PR-OWL 2 RL - A Language for Scalable Uncertainty Reasoning on the Semantic Web

Laécio L. dos Santos[1], Rommel N. Carvalho[1,2], Marcelo Ladeira[1], Li Weigang[1], and Gilson L. Mendes[2]

[1] Department of Computer Science, University of Brasília (UnB)
Brasília, DF, Brazil
{laecio,mladeira,weigang}@cic.unb.br
[2] Department of Research and Strategic Information, Brazilian Office of the Comptroller General (CGU), Brasília, DF, Brazil
{rommel.carvalho,liborio}@cgu.gov.br

Abstract. Probabilistic OWL (PR-OWL) improves the Web Ontology Language (OWL) with the ability to treat uncertainty using Multi-Entity Bayesian Networks (MEBN). PR-OWL 2 presents a better integration with OWL and its underlying logic, allowing the creation of ontologies with probabilistic and deterministic parts. However, there are scalability problems since PR-OWL 2 is built upon OWL 2 DL which is a version of OWL based on description logic SROIQ(D) and with high complexity. To address this issue, this paper proposes PR-OWL 2 RL, a scalable version of PR-OWL based on OWL 2 RL profile and triplestores (databases based on RDF triples). OWL 2 RL allows reasoning in polynomial time for the main reasoning tasks. This paper also presents First-Order expressions accepted by this new language and analyzes its expressive power. A comparison with the previous language presents which kinds of problems are more suitable for each version of PR-OWL.

1 Introduction

Web Ontology Language (OWL) is the main language in the Semantic Web for creating ontologies. It lacks the capacity for treating uncertainty, limiting its application in several kinds of domains. Various approaches have been proposed to solve this issue using different formalisms, such as Bayesian networks, fuzzy logic, and Dempster-Shaffer theory. One of these approaches, Probabilistic OWL (PR-OWL) [7] adds uncertainty treatment capacity to OWL using Multi-Entity Bayesian Networks (MEBN) [11], which is a very expressive First-Order Probabilistic Logic. PR-OWL has been implemented in UnBBayes [3], which is an open source framework for probabilistic graphical models. PR-OWL 2 [4] extends the previous language adding a tight and better integration between OWL existing concepts and properties and PR-OWL new ones. A PR-OWL 2 implementation also was developed in UnBBayes, using Protégé [4] and its HermiT [13] default

[3] http://unbbayes.sourceforge.net/
[4] http://protege.stanford.edu/

OWL DL reasoner for modeling and reasoning with the deterministic part of the ontology.

PR-OWL 2 implementation, however, has some scalability problems due to the time complexity of OWL 2 DL reasoners to solve complex expressions. This hinders the ability to work with domains that have large assertive databases. One example is the public procurement fraud detection domain developed in Brazil, for which a probabilistic ontology was created using PR-OWL 2 [5]. Although the probabilistic ontology has been successfully tested with simple cases, in a real situation, using government databases, millions of triples will be needed, making the reasoning intractable with PR-OWL 2 and its current implementation.

The solution proposed for overcoming this limitation is to use triplestores together with the OWL 2 RL profile to create a new version of PR-OWL 2, named PR-OWL 2 RL. The OWL 2 RL [17] profile allows implementations with reasoning in polynomial time for the main reasoning tasks in systems based on rules. The reasoning is mainly processed by materialization, where the rule set is evaluated when new statements are included in the base, as well as the new knowledge that is derived by them.

The proposal of this new language requires: 1) to review the PR-OWL language according the OWL 2 RL syntax restrictions; 2) a new algorithm to evaluate the MEBN first-order formulas using triplestores; and 3) to design a scalable algorithm for generating Situation Specific Bayesian Networks (SSBN). This paper discusses the first two issues.

This paper is organized as follows. Section 2 describes some relevant concepts for the understanding of this work: MEBN, OWL and PR-OWL. Section 3 presents PR-OWL 2 bottlenecks that motivated this work. Section 4 introduces the language proposed and shows how the first-order formulas can be evaluated using the SPARQL language. Finally, Section 5 presents some conclusions and possible future work.

