

## Exercise Sheet 8: Expressivity of SPARQL

Maximilian Marx, Markus Krötzsch

Knowledge Graphs, 2019-12-10, Winter Term 2019/2020

**Exercise 8.1.** 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  $\geq 100$
2. Find nodes that are connected by an `eg:edge` path of length  $\leq 100$
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 8.2.** Given a formula  $\varphi$  of propositional logic, show how to decide  $\varphi \in \mathbf{SAT}$  using a SPARQL query that does not contain any BGPs.

**Exercise 8.3.** Find a family of SPARQL queries that produce solutions where a variable name is mapped to a value that requires an exponential number of characters to write down (measured in the size of the query and RDF graph). What can you say about the growth of the result's size with respect to the size of the RDF graph when keeping the query fixed?

**Exercise 8.4.** Which of the following QBF are satisfiable? Why?/Why not?

1.  $\exists p_1. p_1$
2.  $\forall p_1. p_1$
3.  $\exists p_1. \perp$
4.  $\forall p_1. \exists p_2. p_2 \rightarrow p_1$
5.  $\forall p_1. \exists p_2. \forall p_3. (p_1 \vee p_2) \wedge p_3$
6.  $\forall p_1. \forall p_2. \exists p_3. \forall p_4. (p_1 \wedge p_2 \rightarrow p_4) \vee \neg p_3$