Inconsistences in Hybrid Knowledge Bases

Daria Stepanova
Institute of Information Systems, Vienna University of Technology
Favoritenstraße 9-11, A-1040 Vienna, Austria
dasha@kr.tuwien.ac.at

Abstract

Hybrid knowledge bases (KBs) are a well-known paradigm for combining nonmonotonic rules and description logic ontologies, which gained increasing attention in the past years. Such hybrid KBs have the potential of becoming very powerful tools for knowledge representation and reasoning.

However, the interaction between rules and ontologies might cause undesired inconsistencies, making the whole system unusable. The research area in which the thesis will be stated is devoted to the investigation of the notion of inconsistencies and development of the methodology for dealing with them within hybrid KBs.

1 Introduction and Problem Description

The increasing popularity of the World Wide Web and distributed systems has created the need for hybrid formalisms that combine nonmonotonic rules and description logic ontologies. Various approaches for such combination have been proposed (see (de Bruijn et al. 2009) for overview).

In this thesis we focus on Description Logic (DL-) programs (Eiter et al. 2008), which couple rules and ontologies in a loose manner. The data sources in DL-programs are treated separately, but a bidirectional information flow between them is arranged. This makes them highly expressive systems capable of solving complex reasoning problems on top of ontologies. The interaction between rules and ontologies, however, might give rise to undesired inconsistencies, making the whole system broken and thus unusable.

Unfortunately, currently available systems suffer from the inability to deal with inconsistencies with ease, which forms a major obstacle for their wider acceptance. The development of a sophisticated framework for handling inconsistencies in DL-programs is therefore the overall goal of this PhD project.

2 State of the Art

Description Logics (DLs) are applied as a mature logical formalism, which is focused on specifying and reasoning in relation to conceptual knowledge. *Rules* in the sense of logic programming are a well-accepted paradigm for declarative problem solving designed to target issues associated with nonmonotonic inference.

Many applications, however, require the combination of features of both DLs and rules. Thus, the natural solution of combining these formalisms has given rise to the notion of *Hybrid Knowledge Bases* (Lu, Nerode, and Subrahmanian 1996).

Figure 1: DL-program Π over a family ontology

$$\mathcal{O} = \left\{ \begin{array}{ll} (1)\ \mathit{Child} \sqsubseteq \exists.\mathit{hasParent} & (4)\ \mathit{Male}(\mathit{pat}) \\ (2)\ \mathit{Adopted} \sqsubseteq \mathit{Child} & (5)\ \mathit{Male}(\mathit{john}) \\ (3)\ \mathit{Female} \sqsubseteq \neg \mathit{Male} & (6)\ \mathit{hasParent}(\mathit{john}, \mathit{pat}) \end{array} \right\}$$

$$P = \left\{ \begin{array}{l} (7)\ \mathit{ischildof}(\mathit{john}, \mathit{alex}); \quad (8)\ \mathit{boy}(\mathit{john}); \\ (9)\ \mathit{hasfather}(X, Y) \leftarrow \mathrm{DL}[\mathit{Male} \uplus \mathit{boy}; \mathit{Male}](Y) \\ \qquad\qquad\qquad\qquad\qquad \mathrm{DL}[; \mathit{hasParent}](X, Y); \\ (10)\ \bot \leftarrow \mathrm{not}\ \mathrm{DL}[; \mathit{Adopted}](X), Y_1 \neq Y_2, \\ \qquad\qquad \mathit{hasfather}(X, Y_1), \mathit{ischildof}(X, Y_2), \\ \qquad\qquad \mathrm{not}\ \mathrm{DL}[\mathit{Child} \uplus \mathit{boy}; \neg \mathit{Male}](Y_2); \end{array} \right\}$$

