Julius Rios

COP 5859

Final Project Paper

The state of the art of OWL

**Abstract.** The state of the art of OWL profiles are constantly evolving. Researchers are developing new ways to enhance these profiles, to increase their usability and practicality. Three such groups will be discussed. The first group of researchers developed a new way to increase OWL 2 EL capabilities in processing larger ontologies The second group developed an application for fast atomic decomposition and module extraction of OWL-EL ontologies. The third group developed a web application TRILL on SWISH that allows the user to write probabilistic Description Logic (DL) theories, and compute the probability of queries with just a web browser.

**Introduction**

The purpose of this research paper is to discuss the State of the Art in OWL Profiles. In this paper I will discuss three different groups of researchers and the applications they have developed. Two of the applications were developed for only one of the OWL 2 profiles, OWL 2 EL. I could not find any research in regards to the other two profiles, OWL 2 RL and OWL 2QL. However, I did find an application for the expressive sublanguage OWL 2DL.

**Applications**

The first application comes from a group of four researchers named Raghava Mutharaju, Pascal Hitzler, Prabhaker Mateti, and Freddy Lecue. Their paper is titled “Distributed and Scalable OWL EL Reasoning”. The paper was referenced from the ESWC website. The system they developed was for the OWL 2 EL reasoner. According to the W3C website, “OWL 2 EL is particularly useful in applications employing ontologies that contain very large numbers of properties and/or classes. This profile captures the expressive power used by many such ontologies and is a subset of OWL 2 for which the basic reasoning problems can be performed in time that is polynomial with respect to the size of the ontology [[*EL++*](http://www.w3.org/TR/owl2-profiles/#ref-ELpp)] (see [Section 5](http://www.w3.org/TR/owl2-profiles/#Computational_Properties) for more information on computational complexity). Dedicated reasoning algorithms for this profile are available and have been demonstrated to be implementable in a highly scalable way. The EL acronym reflects the profile's basis in the EL family of description logics [[*EL++*](http://www.w3.org/TR/owl2-profiles/#ref-ELpp)], logics that provide only existential quantification.” An example of an OWL 2 EL application is it’s use for the SNOMED CT, which is a very large biomedical ontology. According to this group of researchers, OWL 2 EL provides sufficient expressivity to model large biomedical ontologies as well as streaming data such as traffic, while at the same time allows for efficient reasoning services. Existing reasoners for OWL 2 EL, however, use only a single machine and are thus constrained by memory and computational power. At the same time, the automated generation of ontological information from streaming data and text can lead to very large ontologies which can exceed the capacities of these reasoners. Therefore they describe a distributed reasoning system that scales well using a cluster of commodity machines. They also apply the system to a use case on city traffic data and show that it can handle volumes which cannot be handled by current single machine reasoners.

The group developed a reasoner called DistEL, an open source distributed reasoner. DistEL is GPL open-sourced at https://github.com/raghavam/DistEL. Its usage and build are fully documented and it works on publicly available ontologies. The distributed algorithms along with the data distribution and load balancing scheme were described. As far as the group is aware of, this is the first such work for the EL++ description logic. The group demonstrate that DistEL scales well and also achieves reasonable speedup through parallelization. It can handle ontologies much larger than what current other reasoners are capable of. They presented a traffic data application where ontologies are generated from streaming data. They show that existing reasoners were not able to classify traffic data and other large ontologies. Their system on the other hand handles these large ontologies and shows good speedup with increase in the number of machines in the cluster. Ontologies continue to grow and to hope to keep their representations in the main memory of single machines, no matter how powerful and expensive, is hardly realistic. Large farms of commodity inexpensive machines will push the field of ontology reasoning. The group plans to further explore approaches to efficiently manage communication overhead, including other ontology partitioning strategies as well as alternate classification approaches and rule sets such as the one from ELK. They also plan to do performance modeling and fine-grained analysis on larger datasets, with higher number of nodes in the cluster. Alternatives to the usage of Redis including developing custom storage and data structure solutions can also be looked into.

The second application was developed by two researchers Francisco Martín-Recuerda and Dirk Walther. Their paper is titled “HyS: Fast Atomic Decomposition and Module Extraction of OWL-EL ontologies”. The paper was referenced from the ESWC website. In their paper, they present HyS, an application for fast atomic decomposition and module extraction of OWL-EL ontologies. They plan to extend the implementation to support both, OWL 2 ontologies and locality, in the future. A module is a subset of an ontology that includes all the axioms required to define a set of terms and the relationships between them. Efficient approximations of minimal modules have been suggested based on the notion of locality. Module extraction facilitates the reuse of existing ontologies. The number of all possible modules of an ontology can be exponential to the number of terms or axioms of the ontology. Atomic decomposition consists of a polynomial size representation of all possible modules of an ontology. This representation can facilitate the creation, reuse and maintenance of OWL ontologies. Moreover, atomic decomposition can contribute to improve the performance of existing algorithms for locality-based module extraction.

