Axiom-oriented Reasoning to Deal with Inconsistency Between Ontology and Knowledge Base

Tuan A. Luu[1], Tho. T Quan[1], Tru H. Cao[1] and Jin-Song Dong[2]

[1]Faculty of Computer Science and Engineering
Ho Chi Minh City University of Technology
Vietnam
[2] School of Computing
National University of Singapore
Singapore
qttho@cse.hcmut.edu.vn

Abstract. When deployed in practical applications, Ontologies and KBs often suffer various kinds of inconsistency, which limit the applications performances significantly. In this paper, we propose a framework to reason inconsistency between Ontology and KB and refine the inconsistency accordingly. To make our framework efficient, we only focus on reasoning a part responsible for the inconsistency, rather than the whole structures of Ontology and KB. Moreover, to improve the execution speed of algorithms employed in the framework, we also discuss an axiom-oriented strategy to reason on a reduced space of formula to be inferred in Ontology and KB.

1 Introduction

The Semantic Web [1] is developed as a concept of how computers, people, and the Web can work together more effectively than it is possible now. Ontology and Knowledge Base (KB) are two significant elements of the Semantic Web. However, when used in practical applications, Ontologies and KBs always suffer inconsistencies due to various reasons. In recent literature, there are two emerging approaches following this direction: to diagnose and repair inconsistency in Ontology by finding minimal inconsistent subset [2] ; and to reason in inconsistent Ontology and KB based on maximum consistent subset constructed [3] .

In this paper, we propose a framework to handle inconsistency between Ontology and KB. It is done by reasoning to find the part responsible for the inconsistency and then refining the detected inconsistencies accordingly. In addition, to reduce the complexity cost of algorithms employed in the framework, we also develop an axiom-oriented strategy to isolate and detect the axioms responsible for the inconsistency. The rest of the paper is organized as follows. Section 2 presents formal definitions of Ontology and Knowledge Base. Section 3 discusses inconsistency between Ontology and KB. In Section 4, the general framework for inconsistency detecting and repairing is given. Section 5 gives discussion of the axiom-oriented strategy to deal with inconsistency. Finally, Section 6 concludes the paper.

2 Ontology and Knowledge Base

Definition 1 (Ontology). An ontology is a structure $O = (C; T; R; A; \leq_C; \leq_T; \delta_R; \delta_A; \tau_T; S_A)$. It consists of disjoint sets of concepts (or classes) C, types T, relations R, attributes A, and values V. The partial orders $\leq_C$ (on C) and $\leq_T$ (on T) define a concept hierarchy and a type hierarchy, respectively. The function δ_R: $R \rightarrow C^2$ provides relation signatures (i.e., for each relation, the function specifies which concepts may be linked by this relation); while the function δ_A: $A \rightarrow C \times T$ provides attribute signatures (for each attribute, the function specifies to which concept the attribute belongs and what is its data type); and τ_T: $T2^V$ is the assignment of values to types. S_A is a set of axioms, restrictions between concepts and attributes.

Example 1. We define *Football Ontology* $O = (C; T; R; A; \leq_C; \leq_T; \delta_R; \delta_A; \tau_T; S_A)$ where

C = {football-player, person, club, city }
$\leq_C$ = {football-player ⊆ person}
T = {integer}
R = {live-in, locate-in, play-for, has-wife}
A = {age, height, weight}
δ_R = {live-in → football-player x city,live-in → person x city,locate-in → club x city, play-for → football-player x club, has-wife → football-player x football-player}
δ_A = {age→football-playerx integer,height → football-player x integer, weight → football-player x integer}
S_A = {(O_1) football-player(x) Λ club(y) Λ city (z) Λ play-for(x, y) Λ locate-in(y, z) → live-in(x, z) *// football player plays for club will live in the cty that the club locates*.
(O_2) football-player(x) Λ city(y) Λ city (z) Λ live-in(x, y) Λ live-in(x, z) → y = z *// football player is not living in more than one city.*
(O_3) football-player(x) Λ has-wife(x, y)Λcity(z) Λ live-in(y, z) → live-in(x,z) *// football player who has wife will lives in the city will live in the same city as her wife's*.
(O_4) club(x) Λ locate-in(x, z) Λ club(y) Λ locate-in(y, z) → x = y *// each city has not more than one club*.}

Definition 2 (Knowledge Base). A Knowledge Base (KB) is a structure $K = (C; R; A; I; V; \tau_C; \tau_R; \tau_A)$. It consists of disjoint sets of concepts (or classes) C, relations R, attributes A, individuals I and values V. The function τ_C: $C2^I$ is the assignment of instances to concepts), the function τ_R: $R \rightarrow 2^{I \times I}$ defines relations between instances, and τ_A: $A \rightarrow 2^{I \times V}$ defines attributes of instances.

