Unit 2 – Rule-based aggregation and integration of Semantic Web data

Axel Polleres

DERI, National University of Ireland, Galway

Reasoning Web Summer School 2008

Unit Outline

1. What's missing? Rules!

2. Common formats for Rule Interchange on the Web – RIF

3. Features of Rule Languages

3.1 RDF Data Import
3.2 RDF Schema Support
3.3 OWL Support
3.4 Modules, Context, and Named Graphs
3.5 Blank Nodes and Function Symbols
3.6 Built-in Predicates and Functions
3.7 Defaults and Negation as Failure
3.8 Unstratified Negation, Disjunction, and Constraints

What more do I need on top of RDF(S) +OWL?

Express and evaluate rules on top of aggregated Web data. Examples:

- RDFS entailment rules, e.g. (rdfs3) from [Hayes, 2004]:

  IF an RDF graph contains triples `P rdfs:range C.` and `S P O.`
  THEN the triple `O rdf:type C` is entailed

- Custom Rules, beyond RDFS' and OWL's expressivity, e.g. to define complex sub-property relations:

  IF `A dc:partOf C.` and `C swrc:editor E .`
  THEN `A ex:editedBy E .`

- Rules beyond Horn rules, e.g. with disjunctive heads, default rules:

  IF `R a ex:Senior.` and *no evidence that* `R a ex:ConflictingReviewer .`
  THEN `R a ex:CandidtateReviewer`.

Unit Outline

RIF: Towards a common rule interchange format

- W3C Rule Interchange Format working group (RIF)[1] established December 2005
- like all W3C WGs: industrial and academic participants
- not only rules for RDF, but also production rules, business rules, policies, etc.
- recent "last call working drafts", 30 July 2008:
  - *RIF Basic Logic Dialect (BLD)* [RIF-BLD, 2008]
  - *RIF RDF and OWL Compatibility* [RIF-RDFOWL, 2008],

We only use RIF's presentation syntax here, more details on RIF, cf. [Boley *et al.*, 2007], as well as the latest RIF drafts.

[1] http://www.w3.org/2005/rules/wg

Some example Rules on top of RDF data 2/2

Before arriving at a common rule language for the Web, several rules languages/systems already out there...

Let's consider the RDFS entailment rule (rdfs3) mentioned before:

> *IF an RDF graph contains triples (P rdfs:range C) and (S P O)*
> THEN the triple `O rdf:type C` is entailed

Can be written as a Horn rule as follows (using the $triple$ predicate notation):

$\forall$ `?S,?P,?O,?C triple(?O,"rdf:type",?C)` $\leftarrow$
`( triple(?P,"rdf:range",?C)` $\wedge$ `triple(?S,?P,?O) )`

Note again: The unary/binary predicate version would go outside first-order:

$\forall$ `?S,?P,?O,?C "rdf:type"(?O,?C)` $\leftarrow$
`( "rdf:range"(?P,?C)` $\wedge$ `?P(?S,?O) )`

Slotted/F-Logic version works as well:

$\forall$ `?S,?P,?O,?C ?O#?C` $\leftarrow$
`( ?P[rdf:range->?C]` $\wedge$ `?S[?P->?O] )`

Let's see how this looks in several existing rules languages for RDF...

Some SW Rules Languages: TRIPLE

TRIPLE:

- M.Sintek, S.Decker, A.Harth, 2002
- Frame syntax, similar to F-Logic
- Special syntax to import RDF, define namespaces, etc.

```
rdf:= 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'.
rdfs:= 'http://www.w3.org/2000/01/rdf-schema#'.
type := rdf:type.
range := rdfs:range.

FORALL O,C O[type->C] <- EXISTS  S,P (S[P->O] AND P[range->C]).
```

Some SW Rules Languages: JENA

JENA:

- HP Labs Bristol
- proprietary syntax
- natively dealing with RDF, rules as add-on part of Jena API.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.

