delta_2) C.
\end{aligned}$$

A *fuzzy knowledge base* (ƒ*KB)* is $\langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T}$ and $\mathcal{A}$ are finite sets of terminological axioms and ƒassertions respectively.

Example 11 *A* ƒ*KB* $\mathfrak{f}\mathcal{K} = \langle \{A \sqsubseteq \forall R.\neg B\}, \{a : \forall R.C \geq \mathit{VeryTrue}\} \rangle$.

An ƒ-interpretation $\mathcal{I}$ *satisfies* (is a *model* of) a TBox $\mathcal{T}$ iff $\mathcal{I}$ satisfies each element in $\mathcal{T}$. $\mathcal{I}$ *satisfies* (is a *model* of) an ABox $\mathcal{A}$ iff $\mathcal{I}$ satisfies each element in $\mathcal{A}$. $\mathcal{I}$ *satisfies* (is a *model* of) a ƒKB $\mathfrak{f}\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ iff $\mathcal{I}$ satisfies both $\mathcal{A}$ and $\mathcal{T}$. Given a ƒKB $\mathfrak{f}\mathcal{K}$ and a ƒassertion $\mathfrak{f}\alpha$. We say that $\mathfrak{f}\mathcal{K}$ *entails* $\mathfrak{f}\alpha$ (denoted $\mathfrak{f}\mathcal{K} \models \mathfrak{f}\alpha$) iff each model of $\mathfrak{f}\mathcal{K}$ satisfies $\mathfrak{f}\alpha$.

3 Transforming $\mathcal{ALC}_{\mathcal{FL}}$ into $\mathcal{ALCH}$

We will introduce a satisfiability preserving transformation from $\mathcal{ALC}_{\mathcal{FL}}$ into $\mathcal{ALCH}$ in this section. First, we illustrate the basic idea which is similar to the one in [17] which is the first efforts in this direction. There is also other more efficient representation in [3].

Consider a monotonic HA $AX = (X, \{\mathit{True}, \mathit{False}\}, H, >)$. In the following, we assume that $c \in \{c^+, c^-\}$ where $c^+ = \mathit{True}, c^- = \mathit{False}$, $\sigma \in H^*$, $\sigma c \in X$ and $\bowtie \in \{\geq, >, \leq, <\}$. Assume we have an $\mathcal{ALC}_{\mathcal{FL}}$ knowledge base, $\mathfrak{f}\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{A} = \{\mathfrak{f}\alpha_1, \mathfrak{f}\alpha_2, \mathfrak{f}\alpha_3, \mathfrak{f}\alpha_4\}$ and $\mathfrak{f}\alpha_1 = \langle a : A \geq \mathit{True} \rangle$, $\mathfrak{f}\alpha_2 = \langle b : A \geq \mathit{VeryTrue} \rangle$, $\mathfrak{f}\alpha_3 = \langle a : B \leq \mathit{False} \rangle$, and $\mathfrak{f}\alpha_4 = \langle b : B \leq \mathit{VeryFalse} \rangle$ where A, B are concept names. We introduce four new concept names: $A_{\geq \mathit{True}}$, $A_{\geq \mathit{VeryTrue}}$, $B_{\leq \mathit{False}}$ and $B_{\leq \mathit{VeryFalse}}$. The concept name $A_{\geq \mathit{True}}$ represents the set of individuals that are instances of A with degree greater and equal to *True*. The concept name $B_{\leq \mathit{VeryFalse}}$ represents the set of individuals that are instances of B with degree less and equal to *VeryFalse*. We can map the fuzzy assertions into classical assertions:

$\langle a : A \geq \mathit{True} \rangle \rightarrow \langle a : A_{\geq \mathit{True}} \rangle$,
$\langle b : A \geq \mathit{VeryTrue} \rangle \rightarrow \langle b : A_{\geq \mathit{VeryTrue}} \rangle$,
$\langle a : B \leq \mathit{False} \rangle \rightarrow \langle a : B_{\leq \mathit{False}} \rangle$,
$\langle b : B \leq \mathit{VeryFalse} \rangle \rightarrow \langle b : B_{\leq \mathit{VeryFalse}} \rangle$.

We also need to consider the relationships among the newly introduced concept names. Because $\mathit{VeryTrue} > \mathit{True}$, it is easy to get if a truth value $\sigma c \geq \mathit{VeryTrue}$ then $\sigma c \geq \mathit{True}$. Thus, we obtain a new inclusion $A_{\geq \mathit{VeryTrue}} \sqsubseteq A_{\geq \mathit{True}}$. Similarly for B, because $\mathit{VeryFalse} < \mathit{False}$, a truth value $\sigma c \leq \mathit{VeryFalse}$ implies $\sigma c \leq \mathit{False}$ too. Then the inclusion $B_{\leq \mathit{VeryFalse}} \sqsubseteq B_{\leq \mathit{False}}$ is obtained.

Now, let us proceed with the mappings. Let $\mathfrak{f}\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be an $\mathcal{ALC}_{\mathcal{FL}}$ knowledge base. We are going to transform $\mathfrak{f}\mathcal{K}$ into an $\mathcal{ALCH}$ knowledge base $\mathcal{K}$. We assume $\sigma c \in [\inf(\mathit{False}), \sup(\mathit{True})]$ and $\bowtie \in \{\geq, >, \leq, <\}$.

The transformation of ABox

In order to transform $\mathcal{A}$, we define two mappings θ and ρ to map all the assertions in $\mathcal{A}$ into classical assertions. Notice that we do not allow assertions of the forms

$(a,b) : R < \sigma c$ and $(a,b) : R \leq \sigma c$ although they are legal forms of assertions in $\mathcal{ALC}_{\mathcal{FL}}$ because they related to 'negated role' which is not part of classical $\mathcal{ALCH}$.

We use the mapping ρ to encode the basic idea we present at the beginning of this section. The mapping ρ combines the $\mathcal{ALC}_{\mathcal{FL}}$ concept term, the $\bowtie$ and the fuzzy value σc together into one $\mathcal{ALCH}$ concept term.

Let A be a concept name, C, D be concept terms and R be a role name. For roles we have simply

$$\rho(R, \bowtie \sigma c) = R_{\bowtie \sigma c}.$$

For concept terms, the mapping ρ is inductively defined on the structures of concept terms:
For $\top$,

$$\rho(\top, \bowtie \sigma c) = \begin{cases} \top \text{ if } \bowtie \sigma c = \geq \sigma c \\ \top \text{ if } \bowtie \sigma c = > \sigma c, \sigma c < \sup(c^+) \\ \bot \text{ if } \bowtie \sigma c = > \sup(c^+) \\ \top \text{ if } \bowtie \sigma c = \leq \sup(c^+) \\ \bot \text{ if } \bowtie \sigma c = \leq \sigma c, \sigma c < \sup(c^+) \\ \bot \text{ if } \bowtie \sigma c = < \sigma c. \end{cases}$$

For $\bot$,

$$\rho(\bot, \bowtie \sigma c) = \begin{cases} \top \text{ if } \bowtie \sigma c = \geq \inf(c^-) \\ \bot \text{ if } \bowtie \sigma c = \geq \sigma c, \sigma c > \inf(c^-) \\ \bot \text{ if } \bowtie \sigma c = > \sigma c \\ \top \text{ if } \bowtie \sigma c = \leq \sigma c \\ \top \text{ if } \bowtie \sigma c = < \sigma c, \sigma c > \inf(c^-) \\ \bot \text{ if } \bowtie \sigma c = < \inf(c^-). \end{cases}$$

For concept name A,

$$\rho(A, \bowtie \sigma c) = A_{\bowtie \sigma c}.$$

For concept conjunction $C \sqcap D$,

$$\rho(C \sqcap D, \bowtie \sigma c) = \begin{cases} \rho(C, \bowtie \sigma c) \sqcap \rho(D, \bowtie \sigma c) \text{ if } \bowtie \in \{\geq, >\} \\ \rho(C, \bowtie \sigma c) \sqcup \rho(D, \bowtie \sigma c) \text{ if } \bowtie \in \{\leq, <\}. \end{cases}$$

For concept disjunction $C \sqcup D$,

$$\rho(C \sqcup D, \bowtie \sigma c) = \begin{cases} \rho(C, \bowtie \sigma c) \sqcup \rho(D, \bowtie \sigma c) \text{ if } \bowtie \in \{\geq, >\} \\ \rho(C, \bowtie \sigma c) \sqcap \rho(D, \bowtie \sigma c) \text{ if } \bowtie \in \{\leq, <\}. \end{cases}$$

For concept negation $\neg C$,

$$\rho(\neg C, \bowtie \sigma c) = \rho(C, \neg \bowtie \sigma \bar{c}),$$

where $\neg \geq = \leq, \neg > = <, \neg \leq = \geq, \neg < = >$.

For modifier concept δC,

$$\rho(\delta C, \bowtie \sigma c) = \rho(C, \bowtie \sigma\delta c).$$

For existential quantification $\exists R.C$,

$$\rho(\exists R.C, \bowtie \sigma c) = \begin{cases} \exists\rho(R, \bowtie \sigma c).\rho(C, \bowtie \sigma c) \text{ if } \bowtie \in \{\geq, >\} \\ \forall\rho(R, - \bowtie \sigma c).\rho(C, \bowtie \sigma c) \text{ if } \bowtie \in \{\leq, <\}, \end{cases}$$

where $- \leq = >$ and $- < = \geq$.

For universal quantification $\forall R.C$,

$$\rho(\forall R.C, \bowtie \sigma c) = \begin{cases} \forall\rho(R, + \bowtie \sigma\bar{c}).\rho(C, \bowtie \sigma c) \text{ if } \bowtie \in \{\geq, >\} \\ \exists\rho(R, \neg \bowtie \sigma\bar{c}).\rho(C, \bowtie \sigma c) \text{ if } \bowtie \in \{\leq, <\}, \end{cases}$$

where $+ \geq = >$ and $+ > = \geq$.

θ maps fuzzy assertions into classical assertions using ρ. Let $\mathfrak{f}\alpha$ be a fassertion in $\mathcal{A}$, we define it as follows.

$$\theta(\mathfrak{f}\alpha) = \begin{cases} a : \rho(C, \bowtie \sigma c) \text{ if } \mathfrak{f}\alpha = \langle a : C \bowtie \sigma c\rangle \\ (a, b) : \rho(R, \bowtie \sigma c) \text{ if } \mathfrak{f}\alpha = \langle (a, b) : R \bowtie \sigma c\rangle. \end{cases}$$

Example 12 *Let* $\mathfrak{f}\alpha = \langle a : Very(A \sqcap B) \leq LessFalse\rangle$, *then*

$$\begin{aligned} \theta(\mathfrak{f}\alpha) &= a : \rho(Very(A \sqcap B), \leq LessFalse) \\ &= a : \rho((A \sqcap B), \leq LessVeryFalse) \\ &= a : \rho(A, \leq LessVeryFalse) \sqcup \rho(B, \leq LessVeryFalse) \\ &= a : A_{\leq LessVeryFalse} \sqcup B_{\leq LessVeryFalse}. \end{aligned}$$

We extend θ to a set of fassertions $\mathcal{A}$ point-wise,

$$\theta(\mathcal{A}) = \{\theta(\mathfrak{f}\alpha) \mid \mathfrak{f}\alpha \in \mathcal{A}\}.$$

According to the rules above, we can see that $|\theta(\mathcal{A})|$ is linearly bounded by $|\mathcal{A}|$.

4 The transformation of TBox

The new TBox is a union of two terminologies. One is the newly introduced TBox (denoted by $\mathcal{T}(N^{\mathfrak{f}\mathcal{K}})$ which is the terminology relating to the newly introduced concept names and role names. The other one is $\kappa(\mathfrak{f}\mathcal{K}, \mathcal{T})$ which is reduced by a mapping κ from the TBox of an $\mathcal{ALC}_{\mathcal{FL}}$ knowledge base.

The newly introduced TBox

Many new concept names and new role names are introduced when we transform an ABox. We need a set of terminological axioms to define the relationships among those new names.

We need to collect all the linguist terms σc that might be the subscript of a concept name or a role name. It means that not only the set of linguistic terms that appears in the original ABox but also the set of new linguist terms which

are produced by applying the ρ for modifier concepts should be included. Let A be a concept name, R be a role name.

$$X^{\mathfrak{f}\mathcal{K}} = \{\sigma c \mid \langle \alpha \bowtie \sigma c\rangle \in \mathcal{A}\} \cup \ \{\sigma\delta c \mid \rho(\delta C, \bowtie \sigma c) = \rho(C, \bowtie \sigma\delta c)\}.$$

such that δC occurs in $\mathfrak{f}\mathcal{K}$.
We define a sorted set of linguistic terms,

$$N^{\mathfrak{f}\mathcal{K}} = \{\inf(\mathit{False}), W, \sup(\mathit{True})\} \cup \ X^{\mathfrak{f}\mathcal{K}} \cup \{\sigma\bar{c} \mid \sigma c \in X^{\mathfrak{f}\mathcal{K}}\} = \{n_1, \ldots, n_{|N^{\mathfrak{f}\mathcal{K}}|}\}$$

where $n_i < n_{i+1}$ for $1 \leq i \leq |N^{\mathfrak{f}\mathcal{K}}| - 1$ and $n_1 = \inf(\mathit{False}), n_{|N^{\mathfrak{f}\mathcal{K}}|} = \sup(\mathit{True})$.

Let $\mathcal{T}(N^{\mathfrak{f}\mathcal{K}})$ be the set of terminological axioms relating to the newly introduced concept names and role names.

Definition 13. *Let $\mathcal{A}^{\mathfrak{f}\mathcal{K}}$ and $\mathcal{R}^{\mathfrak{f}\mathcal{K}}$ be the sets of concept names and role names occurring in $\mathfrak{f}\mathcal{K}$ respectively. For each $A \in \mathcal{A}^{\mathfrak{f}\mathcal{K}}$, for each $R \in \mathcal{R}^{\mathfrak{f}\mathcal{K}}$, for each $1 \leq i \leq |N^{\mathfrak{f}K}| - 1$ and for each $2 \leq j \leq |N^{\mathfrak{f}K}|$, $\mathcal{T}(N^{\mathfrak{f}\mathcal{K}})$ contains*

$$\begin{array}{rcl}
A_{\geq n_{i+1}} \sqsubseteq A_{>n_i}, & & A_{>n_i} \sqsubseteq A_{\geq n_i}, \\
A_{<n_j} \sqsubseteq A_{\leq n_j}, & & A_{\leq n_i} \sqsubseteq A_{<n_{i+1}}, \\
A_{\geq n_j} \sqcap A_{<n_j} \sqsubseteq \bot & , & \top \sqsubseteq A_{\geq n_j} \sqcup A_{<n_j}, \\
A_{>n_i} \sqcap A_{\leq n_i} \sqsubseteq \bot & , & \top \sqsubseteq A_{>n_i} \sqcup A_{\leq n_i}, \\
R_{\geq n_{i+1}} \sqsubseteq R_{>n_i}, & & R_{>n_i} \sqsubseteq R_{\geq n_i}.
\end{array}$$

where $n \in N^{\mathfrak{f}\mathcal{K}}$.

$n_{i+1} > n_i$ because $N^{\mathfrak{f}\mathcal{K}}$ is a sorted set. Then if an individual is an instance of a concept name with degree $\geq n_{i+1}$ then the degree is also $> n_i$. The first terminological axiom shows that if an individual is an instance of $A_{\geq n_{i+1}}$ then it is an instance of $A_{>n_i}$ as well. Similarly, if an individual is an instance of a concept name with degree $\leq n_i$ then the degree is also $< n_{i+1}$. The third terminological axiom shows that if an individual is an instance of $A_{\leq n_i}$ then it is also an instance of $A_{<n_{i+1}}$. $A_{\geq n_j} \sqcap A_{<n_j} \sqsubseteq \bot$ because there is no individual such that it is an instance of a concept name with degree $\geq n_j$ and with degree $< n_j$ at the same time.

$\mathcal{T}(N^{\mathfrak{f}\mathcal{K}})$ contains $8|\mathcal{A}^{\mathfrak{f}\mathcal{K}}|(|\mathcal{N}^{\mathfrak{f}\mathcal{K}}| - 1)$ plus $2|\mathcal{R}^{\mathfrak{f}\mathcal{K}}|(|\mathcal{N}^{\mathfrak{f}\mathcal{K}}| - 1)$ terminological axioms.

The mapping κ

κ maps the fuzzy TBox into the classical TBox.

Definition 14. *Let C, D be two concept terms and $C \sqsubseteq D \in \mathcal{T}$. For all $n \in N^{\mathfrak{f}\mathcal{K}}$*

$$\begin{array}{r}
\kappa(\mathfrak{f}\mathcal{K}, C \sqsubseteq D) = \bigcup_{n \in N^{\mathfrak{f}\mathcal{K}}, \bowtie \in \{\geq, >\}} \{\rho(C, \bowtie n) \sqsubseteq \rho(D, \bowtie n)\} \\
\bigcup_{n \in N^{\mathfrak{f}\mathcal{K}}, \bowtie \in \{\leq, <\}} \{\rho(D, \bowtie n) \sqsubseteq \rho(C, \bowtie n)\}
\end{array} \tag{3}$$

We extend κ to a terminology $\mathcal{T}$ point-wise. For all $\tau \in \mathcal{T}$

$$\kappa(\mathfrak{f}\mathcal{K}, \mathcal{T}) = \cup_{\tau \in \mathcal{T}} \kappa(\mathfrak{f}\mathcal{K}, \tau).$$

The satisfiability preserving theorem

Now we can define the *reduction* of $\mathfrak{f}\mathcal{K}$ into an $\mathcal{ALCH}$ knowledge base, denoted $\mathcal{K}(\mathfrak{f}\mathcal{K})$,

$$\mathcal{K}(\mathfrak{f}\mathcal{K}) = \langle \mathcal{T}(N^{\mathfrak{f}\mathcal{K}}) \cup \kappa(\mathfrak{f}\mathcal{K}, \mathcal{T}), \theta(\mathcal{A}) \rangle.$$

The transformation can be done in polynomial time. The soundness and completeness of the algorithm is guaranteed by the following satisfiability preserving theorem.

Theorem 15 *Let $\mathfrak{f}\mathcal{K}$ be an $\mathcal{ALC}_{\mathcal{FL}}$ knowledge base. Then $\mathfrak{f}\mathcal{K}$ is satisfiable iff the $\mathcal{ALCH}$ knowledge base $\mathcal{K}(\mathfrak{f}\mathcal{K})$ is satisfiable.*

Proof. Please refer to my thesis [18] which can be download from my homepage.[2]

5 Discussion

In this paper, we have presented a satisfiability preserving transformation of $\mathcal{ALC}_{\mathcal{FL}}$ into $\mathcal{ALCH}$ which is with general TBox and role hierarchy. Since all other reasoning tasks such as entailment problem and subsumption problem can be reduced to satisfiability problem, this result allows for algorithms and complexity results that were found for $\mathcal{ALCH}$ to be applied to $\mathcal{ALC}_{\mathcal{FL}}$.

As for the complexity of the transformation, we know the fact that $|\theta(\mathcal{A})|$ is linearly bounded by $|\mathcal{A}|$, $|\mathcal{T}(N^{\mathfrak{f}\mathcal{K}})| = 8|\mathcal{A}^{\mathfrak{f}\mathcal{K}}|(|\mathcal{N}^{\mathfrak{f}\mathcal{K}}| - 1) + 2|\mathcal{R}^{\mathfrak{f}\mathcal{K}}|(|\mathcal{N}^{\mathfrak{f}\mathcal{K}}| - 1)$ and $\kappa(\mathfrak{f}\mathcal{K}, \mathcal{T})$ contains at most $4|\mathcal{T}||N^{\mathfrak{f}\mathcal{K}}|$. Therefore, the resulted classical knowledge base (at most polynomial size) can be constructed in polynomial time.

There exist some reasoners for fuzzy DLs, e.g. *FiRE* [15], *GURDL* [5], *DeLorean* [2], *GERDS* [6], *YADLR* [9] and *fuzzyDL* [4]. Among them, *fuzzyDL* allows modifiers defined in terms of linear hedges and triangular functions and *DeLorean* supports triangularly-modified concept. So if we can transform variety of fuzzy DLs into classical DLs then we can use the already existing DL systems to do the reasoning of fuzzy DLs.

6 Acknowledgments

Thank Pascal Hiltzler and Martin Caminada for their comments on this paper.

References

1. Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
2. Fernando Bobillo, Miguel Delgado, and Juan Gómez-Romero. Optimizing the crisp representation of the fuzzy description logic SROIQ. In *URSW*, 2007.

[2] http://icr.uni.lu/ yining

3. Fernando Bobillo, Miguel Delgado, Juan Gómez-Romero, and Umberto Straccia. Fuzzy description logics under gödel semantics. *Int. J. Approx. Reasoning*, 50(3):494–514, 2009.
4. Fernando Bobillo and Umberto Straccia. fuzzyDL: An expressive fuzzy description logic reasoner. In *2008 International Conference on Fuzzy Systems (FUZZ-08)*. IEEE Computer Society, 2008.
5. Volker Haarslev, Hsueh-Ieng Pai, and Nematollaah Shiri. Optimizing tableau reasoning in alc extended with uncertainty. In *Description Logics*, 2007.
6. Hashim Habiballa. Resolution strategies for fuzzy description logic. In *EUSFLAT Conf. (2)*, pages 27–36, 2007.
7. Steffen Hölldobler, Dinh-Khac Dzung, and Tran Dinh-Khang. The fuzzy linguistic description logic $\mathcal{ALC}_{\mathcal{FL}}$. In *Proceedings of the Eleventh International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*, pages 2096–2103, 2006.
8. Steffen Hölldobler, Hans-Peter Störr, and Dinh Khang Tran. The fuzzy description logic $\mathcal{ALC}_{\mathcal{FH}}$ with hedge algebras as concept modifiers. *Journal of Advanced Computational Intelligence and Intelligent Informatics (JACIII)*, 7(3):294–305, 2003.
9. Stasinos Konstantopoulos and Georgios Apostolikas. Fuzzy-dl reasoning over unknown fuzzy degrees. In *OTM Workshops (2)*, pages 1312–1318, 2007.
10. Thomas Lukasiewicz and Umberto Straccia. Managing uncertainty and vagueness in description logics for the semantic web. *Journal of Web Semantics*, 6:291–308, 2008.
11. Cat-Ho Nguyen, Dinh-Khang Tran, Van-Nam Huynh, and Hai-Chau Nguyen. Hedge algebras, linguistic-valued logic and their application to fuzzy reasoning. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 7(4):347–361, 1999.
12. Cat-Ho Nguyen and W. Wechler. Hedge algebras: an algebraic approach to structure of sets of linguistic truth values. *Fuzzy Sets and Systems*, 35(3):281–293, 1990.
13. Cat-Ho Nguyen and W. Wechler. Extended hegde algebras and their application to fuzzy logic. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 52:259–281, 1992.
14. H.-N. Nguyen S. Hölldobler and D.-K. Tran. The fuzzy description logic $\mathcal{ALC}_{\mathcal{FLH}}$. In *Proc. 9th IASTED International Conference on Artificial Intelligence and Soft Computing*, pages 99–104, 2005.
15. N. Simou and S. Kollias. Fire : A fuzzy reasoning engine for impecise knowledge. K-Space PhD Students Workshop, Berlin, Germany, 14 September 2007, 2007.
16. Umberto Straccia. Reasoning within fuzzy description logics. *JAIR*, 14:137–166, 2001.
17. Umberto Straccia. Transforming fuzzy description logics into classical description logics. In *Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA-04)*, number 3229 in Lecture Notes in Computer Science, pages 385–399, Lisbon, Portugal, 2004. Springer Verlag.
18. Yining Wu. Transforming fuzzy description logic $\mathcal{ALC}_{\mathcal{FL}}$ into classical description logic $\mathcal{ALCH}$. *Master Thesis*, 2007.
19. Lotfi A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning - I. *Information Sciences*, 8(3):199–249, 1975.
20. Lotfi A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning - II. *Information Sciences*, 8(4):301–357, 1975.
21. Lotfi A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning- III. *Information Sciences*, 9(1):43–80, 1975.

PrOntoLearn: Unsupervised Lexico-Semantic Ontology Generation using Probabilistic Methods

Saminda Abeyruwan[1], Ubbo Visser[1], Vance Lemmon[2], and Stephan Schürer[3]

[1] Department of Computer Science, University of Miami, Florida, USA
{saminda,visser}@cs.miami.edu
[2] The Miami Project to Cure Paralysis, University of Miami Miller School of Medicine, Florida, USA
vlemmon@miami.edu
[3] Department of Molecular and Cellular Pharmacology, University of Miami Miller School of Medicine, Florida, USA
sschuerer@med.miami.edu

Abstract. Formalizing an ontology for a domain manually is well-known as a tedious and cumbersome process. It is constrained by the knowledge acquisition bottleneck. Therefore, researchers developed algorithms and systems that can help to automatize the process. Among them are systems that include text corpora for the acquisition. Our idea is also based on vast amount of text corpora. Here, we provide a novel unsupervised bottom-up ontology generation method. It is based on lexico-semantic structures and Bayesian reasoning to expedite the ontology generation process. We provide a quantitative and two qualitative results illustrating our approach using a high throughput screening assay corpus and two custom text corpora. This process could also provide evidence for domain experts to build ontologies based on top-down approaches.

Keywords: Ontology Modeling, Ontology Learning, Probabilistic Methods

1 Introduction

An ontology is a formal, explicit specification of a shared conceptualization [10], [22]. Formalizing an ontology for a given domain with the supervision of domain experts is a tedious and cumbersome process. The identification of the structures and the characteristics of the domain knowledge through an ontology is a demanding task. This problem is known as the knowledge acquisition bottleneck (KAB) and a suitable solution presently does not exist.

There exists a large number of text corpora available from different domains (e.g., the BioAssay high throughput screening assays[4]) that need to be classified into ontologies to faciliate the discovery of new knowledge. A domain of discourse

[4] http://bioassayontology.org/

(i.e., sequential number of sentences) shows characteristics such as 1) redundancy 2) structured and unstructured text 3) noisy and uncertain data that provide a degree of belief 4) lexical disambiguity, and 5) semantic heterogeneity problems. We discuss in depth the importance of these characteristics in section 3. Our goal in this research is to provide a novel method to construct an ontology from the evidence collected from the corpus. In order to achieve our goal, we use the lexico-semantic features of the lexicon and probabilistic reasoning to handle the uncertainty of features. Since our method is applied to build an ontology for a corpus without domain experts, this method can be seen as an unsupervised learning technique. Since the method starts from the evidence present in the corpus, it is can be seen as a reverse engineering technique. We use WordNet[5] to handle lexico-semantic structures, and the Bayesian reasoning to handle degree of belief of an uncertain event. We implement a Java based application to serialize the learned conceptualization to OWL DL[6] format.

The rest of the paper is organized as follows: section 2 provides a broad investigation of the related work. Section 3 provides details of our research approach. Section 4 provides a detail description of the experiments based on three different text corpora and the discussion. Finally, section 5 provides the summary and the future work.

2 Related Work

The problem of learning a conceptualization from a corpus has been studied in many disciplines such as machine learning, text mining, information retrieval, natural language processing, and Semantic Web. Table 1 shows the pros and cons of different techniques to solve the problem of *ontology learning*. Each method covers some portion of the problem and each method learns the conceptualization from terms, and present it as taxonomies and axioms to an ontology. On the other hand, most of the methods use a top-down approach, i.e., an initial classification of an ontology is given. The uncertainty inherited from the domain is usually dealt with by a domain expert, and the conceptualization is normally defined using predefined rules or templates. These methods show the characteristics of a semi-supervised and a semi-automated learning paradigm.

3 Approach

Our research focuses on an unsupervised method to quantify the degree of belief that a grouping of words in the corpus will provide a substantial conceptualization of the domain of interest. The degree of belief in world states influences the uncertainty of the conceptualization. The uncertainty arises from partial observability, non-determinism, laziness and theoretical and practical ignorance [19]. The partial observability arises from the size of the corpus. Even though

[5] http://wordnet.princeton.edu/
[6] http://www.w3.org/TR/owl-guide/

Table 1. The summary of the related work. Probabilistic learning (PR), never ending language learning (NELL), discovery and aggregation of relations in text (DART), recognizing textual entailment (RTE), automated theorem proving (ATP), natural language understanding (NLU), formal concept analysis (FCA), and ontology population (OP).

Work	Purpose	T-Box	A-Box	Method
PR [9], [12], [14] and [17]	reasoning	available	available	prob. theory
NELL [3]	24×7 learning	fixed	dynamic	ML techniques
DART [7]	world knowledge	×	×	semi-automated
RTE [2], and [13]	entailment	×	×	ATP
NLU [20]	commonsense rules	×	×	semi-supervised
Text2Onto [6]	ontology learning	√	√	semi-supervised
LexO [24]	complex classes	√	×	semi-supervised
FCA [5]	taxonomy	√	×	FCA
OP [4], and [23]	ontology population	available	available	semi-/supervised

a corpus many be large, it might not contain all the necessary evidence of an event of interest. A corpus contains ambiguous statements about an event that leads to a non-determinism of the state of the event. The laziness arises from the too much work that needs to be done in order to learn exceptionless rules and it is too hard to learn such rules. The theoretical and practical ignorance arises from lack of complete evidence and it is not possible to conduct all the necessary tests to learn a particular event. Hence, the domain knowledge, and in our case the domain conceptualization, can at best provide only a degree of belief of the relevant groups of words. We use probability theory to deal with the degrees of belief. As mentioned in [19], the probability theory has the same ontological commitment as the formal logic, though the epistemological commitment differs. The process of learning and presenting a probabilistic conceptualization is divided into four phases as shown in Figure 1. They are, 1) pre-processing 2) syntactic analysis 3) semantic analysis, and 4) representation.

3.1 Pre-processing

A corpus contains a plethora of structured and unstructured sentences. A lexicon of a language is its vocabulary built from lexemes [11], [15]. A lexicon contains words belonging to a language and in our work individual words from the corpus. In pure form, the lexicon may contain words that appear frequently in the corpus but have little value in formalizing a meaningful criterion. These words are called stop words or in our terminology: negated lexicon, and they are excluded from the vocabulary. We, first, part-of-speech tagged the corpus with the Penn Treebank English POS tag set [16]. We use the subset of tagset NN, NNP, NNS, NNPS, JJ, JJR, JJS, VB, VBD, VBG, VBN, VBP, and VBZ. The word length W_L above some threshold W_{L_T} is also considered. The length of a word, with respect to

POS context, is the sequence of characters or symbols that made up the word. By default, we consider that a word with $W_L > 2$ sufficiently formalizes to some criterion.

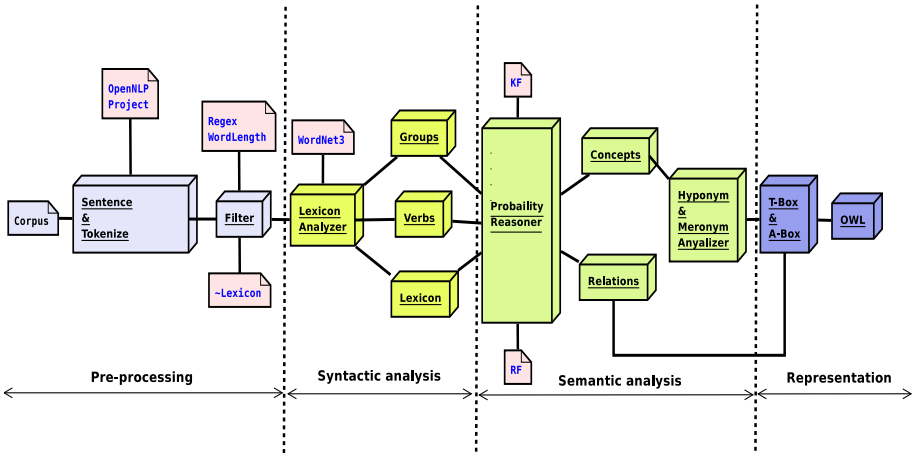

Fig. 1. Overall process: process categorizes into four phases; pre-processing, syntactic analysis, semantic analysis & representation

The pure form of the lexicon might contain words that need to be further purified according to some criterion. We use regular expressions for this task. Then we normalize and case-fold the words [15]. In addition to this there are families of derivationally related words with similar meanings. We use stemming and lemmatization to reduce the inflectional forms and derivational forms of a word to a common base form [15]. We achieve this with the aid of WordNets' stemming algorithms. We couple the knowledge of POS tag of the word to get the correct context when finding the common base form.

3.2 Syntactic Analysis

The primary focus on this phase is to look at the structure of the sentences and learn the associations among the vocabulary. We assume that each sentence of the corpus follows the POS pattern 1. 1,

$$(Subject_{NounPhrase}+)(Verb+)(Object_{NounPhrase}+) \tag{1}$$

We hypothesize that the associations learned from this phase provides the potential candidates for concepts and relations of the ontology. But the vocabulary itself does not provide sufficient ontology concepts. We use a notion of grouping of consecutive sequence of words to form an OWL concept. This grouping is done using an appropriate N-gram model [1]. We illustrate this idea using Figure 2.

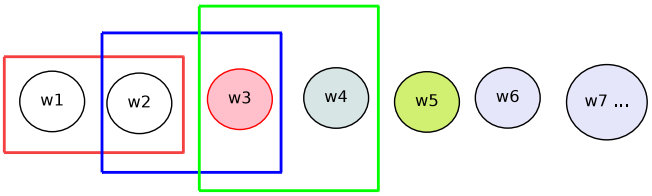

Fig. 2. An example three-gram model

The group $w_1 \circ w_2$ forms a potential concept in the conceptualization. We use the notation $x \circ y$ to show that the word y is appended to the word x. The groups $w_2 \circ w_3$, $w_3 \circ w_4$ etc. form other potential concepts in the conceptualization. Word w_3 comes after group $w_1 \circ w_2$. According to the Bayes viewpoint, we collect information to estimate the probability $P(w_3|\{w_1 \circ w_2\})$, which will be used to form IS-A relationships, $w_1 \circ w_2 \sqsubseteq w_3$ using an independent Bayesian network with conditional probability $P(\{w_1 \circ w_2\}|w_3)$. In addition to this, we count the groups appear in the left hand side and the right hand side of the expression 1 and the association of of these groups given the verbs. These counts are used in the third phase to create the relations among concepts.

3.3 Semantic Analysis

This phase conducts the semantic analysis with probabilistic reasoning, which constitutes the most important operation of our work. This phase determines the conceptualization of the domain using a probability distribution for IS-A relations and relations among the concepts. Our main definition of concept learning is given in Definition 1.

