A New Approach of Reasoning with Inconsistent Ontologies

Jun Fang
School of Automation
Northwestern Polytechnical University
China
junfang@nwpu.edu.cn

Zhisheng Huang
Computer Science Department
Vrije Universiteit Amsterdam
The Netherlands
huang@cs.vu.nl

ABSTRACT

Reasoning with inconsistent ontologies is to use an inconsistency reasoner to get meaningful answers from inconsistent ontologies. In this paper, we propose an improved inconsistency reasoner which selects some consistent subsets by using minimal inconsistent sets(MIS) and a resolution method to improve the run-time performance of the reasoning processing. A minimal inconsistent set contains a minimal explanation for the inconsistency of an ontology. Thus, it can replace the consistency checking operation, which is executed frequently in the existing approaches. When selecting subsets of the inconsistent ontology, we select formulas which can be directly or indirectly resolved with the negation of the query formula, because only those formulas have effects on the consequences of the reasoner. In this paper, we prove that the complexity of the reasoning processing is reduced a lot. Furthermore, the experimental results show that the run-time performance of the inconsistency reasoner has been significantly improved.

1. INTRODUCTION

Reasoning with inconsistent ontologies is a very important research topic in the Semantic Web, as there exist many ontologies on the Web and the integration of the existing ontologies would lead to an inconsistency easily [1, 2, 3]. Generally, there are two ways to deal with inconsistency in ontologies. One is to pinpoint the inconsistency and repair it before reasoning [4, 5]. Another is to directly perform the reasoning task over inconsistent ontologies. The former is called *diagnosis and repair of inconsistent ontologies*, whereas the latter is called *reasoning with inconsistent ontologies*. Reasoning with inconsistent ontologies is more suitable in the Web scenarios, as we may not have the right to repair inconsistency in imported ontologies or the inconsistent ontologies may be too large to be repaired effectively. Moreover, part of ontologies may be changed too rapidly or too frequently to allow for any meaningful repair. A reasoner for reasoning with inconsistent ontologies is called *an inconsistency reasoner*.

The existing main work on reasoning with inconsistent ontologies includes paraconsistent reasoning based

on four-valued logic in [6, 7], uncertainty reasoning based on possibilistic logic in [8], and the inconsistency reasoning based on selection of consistent subset in [2, 3]. The first two methods need to extend the existing Description Logic(DL) languages, hence they can not be directly used for currently DL-based ontologies. The third one is more practical, as it can directly reason with DL-based ontologies.

Given an inconsistent ontology and a boolean query, an inconsistency reasoner[1] may use a syntactic relevance-based selection function to select a subset of the inconsistent ontology step by step (We call this inconsistency reasoner [2] the *syntax-based inconsistency reasoner* in this paper.), until the selected subset entails the query or its negative query, or the selected subset is inconsistent. The syntactic relevance-based selection function is based on in common symbols appear in the computed formulas. When the selected subset is inconsistent, the reasoner uses an *over-determined processing* (ODP) to perform a breadth-first search among the subsets with a decreasing cardinality until it finds a consistent subset. It is easy to observe that a syntax-based inconsistency reasoner needs to execute many entailment and consistency checking operations. The run-time performance of that kind of reasoner can be improved.

In this paper, we propose an approach which uses *minimal inconsistent set* (MIS) and a *resolution* method to improving the run-time performance of an inconsistency reasoner.

For an inconsistent ontology, a minimal inconsistent set is a minimal formula[2] set for explaining one inconsistency of the ontology. A minimal inconsistent set is inconsistent. Furthermore, removing any one formula from the set would make it consistent. If a subset of the original ontology is inconsistent, it must contain one minimal inconsistent set. Hence, our inconsistency reasoner firstly calculates all minimal inconsistent sets of the inconsistent ontology off-line. In the process of inconsistency reasoning, we can execute the consistency checking for one subset by determine if it contains one

---

[1]In the rest of this paper, we use inconsistency reasoner to denote the inconsistency reasoner which is based on the framework of reasoning with inconsistent ontologies in [2].

[2]The formulas in this paper mean First-order Logic formulas. Our work is built on FOL by default. It does not lose any generality as DL axioms can be transformed into FOL formulas [9].

minimal inconsistent set.

We also notice that not all syntactically relevant formulas have an effect on the entailment result of the reasoner. According to the resolution technique, only the formulas which can be directly or indirectly resolved with the negation of the boolean query formula can influence the entailment result. Two normal formula[3] ϕ, ψ can be resolved iff there exists an atomic formula A such that A is in one of these normal formulas and $\neg A$ is in the other. In the process of our inconsistency reasoning, we can select only the formulas which can be resolved with the negation of the boolean query formula. So the selected subset is smaller and more relevant to the reasoning process.

