SSSC 2011
Reasoning

Barry Norton [1, formerly 2]
[1] Ontotext AD, BG/UK
[2] AIFB, Karlsruhe Institute of Technology, DE

Berkeley, US; August 9, 2011
RDF Schema

- RDF Schema (RDFS) is the simplest language for two tasks with respect to the RDF data model:
  - **Expectation** – nominate:
    - the ‘types’, i.e., *classes*, of things we might make assertions about, and
    - the *properties* we might apply, as predicates in these assertions, to capture their relationships
  - **Inference** – given a set of assertions, using these classes and properties, specify what should be inferred about assertions that are *implicitly* made

[http://www.w3.org/TR/rdf-schema/](http://www.w3.org/TR/rdf-schema/)
RDF Schema – Predicates and Resources

- RDF Schema introduces
- resources and predicates with (limited) inference:
  - rdfs:Resource
  - rdfs:Literal, rdfs:Datatype, (rdf:XMLLiteral)
  - rdfs:Class, rdfs:subClassOf
  - (rdf:Property), rdfs:subPropertyOf
  - rdfs:range, rdfs:domain
- some predicates with no inference:
  - rdfs:comment
  - rdfs:label
  - rdfs:seeAlso
  - rdfs:isDefinedBy
RDFS Inference

Recall:

- **Schema**
  - vocab:CommercialFlight
  - rdfs:subClassOf
  - vocab:Flight.

- **Existing fact**
  - flights:AI288 rdf:type
  - vocab:CommercialFlight.

- **Inferred fact**
  - flights:AI288 rdf:type
  - vocab:Flight.

- **We expect** to use this vocabulary to make assertions about flights.
- **Having made** such an assertion...
- **Inferences** can be drawn that we did not explicitly make.
This is a result of a set of ‘semantic conditions’ that are applied in the RDFS Semantics:

\[
x \text{ is in } \text{ICEXT}(y) \text{ if and only if } \langle x, y \rangle \text{ is in } \text{IEXT}(l(\text{rdf:type}))
\]

\[
\text{IC} = \text{ICEXT}(l(\text{rdfs:Class}))
\]

\[
\text{IR} = \text{ICEXT}(l(\text{rdfs:Resource}))
\]

\[
\text{LV} = \text{ICEXT}(l(\text{rdfs:Literal}))
\]

\[
\ldots
\]

\[
\text{If } \langle x, y \rangle \text{ is in } \text{IEXT}(l(\text{rdfs:subClassOf})) \text{ then } x \text{ and } y \text{ are in } \text{IC} \text{ and } \text{ICEXT}(x) \text{ is a subset of } \text{ICEXT}(y)
\]

Note: it is not necessary to understand the details, only that the symbols (URIs, etc.) in the model have interpretations (I) and that resource’s interpretations are members of classes (are their extent) and pairs of resource’s interpretations members of predicates interpretations (their extent)

http://www.w3.org/TR/2004/REC-rdf-mt-20040210/
RDFS Inference from Schema

Note, therefore, that the schema in itself leads to inference:

Schema vocab:CommercialFlight
  rdfs:subClassOf
  vocab:Flight.

Inferred vocab:CommercialFlight a rdfs:Class.
  facts vocab:Flight a rdfs:Class.

This is also captured in the set of *axiomatic triples*, including:

  rdfs:subClassOf rdfs:domain rdfs:Class.
  rdfs:subClassOf rdfs:range rdfs:Class.
To see how these apply, consider two further inference rules:

<table>
<thead>
<tr>
<th>If ( &lt;x,y&gt; ) is in ( \text{IEXT}(I(\text{rdfs:domain})) ) and ( &lt;u,v&gt; ) is in ( \text{IEXT}(x) ) then ( u ) is in ( \text{ICEXT}(y) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>If ( &lt;x,y&gt; ) is in ( \text{IEXT}(I(\text{rdfs:range})) ) and ( &lt;u,v&gt; ) is in ( \text{IEXT}(x) ) then ( v ) is in ( \text{ICEXT}(y) )</td>
</tr>
</tbody>
</table>

Recall:

**Schema**

\[ \text{vocab:from } \text{rdfs:range} \ <\text{http://dbpedia.org/ontology/City}>.\]

**Existing**

\[ \text{<http://example.com/flights/AI288>} \]

