Approximating Description Logic Classification for Semantic Web Reasoning

Perry Groot[1], Heiner Stuckenschmidt[2], and Holger Wache[2]

[1] Radboud University Nijmegen, Toernooiveld 1,
6500GL Nijmegen, The Netherlands
Perry.Groot@science.ru.nl,
[2] Vrije Universiteit Amsterdam, de Boelelaan 1081a,
1081HV Amsterdam, The Netherlands
{holger,heiner}@cs.vu.nl.

Abstract. In many application scenarios, the use of the Web ontology language OWL is hampered by the complexity of the underlying logic that makes reasoning in OWL intractable in the worst case. In this paper, we address the question whether approximation techniques known from the knowledge representation literature can help to simplify OWL reasoning. In particular, we carry out experiments with approximate deduction techniques on the problem of classifying new concept expressions into an existing OWL ontology using existing Ontologies on the web. Our experiments show that a direct application of approximate deduction techniques as proposed in the literature in most cases does not lead to an improvement and that these methods also suffer from some fundamental problems.

1 Introduction and Motivation

A strength of the current proposals for the foundational languages of the Semantic Web is that they are all based on formal logic. This makes it possible to formally reason about information and derive implicit knowledge. However, this reliance on logics is not only a strength but also a weakness. Traditionally, logic has always aimed at modelling idealised forms of reasoning under idealised circumstances. Clearly, this is not what is required under the practical circumstances of the Semantic Web. Instead, the following are all needed:

- reasoning under time-pressure
- reasoning with other limited resources besides time
- reasoning that is not 'perfect' but instead 'good enough' for given tasks under given circumstances

It is tempting to conclude that symbolic, formal logic fails on all these counts, and to abandon that paradigm. Our aim is to keep the advantages of formal logic in terms of definitional rigour and reasoning possibilities, but at the same time address the needs of the Semantic Web.

Research in the past few years has developed methods with the above properties while staying within the framework of symbolic, formal logic. However, many of those

previously developed methods have never been considered in the context of the Semantic Web. Some of them have only been considered for some very simple underlying description languages [1]. As the languages proposed for modelling Ontologies in the Semantic Web are becoming more and more complex, it is an open question whether those approximation methods are able to meet the practical demands of the Semantic Web. In this article, we look at approximation methods for Description Logics (DLs), which are closely related to some of the currently proposed Semantic Web languages, e.g., OWL.

The remainder of the article is structured as follows. Section 2 gives an overview of various approximation approaches and techniques useful in the context of the Semantic Web. Thereafter, the article focuses on the investigation of a particular approximation technique in the context of a particular reasoning task. Section 3 describes the approximation approach used. Section 4 describes the reasoning task focused on. Section 5 gives experimental results of the approximation method applied to the classification of concepts in a number of Ontologies. Section 6 gives conclusions, discusses the results and the applicability of the analysed approximation approach to the Semantic Web.

2 Approximation Approaches

A typical architecture for a KR system based on DLs can be sketched as in Figure 1 [2], which contains three components that can be approximated to obtain a simplified system that is more robust and more scalable. These components are: (1) the underlying description language, (2) the knowledge base, and (3) the reasoner. The knowledge base itself comprises two components (TBox and ABox), which can also be approximated as a whole or separately. Some general approximation techniques that can be applied to one or more of these components are the following:

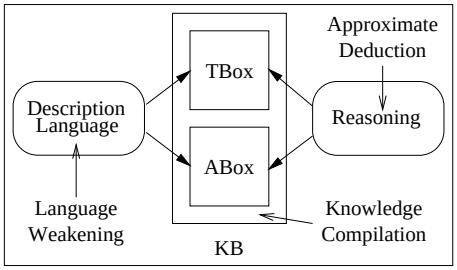

Fig. 1. Architecture of a KR system based on Description Logic together with possible approximation approaches.

Language Weakening: The idea of language weakening is based on the well-known trade-off between the expressiveness and the reasoning complexity of a logical language. By weakening the logical language in which a theory is encoded, we are able to trade the completeness of reasoning against run-time. For example, [3] shows how hierarchical knowledge bases can be used to reason approximately with disjunctive information. The logic that underlies OWL Full for example is known to be intractable, reasoners can use a slightly weaker logic (e.g., OWL Lite) that still allows the computation of certain consequences. This idea can be further extended by starting with a very simple language and iterating over logics of increasing strength supplementing previously derived facts.

Knowledge Compilation: In order to avoid complexity at run-time, knowledge compilation aims at pre-processing the ontology off-line such that on-line reasoning becomes faster. For example, this can be achieved by explicating hidden knowledge. Derived facts are added to the original theory as axioms, avoiding the need to deduce them again. In the case of ontological reasoning, implicit subsumption and membership relations are good candidates for compilation. For example, implicit subsumption relations in an OWL ontology could be identified using a DL reasoner, the resulting more complete hierarchy could be encoded e.g., in RDF schema and used by systems that do not have the ability to perform complex reasoning. This example can be considered to be a transformation of the DL language. When one transforms an ontology into a less expressive DL language [4, 5], this often results in an approximation of the original ontology.

Approximate Deduction: Instead of modifying the logical language, approximations can also be achieved by weakening the notion of logical consequence [1, 6]. The approximated consequences are usually characterised as sound but incomplete, or complete but unsound. Only [1] has made some effort in the context of DLs.