The general strategy of our inconsistency reasoner is: it firstly checks the resolubility to select a subset of the inconsistent ontology monotonically, until the subset contains all resolvably relevant formulas or one minimal inconsistent set, which means it is inconsistent. Then it removes the last added formula(s) of the minimal inconsistent set(s) from the subset to resume its consistency[4]. Finally, a standard reasoner is used to compute the standard entailment between the consistent subset and the query formula. It is easy to see that our inconsistency reasoner just needs to execute only one entailment operation on the fly, so it is more efficient, compared with the syntax-based inconsistency reasoner. The experimental results in Section 4 show that the run-time performance has been significantly improved.

This paper is organized as follows: In Section 2, we overview the existing work on inconsistency reasoners, which includes several formal definitions and the syntax-based inconsistency reasoner. In Section 3, the details of our new inconsistency reasoner by using minimal inconsistent sets and resolution method are proposed. In Section 4, we report the experiments of our methods with several inconsistent ontologies. In Section 5, we discuss related work. The conclusion and future work are presented in Section 6.

2. PREVIOUS WORK ON INCONSISTENCY REASONER

In this section, we overview the work of inconsistency reasoner proposed in [2] firstly.

2.1 Formal Definitions of Inconsistency Reasoners

The conclusions drawn from an inconsistent ontology by using the standard reasoning may be completely meaningless, because of the *explosive* problem. Thus an inconsistency reasoner uses a *nonstandard entailment* to reason with inconsistent ontologies. In the following, we will use the notation $\models$ to denote the standard entailment, and the notation $\approx$ to denote a nonstandard entailment.

---

[3]The normal formula means it is a Conjunction Normal Form(CNF) or a Disjunction Normal Form(DNF).
[4]The selected subset may contain several minimal inconsistent sets at the same time, so we may need to remove one formula from each minimal inconsistent set.

DEFINITION 1 (NONSTANDARD ENTAILMENT $\approx$). *A nonstandard entailment $\approx$ satisfies the following two requirements:*

1. *Soundness. A nonstandard entailment $\approx$ is sound if the formulas that follow from an inconsistent ontology $\mathcal{O}$ follow from a consistent subontology of $\mathcal{O}$ using classical reasoning, namely, $\mathcal{O} \approx \phi \Rightarrow \exists(\mathcal{O}' \subseteq \mathcal{O})(\mathcal{O}' \nvDash \bot \text{ and } \mathcal{O}' \vDash \phi)$.*

2. *Meaningfulness. An answer given by an inconsistency reasoner is meaningful iff it is consistent and sound. Namely, it requires not only the soundness condition, but also $\mathcal{O} \approx \phi \Rightarrow \mathcal{O} \napprox \neg\phi$. A nonstandard entailment $\approx$ is said to be meaningful iff all of the answers are meaningful.*

A nonstandard entailment uses *four valued logic* to distinguish the following four states of the answers:

DEFINITION 2 (ANSWERS OF INCONSISTENCY REASONER). *Given an inconsistent ontology $\mathcal{O}$, the inconsistency reasoner $\approx$ can provide four possible answers for a boolean query ϕ.*

1. *Over-determined: $\mathcal{O} \approx \phi$ and $\mathcal{O} \approx \neg\phi$.*

2. *Accepted: $\mathcal{O} \approx \phi$ and $\mathcal{O} \napprox \neg\phi$.*

3. *Rejected: $\mathcal{O} \napprox \phi$ and $\mathcal{O} \approx \neg\phi$.*

4. *Undetermined: $\mathcal{O} \napprox \phi$ and $\mathcal{O} \napprox \neg\phi$.*

2.2 A Syntax-based Inconsistency Reasoner

A syntax-based inconsistency reasoner uses a syntactic relevance-based selection function to determine which consistent subsets of an inconsistent ontology should be considered during its reasoning process. The process is described in Figure 1.

