

CLARK  PARSIA

Modeling with Rules in Practice

Héctor Pérez-Urbina
Clark & Parsia, LLC
hector@clarkparsia.com



Clark & Parsia

- Specializes in Semantic Web, Web services, and advanced AI technologies for federal and enterprise customers
- Software development and integration services
- Software products for end-user and OEM use

Rules rule!

- Many of our customers prefer rules over axioms
- Sometimes axioms are not sufficiently expressive or expressive in the *right* way

Working with Rules

Why is the syntax so verbose?

```
<swrl:Variable rdf:about="urn:swrl#parent" />
<swrl:Variable rdf:about="urn:swrl#child" />
<swrl:Imp>
  <swrl:head>
    <swrl:AtomList>
      <rdf:rest rdf:resource="&rdf:nil" />
      <rdf:first>
        <swrl:IndividualPropertyAtom>
          <swrl:propertyPredicate rdf:resource="http://www.
example.org#hasFather" />
            <swrl:argument1 rdf:resource="urn:swrl#child"/>
            <swrl:argument2 rdf:resource="urn:swrl#parent"/>
          </swrl:IndividualPropertyAtom>
        </rdf:first>
      </swrl:AtomList>
```

...

CLARK  PARSIA



IF-THEN Syntax

```
PREFIX :<http://www.example.org#>
```

```
IF {  
    ?x :hasParent ?y.  
    ?y rdf:type :Male.  
}  
THEN {  
    ?x :hasFather :?y.  
}
```

Datalog Syntax

```
PREFIX :<http://www.example.org#>
```

```
:hasParent(?child, ?parent), :Male(?parent)
```

```
->
```

```
:hasFather(?child, ?parent).
```

IF-THEN Syntax

```
@prefix rule: <http://rule.stardog.com/> .
```

```
[ ] a rule:SPARQLRule ;  
  rule:content "  
    PREFIX :<http://www.example.org#>  
    IF {  
      ?x :hasParent ?y.  
      ?y rdf:type :Male.  
    }  
    THEN {  
      ?x :hasFather :?y.  
    }  
  " .
```

CLARK  PARSIA



Datalog Syntax

```
@prefix rule: <http://rule.stardog.com/> .
```

```
[ ] a rule:SPARQLRule ;
```

```
  rule:content "
```

```
    PREFIX :<http://www.example.org#>
```

```
    :hasParent(?child, ?parent), :Male(?parent)
```

```
  ->
```

```
    :hasFather(?child, ?parent).
```

```
" .
```

Is Protégé the only option?

The screenshot shows the Stardog web interface for a test ontology. The browser address bar shows the URL `http://stardog.clarkparsia.com/TestKB.owl`. The interface has several tabs: "Active Ontology", "Entities", "Classes", "Object Properties", "Data Properties", "Individuals", and "DL Query". The "Active Ontology" tab is selected, and the "Annotations" section is empty. Below it, the "Rules" section is active, displaying a list of rules with expand/collapse icons on the right. The rules are:

