

A Method of Contrastive Reasoning with Inconsistent Ontologies

Jun Fang¹, Zhisheng Huang², and Frank van Harmelen²

¹ School of Automation, Northwestern Polytechnical University, China
junfang@nwpu.edu.cn

² Department of Computer Science, Vrije Universiteit Amsterdam, The Netherlands
{huang,Frank.van.Harmelen}@cs.vu.nl

Abstract. Contrastive reasoning is the reasoning with contrasts which are expressed as contrary conjunctions like the word "but" in natural language. Contrastive answers are more informative for reasoning with inconsistent ontologies, as compared with the usual simple Boolean answer, i.e., either "yes" or "no". In this paper, we propose a method of computing contrastive answers from inconsistent ontologies. The proposed approach has been implemented in the system CRION (Contrastive Reasoning with Inconsistent ONtologies) as a reasoning plug-in in the LarKC (Large Knowledge Collider) platform. We report several experiments in which we apply the CRION system to some realistic ontologies. This evaluation shows that contrastive reasoning is a useful extension to the existing approaches of reasoning with inconsistent ontologies.

1 Introduction

1.1 Motivation

Contrastive reasoning is the reasoning with contrasts which are expressed as contrary conjunctions like the word "but" in natural language [1, 2, 3]. For instance, in real life, one would say that "all cars are polluting, *but* hybrid cars are not polluting"; or one would say that "The conference will be held in Holland, *but* not in Amsterdam". The first example expresses an exception that contradicts a general rule; the second example is contrary to a general expectation that one may have (namely that conferences in Holland are generally held in Amsterdam).

There exist some previous works on contrastive reasoning [1, 2, 3]. These previous works consider contrastive reasoning as a supplement for non-monotonic reasoning: they use non-monotonic reasoning to compute the all implications, and then determine the contrasts among them. Compared with normal Boolean query answering, contrastive reasoning gives users not only an answer to the original query, but also some *contrastive answers*.

Such contrastive reasoning has two main goals:

- *Avoidance of misleading information by extending the answer.* Contrastive answers provide not only an answer to the original query, but also some relevant contrasting answers. In our introductory example, the simple answer that all cars are polluting is misleading because hybrid cars are an exception to this rule.
- *Effective influence with surprising answers.* A psychologically effective influence can be achieved by providing an additional answer that is unexpected. In our introductory conference example, the contrastive answer that the conference is not in Amsterdam is surprising against the background expectation that all conferences in Holland will be in Amsterdam.

Contrastive reasoning exposes the contradiction that exists either between a knowledge base and external expectations (as in the conference example), or contradictions between different parts of the knowledge base (as in the polluting cars example). Because of this, contrastive reasoning is also very useful for reasoning with inconsistent ontologies, because it does not simply respond to queries with a Boolean answer of either “yes” or “no”, but also provides an informative answer with some “surprising” information.

Reasoning with inconsistent ontologies is a particularly important research topic in the Semantic Web for several reasons: i) Integration of existing ontologies easily leads to an inconsistency. ii) It may be ineffective or even impossible to repair inconsistencies before reasoning as the inconsistent ontologies may be too large or we may not have the right to repair inconsistencies in imported ontologies. iii) Ontologies may change at a high frequency and hence do not allow for any meaningful repair. In this paper, we will therefore focus on reasoning with inconsistent ontologies.

1.2 Simple Example

We consider a fragment of the well known MadCow ontology shown in Table 1¹, in which *MadCow* is defined as a *Cow* which eats brains of *Sheep* and *Cow* is defined as a *Vegetarian*, which leads to an inconsistency in the ontology.

Table 1. Fragment of the MadCow ontology

$Cow \sqsubseteq Vegetarian$	$MadCow(the_MadCow)$
$MadCow \sqsubseteq Cow \sqcap \exists eat.((\exists partof.Sheep) \sqcap Brain)$	
$Sheep \sqsubseteq Animal$	$Vegetarian \sqsubseteq \forall eat.\neg Animal$
$Vegetarian \sqsubseteq Animal \sqcap \forall eat.\neg(\exists partof.Animal)$	

When we ask “Is Cow a Vegetarian?”, the current methods will answer “yes”. However, using contrastive reasoning, the answer “yes, *but* MadCow is not a Vegetarian” is more informative. The latter answer has a touch of contrast (or surprise), which would provide more instructive information for users.

¹ We add *MadCow(the_MadCow)* to expose the inconsistency.

1.3 Structure and Contributions of This Paper

We have presented the initial framework of contrastive reasoning in [4]. In this paper, we propose a method of computing contrastive answers from inconsistent ontologies. We introduce contrastive reasoning in the general setting of First-order Logic (FOL). The proposed approach has been implemented in the system CRION as a reasoning plug-in in the LarkC platform². We will report our experiments of applying the proposed approach to some realistic ontologies. The experiments show that contrastive reasoning is a useful form of reasoning with inconsistent ontologies.

Summarizing, the main contributions of this paper are (1) a general approach of contrastive reasoning; (2) a method of computing contrastive answers; (3) the implementation of the CRION system that computes contrastive answers; and (4) evaluation of CRION, using human subjects to score the effectiveness of contrastive answers to queries.

