A Hybrid DL and RL Based Reasoner for Optimum Entailments in Ontologies

Deepika Chaudhary¹, Jaiteg Singh² and D. P. Kothari³

¹Chitkara University Institute of Engineering and Technology, Chitkara University, Patiala – 140401, Punjab, India; Deepika.chaudhary@chitkara.edu.in
²CURIN, Chitkara University, Patiala – 140401, Punjab, India; Jaiteg.singh@chitkara.edu.in
³S. B. Jain Institute of Technology, Management and Research, Nagpur – 441501, Maharashtra, India; dpkvits@gmail.com

Abstract

Objective: To present a reasoner based on the hybrid approach which integrates the Description Logic (DL) reasoner with Rule Logic (RL) based reasoner for optimizing the reasoning process in ontology’s. Methods/Statistical Analysis: The idea here is to integrate the capabilities of both atomic reasoners. This hybrid reasoner will first load the ontology and make this ontology active; it will then separate the Terminological axioms (Tbox) from the Assertion axiom (Abox). It will thus call the Description Logic (DL) based reasoner for Tbox reasoning which will perform reasoning by classifying and realizing the elements of a given ontology. Afterwards it will upload the rules set for performing Rule based entailments. It will thus perform the assertion reasoning based on rules set and the given axioms. Findings: This approach integrates the effective and efficient inferencing capability of a DL algorithm based reasoner with the scalability of rule based reasoners. The experimental research shows that by following the hybrid approach the entailments can be made more efficient and scalable as compared to the other reasoners. This framework was successful in performing the Tbox entailments by calling the Pellet reasoner and it achieves a faster execution of Abox queries through the use of Generic Rule Reasoner of Jena Framework which perform the entailments in the RDF Schema and the rule set. This approach enriches the reasoning task in two ways first it separates the two schemas of any given ontology and secondly it handles the memory more efficiently. Applications: This framework gives us an efficient way for performing entailments in any kind of ontology whether it is a simple ontology or ontology based on OWL2 profile languages.

Keywords: Description Logic, Entailments, Inference, Jena, Ontology, OWL DL, OWL QL, OWL RL, Pellet, Reasoner

1. Introduction

Since last few years researchers have a heed on ontology’s as it plays a very vital role in semantic web. Ontology is an archive made by a particular vocabulary used to give description about a concept along with some given assumptions regarding its meaning. Ontology is defined as a formal explicit specification of shared conceptualization of domain of interest. Here the words formal explicit specification means that ontology’s in Semantic web can only be useful if it comes with a proper semantics. As per the recommendations of W3C, Description Logic (DLs) is the basic building block of ontology’s whose responsibility is to guard the logics. The Semantic web stack consists of several layers and for each layer there is a defined mark up language and logic.

* Author for correspondence
In this section we discuss the working of each layer with the applied logic. To define the ontology vocabulary layer the very expressive DL’s SHOIN (D) is used. DL’s SHOIN (D) is the decidable fragments of First Order Predicate Logic. If we have a closer look into the Figure 1 we will find that the rules are built on top of the ontology’s. The Horn Clausal Logic (HCL) defines the formal specifications for rules and both these set of logics are incomparable as regard to its expressiveness and semantics. In order to bridge the gap between these two logical layers based on different set of semantics either the homogenous or hybrid approach can be used. In homogenous approach there is no differentiation between the rules and the ontology and both are treated as a single entity and a single logic language is used for inference. The example of one such language is the EOR [Entailment based owl reasoning], the reasoners which work on EOR are OWLIM, OWL Jess KB, CARIN etc. On the other hand in hybrid approach we combine various atomic systems. The systems which are working with two different parts of a same knowledge base i.e. the basic concepts (OWL Vocabulary) and the rule set. Therefore it forms a modular architecture, it combines the DL/RL reasoners to work on the ontology layer and a rule engine to extract and entail the rules of a given ontology. The major objective of this paper is to prove that process of entailment which works on both the layers; ontology and the rules can be improvised in terms of scalability, efficiency and simplicity when by merging the strong points of both these approaches. We here present an analysis of hybrid DL and RL based reasoner to work on the ontology layer and a rule engine to extract and entail the rules of a given ontology. The major objective of this paper is to prove that process of entailment which works on both the layers; ontology and the rules can be improvised in terms of scalability, efficiency and simplicity when by merging the strong points of both these approaches. We here present an analysis of hybrid DL and RL based reasoner which perform entailments on two different kinds of statement: TBOX (Ontology) statements which describe the implied knowledge that is terminological knowledge and ABOX (Rules) statements i.e. describe extensional knowledge about the individuals on OWL2 Profile Languages. The objective of this study is

- To highlight the challenges in reasoning while working with homogenous approach.
- To review the literature and related work in this direction.
- To showcase the working of the proposed approach by explaining the reasoning process and giving an idea of basics of Description Logic. In this section we have also discussed the classification process to perform entailments.
- To present the methodology adopted and to discuss the experimental results of reasoning and at the last we summarize.