<table>
<thead>
<tr>
<th>fact</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ \text{vocab:from } \text{<a href="http://dbpedia.org/resource/Vienna%7D">http://dbpedia.org/resource/Vienna}</a>. ]</td>
</tr>
</tbody>
</table>

**Inferred**

\[ \text{<http://dbpedia.org/resource/Vienna>} \]

<table>
<thead>
<tr>
<th>fact</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ a \ &lt;\text{<a href="http://dbpedia.org/ontology/City%7D%3E">http://dbpedia.org/ontology/City}&gt;</a>. ]</td>
</tr>
</tbody>
</table>
Other axiomatic triples (ignoring datatypes and containers) are:

```
rdf:type rdfs:domain rdfs:Resource .
```
```
`
Another way that properties can cause inference is by being related in subproperty hierarchies:

Schema

:wife_of rdfs:subPropertyOf :married_to.
.married_to rdfs:domain :Spouse;
rdfs:range :Spouse.
.wife_of rdfs:domain :Woman;
rdfs:range :Man.

Existing fact

Inferred facts
:anne a :Woman;
a :Spouse;
.married_to :david.
david a :Man; a :Spouse.

Note that there is no problem to be an instance of more than one class.
This does not mean that Woman is a subclass of Spouse or vice versa.
RDFS Inference Limitations

Note that we might wish further inferences, but these are beyond the reasoning power of RDFS and require OWL:

Schema

:wife_of rdfs:subPropertyOf :married_to.
:married_to rdfs:domain :Spouse;
     rdfs:range :Spouse.
:wife_of rdfs:domain :Woman;
     rdfs:range :Man.

Existing fact


Inferred facts

:anne a :Woman;
   a :Spouse;
   :married_to :david.
:married_to :david.
:david a :Man; a :Spouse.

Cannot model with RDFS that x being married to y implies y is married to x

Not :david :married_to :anne.

Cannot model with RDFS that x being wife to y implies y is husband to x
RDFS Lack of Consistency Check

Note furthermore that we might infer what seem like inconsistent facts, but RDFS cannot constrain these:

Schema:
:wife_of rdfs:subPropertyOf :married_to.
:married_to rdfs:domain :Spouse;
   rdfs:range :Spouse.
:wife_of rdfs:domain :Woman;
   rdfs:range :Man.

Existing fact:
:david a :Man.

Inferred facts:
:david a :Woman.

etc.

There is no contradiction here, and this mis-modelling is not automatically diagnosed.
RDFS Summary

Resource Description Framework Schema:

- Allows schemas to be defined for RDF using RDF – on the basis of assertions using specific resources and predicates
- Allows the expectation of the properties to be applied to given classes to be documented
- Allows facts to be inferred from assertions, especially concerning the classification of resources
- Is somewhat limited in terms of the inferences that can be provided
- Does not provide a notion of consistency, or a system of constraints – all assertions and inferences are valid
The Web Ontology Language (OWL) first adds more powerful constructs, allowing further inference over RDF-based models. We shall consider some OWL constructs in the context of Linked Data.

Ontology

<http://www.geonames.org/ontology#parentFeature>
rdf:type owl:TransitiveProperty

Existing facts

owl:sameAs <http://dbpedia.org/resource/Vienna_International_Airport>
<http://sws.geonames.org/2761335/>
owl:sameAs <http://www.geonames.org/ontology#parentFeature>
<http://sws.geonames.org/2770542/>
owl:sameAs <http://dbpedia.org/resource/Austria>

Inferred fact

<http://dbpedia.org/resource/Vienna_International_Airport>
<http://www.geonames.org/ontology#parentFeature>
<http://dbpedia.org/resource/Austria>
OWL Consistency

Unlike RDFS, OWL does not simply infer new triples over RDF models, but also adds a notion of *consistency* and axioms that *constrain* models.

### Ontology

<http://xmlns.com/foaf/0.1/mbox>

rdf:type owl:InverseFunctionalProperty

### Existing facts

owl:differentFrom

sssc11:instructor1\(\rightarrow\) sssc1:instructor2

<http://xmlns.com/foaf/0.1/mbox>

"BarryNorton@GMail.com"   "BarryNorton@GMail.com"

### Inferred

Inconsistency