Definition 1. *The set $W = \{w_1, \ldots, w_n\}$ represents words of the vocabulary and each w_i has a prior probability $\theta_i > \tau$. τ is a prior threshold, which is known as the knowlege factor. The set $G = \{g_1, \ldots, g_m\}$ represents N-gram groups learned from the corpus and each g_j has a prior probability η_j. When $w \in W$ and $g \in G$, $P(w|g)$ is the likelihood probability π learned from the corpus. The entities w and g represent the potential concepts of the conceptualization and the set W provide the potential super-concepts of the conceptualization. Within this environment, an IS-A relationship between w and g is given by the posterior probability $P(g|w)$ and this is represented with a Bayesian network having two nodes w and g and is modeled by the equation,*

$$P(g|w) = \frac{\pi \times \eta}{\sum_i p(w|g_i) \times p(g_i)}. \tag{2}$$

Using the Definition 1, the probabilistic conceptualization of a domain is defined as follows.

Definition 2. *The probabilistic conceptualization of the domain is represented by an n-number of independent Bayesian networks sharing groups.*

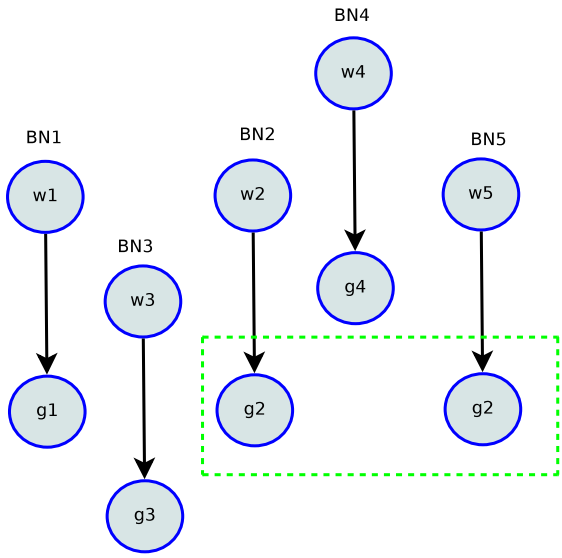

Fig. 3. w_1, w_2, w_3, w_4 and w_5 are super-concepts. g_1, g_2, g_3 and g_4 are candidate sub-concepts. There are 5 independent Bayesian networks. Bayesian networks 2 and 5 share the group g_2 when representing the concepts of the conceptualization

Figure 3 shows a simple example of the Definition 2. The interpretation of Definition 2 is: Let a set G contains an n-number of finite random variables $\{g_1, \ldots, g_n\}$. There exist a group g_i, which is shared by m words $\{w_1, \ldots, w_m\}$. Then, with respect to the Bayesian framework, BN_i of $P(g_i|w_i)$ is calculated and $max(P(g_i|m_i))$ is selected for the construction of the ontology. This means that if there exists two Bayesian networks and the Bayesian network one is given by the pair w_1, g_1 and the Bayesian network two is given by the pair $\{w_2, g_1\}$ then the Bayesian network that has the most substantial IS-A relationship is obtained through $max_{BN_i}(P(g_1|w_1))$ and this network is retained and other Bayesian networks will be ignored when building the ontology. If all $P(g_1|w_1)$ remains equal, then the Bayesian network with the highest super-concept probability will be retained. These two conditions will resolve any naming issues.

The next step is to induce the relationships to complete the conceptualization. In order to do this, we need to find semantics associated with each *verb*. We hypothesize that relations are generated by the verbs and the definition is as follows.

Definition 3. *The relationships of the conceptualization are learned from the syntactic structure model by the expression 1 and the semantic structure model by the lambda expression* $\lambda obj.\lambda sub.Verb(sub, obj)$*, where* β*-reduction is applied for obj and sub of the expression 1. If there exists a verb* V *between two groups of concepts* C_1 *and* C_2*, the relationship of the triple* (V, C_1, C_2) *is written as* $V(C_1, C_2)$ *and model with conditional probability* $P(C_1, C_2|V)$*. The Bayesian*

network for relationship is and the model semantic relationship is given by,

$$P(C_1, C_2|V) = p(C_1|V)p(C_2|V) \rightarrow V(C_1, C_2)$$

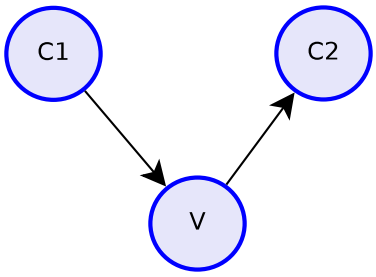

Fig. 4. Bayesian networks for relations modeling. C_1 and C_2 are groups and V is a verb

The relations learned from Defintions 3 needs to be subjected to a lower bound. The lower bound is known as the *relations factor*. When the corpus is substantially large, the number of relations is proportional to the number of verbs. Not all relations may relevant and the factor is used as the threashold. A verb may have antonyms. If a verb is associated with some concepts and these concepts happen to be associated with a antonym, the verb with the highest Bayesian probability value is selected for the relations map and the other relationship will be removed. Finally, the probabilistic conceptualization is serialized as an OWL DL ontology in the representation phase.

Our implementation of the above phases is based on Java 6 and it is named as `PrOntoLearn` (Probabilistic Ontology Learning).

4 Experiments

We have conducted experiments on three main data corpora, 1) the PCAssay, of the BioAssay Ontology (BAO) project, Department of Molecular and Cellular Pharmacology University of Miami, School of Medicine 2) a sample collection of 38 PDF files from ISWC 2009 proceedings, and 3) a substantial portion of the web pages extracted from the University of Miami, Department of Computer Science[7] domain . We have constructed ontologies for all three corpora with different parameter settings.

The first corpus contains high throughput screening assays performed on various screening centers. This corpus grows rapidly each month. We specifically limited our dataset to assays available on the 1^{st} of January 2010. Table 2 provides the statistics of the corpus. We extract the vocabulary generated

[7] http://www.cs.miami.edu

from *[a-zA-Z]+[-_]?\w** regular expression, and normalized them to create the vocabulary.

Table 2. The PCAssay (the BioAssay Ontology project) corpus statistics

Title	Statistics	Description
Documents	1,759	All documents are XHTML formated with a given template
Unique $ConceptWords$	13,017	Normalized candidate concept words from NN, NNP, NNS, JJ, JJR & JJS using [a-zA-Z]+[-_]?\w*
Unique $Verbs$	1,337	Normalized verbs from VB, VBD, VBG, VBN, VBP & VBZ using [a-zA-Z]+[-_]?\w*
Total $ConceptWords$	631,623	
Total $Verbs$	109,421	
Total Lexicon	741,044	$Lexicon = ConceptWords \bigcap Verbs$
Total $Groups$	631,623	

The average file size of the corpus is approximately 6 Kb. We conducted these experiments in a Genuine Intel(R) CPU 585 @ 2.16GHz, 32 bits, 2 Gb Toshiba laptop. It is found that the time required to build the conceptualization grows linearly. We use precision, recall and F1 measures to evaluate the ontology and recommendations from domain experts, specially to get comments on the generated bioassay ontology. The ontology that is generated is too large to show in here.Instead, we provide a few distinct snapshots of the ontology with the help of Protégé OWLViz plugin. Figures 5 and 6 show snapshots of the ontology created from the BioAssay Ontology corpus for input parameters $KF = 0.5$, N-gram $= 3$, and $RF = 0.9$. Figure 5 shows the IS-A relationships and Figure 6 shows the binary relationships.

According to experts, the ontology contains rich set of vocabulary, which is very useful for top-down ontology construction. The experts also mentioned that the ontology has good enough structure. The *www.cs.miami.edu* corpus is used to calculate quantitative measurements. The gold standard based approaches such as precision (P) and recall (R) and F-measure (F_1) are used to evaluate ontologies [8]. We use a slightly modified version of [21] as our reference ontology. Table 3 shows the results. The average precision of the constructed ontology is approximately 42%. It is to be noted that we use only one reference ontology. If we use another reference ontology the precision values varies. This means that the precision value depends on the available ground truth.

The results show that our method creates an ontology for any given domain with acceptable results. This is shown in the precision value, if the ground truth

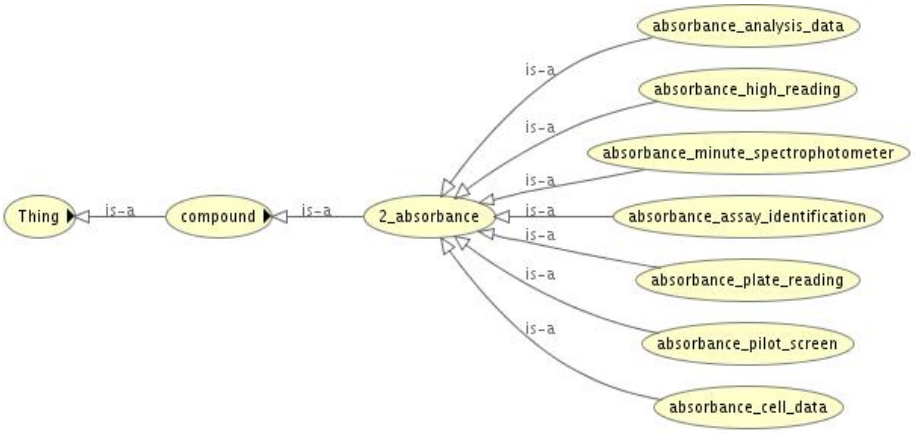

Fig. 5. An example snapshot of the BioAssay Ontology corpus with IS-A relations

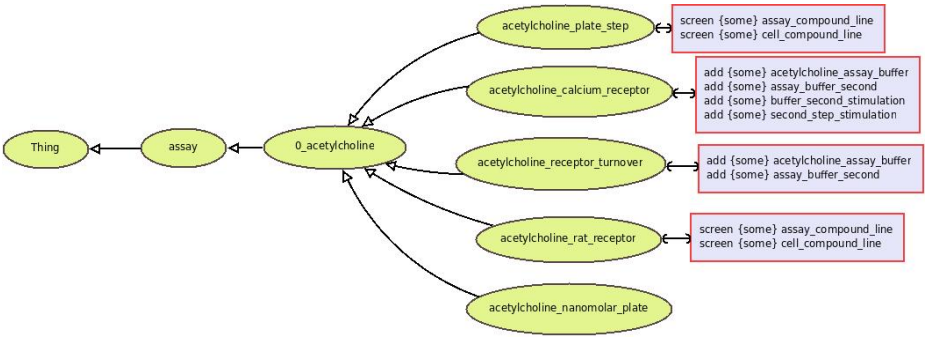

Fig. 6. An example snapshot of the BioAssay Ontology corpus with binary relations

Table 3. Precision, recall and F1 measurement for N-gram=4 and RF=1 using extended reference ontology

KF	Precision	Recall	F1
0.1	0.424	1	0.596
0.2	0.388	1	0.559
0.3	0.445	1	0.616
0.4	0.438	1	0.609
0.5	0.438	1	0.609
0.6	0.424	1	0.595
0.7	0.415	1	0.587
0.8	0.412	1	0.583
0.9	0.405	1	0.576
1.0	0.309	1	0.472

is available. On the other hand, if the domain does not have ground truth the results are subject to domain expert evaluation of the ontology. One of the potential problems we have seen in our approach is search space. Since our method is unsupervised, it tends to search the entire space for results, which is computationally costly. We thus need a better method to prune the search space so that out method provide better results. According to domain experts, our method extracts good vocabulary but provides a flat structure. They have proposed a sort of a semi-supervised approach to correct this problem, by combining the knowledge from domain experts and results produced by our system. We left the detailed investigation for future work.

Since our method is based on the Bayesian reasoning (which uses N-gram probabilities), it is paramount that the corpus contains enough evidence of the redundant information. This condition requires that the corpus to be large enough so that we can hypothesize that the corpus provides enough evidence to build the ontology.

We hypothesize that a sentence of the corpus would generally be subjected to the grammar rule given in expression 1. This constituent is the main factor that uses to build the relationships among concepts. In NLP, there are many other finer grained grammar rules that specifically fit for given sentences. If these grammar rules are used, we believe we can build a better relationship model. We have left this for future work.

At the moment our system does not distinguish between concepts and the individuals of the concepts. The learned A-Box primarily consists of the probabilities of each concepts. This is one area where we are eager to work on. Using the state-of-the art NLP techniques, we plan to fill this gap in a future work. Since our method has the potential to be used in any corpus, it could be seen that the lemmatizing and stemming algorithms that are available in WordNet would not recognize some of the words. Specially in the BioAssay corpus, we observe that some of the domain specific words are not recognized by WordNet. We use the Porter stemming algorithm [18] to get the word form and it shows that this algorithm constructs peculiar word forms. Therefore, we deliberately remove it from the processing pipeline.

The complexity of our algorithms is as follows. The bootstrapping algorithm available in the syntactic layer has a worst case running time of $O(M \times max(s_j) \times max(w_k))$, where M is the number of documents, s_j is a the number of sentences in a document, and w_k is the number of words in a sentence. The probabilistic reasoning algorithm has the worst case running time of $O(|\mathcal{L}| \times |SuperConcepts|)$, where $|\mathcal{L}|$ is the size of the lexicon and $|SuperConcepts|$ is the size of the super concepts set. The ontologies generated from the system are consistent with Pellet[8] and FaCT++[9] reasoners.

Finally, our method provides a process to create a lexico-semantic ontology for any domain. For our knowledge, this is a very first research on this line of

[8] http://clarkparsia.com/pellet
[9] http://owl.man.ac.uk/factplusplus/

work. So we continue our research along this line and to provide better results for future use.

5 Conclusion

We have introduced a novel process to generate an ontology for any random text corpus. We have shown that our process constructs a flexible ontology. It is also shown that in order to achieve high precision, it is paramount that the corpus should be large enough to extract important evidence. Our research has also shown that probabilistic reasoning on lexico-semantic structures is a powerful solution to overcome or at least mitigate the knowledge acquisition bottleneck. Our method also provides evidence to domain experts to build ontologies using a top-down approach. Though we have introduced a powerful technique to construct ontologies, we believe that there is a lot of work that can be done to improve the performance of our system. One of the areas our method lacks is the separation between concepts and individuals. We would like to use the generated ontology as a seed ontology to generate instances for the concepts and extract the individuals already classified as concepts. Finally, we would like to increase the lexicon of the system with more tags available from the Penn Treebank tag set. We believe that if we introduce more tags into the system, our system can be trained to construct human readable (friendly) concepts and relations names.

Acknowledgements

This work was partially funded by the NIH grant RC2 HG005668.

References

1. Banerjee, S., Pedersen, T.: The design, implementation and use of the n-gram statistics package. In: In Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics. pp. 370–381 (2003)
2. Bos, J., Markert, K.: Recognising textual entailment with logical inference. In: HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing. pp. 628–635. Association for Computational Linguistics, Morristown, NJ, USA (2005)
3. Carlson, A., Betteridge, J., Wang, R.C., Hruschka, Jr., E.R., Mitchell, T.M.: Coupled semi-supervised learning for information extraction. In: WSDM '10: Proceedings of the third ACM international conference on Web search and data mining. pp. 101–110. ACM, New York, NY, USA (2010)
4. Chemudugunta, C., Holloway, A., Smyth, P., Steyvers, M.: Modeling documents by combining semantic concepts with unsupervised statistical learning. In: ISWC '08: Proceedings of the 7th International Conference on The Semantic Web. pp. 229–244. Springer-Verlag, Berlin, Heidelberg (2008)
5. Cimiano, P., Hotho, A., Staab, S.: Learning concept hierarchies from text corpora using formal concept analysis. Journal of Artificial Intelligence research 24, 305–339 (2005)

6. Cimiano, P., Völker, J.: Text2Onto - A Framework for Ontology Learning and Data-driven Change Discovery (2005)
7. Clark, P., Harrison, P.: Large-scale extraction and use of knowledge from text. In: K-CAP '09: Proceedings of the fifth international conference on Knowledge capture. pp. 153–160. ACM, New York, NY, USA (2009)
8. Dellschaft, K., Staab, S.: Strategies for the evaluation of ontology learning. In: Proceeding of the 2008 conference on Ontology Learning and Population: Bridging the Gap between Text and Knowledge. pp. 253–272. IOS Press, Amsterdam, The Netherlands, The Netherlands (2008)
9. Ding, Z., Peng, Y.: A Probabilistic Extension to Ontology Language OWL. In: HICSS '04: Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 4. p. 40111.1. IEEE Computer Society, Washington, DC, USA (2004)
10. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition 5(2), 199–220 (1993)
11. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Prentice Hall, Pearson Education International, 2. edn. (2009)
12. Koller, D., Levy, A., Pfeffer, A.: P-CLASSIC: A tractable probabilistic description logic. In: Proceedings of AAAI-97. pp. 390–397 (1997)
13. Lin, D., Pantel, P.: Discovery of inference rules for question-answering. Natural Language Engineering 7(4), 343–360 (2001)
14. Lukasiewicz, T.: Probabilistic description logics for the semantic web. Tech. rep., Nr. 1843-06-05, Institut fur Informationssysteme, Technische Universitat Wien (2007)
15. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA (2008)
16. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of english: the penn treebank. Comput. Linguist. 19(2), 313–330 (1993)
17. Poon, H., Domingos, P.: In: Proceedings of the Forty-Eighth Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden. ACL (2010)
18. Porter, M.F.: An algorithm for suffix stripping. Program 14(3), 130–137 (1980)
19. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall, 3rd edn. (2009)
20. Salloum, W.: A question answering system based on conceptual graph formalism. In: KAM '09: Proceedings of the 2009 Second International Symposium on Knowledge Acquisition and Modeling. pp. 383–386. IEEE Computer Society, Washington, DC, USA (2009)
21. SHOE: Example computer science department ontology, `http://www.cs.umd.edu/projects/plus/SHOE/cs.html`, last visited on June 14, 2010
22. Studer, R., Benjamins, V.R., Fensel, D.: Knowledge engineering: Principles and methods. Data and Knowledge Engineering 25(1-2), 161–197 (1998)
23. Tanev, H., Magnini, B.: Weakly supervised approaches for ontology population. In: Proceeding of the 2008 conference on Ontology Learning and Population: Bridging the Gap between Text and Knowledge. pp. 129–143. IOS Press, Amsterdam, The Netherlands (2008)
24. Völker, J., Hitzler, P., Cimiano, P.: Acquisition of owl dl axioms from lexical resources. In: ESWC '07: Proceedings of the 4th European conference on The Semantic Web. pp. 670–685. Springer-Verlag, Berlin, Heidelberg (2007)

Efficient approximate SPARQL querying of Web of Linked Data

B.R.Kuldeep Reddy and P.Sreenivasa Kumar

Indian Institute of Technology Madras,
Chennai, India
{brkreddy,psk}@cse.iitm.ac.in

Abstract. The web of linked data represents a globally distributed dataspace which can be queried using the SPARQL query language. However, with the growth in size and complexity of the web of linked data, it becomes impractical for the user to know enough about its structure and semantics for the user queries to produce enough answers. This problem is addressed in the paper by making use of ontologies available on the web of linked data to produce approximate results. The existing approach, which generates multiple relaxed queries and executes them sequentially one by one, is improved by integrating the approximation steps with the query execution itself. Thus, by performing query relaxation on-the-fly at runtime, the shared data between relaxed queries are not fetched repeatedly, resulting in significant performance benefits. Further opportunities for optimization during query execution are identified and are used to prune relaxation steps which do not produce results. The implementation of our approach demonstrates its efficacy.

1 Introduction

The traditional World Wide Web has allowed sharing of documents among users on a global scale. The documents are generally represented in HTML, XML formats and are accessed using URL and HTTP protocols creating a global information space. However, in the recent years the web has evolved towards a web of data [1] as the conventional web's data representation sacrifices much of its structure and semantics [2] and the links between documents are not expressive enough to establish the relationship between them. This has lead to the emergence of the global data space known as Linked Data[2].

Linked data basically interconnects pieces of data from different sources utilizing the existing web infrastructure. The data published is machine readable that means it is explicitly defined. Instead of using HTML, linked data uses RDF format to represent data. The connection between data is made by typed statements in RDF which clearly defines the relationship between them resulting in a web of data.

Berners-Lee outlined a set of Linked Data Principles for publishing data on the Web [3] in a way that all published data becomes part of a single global data space:

1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names
3. When someone looks up a URI, provide useful information, using the standards(RDF, SPARQL)
4. Include links to other URIs, so that they can discover more things

The RDF model describes data in the form of subject, predicate and object triples. The subject and object of a triple can be both URIs that each identify an entity, or a URI and a string value respectively. The predicate denotes the relationship between the subject and object, and is also represented by a URI. SPARQL is the query language proposed by W3C recommendation to query RDF data [4]. A SPARQL query basically consists of a set of triple patterns. It can have variables in the subject,object or predicate positions in each of the triple pattern. The solution consists of binding these variables to entities which are related with each other in the RDF model according to the query structure.

There have been a number of approaches proposed to query the web of linked data. One direction has been to crawl the web by following RDF links and build an index of discovered data. The queries are then executed against these indexes. This approach is followed by Sindice[5], Swoogle[6]. Another approach has been to follow the federated query processing concepts [7], as in DARQ[8], which decomposes a SPARQL query in subqueries, forwards these subqueries to multiple, distributed query services, and, finally, integrates the results of the subqueries. Another execution approach for evaluating SPARQL queries on linked data is proposed in [9]. It is basically a run-time approach which executes the query by asynchronously traversing RDF links to discover data sources at run-time. SPARQL query execution takes place by iteratively dereferencing URIs to fetch their RDF descriptions from the web and building solutions from the retrieved data. The SPARQL query execution according to [9] is explained with an example below.

```
SELECT ?prof ?publ WHERE
{
 <http://site//univ> univ:hasPublications ?publ
      ?publ  univ:authoredBy  ?prof
      ?prof  rdf:type  Professor
}
```

Fig. 1. Example SPARQL query

Example. The SPARQL query shown in Figure 1 searches for Professors employed by the university who have authored a publication. The query execution begins by fetching the RDF description of the university by dereferencing its URI. The fetched RDF description is then parsed to gather a list of all of its pub-

lications. Parsing is done by looking for triples that match the first pattern in the query. The object URIs in the matched triples form the list of publications in the university. Lets say `<http://site1/publ1.rdf>`, `<http://site2/publ2.rdf>`, `<http://publ3/Mary.rdf>` were found to be the papers. The query execution proceeds by fetching the RDF descriptions corresponding to the three publications. Lets say first publ1's graph is retrieved. It is parsed to check for triples matching the second query pattern and it is found that publ1 was authored by John `<http://site4/John.rdf>`. John's details are again fetched and the third triple pattern in the query is searched in the graph to see whether he is of type Professor and if he is, the result of query is formed and displayed as output. Publ1's and Publ2's graphs and their author details would also be retrieved and the query execution proceeded in a way similar to Publ1's.

Consider the situation where the retrieved list publications authored by the professors may not meet the requirements of the user in which case query conditions can be relaxed to produce more results. For example, instead of looking for only Professors, the query can be generalized by searching for all types of people including lectures,graduate students etc. The algorithm presented in [10] generates relaxed SPARQL queries and executes them sequentially one after another to generate approximate answers. The algorithm was designed to work on centralized RDF repositories and the approach is extended to the web of linked data in this paper. The relaxed SPARQL queries formed share many query conditions in common, which are not utilized to optimize the queries. Especially in a distributed environment, like the web of linked data, avoiding repeated fetching of data shared across the queries results in significant performance benefits.

```
SELECT ?prof ?publ WHERE
{
 <http://site//univ> univ:hasPublications ?publ
      ?publ  univ:authoredBy  ?prof
      ?prof  rdf:type  Faculty
}
```

```
SELECT ?prof ?publ WHERE
{
 <http://site//univ> univ:hasPublications ?publ
      ?publ  univ:authoredBy  ?prof
      ?prof  rdf:type  Person
}
```

Fig. 2. Relaxed SPARQL query

Example. Figure 2 gives the two similar queries formed after the query term Professor is replaced by Faculty and Person terms using RDFS ontology, which are then executed sequentially. However, the first two predicates are common between the two queries, therefore to achieve efficiency the information corresponding to them can be fetched once. Hence, instead of generating the two queries the execution of the original query can continue by dereferencing the URIs corresponding to Publ1 and its author and retrieving their RDF descriptions, and the check performed by the third predicate to see whether the author is a Professor or not is only replaced by Faculty and Person at the last step.

This allows the retrieval of the shared data only once instead of twice had the existing approach been followed.

The goal of this paper is to perform approximate SPARQL querying of the web of linked data. This paper extends the approach presented in [10] for relaxed querying of centralized RDF repositories to the context of web of linked data and along with the execution approach presented in [9] takes into account different namespaces being used. The idea of delaying query relaxation to run-time is introduced in order to optimize the query performance and the various other optimization opportunities present are also recognized and used.

2 Similarity Measures

In [10] the similarity measures were defined to allow ranked approximate answers. However the measures were designed for centralized RDF repositories and considered only one ontology. In the context of linked data, each user publishing data has the freedom to define his own ontology, but according to the principles of linked data it has to be mapped to existing ontologies, therefore we assume such mappings exist for the purposes of this paper.

A triple pattern can be replaced by terms in the ontology in a number of ways. Therefore, there is a need to attach a score to each relaxation which can be used to rank them to ensure the quality of results. The score given to each relaxation measures the similarity of it to the original triple pattern. Highest scoring relaxation are executed first followed by others in the decreasing order of the similarity score. For example, we would rank the relaxation from (?X,type,professor) to (?X,type,faculty) higher than (?X,type,professor) to (?X,type,person) as the former is more similar to the original triple pattern. A SPARQL query consists of a basic graph pattern which in turn consists of triple patterns. Therefore, the score associated with an answer to a SPARQL query is computed by aggregating the scores of relaxed triple patterns. Each triple pattern consists of a subject, predicate and object parts, and each of them can be potentially relaxed. Their aggregated score gives the score of the triple pattern.

Similarity between nodes In a triple pattern t_1, if the subject/object node belongs to class c_1 in the RDFS ontology and is relaxated to class c_2 using the ontology we use the idea of Least Common Ancestor to compute the similarity of the two triple patterns. The Least Common Ancestor denotes the depth of the common ancestor superclass of the two classes from the root in the RDFS ontology.

$$score(c_1, c_2) = \frac{2 * Depth(LCA(c_1, c_2))}{depth(c_1) + depth(c_2)}$$

Similarity between predicates In a triple pattern t_1, if the predicate belongs to class p_1 in the RDFS ontology and is relaxed to class p_2 using the ontology we use the idea of Least Common Ancestor to compute the similarity of the two triple patterns similar to that done for subject/object nodes. The Least Common Ancestor denotes the depth of the common ancestor superproperty of

the two classes from the root in the RDFS ontology.

$$score(p_1, p_2) = \frac{2 * Depth(LCA(p_1, p_2))}{depth(p_1) + depth(p_2)}$$

Similarity between triple patterns If the triple pattern t_1-$(s_1, p1, o1)$ is relaxed to t_2-(s_2, p_2, o_2) we aggregate the similarity scores of the triple pattern constituents to compute the overall similarity score of relaxed triple pattern.

$$similarity(t_1, t_2) = score(s_1, s_2) + score(p_1, p_2) + score(o_1, o_2)$$

Score of an answer The bindings of the relaxed SPARQL queries form the answers to the original SPARQL query. Since the original query is relaxed in a number of ways we need a measure to rank the relevant answers to ensure the quality of results. Thus, we define the score of each relevant answer as the similarity of its corresponding relaxed SPARQL query from which it is produced to the original SPARQL query. The similarity between the two queries is obtained by combining the similarities of the triple patterns in them. Suppose the answer A is obtained from query $Q^{'}(t_1^{'}, t_2^{'}, t_3^{'}...t_n^{'})$ which was formed after the original query $Q(t_1, t_2, t_3....t_n)$ was relaxed.

$$score(A) = \sum_{i=1}^{n} similarity(t_i, t_i^{'})$$

3 Query Processing Algorithms

[10] presents an approach to generate relaxed SPARQL queries from the original SPARQL query using RDFS ontology. It produces many relaxed versions and assigns scores to them based on the similarity to the original query. The relaxed queries are then executed one by one sequentially in the descending order of their scores to get ranked approximate answers. However, the SPARQL queries generated have many query conditions in common. Therefore, the sequential execution approach of all the queries involves needlessly fetching the same data repeatedly. In this section we present an optimized query processing algorithm where relaxed queries are generated and answered on-the-fly during query execution resulting in significant performance benefits.

Algorithm 1 describes the approach presented in [10] that can be extended to produce approximate answers in the web of linked data. Lines 2-7 denote the steps taken to generate multiple relaxed queries. The relaxation procedure is described as a graph, called a relaxation graph here. First the given query is put as a root in the relaxation graph. Then each triple is relaxed one-by-one and the new query produced as a result is inserted as a child node of the query node in the relaxation graph that led to it being produced. Each triple relaxation is accompanied by computing its relaxation score and this score is attached to its corresponding relaxed query. This process is repeated till all possible relaxed queries are generated. Lines 11-18 execute the relaxed queries produced earlier sequentially one by one. To generate ranked approximate results and ensure

the quality of answers the relaxed queries are executed in the descending order of their similarity scores with the original query. The relaxed query with the maximum score is executed first following which the next query to be executed is chosen with the highest score amongst its children and so on.

Algorithm 1: Existing Approach

Input : :Query Q
Output: :Approximate answers

1 $relaxationGraph = \phi$
2 Insert Q as root in $relaxationGraph$
3 **while** $Q \neq \phi$ **do**
4 **foreach** *Triple* t_i *in* Q **do**
5 Relax t_i to $t_i^{'}$
6 compute the score of approximation
7 Insert $Q_i^{'}$ as a succeeding node of Q in $relaxationGraph$
8 $Q \Leftarrow Q_{siblingNode}$ or $Q_{succeedingNode}$
9 $Result = \phi$
10 $Candidates = \phi$
11 Insert Q's succeeding nodes from $relaxationGraph$ into $Candidates$.
12 **while** $Candidates > 0$ **do**
13 Select Q_i with maximum score from $Candidates$
14 Insert Q_i succeeding nodes $relaxationGraph$ into $Candidates$
15 $R \Leftarrow Execute(Q_i)$
16 $Result = Result \cup R$
17 Add Q_i to processed
18 Remove Q_i from $Candidates$
19 Return $Result$

Figure 3 describes the execution of two queries of figure 2. The two queries are generated from the query of figure 1 as described by algorithm 1. The query in figure 1 finds the professors in the university who have authored a publication. To get approximate answers, the query is relaxed by producing two queries in which the query condition professor is replaced by faculty and persons. The left box in figure 4 shows the execution of left query in figure 2 and similarly for the box on the right. As we can see, many of the URIs dereferenced are the same in both the cases. For both of them, the query execution takes place by first dereferencing the university's URI to retrieve its RDF graph. Then the details of its publications publ1,publ2 and publ3 followed by its authors, John,Peter and Mary, are fetched. The existing approach repeats this process twice for each of the relaxed query when instead we can fetch the shared information once and then perform the relaxation. This motivates us to integrate the approximation process with the execution of the query and is described in algorithm 2.

Algorithm 2 describes the proposed approach in this paper for efficient approximate answering. Lines 3-16 repeat for each query predicate in the given

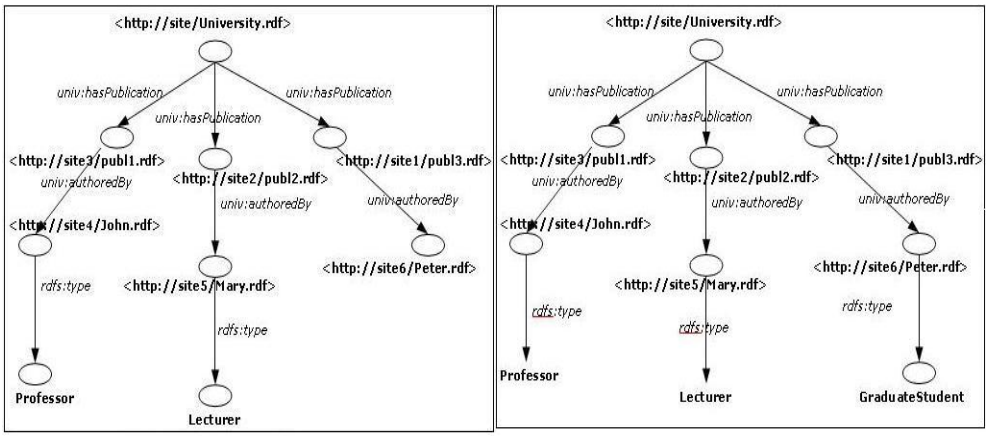

Fig. 3. Execution with existing approach

query. It begins with the seed, fetching its RDF graph. Then presence of the query predicate is checked for in the fetched RDF graph. If it is present the relaxation score for the predicate in the graph is given the maximum value of 1.0. Predicates belonging to different namespaces are assumed to be mapped in accordance with the linked data principles. Otherwise, using the metrics described in the earlier section the similarity score for each predicate in the RDF graph is computed. The predicates are then sorted in the descending order of their scores. The query execution proceeds by updating the seed with the object URIs of the predicates, which are then dereferenced to retrieve their graphs. Further similarities are computed and this process is repeated till a set of leaf values are produced. The path from the root to the leaf values in lines 17-19 along the relaxed predicates gives the approximate answers.

Figure 4 shows the query execution with the proposed approach for the query in figure 1. The query execution takes place by fetching the university's details, the details its publications and their authors just once. Once the publication's authors details have been retrieved the third predicate checking whether the person is of type professor can be relaxed to check for all people in the university like lecturers and graduate students. Thus in effect the relaxation mechanism has been delayed to be performed on-the-fly at run-time and by doing so the shared data is not fetched repeatedly which results in significant performance benefits.

4 Optimizations

The query processing described in the last section works by relaxing the query on-the-fly during query execution. This approach serves well to optimize the query but there are further opportunities that arise during query execution that can be exploited to optimize the query. To do so the vocabulary(RDFS/OWL) describing the resources which gives the domains and ranges of various predicates

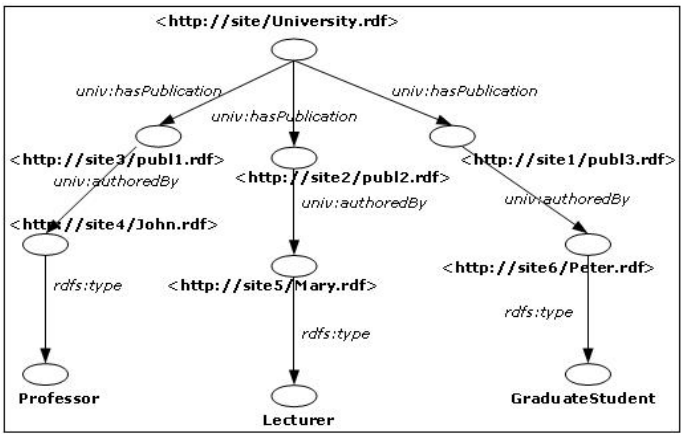

Fig. 4. Execution with proposed approach

as well as the subclass/superclass hierarchy details of all classes is considered. There are two cases that arise.

