On Logical Consequence for Collections of OWL Documents

Yuanbo Guo and Jeff Heflin

Computer Science & Engineering Dept.,
Lehigh University, Bethlehem, PA18015, USA
{yug2, heflin}@cse.lehigh.edu

Abstract. In this paper, we investigate the (in)dependence among OWL documents with respect to the logical consequence when they are combined, in particular the inference of concept and role assertions about individuals. On the one hand, we present a systematic approach to identifying those documents that affect the inference of a given fact. On the other hand, we consider ways for fast detection of independence. First, we demonstrate several special cases in which two documents are independent of each other. Secondly, we introduce an algorithm for checking the independence in the general case. In addition, we describe two applications in which the above results have allowed us to develop novel approaches to overcome some difficulties in reasoning with large scale OWL data. Both applications demonstrate the usefulness of this work for improving the scalability of a practical Semantic Web system that relies on the reasoning about individuals.

1 Introduction

To fully exploit the power of the Semantic Web, it is crucial to reason, in an efficient and scalable way, with its data, e.g., those described in the OWL language [4]. Representative OWL reasoners of today such as FaCT [12] and Racer [9], implemented upon the most advanced reasoning mechanisms, have seen some successful applications. However, the scalability of those systems is still far from satisfactory in the context of the Semantic Web, which represents much larger problem sizes than those traditionally found in artificial intelligence. For instance, as Haarslev and Möller [10] pointed out, the scale of data that Racer could appropriately handle is still rather limited (only up to 30,000 individuals in their experimental setting) given non-naïve ontologies and instances. Hence, a great challenge remains in order to implement a practical Semantic Web system that is capable of reasoning on large scale data.

In this paper, we investigate a specific issue in this regard. We consider the situation when a system needs to reason over a large collection of OWL documents in order to answer queries against them. If scalability was not an issue, we could load the entire set of documents into a contemporary OWL reasoner and then issue queries in a normal fashion. However, this will not work in practice considering the large number of documents the system has to deal with. For instance, practical reasons such as memory limitations might simply prevent the system from processing a document collection in its entirety.

Y. Gil et al. (Eds.): ISWC 2005, LNCS 3729, pp. 338 – 352, 2005.

This work explores ways for improving the scalability of the system in performing the above kind of task. Note, here we assume we have access to a powerful OWL reasoner and we are not concerned with the implementation or optimization of the reasoner per se. Instead, we consider how large problems that cannot typically be solved by the reasoner can be reduced into problems that can be solved.

Specifically, we have the following considerations. First, given a set of documents, it might be the case that, only a subset of the documents is needed in order for a specific statement to be true. In other words, we do not need to combine the whole set and perform expensive reasoning on it in order to guarantee a correct inference with respect to that statement. Furthermore, if we could show that a set of documents, when combined, would not form any conclusions that are not supported by any of the documents individually, then we may adopt a divide-and-conquer approach for the related reasoning tasks on those documents.

The above considerations have led us to research on what we call dependence and independence relationships among OWL documents. Now that the notions apply to general logical knowledge bases, we define them in general as follows:

Def. 1.1. *Let K be the set of knowledge bases* $\{\mathcal{K}_1,\ldots,\mathcal{K}_n\}$ $(n>1)$*. The members of K are* ***logically dependent on each other with respect to a sentence*** α *iff K is a minimal set such that* $K \models \alpha$.

Def. 1.2. *Let K be the set of knowledge bases* $\{\mathcal{K}_1,\ldots,\mathcal{K}_n\}$ $(n>1)$*. The members of K are* ***logically independent of each other*** *iff for every sentence* α*, there are no members of K that are logically dependent on each other with respect to* α.

What we shall look into in this work is the logical (in)dependence relationships among OWL documents with respect to OWL assertions. Here, we refer to an OWL document as an RDF/XML-syntax document that conforms to the specification by the OWL reference [4: Section 2]. In addition, to clarify the meaning of "$K \models \alpha$" in the above definitions when applied to OWL, we say a set of OWL documents entail an assertion α iff α is entailed by the union of the imports closure[1] of every document in the set. Note that this is different from the notion of entailment defined in ongoing research on distributed description logics [2], which focuses on preventing the propagation of inconsistency.

This work will focus on OWL Lite and OWL DL, the two decidable sublanguages of OWL. Since OWL Lite and OWL DL are logically equivalent to DL $\mathcal{SHIF}(\mathbf{D})$ and DL $\mathcal{SHOIN}(\mathbf{D})$ respectively [13], throughout the discussion, we will regard an OWL document as a description logic (DL) knowledge base consisting of a TBox $\mathcal{T}$ (equivalent to the parse result of the ontology) and an ABox $\mathcal{A}$ (equivalent to the parse result of the instance data committing to the ontology). We use $(\mathcal{T}, \mathcal{A})$ to denote such a knowledge base. To facilitate the discussion, we shall use the following terms, which are not conventionally used in the literature:

[1] Imports closure of an OWL document is the information in the document unioned with the information in the imports closure of documents that are imported by that document [19].

An OWL DL knowledge base (resp. an OWL DL TBox, an OWL DL ABox) is the result of parsing an OWL DL document (resp. the ontology part of an OWL DL document, the instance data part of an OWL DL document) into a DL knowledge base (resp. a DL TBox, a DL ABox). Similarly for OWL Lite knowledge base, OWL Lite TBox, and OWL Lite ABox. In the subsequent discussion, for brevity, "DL" will be omitted without confusion.

As a final remark, as the first step of the work, we consider OWL documents that commit to a common ontology. Additionally, we will concentrate on ABox reasoning, in particular the inference of concept assertions and role assertions. This is motivated by the expectation that the instance data will greatly outnumber the ontologies on the Semantic Web.

2 Identifying the Logical Dependence

In this section, we examine the dependence relationship among OWL documents. Specifically, we attempt to define a systematic way to identify what documents, when combined together, may affect the inference of a specific assertion. We base our approach on the identification of the relevant assertion set to a given assertion, as defined below. Again, we first define the notions in general. Then, we give a specific account of the notions for ABox assertions.

Def. 2.1. *A set S of sentences is a* ***relevant sentence set*** *to the logical entailment of a sentence* α*, denoted Rel(S,* α*), iff* $\mathcal{S} \models \alpha$.

