

Semantic Days 2010 Tutorial

Semantic Web Technologies

Lecture 3: The SPARQL Query Language

Martin Giese

31st May 2010



DEPARTMENT OF
INFORMATICS



UNIVERSITY OF
OSLO

- SPARQL Protocol And RDF Query Language

SPARQL

- SPARQL Protocol And RDF Query Language
- Documentation:

SPARQL

- SPARQL Protocol And RDF Query Language

- Documentation:

Queries <http://www.w3.org/TR/rdf-sparql-query/>
Language for submitting “graph pattern” queries

SPARQL

- SPARQL Protocol And RDF Query Language

- Documentation:

- **Queries** <http://www.w3.org/TR/rdf-sparql-query/>
Language for submitting “graph pattern” queries

- **Protocol** <http://www.w3.org/TR/rdf-sparql-protocol/>
Protocol to submit queries to a server (“endpoint”)

SPARQL

- SPARQL Protocol And RDF Query Language
- Documentation:

Queries <http://www.w3.org/TR/rdf-sparql-query/>
Language for submitting “graph pattern” queries

Protocol <http://www.w3.org/TR/rdf-sparql-protocol/>
Protocol to submit queries to a server (“endpoint”)

Results <http://www.w3.org/TR/rdf-sparql-XMLres/>
XML format in which results are returned

SPARQL

- SPARQL Protocol And RDF Query Language

- Documentation:

Queries <http://www.w3.org/TR/rdf-sparql-query/>
Language for submitting “graph pattern” queries

Protocol <http://www.w3.org/TR/rdf-sparql-protocol/>
Protocol to submit queries to a server (“endpoint”)

Results <http://www.w3.org/TR/rdf-sparql-XMLres/>
XML format in which results are returned

- Try it out:

SPARQL

- SPARQL Protocol And RDF Query Language

- Documentation:

Queries <http://www.w3.org/TR/rdf-sparql-query/>
Language for submitting “graph pattern” queries

Protocol <http://www.w3.org/TR/rdf-sparql-protocol/>
Protocol to submit queries to a server (“endpoint”)

Results <http://www.w3.org/TR/rdf-sparql-XMLres/>
XML format in which results are returned

- Try it out:

DBLP <http://dblp.13s.de/d2r/snorql/>

SPARQL

- SPARQL Protocol And RDF Query Language

- Documentation:

Queries <http://www.w3.org/TR/rdf-sparql-query/>
Language for submitting “graph pattern” queries

Protocol <http://www.w3.org/TR/rdf-sparql-protocol/>
Protocol to submit queries to a server (“endpoint”)

Results <http://www.w3.org/TR/rdf-sparql-XMLres/>
XML format in which results are returned

- Try it out:

DBLP <http://dblp.13s.de/d2r/snorql/>

DBpedia <http://dbpedia.org/sparql>

SPARQL

- SPARQL Protocol And RDF Query Language

- Documentation:

- **Queries** <http://www.w3.org/TR/rdf-sparql-query/>
Language for submitting “graph pattern” queries

- **Protocol** <http://www.w3.org/TR/rdf-sparql-protocol/>
Protocol to submit queries to a server (“endpoint”)

- **Results** <http://www.w3.org/TR/rdf-sparql-XMLres/>
XML format in which results are returned

- Try it out:

- **DBLP** <http://dblp.13s.de/d2r/snorql/>

- **DBpedia** <http://dbpedia.org/sparql>

- **DBtunes** <http://dbtune.org/musicbrainz/snorql/>

SPARQL

- SPARQL Protocol And RDF Query Language

- Documentation:

Queries <http://www.w3.org/TR/rdf-sparql-query/>
Language for submitting “graph pattern” queries

Protocol <http://www.w3.org/TR/rdf-sparql-protocol/>
Protocol to submit queries to a server (“endpoint”)

Results <http://www.w3.org/TR/rdf-sparql-XMLres/>
XML format in which results are returned

- Try it out:

DBLP <http://dblp.13s.de/d2r/snorql/>

DBpedia <http://dbpedia.org/sparql>

DBtunes <http://dbtune.org/musicbrainz/snorql/>

World DB <http://sws.ifi.uio.no/d2rq/snorql/>

Simple Examples

- DBLP contains computer science publications

Simple Examples

- DBLP contains computer science publications
- vocabulary of RDF version:

Simple Examples

- DBLP contains computer science publications
- vocabulary of RDF version:
 - author of a document: `dc:creator`

Simple Examples

- DBLP contains computer science publications
- vocabulary of RDF version:
 - author of a document: `dc:creator`
 - title of a document: `dc:title`

Simple Examples

- DBLP contains computer science publications
- vocabulary of RDF version:
 - author of a document: `dc:creator`
 - title of a document: `dc:title`
 - name of a person: `foaf:name`

Simple Examples

- DBLP contains computer science publications
- vocabulary of RDF version:
 - author of a document: `dc:creator`
 - title of a document: `dc:title`
 - name of a person: `foaf:name`

People called “Martin Giese”

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?mg WHERE {
    ?mg foaf:name "Martin Giese" .
}
```

Simple Examples

- DBLP contains computer science publications
- vocabulary of RDF version:
 - author of a document: `dc:creator`
 - title of a document: `dc:title`
 - name of a person: `foaf:name`

People called “Martin Giese”

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?mg WHERE {
    ?mg foaf:name "Martin Giese" .
}
```

Answer:

?mg
http://dblp.l3s.de/d2r/resource/authors/Martin_Giese

Simple Examples (cont.)

Publications by people called "Martin Giese"

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
SELECT ?pub WHERE {
    ?mg foaf:name "Martin Giese" .
    ?pub dc:creator ?mg .
}
```

Simple Examples (cont.)

Publications by people called "Martin Giese"

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
SELECT ?pub WHERE {
    ?mg foaf:name "Martin Giese" .
    ?pub dc:creator ?mg .
}
```

Answer:

?pub

<http://dblp.13s.de/d2r/resource/publications/conf/cade/Giese01>

<http://dblp.13s.de/d2r/resource/publications/conf/cade/BeckertGHKRSS07>

<http://dblp.13s.de/d2r/resource/publications/conf/fase/AhrendtBBGHHMS02>

<http://dblp.13s.de/d2r/resource/publications/conf/jelia/AhrendtBBGHHMS00>

<http://dblp.13s.de/d2r/resource/publications/conf/lpar/Giese06>

...

Simple Examples (cont.)

Titles of publications by people called "Martin Giese"

```
SELECT ?title WHERE {  
  ?mg foaf:name "Martin Giese" .  
  ?pub dc:creator ?mg .  
  ?pub dc:title ?title .  
}
```

Simple Examples (cont.)

Titles of publications by people called "Martin Giese"

```
SELECT ?title WHERE {  
  ?mg foaf:name "Martin Giese" .  
  ?pub dc:creator ?mg .  
  ?pub dc:title ?title .  
}
```

Answer:

?title
"Incremental Closure of Free Variable Tableaux."^^xsd:string
"The KeY system 1.0 (Deduction Component)."
"The KeY System: Integrating Object-Oriented Design and Formal Methods."
"The KeY Approach: Integrating Object Oriented Design and Formal Verification."
"Saturation Up to Redundancy for Tableau and Sequent Calculi."
...

Simple Examples (cont.)

Names of people who have published with “Martin Giese”

```
SELECT ?name WHERE {  
  ?mg foaf:name "Martin Giese" .  
  ?pub dc:creator ?mg .  
  ?pub dc:creator ?other .  
  ?other foaf:name ?name.  
}
```

Simple Examples (cont.)

Names of people who have published with “Martin Giese”

```
SELECT ?name WHERE {  
  ?mg foaf:name "Martin Giese" .  
  ?pub dc:creator ?mg .  
  ?pub dc:creator ?other .  
  ?other foaf:name ?name.  
}
```

Answer:

?name
"Martin Giese"
"Bernhard Beckert"
"Martin Giese"
"Reiner Hähnle"
"Vladimir Klebanov"
...

Simple Examples (cont.)

