Exercise Sheet 8: Expressivity of SPARQL

Maximilian Marx, Markus Krötzsch

Knowledge Graphs, 2025-01-07, Winter Term 2024/2025

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 ≥ 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 8.2. Can you write a SPARQL query for the Wikidata Query Service[1] that finds all persons related to Q1339 (“Johann Sebastian Bach”) by a path going through P40 (“child”), P25 (“mother”), or P26 (“spouse”) edges, such that every person on this path has a statement for property P1303 (“instrument”) with value Q1444 (“Organ”)? How?/Why not?

Does anything change if you relax the restriction on all persons on the path and only require that they have a statement for property P1303 (“instrument”) with an arbitrary value?

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?

[1] https://query.wikidata.org

Exercise 8.4. Consider the Datalog program P

$$\begin{aligned}
\mathsf{Parent}(x, y) &:\!- \mathsf{father}(x, y) \\
\mathsf{Parent}(x, y) &:\!- \mathsf{mother}(x, y) \\
\mathsf{Ancestor}(x, y) &:\!- \mathsf{Parent}(x, y) \\
\mathsf{Ancestor}(x, z) &:\!- \mathsf{Parent}(x, y), \mathsf{Ancestor}(y, z) \\
\mathsf{Result}(y) &:\!- \mathsf{Ancestor}(\mathrm{alice}, y)
\end{aligned}$$

and the facts

$$\begin{aligned}
\mathsf{mother}(\mathrm{alice}, \mathrm{barbara}) \qquad & \mathsf{father}(\mathrm{alice}, \mathrm{bob}) \\
\mathsf{mother}(\mathrm{barbara}, \mathrm{christine}) \qquad & \mathsf{father}(\mathrm{barbara}, \mathrm{charles}) \\
\mathsf{mother}(\mathrm{dave}, \mathrm{emmy}) \qquad & \mathsf{father}(\mathrm{bob}, \mathrm{dave})
\end{aligned}$$

Compute all query results for $\langle P, \mathsf{Result} \rangle$.