Querying Description Logics

Petr Křemen

1 SPARQL and Ontology Querying

1.1 SPARQL Query Structure

SPARQL Language [SS13] is aimed at querying RDF(S) [GB04] documents. As OWL 2 [MPSP09] is an extension of RDF(S), we will use it here as a syntax for OWL 2 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).

1.2 Expressing conjunctive queries in SPARQL syntax

Example 1.1. Consider a conjunctive query (without full URIs)

$$Q_1(?X, ?Y) \leftarrow Professor(?X), worksFor(?X, CVUT), name(?X, ?Y). \quad (1)$$

Its SPARQL counterpart is :

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

SELECT ?X ?Y
WHERE {
```

```
  ?X a u:Professor .
  ?X u:worksFor u:CVUT .
  ?X u:name ?Y .
}
```

If we are not interested in the binding of the particular name Y, we would pose a conjunctive query

$$Q_2(?X) \leftarrow Professor(?X), worksFor(?X, CVUT), name(?X, ?Y) \tag{2}$$

To reflect this change in SPARQL, we could simply remove the variable from the SELECT clause, obtaining:

```
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 .
}
```

However, this SPARQL query doesn't return individuals (bindings for $?X$) for which there is no name known, but their existence is inferred (e.g. i in axiom $(\exists name \cdot \top)(i)$). To capture all cases, we need to use a *bnode* instead of the (distinguished) variable:

```
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 .
}
```

1.3 Avoiding complex concepts in SPARQL

To avoid rather complex RDF/XML syntax for complex OWL class descriptions in SPARQL queries, we can define a new concept in the description logic theory and query the new concept instead.

Example 1.2. To express a conjunctive query, e.g.

$$Q_2'(?X) \quad \leftarrow \quad (Professor \sqcap \exists name \cdot \top)(?X), worksFor(?X, CVUT),$$

we will define a new concept $ProfessorWithName$ in the description logic theory (OWL Class in Protege):

$$ProfessorWithName \equiv Professor \sqcap \exists name \cdot \top$$

and reformulate the query into

$$Q_2''(?X) \leftarrow ProfessorWithName(?X), worksFor(?X, CVUT).$$

Queries Q_2' and Q_2'' are equivalent.

For some more interesting SPARQL examples, refer to the last section of this document.

2 Conjunctive Queries Practically

1. Download Pellet 2.3.1 from the course web pages.
2. Pellet is a command-line tool. Use `./pellet.sh help` (or `pellet.bat help`) command to find out details about its usage.
3. Download the wine ontology from `http://www.w3.org/TR/owl-guide/wine.rdf` and save it into the Pellet home directory.
4. Open the wine ontology in Protege and insert a new instance of `Wine` into the ontology.
5. Download an example query from the seminar web pages. This query finds all wines (instances of `Wine`) :

   ```
   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 your result from the previous point to the 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.

3 (OPTIONAL) SPARQL Queries on DBPedia

DBPedia is imprecise, yet interesting data source. Use the `http://dbpedia-live.openlinksw.com/sparql/` endpoint and pose the following queries.

Ex. 1 — Find all types (classes) of a Gray wolf in DBPedia.

Answer (Ex. 1) —

```
SELECT DISTINCT ?o
WHERE
{
  <http://dbpedia.org/resource/Gray_wolf> a ?o
}
```

Ex. 2 — How many world countries have more than one official language ?

Answer (Ex. 2) —

```
SELECT count(?country) {
  SELECT DISTINCT ?country, COUNT(?language)
  WHERE
  {
    ?country dbpedia-owl:officialLanguage ?language .
  }
  GROUP BY ?country
  HAVING COUNT(?language) > 1
}
```

Ex. 3 — Find five countries with the highest population.

Answer (Ex. 3) —

```
SELECT DISTINCT ?country ?cnt
WHERE
{
  ?country a dbpedia-owl:Country .
  ?country dbpprop:populationCensus ?cnt
}
ORDER BY DESC(?cnt)
LIMIT 5
```

Ex. 4 — Find all female ancestors of the Prince Charles.

Answer (Ex. 4) —

```
SELECT * WHERE {
```

```
    <http://dbpedia.org/resource/Charles,_Prince_of_Wales>
        (<http://dbpedia.org/property/mother>)+ ?s.
}
LIMIT 10
```

Reference

[GB04] Ramanathan V. Guha and Dan Brickley. RDF Vocabulary Description Language 1.0: RDF Schema [online]. W3C Recommendation, W3C, 2004. Available at `http://www.w3.org/TR/2004/REC-rdf-schema-20040210`, cit. 11/1/2012.

[MPSP09] Boris Motik, Peter F. Patel-Schneider, and Bijan Parsia. OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax [online]. W3C Recommendation, W3C, 2009. Available at `http://www.w3.org/TR/2009/REC-owl2-syntax-20091027`, cit. 11/1/2012.

[SS13] Andy Seaborne and Harris Steve. SPARQL 1.1 query language [online]. W3C Recommendation, W3C, 2013. Available at `http://www.w3.org/TR/sparql11-query`, cit. 1/4/2013.