Rule-based query answering method for a knowledge base of economic crimes

PhD thesis @ RuleML 2011 Doctoral Consortium

Jaroslaw Bak
Institute of Control and Information Engineering
Poznan University of Technology
Outline

1. Problem statement
2. Overview of existing solutions
3. Rule-based query answering methods
   • with hybrid reasoning
   • with forward reasoning and extended rules
4. Rule-based mapping
5. Knowledge base of economic crimes – The Hydra case
6. Example queries and results
7. Conclusion and future work
Problem statement (1)

Research question:

How to efficiently query a relational database at the conceptual level defined in a rule-based system?
Problem statement (2)

Three main issues:

1. Rule-based query answering.
2. The combination of a rule-based system and a relational database.
3. The construction of the knowledge base (i.e. knowledge of economic crimes).
Overview of existing solutions (1)

- Backward chaining vs. forward chaining and OpenRuleBench initiative – Prolog wins [Bry et al]
- Magic Transformation [Ramakrishnan et al, Eiter et al] – a program $P$ and a query $Q$ are transformed into a new program, $\text{magic}(P \cup Q)$
- Implementation alternatives for bottom-up evaluation – push and pull methods [Brass]
Overview of existing solutions (2)

• The combination of rules with relational databases [Motik et al., Calvanese et al, Lukácsy et al, Hustadt et al]

• The problem of applying rules and ontology in economic crimes is quite new – some works in information management exist [Biasotii et al, Casellas et al], not reasoning [Breuker]

• FFPoirot project – fraudulent Internet investment pages and the Nigerian letter fraud [Zhao et al]
Rule-based query answering method

Assumptions

- We use a production rule system (the Jess engine) with forward and backward chaining
- We express the conceptual knowledge with the Horn-SHIQ ontology combined with SWRL (Horn-like) rules
- We use conjunctive queries in the form of directed graphs
- We use Datalog Safety and DL-safe rules to achieve the decidability of our system
- We apply our methods with the knowledge base of economic crimes: fraudulent disbursement and money laundering
- Our approaches are implemented in the Semantic Data Library (SDL)
• Two Jess instances:
  – backward engine is used to gather data from a relational database
  – forward engine is used to answer a query
• Rules are expressed in the Jess language and obtained from the transformation of an OWL ontology with SWRL rules
• We developed an algorithm for grouping SQL queries
• This approach was presented at the RuleML2009 conference
Rule-based query answering method

Hybrid reasoning (2)

Get QUERYRULE activations

For each concept/role occurring in the Query do:

1. Query
2. Run engine
3. Add need-X fact
   where X is the current concept/role
4. RunQueriesFromJess call
5. Results as Jess facts
6. Results as Jess facts
7. SQL queries
8. Results
9. Results
10. Copy Jess facts, remember values of the variables, clear the working memory of backward chaining engine and go to the ④
11. Forward Chaining Engine
12. Answer

 Query as body of the rule; create QUERYRULE

SDL-API

JessDBAccess class

DB
Rule-based query answering method
Forward reasoning with extended rules (1)

• Based on the modified magic transformation
• We use extended rules - rules that are generated automatically from the basic ones for the evaluation purposes
• Extended rules are query-independent – in contrast to the magic transformation
• Generation of the extended rules are based on the particular sip algorithm – goal- and dependency-oriented generation
• One instance of the Jess engine is used
• This method is more efficient than the hybrid one
• Accepted for the presentation at the RuleML2011@BRF conference
Rule-based query answering method

Forward reasoning with extended rules (2)

\[ p_1(?x, ?y), p_2(?y, ?w), p_3(?w), ?w \neq ?x \Rightarrow h_1(?x, ?w) \]

\[ p_1(?x, ?y), p_2(?y, ?w), p_3(?w), ?w \neq ?x, h_1(?x, ?w)^C \]

\[ \Rightarrow h_1(?x, ?w) \]

Applicable to:

\[ h_1(?x, ?w) \]

\[ h_1(nil, nil)^C \Rightarrow p_1(nil, nil)^C \]

\[ h_1(nil, nil)^C \Rightarrow p_2(nil, nil)^C \]

\[ \text{Etc.} \]

\[ h_1(?x, ?), p_1(?x, ?y) \Rightarrow p_2(?y, nil)^C \]

\[ h_1(?x, ?)^C, ?x \neq nil \Rightarrow p_1(?x, nil)^C \]

\[ \text{Etc.} \]

\[ h_1(?x, ?w)^C, ?w \neq nil \Rightarrow p_2(nil, ?w)^C \]

\[ h_1(?x, ?w)^C, p_2(?y, ?w) \Rightarrow p_1(nil, ?y)^C \]

\[ \text{Etc.} \]
Rule-based query answering method
The architecture of the SDL library

Ontology-based knowledge transformed into rules

Rule-based knowledge

Jess engine(s)

Query

Answer

Relational Database

Mapping rules
Rule-based mapping

• "Essential" concept/role:
  Buyer is-a Company is-a Institution

• For every essential concept/role a SQL query of the following form is created:

```
SELECT [R] FROM [T] <WHERE> <C, AND, OR>
```

where R – result columns, T – tables, C – constraints, AND, OR – optional SQL commands

• Example: AdultMan

```
SELECT id FROM persons
WHERE age>18
AND gender='Male';
```
Rule-based mapping - Example

