Exercise Sheet 9: SPARQL and OWL

Maximilian Marx, Markus Krötzsch

Knowledge Graphs, 2026-01-14, Winter Term 2025/2026

Exercise 9.1. Consider the RDF graph G:

```
eg:x eg:edge eg:x ;
     eg:value 1 .
eg:y eg:edge eg:x, eg:y ;
     eg:value 2 .
eg:z eg:edge eg:x, eg:y, eg:z ;
     eg:value 3 .
```

Evaluate the following expression of the SPARQL algebra over G:

```
Group(⟨ ?s ⟩,
  LeftJoin(
    BGP_G(?s eg:value ?v),
    BGP_G(?s eg:edge ?o),
    ?s != ?o)
)
```

The semantics of grouping are as follows: Consider some list of expressions $\Phi = \langle \varphi_1, \ldots, \varphi_n \rangle$. For a solution mapping μ, define $\Phi(\mu)$ as the list $\langle \varphi_1(\mu), \ldots, \varphi_n(\mu) \rangle$ of values obtained by evaluation these expressions for the bindings of μ. Then

$$\mathsf{Group}(\Phi, M) = \left\{ \Phi(\mu) \mapsto \{\mu' \in M \mid \Phi(\mu') = \Phi(\mu)\} \mid \mu \in M \right\}$$

Exercise 9.2. Which of the following graph patterns are expressible in SPARQL? Explain your answer by either giving a SPARQL query or by arguing why there is none.

1. Find nodes that are connected by an `eg:edge` path of length ≥ 100
2. Find nodes that are connected by an `eg:edge` path of length ≤ 100
3. Find nodes that are connected by an `eg:edge` path of length $\neq 100$
4. Find nodes that are not connected by an `eg:edge` path of length 100
5. In a graph with a `eg:parent` property, find nodes with a common ancestor
6. In a graph with a `eg:parent` property, find nodes that are cousins (of any degree)
7. Find nodes that are connected by `eg:propA` but not by `eg:propB`
8. Find nodes that are connected by an `eg:propA` path, but not by an `eg:propB` path
9. Find nodes that are connected by a path of nodes as in 7
10. Find nodes connected by an arbitrary path
11. Find nodes connected by an arbitrary path of even length
12. Check if the graph contains an even number of nodes

Exercise 9.3. Consider the following ontology O:

```
SubClassOf(
  eg:Road
  ObjectHasValue( eg:leadsTo eg:rome )
)
SubClassOf( eg:Highway eg:Road )
ClassAssertion( eg:Highway eg:a1 )
```

Find an interpretation $\mathcal{I}$ such that $\mathcal{I} \models O$.

Exercise 9.4. Show that the following assertional axioms are syntactic sugar in OWL:

1. `ClassAssertion(` C e `)`
2. `ObjectPropertyAssertion(` P e f `)`
3. `DataPropertyAssertion(` P e ℓ `)`
4. `EquivalentObjectProperties(` P Q `)`
5. `SameIndividual(` e f `)`
6. `DifferentIndividuals(` e f `)`
7. `DisjointClasses(` C D `)`
8. `NegativeObjectPropertyAssertion(` P e f `)`

Exercise 9.5. For each of the following axioms, decide whether they are syntactic sugar in OWL and show how they can be expressed using other axioms, or give a reason why they cannot be expressed otherwise.

1. `SymmetricObjectProperty(` P `)`
2. `AsymmetricObjectProperty(` P `)`
3. `DisjointObjectProperty(` P Q `)`
4. `IrreflexiveObjectProperty(` P `)`
5. `FunctionalObjectProperty(` P `)`
6. `TransitiveObjectProperty(` P `)`