The approximation method focused on in this article belongs to the last category, however there is not always a clear classification of one method to the three categories defined above. In the following section we discuss in more detail what is meant by approximating DLs in the remainder of this paper.

3 Approximating Description Logics

The elements of a DL are concept expressions and determining their satisfiability is the most basic task. Other reasoning services (e.g., subsumption, classification, instance retrieval) can often be restated in terms of satisfiability checking [2]. With approximation in DLs, we mean determining the satisfiability of a concept expression through some other means than computing the satisfiability of the concept expression itself. This use of approximation differs with other work on approximating DLs [4, 5] in which a concept expression is translated to another concept expression, defined in a second typically less expressive DL.

In our approach (originally proposed in [1]), in a DL only other, somehow 'related', concept expressions can be used that are in some way 'simpler' when determining their satisfiability. For example, a concept expression can be related to another concept expression through its subsumption relation, and a concept expression can be made simpler by either forgetting some of its subconcepts or by replacing some of its subconcepts with simpler concepts. In particular, there are two ways that a concept expression C can be approximated by a related simpler concept expression D. Either the concept expression C is approximated by a weaker concept expression D (i.e., less specific, $C \sqsubseteq D$) or by a stronger concept expression D (i.e., more specific, $D \sqsubseteq C$). When $C \sqsubseteq D$, unsatisfiability of D implies unsatisfiability of C. When $D \sqsubseteq C$, satisfiability of D implies satisfiability of C. Note that this is similar to set theory. For two sets C, D, when $C \subseteq D$ holds, emptiness of D implies emptiness of C, and when $D \subseteq C$ holds, non-emptiness of D implies non-emptiness of C.

In [1] Cadoli and Schaerf propose a syntactic manipulation of concept expressions that simplifies the task of checking their satisfiability. The method generates two sequences of approximations, one sequence containing weaker concepts and one sequence containing stronger concepts. The sequences of approximations are obtained by substituting a substring D in a concept expression C by a simpler concept.

More precisely, for every substring D they define the depth of D to be 'the number of universal quantifiers occurring in C and having D in its scope' [1]. The scope of $\forall R.\phi$ is ϕ which can be any concept term containing D. Using the definition of depth a sequence of weaker approximated concepts can be defined, denoted by $C_i^{\top}$, by replacing every existentially quantified subconcept, i.e., $\exists R.\phi$ where ϕ is any concept term, of depth greater or equal than i by $\top$. Analogously, a sequence of stronger approximated concepts can be defined, denoted by $C_i^{\bot}$, by replacing every existentially quantified subconcept of depth greater or equal than i by $\bot$. The concept expressions are assumed to be in negated normal form (NNF) before approximating them. These definitions lead to the following result:

Theorem 1. *For each i, if $C_i^{\top}$ is unsatisfiable then $C_j^{\top}$ is unsatisfiable for all $j \geq i$, hence C is unsatisfiable. For each i, if $C_i^{\bot}$ is satisfiable then $C_j^{\bot}$ is satisfiable for all $j \geq i$, hence C is satisfiable.*

These definitions are illustrated by the following concept expression in NNF taken from the Wine ontology[3]

$$\texttt{Merlot} \equiv \texttt{Wine} \sqcap \leq 1\,\text{madeFromGrape}.\top \sqcap \exists\,\text{madeFromGrape}.\{\mathit{MerlotGrape}\},$$

which states that a Merlot wine is a wine that is made from the Merlot grape and no other grape. This concept expression contains no $\forall$-quantifiers. Therefore the depth of the only existentially quantified subconcept '$\exists$madeFromGrape.$\{\mathit{MerlotGrape}\}$' is 0. Substituting either $\top$ or $\bot$ leads to the following approximations for level 0:

$$\texttt{Merlot}_0^{\top} \equiv \texttt{Wine} \sqcap (\leq 1\,\text{madeFromGrape}.\top) \sqcap \top,$$

$$\texttt{Merlot}_0^{\bot} \equiv \texttt{Wine} \sqcap (\leq 1\,\text{madeFromGrape}.\top) \sqcap \bot.$$

No subconcepts of level 1 appear in the concept expression for `Merlot`. Therefore, $\texttt{Merlot}_1^{\top}$ and $\texttt{Merlot}_1^{\bot}$ are equivalent to `Merlot`. The nesting of existential and universal quantifiers is an important measure of the complexity of satisfiability checking when considered from a worst case complexity perspective [8]. This is a motivation for Cadoli and Schaef to make their specific substitution choices. Furthermore, they are able to show a relation between $C_i^{\top}$- and $C_i^{\bot}$-approximation and their multi-valued logic based on S-1- and S-3-interpretations [1]. Therefore, properties obtained for S-1- and S-3-approximation also hold for $C_i^{\top}$- and $C_i^{\bot}$-approximation. These properties include the following: (1) Semantically well founded, i.e., there is a relation with a logic that can be used to give meaning to approximate answers; (2) Computationally attractive, i.e., approximate answers are cheaper to compute than the original problem; (3) Duality, i.e., both sound but incomplete and complete but unsound approximations can be

[3] A wine and food ontology which forms part of the OWL test suite [7].

constructed; (4) Improvable, i.e., approximate answers can be improved while reusing previous computations; (5) Flexible, i.e., the method can be applied to various problem domains. These properties were identified by Cadoli and Schaerf to be necessary for any approximation method.