[rdfs3: (?s ?p ?o) (?p rdfs:range ?c) -> (?o rdf:type ?c)]
```

Some SW Rules Languages: N3

N3:

- W3C people, Dan Connolly, Tim Berners-Lee
- syntax extends N-Triples RDF syntax by rules
- natively extension of RDF, implemented in a prototype system (cwm).

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix log: <http://www.w3.org/2000/10/swap/log#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

{ <#p> rdfs:range <#c>. <#s> <#p> <#o> . }
                log:implies { <#o> rdf:type <#c> }.
```

Some SW Rules Systems: F-Logic á la FLORA-2

FLORA-2:

- M. Kifer et al.
- A reference implementation for F-Logic with RDF support
- Additional support for higher-order modeling via HiLog

```
:- iriprefix rdf = 'http://www.w3.org/2000/01/rdf-schema#'.

?O[rdf#type->?C] :- ?S[?P->?O], ?P[rdf#range->?C].
```

Some SW Rules Systems: dlvhex

dlvhex:

- T.Eiter, R. Schindlauer, T.Krennwallner, developed within REWERSE
- SW rules engine on top of the dlv system, stable model semantics
- Prolog-style syntax, special predicates for RDF import, namespaces, etc.

```
#namespace("rdf","http://www.w3.org/1999/02/22-rdf-syntax-ns#")
#namespace("rdfs","http://www.w3.org/2000/01/rdf-schema#")

triple(O,rdf:type,C) :-  triple(P,rdfs:range,C), triple(S,P,O).
triple(S,P,O) :-
       &rdf["http://UrlWithRdfData.example.org/data.rdf"](S,P,O).
```

Some SW "Rules" Languages: SPARQL!

SPARQL:

- W3C query language standard
- As mentioned before, SPARQL's `CONSTRUCT` queries may be viewed as rules as well
- Syntax a bit like merging SQL with N3/Turtle.

```
CONSTRUCT { ?O rdf:type ?C }
WHERE { ?P rdf:range ?C . ?S ?P ?O . }
```

Issues:

- No recursive/fixpoint evaluation in standard engines
- No combination of several CONSTRUCTs in standard engines

Finally: RIF Presentation Syntax

1) Yet another prefix declaration mechanism:

```
Prefix(xs http://www.w3.org/2001/XMLSchema#)
Prefix(rdfs http://www.w3.org/2000/01/rdf-schema#)
Prefix(owl http://www.w3.org/2002/07/owl#)
Prefix(foaf http://xmlns.com/foaf/0.1/)
Prefix(ex http://www.example.org/)
```

2) "ASCII-writable" presentation syntax, borrows from both common logic (Connectives, Quantifiers), F-Logic (Frames), SPARQL/Turtle (variables, data-typed constants):

```
Forall ?S,?P,?O,?C ( ?O#?C :- And ( ?P[rdf:range->?C] ?S[?P->?O] ) )
```

RIF Examples 1/3

The set of **conflicting reviewers**, that is, either persons having the same names as individuals the authors know personally, according to their FOAF files...

```
Forall ?P ?A ?P1 ?N
 ( ?P#ex:ConflictingReviewer :-  And(
      <http://dblp.l3s.de/d2r/page/publications/conf/rweb/EiterIKP08>
         [dc:creator -> ?A]
      ?A[foaf:knows -> ?P1]
      ?P1[foaf:name -> ?N]
      ?P[foaf:name -> ?N]
      ?P#foaf:Person
    )
 )
```

(1)

RIF Examples 2/3

. . . or, persons having the same names as people that, according to DBLP, co-authored papers with the authors of the paper in question.

```
Forall ?P ?A ?Pub ?P1 ?N
 ( ?P#ex:ConflictingReviewer :- And(
     <http://dblp.l3s.de/d2r/page/publications/conf/rweb/EiterIKP08>
         [dc:creator -> ?A]
     ?Pub[dc:creator -> ?A]
     ?Pub[dc:creator -> ?P1]
     ?P1[ foaf:name -> ?N]
     ?P[ foaf:name -> ?N]
     ?P#foaf:Person
   )
 )
```

(2)

RIF Examples 3/3

People with the same names as people who have published in the same conferences or journals as the authors are, **possible reviewers**.

```
Forall ?P ?A ?Pub ?ConfOrJournal ?P1 ?N
 ( ?P#ex:CandidateReviewer :- And(
     <http://dblp.l3s.de/d2r/page/publications/conf/rweb/EiterIKP08>
         [dc:creator -> ?A]
     ?Pub[dc:creator -> ?A]
     ?Pub[dcterms:partOf  -> ?ConfOrJournal]
     ?Pub1[dcterms:partOf -> ?ConfOrJournal]
     ?Pub1[dc:creator -> ?P1]
     ?P1[ foaf:name -> ?N]
     ?P[ foaf:name -> ?N]
     ?P#foaf:Person
   )
 )
```
(3)

Unit Outline

RDF Data Import

Available rules systems provide RDF import facilities, either

- by import directives, external to the rules language, or
- by special built-in predicates as part of the rule language to import RDF graphs.

