SPARQL (2): An RDF Query Language

Camilo Thorne

Room 00.012
Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
+49 (0) 711 685-81369
camilo.thorne@ims.uni-stuttgart.de

Semantic Web, SS 2017

Institut für Maschinelle Sprachverarbeitung

INFO: Assignment 1

- Deadline extension: Friday 9.6.17, 23h00
- I've restructured the ILIAS repo

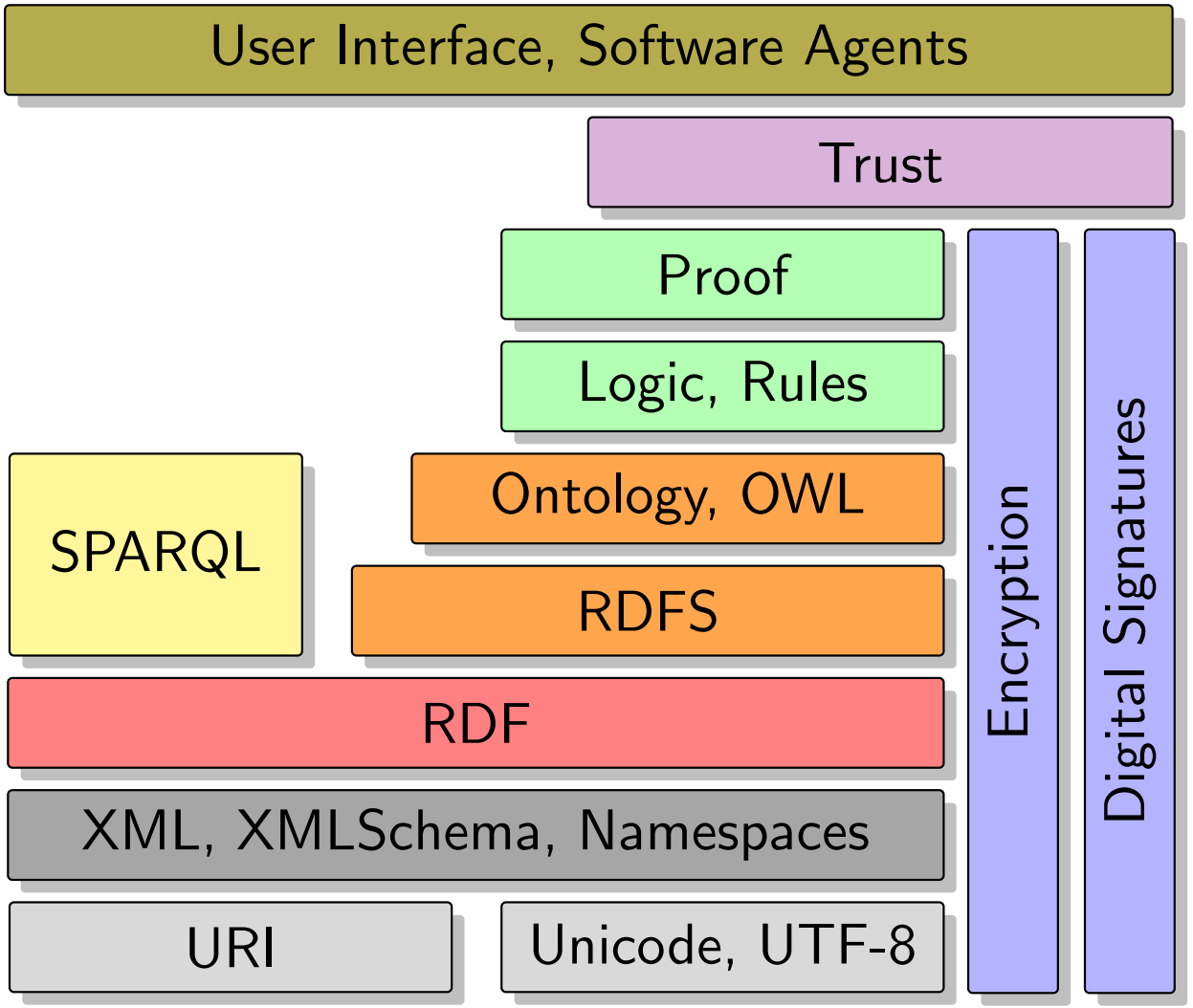
Institut für Maschinelle Sprachverarbeitung

