SAARLAND UNIVERSITY ARTIFICIAL INTELLIGENCE
Dr. Álvaro Torralba and Prof. Wolfgang Wahlster
Dr. Cosmina Croitoru, Daniel Gnad, Marcel Steinmetz
Yannick Körber, Michael Barz
Christian Bohnenberger, Sophian Guidara, Alexander Rath,
El Khansa Rekik, Julia Wichlacz, Anna Wilhelm

Practical Sheet 5.

Solutions due Thursday, **July 12**, 23:59, in the Moodle system.

In this sheet, your task is to write SPARQL queries and run them on an owl ontology. You can find additional information about SPARQL on [1] and [2]. You can find additional information about Jena on [3]. We provide a Java executable (`*.jar`)[4], which allows you to test your queries. Write your query into a text file and run

```
java -jar sparql.jar -o pizza.owl  -q your_query.sparql
```

To forward the output into a file you can add "`> output`" at the end of the command.

Submission instructions Put all queries, their results, and the answers to the questions into a single PDF or TXT, and add it into an archive called "name1-name2-name3.zip", where "name1", "name2", "name3" are the family names of all authors. Additionally, add a file called "authors.txt" to the archive that contains one line per author, detailing the full name and matriculation number. Please do **not** add any subfolders to your archive. To upload the archive to the Moodle system, go to the corresponding assignment, click "Add submission", and upload it. Note that only one author per group needs to do the submission.

Exercise 7. (20 Points)

The pizza ontology contains information about different pizza types, their toppings and origin. You should get an overview of the ontology by opening it in Protégé (it is installed in the virtual machine). File → Open from URL → and add `http://protege.stanford.edu/ontologies/pizza/pizza.owl`. For each task write down the query and the query result. Use the following prefixes:

[1] `https://jena.apache.org/tutorials/sparql.html`
[2] `https://www.w3.org/TR/rdf-sparql-query/`
[3] `https://jena.apache.org/tutorials/rdf_api.html`
[4] `https://cloud.dfki.de/owncloud/index.php/s/isHx4Ne8QseS5Nj`

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX pizza: <http://www.co-ode.org/ontologies/pizza/pizza.owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
```

1. Write a SPARQL query (and the query result) that lists all countries in the ontology.
2. Write a SPARQL query (and the query result) that lists all meat toppings (see pizza:MeatTopping) and sorts them alphabetically by their names. *Hint: Using an asterisk (rdfs:subClassOf*) will provide all sub-classes.*
3. Write a SPARQL query (and the query result) that lists all "hot" toppings (see pizza:Hot). Hint: You should have a look at the concept called `blank Nodes`[5]
4. Write a SPARQL query (and the query result) that lists all pizzas with cheese topping. Make sure that the result does not contain duplicates.
5. The ontology defines a class CheesyPizza through

```
  Pizza and (hasTopping some CheeseTopping)
  hasBase some PizzaBase
```

   Why doesn't the following query return the same result as the previous one?

```
SELECT ?P
WHERE {
      ?P rdfs:subClassOf* pizza:CheeseyPizza .
}
```

6. Open the pizza ontology in Protégé and start the reasoner (e.g. Reasoner → HermiT, Reasoner → Start reasoner). Compare the result of your query with the sub classes of CheesyPizza in an inferred view (e.g. Entities / Class Hierarchy (inferred)).

[5] `https://jena.apache.org/tutorials/rdf_api.html#glos-blank%20node`