PERFORMANCE EVALUATION of SEMANTIC REASONERS

Prof. Neha Dalwadi[#1], Prof. Bhaumik Nagar[*2], Prof. Ashwin Makwana[$3]

[#] *Computer Engineering, Vadodara Institute of Technology*

Kotambi, Dist. Vadodara, India

[1]neha.dalwadi@gmail.com

[$] *Computer Engineering, Chandubhai S Patel Institute of Technology*
Changa, Dist. Anand, India
[3]ashwinmakwana.ce@ecchanga.ac

[*] *National Institute of Design*
Paldi, Ahmedabad, India
[2]bhaumik@bhaumiknagar.com

***Abstract*- As the performance of semantic reasoners change significantly with respect to all included characteristics, and therefore requires assessment and evaluation before selecting an appropriate reasoner for a given application. There are number of inference engines like Pellet, FaCT++, Hermit, RacerPro, KaON2, F-OWL and BaseVISor. Some of them are reviewed and tested for few prebuilt ontologies. This paper proposes performance evaluation and comparison of semantic reasoner for the ontologies of Health and Anatomy domain. Reasoners are characterized based on reasoning method, reasoning algorithm, computational complexity, classification, scalability, query and rule support.**

Keywords- OWL, Inference Engine, Semantic reasoner.

I. INTRODUCTION

Semantic web is an emerging technology of next generation web paradigm, providing machine understandable information that is based on meaning. (Tim Berners-Lee, 2001). OWL (Web Ontology Language) is knowledge representation language in semantic web, which describes basic concepts and relationships among them. To understand and use data encoded in semantic web documents, represented in OWL, requires an inference engine. OWL is further classified based on expressivity as OWL Lite and OWL DL, OWL Full that are based on Description Logic (DL). OWL Lite is less expressive than OWL DL and OWL Full is fully expressive language. Implicit knowledge can be inferred from the given descriptions of concepts and roles in OWL. There are two components of DL knowledge base: Terminology Box (TBox) and Assertion Box (ABox). Most of the scalability issues refer to size of TBox and ABox used with inference engines.

A. *Objective* - To gain familiarity with inference engine and techniques to infer new knowledge from semantic web. To identify reasoner characteristics that influences the choice of semantic reasoner for a given semantic application.

II. SEMANTIC REASONERS

A Semantic reasoner is a program that infers new set of explicitly asserted axioms or facts. It provides several reasoning tasks like classification, consistency checking, satisfiability checking of concepts (Classes) and ontology, querying etc. Different inference algorithms and inference engines are used to discover new facts and knowledge with higher accuracy, scalability, efficiency and much smaller rule list. Performance analysis is done on following semantic reasoner:

A. *Pellet*- It is an OWL-DL reasoner implemented in JAVA, reasoning on SHIN (D) and SHON (D). Performs data type reasoning, individual reasoning, absorption, nominal support, semantic branching, lazy unfolding, TBox partitioning and optimization in ABox query answering makes it more attractive for semantic web based applications. Pellet system is based on tableaux algorithm [1].

B. *Fact++*- Inference engine based on F-Logic, an approach of defining frame based systems in logic. It employs tableaux algorithm for SHOIQ DL. Performs absorption, consistency check, extracts hidden knowledge base, support complex reasoning by importing rules, synonym replacement and model merging [2].

C. *Hermit*- Hermit is an OWL-DL reasoner. Performs consistency checking, identify subsumption relationships between classes, check satisfiability and other reasoning tasks. It is based on hyper-tableau calculus which provides more efficient reasoning than tableaux algorithm.

D. *RacerPro*- It is an OWL-DL (without nominals) reasoner implemented using Lisp, reasoning on SHIQ. It performs basic reasoning tasks such as satisfiability, subsumption checking, consistency, ABox query etc. It is based on tableaux algorithm follows optimization strategies for better performance [3].

III. MOTIVATION

The semantic web data are annotated with semantic mark-ups which are included in OWL. Therefore research and implementation of OWL inference engine is important topic today. There are many semantic reasoners available for reasoning on semantic web application as described few of them above. In state-of-the-art semantic web reasoners, reasoning is performed for OWL-DL which has high worst complexity. There is an assumptions that these semantic reasoner works well in realistic semantic applications, for that there is a need to analyze the performance evaluation of semantic reasoners.

There are number of inference engines available for reasoning, but their performance depend on reasoning capability and knowledge base characteristics like expressivity and size. Due to that it is difficult to choose particular inference engine for a specific domain application and this become more complex when evaluations are based on large-scale of ontology.

