Reasoning Methods in Semantic Web

Anriette Michel Fouad Bishara, Lia Kurtanidze, Mikheil Rukhaia

Abstract— **Semantic Web is a collection of different technologies, where most of them is already standardized. The main purpose of these technologies is to describe semantic content of the web, i.e. their meaning and sense, in the format understood by computers. As a consequence, computer programs will be able to use more (human) knowledge to do assigned tasks.**

In this paper we overview the ontology and logic layers of the semantic web stack. Although ontology languages are standardized by W3C, there are still many problems remaining, which are related to reasoning over the ontologies.

On the logic layer of the semantic web stack are considered unranked languages, where function and predicate symbols do not have a fixed arity. Such languages can naturally model XML documents and operations on them. In this paper we present survey of reasoning methods over such unranked languages.

Index Terms— **Description Logic, query answering, semantic web, web data extraction**

I. INTRODUCTION

From its beginning, development of semantic web technologies was closely related to the Internet. The name itself, Semantic Web, was introduced by Tim Berners-Lee, who was a founder of this scientific direction [1]. The main idea of the semantic web is to have knowledge available for wide auditory (the purpose of WWW itself) and to utilize this knowledge by developing systems for searching, browsing and evaluation. Thus, main technologies in semantic web are knowledge representation formats and different forms of knowledge.

The semantic web, as scientific direction, was almost pronounced dead, when major IT companies started to invest money in it. As a consequence, the field is alive, growing rapidly and has big financial support from research organizations and industry.

Semantic Web is a collection of different technologies, where most of them is already standardized [2]. The main purpose of these technologies is to describe semantic content of the web, i.e. their meaning and sense, in the format understood by computers. As a consequence, computer programs will be able to use more (human) knowledge to do assigned tasks. Nowadays, the main research is concentrated on the ontology, logic and proof layers [3], [4].

Ontologies are called machine-processable formalisms for knowledge description. Their purpose is to describe objects according to domain of interests. For example, modern libraries (especially online ones) use model that is based on books content and search is carried out according to author, title, publisher and the like.

Anriette Michel Fouad Bishara, Faculty of Computer Technologies and Engineering, International Black Sea University, Tbilisi, Georgia

Lia Kurtanidze, Department of Informatics, Faculty of Business, Computing and Social Sciences, St. Andrew the First-Called Georgian University of the Patriarchate of Georgia, Tbilisi, Georgia

Mikheil Rukhaia, Institute of Applied Mathematics, Tbilisi State University, Tbilisi, Georgia

Although ontology languages are standardized by W3C (e.g. OWL [5]), there are still many problems remaining. One of the most important problem is related to reasoning over such ontologies. It is important to link ontology layer with logic layer. In this paper we summarize existing approaches and propose some new ideas.

Another very important layer of semantic web is the logic layer, which is not yet standardized. This layer is related to reasoning over the knowledge, so called knowledge derivation. Humans derive new knowledge from existing facts in their mind, but the first formalization of this process was given by Aristotle with simple syllogism: if A implies B and B implies C, then A implies C. More serious work in this direction was resumed in nineteenth century, when George Boole created propositional logic (often called Boolean logic), and Gottlob Frege introduced notions of quantifiers, which is considered as basis for first-order logic. Later, Whitehead and Russel conjectured that any mathematical truth can be obtained from several axioms by logical deduction. They gave formalization of set theory and arithmetic in a strong deductive way [6]. Based on these results, Hilbert created his well-known program and introduced Hilbert's system [7], an alternative to logical deduction. But in 1930s Godel proved, by his famous incompleteness theorems [8], that formalization of mathematics in this way is not possible. Analogous result was obtained by Turing, who proved that there is no algorithm, which tells us whether an arbitrary computer program will terminate (so called Halting Problem, [9]).

