Approaches to Handling Ontology Inconsistency

FANG Yishu and XIA Yingying
Last updated: 10 May 2013

Abstract. The Semantic Web (SW) is a constantly changing and collaborative environment. Knowledge base is the core component of the SW. It is reasonable that, as the formal representation of knowledge in the SW, ontology may encounter all kind of problems, especially inconsistency. Reasoning techniques for ontologies that can handle all kinds of inconsistency are needed. The inconsistency handling problem has recently attracted a lot of attention. Existing approaches can be roughly divided into two families: either resolving inconsistency or reasoning with inconsistency. In this paper, we give a brief review of those two families of approaches.

1. Introduction

Knowledge representation plays a critical role in the successful construction of the Semantic Web (SW). Ontology has been regarded as the formal representation of the knowledge of concepts and the relationships among them (Thomas, 2013). As the SW is treated as a constantly changing and collaborative environment, it is reasonable that ontologies may be exposed to all kinds of problems, especially inconsistency. Ontology inconsistency may occur due to a number of reasons, such as differences in conceptualization, contradictory domain knowledge, errors in conceptualization, and different realizations of the concepts and properties in different contexts and applications (Hussain, De Roo, Daniyal, & Abidi, 2011). If the ontology itself is error prone then dependent applications have to face the risks of some critical problems and uncertainty. So the ability of handling inconsistency one is of the most important parts in Ontology Engineering in the process of the evaluation of ontology. The inconsistency handling problem has recently attracted a lot of attention and this paper will take a close look at those proposed approaches.

2. Scope

According to the survey conducted by Bell, Qi, and Liu (2007), two main families of approaches are widely used to deal with the inconsistency: resolving inconsistency and reasoning with inconsistency. With regard to resolving inconsistency, the debugging and repair approaches are firstly introduced to handle incoherence which is more description logic (DL) specific. The idea is that diagnosing terminological incoherence and then selecting the axioms or subsets of axioms causing incoherence to delete. Furthermore, to solve inconsistency we can weaken the terminological axioms for a finer granularity (but more computational difficult) or just delete an axiom if it is an assertion one. With the underlying knowledge base unchanged, Paraconsistent Reasoning, Uncertainty Reasoning, and Coherence-based Reasoning are employed to illustrate with inconsistent ontologies. The first method expends the semantics to true, false, underdetermined and overdetermined, i.e. both true and false, allowing the reasoner to draw a contradictory conclusion. The next one uses possibilitistic logic to dealing with uncertainty and handling inconsistency. The last one uses a selection function to select consistent axioms that are only related to query to do the reasoning.

In this paper, we will give a comprehensive review of those two families of existing approaches for ontologies inconsistency handling.

3. Body

3.1 Preliminaries

In this section, we introduce some required notations and definitions defined by Baader (2003).

Definition 1 (DL-based ontology): a DL-based ontology $\mathcal{O} = (\mathcal{T},\mathcal{A})$ is a set T of concept axioms (TBox) and role axioms, and a set A of assertional axioms (ABox). Concept axioms have the form $C \sqsubseteq D$, where C and D are concept descriptions and role axioms are expressions of the form $R \sqsubseteq S$, where R and S are role descriptions. We call both concept axioms and role axioms as terminology axioms. The ABox contains concept assertions of the form C(a) where C is a concept and a is an individual name, and role assertions of the form R(a,b), where R is a role and a and b are individual names.

Definition 2 (Interpretation): an interpretation $\mathcal{I}=(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a non-empty domain set $\Delta^{\mathcal{I}}$ and an interpretation function $\cdot^{\mathcal{I}}$, which maps from individuals, concepts and roles to elements of the domain, subsets of the domain and binary relations on the domain, respectively.

Definition 3 (Model of an ontology): an interpretation $\mathcal{I}$ is called a model of an ontology, *iff* it satisfies each axiom in the ontology.

3.2 Resolving Inconsistency

Flouris, Huang, Pan, Plexousakis, and Wache (2006) define an ontology as **incoherent** one *iff* there is an unsatisfiable concept name in this ontology and an ontology as **inconsistent** one *iff* it has no model. The incoherence can be seen as a potential case of inconsistency and it is more DL-specific. In this section, we will firstly apply the debugging and repair approach to resolve incoherence. Then we will introduce several approaches to deal with inconsistency.

