

Semantic Web – Practical Work #1

Thomas Raimbault

Reminder

The Resource Description Framework (**RDF**) is a W3C Recommendation for describing **resources**.

RDF is a **data model**, where resources are described by **triples**, *i.e.* in the form (*subject, predicate, object*). The subject denotes the resource, the predicate denotes a property of the resource, which expresses a relationship between the subject and the object, and the object denotes the property's value. The object is either a resource (that could be described too), or a literal value (as primitive value, like integers, dates, or strings).

Example :

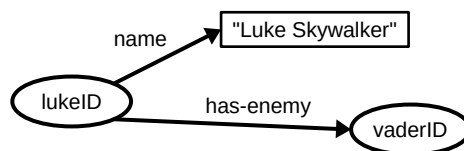
| Subjects | Predicates | Objects |
|----------|------------|------------------|
| lukeID | name | "Luke Skywalker" |
| lukeID | has-enemy | vaderID |

Note that resources have to be identified in a unique way. The W3C recommend to use **URI** (Universal Resource Identifier, ~ URL). For instance, lukeID should be like <http://sample.com/foo/luke> or <http://sample.com/foo#luke>.

A RDF document is a set of triples, also called RDF dataset, or RDF database.

RDF is also a **graph model**. Indeed, a RDF dataset can be seen as a labeled directed multi-graph, where nodes correspond to subjects and objects from triples, and arcs correspond to predicates (directed from subject to object). Generally, resources are represented by ellipses, and literals by rectangles.

The same example as graph:



Apache Jena

[Apache Jena](#) framework (tools and API) will be used throughout the module.

Warning: the Java Development Kit (**JDK**) – including the Java Runtime Environment (JRE) – has to be installed on your computer.

- * Standard Edition (SE) SDK : <https://www.oracle.com/technetwork/java/javase/downloads/index.html>
- * Enterprise Edition (EE) SDK : <https://www.oracle.com/technetwork/java/javasee/downloads/index.html>

Step 1:

Download the latest Apache Jena

- [apache-jena-3.10.0.zip](#)
- <https://jena.apache.org/download/>

Step 2:

Unzip the zip file, if possible with a short file path. (e.g. at "C:\tools\apache-jena-3.10.0\")

Note: you can/should rename the folder, for instance by "jena\" (rather than "apache-jena-3.10.0\")

Step 3:

Open a Console, and print the help of the tool `riot`

- On Windows, go to in the sub-folder `bat\`
 - `> riot.bat --help`
- On Linux and Mac, go to in the sub-folder `bin/`
 - `> ./riot --help`

Remark: firstly, you need to set the `JENA_HOME` variable

- On Windows
 - `> set JENA_HOME=path\to\jena_folder`
 - The Windows' environment variable `PATH` could be modified:
`PATH=...;path\to\jena_folder\bat`
- On Linux and Mac
 - `> export JENA_HOME=path/to/jena_folder`

If help is printed, go to the first exercise.

Exercise 1 RDF document

With a simple text editor, like NotePad++,

- **write in N-Triples format** the information from the bellow paragraph that describes some **facts**:
 - « Star Wars IV's characters are Luke Skywalker, Han Solo, Leia and Darth Vader, who are respectively played by Mark Hamill (who are 67), Harrison Ford, Carrie Fisher and David Prowse. Luke and his sister Leia are friends of Han. Luke's enemy is Vader while Vader is Luke's father. »
- where predicates' IDs are in the form `http://sample/prop/ID`
- where others resources' IDs are in the form `http://sample/ID`
- Save your RDF triples in a file named "`starwars4.nt`".

About N-Triples format (by simplifying):

- One triple per line, terminated by a dot ('.').
- The subject, predicate, and object MUST be followed by a single space.
- IRIs are enclosed in '<' and '>'.

Check syntax with the following command line (content in '[' and ']' is optional), for instance to know if you forget a dot at the end of a triple:

```
> riot.bat --validate [--syntax=NTRIPLES] starwars4.nt
```

Note: RIOT means "Rdf Input/Output Technology".

Exercise 2 Input/Output formatting

RDF is **an abstract model** with several serialization formats (i.e. file or exchange formats). So, XML is not the only format for RDF!

Some serialization formats:

| format | with RIOT | file extension | about |
|-----------|----------------|----------------|---|
| RDF/XML | RDFXML | .rdf | https://www.w3.org/TR/rdf-syntax-grammar/ |
| N-Triples | NTRIPLES ou NT | .nt | https://www.w3.org/TR/n-triples/ |
| Turtle | TURTLE ou TTL | .ttl | https://www.w3.org/TR/turtle/ |
| JSON | RDFJSON | .json | https://www.w3.org/TR/rdf-json/ |

Using RIOT in command line, **convert** your file "`starwars4.nt`" (in N-Triples format)

- to Turtle format (save file in "`starwars4.ttl`")
- to XML format (save file in "`starwars4.rdf`")
- to JSON format (save file in "`starwars4.json`")

Then edit the new created files to check/see the conversions.