- `TestKB:f1(?<urn:swrl#x>, ?<urn:swrl#y>), TestKB:f3(?<urn:swrl#x>, ?<urn:swrl#z>), swrlb:add(?<urn:swrl#z>, 17.0f, ?<urn:swrl#y>) -> TestKB:f4(?<urn:swrl#x>, ?<urn:swrl#z>)`
- `TestKB:f3(?<urn:swrl#x>, ?<urn:swrl#y>), swrlb:greaterThan(?<urn:swrl#y>, 0.0f) -> TestKB:f7(?<urn:swrl#x>, 35.0f)`
- `TestKB:f1(?<urn:swrl#x>, ?<urn:swrl#y>), swrlb:add(?<urn:swrl#w>, 2.0f, ?<urn:swrl#z>), swrlb:multiply(?<urn:swrl#l>, 4.0f, ?<urn:swrl#w>), swrlb:subtract(?<urn:swrl#z>, 1.0f, ?<urn:swrl#y>) -> TestKB:f2(?<urn:swrl#x>, ?<urn:swrl#l>)`
- `TestKB:f2(?<urn:swrl#x>, ?<urn:swrl#z>), TestKB:f3(?<urn:swrl#x>, ?<urn:swrl#y>), swrlb:lessThan(?<urn:swrl#z>, ?<urn:swrl#y>), swrlb:lessThanOrEqual(?<urn:swrl#z>, ?<urn:swrl#y>) -> TestKB:f6(?<urn:swrl#x>, ?<urn:swrl#z>)`
- `TestKB:f2(?<urn:swrl#x>, ?<urn:swrl#z>), TestKB:f3(?<urn:swrl#x>, ?<urn:swrl#y>), swrlb:greaterThan(?<urn:swrl#y>, ?<urn:swrl#z>) -> TestKB:f5(?<urn:swrl#x>, ?<urn:swrl#y>)`
- `TestKB:f2(?<urn:swrl#x>, ?<urn:swrl#y>), swrlb:add(?<urn:swrl#z>, 1.0f, ?<urn:swrl#y>), swrlb:multiply(?<urn:swrl#w>, -1.0f, ?<urn:swrl#z>) -> TestKB:f3(?<urn:swrl#x>, ?<urn:swrl#w>)`

At the bottom of the interface, there is a checkbox for "Show Inferences" which is checked, and a button for "Reasoner -> Start reasoner".

Understanding Rules

So, I'm not in QL anymore?

- OWL profiles intended for tractability
- Rules can quickly take you out of a profile

```
:hasParent(?child, ?parent), :Male(?parent)  
->  
:hasFather(?child, ?parent).
```

How can I create new individuals?

- Axioms can infer the *existence* of some unnamed individuals

```
:Father rdfs:subClassOf :hasChild some owl:Thing.
```

- Can SWRL help me to create *new* named individuals?

```
:Father(?father)
```

```
->
```

```
:hasChild(?father, ?child).
```

How can I create new individuals?

```
:Father(?father), BIND(UUID() as ?child)
```

->

```
:hasChild(?father, ?child).
```

```
:Father(?father), BIND(UUID() as ?child)
```

->

```
:hasChild(?father, ?child), Child(?child).
```

Reasoning with Rules

When does a rule fire?

```
:hasParent(?child, ?parent), :Male(?parent)  
-> hasFather(?child, ?parent).
```

```
:HectorSr :hasParent :Jose.
```

```
:Jose rdf:type :Male.
```

```
:Hector :hasFather :HectorSr.
```


Materialization

```
:hasParent(?child, ?parent), :Male(?parent)  
-> :hasFather(?child, ?parent).
```

```
:HectorSr :hasParent :Jose.
```

```
:Jose a :Male.
```

```
:Hector :hasFather :HectorSr.
```

```
:HectorSr :hasFather :Jose.
```

Query Answering

```
SELECT ?father WHERE {  
  ?child :hasFather ?father.  
}
```

```
{?father -> :Jose, ?father -> :HectorSr}
```

Query Rewriting

```
SELECT ?father WHERE {  
  ?child :hasFather ?father. }
```

`:hasParent(?child, ?parent), :Male(?parent) -> :hasFather(?child, ?parent).`

```
SELECT ?father WHERE {  
  { ?child :hasFather ?father. }  
  UNION  
  { ?child :hasParent ?father.  
    ?father a :Male. }  
}
```

CLARK Φ PARSIA



Query Answering

```
SELECT ?father WHERE {  
  { ?child :hasFather ?father. }  
  UNION  
  { ?child :hasParent ?father.  
    ?father a :Male. }  
}
```

```
:HectorSr :hasParent :Jose.
```

```
:Jose rdf:type :Male.
```

```
:Hector :hasFather :HectorSr.
```

```
{?father -> :Jose, ?father -> :HectorSr}
```

CLARK Φ PARSIA



Other Issues

SWRL Built-Ins are not enough

- Logarithms
 - `stardog:log`
 - `stardog:ln`
- Trigonometric Functions
 - `stardog:atan`
 - `stardog:asin`
 - ...
- Degrees and Radians
 - `stardog:toDegrees`
 - `stardog:toRadians`

Thank you!