Def. 2.2. *A set S of sentences is a* ***minimal relevant sentence set*** *to the logical entailment of a sentence α, denoted MinRel(S, α), iff S is a minimal set such that Rel (S, α).*

Def. 2.3. *Given a consistent TBox* $\mathcal{T}$*, a set S of ABox assertions is a* ***relevant ABox assertion set*** *to the logical entailment of an ABox assertion α, denoted* $Rel_{\mathcal{T}}(S, \alpha)$*, iff* $<\mathcal{T},S> \models \alpha$.

Def. 2.4. *Given a consistent TBox* $\mathcal{T}$*, a set S of ABox assertions is a* ***minimal relevant ABox assertion set*** *to the logical entailment of an ABox assertion α, denoted* $MinRel_{\mathcal{T}}(S, \alpha)$*, iff S is a minimal set such that* $Rel_{\mathcal{T}}(S, \alpha)$.

Now we are able to establish a relationship between relevance and dependence for OWL knowledge bases with the same TBox, i.e., they commit to the same ontology.

Proposition 2.1. *Let K be the set of OWL knowledge bases* $\mathcal{K}_1=(\mathcal{T}, \mathcal{A}_1),\ldots, \mathcal{K}_n=(\mathcal{T}, \mathcal{A}_n)$ *(n>1), wherein* $\mathcal{T}$ *is consistent. The members of K are logically dependent on each other with respect to an ABox assertion α iff K is a minimal set such that there exist* $\alpha_1,\ldots,\alpha_m$ *such that* $MinRel_{\mathcal{T}}(\{\alpha_1,\ldots,\alpha_m\}, \alpha)$ *and for every* α_i *(i=1,…,m) there exists* $\mathcal{A}_j$ *(j=1,…,n) such that* $\alpha_i \in \mathcal{A}_j$.

The above proposition, in effect, indicates a way of determining the dependence among a set of OWL knowledge bases with respect to a given assertion, i.e., by look-

ing for relevant assertion sets to that assertion. Therefore, the remaining question is how to identify those relevant sets. We begin by identifying a set of inference rules for an OWL Lite knowledge base, as shown below.

R1) If $\alpha \in \mathcal{A}$ then $\mathcal{A} \vdash \alpha$

R2) If $\mathcal{T} \models C_1 \sqcap \cdots \sqcap C_n \sqsubseteq C$ and $\mathcal{A} \vdash a:C_1, \ldots, a:C_n$, then $\mathcal{A} \vdash a:C$

R3) If $\mathcal{T} \models R_1 \sqsubseteq R_2$ and $\mathcal{A} \vdash \langle a,b \rangle:R_1$ then $\mathcal{A} \vdash \langle a,b \rangle:R_2$

R4) If $\mathcal{T} \models U_1 \sqsubseteq U_2$ and $\mathcal{A} \vdash \langle a,v \rangle:U_1$ then $\mathcal{A} \vdash \langle a,v \rangle:U_2$

R5) If $\mathcal{A} \vdash \langle a,b \rangle:R$ then $\mathcal{A} \vdash \langle b,a \rangle:R^-$

R6) For $R \in \boldsymbol{R}_+$, if $\mathcal{A} \vdash \langle a,b \rangle:R$ and $\langle b,c \rangle:R$ then $\mathcal{A} \vdash \langle a,c \rangle:R$

R7) If $\mathcal{A} \vdash \langle a,b \rangle:R$ and $b:C$ then $\mathcal{A} \vdash a: \exists R.C$

R8) If $\mathcal{A} \vdash \langle a,v \rangle:U$ and $v \in D$ then $\mathcal{A} \vdash a: \exists U.D$

R9) If $\mathcal{A} \vdash a: \forall R.C$ and $\langle a,b \rangle:R$ then $\mathcal{A} \vdash b:C$

R10) If $\mathcal{A} \vdash \langle a,b \rangle:R$ then $\mathcal{A} \vdash a: \geq 1R$

R11) If $\mathcal{A} \vdash \langle a,v \rangle:U$ then $\mathcal{A} \vdash a: \geq 1U$

R12) If $\mathcal{A} \vdash a: \leq 1R$, $\langle a,b_1 \rangle:R$, and $\langle a,b_2 \rangle:R$ then $\mathcal{A} \vdash b_1=b_2$

R13) If $\mathcal{A} \vdash a=b$ then $\mathcal{A} \vdash b=a$

R14) If $\mathcal{A} \vdash a=b$ and $a:C$ (resp. $\langle a,c \rangle:R$, $\langle c,a \rangle:R$, $a=c$) then $\mathcal{A} \vdash b:C$ (resp. $\langle b,c \rangle:R$, $\langle c,b \rangle:R$, $b=c$)

R15) If $v_1 = v_2$ and $\mathcal{A} \vdash \langle a,v_1 \rangle:U$ then $\mathcal{A} \vdash \langle a,v_2 \rangle:U$

We have defined the above rules by referring to the work of Royer and Quantz [20, 21] with several extensions and adaptations. They described a generic approach to deriving, via Sequent Calculus, complete inference rules for description logics. They also provided the result for a specific description logic, which is generally more expressive than OWL Lite if we ignore datatypes and equality. We extend those rules by taking into account datatypes and equality[2]. Moreover, we handle the inference involving subsumption differently. They provided over 100 rules for subsumption. We chose not to include them since our focus is on ABox assertions and those rules greatly complicate the derivation of the relationship we are looking for. Instead, we remedy the absence of those subsumption rules by relying on the entailment of the TBox, as Rules 2 to 4 show. From the implementation point of view, this means we resort to the reasoner for checking subsumptions. We consider this as a reasonable method especially when the application involves a large amount of instance data over a relatively small number of ontologies.

As some other remarks, in defining the rules we only consider a consistent knowledge base.[3] Also, for simplicity we assume no untyped data values in the knowledge

[2] OWL supports the expression of equality. Also the language does not make the unique names assumption.

[3] These rules may easily be extended to support the inconsistent case and then we consider they could potentially be exploited for other kinds of tasks such as debugging inconsistencies. However, that is beyond the interest of this paper.

base. In addition, we rely upon the reasoner to reason about the data values such as deciding if they belong to a specific datatype and their equality (refer to Rules 8 and 15). Moreover, in the rules and also the subsequent discussion, we assume that every assertion $a{:}C_1 \sqcap \cdots \sqcap C_n$ is represented as $a{:}C_1,\ldots,a{:}C_n$. Finally, our rule set is incomplete for OWL DL, which additionally supports *oneOf*. As Royer and Quantz pointed out, taking account of enumerated classes would greatly complicate the resultant rule set. Thus they did not give a complete set of rules with respect to reasoning about enumerated classes, and neither did we. We leave this as an open issue.