This paper is organized as follows: Section 2 presents a general approach of contrastive reasoning with inconsistent ontologies. Section 3 explores how to compute contrastive answers in FOL. Section 4 discusses the implementation of CRION, reports the experiments of CRION with several inconsistent ontologies and presents the evaluation of CRION. After a discussion of related work in Section 5, the last section includes conclusions and future work.

2 Formalization of Contrastive Reasoning

2.1 Nonstandard Entailment for Inconsistent ontologies

The classical entailment in logics is *explosive*: any formula is a logical consequence of a contradiction. Therefore, conclusions drawn from an inconsistent knowledge base by classical inference may be completely meaningless. The general task of any system that reasons with inconsistent ontologies is: given an inconsistent ontology, return *meaningful* answers to queries. In [5], a general framework of reasoning with inconsistent ontologies has been developed. In that framework, an answer is “meaningful” if it is supported by a selected consistent subset of the inconsistent ontology, while its negation is not supported by the selected subset. PION is a system for reasoning with inconsistent ontologies, which can return such meaningful answers [5]. In the following, we will use the notation \models to denote the standard entailment, and the notation \approx to denote a nonstandard entailment.

Definition 1 (Nonstandard Entailment \approx [5]). *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 subset of \mathcal{O} using classical reasoning: $\mathcal{O} \approx \alpha \Rightarrow \exists(\mathcal{O}' \subseteq \mathcal{O})(\mathcal{O}' \not\models \perp \text{ and } \mathcal{O}' \models \alpha)$.*

² <http://www.larkc.eu>

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 $\mathcal{O} \models \alpha \Rightarrow \mathcal{O} \not\models \neg\alpha$. A nonstandard entailment \models is said to be meaningful iff all of the answers are meaningful.

Properties of \models are similar to those of the standard entailment \models . However, there is an important exception. Given an inconsistent \mathcal{O} and two formulas α and β with $\mathcal{O} \models \alpha$ and $\mathcal{O} \models \beta$, we cannot always conclude $\mathcal{O} \models \alpha \wedge \beta$. One reason for it is that the selected subset that supports $\mathcal{O} \models \alpha$ may differ from the selected subset that supports $\mathcal{O} \models \beta$, while the union of the two subsets may be inconsistent; another reason is that $\alpha \wedge \beta$ may be a contradiction.

2.2 Contrastive Answers

Using the previous definition of nonstandard entailment, we can now define our central notion of *contrastive answers*. Informally, a contrastive answer contains three parts:

- **Original formula.** A formula which answers the original query³;
- **Contrastive formula.** A formula which contrasts with the original answer formula;
- **Clarification formula** A formula that explains the reason why the contradiction occurs. The clarification formula need not (but may) be implied by the ontology. In some application scenarios, the clarification formulas may be omitted in the query answer if the user does not require an explanation of contrastive answers.

In the MadCow example, when considering the query “Is Cow a Vegetarian?”, “Cow is a Vegetarian” is the original answer to the query, “MadCow is not a Vegetarian” is a contrastive formula, while “the_MadCow is a MadCow and MadCow is a Cow” is a clarification formula which explains why “Cow is a Vegetarian” and “MadCow is not a Vegetarian” are contrastive. This leads us to the formal definition of contrastive answers⁴:

Definition 2 (Contrastive Answer). *Given an inconsistent ontology \mathcal{O} , a contrastive answer $\mathcal{O} \models \alpha$ but γ although β contains the following parts: an original formula α , a contrastive formula γ , and a clarification formula β , such*

³ Note that formulas in this paper mean First-order Logic formulas. Our work is built on FOL. Without loss of generality, a Description Logic axiom can be transformed into a (conjunctive) FOL formula. Thus, in the following, we will consider only a single formula.

⁴ In this paper, we focus on the approach of reasoning with inconsistent ontologies, in which a clarification formula is derivable from the ontology. We leave the cases of the clarification formula as an expectation (like that in the conference example) for future work.

that: $\mathcal{O} \models \alpha$, $\mathcal{O} \models \beta$ and $\mathcal{O} \models \gamma$, $\alpha \wedge \beta$ is not a contradiction, $\gamma \wedge \beta$ is not a contradiction, but $\alpha \wedge \beta \wedge \gamma$ is a contradiction⁵.

Sometimes it is not necessary to state the clarification formula explicitly in a contrastive answer. That leads to the following definition.

Definition 3 (Contrastive Answer without Explanation). *Given an inconsistent ontology \mathcal{O} , $\mathcal{O} \models \alpha$ **but** γ is a contrastive answer without explanation if there exists a formula β such that $\mathcal{O} \models \alpha$ **but** γ **although** β is a contrastive answer.*

The definitions above imply that contrastive answers have a nice exchange property. Namely, more contrastive answers can be obtained by exchanging the original formula, the contrastive formula and the clarification formula.

For instance, in the MadCow example, “Cow is a Vegetarian”, but “Mad-Cow is not a Vegetarian”, although “the_MadCow is a MadCow and MadCow is a Cow” is a contrastive answer. It is easy to observe that the symmetric answers such as “Madcows is not a vegetarian”, but “the_MadCow is a Mad-Cow and MadCow is a Cow”, although “Cow is a vegetarian” are contrastive answers.