Although the proposed method by Cadoli and Schaerf [1] satisfies the needs of the Semantic Web identified in Section 1 in theory, little is known about the applicability of their method to practical problem solving. Few results have been obtained for S-1- and S-3-approximation when applied to propositional logic [9–11], but no results are currently known to the authors when their proposed method is applied to DLs. Current work focuses on empirical validation of their proposed method. Furthermore, DLs have changed considerably in the last decade. Cadoli and Schaerf proposed their method for approximating the language $\mathcal{ALE}$ (they also give an extension for $\mathcal{ALC}$), but $\mathcal{ALE}$ has a much weaker expressivity then the languages that are currently proposed for ontology modeling on the Semantic Web such as OWL. The applicability of their method to a more expressive language like OWL is an open question. Current work takes the method of Cadoli and Schaerf as a basis and focuses on extending it to more expressive DLs.

4 Approximating Classification

The problem of classification is to arrange a complex concept expression into the subsumption hierarchy of a given TBox. We choose this task for two reasons. First, the worst-case complexity of a classification algorithm for expressive representation languages like OWL-DL is known to be intractable. Efficient alternatives have only been proposed for subsets of DLs [12].

Second, classification is a very important part of many other reasoning services and applications. For example, classification is used to generate the subsumption hierarchy of the concept descriptions in an ontology. Furthermore, classification is used in the task of retrieving instances. From a theoretical point of view, checking whether an instance i is member of a concept Q can be done by proving the unsatisfiability of $\neg Q(i)$. Doing this for all existing instances, however, is intractable. Therefore, most DL systems use a process that reduces the number of instance checks. It is assumed that the ontology is classified and all instances are assigned to the most specific concept they belong to. Instance retrieval is then done by first classifying the query concept Q in the subsumption hierarchy and then selecting the instances of all successors of Q and of all direct predecessors of Q that pass the membership test in Q. We conclude that there is a lot of potential for approximating the classification task.

In the following, we first describe the process of classification in DL systems. Afterwards we explain how the approximation technique introduced in Section 3 can be used to approximate (part of) this problem.

For classifying a concept expression Q into the concept hierarchy (Algorithm 1) a number of subsumption tests are required for comparing the query concept with other concepts C_i in the hierarchy. As the classification hierarchy is assumed to be known, the number of subsumption tests can be reduced by starting at the highest level of the hierarchy and to move down to the children of a concept only if the subsumption test is positive. The most specific concepts w.r.t. the subsumption hierarchy which passed the

Algorithm 1 classification

Require: A classified concept hierarchy with root *Root*
Require: A query concept *Q*
  VISITED := $\varnothing$
  RESULT := $\varnothing$
  GOALS := $\{\top\}$
  while Goals $\neq \varnothing$ **do**
    C $\in$ Goals *where* {*direct parents of* C} $\subseteq$ Visited
    GOALS := Goals $\setminus$ {C}
    VISITED := Visited $\cup$ {C}
    if subsumed-by(Q,C) **then**
      GOALS := Goals $\cup$ {*direct children of* C}
      RESULT := (Result $\cup$ {C}) $\setminus$ {*all ancestors of* C}
    end if
  end while
  if |Result| = 1 $\wedge$ subsumed-by(C,Q) **then**
    EQUAL := 'yes'
  else
    EQUAL := 'no'
  end if
  return Equal, Result

subsumption test are collected for the results. At the end of the algorithm, we check if the result is subsumed by Q as this implies that both are equal.

Algorithm 1 contains more than one step that can be approximated. For example, the subsumption tests, represented by `subsumed-by(X,Y)` in the algorithm, can be approximated using the method of Cadoli and Schaerf.

The subsumption test $Q \sqsubseteq C$ can be reformulated into the unsatisfiability test of $Q \sqcap \neg C$ (Algorithm 2). The idea is to replace standard subsumption checks by a series of approximate checks of increasing exactness. In particular, we use weaker approximations $C_i^{\top}$ in the $C^{\top}$`-subsumption` algorithm (Algorithm 3) and stronger approximations $C_i^{\bot}$ in the $C^{\bot}$`-subsumption` algorithm (Algorithm 4). (Note the difference between Algorithm 2 and Algorithms 3 and 4.) The approximations are easily constructed in a linear way. When the approximation at a certain level I does not lead to a conclusion (based on Theorem 1) the level I is increased by one. This is repeated until the original concept expression is obtained, i.e., the exact subsumption test has to be performed. Algorithm 5 integrates both approximations in one procedure. The approximate versions,

Algorithm 2 subsumption

Require: A complex concept expression *C*
Require: A Query *Q*
  CURRENT := $Q \sqcap \neg C$
  RESULT := unsatisfiable(Current)
  return Result

Algorithm 3 $C^{\top}$-subsumption

Require: A complex concept expression C
Require: A Query Q
  I := 0
  repeat
    CURRENT := $(Q \sqcap \neg C)_I^{\top}$
    RESULT := unsatisfiable(Current)
    if Result = 'true' **then**
      break
    end if
    I := I+1
  until Current = $Q \sqcap \neg C$
  return Result

Algorithm 4 $C^{\bot}$-subsumption

Require: A complex concept expression C
Require: A Query Q
  I := 0
  repeat
    CURRENT := $(Q \sqcap \neg C)_I^{\bot}$
    RESULT := unsatisfiable(Current)
    if Result = 'false' **then**
      break
    end if
    I := I+1
  until Current = $Q \sqcap \neg C$
  return Result