The above rule set may not be suitable for implementing a practical reasoner due to its special handling of TBox related inferences. However, these natural deduction-style inference rules can facilitate the identification of the relevant assertion sets to a given assertion, as shown in Proposition 2.2. The proposition is obvious by following the rule firing relations (refer to the right column for the corresponding rules).

Proposition 2.2. *In OWL Lite, given a consistent TBox $\mathcal{T}$, a consistent set S of ABox assertions, and a concept or role assertion α, $Rel_{\mathcal{T}}(S, \alpha)$ iff S is at least one of the following sets:*

For any α:

- $\{ \alpha \}$ (R1)

If α is a:C:

- $S_1 \cup \cdots \cup S_n$ *wherein* $Rel_{\mathcal{T}}(S_1, a{:}C_1)$, …, $Rel_{\mathcal{T}}(S_n, a{:}C_n)$, $\mathcal{T} \models C_1 \sqcap \cdots \sqcap C_n \sqsubseteq C$ (R2)
- $S_1 \cup S_2$ *wherein* $Rel_{\mathcal{T}}(S_1, b{:}\forall R.C)$, $Rel_{\mathcal{T}}(S_2, \langle b,a\rangle{:}R)$ (R9)
- $S_1 \cup S_2$ *wherein* $Rel_{\mathcal{T}}(S_1, a{=}b)$, $Rel_{\mathcal{T}}(S_2, b{:}C)$ (R14)

If α is a: $\exists R.C$:

- $S_1 \cup S_2$ *wherein* $Rel_{\mathcal{T}}(S_1, \langle a,b\rangle{:}R)$, $Rel_{\mathcal{T}}(S_2, b{:}C)$ (R7)

If α is a: $\exists U.D$:

- S *wherein* $Rel_{\mathcal{T}}(S, \langle a,v\rangle{:}U)$, $v \in D$ (R8)

If α is a:$\geq 1R$:

- S *wherein* $Rel_{\mathcal{T}}(S, \langle a,b\rangle{:}R)$ (R10)

If α is a:$\geq 1U$):

- S *wherein* $Rel_{\mathcal{T}}(S, \langle a,v\rangle{:}U)$ (R11)

If α is $\langle a,b\rangle{:}R$:

- S *wherein* $Rel_{\mathcal{T}}(S, \langle a,b\rangle{:}T)$, $\mathcal{T} \models T \sqsubseteq R$ (R3)
- S *wherein* $Rel_{\mathcal{T}}(S, \langle b,a\rangle{:}R^-)$ (R5)
- $S_1 \cup S_2$ *wherein* $Rel_{\mathcal{T}}(S_1, \langle a,c\rangle{:}R)$, $Rel_{\mathcal{T}}(S_2, \langle c,b\rangle{:}R)$, $R \in \boldsymbol{R}_+$ (R6)
- $S_1 \cup S_2$ *wherein* $Rel_{\mathcal{T}}(S_1, a{=}c)$, $Rel_{\mathcal{T}}(S_2, \langle c,b\rangle{:}R)$, *or* (R14)
  $Rel_{\mathcal{T}}(S_1, b{=}c)$, $Rel_{\mathcal{T}}(S_2, \langle a,c\rangle{:}R)$ (R14)

If α is $\langle a,v\rangle{:}U$:

- S *wherein* $Rel_{\mathcal{T}}(S, \langle a,v\rangle{:}V\}$, $\mathcal{T} \models V \sqsubseteq U$ (R4)
- S *wherein* $Rel_{\mathcal{T}}(S, \langle a,w\rangle{:}U)$, $v{=}w$ (R15)

If α *is a=b:*

- $S_1 \cup S_2 \cup S_3$
  wherein $Rel_{\mathcal{T}}(S_1, c:\leq 1R)$, $Rel_{\mathcal{T}}(S_2, \langle c,a\rangle:R)$, $Rel_{\mathcal{T}}(S_2, \langle c,b\rangle:R)$ (R12)
- S *wherein* $Rel_{\mathcal{T}}(S, b=a)$ (R13)
- $S_1 \cup S_2$ *wherein* $Rel_{\mathcal{T}}(S_1, c=a)$, $Rel_{\mathcal{T}}(S_2, c=b)$ (R14)

Since determining Min$Rel_{\mathcal{T}}$is straightforward given the information of $Rel_{\mathcal{T}}$, Propositions 2.1 and 2.2 have rendered us a systematic way of identifying the dependent knowledge base set with respect to a specific assertion. Next, we show an application of the approach.

Application I

In another work [8], we aim at developing a query answering system for a repository of OWL documents. One important functionality of the system is to answer a user's query about the minimal subsets of documents in the repository that entail a specific assertion. Currently we only consider queries about concept instances on OWL Lite documents. In order to improve query time, the system performs preprocessing of the documents during loading and records which new assertion is entailed by which minimal subsets of documents. To that end, it enumerates the subsets of the documents in the order of their sizes, i.e. single document first, and then the combinations of two documents, and so on, and performs reasoning on those consistent combinations in sequence.

Obviously, processing the document combinations in this fashion is inefficient since the potential number of combinations the system has to handle increases exponentially in the number of documents. Given Propositions 2.1 and 2.2, we are able to employ a different strategy: we perform reasoning on the whole union of the documents immediately after processing every individual document. Then for each inferred concept assertion α from the union, we directly identify, according to those propositions, the minimal document sets (containing more than one document) that entail α, in other words, the sets of documents that are dependent on each other with respect to α.

Although an efficient implementation of the propositions is still a remaining issue, we consider that, with a proper mechanism for indexing and searching the assertions, the complexity of identifying the dependent subsets among a collection of documents will be no more than that of performing reasoning on the union of the entire set. Thus, we can expect prominent performance improvement in cases where new assertions that are entailed by more than one document are relatively few.

To give readers a flavor of this, we introduce an initial evaluation---the system is still under development. We used 100 small OWL Lite documents (committing to the same ontology). These documents are adaptations from the test data of the Lehigh University Benchmark [7], which simulates a realistic domain. We conducted reasoning on the union of the entire document set. Then, we tried to identify the dependent subsets of documents with respect to each of the concept assertions entailed by the union. As a result, we pinpointed 43 subsets each containing two documents from 16 assertions, which were new entailments by the union than the individual documents. Note, in this case, none of those assertions captured 3 or more documents. This is a

significant improvement considering that, before the strategy is adopted, the system had to handle the level of 2^{100} document combinations in order to completely find those subsets.