Case1: If a predicate p is replaced by $p^{'}$ with the subsequent predicate q, and that $range(p^{'}) \cap domain(q)$ is NULL the current relaxation of p is pruned as it will not produce results. There may be a situation when the subsequent predicate q is relaxed to $q^{'}$ and $range(p^{'}) \cap domain(q^{'})$ is not NULL in which case some results are missed. Therefore, a minimum threshold for the score of relaxation is maintained, and if the intersection of $range(p^{'}) \cap domain(q)$ is NULL the relaxation is pruned only if the score is below the threshold.

Case2: If an object o is replaced by $o^{'}$ with the subsequent predicate q, and that $o^{'} \cap domain(q)$ is NULL the current relaxation can be pruned as it will not produce results. There may be a situation when the subsequent predicate q is relaxed to $q^{'}$ and $o^{'} \cap domain(q^{'})$ is not NULL in which some results are missed. Therefore, a minimum threshold is maintained for the score of relaxation, and if the intersection of $o^{'} \cap domain(q)$ is NULL the relaxation is pruned only if the score is below the threshold.

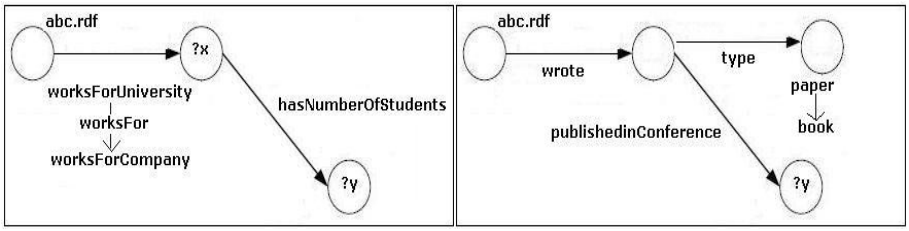

Fig. 5. Examples

Algorithm 2: Proposed Approach

Input : :Query Q
Output: :Approximate answers

let γ be the threshold
seed = intial set of uris
foreach $queryPredicate_k$ *in* Q **do**
  while *seed* $\neq \phi$**do**
    foreach $seed_i$ **do**
      Dereference $seed_i$ *and retrieve its RDF graph* R
      Remove $seed_i$ *from seed*
      foreach *predicate* p_j *in* R **do**
        if p_j *matches the corresponding query predicate* $queryPredicate_k$ **then**
          $relaxScore(p_j) = 1$
          if $p_{j_{object}} isbound$ **then**
            compute $relaxScore(p_{j_{object}})$ *with* $queryPredicate_{k_{object}}$
          else
            compute the $relaxScore(p_j)$ *with* $queryPredicate_k$
    Sort all p_j *in the descending order of their relaxScores.*
    foreach p_j **do**
      if $relaxScore(p_j) > \gamma$ **then**
        if $p_{j_{object}} isnotbound$ **then**
          $seed \Leftarrow seed \cup p_{j_{object}}$
  foreach $seed_i$ *in seed* **do**
    Retrieve the path p from $seed_i$ to root
    Return p as the approximate answer

Figures 5 illustrates the two cases. The figure on the left shows the query during whose execution the predicate "worksForUniversity" is relaxed to "worksFor". If there is a predicate "worksForCompany" in the retrieved RDF graph of the entity and as it is a subproperty of "worksFor" the query condition is relaxed to "worksForCompany". But the domain of the predicate succeeding it, that is "hasNumberOfStudents", is the class of universities whereas the range of the predicate "worksinCompany" is the class of Companies whose intersection is NULL. Thus this relaxation is pruned. But there is a possibility that the relaxation of the next predicate is from "hasNumberofStudents" to "hasnumberofEmployees". In which case the domain of the new relaxed predicate is the class of companies whose intersection with the range of earlier relaxed predicate is again the class of companies. Hence, if the first relaxation had not been discarded results could have been produced. To handle this situation, the score of relaxation is taken into account. If the score is above a certain predefined threshold, the relaxation is allowed and the query execution proceeds as usual.

The figure on the right shows the query during whose execution the object node "paper" is relaxed to the class of "books". However, the next predicate "publishedinConference" has the class of papers as it domain. Hence, the relaxation to class of books produces a NULL set and can be pruned.

Algorithm 3: Optimizations

Input : :Query Q
Output: :Decision on whether to continue with current approximation

```
1  let t denote the triple being handled, which is approximated to t'
2  let q be the predicate succeeding t
3  let γ be the threshold on the score of approximation
4  if predicate p is relaxed to p' then
5      if range(p') ∩ domain(q) == NULL then
6          if score(t) < γ then
7              try different relaxation of p
8  if object node o is relaxed to o' then
9      if o' ∩ domain(q) == NULL then
10         if score(t) < γ then
11             try different relaxation of o
```

5 Experiments

The experiments were conducted on a Pentium 4 machine running windows XP with 1 GB main memory. All the programs were written in Java. The synthetic data used for the simulations was generated with the LUBM benchmark data generator [11]. The LUBM benchmark is basically an university ontology that describes all the constituents of a university like its faculty,courses,students etc. The synthetic data is represented as a web of linked data with 200,890 nodes denoting entities and 500,595 edges denoting the relationships between them. The efficacy of the proposed idea was demonstrated by executing a set of queries in figure 6 used in [10] on the simulated web of linked data of a university and comparing the results with the existing approach. Each of the query below can be relaxed in a number of ways and the existing approach generates relaxed queries and executes them sequentially one by one whereas in contrast the proposed approach integrates the process of relaxation with the query execution to produce approximate answers. The time taken to execute the query is proportional to the number of URIs resolved to fetch their RDF descriptions during the course of query execution. Therefore, this paper uses the reduction in the number of URIs fetched as a metric to judge the results as the web of linked data was simulated on a single machine.

Q_1:(?x,type,TeachingAssitant)(?x,teachingAssistantOf,http://www.Department0.University0/Course3)(?x,mastersDegreeFrom,http://www.Department0.University0.edu)
Q_2:(?x,teacherOf,?z)(?x,ub:worksFor,University0)(?x,type,AssistantProfessor)
Q_3:(?x,advisor,?y)(?y,type,AssistantProfessor)(?y,researchInterest',Research12')(?y,worksFor,http://www.University0.edu)
Q_4:(?x,advisor,?y)(?y,type,Professor)(?y,worksFor, http://www.University0.edu)
Q_5:(?x,type,JournalAticle)(?y,publicationAuthor,?x) (?y,type,Professor)

Fig. 6. Queries

Query 1 searches for the teaching assistants of a particular course who have a masters degree from a particular university. Approximate answers are generated by relaxing the constraints step by step on the teaching assistant that is the teaching assistant can handle any course and have a master's degree from any university. Query 2 searches for assistant professors who teach a graduate course. Approximate answers are produced by relaxing the conditions in steps to look for all faculty who teach any course. Query 3 looks for assistant professor advisors who have a particular research interest. The query is again relaxed in steps by searching for all the people in the university who have any research interest. Query 4 searches for advisors who are professors and work for a particular university. Approximate answers are produced by looking for advisors who can be any type of faculty and who work for any university. Query 5 searches for professors who have authored a journal article. Approximate answers are produced step by step by looking for all persons including graduate students who have authored any type of paper. Figure 7 shows the number of URIs of entities dereferenced with the existing and the proposed approaches. The query performance improves by 75% for query 1, 80% for query 2, 83% for query 3, 78% for query 4 and 67% for query 5.

6 Conclusions And Future Work

The paper presented an approach towards allowing approximate querying of the web of linked data. The proposed idea produces approximate answers by relaxing the query conditions on-the-fly during query execution using the ontologies available on the web of linked data, in contrast with existing approach which generates multiple relaxed queries and executes them sequentially. The advantage of proposed approach is that it is able to avoid fetching the shared data between the relaxed queries repeatedly, which results in significant performance benefits as shown in the experiments. Future work includes investigation of other schemes, like top-k systems, towards producing approximate answers.

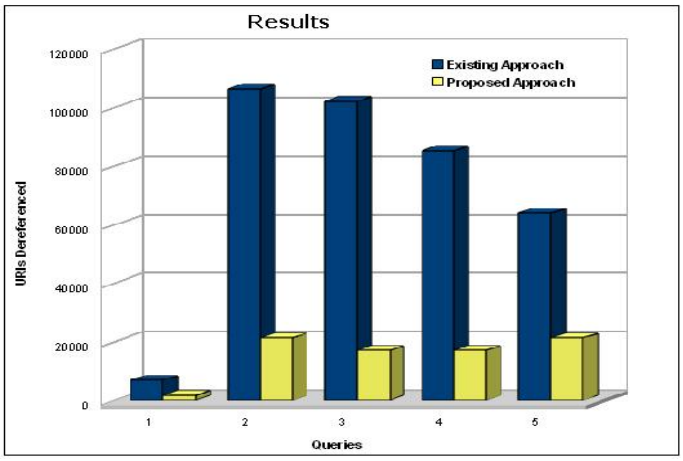

Fig. 7. Results

References

1. Franklin, M.: From databases to dataspaces: A new abstraction for information management. SIGMOD Record **34** (2005) 27–33
2. Bizer, C., Heath, T., Berners-Lee, T.: Linked data – the story so far. International Journal on Semantic Web and Information Systems **5**(3) (2009) 1–22
3. Berners-Lee, T.: Linked data - design issues. web page (2006)
4. Prud'hommeaux, E., Seaborne, A.: SPARQL query language for RDF. W3C recommendation, World Wide Web Consortium (2008)
5. Tummarello, G., Delbru, R., Oren, E.: Sindice.com: Weaving the open linked data. (2008) 552–565
6. Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R.S., Peng, Y., Reddivari, P., Doshi, V.C., Sachs, J.: Swoogle a semantic web search and metadata engine. In: Proc. 13th ACM Conf. on Information and Knowledge Management. (Nov. 2004)
7. Sheth, A.P., Larson, J.A.: Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Comput. Surv. **22**(3) (1990) 183–236
8. Quilitz, B., Leser, U.: Querying distributed rdf data sources with sparql. In Hauswirth, M., Koubarakis, M., Bechhofer, S., eds.: Proceedings of the 5th European Semantic Web Conference. LNCS, Berlin, Heidelberg, Springer Verlag (June 2008)
9. Hartig, O., Bizer, C., Freytag, J.C.: Executing SPARQL queries over the web of linked data. In: ISWC 2009: Proceedings of the 8th International Semantic Web Conference, Chantilly, VA, USA. (2009) 293–309
10. Huang, H., Liu, C., Zhou, X.: Computing relaxed answers on rdf databases. In Bailey, J., Maier, D., Schewe, K.D., Thalheim, B., Wang, X.S., eds.: WISE. Volume 5175 of Lecture Notes in Computer Science., Springer (2008) 163–175
11. Guo, Y., Pan, Z., Heflin, J.: Lubm: A benchmark for owl knowledge base systems. J. Web Sem. **3**(2-3) (2005) 158–182

Semantic Query Extension through Probabilistic Description Logics

José Eduardo Ochoa Luna[1], Kate Revoredo[2], and Fabio Gagliardi Cozman[1]

[1] Escola Politécnica, Universidade de São Paulo,
Av. Prof. Mello Morais 2231, São Paulo - SP, Brazil
[2] Departamento de Informática Aplicada, Unirio
Av. Pasteur, 458, Rio de Janeiro, RJ, Brazil
eduardo.ol@gmail.com,katerevoredo@uniriotec.br,fgcozman@usp.br

Abstract. This paper presents a novel approach for semantic query extension using a probabilistic description logic. Concepts that are related to a keyword-based query are used for finding other concepts and relations through the use of a relational Bayesian network built using the probabilistic description logic CR$\mathcal{ALC}$. Furthermore, probabilistic assessments allow us to rank the information returned by search. Examples and issues of importance in real world applications are discussed.

1 Introduction

This paper focuses on the use of ontologies to improve keyword-based search. The concepts of a given ontology are taken as annotations for documents or text fragments, thus providing background knowledge and enabling intelligent search and browsing facilities. Hence the ontological knowledge augments unstructured text with links to relevant concepts. For example, articles "Life of the probabilistic fish" and "A new kind of aquatic vertebrate with probabilistic processing" are all instances of the concept Publication; in a keyword-based search, the query "Publications about probabilistic fish" would return only the former paper. However connections amongst concepts are important to indicate further results. An ontology can then be employed for *semantic query extension*; that is, for deriving terms that lead to relevant results for the query. For example, the concept Publication is related to the concept Author; a semantic query extension strategy could use this information and reason that the second paper is a valid result as Professor G. Rouper is an author of both papers.

There is always uncertainty in this sort of reasoning. In particular, it may not be possible to guarantee that a concept is related to the ones in the query. Thus, it would be interesting if the semantic query extension system could handle the *probability* of a concept conditioned on the concepts mentioned in the query. In our example, the information about Author is valuable only if the probability of it influencing the contents of a paper is high.

An ontology can be represented through a description logic [3], which is typically a decidable fragment of first-order logic that tries to reach a practical

balance between expressivity and complexity. To represent uncertainty, a probabilistic description logic must be contemplated. The literature contains a number of proposals for probabilistic description logics [10, 11, 25]. In this paper we adopt a recently proposed probabilistic description logic, called Credal $\mathcal{ALC}$ (CR$\mathcal{ALC}$) [6], that extends the popular logic $\mathcal{ALC}$ [3]. In CR$\mathcal{ALC}$ one can specify sentences such as $P(\mathsf{Professor}|\mathsf{Researcher}) = 0.4$, indicating the probability that an element of the domain is a Professor given that it is a Researcher. These sentences are called *probabilistic inclusions*. Exact and approximate inference algorithms that deal with probabilistic inclusions have been proposed [6, 7], using ideas inherited from the theory of relational Bayesian networks [12].

In this paper, we propose an algorithm that receives keyword-based queries and that takes semantic information about the domain of the application to obtain results that are not possible in standard information retrieval. The idea here is to obtain all concept instances that are related to a given word even if that word does not appear with the concept. The system can infer relations through the probabilistic description logic CR$\mathcal{ALC}$, finding concepts probabilistically related to the ones in the query, and making it possible to retrieve concepts that do not contain any of the specified words. The information related to the chosen concepts is the set of query results, and they are returned ranked by their probability.

Section 2 reviews relevant elements of information retrieval and the probabilistic description logic CR$\mathcal{ALC}$. Section 3 presents our proposal information retrieval system. Section 4 presents some preliminary experiments. Section 5 reviews some related work and Section 6 concludes the paper.

2 Background

In this section, we review keyword-based information retrieval and the probabilistic description logic CR$\mathcal{ALC}$.

2.1 Information Retrieval Models

The field of information retrieval (IR) [14] has been defined as the subject concerned with the representation, storage, organization, and access of information items. One example of traditional IR technique is the Boolean model [23]. A document d is then represented by the vector $\overrightarrow{x} = (x_1, ..., x_M)$ where $x_t = 1$ if term t is present in document d and $x_t = 0$ otherwise. The procedure searches for documents that satisfy a query in the form of a Boolean expression of terms. Thus, if a query such as x_1 AND x_2 OR x_3 is provided, this technique retrieves documents where $x_1 = 1$ and $x_2 = 1$ simultaneously or $x_3 = 1$.

Another sort of model for IR is based on logical representations [4, 5, 13]. The task can be described as the extraction, from a given document base, of those documents d that, given a query q, make the formula $d \rightarrow q$ valid, where d and q are formulas of a chosen logic and "$\rightarrow$" denotes logical implication. In this paper, we are interested in logical representations that consider symbols d and q as

terms (i.e. expressions denoting objects or sets of objects). Different formalisms have been proposed with these goals. An example is the terminological logic for IR proposed in [15]. In that logic, documents are represented by individual constants, whereas a class of documents is represented as a concept, and queries are described as concepts. Given a query q, the task is to find all those documents d such that $q(d)$ holds. The evaluation of $q(d)$ uses the set of assertions describing documents; that is, instead of evaluating whether d is related to q, evaluate if "individual d is an instance of the class concept q".

2.2 Probabilistic Description Logics and CR$\mathcal{ALC}$

A description logic (DL) offers a formal language where one can describe knowledge such as "A Professor is a Person who works in an Organization". To do so, a DL typically uses a decidable fragment of first-order logic [3], and tries to reach a practical balance between expressivity and complexity. The last decade has seen a significant increase in interest in DLs as a vehicle for large-scale knowledge representation, for instance in the semantic web. Indeed, the language OWL [1], proposed by the W3 consortium as the data layer of their architecture for the semantic web, is an XML encoding for quite expressive DLs.

Knowledge in a DL is expressed using *individuals*, *concepts*, and *roles*. The semantics is given by a *domain* $\mathcal{D}$ and an *interpretation* $\cdot^{\mathcal{I}}$. Individuals represent objects through names from a set of names $N_{\mathsf{I}} = \{\mathsf{a}, \mathsf{b}, \ldots\}$. Each *concept* in the set of concepts $N_{\mathsf{C}} = \{\mathsf{C}, \mathsf{D}, \ldots\}$ is interpreted as a subset of a domain $\mathcal{D}$ (a set of objects). Each *role* in the set of roles $N_{\mathsf{R}} = \{\mathsf{r}, \mathsf{s}, \ldots\}$ is interpreted as a binary relation on the domain. Objects correspond to constants, concepts to unary predicates, and roles to binary predicates in first order logic. Concepts and roles are combined to form new concepts using a set of *constructors*. Constructors in the $\mathcal{ALC}$ logic are *conjunction* ($C \sqcap D$), *disjunction* ($C \sqcup D$), *negation* ($\neg C$), *existential* restriction ($\exists r.C$), and *value* restriction ($\forall r.C$). *Concept inclusions/definitions* are denoted respectively by $C \sqsubseteq D$ and $C \equiv D$, where C and D are concepts. Concept ($C \sqcup \neg C$) is denoted by $\top$, and concept ($C \sqcap \neg C$) is denoted by $\bot$.

The probabilistic description logic (PDL) CR$\mathcal{ALC}$ [7] is a probabilistic extension of the DL $\mathcal{ALC}$ that adopts an interpretation-based semantic. It keeps all constructors of $\mathcal{ALC}$, but only allows concept names in the left hand side of inclusions/definitions. Additionally, in CR$\mathcal{ALC}$ one can have probabilistic inclusions such as $P(C|D) = \alpha, P(r) = \beta$ for concepts C and D, and for role r. For any element of the domain, the probability that this element is in C, given that it is in D is α. If the interpretation of D is the whole domain, then we simply write $P(C) = \alpha$. The semantics of these inclusions is roughly as follows (a formal definition can be found in [7]):

$$\forall x \in \mathcal{D} \ : \ P(C(x)|D(x)) = \alpha \quad \textit{and} \quad \forall x \in \mathcal{D}, y \in \mathcal{D} \ : \ P(r(x,y)) = \beta.$$

We assume that every terminology is acyclic; no concept uses itself. This assumption allows one to represent any terminology $\mathcal{T}$ through a relational Bayesian

network (RBN). A directed acyclic graph, denoted by $\mathcal{G}(\mathcal{T})$, has each concept name and role name as a node, and if a concept C directly uses concept D, if C appear in the left and D in the right hand sides of an inclusion/definition, then D is a *parent* of C in $\mathcal{G}(\mathcal{T})$. Each existential restriction $\exists r.C$ and value restriction $\forall r.C$ is added to the graph $\mathcal{G}(\mathcal{T})$ as nodes, with an edge from r to each restriction directly using it. Each restriction node is a *deterministic* node in that its value is completely determined by its parents. Considers the following example.

Example 1. Consider a terminology $\mathcal{T}_1$ with concepts $\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}$. Suppose $P(\mathsf{A}) = 0.9, \mathsf{B} \sqsubseteq \mathsf{A}, \mathsf{C} \sqsubseteq \mathsf{B} \sqcup \exists r.\mathsf{D}, P(\mathsf{B}|\mathsf{A}) = 0.45, P(\mathsf{C}|\mathsf{B} \sqcup \exists r.\mathsf{D}) = 0.5$, and $P(\mathsf{D}|\forall r.\mathsf{A}) = 0.6$. The last three assessments specify beliefs about partial overlap among concepts. Suppose also $P(\mathsf{D}|\neg\forall r.\mathsf{A}) = \epsilon \approx 0$ (conveying the existence of exceptions to the inclusion of D in $\forall r.\mathsf{A}$). Figure 1 depicts $\mathcal{G}(\mathcal{T})$.

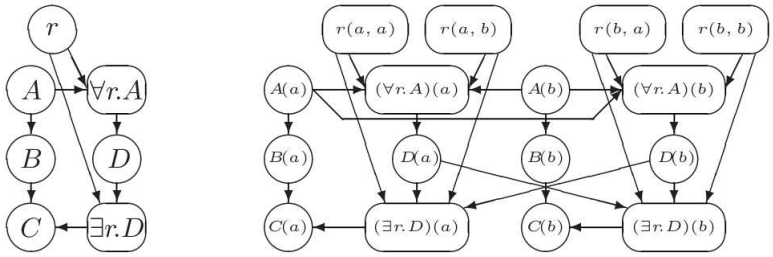

Fig. 1. $\mathcal{G}(\mathcal{T})$ for terminology $\mathcal{T}$ in Example 1 and its grounding for domain $\mathcal{D} = \{a, b\}$.

The semantics of CR$\mathcal{ALC}$ is based on probability measures over the space of interpretations, for a fixed domain. Inferences, such as $P(\mathsf{A}_o(\mathsf{a}_0)|E)$, where E is a set of evidences, can be computed by propositionalization and probabilistic inference (for exact calculations) or by a first order loopy propagation algorithm (for approximate calculations) [7]. Considering the domain $\mathcal{D} = \{a, b\}$ the grounding of $\mathcal{G}(\mathcal{T})$ of Example 1 is shown in Figure 1.

3 Semantic Query Extension with CR$\mathcal{ALC}$

In the last decade several proposals have been made for semantic information retrieval. Boolean and vector space procedures, for example, have corresponding semantic versions [26, 20, 19, 8] and [27, 2, 9] respectively. We refer to [24] for a more detailed review. *Query extension* (or *query suggestion*) is a strategy often used in search engines to derive queries that are able to return more useful search results than original queries [14]. Most popular search engines provide facilities that let users complete, specify, or reformulate their queries. *Semantic query extension* is a special type of query extension based on the identification

of semantic concepts contained in user queries [16]. For example, the result for query "Publications of probabilistic description logic" can be improved when a system that considers semantics extends the query to consider also the concept Author instead of only the concept Publication.

In [18] we employed the PDL CR$\mathcal{ALC}$, combined with traditional IR, to retrieve documents relevant to the query when analyzing the terms of the query separately. In this paper, we claim that the PDL CR$\mathcal{ALC}$ can also be useful for semantic query extension so as to obtain documents that are related to a given word even if that word does not appear with the concept. Therefore, a probabilistic ontology to model the domain represented by the documents is created. This probabilistic ontology is represented through the PDL CR$\mathcal{ALC}$ and can be learned from data (we refer to [17, 21] for detailed information on how to learn CR$\mathcal{ALC}$ sentences from data). Then, the documents are linked to this ontology through indexes. Texts on documents are indexed and these texts are properties in the corresponding ontology. Therefore, documents and ontology are decoupled, but at the same time are related by sharing the same indexed text. The ontology and the indexed documents are input for our semantic search process. The semantic search process is divided in three parts: (i) search, (ii) query extension and (iii) ranking the results according to their relevance. The key design choices for each task are described as follows.

Search Procedure Given a query as a set of keywords, the concepts and roles related to it are found through three steps. First, a keyword-based search is performed finding the set of documents related to the keywords provided by the user. Next, the concepts and roles related to these documents are found through the corresponding indexes (therefore, the concept properties are also identified). Finally, a relational Bayesian network propositionalized is built where the concepts selected are evidence in this network. This relational Bayesian network is the input for the query extension phase.

Query Extension Procedure Expanding a given query involves adding terms and/or operators to the original query in order to improve results. In our proposal, the ontology provides terms that may be added to the query. Inference is performed in the relational Bayesian network found during search. The probability of all concepts that are not evidence in the RBN is inferred. A threshold is considered and the concepts with a probability higher than this threshold are selected and provided as input for the ranking results phase.

Ranking Procedure In this phase the documents related to the concepts selected by the query extension step are retrieved and ranked according to their probability. Then, these documents are shown together with the documents firstly selected in the search process step. It is worth noting that the documents selected in the search process are reordered according their probabilities; that is, a merged ordered list of documents is exhibited to the user.

There are two main drawbacks with this proposal. The first is the size of ontologies and the second is the amount of instances that are obtained after propositionalization. In principle, these issues prevent us from performing probabilistic inference on real world domains and therefore limit our framework to limited size domains. Fortunately, we can resort to variational methods in order to perform approximate inference [7] making possible the application of our proposal.

4 Preliminary Results

Experiments were performed on a real world dataset: the Lattes Curriculum Platform[3], a public repository containing data about Brazilian researchers in HTML format. Due its content is quite structured (sections such as name, address education, etc. are well defined) it is clearly possible to construct a probabilistic ontology from it. We randomly selected 1964 web documents to this task, learning the probabilistic terminology from data with the CR$\mathcal{ALC}$ learning algorithm presented in [21]. The complete probabilistic terminology is given by:

$P(\mathsf{Person}) = 0.9$
$P(\mathsf{Publication}) = 0.5$
$P(\mathsf{Board}) = 0.33$
$P(\mathsf{Supervision}) = 0.35$
$P(\mathsf{hasPublication}) = 0.85$
$P(\mathsf{hasSupervision}) = 0.6$
$P(\mathsf{hasParticipation}) = 0.78$
$P(\mathsf{wasAdvised}) = 0.15$
$P(\mathsf{hasSameInstitution}) = 0.4$
$P(\mathsf{sharePublication}) = 0.22$
$P(\mathsf{sameExaminationBoard}) = 0.19$

$\mathsf{Researcher} \equiv \mathsf{Person} \sqcap (\exists \mathsf{hasPublication.Publication} \sqcap \exists \mathsf{hasSupervision.Supervision} \sqcap \exists \mathsf{hasParticipation.Board})$

$P(\mathsf{NearCollaborator} \mid \mathsf{Researcher} \sqcap \exists \mathsf{sharePublication}.\exists \mathsf{hasSameInstitution}.\exists \mathsf{sharePublication.Researcher}) = 0.95$

$\mathsf{FacultyNearCollaborator} \equiv \mathsf{NearCollaborator} \sqcap \exists \mathsf{sameExaminationBoard.Researcher}$

$P(\mathsf{NullMobilityResearcher} \mid \mathsf{Researcher} \sqcap \exists \mathsf{wasAdvised}.\exists \mathsf{hasSameInstitution.Researcher}) = 0.98$

$\mathsf{StrongRelatedResearcher} \equiv \mathsf{Researcher} \sqcap (\exists \mathsf{sharePublication.Researcher} \sqcap \exists \mathsf{wasAdvised.Researcher})$

$\mathsf{InheritedResearcher} \equiv \mathsf{Researcher} \sqcap (\exists \mathsf{sameExaminationBoard.Researcher} \sqcap \exists \mathsf{wasAdvised.Researcher})$

Text on web documents was indexed according to linked properties on the ontology. When a keyword occurs within a given property, the keyword brings evidence about instance of properties for a given concept. The former probabilistic terminology acts as template for concept and property instances.

The overall process is detailed as follows. Assume we pose a query on "Bayesian networks" (the Lucene [4] search engine was used to do so), the system retrieves

[3] http://lattes.cnpq.br/.
[4] http://lucene.apache.org/

an ordered list of 20 researchers with links to Lattes curriculum as depicted in Figure 2.

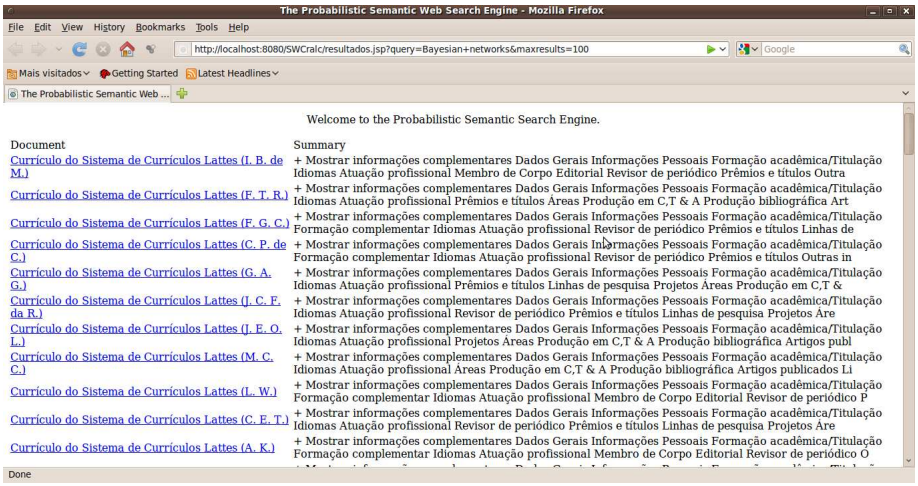

Fig. 2. Traditional query.

Suppose the user intends to follow each link and to inspect where "Bayesian networks" is located, so as to determine relevance of the document retrieved. In our setting these 20 results are candidate documents that could be further extended. Actually, these results are candidate instance concepts in the probabilistic terminology.

Furthermore, because of indexing on text properties, we are able to instantiate specific properties where the query occurs. This step allow us to "propositionalize" the inherent relational Bayesian network associated with the probabilistic ontology. Furthermore, in this probabilistic setting, each query occurrence inside properties denotes evidence on corresponding nodes. For instance, if Researcher(0) contains the query keyword on a given publication the corresponding node hasPublication$(0,1)$ is set to true. Some roles also allow us to state relationships among concept instances (the sharePublication$(0,2)$ role relates Researcher(0) and Researcher(2) through a shared publication) and therefore enforce likelihood of related concepts that leads to extensions of the original query. The resulting relational Bayesian network after propositionalization is shown in Figure 3.

Probabilistic inference is performed on the relational Bayesian network to obtain semantic query extensions; that is, top related concepts and top related researchers to the query are added to results. The extended results page is depicted in Figure 4. Some new entries were added to the former results page (for instance, the researcher P. E. M. was added because of its strong relationship

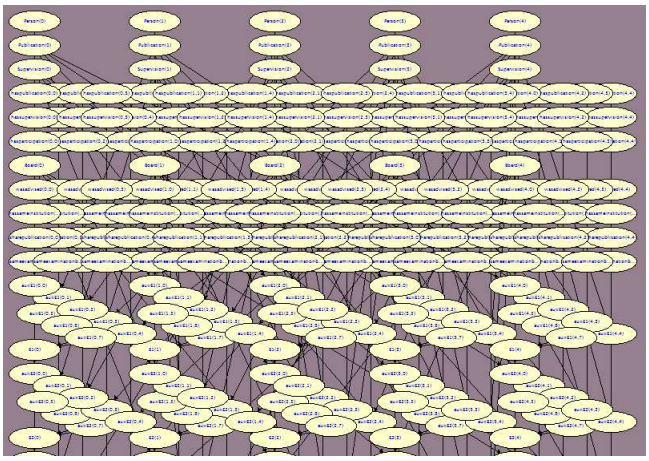

Fig. 3. Relational Bayesian network after propositionalization.

with a top researcher on "Bayesian networks"). In addition, the final research list has extended information with links to specific properties and concepts rather than uninformative snippet texts.

Probabilistic reasoning also allows us to obtain a probabilistic ranking. Intuitively, higher evidence on a given topic gives rise to a better ranking position. The previous ranking in Figure 2 returned the three following researchers: I.B. de M., F. T. R. and F. G. C. Conversely, our probabilistic logic setting returns a modified order: F. G. C., I.B. de M. and A. C. F. O. A relational Bayesian network model allow us to further investigate these results. The higher ranking was attributed to researcher F. G. C. due to evidence of query topic on publications, advising works and participations of examination boards ($\mathsf{P}(\mathsf{Researcher}(\mathsf{F.G.C.}) | \mathsf{hasPublication.P}, \mathsf{advises.S}, \mathsf{participate.B}) = \alpha$). The rest of the ranking was obtained accordingly.

To evaluate results obtained by our approach, two types of tests were conducted. The first type focuses on searching researchers that best match several topics (given as keywords). The aim of this test is evaluate whether the semantic search return meaningful results. In order to do so, we have chosen random topics such as "Bayesian networks", "probabilistic logic", "pattern recognition" and so on with well established research groups in Brazil. Lists of researchers and related concepts were evaluated qualitatively. All 20 topics evaluated had positive analysis. Note that the analysis of results for semantic searches is still an open issue; in fact, there is no standard evaluation benchmarks that contain all required information to judge the quality of the current semantic search methods [9].

The second test addresses the ranking problem; that is, are the top researchers listed first for every topic? This issue is linked to probabilistic as-

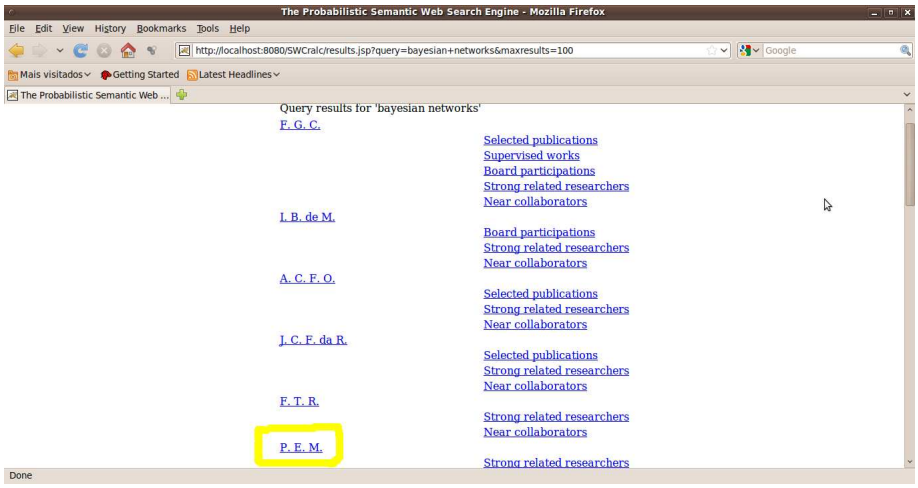

Fig. 4. Final extended result.

sessments that denote strength of relationships among instances, and give rise to a 99% positive analysis.

