RuQAR: Reasoning with OWL 2 RL Using Forward Chaining Engines

Jaroslaw Bak

Institute of Control and Information Engineering
Poznan University of Technology
Piotrowo 3a, 60-965 Poznan, Poland
Email: jaroslaw.bak@put.poznan.pl

June 6th, 2015
Outline

1. What is RuQAR?
2. Motivation
3. The RuQAR Framework
   ▶ Features
   ▶ Architecture and Implementation
4. Evaluation
5. Conclusions and Future Work
Outline

1. What is RuQAR?
Outline

1. What is RuQAR?
2. Motivation
Outline

1. What is RuQAR?
2. Motivation
3. The RuQAR Framework
Outline

1. What is RuQAR?
2. Motivation
3. The RuQAR Framework
   - Features
Outline

1. What is RuQAR?
2. Motivation
3. The RuQAR Framework
   - Features
   - Architecture and Implementation
Outline

1. What is RuQAR?
2. Motivation
3. The RuQAR Framework
   ▶ Features
   ▶ Architecture and Implementation
4. Evaluation
Outline

1. What is RuQAR?
2. Motivation
3. The RuQAR Framework
   - Features
   - Architecture and Implementation
4. Evaluation
5. Conclusions and Future Work
What is RuQAR?

Rule-based Query Answering and Reasoning framework

Supports ABox reasoning and query answering with OWL 2 RL ontologies executed by the forward chaining rule reasoners Jess and Drools

The main goal of this tool is to provide efficient ABox reasoning as well as query answering within OWL 2 RL profile
What is RuQAR?

- Rule-based Query Answering and Reasoning framework
What is RuQAR?

- Rule-based Query Answering and Reasoning framework
- Supports ABox reasoning and query answering with OWL 2 RL ontologies executed by the forward chaining rule reasoners Jess and Drools
What is RuQAR?

- Rule-based Query Answering and Reasoning framework
- Supports ABox reasoning and query answering with OWL 2 RL ontologies executed by the forward chaining rule reasoners Jess and Drools
- The main goal of this tool is to provide efficient ABox reasoning as well as query answering within OWL 2 RL profile
Motivation

We like rules :) OWL 2 RL reasoners are known to perform poorly with large ABoxes. Description logic-based reasoners handle the TBox entailments better than the ABox ones. ABox reasoning can be performed more efficiently by a rule engine. The official list of OWL 2 reasoners supporting OWL 2 RL is limited.
Motivation

- We like rules :)

OWL 2 RL reasoners are known to perform poorly with large ABoxes. Description logic-based reasoners handle the TBox entailments better than the ABox ones. ABox reasoning can be performed more efficiently by a rule engine. The official list of OWL 2 reasoners supporting OWL 2 RL is limited.
Motivation

- We like rules :) 
- OWL 2 RL reasoners are known to perform poorly with large ABoxes
Motivation

- We like rules :)
- OWL 2 RL reasoners are known to perform poorly with large ABoxes
- Description logic-based reasoners handle the TBox entailments better than the ABox ones
Motivation

- We like rules :)
- OWL 2 RL reasoners are known to perform poorly with large ABoxes
- Description logic-based reasoners handle the TBox entailments better than the ABox ones
- ABox reasoning can be performed more efficiently by a rule engine
Motivation

- We like rules :)
- OWL 2 RL reasoners are known to perform poorly with large ABoxes
- Description logic-based reasoners handle the TBox entailments better than the ABox ones
- ABox reasoning can be performed more efficiently by a rule engine
- The official list of OWL 2 reasoners supporting OWL 2 RL is limited
Motivation

An application of an OWL 2 RL reasoner together with a currently being used forward reasoning engine can be a tricky task. Lack of tools that can generate rules for different reasoning engines. Usually, reasoning engines require data to be stored in the main memory. Lack of native and efficient rule sets that support OWL 2 RL reasoning in many popular rule engines, especially considering ABox reasoning.