IV. METHODOLOGY

For an ideal comparison of semantic reasoner, it would be desire to run all reasoners via same interface. Protégé 4.1 is an interface for Pellet, Fact++, Hermit and RacerPro reasoners have been used for experiment. Protégé is a GUI tool to create, update and test set of ontologies. To measure performance the latest available versions of the reasoners have been used: Fact++ v1.5, Hermit- v3.0, Pellet- v2.3, and RacerPro- v2.0.

A. Data Set- The data set contains most of the ontologies that are well established and widely used for testing reasoning services. For performance evaluation Health and Anatomy based ontologies taken as data set, sub-domain of Bio-Medical ontology [4].

1. Anatomy domain based ontologies -

Bila.owl – Bilateria Anatomy

AEO.owl – Anatomical Entity Ontology

DDAnatomy.owl - Dictyostelium discoideum anatomy Ontology, A structured controlled

Vocabulary of the anatomy of the slime-mould *Dictyostelium discoideum*

Cell.owl - The Cell Ontology is designed as a structured controlled vocabulary for cell types. Use by the model organism and other bioinformatics database.

DC_Cell.owl – Dendrite Cell Ontology, Representation of types of dendrite cell. Note that the domain of this ontology is wholly subsumed by the domain of the Cell ontology (CL).

2. Health domain based ontologies-
AERO.owl - The Adverse Event Reporting Ontology is an ontology aimed at supporting clinicians at the time of data entry, increasing quality and accuracy of reported adverse events.

Doid.owl – Human Disease Ontology, Creating a comprehensive hierarchical controlled vocabulary for human disease representation.

Flu.owl – Influenza Ontology

Idomal.owl – Malaria Ontology, application ontology to cover all aspects of malaria (clinical, epidemiological, biological, etc) as well as the intervention attempts to control it.

B. CPU configuration- CPU is managed to provide minimum of 70 percent of processing power to the protégé. This was done with the help of “AUTORUN” tool. This tool provide the facility to halt or stop a process, a service or anything which is running on OS and gives complete control of processing and memory power consumption. CPU– Intel core I5 processor runs at 2.40 GHz, RAM – 4 GB, SYSTEM – 32 bit OS.

C. Performance Measures- Any semantic web application needs to get the response from reasoner effectively and efficiently, for that following measures are evaluated:

- Load Time: The time to load ontology in system and check ABox consistency before performing any reasoning task.

- Classification Time: The time that is needed to classify the concepts of given ontology and generate class hierarchy to solve further reasoning tasks.

- Inferred Axioms: Number of axioms retrieved after performing reasoning. It is required to check inference capability of reasoner.
- Query Execution Time: The time starts with executing query and ends when all query results were store in local variable.

V. RESULTS

Result TABLE I shows comparison between semantic reasoners available as plug-in in protégé with the parameters of time taken for classification in milliseconds, loading time in milliseconds, inferred axioms, number of classes and logical axiom count. From TABLE I below shows the graphical results of ontology.