5 Related Work

Our framework for semantic query extension has been influenced by previous works, which we now briefly review.

The work in [22] describes a semantic search that is based on keywords, but at the same time uses the semantic information about the domain of interest to obtain results that are not possible with traditional searches. Differently from traditional searches, the work obtains all concept instances that are related to a given word even if that word does not appear inside the concept. The system can infer relations through a spread activation algorithm, making it possible to retrieve concepts that do not contain any of the specified words. Given an initial set of activated concepts and some restrictions, activation flows through the instance network reaching other concepts which are closely related to the initial concepts. One of the ideas is to extract knowledge from the ontology and its instances to obtain a numerical weight for each existing relation instance in the model. The result is an hybrid instances network, where each relation instance has both a semantic label and numerical weight. The intuition behind this idea is that better results in the search process can be achieved using the semantic information together with the sub-symbolic (numerically encoded) information extracted from the instances. The present work is different in that it uses a relational Bayesian network to find other concepts related to the one in the query. Therefore, it also finds the probability associated to the concepts.

In [16] the most relevant concepts for the full query and for each contiguous sequence of n words of the query are collected; then, a supervised machine learning method is used to decide which of the retrieved concepts should be kept and which should be discarded. In order to train the learning algorithm, queries submitted and manually linked to relevant DBpedia concepts are used as datasets [28]. The task: given a query (within a session, for a given user), produce a ranked list of concepts from DBpedia that are mentioned or meant in the query. These concepts could then be used to suggest contextual information, such as text snippets from the Wikipedia article. One difference to the present proposal is that we do handle uncertainty explicitly; also, we do not change the original query.

Another complete framework was proposed in [9]. Basically, two tasks were addressed. The first, understanding the natural language user request and retrieving an answer in the form of pieces of ontological knowledge. The user's query is processed and translated into the terminology of available ontologies, thus retrieving a list of ontological entities as a response. In the second task, relevant documents are retrieved and ranked based on the previously retrieved pieces of ontological knowledge. Just as traditional ranking algorithms are based on keyword weighting, their approach relies on measuring the relevance of each individual association between semantic concepts and web documents. This work is related to ours because it also maintains the search process decoupled (ontology and text are explored separately). The difference relies on the consideration of uncertainty in the present work.

6 Conclusion

We have presented a framework for retrieving information using a mix of web documents and probabilistic ontologies. The idea is to extract semantic information in two steps. In the first step, a probabilistic ontology is constructed based on a set of documents. The second step searches for instance concepts that best match a given user query. The algorithm links ontology properties to indexed documents in such a way that properties are instantiated in response to queries.

By handling properties and concepts we can instantiate related concepts and therefore obtain a meaningful relational Bayesian network to perform inference and to obtain a ranking of concepts. Experiments focused on a real-world domain (the Lattes scientific repository) suggest that this approach does lead to improved query results.

Acknowledgements

The first author is supported by CAPES and the third author is partially supported by CNPq. The work reported here has received substantial support through FAPESP grant 2008/03995-5.

References

1. G. Antoniou and F. van Harmelen. *Semantic Web Primer*. MIT Press, 2008.
2. K. Anyanwu, A. Maduko, and A. Sheth. SemRank: ranking complex relationship search results on the semantic web. In *Proceedings of the 14th international conference on World Wide Web*, pages 117–127, New York, NY, USA, 2005. ACM.
3. F. Baader and W. Nutt. Basic description logics. In *Description Logic Handbook*, pages 47–100. Cambridge University Press, 2002.
4. J. Cornelis and A. van Rljsbergen. New theoretical framework for information retrieval. In *ACM Conf. on Research and Development in Information Retrieval (SIGIR)*, pages 194–200, 1986.
5. J. Cornelis and A. van Rljsbergen. A non-classical logic for information retrieval. *The Computer Journal*, 29:481–485, 1986.
6. F.G. Cozman and R.B. Polastro. Loopy propagation in a probabilistic description logic. In Sergio Greco and Thomas Lukasiewicz, editors, *Second International Conference on Scalable Uncertainty Management*, Lecture Notes in Artificial Intelligence (LNAI 5291), pages 120–133. Springer, 2008.
7. F.G. Cozman and R.B. Polastro. Complexity analysis and variational inference for interpretation-based probabilistic description logics. In *Conference on Uncertainty in Artificial Intelligence*, pages 1–9, 2009.
8. L. Ding, T. Finin, A. Joshi, Y. Peng, R. Pan, and P. Reddivari. Search on the semantic web. *Computer*, 38:62–69, 2005.
9. M. Fernandez, V. Lopez, M. Sabou, V. Uren, D. Vallet, E. Motta, and P. Castells. Semantic search meets the web. In *Proceedings of the 2nd IEEE International Conference on Semantic Computing*, pages 253–260, Washington, DC, USA, 2008. IEEE Computer Society.
10. J. Heinsohn. Probabilistic description logics. In *International Conf. on Uncertainty in Artificial Intelligence*, pages 311–318, 1994.
11. M. Jaeger. Probabilistic reasoning in terminological logics. In *Principals of Knowledge Representation (KR)*, pages 461–472, 1994.
12. M. Jaeger. Relational Bayesian networks: a survey. *Linkoping Electronic Articles in Computer and Information Science*, 6, 2002.
13. M. Lalmas and P. Bruza. The use of logic in information retrieval modelling. *The Knowledge Engineering Review*, 13:263–295, 1998.
14. C. Manning, P. Raghavan, and H. Schütze, editors. *Introduction to Information Retrieval*. Cambridge, 2008.
15. C. Meghini, F. Sebastiani, U. Straccia, and C. Thanos. A model of information retrieval based on a terminological logic. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 298–307, New York, NY, USA, 1993. ACM.
16. E. Meij, M. Bron, B. Huurnink, L. Hollink, and M. de Rijke. Learning semantic query suggestions. In *8th International Semantic Web Conference*, pages 424–440. Springer, 2009.
17. J. Ochoa-Luna and F.G. Cozman. An algorithm for learning with probabilistic description logics. In *5th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW) at the 8th International Semantic Web Conference (ISWC)*, pages 63–74, Chantilly, USA, 2009.
18. J. Ochoa-Luna, K. Revoredo, and F.G. Cozman. Semantic query extension using query contexts and probabilistic description logics. In *Proceedings of the 3rd International Workshop on Web and Text Intelligence*. To appear, 2010.

19. B. Popov, A. Kiryakov, D. Ognyanoff, D. Manov, and A. Kirilov. Kim – a semantic platform for information extraction and retrieval. *Nat. Lang. Eng.*, 10(3-4):375–392, 2004.
20. R. Guha R., McCool, and E. Miller. Semantic search. In *Proceedings of the 12th international conference on World Wide Web*, pages 700–709, New York, NY, USA, 2003. ACM.
21. K. Revoredo, J. Ochoa-Luna, and F.G. Cozman. Learning terminologies in probabilistic description logics. In *Proceedings of the 20th Brazilian Symposium on Artificial Intelligence*. To appear, 2010.
22. C. Rocha, D. Schwabe, and M. Aragao. A hybrid approach for searching in the semantic web. In *Proceedings of the 13th international conference on World Wide Web*, pages 374–383, New York, NY, USA, 2004. ACM.
23. G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.
24. P. Scheir, V. Pammer, and S. Lindstaedt. Information retrieval on the semantic web - does it exist? In *In LWA 2007, Lernen - Wissensentdeckung - Adaptivität, 24.-26.9. 2007 in Halle/Saale (in this volume*, 2007.
25. F. Sebastiani. A probabilistic terminological logic for modelling information retrieval. In *ACM Conf. on Research and Development in Information Retrieval (SIGIR)*, pages 122–130, 1994.
26. A. Sheth, C. Bertram, D. Avant, B. Hammond, K. Kochut, and Y. Warke. Managing semantic content for the web. *IEEE Internet Computing*, 6(4):80–87, 2002.
27. N. Stojanovic, N. Studer, and R. Stojanovic. An approach for the ranking of query results in the semantic web. In *Proceedings of the 2nd International Semantic Web Conference*, pages 500–516, 2003.
28. I. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2005.

Finite Fuzzy Description Logics: A Crisp Representation for Finite Fuzzy $\mathcal{ALCH}$

Fernando Bobillo[1] and Umberto Straccia[2]

[1] Dpt. of Computer Science and Systems Engineering, University of Zaragoza, Spain
[2] Istituto di Scienza e Tecnologie dell'Informazione (ISTI - CNR), Pisa, Italy
Email: fbobillo@unizar.es, straccia@isti.cnr.it

Abstract. Fuzzy Description Logics (DLs) are a formalism for the representation of structured knowledge affected by imprecision or vagueness. In the setting of fuzzy DLs, restricting to a finite set of degrees of truth has proved to be useful. In this paper, we propose finite fuzzy DLs as a generalization of existing approaches. We assume a finite totally ordered set of linguistic terms or labels, which is very useful in practice since expert knowledge is usually expressed using linguistic terms. Then, we consider any smooth t-norm defined over this set of degrees of truth. In particular, we focus on the finite fuzzy DL $\mathcal{ALCH}$, studying some logical properties, and showing the decidability of the logic by presenting a reasoning preserving reduction to the non-fuzzy case.

1 Introduction

It has been widely pointed out that classical ontologies are not appropriate to deal with imprecise and vague knowledge, which is inherent to several real-world domains. Since fuzzy logic is a suitable formalism to handle these types of knowledge, there has been an important interest in generalize the formalism of Description Logics (DLs) [1] to the fuzzy case [2].

It is well known that different families of fuzzy operators (or fuzzy logics) lead to fuzzy DLs with different properties [2]. For example, Gödel and Zadeh fuzzy logics have an idempotent conjunction, whereas Łukasiewicz and Product fuzzy logic do not. Clearly, different applications may need different fuzzy logics.

In fuzzy DLs, some fuzzy operators imply logical properties which are usually undesired. For instance, in Zadeh fuzzy logic concepts and roles do not fully subsume themselves [3]. Furthermore, Łukasiewicz logic may not be suitable for combining information, as the conjunction easily collapses to zero [4]. Hence, the study of new fuzzy operators is an interesting topic.

Assuming a finite set of degrees of truth is useful in the setting of fuzzy DLs, [3,5,6]. In the Zadeh case it is interesting for computational reasons [3]. In Gödel logic, it is necessary to show that the logic verifies the Witnessed Model Property [7]. In Łukasiewicz logic, it is necessary to obtain a non-fuzzy representation of the fuzzy ontology [6]. A question that immediately arise is whether this assumption is possible when different fuzzy logics are considered.

There is a recent promising line of research that tries to fill the gap between mathematical fuzzy logic and fuzzy DLs [7,8,9]. Following this path, we build on the previous research on finite fuzzy logics [10,11,12] and propose a generalization of the different fuzzy DLs under finite degrees of truth that have been proposed, as we consider any smooth t-norm defined over a chain of degrees of truth.

Instead of dealing with degrees of truth in $[0, 1]$, as usual in fuzzy DLs, we will assume a finite (totally ordered) set of linguistic terms or labels. For instance, $\mathcal{N} = \{\texttt{false}, \texttt{closeToFalse}, \texttt{neutral}, \texttt{closeToTrue}, \texttt{true}\}$. This makes possible to abstract from the numerical interpretations of these labels.

The use of linguistic labels as degrees in fuzzy DLs has already been proposed. U. Straccia proposed to take the degrees from an uncertainty lattice [13]. To guarantee soundness and completeness of the reasoning, the set of labels is assume to be finite. A recent extension of this work by other authors considers Zadeh $\mathcal{SHIN}$ [14]. Nowadays, finite chains are receiving more attention, since they are one of the building blocks of the first order t-norm based logic $L^*_\sim(\mathbf{S})\forall$, which can be used to define several related fuzzy DLs [8,9].

The benefits of this paper are two-fold: firstly, since experts' knowledge is usually expressed using a set of linguistic terms [11], the process of knowledge acquisition is easier. Secondly, we make possible to use new fuzzy operators in the setting of fuzzy DLs for the first time.

The remainder is organized as follows. Section 2 includes some preliminaries on finite fuzzy logics. Then, Section 3 defines a fuzzy extension of the DL $\mathcal{ALCH}$ based on finite fuzzy logics and discusses some logical properties. Section 4 shows the decidability of the logic by providing a reduction of fuzzy $\mathcal{ALCH}$ into crisp $\mathcal{ALCH}$. Finally, Section 5 sets out some conclusions and ideas for future research.

2 Finite Fuzzy Logics

Fuzzy set theory and fuzzy logic were proposed by L. Zadeh [15] to manage imprecise and vague knowledge. Here, statements are not either true or false, but they are a matter of degree.

Let X be a set of elements called the reference set, and let $\mathcal{S}$ be a totally ordered scale with e as minimum element and u as maximum. A *fuzzy subset* A of X is defined by a membership function $A(x) : X \to \mathcal{S}$ which assigns any $x \in X$ to a value in $\mathcal{S}$. Similarly as in the classical case, e means no-membership and u full membership, but now a value between them represents to which extent x can be considered as an element of X.

All crisp set operations are extended to fuzzy sets. The intersection, union, complement and implication are performed by a t-norm function, a t-conorm function, a negation function, and an implication function, respectively.

In the following, we consider finite chains of degrees of truth [10,11,12]. A *finite chain* of degrees of truth is a totally ordered set $\mathcal{N} = \{0 = \gamma_0 < \gamma_1 < \cdots < \gamma_p = 1\}$, where $p \geq 1$. For our purposes all finite chains with the same number of elements are equivalent. $\mathcal{N}$ can be understood as a set of linguistic terms or labels. For example, $\{\texttt{false}, \texttt{closeToFalse}, \texttt{neutral}, \texttt{closeToTrue}, \texttt{true}\}$.

Table 1. Popular fuzzy logics over a finite chain

Family	$\gamma_i \otimes \gamma_j$	$\gamma_i \oplus \gamma_j$	$\ominus\gamma_i$	$\gamma_i \Rightarrow \gamma_j$
Zadeh	$\min\{\gamma_i, \gamma_j\}$	$\max\{\gamma_i, \gamma_j\}$	γ_{p-i}	$\max\{\gamma_{p-i}, \gamma_j\}$
Gödel	$\min\{\gamma_i, \gamma_j\}$	$\max\{\gamma_i, \gamma_j\}$	$\begin{cases} \gamma_p, & \gamma_i = 0 \\ \gamma_0, & \gamma_i > 0 \end{cases}$	$\begin{cases} \gamma_p, & \gamma_i \leq \gamma_j \\ \gamma_j, & \gamma_i > \gamma_j \end{cases}$
Łukasiewicz	$\gamma_{\max\{i+j-p,0\}}$	$\gamma_{\min\{i+j,p\}}$	γ_{p-i}	$\gamma_{\min\{p-i+j,p\}}$

In the rest of the paper, we will use the following notion: $\mathcal{N}^+ = \mathcal{N} \setminus \{\gamma_0\}$, $+\gamma_i = \gamma_{i+1}$, $-\gamma_i = \gamma_{i-1}$. Let us also denote by $[\gamma_i, \gamma_j]$ the finite chain given by the subinterval of all $\gamma_k \in \mathcal{N}$ such that $i \leq k \leq j$.

T-norms, t-conorms, negations and implications can be restricted to finite chains. Table 1 shows some popular examples: Zadeh, Gödel, and Łukasiewicz.

The *smoothness condition* is a discrete counterpart of continuity on $[0, 1]$. A function $f : \mathcal{N} \to \mathcal{N}$ is *smooth* iff it satisfies the following condition for all $i \in \mathcal{N}^+$ $f(\gamma_i) = \gamma_j$ implies that $f(\gamma_{i-1}) = \gamma_k$ with $j - 1 \leq k \leq j + 1$. A binary operator is smooth when it is smooth in each place.

A *t-norm* on $\mathcal{N}$ is a function $\otimes : \mathcal{N}^2 \to \mathcal{N}$ satisfying commutativity, associativity, monotonicity, and some boundary conditions. Smoothness for t-norms is equivalent to the divisibility condition in $[0, 1]$, i.e., $\gamma_i \leq \gamma_j$ if and only if there exists $\gamma_k \in \mathcal{N}$ such that $\gamma_j \otimes \gamma_k = \gamma_i$. A t-norm $\otimes$ is *Archimedean* iff $\forall \gamma_1, \gamma_2 \in \mathcal{N} \setminus \{\gamma_0, \gamma_p\}$ there is $n \in \mathbb{N}$ such that $\gamma_1 \otimes \gamma_1 \cdots \otimes \gamma_1$ (n times) $< \gamma_2$.

Proposition 1. *There is one and only one Archimedean smooth t-norm on $\mathcal{N}$ given by $\gamma_i \otimes \gamma_j = \gamma_{\max\{0, i+j-p\}}$. Moreover, given any subset J of $\mathcal{N}$ containing γ_0, γ_p, there is one and only one smooth t-norm $\otimes^J$ on $\mathcal{N}$ that has J as the set of idempotent elements. In fact, if J is the set $J = \{0 = \gamma_{i_0} < \gamma_{i_1} < \cdots < \gamma_{i_{m-1}} < \gamma_{i_m} = 1\}$ such a t-norm is given by:*

$$\gamma_i \otimes^J \gamma_j = \begin{cases} \gamma_{\max\{i_k, i+j-i_{k+1}\}} & \text{if } \gamma_i, \gamma_j \in [i_k, i_{k+1}] \text{ for some } 0 \leq k \leq m-1 \\ \gamma_{\min\{i,j\}} & \text{otherwise} \end{cases}$$

Note that the Archimedean smooth t-norm happens with $J = \{\gamma_0, \gamma_p\}$, and that the minimum happens with $J = \mathcal{N}$. It is worth to note that, as a consequence of Proposition 1, a finite smooth product t-norm is not possible.

Example 1. Given the finite chain $\mathcal{N} = \{\gamma_0, \gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5\}$ and the set $J = \{\gamma_0, \gamma_3, \gamma_5\}$, $\otimes^J$ is defined as:

	γ_0	γ_1	γ_2	γ_3	γ_4	γ_5
γ_0	γ_0	γ_0	γ_0	γ_0	γ_0	γ_0
γ_1	γ_0	γ_0	γ_0	γ_1	γ_1	γ_1
γ_2	γ_0	γ_0	γ_1	γ_2	γ_2	γ_2
γ_3	γ_0	γ_1	γ_2	γ_3	γ_3	γ_3
γ_4	γ_0	γ_1	γ_2	γ_3	γ_3	γ_4
γ_5	γ_0	γ_1	γ_2	γ_3	γ_4	γ_5

A negation function $\ominus$ on $\mathcal{N}$ is *strong* if it verifies $\ominus(\ominus\gamma) = \gamma, \forall \gamma \in \mathcal{N}$. There is only one strong negation on $\mathcal{N}$ and it is given by $\ominus\gamma_i = \gamma_{p-i}$

Given a smooth t-norm $\otimes$ and the strong negation $\ominus$, we can define the *dual* t-conorm $\oplus_\otimes$, as the function satisfying $\gamma_i \oplus_\otimes \gamma_j = \ominus((\ominus\gamma_i) \otimes (\ominus\gamma_j))$.

Proposition 2. *There is one and only one Archimedean smooth t-conorm on $\mathcal{N}$ given by $\gamma_i \oplus \gamma_j = \gamma_{\min\{p,i+j\}}$. Moreover, given any subset J of $\mathcal{N}$ containing γ_0, γ_p, there is one and only one smooth t-conorm $\oplus^J$ on $\mathcal{N}$ that has J as the set of idempotent elements. In fact, if J is the set $J = \{0 = \gamma_{i_0} < \gamma_{i_1} < \cdots < \gamma_{i_{m-1}} < \gamma_{i_m} = 1\}$ such a t-conorm is given by:*

$$\gamma_i \oplus^J \gamma_j = \begin{cases} \gamma_{\min\{i_{k+1}, i+j-i_k\}} & \text{if } \gamma_i, \gamma_j \in [i_k, i_{k+1}] \text{ for some } 0 \leq k \leq m-1 \\ \gamma_{\max\{i,j\}} & \text{otherwise} \end{cases}$$

Note that the Archimedean smooth t-conorm happens with $J = \{\gamma_0, \gamma_p\}$, and that the maximum happens with $J = \mathcal{N}$.

A binary operator $\Rightarrow\colon \mathcal{N}^2 \to \mathcal{N}$ is said to be an *implication*, if it is non-increasing in the first place, non-decreasing in the second place, and satisfies some boundary conditions.

Given a smooth t-norm $\otimes$ and the strong negation $\ominus$, an *S-implication* $\Rightarrow_{s\otimes}$ is the function satisfying $\gamma_i \Rightarrow_{s\otimes} \gamma_j = \ominus(\gamma_i \otimes (\ominus\gamma_j)) = (\ominus\gamma_i) \oplus \gamma_j$.

Proposition 3. *Let $\otimes^J : \mathcal{N}^2 \to \mathcal{N}$ be a smooth t-norm with $J = \{0 = \gamma_{i_0} < \gamma_{i_1} < \cdots < \gamma_{i_{m-1}} < \gamma_{i_m} = 1\}$. Then, the implication $\Rightarrow_{s\otimes}$ is given by:*

$$\gamma_i \Rightarrow_{s\otimes} \gamma_j = \begin{cases} \gamma_{\min\{p-i_k, i_{k+1}+j-i\}} & \text{if } \exists \gamma_{i_k} \in J \text{ such that } \gamma_{i_k} \leq \gamma_i, \gamma_{p-j} \leq \gamma_{i_{k+1}} \\ \gamma_{\max\{p-i,j\}} & \text{otherwise} \end{cases}$$

The Kleene-Dienes implication happens with the minimum t-norm, and the Łukasiewicz implication happens with the Archimedean t-norm.

Given a smooth t-norm $\otimes$, an *R-implication* $\Rightarrow_{r\otimes}$ can be defined as $\gamma_i \Rightarrow_{r\otimes} \gamma_j = \max\{\gamma_k \in \mathcal{N} | (\gamma_i \otimes \gamma_k) \leq \gamma_j\}$, for all $\gamma_i, \gamma_j \in \mathcal{N}$.

Proposition 4. *Let $\otimes^J : \mathcal{N}^2 \to \mathcal{N}$ be a smooth t-norm with $J = \{0 = \gamma_{i_0} < \gamma_{i_1} < \cdots < \gamma_{i_{m-1}} < \gamma_{i_m} = 1\}$. Then, the implication $\Rightarrow_{r\otimes}$ is given by:*

$$\gamma_i \Rightarrow_{r\otimes} \gamma_j = \begin{cases} \gamma_p & \text{if } \gamma_i \leq \gamma_j \\ \gamma_{i_{k+1}+j-i} & \text{if } \exists \gamma_{i_k} \in J \text{ such that } \gamma_{i_k} \leq \gamma_j < \gamma_i \leq \gamma_{i_{k+1}} \\ \gamma_j & \text{otherwise} \end{cases}$$

Example 2. Given the t-norm in Example 1, $\Rightarrow_{r\otimes}$ is defined as follows, where the first column is the antecedent and the first row is the consequent:

	γ_0	γ_1	γ_2	γ_3	γ_4	γ_5
γ_0	γ_5	γ_5	γ_5	γ_5	γ_5	γ_5
γ_1	γ_2	γ_5	γ_5	γ_5	γ_5	γ_5
γ_2	γ_1	γ_2	γ_5	γ_5	γ_5	γ_5
γ_3	γ_0	γ_1	γ_2	γ_5	γ_5	γ_5
γ_4	γ_0	γ_1	γ_2	γ_4	γ_5	γ_5
γ_5	γ_0	γ_1	γ_2	γ_3	γ_4	γ_5

Gödel implication happens with the minimum t-norm, and the Łukasiewicz implication happens with the Archimedean t-norm.

A *QL-implication* is an implication verifying $\gamma_i \Rightarrow \gamma_j = (\ominus\gamma_i) \oplus (\gamma_i \otimes \gamma_j)$.

Proposition 5. *Let $\otimes : \mathcal{N}^2 \to \mathcal{N}$ be a smooth t-norm. The operator $\gamma_i \Rightarrow \gamma_j = (\ominus\gamma_i) \oplus (\gamma_i \otimes \gamma_j)$ is a QL-implication iff $\oplus$ is the Archimedean smooth t-conorm. Moreover, in this case, $\gamma_i \Rightarrow_{ql\otimes} \gamma_j = \gamma_{p-i+z}$ for all $\gamma_i, \gamma_j \in \mathcal{N}$, where $\gamma_z = \gamma_i \otimes \gamma_j$.*

Proposition 6. *Let $\otimes^J : \mathcal{N} \times^J \mathcal{N} \to \mathcal{N}$ be a smooth t-norm with $J = \{0 = \gamma_{i_0} < \gamma_{i_1} < \cdots < \gamma_{i_{m-1}} < \gamma_{i_m} = 1\}$. Then, the implication $\Rightarrow_{ql\otimes}$ is given by:*

$$\gamma_i \Rightarrow_{ql\otimes} \gamma_j = \begin{cases} \gamma_{\max\{p-i+i_k, p+j-i_{k+1}\}} & \text{if } \gamma_i, \gamma_j \in [i_k, i_{k+1}] \text{ for some } 0 \leq k \leq m-1 \\ \gamma_{p-i+j} & \text{if } \gamma_j \leq i_k \leq \gamma_i \text{ for some } i_k \in J \\ \gamma_p & \text{otherwise} \end{cases}$$

The Łukasiewicz implication happens with the minimum t-norm, and the KleeneDienes implication happens with the Archimedean t-norm (note the difference with respect to S-implications).

Interestingly, $\Rightarrow_{s\otimes}$ and $\Rightarrow_{ql\otimes}$ are smooth if and only if so is $\otimes$, but the smoothness condition is not preserved in general for R-implications.

Finally, we can also define D-implications. The name is due to the equivalence to the Dishkant arrow in orthomodular lattices. Note that D-implication are sometimes called NQL-implication. A *D-implication* is an implication satisfying $\gamma_i \Rightarrow \gamma_j = ((\ominus\gamma_i) \otimes (\ominus\gamma_j)) \oplus \gamma_j$ for all $\gamma_i, \gamma_j \in \mathcal{N}$. However, QL-implications and D-implications on $\mathcal{N}$ actually coincide. Given a set J and $\bar{J} = \{\gamma_{p-x} | \gamma_x \in J\}$, then $\Rightarrow_{ql\otimes^J}$ is equivalent to $\Rightarrow_{d\otimes^{\bar{J}}}$.

The notions of fuzzy relation, inverse relation, composition of relations, reflexivity, symmetry and transitivity can trivially be restricted to $\mathcal{N}$.

3 Finite Fuzzy $\mathcal{ALCH}$

In this section we define fuzzy $\mathcal{ALCH}$, a fuzzy extension of $\mathcal{ALCH}$ where:

- Concepts denote fuzzy sets of individuals.
- Roles denote fuzzy binary relations.
- Degrees of truth are taking from a finite chain $\mathcal{N}$.
- Axioms have a degree of truth associated.
- The fuzzy connectives used are a smooth t-norm $\otimes$ on $\mathcal{N}$, the strong negation $\ominus$ on $\mathcal{N}$, the dual t-conorm $\oplus$, and the implications $\Rightarrow_{s\otimes}, \Rightarrow_{r\otimes}, \Rightarrow_{ql\otimes}$.

In this paper, we will assume the reader to be familiar with classical DLs (for details, we refer to [1]).

3.1 Definition

Notation. In the rest of this paper, C, D are (possibly complex) concepts, A is an atomic concept, R is a role, a, b are individuals, $\bowtie \in \{\geq, <, \leq, >\}$, $\lhd \in \{\geq, >\}$, $\rhd \in \{\leq, <\}$. We will also use $\equiv$ to denote semantical equivalence, and we will not write $\otimes$ in the subscripts of the implications.

Syntax. Finite fuzzy $\mathcal{ALCH}$ assumes three alphabets of symbols, for concepts, roles and individuals. A *Fuzzy Knowledge Base* (KB) contains a finite set of axioms organized in a fuzzy ABox $\mathcal{A}$ (axioms about individuals), a fuzzy TBox $\mathcal{T}$ (axioms about concepts), and a fuzzy RBox $\mathcal{T}$ (axioms about roles).

The syntax of fuzzy concept, roles, and axioms are shown in Table 2. Note that in fuzzy $\mathcal{ALCH}$, all fuzzy roles are atomic.

Remark 1. As opposed to the crisp case, there are three types of universal restrictions, fuzzy GCIs, and fuzzy RIAs. In fact, the different subscripts $_s$, $_r$, and $_{ql}$ denote an S-implication, R-implication, and QL-implication, respectively.

Semantics. A fuzzy interpretation $\mathcal{I}$ is a pair $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is a non empty set (the interpretation domain) and $\cdot^{\mathcal{I}}$ is a fuzzy interpretation function mapping *(i)* every individual a onto an element $a^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, *(ii)* every concept C onto a function $C^{\mathcal{I}} : \Delta^{\mathcal{I}} \to \mathcal{N}$, and *(iii)* every role R onto a function $R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to \mathcal{N}$. The fuzzy interpretation function is extended to fuzzy *complex concepts* and *axioms* as shown in Table 2.

$C^{\mathcal{I}}$ denotes the membership function of the fuzzy concept C with respect to the fuzzy interpretation $\mathcal{I}$. $C^{\mathcal{I}}(x)$ gives us the degree of being x an element of the fuzzy concept C under $\mathcal{I}$. Similarly, $R^{\mathcal{I}}$ denotes the membership function of the fuzzy role R with respect to $\mathcal{I}$. $R^{\mathcal{I}}(x, y)$ gives us the degree of being (x, y) an element of the fuzzy role R.

Remark 2. Note an important difference with previous work in fuzzy DLs. Usually, $\cdot^{\mathcal{I}}$ maps every concept C onto a function $C^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0, 1]$, and every role R onto $R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0, 1]$. Consequently, a fuzzy KB $\{\langle a : C > 0.5\rangle, \langle a : C < 0.75\}$ is satisfiable, by taking $C^{\mathcal{I}}(a) \in (0.5, 0.75)$. But now, given $\mathcal{N} = \{\texttt{false}, \texttt{closeToFalse}, \texttt{neutral}, \texttt{closeToTrue}, \texttt{true}\}$, a fuzzy KB $\{\langle a : C > \texttt{closeToFalse}\rangle, \langle a : C < \texttt{neutral}\}$ is not satisfiable, since $C^{\mathcal{I}}(a) \in \mathcal{N}$.

Witnessed models. In order to correctly manage infima and suprema in the reasoning, we need to define the notion of *witnessed* interpretations [7]. A fuzzy interpretation $\mathcal{I}$ is *witnessed* iff, for every formula, the infimum corresponds to the minimum and the supremum corresponds to the maximum. Our logic also enjoys the Witnessed Model Property (WMP) (all models are witnessed), because the number of degrees of truth in the models of the logic is finite [7].

Reasoning tasks. We will define the most important reasoning tasks and show that all of them can be reduced to fuzzy KB satisfiability.

- *Fuzzy KB satisfiability.* A fuzzy interpretation $\mathcal{I}$ *satisfies* (is a model of) a fuzzy KB $\mathcal{K} = \langle \mathcal{A}, \mathcal{T}, \mathcal{R}\rangle$ iff it satisfies each element in $\mathcal{A}$, $\mathcal{T}$ and $\mathcal{R}$.
- *Concept satisfiability.* C is α-satisfiable w.r.t. a fuzzy KB $\mathcal{K}$ iff $\mathcal{K} \cup \{\langle a : C \geq \alpha\rangle\}$ is satisfiable, where a is a new individual, which does not appear in $\mathcal{K}$.
- *Entailment.* A fuzzy concept assertion $\langle a : C \bowtie \alpha\rangle$ is entailed by a fuzzy KB $\mathcal{K}$ (denoted $\mathcal{K} \models \langle a : C \bowtie \alpha\rangle$) iff $\mathcal{K} \cup \{\langle a : C \neg\bowtie \alpha\rangle\}$ is unsatisfiable. Furthermore, $\mathcal{K} \models \langle (a, b) : R \geq \alpha\rangle$ iff $\mathcal{K} \cup \{\langle b : B \geq \gamma_p\rangle\} \models \langle a : \exists R.B \geq \alpha\rangle$, where B is a new concept.

Table 2. Syntax and semantics of finite fuzzy $\mathcal{ALCH}$

Element	Syntax	Semantics
Concepts	$\top$	γ_p
	$\bot$	γ_0
	A	$A^{\mathcal{I}}(x)$
	$C \sqcap D$	$C^{\mathcal{I}}(x) \otimes D^{\mathcal{I}}(x)$
	$C \sqcup D$	$C^{\mathcal{I}}(x) \oplus D^{\mathcal{I}}(x)$
	$\neg C$	$\ominus C^{\mathcal{I}}(x)$
	$\forall_s R.C$	$\inf_{y \in \Delta^{\mathcal{I}}} \{R^{\mathcal{I}}(x,y) \Rightarrow_s C^{\mathcal{I}}(y)\}$
	$\forall_r R.C$	$\inf_{y \in \Delta^{\mathcal{I}}} \{R^{\mathcal{I}}(x,y) \Rightarrow_r C^{\mathcal{I}}(y)\}$
	$\forall_{ql} R.C$	$\inf_{y \in \Delta^{\mathcal{I}}} \{R^{\mathcal{I}}(x,y) \Rightarrow_{ql} C^{\mathcal{I}}(y)\}$
	$\exists R.C$	$\sup_{y \in \Delta^{\mathcal{I}}} \{R^{\mathcal{I}}(x,y) \otimes C^{\mathcal{I}}(y)\}$
Roles	R	$R^{\mathcal{I}}(x,y)$
ABox axioms	$\langle a\!:\!C \bowtie \gamma \rangle$	$C^{\mathcal{I}}(a^{\mathcal{I}}) \bowtie \gamma$
	$\langle (a,b)\!:\!R \bowtie \gamma \rangle$	$R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) \bowtie \gamma$
TBox axioms	$\langle C \sqsubseteq_s D \rhd \gamma \rangle$	$\inf_{x \in \Delta^{\mathcal{I}}} \{C^{\mathcal{I}}(x) \Rightarrow_s D^{\mathcal{I}}(x)\} \rhd \gamma$
	$\langle C \sqsubseteq_r D \rhd \gamma \rangle$	$\inf_{x \in \Delta^{\mathcal{I}}} \{C^{\mathcal{I}}(x) \Rightarrow_r D^{\mathcal{I}}(x)\} \rhd \gamma$
	$\langle C \sqsubseteq_{ql} D \rhd \gamma \rangle$	$\inf_{x \in \Delta^{\mathcal{I}}} \{C^{\mathcal{I}}(x) \Rightarrow_{ql} D^{\mathcal{I}}(x)\} \rhd \gamma$
RBox axioms	$\langle R_1 \sqsubseteq_s R_2 \rhd \gamma \rangle$	$\inf_{x,y \in \Delta^{\mathcal{I}}} \{R_1^{\mathcal{I}}(x) \Rightarrow_s R_2^{\mathcal{I}}(x)\} \rhd \gamma$
	$\langle R_1 \sqsubseteq_r R_2 \rhd \gamma \rangle$	$\inf_{x,y \in \Delta^{\mathcal{I}}} \{R_1^{\mathcal{I}}(x) \Rightarrow_r R_2^{\mathcal{I}}(x)\} \rhd \gamma$
	$\langle R_1 \sqsubseteq_{ql} R_2 \rhd \gamma \rangle$	$\inf_{x,y \in \Delta^{\mathcal{I}}} \{R_1^{\mathcal{I}}(x) \Rightarrow_{ql} R_2^{\mathcal{I}}(x)\} \rhd \gamma$

- *Greatest lower bound.* The greatest lower bound of a concept or role assertion τ is defined as the $\sup\{\alpha : \mathcal{K} \models \langle \tau \geq \alpha \rangle\}$. It can be computed performing at most $\log |\mathcal{N}|$ entailment tests [16].
- *Concept subsumption*: Under an S-implication, D subsumes C with degree α ($C \sqsubseteq_s D \geq \alpha$) w.r.t. a fuzzy KB $\mathcal{K}$ iff $\mathcal{K} \cup \{a\!:\!\neg C \sqcup D < \alpha\}$ is unsatisfiable, where a is a new individual. Under an R-implication, D subsumes C ($C \sqsubseteq_r D$) w.r.t. a fuzzy KB $\mathcal{K}$ iff, for every $\alpha \in \mathcal{N}$, $\mathcal{K} \cup \{a\!:\!C \geq \alpha\} \cup \{a\!:\!D < \alpha\}$ is unsatisfiable, where a is a new individual. Under a QL-implication, D subsumes C with degree α ($C \sqsubseteq_{ql} D \geq \alpha$) w.r.t. a fuzzy KB $\mathcal{K}$ iff $\mathcal{K} \cup \{a\!:\!\neg C \sqcup (C \sqcap D) < \alpha\}$ is unsatisfiable, where a is a new individual.

