Offshore Holdings Analytics Using Datalog+ RuleML Rules

Mohammad Sadnan Al Manir Christopher J.O. Baker

Department of Computer Science & Applied Statistics
University of New Brunswick

RuleML 2014 Rulebase Competition
Prague, Czech Republic
August 19, 2014

Outline

1. Motivation
2. Background & Contribution
3. Rulebase: Formalizing Facts, Rules and Queries
4. Summary and Future Work
5. Extra Slides

Analytics

- Discovery and communication of meaningful patterns in data
- Valuable for areas rich in recorded information
- Relies on the simultaneous application of
  - statistics, computer programming, operations research to quantify performance
- Favours data visualization

Datalog+ Rules

- New features of Deliberation RuleML 1.01
  - Combine Datalog extensions defining Datalog+ such as
    - Existential Rules
    - Equality Rules
    - Integrity Rules
  - All of the above features are allowed in rule heads

Goals

- Formalize use case on Offshore Holdings
- Discover meaningful patterns
- Use the patterns in analytics

Offshore Holdings

- Exposure of 130,000 offshore accounts in 2013
    - International Consortium of Investigative Journalists (ICIJ)
    - Ten offshore jurisdictions investigated
    - Offshore Leaks Database: 2.5 million records
    - Relationships and networks among
        - people or companies and offshore entities
        - director, shareholder, trustee etc.
    - Linked to 170 countries

Our Contribution

- Evaluate effectiveness of Datalog+ RuleML
    - Building a rulebase
        - Based on a subset of Offshore Leaks Database
        - Relations derived among entities
        - E.g.: Top-level officials owning Offshore Holdings
        - Difficult to find such connections manually
    - Incremental step-by-step rule authoring
    - 37 transactions in the rulebase
        - 14 Asserts
        - 3 Retracts
        - 20 Queries
    - Covers the 3 Datalog extensions of Datalog+
    - Validated against RELAX NG Compact Syntax (RNC) schemas

Rulebase

- Fact: The president of `Azerbaijan` is `Ilham`
- Query: Who is the president of Azerbaijan?
- Result: binding of `x` to `Ilham`

```
<RuleML xmlns="http://ruleml.org/spec">
  <Assert>
    <Atom>
      <Rel>president</Rel>
      <Ind>Ilham</Ind>
      <Ind>Azerbaijan</Ind>
    </Atom>
  </Assert>
  <Query>
    <Atom>
      <Rel>president</Rel>
      <Var>x</Var>
      <Ind>Azerbaijan</Ind>
    </Atom>
  </Query>
  ...
</RuleML>
```

Rulebase (Cont’d)

- No one is both the president and the prime minister of a country (empty `<Or/>` used for falsity)

```
<Assert>
  <Forall>
    <Var>x</Var>
    <Implies>
      <if>
        <And>
          <Atom>
            <Rel>president</Rel>
            <Var>x</Var>
          </Atom>
          <Atom>
            <Rel>primeMinister</Rel>
            <Var>x</Var>
          </Atom>
        </And>
      </if>
      <then>
        <Or/>
      </then>
    </Implies>
  </Forall>
</Assert>
```

Rulebase (Cont'd)

- `Ilham` is the prime minister of `Azerbaijan`

```
<Assert>
  <Atom>
    <Rel>primeMinister</Rel>
    <Ind>Ilham</Ind>
    <Ind>Azerbaijan</Ind>
  </Atom>
</Assert>
```

- Is there any inconsistency?

```
<Query>
  <Or/>
</Query>
```

Rulebase (Cont'd)

- `Ilham` is the prime minister of `Azerbaijan`

```
<Assert>
  <Atom>
    <Rel>primeMinister</Rel>
    <Ind>Ilham</Ind>
    <Ind>Azerbaijan</Ind>
  </Atom>
</Assert>
```

- Is there any inconsistency?

```
<Query>
  <Or/>
</Query>
```

Rulebase (Cont'd)