The tool HyS, is a Java implementation for computing the atomic decomposition and locality-based modules of OWL-EL ontologies. HyS computes a hypergraph representation of the modular structure of an ontology. This hypergraph representation contains the atomic decomposition of the input ontology, and it allows to extract modules for a given signature. The two reaserhers provide an experimental evaluation of HyS with a selection of large and prominent biomedical ontologies, most of which are available in the NCBO Bio portal. They demonstrate that HyS outperforms FaCT++ and the OWLAPI-AD in computing the atomic decomposition of all biomedical ontologies tested. In some cases a staggering speedup of over 1 000 times could be realized. Moreover, HyS significantly outperforms FaCT++ and the OWLAPI in extracting syntactic locality modules. The group plans to extend HyS to support both locality and full SROIQ-ontologies. They are interested in investigating the possibility to compute strongly connected components in hypergraphs in less than quadratic time. Such a result would improve the performance of computing mutual reachability in the axiom dependency hypergraph for ontologies whose locality-based dependencies can only be represented by hyper edges with more than one node in the tail.

The third application comes from a group of five researchers named Elena Bellodi, Evelina Lamma, Fabrizio Riguzzi, Riccardo Zese and Giuseppe Cota. Their paper is titled “A Web Interface to a Probabilistic OWL Reasoner”. The paper was referenced from the ESWC website. The system they developed was for the expressive sublanguage OWL 2 DL. According to the W3C website,” *OWL DL* supports those users who want the maximum expressiveness without losing computational completeness (all entailments are guaranteed to be computed) and decidability (all computations will finish in finite time) of reasoning systems. OWL DL includes all OWL language constructs with restrictions such as type separation (a class cannot also be an individual or property, a property cannot also be an individual or class). OWL DL is so named due to its correspondence with *description logics*, a field of research that has studied a particular decidable fragment of first order logic. OWL DL was designed to support the existing Description Logic business segment and has desirable computational properties for reasoning systems”.

The group present the web application TRILL on SWISH that allows the user to write probabilistic Description Logic (DL) theories, and compute the probability of queries with just a web browser. Various probabilistic extensions of DLs have been proposed in the recent past, since uncertainty is a fundamental component of the Semantic Web. They consider probabilistic DL theories following the DISPONTE semantics. Axioms of a DISPONTE Knowledge Base (KB) can be annotated with a probability and the probability of queries can be computed with inference algorithms. TRILL is a probabilistic reasoner for DISPONTE KBs that is implemented in Prolog and exploits its backtracking facilities for handling the non-determinism of the tableaux algorithm. TRILL on SWISH exploits SWISH, a recently proposed web framework for logic programming, based on various features and packages of SWI-Prolo. The program and the query are sent to the server, which returns the answer(s) to the user. TRILL on SWISH also allows users to cooperate in writing a probabilistic DL theory. By providing a web-based system, the users can experiment with Probabilistic DLs without the need to install any system. In this way they aim to reach out to a wider audience and popularize the Probabilistic Semantic Web. In the future, they are planning to extend it with the possibility of expressing KBs with other OWL syntaxes (such as Functional, Turtle, etc.), and with learning algorithms.

**Conclusion**

In conclusion, the topic for this research paper is the State of the Art in OWL Profiles In this paper I discussed three different groups of researchers and the applications they have developed. Two of the applications were developed for only one of the OWL 2 profiles, OWL 2 EL. The first group of researchers developed a new way to increase OWL 2 EL capabilities in processing larger ontologies The second group developed an application for fast atomic decomposition and module extraction of OWL-EL ontologies. The third group developed an application for the expressive sublanguage OWL 2DL, which was a web application called TRILL on SWISH that allows the user to write probabilistic Description Logic (DL) theories, and compute the probability of queries with just a web browser.

References

http://www.w3.org/TR/owl2-profiles/

http://2015.eswc-conferences.org/

http://www.w3.org/Consortium/

“Distributed and Scalable OWL EL Reasoning”, http://daselab.cs.wright.edu/pub/2015-ESWC-distel.pdf

“HyS: Fast Atomic Decomposition and Module Extraction of OWL-EL ontologies”,http://www.semantic-web-journal.net/system/files/swj1009.pdf

“A Web Interface to a Probabilistic OWL Reasoner”, http://www.semantic-web-journal.net/system/files/swj985.pdf