

Assignment 1 (Partial Solutions)

Camilo Thorne

Room 00.012

Institut für Maschinelle Sprachverarbeitung

Universität Stuttgart

+49 (0) 711 685-81369

camilo.thorne@ims.uni-stuttgart.de

Semantic Web, SS 2017

Outline

- 1 KR
- 2 RDF
- 3 RDFS

Outline

- 1 KR
- 2 RDF
- 3 RDFS

Terminological vs Concrete Knowledge

What is the role of terminological knowledge (e.g. *Every company is an organization.*), and what is the role of facts (e.g. *Microsoft is an organization. Microsoft is headquartered in Redmond.*) in a knowledge-based system?

Answer

① Terminological knowledge

- is used to define relevant concepts of a problem domain
- is used to describe general characteristics of entities and interrelations between entities of a specific domain
- allows to abstract from facts and can be used to derive new concrete knowledge

② Concrete knowledge

- states concrete facts
- states explicit information about specific entities

Remark: An **ontology** requires both kinds of knowledge.

Outline

- 1 KR
- 2 RDF
- 3 RDFS

- triples having a *subject*, a *predicate* and an *object*
- Different syntactic notations, e.g. **XML**, **N-Triples** or **Notation3 (N3)**

RDF – N3

```
@prefix model: <http://example.com/model1/> .  
@prefix cdk: <http://example.com/chemistrydevelopmentkit/> .  
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .  
@prefix owl: <http://www.w3.org/2002/07/owl#> .
```

```
model:atom1
```

```
  a      cdk:Atom ;  
  cdk:hasFormalCharge "1" ;  
  cdk:symbol "O" .
```

```
model:atom2
```

```
  a      cdk:Atom ;  
  cdk:symbol "C" .
```

```
model:mol1
```

```
  a      cdk:Molecule ;  
  rdfs:label "Methanol" ;  
  owl:sameAs <http://rdf.openmolecules.net/?InChI=1/CH4O/c1-2/h2H,1H3> ;  
  cdk:hasAtom model:atom2 ,  
              model:atom1 ;  
  cdk:hasBond model:bond1 .
```


RDF Graph

```
@prefix model: <http://example.com/model1/> .
@prefix cdk: <http://example.com/chemistrydevelopmentkit/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl:
<http://www.w3.org/2002/07/owl#> .
```

model:atom1

```
a      cdk:Atom ;
cdk:hasFormalCharge "1" ;
cdk:symbol "O" .
```

model:atom2

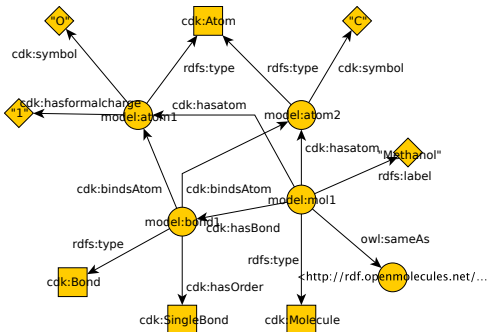
```
a      cdk:Atom ;
cdk:symbol "C" .
```

model:mol1

```
a      cdk:Molecule ;
rdfs:label "Methanol" ;
cdk:hasAtom model:atom2 ,
            model:atom1 ;
cdk:hasBond model:bond1 .
owl:sameAs <http://rdf.
openmolecules.net/
?InChI=1/CH4O/c1-2/h2H,1H3> ;
```

model:bond1

```
a      cdk:Bond ;
cdk:bindsAtom model:atom1 ,
              model:atom2 ;
cdk:hasOrder cdk:SingleBond .
```



What You See Is What You Get

Describe the information you have about *model:mol1* in your own words.

Answer

The methanol molecule consists of 2 atoms, namely C and O, which both are bound by single bonds. Methanol is the same thing as the one described in the resource (URI)

```
<http://rdf.openmolecules.net/?InChI=1/CH4O/c1-2/h2H,1H3>
```

FOAF

One widely known RDF vocabulary is FOAF (“Friend of a Friend”) which is useful for expressing metadata about people, and their interests, relationships and activities. You can find an introduction here <http://www.xml.com/pub/a/2004/02/04/foaf.html> and the complete specification here <http://xmlns.com/foaf/spec>.

