# Querying Description Logics

Petr Křemen

## 1 SPARQL and Ontology Querying

SPARQL Language is aimed at querying RDF(S) documents. As OWL is an extension of RDF(S), we will use it here as a syntax for OWL conjunctive queries. We will use SPARQL queries of the form

```
PREFIX <prefix1>:<URI1>
PREFIX <prefix2>:<URI2>
SELECT <vars>
WHERE {
      <triple1> .
      <triple2> .
      ...
      <tripleN> .
}
```

where `<vars>` is a list of variables (identifier started with a sign "?") delimited with spaces and `<tripleI>` is a triple of the form `subj pred obj`, where each `subj`, `pred` and `obj` is either a variable, or individual URI (also in a shortened form using a PREFIX). Additionally, `obj` can be literal (string in double-quotes). A query example might be

```
PREFIX u: <http://krizik.felk.cvut.cz/university.owl#>

SELECT ?X
WHERE {
  ?X a u:Professor .
  ?X u:worksFor u:CVUT .
  ?X u:name _:Y .
}
```

This SPARQL query expresses a conjunctive query (without full URIs) $Q(?X) \leftarrow Professor(?X), worksFor(?X, CVUT), name(?X, ?Y)$.

## 2 Conjunctive Queries Practically

1. Download the Pellet system from the seminar web pages.

2. In `pellet.bat` setup proxy:

   - delete the line
     `if "%pellet_java_args%"=="" set pellet_java_args=-Xmx512m`
   - in the last row replace `%pellet_java_args%` with the string
     `-Xmx512m -Dhttp.proxyHost=proxy.felk.cvut.cz -Dhttp.proxyPort=80`

3. Download the wine ontology from `http://www.w3.org/TR/owl-guide/wine.rdf` and save it into the Pellet home directory.

4. Using Protege, insert a new instance of `Wine` into the ontology.

5. Download an example query from the seminar web pages. This query finds all regions (instancec of `Region`) :

   ```
   pellet.bat query -q <file-with-query> <file-with-ontology>
   ```

6. In the file with a query, replace the distinguished variable `?Y` with an undistinguished variable `_:Y` and compare results (use the `-bnode` switch for Pellet)

7. Check that you got the same result as in the DL-query tab (How to formulate such query ?).

8. Formulate and evaluate a query that

   - finds all regions in USA together with dry wines produced in these regions.
   - finds all regions in USA that produce both dry and sweet wines.

9. Insert a new type `locatedIn some Region` to the individual `ItalianRegion`. Then, formulate a query that finds all wines that are produced in some (arbitrary) super-region of Italy (i.e. region in which `ItalianRegion` is located in (`locatedIn`)). Use the `--bnode` parameter in the Pellet command line to correctly evaluate the query.