We really like rules ;)

Jaroslaw Bak (PUT)
Motivation

- An application of an OWL 2 RL reasoner together with a currently being used forward reasoning engine can be a tricky task.
Motivation

- An application of an OWL 2 RL reasoner together with a currently being used forward reasoning engine can be a tricky task
- Lack of tools that can generate rules for different reasoning engines
An application of an OWL 2 RL reasoner together with a currently being used forward reasoning engine can be a tricky task.

Lack of tools that can generate rules for different reasoning engines.

Usually, reasoning engines require data to be stored in the main memory.
Motivation

- An application of an OWL 2 RL reasoner together with a currently being used forward reasoning engine can be a tricky task
- Lack of tools that can generate rules for different reasoning engines
- Usually, reasoning engines require data to be stored in the main memory
- Lack of native and efficient rule sets that support OWL 2 RL reasoning in many popular rule engines, especially considering ABox reasoning
Motivation

- An application of an OWL 2 RL reasoner together with a currently being used forward reasoning engine can be a tricky task
- Lack of tools that can generate rules for different reasoning engines
- Usually, reasoning engines require data to be stored in the main memory
- Lack of native and efficient rule sets that support OWL 2 RL reasoning in many popular rule engines, especially considering ABox reasoning
- We really like rules ;}
RuQAR’s Features

We aim at providing an easy-to-use framework for performing ABox reasoning with OWL 2 RL ontologies in any forward chaining rule engine, so it can be used in many rule-based applications. An ontology needs to be transformed into rules that are readable by a chosen engine.

Jaroslaw Bak (PUT)
RuQAR’s Features

- We aim at providing an easy-to-use framework for performing ABox reasoning with OWL 2 RL ontologies in any forward chaining rule engine which such that it can be used in many rule-based applications.
RuQAR’s Features

- We aim at providing an easy-to-use framework for performing ABox reasoning with OWL 2 RL ontologies in any forward chaining rule engine which such that it can be used in many rule-based applications.
- An ontology needs to be transformed into rules that are readable by a chosen engine.
The Abstract Syntax of Rules and Facts (ASRF) which is used to rise an abstraction level providing more universal representation of rules and facts. ASRF enables easy translation into the language of any rule engine. An implementation of mappings between ASRF and the language of the engine is required.
The Abstract Syntax of Rules and Facts (ASRF) which is used to rise an abstraction level providing more universal representation of rules and facts.
The Abstract Syntax of Rules and Facts (ASRF) which is used to rise an abstraction level providing more universal representation of rules and facts.

ASRF enables easy translation into the language of any rule engine.
The Abstract Syntax of Rules and Facts (ASRF) which is used to rise an abstraction level providing more universal representation of rules and facts.

ASRF enables easy translation into the language of any rule engine.

An implementation of mappings between ASRF and the language of the engine is required.
Rule ClassHierarchyRule_1_LUBM
If
  (Triple
    (Subject ?x)
    (Predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#type")
    (Object "http://swat.cse.lehigh.edu/onto/univ-bench.owl#ResearchAssistant"))
Then
  (Triple
    (Subject ?x)
    (Predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#type")
    (Object "http://swat.cse.lehigh.edu/onto/univ-bench.owl#Person"))
End
RuQAR’s Features – ASRF Example

Rule ClassHierarchyRule

If

(Triple
  (Subject ?x)
  (Predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#type")
  (Object "http://swat.cse.lehigh.edu/onto/univ-bench.owl#ResearchAssistant"))

Then

(Triple
  (Subject ?x)
  (Predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#type")
  (Object "http://swat.cse.lehigh.edu/onto/univ-bench.owl#Person"))

End

(Triple
  (Subject "http://www.Department6.University0.edu/GraduateStudent22")
  (Predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#type")
  (Object "http://swat.cse.lehigh.edu/onto/univ-bench.owl#ResearchAssistant"))

(Triple
  (Subject "http://www.Department6.University0.edu/GraduateStudent22")
  (Predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#type")
  (Object "http://swat.cse.lehigh.edu/onto/univ-bench.owl#Person"))
We transform an OWL 2 ontology into a set of rules and a set of facts expressed in ASRF.
RuQAR’s Features – Transformation Schema

