# Reasoning with Large A-Boxes in Fuzzy Description Logics using DL reasoners: An experimental evaluation

P. Cimiano[1], P. Haase[1], Q. Ji[1], T. Mailis[2], G. Stamou[2], G. Stoilos[2], T. Tran[1], V. Tzouvaras[2]

[1]Institute AIFB, Universität Karlsruhe, Germany
{pci,pha,qiji,dtr}@aifb.uni-karlsruhe.de

[2]National and Technical University of Athens
{theofilos,gstam,gstoil,tzouvaras}@image.ntua.gr

**Abstract.** While knowledge representation languages developed in the context of the Semantic Web build mainly on crisp knowledge, fuzzy reasoning seems to be needed for many applications, e.g. for retrieval of images or other resources. However, techniques for fuzzy reasoning need to scale to large A-Boxes for practical applications. As a first step towards clarifying whether fuzzy reasoning techniques can indeed scale, we examine a specific point in space. Earlier research has shown that fuzzy description logics can be transformed to crisp description logics in a satisfiability-preserving fashion. This thus opens the possibility of using standard description logic reasoners for reasoning with fuzzy OWL ontologies. As the transformation produces a quadratic blow-up of the T-Box, a crucial question is if such an approach is feasible from a performance point of view. We provide an empirical evaluation on four different ontologies of varying complexity and with generated A-Boxes of up to a million individuals. To our knowledge, we thus provide the first systematic and empirical evaluation of a fuzzy reasoning approach based on a reduction to standard description logics.

## 1 Introduction

Current knowledge representation formalisms for the Semantic Web such as OWL or RDF(S) rely on crisp logics. However, it has become clear that the assumption that all knowledge is crisp is unsustainable. There are various reasons for this. First of all, the large amount of knowledge needed to bootstrap the semantic web can neither be acquired by hand nor by simply importing legacy data. Thus, knowledge acquisition techniques are needed. However, knowledge acquisition techniques building on machine learning or natural language processing techniques are inherently error-prone. Consequently, the need arises to represent the certainty or confidence of the extracted information. Further, for the purpose of retrieval, e.g. of images or other data, we need ways to assess the degree to which a certain answer matches the information needs of a query (see [1]). Restricting ourselves to the crisp case would yield very brittle systems which only return answers in case information is perfectly modeled and perfectly matches the user query. Finally, there are concepts which are inherently fuzzy and which can not be represented in a crisp way. These are concepts representing linguistic variables in the sense of Zadeh [2], e.g. such as *hot*, *young*, *cheap* etc. Thus, various extensions to Semantic Web languages have been presented to deal with imperfect information, i.e. probabilistic approaches such as [3], [4] for OWL, [5] for RDF as well as approaches based on Fuzzy Logics such as [6] and [7]. For an overview of non-crisp ontology formalisms the reader is referred to [8]. Whatever the formalism used is, one requirement is of crucial importance: the availability of scalable reasoning procedures to deal with imperfect knowledge.

In this paper, we thus ask ourselves a straightforward question with a not so straightforward answer: are current reasoning procedures mature and scalable enough to reason with a large amount of imperfect (e.g. fuzzy) information? Giving a principled and empirically grounded answer to this problem is certainly out of the scope of this and many articles to be published. In fact, we are convinced that it is only through systematic experimental evaluation that we will find out which techniques can scale towards real scenarios with millions of facts to be stored and retrieved.

As a first step towards answering this question, the contribution of this paper is to examine a very specific setting, i.e. we assume that knowledge is represented using fuzzy description logics and analyze the scalability of the KAON2 reasoner on the knowledge bases resulting from reducing Fuzzy Description Logics (see [6, 7]) to standard description logics based on the idea described in [9]. KAON2 is a resolution-based $\mathcal{SHIQ}(\mathbf{D})$-reasoner which transforms a $\mathcal{SHIQ}(\mathbf{D})$ knowledge base into a Disjunctive Datalog Program. This allows to reuse optimization techniques from database technology to reason with large A-Boxes. The aim of our paper is to explore if one can indeed reason with large Fuzzy A-Boxes, a necessary requirement for large-scale knowledge management applications. In this sense this paper can be seen as a contribution towards a larger goal: the one of clarifying whether fuzzy knowledge representation can indeed scale towards meeting the size of large-scale applications.

The structure of this paper is as follows: in Section 2, we introduce the logic $\mathrm{f}_{KD}$-$\mathcal{SHIN}$ as well as the reduction to crisp OWL presented already in [7] in order to make the paper self-contained. In Section 3 we briefly describe the OWL-DL reasoner used in our experiments, i.e. the KAON2 reasoner. In Section 4 we present our experiments and results. Finally, in Section 5 we discuss some related work and conclude.

## 2 Fuzzy Description Logics and their reduction to Classical DLs

In this section we provide a brief introduction to the fuzzy Description Logic (DL) $\mathrm{f}_{KD}$-$\mathcal{SHIN}$ [10]. $\mathrm{f}_{KD}$-$\mathcal{SHIN}$ is a fuzzy extension of the $\mathcal{SHIN}$ DL [11] which uses the fuzzy operators $1-x$, max and min for performing the fuzzy set theoretic operations. For a general fuzzy extension of $\mathcal{SHIN}$ please refer to [6].

Fuzzy Description Logics [12] have been proposed as powerful knowledge representation languages capable of capturing vague (fuzzy) knowledge that exists in many applications. Fuzzy DLs and ontologies usually keep the same syntax as their crisp (classical) counterpart, without adding any additional degrees to them; notable exceptions are [6] and [13]. Thus, using concepts and roles we can build *concept descriptions* in the usual way by using *conjunctions* ($C \sqcap D$), *disjunctions* ($C \sqcup D$), *negation* ($\neg C$), *existential* ($\exists R.C$) and *universal quantification* ($\forall R.C$) as well as *cardinality restrictions* ($\geq nR$, $\leq nR$). Similarly, concept and role axioms, like *concept equivalence* ($C \equiv D$), *concept subsumption* ($C \sqsubseteq D$), *role inclusions* ($R \sqsubseteq S$) and *transitivity* ($\mathsf{Trans}(R)$) can be defined in the usual way. Thus, the definitions of a TBox and an RBox are as usual.