(defrule Def-MoneyTransferTo

  ?r<- (need-Triple

    (p "MoneyTransferTo") (s ?x) (o ?y))

=>

  (bind ?query (str-cat

    "SELECT id, receiver FROM transfers;"))

  (?*access* runQueriesFrom Jess

    "Def-MoneyTransferTo"

    ?query

    "s;id;o;receiver;p;MoneyTransferTo;"

    (str-cat ?x ";" ?y ";")

    "triple"

    ?*conn*

    (engine))

  (retract ?r)

)
Knowledge base of economic crimes

The Hydra Case

Fraudulent Disbursement

- Comp. A
  - Service/goods: invoice 976, wire 976
  - Fictitious work statement
  - Work done by Comp. A itself
  - Owner of Comp. A

- Comp. B
  - Service/goods: invoice 854, wire 854
  - Fictitious work statement
  - CEO of Comp. B

- Single person Comp. C
  - Service/goods: invoice 450
  - No work statement
  - Owner of Comp. C

- Single person Comp. D
  - Service/goods
  - Owner of Comp. D

Money Laundering

- Difference between transfers 122 (commission)
- Attempt of withdrawal of 450 cash
- Intended cash flow
Example queries and results

Generation of the simulated input data

- Information about employees and their position in a company
- Invoices with all obligatory elements (payer, seller, product, etc.)
- Work approval documents (or the lack of them)
- Signatures on documents
- Goods and services
- Companies and their legal form
- Money turnovers: money transfers, payments and withdrawals
- Legal articles (name, ID and content)
- Information about illicit personal gains and damages to companies (with values)
- Other facts, like who knows about what (Person knowsAbout document) – these data come from testimonies.
Example queries and results

Queries

1. IllicitPersonalGain" `<rdf:type>` ?g `<isAccompaniedBy>` ?p `< achieves>` ?c `<rdf:type>` Company `<hasValue>` ?h `<rdf:type>` HighValue

2. 12 `< rdf:type` Director `<worksFor` ?c `<rdf:type` Company `<worksFor` 11 `<rdf:type` Director `< fallsUnder` ?a `<rdf:type` Company `Principal`

3. ?d `<isSignedBy` ?p `<rdf:type` Person `<rdf:type` FalsifiedComplex `<InternalLegalDocument` ?d `<hasValue` ?h `<rdf:type` HighValue `< achieves` ?g `<rdf:type` Company `< isAccompaniedBy` ?c `< rdf:type` Company

4. 11 `< rdf:type` Director `< fallsUnder` Art.296§1 `< rdf:type` ?a `< rdf:type` Company `Principal`

5. 84 `< rdf:type` ?a1 `< fallsUnder` Art.299§1 `< rdf:type` ?a2 `< fallsUnder` Art.299§5
# Example queries and results

## Results

<table>
<thead>
<tr>
<th>Query and info</th>
<th>Database 20</th>
<th>Database 100</th>
<th>Database 200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query 1 Results</td>
<td>781</td>
<td>219</td>
<td>1328</td>
</tr>
<tr>
<td>Rules fired</td>
<td>54</td>
<td>54</td>
<td>474</td>
</tr>
<tr>
<td></td>
<td>74</td>
<td>251</td>
<td>441</td>
</tr>
<tr>
<td>Query 2 Results</td>
<td>2734</td>
<td>437</td>
<td>37141</td>
</tr>
<tr>
<td>Rules fired</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1076</td>
<td>1 506</td>
<td>36260</td>
</tr>
<tr>
<td>Query 3 Results</td>
<td>2875</td>
<td>359</td>
<td>36344</td>
</tr>
<tr>
<td>Rules fired</td>
<td>18</td>
<td>18</td>
<td>322</td>
</tr>
<tr>
<td></td>
<td>1 367</td>
<td>2 005</td>
<td>38457</td>
</tr>
<tr>
<td>Query 4 Results</td>
<td>5437</td>
<td>1 859</td>
<td>128719</td>
</tr>
<tr>
<td>Rules fired</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2040</td>
<td>5 467</td>
<td>58 520</td>
</tr>
<tr>
<td>Query 5 Results</td>
<td>9312</td>
<td>1 234</td>
<td>Time exceeded 10 minutes</td>
</tr>
<tr>
<td>Rules fired</td>
<td>1</td>
<td>1</td>
<td>61 199</td>
</tr>
<tr>
<td></td>
<td>2540</td>
<td>5 828</td>
<td></td>
</tr>
</tbody>
</table>
Conclusions

- We developed two approaches for rule-based query answering task: the hybrid reasoning and extended rules.
- Approaches extend functionality of the Jess reasoning engine.
- Approaches were tested with our knowledge base of economic crimes.
- SDL library enables to query a relational database in terms of ontology concepts/roles.
- Answer is always up-to-date.
- Queries are in the form of directed graphs.
Future work

- More **optimizations** in the query answering method
- Method for **rule-dependent** sips
- **Comparison** to other approaches (OpenRuleBench, OWLim, DLEJena etc.)
- **Extension** of rule-based knowledge base of economic crimes
- More **formal** description
- **Graphical** user interface for queries execution, rules creation and data (facts) analysis

- **FINISH THE PhD 😊**
THE END

Thanks for listening! 😊