3 Detecting the Logical Independence – Special Cases

Hereafter, we will switch our focus to the detection of independence. Unlike dependence, the notion of independence is not defined with respect to a specific statement. Instead, we are interested in more general relationships such as independence with respect to all ABox assertions of certain forms. The result in the previous section provides an indirect way for determining this kind of independence between a set of OWL knowledge bases *K*, i.e., by showing that, for every applicable assertion, there are no members of *K* that are mutually dependent with respect to that assertion. However, this is obviously a very inefficient approach since we have to enumerate and test all possible assertions. In the following two sections, we explore some faster ways. We begin with special cases, in which we demonstrate that, under certain conditions, two OWL documents are independent of each other with respect to the assertions of specific forms.

First, we introduce the following notation:

Ind($\mathcal{K}$): The set of individual names in the OWL knowledge base $\mathcal{K}$.

The following theorem reveals the independence relationship between two OWL knowledge bases with disjoint sets of individual names.

Theorem 3.1. *Let $\mathcal{K}_1=(\mathcal{T}, \mathcal{A}_1)$ and $\mathcal{K}_2=(\mathcal{T}, \mathcal{A}_2)$ be two OWL knowledge bases. If $Ind(\mathcal{K}_1)\cap Ind(\mathcal{K}_2)=\varnothing$, then $\mathcal{K}_1$ and $\mathcal{K}_2$ are logically independent of each other with respect to any concept assertion, role assertion, or equality assertion*[4].

Proof. Let $\mathcal{K}=(\mathcal{T}, \mathcal{A}_1 \cup \mathcal{A}_2)$. This is equivalent to proving that for every assertion α being any form of a:C, <a,b>:R, <a,v>:U or a=b, $\mathcal{K} \models \alpha$ iff either $\mathcal{K}_1 \models \alpha$ or $\mathcal{K}_2 \models \alpha$.

(<=) It is trivially true because OWL is monotonic.

(=>) Since $\mathcal{K} \models \alpha$, every model $\mathcal{I}$ of $\mathcal{K}$ satisfies α (1). If either $\mathcal{K}_1$ or $\mathcal{K}_2$ is inconsistent, since everything can be deduced from an inconsistent knowledge base, the theorem is proved. In case both $\mathcal{K}_1$ and $\mathcal{K}_2$ are consistent, suppose $\mathcal{K}_1 \not\models \alpha$ and $\mathcal{K}_2 \not\models \alpha$ (2). Then there must exist a model $\mathcal{I}_1 = (\Delta^{\mathcal{I}1}, \cdot^{\mathcal{I}1})$ (resp. $\mathcal{I}_2 = (\Delta^{\mathcal{I}2}, \cdot^{\mathcal{I}2})$) of $\mathcal{K}_1$ (resp. $\mathcal{K}_2$) that does not satisfy α. We assume that $\Delta^{\mathcal{I}1}$ and $\Delta^{\mathcal{I}2}$ are disjoint (3). We could make the assumption because if that is not the case, we can always replace $\mathcal{I}_1$ with another interpretation $\mathcal{I}_1$' such that 1) $\Delta^{\mathcal{I}1'}$ and $\Delta^{\mathcal{I}2}$ are disjoint, and 2) every domain object in $\Delta^{\mathcal{I}1'}$ has a counterpart in $\Delta^{\mathcal{I}1}$ and stands in the same place in $\cdot^{\mathcal{I}1'}$ as its counterpart does in $\cdot^{\mathcal{I}1}$, and vice versa.

Now define an interpretation $\mathcal{I}=(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ for $\mathcal{K}$ wherein,

- $\Delta^{\mathcal{I}} = \Delta^{\mathcal{I}1} \cup \Delta^{\mathcal{I}2}$
- For every concept C, $C^{\mathcal{I}} = C^{\mathcal{I}1} \cup C^{\mathcal{I}2}$

[4] This does not hold for inequality. For instance, if $\mathcal{T}$ claims the concepts C_1 and C_2 as disjoint whereas $\mathcal{A}_1$ and $\mathcal{A}_2$ assert $a:C_1$ and $b:C_2$ respectively, then we can infer that $a \neq b$ from the union of $\mathcal{K}_1$ and $\mathcal{K}_2$.

- For every (object and datatype) role R, $R^{\mathcal{I}} = R^{\mathcal{I}1} \cup R^{\mathcal{I}2}$
- For every individual a:[5]
  If $a \in \mathrm{Ind}(\mathcal{A}_1)$, $a^{\mathcal{I}} = a^{\mathcal{I}1}$; otherwise, $a^{\mathcal{I}} = a^{\mathcal{I}2}$
- For every data value v, $v^{\mathcal{I}} = v^{\mathbf{D}}$
- The concrete domain $\Delta_{\mathbf{D}}{}^{\mathcal{I}} = \Delta_{\mathbf{D}}{}^{\mathcal{I}1} = \Delta_{\mathbf{D}}{}^{\mathcal{I}2}$

Next we prove that $\mathcal{I}$ is a model of $\mathcal{K}$. As it has been shown that OWL DL entailment is reducible to DL $\mathcal{SHOIN}(\mathbf{D})$ satisfiability [13], we will show that $\mathcal{I}$ is a model of $\mathcal{K}$ based on the semantics of $\mathcal{SHOIN}(\mathbf{D})$ (cf. Fig. 3 of [13]).[6]

First, it is straightforward to show from the definition of $\mathcal{I}$ that $\mathcal{I}$ satisfies the semantics relating to atomic concept, datatype, role, individual, data value, inverse role, top/bottom concept, disjunction, oneOf, concept inclusion, and role inclusion. For example,

$$(C_1 \sqcup C_2)^{\mathcal{I}} = (C_1 \sqcup C_2)^{\mathcal{I}1} \cup (C_1 \sqcup C_2)^{\mathcal{I}2} = (C_1{}^{\mathcal{I}1} \cup C_2{}^{\mathcal{I}1}) \cup (C_1{}^{\mathcal{I}2} \cup C_2{}^{\mathcal{I}2})$$
$$= (C_1{}^{\mathcal{I}1} \cup C_1{}^{\mathcal{I}2}) \cup (C_2{}^{\mathcal{I}1} \cup C_2{}^{\mathcal{I}2}) = C_1{}^{\mathcal{I}} \cup C_2{}^{\mathcal{I}}$$