- We transform an OWL 2 ontology into a set of rules and a set of facts expressed in ASRF
We transform an OWL 2 ontology into a set of rules and a set of facts expressed in ASRF.
RuQAR’s Features – Transformation Schema

- We transform an OWL 2 ontology into a set of rules and a set of facts expressed in ASRF

OWL 2 Ontology → HermiT (TBox Reasoning) → Inferred OWL 2 Ontology → Ontology Transformation → OWL 2 Ontology Written in ASRF
RuQAR’s Features – Transformation Schema

Table: Currently supported OWL 2 RL entailment rules.

<table>
<thead>
<tr>
<th>OWL 2 RL Specification Table</th>
<th>Supported Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 4. The Semantics of Equality</td>
<td>eq-sym, eq-rep-p, eq-rep-o</td>
</tr>
<tr>
<td></td>
<td>eq-trans, eq-rep-s, eq-rep-o</td>
</tr>
<tr>
<td>Table 5. The Semantics of Axioms about Properties</td>
<td>prp-dom, prp-fp, prp-symp, prp-eqp1, prp-eqp2, prp-inv1, prp-inv2</td>
</tr>
<tr>
<td></td>
<td>prp-rng, prp-ifp, prp-trp, prp-spo1, prp-inv1, prp-inv2</td>
</tr>
<tr>
<td>Table 6. The Semantics of Classes</td>
<td>cls-int1, cls-uni, cls-svf2, cls-hv1, cls-maxc2</td>
</tr>
<tr>
<td></td>
<td>cls-int2, cls-svf1, cls-avf, cls-hv2</td>
</tr>
<tr>
<td>Table 7. The Semantic of Class Axioms</td>
<td>cax-sco, cax-eqc2</td>
</tr>
<tr>
<td></td>
<td>cax-eqc1, cax-eqc2</td>
</tr>
</tbody>
</table>
RuQAR’s Features – Jess, Drools, SWRL

ASRF sets can be translated into Drools and Jess languages.

Semantic Web Rule Language is also supported.

RuQAR checks whether a rule is safe or not.
RuQAR’s Features – Jess, Drools, SWRL

- ASRF sets can be translated into Drools and Jess languages
RuQAR’s Features – Jess, Drools, SWRL

- ASRF sets can be translated into Drools and Jess languages
- Semantic Web Rule Language is also supported
RuQAR’s Features – Jess, Drools, SWRL

- ASRF sets can be translated into Drools and Jess languages
- Semantic Web Rule Language is also supported
- RuQAR checks whether a rule is safe or not
RuQAR’s Features – Database Mapping

Mapping method between an ontology and a relational database

Query answering functions

Storing ABox in a relational database using simple mappings
RuQAR’s Features – Database Mapping

- Mapping method between an ontology and a relational database (R2RML is an ongoing work)
RuQAR’s Features – Database Mapping

- Mapping method between an ontology and a relational database (R2RML is an ongoing work)
- Query answering functions
RuQAR’s Features – Database Mapping

- Mapping method between an ontology and a relational database (R2RML is an ongoing work)
- Query answering functions
- Storing ABox in a relational database using simple mappings
SELECT Col₁ FROM ∗ WHERE (Col₁ is not NULL);

SELECT Col₁, Col₂ FROM ∗ WHERE ((Col₁ is not NULL) AND (Col₂ is not NULL));

SELECT Col₁, Col₂, Col₃ FROM ∗ WHERE
   ((Col₁ is not NULL) AND (Col₂ is not NULL)) AND (Col₃ is not NULL));

where:

- Colᵥ are the attributes (columns) that occur in the result of a query,
- ∗ is an SQL statement; it can contain SQL commands - e.g. nested Select query or a table name,
- (Colᵥ is not NULL) means NULL results are not allowed.
RuQAR's Architecture and Implementation
RuQAR’s Architecture and Implementation

Diagram showing the architecture of RuQAR with components such as Reasoning Manager, Query Answering, Transformations, ASRF, Mapping, OWL-API, and RDB.
Evaluation