**Table 1.** Semantics of concepts descriptions and axioms

| Constructor | Syntax | Semantics |
|---|---|---|
| top | $\top$ | $\top^{\mathcal{I}}(a) = 1$ |
| bottom | $\bot$ | $\bot^{\mathcal{I}}(a) = 0$ |
| general negation | $\neg C$ | $(\neg C)^{\mathcal{I}}(a) = 1 - C^{\mathcal{I}}(a)$ |
| conjunction | $C \sqcap D$ | $(C \sqcap D)^{\mathcal{I}}(a) = \min(C^{\mathcal{I}}(a), D^{\mathcal{I}}(a))$ |
| disjunction | $C \sqcup D$ | $(C \sqcup D)^{\mathcal{I}}(a) = \max(C^{\mathcal{I}}(a), D^{\mathcal{I}}(a))$ |
| exists restriction | $\exists R.C$ | $(\exists R.C)^{\mathcal{I}}(a) = \sup_{b \in \Delta^{\mathcal{I}}} \{\min(R^{\mathcal{I}}(a,b), C^{\mathcal{I}}(b))\}$ |
| value restriction | $\forall R.C$ | $(\forall R.C)^{\mathcal{I}}(a) = \inf_{b \in \Delta^{\mathcal{I}}} \{\max(1 - R^{\mathcal{I}}(a,b), C^{\mathcal{I}}(b))\}$ |
| at-most | $\leq pR$ | $(\leq pR)^{\mathcal{I}}(a) = \inf\limits_{b_1,\ldots,b_{p+1} \in \Delta^{\mathcal{I}}} \max(1 - \max\limits_{i=1}^{p+1} R^{\mathcal{I}}(a,b_i), \max\limits_{i<j}\{b_i = b_j\})$ |
| at-least | $\geq pR$ | $(\geq pR)^{\mathcal{I}}(a) = \sup\limits_{b_1,\ldots,b_p \in \Delta^{\mathcal{I}}} \min(\min\limits_{i=1}^{p} R^{\mathcal{I}}(a,b_i), \min\limits_{i<j}\{b_i \neq b_j\})$ |
| inverse role | $R^-$ | $(R^-)^{\mathcal{I}}(b,a) = R^{\mathcal{I}}(a,b)$ |
| equivalence | $C \equiv D$ | $\forall a \in \Delta^{\mathcal{I}}.C^{\mathcal{I}}(a) = D^{\mathcal{I}}(a)$ |
| sub-concept | $C \sqsubseteq D$ | $\forall a \in \Delta^{\mathcal{I}}.C^{\mathcal{I}}(a) \leq D^{\mathcal{I}}(a)$ |
| transitive role | $\mathsf{Trans}(R)$ | $\forall a,b \in \Delta^{\mathcal{I}}.R^{\mathcal{I}}(a,b) \geq \sup_{c \in \Delta^{\mathcal{I}}}\{t(R^{\mathcal{I}}(a,c), R^{\mathcal{I}}(c,b))\}$ |
| sub-role | $R \sqsubseteq S$ | $\forall a,b \in \Delta^{\mathcal{I}}.R^{\mathcal{I}}(a,b) \leq S^{\mathcal{I}}(a,b)$ |
| concept assertions | $(a : C)\bowtie n$ | $C^{\mathcal{I}}(a^{\mathcal{I}})\bowtie n$, $\bowtie \in \{\geq, >, \leq, <\}$ |
| role assertions | $((a,b) : R) \rhd n$ | $R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) \rhd n$, $\rhd \in \{\geq, >\}$ |

Fuzzy DLs extend the syntax of assertions with membership degrees, thus allowing to create *fuzzy assertions* (see [12, 10, 14, 15]) as well as to annotate individual axioms (assertions) with degrees of membership. More formally, a *fuzzy assertion* [12] is of the form $(a : C)\bowtie n$ or $((a,b) : R) \rhd n$, where $\bowtie \in \{\geq, >, \leq, <\}$, $\rhd \in \{\geq, >\}$ and $a, b$ are individuals, and $n \in (0,1]$. In many cases we write $(a : C) = n$ instead of writing two fuzzy assertions of the form $(a : C) \geq n$ and $(a : C) \leq n$. For example, one is able to state that *grass* is $\mathsf{Green}$ to a degree of at least 0.7, writing $(grass : \mathsf{Green}) \geq 0.7$, or that *athens* is $\mathsf{nearTo}$ *rome* to a degree of at least 0.8, by the axiom $((athens, rome) : \mathsf{nearTo}) \geq 0.8$. A set of fuzzy assertions defines a (fuzzy) ABox.

The semantics of fuzzy DLs are provided by *fuzzy interpretations* [12], which extend classical interpretations to the unit interval $[0, 1]$. A fuzzy interpretation is a pair $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ where the domain $\Delta^\mathcal{I}$ is a non-empty set of objects and $\cdot^\mathcal{I}$ is a *fuzzy interpretation function*, which maps

- an individual name $a \in \mathbf{I}$ to an element $a^\mathcal{I} \in \Delta^\mathcal{I}$,
- a concept name $\mathsf{A} \in \mathbf{C}$ to a membership function $\mathsf{A}^\mathcal{I} : \Delta^\mathcal{I} \to [0, 1]$,
- a role name $R \in \mathbf{R}$ to a membership function $R^\mathcal{I} : \Delta^\mathcal{I} \times \Delta^\mathcal{I} \to [0, 1]$.

Intuitively, an object (pair of objects) can now belong to a fuzzy concept (role) to any degree between 0 and 1. For example, $\mathsf{HotPlace}^\mathcal{I}(\mathtt{Rome}^\mathcal{I}) = 0.7$, means that $\mathtt{Rome}^\mathcal{I}$ is a hot place to a degree equal to 0.7. Fuzzy interpretations can be extended to interpret $\text{f}_{KD}$-$\mathcal{SHIN}$-concepts and roles with the aid of the fuzzy operators $1 - x$, min and max. The complete semantics for concept descriptions, concept and role axioms are depicted in Table 1.