### 2. Challenges Faced with Homogenous Approach

In terms of Reasoning, ontology’s exhibit many features. In some cases ontology’s may be defined using very expressive language, in other they can have large number of instances associated with it, also there can be scenarios where the defined ontology’s are highly cyclic in nature. If ontology is a mix of all the above said traits then it would be very difficult to classify such ontology’s through an atomic reasoner which is based on homogenous approach. In Homogenous approach a single reasoner is used to infer a given ontology. Although it is difficult for an atomic reasoner to perform entailments on ontology’s having all the above said traits but if in case we are able to infer such ontology’s we require machines with large memory and powerful functional capacities. Here, we present some of the challenges associated while working with homogenous approach. If we talk of functional challenges, the tableau based reasoners are useful for both simple DL and Expressive DL based ontology’s. The reasoner which uses tableau algorithms is Hermit and Pellet. On the other hand if the ontology’s are associated with multiple instances then we require a reasoner which infers these ontology’s in minimum memory requirement using a rule base. The representative reasoner in such case is Bossam and RETE based rule reasoner. There can be cases when ontology is a mix of all the traits and it would be a curse to use a single reasoner to classify it. In those cases when the ontology’s is highly complex then we need to follow a hybrid reasoner which combines the traits of two individual reasoners into one. This means that if we have two languages R and S where R is the rule language and S is some Description Logic language and in both of these languages the variable and constant are common but relationships are disjoint which can be reasoned and queried by respective languages QR and QS then a reasoner for such hybrid languages can be constructed by merging the two atomic reasoners together in this paper we present a hybrid reasoner which integrates a Description Logic based reasoner and Rule Based reasoner, the reasoner so designed should be sound and complete in according to the semantics and can be best utilized for optimum entailments and efficient query answering. This approach can practically be applied where the entailments can be divided into processes TBOX and ABOX and according to the nature of queries can be
assigned to two separate reasoners, a Description Logic based reasoner for answering the Terminological queries and a Rule reasoner for answering ABOX queries.

3. Background

Reasoning is the process of discovering new relationships based on the data and based on some additional information described using vocabularies and rules. The Description Logic (DL) is used to describe the concepts of a given domain which is also known as terminological knowledge. The basic unit for this logic is termed as Axiom. The axioms are the building blocks of DL or are the logical statement related to a role or a concept. Basic DL reasoning problem includes Concept satisfiability, Concept Subsumption, Consistency of ABox with respect to TBox, Realization and retrieval of Individuals. The semantic of any ontology can be captured using entailments. Entailments are the set of rules used to transform new assertions from the existing ones. For performing entailments it is required that a RDF triple can be defined as subject, predicate and an object which can be referred as \(<s, p, o>\). In RDF triple certain relationship holds between the subject and the object and is denoted through predicate. The meaning of asserting an RDF graphs means inferring all the triples contained in it. Therefore assertion can be defined as a conjunction (Logical and) of all the statements corresponding to all the triples it holds. The triples in this sense can be classified as: Terminological (TBox), Assertional (ABox). The T triples describe information such as parent child relationships, equivalence of classes and property types etc. And A-triples gives information such as same As/different From, domain, Functional Property etc. The process of reasoning in DL systems is based on two distinct pillars that cooperate. By definition the Knowledge Base of any Semantic Web Application is based upon two KB’s i.e. the ontology (TBox) and the Rules (Abox). In Description logic there is a thin layer which differentiates between the terminological box and assertional Box. If TBox describes the concepts, the ABox gives a description of individuals which belongs to these concept hierarchies known as instances, for example the classification problem is related to TBox but instance checking is a ABox related problem. The primary reason for separating two statements in our approach is to reduce the complexity of a reasoner which might process these two statements separately.

5. In their research has stated that the Description Logic reasoner can handle TBOX reasoning more efficiently than any other reasoner and ABOX reasoner is suitable for answering complex rule based queries. In his paper has presented a prototypic system that integrates normal clauses having a well defined semantics and ontology’s specified in Description Logics. They have merged the XSB Prolog for rule based reasoning and existing OWL reasoner for ontology reasoning. This approach combines the Declarative Semantics of Logic Programming with first order Semantics of Description Logic and has proposed that various reasoners can be built based on existing rule and OWL reasoned. In his paper has presented a real approach that infers the ontologies and rules in accordance with the semantic web standards and available tools. They have combined the OWL/RuleML using Protege OWL Plug in. Had presented a framework which is inspired by the plus points of both the homogenous and hybrid approaches for integration of rules and ontologies. Their idea was to combine the TBox inference capabilities of a DL reasoner with Rule based ABox related entailment. We hereby present few existing systems which are based on the concept of integration of different Rule based Reasoner along with OWL DL based Reasoners. The AL-log system integrates the Data log with DL based ALC. The ALC DL is a simple DL based language with various class constructors like intersection Of, union Of, complement Of. All the DL queries in this system are restricted to ground and open clauses. In CARIN an integration of Data log with Description Logic ALCNR is done. This system implements class constructor along with intersection constructor for properties. This system perceives the class instance membership queries along with property instance membership queries in rule bodies. The complexity of this generality is undecidability of query answering. Based on the Suggestions given by the above authors we hereby present a DL and RL based hybrid reasoner.