RDF (Modeling)

FOAF Classes

Look at the definitions of the following classes: `foaf:Group`, `foaf:Person`, `foaf:Organization`. Mark (e.g. underline) all relevant entities in the given text with regard to the foaf-classes mentioned above.

Metallica is an American heavy metal band from Los Angeles, California. The band's fast tempos, instrumentals, and aggressive musicianship placed them as one of the founding "big four" of thrash metal alongside Slayer, Megadeth, and Anthrax. Metallica formed in 1981 when James Hetfield responded to an advertisement that drummer Lars Ulrich had posted in a local newspaper. As of 2003, the line-up features long-time lead guitarist Kirk Hammett (who joined the band in 1983) and bassist Robert Trujillo (a member since 2003) alongside Hetfield and Ulrich. Previous members of the band are lead guitarist Dave Mustaine (who went on to found the band Megadeth), and bassists Ron McGovney, Cliff Burton and Jason Newsted. The band also had a long collaboration with producer Bob Rock, who produced all of its albums from 1990 to 2003 and served as a temporary bassist between the departure of Newsted and the hiring of Trujillo. The band earned a growing fan-base in the underground music community and critical acclaim with its first four albums, with their third, Master of Puppets (1986), described as one of the most influential and "heavy" thrash metal albums. Metallica achieved substantial commercial success with their eponymous fifth album (also known as The Black Album), which debuted at number one on the Billboard 200.

Source: <https://en.wikipedia.org/wiki/Metallica>

`Foaf:Person`: James Hetfield, Lars Ulrich, Kirk Hammett, Robert Trujillo, Dave Mustaine, Ron McGovney, Cliff Burton, Jason Newsted, Bob Rock

RDF (Modeling)

`Foaf:Person`: James Hetfield, Lars Ulrich, Kirk Hammett, Robert Trujillo, Dave Mustaine, Ron McGovney, Cliff Burton, Jason Newsted, Bob Rock

`Foaf:Group`: Slayer, Megadeth, Anthrax, Metallica, Megadeth

`Foaf:Person`: James Hetfield, Lars Ulrich, Kirk Hammett, Robert Trujillo, Dave Mustaine, Ron McGovney, Cliff Burton, Jason Newsted, Bob Rock

`Foaf:Group`: Slayer, Megadeth, Anthrax, Metallica, Megadeth

`Foaf:Organization`: None

Model information about Metallica

Create an RDF model for the text about Metallica. Only use the classes `foaf:Group`, `foaf:Person` and `foaf:Organization` and the following properties: `foaf:birthday`, `foaf:member`, `foaf:knows`.

RDF (Modeling)

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dbp="http://dbpedia.org/page/"
xmlns:foaf="http://xmlns.com/foaf/0.1/"
xmlns:ex="http://example.org/CustomEntity/">
```

```
<foaf:Person rdf:about=dbp:James_Hetfield>
  <foaf:name>James Hetfield</foaf:name>
  <foaf:birthday>"03-04"</foaf:birthday>
  <foaf:knows rdf:resource=dbp:Lars_Ulrich></foaf:knows>
  <foaf:member rdf:resource=dbp:Metallica></foaf:member>
</foaf:Person>
```

```
<foaf:Person rdf:about=dbp:Lars_Ulrich>
  <foaf:name>Lars Ulrich</foaf:name>
  <!-- etc -->
</foaf:Person>
```