3.2 Logical Properties

It can be easily shown that finite fuzzy $\mathcal{ALCH}$ is a sound extension of crisp $\mathcal{ALCH}$, because fuzzy interpretations coincide with crisp interpretations if we restrict the membership degrees to $\{\gamma_0 = 0, \gamma_p = 1\}$.

Proposition 7. *Finite fuzzy $\mathcal{ALCH}$ interpretations coincide with crisp interpretations if we restrict the membership degrees to $\{\gamma_0 = 0, \gamma_p = 1\}$.*

The following properties are extensions to a finite chain $\mathcal{N}$ of properties for Zadeh fuzzy DLs [3] and Łukasiewicz fuzzy DLs [6].

1. *Concept simplification*: $C \sqcap \top \equiv C$, $C \sqcup \bot \equiv C$,$C \sqcap \bot \equiv \bot$, $C \sqcup \top \equiv \top$, $\exists R.\bot \equiv \bot$, $\forall_s R.\top \equiv \top$, $\forall_r R.\top \equiv \top$, $\forall_{ql} R.\top \equiv \top$.
2. *Involutive* negation: $\neg\neg C \equiv C$,
3. *Excluded middle and contradiction*: In general, $C \sqcup \neg C \not\equiv \top$, $C \sqcap \neg C \not\equiv \bot$,
4. *Idempotence* of conjunction/disjunction: In general, $C \sqcap C \not\equiv C$, $C \sqcup C \not\equiv C$.

5. *De Morgan* laws: $\neg(C \sqcup D) \equiv \neg C \sqcap \neg D$, $\neg(C \sqcap D) \equiv \neg C \sqcup \neg D$,
6. *Inter-definability of concepts*: $\bot \equiv \neg\top$, $\top \equiv \neg\bot$, $C \sqcap D \equiv \neg(\neg C \sqcap \neg D)$, $C \sqcup D \equiv \neg(\neg C \sqcup \neg D)$, $\forall_s R.C \equiv \neg\exists R.(\neg C)$, $\exists R.C \equiv \neg\forall_s R.(\neg C)$. However, in general, $C \sqcap D \not\equiv \neg(\neg C \sqcup \neg D)$, $C \sqcup D \not\equiv \neg(\neg C \sqcap \neg D)$, $\forall_r R.C \not\equiv \neg\exists R.(\neg C)$, $\exists R.C \not\equiv \neg\forall_r R.(\neg C)$, $\forall_{ql} R.C \not\equiv \neg\exists R.(\neg C)$, $\exists R.C \not\equiv \neg\forall_{ql} R.(\neg C)$.
7. *Inter-definability of axioms*: $\langle \tau > \beta \rangle \equiv \langle \tau > +\beta \rangle$, $\langle \tau < \alpha \rangle \equiv \langle \tau \leq -\alpha \rangle$.
8. *Contrapositive symmetry*: $C \sqsubseteq_s D \equiv \neg D \sqsubseteq_s \neg C$. However, in general, $C \sqsubseteq_r D \not\equiv \neg D \sqsubseteq_r \neg_s C$,$C \sqsubseteq_{ql} D \not\equiv \neg D \sqsubseteq_{ql} \neg_s C$.
9. *Modus ponens*: $\langle a : C \rhd \gamma_1 \rangle$ and $\langle C \sqsubseteq_r D \rhd \gamma_2 \rangle$ imply $\langle a : D \rhd \gamma_1 \otimes \gamma_2 \rangle$, $\langle (a,b) : R \rhd \gamma_1 \rangle$ and $\langle R \sqsubseteq_r R' \rhd \gamma_2 \rangle$ imply $\langle (a,b) : R' \rhd \gamma_1 \otimes \gamma_2 \rangle$.
10. *Self-subsumption*: $(C \sqsubseteq_r C)^{\mathcal{I}} = \gamma_p$, $(R \sqsubseteq_r R)^{\mathcal{I}} = \gamma_p$. However, in general, $(C \sqsubseteq_s C)^{\mathcal{I}} \neq \gamma_p$, $(R \sqsubseteq_s R)^{\mathcal{I}} \neq \gamma_p$, and $(C \sqsubseteq_{ql} C)^{\mathcal{I}} \neq \gamma_p$, $(R \sqsubseteq_{ql} R)^{\mathcal{I}} \neq \gamma_p$.

Remark 3. Inter-definability of axioms makes it possible to restrict to fuzzy axioms of the forms $\langle \tau \geq \alpha \rangle$ and $\langle \tau \leq \beta \rangle$.

4 A Crisp Representation for Finite Fuzzy $\mathcal{ALCH}$

In this section we show how to reduce a fuzzy KB into a crisp KB. The procedure is satisfiability-preserving, so existing DL reasoners could be applied to the resulting KB. The basic idea is to create some new crisp concepts and roles, representing the α-cuts of the fuzzy concepts and relations, and to rely on them. Next, some new axioms are added to preserve their semantics and finally every axiom in the ABox, the TBox and the RBox is represented, independently from other axioms, using these new crisp elements.

Before proceeding formally, we will illustrate this idea with an example.

Example 3. Consider the smooth t-norm on $\mathcal{N}$ used in Example 1, and let us compute some α-cuts of the fuzzy concept $A_1 \sqcap A_2$ (denoted $\rho(A_1 \sqcap A_2, \geq \alpha)$).

To begin with, let us consider $\alpha = \gamma_2$. By definition, this set includes the elements of the domain x satisfying $A_1^{\mathcal{I}}(x) \otimes A_2^{\mathcal{I}}(x) \geq \gamma_2$. There are two possibilities: *(i)* $A_1^{\mathcal{I}}(x) \geq \gamma_2$ and $A_2^{\mathcal{I}}(x) \geq \gamma_3$, or *(ii)* $A_1^{\mathcal{I}}(x) \geq \gamma_3$ and $A_2^{\mathcal{I}}(x) \geq \gamma_2$. Hence, $\rho(A_1 \sqcap A_2, \geq \gamma_2) = \Big(\rho(A_1, \geq \gamma_2) \sqcap \rho(A_2, \geq \gamma_3)\Big) \sqcup \Big(\rho(A_1, \geq \gamma_3) \sqcap \rho(A_2, \geq \gamma_2)\Big)$.

Now, let us consider $\alpha = \gamma_3$. Now, there is only one possibility: $A_1^{\mathcal{I}}(a^{\mathcal{I}}) \geq \gamma_3$ and $A_2^{\mathcal{I}}(a^{\mathcal{I}}) \geq \gamma_3$. Hence, $\rho(A_1 \sqcap A_2, \geq \gamma_3) = \rho(A_1, \geq \gamma_3) \sqcap \rho(A_2, \geq \gamma_3)$.

Observe that for idempotent degrees ($\alpha \in J$) the case is the same as in finite Zadeh and Gödel fuzzy logics [3,5], whereas for non-idempotent degrees the case is similar as in finite Łukasiewicz fuzzy logic [6].

4.1 Adding New Elements

Let $\mathbf{A}$ be the set of atomic fuzzy concepts and $\mathbf{R}$ the set of atomic fuzzy roles in a fuzzy KB $\mathcal{K} = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$, respectively. For each $\alpha \in \mathcal{N}^+$, for each $A \in \mathbf{A}$, a new atomic concepts $A_{\geq\alpha}$ is introduced. $A_{\geq\alpha}$ represents the crisp set of individuals which are instance of A with degree higher or equal than α i.e the α-cut of A. Similarly, for each $R \in \mathbf{R}$, a new atomic role $R_{\geq\alpha}$ is created.

Remark 4. The atomic elements $A_{\geq\gamma_0}$ and $R_{\geq\gamma_0}$ are not considered because they are always equivalent to the $\top$ concept. Also, as opposite to previous works [3,5,6] we are not introducing elements of the forms $A_{>\beta}$ and $R_{>\beta}$ (for each $\beta \in \mathcal{N} \setminus \{\gamma_p\}$), since now $A_{>\gamma_i}$ is equivalent to $A_{\geq\gamma_{i+1}}$, and $R_{>\gamma_i}$ is equivalent to $R_{\geq\gamma_{i+1}}$.

The semantics of these newly introduced atomic concepts and roles is preserved by some terminological and role axioms. For each $1 \leq i \leq p-1$ and for each $A \in \mathbf{A}$, $T(\mathcal{N})$ is the smallest terminology containing these axioms: $A_{\geq\gamma_{i+1}} \sqsubseteq A_{\geq\gamma_i}$. Similarly, for each $R_A \in \mathbf{R}$, $R(\mathcal{N})$ is the smallest terminology containing these axioms: $R_{\geq\gamma_{i+1}} \sqsubseteq R_{\geq\gamma_i}$.

Remark 5. Again, note that the number of new axioms needed here is less than the number needed in similar works [3,5,6], since we do not need to deal with elements of the forms $A_{>\beta}$ and $R_{>\beta}$.

4.2 Mapping Fuzzy Concepts, Roles and Axioms

Fuzzy concept and role expressions are reduced using mapping ρ, as shown in the top part of Table 3. Given a fuzzy concept C, $\rho(C, \geq \alpha)$ is a crisp set containing all the elements which belong to C with a degree greater or equal than α. The other cases $\rho(C, \bowtie \gamma)$ are similar. ρ is defined in a similar way for fuzzy roles. Furthermore, axioms are reduced as in the bottom part of Table 3, where $\kappa(\tau)$ maps a fuzzy axiom τ in finite fuzzy $\mathcal{ALCH}$ into a set of crisp axioms in $\mathcal{ALCH}$.

The reduction of the conjunction considers every pair $\gamma_x, \gamma_y \in (\gamma_{i_k}, \gamma_{i_{k+1}}]$ such that $\alpha \in (\gamma_{i_k}, \gamma_{i_{k+1}}]$, and $x + y = i_{k+1} + z$, with $\alpha = \gamma_z$. Note that the reduction does not consider a closed internal of the form $[\gamma_{i_k}, \gamma_{i_{k+1}}]$. The reason is that, if α is idempotent and we set $\gamma_{i_{k+1}} = \alpha$, the result is correct ($\gamma_x = \gamma_y = \alpha$). However, setting $\gamma_{i_k} = \alpha$ would yield an incorrect result. Similarly, the reduction of the disjunction also considers a closed interval.

When dealing with R-implications and QL-implications, we consider optimal pairs of elements, to get efficient representation that avoids superfluous elements.

Definition 1. *Let $\Rightarrow$ be an implication in $\mathcal{N}$, and let $\gamma_x, \gamma_y \in \mathcal{N}^+$. (γ_x, γ_y) is a* ($\Rightarrow_{\geq\alpha}$)-optimal *pair iff* (i) $\gamma_x \Rightarrow \gamma_y \geq \alpha$, (ii) *there are no $\gamma'_x, \gamma'_y \in \mathcal{N}^+$ such that $\gamma'_x \Rightarrow \gamma'_y \geq \alpha$, and such that either $\gamma'_x < \gamma_x$ or $\gamma'_y < \gamma_y$.*

Definition 2. *Let $\Rightarrow$ be an implication in $\mathcal{N}$, and let $\gamma_x \in \mathcal{N}^+, \gamma_y \in \mathcal{N}$. (γ_x, γ_y) is a* ($\Rightarrow_{\leq\beta}$)-optimal *pair iff* (i) $\gamma_x \Rightarrow \gamma_y \leq \beta$, (ii) *there are no $\gamma'_x, \gamma'_y \in \mathcal{N}^+$ such that $\gamma'_x \Rightarrow \gamma'_y \leq \beta$, and such that either $\gamma'_x < \gamma_x$ or $\gamma'_y > \gamma_y$.*

Example 4. Given the R-implication in Example 2, the ($\Rightarrow_{\geq\gamma_3}$)-optimal pairs are (γ_3, γ_3), (γ_2, γ_2), and (γ_1, γ_1); and the ($\Rightarrow_{\leq\gamma_3}$)-optimal pairs are (γ_5, γ_3), (γ_3, γ_2), (γ_2, γ_1), and (γ_1, γ_0).

Note that R-implications are, in general, non smooth (see Example 2). Hence, a pair of elements γ_1, γ_y such that $\gamma_x \Rightarrow_r \gamma_y = \alpha$ might not exist, and thus we have to consider an inequality of the form $\gamma_x \Rightarrow_r \gamma_y \geq \alpha$. In QL-implications, due to the optimality condition, $=$ and $\geq$ yield the same result.

Table 3. Mapping of concepts, roles, and axioms

$\rho(\top, \geq \alpha)$ $\rho(\top, \leq \beta)$	$\top$ $\bot$
$\rho(\bot, \geq \alpha)$ $\rho(\bot, \leq \beta)$	$\bot$ $\top$
$\rho(A, \geq \alpha)$ $\rho(A, \leq \beta)$	$A_{\geq \alpha}$ $\neg A_{\geq +\beta}$
$\rho(\neg C, \bowtie \gamma)$	$\rho(C, \bowtie^{-} \ominus \gamma)$
$\rho(C \sqcap D, \geq \alpha)$	$\bigsqcup_{\gamma_x, \gamma_y} \{\rho(C, \geq \gamma_x) \sqcap \rho(D, \geq \gamma_y)\}$ for every pair γ_x, γ_y such that $\alpha, \gamma_x, \gamma_y \in (\gamma_{i_k}, \gamma_{i_{k+1}}]$, and $x + y = i_{k+1} + z$, with $\gamma_z = \alpha$
$\rho(C \sqcap D, \leq \beta)$	$\rho(\neg C \sqcup \neg D, \geq \ominus \beta)$
$\rho(C \sqcup D, \geq \alpha)$	$\rho(C, \geq \alpha) \sqcup \rho(D, \geq \alpha) \bigsqcup_{\gamma_x, \gamma_y} \{\rho(C, \geq \gamma_x) \sqcap \rho(D, \geq \gamma_y)\}$ for every pair γ_x, γ_y such that $\alpha, \gamma_x, \gamma_y \in (\gamma_{i_k}, \gamma_{i_{k+1}}]$, and $x + y = i_k + z$, with $\gamma_z = \alpha$
$\rho(C \sqcup D, \leq \beta)$	$\rho(\neg C \sqcap \neg D, \geq \ominus \beta)$
$\rho(\exists R.C, \geq \alpha)$	$\bigsqcup_{\gamma_x, \gamma_y} \{\exists \rho(R, \geq \gamma_x).\rho(C, \geq \gamma_y)\}$ for every pair $\gamma_x, \gamma_y \in (\gamma_{i_k}, \gamma_{i_{k+1}}]$ such that $\gamma \in (\gamma_{i_k}, \gamma_{i_{k+1}}]$, and $x + y = i_{k+1} + z$, with $\gamma_z = \alpha$
$\rho(\exists R.C, \leq \beta)$	$\rho(\forall_s R.(\neg C), \geq \ominus \beta)$
$\rho(\forall_s R.C, \geq \alpha)$	$\bigsqcap_{\gamma_x, \gamma_y} \{\forall \rho(R, \geq \gamma_x).\rho(C, \geq \gamma_y)\}$ for every pair γ_x, γ_y such that $\gamma_x \in (\gamma_{i_k}, \gamma_{i_{k+1}}], \alpha, \gamma_y \in (\gamma_{p-i_{k+1}}, \gamma_{p-i_k}]$, and $y - i = z - i_{k+1}$, with $\gamma_z = \alpha$
$\rho(\forall_s R.C, \leq \beta)$	$\rho(\exists R.(\neg C), \geq \ominus \beta)$
$\rho(\forall_r R.C, \geq \alpha)$	$\bigsqcap_{\gamma_x, \gamma_y} \{\forall \rho(R, \geq \gamma_x).\rho(C, \geq \gamma_y)\}$ for every pair $\gamma_x, \gamma_y \in \mathcal{N}^+$ such that γ_x, γ_y are $(\Rightarrow_{r\ \geq \alpha})$-optimal
$\rho(\forall_r R.C, \leq \beta)$	$\bigsqcup_{\gamma_x, \gamma_y} \{\exists \rho(R, \geq \gamma_x).\rho(C, \leq \gamma_y)\}$ for every pair $\gamma_x \in \mathcal{N}^+, \gamma_y \in \mathcal{N}$ such that γ_x, γ_y are $(\Rightarrow_{r\ \leq \beta})$-optimal
$\rho(\forall_{ql} R.C, \geq \alpha)$	$\bigsqcap_{\gamma_x, \gamma_y} \{\forall \rho(R, \geq \gamma_x).\rho(C, \geq \gamma_y)\}$ for every pair $\gamma_x, \gamma_y \in \mathcal{N}^+$ such that γ_x, γ_y are $(\Rightarrow_{ql\ \geq \alpha})$-optimal
$\rho(\forall_{ql} R.C, \leq \beta)$	$\bigsqcup_{\gamma_x, \gamma_y} \{\exists \rho(R, \geq \gamma_x).\rho(C, \leq \gamma_y)\}$ for every pair $\gamma_x \in \mathcal{N}^+, \gamma_y \in \mathcal{N}$ such that γ_x, γ_y are $(\Rightarrow_{ql\ \leq \beta})$-optimal
$\rho(R, \geq \alpha)$ $\rho(R, \leq \beta)$	$R_{\geq \alpha}$ $\neg R_{\geq +\beta}$
$\kappa(\langle a : C \bowtie \gamma \rangle)$	$\{a : \rho(C, \bowtie \gamma)\}$
$\kappa(\langle (a, b) : R \bowtie \gamma \rangle)$	$\{(a, b) : \rho(R, \bowtie \gamma)\}$
$\kappa(\langle C \sqsubseteq_s D \geq \alpha \rangle)$	$\bigsqcup \{\rho(C, \geq \gamma_x) \sqsubseteq \rho(D, \geq \gamma_y)\}$ for every pair γ_x, γ_y such that $\gamma_x \in (\gamma_{i_k}, \gamma_{i_{k+1}}], \alpha, \gamma_y \in (\gamma_{p-i_{k+1}}, \gamma_{p-i_k}]$, and $y - \gamma_i = z - \gamma_{i_{k+1}}$, with $\gamma_z = \alpha$
$\kappa(\langle C \sqsubseteq_r D \geq \alpha \rangle)$	$\bigsqcup \{\rho(C, \geq \gamma_x) \sqsubseteq \rho(D, \geq \gamma_y)\}$ for every pair $\gamma_x, \gamma_y \in \mathcal{N}^+$ such that γ_x, γ_y are $(\Rightarrow_{r\ \geq \alpha})$-optimal
$\kappa(\langle C \sqsubseteq_{ql} D \geq \alpha \rangle)$	$\bigsqcup \{\forall \rho(C, \geq \gamma_x) \sqsubseteq \rho(D, \geq \gamma_y)\}$ for every pair $\gamma_x, \gamma_y \in \mathcal{N}^+$ such that γ_x, γ_y are $(\Rightarrow_{ql\ \geq \alpha})$-optimal
$\kappa(\langle R_1 \sqsubseteq_s R_2 \geq \alpha \rangle)$	$\bigsqcup \{\rho(R_1, \geq \gamma_x) \sqsubseteq \rho(R_2, \geq \gamma_y)\}$ for every pair γ_x, γ_y such that $\gamma_x \in (\gamma_{i_k}, \gamma_{i_{k+1}}], \alpha, \gamma_y \in (\gamma_{p-i_{k+1}}, \gamma_{p-i_k}]$, and $y - \gamma_i = z - \gamma_{i_{k+1}}$, with $\gamma_z = \alpha$
$\kappa(\langle R_1 \sqsubseteq_r R_2 \geq \alpha \rangle)$	$\bigsqcup \{\rho(R_1, \geq \gamma_x) \sqsubseteq \rho(R_2, \geq \gamma_y)\}$ for every pair $\gamma_x, \gamma_y \in \mathcal{N}^+$ such that γ_x, γ_y are $(\Rightarrow_{r\ \geq \alpha})$-optimal
$\kappa(\langle R_1 \sqsubseteq_{ql} R_2 \geq \alpha \rangle)$	$\bigsqcup \{\rho(R_1, \geq \gamma_x) \sqsubseteq \rho(R_2, \geq \gamma_y)\}$ for every pair $\gamma_x, \gamma_y \in \mathcal{N}^+$ such that γ_x, γ_y are $(\Rightarrow_{ql\ \geq \alpha})$-optimal

$\kappa(\mathcal{A})$ (resp. $\kappa(\mathcal{T})$, $\kappa(\mathcal{R})$) denotes the union of the reductions of every axiom in $\mathcal{A}$ (resp. $\mathcal{T}$, $\mathcal{R}$). $\mathsf{crisp}(\mathcal{K})$ denotes the reduction of a fuzzy KB $\mathcal{K}$. A fuzzy KB $\mathcal{K} = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ is reduced into a KB $\mathsf{crisp}(\mathcal{K}) = \langle \kappa(\mathcal{A}), T(\mathcal{N}) \cup \kappa(\mathcal{T}), R(\mathcal{N}) \cup \kappa(\mathcal{R}) \rangle$.

4.3 Properties of the Reduction

Correctness. The following theorem, showing the logic is decidable and that the reductions preserves reasoning, can be shown.

Theorem 1. *The satisfiability problem in finite fuzzy $\mathcal{ALCH}$ is decidable. Furthermore, a finite fuzzy $\mathcal{ALCH}$ fuzzy KB $\mathcal{K}$ is satisfiable iff $\mathsf{crisp}(\mathcal{K})$ is.*

Complexity. In general, the size of $\mathsf{crisp}(\mathcal{K})$ is $\mathcal{O}(|\mathcal{K}| \cdot |\mathcal{N}|^k)$, being k the maximal depth of the concepts appearing in $\mathcal{K}$. In the particular case of finite Zadeh fuzzy logic, the size of $\mathsf{crisp}(\mathcal{K})$ is $\mathcal{O}(|\mathcal{K}| \cdot |\mathcal{N}|)$ [3]. For other fuzzy operators the case is more complex because we cannot infer the exact values of the degrees of truth, so we need to build disjunctions or conjunctions over all possible degrees of truth.

Modularity. The reduction of an ontology can be reused when adding new axioms if they do not introduce new atomic concepts and roles. In this case, it remains to add the reduction of the new axioms. This allows to compute the reduction of the ontology off-line and update $\mathsf{crisp}(\mathcal{K})$ incrementally. The assumption that the basic vocabulary is fully expressed in the ontology is reasonable because ontologies do not usually change once that their development has finished.

5 Conclusions and Future Work

This paper has set a general framework for fuzzy DLs with a finite chain of degrees of truth $\mathcal{N}$. $\mathcal{N}$ can be seen as a finite totally ordered set of linguistic terms or labels. This is very useful in practice, since expert knowledge is usually expressed using linguistic terms and avoiding their numerical interpretations.

Starting from a smooth finite t-norm on $\mathcal{N}$, we define the syntax and semantics of fuzzy $\mathcal{ALCH}$. The negation function and the t-conorm are imposed by the choice of the t-norm, but there are different options for the implication function. For this reason, whenever this is possible (i.e., in universal restriction concepts and in inclusion axioms), the language allows to use three different implications. We have studies some of the logical properties of the logic. This will help the ontology developers to use the implication that better suit their needs.

The decidability of the logic has been shown by presenting a reasoning preserving reduction to the crisp case. Providing a crisp representation for a fuzzy ontology allows reusing current crisp ontology languages and reasoners, among other related resources. The complexity of the crisp representation is higher than in finite Zadeh fuzzy DLs, because it is necessary to build disjunctions or conjunctions over all possible degrees of truth. However, Zadeh fuzzy DLs have some logical problems [3] which may not be acceptable in some applications, where alternative operators such as those introduced in this paper could be used.

As future work we will study more expressive logics than $\mathcal{ALCH}$, applying the ideas in the previous work DLs [3,5,6], with the aim of providing the theoretical basis of a fuzzy extension of OWL 2 under finite chain of degrees of truth.

Acknowledgement

F. Bobillo acknowledges support from the Spanish Ministry of Science and Technology (project TIN2009-14538-C02-01) and Ministry of Education (program José Castillejo, grant JC2009-00337).

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F.: The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
2. Lukasiewicz, T., Straccia, U.: Managing uncertainty and vagueness in description logics for the semantic web. Journal of Web Semantics **6**(4) (2008) 291–308
3. Bobillo, F., Delgado, M., Gómez-Romero, J.: Crisp representations and reasoning for fuzzy ontologies. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **17**(4) (2009) 501–530
4. Cerami, M., Esteva, F., Bou, F.: Decidability of a description logic over infinite-valued product logic. Proceedings of the 12th International Conference on Principles of Knowledge Representation and Reasoning (KR 2010) 203–213
5. Bobillo, F., Delgado, M., Gómez-Romero, J., Straccia, U.: Fuzzy description logics under Gödel semantics. Int. J. Approximate Reasoning **50**(3) (2009) 494–514
6. Bobillo, F., Straccia, U.: Towards a crisp representation of fuzzy description logics under Łukasiewicz semantics. Proceedings of the 17th International Symposium on Methodologies for Intelligent Systems (ISMIS 2008). Volume 4994 of Lecture Notes in Computer Science, Springer-Verlag (2008) 309–318
7. Hájek, P.: Making fuzzy description logic more general. Fuzzy Sets and Systems **154**(1) (2005) 1–15
8. García-Cerdaña, A., Armengol, E., Esteva, F.: Fuzzy description logics and t-norm based fuzzy logics. Int. J. Approximate Reasoning **51** (2010) 632–655
9. Cerami, M., García-Cerdaña, A.,, Esteva, F.: From classical description logic to n–graded fuzzy description logic. In: Proceedings of the 19th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2010), IEEE Press (2010) 1506–1513
10. Mas, M., Monserrat, M., Torrens, J.: S-implications and r-implications on a finite chain. Kybernetika **40**(1) (2004) 3–20
11. Mayor, G., Torrens, J.: On a class of operators for expert systems. International Journal of Intelligent Systems **8**(7) (1993) 771–778
12. Mas, M., Monserrat, M., Torrens, J.: On two types of discrete implications. International Journal of Approximate Reasoning **40**(3) (2005) 262–279
13. Straccia, U.: Description logics over lattices. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **14**(1) (2006) 1–16
14. Jiang, Y., Tang, Y., Wang, J., Deng, P., Tang, S.: Expressive fuzzy description logics over lattices. Knowledge-Based Systems **23**(2) (2010) 150–161
15. Zadeh, L.A.: Fuzzy sets. Information and Control **8** (1965) 338–353
16. Straccia, U.: Reasoning within fuzzy description logics. Journal of Artificial Intelligence Research **14** (2001) 137–166

PR-OWL 2.0 - Bridging the gap to OWL semantics

Rommel N. Carvalho, Kathryn B. Laskey, and Paulo C.G. Costa

Center of Excellence in C4I,
George Mason University, USA
rommel.carvalho@gmail.com, {klaskey,pcosta}@gmu.edu
http://www.gmu.edu

Abstract. The past few years have witnessed an increasingly mature body of research on the Semantic Web, with new standards being developed and more complex use cases being proposed and explored. As complexity increases in SW applications, so does the need for principled means to cope with uncertainty inherent to real world SW applications. Not surprisingly, several approaches addressing uncertainty representation and reasoning on the Semantic Web have emerged [3, 4, 6, 7, 10, 11, 13, 14]. For example, PR-OWL [3] provides OWL constructs for representing Multi-Entity Bayesian Network (MEBN) [8] theories. This paper reviews some shortcomings of PR-OWL 1 [2] and describes how they will be addressed in PR-OWL 2. A method is presented for mapping back and forth from triples into random variables (RV). The method applies to triples representing both predicates and functions. A complex example is given for mapping an n-ary relation using the proposed schematic.

Keywords: uncertainty reasoning, OWL, PR-OWL, MEBN, probabilistic ontology, Semantic Web, compatibility.

1 Introduction

Appreciation is growing within the Semantic Web community of the need to represent and reason with uncertainty. In recognition of this need, the World Wide Web Consortium (W3C) created the Uncertainty Reasoning for the World Wide Web Incubator Group (URW3-XG) in 2007 to identify requirements for reasoning with and representing uncertain information in the World Wide Web. The URW3-XG concluded that standardized representations were needed to express uncertainty in Web-based information [9]. A candidate representation for uncertainty reasoning in the Semantic Web is Probabilistic OWL (PR-OWL) [3], an OWL upper ontology for representing probabilistic ontologies based on Multi-Entity Bayesian Networks (MEBN) [8].

Compatibility with OWL was a major design goal for PR-OWL [3]. However, there are several ways in which the initial release of PR-OWL falls short of complete compatibility. First, there is no mapping in PR-OWL to properties of OWL. Second, although PR-OWL has the concept of meta-entities, which allows

the definition of complex types, it lacks compatibility with existing types already present in OWL.

These problems have been noted in the literature [12]:

> PR-OWL does not provide a proper integration of the formalism of MEBN and the logical basis of OWL on the meta level. More specifically, as the connection between a statement in PR-OWL and a statement in OWL is not formalized, it is unclear how to perform the integration of ontologies that contain statements of both formalisms.

This paper justifies the need for a formal mapping between random variables defined in PR-OWL and concepts defined in OWL, and proposes an approach to such a mapping. We first present a solution that is sufficient for binary relations. Next, we present a more robust solution that allows the user to define PR-OWL random variables with arbitrarily many arguments, while maintaining a 2-way mapping to OWL concepts. Finally, we present a schematic for the mapping back and forth from triples into random variables.

2 Why map PR-OWL Random Variables to OWL Concepts?

PR-OWL was proposed as an extension to the OWL language based on MEBN, which can express a probability distribution on interpretations of any first-order theory. In PR-OWL, a probabilistic ontology (PO) has to have at least one individual of class `MTheory`, which is basically a label linking a group of MFrags that collectively form a valid MTheory. In actual PR-OWL syntax, that link is expressed via the object property `hasMFrag` (which is the inverse of object property `isMFragIn`). Individuals of class `MFrag` are comprised of nodes. Each individual of class `Node` is a random variable (RV) and thus has a mutually exclusive, collectively exhaustive set of possible states. In PR-OWL, the object property `hasPossibleValues` links each node with its possible states, which are individuals of class `Entity`. Finally, random variables (represented by the class `Node` in PR-OWL) have unconditional or conditional probability distributions, which are represented by class `ProbabilityDistribution` and linked to their respective nodes via the object property `hasProbDist`.

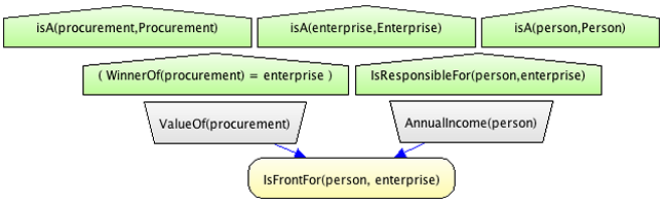

Fig. 1. Front of an Enterprise MFrag.

As a running example, we consider an OWL ontology for the public procurement domain. The ontology defines concepts such as procurement, winner of a procurement, members of a committee responsible for a procurement, etc.

Now, imagine we want to define some uncertain relations about this domain. For example, if an enterprise wins a procurement for millions of dollars, but the responsible person for this enterprise makes less than 10 thousand dollars a year, the responsible person may be a front. That is, we can identify potential fronts by examining the value of the procurement and the income of the responsible person. Figure 1 shows this probabilistic relation defined using PR-OWL in an open-source tool for probabilistic reasoning, UnBBayes [1]. In the figure, we see that a person's income and the value of a procurement influence whether the person is front for the procurement. The green pentagons at the top of the figure show conditions that must be met for the probabilistic relationship to apply; e.g., that the person we are considering as a possible front must be responsible for the enterprise we are examining.

Listing 1.1. Definition of `WinnerOf` RV in PR-OWL 1

```
<owl:Thing rdf:about="#WinnerOf_RV">
  <rdf:type rdf:resource="#Domain_Res"/>
  <hasPossibleValues rdf:resource="#Enterprise"/>
  <isResidentNodeIn rdf:resource="#ProcurementInfo_MFrag"/>
  <hasArgument rdf:resource="#WinnerOf_1"/>
</owl:Thing>

<owl:Thing rdf:about="#WinnerOf_1">
  <rdf:type rdf:resource="#SimpleArgRelationship"/>
  <hasArgNumber rdf:datatype="&xsd;int">1</hasArgNumber>
  <hasArgTerm rdf:resource=
    "#ProcurementInfo_MFrag.procurement"/>
  <isArgumentOf rdf:resource="#WinnerOf_RV"/>
</owl:Thing>

<owl:Thing rdf:about="#ProcurementInfo_MFrag.procurement">
  <rdf:type rdf:resource="#OVariable"/>
  <isOVariableIn rdf:resource="#ProcurementInfo_MFrag"/>
  <isSubsBy rdf:resource="#Procurement"/>
  <isArgTermIn rdf:resource="#WinnerOf_1"/>
</owl:Thing>
```

We would like to be able to tie this fragment of probabilistic knowledge with domain knowledge already represented in an OWL ontology. That is, we might have a database containing instances of persons and enterprises, linked to an OWL ontology defining their semantics (e.g., that persons can be responsible for enterprises). Accessing this information should be trivial once the definitions in the ontology were made available and permission was granted to retrieve data from the database. However, for PR-OWL to make use of this knowledge, there must be a way to link PR-OWL random variables (RVs) with concepts defined

in OWL. The current version of PR-OWL has no standard way to establish such links.