Informally, a hybrid KB can be viewed as a pair $\Pi = (\mathcal{O}, P)$, where $\mathcal{O}$ is a DL-based ontology and P is a set of logical rules. Approaches for defining hybrid KBs include tight coupling (SWRL (Horrocks et al. 2004), r-hybrid KBs (Rosati 2005), etc.), embedding (MKNF KBs (Motik and Rosati 2010), G-hybrid KBs (Heymans et al. 2006), etc.) and loose coupling (DL-programs (Eiter et al. 2008) and F-Logic KBs (Heymans et al. 2010)). The tight coupling approaches define the interface based on common models. The embedding approaches define the interface based on embeddings of both the ontology and the rules in a single unifying non-monotonic formalism. In the loose coupling approach (de Bruijn et al. 2009) the ontology and the rules act separately but communicate via a well-defined interface.

In this thesis, we mostly look at loosely coupled hybrid KBs and study DL-programs (Eiter et al. 2008) as a prominent example which realizes this approach.

Example 1 *Consider the DL-program* Π *in Figure 1, representing information about children of a primary school and their parents. It consists of an ontology* $\mathcal{O}$ *with a taxonomy* $\mathcal{T}$ *of concepts in (1)-(3) and a data part (i.e., assertions)* $\mathcal{A}$ *about some individuals in (4)-(6). The rules* P *contain facts (7), (8) and logic rules: (9) identifies fathers from the ontology, upon some feeding information; (10) checks the constraint that a child has for sure at most one father, unless it is adopted.*

The bidirectional flow of information in the DL-programs between the logic program and the DL ontology is achieved via so-called DL-atoms. For instance, a DL-atom

$DL[Male \uplus boy; Male](t)$ first enriches the concept $Male$ of $\mathcal{O}$ by constants that are in the rule predicate boy and then queries the updated ontology for all assertions $Male(t)$. The semantics of DL-programs is given in terms of answer sets which are minimal models of the program reduct. In this work we look at weak and flp-answer sets (see (Eiter et al. 2005) for formal definitions).

Unfortunately, the interplay between rules and ontology in DL-programs might often lead to inconsistencies, i.e. absence of an answer set. This is observed in Example 1, where the constraint (10) is applied since $john$ being not adopted has pat as a father according to the ontology and also possibly $alex$ due to the rule part.

Regardless of the reasons for inconsistencies in logical formalisms all approaches for dealing with them fall into two groups. Methods from the first group diagnose and repair inconsistenies. Approaches from the second group simply avoid inconsistencies and apply non-standard reasoning methods in order to obtain meaningful answers. A good overview of consistency-related issues in different formalisms is given in (Fink et al. 2011).

Inconsistencies in ontologies Approaches for fixing inconsistent ontologies have been studied in many works. For example, a suitable technique is proposed in (Ji et al. 2009), which generates minimal inconsistent subsets and removes them from ontology. A similar approach reported in (Haase et al. 2005) provides methodologies for extracting minimal inconsistent and maximal consistent subontologies. The authors of (Hussain et al. 2011) deal with the identification of contradiction derivations under the integrity constraint rules defined in a logic program. The recent works (Lembo et al. 2011; Bienvenu 2012) study consistent query answering in ontologies.

Inconsistencies in Rules The efforts towards detecting and solving inconsistencies in logic rules are mostly described in the papers that focus on debugging of logic programs (Gebser et al. 2008; Syrjänen 2006). The approach by Syränen (Syrjänen 2006) addresses the issue of debugging incoherent logic programs, which is adapted from the field of symbolic diagnosis (Reiter 1987). A generalization of the same problem is given in the work (Gebser et al. 2008), which provides explanations why interpretations are not answer sets of a program under consideration. The consistency-restoring rules of Balduccini and Gelfond (Balduccini and Gelfond 2003) form another related approach in this regard.