2 Fundamentals

This section presents some concepts necessary for the understanding of this paper. Section 2.1 presents Multi-Entity Bayesian Networks, the formalism used by PR-OWL to deal with uncertainty in the OWL language. Section 2.2 presents the OWL language and its versions, including the OWL 2 RL profile, and triplestores. Section 2.3 presents PR-OWL, its extension, PR-OWL 2, and its implementation in the UnBBayes Framework.

2.1 Multi-Entity Bayesian Networks

Multi-Entity Bayesian Networks (MEBN) is a formalism for representing first-order probabilistic knowledge bases [11]. MEBN joins Bayesian networks with First-Order Logic (FOL), augmenting the expressive power of the first by allowing uncertainty representation and reasoning in situations where the quantity of random variables is unknown.

MEBN models the domain using a MEBN Theory (MTheory), which is composed of random variables that together have a unique joint probability distribution. The knowledge is divided into MEBN Fragments (MFrags). Each MFrag is composed by resident nodes, input nodes, and context nodes. Resident nodes are random variables for which the Local Probability Distribution (LPD) is defined in the MFrag where they are. Input nodes are references to resident nodes defined in a different MFrag. Context nodes contain restrictions that need to be satisfied in order to correctly instantiate the corresponding MFrag. The nodes represent entity attributes and relationships between entities. Each node is parameterized with ordinary variables (OV), placeholders filled with entity instances available in the knowledge base during the instantiation of the model.

Figure 1 shows the MFrag `Front Of Enterprise` of the Procurement Fraud ontology [5]. This probabilistic ontology was designed to identify frauds in public procurements in Brazil using the data available in the Brazilian Office of the Comptroller General (CGU). In this MFrag, the resident node `isFrontFor` (node 9) refers to the probability that a person is a front for an enterprise. It is influenced by the input nodes `hasValue`, `hasAnnualIncome`, and `hasEducationLevel` (nodes 6–8). The context nodes (nodes 1–5) show which restrictions need to be satisfied in order to instantiate this MFrag. Nodes 4 and 5, for example, say that the procurement has to be finished and the person of interest has to be responsible for the enterprise that won the procurement.

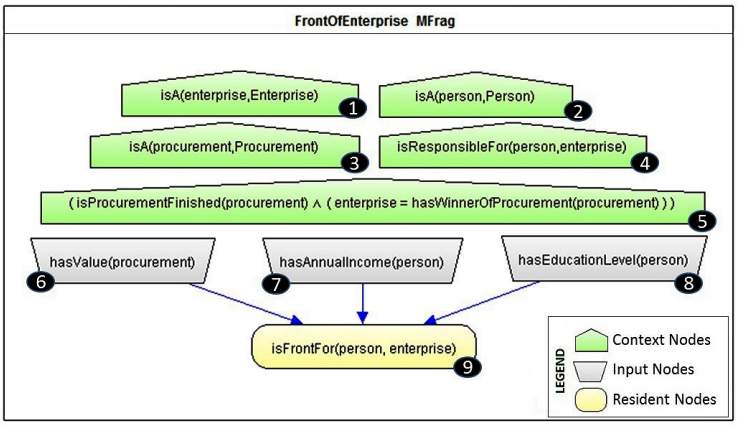

Fig. 1. MFrag `Front Of Enterprise` for the Procurement Fraud domain

An MTheory works like a template, which is instantiated giving the query nodes and the evidence to build a Situation-Specific Bayesian Network (SSBN), a Bayesian Network with all nodes computationally relevant to answer the queries. Laskey presents in [11] an algorithm for generating a SSBN that expands the network from both the queries and findings in order to build a grand BN which is pruned by removing barren, nuisance, and d-separated nodes.

2.2 OWL

OWL is the main Semantic Web language for building ontologies. OWL 2, its current version, became a W3C recommendation in 2009 [16].

