Exercise Sheet 10: Datalog in Practice

Maximilian Marx, Markus Krötzsch

Knowledge Graphs, 2025-01-21, Winter Term 2024/2025

Exercise 10.1. Solve Exercise 2.5 using Nemo[1]: Write a Datalog program extracting, from the `coauthors.nt.gz` graph[2], the *connected component* containing <http://dblp.uni-trier.de/pers/s/Studer:Rudi>, i.e, extract the induced subgraph that

- contains <http://dblp.uni-trier.de/pers/s/Studer:Rudi>,
- contains all nodes reachable from <http://dblp.uni-trier.de/pers/s/Studer:Rudi> by some path, and
- contains all edges that are present in the full graph between these nodes.

Note that, while an RDF graph is inherently directed, edges in `coauthors.nt.gz` are symmetric, i.e., the graph is essentially undirected.
Hint: `authorship-snippet.nt.gz`[2] contains <http://dblp.uni-trier.de/pers/s/Studer:Rudi> and can be used for testing.
Hint: You can use, e.g., https://media.githubusercontent.com/media/knowsys/Course-Knowledge-Graphs/refs/heads/main/data/dblp/authorship-snippet.nt.gz as the resource in the Nemo web UI to avoid downloading the graph onto your computer.

Exercise 10.2. Use Nemo[1] to to solve the query from Exercise 8.2 using a Datalog program and the Wikidata SPARQL endpoint[3]: find 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").

Exercise 10.3. DBpedia is a knowledge graph based on information extracted from Wikipedia. Use Nemo[1] and the Wikidata[3] and DBpedia[4] SPARQL endpoints to integrate and compare the *parent* relations from DBpedia and Wikidata: Use SPARQL queries to fetch both *parent* relationships (for Wikidata, you can restrict to items with articles on English Wikipedia). Make sure your queries include a common feature that can be used for integration, e.g., the URL of the related Wikipedia article.[5] Lastly, use rules to compute the total number of (unique) relations found in both graphs, in Wikidata only, and in DBpedia only.

[1] https://tools.iccl.inf.tu-dresden.de/nemo/next
[2] https://github.com/knowsys/Course-Knowledge-Graphs/tree/main/data/dblp
[3] https://query.wikidata.org
[4] https://dbpedia.org/sparql
[5] DBpedia still stores `http` URLs, whereas Wikidata uses `https`. You can use `SUBSTR` and `BIND` in your SPARQL queries to align such URLs.

Exercise 10.4. Which of the following graph patterns are expressible as (stratified) Datalog queries (all predicates mentioned are binary)? Explain your answer by either giving a Datalog query or by arguing why there is none.

1. Find nodes that are connected by an edge path of length ≥ 100
2. Find nodes that are connected by an edge path of length ≤ 100
3. Find nodes that are connected by an edge path of length $\neq 100$
4. Find nodes that are not connected by an edge path of length 100
5. In a graph with a parent predicate, find nodes with a common ancestor
6. In a graph with a parent predicate, find nodes that are cousins (of any degree)
7. Find nodes that are connected by predA but not by predB
8. Find nodes that are connected by an predA path, but not by an predB 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