INFO: Assignment 1

- Deadline extension: Friday 9.6.17, 23h00
- I've restructured the ILIAS repo
- Added:
  1. Scans of the examples done on the board (see Lectures → Board Examples)
  2. A bunch of tutorials on: set theory and relations, KR, graphs and namespaces (see Tutorials and Cheat Sheets → Tutorials)
  3. A bunch of cheatsheets on semantic web standards (see Tutorials and Cheat Sheets → Cheat Sheets)
- Removed: advanced literature

Institut für Maschinelle Sprachverarbeitung

The Semantic Web Stack [W3C, Tim Berners-Lee]



Institut für Maschinelle Sprachverarbeitung

Outline

Institut für Maschinelle Sprachverarbeitung

Outline

Institut für Maschinelle Sprachverarbeitung

SPARQL: Advanced Syntax

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?book ?author ?genre
WHERE {
   ?book dbo:author ?author .
   ?book dbo:author dbr:J._R._R._Tolkien .
   ?author rdf:type ?personclass .
   ?book dbo:literaryGenre ?genre .
}
```

This DBpedia SPARQL query returns a list of answers, but does it return all what Dbpedia knows about Tolkien?

Institut für Maschinelle Sprachverarbeitung

OPTIONAL

- One common problem in RDFS ontologies is that some property objects may consist of empty (unknown) values "" (empty strings)
- Body joins/merges among triples are only defined over **non-empty values**, triples with empty values are ignored
- But, we would like to obtain at least partial answers, if some of the information is there
- We can "force" merges/joins to be run over empty values by enclosing the problematic pattern(s) with `OPTIONAL{...}`
- `OPTIONAL` creates a dummy value (a so-called **null**) for each empty value, thus allowing to compute a non-empty join or merge

Institut für Maschinelle Sprachverarbeitung

OPTIONAL

- Our query lists only books that have a genre specified with `dbo:literaryGenre`, books without genre are not listed:

```
?book dbo:author ?author .
?book dbo:literaryGenre ?genre .
```

Institut für Maschinelle Sprachverarbeitung

OPTIONAL

- Our query lists only books that have a genre specified with `dbo:literaryGenre`, books without genre are not listed:

```
?book dbo:author ?author .
?book dbo:literaryGenre ?genre .
```

- To list all books and then, if they have one, also their genre, we can use `OPTIONAL`

Institut für Maschinelle Sprachverarbeitung

OPTIONAL

- Our query lists only books that have a genre specified with `dbo:literaryGenre`, books without genre are not listed:

```
?book dbo:author ?author .
?book dbo:literaryGenre ?genre .
```

- To list all books and then, if they have one, also their genre, we can use `OPTIONAL`

- Several statement occur within `OPTIONAL`, they are interpreted as a join or merge
  - within optional, the join or merge will behave as usual
  - if there are empty values, the result of the **nested triples** will be empty
  - but the **complete query** will still return a (partial) answer

Institut für Maschinelle Sprachverarbeitung

OPTIONAL Example (1)

Query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?book ?author ?genre
WHERE {
   ?book dbo:author ?author .
   ?book dbo:author dbr:J._R._R._Tolkien .
   ?author rdf:type ?personclass .
   OPTIONAL{?book dbo:literaryGenre ?genre .}
}
```

Result:

?book	?author	?genre
...		
dbr:The_Hobbit	dbr:J._R._R._Tolkien	dbr:Children's_literature
dbr:The_Hobbit	dbr:J._R._R._Tolkien	dbr:Fantasy
dbr:The_Adventures_of_Tom_Bombadil	dbr:J._R._R._Tolkien	
dbr:The_Quest_of_Erebor	dbr:J._R._R._Tolkien	dbr:Fantasy
dbr:Songs_for_the_Philologists	dbr:J._R._R._Tolkien	
...		