The direct relevance between two formulas is defined as a binary relation on formulas: $\mathcal{R} \subseteq \mathbf{L} \times \mathbf{L}$, where $\mathbf{L}$ is an ontology language. Given any direct relevance relations $\mathcal{R}$, we can extend it to a relation $\mathcal{R}^+$ on a formula and a formula set, i.e. $\mathcal{R}^+ \subseteq \mathbf{L} \times \mathcal{P}(\mathbf{L})$, as follows:

$$< \phi, \Sigma > \in \mathcal{R}^+ \text{ iff } \exists \psi \in \Sigma \text{ such that } < \phi, \psi > \in \mathcal{R}$$

In other words, a formula ϕ is relevant to a formula set Σ iff there exists a formula $\psi \in \Sigma$ such that ϕ and ψ are relevant. Two formulas ϕ, ϕ' are k-relevant with respect to a formula set Σ iff there exist formula $\psi_0, \ldots, \psi_{k+1} \in \Sigma$ such that $\phi = \psi_0, \psi_{k+1} = \phi'$ and all ψ_i and ψ_{i+1} are directly relevant.

We can use the relevance relation above to define a selection function as follows:

- $s(\Sigma, \phi, 0) = \emptyset$

- $s(\Sigma, \phi, 1) = \{\psi \in \Sigma | \phi \text{ and } \psi \text{ directly relevant}\}$

- $s(\Sigma, \phi, k) = \{\psi \in \Sigma | \psi \text{ is directly relevant to } s(\Sigma, \phi, k-1)\}$ for $k > 1$

Two formula ϕ and ψ are directly syntactically relevant iff there is a common name which appears in both formulas, i.e, $Sig(\phi) \cap Sig(\psi) \neq \emptyset$.
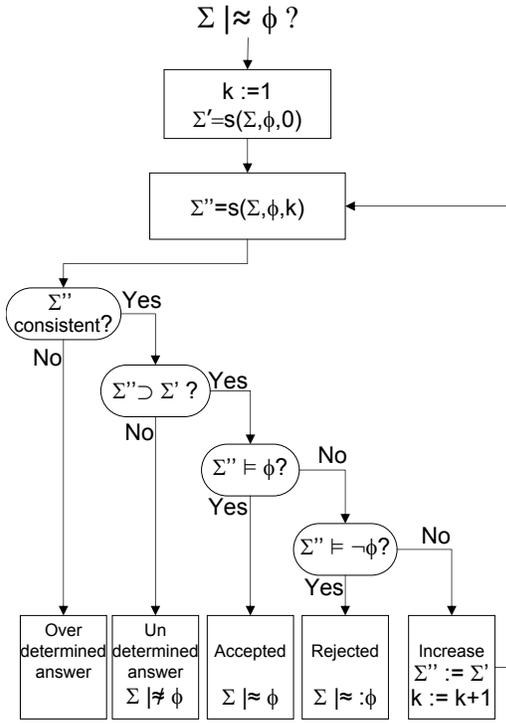
$$\Sigma \,|\!\approx\, \phi \;?$$

k :=1
$\Sigma' = s(\Sigma, \phi, 0)$

$\Sigma'' = s(\Sigma, \phi, k)$

Σ'' consistent? — Yes

No

$\Sigma'' \supset \Sigma'$? — Yes

No

$\Sigma'' \vDash \phi$? — No

Yes

$\Sigma'' \vDash \neg\phi$? — No

Yes

Over determined answer $\Sigma \,|\!\not\approx\, \phi$

Un determined answer $\Sigma \,|\!\approx\, \phi$

Accepted $\Sigma \,|\!\approx\, \phi$

Rejected $\Sigma \,|\!\approx\, :\phi$

Increase $\Sigma'' := \Sigma'$ k := k+1

Figure 1: Process of $\,|\!\approx_S$

The syntactic relevance-based selection function usually grows up to an inconsistent set rapidly, which leads to too many undetermined answers. An *Over-determined processing*(ODP) is used to reduce the number of over-determined answers, and hence improve the syntax-based inconsistency reasoner. The ODP performs a breadth-first search among the subsets of $s(\Sigma, \phi, k) - s(\Sigma, \phi, k - 1)$ with a decreasing cardinality until we find the first consistent subset. We have the following proposition about the complexity of the syntax-based inconsistency reasoner[5]:

PROPOSITION 1. *Let n be the cardinality of an ontology $\mathcal{O}$, the complexity of $\vDash$ be E, k be $n - |\Sigma'|$ which is the cardinality difference between the ontology $\mathcal{O}$ and its maximal consistent subset Σ'[6], and the complexity of the consistency checking be C.*
The complexity of the syntax-based inconsistency reasoner $\,|\!\approx_S$ is $n \times (C + 2E) + n^k \times C$.