The direct model-theoretic semantics of OWL 2 is called Direct Semantics, which is strongly related to the semantics of description logics [16]. The Direct Semantics assigns meaning directly to ontology structures, in a way compatible with the semantics of the SROIQ description logic [16]. Description Logics are subsets of FOL that model the domain based on its classes and properties. It represents the knowledge by first defining the relevant concepts of the domain (TBox), and then using these concepts to specify properties of objects and individuals occurring in the domain (ABox) [1]. Different description logics have been created, trying to get a favorable trade-off between expressiveness and complexity. OWL 2 DL is based on $\mathcal{SROIQ}$(D). Several reasoners based on tableau algorithms were created for OWL 2 DL. Some examples are HermiT [13], Pellet [12] and FaCT++ [15].

OWL 2 has three different profiles (syntactic subsets): OWL 2 EL, OWL 2 QL, and OWL 2 RL. All of them are more restrictive than OWL 2 DL and trades off OWL 2's expressive power for computational or implementation benefits. In these profiles, most types of reasoning can be made in polynomial time. OWL 2 EL is suitable for ontologies with a very large but simple TBox. OWL 2 QL is suitable to work with conjunctive queries, permitting the use of ontological reasoning in systems like relational databases through a query rewriting approach. OWL 2 RL, based on Datalog and in R-entailment [14], is suitable to allow an easy implementation in systems based on rules.

W3C proposes a set of rules called OWL 2 RL/RDF that implements the OWL 2 RL profile. This set of rules is based on RDF Semantic, where the knowledge is organized in graphs, composed by RDF triples. Each RDF triple is composed by a subject linked to an object by a property. The reasoning is made through rules, where given a set of specific triples and a rule, we can get another expression that follows logically from the rule. The Theorem PR1 [17] states some conditions that guarantee that the ontology O_2 entailed from O_1 under the Direct Semantics is the same entailed under the first-order axiomatization of RDF semantics using the OWL 2 RL/RDF rules:

- neither O_1 nor O_2 contains an IRI (International Resource Identifier) that is used for more than one type of entity;
- O_1 does not contain the following axioms:
  - `SubAnnotationPropertyOf`,
  - `AnnotationPropertyDomain`,
  - `AnnotationPropertyRange`; and
- each axiom in O_2 is an assertion of the form as specified below, for a_1, a_2, ..., a_n a named individual:
  - `ClassAssertion`(C a) where C is a class,
  - `ObjectPropertyAssertion`(OP a_1 a_2) where OP is an object property,
  - `DataPropertyAssertion`(DP a v) where DP is a data property, or

- `SameIndividual`($a_1 \ldots a_n$).

The OWL 2 RL profile is implemented by some triplestores. Triplestores are databases that organize the knowledge in graphs composed by RDF triples. They are becoming very useful and there are a lot of commercial (*e.g.*, GraphDB, Oracle Spatial and Graph, and AllegroGraph) and free implementations (*e.g.*, Sesame). They normally implement the RDF/RDFS entailment rules, using materialization for expanding the rules when new declarations are added to the base. SPARQL is the main language used for querying RDF databases. It is very similar to SQL, acting over generalized RDF graphs. Most triplestores accept, in addiction to RDFS, inference with some constructions of OWL. Implementations of the OWL 2 RL profile are common.

2.3 PR-OWL

Probabilistic OWL (PR-OWL) is an extension of the OWL language that permits the creation of probabilistic ontologies [7]. It works as an upper-ontology, consisting of a set of classes, subclasses, and properties that allow modeling the uncertainty part of the ontology using MEBN. Figure 2 shows the main concepts involved in a PR-OWL ontology. The probabilistic ontology is modeled using the MTheory class, composed by a set of MFrags. These MFrags must collectively form a consistent MTheory. The MFrags are built from random variables, which have a probabilistic distribution and an exhaustive set of possible states.