Institut für Maschinelle Sprachverarbeitung

OPTIONAL Example (2)

Query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?book ?author ?genre
WHERE {
   ?book dbo:author ?author .
   ?author rdf:type ?personclass .
   OPTIONAL{?book dbo:literaryGenre ?genre .}
   OPTIONAL{?book dbp:releaseDate ?releasedate .}
}
```

Result:

?book	?author	?genre	?releasedate
...			
dbr:The_Hobbit	dbr:J._R._R._Tolkien	dbr:Children's_literature	1937-09-21
dbr:The_Hobbit	dbr:J._R._R._Tolkien	dbr:Fantasy	1937-09-21
dbr:The_Adventures_of_Tom_Bombadil	dbr:J._R._R._Tolkien		
dbr:The_Quest_of_Erebor	dbr:J._R._R._Tolkien	dbr:Fantasy	
dbr:Songs_for_the_Philologists	dbr:J._R._R._Tolkien		1936
...			

Institut für Maschinelle Sprachverarbeitung

OPTIONAL Example (3)

Query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?book ?author ?genre
WHERE {
   ?book dbo:author ?author .
   ?author rdf:type ?personclass .
   OPTIONAL{ ?book dbo:literaryGenre ?genre .
                   ?book dbp:releaseDate ?releasedate . }
}
```

Result:

?book	?author	?genre	?releasedate
...			
dbr:The_Hobbit	dbr:J._R._R._Tolkien	dbr:Children's_literature	1937-09-21
dbr:The_Hobbit	dbr:J._R._R._Tolkien	dbr:Fantasy	1937-09-21
dbr:The_Adventures_of_Tom_Bombadil	dbr:J._R._R._Tolkien		
dbr:The_Quest_of_Erebor	dbr:J._R._R._Tolkien		
dbr:Songs_for_the_Philologists	dbr:J._R._R._Tolkien		
...			

Institut für Maschinelle Sprachverarbeitung

Quiz: OPTIONAL

Some mountains have an alternative name (dbp:altname), some have a first person to ascent (dbo:firstAscentPerson), breaking

```
SELECT ?name ?altname ?person
WHERE {
       ?mountain rdf:type dbo:Mountain .
       ?mountain dbp:name ?name .
       ?mountain dbp:altname ?altname .
       ?mountain dbo:firstAscentPerson ?person .
  }
```

What happens if we replace the last two triple patterns by

A)
```
OPTIONAL { ?mountain dbp:altname ?altname . }
OPTIONAL { ?mountain dbo:firstAscentPerson ?person . }
```

B)
```
OPTIONAL { ?mountain dbp:altname ?altname .
           ?mountain dbo:firstAscentPerson ?person . }
```

C)
```
?mountain dbp:altname ?altname .
OPTIONAL { ?mountain dbo:firstAscentPerson ?person . }
```

Institut für Maschinelle Sprachverarbeitung

Outline

Institut für Maschinelle Sprachverarbeitung

FILTER

- Sometimes it is useful to restrict the values of a variable via some **Boolean condition**

Institut für Maschinelle Sprachverarbeitung

FILTER

- Sometimes it is useful to restrict the values of a variable via some **Boolean condition**

- Example: List all books published after 1970
- `FILTER(C)` can be used to filter out lines in the result for which the Boolean condition `C` evaluates to `false`
- A filter is always applied to the whole pattern group in which the filter appears
- Filters can be connected with boolean operators: `&&`, `||`, `!`

Institut für Maschinelle Sprachverarbeitung

FILTER

- Sometimes it is useful to restrict the values of a variable via some **Boolean condition**