Algorithm 5 $C_I^{\bot}$-$C_I^{\top}$-subsumption

Require: A complex concept expression C
Require: A Query Q
  I := 0
  repeat
    CURRENT := $(Q \sqcap \neg C)_I^{\bot}$
    RESULT := unsatisfiable(Current)
    if Result = 'false' **then**
      break
    end if
    CURRENT := $(Q \sqcap \neg C)_I^{\top}$
    RESULT := unsatisfiable(Current)
    if Result = 'true' **then**
      break
    end if
    I := I+1
  until Current = $Q \sqcap \neg C$
  return Result

i.e., $C^{\top}$ `-subsumption`, $C^{\bot}$ `-subsumption`, and $C_I^{\bot}$ -$C_I^{\top}$ `-subsumption` will replace the method `subsumed-by` in Algorithm 1 in the forthcoming experiments.

While these approximate versions can in principle be applied to all occurrences of subsumption tests, we restricted the use of approximations to the first part of the algorithm where the query concept is classified into the hierarchy.

Each DL reasoner (e.g., Fact [13], Racer [14, 15]) implements the classification functionality internally. In order to obtain comparable statements about approximate classification, independently from the implementation of a particular DL reasoner, which may use highly optimised heuristics, we implement our own and independent classification method. The classification procedure was built on top of an arbitrary DL reasoner according to Algorithm 1, which can call the various approximation forms stated in Algorithms 3, 4, and 5 The satisfiability tests are propagated to the DL reasoner through the DIG interface [16] as depicted in Figure 2.

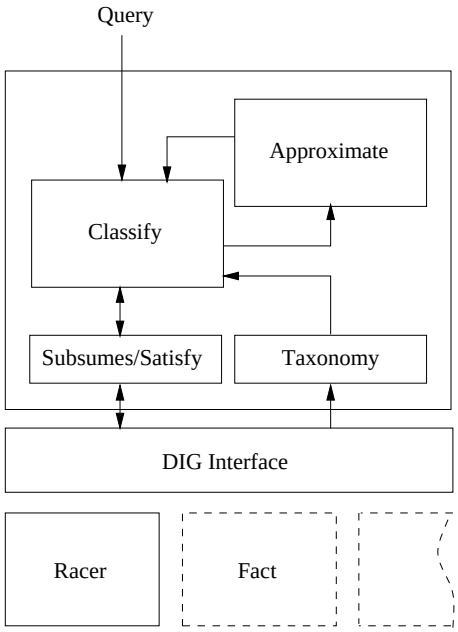

Fig. 2. Architecture of experimental setup.

5 Experiments

The main question focused on in the experiments is which form of approximation, i.e., $C_i^{\top}$, $C_i^{\bot}$, or their combination, can be used to reduce the complexity of the classification task. The focus of the experiments will not be on the overall computation time, but on the number of operations needed. The goal of approximation is to replace costly reasoning operations by a (small) number of cheaper approximate reasoning operations. The suitability of the method of Cadoli and Schaerf therefore depends on the number of classical reasoning operations that can be replaced by their approximate counterparts without changing the result of the computation.

In the experiments queries are generated automatically. The system randomly selects a number of concept descriptions from the loaded ontology. These definitions are used as queries and are reclassified into the subsumption hierarchy. Note that the queries are first randomly selected, then they are used in the experiments with all forms of approximation.

The first experiments were made with the TAMBIS ontology in which we (re)classified 16 unfoldable concept definitions.[4] Only the approximation method originally suggested by Cadoli and Schaerf [1] for $\mathcal{ALE}$ (described in Section 3) was used.

[4] A biochemistry ontology developed in the TAMBIS project [17].

The results of the first experiments are shown in Table 1, which is divided into four columns. Each column reports the number of subsumption tests when using a certain form of approximation. The first column reports results for the experiment with normal classification (i.e., without approximation), the second column for $C_i^{\perp}$-approximation, the third column for $C_i^{\top}$-approximation, and the fourth column for a combination of $C_i^{\perp}$- and $C_i^{\top}$-approximation.

Each column of Table 1 is divided into a number of smaller rows and columns. The rows represent the level of the approximation used, where N denotes normal subsumption testing, i.e., without approximation. The columns represent whether the subsumption test resulted in true or false.[5] This distinction is important, because Theorem 1 tells us that only one of those two results will immediately lead to a reduction in complexity, while for the other result approximation has to continue at the next level. This continues until no more approximation steps can be done.

Table 1. Subsumption tests for the reclassification of 16 concepts in TAMBIS.

	normal			$C_i^{\perp}$			$C_i^{\top}$			$C_i^{\perp}$&$C_i^{\top}$		
		true	**false**		**true**	**false**		**true**	**false**		**true**	**false**
Tambis (16)				$C_0^{\perp}$	157	32	$C_0^{\top}$	8	181	$C_0^{\perp}$	157	32
				$C_1^{\perp}$	0	0	$C_1^{\top}$	0	0	$C_0^{\top}$	8	149
	N	24	279	N	24	247	N	16	279	N	16	247

The first column shows that for the reclassification of 16 concepts in the TAMBIS ontology, 24 true subsumption tests and 279 false subsumption tests were needed.