Proposition 1 (Exchange Property of Contrastive Answers). *For an inconsistent ontology \mathcal{O} and three formulas α , β , γ , the following hold:*

- Exchange: $\mathcal{O} \models \alpha$ **but** γ **although** $\beta \Rightarrow \mathcal{O} \models \gamma$ **but** α **although** β
- Conditional Lifting: $\mathcal{O} \models \alpha$ **but** γ **although** β and $\alpha \wedge \gamma$ is not a contradiction $\Rightarrow \mathcal{O} \models \beta$ **but** γ **although** α
- Conditional Shifting: $\mathcal{O} \models \alpha$ **but** γ **although** β and $\alpha \wedge \gamma$ is not a contradiction $\Rightarrow \mathcal{O} \models \alpha$ **but** β **although** γ

Proof. It can be easily proved by using the definition of contrastive answer. 1) If $\mathcal{O} \models \alpha$ **but** γ **although** β , then $\mathcal{O} \models \alpha$ and $\mathcal{O} \models \beta$ and $\mathcal{O} \models \gamma$ and $\alpha \wedge \beta$ is not a contradiction and $\gamma \wedge \beta$ is not a contradiction and $\alpha \wedge \beta \wedge \gamma$ is a contradiction, according to definition 2, $\mathcal{O} \models \gamma$ **but** α **although** β . Conditional symmetry properties are proved in a similar way.

These exchange properties do not mean that α , β and γ do not differ in any way. Although they can be formally interchanged in the above way, such an interchange implies a change in the epistemological status of the formula: The original formula α is the answer to the original query. Thus, it is considered to be the most important one. The contrastive formula γ is an additional answer. The clarification formula β provides some information to explain the reason why the contradiction occurs, which may be ignored if an explanation is not necessary.

⁵ A formula is a contradiction iff there does not exist a model which can satisfy the formula, an ontology is inconsistent iff there does not exist a model which can satisfy all formulas in the ontology.

Besides the exchange property above, contrastive answers also have the expansion property: the formula α in the contrastive answer can be expanded with other formula α' if the conjunction $\alpha \wedge \alpha'$ is an answer of \models .

Proposition 2 (Expansion Property of Contrastive Answers). *For an inconsistent ontology \mathcal{O} , and three formulas α, β, γ , the following hold:*

- $\mathcal{O} \models \alpha$ **but** γ **although** β and $\alpha \wedge \alpha' \wedge \beta$ is not a contradiction and $\mathcal{O} \models \alpha \wedge \alpha' \Rightarrow \mathcal{O} \models \alpha \wedge \alpha'$ **but** γ **although** β
- $\mathcal{O} \models \alpha$ **but** γ **although** β and $\alpha \wedge \beta \wedge \beta'$ is not a contradiction and $\beta \wedge \beta' \wedge \gamma$ is not a contradiction and $\mathcal{O} \models \beta \wedge \beta' \Rightarrow \mathcal{O} \models \alpha$ **but** γ **although** $\beta \wedge \beta'$
- $\mathcal{O} \models \alpha$ **but** γ **although** β and $\beta \wedge \gamma \wedge \gamma'$ is not a contradiction and $\mathcal{O} \models \gamma \wedge \gamma' \Rightarrow \mathcal{O} \models \alpha$ **but** $\gamma \wedge \gamma'$ **although** β

Proof 1. *If $\mathcal{O} \models \alpha$ **but** γ **although** β , then $\mathcal{O} \models \alpha$ and $\mathcal{O} \models \beta$ and $\mathcal{O} \models \gamma$ and $\alpha \wedge \beta \wedge \gamma$ is a contradiction, so $\alpha \wedge \alpha' \wedge \beta \wedge \gamma$ is also a contradiction. Furthermore, $\alpha \wedge \alpha' \wedge \beta$ is not contradictions. Since $\mathcal{O} \models \alpha \wedge \alpha'$, according to Definition 2, $\mathcal{O} \models \alpha \wedge \alpha'$ **but** γ **although** β . Other situations can be proved similarly.*

3 Computing Contrastive Answers

In this section, we will propose a method of obtaining contrastive answers. Given an inconsistent ontology \mathcal{O} and a closed FOL formula α , if $\mathcal{O} \models \alpha$, how can we obtain related contrastive answers $\mathcal{O} \models \alpha$ **but** γ **although** β .

Our approach of computing contrastive answers is an extension to the method for reasoning with inconsistent ontologies proposed in [5]. To make this paper self contained, we will first give a brief overview of the general approach of reasoning with inconsistent ontologies, which is developed in [5, 6].

3.1 The PION Approach

Selection functions are central in the PION approach of reasoning with inconsistent ontologies. It is used to determine which consistent subsets of an inconsistent ontology should be considered during the reasoning process. The selection function can either be syntactic, e.g. using a syntactic relevance measure, or can be based on semantic relevance, such as using the co-occurrence of terms in search engines like Google [7].

