Dealing with Inconsistencies in *DL-Lite* Ontologies [⋆]

Liping Zhou

Beijing Jiaotong University, Beijing, China
dearliping@gmail.com

1 The research problem

As a shared conceptualization of a particular domain, ontologies play an important role for the success of the Semantic Web. However it is often difficult to create an absolutely consistent ontology. Inconsistency can occur due to several reasons, such as modeling errors, migration or merging ontologies, and ontology evolution. So it is essential to study how to deal with inconsistent ontologies. Many approaches have been proposed to solve this problem. These approaches are mainly used for dealing with inconsistency in expressive Description Logics(DLs). In our work, we consider inconsistency handling in the *DL-Lite* family, which is a family of DLs that preserve tractable reasoning and are specifically tailored to deal with large amounts of data. Like other DLs, inconsistencies in *DL-Lite* can also easily occur because disjoint axioms are allowed.

2 The state of art

There are two main ways to deal with inconsistent ontologies [5]. One way is to simply avoid the inconsistency and to apply a non-standard reasoning method to obtain meaningful answers. For example, Huang et al. in [5] have proposed a general framework for reasoning with inconsistent ontologies by using a selection function. In [7, 8, 12], four-valued logics have been applied to reason with inconsistent ontologies.

The other is to resolve logical modeling errors whenever a logical problem is encountered. For example, some researchers proposed methods to debug erroneous terminologies and have them repaired when inconsistencies are detected [17, 18, 13, 16, 11, 4]. Some researchers considered resolving the logical inconsistencies during ontology evolution through some revision operators (see, for example, [15]).

When resolving inconsistencies, we always need some additional information to decide which solutions are better. Therefore, some researchers provided methods to measure the incoherence [14] or inconsistency [10, 9, 3] of ontologies. For example, some researchers provided methods for measuring inconsistency of axioms to identify

[⋆] I would like to acknowledge my supervisor Houkuang Huang, my co-supervisor Youli Qu, my external supervisors Zhisheng Huang and Guilin Qi for their guidance, support and fruitful discussions. Our research is supported by the National Grand Fundamental Research 973 Program of China under grant (No.2007CB307100, No.2007CB307106) and by the Specialized Research Foundation of Doctoral Program of Higher Education of China under grant (No.20050004008).

which axioms need to be removed or modified to resolve an inconsistency (see [3, 10]). Qi and Hunter in [14] provide a method for measuring the incoherence of an ontology.

The above approaches to dealing with inconsistency are usually based on expressive DLs which suffer from worst-case exponential time behavior of reasoning [1]. This may hinder their applications to very large real life ontologies.

As a family of tractable DLs, *DL-Lite* can keep all the reasoning tasks tractable, in particular, with polynomial time complexity with respect to the size of the knowledge base [2]. Recently, there are some researchers began to study the problem of dealing with inconsistent *DL-Lite* ontologies. For example, Lembo et al. in [6] provided an approach for consistent query answering over ontologies specified in *DL-Lite*. Ma et al. in [8] proposed four-valued semantics for *DL-Lite*.

3 Proposed Approach

In our work, we mainly focus on how to handle inconsistency in *DL-Lite*. We will explore the following approaches. The list of approaches is not exhausting and new approach may be considered. First, we will work on a method for debugging a *DL-Lite* ontology by exploring the specific feature of *DL-Lite*. Second, we will work on a method for consistent query answering in *DL-Lite* based on multi-valued semantics. Third, we will propose a novel method for measuring inconsistency in a *DL-Lite* ontology. The main expected results of our research are as follows:

1. To give a method for measuring the inconsistency degree of *DL-Lite* ontologies. The inconsistency measures can be used to evaluate the quality of the ontology, to rank axioms in an ontology and to give guidance to resolve inconsistency. We expect that the inconsistency degree of an ontology is a single value. Unlike the work given in [10], our work uses a single value as the inconsistency degree of an ontology. Arguably, this single value is easier to be used to deal with inconsistencies, we will find an algorithm to compute the inconsistency degree in polynomial time.

2. To find a debugging algorithm tailored for *DL-Lite* ontologies. We will explore the tracing techniques for consistency checking algorithm in *DL-Lite*.