Moreover, based on the definition of $\mathcal{I}$ plus the assumption (3), we can show that $\mathcal{I}$ satisfies the semantics relating to conjunction, negation, retrictions, and transitive role. For instance,

$$(\exists R.C)^{\mathcal{I}} = (\exists R.C)^{\mathcal{I}1} \cup (\exists R.C)^{\mathcal{I}2}$$
$$= \{x \mid \exists y.\langle x,y\rangle \in R^{\mathcal{I}1} \text{ and } y \in C^{\mathcal{I}1}\} \cup \{x \mid \exists y.\langle x,y\rangle \in R^{\mathcal{I}2} \text{ and } y \in C^{\mathcal{I}2}\}$$
$$= \text{via (3)} = \{x \mid \exists y.\langle x,y\rangle \in R^{\mathcal{I}1} \cup R^{\mathcal{I}2} \text{ and } y \in C^{\mathcal{I}1} \cup C^{\mathcal{I}2}\}$$
$$= \{x \mid \exists y.\langle x,y\rangle \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}$$

Lastly, for every ABox assertion β in $\mathcal{K}$, we can show $\mathcal{I}$ satisfies β based on the definition of $\mathcal{I}$ and the fact that β is originally from either $\mathcal{K}_1$ or $\mathcal{K}_2$ and thus either $\mathcal{I}_1$ or $\mathcal{I}_2$ satisfies β. For example, if $a{:}C$ is from $\mathcal{K}_1$ and therefore satisfied by $\mathcal{I}_1$, then $a^{\mathcal{I}} = a^{\mathcal{I}1}$. Since $a^{\mathcal{I}1} \in C^{\mathcal{I}1}$ and $C^{\mathcal{I}1} \subseteq C^{\mathcal{I}}$, $a^{\mathcal{I}} \in C^{\mathcal{I}}$. Thus $\mathcal{I}$ satisfies $a{:}C$.

So far, we can conclude that $\mathcal{I}$ is a model of $\mathcal{K}$. Next we show $\mathcal{I}$ however does not satisfy α. If α is of the form $a{:}C$, according to the definition of $\mathcal{I}$, $a^{\mathcal{I}}$ equals to either $a^{\mathcal{I}1}$ or $a^{\mathcal{I}2}$. Since neither $\mathcal{I}_1$ nor $\mathcal{I}_2$ satisfies α, $a^{\mathcal{I}1} \notin C^{\mathcal{I}1}$ and $a^{\mathcal{I}2} \notin C^{\mathcal{I}2}$. In addition, given the assumption of (3), we have $a^{\mathcal{I}1} \notin C^{\mathcal{I}2}$ and $a^{\mathcal{I}2} \notin C^{\mathcal{I}1}$. Hence, $a^{\mathcal{I}} \notin C^{\mathcal{I}1} \cup C^{\mathcal{I}2}$ and thus $a^{\mathcal{I}} \notin C^{\mathcal{I}}$, which means $\mathcal{I}$ does not satisfy α. With similar arguments, we can show that $\mathcal{I}$ does not satisfy α of the form $\langle a,b\rangle{:}R$, $\langle a,v\rangle{:}U$ or $a{=}b$ either.

In conclusion, $\mathcal{I}$ is a model of $\mathcal{K}$ but $\mathcal{I}$ does not satisfy α. This is contradictory to (1), which means assumption (2) does not hold. Therefore, either $\mathcal{K}_1 \models \alpha$ or $\mathcal{K}_2 \models \alpha$.

The theorem below considers the independence between OWL knowledge bases in terms of deriving inconsistency.

Theorem 3.2. *Under the same precondition of Theorem 3.1, $\mathcal{K}$ is inconsistent iff either $\mathcal{K}_1$ or $\mathcal{K}_2$ is inconsistent.*

---

[5] Note that for every enumerated class $\{o_1,\ldots, o_n\}$, since $\mathrm{Ind}(\mathcal{K}_1)$ and $\mathrm{Ind}(\mathcal{K}_2)$ are disjoint, $o_1,\ldots, o_n$ could not occur in both knowledge bases.

[6] For simplicity, we ignore the translation of OWL DL into $\mathcal{SHOIN}(\mathbf{D})$ since it maintains the satisfaction of theorem's precondition.

Proof. Again the (<=) part is obvious. (=>) That $\mathcal{K}$ is inconsistent means $\mathcal{K}$ has no models. Suppose both $\mathcal{K}_1$ and $\mathcal{K}_2$ are consistent and thus have a model respectively, we could find a model $\mathcal{I}$ for $\mathcal{K}$ as we do in the proof of Theorem 3.1. Therefore, $\mathcal{K}_1$ and $\mathcal{K}_2$ could not be both consistent.

Corollary 3.1. *Theorem 3.1 (and also Theorem 3.2) still holds if every individual* $a \in Ind(\mathcal{A}_1) \cap Ind(\mathcal{A}_2)$ *appears only in those assertions shared by* $\mathcal{A}_1$ *and* $\mathcal{A}_2$.

Proof. (<=) It is trivially true again. (=>) Define $\mathcal{K}_2'=(\mathcal{T}, \mathcal{A}_2')$ by removing from $\mathcal{A}_2$ all the assertions that are also in $\mathcal{A}_1$. In this way, $\mathcal{K}_1 \cup \mathcal{K}_2'$ still equals to $\mathcal{K}$, but $Ind(\mathcal{K}_1)$ and $Ind(\mathcal{K}_2')$ become disjoint. Therefore according to Theorem 3.1, for every ABox assertion α except the inequality assertion, $\mathcal{K} \models \alpha$ implies either $\mathcal{K}_1 \models \alpha$ or $\mathcal{K}_2' \models \alpha$. Since $\mathcal{K}_2$ subsumes $\mathcal{K}_2'$, based on monotonicity, we have $\mathcal{K} \models \alpha$ implies either $\mathcal{K}_1 \models \alpha$ or $\mathcal{K}_2 \models \alpha$. In a similar fashion, we can prove the cases for Theorem 3.2.

Next, we will consider a different situation by removing the disjointness requirement on individual names while imposing another restriction: we look at OWL documents that contain only RDF(S) [23] features. We look at the RDF(S) fragment of OWL considering oftentimes applications do not need the full expressivity of OWL and the fact that RDF-style documents occupy a considerable portion of the Semantic Web we have seen so far.