Nevertheless, from the 1960s a new direction, called Artificial Intelligence was born. Its main purpose was to automatize reasoning and derivation of new knowledge from facts. The main problem in this direction was that reasoning worked on small problems under limited knowledge and was not able to handle big problems. Later it was proved, that this limitation is not due to hardware, but algorithms chosen, and even unavoidable in some cases. Because of this, a notion of scalability was proposed, which is the main requirement in semantic web technologies. Scalability means to create algorithms, oriented on practical problems, which will solve bigger problems on a better computer. Thanks to such algorithms, nowadays we have so called expert systems, which are used in almost all areas, namely in medicine, biology and the like. The most important fact is that in many cases such systems perform better than humans.

Nowadays, on the logic layer of the semantic web are considered formal languages, which are based on unranked alphabets. This means that functional and/or predicate symbols do not have a fixed arity. Such language can naturally model XML documents and operation over them [10] and many more. These kind of languages are called unranked languages. One of the most interesting formalisms, based on an unranked alphabet, is Common Logic [11]. It is a logic, which is used to exchange information between different systems and networks. It was given ISO/IEC standard in

2007. In this paper we will discuss reasoning methods over such languages and show its connection with ontology layer.

II. Preliminaries

Semantic web is a collection of different technologies, where at the bottom of the stack we have XML, which is a language allowing to write structured documents by user defined vocabulary. XML is famous for its ability to transform structured information from site to site over the web. There are several XML-based knowledge representation languages for the Semantic Web, such as RDF, RDF Schema, and OWL.

A. *Resource Description Framework*

Resource Description Framework is a formal language for describing structured information. It is a data model used to represent information about resources. Its main intention is not to display documents, but to allow their further processing and re-combination of the information contained in them. RDF document is a directed graph with labeled nodes and edges.

The fundamental concepts of RDF are resources, properties and statements [12]. In the Semantic Web everything is a resource. Resources are objects we want to talk about, for example people, books, cars, search queries and so on. They are represented by Universal Resource Identifier (URI). Properties are a special kind of resources. They describe relationships between resources, for example "is parent of", "has child" and so on. Statements are *(subject; Predicate; object)* triples, expressing that some resource (*subject*) is related to another resource or a value (*object*) through the property (*Predicate*). We can consider this triple as a logical formula *Predicate(subject, object)*. RDF offers only binary predicates.

The RDF Vocabulary Description Language (RDF Schema) is an extension of RDF [13]. It introduces the notions of class and property and provides mechanisms for specifying class hierarchies, property hierarchies and for defining domains and ranges of properties.

RDF Schema (RDFS) is a universal language that lets users describe resources using their own vocabularies. RDF does not make any particular assumption about application domains and does not define semantics of domain. It is a semantic extension of basic RDF essentially by giving special meaning to the properties *rdfs:subClassOf* and *rdfs:subPropertyOf*, as well as to several types (like *rdfs:Class, rdfs:Resource, rdfs:Literal, rdfs:Datatype* etc.), in order to express simple taxonomies among properties and resources.

RDF Schema provides modeling primitives for expressing the information about class hierarchy, property hierarchy, defining domains and ranges of properties and so on. It uses RDF language itself. The modeling primitives of RDF Schema are defined using resources and properties. So, RDFS document is just RDF document written in XML syntax.

The RDF Schema language class and property system is similar to the general principles of object-oriented programming paradigms, but there are major differences as well. One point is that instead of defining a class in terms of the properties its instances may have, the RDF vocabulary description language describes properties in terms of the classes of resource to which they apply. For example, we could define the *ex:author* property to have a domain of *ex:Document* and a range of *ex:Person*, whereas a classical object oriented system would define a class *Book* with an attribute called *author* of type *Person*. Another major difference is that classes can have multiple super-classes in RDF Schema.

B. *Web Ontology Language*

There was a need to develop more expressive ontology language than RDF Schema. For example, RDF Schema cannot express cardinality constraints and properties like transitivity, symmetry, etc.