We can now define the inference problems for fuzzy DLs. A fuzzy knowledge base $\Sigma$ is *satisfiable* (*unsatisfiable*) iff there exists (does not exist) a fuzzy interpretation $\mathcal{I}$ which satisfies all axioms in $\Sigma$. An f-$\mathcal{SHIN}$-concept $C$ is *n-satisfiable* w.r.t. $\Sigma$ iff there exists a model $\mathcal{I}$ of $\Sigma$ for which there is some $a \in \Delta^\mathcal{I}$ such that $C^\mathcal{I}(a) = n$, and $n \in (0, 1]$; $C$ subsumes $D$ w.r.t. $\Sigma$ iff for every model $\mathcal{I}$ of $\Sigma$ we have $\forall d \in \Delta^\mathcal{I}, C^\mathcal{I}(d) \leq D^\mathcal{I}(d)$; a fuzzy ABox $\mathcal{A}$ is *consistent* (*inconsistent*) w.r.t. a fuzzy TBox $\mathcal{T}$ and RBox $\mathcal{R}$ if there exists (does not exist) a model $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{R}$ that satisfies each assertion in $\mathcal{A}$. Given a fuzzy concept axiom, a fuzzy role axiom or a fuzzy assertion $\phi$, $\Sigma$ *entails* $\phi$, written $\Sigma \models \phi$, iff for all models $\mathcal{I}$ of $\Sigma$, $\mathcal{I}$ satisfies $\phi$.

Currently, there are many proposals for reasoning in various different fragments of fuzzy Description Logics ranging from tableaux-based [12, 10] to optimization-based [16, 17]. Straccia proposed in [9] a method for reducing an $\text{f}_{KD}$-$\mathcal{ALCH}$ knowledge base to a classical $\mathcal{ALCH}$ one. The purpose of the reduction is to reduce the reasoning problem from f-DLs to crisp DLs in order to allow to use existing optimized reasoners, like FaCT [18], Pellet [19] or KAON2 (see Section 3) to perform the inference tasks of fuzzy DLs. Later, Bobillo et al. extended the method [13] to cover the full fuzzy OWL language, while quite recently a number of optimisations of the method have been also proposed [20]. In the following we sketch the main idea and refer the interested reader to [9, 13, 20] for a detailed presentation.

The main idea behind the reduction is that a fuzzy assertion of the form $(a : C) \geq n$, where $a$ is an individual and $n \in [0, 1]$ can be represented by a crisp assertion of the form $a : C_{\geq n}$, where $C_{\geq n}$ is a new concept. Intuitively, $C_{\geq n}$ stands for the set of objects that belong to $C$ to a degree greater or equal than $n$. Then there are two important points for the reduction:

1. The number of different membership degrees we have to consider in the reduction is finite (this is a property of $\text{f}_{KD}$-DLs) and more precisely defined by the set: $N^\Sigma = \{0, 0.5, 1\} \cup \{n, 1 - n \mid (a : C) \bowtie n \in \mathcal{A} \text{ or } ((a, b) : R) \bowtie n \in \mathcal{A}\}$.
2. In order for the reduction to be satisfiability-preserving, additional concept and role axioms need to be added. For example, if $n_1, n_2$ are two degrees from $N^\Sigma$ and $n_1 \leq n_2$, then it obviously holds that $A_{\geq n_2} \sqsubseteq A_{\geq n_1}$. In summary

for each atomic concept in the fuzzy KB the following axioms are added [9, 13]:

$$
\begin{array}{llll}
(1) & A_{\geq n_{i+1}} \sqsubseteq A_{>n_i}, & & A_{>n_i} \sqsubseteq A_{\geq n_i} \\
(2) & A_{<n_j} \sqsubseteq A_{\leq n_j}, & & A_{\leq n_i} \sqsubseteq A_{<n_{i+1}} \\
(3) & A_{\geq n_j} \sqcap A_{<n_j} \sqsubseteq \bot, & & A_{>n_i} \sqcap A_{\leq n_i} \sqsubseteq \bot \\
(4) & \top \sqsubseteq A_{\geq n_j} \sqcup A_{<n_j}, & & \top \sqsubseteq A_{>n_i} \sqcup A_{\leq n_i}
\end{array}
$$

where $1 \leq i \leq |N^{\Sigma}| - 1$ and $2 \leq j \leq |N^{\Sigma}|$. Note that axioms with concepts $A_{\leq n}$ and $A_{<n}$ are necessary since the fuzzy ABox might contain fuzzy assertions of the form $(a : A) \leq n$ or $(a : A) < n$. Similarly for each atomic roles we have: $R_{\geq n_{i+1}} \sqsubseteq R_{>n_i}, R_{>n_i} \sqsubseteq R_{\geq n_i}$.

Finally, concept and role axioms in the fuzzy KB should also be reduced [9, 13]. Thus, if $|\mathbf{C}|$ and $|\mathbf{R}|$ is the number of the different atomic concepts and roles in the original fuzzy KB, respectively, and $|N^{\Sigma}|$ the number of different membership degrees that appear in the fuzzy ABox, then the reduced crisp KB would contain as much as $8|\mathbf{C}|(|N^{\Sigma}| - 1)$ concept axioms and $2|\mathbf{R}|(|N^{\Sigma}| - 1)$ role inclusion axioms of this kind of satisfiability-preserving axioms. On the other hand, as shown in [9], there are additionally $4|\mathcal{T}||N^{\Sigma}|$ concept and $2|\mathcal{R}||N^{\Sigma}|$ role inclusion axioms, where $|\mathcal{T}|$ and $|\mathcal{R}|$ is the number of concept and role axioms in the original fuzzy KB. Due to the fact that the number of axioms increases quadratically, the performance of DL reasoners can heavily deteriorate. For this reason, Bobillo et. al. [20] have proposed a number of optimisations. In particular, they note the concept equivalences $A_{\leq n} \equiv \neg A_{>n}$ and $A_{<n} \equiv \neg A_{\geq n}$, thus they map fuzzy assertions of the form $(a : A) \leq n$ and $(a : A) < n$ in the ABox to $a : \neg A_{>n}$ and $a : \neg A_{\geq n}$ respectively. Then, using the laws of contradiction and excluded middle, it can be proved that all axioms of the forms (2) to (4) above are unnecessary. Roughly, speaking this optimisation reduces the number of concept axioms in the reduced TBox to about a quarter compared to the unoptimised one. In our experiments we will in particular also compare the performance of the optimised vs. the unoptimised reduction.