A. Results of Anatomy domain based ontology

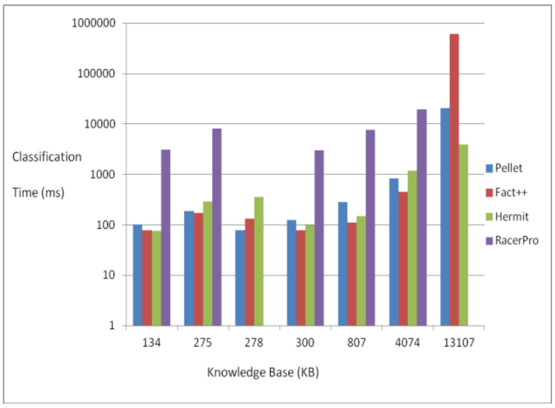

Fig. 1 Knowledge Base vs. Classification Time

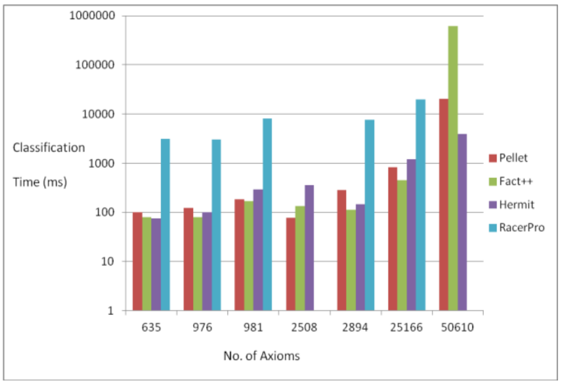

Fig. 2 Number of Axioms vs. Classification Time

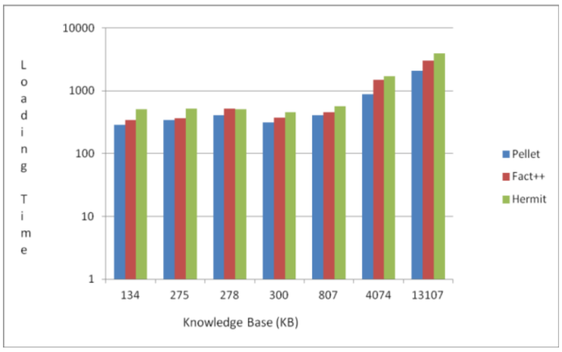

Fig. 3 Knowledge Base vs. Loading Time

From Fig. 1, 2 and 3, it can be observed that Hermit performs better for all ontology compare to other reasoner. And performance of RacerPro decreases as the knowledgebase increases. Also observed for Cell ontology RacerPro goes in infinite state due to large number of classes and axioms not able to handle it. Performance of FaCT++ is good for small size of knowledgebase but it also degrades as the size increases and gives poor performance than other reasoners. Hermit and Pellet supports large number of axioms and knowledgebase but as the number of axioms increased performance of FaCT++ and RacerPro degrades.

B. Results of Health domain based ontology

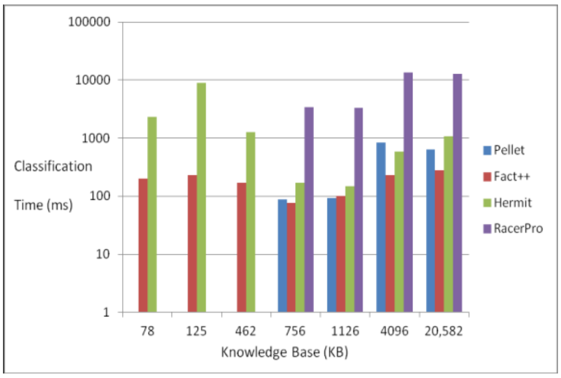

Fig. 4 Knowledge Base vs. Classification Time

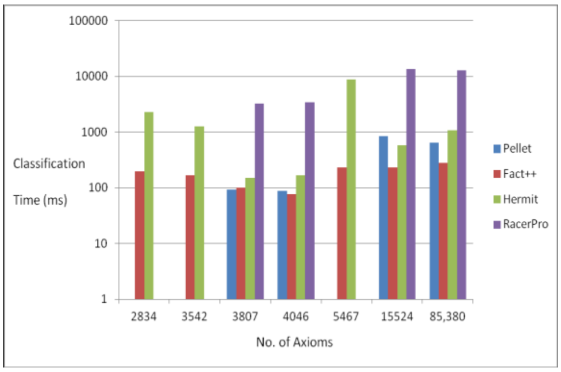

Fig. 5 Number of Axioms vs. Classification Time

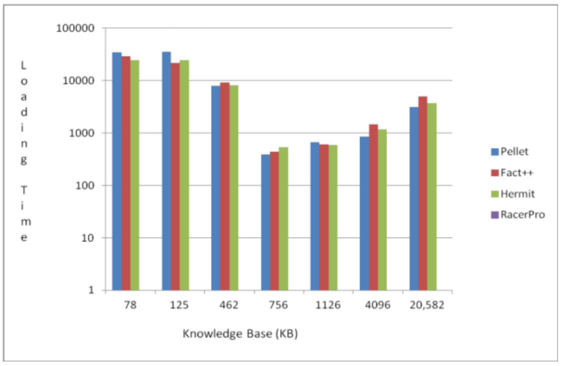

Fig. 6 Knowledge Base vs. Loading Time

From fig. 4, 5 and 6, it can be observed that Hermit performs best for all ontologies over other reasoners. First three ontologies of Health domain as shown in TABLE I contain ABox data that is number of individuals and can observe that as the size of ABox increases the performance of all reasoners degrades. Hermit and Pellet supports large number of axioms and knowledgebase but as the number of axioms increased performance of Pellet and RacerPro degrades. For Health domain based ontologies FaCT++ gives a better performance because it can handle large number of TBox and ABox data.

VI. CONCLUSION

Based on experiments on different domain based ontologies it is important to understand the characteristics in order to select an adequate reasoner for a given reasoning task. There is no clear "winner" reasoner that performs well for all types of ontologies and reasoning task, for example as shown here FaCT++ is not appropriate for Anatomy based ontologies but it gives better results for Health domain based ontologies than other reasoners. It is understood from the results that reasoners vary significantly with regard to the characteristics of reasoners. Therefore, assessment and evaluation of reasoners is needed before selecting a reasoner for a real-life application. With respect to performance of reasoner, it is desirable to note that how long the user has to wait for classification results.