PROOF. In every loop of the selection process, we need to perform one consistency checking operation and at most two entailment operation. In the worst case, it needs n loops, so complexity of selection process is $n \times (C + 2E)$. Since complexity of ODP is $n^k \times C$. Complexity of $\,|\!\approx_S$ is $n \times (C + 2E) + n^k \times C$. □

----

[5]Please note that syntax-based inconsistency reasoner contains the syntactically linear selection process as well as ODP.
[6]Σ' is a maximal consistent subset of the ontology $\mathcal{O}$ iff $(\Sigma' \subseteq \mathcal{O}) \wedge (\Sigma' \,|\!\not\approx\, \bot) \wedge (\forall \Sigma'' \supset \Sigma' \wedge \Sigma'' \subseteq \mathcal{O})(\Sigma'' \,|\!\approx\, \bot) \wedge \forall \phi(\Sigma' \vDash \phi \Leftrightarrow \mathcal{O} \,|\!\approx\, \phi)$

3. INCONSISTENCY REASONER BASED ON MIS AND RESOLUTION

It has been known for long time that minimal inconsistent subsets of an ontology are a cornerstone for analyzing inconsistencies. For instance, to recover the consistency, one has just to remove one formula from each minimal inconsistent subset [10].

DEFINITION 3 (MINIMAL INCONSISTENT SET(*MIS*)). *Given an inconsistent ontology $\mathcal{O}$, a formula set $\mathcal{O}'$ is a minimal inconsistent set of $\mathcal{O}$ iff it satisfies i) $\mathcal{O}' \subseteq \mathcal{O}$, ii) $\mathcal{O}' \vDash \bot$, and iii) $\forall \mathcal{O}''(\mathcal{O}'' \subset \mathcal{O}' \rightarrow \mathcal{O}'' \not\vDash \bot)$. We use $MIS(\mathcal{O})$ to denote all minimal inconsistent sets in $\mathcal{O}$, i.e., $MIS(\mathcal{O}) = \{\mathcal{O}' \subseteq \mathcal{O} | \mathcal{O}' \vDash \bot \wedge \forall \mathcal{O}''(\mathcal{O}'' \subset \mathcal{O}' \rightarrow \mathcal{O}'' \not\vDash \bot)\}$.*

In our inconsistency reasoner, we check if a formula set includes a minimal inconsistent subset to determine whether a formula set is consistent. All minimal inconsistent sets for an inconsistent ontology can be computed off-line, so execution of consistent checking in the inconsistency reasoning is very quick. Computation of all minimal inconsistent sets can be transformed into calculation of finding all justifications [11, 12], which are minimal formula sets sufficient to produce an entailment. It is intensively researched recently and some effective approaches already exist [13, 14, 15].

DEFINITION 4 (JUSTIFICATION [13]). *Let $\mathcal{O} \vDash \phi$, where ϕ is a formula and $\mathcal{O}$ is a consistent ontology. A fragment $\mathcal{O}' \subseteq \mathcal{O}$ is a justification for ϕ in $\mathcal{O}$, if $\mathcal{O}' \vDash \phi$, and $\mathcal{O}'' \not\vDash \phi$ for every $\mathcal{O}'' \subset \mathcal{O}'$. we use $JUST(\phi, \mathcal{O})$ to denote all justifications for ϕ in $\mathcal{O}$.*

In the following, we present the notion of resolvable relevance and the whole process of our inconsistency reasoning.

DEFINITION 5 (DIRECT RESOLVABLE RELEVANCE). *Two conjunctive normal formula ϕ, ψ are directly resolvably relevant iff there exists an atomic formula A such that A is in one of these normal formulas and $\neg A$ is in the other. Alternatively we call ϕ and ψ have a resolvent or ϕ and ψ can be resolved.*

DEFINITION 6 (DIRECTLY RELEVANT TO A SET). *A formula ϕ is directly resolvably relevant to a formula set Σ iff there exists a formula $\psi \in \Sigma$ such that ϕ and ψ are directly resolvably relevant.*

DEFINITION 7 (K-RELEVANCE). *Two formulas ϕ and ϕ' are resolvably k-relevant with respect to a formula set Σ iff there exist formulas $\psi_1, \ldots, \psi_k \in \Sigma$ such that ϕ and ψ_1, $res(\phi, \psi_1)$[7] and ψ_2, ..., and $res(\ldots(res(res(\phi, \psi_1), \psi_2), \ldots), \psi_k)$ and ϕ' are directly resolvably relevant.*

DEFINITION 8 (K-RELEVANCE TO A SET). *A formula ϕ is resolvably k-relevant to a formula set Σ iff there exists a formula $\psi \in \Sigma$ such that ϕ and ψ are resolvably k-relevant with respect to Σ.*

----

[7]Function $res(\phi, \psi_1)$ denotes the resolvent of the two resolvable formula ϕ and ψ_1.