3. To design algorithms for query answering over inconsistent *DL-Lite* ontologies. Based [6], we plan to find a solution for query answering over inconsistency *DL-Lite* ontologies based on multi-valued semantics.

4. To implement the proposed algorithm for dealing with inconsistencies in *DL-Lite* ontologies.

In this paper, we will elaborate the first task, i.e., measuring inconsistency in *DL-Lite* ontologies. Our work is based on a three-valued semantics. Compared with classical semantics, three-valued semantics allows for a third truth contradictory such that even an inconsistency ontology can has three-valued models. Since we only aim to analyze inconsistency, there is no need to adopt other multi-valued semantics, such as four-valued semantics which contains a fourth truth value for expressing incomplete knowledge.

4 Research methodology and results

4.1 Three-valued semantics

Inspired by the four-valued semantics [7, 8], we give a three-valued semantics for *DL-Lite*. Based on the three-valued semantics, the truth values about concepts or roles have three values instead of the classical two. The three truth values are $true$, $false$ and $contradictory$, we use the symbols $t, f, \mathbb{B}$ respectively to denote them. For a given domain Δ, we give each concept or role an extended truth value $\langle P, N\rangle$, where P (resp., N) is the subset of Δ (for concept) or $\Delta \times \Delta$ (for role) that supports concept or role to be true (resp., false). We denote $proj^{+}(\langle P, N\rangle) = P$ and $proj^{-}(\langle P, N\rangle) = N$. Given a three-valued interpretation I, we say that I satisfies a concept inclusion axiom $B \sqsubseteq C$ (resp., a role inclusion axiom $R_1 \sqsubseteq R_2$) if $proj^{+}(B^I) \subseteq proj^{+}(C^I)$ (resp., $proj^{+}(R_1^I) \subseteq proj^{+}(R_2^I)$). I satisfies a function assertion (Funct Q) if $\forall x, y, z, (x, y) \in proj^{+}(Q^I) \wedge (x, z) \in proj^{+}(Q^I) \rightarrow (y, z) \in proj^{+}((=)^I)$. Furthermore, I satisfies an atomic concept assertion $C(a)$ (resp., an atomic role assertion $R(a, b)$) if $a^I \in proj^{+}(C^I)$ (resp., $(a^I, b^I) \in proj^{+}(R^I)$). A three-valued model of a *DL-Lite* ontology $\mathcal{K}$ is a three-valued interpretation I which satisfies each assertion and each axiom in $\mathcal{K}$. A *DL-Lite* ontology is three-valued satisfiable (unsatisfiable) if there exists (does not exist) such a model. Furthermore, any *DL-Lite* ontology has the finite model property under three-valued semantics.

4.2 Approach to measuring inconsistency

As we have discussed above, we want to compute a single inconsistency degree of a *DL-Lite* ontology. We start with studying on the conflicting atomic assertions. Considering the novel property of *DL-Lite* that a *DL-Lite* knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ is satisfiable iff $db(\mathcal{A})$ is a model of $\langle cln(\mathcal{T}), \mathcal{A})\rangle$ [2] and a contradiction on a *DL-Lite* KB may hold only if a membership assertion in the ABox contradicts a functionality assertion or a negative inclusions (NI) implied by the closure $cln(\mathcal{T})$ [2], we think that $db(\mathcal{A})$ and $\langle cln(\mathcal{T}), \mathcal{A}\rangle$ may be useful for measuring the inconsistency degree. We plan to do as follows:

To compute all membership assertions that maybe cause inconsistency. Consider a *DL-Lite* ontology $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ without functional assertions, we find that all conflicting assertions in $\langle cln(\mathcal{T}), \mathcal{A}\rangle$ obtained through a three-valued preferred model with the domain $\Delta^{db(\mathcal{A})}$ are also in $\mathcal{A}$. Furthermore, we find that $\mathcal{K}$ will be consistent when we remove from $\mathcal{K}$ the conflict assertions of $\langle cln(\mathcal{T}), \mathcal{A})\rangle$.