Theorem 3.3. *Let* $\mathcal{K}_1=(\mathcal{T}, \mathcal{A}_1)$ *and* $\mathcal{K}_2=(\mathcal{T}, \mathcal{A}_2)$ *be two OWL knowledge bases. If both knowledge bases are limited to the RDF(S) fragment, then* $\mathcal{K}_1$ *and* $\mathcal{K}_2$ *are logically independent of each other with respect to any assertion* α *of the form of* $a:C$ *or* $\langle a,b\rangle:p$.

Proof. It is equivalent to proving that $\mathcal{K}_1 \cup \mathcal{K}_2 \models \alpha$ iff either $\mathcal{K}_1 \models \alpha$ or $\mathcal{K}_2 \models \alpha$.(<=) Once again it is obvious. (=>) Table 3.1 illustrates how we can transfer an RDF(S) statement into a FOL rule. As can be seen, if a knowledge base contains only FOL rules mapping to RDF(S) statements, it will fit into Horn Logic. This means we could apply a simple forward chaining reasoning on that knowledge base (in this case $\mathcal{K}_1 \cup \mathcal{K}_2$) to get a sound and complete inferencing.

Table 3.1. Correspondence between OWL and DL and between DL and FOL (RDF(S) fragment) [22]

OWL Fact/Axiom	DL Syntax	FOL Rule
a type *C*	$a:C$	$C(a)$
a P b	$\langle a,b\rangle:P$	$P(a,b)$
rdfs:subclassOf	$C_1 \sqsubseteq C_2$	$\forall x.C_1(x) \rightarrow C_2(x)$
rdfs:subpropertyOf	$P_1 \sqsubseteq P_2$	$\forall x,y.P_1(x,y) \rightarrow P_2(x,y)$
rdfs:domain	$\top \sqsubseteq \forall P^-.C$	$\forall x,y.P(x,y) \rightarrow C(x)$
rdfs:range	$\top \sqsubseteq \forall P.C$	$\forall x,y.P(x,y) \rightarrow C(y)$

Furthermore, since every rule presented in the above table has only one antecedent, any proof tree generated on the knowledge base actually reduces to a chain-like structure, starting from a fact in the input till the goal fact. In other words, no proof trees

for a new fact involve more than one fact from the input: for two input facts to be used in a proof, there must be a step in the proof that involves two facts that are respectively either the input facts themselves or facts derived from the input facts. However, this is impossible given the above mentioned structure of the proof tree. Thus the theorem is proved.

Application II

Next, we will show an application of the above results. In yet another work [7], we are conducting benchmarks of OWL knowledge base systems with respect to queries upon the instance data. To facilitate evaluating the query completeness and soundness of the systems under test, we intended to use Racer to generate the answer set as a basis for comparison. However, a big problem was that Racer at the current stage is incapable of handling the dataset used in the benchmark in our experimental environment. The smallest dataset used in the benchmark consists of 15 OWL documents. However, as shown in [11], due to Racer has to perform consistency check before answering queries, it could only load up to 5 of the documents (9555 individuals).

Nevertheless, we were able to overcome this problem by virtue of Corollary 3.1. In the benchmark, a dataset consists of multiple OWL Lite documents that commit to the same ontology (also in OWL Lite). We have found out that these documents meet the precondition in Corollary 3.1 with a handful of exceptions[7]. Furthermore, by focusing on those exceptional assertions we have figured out without difficulty that they will not lead to any inferences or inconsistencies across multiple documents. Therefore, we could conclude that these documents are independent of each other with respect to the kind of inference we need, i.e., concept and role instance retrieval. This means we could let Racer do such reasoning on one document from the test set at a time and still guarantee a sound and complete inference by taking the union of the results on every individual document later on. In this way, we have found a solution the above problem.

4 Detecting the Logical Independence – The General Case

In the previous section, we introduced several ways of quickly detecting the independence of OWL knowledge bases in some special cases. Now we look at the general case, i.e., for arbitrary knowledge bases. We have realized that this is a challenging task. Here we present our initial results. The algorithm below is responsible for checking the independence between two OWL knowledge bases.

```
1 procedure CHECK-INDEPENDENCE(𝒦1, 𝒦2)
2 input: OWL knowledge bases 𝒦1=(𝒯, 𝒜1) and 𝒦2=(𝒯, 𝒜2)
3 output: indicate if 𝒦1 and 𝒦2 are independent with
   respect to any concept assertion or role assertion
4 begin
5   I:= Ind(𝒜1) ∩ Ind(𝒜2); //overlap in individual names
6   if I = ∅ then return true;
```

[7] These documents have few overlap in individual names because they are dedicated to the description of different organizations (i.e. academic departments) and their affiliated persons.

```
7   if CHECK-AND-REALIZE(A1)=false then return true;
8   if CHECK-AND-REALIZE(A2)=false then return true;
9   O := { a:C ∈ A1 ∪ A2 | a ∈ I } ∪ {<a,b>:R ∈ A1 ∪ A2 | a ∈ I or b ∈ I } ∪
        { a:C ∈ A1 ∪ A2 | ∃b∈I. <a,b>:R ∈ A1 ∪ A2 or <b,a>:R ∈ A1 ∪ A2 };
10  AO1 := A1 ∪ O;  //Combine A1 and O
11  if CHECK-AND-REALIZE(AO1)=false then return false;
12  if new assertions have been added to AO1 then return
    false;
13  AO2 := A2 ∪ O;  //Combine A2 and O
14  repeat 11-12 for AO2;
15  return true;
16end
17procedure CHECK-AND-REALIZE (A)
18  Check consistency of A;
19  if A is inconsistent then return false;
20  Perform ABox reasoning on A and update A accordingly,
    i.e.,
    A = A ∪
    {a:C | A ⊨ a:C and C is among the most specific con-
    cepts that a is an instance of} ∪
    {<a,b>:R | A ⊨ <a,b>:R and R is among the most spe-
    cific roles that <a,b> is an instance of};
21  return true;
22end
```

Fig. 4.1 illustrates the key operations in the algorithm.