- Yes, `Ilham` cannot hold both the post of a president and a prime minister of `Azerbaijan`
- Retract

```
<Retract>
  <Atom>
    <Rel>primeMinister</Rel>
    <Ind>Ilham</Ind>
    <Ind>Azerbaijan</Ind>
  </Atom>
</Retract>
```

- Is there any inconsistency?
  - Fails

Rulebase (Cont’d)

- Yes, `Ilham` cannot hold both the post of a president and a prime minister of `Azerbaijan`
- Retract

```
<Retract>
  <Atom>
    <Rel>primeMinister</Rel>
    <Ind>Ilham</Ind>
    <Ind>Azerbaijan</Ind>
  </Atom>
</Retract>
```

- Is there any inconsistency?
  - Fails

Rulebase (Cont'd)

- Yes, `Ilham` cannot hold both the post of a president and a prime minister of `Azerbaijan`
- Retract

```
<Retract>
  <Atom>
    <Rel>primeMinister</Rel>
    <Ind>Ilham</Ind>
    <Ind>Azerbaijan</Ind>
  </Atom>
</Retract>
```

- Is there any inconsistency?
    - Fails

Rulebase (Cont'd)

- Two persons have family ties if they are either married to each other, or there is a parent-child relationship between them, or they are distant relatives

$$\forall x, y : \mathit{marriedTo}(x, y) \lor \mathit{hasChild}(x, y) \lor \mathit{hasDistantRelative}(x, y) \Rightarrow \mathit{hasFamilyTies}(x, y)$$

- `Ilham` is married to `Mehriban`
- `Ilham` has a daughter `Arzu` (specializing *hasChild*)
- `Ilham` has a daughter `Leyla` (specializing *hasChild*)

Rulebase (Cont'd)

- Everything that operates either as an intermediary company or as an offshore company is designated as a company

  $$\forall x : \mathit{intermediaryCompany}(x) \lor \mathit{offshoreCompany}(x) \Rightarrow \mathit{company}(x)$$

- `ArborInvestmentsLtd` operates as an intermediary company
- `NaziqAndPartners` operates as an intermediary company
- `HarvardManagementLtd` operates as an offshore company

Rulebase (Cont’d)

- Nothing is both an onshore company and an offshore company

```
<Assert>
  <Forall>
    <Var>x</Var>
    <Implies>
      <if>
        <And>
          <Atom>
            <Rel>onshoreCompany</Rel>
            <Var>x</Var>
          </Atom>
          <Atom>
            <Rel>offshoreCompany</Rel>
            <Var>x</Var>
          </Atom>
        </And>
      </if>
      <then>
        <Or/>
      </then>
    </Implies>
  </Forall>
</Assert>
```

Rulebase (Cont'd)

- HarvardManagementLtd is an onshore company

```
<Assert>
  <Atom>
    <Rel>onshoreCompany</Rel>
    <Ind>HarvardManagementLtd</Ind>
  </Atom>
</Assert>
```

- Is there any inconsistency?

```
<Query>
  <Or/>
</Query>
```

Rulebase (Cont'd)

- HarvardManagementLtd is an onshore company

```
<Assert>
  <Atom>
    <Rel>onshoreCompany</Rel>
    <Ind>HarvardManagementLtd</Ind>
  </Atom>
</Assert>
```

- Is there any inconsistency?

```
<Query>
  <Or/>
</Query>
```

Rulebase (Cont'd)

- Yes, `HarvardManagementLtd` cannot be both an onshore company and an offshore company
- Retract

```
<Retract>
  <Atom>
    <Rel>onshoreCompany</Rel>
    <Ind>HarvardManagementLtd</Ind>
  </Atom>
</Retract>
```

- Is there any inconsistency?
  - Fails

Rulebase (Cont'd)

- Yes, HarvardManagementLtd cannot be both an onshore company and an offshore company
- Retract

```
<Retract>
  <Atom>
    <Rel>onshoreCompany</Rel>
    <Ind>HarvardManagementLtd</Ind>
  </Atom>
</Retract>
```

- Is there any inconsistency?
  - Fails

Rulebase (Cont'd)

