Thoughts on Rulelog and Planning of a Mini-Series on Rules for Ontolog Forum

Benjamin Grosof*

July 25, 2013

Ontolog Forum‡

Globally accessible webconference session

* Benjamin Grosof & Associates, http://www.mit.edu/~bgrosof/
and
Coherent Knowledge Systems http://www.coherentknowledge.com

‡ http://ontolog.cim3.net/cgi-bin/wiki.pl?ConferenceCall_2013_07_25

Aspects to consider for mini-series on rules

- **Rulelog in more detail – incl. draw on existing tutorial material**
  - Concepts and logical foundations of its knowledge representation and reasoning (KRR)
    - Put meta in knowledge, as web/markup puts meta in data
    - Defeasibility (exception-ability). Bounded rationality. Higher-order. Provenance.
  - **Use cases and applications. Requirements analysis.
    - Financial, health, legal, intelligence analysis, science, education, …
    - Ontology mapping. Policies, regulations, contracts, laws. Explanation-based edutech.
  - **Standards and interoperability with other semantic/rule KRR
    - RIF, RuleML, LegalRuleML. SQL, SPARQL. FOL, Common Logic. OWL.
  - **Tools, incl. open-source. XSB, Flora-2, Coherent.
  - *Adapt existing conference-tutorial material given at AAAI-13, ISWC-2012, other conf.'s*
    - *http://www.mit.edu/~bgrosof/#AAAI13RulesTutorial*
  - *Follow up OF session* http://ontolog.cim3.net/cgi-bin/wiki.pl?ConferenceCall_2013_06_20

- **Natural language (NLP) for rule authoring and HCI**
  - Textual logic: unrestricted English mapped into (and from) logic, using logic-based NLP
    - Textual terminology. Rapid interactive disambiguation.
  - Restricted NL. SBVR.

- **More ideas:**
  - Focus/drill on **'d items above
  - Visualization, incl. for rule authoring and HCI
  - Combination of logical with probabilistic/statistical incl. for data analysis/mining and discovery

- Optional Slides Follow about Rulelog and Textual Logic

Rulelog: Overview

- **First KRR to meet central challenge:**

  rich -- higher order logic formulas, incl. as target for text interpretation
  + defeasible -- handle exceptions, change in K, change in world
  + tractable -- polynomial-time in worst-case
- **New rich logic: based on databases, not classical logic**
  - Expressively extends normal declarative logic programs (LP)
  - Transforms into LP (the logic of DB's (SQL, SPARQL) and pure Prolog)
    - Production/ECA business rules expressiveness is similar to DB's
  - LP (not FOL) is "the 99%" of practical structured info mgmt. today
- **In draft as industry standard (RuleML submission to W3C RIF and …)**
- **Associated new reasoning techniques to implement it**
- **Prototyped in Vulcan's SILK. Commercialization by Coherent Knowledge Systems.**
  - Mostly open source: Flora-2 and XSB Prolog
- **Applications:** college-level science (e.g., AP Biology), legal analysis and reasoning (Regulation W), financial compliance (Financial Industry Business Ontology), health care treatment protocols, national intelligence, privacy,

Biology Example in Rulelog

- Biology information about cells and nuclei:

```
“A eukaryotic cell has a nucleus.”
@[id->i1, tag->r1] forall(?x)^(?x(is(a(eukaryotic(cell)))) ==> ?x(has(a(nucleus)))).
“A red blood cell has no nucleus.”
@[id->i2, tag->r2] forall(?x)^(?x(is(a(red(blood(cell))))) ==> neg ?x(has(a(nucleus)))).
“A eukaryotic cell during anaphase has no nucleus.”
@[id->i3, tag->r3] forall(?x)^(?x(is(a(eukaryotic(cell(during(anaphase)))))) ==> neg ?x(has(a(nucleus)))).
```

- Prioritization:

```
\overrides(r2,r1).
\overrides(r3, r1).
```

- Ontology information:

```
@[strict] red(blood(cell)) ## eukaryotic(cell).
 @[strict] eukaryotic(cell(during(anaphase))) ## eukaryotic(cell) .
cell41(is(a(eukaryotic(cell)))) . // Some cells
cell52 # red(blood(cell)).
cell63(is(a(eukaryotic(cell(during(anaphase)))))) .
```

- Queries:

```
?- ?x(has(?y(nucleus))). // What has or doesn't have a nucleus?
?- cell41(has(a(nucleus))) .  // is true
?- neg cell52(has(a(nucleus))) .  // is true, and without the neg is false
```