Ideally, new web ontology language would extend RDF Schema, but naive extension of RDF Schema with logic leads to uncontrollable computational properties. Thus, OWL [14] is based on a logic family called Description Logics, which are usually decidable fragments of first-order predicate logic (there are some undecidable description logics, but they are rarely used in practice).

The main notions in DLs are *concepts* and *roles*. *Concept* names are equivalent to unary predicates and concepts itself to formulae with one free variable. *Role* names are equivalent to binary predicates and roles itself to formulae with two free variables. *Individuals* are equivalent to constants. Operators $\forall$ and $\exists$ are restricted so that the language is decidable.

Let A be an atomic class, and R be an (abstract) role; class expressions C, D are constructed using the following rule:

$$C, D ::= A \mid \top \mid \bot \mid \neg C \mid C \sqcup D \mid C \sqcap D \mid \forall R.C \mid \exists R.C$$

In general, a DL knowledge base is a pair (TB, AB), where TB is a set of *terminological boxes* (TBox) and AB is a set of *assertional boxes* (ABox).

TBox contains class and role definitions and assertions about them. In particular, it formalizes *subset* and *equivalence* relations. *Subset* is typically written as $C \sqsubseteq D$ which means that D subsumes C, i.e. the class (or role) D is more general than the class (role) C (e.g., $Man \sqsubseteq Person$). *Equivalence* is denoted as $C \equiv D$ and is often used to define left-hand side classes. For example, $Woman \equiv Person \sqcap Female$ defines a woman as a female person.

ABox contains the facts about the individuals belonging to some classes or connected to other individuals via roles. For example, *Person(john)* states that the individual *john* is a *Person*; and *MarriedTo(john, marry)* stated that john is married to marry. Formally, it is not allowed to have ABox of the form $(Person \sqcap Female)(marry)$, but we can define new class Woman (using a TBox as it is shown above) and write *Woman(marry)*. This process is called ABox *reduction* in the literature [15].

Applying some kind of syntactic restriction on TBoxes, different sublanguages of OWL can be defined. These sublanguages have different computational properties and expressive power. In practice it is common to use the sublanguages that are less expressive, but reasoning over the ontologies is at most polynomial.

III. Reasoning Methods

Reasoning on the Semantic Web is a process of deriving new knowledge from a particular ontology (knowledge base). The main reasoning tasks are:

Subsumption: whether a class C is a subclass of D (i.e., whether the fact $C \sqsubseteq D$ is derivable).

Class equivalence: whether a class C is equivalent to a class D (i.e., whether the fact $C \equiv D$ is derivable).

Instance checking: whether an individual i belongs to a class C (i.e., derive fact C(i)).

Class disjointness: whether two classes C and D are disjoint (i.e., whether the fact C ⊓ D ⊑ ⊥ is derivable).

Class consistency: whether a given class C is consistent (i.e., whether the fact C ⊑ ⊥ is not derivable).

All these tasks are reducible to the task of *global consistency*: whether the given knowledge base is satisfiable. The reduction idea is simple – add negation of the task to the knowledge base and check its global consistency. For example, if a knowledge K is given and we are interested if it implies that a class C contains an individual i, the knowledge base K ∪ {¬C(i)} must be unsatisfiable.

Description Logics are fragments of first-order logic, thus in principle it is possible to transform every OWL statement into a first-order formula and use well known reasoning methods of first-order logic for satisfiability checking of the ontology. Such naïve approach would be highly inefficient. Thus, deduction algorithms from first-order logic must be adjusted to the description logic settings. The most successful approach for description logics to date is based on tableaux algorithms. All major Description Logic reasoners (e.g. Racer [16], FaCT++ [17], Pellet [18], etc.) use tableaux as their main reasoning method (see e.g. [19], [20]).