The second column shows that $C_i^{\perp}$-approximation leads to a change in normal subsumption tests. Compared to the normal case, the number of false subsumption tests are reduced from 279 to 247. However, the 24 true subsumption tests are not reduced. Note that 32 (279 - 247) false subsumption tests are replaced by 157 true $C_0^{\perp}$-subsumption tests and 32 false $C_0^{\perp}$-subsumption tests.[6]

The third column shows that $C_i^{\top}$-approximation also leads to a change in normal subsumption tests, but quite different when compared to $C_i^{\perp}$-approximation. With $C_i^{\top}$-approximation we reduce the true subsumption tests from 24 to 16. However, the 279 false subsumption tests are not reduced. Note that 8 (24 - 16) true subsumption tests are replaced by 8 true $C_0^{\top}$-subsumption tests and 181 false $C_0^{\top}$-subsumption tests. Analogously to $C_i^{\perp}$-approximation, no $C_i^{\top}$-approximation was used when this would not lead to a change in the subsumption expression.

The fourth column shows the combination of $C_i^{\perp}$- and $C_i^{\top}$-approximation by using the approximation sequence $C_0^{\perp}, C_0^{\top}, C_1^{\perp}, C_1^{\top}, ..., C_{n-1}^{\perp}, C_{n-1}^{\top}, \mathit{normal}$. This combination

[5] We will use the shorthand 'true subsumption test' and 'false subsumption test' to indicate these two distinct results.

[6] Note that the numbers do not add up. The reason for this is that approximation is not used when there is no change in the subsumption expression after approximation, i.e., when $C_i^{\perp} = C$ the DL reasoner is not called and no subsumption check for $C_i^{\perp}$ is performed.

leads to a reduction of normal subsumption tests, which is the combination of the reductions found when using $C_i^{\perp}$- or $C_i^{\top}$-approximation by itself. The true subsumption tests are reduced from 24 to 16 and the false subsumption tests are reduced from 279 to 247. Note that the reduction of 8 (24 - 16) true subsumption tests and 32 (279 - 247) are now replaced by 157 true $C_0^{\perp}$-subsumption tests, 32 false $C_0^{\perp}$-subsumption tests, 8 true $C_0^{\top}$-subsumption tests, and 149 false $C_0^{\top}$-subsumption tests.

5.1 Analysis of $C_i^{\perp}$-/$C_i^{\top}$-Approximation

The approximation of concept classification in the TAMBIS ontology using the method of Cadoli and Schaerf reveals at least four points of interest. First, Table 1 shows that using $C_i^{\perp}$-approximation can only lead to a reduction of the false subsumption tests and $C_i^{\top}$-approximation can only lead to a reduction of the true subsumption tests. These results could be expected as they follow from Theorem 1 and are reflected by Algorithm 3, 4, and 5. Using Theorem 1 we have the following reasoning steps for $C_i^{\perp}$-approximation:

$$\begin{aligned} \texttt{Query} \not\sqsubseteq \texttt{Concept} &\Leftrightarrow (\texttt{Query} \sqcap \neg\texttt{Concept}) \text{ is satisfiable} \\ &\Leftarrow (\texttt{Query} \sqcap \neg\texttt{Concept})_i^{\perp} \text{ is satisfiable.} \end{aligned}$$

Hence, when $(\texttt{Query} \sqcap \neg\texttt{Concept})_i^{\perp}$ *is not satisfiable*, nothing can be concluded and approximation can not lead to any gain.

Using Theorem 1 we have the following reasoning steps for $C_i^{\top}$-approximation:

$$\begin{aligned} \texttt{Query} \sqsubseteq \texttt{Concept} &\Leftrightarrow (\texttt{Query} \sqcap \neg\texttt{Concept}) \text{ is not satisfiable} \\ &\Leftarrow (\texttt{Query} \sqcap \neg\texttt{Concept})_i^{\top} \text{ is not satisfiable.} \end{aligned}$$

Hence, when $(\texttt{Query} \sqcap \neg\texttt{Concept})_i^{\top}$ *is satisfiable*, nothing can be concluded and approximation can not lead to any gain.

Second, no approximations are used on a level higher than zero. This is a direct consequence of the TAMBIS ontology containing no nested concept definitions. Further on, we show this to be the case for most ontologies found in practice.

Third, both $C_i^{\perp}$- and $C_i^{\top}$-approximation are not applied in all subsumption tests that are theoretically possible. With normal classification 303 (24 + 279) subsumption tests are needed. However, with $C_i^{\perp}$-approximation in only 189 (157 + 32) cases approximation was actually used. In the remaining 114 (303 - 189) cases approximation had no effect on the concept definitions, i.e., $C_i^{\perp} = C$, and no test was therefore performed. Hence, in 38% of the subsumption tests, approximation was not used. Similar observations hold for $C_i^{\top}$-approximation. This observation indicates that $C_i^{\perp}$-/$C_i^{\top}$-approximation is not very useful (at least for the TAMBIS ontology) for approximating classification in an ontology.