Given an ontology (i.e., a formula set) \mathcal{O} and a query α , a selection function s returns a subset of \mathcal{O} at each step $k > 0$. Let \mathbf{L} be the ontology language, which is denoted as a formula set. A selection function s is then a mapping $s : \mathcal{P}(\mathbf{L}) \times \mathbf{L} \times N \rightarrow \mathcal{P}(\mathbf{L})$ such that $s(\mathcal{O}, \alpha, k) \subseteq \mathcal{O}$.

A formula ϕ is syntactic relevant to a formula set Σ iff there exists a formula $\psi \in \Sigma$ such that ϕ and ψ are directly relevant. We can use the relevance relation above to define a selection function as follows:

- $s(\Sigma, \phi, 0) = \emptyset$
- $s(\Sigma, \phi, 1) = \{\psi \in \Sigma \mid \phi \text{ and } \psi \text{ directly relevant}\}$
- $s(\Sigma, \phi, k) = \{\psi \in \Sigma \mid \psi \text{ is directly relevant to } s(\Sigma, \phi, k-1)\}$ for $k > 1$

In this paper, we use the syntactic method [5] to measure relevance between formulas. Two formula ϕ and ψ are directly syntactically relevant iff there is a common name which appears in both formulas. Although the syntactic-relevance-based selection function is specific and seems to be simple, the experiments show that even this simple selection function can obtain intuitive results in most cases for reasoning with inconsistent ontologies [5]. Furthermore, our approach of contrastive reasoning is independent of any specific selection function, because the syntactic-relevance-based selection function can be replaced with any other kinds of selection functions, like one with Normalized Google Distance [7].

The general strategy for reasoning with inconsistent ontologies is: given the syntactic selection function, we select a consistent subset from an inconsistent ontology. Then we apply standard reasoning on the selected subset to find meaningful answers. If a satisfying answer cannot be found, we use the selection function to extend the selected set for further reasoning. If an inconsistent subset is selected, we apply “over-determined processing” (ODP) [5]. One of the ODP strategies is to find a maximal consistent subset of the selected set. If the (firstly selected) maximal consistent subset entails the query, the algorithm will return ‘yes’, otherwise it will return ‘no’. A linear extension strategy with ODP for the evaluation of a query ‘ $\mathcal{O} \models \alpha$?’ is described in Algorithm 1.

Algorithm 1. Linear extension strategy for evaluating $\mathcal{O} \models \alpha$

```

1:  $\Omega := \emptyset$ 
2:  $k := 0$ 
3: repeat
4:    $k := k + 1$ 
5:    $\Omega' := s(\mathcal{O}, \alpha, k)$ 
6:   if  $\Omega' \subseteq \Omega$  then
7:     return  $\mathcal{O} \not\models \alpha$ 
8:   end if
9:   if  $\Omega'$  inconsistent then
10:     $\Omega'' := \text{maximal\_consistent\_subontology}(\Omega')$ 
11:    if  $\Omega'' \models \alpha$  then
12:      return  $\mathcal{O} \models \alpha$ 
13:    else
14:      return  $\mathcal{O} \not\models \alpha$ 
15:    end if
16:   end if
17:    $\Omega := \Omega'$ 
18: until  $\Omega' \models \alpha$ 
19: return  $\mathcal{O} \models \alpha$ 

```

3.2 The CRION Approach

In the following, we propose an algorithm for obtaining contrastive answers, based on the PION approach described above. From the definition of contrastive answers, the conjunction of the original formula α , the contrastive formula γ , and the clarification formula β must lead to a contradiction, i.e., $\{\alpha, \beta, \gamma\} \models \perp$. That means that, given an original answer α which is obtained by using the PION approach, we can try to obtain the contrastive formula and the clarification formula, by considering a *minimal inconsistent set* which contains α . A minimal inconsistent set is a minimal formula set that explains the inconsistency of an inconsistent ontology.

Definition 4 (Minimal Inconsistent Set (MIS)). *Given an inconsistent ontology \mathcal{O} , a formula set \mathcal{O}' is a minimal inconsistent set (MIS) of \mathcal{O} iff it satisfies the conditions: i) $\mathcal{O}' \subseteq \mathcal{O}$, ii) $\mathcal{O}' \models \perp$, and iii) $\forall \mathcal{O}'' (\mathcal{O}'' \subset \mathcal{O}' \Rightarrow \mathcal{O}'' \not\models \perp)$.*

A minimal consistent set is akin to a justification [8], which is a minimal formula set to explain the entailment. In [9], the justification method [8] is used to compute minimal consistent sets in inconsistent ontologies. In this paper, we are interested in computing one *MIS* which includes one specified formula, i.e. the original formula which needs to be included in the minimal inconsistent set of the inconsistent ontology.

Algorithm 2 describes the process for computing such a specific *MIS*. The algorithm is taken from algorithm 2 in [10] with a few modifications, it applies binary search to quickly find a *MIS*. The algorithm partitions the ontology into two halves, and checks whether one of them is inconsistent. If yes, it goes to the recursion on that half, throwing away half of the axioms in one step. Otherwise, essential axioms are in both halves. In this case, the algorithm goes on the recursion on each half, using the other half as the support set.