The main contribution to this work is to test the performance of reasoners for biomedical ontologies that is useful for developers as well as users of semantic web applications.

VII. FUTURE WORK

From analysis of reasoners it is required to make the ABox consistent. For real-life applications size of ABox will be increased and currently there is no such reasoner that can support it. The evolutionary algorithms or inference mechanisms needs to be implemented to deal with large ABox data and response time to user's need. This work can be extended by implementing efficient databases for large, scalable ontologies that can be supported with available inference engines.

REFERENCES

[1] *Pellet: An owl dl reasoner*. B. Parsia, E. Sirin, In: Proc. International Semantic Web Conference, 2005.

[2] "A tableaux decision procedure for SHOIQ", I. Horrocks, U. Sattler In: Proceedings of Nineteenth International Joint Conference on Artificial Intelligence. (2005).

[3] *Racer: A Core Inference Engine for the Semantic Web*, Volker Haarslev and Ralf M¨oller. (2003).

[4] http://www.obofoundry.org

[5] Evaluation of Ontologies and DL Reasoners, Muhammad Fahad, Abdul Qadir and Syed Adnan Hussain Shah.

[6] "Structured Objects in OWL representation and reasoning". Boris Motik, Burnard Grau and Ulrike Sattler, 2008

[7] "Towards an ontology based web data exraction" Hicham Snoussi, Laurent Magnin and Jian-Yun Nie

[8] *Reducing OWL entailment to description logic satisfiability*, Ian Horrocks and Peter Patel-Schneider Journal of Web Semantics, 1(4):345-357, 2004.

TABLE I
COMPARATIVE ANALYSIS OF INFERENCE ENGINE BASED ON PERFORMANCE

Inference Engines / Ontology	DL Expressivity	Size	No. Of Class/Axioms (class+property+annotation) /Individuals	Pellet	Fact++	Hermit	Racer Pro
Anatomy Domain based Ontology							
Bila.owl (Inferred Axioms)	ALEHI+	134	114/(132+7+496)/0	100 ms (3) 287	79 ms (5) 340	76 ms (5) 503	3118 ms (3)
DC_Cell.owl	ALC	275	174/(313+0+668)/0	185 ms (55) 343	171 ms (55) 365	288 ms (55) 517	8150 ms (55)
AEO.owl	S	278	244/(355+2+2151)/0	77 ms (10) 409	133 ms (10) 518	353 ms (10) 511	OutOfMemoryError: Java Heap Space
DDAnatomy.owl	ALE+	300	138/(378+1+597)/0	123 ms (2) 315	79 ms (2) 369	99 ms (2) 456	3018 ms (2)
AAO.owl	ALE+	807	700/(696+2+2196)/0	282 ms (2) 408	112 ms (2) 456	147 ms (2) 563	7602 ms (2)
ATO.owl	ALE	4074	6135/(12163+0+13003)/0	831 ms (1) 882	451 ms (1) 1499	1185 ms (1) 1682	19571 ms (1)
Cell.owl	SR	13107	4090/(11184+28+39398)/0	20599 ms (174) 2055	606219 ms (180) 3035	3895 ms (180) 3882	Undefined Time
Health Domain based Ontologies							
AERO.owl	SROIQ (D)	78	309/(454+195+2185)/25	Undefined Time 35029	199 ms (273) 29049	2311 ms (273) 24514	OutOfMemoryError: Java Heap Space
Flu.owl	SROIN (D)	125	734/(1352+133+3982)/33	Undefined Time 35991	233 ms (231) 22050	8858 ms (231) 24422	OutOfMemoryError: Java Heap Space

IDO.owl	SROIF	462	509/(1019+74 +2449)/17	Undefined Time 7918	169 ms (159) 9224	1265 ms (165) 8152	OutOfMemoryError: Java Heap Space
Symp.owl	AL	756	936/(841+0+3205)/0	88 ms (1) 386	77 ms (1) 442	169 ms (1) 540	3401 ms (1)
Mpath.owl	ALE+	1126	754/(807+1+2999)/0	92 ms (2) 664	100 ms (2) 601	149 ms (2) 595	3268 ms (2)
Idomal.owl	ALERI+	4096	2417/(3148+11 +12365)/0	838 ms (11) 866	230 ms (11) 1485	586 ms (11) 1167	13523 ms (11)
Doid.owl	AL	20,582	8610/(6753+0 +78627)/0	641 ms (15) 3128s	279 ms (15) 5057	1072 ms (15) 3727	12667 ms (15)