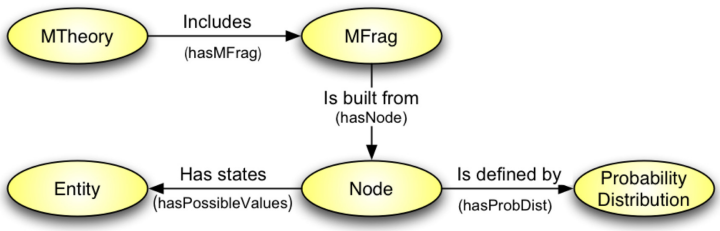

Fig. 2. PR-OWL Main Concepts

UnBBayes has an implementation of PR-OWL and MEBN [3, 6] that allows the design of an MTheory using a graphical user interface (GUI). A pseudo-code can be used for defining the LPDs. The MTheory is stored in a knowledge base, supported by the PowerLoom Knowledge Representation and Reasoning (KR&R) System[5]. The algorithm for generating SSBNs is based on the one proposed in [11].

PR-OWL 2 [4] extends PR-OWL by having a better built-in integration between OWL and MEBN. The main concept used for this is the property

[5] http://www.isi.edu/isd/LOOM/PowerLoom/

`definesUncertaintyOf` that links a random variable to an OWL property. The additional properties `isObjectIn` and `isSubObjectIn` allow the mapping of both domain and range of the OWL property to its corresponding concept in MEBN. PR-OWL 2 also has other improvements, like the support to polymorphism and the use of OWL datatypes. A plug-in for PR-OWL 2 was developed in UnBBayes, using Protégé for modeling the deterministic parts of the ontology. Protégé is a popular open source framework for editing ontologies. Moreover, HermiT is used for evaluating the context nodes and for getting information about findings.

3 Description of the Problem

PR-OWL 2 and its implementation in UnBBayes have some scalability and expressibility problems. OWL 2 DL, which is used in PR-OWL 2 definition/implementation, is based on description logic $\mathcal{SROIQ}$(D) that has complexity N2EXPTIME-complete [10] for the main reasoning problems: ontology consistency, class expression satisfiability, class expression subsumption, and instance checking. This class of complexity comprises the problems solvable by nondeterministic algorithm in time at most double exponential in the size of the input [17]. OWL 2 DL reasoners are normally based on tableau algorithms. Donini [8] states two different sources of complexity in tableau calculi: the AND-Branching, responsible for the exponential size of a single candidate model, and the OR-Branching, responsible for the exponential number of different candidate models. This exponential complexity of OWL 2 DL reasoners makes the queries more time/space consuming, the larger/more complex the knowledge base is. Thus, making it inviable for several cases. Furthermore, most OWL reasoners are limited to the available memory of the computational resource used, since the database needs to be loaded into memory to allow inference. This clearly does not scale to real and large databases.

We also have scalability problems because of the use of Protégé's GUI and API in UnBBayes' PR-OWL 2 implementation. We made tests using LUBM ontologies to verify the performance of this implementation. LUBM (Lehigh University Benchmark) [9] is a very popular benchmark for reasoners and triplestores. Using an i5 machine with 3GB of memory dedicated to run Protégé, we could not load nor initialize the reasoner with the LUBM 100, an assertive base containing 2,779,262 instances of classes and 11,096,694 instances of properties. We used the HermiT reasoner, where the initialization consists of building the class hierarchy, classifying object and data properties, computing instances of all classes and object properties, and calculating same as individual. This initialization is necessary to solve the queries. LUBM 100 base has 1,06 GB when stored in an OWL file in XML format, making it clear that the structure used by Protégé adds a great overhead to PR-OWL 2 implementation.

This scalability problems limit the use of PR-OWL in domains with large assertive bases. In the domain of procurement fraud detection [5], for example, the assertive base can easily have millions of assertions. This makes it unsuit-

able to using an OWL reasoner for the necessary deterministic reasoning, which comprises of evaluating the FOL expressions and searching for findings.

Since PR-OWL 2 is written in OWL 2 DL, one possibility is to use an OWL 2 DL reasoner for solving the FOL formulas during MEBN reasoning. PR-OWL 2 current implementation in UnBBayes does that.

Evaluating MEBN FOL formulas using an OWL DL reasoner requires some workarounds. The Table 1 presents the formulas that are allowed in the current implementation of UnBBayes, where `ov` are ordinary variables, `CONST` are constants, and `booleanRV` are Boolean random variables. Expressions with connectives and quantifiers are not allowed in this version.