Our resolution-based selection function rs is defined as follows:

- $rs(\Sigma, \phi, 0) = \emptyset$

- $rs(\Sigma, \phi, 1) = \{\psi \in \Sigma | \phi \text{ and } \psi \text{ directly resolvably relevant}\}$

- $rs(\Sigma, \phi, k) = \{\psi \in \Sigma | \psi \text{ is resolvably } k\text{-relevant to } \phi \text{ for } k > 1\}$

Given a reasoning task $\mathcal{O} \approx\!\!\!\mid \phi$?, our inconsistency reasoner $\approx\!\!\!\mid_{MR}$ uses the function rs to select the subset Σ which is relevant to the negation of the query formula, until the selected subset contains all resolvably relevant formulas or it is inconsistent, which is accomplished by checking if the subset contains one minimal inconsistent set after every selection step. When the subset Σ is inconsistent, we remove the last added formula(s) of the minimal inconsistent set(s) from the subset. Finally, we perform $\Sigma \models \phi$ and $\Sigma \models \neg\phi$ to obtain the answer. The whole process for processing $\mathcal{O} \approx\!\!\!\mid_{MR} \phi$? is described in Algorithm 1, where all minimal inconsistent set $MIS(\mathcal{O}) = \{m_1, \ldots, m_i, \ldots\}$.

Algorithm 1 Computing $\mathcal{O} \approx\!\!\!\mid_{MR} \phi$

1: $rs(\Sigma, \neg\phi, 0) := \emptyset$
2: $k := 1$
3: **while** $rs(\Sigma, \neg\phi, k) \supset rs(\Sigma, \neg\phi, k - 1)$ **do**
4:   **if** Σ contains one minimal inconsistent set **then**
5:     remove one formula in $(rs(\Sigma, \neg\phi, k) \setminus rs(\Sigma, \neg\phi, k - 1)) \cap m_i$ from Σ for each $m_i \subseteq \Sigma$;
6:     **break**;
7:   **end if**
8:   $k := k + 1$
9: **end while**
10: **if** $\Sigma \models \phi$ **then**
11:   **return** *accepted*;
12: **else if** $\Sigma \models \neg\phi$ **then**
13:   **return** *rejected*;
14: **else**
15:   **return** *undetermined*;
16: **end if**

PROPOSITION 2. *The inconsistent reasoner $\approx\!\!\!\mid_{MR}$ satisfies the following properties:*
(a) never over-determined,
(b) may be undetermined,
(c) always sound,
(d) always meaningful.

If an consistent formula set S entails a formula ϕ, then $\neg\phi \cup S$ is inconsistent. To conclude the inconsistency, we can perform resolution among $\neg\phi$ and other formulas which are resolvably relevant to it, finally we can obtain the $\bot$. Hence, whether an ontology entails a boolean query only depends on formulas which are resolvably relevant to negation of it, our reasoner $\approx\!\!\!\mid_{MR}$ can inconsistently entail all formulas entailed by the syntax-based reasoner $\approx\!\!\!\mid_S$.

PROPOSITION 3. *Given an inconsistent ontology $\mathcal{O}$.*
$\forall\phi(\mathcal{O} \approx\!\!\!\mid_S \phi \Rightarrow \mathcal{O} \approx\!\!\!\mid_{MR} \phi)$.

Please note that the opposite direction is not always satisfied, i.e., $\exists\phi(\mathcal{O} \approx\!\!\!\mid_{MR} \phi \not\Rightarrow \mathcal{O} \approx\!\!\!\mid_S \phi)$. For instance, let the inconsistent ontology[8] $\mathcal{O}$ be $\{A \sqsubseteq B, B \sqsubseteq C, A \sqsubseteq \neg B, C \sqsubseteq D, D \sqsubseteq E\}$, and the query formula ϕ be $B \sqsubseteq E$.

Using the syntax-based reasoner, it selects $s(\Sigma, \neg\phi, 1) = \{A \sqsubseteq B, B \sqsubseteq C, A \sqsubseteq \neg B, D \sqsubseteq E\}$, it is inconsistent and then calls an ODP. The ODP returns a consistent subset $S_1 := \{A \sqsubseteq B, B \sqsubseteq C, D \sqsubseteq E\}$ and performs the classical reasoning. Finally it returns $S_1 \not\models \phi$. Hence, $\mathcal{O} \not\approx\!\!\!\mid_S \phi$.

