

# INFO-H509: XML & Web Technologies

## Semantic Web Exercises

Professeur: Stijn Vansummeren

Assistant: Michael Waumans

2014–2015

---

Sur le site web, vous trouverez un ensemble de documents RDF (en RDF/XML et au format Turtle) qui décrivent un catalogue simplifié de cours. Ils sont nécessaires pour réaliser les exercices suivants.

## 1 RDF

### Exercice 1.1

1. Inspectez le contenu du fichier `staff.rdf` (ce fichier est au format RDF/XML). Tracez le graphe RDF correspondant sur papier. **Note:** Vous pouvez vérifier l'exactitude de votre solution en employant l'outil en ligne disponible à l'adresse <http://www.w3.org/RDF/Validator/>: copiez le contenu du fichier et collez le dans la boîte de texte. Veuillez vérifier que l'option *Triples and/or graph* est sélectionnée. Celle-ci se trouve sous *Display Result Options* (Voir figure 1.)
2. Complétez votre graphe en ajoutant les informations à propos du cours *XML Technologies* que vous trouverez dans le fichier `catalogue.rdf`.
3. Complétez ensuite votre graphe en prenant en compte les informations du fichier `infoh509.ttl`.
4. En vous basant sur votre graphe, écrivez les triplets dont vous auriez besoin pour enregistrer les informations du second cours d'XML Technologies dans ce graphe.

### Exercice 1.2

Ouvrez le fichier `catalog.rdf` dans un éditeur de texte, et ajoutez-y un nouveau cours avec le titre 'Object Oriented Programming', dont le mnémonique est INFO-H-200. Ce cours est donné par le professeur Zimanyi et son assistant est appelé Boris Verhaegen.

Ensuite, modifiez le fichier `catalog.rdf` afin d'enregistrer le fait que le cours d'INFO-H-303 'Base de données' est un prérequis pour le cours 'Distributed Information Systems'. (Vous pouvez utiliser le prédicat `ulb:prerequisite` pour cela.)

Aussi, ajoutez le fait que le cours INFO-H-100 est un prérequis au cours 'Object Oriented Programming'. (Vous pouvez à nouveau utiliser le prédicat `ulb:prerequisite` for this.)

### Exercice 1.3

Ecrivez un document, en utilisant la syntaxe Turtle, qui décrive qui vous êtes (dans le même esprit que l'information présente dans le fichier `staff.ttl`). Indiquez dans ce fichier que vous suivez le cours 'XML Technologies'. Tous les nouveaux termes dont vous pourriez avoir besoin d'inventer afin de réaliser cet exercice devront être ajoutés sous [http://www.example.org/\[votre nom\]/](http://www.example.org/[votre nom]/).

Home Documentation Feedback

Check and Visualize your RDF documents

olde servlet

Enter a URI or paste an RDF/XML document into the text field above. A 3-tuple (triple) representation of the corresponding data model as well as an optional graph

Check by Direct Input

<?xml version="1.0"?>  
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dc="http://purl.org/dc/elements/1.1/">  
<rdf:Description rdf:about="http://www.w3.org/">  
<dc:title>World Wide Web Consortium</dc:title>  
</rdf:Description>  
</rdf:RDF>

Copy and paste RDF/XML format here.

Set to "Triples & Graph" to get a drawing of the graph.

Parse RDF Restore the original example Clear the textarea

Display Result Options:  
Triples and/or Graph: Triples and Graph  
Graph format: PNG - embedded

Paste an RDF/XML document into the following text field to have it checked. More options are available in the [Extended interface](#).

Figure 1: The RDF Validator Tool

## 2 RDF Schema

### Exercise 2.1

Le fichier `inference.ttl` contient un schema d'ontologie RDF qui spécifie que les gens travaillant à l'université, soit `ulb:Professor` et `ulb:PHDStudent`, sont membres du personnel académique (`ulb:Faculty`).

1. Les fichiers de données disponibles sur la page du cours contiennent aussi un fichier Java JAR qui nous permettra d'imprimer tous les triplets RDF qui peuvent être inférés depuis un set de documents RDF donnés. Pour montrer l'ensemble des triplets qui peuvent être inférés en interprétant le schema RDF disponible, vous pouvez appeler le programme de la manière suivante :

```
java -jar infertools.jar [--rdfs | --owlfull] <list of files containing rdf>
```

- Par défaut, `-owlfull` est supposé. Dans ce mode, les inférences OWL, incluant les triplets axiomatiques, seront faites. Notez que l'outil utilise Jena ([jena.apache.org](http://jena.apache.org)), une librairie pour faire du raisonnement OWL; l'algorithme fourni par cette boîte à outil n'est pas complet néanmoins, mais, devrait être suffisant pour réaliser ces exercices.
- L'option `-rdfs` provoque la réalisation d'inférence uniquement sur le vocabulaire du schéma RDF. Les triplets axiomatique ne sont pas sortis en output.

Ainsi, depuis le 'Windows Command Prompt' ou un terminal Linux, dans le dossier où vous avez extrait les fichiers de données, vous pourrez utiliser la commande suivant afin de réaliser les inférences sur le schéma RDF sur `staff.rdf` en utilisant les règle situées dans `inference.ttl`.

```
java -jar infertools.jar --rdfs inference.ttl staff.rdf
```

2. Lancez cette commande et inspecter l'output. Notez que tous les professeurs et étudiants doctorants sont maintenant aussi classés comme appartenant à la classe `ulb:Faculty`.
3. Modifiez le fichier `inference.ttl` en ajoutant une règle qui spécifie que les membres du personnel (`ulb:Faculty`) sont des gens (`foaf:Person`).
4. Lancez à nouveau la commande ci-dessus et inspectez le fichier d'output. Vérifiez que votre règle fonctionne comme espéré.