Listing 1.1 presents how the RV `WinnerOf_RV` from Figure 1 is defined in PR-OWL today. This RV is defined as follows:

- It is a domain resident node (line 2)
- Its possible values (range) are instances of `Enterprise` (line 3)
- Its home MFrag is `ProcurementInfo_MFrag` (line 4)
- It has one argument (domain) `WinnerOf_1` (line 5)
- `WinnerOf_1` is the first argument (line 10)
- `WinnerOf_1` is related to the variable `ProcurementInfo_MFrag.procurement` (lines 11-12)
- `ProcurementInfo_MFrag.procurement` is an ordinary variable (line 17)
- `ProcurementInfo_MFrag.procurement` is defined in the `ProcurementInfo_MFrag` (line 18)
- `ProcurementInfo_MFrag.procurement` can only be replaced by instances of `Procurement` (line 19)

Listing 1.2 is a suggested definition of the object property winnerOf in OWL. This property is defined as follows:

- It is an object property (line 1)
- It is a functional property (line 2)
- Its domain is the instances of `Procurement` (line 3)
- Its range is the instances of `Enterprise` (line 4)

Listing 1.2. Definition of `winnerOf` object property in OWL

```
1 <owl:ObjectProperty rdf:about="#winnerOf">
2     <rdf:type rdf:resource="&owl;FunctionalProperty"/>
3     <rdfs:domain rdf:resource="#Procurement"/>
4     <rdfs:range rdf:resource="#Enterprise"/>
5 </owl:ObjectProperty>
```

Comparing the two definitions `winnerOf` and `WinnerOf_RV`, we can see that they are consistent, since their domain/arguments and range/possible values are the same, `Procurement` and `Enterprise`, respectively. However, there is no property that explicitly relates these two concepts, and there is no implicit way of figuring out that they should be related besides the fact that their names are similar (`winnerOf` and `WinnerOf_RV`). Therefore, we would not have access to the semantics of the term `winnerOf` defined in our ontology when defining its probabilistic relations using the new and unrelated term `WinnerOf_RV` defined in our probabilistic ontology.

This simple example demonstrates the need to define a reference from every probabilistic definition involving a concept to its OWL definition. In other words, full compatibility with OWL requires modifications to PR-OWL that guarantee the preservation of OWL's semantics.

A simple solution to this mapping problem is presented in Listing 1.3. By adding the property `defineUncertaintyOf` which states that a random variable defines the uncertainty relations of a specific property, we could state that `WinnerOf_RV` defines the uncertainty of the object property `winnerOf` (line 3). In order to make this definition consistent we would need to add some axioms to our language stating that the possible values of the RV must be the same as the range defined in the property for which this RV defines the uncertainty. Similar axioms would be needed for its domain.

Listing 1.3. Definition of `WinnerOf` RV with mapping information to its OWL concept

```
1  <owl:Thing rdf:about="#WinnerOf_RV">
2    <rdf:type rdf:resource="#Domain_Res"/>
3    <defineUncertaintyOf rdf:resource="#winnerOf"/>
4    <hasPossibleValues rdf:resource="#Enterprise"/>
5    <isResidentNodeIn rdf:resource="#ProcurementInfo_MFrag"/>
6    <hasArgument rdf:resource="#WinnerOf_1"/>
7  </owl:Thing>
8
9  <owl:Thing rdf:about="#WinnerOf_1">
10   <rdf:type rdf:resource="#SimpleArgRelationship"/>
11   <hasArgNumber rdf:datatype="&xsd;int">1</hasArgNumber>
12   <hasArgTerm rdf:resource=
13     "#ProcurementInfo_MFrag.procurement"/>
14   <isArgumentOf rdf:resource="#WinnerOf_RV"/>
15 </owl:Thing>
16
17 <owl:Thing rdf:about="#ProcurementInfo_MFrag.procurement">
18   <rdf:type rdf:resource="#OVariable"/>
19   <isOVariableIn rdf:resource="#ProcurementInfo_MFrag"/>
20   <isSubsBy rdf:resource="#Procurement"/>
21   <isArgTermIn rdf:resource="#WinnerOf_1"/>
22 </owl:Thing>
```

3 Mapping n-ary relations

In Section 2 we presented a simple solution to map OWL concepts to random variables defined in PR-OWL. In this section we will show that the presented solution is not enough to cover the full expressiveness of PR-OWL. In particular, this solution cannot represent uncertainty for n-ary functions and relations.

Imagine extending our example to a situation in which a group of enterprises can win a procurement. Moreover, there will be a price associated with each enterprise on the contract. Therefore, instead of comparing the value of the procurement as a whole to try to identify the owner of the enterprise as a front (as shown on Figure 1), we need to consider only the part of that total associated to that specific enterprise, as shown in Figure 2.

Note that we now have a ternary relation which associates an enterprise, a contract, and the amount awarded by the contract to the enterprise. As a

functional relation, this is represented by the two-argument function `priceOf(-contract,enterprise)`.

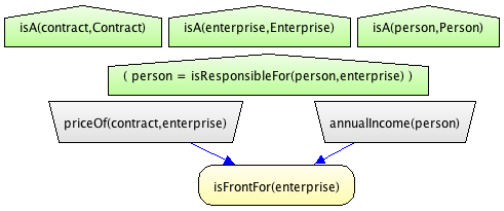

Fig. 2. Front of an Enterprise MFrag using `priceOf(contract,enterprise)`.

Listing 1.4. Problem when trying to define n-ary relations as simple binary relations

```
<owl:ObjectProperty rdf:about="#hasPrice">
    <rdfs:domain rdf:resource="#Contract"/>
    <rdfs:range rdf:resource="#Money"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#hasEnterprise">
    <rdfs:domain rdf:resource="#Contract"/>
    <rdfs:range rdf:resource="#Enterprise"/>
</owl:ObjectProperty>

<Contract rdf:about="#contract1">
    <rdf:type rdf:resource="&owl;Thing"/>
    <hasEnterprise rdf:resource="#enterprise1"/>
    <hasEnterprise rdf:resource="#enterprise2"/>
    <hasPrice rdf:resource="#price1"/>
    <hasPrice rdf:resource="#price2"/>
</Contract>

<Money rdf:about="#price1">
    <rdf:type rdf:resource="&owl;Thing"/>
    <valueOf rdf:datatype="&xsd;float">10000</valueOf>
    <currencyOf rdf:resource="#Dollar"/>
</Money>

<owl:Thing rdf:about="#price2">
    <rdf:type rdf:resource="#Money"/>
    <valueOf rdf:datatype="&xsd;float">500000</valueOf>
    <currencyOf rdf:resource="#Dollar"/>
</owl:Thing>
```

Suppose that we want to represent that `enterprise1` was hired for $10,000.00 and `enterprise2` for $500,000.00 both in `contract1`. The problem is that OWL

supports only binary relations. As shown in Listing 1.4, if we tried to represent this situation using binary relations with the class `Contract`, we would be unable to distinguish whether `enterprise1` has price of $10,000.00, `price1`, or $500,000.00, `price2`.

Listing 1.5. Defining n-ary relations in OWL

```
<owl:ObjectProperty rdf:about="#contractOf">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource="_:id1"/>
    <rdfs:range rdf:resource="#Contract"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#enterpriseOf">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource="_:id1"/>
    <rdfs:range rdf:resource="#Enterprise"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#priceOf">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource="_:id1"/>
    <rdfs:range rdf:resource="#Money"/>
</owl:ObjectProperty>

<owl:Thing rdf:about="#3aryInstance1">
    <rdf:type rdf:resource="_:id1"/>
    <contractOf rdf:resource="#contract1"/>
    <enterpriseOf rdf:resource="#enterprise1"/>
    <priceOf rdf:resource="#price1"/>
</owl:Thing>

<owl:Thing rdf:about="#3aryInstance2">
    <rdf:type rdf:resource="_:id1"/>
    <contractOf rdf:resource="#contract1"/>
    <enterpriseOf rdf:resource="#enterprise2"/>
    <priceOf rdf:resource="#price2"/>
</owl:Thing>
```

As shown in Figure 3, one way to overcome this problem is to create a blank node which has three functions mapping to each of the 3 arguments of our ternary relation. Notice that these 3 binary relations (`contractOf`, `enterpriseOf`, and `priceOf`) have to be functions, otherwise we would have the same problem we had in Listing 1.4. Listing 1.5 presents this representation in OWL (for more details on how to define n-ary relations in OWL, see [5]).

When we try to apply the simple solution given in Section 2, we realize that it is not suitable for RVs with more than one argument. This is due to the fact that we assume the following:

1. The range from the property associated to the `defineUncertaintyOf` has to be the same type as the value of the RV's `hasPossibleValues` property; and
2. The domain from the property associated to the `defineUncertaintyOf` has to be the same type as the only RV's argument (`hasArgument` → `hasArgTerm` → `isSubsBy`).

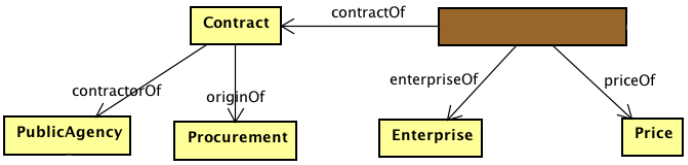

Fig. 3. An initial ontology with an n-ary relation between `Price`, `Enterprise`, and `Contract` using a blank node.

So, what happens with the other arguments of the RV? What do they map to? Notice also that there is no argument in `priceOf(contract,enterprise)` that relates to the domain of the OWL property `priceOf`. In other words, there is no argument that "points" to the blank node we defined in Figure 3 and Listing 1.5. Besides, having only the property `defineUncertaintyOf` relating to the OWL property `priceOf` tells us nothing about what `contract` and `enterprise` are and where they come from. As a matter of fact, we need to have a reference to all the binary properties that we use to represent the n-ary relation we want. Therefore, in this case, we also need to have a mapping to both `contractOf` and `enterpriseOf`.

Taking a closer look, we realize that all three properties of interest (`priceOf`, `contractOf`, and `enterpriseOf`) have the same domain (the blank node) and their range, `Money`, `Contract`, and `Enterprise`, map directly to the possible values of our RV of interest, to the argument `contract`, and to the argument `enterprise`, respectively. Listing 1.6 shows a more complex and robust solution that covers this case and any other n-ary relation for which we might want to define uncertainty.

Listing 1.6 states that the RV `priceOf_RV` defines the probabilistic semantics of the property `priceOf`, which already has an OWL semantics (line 3). Lines 4 and 5 ensure that the domain and range from the OWL property match the RV domain (`hasDomain`) and range (`hasPossibleValues`), respectively. Lines 7 and 8 say that this RV has two arguments. Lines 13-15 define the first argument as being the variable `contract`, and lines 30-32 define the second argument as the variable `enterprise`. Lines 22-24 specify that the contract variable is used as the object (`objectIn`) of the OWL property `contracOf`, thus it can only be substituted by (`isSubsBy`) the class that is the range of the `contractOf` property, which is `Contract`. In addition, the domain also has to be the same (`hasDomain`), which, in this case, is the blank node `_:id1`.

Listing 1.6. Robust solution for defining n-ary RVs and mapping them to the OWL concepts that define their semantics

```
1  <owl:Thing rdf:about="#priceOf_RV">
2    <rdf:type rdf:resource="#Domain_Res"/>
3    <defineUncertaintyOf rdf:resource="#priceOf"/>
4    <hasDomain rdf:resource="_:id1"/>
5    <hasPossibleValues rdf:resource="#Money"/>
6    <isResidentNodeIn rdf:resource="#FrontOfEnterprise_MFrag"/>
7    <hasArgument rdf:resource="#priceOf_1"/>
8    <hasArgument rdf:resource="#priceOf_2"/>
9  </owl:Thing>
10
11 <owl:Thing rdf:about="#priceOf_1">
12   <rdf:type rdf:resource="#SimpleArgRelationship"/>
13   <hasArgNumber rdf:datatype="&xsd;int">1</hasArgNumber>
14   <hasArgTerm rdf:resource=
15     "#FrontOfEnterprise_MFrag.contract"/>
16   <isArgumentOf rdf:resource="#priceOf_RV"/>
17 </owl:Thing>
18
19 <owl:Thing rdf:about="#ProcurementInfo_MFrag.contract">
20   <rdf:type rdf:resource="#OVariable"/>
21   <isOVariableIn rdf:resource="#FrontOfEnterprise_MFrag"/>
22   <objectIn rdf:resource="#contractOf"/>
23   <hasDomain rdf:resource="_:id1"/>
24   <isSubsBy rdf:resource="#Contract"/>
25   <isArgTermIn rdf:resource="#priceOf_1"/>
26 </owl:Thing>
27
28 <owl:Thing rdf:about="#priceOf_2">
29   <rdf:type rdf:resource="#SimpleArgRelationship"/>
30   <hasArgNumber rdf:datatype="&xsd;int">2</hasArgNumber>
31   <hasArgTerm rdf:resource=
32     "#FrontOfEnterprise_MFrag.enterprise"/>
33   <isArgumentOf rdf:resource="#priceOf_RV"/>
34 </owl:Thing>
35
36 <owl:Thing rdf:about="#ProcurementInfo_MFrag.enterprise">
37   <rdf:type rdf:resource="#OVariable"/>
38   <isOVariableIn rdf:resource="#FrontOfEnterprise_MFrag"/>
39   <objectIn rdf:resource="#enterpriseOf"/>
40   <hasDomain rdf:resource="_:id1"/>
41   <isSubsBy rdf:resource="#Enterprise"/>
42   <isArgTermIn rdf:resource="#priceOf_2"/>
43 </owl:Thing>
```

The same thing goes for the `enterprise` variable. In lines 39-41 we define that the `enterprise` variable is in fact used as the object (`objectIn`) of the OWL property `enterpriseOf`, thus it can only be substituted by (`isSubsBy`)

the class that is the range of the `enterpriseOf` property, which is `Enterprise`. In addition, the domain also has to be the same (`hasDomain`), which, in this case, is the blank node `_:id1`.

4 The bridge joining OWL and PR-OWL

The key to building the bridge that connects the deterministic ontology defined in OWL and its probabilistic extension defined in PR-OWL is to understand how to translate one to the other. On the one hand, given a concept defined in OWL, how should its uncertainty be defined in PR-OWL in a way that maintains its semantics defined in OWL? On the other hand, given a random variable defined in PR-OWL, how should it be represented in OWL in a way that respects its uncertainty already defined in PR-OWL? Examples of our proposed translation were given above. Here, a schematic is given in Figure 4 for the 2-way mapping between triples and random variables. Functions and predicates are considered as separate cases.

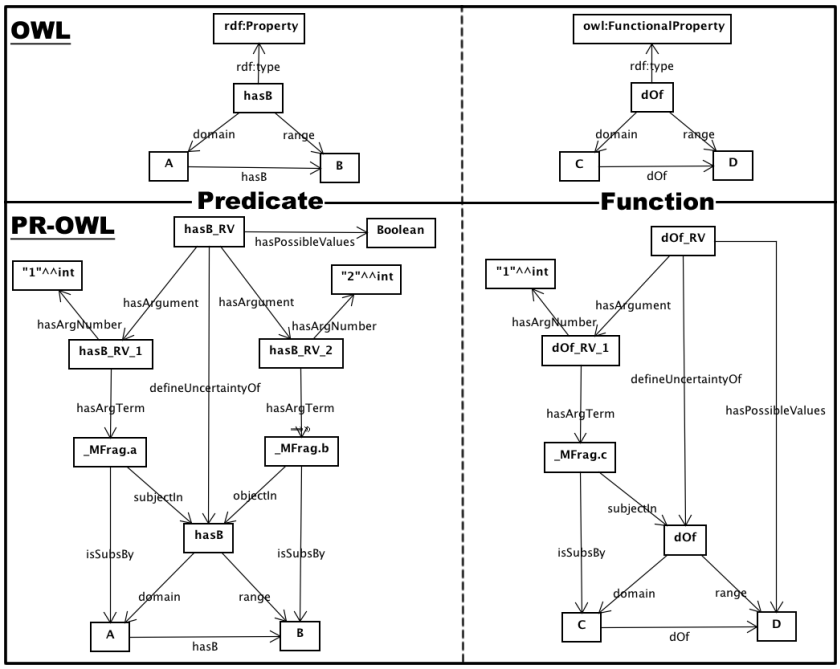

Fig. 4. The bridge joining OWL and PR-OWL.

If a property (`hasB` or `dOf`) is defined in OWL, then its domain and range are already represented (`A` and `B`; `C` and `D`, respectively). The first thing to be done is

to create the corresponding RV in PR-OWL (`hasB_RV` and `dOf_RV`, respectively) and link it to this OWL property through the property `defineUncertaintyOf`.

For binary relations, the domain of the property (`A` and `C`, respectively) will usually be the type (`isSubsBy`) of the variable (`_MFrag.a` and `_MFrag.c`, respectively) used in the first argument (`hasB_RV_1` and `dOf_RV_1`, respectively) of the RV. For n-ary relations see Section 3.

If the property is non-functional (`hasB`), then it represents a predicate that may be true or false. Thus, instead of having the possible values of the RV in PR-OWL (`hasB_RV`) being the range of the OWL property (`B`), it must be `Boolean`. So, its range (`B`) has to be mapped to the second argument (`hasB_RV_2`) of the RV, the same way the domain (`A`) was mapped to the first argument (`hasB_RV_1`) of the RV. On the other hand, if the the property is functional (`dOf`), the possible values of its RV (`dOf_RV`) must be the same as its range (`B`).

It is important to note that not only is the RV linked to the OWL property by the `defineUncertaintyOf`, but also to the variables by either `subjectIn` or `objectIn`, depending on what they refer to (domain or range of the OWL property, respectively). This feature is especially important when dealing with n-ary relations, where each variable will be associated with a different OWL property (see Section 3) for details).

Finally, if the RV is already defined in PR-OWL with all its arguments and its possible values, the only thing that needs to be done is to create the corresponding OWL property, link the RV to it using the `defineUncertaintyOf` and make sure that the domain and range of the property matches the RV definition, as explained previously.

The mapping described in this Section provides the basis for a formal definition of consistency between a PR-OWL probabilistic ontology and an OWL ontology, in which rules in the OWL ontology correspond to probability one assertions in the PR-OWL ontology. A formal notion of consistency can lead to development of consistency checking algorithms.

5 Conclusion

Although the semantics was not formally defined, this paper provided both the syntax and a more in depth description of one of the major changes in PR-OWL 2: a formal mapping between OWL concepts and PR-OWL random variables. First, the importance of a formal mapping was justified through an example. Second, a simple solution sufficient for 2-way relations was presented. Next, a more complex and robust solution covering n-ary random variables was presented. Finally, a schematic was given for how to do the mapping back and forth between PR-OWL random variables and OWL triples (both predicates and functions).

As future work, this schematic will be formally defined by explicitly defining its semantics. This will be a major contribution of PR-OWL 2. Moreover, a formalization of an algorithm for performing the mapping from OWL concepts to PR-OWL RVs, and vice-versa, will be proposed. In addition, PR-OWL 2 will address other issues described in [2].

Acknowledgments. The authors would like to thank the Brazilian Office of the Comptroller General (CGU) for their active support since 2008 and for providing the human resources necessary to conduct this research.

References

1. UnBBayes - the UnBBayes site. http://unbbayes.sourceforge.net/.
2. Rommel Novaes Carvalho, Kathryn B. Laskey, and Paulo Cesar G. Costa. Compatibility formalization between PR-OWL and OWL. In *Proceedings of the First International Workshop on Uncertainty in Description Logics (UniDL) on Federated Logic Conference (FLoC) 2010*, Edinburgh, UK, July 2010.
3. Paulo C. G Costa. *Bayesian Semantics for the Semantic Web*. PhD, George Mason University, July 2005. Brazilian Air Force.
4. Zhongli Ding, Yun Peng, and Rong Pan. BayesOWL: uncertainty modeling in semantic web ontologies. In *Soft Computing in Ontologies and Semantic Web*, pages 3–29. 2006.
5. Patrick Hayes and Alan Rector. Defining n-ary relations on the semantic web. http://www.w3.org/TR/swbp-n-aryRelations/, 2006.
6. Jochen Heinsohn. Probabilistic description logics. In *Proceedings of the 10th Annual Conference on Uncertainty in Artificial Intelligence (UAI-94)*, pages 311–318, Seattle, Washington, USA, 1994. Morgan Kaufmann.
7. Daphne Koller, Alon Levy, and Avi Pfeffer. P-CLASSIC: a tractable probabilistic description logic. *IN PROCEEDINGS OF AAAI-97*, pages 390—397, 1997.
8. Kathryn Blackmond Laskey. MEBN: a language for first-order bayesian knowledge bases. *Artif. Intell.*, 172(2-3):140–178, 2008.
9. Kenneth Laskey and Kathryn Laskey. Uncertainty reasoning for the world wide web: Report on the URW3-XG incubator group. URW3-XG, W3C, 2008.
10. Thomas Lukasiewicz. Expressive probabilistic description logics. *Artificial Intelligence*, 172(6-7):852–883, April 2008.
11. J Z Pan, G Stoilos, G Stamou, V Tzouvaras, and I Horrocks. f-SWRL: a fuzzy extension of SWRL. *Journal of Data Semantics VI*, 4090/2006:28–46, 2006.
12. Livia Predoiu and Heiner Stuckenschmidt. Probabilistic extensions of semantic web languages - a survey. In *The Semantic Web for Knowledge and Data Management: Technologies and Practices*. Idea Group Inc, 2008.
13. Umberto Straccia. A fuzzy description logic for the semantic web. In *FUZZY LOGIC AND THE SEMANTIC WEB, CAPTURING INTELLIGENCE*, pages 167—181. Elsevier, 2005.
14. Jia Tao, Zhao Wen, Wang Hanpin, and Wang Lifu. PrDLs: a new kind of probabilistic description logics about belief. In *New Trends in Applied Artificial Intelligence*, pages 644–654. 2007.

Learning Sentences and Assessments in Probabilistic Description Logics

José Eduardo Ochoa Luna[1], Kate Revoredo[2], and Fabio Gagliardi Cozman[1]

[1] Escola Politécnica, Universidade de São Paulo,
Av. Prof. Mello Morais 2231, São Paulo - SP, Brazil
[2] Departamento de Informática Aplicada, Unirio
Av. Pasteur, 458, Rio de Janeiro, RJ, Brazil
eduardo.ol@gmail.com,katerevoredo@uniriotec.br,fgcozman@usp.br

Abstract. The representation of uncertainty in the semantic web can be eased by the use of learning techniques. To completely induce a probabilistic ontology (that is, an ontology encoded through a probabilistic description logic) from data, two basic tasks must be solved: (1) learning concept definitions and (2) learning probabilistic inclusions. In this paper we propose and test an algorithm that learns concept definitions using an inductive logic programming approach and then learns probabilistic inclusions using relational data.

1 Introduction

Probabilistic Description Logics (PDLs) have been extensively investigated in the last few years [5, 8, 19, 7]. The goal is to represent uncertainty in the context of classical description logics. So far probabilistic description logics have been mostly restricted to academic purposes, as caveats in syntax and semantics have prevented them from spreading into several domains. Additionaly, it can be hard to elicit the probability component of a particular set of sentences.

The probabilistic description logic CR$\mathcal{ALC}$ [6, 22, 7] allows one to perform probabilistic reasoning by adding uncertainty capabilities to the logic $\mathcal{ALC}$ [2]. Previous efforts for learning CR$\mathcal{ALC}$ have separately focused on concept definitions [20] and probabilistic inclusions [24]. In this paper, we present an algorithm for learning concept definitions and probabilistic inclusions at once; i.e., we discuss how to construct the whole probabilistic terminology based on CR$\mathcal{ALC}$ from relational data. We expect that learning techniques can accomodate together background knowledge and deterministic and probabilistic concepts, giving each component its due relevance.

The algorithm we propose is mostly based on inductive logic programming (ILP) [9] techniques with a probabilistic twist. A search for the best concept description is performed. At the end of this search a decision is made as to whether to consider the concept description found or to insert a probabilistic inclusion based on this concept.

The paper is organized as follows. Section 2 reviews basic concepts of description logics, probabilistic description logics, CR$\mathcal{ALC}$ and machine learning

in a deterministic setting. Section 3 presents our algorithm for probabilistic description logic learning. Experiments are discussed in Section 4, and Section 5 concludes the paper.

2 Basics

The aim of this paper is to learn probabilistic terminologies from data. In this section we briefly review both deterministic and probabilistic components of probabilistic description logics. In addition, machine learning in a deterministic setting is discussed.

2.1 Description Logics

Description logics (DLs) form a family of representation languages that are typically decidable fragments of first order logic (FOL) [2]. Knowledge is expressed in terms of *individuals*, *concepts*, and *roles*. The semantic of a description is given by a *domain* $\mathcal{D}$ (a set) and an *interpretation* $\cdot^{\mathcal{I}}$ (a functor). Individuals represent objects through names from a set $N_{\mathsf{I}} = \{a, b, \ldots\}$. Each *concept* in the set $N_{\mathsf{C}} = \{C, D, \ldots\}$ is interpreted as a subset of a domain $\mathcal{D}$. Each *role* in the set $N_{\mathsf{R}} = \{r, s, \ldots\}$ is interpreted as a binary relation on the domain.

Concepts and roles are combined to form new concepts using a set of *constructors*. Constructors in the $\mathcal{ALC}$ logic are *conjunction* ($C \sqcap D$), *disjunction* ($C \sqcup D$), *negation* ($\neg C$), *existential* restriction ($\exists r.C$), and *value* restriction ($\forall r.C$). *Concept inclusions/definitions* are denoted respectively by $C \sqsubseteq D$ and $C \equiv D$, where C and D are concepts. Concepts ($C \sqcup \neg C$) and ($C \sqcap \neg C$) are denoted by $\top$ and $\bot$ respectivelly. Information is stored in a *knowledge base* ($\mathcal{K}$) divided in two parts: the TBox (terminology) and the ABox (assertions). The TBox lists concepts and roles and their relationships. A TBox is acyclic if it is a set of concept inclusions/definitions such that no concept in the terminology uses itself. The ABox contains assertions about objects.

Given a knowledge base $\mathcal{K} =< \mathcal{T}, \mathcal{A} >$, the reasoning services typically include (i) consistency problem (to check whether the $\mathcal{A}$ is consistent with respect to the $\mathcal{T}$); (ii) entailment problem (to check whether an assertion is entailed by $\mathcal{K}$; note that this generates class-membership assertions $\mathcal{K} \models C(a)$, where a is an individual and C is a concept); (iii) concept satisfiability problem (to check whether a concept is subsumed by another concept with respect to the $\mathcal{T}$). The latter two reasoning services can be reduced to the consistency problem [2].

2.2 Probabilistic Description Logics and CR$\mathcal{ALC}$

Several probabilistic descriptions logics (PDLs) have appeared in the literature. Heinsohn [12], Jaeger [14] and Sebastiani [25] consider probabilistic inclusion axioms such as $P_{\mathcal{D}}(\mathsf{Professor}) = \alpha$, meaning that a randomly selected object is a $\mathsf{Professor}$ with probability α. This characterizes a *domain-based* semantics: probabilities are assigned to subsets of the domain $\mathcal{D}$. Sebastiani also allows inclusions

such as $P(\mathsf{Professor}(\mathsf{John})) = \alpha$, specifying probabilities over the interpretations themselves. For example, one interprets $P(\mathsf{Professor}(\mathsf{John})) = 0.001$ as assigning 0.001 to be the probability of the set of interpretations where John is a Professor. This characterizes an *interpretation-based* semantics.

The PDL CR$\mathcal{ALC}$ is a probabilistic extension of the DL $\mathcal{ALC}$ that adopts an interpretation-based semantics. It keeps all constructors of $\mathcal{ALC}$, but only allows concept names on the left hand side of inclusions/definitions. Additionally, in CR$\mathcal{ALC}$ one can have probabilistic inclusions such as $P(C|D) = \alpha$ or $P(r) = \beta$ for concepts C and D, and for role r. If the interpretation of D is the whole domain, then we simply write $P(C) = \alpha$. The semantics of these inclusions is roughly (a formal definition can be found in [7]) given by:

$$\forall x \in \mathcal{D} \ : \ P(C(x)|D(x)) = \alpha,$$

$$\forall x \in \mathcal{D}, y \in \mathcal{D} \ : \ P(r(x,y)) = \beta.$$

We assume that every terminology is acyclic; no concept uses itself. This assumption allows one to represent any terminology $\mathcal{T}$ through a directed acyclic graph. Such a graph, denoted by $\mathcal{G}(\mathcal{T})$, has each concept name and role name as a node, and if a concept C directly uses concept D, that is if C and D appear respectively in the left and right hand sides of an inclusion/definition, then D is a *parent* of C in $\mathcal{G}(\mathcal{T})$. Each existential restriction $\exists r.C$ and value restriction $\forall r.C$ is added to the graph $\mathcal{G}(\mathcal{T})$ as nodes, with an edge from r to each restriction directly using it. Each restriction node is a *deterministic* node in that its value is completely determined by its parents.

The semantics of CR$\mathcal{ALC}$ is based on probability measures over the space of interpretations, for a fixed domain. Inferences, such as $P(\mathsf{A}_o(\mathsf{a}_0)|\mathcal{A})$ for an ABox $\mathcal{A}$, can be computed by propositionalization and probabilistic inference (for exact calculations) or by a first order loopy propagation algorithm (for approximate calculations) [7].

2.3 Learning Description Logics

The use of ontologies for knowledge representation has been a key element of proposals for the Semantic Web [1]. However, constructing ontologies from scratch can be a bundersome and time consuming task [10]. Nowadays, mainly due to the availability of data, learning of ontologies has turn out to be an interesting alternative. Indeed, considerable effort is currently invested into developing automated means for the acquisition of ontologies [16].

Most early approaches were only capable of learning simple ontologies such as taxonomic hierarchies. Some recent approaches such as YINYANG [13], DL-FOIL [10] and DL-Learner [18] have focused on learning expressive terminologies (we refer to [20] for a detailed review on learning description logics). To some extent, all these approaches have been inspired by Inductive Logic Programming (ILP) techniques, in that they try to transfer ILP methods to description logic settings. The goal of learning in such deterministic languages is generally to find a correct concept with respect to given examples. A formal definition is:

Definition 1. *Given a knowledge base $\mathcal{K}$, a target concept* Target *such that* Target $\notin \mathcal{K}$*, a set $E = E_p \cup E_n$ of positive and negative examples given as assertions for* Target*, the goal of learning is to find a concept definition C(*Target $\equiv C$*) such that $\mathcal{K} \cup C \models E_p$ and $\mathcal{K} \cup C \not\models E_n$.*

A sound concept definition for Target must cover all positive examples and none of the negative examples. A learning algorithm can be constructed as a combination of (1) a refinement operator, which defines how a search tree can be built, (2) a search algorithm, which controls how the tree is traversed, and (3) a scoring function to evaluate the nodes in the tree defining the best one.

The refinement operator Refinement operators allow us to find candidate concept definitions through two basic tasks: generalization and specialization [17]. Such operators in both ILP and description logic learning rely on θ-subsumption to establish an ordering so as to traverse the search space. If a concept C subsumes a concept $D(D \sqsubseteq C)$, then C covers all examples which are covered by D, which makes subsumption a suitable order. Arguably the best refinement operator for description logic learning is the one available in the DL-Learner system [17, 18], as this operator has been proved to be complete, weakly complete and proper (see [17] for details).

The score function In a deterministic setting a cover relationship simply tests whether, for given candidate concept definition (C), a given example e holds; that is, $\mathcal{K} \cup C \models e$ where $e \in E_p$ or $e \in E_n$. In this sense, a cover relationship $cover(e, \mathcal{K}, C)$ indicates whether a candidate concept covers a given example. A cover relationship is commonly evaluated by instance checking [10].

In description logic learning one often compares candidates through score functions based on the number of positive/negative examples covered. To avoid overfitting on concepts, horizontal expansions[3] are also explored [18]. For instance, in DL-Learner a fitness relationship considers the number of positive examples as well as the length of solutions when expanding candidates in the tree search.

The algorithm to traverse the search space The learning algorithm depends basically on the way we traverse the candidate concepts obtained after applying refinement operators. In a deterministic setting the search for candidate concepts is often based on the FOIL [23] algorithm. There are also different approaches (for instance, DL-Learner, an approach based on genetic algorithms [16], and one that relies on horizontal expansion and redundance checking to traverse search trees [18]).

[3] Given a node in a search tree, the horizontal expansion is its upper bound on the length of child concepts.

3 Learning the PDL CR$\mathcal{ALC}$

A probabilistic terminology consists of both concepts definitions and probabilistic components (probabilistic inclusions in CR$\mathcal{ALC}$). We aim at automatically identifying from data sound deterministic concepts and consistent probabilistic inclusions. A key design choice in learning under a combined approach is to give a due relevance to each component.

It is worth noting that there are well established deterministic concepts such as $\mathsf{Father} \equiv \mathsf{Male} \sqcap \mathsf{hasChild}.\top$ for which it would be unnecessary to find a probabilistic interpretation. On the other hand, there are concepts with natural probabilistic assessments such as $P(\mathsf{FlyingBird}|\mathsf{Bird}) = \alpha$. In principle, a learning algorithm should be able to deal with these subtleties.

We argue that negative and positive examples underlie the choice of either a concept definition or a probabilistic inclusion. In a deterministic setting we expect to find concepts covering all positive examples, which is not always possible. It is common to allow flexible heuristics that deal with these issues. Moreover, there are several examples that cannot be ascribed to candidate hypotheses[4]. Uncertainty stems from such missing information. Therefore, when we are unable to find a concept definition that covers all positive examples we assume such hypothesis as candidates to be a probabilistic inclusion and we begin the search for the best probabilistic inclusion that fits the examples.

