COMP318

Ontologies and Semantic Web

SPARQL - Part 1

Dr Valentina Tamma

V.Tamma@liverpool.ac.uk

Where were we

- RDF data model & RDFS schema language
  - Vocabulary and model
- Entailment in RDF & RDFS
  - Simple entailment
  - RDFS entailment

SPARQL in general

- SPARQL Protocol and RDF Query Language
- SPARQL Query Language for RDF
  - Declarative
  - Based on the RDF data model (triples/graph)
- SPARQL Query Results XML Format
  - Representation of the results of SPARQL queries
- SPARQL Protocol for RDF
  - Transmission of SPARQL queries and the results
  - SPARQL endpoint: Web service that implements the protocol

SPARQL

- SPARQL is the query language for querying RDF. It allows users to:
  - Pull values from ***structured*** and ***semi-structured*** data
  - Explore data by querying ***unknown relationships***
  - Perform ***complex joins*** of ***disparate databases*** in a single, simple query
  - ***Transform RDF data*** from one vocabulary to another

Assumptions

- Data is represented in RDF
  - Resources are represented by URIs
    - possibly abbreviated as ***prefixed names***
  - Objects can be ***literals:*** strings, integers, booleans...
- We use Turtle syntax:
  - URIs:
    - `<http://www.sw-example.com/resource>` or `prefix:name`
  - literals:
    - `“plain string” “14.1”^^xsd:float` or `”string with language”@en`
  - triples:
    - `pref:subject anOtherPref:predicate “object”`

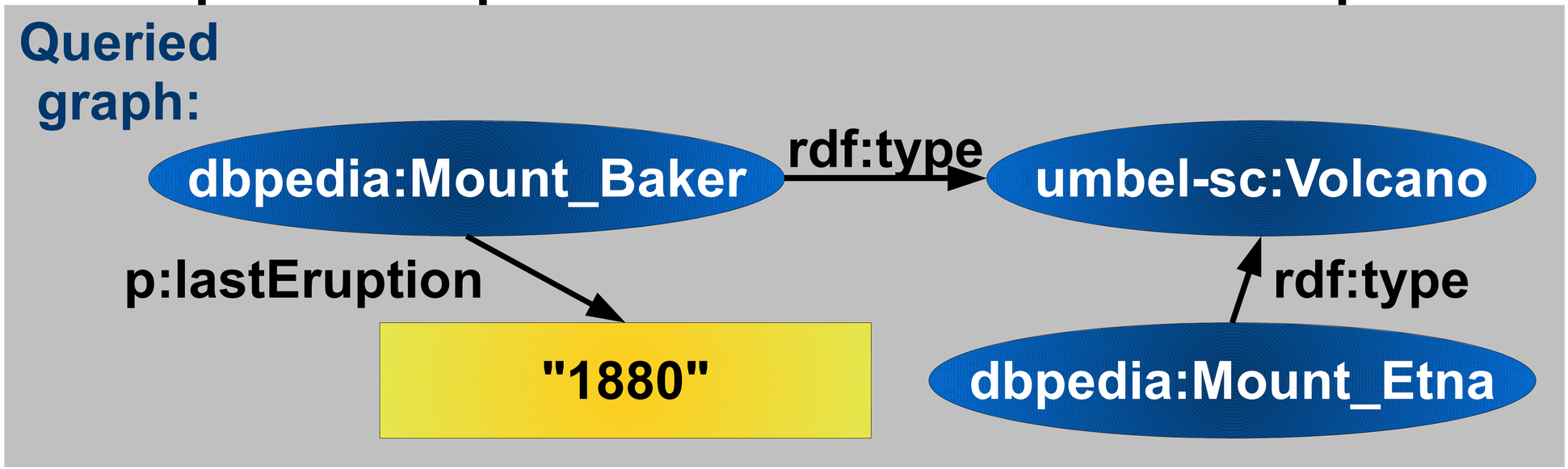
subject - predicate - object
subject - predicate - object
subject - predicate - object
subject - predicate - object
subject - predicate - object
subject - predicate - object
subject - predicate - object
subject - predicate - object
subject - predicate - object
subject - predicate - object
subject - predicate - object

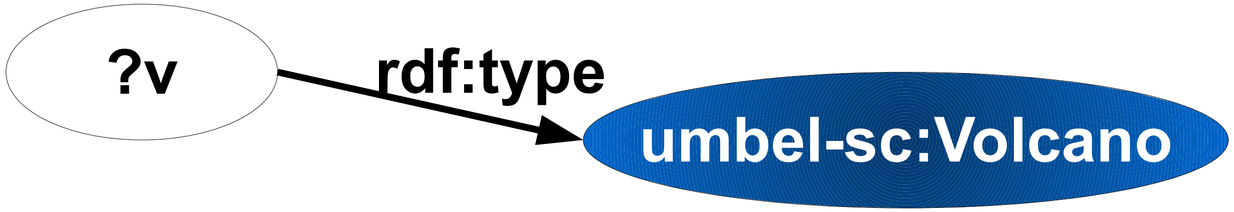
SPARQL versions

- SPARQL 1.0 (2008) included:
  - SPARQL 1.0 Query Language
  - SPARQL 1.0 Protocol
  - SPARQL Results XML Format