Rulelog: more details

- Defeasibility based on ***argumentation theories (AT)*** [Wan, Grosof, Kifer 2009]
  - Meta-rules (~10's) specify principles of debate, thus when rules have exceptions
  - Prioritized conflict handling. Ensures consistent conclusions. Efficient, flexible, sophisticated defeasibility.
- ***Restraint***: semantically clean **bounded rationality** [Grosof & Swift, AAAI-13]*
  - Leverages "undefined" truth value to represent "not bothering"
  - Extends well-foundedness in LP
- ***Omniformity***: higher-order logic formula syntax, incl. hilog, rule id's
  - Omni-directional disjunction. Skolemized existentials. [Grosof (invited), RuleML-2013]*
  - Avoids general reasoning-by-cases (cf. unit resolution).
- Sound interchange of K with all major standards for sem web K
  - Both FOL & LP, e.g.: RDF(S), OWL-DL, SPARQL, CL
- Reasoning techniques based on extending tabling in LP inferencing
  - Truth maintenance, justifications incl. why-not, trace analysis for KA debug, term abstraction, delay subgoals [Andersen et al, RuleML-2013 (Challenge)]

For more info, see [Grosof et al, AAAI-13 Tutorial]* – largely about Rulelog

* preprint/prelim-v. already avail.

Example: Ontology Translation, leveraging hilog and exceptions

```
/*  Company BB reports operating earnings using R&D operating cost which includes price of a
    small company acquired for its intellectual property.  Organization GG wants to view
    operating cost more conventionally which excludes that acquisition amount.  We use rules to
    specify the contextual ontological mapping.  */
 @normallyBringOver  ?categ(GG)(?item)  :- ?categ(BB)(?item).
 @acquisitionsAreNotOperating   neg ?categ(GG)(?item) :-
      acquisition(GG)(?item) and (?categ(GG) :: operating(GG)).
 \overrides(acquisitionsAreNotOperating, normallyBringOver).  /* exceptional */
 acquisition(GG)(?item) :- price_of_acquired_R_and_D_companies(BB)(?item).
 R_and_D_salaries(BB)(p1001).   p1001[amount -> $25,000,000].
 R_and_D_overhead(BB)(p1002).   p1002[amount -> $15,000,000].
 price_of_acquired_R_and_D_companies(BB)(p1003).   p1003[amount -> $30,000,000].
 R_and_D_operating_cost(BB)(p1003).  /* BB counts the acquisition price item in this category */
 R_and_D_operating_cost(GG) :: operating(GG).
 Total(R_and_D_operating_cost)(BB)[amount -> $70,000,000].  /* rolled up by BB cf. BB's definitions */
 Total(R_and_D_operating_cost)(GG)[amount -> ?x] :- … .  /* roll up the items for GG cf. GG's definitions */
```

As desired: |=   R_and_D_salaries(GG)(p1001)