## 3 OWL DL Reasoning with KAON2

Reasoning with KAON2 is based on special-purpose algorithms which have been designed for dealing with large ABoxes. They are described in more detail in [21], such that we only present a birds' eyes perspective here. The underlying rationale of the reasoning procedures is that algorithms for deductive databases have proven to be efficient in dealing with large numbers of facts. The KAON2 approach utilises this by transforming OWL DL ontologies to disjunctive datalog, and by the subsequent application of the mentioned and established algorithms for dealing with disjunctive datalog [22].

A birds' eyes perspective on the KAON2 approach is depicted in Figure 1. KAON2 can handle $\mathcal{SHIQ}(\mathbf{D})$[1] description logic ontologies, which corresponds

[1] $\mathcal{SHIQ}(\mathbf{D})$ provides datatypes and qualified number restrictions in addition to the $\mathcal{SHIN}$ description logic used for our fuzzy reasoning.

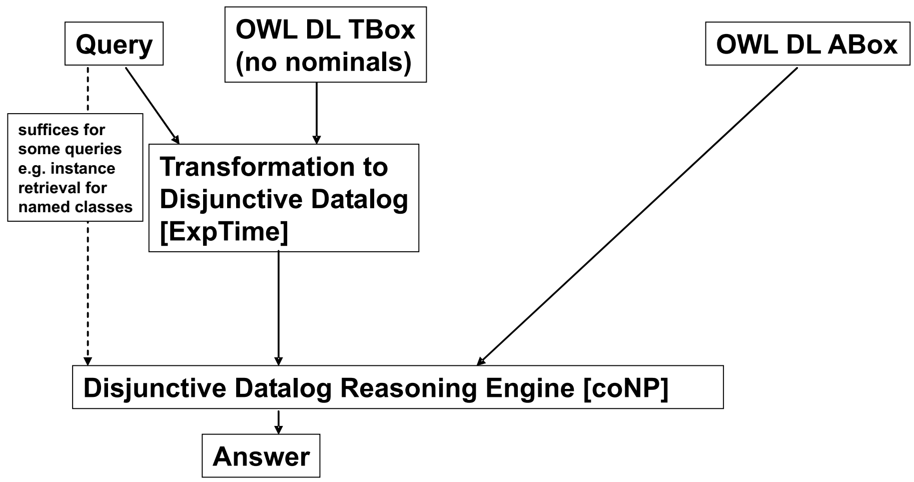


**Fig. 1.** KAON2 approach to reasoning

roughly to OWL DL without nominals. The TBox is processed together with a query by the transformation algorithm, which is described in more detail below and which returns a disjunctive datalog program. This, together with an ABox, is then fed into a disjunctive datalog reasoner which eventually returns an answer to the query. In some cases, e.g. when querying for instances of named classes, the query does not need to be fed into the transformation algorithm but instead needs to be taken into account only by the datalog reasoner.

The transformation algorithm accepts a $\mathcal{SHIQ}$ (or $\mathcal{SHIQ}(\mathbf{D})$) TBox and returns a disjunctive datalog program. Note that the returned program is in general not logically equivalent to the input TBox; the exact relationship is given below in Theorem 1.

The steps of the algorithm can roughly be described as follows. (1) Transitivity axioms for roles $S$ are replaced by adding axioms of the form $\forall R.C \sqsubseteq \forall S.\forall S.C$ whenever $S \sqsubseteq R$. This is a standard method for eliminating transitivity axioms, such that the resulting knowledge base is satisfiable if and only if the original knowledge base is. This ensures that the translation can be used to solve typical $\mathcal{SHIQ}$ reasoning problems by reducing them to unsatisfiability of a $\mathcal{SHIQ}$ knowledge base.

Employing the fact that $\mathcal{SHIQ}$ can be regarded as a subset of first-order logic, step (2) uses standard algorithms to transform the knowledge base into conjunctive normal form. This involves eliminating existential quantifiers by *Skolemization*, such that function symbols must be introduced into the knowledge base.

Next, in step (3), the obtained set of clauses is partially *saturated* by adding logical consequences. This is the crucial step of the algorithm where one has to compute enough consequences to allow for a reduction to function-free Datalog. Since the computational complexity is ExpTime for $\mathcal{SHIQ}$ but only NP for disjunctive Datalog, it should not come as a surprise that this transformation step can be exponential in the size of the input. The details of this step are rather sophisticated such that we refer to [21] for details and proofs.

Now function symbols can safely be eliminated in step (4). To ensure that this process still preserves satisfiability of the knowledge base, one has to add a linear number of auxiliary axioms. Finally, it is easy to syntactically transform the resulting set of clauses into a disjunctive Datalog program in step (5).

Due to the transformations in steps (1) and (2), the output of the algorithm is in general not logically equivalent to the input. Since all major reasoning tasks for $\mathcal{SHIQ}$ can be reduced to satisfiability checking, it is sufficient for the transformation to preserve satisfiability, as shown in the following theorem from [21].

**Theorem 1.** *Let $K$ be a $\mathcal{SHIQ}(\mathbf{D})$ knowledge base and $D(K)$ be the datalog output of the KAON2 transformation algorithm on input $K$. Then the following claims hold.*

- *$K$ is unsatisfiable if and only if $D(K)$ is unsatisfiable.*
- *$K \models \alpha$ if and only if $D(K) \models \alpha$, where $\alpha$ is of the form $A(a)$ or $R(a,b)$, for $A$ a named concept and $R$ a role.*
- *$K \models C(a)$ for a nonatomic concept $C$ if and only if, for $Q$ a new atomic concept, $D(K \cup \{C \sqsubseteq Q\}) \models Q(a)$.*

A performance evaluation of KAON2 is reported in [23]. It shows that KAON2 is indeed superior to other reasoners in most cases where size of the ABox dominates compared to the size of the TBox. This is exactly the reason why we have decided to use KAON2 in the context of our experiments with large A-Boxes.