In the area of paraconsistent reasoning for logic programs the most prominent works include (Przymusinski 1991; Sakama and Inoue 1995). For more discussion see (Eiter, Leone, and Saccà 1997). In the work (Eiter, Fink, and Moura 2010) a semantic characterization of semi-stable models in terms of bi-models and of semi-equilibrium models is given. A recent work (Denecker, Bruynooghe, and Vennekens 2012) surveys and explains the application of the approximation fixpoint theory to the semantics of logic programming and answer set programming and generalizations of these.

Inconsistencies in Combination of Logical Formalisms In general, combining different pieces of knowledge is more vulnerable for contradiction than individual representations. The important results in this respect have been obtained in (Eiter et al. 2010; Brewka and Eiter 2007; Bögl et al. 2010) where the authors focused on explaining inconsistency in multi-context systems (MCS), in which decentralized and heterogeneous system parts interact via nonmonotonic bridge rules. The techniques described in (Eiter et al. 2010) characterize inconsistency in terms of involved bridge rules: either by pointing out rules which need to be altered for restoring consistency, or by finding combinations of rules which cause inconsistency.

The problem of inconsistencies in DL-programs, is touched in the work (Pührer, Heymans, and Eiter 2010). The semantics that is sensitive for inconsistencies caused by DL-atoms was introduced there. Some preliminary work on defining the notion of diagnosis and diagnostic repair for inconsistent DL-programs is presented in (Puehrer et al. 2011). The rough idea is to apply minimal changes to the ontology s.t. the resulting DL-program is consistent.

Despite being an important problem for many applications, paraconsistent reasoning in the context of loosely coupled knowledge formalisms has been barely explored to the best of our knowledge. In the tight coupled formalisms the framework for paraconsistent reasoning is proposed in (Fink 2012; Ma, Qi, and Hitzler 2011).

3 Goal of the Research

In addressing the engineering requirements of DL-programs as argued above, the main goal of this project is to develop a powerful framework for handling inconsistencies in the considered formalism. We strive to formulate theoretical approaches for handling inconsistencies as well as to develop concrete tools which would help to materialize the developed techniques.

The key objectives of our theoretical work shall more specifically comprise of the following goals:

- *Repair Semantics for Inconsistent DL-programs*
  The first goal of this project is to develop a repair semantics for inconsistent DL-programs and perform its extensive complexity analysis.
- *Algorithm Design*
  The second goal of our project is to design practically relevant algorithms for repairing inconsistent DL-programs and propose their possible optimizations.

On the practical side, the research work will focus on bringing together the advanced results and methodologies developed within the project into a coherent system architecture. In particular, the following practical issues will be solved:

- *Implementation*
  As a proof-of-concept, one of our major goals is the implementation of the new algorithms emerging from our research and the incorporation of those into the DLVHEX system (Eiter et al. 2006). The practical techniques require a lot of investigation since such system plug-in will be implemented from scratch.
- *Evaluation and Experiments*
  The last but not least goal is the development of benchmarks for evaluation of the methods elaborated within the PhD project.

4 Preliminary Results Accomplished

In our research we assume that the ontology and the rules of the DL-program are consistent when considered separately and the inconsistencies arise as a result of their combination. The reasons for inconsistencies therefore lie in the wrong values of some DL-atoms occurring in the DL-program.

We started the work by identifying DL-atoms which have always the same value regardless of the ontology and interpretation of the DL-program at hand. Knowledge about such DL-atoms helps to decide whether a DL-program repair exists and it can also be used for optimization purposes. We called such DL-atoms independent and developed a sound and complete calculus for their derivation (Eiter, Fink, and Stepanova 2012). More specifically, independent DL-atoms fall into two categories: tautologic and contradictory. We have shown that checking whether a given DL-atom is independent can be done efficiently.