Tableaux calculus [21] is based on the principle of refutation. When a formula is given, it is negated and according to some rules decomposed to subformulas. This decomposition produces a tree of formulas. If every branch of the tree is closed, then the given formula is valid. A branch of a tableaux is closed if it contains both, formula and its negation; otherwise it is open. Tableau has advantage over other proof systems in that it can also build a model for satisfiable formula, or find a counter-example for non-valid formula. The model is extracted from the open branches of a tableaux.

There are many refinements and modification of the tableaux calculus in the literature (see e.g. [22], [23]). This includes tableaux for intuitionistic, temporal, modal, substructural, nonmonotonic, many-valued logics and the like.

Another tableaux method in the context of Semantic Web was developed in [24]. The classical first-order Tableaux calculus was extended with formulas over unranked terms. Unranked unification procedure was integrated into the calculus as a mechanism that decides whether a branch can be closed. It selects terms for replacement in quantification rules. Unranked unification was introduced in [25] and proved not to be finitary in general, that can cause non-termination of the given algorithm. The classical example of non-finitary unification is the pair *f(x,a)* and *f(a,x)*, where *x* is a sequence variable; the unifiers are [*x* => ()], [*x* => *a*], [*x* => (*a*,*a*)], etc. Nevertheless, the termination can be achieved in practice by restricting unification to matching, a special case when one of the unifiable terms is ground, i.e. does not contain sequence (unranked) variables. Such restriction makes sense, because although the query might contain sequence variables, the knowledge base contains only ground terms. Matching proved to be complete and finitary in [26].

Another important technique in refutational reasoning is Skolemization, which eliminates existential quantifiers. It is sometimes called an extension method, because it introduces new symbols in the signature of a formula. Very important feature of skolemization is that it loses logical equivalence, but preserves sat- or validity-equivalence.

Skolemization procedure is well studed for classical first-order logic [27], Constrained Logic [28], Intuitionistic Logic [29], Fuzzy Logics [30], Lukasiewicz Logic [31], Probabilistic Logic [32] and the like.

A skolemization procedure for unranked logics was presented in [33]. It can be used as an important part of resoning method together with unranked tableaux calculus. There are various ways to define skolemization:

Prenex: the traditional way to get skolem normal form of a formula. First, the formula is transformed to prenex normal form and then existential quantifiers are removed by replacing the corresponding bound variables by new function symbols, where sort of the function symbols are determined according to the number of universal quantifiers, preceding the existential one and according to the sort of variables that these quantifiers are binding.

Structural: this method does not need transformation to prenex normal form. It is a bit more general, because it can eliminate strong quantifiers from a formula. The rule is similar -- strong quantifier (Qx) depends on the weak quantifiers having (Qx) in their scope. It is possible to remove weak quantifiers in the same way, but it is called Herbandization in the literature [27].

Antiprenex: this method is similar to structural skolemization, but contrary to prenex normal form, quantifiers are shifted deep inside the formula using quantifier shifting rules, to minimize the range of quantifiers. This leads to smaller skolem terms in general.

It is easy to see, that different skolemization methods produce formulae of the similar length (the number of symbols) and logical complexity (the number of logical connectives). In [27] it was shown, that in terms of proof complexity, the particular form of skolemization actually matters, since it might destroy some information encoded inside a formula. Thus, skolemization should be considered as an integral part of the inference process and not as a preprocessing step of minor importance.

Acknowledgment

This work was supported by Shota Rustaveli National Science Foundation grant number PhD_F_17_218.

References

[1] Tim Berners-Lee, James Hendler, Ora Lassila. The Semantic Web. Scientific American Magazine, pp.34-43, 2001.

[2] Tim Berners-Lee, Wendy Hall, Nigel Shadbolt. The Semantic Web Revisited. IEEE Intelligent Systems, pp.96-101, 2006.