|=   neg R_and_D_operating_cost(GG)(p1003) /* GG doesn't count it */

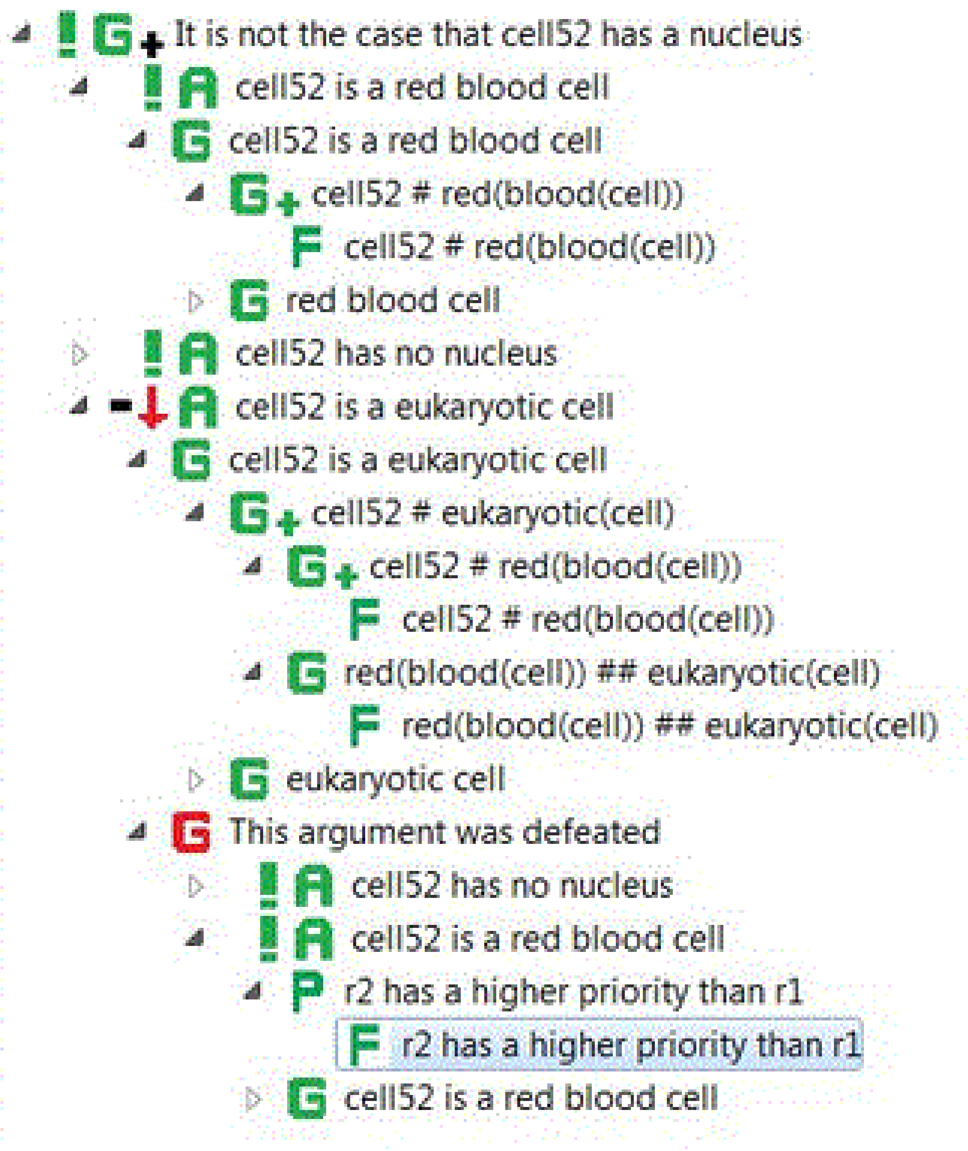
|=   Total(R_and_D_operating_cost)(GG)[amount -> $40,000,000]

Notation: @… declares a rule tag. ? prefixes a variable. :- means if. X :: Y means X is a subclass of Y.
\overrides(X,Y) means X is higher priority than Y.

Textual Logic Approach: Overview

- **Logic-based text interpretation & generation, for KA & QA**
  - Map text to logic (“text interpretation”): for K and Q’s
  - Map logic to text (“text generation”): for viewing K, esp. for justifications of answers (A’s)
  - Map based on logic
- **Textual terminology – phrasal style of K**
  - Use words/word-senses directly as logical constants
  - Natural composition: textual phrase ↔ logical term
- **Interactive logical disambiguation technique**
  - Treats: parse, quantifier type/scope, co-reference, word sense
  - Leverages lexical ontology – large-vocabulary, broad-coverage
  - Initial restriction to stand-alone sentences – “straightforward” text
    - Minimize ellipsis, rhetoric, metaphor, etc.
  - Implemented in Automata Linguist™
- **Leverage defeasibility of the logic**
  - For rich logical K: handle exceptions and change
    - Incl. for NLP itself: “The thing about NL is that there’s a gazillion special cases” [Peter Clark]

Query Justification in Rulelog (SILK)



```
◢ ! G + It is not the case that cell52 has a nucleus
    ◢ ! A cell52 is a red blood cell
        ◢ G cell52 is a red blood cell
            ◢ G + cell52 # red(blood(cell))
                    F cell52 # red(blood(cell))
            ▷ G red blood cell
    ▷ ! A cell52 has no nucleus
    ◢ - ↓ A cell52 is a eukaryotic cell
        ◢ G cell52 is a eukaryotic cell
            ◢ G + cell52 # eukaryotic(cell)
                ◢ G + cell52 # red(blood(cell))
                        F cell52 # red(blood(cell))
                ◢ G red(blood(cell)) ## eukaryotic(cell)
                        F red(blood(cell)) ## eukaryotic(cell)
            ▷ G eukaryotic cell
        ◢ G This argument was defeated
            ▷ ! A cell52 has no nucleus
            ◢ ! A cell52 is a red blood cell
                ◢ P r2 has a higher priority than r1
                        F r2 has a higher priority than r1
                ▷ G cell52 is a red blood cell
```

G True literal

G False literal

F Fact

A True rule body (argument) supporting a literal

P Prioritization rule between two rule tags

↓ Refutation: another argument on the other side had a higher priority

! Live argument

+ There are more arguments to see (pro, con, both)