Moreover, on the theoretical level we have formalized the problem of repairing DL-programs and introduced the notions of repair and repair answer set (Eiter, Fink, and Stepanova 2013). We assumed that the rule part of the DL-program and the ontology TBox are well-developed (as it indeed often happens) and the reasons for inconsistencies lie in the ontology ABox. The novel notions of repair and repair answer set are therefore based on changes of the ontology data part that enable answer sets. For instance, deletion of $hasParent(john, pat)$ from $\mathcal{A}$ in Example 1 leads to a repair $\mathcal{A}' = \{Male(pat), Male(john)\}$ under which $I' = \{ischildof(john, alex), boy(john)\}$ is an flp-repair answer set.

We have shown that repair answer sets do not have higher complexity than ordinary ones (more specifically, weak and FLP answer sets) in case if queries in DL-atoms can be evaluated in polynomial time. To ensure this property, we concentrated on the Description Logic $DL\text{-}Lite_{\mathcal{A}}$ (Calvanese et al. 2007), which is a prominent DL particularly useful for ontology based data access (OBDA).

As clearly not all repairs are equally attractive for a given scenario, in order to distinct between the repairs we introduced the preference relation realized by a selection function σ. The latter selects preferred repairs from a set of all candidates. We studied selection functions that do not introduce additional complexity for computing preferred repair answer sets, e.g. bounded $\delta^{\pm}$-change, deletion, addition under bounded opposite polarity and others. For instance, while $\mathcal{A}'$ from above is a deletion repair for Π, the ABox $\mathcal{A}'' = \{Male(pat), Male(john), hasParent(john, pat), female(alex)\}$ satisfies the criteria for being a k-bouded addition repair, if $k \geq 1$. The repair $\mathcal{A}''$ makes $I'' = I' \cup \{hasfather(john, pat)\}$ a repair answer set for Π.

We showed how an algorithm for evaluating DL-programs (Eiter et al. 2005) can be extended to compute repairs resp. repair answer sets, with possibly integrated selection criteria. The evaluation of the DL-program Π is based on the program rewriting $\hat{\Pi}$, where DL-atoms are substituted by normal atoms and additional guessing rules on their values are added to the program. The answer set of the rewritten program is also an answer set of the original DL-program if the real values of all DL-atoms coincide with the guessed ones and the minimality check succeeds. While adapting this approach for the repair computation a novel interesting *generalized ontology repair problem* (ORP) was introduced. The latter is based on an answer set candidate and DL-atoms of the program. The solution to an ORP is an ABox which ensures simultaneous entailment and non-entailment of sets of queries under possible updates.

The naive implementation of the repair answer set computation goes through all answer sets I of the replacement program $\hat{\Pi}$ and checks all possible ABoxes to see whether under any of them I becomes an answer set of Π. While natural this approach does not to scale well for practical application since the number of ABoxes to be checked might be large in general.

Therefore, we proposed an alternative improved approach for repair computation which is based on the notion of *support sets*. Intuitively, a support set for a ground DL-atom $a = \mathrm{DL}[\lambda; Q](t)$ is a part of its input which together with the ontology TBox is sufficient for $Q(t)$ to be derived. Our method is to precompute small support sets for all DL-atoms on a nonground level by exploiting TBox classification. Then for each candidate interpretation the ground instantiations of support sets are effectively obtained. These help to prune the search space of the model candidates and also to construct the ABox repair. The described approach is particularly attractive for $DL\text{-}Lite_{\mathcal{A}}$ ontologies, for which support sets are small and easily computable.

We implemented the above algorithm as part of the DLVHEX system and evaluated it on a number of benchmark scenarios. The results of the experiments proved effectiveness of our approach.

5 Open Issues and Expected Achievements

For future work in remains to improve the implementation of the inconsistency framework to enable computing different σ-preferred repairs for DL-programs. We plan to consider maximal deletion repairs, repairs with at most k new assertions, discussed in (Eiter, Fink, and Stepanova 2013). Moreover, we will realize other practically attractive criterias for repair selection. For example, one might be interested in deleting only assertions with certain ontology predicates or to restrict repairs to selected individuals.