4. Methodology

In this section we present the algorithm which takes
ontology and entailment rules as input and answer the queries through the use of materialized ontology in forms of inferred triples. But before describing the algorithms lets have an eye on the key terms and the tools which are required to build the proposed reasoner.

4.1 RDF Entailment Regimes

Definition 1
Entailment can be expressed as a semantic relationship among statements which holds whenever the truth of the first statement guarantees the truth of the second one. Similarly, whenever it is impossible for the first statement to be true the second statement always be false. Therefore, any interpretation which satisfies the first also satisfies the second.

RDF defines 4 entailment regimes:
- Simple entailment: These kinds of entailments match the structure of the ontology and does not interpret any RDF vocabulary,
- RDFS entailment rules: are the set of rules given by W3C for the interpretation of RDF and RDFS vocabularies
- D entailment: These are the set of rules for the entailments of data types.

4.2 Tools for Semantic Web Programming

In Table 1 we present a set of tools, knowledge of which acts as a prerequisite for complete understanding of this framework. The tools are divided into three categories: the Editors, the interface and few already existing reasoners. Editors are required for complete understanding of Ontology, The Interface is required to integrate reasoner with ontology and before development of the new reasoner, a complete understanding of the existing reasoner is must.

4.3 Algorithm

Here, we present the proposed algorithm which is based on the hybrid approach and calls two separate reasoners to entail ontologies which are a mix of traits discussed in the previous section.

Input: Ontology (Tbox, Abox), Entailment Rules.
Output: Materialized Ontology in the forms of inferred Triples used for Query Answering.

- Step 1: Input Ontology to the Reasoner.
- Step 2: Classifying Axioms into Schema Based (TBOX) and Assertional Based (ABOX).
- Step 3: Make a call to a Generic Description Logic Reasoner for entailment of Terminological based Triples.
- Step 4: Uploading the rule set for in the Agenda (The knowledge base containing the Domain specific Rules)
- Step 5: Calling a Rule based reasoner for Domain Specific (ABOX) entailments.

We conducted experiments to test the above algorithms by making a call to Pellet and Hermit reasoners (The Generic Description Logic Reasoner) and we have used Jena based Generic Rule Forward Chaining Reasoner for ABox specific Entailments. The whole of the study used OWLAPI and Jena Framework. Three different ontology’s falling under different OWL Profiles was given as Input to Reasoner and the results for the same are discussed in the section below.

Table 1. Semantic web tools

<table>
<thead>
<tr>
<th>Ontology Editors</th>
<th>Interface</th>
<th>Reasoners</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protégé – An free open Source knowledge management system used for the purpose of ontology editing, comes with DL reasoners and uses a DIG interface at backend for consistency check.</td>
<td>DIG- DL implementation Group is a standardised XML interface to Description Logic systems.</td>
<td>There are lot many reasoners which works on OWL and DL Systems to name few we have: Fact++ is a c++ free open-source reasoner. Pellet is a Java based commercial reasoner. HermiT is a reasoner which is based on the hypertableaux calculus. Jena works on RDF and OWL. The reasoning process in Jena is seen as graph to graph transformation which produces graph of virtual triples.</td>
</tr>
<tr>
<td>Swoop – a product of university of Maryland is an open source ontology editor.</td>
<td>OWL API – Is a Java based open source used for creation and modification of ontologies.</td>
<td></td>
</tr>
</tbody>
</table>
5. Discussions

For the purpose of our study we loaded different ontology’s and tested the result on the basis of parameters like size of the ontology, the time taken to classify and materialize the axiom. Basically three types of Ontologies were loaded which falls under different profiles OWL DL, OWL EL, OWL QL. For OWL DL profile we loaded UnivBench ontology and the reasoner classified and realized 45 elements in it took 9.07 seconds to separate the axioms and 0.015 seconds to perform ABox entailment before query answering. In the next phase we loaded Galen ontology and the reasoner classified 23143 elements in 11.817 seconds and after which it failed to perform ABox reasoning as this ontology comes under the category of large ontology’s which requires a different set of rules to be loaded. The experiment was also performed on a prototypic ontology designed by the authors for Mobiles. The reasoner classified 9 elements and it took 3 seconds only to separate the TBox and ABox axiom as it was having certain inconsistency involved which was given as a warning message to the user. For ontology Pizza.owl the reasoner classified 101 elements in just 6.108 seconds.

6. Results and Findings

As compared with a homogenous approach the reasoners which are based on the hybrid approach has the following benefits.

- These reasoners are able to deal with more expressive DL.
- These reasoners show the results which are more sound and complete.
- These reasoners are able to perform entailments on ontology’s which are complicated in nature and can only be entailed by powerful machines.

7. Conclusion

In this paper we have analyzed the working of hybrid reasoner on OWL2 Profile Languages. One OWL file from each OWL2 profile RL, QL, EL was given as an input to the hybrid reasoner and was analyzed on various factors. In future we can analyze the working of this reasoner on large ontology’s which has an extended dataset stored in a database.

8. References

17. Kattenstroth H, May W, Schenk F. Combining OWL with