Approaches to Resolving Incoherent

Bell et al. (2007) state that to solve incoherence, the debugging and repair approaches are usually used. Schlobach and Cornet (2003) initiates a debugging approach focused on unfoldable $\mathcal{ALC}$ TBoxes using Boolean methods for minimal unsatisfiability-preserving sub-TBoxes and a way of calculating terminological diagnoses using Reiter's algorithms. However, this approach is limited to unfoldable $\mathcal{ALC}$ TBoxes and is tableaux reasoner-dependent. A more suitable black-box solution to deal with incoherence is proposed by Kalyanpur, Parsia, Grau, and Sirin (2006) . The First step is computing a single MUPS which can be defined as a minimal incoherence-preserving sub-TBox of an incoherent TBox. Secondly, retrieve the remaining ones by utilizing the Hitting Set approach. To solve the incoherence, we can select the axioms or subsets of axioms causing incoherence to delete. But this solution needs to remove an axiom even if only part of the axiom is responsible for the conflict. Lam, Pan, Sleeman, and Vasconcelos (2006) propose a finer-grained approach to rewriting the problematic axioms in an ontology by revising the classical tableaux algorithm. This approach is able to pinpoint the axioms involved in incoherence and to capture which parts of the axioms are responsible for the unsatisfiability of concepts as well.

Approaches to Resolving Inconsistency

Support for ontology evolution is extremely important in ontology engineering and application of ontologies considering the dynamic environments. Furthermore, in the process of ontology evolution, a critical aspect is to guarantee consistency of the ontology once several changes take place, considering the semantics of the ontology change as well (Haase & Stojanovic, 2005) . The approach to preserving ontology consistency in this inevitable evolution consists of two main phrases: Inconsistency Detection and Change Generation (Bell et al., 2007). Inconsistency Detection is responsible for checking the consistency of an ontology and identifying the parts in the ontology that violate original consistent conditions. Change Generation is responsible for generating extra changes that resolve detected inconsistencies to maintain the consistency of the ontology. Haase and Stojanovic (2005) have presented two kinds of algorithms to detect inconsistencies relied on the idea of establishing a selection function to define the relevant axioms that contribute to the inconsistency. The first one aims at finding a maximal consistent subontology and the second one aims at finding a minimal inconsistent subontology. The method of repairing inconsistency based on the above detecting approaches is removing the complete axioms from the ontology. Another alternative which is finer granularity is just weakening the relevant terminology axioms. Meyer, Lee, and Booth (2005) give an algorithm, called refined conjunctive maxi-adjustment, to weaken conflicting information. In this algorithm, they introduced a DL expression which is defined as cardinality restrictions on concepts to deal with a terminological axiom. However, with regard to weakening an assertional axiom, they simply delete it. This algorithm is proved to be finer-grained but more computational complexity.

3.3 Reasoning with Inconsistency

In general, there are three main methods that can be used to reasoning with inconsistency: paraconsistent reasoning, uncertainty reasoning and coherence-based reasoning.

Paraconsistent Reasoning

This method is proposed by Ma, Hitzler, and Lin (2007) and then extended by Ma and Hitzler (2009). The so-called paraconsistent semantics is employed here to deal with the inconsistency problem. In addition to the true value and the false value of classical two-value semantics, another two truth values are added: undefined and overdefined (or overdetermined, contradictory). The undefined value means neither true nor false, while the overdefined value stands for contradictory information, means both inferred as right and wrong at the same time. By applying the four-valued semantics, we may infer some contradictory conclusions; those conclusions may in turn inform upper layers to take more attention about this.

Uncertainty Reasoning

Based on the same idea of the former method, this one also tries to extend description logics to achieve the goal of handling ontology inconsistency. Possibilistic logic is proposed as a powerful tool to deal with uncertainty and inconsistency due to its powerful representation ability. From this point of view, Qi, Ji, Pan, and Du (2011) argue that possibilistic logic could be appropriate here as an extension of description logics. However, there are some drawbacks by using possibilistic inference services. For instance, it always suffers from the drowning problem, that is, axioms whose confidence degrees are less than or equal to the inconsistency will not be used. They adopt a drowning-free variant of possibilistic inference, called linear order inference to solve this problem. They propose an algorithm for computing the inconsistency degree of a possibilistic description logic knowledge base. Another algorithm is also proposed by them for the linear order inference.