So far we have focused on DL-programs with ontologies in $DL\text{-}Lite_{\mathcal{A}}$. Another issue for further research is to look at other DLs, such as $\mathcal{EL}$.

Even though the evaluation of the developed inconsistency framework has been already started, a further in-depth analysis is still required. The latter involves creation of benchmarks from real world applications, which is nontrivial since no compared benchmarks exist.

Acknowledgements I would like to thank my supervisor Prof. Dr. Thomas Eiter and Dr. Michael Fink for their ongoing support.

References

Balduccini, M., and Gelfond, M. 2003. Logic programs with consistency-restoring rules. In *AAAI 2003 Spring Symposium Series*, 9–18.

Bienvenu, M. 2012. On the complexity of consistent query answering in the presence of simple ontologies. In *AAAI*. AAAI Press.

Bögl, M.; Eiter, T.; Fink, M.; and Schüller, P. 2010. The mcs-ie system for explaining inconsistency in multi-context systems. In *JELIA*, volume 6341 of *Lecture Notes in Computer Science*, 356–359. Springer.

Brewka, G., and Eiter, T. 2007. Equilibria in heterogeneous nonmonotonic multi-context systems. In *AAAI*, 385–390. AAAI Press.

Calvanese, D.; Giacomo, G. D.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2007. Tractable reasoning and efficient query answering in description logics: The *dl-lite* family. *J. Autom. Reasoning* 39(3):385–429.

2005. *IJCAI-05, Proc. of the 19th Int. Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30-August 5, 2005*, Professional Book Center.

2010. *Proc. of the 20th Int. Conf., KR 2010, Toronto, Ontario, Canada, May 9-13, 2010*, AAAI Press.

de Bruijn, J.; Bonnard, P.; Citeau, H.; Dehors, S.; Heymans, S.; Pührer, J.; and Eiter, T. 2009. Combinations of rules and ontologies: State-of-the-art survey of issues. Technical Report Ontorule D3.1, Ontorule Project Consortium. http://ontorule-project.eu/.

Denecker, M.; Bruynooghe, M.; and Vennekens, J. 2012. Approximation fixpoint theory and the semantics of logic and answers set programs. In *Correct Reasoning*, volume 7265 of *Lecture Notes in Computer Science*, 178–194. Springer.

Eiter, T.; Ianni, G.; Schindlauer, R.; and Tompits, H. 2005. A uniform integration of higher-order reasoning and external evaluations in answer-set programming. In *IJCAI* (2005), 90–96.

Eiter, T.; Ianni, G.; Schindlauer, R.; and Tompits, H. 2006. Effective integration of declarative rules with external evaluations for semantic-web reasoning. In *ESWC*, volume 4011 of *LNCS*, 273–287. Springer.

Eiter, T.; Ianni, G.; Lukasiewicz, T.; Schindlauer, R.; and Tompits, H. 2008. Combining answer set programming with description logics for the Semantic Web. 172(12-13):1495–1539.

Eiter, T.; Fink, M.; Schüller, P.; and Weinzierl, A. 2010. Finding explanations of inconsistency in multi-context systems. In *KR* (2010).

Eiter, T.; Fink, M.; and Moura, J. 2010. Paracoherent answer set programming. In *KR* (2010).

Eiter, T.; Fink, M.; and Stepanova, D. 2012. Semantic independence in dl-programs. In *RR*, volume 7497 of *Lecture Notes in Computer Science*, 58–74. Springer.

Eiter, T.; Fink, M.; and Stepanova, D. 2013. Data repair of inconsistent dl-programs. In *IJCAI*. IJCAI/AAAI.

Eiter, T.; Leone, N.; and Saccà, D. 1997. On the partial semantics for disjunctive deductive databases. *Ann. Math. Artif. Intell.* 19(1-2):59–96.

Fink, M.; Ghali, A. E.; Chniti, A.; Korf, R.; Schwichtenberg, A.; Lévy, F.; Pührer, J.; and Eiter, T. 2011. D2.6 Consistency maintenance. Final report. Technical Report 2.6, ONTORULE ICT-2009-231875 Project.