[3] Naeem Khalid Janjua, Farookh Khadeer Hussain. Development of a Logic Layer in the Semantic Web: Research Issues. In Proceedings of the Sixth International Conference on Semantics, Knowledge and Grids (SKG10). IEEE Computer Society, 367-370, 2010. DOI=http://dx.doi.org/10.1109/SKG.2010.61

[4] Alexander Maedche. Ontology learning for the semantic web. Vol. 665. Springer Science & Business Media, 2012.

[5] Grigoris Antoniou, Frank Van Harmelen. Web ontology language: Owl. In Handbook on ontologies, pp. 67-92. Springer Berlin Heidelberg, 2004.

[6] Alfred Whitehead, Bertrand Russell. Principia Mathematica. Cambridge University Press, 1910-1913.

[7] David Hilbert, Wilhelm Ackermann. Grundzüge der theoretischen Logik. J. Springer, 1928.

[8] Kurt Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme. Monatshefte für Mathematik und Physik, 38: 173–198, 1931.

[9] Alan Turing. On computable numbers, with an application to the Entscheidungsproblem. Proceedings of the London Mathematical Society, 42(2): 230–265, 1937.

[10] Jorge Coelho, Besik Dundua, Mário Florido, and Temur Kutsia. A rule-based approach to XML processing and web reasoning. In Proceedings of the Fourth international conference on Web reasoning and rule systems (RR'10), pp164-172. Springer-Verlag, Berlin, Heidelberg, 2010.

[11] Information technology – Common Logic (CL): a framework for a family of logic based languages, 2007. International Standard ISO/IEC 24707, 1st edn.

[12] Grigoris Antoniou, Frank Harmelen. A Semantic Web Primer. The MIT Press, Cambridge, Massachusetts Institute of Technology, London, England, 2004.

[13] W3C 2014. RDF Schema 1.1. W3C Recommendation 25 February 2014. Available at https://www.w3.org/TR/rdf-schema/

[14] W3C 2012. OWL 2 Web Ontology Language Document Overview (Second Edition). W3C Recommendation 11 December 2012. Available at https://www.w3.org/TR/owl2-overview/

[15] Pascal Hitzler, Markus Krotzsch, and Sebastian Rudolph. Foundations of Semantic Web Technologies. CRC press, 2009.

[16] Volker Haarslev and Ralf Moller. Racer system description. In International Joint Conference on Automated Reasoning, pages 701–705. Springer, 2001.

[17] Dmitry Tsarkov and Ian Horrocks. Fact++ description logic reasoner: System description. In International Joint Conference on Automated Reasoning, pages 292–297. Springer, 2006

[18] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical owl-dl reasoner. Web Semantics: science, services and agents on the WWW, 5(2):51–53, 2007

[19] Ian Horrocks and Andrei Voronkov. Reasoning support for expressive ontology languages using a theorem prover. In Foundations of Information and Knowledge Systems, pages 201–218. Springer, 2006

[20] Christoph Benzmuller and Adam Pease. Progress in Automating Higher-Order Ontology Reasoning. In Proceedings of the Second International Workshop on Practical Aspects of Automated Reasoning, IJCAR, pp. 22-32, 2010.

[21] Raymond Smullyan. First-Order Logic. Springer, 1968.

[22] Alan J. Robinson and Andrei Voronkov. Handbook of automated reasoning, volume 1. Elsevier, North Holland, 2001.

[23] Marcello D'Agostino, Dov M. Gabbay, Reiner Hahnle, and Joachim Posegga. Handbook of tableau methods. Springer Science & Business Media, 2013.

[24] Besik Dundua, Lia Kurtanidze, Mikheil Rukhaia. Unranked Tableaux Calculus and its Web-related Applications. IEEE Conference on Electrical and Computer Engineering. IEEE Xplore Digital Library, pp. 1181-1184, 2017.

[25] Temur Kutsia. Solving equations with sequence variables and sequence functions. Journal of Symbolic Computation, volume 42(3):352–388, 2007.