The main idea of the CRION approach is to extend the linear extension strategy of the PION approach until the selected set $\Omega \cup \{\alpha\}$ is inconsistent. We have then obtained a minimal inconsistent set which includes α by using algorithm 2. Then we pick up a clarification formula β in the *MIS* and construct a contrastive formula γ from the *MIS*. A straightforward approach to construct the formula γ is to take the conjunction of some subset of the *MIS*. We call that approach *Contrastive Answer by Conjunction* (CAC).

Given an original answer $\mathcal{O} \approx \alpha$ which is obtained as the selected set $s(\mathcal{O}, \alpha, k)$ at step k , the CAC algorithm for obtaining contrastive answers is described in Algorithm 3. In the algorithm, we use S_c to denote the set of returned contrastive answers. If the S_c is \emptyset , then that means that there are no contrastive answers for the query.

The algorithm consists of the three main steps: i) extend the selected set until it becomes inconsistent, ii) find a minimal inconsistent set which includes α , and iii) construct the clarification formula β and the contrastive formula γ . It is not hard to prove the following proposition.

Proposition 3 (Soundness of the CAC Algorithm). *The contrastive answers obtained in Algorithm 3 are sound.*

Algorithm 2. *mis_binarySearch*(S, \mathcal{O}, α)**Assume:** $|\mathcal{O}| > 1$ in the initial step

```

1: if  $|\mathcal{O}| == 1$  then
2:   return  $\mathcal{O}$ 
3: end if
4:  $S_1, S_2 := \text{halve}(\mathcal{O})$ 
5: if  $S \cup S_1$  is inconsistent then
6:   return mis_binarySearch( $S, S_1, \alpha$ )
7: else if  $S \cup S_2$  is inconsistent then
8:   return mis_binarySearch( $S, S_2, \alpha$ )
9: end if
10:  $S'_1 := \text{mis\_binarySearch}(S \cup S_2, S_1, \alpha)$ 
11:  $S'_2 := \text{mis\_binarySearch}(S \cup S'_1, S_2, \alpha)$ 
12: if  $\alpha \in S'_1 \cup S'_2$  then
13:   return  $S'_1 \cup S'_2$ 
14: end if
15: return  $\emptyset$ 

```

Algorithm 3. Contrastive Answers by Conjunction (CAC)

```

1:  $S_c := \emptyset$ 
2:  $j := k$ 
3:  $\Omega := s(\mathcal{O}, \alpha, j)$ 
4: while  $\Omega \cup \{\alpha\}$  consistent do
5:    $j := j + 1$ 
6:   if  $s(\mathcal{O}, \alpha, j) \subseteq \Omega$  then
7:     return  $\emptyset$ 
8:   end if
9:    $\Omega := s(\mathcal{O}, \alpha, j)$ 
10: end while
11:  $\Omega' := \text{mis\_binarySearch}(\emptyset, \Omega \cup \{\alpha\}, \alpha)$ 
12: for  $\rho \in \Omega'$  do
13:   if  $\{\alpha, \rho\}$  consistent then
14:      $\beta := \rho$ 
15:      $\gamma := \bigwedge(\Omega' - \{\alpha, \beta\})$ 
16:     if  $\{\beta, \gamma\}$  consistent and  $\mathcal{O} \models \gamma$  then
17:        $S_c = S_c \cup \{\mathcal{O} \models \alpha \text{ but } \gamma \text{ although } \beta\}$ 
18:     end if
19:   end if
20: end for
21: return  $S_c$ 

```

Proof. If the algorithm returns an answer $\mathcal{O} \models \alpha$ **but** γ **although** β in S_c , we want to prove that α **but** γ **although** β is indeed a contrastive answer. From the given condition, we already have that $\mathcal{O} \models \alpha$. Since any formula is considered to be always the most (syntactically or semantically) relevant to itself, we have $\mathcal{O} \models \rho$ for any formula ρ such that $\rho \in \mathcal{O}$ and $\neg\rho \notin \mathcal{O}$ ⁶. Thus, we have $\mathcal{O} \models \beta$. From the algorithm, we have $\mathcal{O} \models \gamma$. It is easy to see that $\alpha \wedge \beta$ is not a contradiction and $\beta \wedge \gamma$ is not a contradiction, because $\{\alpha, \beta\}$ is consistent and $\{\beta, \gamma\}$ is consistent. Furthermore, from the algorithm, we know that $\alpha \wedge \beta \wedge \gamma$ is a contradiction because of the inconsistency of Ω' . Thus, we have the conclusion.

It is easy to see that the CAC algorithm can always terminate and that its computational cost is not significantly increased, compared with the complexity of the existing approaches in reasoning with inconsistent ontologies.

In the CAC algorithm, the contrastive formula is a conjunction of formulas selected from the ontology. We are more interested in contrastive formulas which are implied by a consistent subset of the minimal inconsistent set, rather than its subformulas which are contained by the ontology explicitly. Those contrastive answers may be obtained through the CAC approach by using the exchange property in Proposition 1.