Names of people who have published with “Martin Giese”

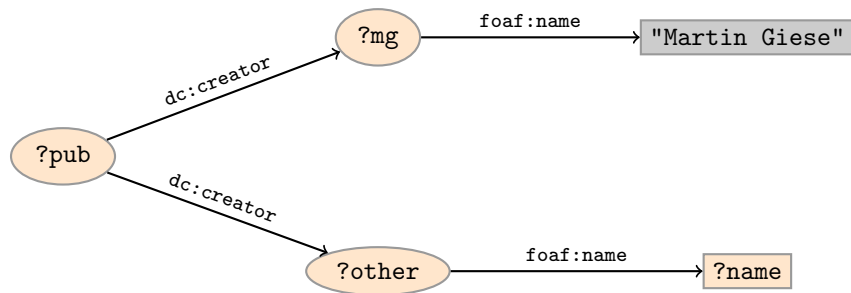
```
SELECT DISTINCT ?name WHERE {  
  ?mg foaf:name "Martin Giese" .  
  ?pub dc:creator ?mg .  
  ?pub dc:creator ?other .  
  ?other foaf:name ?name.  
}
```

Answer:

?name
"Martin Giese"
"Bernhard Beckert"
"Reiner Hähnle"
"Vladimir Klebanov"
"Philipp Rümmer"
...

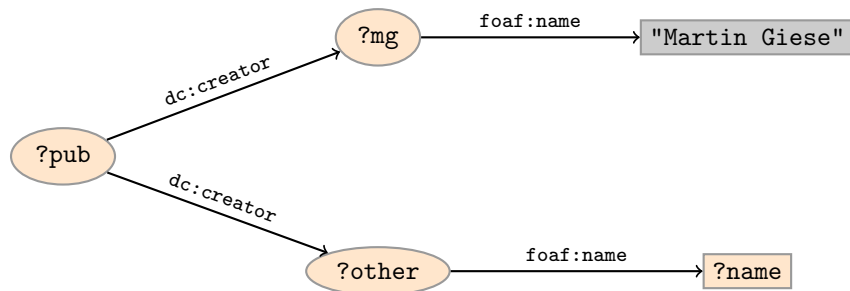
Graph Patterns

The previous SPARQL query as a graph:



Graph Patterns

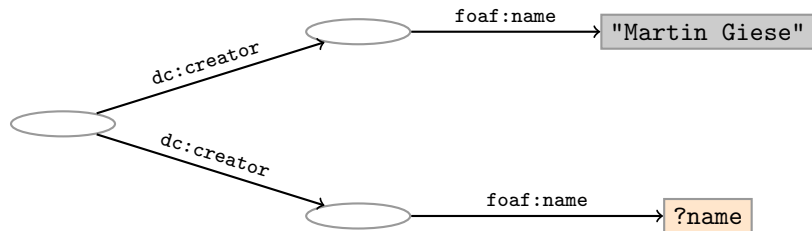
The previous SPARQL query as a graph:



Assign values to variables to make this a sub-graph of the RDF graph!

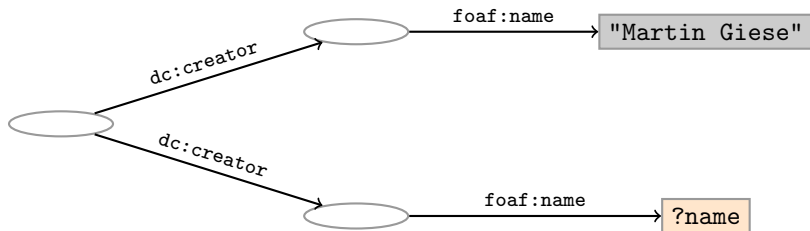
Graph with blank nodes

Variables not SELECTed can equivalently be blank:



Graph with blank nodes

Variables not SELECTed can equivalently be blank:



Assign values to variables **and blank nodes** to make this a sub-graph of the RDF graph!

SPARQL Query with blank nodes

Names of people who have published with “Martin Giese”

```
SELECT DISTINCT ?name WHERE {  
  _:mg foaf:name "Martin Giese" .  
  _:pub dc:creator _:mg .  
  _:pub dc:creator _:other .  
  _:other foaf:name ?name.  
}
```

SPARQL Query with blank nodes

Names of people who have published with “Martin Giese”

```
SELECT DISTINCT ?name WHERE {  
  _:mg foaf:name "Martin Giese" .  
  _:pub dc:creator _:mg .  
  _:pub dc:creator _:other .  
  _:other foaf:name ?name.  
}
```

The same with blank node syntax

```
SELECT DISTINCT ?name WHERE {  
  _:pub dc:creator [foaf:name "Martin Giese"] .  
  _:pub dc:creator _:other .  
  _:other foaf:name ?name.  
}
```