Fourth, apart from the successful reduction of normal subsumption tests, we must also consider the cost for obtaining the reduction. For example, with $C_i^{\perp}$-approximation we obtained a reduction in 32 false subsumption tests, i.e., 32 normal false subsumption tests could be replaced by 32 cheaper false $C_0^{\perp}$-subsumption tests, however it also

cost an extra 157 true $C_0^{\bot}$-subsumption tests that did not lead to any reduction. As nothing can be deduced from these 157 true $C_0^{\bot}$-subsumption tests, these computations are wasted and reduce the gain obtained with the 32 reduced false subsumption tests considerably. Obviously, these unnecessary true $C_0^{\bot}$-subsumption tests should be minimised. No final verdict can be made however, because it all depends on the computation time needed to compute the normal subsumption tests and $C_0^{\bot}$-subsumption tests, but 157 seems rather high. Similar observations hold for $C_i^{\top}$-approximation.

Analysing the high amount of unnecessary subsumption tests, we discovered a phenomenon, which we call *term collapsing*. We illustrate term collapsing through an example taken from the Wine ontology. Suppose that during a classification the subsumption test $\texttt{Query} \sqsubseteq \texttt{WhiteNonSweetWine}$ is generated. The definition for $\texttt{WhiteNonSweetWine}$ is:

$$\texttt{Wine} \sqcap \exists \text{hasColor}.\{\textit{White}\} \sqcap \forall \text{hasSugar}.\{\textit{OffDry}, \textit{Dry}\}.$$

The subsumption query is first transformed into a satisfiability test, i.e., $\texttt{Query} \sqsubseteq \texttt{WhiteNonSweetWine} \Leftrightarrow \texttt{Query} \sqcap \neg\texttt{WhiteNonSweetWine}$ *is unsatisfiable*, because $C_i^{\bot}$-/$C_i^{\top}$-approximation is defined in terms of satisfiability checking.
The definition of $\neg\texttt{WhiteNonSweetWine}$ is

$$\begin{aligned} &\equiv \neg(\texttt{Wine} \sqcap \exists \text{hasColor}.\{\textit{White}\} \sqcap \forall \text{hasSugar}.\{\textit{OffDry}, \textit{Dry}\}) \\ &\equiv \neg\texttt{Wine} \sqcup \forall \text{hasColor}.\neg\{\textit{White}\} \sqcup \exists \text{hasSugar}.\neg\{\textit{OffDry}, \textit{Dry}\}. \end{aligned}$$

and therefore the approximation $(\neg\texttt{WhiteNonSweetWine})_0^{\top}$ is

$$\begin{aligned} &\equiv (\neg\texttt{Wine} \sqcup \forall \text{hasColor}.\neg\{\textit{White}\} \sqcup \exists \text{hasSugar}.\neg\{\textit{OffDry}, \textit{Dry}\})_0^{\top} \\ &\equiv (\neg\texttt{Wine})_0^{\top} \sqcup (\forall \text{hasColor}.\neg\{\textit{White}\})_0^{\top} \sqcup (\exists \text{hasSugar}.\neg\{\textit{OffDry}, \textit{Dry}\})_0^{\top} \\ &\equiv \neg\texttt{Wine} \sqcup \forall \text{hasColor}.\neg\{\textit{White}\} \sqcup \top \\ &\equiv \top. \end{aligned}$$

Therefore, approximating the expression $\texttt{Query} \sqcap \neg\texttt{WhiteNonSweetWine}$ results in checking unsatisfiability of $\texttt{Query}_0^{\top}$, i.e., $(\texttt{Query} \sqcap \neg\texttt{WhiteNonSweetWine})_0^{\top} \Leftrightarrow \texttt{Query}_0^{\top} \sqcap \top \Leftrightarrow \texttt{Query}_0^{\top}$ *is unsatisfiable*. This test most likely fails, because in a consistent ontology $\texttt{Query}$ will be satisfiable and as $\texttt{Query}$ is more specific than $\texttt{Query}_0^{\top}$, i.e., $\texttt{Query} \sqsubseteq \texttt{Query}_0^{\top}$, the latter will be satisfiable.

Analogously, applying $C_i^{\bot}$-approximation may result in a collapse of the $\texttt{Query}$ to $\bot$. This occurs whenever $\texttt{Query}$ contains a conjunction with at least one $\exists$-quantifier. In this case, the entire subsumption test is collapsed into checking the satisfiability of $\bot$. As $\bot$ can never be satisfied, this results in an unnecessary subsumption test.

For the TAMBIS ontology we counted the numbers of occurrences of term collapsing in approximated concept expressions. With $C_i^{\top}$-approximation 65 terms out of 181 collapsed. In other words, 35.9% of the approximated false subsumption tests are obviously not needed and should be avoided. With $C_i^{\bot}$-approximation it is even more severe drastic: 157 terms out of 157 collapsed. With $C_i^{\bot}$- and $C_i^{\top}$-approximation 190 terms out of 306 collapsed. In other words 62.1% of the approximated subsumption tests are not needed.

An additional linear time test could be added to the approximation algorithm to detect term collapsing. However, an optimised DL reasoner with lazy evaluation would perform this simple test in a similar way. Experiments indeed show that the DL reasoner quickly detects term collapsing.

Summarising, using the proposed approximation method by Cadoli and Schaerf [1] on query classification in the TAMBIS ontology leads to many collapsing terms. Furthermore, in only a few cases expensive subsumption tests are replaced by cheaper approximated subsumption tests. These results indicate that their approximation method does not fit practical situations well. A different approximation method may provide better approximation.