Table 1. Types of context node formulas accepted in the PR-OWL 2 implementation

Formula	Negation
$ov_1 = ov_2$	NOT ($ov_1 = ov_2$)
booleanRV(ov_1 [, ov_2 , ...])	NOT booleanRV(ov_1 [, ov_2, ...])
ov_0 = nonBooleanRV(ov_1)	NOT (ov_0 = nonBooleanRV(ov_1))
ov_0 = nonBooleanRV(ov_1 [, ov_2, ...])	
CONST = nonBooleanRV(ov_1 [, ov_2 , ...])	
nonBooleanRV(ov_1 [, ov_2 , ...]) = CONST	
nonBooleanRV(ov_1) = ov_0	NOT (nonBooleanRV (ov_1) = ov_0)
nonBooleanRV(ov_1 [, ov_2 , ...]) = ov_0	

4 PR-OWL based on OWL 2 RL profile

In order to overcome the limitations presented, we propose PR-OWL 2 RL, a more scalable version of PR-OWL based in the OWL 2 RL profile. The purpose is to use an RDF triplestore database for both the storage and reasoning with very large ontologies represented as RDF triples. This is possible because OWL 2 RL allows reasoning in polynomial time for the main reasoning tasks. SPARQL is the common query language used with RDF triples.

Since PR-OWL 2 is written in OWL 2 DL, some adjustments are necessary to adapt it for OWL 2 RL. This is due to the fact that OWL 2 RL imposes several syntactic restrictions on the OWL expressions. Running a validator developed by the Manchester University [6] we found the following unsupported features:

1. Use of non-superclass expression where superclass expression is required;
2. Use of non-subclass expression where subclass expression is required;
3. Use of non-equivalent-class expression where equivalent-class expression is required; and
4. Use of unsupported data range.

[6] http://mowl-power.cs.man.ac.uk:8080/validator/

Figure 3 shows examples for each kind of unsupported feature. The first case occurs for several reasons, such as the use of existential quantifier and disjunction on the right side of a `subClass` expression or in range/domain expressions, the use of `owl:Thing` as superclass or in range/domain expressions, and the use of the qualified restriction `exactly`. The second occurs in the class `owl:Thing`, that is setted as a subclass of the restriction `'hasUID only String'`. The property `hasUID` is used to guarantee that every instance in PR-OWL has a unique identifier (a requisite necessary to work with MEBN, where each possible state has to be unique). The third occurs in all `equivalent` expressions of PR-OWL 2, which includes conjunctions, min/max/exactly cardinality expressions, and universal/existential quantifiers. OWL 2 RL is very restrictive in relation to equivalent expressions, allowing only `Class`, `intersection`, and `hasValue` expressions. Finally, the fourth occurs in the `isRepresentedAs` range expression, where all possible formats to represent the probabilistic distributions are listed.

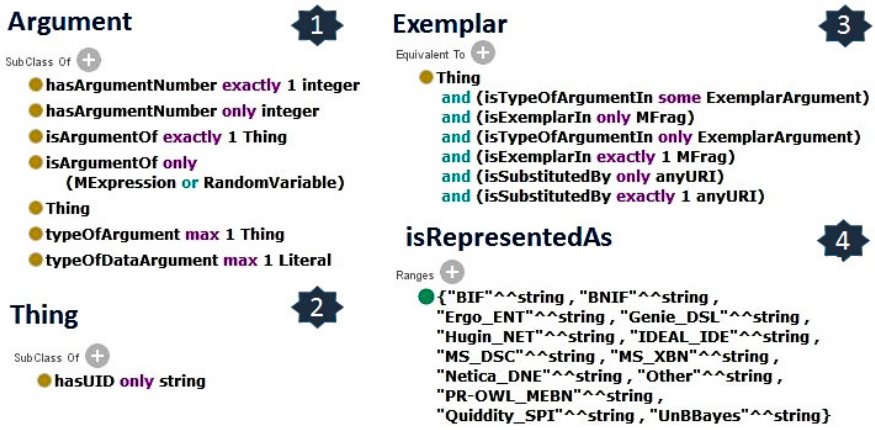

Fig. 3. Examples of disallows in PR-OWL 2

Since the syntax of OWL 2 RL/RDF is based on RDF, it permits generalized RDF graphs, not having the several restrictions of OWL 2 DL. The pure OWL 2 RL profile, however, has restrictions to allow reasoning also with the Direct Semantics. We choose to adapt PR-OWL 2 RL with the OWL 2 RL restrictions for keeping the compatibility with both semantics. In order to make this adaptation it is necessary to fix the unsupported features listed above.