- Yes, `HarvardManagementLtd` cannot be both an onshore company and an offshore company
- Retract

```
<Retract>
  <Atom>
    <Rel>onshoreCompany</Rel>
    <Ind>HarvardManagementLtd</Ind>
  </Atom>
</Retract>
```

- Is there any inconsistency?
  - Fails

Rulebase (Cont'd)

- Anyone is a shareholder of a company if and only if there exists a certain level of stocks of that company which he/she owns (*ownsStockin-atLevel*), namely Small-cap level or Medium-cap level or Large-cap level

  $$\forall x, y : \mathit{shareHolderOf}(x, y) \equiv \exists z : \mathit{ownsStockin\text{--}atLevel}(x, y, z)$$

- `Arzu` owns `Medium-cap` level of stocks in `ArborInvestmentsLtd`

  $$\mathit{ownsStockin\text{--}atLevel}(\texttt{Arzu}, \texttt{ArborInvestmentsLtd}, \texttt{Medium-cap})$$

Rulebase (Cont'd)

- Every stock owner owns at most one level of stocks of a company

$$\forall x, y, z_1, z_2 : \mathit{ownsStockin\text{–}atLevel}(x, y, z_1) \land \mathit{ownsStockin\text{–}atLevel}(x, y, z_2) \Rightarrow z_1 = z_2$$

Rulebase (Cont'd)

- Is there any individual who owns Small-cap level of stocks in ArborInvestmentsLtd?
  - Fails because no individuals found who own Small-cap level of stocks
- List all shareholders of ArborInvestmentsLtd
  - `Arzu` is the shareholder

Rulebase (Cont'd)

- There is a link between a person and a company if the person is a director or a shareholder of the company or he/she owns a level of stocks in the company

$$\forall x, y \; \exists z : \mathit{directorOf}(x, y) \;\vee\; \mathit{shareHolderOf}(x, y)$$
$$\vee\; \mathit{ownsStockin\text{–}atLevel}(x, y, z) \Rightarrow \mathit{hasLinksTo}(x, y)$$

- Which individuals and companies are linked to each other?
  - `Arzu` and `ArborInvestmentsLtd`

Rulebase (Cont'd)

- If a company C$_1$ manages assets of another company C$_2$ and the latter, i.e. C$_2$, manages assets of yet another company C$_3$, then company C$_1$ manages the assets of the company C$_3$

$$\forall x, y, z \colon \mathit{manageAssets}(x, y) \land \mathit{manageAssets}(y, z) \Rightarrow \mathit{manageAssets}(x, z)$$

- `ArborInvestmentsLtd` manages assets of `NaziqAndPartners`
- `NaziqAndPartners` manages assets of `HarvardManagementLtd`

Rulebase (Cont'd)

- Find all the companies whose assets are being managed and the companies managing them

```
<Query>
  <Atom>
    <Rel>manageAssets</Rel>
    <Var>x</Var>
    <Var>z</Var>
  </Atom>
</Query>
```

- `ArborInvestmentsLtd` is, by the above transitivity rule, managing assets of the distant company `HarvardManagementLtd`

Rulebase (Cont'd)

- The president of a country may have offshore investments in a company if his/her family members have links to companies managing assets in that offshore company

$$\begin{aligned}\forall p, c, fm, ic, oc : \mathit{president}(p, c) \wedge \mathit{hasFamilyTies}(p, fm) \\ \wedge \mathit{hasLinksTo}(fm, ic) \wedge \mathit{manageAssets}(ic, oc) \\ \Rightarrow \mathit{possiblyHasOffshoreInvestmentIn}(p, oc)\end{aligned}$$

Rulebase (Cont'd)

- Find individuals and companies in which the individuals possibly have offshore investments

```
<RuleML xmlns="http://ruleml.org/spec">
  ...
  <Query>
    <Atom>
      <Rel>possiblyHasOffshoreInvestmentIn</Rel>
      <Var>p</Var>
      <Var>oc</Var>
    </Atom>
  </Query>
</RuleML>
```