- SPARQL 1.1 (2013) includes:
  - SPARQL 1.1 versions of SPARQL Query and SPARQL Protocol
  - SPARQL 1.1 Update
  - SPARQL 1.1 Graph Store HTTP Protocol
  - SPARQL 1.1 Service Description
  - SPARQL 1.1 Entailments
  - SPARQL 1.1 Basic Federated Query

Main principle of SPARQL queries



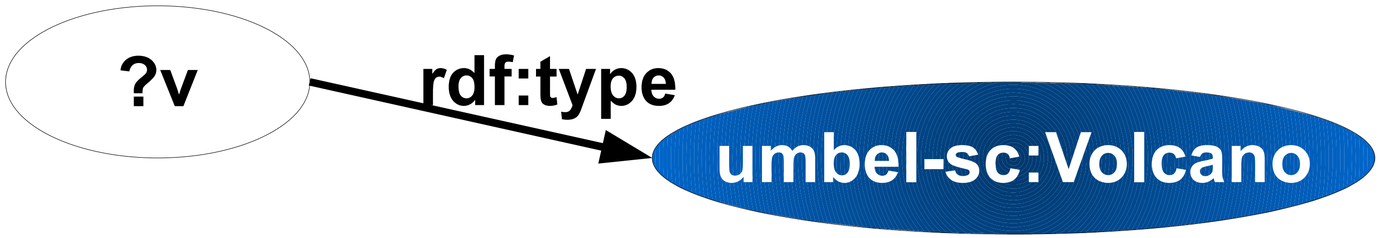


Results:

?v
dbpedia:Mount_Baker
dbpedia:Mount_Etna

Main principle of SPARQL queries

- SPARQL based on the principle of pattern matching
  - Describe subgraphs of the queried RDF graph
  - Subgraphs that match your description yield a result
    - i.e. graph patterns (i.e. RDF graphs with variables)



SPARQL queries

- PREFIX
  - Prefix mechanism for abbreviating URIs
    - **`PREFIX foo: <http://example.com/resources/>`**
- Query Pattern
  - Result clause
    - Identifies the variables to be returned in the query answer
    - **`SELECT ...`**
- DATASET DEFINITION
  - Name(s) of the graph(s) to be queried
    - **`FROM ...`**
- QUERY PATTERN
  - Specifying what to query for in the dataset
    - Query pattern as a list of triple patterns

```
WHERE {
...
}
```

- QUERY MODIFIERS
  - **`ORDER BY ...`**

Query execution

- SPARQL queries are executed against RDF datasets (made by RDF graphs).

- A SPARQL ***endpoint*** accepts queries and returns results via HTTP.
  - endpoint is the entry point to a service, a process, or a queue or topic destination
  - ***Generic*** endpoints will query any Web-accessible RDF data
  - ***Specific*** endpoints are hardwired to query against particular datasets

SPARQL Result set

- A 'binding' is a pair (variable, RDF term).
  - In this result set, there are three variables: x, y and z
  - Each solution is shown as one row in the body of the table.
- Single solution
  - `x` is bound to "Alice",
  - `y` is bound to `<http://example/a>`,
  - `z` is not bound to any RDF term.

```
 x             y                z
=========================================


"Alice"    http://example/a
```

URI abbreviation: PREFIX

- Mechanism for namespace abbreviation
- Syntax:

```
PREFIX abbr: <URI>
```

- Example:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

- Default:

```
PREFIX : <URI>
```

- Example:

```
PREFIX : <http://example.org/myOntology#>
```

URI abbreviation: PREFIX

- Prefix definitions enable CURIEs in the query
  - CURIE: defines a generic, abbreviated syntax for expressing Uniform Resource Identifiers (URIs).
  - Abbreviated URI expressed in a compact syntax
    - may be found in both XML and non-XML grammars.
  - A CURIE may be considered a datatype.

```
<! RDF graph prologue —>

PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX : <http://example.org/myOntology#>
```

Query Result Forms

- `SELECT`: Projection of query result
- `CONSTRUCT`: Returning RDF Graph
- `DESCRIBE`: Returning descriptions of RDF resource
  - not treated here
- `ASK`: “yes/no” query

Query Result Forms

- `SELECT`: Projection of query result
- `CONSTRUCT`: Returning RDF Graph
- `DESCRIBE`: Returning descriptions of RDF resource
  - not treated here
- `ASK`: “yes/no” query

COMP318

Ontologies and Semantic Web

End of SPARQL - Part 1

Dr Valentina Tamma

`V.Tamma@liverpool.ac.uk`