A more general approach to obtaining those contrastive answers is to consider a contrastive formula γ which is a non-trivial consequence of the selected set, i.e., $\gamma \in Cn(\Omega' - \{\alpha, \beta\}) - Cn(\emptyset)$ in the algorithm⁷. We call that approach *Contrastive Answer by Logical Consequence* (CALC). The CALC algorithm is a revision of the CAC algorithm by constructing a formula γ which satisfies the condition above and inserting a step of contradiction checking for $\alpha \wedge \beta \wedge \gamma$ before Step 16 in Algorithm 3. There are various strategies to construct a contrastive formula γ for the CALC approach (e.g., depth-first search, breadth-first search, and best-first search). We will leave the investigation of variant CALC algorithms for future work.

4 Implementation and Evaluation

4.1 Implementation

We have implemented the prototype of CRION⁸ as a reasoning plug-in in the LarKC Platform. by using Pellet⁹ and OWLAPI¹⁰. Given a query answer in

⁶ In this paper we consider only the ontology \mathcal{O} in which there exists no a formula ρ such that $\rho \in \mathcal{O}$ and $\neg\rho \in \mathcal{O}$. Namely, the inconsistency in \mathcal{O} is not explicit. This condition is generally satisfied in practice.

⁷ Cn is a consequence operator such that $Cn(\mathcal{O}) = \{\varphi | \mathcal{O} \models \varphi\}$. $Cn(\emptyset)$ is the tautology set, which is considered to be trivial.

⁸ https://larkc.svn.sourceforge.net/svnroot/larkc/branches/Release_1.1_candidate/plugins/reason/CRION/

⁹ <http://clarkparsia.com/pellet/>

¹⁰ <http://owlapi.sourceforge.net/>

an inconsistent Description Logic (DL) ontology, CRION calculates contrastive answers based on the CAC approach. CRION uses PION¹¹ to compute the non-standard entailment in an inconsistent ontology. Syntax-based selection function defined in [5] is used in PION.

4.2 Evaluation

We have tested the CRION prototype by applying it to inconsistent ontologies. For that test, we selected two group of ontologies. The first group are several ontologies from the TONES ontology repository¹². Those ontologies are selected, because i) they are inconsistent, ii) Pellet supports them, and iii) we are familiar with the domains of those ontologies.

In order to test the run-time performance of our method in large scale ontologies, we construct the second group of ontologies by modifying from the LUBM¹³ benchmark ontology by inserting a specified number of conflicts using the Injector tool described in [11], where a conflict is a set of axioms violating a functional role restriction or a disjointness constraint. By LUBM-Lite_{*n+m*} we mean an LUBM-Lite ontology with assertional axioms of *n* universities and with *m* inserted conflicts. The profiles of the selected ontologies are shown in Table 2.

Table 2. Information about ontologies

Ontology	Syntax	#Cons	#Roles	#Inds	#Axioms	#MISs
MadCow	$\mathcal{ALCHOIN}(\mathbf{D})$	54	16	67	143	1
Pizza	\mathcal{SHION}	101	8	106	818	2
Economy	$\mathcal{ALCH}(\mathbf{D})$	338	45	818	1,947	51
Transportation	$\mathcal{ALCH}(\mathbf{D})$	446	89	629	1,786	62
LUBM-Lite1 ₊₂₀	$\mathcal{SHLF}(\mathbf{D})$	100	39	17,190	100,869	20
LUBM-Lite2 ₊₄₀				38,377	230,408	30
LUBM-Lite4 ₊₈₀				78,653	478,740	80
LUBM-Lite8 ₊₁₆₀				163,690	1,002,095	160
LUBM-Lite16 ₊₃₂₀				341,557	2,096,008	320

As the original formulas in a contrastive answer is related to a minimal inconsistent set, for each inconsistent ontology, we select the testing queries from the union of all minimal inconsistent sets calculated by using the explanation method in Pellet¹⁴. We evaluate the approach of contrastive reasoning with respect to the following three aspects:

¹¹ <http://wasp.cs.vu.nl/sekt/pion/>

¹² <http://owl.cs.manchester.ac.uk/repository/>, we expose their inconsistencies by adding a concept assertion for every named concept, i.e., $Con(the_Con)$.

¹³ <http://swat.cse.lehigh.edu/projects/lubm/>

¹⁴ It uses the method of `org.mindswap.pellet.owlapi.Reasoner.getExplanation()`.

- **Frequency:** Given an inconsistent ontology, how often can we obtain a contrastive answer? We measure the frequency by counting the amount of contrastive answers.
- **Usability:** Does the contrastive reasoning really achieve the main goals? Is it really useful or not? We evaluate the usability by examining the results with respect to the two main criteria: i) does it help avoiding misleading information and ii) does it improve the effective influence of the answer? Of course these criteria are necessarily “soft” in nature: they cannot be measured by any formal means, but must be subjected to human judgment.
- **Performance:** Is the contrastive reasoning computationally expensive or not? We evaluate the performance by examining the run-time performance of computing contrastive answers.