- President `Ilham` of `Azerbaijan` whose daughter `Arzu` has links with the company `ArborInvestmentsLtd`, which indirectly manages the assets in the offshore company called `HarvardManagementLtd`

Conclusions

- Datalog+ RuleML 1.01/XML rules
    - Step-by-step incremental formalization
    - Discovery of interesting relationships not seen at plain sight
    - Offshore Holdings analytics
- Cover two sublanguages *datalogplus_min* and *disdatalogplus_min*
- Fine-grained rule sublanguages will facilitate using appropriate resources, e.g. inference engines
- Outlook
    - Datalog+ rule expressiveness calls for future work, e.g. on Datalog+ engines, OWL-RuleML combination, and extending the use case

Rulebase Authoring, Schema Design, and Validation

- XML Editor ▸ XML Copy Editor
- Rulebase ▸ Offshore Holdings analytics rulebase
- Modular sYNtax confiGurator (MYNG 1.01) ▸ Relax NG schema
- Validator Web Service ▸ Validator
- Validation ▸ Validation

Rulebase Authoring, Schema Design, and Validation

- XML Editor ▸ XML Copy Editor

  http://xml-copy-editor.sourceforge.net
- Rulebase ▸ Offshore Holdings analytics rulebase

  http://deliberation.ruleml.org/1.01/exa/RulebaseCompetition2014/OffshoreHoldingAnalytics.ruleml
- Modular sYNtax confiGurator (MYNG 1.01) ▸ Relax NG schema

  http://deliberation.ruleml.org/1.01/relaxng/naffologeq_relaxed.rnc
- Validator Web Service ▸ Validator http://validator.nu/
- Validation ▸ Validation

  http://validator.nu/?doc=http://deliberation.ruleml.org/1.01/exa/RulebaseCompetition2014/OffshoreHoldingAnalytics.ruleml&schema=http://deliberation.ruleml.org/1.01/relaxng/naffologeq_relaxed.rnc

Slides Added
After
Rulebase Competition 2014

Rewriting Equivalence Formula (cf. Slide 17)

- Anyone is a shareholder of a company if and only if there exists a certain level of stocks of that company which he/she owns, namely Small-cap level or Medium-cap level or Large-cap level

$$\forall x, y: \mathit{shareHolderOf}(x, y) \equiv \exists z: \mathit{ownsStockin\text{–}atLevel}(x, y, z)$$

can be rewritten as a conjunction of two implications

$$\forall x, y: ((\mathit{shareHolderOf}(x, y) \Rightarrow \exists z: \mathit{ownsStockin\text{–}atLevel}(x, y, z)$$

$$\wedge\ \exists z: \mathit{ownsStockin\text{–}atLevel}(x, y, z) \Rightarrow (\mathit{shareHolderOf}(x, y))$$

Rewriting Equivalence Formula (Cont'd)

Original Equivalence Formula

```
<Forall>
  <Var>x</Var>
  <Var>y</Var>
  <Equivalent>
    <Atom>
      <Rel>shareHolderOf</Rel>
      <Var>x</Var>
      <Var>y</Var>
    </Atom>
    <Exists>
      <Var>z</Var>
      <Atom>
        <Rel>ownsStockin-atLevel</Rel>
        <Var>x</Var>
        <Var>y</Var>
        <Var>z</Var>
      </Atom>
    </Exists>
  </Equivalent>
</Forall>
```

Rewriting Equivalence Formula (Cont'd)

Original Equivalence Formula

```
<Forall>
  <Var>x</Var>
  <Var>y</Var>
  <Equivalent>
    <Atom>
      <Rel>shareHolderOf</Rel>
      <Var>x</Var>
      <Var>y</Var>
    </Atom>
    <Exists>
      <Var>z</Var>
      <Atom>
        <Rel>ownsStockin-atLevel</Rel>
        <Var>x</Var>
        <Var>y</Var>
        <Var>z</Var>
      </Atom>
    </Exists>
  </Equivalent>
</Forall>
```