Using our inconsistency reasoner $\approx\!\!\!\mid_{MR}$, it selects $rs(\Sigma, \neg\phi, 1) = \{A \sqsubseteq \neg B, B \sqsubseteq C, D \sqsubseteq E\}$, $rs(\Sigma, \neg\phi, 2) = \mathcal{O}$. It is inconsistent as it contain the minimal inconsistent set $\{A \sqsubseteq B, A \sqsubseteq \neg B\}$; then the ODP removes $A \sqsubseteq B$ which is a lastly added formula of the minimal inconsistent set from $S2$ to obtain $S_2 := \{A \sqsubseteq \neg B, B \sqsubseteq C, C \sqsubseteq D, D \sqsubseteq E\}$. As $S_2 \models \phi$, $\mathcal{O} \approx\!\!\!\mid_{MR} \phi$.

Based on Algorithm 1, the complexity of $\approx\!\!\!\mid_{MR}$ decreases a lot, compared with the complexity of $\approx\!\!\!\mid_S$.

PROPOSITION 4   (COMPLEXITY OF $\approx\!\!\!\mid_{MR}$). [9] *Let n be the cardinality of an ontology $\mathcal{O}$, m be the cardinality of $MIS(\mathcal{O})$, the complexity of inclusion checking between two sets be I, and the complexity of $\models$ be E. The complexity of inconsistency reasoner $\approx\!\!\!\mid_{MR}$ is $n \times m \times I + 2E$.*

PROOF. In the algorithm 1, the selection process (from step1 to step9) needs n loops in the worst case. In every loop, we need to check if the selected subset contain one minimal inconsistent set, it is achieved by traversing all minimal inconsistent sets. Hence the complexity for the selection process is $n \times m \times I$. It is easy to observe that we just need to perform at most two entailment operation $\models$ in rest steps. Thus, complexity of $\approx\!\!\!\mid_{MR}$ is $n \times m \times I + 2E$. $\square$

4. IMPLEMENTATION AND EVALUATION

We have implemented the prototype of $\approx\!\!\!\mid_{MR}$ by using Pellet[10] and OWLAPI[11]. It is integrated into PION[12] to compute the nonstandard entailment with DL inconsistent ontologies.

The computation of query answers for DL-based ontologies is similar with that for FOL. However, there is the following difference. DL has two kinds of inconsistencies. One is the classical inconsistency, i.e., the ontology has no interpretation, and another is *incoherence* [16], i.e., there exists an unsatisfiable named concept in the ontology. Hence, when checking inconsistency in the process of $\approx\!\!\!\mid_S$, we need to consider these two inconsistencies. When calculating minimal inconsistent sets, we

---

[8]In this example, we use a DL ontology which contains an unsatisfiable concept, as it is more common in the Semantic Web situation.
[9]Please note that complexity of $\approx\!\!\!\mid_{MR}$ does not contain complexity of calculation of all minimal inconsistent sets, which is finished off-line.
[10]http://clarkparsia.com/pellet/
[11]http://owlapi.sourceforge.net/
[12]http://wasp.cs.vu.nl/sekt/pion/

Table 1: Information about ontologies

Ontology	Syntax	#Cons	#Roles	#Inds	#Axioms	#Unsat. Cons
MadCow	$\mathcal{ALCHOIN}(\mathbf{D})$	55	16	14	89	1
Economy	$\mathcal{ALCH}(\mathbf{D})$	338	45	481	1609	51
Transportation	$\mathcal{ALCH}(\mathbf{D})$	447	89	183	1340	62

Table 2: Run-time performance evaluation of inconsistency reasoner $\models\approx_S$

Ontology	Max execution time	Min execution time	Average execution time
MadCow	0.93s	0.024s	0.14s
Economy	15.85s	0.072s	2.25s
Transportation	13.34s	0.071s	1.71s

need to extend MIS to include all minimal incoherent sets[13].

In addition, as DL axiom doesn't have negation form and normal form, we translate DL axiom into semantically equivalent FOL formula to calculate its negation form or normal form. Two DL axioms are resolvably relevant iff their corresponding FOL formulas are resolvably relevant.

We test the prototype of $\models\approx_{MR}$ by applying it on several inconsistent ontologies. Those inconsistent ontologies are the MadCow ontology, the Economy ontology and the Transportation ontology, which are selected from the TONES ontology repositories[14]. Information of these ontologies are described as table 1.

The testing axiom dataset is constructed by randomly generating 1000, 10000 and 10000 concept inclusion axioms ($\phi := C \sqsubseteq D$ and $\phi \notin \mathcal{O}$) from the MadCow ontology, Economy ontology and Transportation ontology. All minimal inconsistent sets for these inconsistent ontologies are computed off-line by using the explanation method in Pellet[15]. All the following experiments have been carried out on an ordinary PC (2.60 GHz Pentium-4 processor and 2GB of physical memory) with the default Java memory setting.