SPARQL Query with blank nodes

Names of people who have published with “Martin Giese”

```
SELECT DISTINCT ?name WHERE {  
  _:mg foaf:name "Martin Giese" .  
  _:pub dc:creator _:mg .  
  _:pub dc:creator _:other .  
  _:other foaf:name ?name.  
}
```

The same with blank node syntax

```
SELECT DISTINCT ?name WHERE {  
  _:pub dc:creator [foaf:name "Martin Giese"] .  
  _:pub dc:creator [foaf:name ?name] .  
}
```


SPARQL Query with blank nodes

Names of people who have published with “Martin Giese”

```
SELECT DISTINCT ?name WHERE {  
  _:mg foaf:name "Martin Giese" .  
  _:pub dc:creator _:mg .  
  _:pub dc:creator _:other .  
  _:other foaf:name ?name.  
}
```

The same with blank node syntax

```
SELECT DISTINCT ?name WHERE {  
  [ dc:creator [foaf:name "Martin Giese"] ;  
    dc:creator [foaf:name ?name]  
  ]  
}
```

SPARQL Query with blank nodes

Names of people who have published with “Martin Giese”

```
SELECT DISTINCT ?name WHERE {  
  _:mg foaf:name "Martin Giese" .  
  _:pub dc:creator _:mg .  
  _:pub dc:creator _:other .  
  _:other foaf:name ?name.  
}
```

The same with blank node syntax

```
SELECT DISTINCT ?name WHERE {  
  [ dc:creator [foaf:name "Martin Giese"] ,  
              [foaf:name ?name]  
  ]  
}
```

Filters

- Groups may include *constraints* or *filters*

Filters

- Groups may include *constraints* or *filters*

- E.g.

```
{  
  ?x a dbpedia-owl:Place ;  
      dbpprop:population ?pop .  
  FILTER (?pop > 1000000)  
}
```

Filters

- Groups may include *constraints* or *filters*

- E.g.

```
{  
  ?x a dbpedia-owl:Place ;  
      dbpprop:population ?pop .  
  FILTER (?pop > 1000000)  
}
```

- E.g.

```
{  
  ?x a dbpedia-owl:Place ;  
      dbpprop:abstract ?abs .  
  FILTER (lang(?abs) = "no")  
}
```

Filters

- Groups may include *constraints* or *filters*

- E.g.

```
{  
  ?x a dbpedia-owl:Place ;  
      dbpprop:population ?pop .  
  FILTER (?pop > 1000000)  
}
```

- E.g.

```
{  
  ?x a dbpedia-owl:Place ;  
      dbpprop:abstract ?abs .  
  FILTER (lang(?abs) = "no")  
}
```

- Numerical functions, string operations, reg. exp. matching, etc.

Filters

- Groups may include *constraints* or *filters*

- E.g.

```
{  
  ?x a dbpedia-owl:Place ;  
      dbpprop:population ?pop .  
  FILTER (?pop > 1000000)  
}
```

- E.g.

```
{  
  ?x a dbpedia-owl:Place ;  
      dbpprop:abstract ?abs .  
  FILTER (lang(?abs) = "no")  
}
```

- Numerical functions, string operations, reg. exp. matching, etc.
- Answers only matches where filter applies

Optional Patters

- A match can leave some variables *unbound*

Optional Patters

- A match can leave some variables *unbound*
- A *partial* function from variables to RDF terms

Optional Patterns

- A match can leave some variables *unbound*
- A *partial* function from variables to RDF terms
- Queries may include *optional parts*

Optional Patterns

- A match can leave some variables *unbound*
- A *partial* function from variables to RDF terms
- Queries may include *optional parts*
- E.g.

```
{  
  ?x a dbpedia-owl:Place ;  
     dbpprop:population ?pop .  
  OPTIONAL {  
    ?x dbpprop:abstract ?abs .  
    FILTER (lang(?abs) = "no")  
  }  
}
```

Optional Patterns

- A match can leave some variables *unbound*
- A *partial* function from variables to RDF terms
- Queries may include *optional parts*
- E.g.

```
{  
  ?x a dbpedia-owl:Place ;  
    dbpprop:population ?pop .  
  OPTIONAL {  
    ?x dbpprop:abstract ?abs .  
    FILTER (lang(?abs) = "no")  
  }  
}
```