E.g. RIF has a dedicated import directive [RIF-RDFOWL, 2008]:

```
Import( <http://dblp.l3s.de/d2r/data/authors/Thomas_Eiter> )
```

can be parameterized with semantic "profile", e.g. RDFS...

```
Import( <http://dblp.l3s.de/d2r/data/authors/Thomas_Eiter> rdfs )
```

...see next slide.

RDF Schema Support

Is Thomas Eiter a conflicting reviewer? Not according to rules (1)-(2)! both have

`?P#foaf:Person` $\sim$ `?P rdf:type foaf:Person .`

in the prerequisite, but from DBLP we "only" know:

```
@prefix swrc:  http://swrc.ontoware.org/ontology# .
...
<http://dblp.L3S.de/d2r/resource/authors/Thomas_Eiter> rdf:type foaf:Agent .
<http://dblp.L3S.de/d2r/resource/publications/conf/foiks/2002>
            swrc:editor <http://dblp.L3S.de/d2r/resource/authors/Thomas_Eiter>.
```

refers implicitly – by namespace – to the SWRC ontology, which contains

```
swrc:editor rdfs:range swrc:Person.
swrc:Person rdfs:subClassOf foaf:Person.
```

by importing SWRC + adding RDFS entailment rules:

```
Forall ?S,?P,?O,?C ( ?O#?C :- And ( ?P[rdf:range->?C] ?S[?P->?O] ) )
Forall ?O,?C,?D ( ?O#?D :- And ( ?O#?C ?C[rdfs:subClassOf->?D] ) )
```

we could finally infer:

`<http://dblp.L3S.de/d2r/resource/authors/Thomas_Eiter>#foaf:Person`, i.e

`<http://dblp.L3S.de/d2r/resource/authors/Thomas_Eiter> rdf:type foaf:Person .`

OWL Support 1/3

We were cheating in the previous slide...

```
swrc:editor rdfs:range swrc:Person.
```

is not said in the SWRC ontology,but it is an OWL ontology which has:

$$swrc{:}Proceedings \sqsubseteq \forall swrc{:}editor.swrc{:}Person;$$

instead.

...Never mind! Can – just like RDFS entailment – also be translated to a RIF rule:

```
Forall ?P ?Proc( ?P#swrc:Person :- And( ?Proc#swrc:Proceedings
                                        ?Proc[ swrc:editor -> ?P] ) ).
```

But...

OWL Support 2/3

. . . unlike RDFS, not all of OWL can be translated to rules that easily:

$$ex{:}Publication \equiv ex{:}Paper \sqcap \exists ex{:}publishedIn.\top$$

$\Leftarrow$ easy. . .

```
Forall ?P ( ?P#ex:Publication
            :- And( ?P#ex:Paper Exists ?X( ?P[ ex:pulishedIn -> ?X] ) ) )
```
(4)

$\Rightarrow$ problematic!

```
Forall ?P ( And( ?P#ex:Paper Exists ?X( ?P[ ex:pulishedIn -> ?X] ) )
              :- ?P#ex:Publication )
```
(5)

This rule is not Horn!

Conclusion: Some, but not all of OWL can be translated to Horn rules, the subset of OWL DL within Horn is called OWL DLP [Grosof *et al.*, 2003].

OWL Support 3/3

Some inferences in OWL need full OWL reasoning, e.g. for cardinality, equality reasoning, reasoning y cases, etc.
To this end, some rule engines, instead of trying to translate OWL into rules integrate interfaces to call external DL reasoners, or perform different forms of integration, more details in Units 3+4.

Modules, Context, and Named Graphs

How can we mix **different** RDF graphs in the same ruleset? This is not possible with "global" imports statements...

