CEID

MSc on DATA DRIVEN COMPUTING AND DECISION MAKING (DDCDM)

SPARQL

I. Hatzilygeroudis

SPARQL Queries

- Querying an RDF document using XML-based query languages (eg XPath) is problematic, due to many variations of description representation.
- SPARQL is used as the standard under adoption by the W3C.

SPARQL Queries

- Basic queries
  - They are based on matching graph patterns
  - Simplest graph pattern: triple pattern
  - It looks like an RDF triple, but variables are allowed in the subject, predicate, or object positions
  - The combination of triple patterns produces a basic graph pattern
  - An exact match to some graph is required

SPARQL Queries

- Basic queries (cont.)
  - Simple example:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdf: <http://www.w3.org/2000/01/22-rdf-schema#>
SELECT ?c
WHERE
  {
      ?c rdf:type rdfs:Class .
  }
```

(the patterns of triples, where rdf:type is the property and rdfs:Class is the object: retrieval of all classes)

SPARQL Queries

- Basic queries (cont.)
  - Another example:

```
PREFIX uni: <http://www.mydomain.org/uni-ns#>
SELECT ?i
WHERE
  {
      ?i rdf:type rdfs:course .
  }
```

(retrieve all instances of the ‘course’ class)

SPARQL Queries

- Structure select-from-where
  - SELECT: specifies the number and order of data to retrieve
  - FROM: identify the source of the data (optional)
  - WHERE: imposes restrictions on possible responses

  E.g.

```
SELECT ?x ?y
WHERE
{
    ?x uni:phone ?y .
}
```

  (recovery of all staff members' phones)

SPARQL Queries

```
SELECT ?x ?y
WHERE
{
    ?x rdf:type uni:Lecturer ;
        uni:phone ?y .
}
```

```
SELECT ?x ?y
WHERE
{
    ?x rdf:type uni:Lecturer .
    ?x uni:phone ?y .
}
```

(recovery of all teachers and their phones)

1. The term ?x rdf:type uni:Lecturer gathers all instances of the Lecturer class and binds the result to the variable ?x
2. uni:phone ?y gathers all triples with predicate phone
3. The implicit join (because of the "?") limits these triads to those with a common subject with the first ones (?x)

SPARQL Queries

```
SELECT ?n
WHERE
{
    ?x rdf:type uni:Course ;
        uni:isTaughtBy  :949352 .
     ?c uni:name ?n .
     FILTER (?c = ?x) .
}
```

(retrieve all courses taught by the teacher with code (ID) 949352)

FILTER is used to indicate a logical restriction. Here the restriction is the direct join (explicit join) of the variables ?c and ?x (use of equality operator "=").

SPARQL Queries

- Optional patterns
  - Until now a response is returned if there is a complete pattern match in the knowledge base
  - But more flexibility is often required

E.g.

```
<uni:lecturer rdf:about="949352">
  <uni:name>Grigoris Antoniou</uni:name>
</uni:lecturer>

<uni:lecturer rdf:about="949318">
  <uni:name>John Hatzis</uni:name>
  <uni:email>ihatz@cti.gr</uni:email>
</uni:lecturer>
```

It doesn't return the other lecturer's name because he doesn't have an email.

```
SELECT ?name ?email
WHERE
{
    ?x rdf:type uni:Lecturer ;
       uni:name  ?name ;
       uni:email ?email .
}
```

?name	?email
John Hatzis	ihatz@cti.gr

SPARQL Queries

```
<uni:lecturer rdf:about="949352">
  <uni:name>Grigoris Antoniou</uni:name>
</uni:lecturer>

<uni:lecturer rdf:about="949318">
  <uni:name>John Hatzis</uni:name>
  <uni:email>ihatz@cti.gr</uni:email>
</uni:lecturer>
```

```
SELECT ?name ?email
WHERE
{
    ?x rdf:type uni:Lecturer ;
       uni:name  ?name ;
    OPTIONAL  {?x uni:email ?email }
}
```

?name	?email
Grigoris Antoniou	
John Hatzis	ihatz@cti.gr