To solve the unsupported features, we analyzed three alternatives. The first consists in rewriting all expressions of PR-OWL 2 from OWL 2 DL to OWL 2 RL. However, this is not possible due the less expressive power of OWL 2 RL language. Expressions `subClass`, for instance, with existential quantifier on the right side cannot be expressed in this profile. The second alternative consists in rewriting to OWL 2 RL the expressions that can be rewritten, removing the

others, and passing the responsibility of validating them forward to the PR-OWL reasoner. The problem with this alternative is that the resulting ontology is hard to understand due to the rewriting of the expressions to less intuitive axioms and because the restrictions are only partially explicit. The last alternative consists in turning PR-OWL into a lightweight ontology, containing only the class hierarchy and the object and data properties (with its domain and range restrictions). The full validation of the probabilistic model consistency is left to the PR-OWL reasoner. This was the chosen alternative because it results in a simpler ontology, sufficient for expressing the MEBN elements in OWL language.

The FOL formulas in PR-OWL 2 RL are evaluated in a different way than they are in the previous versions of PR-OWL. Using materialization, all implications of an added expression is calculated in load time. In the triplestores implementations, we do not have posterior reasoning: the queries are solved with searches on the database, using the SPARQL language, in a similar way to the SQL language in relational databases. This means that in the knowledge base we already have for example the hierarchy of an instance explicitly. For example, if an instance `a` is of the class `A`, and A is subclass of `B`, then, we will also have both `ClassAssertion(A,a)` and `ClassAssertion(B,a)` information on the base, where the second one was derived from the rule showed below (extracted from [17]).

```
IF T(?c1, rdfs:subClassOf, ?c2) AND T(?x, rdf:type, ?c1)
THEN T(?x, rdf:type, ?c2)
```

If we ask if `a` is instance of class `B`, the result will be `TRUE` because we will get the information `ClassAssertion(B,a)`. The advantage of this approach is that queries are very fast. The disadvantage is that complex reasoning cannot be handled, justifying the OWL 2 RL language restrictions.

The BNF grammar below shows the restrictions on the types of context nodes formulas accepted in PR-OWL 2 RL. The evaluation of these context nodes will be handled using the SPARQL language.

Listing 1.1. BNF Grammar for FOL Expressions in PR-OWL 2 RL

```
<atom>       ::= ov1 == ov2 |
                 booleanRV(ov1, [,ov2 ...]) |
                 nonBooleanRV(ov1, [,ov2 ...]) = ov0 |
                 ov0 = nonBooleanRV(ov1, [,ov2 ...]) |
                 nonBooleanRV(ov1, [,ov2 ...]) = CONST |
                 CONST = nonBooleanRV(ov1, [,ov2 ...])
<negation>   ::= NOT <atom>
<conjunction>::= <atom> [AND <atom>]+
<disjunction>::= <atom> [OR <atom>]+
<formula>    ::= <atom> | <negation> |
                 <conjunction> | <disjunction>
```

Table 2 shows how to evaluate these formulas using SPARQL. To solve the `EQUAL TO` operator between two ordinary variables, we can use the SPARQL `FILTER` construction, limiting the result of a query where the terms are equal.

The evaluation of `AND` and `OR` connectives is possible using period and `UNION` constructions. The negation can be implemented by the PR-OWL 2 reasoner in one of three ways depending on each case: for a not equal expression a `FILTER` can be used with the operator `!=` (different); for a boolean RV it is sufficient to ask if it is equal `FALSE`; and finally, for a not boolean RV, we can use the operator `NOT EXISTS` inside a `FILTER`.