- Example: List all books published after 1970
- `FILTER(C)` can be used to filter out lines in the result for which the Boolean condition `C` evaluates to `false`
- A filter is always applied to the whole pattern group in which the filter appears
- Filters can be connected with boolean operators: `&&`, `||`, `!`

- Similar to an SQL selection

Institut für Maschinelle Sprachverarbeitung

FILTER Example

Query:

```
SELECT ?book ?author ?releasedate
WHERE {
   ?book dbo:author ?author .
   ?book dbo:author dbr:J._R._R._Tolkien .
     {?book dbp:releaseDate ?releasedate .}
   UNION
     {?book dbp:pubDate ?releasedate . }
   FILTER (?releasedate > "1970-01-01"^^xsd:date)
}
```

Result:

?book	?author	?releasedate
dbr:The_Children_of_Húrin	dbr:J._R._R._Tolkien	2007-04-16
dbr:The_Book_of_Lost_Tales	dbr:J._R._R._Tolkien	1983-10-28
dbr:The_Book_of_Lost_Tales	dbr:J._R._R._Tolkien	1984-08-16
...		

Institut für Maschinelle Sprachverarbeitung

FILTER Syntax

- For numerical types, different operators for comparison are defined:
  `<, =, >, <=, >=, !=`
- For other types, only `=` and `!=` are defined
- Literals of incompatible types cannot be compared
- Arithmetic operators can be used in filters:
  `FILTER (?weight / (?height * ?height) >= 25 )`
- There are several other predefined filters (see literature), e.g.:

`BOUND(A)`	true if A is a bound variable
`isURI(A)`	true if A is a URI
`isLITERAL(A)`	true if A is a literal
`DATATYPE(A)`	returns the URI of the datatype of A
`REGEX(A,B)`	true if regular expression B can be found in A

Institut für Maschinelle Sprachverarbeitung

FILTER Example (2)

Query:

```
SELECT ?book ?author ?releasedate
WHERE {
   ?book dbo:author ?author .
   { ?book dbp:releaseDate ?releasedate . }
   UNION
   { ?book dbp:pubDate ?releasedate . }
   FILTER (
        ?releasedate > "1950-01-01"^^xsd:date &&
        !((?releasedate + 5) >= "1970-12-31"^^xsd:date)
   )
}
```

Result:

?book	?author	?releasedate
dbr:I,_Robot	dbr:Isaac_Asimov	1950-12-02
...		

Institut für Maschinelle Sprachverarbeitung

Quiz: FILTER

Determine at which point the line `FILTER (?elevation > 8000)` has to be inserted, so that only 8000ers are retrieved

```
SELECT ?mountain ?name ?elevation ?location
WHERE {
   ?mountain rdf:type dbo:Mountain .
   ?mountain dbp:name ?name .
   [A]
   OPTIONAL {
      ?mountain dbp:location ?location .
      [B]
   }
   OPTIONAL {
      ?mountain dbp:elevationM ?elevation .
      [C]
  }
   [D]
}
```

Institut für Maschinelle Sprachverarbeitung

Outline

Institut für Maschinelle Sprachverarbeitung

UNION

- A sequence of patterns in a SPARQL query is always interpreted as a conjunction (logical **and**)
- To use the logical **or** for parts of the pattern, pattern alternatives must be used
- Pattern alternatives are specified with the `{ ... } UNION { ... }` block

Institut für Maschinelle Sprachverarbeitung

UNION

- A sequence of patterns in a SPARQL query is always interpreted as a conjunction (logical **and**)
- To use the logical **or** for parts of the pattern, pattern alternatives must be used
- Pattern alternatives are specified with the `{ ... } UNION { ... }` block

- Roughly speaking, it returns the **union** of the intermediate results returned for each of its operands