As in description logic learning three tasks are important and should be considered: (1) refinement operators, (2) scoring functions and (3) a traverse search space algorithm. The refinement operator described in 2.3 is used for learning the deterministic component of probabilistic terminologies. The other two tasks were adapted for probabilistic description logic learning as follows.

3.1 The Probabilistic Score Function

In our proposal, since we want to learn probabilistic terminologies, we adopt a probabilistic cover relation given in [15]:

$$cover(e, \mathcal{K}, C) = P(e|\mathcal{K}, C).$$

Every candidate hypothesis together with a given example turns out to be a probabilistic random variable which yields true if the example is covered, and false otherwise. To guarantee soundness of the ILP process (that is, to cover positive examples and not to cover negative examples), the following restrictions are needed:

$$P(e_p|\mathcal{K}, C) > 0, \qquad P(e_n|\mathcal{K}, C) = 0.$$

In this way a probabilistic cover relationship is a generalization of the deterministic cover, and is suitable for a combined approach. Probabilities can be

[4] In some cases the Open World Assumption inherent to description logics prevent us for stating membership of concepts.

computed through Bayes' theorem:

$$P(e|\mathcal{K}, C_1, \ldots, C_k) = \frac{P(C_1, C_2, \ldots, C_k|T)P(T)}{P(C_1, \ldots, C_k)},$$

where $C_1, \ldots, C_k$ are candidate concepts definitions, and T denotes the target concept variable. Here are three possibilities for modeling $P(C_1, \ldots, C_k|T)$: (1) a naive Bayes assumption may be adopted [15] (each candidate concept is independent given the target), and then $P(C_1, \ldots, C_k|T) = \prod_i P(C_i|T)$; (2) the noisy-OR function may be used [20]; (3) a less restrictive option based on tree augmented naive Bayes networks (TAN) may be handy [15]. This last possibility has been considered for the probabilistic cover relationship used in this paper. In each case probabilities are estimated by maximum (conditional) likelihood parameters. The candidate concept definition C_i with the highest probability $P(C_i|T)$ is the one chosen as the best candidate.

As we have chosen a probabilistic cover relationship, our probabilistic score is defined accordingly:

$$score(\mathcal{K}|C) = \prod_{e_i \in E_p} P(e_i|\mathcal{K}, C),$$

where C is the best candidate chosen as described before.

In the probabilistic score we have previously defined, a given threshold allow us differentiate between a deterministic and probabilistic inclusion candidate. Further details are given in the next section.

3.2 The Algorithm to Learn Probabilistic Terminologies

Previous efforts for learning the PDL CR$\mathcal{ALC}$ have separately explored concepts definitions [20] and probabilistic inclusions [24]. In this paper, we advocate for a combined approach where we use a classical approach for traversing the space of deterministic concepts and a probabilistic procedure for generating probabilistic inclusions.

The choice between a deterministic or a probabilistic inclusion is based on a probabilistic score. We start by searching a deterministic concept. If after a set of iterations the score of the best candidate is below a given threshold, a search for a probabilistic inclusion is preferred rather than keep searching for a deterministic concept definition. Then, the current best k-candidates are considered as start point for probabilistic inclusion search. The complete learning procedure is shown in Algorithm 1.

The algorithm starts with an overly general concept definition in the root of the search tree (line 1). This node is expanded according to refinement operators and horizontal expansion criterion (line 4), i.e, child nodes obtained by refinement operators are added to the search tree (line 5). The probabilistic parameters of these child nodes are learned (line 6) and then they are evaluated with the best one chosen for a new expansion (line 3) (alternative nodes based

Require: an initial knowledge base $\mathcal{K} =< \mathcal{T}, \mathcal{A} >$ and a training set E.
1: SearchTree with a node $\{C = \top, h = 0\}$
2: **repeat**
3: choose node $N = \{C, h\}$ with highest probabilistic score in SearchTree
4: expand node to length $h + 1$:
5: add all nodes $D \in$ (refinementOperator(C)) with lenght $=h + 1$
6: learn parameters for all nodes D
7: $N = \{C, h + 1\}$
8: expand alternative nodes according to horizontal expansion factor and $h + 1$[18]
9: **until** stopping criterion
10: $N' =$ best node in SearchTree
11: **if** score(N') $>$ threshold **then**
12: return deterministic concept $C' \in N'$
13: **else**
14: call ProbabilisticInclusion(SearchTree)
15: **end if**

Algorithm 1: Algorithm for learning probabilistic terminologies.

on horizontal expansion factor are also considered (line 8)). This process continues until a stopping criterion is attained (difference for scores is insignificant); After that, the best node obtained is evaluated and if it is above a threshold, a deterministic concept definition is found and returned (line 11). Otherwise, a probabilistic inclusion procedure is called (line 13).

The Algorithm 2 learns probabilistic inclusions. It starts retrieving the best k nodes in the search tree and computing the conditional mutual information for every pair of nodes (line 2). Then an undirected graph is built where the vertices are the k nodes and the edges are weighted with the value of the conditional mutual information [21] for each pair of vertices (lines 4 and 5). A maximum weight spanning tree [4] from this graph is built (line 6) and the target concept is added as a parent for each vertice (line 7). The probabilistic parameters are learned (line 8). This learned TAN-based classifier [11] is used to evaluate the possible probabilistic inclusion candidates (line 9) and the best one is returned.

4 Experiments

In order to evaluate the learning algorithm we have divided the analysis in two stages. In a first stage, the algorithm was compared with, arguably, the best description logic learning algorithm available (the DL-Learner system). The second stage evaluated suitability of the algorithm for learning probabilistic terminologies in real world domains.

The aim of the first stage was to investigate whether by introducing a probabilistic setting the algorithm behaves as well as traditional deterministic approaches in description logic learning tasks. In this preliminar evaluation (as a rule, there is a lack of evaluation standards in ontology learning [18]) we have

Require: SearchTree previously computed
1: **for** each pair of candidates C_i, C_j in first k nodes of the SearchTree **do**
2: compute the conditional mutual information $I(C_i, C_j|T)$
3: **end for**
4: build an undirected graph in which vertices are the k candidates
5: annotate the weight of an edge connecting C_i to C_j by the $I(C_i, C_j|T)$
6: build a maximum weight spanning tree from this graph
7: add T as parent for each C_i
8: learn probabilities for $P(C_i|Parents(C_i))$
9: return the highest probabilistic inclusion $P(T|C') = \alpha$

Algorithm 2: Algorithm for learning probabilistic inclusions.

considered five datasets available in the DL-Learner system and reported in [18]. Evaluation results are shown in Table 1.

Table 1. Description logic learning results

Problem	axioms, examples	DL-learner correct (length)	Combined approach correct(length)
trains	252,10	100(5)	100%(5)
arches	47,5	100%(9)	100%(10)
moral	31,43	100%(3)	100%(5)
poker(pair)	35,49	100%(8)	100%(8)
poker (straight)	45,55	100%(5)	100%(5)

The combined approach was able to learn correct concept definitions. However, in some cases produced longer solutions.

In the second stage we focused on learning of probabilistic terminologies from real world data. Wikipedia[5] was used to do so. Wikipedia articles consist mostly of free text, but also contain various types of structured information in the form of Wiki markup. Such information includes infobox templates, categorization information, images geo-coordinates, links to external Web pages, disambiguation pages, redirects between pages, and links across different language editions of Wikipedia.

In the last years, there were several projects aimed at structuring such huge source of knowledge. Examples include, The DBpedia project [3], which extracts structured information from Wikipedia and turns it into a rich knowledge base, and YAGO [26], a semantic knowledge base based on data from Wikipedia and WordNet[6]. Currently, YAGO knows more than 2 million entities (like persons, organizations, cities, etc.). It knows 20 million facts about these

[5] http://www.wikipedia.org/
[6] wordnet.princeton.edu/

entities. Unlike many other automatically assembled knowledge bases, YAGO has a manually confirmed accuracy of 95%. Several domains ranging from films, places, historical events, wines, etc. have been considered in this ontology. Moreover, facts are given as binary relationships that are suitable for our learning settings. There are approximately 92 relationships available. Examples include actedIn, bornIn, created, discovered describes, diedIn, happenedIn, hasAcademicAdvisor, hasChild, hasHDI, hasWonPrize, influences, isMarriedTo, isPartOf, livesIn, politicianOf, worksAt.

We have used subsets of YAGO facts for learning probabilistic terminologies. Two domais have been mostly explored. The first, related to scientists. The second, related to film directors. In both cases the threshold used was 0.85 and the 20 best candidates were considered in the probabilistic inclusion learning step.

The first dataset consists of 2008 potential scientists for which we have learned concept definitions and probabilistic inclusions. The complete terminology is given below:

$$
\begin{array}{ll}
 & P(\mathsf{Person}) = 0.9 \\
 & P(\mathsf{Topic}) = 0.4 \\
 & P(\mathsf{Year}) = 0.35 \\
 & P(\mathsf{Prize}) = 0.2 \\
 & P(\mathsf{Text}) = 0.25 \\
 & P(\mathsf{EducationalInstitution}) = 0.3 \\
 & P(\mathsf{wrotes}) = 0.4 \\
 & P(\mathsf{hasAcademicAdvisor}) = 0.80 \\
 & P(\mathsf{interestedIn}) = 0.6 \\
 & P(\mathsf{diedOnYear}) = 0.7 \\
 & P(\mathsf{hasWonPrize}) = 0.4 \\
 & P(\mathsf{worksAt}) = 0.85 \\
 & P(\mathsf{influences}) = 0.6 \\
\mathsf{Scientist} \equiv & \mathsf{Person} \\
 & \sqcap(\exists \mathsf{hasAcademicAdvisor.Person} \\
 & \sqcap \exists \mathsf{wrotes.Text} \sqcap \exists \mathsf{worksAt.EducationalInsitution}) \\
P(\mathsf{InfluentialScientist} & \mid \mathsf{Scientist} \sqcap \exists \mathsf{influences.} \\
 & \exists \mathsf{diedOnYear.Year}) = 0.85 \\
P(\mathsf{Musician} & \mid \mathsf{Person} \sqcap \exists \mathsf{hasAcademicAdvisor.} \exists \mathsf{wrote.Text}) = 0.1 \\
\mathsf{HonoredScientist} \equiv & \mathsf{Scientist} \\
 & \sqcap \exists \mathsf{hasWonPrize.Prize}
\end{array}
$$

This resulting CR$\mathcal{ALC}$ terminology can be further investigated by probabilistic inference[7]. The basic task we address is classification. Assume we are interested in classifying a potential scientist given we know he/she has written a book and has an academic advisor:

$$P(\mathsf{Scientist}(0) | \mathsf{Person}(0) \sqcap \exists \mathsf{wrote.Text}(1) \sqcap \mathsf{hasAcademicAdvisor.Person}(2)) = 0.5$$

When further evidence is available the value probability is updated to:

$$
\begin{array}{ll}
P(\mathsf{Scientist}(0) & | \mathsf{Person}(0) \\
 & \sqcap(\exists \mathsf{wrote.Text}(1) \sqcap \exists \mathsf{hasAcademicAdvisor.} \\
 & \exists \mathsf{influences.Person}(3))) = 0.65
\end{array}
$$

[7] Given a domain size, a relational Bayesian network is constructed to do so.

In the second dataset we have collected facts about film directors ranging from classical to contemporary. About 5589 potential directors have been considered. The complete probabilistic terminology is shown below.

$$
\begin{array}{ll}
 & P(\mathsf{Person}) = 0.9 \\
 & P(\mathsf{Prize}) = 0.1 \\
 & P(\mathsf{Year}) = 0.25 \\
 & P(\mathsf{Film}) = 0.3 \\
 & P(\mathsf{isMarriedTo}) = 0.1 \\
 & P(\mathsf{influences}) = 0.35 \\
 & P(\mathsf{hasWonPrize}) = 0.28 \\
 & P(\mathsf{hasChild}) = 0.05 \\
 & P(\mathsf{diedOnYear}) = 0.5 \\
 & P(\mathsf{directed}) = 0.8 \\
 & P(\mathsf{actedIn}) = 0.4 \\
\mathsf{Actor} \equiv & \mathsf{Person} \sqcap \forall \mathsf{actedIn.Film} \\
P(\mathsf{Director} & \mid \mathsf{Person} \sqcap (\exists \mathsf{directed.Film} \sqcap \exists \mathsf{influences}. \\
 & \exists \mathsf{actedIn.Film}) = 0.75 \\
P(\mathsf{FomerActor} & \mid \mathsf{Director} \sqcap \exists \mathsf{actedIn.Film}) = 0.6 \\
\mathsf{HonoredDirector} \equiv & \mathsf{Director} \sqcap \exists \mathsf{hasWonPrize.Prize} \\
\mathsf{FamilyDirector} \equiv & \mathsf{Director} \sqcap (\exists \mathsf{isMarriedTo.Director} \sqcup \exists \mathsf{hasChild.Director}) \\
P(\mathsf{InfluentialDirector} & \mid \mathsf{Director} \sqcap \exists \mathsf{hasWonPrize.Prize} \sqcap \exists \mathsf{influences}. \\
 & \exists \mathsf{isMarriedTo.Director}) = 0.7 \\
P(\mathsf{MostInfluentialDirector} & \mid \mathsf{Director} \sqcap \exists \mathsf{diedOnYear.Year} \sqcap \exists \mathsf{influences}. \\
 & \exists \mathsf{hasWonPrize.Prize}) = 0.8
\end{array}
$$

Learned components range from basic concept definitions such as Actor to probabilistic inclusions for describing most influential directors. Assume we are interested in classifying a person given we know that he/she has acted and directed. According to evidence available:

$$P(\mathsf{Actor}(0) | \mathsf{Person}(0) \sqcap \exists \mathsf{actedIn.Film}(1) \sqcap \exists \mathsf{directed.Film}(2)) = 0.4$$

$$P(\mathsf{Director}(0) | \mathsf{Person}(0) \sqcap \exists \mathsf{actedIn.Film}(1) \sqcap \exists \mathsf{directed.Film}(2)) = 0.55$$

As further evidence is given, probability value changes to:

$$
\begin{array}{ll}
P(\mathsf{Actor}(0) & \mid \mathsf{Person}(0) \\
 & \sqcap (\exists \mathsf{actedIn.Film}(1) \sqcap \exists \mathsf{directed.Film}(2) \\
 & \sqcap \exists \mathsf{influences.Person}(3))) = 0.3
\end{array}
$$

5 Conclusion

We have proposed a method for learning deterministic/probabilistic components of terminologies expressed in CR$\mathcal{ALC}$. Differently from previous approaches, we have produced a combined scheme, where both the deterministic and probabilistic components receive due attention.

This unified learning scheme has the following components: (1) a refinement operator for traversing the search space, (2) probabilistic cover and score relationships for evaluating candidates, (3) a mixed search procedure. Initially, the search aims at finding deterministic concepts. If the score obtained is below a given threshold, a probabilistic inclusion search is conducted (a probabilistic classifier is produced). Experiments with probabilistic terminology in a real-world domain suggest that probabilistic inclusions do lead to improved likelihoods.

Probabilistic description logics offer expressive languages in which to conduct learning, while charging a relatively low cost for inference. The present contribution offers novel ideas for this sort of learning task; we note that the current literature on this topic is rather scarce. Our future work is to investigate the scalability of our learning methods.

Acknowledgements

The first author is supported by CAPES and the third author is partially supported by CNPq. The work reported here has received substantial support through FAPESP grant 2008/03995-5.

References

1. G. Antoniou and F. van Harmelen. *Semantic Web Primer*. MIT Press, 2008.
2. F. Baader and W. Nutt. Basic description logics. In *Description Logic Handbook*, pages 47–100. Cambridge University Press, 2002.
3. C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. DBpedia - a crystallization point for the web of data. *Web Semant.*, 7(3):154–165, 2009.
4. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2001.
5. P. C. G. Costa and K. B. Laskey. PR-OWL: A framework for probabilistic ontologies. In *Proceeding of the 2006 conference on Formal Ontology in Information Systems*, pages 237–249, Amsterdam, The Netherlands, The Netherlands, 2006. IOS Press.
6. F. G. Cozman and R. B. Polastro. Loopy propagation in a probabilistic description logic. In Sergio Greco and Thomas Lukasiewicz, editors, *Second International Conference on Scalable Uncertainty Management*, Lecture Notes in Artificial Intelligence (LNAI 5291), pages 120–133. Springer, 2008.
7. F. G. Cozman and R. B. Polastro. Complexity analysis and variational inference for interpretation-based probabilistic description logics. In *Conference on Uncertainty in Artificial Intelligence*, 2009.
8. C. D'Amato, N. Fanizzi, and T. Lukasiewicz. Tractable reasoning with Bayesian description logics. In *SUM '08: Proceedings of the 2nd international conference on Scalable Uncertainty Management*, pages 146–159, Berlin, Heidelberg, 2008. Springer-Verlag.
9. L. De Raedt, editor. *Advances in Inductive Logic Programming*. IOS Press, 1996.
10. N. Fanizzi, C. D'Amato, and F. Esposito. DL-FOIL concept learning in description logics. In *ILP '08: Proceedings of the 18th International Conference on Inductive Logic Programming*, pages 107–121, Berlin, Heidelberg, 2008. Springer-Verlag.
11. N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–163, 1997.
12. J. Heinsohn. Probabilistic description logics. In *International Conf. on Uncertainty in Artificial Intelligence*, pages 311–318, 1994.
13. L. Iannone, I. Palmisano, and N. Fanizzi. An algorithm based on counterfactuals for concept learning in the semantic web. *Applied Intelligence*, 26(2):139–159, 2007.

14. M. Jaeger. Probabilistic reasoning in terminological logics. In *Principals of Knowledge Representation (KR)*, pages 461–472, 1994.
15. N. Landwehr, K. Kersting, and L. DeRaedt. Integrating Naïve Bayes and FOIL. *J. Mach. Learn. Res.*, 8:481–507, 2007.
16. J. Lehmann. Hybrid learning of ontology classes. In *Proceedings of the 5th International Conference on Machine Learning and Data Mining*, volume 4571 of *Lecture Notes in Computer Science*, pages 883–898. Springer, 2007.
17. J. Lehmann and P. Hitzler. Foundations of refinement operators for description logics. In Hendrick Blockeel, Jude W. Shavlik, and Prasad Tadepalli, editors, *ILP '07: Proceedings of the 17th International Conference on Inductive Logic Programming*, volume 4894 of *Lecture Notes in Computer Science*, pages 161–174. Springer, 2007.
18. J. Lehmann and P. Hitzler. A refinement operator based learning algorithm for the $\mathcal{ALC}$ description logic. In Hendrick Blockeel, Jude W. Shavlik, and Prasad Tadepalli, editors, *ILP '07: Proceedings of the 17th International Conference on Inductive Logic Programming*, volume 4894 of *Lecture Notes in Computer Science*, pages 147–160. Springer, 2007.
19. T. Lukasiewicz. Expressive probabilistic description logics. *Artif. Intell.*, 172(6-7):852–883, 2008.
20. J. E. Ochoa-Luna and F. G. Cozman. An algorithm for learning with probabilistic description logics. In *5th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW) at the 8th International Semantic Web Conference (ISWC)*, pages 63–74, Chantilly, USA, 2009.
21. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: networks of plausible inference*. Morgan Kaufman, 1988.
22. R. B. Polastro and F. G. Cozman. Inference in probabilistic ontologies with attributive concept descriptions and nominals. In *4th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW) at the 7th International Semantic Web Conference (ISWC)*, Karlsruhe, Germany, 2008.
23. J. R. Quinlan and R. M. Cameron-Jones. FOIL: A midterm report. In *Proceedings of the European Conference on Machine Learning*, pages 3–20. Springer-Verlag, 1993.
24. K. Revoredo, J. Ochoa-Luna, and F.G. Cozman. Learning terminologies in probabilistic description logics. In *Proceedings of the 20th Brazilian Symposium on Artificial Intelligence*. To appear, 2010.
25. F. Sebastiani. A probabilistic terminological logic for modelling information retrieval. In *ACM Conf. on Research and Development in Information Retrieval (SIGIR)*, pages 122–130, 1994.
26. F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 697–706, New York, NY, USA, 2007. ACM.

SWRL-F - A Fuzzy Logic Extension of the Semantic Web Rule Language

Tomasz Wiktor Wlodarczyk[1], Martin O'Connor[2], Chunming Rong[1], Mark Musen[2],

[1] University of Stavanger, Norway; [2] Stanford University, USA
tomasz.w.wlodarczyk@uis.no

Abstract. Enhancing Semantic Web technologies with an ability to express uncertainty and imprecision is widely discussed topic. While SWRL can provide additional expressivity to OWL-based ontologies, it does not provide any way to handle uncertainty or imprecision. We introduce an extension of SWRL called SWRL-F that is based on SWRL rule language and uses SWRL's strong semantic foundation as its formal underpinning. We extend it with a SWRL-F ontology to enable fuzzy reasoning in the rule base. The resulting language provides small but powerful set of fuzzy operations that do not introduce inconsistencies in the host ontology.

Keywords: SWRL, SWRL-F, fuzzy logic, fuzzy rules, fuzzy, rule language, risk.

1 Introduction

Fuzzy Logic (FL) has provides a way to express imprecise information and helps in simplifying knowledge representation. For these reasons it is considered to be an important element in Semantic Web (SW) research. Despite the existing research work the problem of supplementing SW with FL remains without implemented, generic, publicly available, standards-based and widely used solution.

In this paper we present SWRL-F, a Fuzzy Logic extension of the Semantic Web Rule Language. It allows expressing imprecise information and helps in simplifying knowledge representation in SWRL. It consists of two parts. SWRL-F ontology that allows representing FL knowledge in the ontology and SWRL rule base, and execution engine that integrates with Protégé [1]. One of the areas where fuzzy logic found significant application are control systems. In this work we based on the control system approach that follows the scheme: collect crisp inputs, fuzzify inputs, perform fuzzy inference, defuzzify inputs, apply crisp outputs [2].

Related Work. Pan et al. [3] propose f-SWRL, a fuzzy extension to SWRL. It includes fuzzy assertions and fuzzy rules, however, does not describe any implementation. Moreover, that approach is criticized in Agarwal and Hitzler [4], who explain that syntax and semantics of f-SWRL actually offer no fuzziness in f-SWRL rules. Bobillo et al. [5] present a semantic fuzzy expert system for a fuzzy balanced scorecard. They use OWL ontology to represent knowledge about variables. They also provide and interface to FuzzyJess to execute fuzzy rules. Protege is used as a development platform; however, implementation focuses only on balanced

scorecard and rules are not based on SWRL. A need for more generic approach is mentioned in conclusions. Stoilos et al. [6] discuss Fuzzy OWL and uncertainty representation with rules. They present a fuzzy reasoning engine that implements a reasoning algorithm for a fuzzy DL language fKD-SHIN. It handles most of OWL features. However, the implementation is proprietary and does not connect directly with any established Semantic Web technologies or tools like OWL, SWRL or Protege. For additional related work one can refer to [7].

Contributions. In SWRL-F we aim to provide a FL extension to SWRL, which is based on standard OWL DL and SWRL. SWRL-F ontology enables description of FL knowledge and its application in SWRL rules. We also implemented a test execution engine and development environment that is publically available[1].

Organization of the Paper. After the Introduction, in Section 2 we explain our design choices for SWRL-F in term of their influence on semantics of rules and logical soundness of ontology. In Section 3 we mention basic constructs of SWRL-F ontology. Further, in Section 4, we describe how to understand and construct fuzzy rules with SWRL-F. We conclude in Section 5.

2 Design Choices

Connection between FL and SW technologies based on DL is a non-trivial problem. We have made four main design choices that influence semantic of the rules and logical soundness of the ontology.

First, SWRL-F must be standard based. It includes anchoring in the well established fuzzy logic scheme. Our leading idea was to follow fuzzy control systems scheme: fuzzification, inference, defuzzification. Moreover, SWRL-F can be fully expressed using OWL and SWRL, by importing SWRL-F ontology that we created. This ontology is purely OWL-based and it is described in the Section 3.

Second, fuzzy inference in SWRL-F is limited to the rules only. This way we can avoid inconsistencies in the ontology. Ontology is used to describe fuzzy knowledge base, however, it can be interpreted in a limited, non-fuzzy way by a DL-reasoner. Until we connect fuzzy rule reasoner knowledge based on SWRL-F ontology has limited use, but it does not create any inconsistencies with standard SW technologies.

Third, fuzzy assertions in SWRL are represented as a standard object property defined in SWRL-F ontology, which has special meaning when interpreted by a fuzzy rule reasoner. It provides the most natural way of expression and can be interpreted (though not in a fuzzy way) by a non-fuzzy rule reasoner.

Fourth, we decided to reuse existing fuzzy rule engine namely FuzzyJess [8]. This allowed us to implement our solution faster and be sure that it will be stable and reasonably efficient. As FuzzyJess is a superset of Jess we could automatically provide compatibility with existing extensions and built-ins available for SWRL and SWRLJESSTab [9]. There is, though, one notable limitation of such approach: not all the OWL constructs can be represented, which follows the limitations as described in [10].

[1] http://protege.cim3.net/cgi-bin/wiki.pl?SWRLF

3 SWRL-F ontology

In order to express necessary fuzzy knowledge, namely fuzzy: sets, terms, variables and values, we have created SWRL-F ontology. Due to limited space, we present here only a few key elements. Representation follows Manchester syntax [11].

```
Class: FuzzyVariable
Class: FuzzyTerm
Class: FuzzyValue
Class: FuzzySet
ObjectProperty: hasFuzzySet
  Domain: FuzzyTerm, FuzzyValue
  Range: FuzzySet
ObjectProperty: hasFuzzyTerm
  Domain: FuzzyVariable
  Range: FuzzyTerm
ObjectProperty: hasFuzzyValue
  Domain: FuzzyVariable
  Range: FuzzyValue
ObjectProperty: hasFuzzyVariable
  Domain: FuzzyValue
  Range: FuzzyVariable
```

4 SWRL-F Rules

Having FuzzyValues and FuzzyTerms described one can construct rules in SWRL-F. To do so we use modified SWRLJessTab. SWRL-F rules are normal SWRL rules that make use of fuzzymatch object property from SWRL-F ontology. If executed using standard rule engine like Jess this property acts as any other object property. However, if run using modified version of SWRLJessTab together with FuzzyJ and FuzzyJess packages, fuzzymatch property allows constructing fuzzy rules.

```
ObjectProperty: fuzzymatch
  Domain: FuzzyValue
  Range: FuzzyTerm
```

Let us analyze a generic example:

`FuzzyValue (?v1) ∧ fuzzymatch(?v1, someFuzzyTerm) ∧ FuzzyValue(?v2) → fuzzymatch(?v2, otherFuzzyTerm)`

The fuzzymatch property is used to calculate degree of membership of FuzzyValue ?v1 in the someFuzzyTerm. FuzzyValues and FuzzyTerms are related by FuzzyVariables. Second use of fuzzymatch allows to bind the value of otherFuzzyTerm to the ?v2 FuzzyValue, basing on the calculated degree of membership.

Many rules can assign new values to the same FuzzyValue. In contrast with standard SWRL where such assertions would not carry any additional semantics, in SWRL-F the values that each rule assigns are then grouped together and collectively

defuzified into one final crisp result. Apart from simplifying management and creation of rules, this allows to create rules in a more natural way.

5 Conclusions

In this paper we presented SWRL-F. It is an extension to SWRL that allows constructing fuzzy rules using lexical variables described it OWL-based ontology. Its general design is based on fuzzy control system approach and together with proper construction of SWRL-F ontology it allows to avoid conflicts between FL and DL in the ontology. SWRL-F can be used to extend any SW application with FL capabilities basing on Protege editor and modified SWRLJessTab.

SWRL-F does not introduce any inconsistencies into a DL-based ontology due to limiting fuzzy inference to rules basing on SWRL-F ontology construction. However, it has the some limitations with regards to OWL representation as explained in [10].

SWRL-F allows easier knowledge management by moving numerical values from rules to ontology. This results in simpler rules and removes hard-coding of those numerical values in rules.

References

[1] "The Protégé Ontology Editor and Knowledge Acquisition System" Available: http://protege.stanford.edu/.

[2] T.J. Ross, "Fuzzy Control Systems," *Fuzzy Logic with Engineering Applications*, Wiley-Blackwell, 2004.

[3] J.Z. Pan, G. Stamou, V. Tzouvaras, and I. Horrocks, "f-SWRL: A Fuzzy Extension of SWRL," *Artificial Neural Networks: Formal Models and Their Applications - ICANN 2005*, 2005, pp. 829-834.

[4] S. Agarwal and P. Hitzler, "Modeling Fuzzy Rules with Description Logics."

[5] F. Bobillo, M. Delgado, J. Gómez-Romero, and E. López, "A semantic fuzzy expert system for a fuzzy balanced scorecard," *Expert Syst. Appl.*, vol. 36, 2009, pp. 423-433.

[6] G. Stoilos, N. Simou, G. Stamou, and S. Kollias, "Uncertainty and the Semantic Web," *IEEE Intelligent Systems*, vol. 21, 2006, pp. 84-87.

[7] T. Lukasiewicz and U. Straccia, "Managing uncertainty and vagueness in description logics for the Semantic Web," *Web Semant.*, vol. 6, 2008, pp. 291-308.

[8] B. Orchard, "Controlling with fuzzy rules," *Jess in Action: Java Rule-Based Systems*, Manning Publications, 2003.

[9] "ProtegeWiki: SWRLJess Tab" Available: http://protege.cim3.net/cgi-bin/wiki.pl?SWRLJessTab.

[10] M. O'connor, H. Knublauch, S. Tu, B. Grosof, M. Dean, W. Grosso, and M. Musen, "Supporting Rule System Interoperability on the Semantic Web with SWRL," *The Semantic Web – ISWC 2005*, 2005, pp. 974-986.

[11] "OWL 2 Web Ontology Language Manchester Syntax" Available: http://www.w3.org/TR/owl2-manchester-syntax/.

A Tractable Paraconsistent Fuzzy Description Logic

Henrique Viana, Thiago Alves, João Alcântara and Ana Teresa Martins

Departamento de Computação, Universidade Federal do Ceará, P.O.Box 12166,
Fortaleza, CE, Brasil 60455-760
{henriqueviana,thiagoalves,jnando,ana}@lia.ufc.br

Abstract. In this paper, we introduce the tractable $pf\text{-}\mathcal{EL}^{++}$ logic, a paraconsistent version of the fuzzy logic $f\text{-}\mathcal{EL}^{++}$. Within $pf\text{-}\mathcal{EL}^{++}$, it is possible to tolerate contradictions under incomplete and vague knowledge. $pf\text{-}\mathcal{EL}^{++}$ extends the $f\text{-}\mathcal{EL}^{++}$ language with a paraconsistent negation in order to represent contradictions. This paraconsistent negation is defined under Belnap's bilattices. It is important to observe that $pf\text{-}\mathcal{EL}^{++}$ is a conservative extension of $f\text{-}\mathcal{EL}^{++}$, thus assuring that the polynomial-time reasoning algorithm used in $f\text{-}\mathcal{EL}^{++}$ can also be used in $pf\text{-}\mathcal{EL}^{++}$.

1 Introduction

A difficult task in a knowledge base that aims to formalise a real world application is to deal with incomplete, imprecise and contradictory information. Hence, it is unreasonable to expect that a knowledge base which allows realistic reasoning based on partial knowledge must always be kept logically consistent. In this sense, in the last century, the paraconsistent logics were designed to handle inconsistencies without deriving anything from a contradiction. Here, we are particularly interested in the paraconsistent logic introduced by Belnap [3]. In addition, there are some logical approaches that attempt to formalise reasoning under incomplete and imprecise knowledge as the fuzzy logic introduced by Zadeh [10].

Although expressive enough to deal with incomplete, imprecise and contradictory information, the satisfiability problem for paraconsistent and fuzzy logics is undecidable. Since real world applications demand efficient inference systems, a family of logics, the Description Logics (DLs) [1], have been proposed. DLs are decidable fragments of classical first-order logic, and they have been customarily used in the definition of ontologies and applications for the Semantic Web.

In [7], a fuzzy logic $f\text{-}\mathcal{EL}^{++}$ with a polynomial-time subsumption algorithm was specially defined to deal with imprecise and vague knowledge. Unfortunately, this logic cannot express negative information. In fact, it was proved that the introduction of the classical negation in DLs leads to undecidability [2].

In this paper, we introduce the tractable $pf\text{-}\mathcal{EL}^{++}$ logic, a paraconsistent version of $f\text{-}\mathcal{EL}^{++}$ that is able to tolerate contradiction under incomplete and vague knowledge. It extends the $f\text{-}\mathcal{EL}^{++}$ language with a paraconsistent negation in order to represent contradictions.

2 Bilattices

In [3] Belnap introduced a logic intended to deal with inconsistent and incomplete information. This logic is capable of representing four truth values: t (true), f (false), $\ddot{\top}$ (overdefined) and $\ddot{\bot}$ (underdefined). The underdefined value represents the total lack of knowledge, while the overdefined one represents the excess of knowledge (conflicts between information). Belnap's logic was generalized by Ginsberg [4], who introduced the notion of bilattices, which are algebraic structures containing an arbitrary number of truth values simultaneously arranged in two partial orders. In the sequel, we will show the definition of bilattices and introduce the particular billatice employed in the representation of fuzzy truth-values in our proposal:

Definition 1 (Complete Bilattice) *Given two complete lattices*[1] $\langle C, \leq_1 \rangle$ *and* $\langle D, \leq_2 \rangle$*, the structure* $\mathcal{B}(C, D) = \langle C \times D, \leq_k, \leq_t, \neg \rangle$ *is a complete bilattice, in which:* $\langle c_1, d_1 \rangle \leq_k \langle c_2, d_2 \rangle$ if $c_1 \leq_1 c_2$ and $d_1 \leq_2 d_2$, $\langle c_1, d_1 \rangle \leq_t \langle c_2, d_2 \rangle$ if $c_1 \leq_1 c_2$ and $d_2 \leq_2 d_1$. *Furthermore,* $\neg : C \times D \to D \times C$ *is a negation operation such that: (1)* $a \leq_k b \Rightarrow \neg a \leq_k \neg b$*, (2)* $a \leq_t b \Rightarrow \neg b \leq_t \neg a$*, (3)* $\neg\neg a = a$*.*

$\mathcal{B}^2 = \langle [0,1] \times [0,1], \leq_t, \leq_k, \neg \rangle$ *is a complete bilattice where* $\neg \langle x_1, x_2 \rangle = \langle x_2, x_1 \rangle$.

In an element $x = \langle x_1, x_2 \rangle$ in $[0,1] \times [0,1]$, x_1 and x_2 represent, respectively, the membership and non-membership degrees of x in $[0,1]$. This means that x_2 can be any value in $[0,1]$ and not necessarily $1 - x_1$ as one would expect in the classical case. It is a very important distinction because it will allow us to identify contradictory truth-values. A truth-value $x = \langle x_1, x_2 \rangle$ is *contradictory* whenever $x_1 + x_2 > 1$.