## Exercice 2.2

Pour l'instant, le fichier `catalog.rdf` ne spécifie aucun type pour les cours 'Databases' et 'Introduction to Computer Programming'. Utilisez les termes de schéma RDF `rdfs:range` et `rdfs:domain` pour ajouter des règles au fichier `inference.ttl` qui spécifient que le prérequis d'un cours est lui aussi un cours. Lancez l'outil de l'exercice 2.2 une nouvelle fois pour vérifier que votre ajout est correct.

**Supplementary exercise:** Dans un style similaire, en continuant d'utiliser `rdfs:range` et `rdfs:domain`, ajoutez des règles qui décrivent les propriétés `lecturer` et `assistant` de manière plus détaillée.

## Exercice 2.3

Créez une nouvelle propriété `workHomepage`, et spécifiez que c'est une sous-propriété de `foaf:homepage`. Modifiez le fichier `staff.rdf` pour utiliser cet propriété.

# 3 OWL

**Note** Pour ces exercices, lorsque vous appelez `infortools.jar`, soyez sûr d'employer le paramètre `-owlfull` pour permettre aux inférences OWL d'être réalisées.

## Exercice 3.1

Rappelez vous que OWL DL définit les propriétés caractéristiques suivantes.

1. `owl:TransitiveProperty`
2. `owl:SymmetricProperty`
3. `owl:FunctionalProperty`
4. `owl:InverseFunctionalProperty`

Pour chacune des propriétés suivantes, listez les propriétés caractéristiques qui pourraient s'appliquer.

1. Le prérequis d'un cours (`ulb:prerequisite`)
2. Le numéro de matricule d'un étudiant
3. la date de naissance
4. `owl:sameAs`
5. `owl:inverseOf`

**Supplementary exercise:** Complétez la description de `ulb:prerequisite` dans le fichier `inference.ttl`. Vérifiez l'effet de cette modification en utilisant l'utilitaire `inferencetool.jar`.

## Exercice 3.2

Définissez la propriété `ulb:teaches` comme l'inverse de `ulb:lecturer`. Définissez aussi le domain et la portée de cette propriété.

## Exercice 3.3

Définissez que `staff:fpicalau` est la même personne que `http://my.opera.com/fpicalausa/xml/foaf#me`.

### Exercice 3.4

Utilisez OWL DL pour modéliser les phrases suivantes :

- La classe **Vegetable** est une sous-classe de **PizzaTopping**
- La classe **PizzaTopping** ne partage aucun élément avec la classe **Pizza**.
- L'individu **aubergine** est un élément de la classe **Vegetable**.
- La propriété abstraite est uniquement utilisée pour les relations entre les éléments des classes **Pizza** et **PizzaTopping**.
- La classe **VegPizza** consiste en les éléments des classes **NoMeatPizza** et **NoFishPizza**.
- La propriété **hasTopping** est une sous-propriété de **hasIngredient**.

Ensuite:

- Ajoutez un individu dans les classes **PizzaTopping** et **Pizza**. Lancez ensuite l'outil d'inférence. Que trouvez-vous ? Qu'est ce qui pourrait remédier à cela ?
- Ajoutez un individu aux classes **NoMeatPizza** et **NoFishPizza**. A quoi vous attendez vous ? Lancez l'outil d'inférence. Vérifiez, en utilisant l'outil d'inférence que vous recevez bien le résultat attendu.

### Exercice 3.5

Continuez l'exercice 3.4, utilisez OWL DL afin de modéliser les phrases suivantes.

1. Chaque pizza possède **tomato** comme topping. (garniture)
2. Chaque pizza dans la classe **PizzaMargarita** a exactement **tomato** et **cheese** comme topping.

## 4 SPARQL

**note** Les fichiers de données disponibles sur le site web du cours contiennent aussi un fichier Java JAR qui nous permet d'exécuter une requête SPARQL sur un set de documents RDF donnés en input. Vous pouvez appeler cette méthode de la manière suivante :

```
java -jar sparqltool.jar data-files [query.sparql]
```

- Ici, **data-files** est une liste séparée par des espaces blancs de fichiers rdf (les extensions reconnues sont: ttl, rdf, n3, nt).
- **query.sparql** est un nom de fichier optionnel avec "sparql" comme extension contenu le query à exécuter.
- Si aucun fichier de query n'est spécifié, le query est lu sur l'input standard jusqu'à atteindre un double return (double retour à la ligne).

### Exercice 4.1

Ecrivez les query SPARQL pour les query suivants. (Tous les query doivent être lancés avec les fichier **catalog.rdf**, **staff.rdf**, et **infoh509.ttl**.)

1. Recupérez les URIs de tous les cours.
2. Recupérez, pour chaque cours, sont titre et le nom du professeur.

3. Le nom de tous les professeurs qui donnent un cours tel que celui-ci enseigne aussi le prérequis de ce cours.
4. Le nom de toutes les personnes (`foaf:Person`) et leur page personnelle, si cette information est disponible (i.e., récupérez juste la personne si la page personnelle n'est pas disponible).
5. Le titre de tous les cours qui ont été organisé en *UA4.218*.
6. Toutes les personnes qui connaissent quelqu'un connaissant M.Vansummeren.