Table 2 describes the run-time performance of syntax-based inconsistency reasoner $\models\approx_S$, and Table 3 illustrates the run-time performance of our improved inconsistency reasoner $\models\approx_{MR}$. For every inconsistent ontology, the average execution time of inconsistency reasoner is computed by dividing the whole execution time for all generated axioms by the total number of the axioms. Maximal execution time, minimal execution time and average execution time of $\models\approx_S$ and $\models\approx_{MR}$ are shown in the second column to the forth column in Table 2 and Table 3 separately.

The results show that the run-time performance of the improved inconsistency reasoner $\models\approx_{MR}$ increases a lot when comparing with $\models\approx_S$, especially for inconsistent ontologies which have many unsatisfied concepts. For

---

[13]The definition of the minimal incoherent set is similar to the definition of minimal *unsatisfiability-preserving sub-ontologies*(MUPS) [4]. An axiom set $\mathcal{O}'$ is a minimal incoherent set of an incoherent ontology $\mathcal{O}$ iff it satisfies i) $\mathcal{O}' \subseteq \mathcal{O}$, ii) $\mathcal{O}'$ is incoherent and iii) $\forall \mathcal{O}''(\mathcal{O}'' \subset \mathcal{O}' \rightarrow \mathcal{O}''$ is coherent).

[14]http://owl.cs.manchester.ac.uk/repository/

[15]It uses method of org.mindswap.pellet.KnowledgeBase .getExplanation().

each inconsistent ontology, although minimal execution time of the inconsistent reasoner $\models\approx_{MR}$ is close to that of the syntax-based inconsistency reasoner $\models\approx_S$, maximal execution time of the inconsistency reasoner $\models\approx_{MR}$ is $\frac{1}{10}, \frac{1}{99}$ and $\frac{1}{49}$ of that of the syntax-based inconsistency reasoner $\models\approx_S$; and average execution time of the inconsistency reasoner $\models\approx_{MR}$ is $\frac{1}{3}, \frac{1}{20}$ and $\frac{1}{15}$ that of the syntax-based inconsistency reasoner $\models\approx_S$.

The reason that minimal execution time of the two inconsistent reasoners are almost equivalent is that the syntax-based inconsistency reasoner $\models\approx_S$ may accept or reject the query in the first several selected subsets. The reason for the significant improvement of run-time performance of the inconsistency reasoner is that $\models\approx_S$ needs many consistency checking and entailment operations, while $\models\approx_{MR}$ needs only one entailment operation. The experimental results coincide with the complexity comparison between the two inconsistency reasoners.

5. RELATED WORK

There are many existing work for reasoning with inconsistent ontologies for its importance. One major work of reasoning with inconsistent ontologies is based on paraconsistent approaches [6, 7, 17]. The main idea is to extend the existing ontology languages to *four-valued logics*, which uses two additional truth values which stand for *underdefined* (i.e neither true nor false) and *overdefined* (or *contradictory*, i.e. both true and false). The method given in [2, 3] firstly selects a consistent subontology based on a selection function, which is defined on the syntactic [2] or general semantic [3] relevance, then reasons over the selected subontology. In [8], the authors introduce the *possibilistics* in DLs by exploiting uncertainty degreee on DL axioms. The method given in [18] applies *lexicographic inference* to DLs, it views a DL-based ontology as a set of axioms with priority, reasoning consequences are inferred from every consistent subontology having the highest lexicographic precedence. In [19], the authors propose a framework for reasoning with inconsistent DL ontologies by expressing DL inconsistent ontologies as *Defeasible Logic Programs* (DeLP).

6. CONCLUSION AND FUTURE WORK

In order to improve the run-time performance of an inconsistency reasoner, we have presented an improved inconsistency reasoner $\models\approx_{MR}$ based on minimal incon-

Table 3: Run-time performance evaluation of inconsistency reasoner $\approx_{MR}$

Ontology	Max execution time	Min execution time	Average execution time
MadCow	0.091s	0.024s	0.037s
Economy	0.16s	0.063s	0.11s
Transportation	0.27s	0.071s	0.11s

sistent sets and a resolution method in this paper. We have proved that the inconsistency reasoner $\approx_{MR}$ can preserve all consequences of the syntax-based inconsistency reasoner $\approx_S$, and the complexity is reduced a lot. The experimental results on several inconsistent ontologies show that the run-time performance of $\approx_{MR}$ has been significantly improved. In the future, we plan to test the improved inconsistency reasoner on more complex and large scale ontologies.

