E05 Processing SPARQL Queries

Quiz

Decide whether the following statements are true or false.

Q5.1 A basic graph pattern consists of one or more triple patterns.

Q5.2 BGPs may contain blank nodes on subject, predicate, and object position.

Q5.3 Basic graph pattern matching in SPARQL takes into account different lexical forms of literals

Q5.4 Of the operators presented in the chapter, the *BGP* operator is the only operator that accesses triples.

Q5.5 Only one *FROM* expression is allowed in a SPARQL query.

Q5.6 Multiple *FILTER* expressions are allowed in a SPARQL query. Only solutions that match all of the *FILTER* expressions are returned.

Q5.7 The empty solution mapping $\mu_\emptyset$ is compatible with any other solution mapping.

Q5.8 The notion of compatibility of solutions can be used to compute the Cartesian product between solution sequences.

Q5.9 Filter expressions in SPARQL take into account different lexical forms of literals.

Q5.10 The following filter expression returns *false*:
`"1"^^xsd:int == (xsd:integer)"1"^^xsd:float`

Q5.11 Queries with a CONSTRUCT query form may lead to new blank nodes in the query results.

Exercises

E5.1 Let *G* be the graph serialized in the following RDF document doc.ttl:

```
@prefix : <doc#> .
:a :p :b .
:b :p :c .
:c :p :d .
```

Now, consider the BGP *P*:

```
  <doc#a> <doc#p> <doc#b> .
```

What is the solution sequence Ω resulting from matching *P* to *G*?

E5.2 Given the basic graph pattern *P*:

```
  <doc#a> <doc#p> <doc#c> .
```

What is the solution sequence Ω resulting from matching *P* to graph *G* from E 5.1?

E5.3 Consider the BGP *P*:

```
  _:x <doc#p> _:y .
```

What is the solution sequence Ω resulting from matching *P* to graph *G* from E 5.1?

E5.4 Given the following BGP *P* with two triple patterns:

```
  ?x <doc#p> ?y . ?y <doc#p> ?z .
```

What is the solution sequence Ω resulting from matching *P* to graph *G* from E 5.1?

E5.5 Given the following BGP *P*:

```
  ?x <doc#p> _:y . _:y <doc#p> ?z .
```

What is the solution sequence Ω resulting from matching *P* to graph *G* from E 5.1?

E5.6 Given the following BGP *P*, where the triple patterns do not share variables[1]:

```
  ?x1 <doc#p> ?y1 .
  ?x2 <doc#p> ?y2 .
```

[1] https://lists.w3.org/Archives/Public/public-sparql-dev/2017AprJun/0008.html

Can you work out from the definitions the solution sequence Ω formatching *P* to graph *G* from E 5.1?

E5.7 Translate the following *WHERE* clause into a SPARQL algebra expression, without expanding prefixes in CURIEs.

```
WHERE {
   ?x :name ?label .
   ?x :radius ?r .
}
```

E5.8 Translate the following *WHERE* clause into a SPARQL algebra expression, without expanding prefixes in CURIEs.

```
WHERE {
   ?x :label ?label .
   FILTER (lang(?label) = "en")
}
```

E5.9 Translate the following *WHERE* clause into a SPARQL algebra expression, without expanding prefixes in CURIEs.

```
WHERE {
      ?x :radius ?r .
      OPTIONAL {
            ?x :label ?label .
            FILTER (lang(?label) = "en")
      }
}
```

E5.10 Translate the following *WHERE* clause into a SPARQL algebra expression.

```
WHERE {
      ?x :satellite ?y .
      {
            ?y :label ?label .
      } UNION {
            ?y :name ?label .
      }
}
```

E5.11 Translate the following WHERE clause into a SPARQL algebra expression.

```
WHERE {
      ?x :radius ?r .
      BIND (?r * 2 AS ?diameter)
```

}

E5.12 Consider the following RDF document with a graph G describing celestial bodies in our solar system.

```
@prefix : <astro#> .

:Sonne :radius 696342 ; :satellite :Merkur, :Venus, :Erde, :Mars .
:Merkur :radius 2439.7 .
:Venus :radius 6051.9 .
:Erde :radius 6372.8 ; :satellite :Mond .
:Mars :radius 3402.5 ; :satellite :Phobos, :Deimos .
:Mond :radius 1737.1 ; :label "Mond"@de, "Moon"@en .
:Phobos :label "Phobos" .
:Deimos :label "Deimos" .
```

The property with URI *:radius* describes the equatorial radius of the celestial body. The property with URI *:satellite* denotes that the celestial body in object position is a satellite of the celestial body in subject position. The property with URI *:label* provides a human-readable label. Consider the BGP P:

```
:Erde :satellite ?x .
?x :label "Moon"@en .
```

Consider the algebra expression *M = BGP(P).* What is the solution sequence Ω resulting from evaluating M on graph G, i.e., *eval(G, M)*?

E5.13 Consider the BGP *P:*

```
  :Erde :satellit ?x .
  ?x :label "Moon" .
```

Consider the algebra expression *M = BGP(P).* What is the solution sequence Ω resulting from evaluating *M* on graph *G* from E 5.12?

E5.14 Let P be:

```
  :Sonne :satellite ?x .
```

Let F be:

```
 (?x != :Merkur && ?x != :Mars)
```

Consider the algebra expression *M = FILTER(F, BGP(P)).* What is the solution sequence Ω resulting from evaluating M on graph G from E 5.12?

E5.15 Let P1 be

```
   ?s1 :radius 6051.9 .
```

Let P2 be

```
  ?s2 :radius 6372.8 .
```

Consider the algebra expression *M = UNION(BGP(P1), BGP(P2)).* What is the solution sequence Ω resulting from evaluating *M* on graph *G* from E 5.12?

E5.16 Let P1 be:

```
?x :radius ?r .
```

Let P2 be:

```
?x :label ?label .
```

Let *M* be *LEFTJOIN(BGP(P1), BGP(P2), true).* What is the solution sequence Ω resulting from evaluating *M* on graph *G* from E 5.12?

E5.17 OPTIONAL vs. UNION

Data:

```
:s :p :o1 .
:o1 :q :o2 .
:s :p :o2 .
```

```
WHERE {
   { :s :p ?x . }
   UNION
   { :s :p ?x .
   ?x :q :o . }
}
```

```
WHERE {
   :s :p ?x .
   OPTIONAL { ?x :q :o . }
}
```

Different solutions!

Learning Goals

G 5.1 Explain the formal definition of basic graph pattern matching and show how to match basic graph patterns to graphs.

G 5.2 Translate a given SPARQL WHERE clause, including UNION, OPTIONAL, FILTER and BIND AS clauses, to a SPARQL algebra expression.

G 5.3 Given a query, explain the handling of the RDF dataset with FROM and FROM NAMED clauses in conjunction with GRAPH.

G 5.4 Evaluate a SPARQL algebra expression on a given RDF dataset and specify the solutions of the entire algebra expression and also of partial expressions.

G 5.5 Generate the final results to a SPARQL abstract query, taking into account the solution sequence of the graph pattern algebra expression and the query form.