## 4 Experiments

In this section, we present the results of our experimental evaluation. First of all, we describe the datasets used, i.e. the ontologies, in more detail (see Subsection 4.1). These belong to different complexity classes in order to be able to analyse performance with respect to different classes of ontologies. Most standard benchmark datasets do not focus on average performance (over a larger amount of queries) nor on performance with respect to data size. Thus, we made two important decisions in the context of our experimental evaluation. First, we decided to automatically generate larger A-Boxes ranging from 100 to 1 Mio. individuals. In order not to bias our results by some generation strategy, we apply three different ways of generating A-Boxes of varying sizes for a given ontology. These strategies are described in more detail in Subsection 4.2. Second, instead of relying on a limited number of predefined queries as done in most benchmarks, we decided to also randomly generate a number of $n$ queries of varying complexity. This will then allow us to analyze the performance of our approach with respect to an average conjunctive query.

### 4.1 Datasets

We have chosen some popular ontologies which have been used in previous benchmarks [23]. These are described in what follows:

- **VICODI**: The VICODI ontology[2] was manually created in the EU-funded VICODI project and is a representative of the RDFS(DL) fragment. This ontology is relatively small and simple since it does not contain any disjunctions, existential quantification or number restrictions.
- **LUBM**: The Lehigh University Benchmark (LUBM)[3] was explicitly designed for OWL benchmarks. It describes the organizational structure of universities. Due to existential restriction on the right side of class expressions, the LUBM ontology is in OWL Lite.
- **Semintec**: The Semintec ontology[4] is about financial services and was created at the University of Poznan. It is also relatively simple like the VICODI ontology without existential quantifiers or disjunctions. But it is a representative of the OWL DL fragment as it contains functional properties and disjointness constraints.
- **Wine**: The Wine ontology[5] is a prominent example of an OWL DL ontology. More precisely, we used a version for the Wine ontology without nominals, as it has been used in previous benchmarks (cf. [23]).

### 4.2 Generating A-Boxes

While the ontologies mentioned above already contain some A-Box assertions, for most of them the A-Boxes are too small for our intended performance evaluations. Therefore, for each of the ontologies, we generated different extended datasets with increasing A-Box size to be able to quantify the relationship between query time and the size of the A-Box. In particular, in order not to bias our quantitative analysis, we experimented with three different ways of generating A-Boxes: *random*, *structural* and *proportional*. We briefly describe these different generation strategies, omitting details due to space limitations. For simplicity, from now on, the individuals always include concept individuals and property individuals if we do not specify which kind of individuals.

- **random generation**: according to this strategy, a number $n$ of individuals are generated by first randomly selecting $m$ properties, generating new individuals of these and randomly deciding whether to create a new domain (range) individual or use an existing one. This yields at most $m$ property and $2m$ concept individuals. Then, the rest to $n$ is filled up with random concept individuals.

[2] `http://www.vicodi.org`
[3] `http://swat.cse.lehigh.edu/projects/lubm/index.htm`
[4] `http://www.cs.put.poznan.pl/alawrynowicz/semintec.htm`
[5] `http://www.schemaweb.info/schema/SchemaDetails.aspx?id=62`

– **structural generation**: according to this strategy, the existing A-Box is copied $m$ times until the final number of individuals is reached, whereby concept and property individuals are renamed. This strategy thus aims at preserving the structure of the original ontology.
– **proportional**: according to this strategy, the conditional probability $P(c_j|i_c)$, i.e. the probability that given that $i_c$ is a concept individual it belongs to concept $c_j$ is calculated in the original A-Box. Further, also $P(p_j|i_p)$ is calculated for properties. The relative frequency of property versus concept individuals is also calculated. Then, new individuals are randomly generated according to the above probabilities. This strategy aims at preserving the distribution in the original ontology.

For each of the ontologies mentioned above, we have thus generated ontologies with 100, 1.000, 10.000, 100.000 and 1 Mio. individuals. Further, in order to generate fuzzy ontologies out of these, a random fuzzy degree has been generated for each of the individuals in the different datasets. The ontologies with the fuzzy degree values have then been reduced using the optimised and the unoptimised reduction.

### 4.3 Queries

To generate various sorts of queries for tests, we use 5 query patterns which vary from the simplest one to retrieve all the individuals of a concept to the complex ones such as:

```
SELECT ?V1 ?V2 ?V3 ?V4 ?V5 ?V6
WHERE {?V1 rdf:type C1 . ?V2 rdf:type C2 . ?V3 R1 ?V4 . ?V5 R2 ?V6}
```

where, $Vi$ $(i = 1...6)$ indicates variables. $C1$ and $C2$ means concepts and $R1$ and $R2$ are relations.

When generating a query, we randomly select one query pattern. For each chosen pattern, we replace each variable in the pattern by randomly choosing one from several distinct variables given beforehand. As for each relation (concept) in the pattern, we randomly choose one relation (concept) from the ontology to be queried.

This way of creating queries is obviously flexible enough to generate an arbitrary number of queries of varying complexity. It is important to mention that, as we only use named classes in our queries, the reduction to Datalog does not need to be performed for each query (see Section 3).