Table 2. Implementing PR-OWL 2 RL FOL expressions using SPARQL

MEBN Expression	SPARQL Expression
AND	. (period)
OR	UNION
EQUAL TO	Use of = inside a FILTER
NOT	It depends on the case

To evaluate expressions where we do not know the value of some ordinary variable, we use the SPARQL `SELECT` construction. If we already know all values, a command `ASK` is used. This command evaluates the expression and returns `TRUE` or `FALSE`. The evaluation of the context nodes is made one by one and the implementation is responsible for keeping the consistency between the ordinary variable values of each node. The following code shows a `SELECT` to get which procurements `ENTERPRISE_1` won (node 5 in Figure 1).

```
SELECT ?procurement
WHERE  { ?procurement rdf:type Procurement .
       ENTERPRISE_1 hasWinnerOfProcurement ?procurement}
```

Finally, for the new language to be useful, it is also necessary to propose a new algorithm for generating a SSBN. The previous SSBN algorithm implemented in PR-OWL 2 starts from the queries set as well as the findings set. Since we can have a large assertive base in PR-OWL 2 RL, making the findings set very large, the previous SSBN construction algorithm might be hindered. We plan to extend the algorithm previously proposed in [6], by starting only from the queries set and removing known issues with it. For instance, the version proposed in [6] does not evaluate the parent nodes of a query, even if they are not d-separated from the evidence.

Using the new language proposed, together with a triplestore and the materialization approach, it is possible to solve the scalability problems presented. The BNF grammar proposed is sufficient to evaluate all context node formulas used in the Procurement Fraud probabilistic ontology.

The Theorem PR1 [17] limits the entailed statements to assertions. The reasoning in PR OWL 2 RL is mainly over the assertive base (ABox), but, based on the use cases already developed for PR-OWL, this does not seem to be a problem.

It is important to note that Costa, the author of PR-OWL, already visualized the possibility of creating more restrictive versions of the language to guarantee tractability [7]. The objective of PR-OWL 2 RL is not to substitute the previous version (in the way that PR-OWL 2 intends to substitute PR-OWL). Both PR-OWL 2 and PR-OWL 2 RL have characteristics that make them suitable for different kind of domains. While PR-OWL 2 is recommended for heavyweight ontologies, with complex expressions, but limited assertive bases, PR-OWL 2 RL is ideal for lightweight ontologies, with simple expressions and a very large knowledge base. This last one, for example, is the case of the ontologies in Linked Data projects.

5 Conclusion and Future Work

Using a less expressive version of OWL for reasoning in polynomial time, PR-OWL 2 RL is developed to work with ontologies containing millions of triples. When used together with RDF triplestores, it can solve the scalability problem of the previous PR-OWL versions. Using a commercial database it is possible to work with billions of triples, making it suitable even for working with Big Data. The restrictions on the expressiveness of OWL 2 RL do not allow it to express some complex statements, but it is sufficient for a lot of domains, such as the Procurement Fraud, and Linked Open Data projects. This paper presented limitations on the first-order expressions used in context node formulas, restricting the use of MEBN logic, but allowing at the same time the same constructs which are implemented and allowed in UnBBayes' PR-OWL 2 plug-in.

A future work that is already under way is the implementation of a plug-in for PR-OWL 2 RL in UnBBayes. In this plug-in we plan to use the triplestore GraphDB Lite, a free version of GraphDB [2]. GraphDB, previously OWLIM, partially implements the OWL 2 RL profile (it does not implement the rules related to datatypes), using the OWL 2 RL/RDF rules and a materialization approach. The UnBBayes' PR-OWL 2 RL plug-in will allow the user to model a probabilistic ontology using the language, to put it into the triplestore, to fill the assertive base, and to build a SSBN from the queries set. Other future work is create new study cases to validate the solution. We also plan to make an extension of the LUBM ontology by adding uncertainty concepts to it, making it possible to construct a benchmark for large probabilistic ontologies.