...but some rule engines allow a sophisticated module mechanism to define context "per Atom":[2]

e.g., rules for extracting $ex{:}Publication$s from different graphs...

```
Forall ?X ( ?X#ex:Publication :-
            ?X#foaf:Document @ <http://dblp.l3s.de/d2r/all/Publications> )
Forall ?X ( ?X#ex:Publication :-
            ?X#foaf:Document @ <http://polleres.net/publications.rdf> )
```

...without importing the other triples of these RDF graphs
...similar to SPARQL's `graph` patterns!

[2] e.g. TRIPLE [Sintek and Decker, 2002], FLORA-2 [Kifer, 2005]

Blank Nodes and Function Symbols

Let's have another look at (5) from above:

```
Forall ?P ( And( ?P#ex:Paper Exists ?X( ?P[ ex:pulishedIn -> ?X] ) )
            :- ?P#ex:Publication )
```

1. And in the head can be split to two rules
2. in rules languages with function symbols we could do the usual "trick": Skolemization! [3]

```
Forall ?P ( ?P#ex:Paper :- ?P#ex:Publication )
Forall ?P ( ?P[ ex:pulishedIn -> sk(?P)] :- ?P#ex:Publication )
```

Similar to blank nodes in the head (construct) in SPARQL...

```
construct { ?P ex:publishedIn _:X } where { ?P rdf:type ex:Publication }
```

...but – especially in combination with other features (recursion, negation) – potentially leads to termination problems (cf. e.g. [Bonatti, 2004])

[3] i.e., replace existentials by new function symbols

Built-in Predicates and Functions

External predicates & functions with fixed semantics: arithmetics, string manipulations, etc.

- Standard list of practical functions and predicates: XQuery/XPath [Malhotra *et al.*, 2007]
- Adopted in RIF's Datatypes and Built-ins List [RIF-DTB, 2008][4]
- Typically, existing rules languages only support subsets of these.

E.g., sub-string matching:

```
Forall ?X ?A ( ?X#ex:Publication :-
                    And( ?X#foaf:Document
                    fn:startsWith(?X,"http://dblp.l3s.de/d2r/") ) )
```

particularly tricky, e.g. external functions in rule heads, string-concatenation:

```
Forall ?X ?G ?S ( ?X[ foaf:name -> fn:con cat(?G," ",?S) ] :-
                  And( ?X [ foaf:givenname-> ?G ] ?X[ foaf:surname-> ?S ] ) )
```

Note: This is beyond what can be done in current SPARQL, cf. [Polleres *et al.*, 2007]

[4] we slightly simplify from RIF's syntax for external predicate calls here

Defaults and Negation as Failure

Example from above:

IF `R a ex:Senior.` and *no evidence that* `R a ex:conflictingReviewer .`
THEN `R a ex:CandidtateReviewer`.

This is typically called "*negation as failure*", or "*default negation*" in rules languages.[5]

```
Forall ?P ( ?P#ex:CandidateReviewer :-
                  And( ?P#ex:Senior Not ( ?P#ex:ConflictingReviewer) ) )
```

Note: Negation as failure is non-monotonic, unlike "classical" negation, in e.g. OWL!

$$\not\propto$$

$$CandidateReviewer \sqsupseteq Senior \sqcap \neg ConflictingReviewer$$

[5] e.g. Prolog, Answer Set Programming, cf. next Units. Note that negation as failure is not in RIF BLD, but e.g. present in the production rules dialect (PRD) [RIF-PRD, 2008].

Unstratified Negation

Similar to the rule from previous slide, rules for "assignment" of reviewers:

```
Forall ?P ( ?P#ex:AvailableReviewer :-
                And( ?P#ex:CandidateReviewer Not ( ?P#ex:AssignedReviewer) ) )
```

Likewise, we could state the other way around:

```
Forall ?P ( ?P#ex:AssignedReviewer :-
                And( ?P#ex:CandidateReviewer Not ( ?P#ex:AvailableReviewer) ) )
```

Recursion over `Not`, also called *unstratified* negation as failure.