Conjunction of Implications

```
<Forall>
  <Var>x</Var><Var>y</Var>
  <And>
    <Implies>
      <if><Atom><Rel>shareHolderOf</Rel>
          <Var>x</Var><Var>y</Var></Atom>
      </if>
      <then><Exists><Var>z</Var>
        <Atom><Rel>ownsStockin-atLevel</Rel>
          <Var>x</Var><Var>y</Var><Var>z</Var>
        </Atom></Exists>
      </then>
    </Implies>
    <Implies>
      <if><Exists><Var>z</Var>
        <Atom><Rel>ownsStockin-atLevel</Rel>
          <Var>x</Var><Var>y</Var><Var>z</Var>
        </Atom></Exists>
      </if>
      <then><Atom><Rel>shareHolderOf</Rel>
          <Var>x</Var><Var>y</Var></Atom>
      </then>
    </Implies>
  </And>
</Forall>
```

Update: Schema Design and Validation (cf. Slide 26)

- Our rulebase ▸ Offshore Holdings analytics rulebase
  - Contains *no* rules with negation-as-failure
- The schema (*naffologeq*) ▸ Relax NG schema
  - Covers First-Order Logic (FOL) with equality, but would also accommodate negation-as-failure
- Validation
  - Schema without negation-as-failure is sufficient
  - ~~naf~~fologeq

Update: Schema Design and Validation (cf. Slide 26)

- Our rulebase ▸ Offshore Holdings analytics rulebase
  - Contains *no* rules with negation-as-failure
- The schema (*naffologeq*) ▸ Relax NG schema
  - Covers First-Order Logic (FOL) with equality, but would also accommodate negation-as-failure
- Validation
  - Schema without negation-as-failure is sufficient
  - ~~naf~~fologeq

Update: Schema Design and Validation (cf. Slide 26)

- Our rulebase ▸ Offshore Holdings analytics rulebase
  - Contains *no* rules with negation-as-failure
- The schema (*naffologeq*) ▸ Relax NG schema
  - Covers First-Order Logic (FOL) with equality, but would also accommodate negation-as-failure
- Validation
  - Schema without negation-as-failure is sufficient
  - ~~naf~~fologeq

Update: Schema Design and Validation (cf. Slide 26)

- Our rulebase ▸ Offshore Holdings analytics rulebase
  - Contains *no* rules with negation-as-failure
- The schema (*naffologeq*) ▸ Relax NG schema
  - Covers First-Order Logic (FOL) with equality, but would also accommodate negation-as-failure
- Validation
  - Schema without negation-as-failure is sufficient
  - ~~naf~~fologeq

Update: Schema Design and Validation (Cont'd)

- Preferred schema (fologeq) ▸ Relax NG schema
    - Covers First-Order Logic (FOL) with equality
- Validated by the Relax NG (fologeq) schema ▸ Validation
- What if negation-as-failure (Naf) is now used?

Update: Schema Design and Validation (Cont'd)

- Preferred schema (fologeq) ▸ Relax NG schema
  - Covers First-Order Logic (FOL) with equality
- Validated by the Relax NG (fologeq) schema ▸ Validation
- What if negation-as-failure (Naf) is now used?

Update: Schema Design and Validation (Cont’d)

- Naf: If ?X is not an investigative journalist then Ilham likes ?X

```
<RuleML xmlns="http://ruleml.org/spec">
  ...
  <Assert>
    <Forall>
      <Var>X</Var>
      <Implies>
        <if>
          <Naf>
            <Atom>
              <Rel>investigativeJournalist</Rel>
              <Var>X</Var>
            </Atom>
          </Naf>
        </if>
        <then>
          <Atom>
            <Rel>likes</Rel>
            <Ind>Ilham</Ind>
            <Var>X</Var>
          </Atom>
        </then>
      </Implies>
    </Forall>
  </Assert>
  ...
</RuleML>
```

Update: Schema Design and Validation (Cont'd)

- Schema (naffologeq) instead of (fologeq) ▸ Relax NG schema
    - Covers First-Order Logic (FOL) with equality, but would also accommodate negation-as-failure
- Validated by the Relax NG (naffologeq) schema ▸ Validation
- Opens yet more avenues for future work