$\mathcal{A}_1$ (after realization)	$\mathcal{A}_2$ (after realization)	I	$\mathcal{O}$
{ o_1:C_1,	{ o_1:C_1,	{ o_1}	{ o_1:C_1,
o_2:C_2,	o_4:C_2,		o_2:C_2,
o_3:C_3,	o_5:C_3,		o_4:C_2,
<o_2, o_1>:R_1,	<o_1, o_4>:R_1,		<o_2, o_1>:R_1,
<o_2, o_3>:R_2 }	<o_4, o_5>:R_3 }		<o_1, o_4>:R_1 }
$\mathcal{AO}_1$(before realization)	$\mathcal{AO}_2$(before realization)		
{ o_1:C_1,	{ o_1:C_1,		
o_2:C_2,	o_2:C_2,		
o_3:C_3,	o_4:C_2,		
o_4:C_2,	o_5:C_3,		
<o_2, o_1>:R_1,	<o_1, o_4>:R_1,		
<o_2, o_3>:R_2,	<o_4, o_5>:R_3,		
<o_1, o_4>:R_1 }	<o_2, o_1>:R_1 }		

Fig. 4.1. Illustration of the algorithm CHECK-INDEPENDENCE. For instance, if R_1 is a transitive role, the algorithm will detect a new inference in $\mathcal{AO}_1$ and thus determine that $\mathcal{K}_1$ and $\mathcal{K}_2$ are not independent.

Proposition 4.1. *CHECK-INDEPENDENCE is sound and complete in determining the independence of two OWL knowledge bases $\mathcal{K}_1=(\mathcal{T}, \mathcal{A}_1)$ and $\mathcal{K}_2=(\mathcal{T}, \mathcal{A}_2)$, wherein T is inconsistent, with respect to the logical entailment of any concept assertion or role assertion.*

Proof (sketch). The algorithm returns true when both knowledge bases do not overlap in individual names, which is supported by Theorem 3.1. (Lines 5-6) In case the overlap does exist, the algorithm conducts reasoning on $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively (Lines 7-8). If either ABox is found inconsistent, $\mathcal{K}_1$ and $\mathcal{K}_2$ are obviously independent since any assertion can be entailed by the single inconsistent knowledge base. If both ABoxes are consistent, the algorithm performs the realization on each of them and updates them accordingly. Next it calculates $\mathcal{O}$ and combines it with $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively (Lines 10-14). If either $\mathcal{AO}_1$ or $\mathcal{AO}_2$ is found inconsistent, we can immediately decide that $\mathcal{K}_1$ and $\mathcal{K}_2$ are not independent (with respect to any assertion). Otherwise, we check if there are any new inferences in $\mathcal{AO}_1$ and $\mathcal{AO}_2$. All the concept assertions and role assertions that have one or more individuals from I are contained in $\mathcal{O}$. Hence $\mathcal{A}_1\backslash\mathcal{O}$ and $\mathcal{A}_2\backslash\mathcal{O}$ are disjoint in individual names, and according to Theorem 3.1, $\mathcal{A}_1\backslash\mathcal{O}$ and $\mathcal{A}_2\backslash\mathcal{O}$ are independent. Therefore, if both $\mathcal{K}_1$ and $\mathcal{K}_2$ are necessary for the inference of a concept or role assertion (and thus are dependent), there have to be some new inferences of concept or role assertion in either $\mathcal{AO}_1$ or $\mathcal{AO}_2$. This can be justified by considering the inference rules described in Section 2[8]. We can assume that the reasoning is carried out by applying those rules. After the reasoning is done on $\mathcal{A}_1$ and $\mathcal{A}_2$, no more rules are applicable to each ABox. Then, by virtue of the forward chaining style of the rules, we can show that, suppose new inferences occur after combining $\mathcal{A}_1$ and $\mathcal{A}_2$, there must first be some inferences (of concept or role assertions) that involve the assertions in $\mathcal{O}$.

The advantage of the above algorithm lies in that, in case two knowledge bases are independent, it allows us to skip the reasoning on the combination of both knowledge bases by doing extra reasoning on two smaller ABoxes (i.e. $\mathcal{AO}_1$ and $\mathcal{AO}_2$ in the algorithm). This is practically useful, for example, when the reasoning performance degrades significant as the size of the knowledge base increases.

5 Related Work

Logical reasoning is usually complex and expensive. Tons of effort has been made to speed up the reasoning, especially for a large knowledge base. Among this is the research on relevance, i.e., the study of what is the relevant part of the knowledge base to a given reasoning task, e.g., a query. Most of the work is oriented towards a specific logical formalism, reasoning task (e.g. entailment, diagnosis, or abduction), and application domain. As a result, a variety of notions of (ir)relevance have emerged, under different names such as (ir)relevance, (in)dependence, irredundancy, influenceability, novelty, separability, and interactivity [6, 17, 14, 15].

[8] Although the rule set given in Section 2 is incomplete for OWL DL, it does not influence the proof here since we can assume the generation of a complete rule set via applying a similar approach to that used in Section 2.

We compare several of those notions that appeared similar to the (in)dependence we defined in this paper. Darwiche [3] and Lang et al. [14] study the notion of conditional independence between propositions. Although they use the same term, their notion of independence is different from ours. The conditional independence they consider is the logical counterpart to probabilistic independence. Therefore, informally speaking, they look at such a relationship between two sets of propositions X and Y that, given some prior information (i.e., the condition), the addition of information about X (i.e., the truth values for the propositions in X) will not lead to the conclusion of any new information about Y. Levesque [16] and Lang et al. [14] introduce a notion of formula separability, which roughly is a relationship between a set S of sentences with respect to a specific sentence α such that every member of S can entail α individually. If we apply this concept to a set S of knowledge bases with respect to the entailment of a statement a, it would become a relationship that says the members of S are separable because every one of them entails a. We believe that this is unrelated to the independence relationship we defined, but we plan to look at this more closely in the future.

In the DL literature, Tsarkov and Horrocks [22] discuss the use of the relevant information in a TBox to compute a subsumption with respect to that TBox. They define relevance as the transitive closure of the following depends relation: A concept or role expression depends on every concept or role that occurs in it, and a concept or role C depends on a concept or role D if D occurs in the definition of C. In addition, a concept C depends on every general conception inclusion in the TBox. Unlike their work, we focus on the relevance between ABoxes and ABox assertions. Therefore, both works are complementary to each other.

Elhaik and Rousset [5] work on identifying the relevant subpart of an ABox (relative to a fixed TBox) to an update, i.e., those facts that may lead to new entailments together with the newly added fact. Among other differences, their work requires all the entailments to be explicitly recorded in the ABox (they encode and store the ABox using a database), moreover, they deal with a rather restricted DL language called core-CLASSIC with the constructors $\sqcap$, $\forall$, $\geq nR$, $\leq nR$ and $\neg$ (on atomic concept only).