3 The pf-$\mathcal{EL}^{++}$ Logic

Here we propose a new Description Logic, pf-$\mathcal{EL}^{++}$, by extending f-$\mathcal{EL}^{++}$ [7] with the negation operator $\neg$. Motivated by [6,5], we will employ a bilattice of truth-values to represent the degree of inclusion and non-inclusion of an individual to a concept. The differences between the syntax of pf-$\mathcal{EL}^{++}$ and f-$\mathcal{EL}^{++}$ concepts is that in our proposal we introduce the negation in the alphabet and $\sqcup$ and $\exists$ are replaced respectively by $\otimes_k$ and $\exists_k$. Now it is also possible to use negation to concepts and atomic roles:

Definition 2 (Concept Semantics) *The semantics of* pf-$\mathcal{EL}^{++}$ *individuals and atomic concepts/roles is given by* $I = (\Delta^I, .^I)$*, where the domain* Δ^I *is a nonempty set of elements and* $.^I$ *is a mapping function defined by: each individual* $a \in N_I$ *is mapped to* $a^I \in \Delta^I$*; each atomic concept name* $A \in N_C$ *is mapped to* $A^I : \Delta^I \to [0,1] \times [0,1]$*; each atomic role name* $R \in N_R$ *is mapped to* $R^I : \Delta^I \times \Delta^I \to [0,1] \times [0,1]$.

Each atomic concept/role C is mapped to a pair $\langle P, N \rangle$, where $P, N \in [0,1]$. Intuitively, P denotes the degree in which an element belongs to C, while N denotes the degree in which it does not belong to C. Note that $P + N$ is not necessarily equal to 1 as in the classical case. We define the functions $proj^+ \langle P, N \rangle = P$ and $proj^- \langle P, N \rangle = N$. Concepts can be interpreted inductively as follows, where for all $x \in \Delta^I$:

[1] Let L be a nonempty set and $\leq$ a partial order on L. The pair $\langle L, \leq \rangle$ is a complete lattice if every subset of L has both a least upper bound and a greatest lower bound according to $\leq$.

Syntax	Semantics
$\top$	$\top^I(x) = \langle 1, 0\rangle$
$\bot$	$\bot^I(x) = \langle 0, 1\rangle$
$\neg C$	$(\neg C)^I(x) = \langle N, P\rangle$, if $C^I(x) = \langle P, N\rangle$
$\{a\}$	$\{a\}^I(x) = \begin{cases} \langle 1, 0\rangle \text{ if } x = a^I \\ \langle 0, 1\rangle \text{ otherwise} \end{cases}$
$C \otimes_k D$	$(C \otimes_k D)^I(x) = \langle min(P_1, P_2), min(N_1, N_2)\rangle$, if $C^I(x) = \langle P_1, N_1\rangle$ and $D^I(x) = \langle P_2, N_2\rangle$
$\exists_k R.C$	$(\exists_k R.C)^I(x) = \langle \sup_{y \in \Delta^I}(min(proj^+(R^I(x,y)), proj^+(C^I(y)))), \sup_{y \in \Delta^I}(min(proj^+(R^I(x,y)), proj^-(C^I(y))))\rangle$

The controversial part refers to $\otimes_k$ and $\exists_k$, which were designed in a way that $\neg(C \otimes_k D)^I(x) = (\neg C \otimes_k \neg D)^I(x)$ and $\neg(\exists_k R.C)^I(x) = (\exists_k R.\neg C)^I(x)$. Roughly speaking we can understand them as the counterpart in $\leq_k$ of conjunction ($\sqcap$) and role restriction ($\exists$) respectively. In fact, we can simulate $\sqcap$ and $\exists$ presented in $f\text{-}\mathcal{EL}^{++}$ respectively as $(C \sqcap D)^I(x) \equiv (C \otimes_k D \otimes_k \top)^I(x)$ and $(\exists R.C)^I(x) \equiv (\exists_k R.C \otimes_k \top)^I(x)$. The problem is that we cannot introduce them in $pf\text{-}\mathcal{EL}^{++}$ language because $\neg(C \sqcap D)^I(x) = (\neg C \sqcup \neg D)^I(x)$ and $\neg(\exists R.C)^I(x) = (\forall R.\neg C)^I(x)$. Then, since our aim is to present a tractable paraconsistent fuzzy extension for $\mathcal{EL}^{++}$, the inclusions of disjunction ($\sqcup$) and universal restriction ($\forall$) in $\mathcal{EL}^{++}$ are not allowed. Otherwise, as proved in [2], the algorithm of decidibility will grow exponentially!

We define the notions of Terminological Box (TBox), Assertional Box (ABox) and ontology in $pf\text{-}\mathcal{EL}^{++}$. For now on, consider $T_1, \ldots, T_k, T$ refer to atomic roles or the negation of them. The semantics of negation of roles is similar to negation of concepts.

Definition 3 (TBox/ABox) *A paraconsistent fuzzy TBox in $pf\text{-}\mathcal{EL}^{++}$ is a finite set of internal fuzzy inclusion axioms ($C \sqsubset_n D$), strong fuzzy inclusion axioms ($C \rightarrow_n D$), internal role inclusion axioms ($T_1 \circ \ldots \circ T_k \sqsubset T$) and strong role inclusion axioms ($T_1 \circ \ldots \circ T_k \rightarrow T$). A paraconsistent fuzzy ABox in $pf\text{-}\mathcal{EL}^{++}$ consists of a finite set of assertion axioms of the form $C(a) \geq n$ and $T(a, b) \geq n$, where $n \in [0, 1]$.*

Definition 4 (Ontology) *An ontology or knowledge base in $pf\text{-}\mathcal{EL}^{++}$ is a set composed by a paraconsistent fuzzy TBox and a paraconsistent fuzzy ABox.*

The semantics of both paraconsistent fuzzy general concept inclusions, role inclusions, concept assertion and role assertion is given as follows, where for all $x, y \in \Delta^I$:

Axiom Name	Syntax	Semantics
Internal f-GCI	$C_1 \sqsubset_n C_2$	$min(proj^+(C_1^I(x)), n) \leq proj^+(C_2^I(x))$
Strong f-GCI	$C_1 \rightarrow_n C_2$	$min(proj^+(C_1^I(x)), n) \leq proj^+(C_2^I(x))$, $min(proj^-(C_2^I(x)), n) \leq proj^-(C_1^I(x))$
Internal RIA	$T_1 \circ \ldots \circ T_k \sqsubset T$	$proj^+([T_1^I \circ^t \ldots \circ^t T_k^I](x,y)) \leq proj^+(T^I(x,y))$
Strong RIA	$T_1 \circ \ldots \circ T_k \rightarrow T$	$proj^+([T_1^I \circ^t \ldots \circ^t T_k^I](x,y)) \leq proj^+(T^I(x,y))$, $proj^-(T^I(x,y)) \leq proj^-([T_1^I \circ^t \ldots \circ^t T_k^I](x,y))$
Concept assertion	$C(a) \geq n$	$proj^+(C^I(a^I)) \geq n$
Role assertion	$T(a, b) \geq n$	$proj^+(T^I(a^I, b^I)) \geq n$

Finally, we show the notions of satisfiability and logical consequence in $pf\text{-}\mathcal{EL}^{++}$:

Definition 5 (Satisfiability) *The satisfiability of an axiom α by a fuzzy interpretation I, denoted $I \models \alpha$, is defined as $I \models C_1 \sqsubseteq_n C_2$ iff $\forall x \in \Delta^I$, $min(proj^+(C_1^I(x)), n) \leq proj^+(C_2^I(x))$. The notion is similarly applied to the other axioms shown in the table above. I is a model of an ontology O iff I satisfies each axiom of O.*

Definition 6 (Logical Consequence) *An axiom α is a logical consequence of an ontology O, denoted by $O \models \alpha$, iff every model of O satisfies α.*

Paraconsistency comes to deal with the principle that $\alpha, \neg\alpha \not\vdash \bot$, where α is an axiom. Note that in $pf\text{-}\mathcal{EL}^{++}$, $\bot$ is not logical consequence of α and $\neg\alpha$. For example, consider the axioms $(C(a) \geq 0)$, $(\neg C(a) \geq 0)$ and $(\bot(a) \geq 1)$. We have that $(C(a) \geq 0), (\neg C(a) \geq 0) \not\vdash (\bot(a) \geq 1)$, because there is an interpretation I (say $C^I(a^I) = \langle 0, 0\rangle$) such that $(C(a) \geq 0)^I$ and $(\neg C(a) \geq 0)^I$ are true and $(\bot(a) \geq 1)^I$ is false.

4 Conclusions and Future Works

In this paper, we introduced $pf\text{-}\mathcal{EL}^{++}$, a paraconsistent extension of the fuzzy description logic $f\text{-}\mathcal{EL}^{++}$, that deals with negation on concepts and roles. Inspired in [6], we can show how to translate $pf\text{-}\mathcal{EL}^{++}$ into $f\text{-}\mathcal{EL}^{++}$, preserving logical consequence, and under linear time and space in the size of the ontology. Since there is an algorithm for deciding fuzzy concept subsumptions operating in polynomial time [8], we know that paraconsistency can be simulated by $f\text{-}\mathcal{EL}^{++}$ without the loss of tractability.

Regarding future works, we plan to investigate and extend another approach to fuzzy $\mathcal{EL}$, presented by Vojtás [9], where conjunction is interpreted as a fuzzy aggregation function rather than fuzzy intersection. Another line of research is to extend tractable DLs to deal with probabilistic and possibilistic knowledge.

References

1. F. Baader. *The Description Logic Handbook: theory, implementation, and applications*. Cambridge University Press, 2003.
2. F. Baader, S. Brand, and C. Lutz. Pushing the el envelope. In *Proc. of IJCAI 2005*, pages 364–369. Morgan-Kaufmann Publishers, 2005.
3. N. D. Belnap. A useful four-valued logic. In J. Michael Dunn and G. Epstein, editors, *Modern Uses of Multiple-Valued Logic*, pages 8–37. D. Reidel, 1977.
4. M. Ginsberg. Multivalued logics: A uniform approach to reasoning in artificial intelligence. *Computational Intelligence*, 4:265–316, 1988.
5. Y. Ma, P. Hitzler, and Z. Lin. Algorithms for paraconsistent reasoning with owl. In *The Semantic Web: Research and Applications. Proceedings of the 4th European Semantic Web Conference, ESWC2007*, pages 399–413. Springer, 2007.
6. Y. Ma, P. Hitzler, and Z. Lin. Paraconsistent reasoning for expressive and tractable description logics. Proceedings of the 21st International Workshop on Description Logics, 2008.
7. T. Mailis, G. Stoilos, N. Simou, and G. Stamou. Tractable reasoning based on the fuzzy el ++ algorithm, 2008.
8. G. Stoilos, G. Stamou, and J. Z. Pan. Classifying fuzzy subsumption in fuzzy-el+. 21st International Workshop on Description Logics, 2008.
9. P. Vojtás. A fuzzy el description logic with crisp roles and fuzzy aggregation for web consulting. *Proc. of the 2nd int. workshop on uncertainty reasoning for the semantic web*, 2007.
10. L.A. Zadeh. Fuzzy sets. *Information Control*, 8:338–353, 1965.

Default Logics for Plausible Reasoning with Controversial Axioms

Thomas Scharrenbach[1], Claudia d'Amato[2], Nicola Fanizzi[2], Rolf Grütter[1], Bettina Waldvogel[1], and Abraham Bernstein[3]

[1] Swiss Federal Institute for Forest, Snow and Landscape Research WSL Birmensdorf, Switzerland
{thomas.scharrenbach, rolf.gruetter, bettina.waldvogel}@wsl.ch
[2] Università degli Studi di Bari Bari, Italy {claudia.damato, fanizzi}@di.uniba.it
[3] University of Zurich, Department of Informatics Zurich, Switzerland
{bernstein}@ifi.uzh.ch

Abstract. Using a variant of Lehmann's Default Logics and Probabilistic Description Logics we recently presented a framework that invalidates those unwanted inferences that cause concept unsatisfiability without the need to remove explicitly stated axioms. The solutions of this methods were shown to outperform classical ontology repair w.r.t. the number of inferences invalidated. However, conflicts may still exist in the knowledge base and can make reasoning ambiguous. Furthermore, solutions with a minimal number of inferences invalidated do not necessarily minimize the number of conflicts. In this paper we provide an overview over finding solutions that have a minimal number of conflicts while invalidating as few inferences as possible. Specifically, we propose to evaluate solutions w.r.t. the quantity of information they convey by recurring to the notion of entropy and discuss a possible approach towards computing the entropy w.r.t. an ABox.

1 Introduction

In the Semantic Web, knowledge is represented by ontologies expressed in the Web Ontology Language OWL. The current standard, OWL2 [1], defines different profiles all of which have some Description Logics as a rough syntactic variant. These Description Logics (DL) are decidable fragments of first-order logics where knowledge is explicitly expressed in axioms and assertions. DL knowledge bases have well-defined model-theoretic semantics. They allow to express knowledge on different levels of expressivity and enable to infer new conclusions from existing knowledge.

When ontologies evolve or one ontology is mapped to another, contradictions may be introduced that cause the knowledge base as a whole to be inconsistent. Yet, for an inconsistent knowledge base any conclusion—even meaningless ones—becomes trivially true. One cause of inconsistency is given by assertions of concepts that are inferred to be unsatisfiable. Hence, it is desirable to prevent

concepts from being inferred unsatisfiable. A knowledge base can become inconsistent for other reasons, but *we propose to start off with conflict-free conceptualizations and apply a method that never infers any concept to be unsatisfiable.*

In the Semantic Web, agents interacting with an ontology assume that both the query and the answer are expressible in OWL2. Furthermore, the answer should have meaningful semantics but not infer conflicts. We therefore demand any formalism allowing for plausible reasoning on controversial information to fulfill the following properties:

1. Permanence: The formalism for knowledge representation is not changed.
2. Coherency: No concept is inferred to be unsatisfiable
3. Autonomy: The procedure shall work automatically.
4. Originality: The original information should be kept.
5. Conservation: As little inferred information as possible shall be lost.

We presented a method for solving unsatisfiable concepts [2] using a combination of Lehmann's Default Logics [3] and Lukasiewicz' Probabilistic Description Logics [4]. Instead of removing (explicit) axioms, we propose to *invalidate those inferences that cause concepts to be inferred unsatisfiable* [5]. While it is possible to reason with *all* information provided, we may still produce contradicting inferences. In this paper we show that minimizing the number of inferences invalidated does not necessarily minimize the number of those conflicts. For finding optimal solutions we propose to evaluate these w.r.t. their information content which requires the definition of the entropy of a solution. We discuss a possible approach towards computing the entropy w.r.t. an ABox and give an outlook on future work.

2 Procedure

For each unsatisfiable concept U of an ontology, its *justifications* $J^k_{U\sqsubseteq\bot}$ [6], i.e. the minimal sets of axioms explaining the conflict, are determined in a first step. *Each of these justifications is split up into two sets*: one that contains all axioms which contain the unsatisfiable concept, $\mathit{\Gamma}^k_{U\sqsubseteq\bot}$ and one that contains all other axioms of that very justification, $\Theta^k_{U\sqsubseteq\bot}$ [2]. Afterwards, the *root unsat justifications* are determined, which are those justifications that do not depend on any other justification [7].

According to the partition scheme of Lehmann's Default Logics, the axioms of the root justifications are put into partitions $\mathcal{U}_0, \ldots, \mathcal{U}_N$ and a separate TBox $\mathcal{T}_\Delta$ such that all concepts in $\mathcal{T}_\Delta \cup \mathcal{U}_n$ are satisfiable for $n = 0, \ldots N$. Thanks to the splitting, we do not have to perform additional satisfiability checks for computing the partition. The resulting Default TBox is a family of (classical) TBoxes: $\mathcal{DT} = (\mathcal{T}_\Delta \cup \mathcal{U}_0, \ldots, \mathcal{T}_\Delta \cup \mathcal{U}_N)$. For such a Default TBox we may either use the inference methods provided by Probabilistic Description Logics [4] or stick to classical reasoning on the single partitions, separately. Either approach defines a deductive closure of the Default TBox as a set of OWL2 axioms, but we prefer the latter approach to change the formalism for reasoning only as little as possible.

Instead of putting all axioms of the root unsat justifications into the partitions, we showed in [5] that we indeed have to put *only two axioms of each root unsat justification into the partitions*—one of each $\Theta^k_{U\sqsubseteq\bot}$ and one of each $\Gamma^k_{U\sqsubseteq\bot}$—while we may put the remaining axioms into $\mathcal{T}_\Delta$. While potentially invalidating less inferences, however, finding partitions may become non-deterministic.

We propose to approximate an optimal solution by a (stochastic) search process: On the one hand, the number of possible solutions is exponential in the number of axioms in the justifications. On the other hand, once the justifications are known, *finding a single valid solution can be performed efficiently*, because the complexity of the approach is dominated by the complexity of finding justifications—a task which has to be performed anyhow.

3 Minimizing Conflicts by Minimizing the Entropy

By invalidating the inferences of the kind $\mathcal{DT} \models U \sqsubseteq \bot$ we ignore the conflicts during reasoning. Yet, inferences such as the co-occurrence of $\mathcal{DT} \models A$ and $\mathcal{DT} \models \neg A$ are still possible but not desired. Hence, a performance measure that assesses the quality of a solution must not only take into account the number of inferences invalidated but, even more important, the number of conflicts still remaining.

Assume the simple TBox $\mathcal{T} = \{B \sqsubseteq A, C \sqsubseteq B, C \sqsubseteq \neg A, \}$ which has two Default TBoxes as potential solutions:

$$\mathcal{DT}^0 \text{ with } \mathcal{T}^0_\Delta = \{C \sqsubseteq B\}, \quad \mathcal{U}^0_0 = \{B \sqsubseteq A\}, \quad \mathcal{U}^0_1 = \{C \sqsubseteq \neg A\}$$
$$\mathcal{DT}^1 \text{ with } \mathcal{T}^1_\Delta = \{C \sqsubseteq \neg A\}, \quad \mathcal{U}^1_0 = \{B \sqsubseteq A\}, \quad \mathcal{U}^1_1 = \{C \sqsubseteq B\}$$

In contrast to the latter, the first Default TBox $\mathcal{DT}^0$ preserves the inference $C \sqsubseteq A$. Yet, in the presence of an ABox that infers the assertion $C(i)$, the assertion $A(i)$ as well as its complement $\neg A(i)$ can be inferred. The second Default TBox $\mathcal{DT}^1$, in contrast, infers only $\neg A(i)$. It is preferred over $\mathcal{DT}^0$, because it contains fewer conflicts than $\mathcal{DT}^1$.

Conflicts potentially reduce the information content of a knowledge base. For minimizing the number of conflicts as well as the number of inferences invalidated we are currently investigating *qualitative measures based on the entropy* of a possible solution. As opposed to methods based on the structure of an ontology [8], we propose that an entropy-measure should take into account the ambiguity of different ABoxes.

In information theory, the entropy measures the average information content of a random variable we are missing when the value of the random variable is not known [9]. If we know the probability mass function p of the random variable X, we may explicitly denote the entropy by $\mathcal{H}(X) = -\sum_{n=0}^{N} p(x_n) \log p(x_n)$. In case $p(x_n) = 0$, then $p(x_n) \log p(x_n) = 0$. We propose to approximate the probability mass function $p_\mathcal{A}$ for the axioms $B \sqsubseteq A \in \mathcal{DT}$ by counting assertions for the concept $(\neg B \sqcup A)$ found by the instance retrieval service of the reasoning process:

$$p_\mathcal{A}(B \sqsubseteq A) = \frac{|\{x \in \mathcal{A}^\mathcal{I} \quad | \quad \mathcal{T}, \mathcal{A} \models (\neg B \sqcup A)(x)\}|}{\sum_{D \sqsubseteq C \in \mathcal{DT}} |\{y \in \mathcal{A}^\mathcal{I} \quad | \quad \mathcal{T}, \mathcal{A} \models (\neg D \sqcup C)(y)\}|}$$

The entropy of a Default TBox $\mathcal{DT}$ measures the information content of its axioms w.r.t. an ABox $\mathcal{A}$: $\mathcal{H}(\mathcal{DT}, \mathcal{A}) = -\sum_{B \sqsubseteq A \in (\mathcal{DT})} p_{\mathcal{A}}(B \sqsubseteq A) \log p_{\mathcal{A}}(B \sqsubseteq A)$. For the Default TBoxes in the example above, we obtain an entropy of $\mathcal{H}(\mathcal{DT}^0) = -\log(1/3)$ and $\mathcal{H}(\mathcal{DT}^1) = -\log(1/2)$ which would make us choose $\mathcal{DT}^1$ rather than $\mathcal{DT}^0$. Our current hypothesis is that *a Default TBox with minimal entropy also minimizes the number of explicit conflicts w.r.t. an ABox.* A prototype implementation is available [4].

4 Conclusion

We recently introduced a framework that never infers any concept to be unsatisfiable while keeping all originally provided information. This allows plausible reasoning on ontologies that possibly contain controversial information—as it is the case for mapped or dynamic ontologies. Finding solutions is non-deterministic and requires optimization techniques that, in turn, require a performance measure for evaluating the quality of possible solutions.
While reasoning ignores conflicts, they are still present in the knowledge base and may lead to sub-optimal results. It was shown that solutions invalidating a minimal number of inferences do not necessarily minimize the number of conflicts still present. For minimizing these we proposed to use an entropy-based performance measure. We provided a definition for the entropy of a solution w.r.t an ABox which is currently being further investigated.

References

1. Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S.: OWL 2 Web Ontology Language Primer. W3C Recommendation, W3C (2009)
2. Scharrenbach, T., Grütter, R., Waldvogel, B., Bernstein, A.: Structure Preserving TBox Repair using Defaults. In: 23rd Intl. Workshop on Description Logics. (2010)
3. Lehmann, D.: Another perspective on default reasoning. Ann. Math. Artif. Intell **15** (1995) 61–82
4. Lukasiewicz, T.: Expressive probabilistic description logics. Art. Intell. **172**(6-7) (2008) 852–883
5. Scharrenbach, T., d'Amato, C., Fanizzi, N., Grütter, R., Waldvogel, B., Bernstein, A.: Unsupervised conflict-free ontology evolution without removing axioms. In: 4th International Workshop on Ontology Dynamics (IWOD-2010). (to appear).
6. Schlobach, S., Cornet, R.: Non-standard reasoning services for the debugging of description logic terminologies. In: Proc. of IJCAI 2003. (2003) 355–362
7. Kalyanpur, A., Parsia, B., Sirin, E., Hendler, J.: Debugging unsatisfiable classes in owl ontologies. Journal of Web Semantics **3**(4) (2005) 268–293
8. Doran, P.S., Tamma, V., Payne, T.R., Palmisano, I.: An entropy inspired measure for evaluating ontology modularization. In: Proc. of KCAP2009. (2009) 73–80
9. Shannon, C.E.: A mathematical theory of communication. Bell System Technical Journal **27** 379–423, 623–656

[4] http://www.wsl.ch/info/mitarbeitende/scharren/owl-defaults/

Tractability of the Crisp Representations of Tractable Fuzzy Description Logics

Fernando Bobillo[1] and Miguel Delgado[2]

[1] Dpt. of Computer Science and Systems Engineering, University of Zaragoza, Spain
[2] Dpt. of Computer Science and Artificial Intelligence, University of Granada, Spain
Email: fbobillo@unizar.es, mdelgado@ugr.es

Abstract. An important line of research within the field of fuzzy DLs is the computation of an equivalent crisp representation of a fuzzy ontology. In this short paper, we discuss the relation between tractable fuzzy DLs and tractable crisp representations. This relation heavily depends on the family of fuzzy operators considered.

Introduction. Despite the undisputed success of ontologies, classical ontology languages are not appropriate to deal with vagueness or imprecision in the knowledge, which is inherent to most of the real world application domains. As a solution, several fuzzy extensions of Description Logics (DLs) have been proposed in the literature. For a good survey we refer the reader to [1].

An important line of research within the field of fuzzy DLs is the computation of an equivalent crisp representation of a fuzzy ontology. This way, it is possible to reason with the obtained crisp ontology, making it possible to reuse classical ontology languages (e.g., OWL 2), DL reasoners, and other resources. It is possible to reason with very expressive fuzzy DLs, and with different families of fuzzy operators (also called *fuzzy logics*), namely Zadeh [2], Gödel [3], and Łukasiewicz [4]. To be precise, in Gödel and Łukasiewicz it is necessary to restrict to the finite case, i.e., where the set of degrees of truth is finite and fixed.

In the last years, there is a growing interest in the study of *tractable DLs*. In these logics, the expressive power is compromised for the efficiency of reasoning. In OWL 2, the current standard language for ontology representation, three fragments (called *profiles*) have been identified, namely *OWL 2 EL*, *OWL 2 QL*, and *OWL 2 RL* [5]. Table 1 shows the relation of some OWL 2 constructors and its fragments. In *OWL 2 EL* and *OWL 2 RL*, the basic reasoning tasks can be performed in a time which is polynomial with respect to the size of the ontology. In *OWL 2 QL*, conjunctive query answering can be performed in LogSpace with respect to the size of the assertions.

Sometimes, the crisp representation of a fuzzy KB enjoys the following property: given a fuzzy ontology $\mathcal{O}$ in a fuzzy DL language $\mathcal{X}$, the crisp representation of $\mathcal{O}$ is in the (crisp) DL $\mathcal{X}$. The objective of this paper is to determine in a precise way when this property is verified, focusing on the case of tractable fuzzy DLs, which is a very interesting case in real-world applications.

Definition 1. *A fuzzy DL language $\mathcal{X}$ is* closed under reduction *iff the crisp representation of a fuzzy ontology in $\mathcal{X}$ is in the (crisp) DL language $\mathcal{X}$.*

Table 1. Summary of the relation among OWL 2 and its three profiles.

OWL 2	OWL 2 EL	OWL 2 QL	OWL 2 RL
Class	✓	✓	✓
ObjectIntersectionOf	✓	restricted	✓
ObjectUnionOf			restricted
ObjectComplementOf		restricted	restricted
ObjectAllValuesFrom			restricted
ObjectSomeValuesFrom	✓	restricted	restricted
DataAllValuesFrom			restricted
DataSomeValuesFrom	✓	✓	restricted
...			
ObjectProperty	✓	✓	✓
DatatypeProperty	✓	✓	✓
...			
ClassAssertion	✓	✓	✓
ObjectPropertyAssertion	✓	✓	✓
SubClassOf	✓	✓	✓
SubObjectPropertyOf	✓	✓	✓
SubDataPropertyOf	✓	✓	✓
...			

In the following, we will assume that $\mathcal{X}$ is not more expressive than $\mathcal{SROIQ}(\mathbf{D})$.

Fuzzy DLs. We assume the reader to be familiar with fuzzy DLs [1]. We note that the many existing proposals usually differ in syntax, semantics, and logical properties. In this paper, we consider fuzzy DLs with the following features:

- Concepts and roles are syntactically the same as in the crisp case.
- Axioms are syntactically the same as in the crisp case, with the exception of concept assertions, role assertions, general concept inclusions (GCIs), and role hierarchies, where a crisp axiom τ is extended with a lower bound as $\langle \tau \rhd \alpha \rangle$, with $\rhd \in \{\geq, >\}$, and $\alpha \in [0, 1]$. For instance, $\langle a : C \sqcap D \geq 0.6 \rangle$ means that the concept assertion $a : C \sqcap D$ is true with degree at least 0.6.
- The semantics of classes, properties and axioms depends on some fuzzy logical operators, namely a t-norm, a t-conorm, a negation, and an implication. For instance, the semantics of the conjunction is given by a t-norm. Fuzzy DLs with different fuzzy operators have many different logical properties.

Crisp representations of fuzzy DLs. The basic idea of the crisp representation is to use some basic crisp concepts and roles, representing the α-cuts of the fuzzy concepts and roles. To keep the semantics of the α-cuts, some axioms must be introduced, namely GCIs and role hierarchies. Finally, every axiom of the fuzzy ontology is represented, independently from other axioms, using these basic crisp elements. An important property of these crisp representations is that, although the number of axioms in the TBox and the RBox increase, the number of axioms in the ABox is constant. Let us illustrate this with an example.

Example 1. Assume that a fuzzy ontology $\mathcal{K}$ includes the set of axioms $\{\langle a : \exists R.C \geq 0.6 \rangle, \langle a : \neg \exists R.C > 0.8 \rangle\}$. The crisp representation of the ontology must consider the crisp concepts $C_{\geq 0.6}, C_{\geq 0.8}$, and the crisp roles $R_{\geq 0.6}, R_{\geq 0.8}$, which

produce the GCI $C_{\geq 0.8} \sqsubseteq C_{\geq 0.6}$ and the role hierarchy $R_{\geq 0.8} \sqsubseteq R_{\geq 0.6}$. Assuming that the t-norm is the minimum and the negation is the standard (Łukasiewicz), the crisp representation of the axioms is $\{a : \exists R_{\geq 0.6}.C_{\geq 0.6}, a : \forall R_{\geq 0.8}.(\neg C_{\geq 0.8})\}$.

The case of Zadeh fuzzy logic. The full details of the crisp representation in Zadeh $\mathcal{SROIQ}(\mathbf{D})$ can be found in [2]. Zadeh logic makes it possible to obtain smaller crisp representations than with Gödel and Łukasiewicz logics. For instance, in Zadeh logic, from $\langle a : C \sqcap D \geq 0.6 \rangle$ we can deduce both $\langle a : C \geq 0.6 \rangle$ and $\langle a : D \geq 0.6 \rangle$. However, in Łukasiewicz logic, this is not possible, and we have to build a disjunction over all the possibilities. In Gödel implication, we have a similar problem. In the case of Zadeh logic, we have the following property:

Property 1. In Zadeh fuzzy logic, a fuzzy DL language $\mathcal{X}$ is closed under reduction iff it includes GCIs and role hierarchies. □

The proof of this property is trivial from the crisp representation [2]. This result applies, for instance, to logics more expressive than $\mathcal{ALCH}$, such as $\mathcal{SROIQ}(\mathbf{D})$. Furthermore, it also applies to the DLs that are equivalent to the profiles *OWL 2 EL*, *OWL 2 QL*, and *OWL 2 RL* (see Table 1).

Example 2. Consider again the fuzzy ontology $\mathcal{K}$ from Example 1, and assume that the language of $\mathcal{K}$ is $\mathcal{ALC}$. Since $\mathcal{ALC}$ does not contain role hierarchies, the second condition of Property 1 fails, and hence fuzzy $\mathcal{ALC}$ is not closed under reduction. This is intuitive, because the crisp representation contains role hierarchies ($R_{\geq 0.8} \sqsubseteq R_{\geq 0.6}$). □

The case of Gödel fuzzy logic. The full details of the crisp representation in Gödel $\mathcal{SROIQ}(\mathbf{D})$ can be found in [3]. This case is very similar to the previous one. In fact, using a similar reasoning, it can be seen that the following property is verified by the three OWL 2 profiles.

Property 2. In Gödel fuzzy logic, a fuzzy DL language $\mathcal{X}$ is closed under reduction iff it verifies each of the following conditions:

- $\mathcal{X}$ includes GCIs.
- $\mathcal{X}$ includes role hierarchies.
- If $\mathcal{X}$ includes universal (all) restrictions, then it also include conjunction. □

The case of Łukasiewicz fuzzy logic. The full details of the crisp representation in Łukasiewicz $\mathcal{ALCHOI}$ can be found in [4].

Property 3. In Łukasiewicz fuzzy logic, a fuzzy DL language $\mathcal{X}$ is not closed under reduction if it verifies some of the following conditions:

- $\mathcal{X}$ does not include GCIs.
- $\mathcal{X}$ does not include role hierarchies.
- $\mathcal{X}$ includes one and only one of the constructors disjunction and conjunction.

– $\mathcal{X}$ includes existential (some) restrictions, but it does not include disjunction.
– $\mathcal{X}$ includes universal (all) restrictions, but it does not include conjunction.

□

Again, the proof of this property is trivial from the crisp representation [4]. The three OWL 2 profiles verify this property. *OWL 2 EL* and *OWL 2 QL* support conjunction but not disjunction (see Table 1); and *OWL 2 RL* allows intersection as a superclass expression, but does not allow disjunction there [5].

Note that this property is formulated in a different way. The reason is that a crisp representation for a fuzzy DL more expressive than $\mathcal{ALCHOI}$ is still unknown. Hence, rather than a general result, we only have a partial one.

Size of the crisp representations. In Zadeh and Gödel *OWL 2 QL* we obtain a crisp ontology where the ABox has the same number of axioms as the original fuzzy ABox. Hence, tractability is preserved, since the complexity of reasoning depends on the number of assertions.

In Zadeh and Gödel *OWL 2 EL* and *OWL 2 RL*, we obtain a crisp ontology in a tractable language. However, the TBox and the RBox are larger than in the original fuzzy ontology. This increase in the size is an issue to consider when dealing with tractable fuzzy DLs from a practical point of view, as reasoning depends on the size of the ontology.

In Gödel *OWL 2 QL*, a fuzzy universal restriction is mapped into a (crisp) conjunction of universal restrictions. Hence, the resulting ontology is bigger than in the Zadeh case. This does not happen in *OWL 2 EL* nor in *OWL 2 QL*, as they do not allow universal restrictions (see Table 1).

In tractable fuzzy DLs, it is specially important to use optimized crisp representations. For instance, domain and range restrictions can be treated as GCIs, but their crisp representation are more efficient if treated as special cases [2].

Acknowledgement. The authors have been partially supported by the Spanish Ministry of Science and Technology (project TIN2009-14538-C02-01).

References

1. Lukasiewicz, T., Straccia, U.: Managing uncertainty and vagueness in description logics for the semantic web. Journal of Web Semantics **6**(4) (2008) 291–308
2. Bobillo, F., Delgado, M., Gómez-Romero, J.: Crisp representations and reasoning for fuzzy ontologies. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **17**(4) (2009) 501–530
3. Bobillo, F., Delgado, M., Gómez-Romero, J., Straccia, U.: Fuzzy description logics under Gödel semantics. Int. J. of Approximate Reasoning **50**(3) (2009) 494–514
4. Bobillo, F., Straccia, U.: Towards a crisp representation of fuzzy description logics under Łukasiewicz semantics. In Proceedings of ISMIS 2008. Volume 4994 of Lecture Notes in Computer Science, Springer-Verlag (2008) 309–318
5. OWL 2 Web Ontology Language Profiles. `http://www.w3.org/TR/owl2-profiles`.