

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.</pre>
                 example.org#hasFather"/>
                          <swrl:argument1 rdf:resource="urn:swrl#child"/>
                          <swrl:argument2 rdf:resource="urn:swrl#parent"/>
                 </swrl:IndividualPropertyAtom>
             </rdf:first>
        </swrl:AtomList>
```

. . .





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/> .

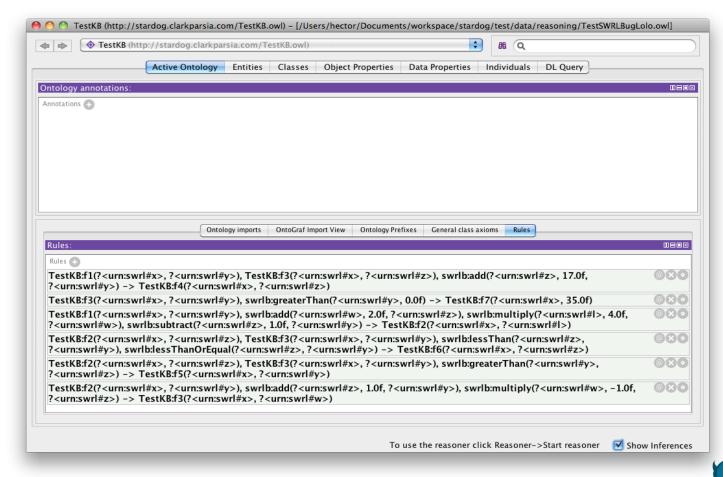
[] a rule:SPARQLRule ;
  rule:content "

        PREFIX :<http://www.example.org#>
        :hasParent(?child, ?parent), :Male(?parent)
        ->
        :hasFather(?child, ?parent).
        " .
```





Is Protégé the only option?





Stardog

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. }
 _ARKФPARSIA
```

Stardog

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 PPARSIA
```



Other Issues





SWRL Built-Ins are not enough

Logarithms

- O stardog:log
- O stardog:ln

Trigonometric Functions

- O stardog:atan
- O stardog:asin
- 0 ...

Degrees and Radians

- O stardog:toDegrees
- O stardog:toRadians





Thank you!