5.2 Further Experiments

Although practical results of $C_i^{\perp}$-/$C_i^{\top}$-approximation are somewhat disappointing for the TAMBIS ontology, similar experiments were made with other ontologies. Table 2 summarises the results of $C_i^{\perp}$-/$C_i^{\top}$-approximation applied to the reclassification of 10 unfoldable concepts in five other Ontologies.

Table 2. Number of subsumption tests for reclassification in five ontologies.

		normal		$C_i^{\perp}$		$C_i^{\top}$		$C_i^{\perp}$&$C_i^{\top}$	
		true	**false**	**true**	**false**	**true**	**false**	**true**	**false**
Dolce (10)	$C_0^{\perp}$	-	-	0	0	-	-	0	0
	$C_0^{\top}$	-	-	-	-	0	0	0	0
	normal	10	113	10	113	10	113	10	113
Galen (10)	$C_0^{\perp}$	-	-	0	0	-	-	0	0
	$C_0^{\top}$	-	-	-	-	0	0	0	0
	normal	10	12190	10	12190	10	12190	10	12190
Monet (10)	$C_0^{\perp}$	-	-	0	0	-	-	0	0
	$C_0^{\top}$	-	-	-	-	0	0	0	0
	normal	20	656	20	656	20	656	20	656
MadCow (10)	$C_0^{\perp}$	-	-	145	0	-	-	145	0
	$C_0^{\top}$	-	-	-	-	5	140	5	140
	normal	66	152	66	152	61	152	61	152
Wine (10)	$C_0^{\perp}$	-	-	228	1	-	-	228	1
	$C_0^{\top}$	-	-	-	-	6	223	6	222
	normal	33	252	33	251	27	252	27	251

For the first three Ontologies in Table 2, the DOLCE[7], Galen[8], and Monet ontology[9], $C_i^{\perp}$- or $C_i^{\top}$-approximation has no effect. In these three Ontologies, $C_i^{\perp}$-/$C_i^{\top}$-

[7] An ontology for linguistic and cognitive engineering [18].
[8] A medical terminology developed in the Galen project [19].
[9] An ontology for mathematical web services [20].

approximation does not change any concept expression and therefore no reduction in normal subsumption tests can be obtained. An analysis of these three Ontologies shows that the Ontologies use some roles and/or attributes, but the $\exists$- and/or $\forall$-quantifiers are very rarely used. For example, the Monet ontology contains 2037 concepts, 34 roles, and 10 attributes. The $\exists$-constructor is only used in 13 definitions (0.64% of all concept definitions). The $\forall$-constructor is only used in 11 cases (0.54% of all concept definitions). Therefore no quantifiers were present in the ten randomly selected queries. As quantifiers are so rare, $C_i^{\perp}$-/$C_i^{\top}$-approximation seems to be useless for those Ontologies.

The next two Ontologies in Table 2, MadCow[10] and Wine, are somewhat artificial because they are developed for demonstrating the expressive power of DLs. $C_i^{\perp}$-/$C_i^{\top}$-approximation was applied to classification in both Ontologies, but this leads to almost no reduction of normal subsumption tests. In the Madcow ontology only 5 true subsumption tests are reduced and in the Wine ontology only 7 subsumption tests are reduced (6 true subsumption tests + 1 false subsumption test). Many more subsumption tests are not reduced. In many cases approximating subsumption tests leads to term collapsing and useless subsumption tests.

6 Conclusions

We argued that the idea of approximate logical reasoning matches the requirements of the Semantic Web in terms of robustness against errors and the ability to cope with limited resources better than conventional reasoning methods. At the same time, approximate logical inference avoids the problems of many numerical approaches for approximate reasoning like the proper interpretation of the numeric values assigned to statements and the problem of acquiring these numbers. We tested a concrete method for approximate logical reasoning in DLs against these claims by applying it to the classification problem on a number of Ontologies. In particular, subsumptions were approximated by sequences of weaker and stronger subsumptions. We showed that in principle both approximations can contribute to the efficiency of query classification.

The main result, however, is that the use of the approximation method for DLs proposed by Cadoli and Schaerf is problematic for two reasons:

- A problematic side effect of using the approximation method is the collapsing of concept expressions leading to many unnecessary approximation steps. This happens either when terms of a disjunction are replaced by $\top$ or terms of a conjunction are replaced by $\perp$. The former case happens when $C_i^{\top}$ is used on a concept that contains a universal quantifier at the top level of the definition. The latter happens when $C_i^{\perp}$ is used on a concept with an existential quantifier at the top level of the definition. This feature of the approach is quite problematic as it excludes an important class of query concepts from the method, namely translations of conjunctive queries which are mostly translated using nested existential quantifications [22].
- The experiments show that only in some cases the method is able to successfully replace subsumption tests by cheaper approximations. In many cases like DOLCE, Galen, and Monet no test could successfully be approximated. This observation

[10] Ontology about mad cows, part of the OWL Reasoning Examples [21].

can be explained by the fact that the approximation method only works on nested expressions that are existentially quantified. Many existing Ontologies, however, do not contain concept expressions with nested expressions. The average ontology on the Semantic Web uses quite simple concept expressions that, if at all, are of depth one. The approximation method by Cadoli and Schaerf was designed based on theoretical considerations to reduce worst case complexity of the subsumption problem, but it does not take practical considerations like the nature of definitions that are likely to be found in Ontologies into account.