Different Rules systems deal differently with unstratified negation, two "standard" ways:

- stable model semantics (now more widely known as *answer set semantics*) [Gelfond and Lifschitz, 1991]
- the well-founded semantics [Van Gelder *et al.*, 1991]

Disjunction

Unstratified Negation often reads unnatural. What we rather wanted to say is something like:

A reviewer is either assigned or available

More natural way to write this would be:

```
Forall ?P ( Or (?P#ex:AssignedReviewer ?P#ex:AvailableReviewer) ) :-
                ?P#ex:CandidateReviewer )
```

E.g. in dlvhex (details next unit) you could write this:

```
triple( P, rdf:type, ex:AssignedReviewer ) v triple( P, rdf:type, ex:AvailableReviewer ) :-
        triple( P, rdf:type, ex:AvailableReviewer )
```

Constraints

Special rules with an empty head, similar to *integrity constraints* in databases. E.g.

Only one reviewer among the candidates can be assigned

```
Forall ?P1 ?P2 (
                :- And( ?P1#ex:AssignedReviewer ?P2#ex:AssignedReviewer
                        ?P1 != ?P2 ) )
```

Here, empty head stands for "contradiction".

Unit 2 Summary

- RIF as a common W3C rule language – work in progress
- Feature summary we'd expect for a reasonable Rule language/system to aggregate Web data:
  - RDF Data Import
  - RDF Schema Support
  - OWL Support
  - Modules, Context, and Named Graphs
  - Blank Nodes and Function Symbols
  - Built-in Predicates and Functions
  - Defaults and Negation as Failure
  - Advanced features: Unstratified Negation, Disjunction, and Constraints


Harold Boley, Michael Kifer, Paula-Lavinia Pătrânjan, and Axel Polleres.
Rule interchange on the web.
In *Reasoning Web 2007*, volume 4636 of *Lecture Notes in Computer Science (LNCS)*, pages 269–309. Springer, September 2007.


Piero A. Bonatti.
Reasoning with infinite stable models.
Artificial Intelligence, 156(1):75–111, 2004.


M. Gelfond and V. Lifschitz.
Classical Negation in Logic Programs and Disjunctive Databases.
New Generation Computing, 9:365–385, 1991.


B. N. Grosof, I. Horrocks, R. Volz, and S. Decker.
Description logic programs: Combining logic programs with description logics.
In *Proceedings WWW-2003*, pages 48–57, 2003.


P. Hayes.
RDF semantics, 2004.
`http://www.w3.org/TR/rdf-mt/`.


Michael Kifer.
Nonmonotonic reasoning in FLORA-2.
In Chitta Baral et al., editors, *LPNMR'05)*, volume 3662 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2005.


Ashok Malhotra, Jim Melton, and Norman Walsh (eds.).
XQuery 1.0 and XPath 2.0 Functions and Operators , January 2007.

W3C Recommendation, available at http://www.w3.org/TR/xpath-functions/.


Axel Polleres, François Scharffe, and Roman Schindlauer.
SPARQL++ for mapping between RDF vocabularies.
In *ODBASE 2007*, volume 4803 of *Lecture Notes in Computer Science* (*LNCS*), pages 878–896, Vilamoura, Algarve, Portugal, November 2007. Springer.


RIF Basic Logic Dialect, July 2008.
W3C Working Draft, available at http://www.w3.org/TR/2007/WD-rif-bld-20071030.


RIF Datatypes and Built-ins 1.0, July 2008.
W3C Working Draft, available at http://www.w3.org/TR/2007/WD-rif-dtb-20080730.


RIF Production Rule Dialect, July 2008.
W3C Working Draft, available at http://www.w3.org/TR/2007/WD-rif-prd-20080730.


RIF RDF and OWL Compatibility, July 2008.
W3C Working Draft, available at
http://www.w3.org/TR/2008/WD-rif-rdf-owl-20080730/.


Michael Sintek and Stefan Decker.
TRIPLE - A Query, Inference, and Transformation Language for the Semantic Web.
In *Proceedings of the First International Semantic Web Conference* (*ISWC 2002*), volume 2342 of *Lecture Notes in Computer Science* (*LNCS*), pages 364–378, 2002.


A. Van Gelder, K. A. Ross, and J. S. Schlipf.
The Well-Founded Semantics for General Logic Programs.
Journal of the ACM, 38(3):620–650, 1991.