Exercise 3 Prefixes

Edit the file "starwars4.ttl"(the one in Turtle format), and define in the first line the following prefixes:

```
@prefix ex: <http://sample.com/> .
@prefix prop: <http://sample.com/prop/> .
```

Then, execute the command line (same output formatting):

```
> riot.bat --output=TURTLE starwars4.ttl
```

What's happened?

What happens

- if `ex` is replaced by `foo`?
- If `ex` is deleted (it is replaced by nothing)? But do not delete the character ':' after `ex` !

Then using RIOT to overwrite your "starwars4.ttl" where IDs are prefixed by the prefixes `ex` and `prop` defined as the base of URIs `http://sample.com/` and `http://sample.com/prop/` (more human readable).

Exercise 4 RDFS

In RDF, a schema – called "RDF Schema" (RDFS) – is a set of classes with certain properties are used to **structure** RDF resources. RDFS elements are also called "RDF vocabularies" to describe some facts.

By simplifying,

- a class is of type `rdfs:Class`
 - example: `sch:Movie rdf:type rdfs:Class .`
- a property is of type `rdf:Property`
 - example: `sch:has-character rdf:type rdf:Property .`
- Classes and properties can respectively be hierarchically organized, as follows
 - example: `sch:Jedi rdfs:subClassOf sch:Creature .`
 - example: `sch:loves rdf:subPropertyOf sch:likes .`
- Domain and range of a property can be specified, as follows
 - example: `sch:has-character rdfs:domain sch:Movie .`
 - example: `sch:has-character rdfs:range sch:Creature .`

Note that all classes and properties do not need to be explicitly declared as `rdfs:Class` or `rdf:Property`. **We talk about inferred information.** Indeed, if you declare for instance that `B rdfs:subClassOf A` it can be deduce that `A` and `B` are two classes. In the same way, if you declare that `ex:SW4 sch:has-character ex:Luke`, then it can be inferred that `ex:SW4` is a `sch:Movie` and `ex:Luke` is a `sch:Creature` (according to the domain and range of the property `sch:has-character`).

Usually, the definitions of classes and properties are stored in a specific RDF file.

So,

- **write in Turtle format** (where prefixes are available) the **schema** that corresponds to:
 - « Star Wars IV is a Movie where there are different kind of creatures, like humans and wookies. Some creatures are Jedis; for instance the human Luke is a Jedi, and Master Yoda – for whom the species is not known – is also a Jedi. The wookie named Chewbacca is the Han's co-pilot on the Millennium Falcon starship. The speed of Millennium Falcon is 1.5 (above the speed of light!) »
 - WARNING: some information here are just facts, then have to be added in the right file (*i.e.* file "starwars4.ttl").
- where classes' IDs are in the form `http://sample/type/ID` (you can define an associated prefix like `type`)
- where properties' IDs are too in the form `http://sample/prop/ID` (the prefix `prop` should still be used)
- the following prefixes should be used
 - `@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .`
 - `@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .`
- Save your schema in a file named "**schema.ttl**".
- Check its syntax with RIOT.

Then, use the following command line to verify that all resources are typed, and that all classes and properties are well defined:

```
> riot.bat --rdfs=schema.ttl starwars4.ttl
```

Exercise 5 Apache Jena : a Java API

Step 1:

Configure you favorite Java IDE.

- This tutorial can help you to start using Jena with Maven:
 - <https://jena.apache.org/download/maven.html>
- Without Maven, see one of these tutorials the see how to use Eclipse:
 - <https://www.iandickinson.me.uk/articles/jena-eclipse-helloworld/>
 - <https://semanticcreatures.com/2015/04/04/getting-started-with-apache-jena-and-eclipse/>

Step 2:

Write a Java code to load RDF facts about Star Wars IV and the associated RDFS schema (if it is more easy, using RIOT you can write facts and the structure in a same file),

then

- add into your code this new information “Han is in love with Leia”
 - Do not forget to add the definition of the property `loves`, which is part of the structure!
- write both facts and the schema into a same RDF/XML file.

To do that, use the official introduction tutorial:

https://jena.apache.org/tutorials/rdf_api.html

Exercise 6 More than a graph!

The interest in manipulating a knowledge graph is that it is possible using rules to infer new knowledge.

Using the documentation at <https://jena.apache.org/documentation/inference/> , write and apply the following rules:

1. the `has-friend` property is symmetric
2. the `has-friend` property is transitive
3. the `has-sister` and `has-brother` are opposite properties
4. an enemy of a friend is an enemy
5. the father of my brother or sister is my father too.

Test your code and check if your defined rules have been well applied.