Fink, M. 2012. Paraconsistent hybrid theories. In *KR*, 391–401. AAAI Press.

Gebser, M.; Pührer, J.; Schaub, T.; and Tompits, H. 2008. A meta-programming technique for debugging answer-set programs. In *AAAI*, 448–453. AAAI Press.

Haase, P.; van Harmelen, F.; Huang, Z.; Stuckenschmidt, H.; and Sure, Y. 2005. A framework for handling inconsistency in changing ontologies. In *Int. Semantic Web Conference*, volume 3729 of *Lecture Notes in Computer Science*, 353–367. Springer.

Heymans, S.; Prediou, L.; Feier, C.; de Bruijn, J.; and van Nieuwenborgh, D. 2006. G-hybrid knowledge bases. In *Proc. of ALPSWS 2006*.

Heymans, S.; Korf, R.; Erdmann, M.; Pührer, J.; and Eiter, T. 2010. F-logic#: Loosely coupling f-logic rules and ontologies. In *Web Intelligence*, 248–255. IEEE.

Horrocks, I.; Patel-Schneider, P. F.; Boley, H.; Tabet, S.; Grosof, B.; and Dean, M. 2004. Swrl: A semantic web rule language combining owl and ruleml. Technical report, World Wide Web Consortium.

Hussain, S.; Roo, J. D.; Daniyal, A.; and Abidi, S. S. R. 2011. Detecting and resolving inconsistencies in ontologies using contradiction derivations. In *COMPSAC*, 556–561. IEEE Computer Society.

Ji, Q.; Haase, P.; Qi, G.; Hitzler, P.; and Stadtmüller, S. 2009. Radon - repair and diagnosis in ontology networks. In *ESWC*, volume 5554 of *Lecture Notes in Computer Science*, 863–867. Springer.

Lembo, D.; Lenzerini, M.; Rosati, R.; Ruzzi, M.; and Savo, D. F. 2011. Inconsistency-tolerant semantics for description logic ontologies (extended abstract). In *SEBD*, 287–294.

Lu, J. J.; Nerode, A.; and Subrahmanian, V. S. 1996. Hybrid knowledge bases. *IEEE Trans. Knowl. Data Eng.* 8(5):773–785.

Ma, Y.; Qi, G.; and Hitzler, P. 2011. Computing inconsistency measure based on paraconsistent semantics. *J. Log. Comput.* 21(6):1257–1281.

Motik, B., and Rosati, R. 2010. Reconciling Description Logics and Rules. 57(5):1–62.

Przymusinski, T. C. 1991. Stable semantics for disjunctive programs. *New Generation Comput.* 9(3/4):401–424.

Puehrer, J.; Ghali, A. E.; Chniti, A.; Korf, R.; Schwichtenberg, A.; Lvy, F.; Heymans, S.; Xiao, G.; and Eiter, T. 2011. Ontorule d2.3 Consistency Maintenance. Intermediate Report. Technical Report 2.3.

Pührer, J.; Heymans, S.; and Eiter, T. 2010. Dealing with inconsistency when combining ontologies and rules using dl-programs. In *ESWC (1)*, volume 6088 of *Lecture Notes in Computer Science*, 183–197. Springer.

Reiter, R. 1987. A theory of diagnosis from first principles. *Artif. Intell.* 32(1):57–95.

Rosati, R. 2005. On the decidability and complexity of integrating ontologies and rules. 3(1):41–60.

Sakama, C., and Inoue, K. 1995. Paraconsistent stable semantics for extended disjunctive programs. *J. Log. Comput.* 5(3):265–285.

Syrjänen, T. 2006. Debugging inconsistent answer-set programs. In *Proc. of the 11th Int. Workshop on Nonmonotonic Reasoning, (NMR'06), UK*, 77–83.