- ?x and ?pop bound in every match, ?abs bound if there is a Norwegian abstract

Optional Patterns

- A match can leave some variables *unbound*
- A *partial* function from variables to RDF terms
- Queries may include *optional parts*
- E.g.

```
{  
  ?x a dbpedia-owl:Place ;  
    dbpprop:population ?pop .  
  OPTIONAL {  
    ?x dbpprop:abstract ?abs .  
    FILTER (lang(?abs) = "no")  
  }  
}
```

- ?x and ?pop bound in every match, ?abs bound if there is a Norwegian abstract
- Groups can contain several optional parts, evaluated separately

Matching Alternatives

- A UNION pattern matches if any of some alternatives matches

Matching Alternatives

- A UNION pattern matches if any of some alternatives matches

- E.g.

```
{  
  { ?book dc:creator ?author ;  
    dc:created ?date . }  
UNION  
  { ?book foaf:maker ?author . }  
UNION  
  { ?author foaf:made ?book . }  
}
```

Matching Alternatives

- A UNION pattern matches if any of some alternatives matches

- E.g.

```
{  
  { ?book dc:creator ?author ;  
    dc:created ?date . }  
UNION  
  { ?book foaf:maker ?author . }  
UNION  
  { ?author foaf:made ?book . }  
}
```

- Variables in matches union of variables in sub-patterns

Matching Alternatives

- A UNION pattern matches if any of some alternatives matches

- E.g.

```
{  
  { ?book dc:creator ?author ;  
    dc:created ?date . }  
  UNION  
  { ?book foaf:maker ?author . }  
  UNION  
  { ?author foaf:made ?book . }  
}
```

- Variables in matches union of variables in sub-patterns
- Match of one pattern leaves rest of variables unbound

Four Types of Queries

SELECT Compute table of bindings for variables

```
SELECT ?a ?b WHERE {  
  [ dc:creator ?a ;  
    dc:creator ?b ]  
}
```

Four Types of Queries

SELECT Compute table of bindings for variables

```
SELECT ?a ?b WHERE {  
  [ dc:creator ?a ;  
    dc:creator ?b ]  
}
```

CONSTRUCT Use bindings to construct a new RDF graph

```
CONSTRUCT {  
  ?a foaf:knows ?b .  
} WHERE {  
  [ dc:creator ?a ;  
    dc:creator ?b ]  
}
```

Four Types of Queries

SELECT Compute table of bindings for variables

```
SELECT ?a ?b WHERE {  
  [ dc:creator ?a ;  
    dc:creator ?b ]  
}
```

CONSTRUCT Use bindings to construct a new RDF graph

```
CONSTRUCT {  
  ?a foaf:knows ?b .  
} WHERE {  
  [ dc:creator ?a ;  
    dc:creator ?b ]  
}
```

ASK Answer (yes/no) whether there is ≥ 1 match

Four Types of Queries

SELECT Compute table of bindings for variables

```
SELECT ?a ?b WHERE {  
  [ dc:creator ?a ;  
    dc:creator ?b ]  
}
```

CONSTRUCT Use bindings to construct a new RDF graph

```
CONSTRUCT {  
  ?a foaf:knows ?b .  
} WHERE {  
  [ dc:creator ?a ;  
    dc:creator ?b ]  
}
```

ASK Answer (yes/no) whether there is ≥ 1 match

DESCRIBE Answer available information about matching resources

Solution Modifiers

- Patterns generate an unordered collection of solutions

Solution Modifiers

- Patterns generate an unordered collection of solutions
- Each solution is a partial function from variables to RDF terms

Solution Modifiers

- Patterns generate an unordered collection of solutions
- Each solution is a partial function from variables to RDF terms
- SELECT treats solutions as a sequence (solution sequence)

Solution Modifiers

- Patterns generate an unordered collection of solutions
- Each solution is a partial function from variables to RDF terms
- SELECT treats solutions as a sequence (solution sequence)
- *Sequence modifiers* can modify the solution sequence:

Solution Modifiers

- Patterns generate an unordered collection of solutions
- Each solution is a partial function from variables to RDF terms
- SELECT treats solutions as a sequence (solution sequence)
- *Sequence modifiers* can modify the solution sequence:
 - Order: ...ORDER BY ?country DESC(?pop)

Solution Modifiers

- Patterns generate an unordered collection of solutions
- Each solution is a partial function from variables to RDF terms
- SELECT treats solutions as a sequence (solution sequence)
- *Sequence modifiers* can modify the solution sequence:
 - Order: ...ORDER BY ?country DESC(?pop)
 - Projection: SELECT ?x ?y WHERE...