References

1. Franz Baader and Werner Nutt. Basic description logics. In *The Description Logic Handbook: Theory, Implementation, and Applications*, pages 43–95, 2003.
2. Barry Bishop and Spas Bojanov. Implementing OWL 2 RL and OWL 2 QL rule-sets for OWLIM. In Michel Dumontier and Mlanie Courtot, editors, *OWLED*, volume 796 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2011.
3. Rommel N. Carvalho, Marcelo Ladeira, Laécio L. Santos, Shou Matsumoto, and Paulo C. G. Costa. A GUI tool for plausible reasoning in the semantic web using MEBN. In *Innovative Applications in Data Mining*, pages 17–45. 2009.

4. Rommel N. Carvalho, Kathryn B. Laskey, and Paulo C. G. da Costa. PR-OWL 2.0 - bridging the gap to OWL semantics. In *Proceedings of the 6th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2010), collocated with the 9th International Semantic Web Conference (ISWC-2010), Shanghai, China, November 7, 2010*, pages 73–84, 2010.
5. Rommel N. Carvalho, Shou Matsumoto, Kathryn B. Laskey, Paulo C. G. da Costa, Marcelo Ladeira, and Laécio L. Santos. Probabilistic ontology and knowledge fusion for procurement fraud detection in brazil. In *Uncertainty Reasoning for the Semantic Web II, International Workshops URSW 2008-2010 Held at ISWC and UniDL 2010 Held at FLoC, Revised Selected Papers*, pages 19–40, 2013.
6. Paulo C. G. Costa, Marcelo Ladeira, Rommel N. Carvalho, Kathryn B. Laskey, Laecio L. Santos, and Shou Matsumoto. A first-order Bayesian tool for probabilistic ontologies. In David Wilson and H. Chad Lane, editors, *FLAIRS Conference*, pages 631–636. AAAI Press, 2008.
7. Paulo Cesar G. da Costa, Kathryn B. Laskey, and Kenneth J. Laskey. PR-OWL: A bayesian ontology language for the semantic web. In *International Semantic Web Conference, ISWC 2005, Galway, Ireland, Workshop 3: Uncertainty Reasoning for the Semantic Web, 7 November 2005*, pages 23–33, 2005.
8. Francesco M. Donini. Complexity of reasoning. In *The Description Logic Handbook: Theory, Implementation, and Applications*, pages 96–136, 2003.
9. Yuanbo Guo, Zhengxiang Pan, and Jeff Heflin. LUBM: A benchmark for OWL knowledge base systems. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(2):158–182, 2005.
10. Yevgeny Kazakov. RIQ and SROIQ are harder than SHOIQ. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Eleventh International Conference, KR 2008, Sydney, Australia, September 16-19, 2008*, pages 274–284, 2008.
11. Kathryn B. Laskey. MEBN: a language for First-Order Bayesian knowledge bases. *Artificial Intelligence*, 172(2-3):140–178, 2008.
12. Bijan Parsia and Evren Sirin. Pellet: An OWL DL reasoner. In *Third International Semantic Web Conference-Poster*, volume 18, 2004.
13. Rob Shearer, Boris Motik, and Ian Horrocks. HermiT: A highly-efficient OWL reasoner. In Catherine Dolbear, Alan Ruttenberg, and Ulrike Sattler, editors, *OWLED*, volume 432 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.
14. Herman J. ter Horst. Combining RDF and part of OWL with rules: Semantics, decidability, complexity. In *The Semantic Web - ISWC 2005, 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10, 2005, Proceedings*, pages 668–684, 2005.
15. Dmitry Tsarkov and Ian Horrocks. FaCT++ description logic reasoner: System description. In *Automated Reasoning, Third International Joint Conference, IJCAR 2006, Seattle, WA, USA, August 17-20, 2006, Proceedings*, pages 292–297, 2006.
16. W3C. OWL 2 Web Ontology Language document overview. http://www.w3.org/TR/owl2-overview/, 2012.
17. W3C. OWL 2 Web Ontology Language profiles. http://www.w3.org/TR/owl2-profiles/, 2012.