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>
  </swrl:head>
```

...
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).
@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.Parser
Datalog Syntax

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

[ rule:SPARQLRule ;
  rule:content "
    PREFIX :<http://www.example.org#>
    :hasParent(?child, ?parent), :Male(?parent)
    ->
    :hasFather(?child, ?parent)."
  ]
Is Protégé the only option?
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?


:HectorSr :hasParent :Jose.
:Jose rdf:type :Male.
:Hector :hasFather :HectorSr.
Materialization


: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. }


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

CLARK® PARSIA
Query Answering

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

:HectorSr :hasParent :Jose.
:Jose rdf:type :Male.
:Hector :hasFather :HectorSr.

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