Solution Modifiers

- Patterns generate an unordered collection of solutions
- Each solution is a partial function from variables to RDF terms
- SELECT treats solutions as a sequence (solution sequence)
- *Sequence modifiers* can modify the solution sequence:
 - Order: ...ORDER BY ?country DESC(?pop)
 - Projection: SELECT ?x ?y WHERE...
 - Distinct: SELECT DISTINCT ?x WHERE...

Solution Modifiers

- Patterns generate an unordered collection of solutions
- Each solution is a partial function from variables to RDF terms
- SELECT treats solutions as a sequence (solution sequence)
- *Sequence modifiers* can modify the solution sequence:
 - Order: ...ORDER BY ?country DESC(?pop)
 - Projection: SELECT ?x ?y WHERE...
 - Distinct: SELECT DISTINCT ?x WHERE...
 - Reduce: SELECT REDUCE ?x WHERE...

Solution Modifiers

- Patterns generate an unordered collection of solutions
- Each solution is a partial function from variables to RDF terms
- SELECT treats solutions as a sequence (solution sequence)
- *Sequence modifiers* can modify the solution sequence:
 - Order: ...ORDER BY ?country DESC(?pop)
 - Projection: SELECT ?x ?y WHERE...
 - Distinct: SELECT DISTINCT ?x WHERE...
 - Reduce: SELECT REDUCE ?x WHERE...
 - Sub-sequence: ...LIMIT 10 OFFSET 50

Solution Modifiers

- Patterns generate an unordered collection of solutions
- Each solution is a partial function from variables to RDF terms
- SELECT treats solutions as a sequence (solution sequence)
- *Sequence modifiers* can modify the solution sequence:
 - Order: ...ORDER BY ?country DESC(?pop)
 - Projection: SELECT ?x ?y WHERE...
 - Distinct: SELECT DISTINCT ?x WHERE...
 - Reduce: SELECT REDUCE ?x WHERE...
 - Sub-sequence: ...LIMIT 10 OFFSET 50
- Applied in this order.

Missing in SPARQL

SPARQL does *not* include (amongst others):

- aggregate functions (count, sum, average, . . .)

Missing in SPARQL

SPARQL does *not* include (amongst others):

- aggregate functions (count, sum, average, . . .)
 - difficult with “open world assumption”

Missing in SPARQL

SPARQL does *not* include (amongst others):

- aggregate functions (count, sum, average, . . .)
 - difficult with “open world assumption”
 - i.e. statements may be true even if they are not asserted in a model

Missing in SPARQL

SPARQL does *not* include (amongst others):

- aggregate functions (count, sum, average, . . .)
 - difficult with “open world assumption”
 - i.e. statements may be true even if they are not asserted in a model
- negation, set difference, i.e. something is *not* in a graph

Missing in SPARQL

SPARQL does *not* include (amongst others):

- aggregate functions (count, sum, average, . . .)
 - difficult with “open world assumption”
 - i.e. statements may be true even if they are not asserted in a model
- negation, set difference, i.e. something is *not* in a graph
 - also not compatible with open world assumption

Missing in SPARQL

SPARQL does *not* include (amongst others):

- aggregate functions (count, sum, average, . . .)
 - difficult with “open world assumption”
 - i.e. statements may be true even if they are not asserted in a model
- negation, set difference, i.e. something is *not* in a graph
 - also not compatible with open world assumption
 - can use FILTER to check that variables are not bound

Missing in SPARQL

SPARQL does *not* include (amongst others):

- aggregate functions (count, sum, average, . . .)
 - difficult with “open world assumption”
 - i.e. statements may be true even if they are not asserted in a model
- negation, set difference, i.e. something is *not* in a graph
 - also not compatible with open world assumption
 - can use FILTER to check that variables are not bound
- updates (add delete triples)

Missing in SPARQL

SPARQL does *not* include (amongst others):

- aggregate functions (count, sum, average, . . .)
 - difficult with “open world assumption”
 - i.e. statements may be true even if they are not asserted in a model
- negation, set difference, i.e. something is *not* in a graph
 - also not compatible with open world assumption
 - can use FILTER to check that variables are not bound
- updates (add delete triples)

Some of this is probably coming. . .