### 4.4 First observations: number of axioms

In order to generate ontologies with Fuzzy ABoxes, we randomly generated fuzzy degrees for the A-Boxes generated as described above. After applying the reduction, we obtained ontologies with a number of axioms (TBox and RBox) as reported in Table 2. It shows the number of axioms for the original ontologies as well as those resulting from the reduction to crisp OWL-DL for different degrees, i.e. 3 (0, 0.5 1), 6 (0, 0.2, 0.4, 0.6, 0.8, 1) and 11 (0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6,

| Ontology | # Axioms | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Original | | | | | | | | | Reduced Ontologies | | | | | |
| | | | | | | | | | | Unoptimized | | | Optimized | | |
| | Class | Prop. | TBox | Subcl. | Disj. | Subpr. | Func. | Dom. | Ran. | 3 | 6 | 11 | 3 | 6 | 11 |
| VICODI | 194 | 10 | 223 | 193 | 0 | 9 | 10 | 10 | 0 | 5.110 | 12.196 | 24.006 | 1.624 | 4.872 | 8.120 |
| LUBM | 43 | 25 | 94 | 36 | 0 | 5 | 25 | 18 | 0 | 1.248 | 2.994 | 5.904 | 480 | 1.440 | 2.400 |
| Semintec | 60 | 16 | 222 | 55 | 113 | 0 | 16 | 16 | 12 | 1.808 | 4.016 | 7.696 | 770 | 2.310 | 3.850 |
| Wine | 141 | 13 | 218 | 126 | 1 | 5 | 6 | 9 | 6 | 4.204 | 9.946 | 19.516 | 1.402 | 4.206 | 7.010 |

**Table 2.** Number of axioms for the original ontologies and after the reduction

0.7, 0.8, 0.9, 1). We can certainly see that the reduction increases the number of axioms dramatically. Interestingly, the optimized reduction produces about a third of the axioms produced by the unoptimized method.

### 4.5 Results

We have carried out performance evaluation tests for the above mentioned four ontologies comparing the query performance for standard query answering with respect to the original ontology as well as fuzzy query answering with respect to the fuzzy ontologies after the reduction using the optimised and unoptimised methods. The fuzzy queries are also randomly generated using the patterns described above, inserting the concepts and relations produced by the reduction, thus yielding "fuzzy" conjunctive queries which are actually crisp queries with respect to the transformed ontologies.

Figure 2 shows the average time for query answering with respect to the automatically generated datasets for the crisp ontologies containing 100 to 1. Mio individuals for the different generation strategies. A first interesting result is that the different generation strategies do not have a major effect on the query time. For this reason, we decided to carry out the remaining experiments using only the proportional generation strategy.

In particular, we tested ontologies with 3, 6 and 11 fuzzy degrees. For all queries, we imposed a time limit of 20 minutes. After that, query evaluation was stopped and counted as 20 minutes. All results are reported averaged over 100 queries. Our first conclusions were the following:

- The unoptimised reduction leads to ontologies which are completely intractable, such that almost none of the queries can be answered within 20 minutes.
- The Wine ontology showed to be completely intractable both with the optimised and unoptimised reduction even using only 3 degrees! This shows that highly expressive (fuzzy) ontologies lead to a high number of axioms when they are reduced to crisp ontologies, thus becoming intractable for state-of-the-art reasoners. Thus, we do not show the detailed results for the Wine ontology from now on.
- Using the optimised reduction, the query time with respect to the original crisp ontologies increases a factor of between 2 and 100 depending on the

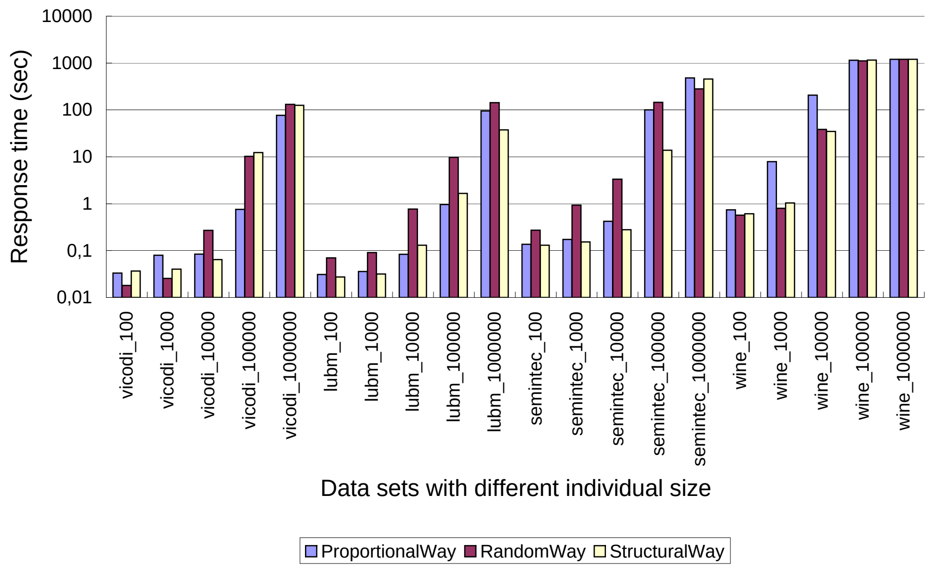


**Fig. 2.** Average query time for the four (crisp) ontologies (VICODI, LUBM, Semintec and OWL) for the three A-Box generation strategies: random, structural and proportional

number of degrees used. We discuss this increase for the ontologies transformed using the optimised reduction in more detail below with respect to the different ontologies.

Figure 3 shows the average querying time for the three ontologies (VICODI, LUBM and Semintec) transformed with the optimised reduction using 3, 6 and 11 degrees. We discuss the results separately for each ontology:

- **VICODI** (RDFS-DL): The results are as expected as there is a clear increase of query time from the original ontology to the transformed ontology with 11 degrees. For 3 degrees, there seems to be an increase of factor 2 of the querying time, while for 6 degrees the increase corresponds to a factor of 10. The increase for 11 degrees is a factor between 10 and 100 (for the dataset with 100.000 individuals).
- **LUBM** (OWL Lite): For LUBM, the results are also more or less as expected. The increase is quite consistently a factor of 10 with the only exception of the dataset with 10.000 individuals
- **Semintec** (OWL-DL): The Semintec ontology shows a similar pattern as the other ontologies: using 3 degrees leads to an increase of query time with a factor of between 1-2; while using 6 degrees and 11 degrees leads to an increase within a factor of around 10 and 100, respectively.