Amir and McIlraith's work on partition-based logical reasoning [1] provides algorithms for reasoning with partitions of related logical axioms in propositional and first-order logic. Like our work in Section 2, they are concerned with reasoning on multiple knowledge bases with overlap in content. However, they are concerned with the overlap in predicates (or propositions) while we look at the overlap in individual names.

McGuinness and Borgida's approach [18] to the explanation of DL reasoning also bases on the use of the natural deduction-style inference rules. Since we have obviously different goals, we face different issues. They work on offering understandable and efficient explanations of subsumption reasoning based on the rules while we derive the dependence relationship between OWL documents in terms of individual reasoning and try to use only a small portion of the rules without those for subsumption.

Distributed Description Logics [2] extends the formalism of DL with the ability to handle complex mappings between domains via the use of so-called bridge rules. Although its semantics is different from that of tradition DL, it could be possible that the future results of this research turn out to be useful to our work.

6 Conclusion and Future Work

Scalability is crucial for the success of the Semantic Web. Great effort has been made on the development of scalable reasoning mechanisms. In this work, we have assumed the use of the reasoner as a black box and taken a different perspective to investigate how to improve the performance of a system that requires reasoning with large scale data. We studied the (in)dependence relationships among OWL documents with respect to the logical consequence of their collections. For most of the work, we have focused on documents that commit to a common ontology, and the inference about concept assertions and role assertions. First, we described a way of identifying those documents that are necessary for the inference of a given assertion. We introduced a notion of relevant ABox assertion set to a given ABox assertion and a systematic approach to identify those sets. Secondly, we revealed two special cases in which two documents are independent of each other with respect to the entailment of assertions of specific forms: when they contain disjoint sets of individual names; and when they contain only RDF(S) statements. Finally, we introduced an algorithm for automatically detecting the independence in the general case. To the best of our knowledge, no prior work has been done to examine OWL documents (or DL knowledge bases) against the relationships of (in)dependence as defined in this paper.

We also described two applications wherein we have developed novel approaches using the above results to solve specific problems. In the first application, we exploit the dependence to help pinpoint from a repository of documents the minimal subsets that have caused the inference of a specific assertion. In the other, we harness the independence to overcome a problem in using the reasoner against large OWL instance data. Both examples demonstrate the potential use of this work in improving the scalability of a Semantic Web system which relies on the reasoning about individuals.

For future work, we intend to extend all the work to the OWL DL language, and to the case in which documents commit to different ontologies. Also, we plan to further improve the algorithm for detecting the independence. At the same time, we intend to implement the approaches in different applications wherein we will conduct empirical evaluations.

Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant No. IIS-0346963.

References

1. Amir, E. and McIlraith, S. Partition-Based Logical Reasoning for First-Order and Propositional Theories, Artificial Intelligence journal, 2003.
2. Borgida, A and Serafini, L. Distributed Description Logics - Assimilating Information from Peer Sources. Journal of Data Semantics (1), 2003.
3. Darwiche, A. A logical notion of conditional independence: properties and applications. Artificial Intelligence 97 (1-2) (1997) 45–82.

4. Dean, M. and Schreiber, G. (Eds). OWL Web Ontology Language Reference, W3C Recommendation 10 February 2004. http://www.w3.org/TR/2004/REC-owl-ref-20040210/
5. Elhaik, Q. and Rousset, M-C. Making an ABox persistent. In Proc. of the 1998 Description Logic Workshop (DL'98).
6. Greiner, R., Pearl, J., Subramanian, D. (Eds.), Artificial Intelligence 97 (1–2) (1997), Special Issue on Relevance.
7. Guo, Y., Pan, Z., and Heflin, J. An Evaluation of Knowledge Base Systems for Large OWL Datasets. In Proc. of the 3rd International Semantic Web Conference (ISWC2004).
8. Guo, Y. and Heflin, J. An Initial Investigation into Querying an Untrustworthy and Inconsistent Web. In ISWC2004 Workshop on Trust, Security and Reputation on the Semantic Web.
9. Haarslev, V. and Möller, R. Racer: A Core Inference Engine for the Semantic Web. In Workshop on Evaluation on Ontology-based Tools, ISWC2003.
10. Haarslev, V. and Möller, R. Optimization Techniques for Retrieving Resources Described in OWL/RDF Documents: First Results. In Proc. of Ninth International Conference on the Principles of Knowledge Representation and Reasoning (KR2004).
11. Haarslev, V., Möller, R., and Wessel, M. Querying the Semantic Web with Racer + nRQL. In Proc. of the Workshop on Description Logics 2004 (ADL2004).
12. Horrocks, I. The FaCT System. In Automated Reasoning with Analytic Tableaux and Related Methods International Conference (Tableaux'98).
13. Horrocks, I. and Patel-Schneider, P. F. Reducing OWL entailment to description logic satisfiability. J. of Web Semantics, 1(4):345-357, 2004.
14. Lang, J., Liberatore, P., and Marquis, P. Conditional independence in propositional logic. Artificial Intelligence Journal, Volume 141(1), October 2002, pp79–121.
15. Lang, J., Liberatore, P., and Marquis, P. Propositional independence: formula-variable independence and forgetting, Journal of Artificial Intelligence Research 18(2003) 391-443.
16. Levesque, H. A completeness result for reasoning with incomplete knowledge bases. In Proc. of KR-98, Sixth International Conference on Principles of Knowledge Representation and Reasoning, 1998.
17. Levy, A.Y., Fikes, R.E., and Sagiv, Y. Speeding up inferences using relevance reasoning: a formalism and algorithms. Artificial Intelligence 97 (1-2) (1997) 83-136.
18. McGuinness, D.L. and Borgida, A. Explaining Subsumption in Description Logics. In Proc. of the 14th International Joint Conference on Artificial Intelligence, 1995.
19. Patel-Schneider, P.F. (Eds). OWL Web Ontology Language Semantics and Abstract Syntax. http://www.w3.org/TR/owl-semantics/
20. Royer, V. and Quantz, J.J. Deriving Inference Rules for Terminological Logics. In Proc. of Logics in AI, European Workshop (JELIA'92).
21. Royer, V. and Quantz, J.J. Deriving Inference Rules for Description Logics: a Rewriting Approach into Sequent Calculi. KIT REPORT 111, Dec. 1993.
22. Tsarkov, D. and Horrocks, I. DL reasoner vs. first-order prover. In Proc. of the 2003 Description Logic Workshop (DL2003).
23. W3C RDF. Resource Description Framework (RDF). http://www.w3.org/RDF/