Frequency. Columns 2, 3 and 4 of Table 3 show that contrastive answers (CAs) occur frequently for inconsistent ontologies. For the MadCow ontology in which there is only one minimal inconsistent set, we have at least 25 contrastive answers for 5 queries. The total numbers of contrastive answers rise to hundreds (408) for the inconsistent ontologies which have dozens (51) of minimal inconsistent sets. For the second group of ontologies, the average numbers of contrastive answers are stable (around 3). There appear to be a reasonable number, and reasonably constant number, of contrastive answers per query across the tested ontologies (1-5). Moreover, note that the total number of contrastive answers will be at least doubled at by using the exchange and expansion property.

Table 3. Evaluation of number of contrastive answers by using CAC

Ontology	Number of queries	Total number of CAs	Average Number of CAs
MadCow	5	25	5.0
Pizza	8	33	4.1
Economy	160	408	2.55
Transportation	159	200	1.25
LUBM-Lite1 ₊₂₀	57	171	3.0
LUBM-Lite2 ₊₄₀	115	359	3.12
LUBM-Lite4 ₊₈₀	207	631	3.05
LUBM-Lite8 ₊₁₆₀	387	1157	3.04
LUBM-Lite16 ₊₃₂₀	703	2126	3.02

Usability. Five researchers score the computed contrastive answers of ontologies in the group one, based on the two main goals, which are discussed in Section 1, namely, avoiding misleading information and improving effective influence of the answer. Those two criteria are marked based on a five point scale: 0=valueless, 1=little value, 2=some value, 3=average value, 4=high value, and 5=perfect value. The average scores are listed in the second column and the third column of

Table 4. For the degree of avoiding misleading information, the scores range from 3.4 (= “average value”) to 4.2 (= “high value”). That means that the contrastive answers are considered to be somewhat useful to avoid misleading information for the four ontologies in our test. For the degree of improving effective influence, they have a very similar range, showing the answers to be somewhat useful for improving effective influence. The fact that all the scores in our small experiment are > 3 indicates that the approach of contrastive reasoning might indeed be useful for reasoning with inconsistent ontologies.

Table 4. Evaluation of value of contrastive answers

Ontology	Average value on	
	avoiding misleading information	improving effective influence
MadCow	4.2	4.0
Pizza	3.6	3.8
Economy	3.4	3.5
Transportation	3.7	3.5

Run-Time Performance. All the experiments are carried out on an ordinary PC (with a 2.60 GHz Pentium-4 processor and 2GB of physical memory, where the maximum Java heap size was set to 1280MB for applying Pellet). The maximal, minimal and average computation time (in seconds) for a query by using the CAC approach are shown in columns 2, 3 and 4 of Table 5.

The experimental results show that for all test ontologies in the first group, the CAC computation time for computing contrastive answers for a query is limited to a small number of seconds. The maximal computation time is just a few seconds (1.1s), the minimal computation time goes even to several milliseconds (0.007s), and the average computation time is less than one second (0.26s). For the large ontologies in the second group, the minimal computation time is only one second, the maximal computation time is less than several minutes when there are millions of axioms in the ontologies, and the average computation time is less than dozens of seconds.

It shows that the calculation of contrastive answers by using the CAC approach does not significantly increase the computational cost. Thus, it is an efficient extension to the existing reasoners with inconsistent ontologies.

4.3 Discussion

Contrastive answers are related with minimal inconsistent sets. One contrastive reasoning method is to calculate all minimal inconsistent sets in the inconsistent ontology (offline) firstly, then compute contrastive answers from these minimal inconsistent sets. In this paper, we use the CAC method for several reasons: i) the calculation of all minimal inconsistent sets is very difficult for a large scale

Table 5. Evaluation of the run-time performance of CAC

Ontology	Max run time	Min run time	Average run time
MadCow	0.22s	0.033s	0.12s
Pizza	1.10s	0.015s	0.26s
Economy	0.44s	0.016s	0.08s
Transportation	0.51s	0.007s	0.11s
LUBM-Lite1 ₊₂₀	1.17s	0.50s	0.58s
LUBM-Lite2 ₊₄₀	16.36s	1.00s	1.77s
LUBM-Lite4 ₊₈₀	37.32s	1.00s	3.27s
LUBM-Lite8 ₊₁₆₀	100.22s	1.00s	7.44s
LUBM-Lite16 ₊₃₂₀	440.19s	1.00s	19.92s

ontology, as demonstrated in [9, 12], ii) ontologies may be dynamical, especially in the Web setting, which may make an offline computation meaningless, iii) the CAC method can obtain a large amount of contrastive answers with a little cost, as shown in the experiments.

It is worth pointing out that algorithm 2 is an incomplete method, i.e., it may not find a *MIS* which includes the specific α although the *MIS* exists in the inconsistent set. The reason why we use it instead of a complete method lies in the complexity of the complete one. Generally, the complete method needs to compute all *MIS*s gradually until finding the required one, which is an *NP-hard* problem. In fact, we have already carried out some experiments to perform contrastive reasoning by using a complete *MIS* calculation method, the results show that the subprogram for calculating the specific *MIS* would not terminate in several hours for some queries with the Economy and Transportation ontologies. Hence, it is impractical.

Contrastive reasoning with DL ontologies can be extended by considering another kind of inconsistency (*incoherence* [13]), i.e., there exists an unsatisfiable named concept in the ontology. Incoherence is very important as many classical inconsistencies are caused by it, e.g., concept assertions of an unsatisfiable concept. In order to deal with incoherence, we need to consider it as well as the classical inconsistency when checking inconsistency in contrastive reasoning with DL ontologies.