We conclude that the use of this specific method of approximating subsumption is often not suited for Semantic Web reasoning. Nevertheless, we believe in the general idea of approximate logical reasoning. The goal is to find an approximation strategy that takes the specifics of Ontologies into account. A particular problem with the current approach is the reliance on alternations of $\forall$- and $\exists$-quantifiers. A straightforward way to modify this approach is to find alternative strategies for selecting subexpressions that are to be replaced by $\top$, $\bot$, or other simpler subconcepts. A good candidate, that will be explored in future work, can use domain knowledge to determine the subset of the vocabulary to be replaced. We could for example first exclude very specific terms and then gradually add more specific ones. This and other options for approximating Semantic Web reasoning will be studied in future research.

References

1. Schaerf, M., Cadoli, M.: Tractable reasoning via approximation. Artificial Intelligence **74** (1995) 249–310
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F.: The Description Logic Handbook - Theory, Implementation and Applications. Cambridge University Press (2003)
3. Borgida, A., Etherington, D.W.: Hierarchical knowledge bases and efficient disjunctive reasoning. In Brachman, R.J., Levesque, H.J., Reiter, R., eds.: KR'89: Principles of Knowledge Representation and Reasoning. Morgan Kaufmann, San Mateo, California (1989) 33–43
4. Baader, F., Küsters, R., Molitor, R.: Rewriting concepts using terminologies. In Cohn, A.G., Giunchiglia, F., Selman, B., eds.: Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000), San Francisco, Morgan Kaufman (2000) 297–308
5. Brandt, S., Küsters, R., Turhan, A.Y.: Approximation and difference in description logics. In Fensel, D., Giunchiglia, F., McGuiness, D., Williams, M.A., eds.: Proceedings of the Eighth International Conference on Principles of Knowledge Representation and Reasoning (KR2002), San Francisco, CA, Morgan Kaufman (2002) 203–214
6. McAllester, D.: Truth maintenance. In: Proceedings of AAAI'90, Morgan Kaufmann (1990) 1109–1116
7. OWL test suite. `http://www.w3.org/TR/owl-test/`.
8. Donini, F., Hollunder, B., Lenzerini, M., Spaccamela, A.M., Nardi, D., Nutt, W.: The complexity of existential quantification in concept languages. Artificial Intelligence **53** (1992) 309–327
9. Groot, P., ten Teije, A., van Harmelen, F.: Towards a Structured Analysis of Approximate Problem Solving: a Case Study in Classification. In Dubois, D., Welty, C., Williams, M., eds.: Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR 2004), Whistler, BC, Canada, AAAI Press (2004) 399–406

10. ten Teije, A., van Harmelen, F.: Exploiting domain knowledge for approximate diagnosis. In Pollack, M., ed.: Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97). Volume 1., Nagoya, Japan, Morgan Kaufmann (1997) 454–459
11. ten Teije, A., van Harmelen, F.: Computing approximate diagnoses by using approximate entailment. In Aiello, G., Doyle, J., eds.: Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR-96), Boston, Massachusetts, Morgan Kaufman (1996)
12. Grosof, B.N., Horrocks, I., Volz, R., Decker, S.: Description logic programs: Combining logic programs with description logic. In: Proc. of the Twelfth International World Wide Web Conference (WWW 2003), ACM (2003) 48–57
13. Horrocks, I.: The FaCT system. In: Proceedings of the second International Conference on Analytic Tableaux and Related Methods. Volume 1397 of Lecture Notes in Artificial Intelligence., Springer (1998) 307–312
14. Haarslev, V., Möller, R.: Race system description. In: Proceedings of the 1999 Description Logic Workshop (DL'99). CEUR Electronic Workshop Proceedings (1999) 130–132
15. Haarslev, V., Möller, R.: Racer system description. In: Proceedings of the International Joint Conference on Automated Reasoning (IJCAR 2001). Volume 2083 of Lecture Notes in Artificial Intelligence., Springer (2001) 701–705
16. Bechhofer, S., Möller, R., Crowther, P.: The dig description logic interface. In: Proceedings of DL2003 International Workshop on Description Logics, Rome (2003)
17. Baker, P., Brass, A., Bechhofer, S., Goble, C., Paton, N., Stevens, R.: TAMBIS: Transparent Access to Multiple Bioinformatics Information Sources. An Overview. In: Proceedings of the Sixth International Conference on Intelligent Systems for Molecular Biology (ISMB'98), Menlow Park, California, AAAI Press (1998) 25–34
18. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A.: Ontology Library. WonderWeb Deliverable D18. Laboratory For Applied Ontology - ISTC-CNR. (20003)
19. Rector, A.L., Nowlan, W.A., Glowinski, A.: Goals for concept representation in the galen project. In: Proceedings of the Seventeenth Annual Symposium on Computer Applications in Medical Care (SCAMC-93), Washington DC, USA (1993) 414–418
20. Caprotti, O., Dewar, M., Turi, D.: Mathematical service matching using description logic and owl. In: To appear in Proceedings 3rd Int'l Conference on Mathematical Knowledge Management (MKM'04). Volume 3119 of Lecture Notes in Computer Science., Springer-Verlag (2004)
21. Bechhofer, S.: OWL Reasoning Examples. University of Manchester (2003) `http://owl.man.ac.uk/2003/why/latest/`.
22. Horrocks, I., Tessaris, S.: A conjunctive query language for description logic aboxes. In: National conference on artificial intelligence (AAAI 2000). (2000) 399–404