Institut für Maschinelle Sprachverarbeitung

UNION Example

Query:

```
SELECT ?book ?author ?releasedate
WHERE {
   ?book dbo:author ?author .
   {
      ?book dbp:releaseDate ?releasedate .
   } UNION {
      ?book dbp:pubDate ?releasedate .
   }
}
```

Result:

?book	?author	?releasedate
dbr:The_Children_of_Húrin	dbr:J._R._R._Tolkien	2007-04-16
dbr:The_Book_of_Lost_Tales	dbr:J._R._R._Tolkien	1983-10-28
dbr:The_Book_of_Lost_Tales	dbr:J._R._R._Tolkien	1984-08-16
...		
dbr:Unfinished_Tales	dbr:J._R._R._Tolkien	
...		

Institut für Maschinelle Sprachverarbeitung

Quiz: UNION

Some locations are given with `dbo:locationArea` some with `dbp:location`. Replace the given query pattern to handle both:

```
SELECT ?mountain ?location
WHERE {
   ?mountain rdf:type dbo:Mountain .
   ?mountain dbo:locationArea ?location .
}
```

A)
```
?mountain rdf:type dbo:Mountain .
{  ?mountain dbo:locationArea ?location . }
UNION   { ?mountain dbp:location ?location . }
```

B)
```
{  ?mountain rdf:type dbo:Mountain .
?mountain dbo:locationArea ?location . }
UNION   { ?mountain dbp:location ?location . }
```

C)
```
?mountain rdf:type dbo:Mountain .
?mountain dbo:locationArea ?location .
OPTIONAL { ?mountain dbp:location ?location . }
```

Institut für Maschinelle Sprachverarbeitung

Combining OPTIONAL and UNION

- We can nest OPTIONAL and UNION query blocks within each other
- The order of the nesting will yield different results
- Reordering the triple joins will yield again different results, depending on how they interplay with OPTIONAL
- We can apply OPTIONAL to any ensuing table, whether the final table, or the intermediate tables in the computation
- Again, SPARQL is mirroring SQL in this regard

Institut für Maschinelle Sprachverarbeitung

OPTIONAL and UNION Example (1)

Query:

```
SELECT ?book ?author ?releasedate
WHERE {
   ?book dbo:author ?author .
   OPTIONAL {
      {
         ?book dbp:releaseDate ?releasedate .
      } UNION {
         ?book dbp:pubDate ?releasedate .
      }
   }
}
```

Result:

?book	?author	?releasedate
...		
dbr:The_Hobbit	dbr:J._R._R._Tolkien	1937-09-21
dbr:The_Adventures_of_Tom_Bombadil	dbr:J._R._R._Tolkien	1962
dbr:The_Quest_of_Erebor	dbr:J._R._R._Tolkien	1980
dbr:Songs_for_the_Philologists	dbr:J._R._R._Tolkien	1936
dbr:The_Children_of_Húrin	dbr:J._R._R._Tolkien	
...		

Institut für Maschinelle Sprachverarbeitung

OPTIONAL and UNION Example (2)

Query:

```
SELECT ?book ?author ?releasedate ?illustrator
WHERE {
   ?book dbo:author ?author .
   {
      ?book dbp:releaseDate ?releasedate .
   } UNION {
      ?book dbp:pubDate ?releasedate .
   }
   OPTIONAL { ?book dbo:illustrator ?illustrator . }
}
```

Result:

?book	?author	?releasedate	?illustrator
...			
dbr:The_Hobbit	dbr:J._R._R._Tolkien	1937-09-21	dbr:J._R._R._Tolkien
dbr:The_Adventures_of_Tom_Bombadil	dbr:J._R._R._Tolkien	1962	dbr:Pauline_Baynes
dbr:The_Quest_of_Erebor	dbr:J._R._R._Tolkien	1980	
dbr:Songs_for_the_Philologists	dbr:J._R._R._Tolkien	1936	
...			