Now we discuss functional assertions. Consider a *DL-Lite* ontology $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$, where $\mathcal{T}$ only contains functional assertions. If $\mathcal{K}$ is inconsistent, we can know that there are some membership assertions in $\mathcal{A}$ contradicting with some functional assertions. By computing preferred models of $\mathcal{K}$ with any domain, we find that the conflict sets obtained in any preferred model with respect to different domains are the same and assertions in conflict sets are all in the form $= (a, b)$ [1]. For example, for $\mathcal{K} = \{(FunctR), R(a, b), R(a, c)\}$, the conflict set is $\{= (b, c)\}$. We also note that the conflict set is only related with $\mathcal{A}$, so we have the following definition.

[1] We enforce the unique name assumption on constants [1].

Definition 1 (Inconsistency Degree) *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite$_{\mathcal{R}}$ or a DL-Lite$_{\mathcal{F}}$ ontology, and let I be a three-valued preferred model of $\langle cln(\mathcal{T}), \mathcal{A} \rangle$ w.r.t $\Delta^{db(\mathcal{A})}$. The inconsistency degree of DL-Lite$_{\mathcal{R}}$ ontology (resp., DL-Lite$_{\mathcal{F}}$ ontology), called $OntoInc\,(\mathcal{K})$, is defined as: $OntoInc(\mathcal{K}) = \frac{|\mathit{ConSet}(I, \langle cln(\mathcal{T}), \mathcal{A} \rangle)|}{|GroundSet(\mathcal{K})|}$ (resp., $OntoInc(\mathcal{K}) = \frac{|\mathit{ConSet}(I, \langle cln(\mathcal{T}), \mathcal{A} \rangle)| + ConEqus(I, \langle cln(\mathcal{T}), \mathcal{A} \rangle)}{|GroundSet(\mathcal{K})|}$), where ConSet$(I, \langle cln(\mathcal{T}), \mathcal{A} \rangle)$ is the set of conflicting atomic individual assertions in $\langle cln(\mathcal{T}), \mathcal{A} \rangle$ regardless of functional assertions and $ConEqus(I, \langle cln(\mathcal{T}), \mathcal{A} \rangle)$ is the set of conflicting assertions obtained through functional assertions and $GroundSet(\mathcal{K})$ is the collection of all possible atomic individual assertions.*

From Definition 1, we can see that the inconsistency degree is only a single value. The next step is only to compute a preferred model of $\langle cln(\mathcal{T}), \mathcal{A} \rangle$ with the domain $\Delta^{db(\mathcal{A})}$ so as to obtain $ConSet(I, \langle cln(\mathcal{T}), \mathcal{A} \rangle)$ and $ConEqus(I, \langle cln(\mathcal{T}), \mathcal{A} \rangle)$.

4.3 Algorithm to compute a preferred model of $\langle cln(\mathcal{T}), \mathcal{A} \rangle$

Using the property that $cln(\mathcal{T})$ only contains NIs and functional assertions and the definition of chase in [2], we can give a definition of chase which mainly adapt to $\langle cln(\mathcal{T}), \mathcal{A} \rangle$ which is finite. Then based on this chase, we give an algorithm to compute a three-valued model of $\langle cln(\mathcal{T}), \mathcal{A} \rangle$ which is a preferred model of $\langle cln(\mathcal{T}), \mathcal{A} \rangle$ with the domain $\Delta^{db(\mathcal{A})}$. So we can obtain $ConSet(I, \langle cln(\mathcal{T}), \mathcal{A} \rangle)$ and $ConEqus(I, \langle cln(\mathcal{T}), \mathcal{A} \rangle)$. We show that the complexity of the algorithm is in PTime. Finally, we can obtain the inconsistent degree of $\langle \mathcal{T}, \mathcal{A} \rangle$ based on Definition 1.

4.4 Evaluation

We plan to implement the algorithm for measuring the inconsistency degree proposed above and give a prototype. Meanwhile we will give some potential applications of our tool. For example, users can input some *DL-Lite* ontologies and obtain the inconsistency degree of each ontology; users can either resolve inconsistency if the degree is high (e.g. greater than 0.7 [10]) or ignore it.

