Large Object Oriented Knowledge Bases: A Challenge for Reasoning Engines

Vinay K. Chaudhri, Stijn Heymans, and Michael Wessel
AI Center
SRI International
Menlo Park, California, USA

Tran Cao Son
Computer Science Department
New Mexico State University
Las Cruces, New Mexico, USA

Abstract

We announce the availability of KB_Bio_101 for research purposes. We explain the origins of this KB and identify the research problems it poses for the state-of-the-art answer set solvers, first order theorem provers and description logic reasoners.

The Knowledge Base KB_Bio_101

The goal of Project Halo[1] is to develop a "Digital Aristotle" — a reasoning system capable of answering novel questions and solving advanced problems in a broad range of scientific disciplines and related human affairs. As part of this effort, SRI has created a knowledge base called KB_Bio_101that represents knowledge from a textbook used for advanced high school and introductory college biology courses. The KB_Bio_101 contains a concept taxonomy for the whole textbook and detailed rules for 20 chapters of the textbook. SRI has tested the educational usefulness of this knowledge base in the context of an electronic version of the book as it is used by students studying from that book[2].

The KB_Bio_101 was originally developed using a knowledge representation and reasoning system called Knowledge Machine (KM) (Clark and Porter 2011). KM supports a variety of representation features that include a facility to define classes and organize them into a hierarchy and define partitions, ability to define relations (also known as slots) and organize them into a relation hierarchy, support for nominals, a facility to define Horn rules, a procedure language, a situation mechanism, and a STRIPS representation for actions. KM performs reasoning by using inheritance, description-logic style classification of individuals, backward chaining over rules, and a heuristic unification. KM supports para-consistent reasoning[3] in the sense that it can perform reasoning even in the face of inconsistencies in the KB. In addition, KM can use its situation mechanism and STRIPS representation of actions to simulate their execution. While the project team has experimented with the use of all of these features, the KB_Bio_101 does not leverage the STRIPS features of KM. We have just completed work to export the KB_Bio_101 in a variety of standard declarative languages, for example, first order logic with equality (Fitting 1996), SILK (Grosof 2009), description logics (Baader et al. 2007) and logic programming under the answer set semantics (Gelfond and Lifschitz 1990).

The KB_Bio_101 is encoded as an object oriented knowledge base (Chaudhri et al. 2013a). The current KB has more than 6000 classes, 6500 subclass and disjointness relationships in the class hierarchy, and several hunderd thousands rules (axioms). The KB_Bio_101 is now freely available for research purposes[4].

Reasoning Problems in KB_Bio_101

The basic reasoning problems in KB_Bio_101 are grouped into different types as follows:

- $\mathbf{Q_1}$: Querying about classes and subclass relations
- $\mathbf{Q_2}$: Querying about properties of individuals
- $\mathbf{Q_3}$: Comparing individuals between classes
- $\mathbf{Q_4}$: Searching for a path with some specified relations between two classes

The first two types of queries focus on the taxonomical hierarchy described by the KB and the last two on the relationships between individuals of one or more classes. Each of these query types can have numerous question templates. For example, for the first query type some example question templates are: What are the subclasses of X? Is it true that class X is a subclass of class Y? Is it true that X and Y are disjoint? etc. Defining numerous question templates for each type of query allows us to capture a large space of queries that the users are interested in asking of the KB_Bio_101. For example,

- is it true that a cell with a nucleus is a prokaryotic cell?
- what are the types of exergonic reactions?
- what are organelle parts of a cell?
- describe the differences and similarities between mitochondria and chloroplasts;

[1] http://www.projecthalo.com/

[2] http://www.aaaivideos.org/2012/inquire_intelligent_textbook/

[3] http://plato.stanford.edu/entries/logic-paraconsistent/

[4] http://www.ai.sri.com/~halo/public/exported-kb/biokb.html

- what process provides raw materials for the citric acic cycle during *cellular respiration*?
- in the absence of oxygen, yeast cells can obtain energy by *which process*?

The detailed input and a possible way for computing the output for each of these queries are given in (Chaudhri et al. 2013b). The current reasoning on KB_Bio_101 is done using KM and implements special algorithms for answering queries of the types $\mathbf{Q_1}$-$\mathbf{Q_4}$. In some queries, only approximated answers are provided. The current reasoning engine also employs a heuristic called unification mapping (UMAP) to unify terms representing objects (Chaudhri and Son 2012).

Challenges for AI-Reasoners

Reasoning in KB_Bio_101 poses a challenge for state-of-the-art AI-reasoning engines for the following reasons:

- The KB_Bio_101 contains rules with function symbols for which the grounding is infinite. A simple example is a KB consisting of a single class person, and a single relation has-parent, and a statement of the form "for each person there exists an instance of the has-parent relation between this person with another individual who is also a person". The skolemized versions of these statements require function symbols. An obvious first challenge that must be addressed is to develop suitable grounding techniques.
- The rules in KB_Bio_101 can define the necessary and sufficient properties of a class that are structured as general graphs as opposed to trees. Furthermore, the class definitions can be circular in that they can refer to each other. Use of graph structures in class descriptions frequently causes undecidability in description logic systems (Motik et al. 2009). Therefore, the computation for queries Q_1 and Q_2 is likely to be intractable.
- Even though rules in KB_Bio_101 follow a small number of axiom templates, the size of this KB indicates that this could be a non-trivial task for state of the art reasoners.
- KB_Bio_101 contains more than 100,000 non-ground rules specifying equality between individual terms. This is because KB_Bio_101 is a fully-specified knowledge base in the sense discussed in (Chaudhri and Son 2012). Computing these rules in each export is a time consuming and complex process. Furthermore, the approach is also not elaboration tolerant in the sense that these equality relations need to be maintained as the knowledge base is updated. A better approach in dealing with underspecification is to use unification mapping (UMAP), as proposed in (Chaudhri and Son 2012). Let us denote with KB_Bio_101^{neq} the KB obtained from KB_Bio_101 by removing the rules for specifying the equality relation. The rules developed for UMAP aim at enforcing the following principles:

(**P1**) *Specificity principle*: in selecting terms for the construction of *umap*-atoms, more specific terms should be preferred over less specific ones.

(**P2**) *Specialization Principle*: Given a relation s and a class c, the application of the specificity principle should be limited to at most one possible value of s at c. Furthermore, if the application of (**P1**) does not violate (**P2**) then (**P1**) should be applied.

(**P3**) *Redundancy Principle*: In the presence of multiple specifications of a relation for an individual, the most-specific relation specification overrides less-specific ones.

(**P4**) *Consistency Principle*: If a unification between x and y takes place at class c then it should be applied in every slot of class c.

Answering queries of the form $\mathbf{Q_1}-\mathbf{Q_4}$ in KB_Bio_101^{neq} would require reasoning with rules for UMAP which is a computationally intensive task. The reason lies in that the UMAP needs to consider the combinatorics of equating different individuals across the class hierarchy.

- The reasoning tasks of computing differences between two concepts and finding relationships between two individuals are computationally intensive tasks. Previous implementations of these tasks rely on graph algorithms and trade completeness for efficiency. These tasks will present a tough challenges to any reasoner.

Benefits

Successfully dealing with the challenging problems will benefit AI-reasoning systems in several ways.

- If current approach—grounding then solving—is maintained in answer set solvers, contemporary reasoners need to develop new grounding techniques to cope with millions of non-ground rules. As such, successfully dealing with the reasoning problems of KB_Bio_101 will provide us with reasoning engines with new grounding technique that scale up to such reasoning.
- The KB presents a real and practical challenge for developers of novel reasoners for non-propositional logical theories (e.g., language with existential quantifiers or function symbols).
- The KB_Bio_101 requires the expressiveness of at least the SHOIQ(Dn) description logic. In addition, it supports graph structured descriptions for which there are no available decidable reasoners. Therefore, it provides an ideal use cases to explore the boundaries of decidable reasoning between different knowledge representation languages (e.g., description logics vs. logic programs).
- The KB_Bio_101 has been posed as a challenge for description logic reasoners (Chaudhri et al. 2013c). It is also expected to be included in the TPTP library (Chaudhri et al. 2013d). It could, thus, be a good shared task for the reasoner olympics to be held as part of Vienna Summer of Logic[5].

[5] http://vsl2014.at/

Acknowledgments

This work has been funded by Vulcan Inc. and SRI International.

References

Baader, F.; Calvanese, D.; McGuinness, D. L.; Nardi, D.; and Patel-Schneider, P. F., eds. 2007. *The Description Logic Handbook: Theory, Implementation and Applications, 2nd Edition*. Cambridge University Press.

Chaudhri, V. K., and Son, T. 2012. Specifying and reasoning with underspecified knowledge bases using answer set programming. In *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning*.

Chaudhri, V. K., Heymans, S., Wessel, M., and Son, T. 2013. Object-Oriented Knowledge Bases in Logic Programming, In *In Technical Communications of the International Conference on Logic Programming*.

Chaudhri, V.; Heymans, S.; Wessel, M.; and Son, T. C. 2012. Query Answering in Object Oriented Knowledge Bases in Logic Programming: Description and Challenge for ASP. In *Workshop on Answer Set Programming and Other Computing Paradigms*, 2013.

Chaudhri, V.; Heymans, S.; Wessel, M. 2013. KB_Bio_101: A Challenge for OWL Reasoners In *Workshop on Evaluating OWL Reasoners*, 2013.

Chaudhri, V.; Heymans, S.; Wessel, M. 2013. KB_Bio_101: A Challenge for TPTP Reasoners In *CADE Workshop on Knowledge Intensive Reasoning*, 2013.

Clark, P., and Porter, B. 2011. *KM (v2.0 and later): Users Manual*.

Motik, B. ; Grau, C.;Horrocks, I.; Sattler U. Representing ontologies using description logics, description graphs, and rules. Artificial Intelligence Journal, 173:1275-1309, 2009.

Fitting, M. 1996. *First-Order Logic and Automated Theorem Proving*. Springer.

Gelfond, M., and Lifschitz, V. 1990. Logic programs with classical negation. In Warren, D., and Szeredi, P., eds., *Logic Programming: Proceedings of the Seventh International Conference*, 579–597.

Grosof, B. N. 2009. SILK: Higher Level Rules with Defaults and Semantic Scalability. In Polleres, A., and Swift, T., eds., *Web Reasoning and Rule Systems, Third International Conference, RR 2009, Chantilly, VA, USA, October 25-26, 2009, Proceedings*, volume 5837 of *Lecture Notes in Computer Science*, 24–25. Springer.