5 Related Work

McGill and Klein address the differences in the use of covariation information implied by contrastive reasoning, which involves comparing the target episode to contrasting background instances [3]. Francez proposes the notion of bilogic as a logical treatment of a contrastive conjunction such as ‘but’, and argues that ordinary logics are not sufficient to express the contrastive nature of ‘but’, because of the neutral conjunction (‘and’) in classical logics [2]. Based on the contrastive operators proposed by Francez, a modal approach to contrastive logic is presented in [1]. Their contrastive logic is actually a simple modal logic which is an extension to the well-known S5 logic with Francez’s contrastive operator.

Default reasoning [14] is somehow similar to contrastive reasoning. It can be considered as a kind of reasoning service where the consequences may be derived only due to lacking evidence of the contrary. However, contrastive reasoning is different from default reasoning, because our approach is based on reasoning with inconsistent ontologies, whereas default reasoning is based on a non-monotonic logic.

6 Conclusions and Future Work

We have presented a general approach for answering queries over inconsistent ontologies by using contrastive reasoning. It is more practical for reasoning with inconsistent ontologies, as it provides not only an original answer, but also more relevant and maybe surprising answers. We have proved that obtaining contrastive answers can be achieved by a slight extension to the existing approach for reasoning with inconsistent ontologies. Furthermore, this extension does not significantly increase the computational cost. Our proposal has been implemented in the system CRION. We have reported several experiments with CRION and have presented an initial evaluation. The tests show that contrastive reasoning is useful and promising for reasoning with inconsistent ontologies.

There is a lot of the future research to be done. Here are just some of them:

- Contrastive Answer by Logical Consequence. As discussed in Section 3, various strategies for obtaining contrastive answers of the CALC approach will be very interesting to gain more useful answers.
- Reasoning with contrastive ontologies. In this paper, we have provided contrastive answers only at the query language level. We have not yet allowed to express contrastive conjunctions in the ontology level. Thus, one of the interesting future works is to reason with inconsistent ontologies that contain contrastive conjunction axioms such as “but”.

Acknowledgment. This work is supported by the European Commission under the 7th framework programme, Large Knowledge Collider (LarKC) Project (FP7-215535).

References

- [1] Meyer, J.J.C., van der Hoek, W.: A modal contrastive logic: The logic of ‘but’. *Annals of Mathematics and Artificial Intelligence* 17, 291–313 (1996)
- [2] Francez, N.: Contrastive logic. *Logic Journal of the IGPL* 3(5), 725–744 (1995)
- [3] McGill, A.L., Klein, J.G.: Counterfactual and contrastive reasoning in causal judgment. *Journal of Personality and Social Psychology* (64), 897–905 (1993)
- [4] Fang, J., Huang, Z., van Frank, H.: Contrastive reasoning with inconsistent ontologies. In: *Proceedings of 2011 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2011)*, pp. 191–194 (2011)
- [5] Huang, Z., van Harmelen, F., ten Teije, A.: Reasoning with inconsistent ontologies. In: *Proceedings of IJCAI 2005*, pp. 454–459 (2005)

- [6] Haase, P., van Harmelen, F., Huang, Z., Stuckenschmidt, H., Sure, Y.: A Framework for Handling Inconsistency in Changing Ontologies. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 353–367. Springer, Heidelberg (2005)
- [7] Huang, Z., van Harmelen, F.: Using Semantic Distances for Reasoning with Inconsistent Ontologies. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318, pp. 178–194. Springer, Heidelberg (2008)
- [8] Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding all Justifications of OWL DL Entailments. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 267–280. Springer, Heidelberg (2007)
- [9] Horridge, M., Parsia, B., Sattler, U.: Explaining Inconsistencies in OWL Ontologies. In: Godo, L., Pugliese, A. (eds.) SUM 2009. LNCS, vol. 5785, pp. 124–137. Springer, Heidelberg (2009)
- [10] Baader, F., Suntisrivaraporn, B.: Debugging SNOMED CT using axiom pinpointing in the description logic \mathcal{EL}^+ . In: KR-MED 2008. CEUR-WS, vol. 410 (2008)
- [11] Du, J., Shen, Y.D.: Computing minimum cost diagnoses to repair populated dl-based ontologies. In: WWW, pp. 565–574 (2008)
- [12] Du, J., Qi, G.: Decomposition-Based Optimization for Debugging of Inconsistent OWL DL Ontologies. In: Bi, Y., Williams, M.-A. (eds.) KSEM 2010. LNCS, vol. 6291, pp. 88–100. Springer, Heidelberg (2010)
- [13] Flouris, G., Huang, Z., Pan, J.Z., Plexousakis, D., Wache, H.: Inconsistencies, negations and changes in ontologies. In: Proc. of AAAI 2006, pp. 1295–1300 (2006)
- [14] Brewka, G.: Preferred subtheories: An extended logical framework for default reasoning. In: Proceedings of IJCAI 1989, pp. 1043–1048 (1989)