Overall, we can conclude that the increase in the time for answering a query is considerable when considering the transformed ontologies, i.e. roughly a factor

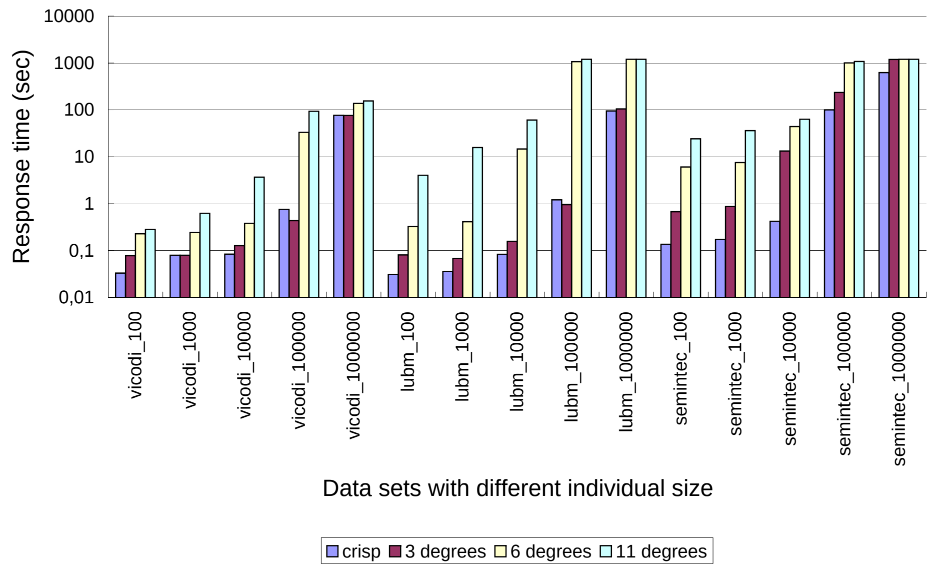


**Fig. 3.** Average query time for the three ontologies (VICODI, LUBM and Semintec) transformed with the optimised reduction using 3, 6 and 11 degrees

between 2 and 100 compared to query answering with respect to the original crisp ontologies. The pattern which emerges is quite clear. For the reduced ontologies with 3 degrees, the query time seems to double compared to standard query answering with respect to the original ontologies. When using 6-11 degrees, the increase is in most cases a factor of at least 10 compared to the time for answering queries with respect to the original crisp ontology. For 11 degrees, the increase can even reach a factor of 100 (e.g. for VICODI with 100.000 individuals), but in most cases there is an increase with a factor of 10. The good news is certainly that for smaller ontologies from 100 to 10.000 individuals, the query time is at most a few (<4) seconds regardless of the number of degrees used. This shows that for smaller ontologies the approach considered is definitely feasible regardless of the degrees considered. For bigger ontologies, using more degrees (6-11) comes at a high price. Thus, applications will clearly need to balance the amount of degrees they need to distinguish with the maximum querying time they can afford. In any case, it seems that fuzzy query answering is feasible in real time (where real time is defined as under a few seconds) in exactly those cases in which query answering with respect to the original crisp ontology is also. In all those cases where fuzzy query answering is not feasible in real time (let's say > 100 seconds), neither standard query answering with respect to the crisp ontologies is. In such a case, one could argue, it is not anymore important whether the queries can be answered in 100 seconds (1.6 minutes) or 1000 seconds (16 minutes) as these times are both clearly beyond a real-time behaviour.

## 5 Related Work and Conclusion

In the last couple of years it has become evident that fuzzy Description Logics could play an important role in several Semantic Web applications that face a significant amount of vague knowledge, like multimedia analysis and retrieval. In order to support inference services for fuzzy-DLs, several different reasoning algorithms have been proposed such as tableaux-based [12, 10], optimization-based [16, 17] as well as techniques that reduce fuzzy-DL reasoning to crisp DL reasoning [9, 13, 20].

Although there has been a significant amount of work on the reduction based DL reasoning methods, there is currently no systematic and complete evaluation of the method[6]. In particular, Straccia was the first to propose the method, showing that reasoning in $f_{KD}$-$\mathcal{ALCH}$ can be reduced to reasoning in $\mathcal{ALCH}$, while later Bobillo et. al. [13] extended the method to reduce $f_{KD}$-$\mathcal{SHOIN}$ (together with *fuzzy subsumption axioms* [6]) to $\mathcal{SHOIN}$. Then Bobillo [20] extended the technique to $\mathcal{SROIQ}$ also proposing a number of optimisation methods that reduced the number of created axioms. Additionally, in [20] there is a first report on an implementation accompanied with a preliminary evaluation. More precisely, the authors have implemented their proposed optimised reduction technique which then they evaluated over a fuzzified version of the Koala ontology[7]. The Koala ontology contains 20 named classes, 15 anonymous classes, 4 object properties, 1 datatype property and 6 individuals. The axioms of the ontology where randomly extended with membership degrees, using 3, 5, 7, 9 and 11 different degrees, thus creating a set of five fuzzy ontologies. Then, the reduction method was evaluated using the Pellet reasoner providing the times for knowledge base satisfiability checking. In contrast, we have mainly focused on the task of query answering. Instead of evaluating with respect to only one small ontology, we have tested four different ontologies corresponding to different complexity classes (i.e. the RDFS DL fragment, OWL Lite and OWL-DL) with respect to different A-Box sizes of up to a million of individuals. Our focus has in fact been to test the scalability of the reduction-based approach depending on A-Box size. This is also the reason why we have used KAON2 as reasoner, which was particularly developed to handle large A-Boxes. Our experimental evaluation has shown that for quite expressive ontologies, especially the Wine ontology, the approach based on the reduction to crisp DL is not feasible for fuzzy query answering. Arguably, the wine ontology was never conceived as a realistic ontology for A-Box reasoning but rather for T-Box reasoning. In any case, in our experiments the ontologies resulting from the reduction to crisp DL where only tractable when using the optimised reduction, which reduced the number of axioms by around 66% compared to the unoptimised reduction.