Coherence-based Reasoning

Compared with uncertainty reasoning by extending existing description logic language, coherence-based reasoning tends to be more practical because reasoning can be directly conducted with DL-based ontologies. Coherence-based reasoning method tries to filter out inconsistent information before reasoning. This can be implemented by starting with an empty (thus consistent) ontology and gradually selecting and adding axioms that do not incur any inconsistency into it. Fang and Huang (2010) have proposed a concept relevance-based framework for reasoning with inconsistent ontologies. A selection function, which is defined on the syntactic or semantic relevance, is employed to select some consistent sub-theories from an inconsistent ontology. After the relevant consistent axioms are selected and the consistent ontology is constructed, standard reasoning method can be used to conduct the actual reasoning.

4. Interpretation and Conclusion

This paper gives a comprehensive review of existing approaches to handling consistency in ontologies. Existing approaches can be divided into two types: resolving inconsistency and reasoning with inconsistency.

When dealing with resolving inconsistency in ontologies, we differentiate logical inconsistency from incoherence. The latter one is more DL-specific and is usually resolved by applying the debugging and repair approaches. In the sense of inconsistency in classical logic, several algorithms of detecting and repairing are reviewed. The basic idea of repairing is deleting or weakening an axiom in the ontology to maintain ontology consistency. The weakening-based approaches are more fine-grained but more computational complexity than the deleting-based approaches.

While reasoning with inconsistency, three methods can be used: paraconsistent reasoning, uncertainty reasoning and coherence-based reasoning. The first two methods involve extending the existing description logic, while the last method involves selecting consistency part of the ontology to do the reasoning.

References

Baader, Franz. (2003). *The description logic handbook: theory, implementation, and applications*: Cambridge: Cambridge University Press.

Bell, David, Qi, Guilin, & Liu, Weiru. (2007). *Approaches to inconsistency handling in description-logic based ontologies.* Paper presented at the On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops.

Fang, Jun, & Huang, Zhisheng. (2010). Reasoning with inconsistent ontologies. *Tsinghua Science & Technology, 15*(6), 687-691.

Flouris, Giorgos, Huang, Zhisheng, Pan, Jeff Z, Plexousakis, Dimitris, & Wache, Holger. (2006). *Inconsistencies, negations and changes in ontologies.* Paper presented at the Proceedings of the National Conference on Artificial Intelligence.

Haase, Peter, & Stojanovic, Ljiljana. (2005). Consistent evolution of OWL ontologies *The Semantic Web: Research and Applications* (pp. 182-197): Springer.

Hussain, Sajjad, De Roo, Jos, Daniyal, Ali, & Abidi, Syed Sibte Raza. (2011). *Detecting and Resolving Inconsistencies in Ontologies Using Contradiction Derivations.* Paper presented at the Computer Software and Applications Conference (COMPSAC), 2011 IEEE 35th Annual.

Kalyanpur, Aditya, Parsia, Bijan, Grau, Bernardo Cuenca, & Sirin, Evren. (2006). Justifications for entailments in expressive description logics: Technical report, University of Maryland Institute for Advanced Computer Studies (UMIACS).

Lam, Sik Chun, Pan, Jeff Z, Sleeman, Derek, & Vasconcelos, Wamberto. (2006). *A fine-grained approach to resolving unsatisfiable ontologies.* Paper presented at the Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on.

Ma, Yue, & Hitzler, Pascal. (2009). Paraconsistent reasoning for OWL 2 *Web Reasoning and Rule Systems* (pp. 197-211): Springer.

Ma, Yue, Hitzler, Pascal, & Lin, Zuoquan. (2007). Algorithms for paraconsistent reasoning with OWL *The Semantic Web: Research and Applications* (pp. 399-413): Springer.

Meyer, Thomas, Lee, Kevin, & Booth, Richard. (2005). *Knowledge integration for description logics*. Paper presented at the PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE.

Qi, Guilin, Ji, Qiu, Pan, Jeff Z, & Du, Jianfeng. (2011). Extending description logics with uncertainty reasoning in possibilistic logic. *International Journal of Intelligent Systems, 26*(4), 353-381.

Schlobach, Stefan, & Cornet, Ronald. (2003). *Non-standard reasoning services for the debugging of description logic terminologies*. Paper presented at the International Joint Conference on Artificial Intelligence.

Thomas, Divya. (2013). A Survey On Various Ontology Debugging Methods. *International Journal of Advanced Research in Electronics and Communication Engineering, 2*(3), pp. 289-292.