Jaroslaw Bak (PUT)

RuQAR Framework

June 6th, 2015

ABox reasoning time (sec)

Jess
Drools
HermiT
Pellet

Semintec_0  Semintec_1  Semintec_2  Semintec_3  Semintec_4
Evaluation

The diagram shows the ABox reasoning time (in seconds) for different systems:
- Jess
- Pellet
- Drools
- Hermit

The systems are compared across different versions labeled as Vicodi_0 to Vicodi_4.
Evaluation

Jaroslaw Bak (PUT)

RuQAR Framework

June 6th, 2015 17 / 20

The diagram shows the ABox reasoning time (sec) for different datasets (LUBM_1 to LUBM_4) using various reasoners:

- Jess
- Drools
- Pellet
- HermiT
Conclusions

RuQAR is a tool that:
- performs ABox reasoning and query answering with OWL 2 RL ontologies executed by forward chaining rule reasoners
- translates an OWL 2 ontology into rules
- allows for using SWRL rules together with ontologies
- provides functions to manage reasoning engines
- can store OWL individuals in a relational database
- is the first implementation of the OWL 2 RL reasoning in Drools and Jess that implements directly the semantics of OWL 2 RL which can be applied in any application requiring efficient ABox reasoning.
Conclusions

RuQAR is a tool that:
RuQAR is a tool that:

- performs ABox reasoning and query answering with OWL 2 RL ontologies executed by forward chaining rule reasoners
Conclusions

RuQAR is a tool that:

- performs ABox reasoning and query answering with OWL 2 RL ontologies executed by forward chaining rule reasoners
- translates an OWL 2 ontology into rules
Conclusions

RuQAR is a tool that:

- performs ABox reasoning and query answering with OWL 2 RL ontologies executed by forward chaining rule reasoners
- translates an OWL 2 ontology into rules
- allows for using SWRL rules together with ontologies
Conclusions

RuQAR is a tool that:

- performs ABox reasoning and query answering with OWL 2 RL ontologies executed by forward chaining rule reasoners
- translates an OWL 2 ontology into rules
- allows for using SWRL rules together with ontologies
- provides functions to manage reasoning engines
RuQAR is a tool that:

- performs ABox reasoning and query answering with OWL 2 RL ontologies executed by forward chaining rule reasoners
- translates an OWL 2 ontology into rules
- allows for using SWRL rules together with ontologies
- provides functions to manage reasoning engines
- can store OWL individuals in a relational database
RuQAR is a tool that:

- performs ABox reasoning and query answering with OWL 2 RL ontologies executed by forward chaining rule reasoners
- translates an OWL 2 ontology into rules
- allows for using SWRL rules together with ontologies
- provides functions to manage reasoning engines
- can store OWL individuals in a relational database
- is the first implementation of the OWL 2 RL reasoning in Drools and Jess that implements directly the semantics of OWL 2 RL which can be applied in any application requiring efficient ABox reasoning
We plan to:

- develop an interface that is based on the R2RML specification
- include more optimizations in query answering as well as reasoning
- provide queries execution with graphical answers
- use the NPD Benchmark for an evaluation
...and Future Work

We plan to:

- develop an interface that is based on the R2RML specification
- include more optimizations in query answering as well as reasoning
- provide queries execution with graphical answers
- use the NPD Benchmark for an evaluation
We plan to:

- develop an interface that is based on the R2RML specification
...and Future Work

We plan to:

- develop an interface that is based on the R2RML specification
- include more optimizations in query answering as well as reasoning
We plan to:

- develop an interface that is based on the R2RML specification
- include more optimizations in query answering as well as reasoning
- provide queries execution with graphical answers
We plan to:

- develop an interface that is based on the R2RML specification
- include more optimizations in query answering as well as reasoning
- provide queries execution with graphical answers
- use the NPD Benchmark for an evaluation
Thank you for listening!

More information available at:
http://etacar.put.poznan.pl/jaroslaw.bak/RuQAR.php

Questions?