Institut für Maschinelle Sprachverarbeitung

OPTIONAL and UNION Example (3)

Query:

```
SELECT ?book ?author ?releasedate ?illustrator
WHERE {
   ?book dbo:author ?author .
   {
      ?book dbp:releaseDate ?releasedate .
   } UNION {
      ?book dbp:pubDate ?releasedate .
      OPTIONAL { ?book dbo:illustrator ?illustrator . }
   }
}
```

Result:

?book	?author	?releasedate	?illustrator
...			
dbr:The_Hobbit	dbr:J._R._R._Tolkien	1937-09-21	
dbr:The_Adventures_of_Tom_Bombadil	dbr:J._R._R._Tolkien	1962	dbr:Pauline_Baynes
dbr:The_Quest_of_Erebor	dbr:J._R._R._Tolkien	1980	
dbr:Songs_for_the_Philologists	dbr:J._R._R._Tolkien	1936	
...			

Institut für Maschinelle Sprachverarbeitung

More . . .

- Modifiers to influence the presentation:

ORDER BY	establishes the ordering of the resulting lines by some variable, ascending or descending
LIMIT	limits the number of solutions returned
OFFSET	causes the solutions generated to start after the specified number of solutions (only predictable in combination with ORDER BY)
DISTINCT	eliminates duplicate solutions (a solution that binds the same variables to the same RDF terms as another solution)

- Different query forms, besides SELECT covered here, there is ASK, DESCRIBE and CONSTRUCT

Institut für Maschinelle Sprachverarbeitung

Exercise: SPARQL

- Write some SPARQL queries that your application might use
- Discuss your solution with your neighbor

- Example: List the title and year of writing (if available) of all songs written by Michael Jackson.

```
SELECT ?title ?year
WHERE {
   ?song rdf:type ex:Song .
   ?song ex:writtenBy ex:MichaelJackson .
   ?song ex:hasTitle ?title .
   OPTIONAL {
      ?song ex:writtenIn ?date .
      ?date ex:hasYear ?year .
   }
}
```

Institut für Maschinelle Sprachverarbeitung

Outline

Institut für Maschinelle Sprachverarbeitung

Summary:

- SPARQL matches **graph patterns** in the query with an RDF graph, if the query contains **variables**, the variables get bound to the parts of the triple that correspond to them
- Several patterns are regarded as a **conjunction**, to express a **disjunction** ("or"), `UNION` can be used
- `OPTIONAL` can be used to include the information **if available**, otherwise the result is returned with the variable unbound
- `FILTER` can be used to **filter** out lines in the result for which the filter expression evaluates to `false`

Institut für Maschinelle Sprachverarbeitung

Outline

Institut für Maschinelle Sprachverarbeitung

Suggested Reading


Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph and York Sure. Semantic Web. Grundlagen. Springer textbook, 2008. (Chapter 7)


Pascal Hitzler, Markus Krötzsch and Sebastian Rudolph. Foundations of Semantic Web Technologies. Chapman & Hall/CRC, 2009. (Chapter 7)

Institut für Maschinelle Sprachverarbeitung

Appendix: Combining OPTIONAL and UNION

- `OPTIONAL` and `UNION` are left-associative:

  ```
  pattern OPTIONAL { pattern } OPTIONAL { pattern }
  ```

  is equivalent to

  ```
  { pattern OPTIONAL { pattern } } OPTIONAL { pattern }
  ```

- Both operators have the same precedence:

  ```
  {s1 p1 o1} OPTIONAL {s2 p2 o2} UNION {s3 p3 o3}
  OPTIONAL {s4 p4 o4} OPTIONAL {s5 p5 o5}
  ```

  is equivalent to

  ```
  { { { {s1 p1 o1} OPTIONAL {s2 p2 o2} } UNION {s3 p3 o3}
     } OPTIONAL {s4 p4 o4} } OPTIONAL {s5 p5 o5}
  ```

- Additional `{}` may always be used to group patterns

Institut für Maschinelle Sprachverarbeitung