```
</rdf:RDF>
```

Outline

- 1 KR
- 2 RDF
- 3 RDFS

RDFS Modeling

- 1 Define a few classes and a small hierarchy of classes for rock band types in Turtle notation
- 2 Define the relationships between the individuals from the previous exercise and the classes from part 1. of this exercise using Turtle notation.
- 3 Define a domain and a range for each of the properties (e.g., `foaf:knows`) of your RDF model.
- 4 Which RDF statements satisfy these constraints, which violate them? Give examples.

I extend the model from slide 16:

① `ex:HardRockBand rdfs:subClassOf ex:RockBand .`
`ex:RockBand rdf:type rdfs:Class .`
`ex:HardRockBand rdf:type rdfs:Class .`

② `ex:Metallica rdf:type ex:ex:HardRockBand .`

③ `foaf:knows rdfs:domain foaf:Person .`
`foaf:knows rdfs:range foaf:Person .`
`foaf:member rdfs:domain foaf:Person .`
`foaf:member rdfs:range ex:Band .`

④ In this example, all statements in my original model (slide 16) satisfy the constraints listed in 1) and 3) (as does the only new fact listed in 2)), but this will depend ultimately on your own model.

Domain and Range

What is the semantics if you specify more than one domain for a property?

Example:

```
ex:containedInBox rdfs:domain ex:Banana .  
ex:containedInBox rdfs:domain ex:Apple .  
ex:containedInBox rdfs:range ex:FruitBox .
```

Looking Deeper into RDFS – Domain and Range

We reason set-theoretically. Let \mathcal{I} be an RDFS models of the previous triples. Then

$$\text{dom}(\text{ex:containedInBox}^{\mathcal{I}}) \subseteq \text{ex:Banana}^{\mathcal{I}} \cap \text{ex:Apple}^{\mathcal{I}}$$

Basically what this means is that following our model anything that is contained in a box is **both** an apple and ...a banana! In general, we get a always an intersection of the typing declarations, even if they are counterintuitive or non-sensical. The model is still consistent. This is because there is no negation to express that bananas are distinct from apples.

RDFS Cyclic Subclasses

What happens if you define cyclic subclass relations? Is there a situation where this is desired?

```
ex:A rdfs:subClassOf ex:B .  
ex:B rdfs:subClassOf ex:C .  
ex:C rdfs:subClassOf ex:A .
```


Looking Deeper into RDFS – Cyclic Subclasses

In general, we get that all classes in the cycle (`ex:A`, `ex:B`, `ex:C`) are equivalent.

Whether this is desired or not, it really depends in the application. In general, this can be bad, as redundancy will induce RDFS reasoners into potentially tons of redundant unnecessary inferences. But it is a common phenomenon in large scale RDFS ontologies, that are complex and dynamic (change over time), as it might be that we weren't apriori aware when adding a new subclass triple, that a cycle would result.

RDFS however has no construct to express explicitly class identity.

Looking Deeper into RDFS – RDF and Turtle Expressiveness

RDF and Turtle

Decide if the following statements can be modeled with RDF/S. If yes, write down the statement in Turtle notation. If not, write down what is missing from RDF/S.

- 1 Every pear is a fruit.
- 2 If customer X ordered box Y, then box Y is ordered by customer X.
- 3 Every fruit box contains fruit flies.
- 4 A customer is never a fruit.
- 5 A fruit box can have only one order number.
- 6 Everything that is in a fruit box is a fruit.
- 7 If something contains fruit, it is a fruit box.
- 8 No fruit box contains meat.

Looking Deeper into RDFS – RDF and Turtle Expressiveness

- 1 Yes: `ex:Pear rdfs:subClassOf ex:Fruit.`
- 2 No: RDFS cannot express inverse properties (in which we invert the sense of a relationship edge)
- 3 Yes:
`ex:contains rdfs:range ex:Flies.`
`ex:contains rdfs:range ex:Fruit.`
`ex:contains rdfs:domain ex:FruitBox.`
- 4 No: RDFS cannot express that customers and fruits are disjoint (no negation)
- 5 No: RDFS cannot express cardinality constraints
- 6 Yes:
`ex:contains rdfs:domain ex:FruitBox.`
`ex:contains rdfs:range ex:Fruit.`
- 7 Yes:
`ex:contains rdfs:range ex:Fruit.`
`ex:contains rdfs:domain ex:FruitBox.`
- 8 No: RDFS cannot express negation (we'd need to assert that fruit –and flies– are disjoint from meat)