Acknowledgments

7. REFERENCES

[1] Hameed, A., Preece, A., Sleeman, D.: Ontology Reconciliation. In Staab, S., Studer, R., eds.: Handbook on Ontologies in Information Systems, Springer Verlag (2003) 231–250

[2] Huang, Z., van Harmelen, F., ten Teije, A.: Reasoning with inconsistent ontologies. In: Proceedings of the International Joint Conference on Artificial Intelligence - IJCAI'05. (2005) 454–459

[3] Huang, Z., van Harmelen, F.: Using Semantic Distances for Reasoning with Inconsistent Ontologies. In: Proceedings of the International Semantic Web Conference 2008 (ISWC08), Lecture Notes in Computer Science 5318, Springer (2008) 178–194

[4] Schlobach, S., Cornet, R.: Non-Standard Reasoning Services for the Debugging of Description Logic Terminologies. In: Proceedings of the eighteenth International Joint Conference on Artificial Intelligence, IJCAI'03. (2003) 355–362

[5] Qi, G., Wang, Y., Haase, P., Hitzler, P.: A Forgetting-based Approach for Reasoning with Inconsistent Distributed Ontologies. In: Proceedings of the International Workshop WORM-08, Ontologies: Reasoning and Modularity, at the 5th European Semantic Web Conference. (2008)

[6] Ma, Y., Hitzler, P., Lin, Z.: Algorithms for Paraconsistent Reasoning with OWL. In: Proceedings of the 4th European Semantic Web Conference. (2007) 399–413

[7] Ma, Y., Hitzler, P.: Paraconsistent Reasoning for OWL 2. In: Proceedings of the 3rd International Conference on Web Reasoning and Rule Systems. (2009) 197–211

[8] Qi, G., Pan, J.Z., Ji, Q.: Extending Description Logics with Uncertainty Reasoning in Possibilistic Logic. In: the Proc. of the 9th European Conference on Symbolic and Quantitave Approaches to Reasoning with Uncertainty (ECSQARU'2007). (2007) 828–839

[9] Motik, B., Sattler, U., Studer, R.: Query Answering for OWL-DL with Rules. In: Journal of Web Semantics, Springer (2004) 549–563

[10] Reiter, R.: A Theory of Diagnosis from First Principles. Artif. Intell. **32**(1) (1987) 57–95

[11] Horridge, M., Parsia, B., Sattler, U.: Explaining Inconsistencies in OWL Ontologies. In: 3rd International Conference on Scalable Uncertainty Management (SUM 2009). Volume 5785 of Lecture Notes in Computer Science., Springer (2009) 124–137

[12] Guilin Qi, Peter Haase, Q.J.: D1.2.3 Diagnosing and repairing inconsistent networked ontologies. In: NeOn Project Deliverable D1.2.3. (2008)

[13] Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding all justifications of owl dl entailments. In: Proceedings of the 6th International Semantic Web Conference. Volume 4825., Springer-LNCS (2007) 267–280

[14] Suntisrivaraporn, B., Qi, G., Ji, Q., Haase, P.: A Modularization-Based Approach to Finding All Justifications for OWL DL Entailments. In: ASWC. (2008) 1–15

[15] Du, J., Qi, G., Ji, Q.: Goal-Directed Module Extraction for Explaining OWL DL Entailments. In: International Semantic Web Conference. (2009) 163–179

[16] Flouris, G., Huang, Z., Pan, J.Z., Plexousakis, D., Wache, H.: Inconsistencies, Negations and Changes in Ontologies. In: Proc. of the 21st National Conference on Artificial Intelligence(AAAI-06). (2006) 1295–1300

[17] Zhang, X., Lin, Z., Wang, K.: Towards a Paradoxical Description Logic for the Semantic Web. In: Sixth International Symposium on Foundations of Information and Knowledge Systems. (2010) 306–325

[18] Du, J., Qi, G., Shen, Y.D.: Lexicographical Inference over Inconsistent DL-Based Ontologies. In: Proceedings of the 2nd International Conference on Web Reasoning and Rule Systems. (2008) 58–73

[19] Gomez, S.A., Chesnevar, C.I., Simari, G.R.: An Argumentative Approach to Reasoning with Inconsistent Ontologies. In Meyer, T., Orgun, M.A., eds.: Knowledge Representation Ontology Workshop (KROW 2008). Volume 90 of CRPIT., Sydney, Australia, ACS (2008) 11–20