Overall, the increase in query time is moderate for small numbers of degrees (e.g. 3-6). Distinguishing 3-6 degrees leads to increased query times with a factor

[6] Note that the landscape is not different with the other reasoning methods for fuzzy-DLs

[7] `http://http://protege.cim3.net/file/pub/ontologies/koala/koala.owl`

between 2-10 for smaller A-Boxes (up to 10.000 individuals) and to a factor of 100 for larger A-Boxes with around 100.000 individuals and more. The interesting conclusion is nevertheless that for rather inexpressive ontologies such as VICODI and LUBM, query answering with the approach analyzed here is definitely feasible for a few degrees (3-6) even with very large A-Boxes as the time increase is below a factor of 10. For applications which really need fuzzy query answering, this increase might be acceptable. Finally, the encouraging result is that for smaller A-Boxes with a number of individuals between 100 and 10.000 query answering takes below a few seconds irrespectively of the degrees used. The interesting conclusion that real time query answering seems to be feasible for the fuzzy ontologies exactly for those A-Box sizes for which it is also feasible for the crisp ontologies is an encouraging result from our point of view.

There are obvious avenues for future work. First, it would be appropriate to experiment with other reasoners such as FaCT or Pellet for reasoning with the reduced ontologies. Further, a necessary next step is to compare with respect to other techniques such as tableaux-based reasoners such as Fire (see [24]). While our envisioned application is retrieval and thus querying, the investigation of several issues from the application point of view seem interesting. First of all, we expect that, in contrast to what we have assumed in this paper, not all concepts in an ontology should be considered as inherently fuzzy. In fact, we expect applications to pose clear requirements on what concepts should be understood as fuzzy and which not. By this, the large amount of axioms produced by the reduction could be dramatically constrained, thus leading to a better performance. In this context it will be necessary to clarify the interactions between fuzzy and non-fuzzy concepts in query answering. Overall, we think that this paper offers a clear step forward towards clarifying whether techniques for reasoning with fuzzy knowledge can be expected to scale up.

## References


1. Meghini, C., Sebastiani, F., Straccia, U.: A model of multimedia information retrieval. Journal of the ACM **48**(5) (2001) 909–970
2. Zadeh, L.: The concept of a linguistic variable and its application to approximate reasoning. Information Sciences **8-9** (1975)
3. Ding, Z., Peng, Y., Pan, R. Studies in Fuzziness and Soft Computing. In: BayesOWL: Uncertainty Modeling in Semantic Web Ontologies. Springer-Verlag (2005) 27
4. da Costa, P., Laskey, K.: PR-OWL: A framework for probabilistic ontologies. In: Proceedings of the International Conference on Formal Ontology in Information Systems. (2006)
5. Udrea, O., Subrahmanian, V.S., Majkic, Z.: Probabilistic rdf. In: Proc. of IRI'06. (2006) 172–177
6. Straccia, U.: Towards a fuzzy description logic for the semantic web. In: Proceedings of the 2nd European Semantic Web Conference. (2005)
7. Stoilos, G., Stamou, G., Tzouvaras, V., Pan, J., Horrocks, I.: Fuzzy OWL: Uncertainty and the semantic web. In: Proc. of the International Workshop on OWL: Experiences and Directions. (2005)

8. Lukasiewicz, T., Straccia, U.: An overview of uncertainty and vagueness in description logics for the semantic web. Technical Report INFSYS Research Report 1843-06-07, Technische Universität Wien (2006)
9. Straccia, U.: 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, Lisbon, Portugal, Springer Verlag (2004) 385–399
10. Stoilos, G., Stamou, G., Tzouvaras, V., Pan, J.Z., Horrocks, I.: Reasoning with very expressive fuzzy description logics. Journal of Artificial Intelligence Research **30**(8) (2007) 273–320
11. Baader, F., McGuinness, D., Nardi, D., Patel-Schneider, P.: The Description Logic Handbook: Theory, implementation and applications. Cambridge University Press (2002)
12. Straccia, U.: Reasoning within fuzzy description logics. Journal of Artificial Intelligence Research **14** (2001) 137–166
13. Bobillo, F., Delgado, M., Gómez-Romero, J.: A crisp representation for fuzzy $\mathcal{SHOIN}$ with fuzzy nominals and general concept inclusions. In: Proc. of the 2nd International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 06), Athens, Georgia. (2006)
14. Straccia, U.: Answering vague queries in fuzzy DL-Lite. In: Proceedings of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, (IPMU-06). (2006) 2238–2245
15. Pan, J., Stamou, G., Stoilos, G., Thomas, E.: Expressive querying over fuzzy DL-Lite ontologies. In: Proceedings of the International Workshop on Description Logics (DL 2007). (2007)
16. Straccia, U.: Description logics with fuzzy concrete domains. In: 21st Conf. on Uncertainty in Artificial Intelligence (UAI-05), Edinburgh (2005)
17. Bobillo, F., Straccia, U.: A fuzzy description logic with product t-norm. In: Proceedings of the IEEE International Conference on Fuzzy Systems (Fuzz-IEEE-07), London. (2007)
18. Horrocks, I.: Using an Expressive Description Logic: FaCT or Fiction? In: Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98), Trento, Italy, June 2-5, 1998, Morgan Kaufmann (1998) 636–649
19. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. Journal of Web Semantics **5** (2007) 51–53
20. Bobillo, F., Delgado, M., Gómez-Romero, J.: Optimising the crisp representation of the fuzzy dl $\mathcal{SROIQ}$. In: Proc. of the 3nd International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 07), Busan, Korea. (2007)
21. Motik, B.: Reasoning in Description Logics using Resolution and Deductive Databases. PhD thesis, Universität Karlsruhe (2006)
22. Cumbo, C., Faber, W., Greco, G., Leone, N.: Enhancing the magic-set method for disjunctive datalog programs. In Demoen, B., Lifschitz, V., eds.: ICLP. Volume 3132 of Lecture Notes in Computer Science., Springer (2004) 371–385
23. Motik, B., Sattler, U.: A comparison of reasoning techniques for querying large description logic ABoxes. In: Proceedings of the 13th International Conference on Logic for Programming Artificial Intelligence and Reasoning (LPAR 2006), Phnom Penh, Cambodia, November, 2006. (2006)
24. Simou, N., Kollias, S.: Fire : A fuzzy reasoning engine for impecise knowledge, K-Space PhD Students Workshop, Berlin, Germany, 14 September 2007 (2007)