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

KARLSRUHE SERVICE RESEARCH INSTITUTE (KSRI)
INSTITUTE OF APPLIED INFORMATICS AND FORMAL DESCRIPTION METHODS (AIFB)

OWLIM GROUP
ONTOTEXT AD

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/

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

RDFS Semantics

Karlsruhe Institute of Technology
ontotext

- This is a result of a set of 'semantic conditions' that are applied in the RDFS Semantics:

W3C Recommendation

RDFS semantic conditions.

x is in ICEXT(y) if and only if <x,y> is in IEXT(I(`rdf:type`))

IC = ICEXT(I(`rdfs:Class`))

IR = ICEXT(I(`rdfs:Resource`))

LV = ICEXT(I(`rdfs:Literal`))

...

If <x,y> is in IEXT(I(`rdfs:subClassOf`)) then x and y are in IC and ICEXT(x) is a subset of 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 facts
```
vocab:CommercialFlight a rdfs:Class.
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 .
```

RDFS Inference from Properties

- To see how these apply, consider two further inference rules:

If <x,y> is in IEXT(I(`rdfs:domain`)) and <u,v> is in IEXT(x) then u is in ICEXT(y)
If <x,y> is in IEXT(I(`rdfs:range`)) and <u,v> is in IEXT(x) then v is in ICEXT(y)

- Recall:

Schema
```
vocab:from rdfs:range <http://dbpedia.org/ontology/City>.
```

Existing fact
```
<http://example.com/flights/AI288>
  vocab:from <http://dbpedia.org/resource/Vienna>.
```

⬇

Inferred fact
```
<http://dbpedia.org/resource/Vienna>
  a <http://dbpedia.org/ontology/City>.
```

RDFS Axiomatic Triples

- Other axiomatic triples (ignoring datatypes and containters) are:

```
rdf:type rdfs:domain rdfs:Resource .
rdfs:domain rdfs:domain rdf:Property .
rdfs:range rdfs:domain rdf:Property .
rdfs:subPropertyOf rdfs:domain rdf:Property .
rdfs:subClassOf rdfs:domain rdfs:Class .
rdf:subject rdfs:domain rdf:Statement .
rdf:predicate rdfs:domain rdf:Statement .
rdf:object rdfs:domain rdf:Statement .
rdfs:member rdfs:domain rdfs:Resource .
rdf:first rdfs:domain rdf:List .
rdf:rest rdfs:domain rdf:List .
rdfs:seeAlso rdfs:domain rdfs:Resource .
rdfs:isDefinedBy rdfs:domain rdfs:Resource .
rdfs:comment rdfs:domain rdfs:Resource .
rdfs:label rdfs:domain rdfs:Resource .
rdf:value rdfs:domain rdfs:Resource .
```

```
rdf:type rdfs:range rdfs:Class .
rdfs:domain rdfs:range rdfs:Class .
rdfs:range rdfs:range rdfs:Class .
rdfs:subPropertyOf rdfs:range rdf:Property .
rdfs:subClassOf rdfs:range rdfs:Class .
rdf:subject rdfs:range rdfs:Resource .
rdf:predicate rdfs:range rdfs:Resource .
rdf:object rdfs:range rdfs:Resource .
rdfs:member rdfs:range rdfs:Resource .
rdf:first rdfs:range rdfs:Resource .
rdf:rest rdfs:range rdf:List .
rdfs:seeAlso rdfs:range rdfs:Resource .
rdfs:isDefinedBy rdfs:range rdfs:Resource .
rdfs:comment rdfs:range rdfs:Literal .
rdfs:label rdfs:range rdfs:Literal .
rdf:value rdfs:range rdfs:Resource .
```

RDFS Inference from Subproperties

- 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
```
:anne :wife_of :david.
```

⬇

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

```
:anne :wife_of :david.
```

Inferred facts

```
:anne a :Woman;
      a :Spouse;
      :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 inferred

```
:david :married_to :anne.
:david :husband_of :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.
:david :wife_of :anne.
```

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

OWL Inference

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/>
            <http://www.geonames.org/ontology#parentFeature>
<http://sws.geonames.org/2770542/>
            <http://www.geonames.org/ontology#parentFeature>
<http://sws.geonames.org/2782113/>
                      <http://dbpedia.org/resource/Austria>
        owl:sameAs
```

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 → sssc1:instructor2

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

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

Inferred

Inconsistency