[26] Temur Kutsia and Mircea Marin. Solving, reasoning, and programming in common logic. In A. Voronkov, V. Negru, T. Ida, T. Jebelean, D. Petcu, S. M. Watt, and D. Zaharie, editors, 14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2012, Timisoara, Romania, September 26-29, 2012, pp. 119–126. IEEE Computer Society, 2012.

[27] Matthias Baaz, Uwe Egly, Alexander Leitsch. Normal form transformations. In A. Robinson and A. Voronkov, editors, Handbook of Automated Reasoning, volume I, chapter 5, pp. 273-333. North Holland, 2001.

[28] Hans-Jürgen Burckert, Bernhard Hollunder, Armin Laux. On skolemization in constrained logics. Annals of Mathematics and Artificial Intelligence, volume 18(2), pp. 95-131. Springer, 1996.

[29] Matthias Baaz, Rosalie Iemhoff. The Skolemization of existential quantifiers in intuitionistic logic. Annals of Pure and Applied Logic, volume 142(1-3), pp. 269-295. Elsevier, 2006.

[30] Matthias Baaz, George Metcalfe. Herbrand theorems and Skolemization for prenex fuzzy logics. In Conference on Computability in Europe, pp. 22-31. Springer, 2008.

[31] Matthias Baaz, George Metcalfe. Herbrand's Theorem, Skolemization and Proof Systems for First-Order Lukasiewicz Logic. Journal of Logic and Computation, volume 20(1), pp. 35-54. Oxford University Press, 2010.

[32] Guy Van den Broeck, Wannes Meert, Adnan Darwiche. Skolemization for weighted first-order model counting. In Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR), pp. 1-10, 2014.

[33] Lia Kurtanidze and Mikheil Rukhaia. Skolemization in Unranked Logics. Submitted to Bulletin of TICMI, 2018.

Anriette Michel Fouad Bishara was born in Aswan, Egypt. She has graduated from Sohag University, Egypt, and obtained her Bachelor degree in commerce. After working as an accountant for almost two years, she decided to change her career and started to attend courses on graphics and animation designs; this led her to continue her study in computer science and obtained her Master degree of Engineering in Computer Sciences from International Black Sea University, Tbilisi, Georgia. Currently she is a PhD student at the faculty of Computer Technologies and Engineering, International Black Sea University, Tbilisi, Georgia. Anriette has participated in about 10 international conferences. Her research interests include semantic web and automated reasoning.

Lia Kurtanidze has graduated from I.Javakhishvili Tbilisi State University and obtained her Bachelor degree in Physics. After she decided to change the field and obtained Master's degree in Informational Technologies and Computer Modeling from Georgian University. Currently, Lia is a PhD student at the Department of Informatics, Faculty of Business, Computing and Social Sciences, in Georgian university. Her major is Computer Technologies and Mathematical Modeling. In parallel Lia is a Mathematics teacher at Tbilisi 107 public school. Her research interests include automated reasoning and proof theory. She has about 5 publications and participated around 10 international conferences. Lia has been involved in two nationally funded research projects, as the principal investigator in one of them.

Mikheil Rukhaia has graduated from Tbilisi State University and obtained his Bachelor degree in Mathematics. He continued Master's study with Erasmus Mundus program and from Vienna University of Technology first he got Master's degree in Computational Logic and then PhD degree in Computer Science. He has spent a year of his post-doc at INRIA saclay Ile-de-france in the Laboratory of Informatics, Ecole Polytechnique. Currently, Mikheil is a researcher at the Institute of Applied Mathematics, Tbilisi State University and an associated professor at the International Black Sea University, Tbilisi, Georgia. His research interests include semantic web, automated reasoning, computational logic and artificial intelligence in general. He has more than 20 publications and participated more than 30 international conferences.

Despite the fact, that Mikheil Rukhaia is a young researcher, he has been involved in several nationally and internationally funded research projects, often as the principal investigator. He is a reviewer of several conferences and journals in his field, is involved in the university teaching in Georgia and works with Bachelor, Master and PhD students.