5 Conclusion

Dealing with inconsistency in ontologies is an important topic in the Semantic Web. In our paper, we mainly discuss how to handle inconsistency in *DL-Lite* ontologies. We first give some proposals for our work, then we illustrate how to measure inconsistency in a *DL-Lite* ontology based on a three-valued semantics.

References

1. Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider. Description logic terminology. In F. Baader et al., editor, *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

2. Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. Autom. Reasoning*, 39(3):385–429, 2007.
3. Xi Deng, Volker Haarslev, and Nematollaah Shiri. Measuring inconsistencies in ontologies. In E. Franconi, M. Kifer, and W. May, editors, *Proceedings of the 4th European Semantic Web Conference*, pages 326–340, 2007.
4. Gerhard Friedrich and Kostyantyn Shchekotykhin. A general diagnosis method for ontologies. In Y. Gil, E. Motta, V. Richard Benjamins, and M. A. Musen, editors, *Proceedings of the 4th International Semantic Web Conference*, pages 232–246, 2005.
5. Zhisheng Huang, Frank van Harmelen, and Annette ten Teije. Reasoning with inconsistent ontologies. In L. Pack Kaelbling and A. Saffiotti, editors, *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 454–459, 2005.
6. Domenico Lembo and Marco Ruzzi. Consistent query answering over description logic ontologies. In M. Marchiori, J.Z. Pan, and C. de Sainte Marie, editors, *Proceedings of the 1st International Conference on Web Reasoning and Rule Systems*, pages 194–208, 2007.
7. Yue Ma, Pascal Hitzler, and Zuoquan Lin. Algorithms for paraconsistent reasoning with OWL. In E. Franconi, M. Kifer, and W. May, editors, *Proceedings of the 4th European Semantic Web Conference*, pages 399–413, 2007.
8. Yue Ma, Pascal Hitzler, and Zuoquan Lin. Paraconsistent reasoning for expressive and tractable description logics. In F. Baader, C. Lutz, and B. Motik, editors, *Description Logics*, 2008.
9. Yue Ma, Guilin Qi, Pascal Hitzler, and Zuoquan Lin. An algorithm for computing inconsistency measurement by paraconsistent semantics. In K. Mellouli, editor, *Proceedings of the 9th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 91–102, 2007.
10. Yue Ma, Guilin Qi, Pascal Hitzler, and Zuoquan Lin. Measuring inconsistency for description logics based on paraconsistent semantics. In K. Mellouli, editor, *Proceedings of the 9th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 30–41, 2007.
11. Bijan Parsia, Evren Sirin, and Aditya Kalyanpur. Debugging OWL ontologies. In A. Ellis and T. Hagino, editors, *Proceedings of the 14th International World Wide Web Conference*, pages 633–640, 2005.
12. Peter F. Patel-Schneider. A four-valued semantics for terminological logics. *Artif. Intell.*, (3):319–351, 1989.
13. Guilin Qi, Peter Haase, Zhisheng Huang, Qiu Ji, Jeff Z. Pan, and Johanna Völker. A kernel revision operator for terminologies - algorithms and evaluation. In A. P. Sheth et al., editor, *Proceedings of the 7th International Semantic Web Conference*, pages 419–434, 2008.
14. Guilin Qi and Anthony Hunter. Measuring incoherence in description logic-based ontologies. In K. Aberer et al., editor, *Proceedings of the 6th International Semantic Web Conference*, pages 381–394, 2007.
15. Guilin Qi, Weiru Liu, and David A. Bell. Knowledge base revision in description logics. In M. Fisher et al., editor, *Proceedings of the 10th European Conference on Logics in Artificial Intelligence*, pages 386–398, 2006.
16. Stefan Schlobach. Diagnosing terminologies. In M. M. Veloso and S. Kambhampati, editors, *Proceedings of the 20th National Conference on Artificial Intelligence*, pages 670–675, 2005.
17. Stefan Schlobach and Ronald Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In G. Gottlob and T. Walsh, editors, *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 355–362, 2003.
18. Stefan Schlobach, Zhisheng Huang, Ronald Cornet, and Frank van Harmelen. Debugging incoherent terminologies. *J. Autom. Reasoning*, 39(3):317–349, 2007.