Example 2. We define *Football KB as* K = $(C; R; A; I; V; \tau_C; \tau_R; \tau_A)$ where:

I = {Beckham, MU, Manchester, Liverpool, Chelsea, Maria)
τ_C = {(K_5) football-player (Beckham), (K_6) club (MU), (K_7) city (Manchester), (K_8) city (Liverpool), (K_9) club (Chelsea)}
τ_R = {(K_{10}) live-in (Beckham, Liverpool), (K_{11}) play-for (Beckham, MU), (K_{12}) locate-in (MU, Manchester), (K_{13}) has-wife (Beckham, Maria), (K_{14}) live-in (Maria, Manchester), (K_{15}) locate-in (Chelsea, Manchester)}
τ_A = { (K_{16}) age (Beckham, 30), (K_{17}) height (Beckham, 180),(K_{18}) weight (Beckham,80)}

3 Inconsistency between Ontology and KB

Although KB (containing concrete data) is always encoded with respect to an ontology (containing a general conceptual model of some domain knowledge), people may find it difficult to understand the logical meaning of the underlying ontology. Hence, people may fail to formulate precisely axioms, which are logically correct, or may specify contradictory statements.

Example 3. Between in *Football Ontology* and *Football KB* defined respectively in Example 1 and Example 2, from (K_5), (K_{10}), (K_{13}), and (K_{14}), we can infer that Beckham lives in Liverpool but has wife living in Manchester. However, from (O_3) we can see that Beckham must live in the same city with his wife. Thus, *Football Ontology* and *Football KB* are inconsistent.

4 Framework for Diagnosing and Repairing Inconsistency Between Ontology and KB

In this section, we present a framework to reason inconsistency between Ontology and KB. The framework is conducted by incorporating the algorithm for debugging inconsistency proposed in [2] and the basic theory of finding the inconsistency introduced in [3] . As shown in Figure 1, the proposed framework consists of three steps as follows:

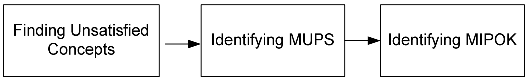

Figure 1. Framework for diagnosing and repairing inconsistency between Ontology and KB

- *Step 1*: It finds all unsatisfied concepts. An unsatisfied concept is a concept that does not have any individual for all models of Ontology and KB.
- *Step 2*: For every unsatisfied concept, we identify a minimal subset axioms and facts that are responsible for an inconsistency, called Minimal Unsatisfied Preserving Sub Ontology and KB (MUPS).
- *Step 3*: From the set of MUPS, we diagnose the smallest subsets of axioms and facts responsible for all inconsistencies, or Minimal Inconsistent Preserving Sub Ontology and KB (MIPOK). Finally, relying on this MIPOK, we will repair this Ontology and KB.

Example 4. We apply the proposed framework to deal with inconsistency between *Football Ontology* and *Football KB* given in Example 1 and Example 2. As a result, *Refined Football Ontology* is redefined as $O_R = (C; T; R; A; \leq_C; \leq_T; \delta_R; \delta_A; \tau_T; S_A)$, where:

C = {football-player, person, club, city }
$\leq_C$ = {football-player $\subseteq$ person}
T = {integer}
R = {live-in, locate-in, play-for, has-wife}
A = {age, height, weight}
δ_R = {live-in → football-player x city, live-in → person x city, locate-in → club x city, play-for → football-player x club, has-wife → football-player x football-player}
δ_A = {age → football-player x integer, height → football-player x integer, weight → football-player x integer}
S_A = {(O_1) football-player(x) ∧ club(y) ∧ city (z) ∧ play-for(x, y) ∧ locate-in(y, z) → live-in(x, z) *// football player plays for club will live in the city that the club locates*.
(O_2) football-player(x) ∧ city(y) ∧ city (z) ∧ live-in(x, y) ∧ live-in(x, z) → y = z *// football player is not living in more than one city.*
(O_3) football-player(x) ∧ has-wife(x, y) ∧ city (z) ∧ live-in(y, z) → live-in(x, z) *// football player who has wife will lives in the city will live in the same city as her wife's*.}