<http://www.w3.org/TR/2009/WD-sparql-features-20090702/>

SPARQL on the 'Net

- Many sites (DBLP, dbpedia, dbtunes, . . .) publish *SPARQL endpoints*

SPARQL on the 'Net

- Many sites (DBLP, dbpedia, dbtunes, . . .) publish *SPARQL endpoints*
- I.e. SPARQL queries can be submitted to a database server that sends back the results

SPARQL on the 'Net

- Many sites (DBLP, dbpedia, dbtunes, ...) publish *SPARQL endpoints*
- I.e. SPARQL queries can be submitted to a database server that sends back the results
- Uses HTTP to submit URL-encoded queries to server
GET /sparql/?query=... HTTP/1.1

SPARQL on the 'Net

- Many sites (DBLP, dbpedia, dbtunes, ...) publish *SPARQL endpoints*
- I.e. SPARQL queries can be submitted to a database server that sends back the results
- Uses HTTP to submit URL-encoded queries to server
GET /sparql/?query=... HTTP/1.1
- Actually defined in terms of W3C Web Services, see
<http://www.w3.org/TR/rdf-sparql-protocol/>

SPARQL on the 'Net

- Many sites (DBLP, dbpedia, dbtunes, . . .) publish *SPARQL endpoints*
- I.e. SPARQL queries can be submitted to a database server that sends back the results
- Uses HTTP to submit URL-encoded queries to server
GET /sparql/?query=... HTTP/1.1
- Actually defined in terms of W3C Web Services, see
<http://www.w3.org/TR/rdf-sparql-protocol/>
- Server responds with XML file encoding result set, see
<http://www.w3.org/TR/rdf-sparql-XMLres/>

SPARQL on the 'Net

- Many sites (DBLP, dbpedia, dbtunes, . . .) publish *SPARQL endpoints*
- I.e. SPARQL queries can be submitted to a database server that sends back the results
- Uses HTTP to submit URL-encoded queries to server
GET /sparql/?query=... HTTP/1.1
- Actually defined in terms of W3C Web Services, see
<http://www.w3.org/TR/rdf-sparql-protocol/>
- Server responds with XML file encoding result set, see
<http://www.w3.org/TR/rdf-sparql-XMLres/>
- Nothing you would want to do manually!

SPARQL on the 'Net

- Many sites (DBLP, dbpedia, dbtunes, . . .) publish *SPARQL endpoints*
- I.e. SPARQL queries can be submitted to a database server that sends back the results
- Uses HTTP to submit URL-encoded queries to server
GET /sparql/?query=... HTTP/1.1
- Actually defined in terms of W3C Web Services, see
<http://www.w3.org/TR/rdf-sparql-protocol/>
- Server responds with XML file encoding result set, see
<http://www.w3.org/TR/rdf-sparql-XMLres/>
- Nothing you would want to do manually!
 - Mostly, there is a web page for manual entry of queries

SPARQL on the 'Net

- Many sites (DBLP, dbpedia, dbtunes, ...) publish *SPARQL endpoints*
- I.e. SPARQL queries can be submitted to a database server that sends back the results
- Uses HTTP to submit URL-encoded queries to server
GET /sparql/?query=... HTTP/1.1
- Actually defined in terms of W3C Web Services, see
<http://www.w3.org/TR/rdf-sparql-protocol/>
- Server responds with XML file encoding result set, see
<http://www.w3.org/TR/rdf-sparql-XMLres/>
- Nothing you would want to do manually!
 - Mostly, there is a web page for manual entry of queries
 - RDF APIs like Jena can send queries to endpoints

Exercise: SPARQL

- Write SPARQL queries.
- Run queries on SPARQL endpoint at `http://sws.ifi.uio.no/d2rq/snorql/`.
- Browse dataset using web browser `http://sws.ifi.uio.no/d2rq/`.
- Many queries, choose according to your skills.

Go to `http://sws.ifi.uio.no/semdays2010/` for more information.