Refined Football KB is redefined *as* $K_R = (C; R; A; I; V; \tau_C; \tau_R; \tau_A)$ where:

I = {Beckham, MU, Manchester, Liverpool, Chelsea, Maria)
V = {30, 80, 180}
τ_C = {(K_5) football-player (Beckham), (K_6) club (MU), (K_7) city (Manchester), (K_8) city (Liverpool), (K_9) club (Chelsea)}
τ_R = {(K_{11}) play-for (Beckham, MU), (K_{12}) locate-in (MU, Manchester), (K_{13}) has-wife (Beckham, Maria), (K_{14}) live-in (Maria, Manchester),(K_{15}) locate-in (Chelsea, Manchester)}
τ_A = {(K_{16}) age (Beckham, 30), (K_{17}) height (Beckham, 180), (K_{18}) weight (Beckham, 80)}

5 Axiom-oriented Construction of MUPS

In [2] and [3] , the authors have proposed an algorithm to find MUPS, as presented in Figure 2. However, because we only focus on solving the inconsistency between Ontology and KB, i.e. inconsistency occurs in the relations between facts and axioms, so we can apply an axiom-oriented strategy in the selection function. It is carried out using the following selection rules.

Rule 1 (Axiom-Related Selection). Only add to the *final_set* mentioned in Algorithm 1 formulae that are not only directly relevant to this set but also directly relevant to at least an axiom in Ontology.

Rule 2 (Onto-KB Selection). Only consider the *subset* S_1 and *subset* T_1 mentioned in Algorithm 1 if the formulae in them occur in both Ontology and KB.

Algorithm 1. Finding MUPS of an unsatisfied concept *c*.

```
Input: Unsatisfied concept c with set of formulae ∑.
Output: set MUPS corresponding to c.
Process:
    set S = {c}, final_set = Ø.
    from S find set of formulae S' that is directly relevant to S.
    if  S' is consistent then
       set S = S'.
       repeat
          Find new set of formulae S' that is direct relevant to S
          if  S' is consistent then S = S'
       until c is inconsistent in S'
    end if
    set T = S' – S
    for all subset T1 of T and all subset S1 of S
       if c is inconsistent in {T1 U S1} then final_set = final_set U {T1 U S1}
    end for
    MUPS(∑ , c) := Minimality-Checking(final_set)
    return MUPS(∑ , c)
```

Figure 2. Algorithm for finding MUPS of an unsatisfied concept *c*.

Example 5. Consider *Football Ontology* and *Football KB* given in Example 1 and Example 2. The effectiveness of using axiom-oriented approach is demonstrated, as the numbers of subsets generated when calculated MUPS(∑, *football-player*) are 2^{32} and $2^{26} - 2^{21}$ in non axiom-oriented and axiom-oriented methods, respectively.

6 Conclusion

In this paper, we first introduced inconsistency occurring between Ontology and KB. Then, we proposed some refinements and improvements for an effective framework to solve the inconsistency between Ontology and KB in the reasonable complexity and time. Generally, our proposed framework only focuses on axioms, rather than the whole structure of ontology. Hence, our approach is highly potential in terms of reducing computational cost, as compared to similar existing work.

References

[1] Tim Berners-Lee, James Hendler and Ora Lassila. *The Semantic Web.* Scientific American, 284(5) , pages 34-43, May 2001.

[2] Stefan Schlobach and Zhisheng Huang, *Inconsistent Ontology Diagnosis: Framework and Prototype*. SEKT/2005/D3.6.1/v0.9, SEKT EU-IST-2003-506826, October 2005.

[3] Zhisheng Huang, Frank van Harmelen, and Annette ten Teije. *Reasoning with Inconsistent Ontologies*. Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI 2005), August 2005.

[4] Marc Ehrig, Peter Haase, Mark Hefke, Nenad Stojanovic. *Similarity for Ontologies – a Comprehensive Framework*. Proceedings of the 13th European Conference on Information Systems (ECIS 2005), May 2005.

[5] Steffen Staab and Heiner Stuckenschmidt. *Semantic Web and Peer-to-Peer*. Springer-Verlag Berlin Heidelberg, 2006.