

Exercise Sheet 11: Cypher

Maximilian Marx, Markus Krötzsch
Knowledge Graphs, 2020-01-14, Winter Term 2019/2020

Exercise 11.1. Consider the following cypher query from example 11.17:

```
MATCH (prof {occupation: "Professor" }) -[:SPOUSE] -()  
MATCH (prof) -[:HAS_CHILD] ->(child)  
RETURN prof, count(child)
```

Do the query results change if `count(DISTINCT child)` is used instead?

Exercise 11.2. Which of the following graph patterns are expressible in Cypher? Explain your answer by either giving a Cypher 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` relationship type, find nodes with a common ancestor
6. In a graph with a `:PARENT` relationship type, find nodes that are cousins (of any degree)
7. Find nodes that are connected by `:PROP_A` but not by `:PROP_B`
8. Find nodes that are connected by a `:PROP_A` path, but not by a `:PROP_B` 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 11.3. Neo4j provides numerous extension over the openCypher language, including the list predicate functions `all`¹ and `any`², that check whether a condition is true for all elements (or any element, respectively) of a list.

Show that these two functions are sufficient to encode **TRUEQBF** in a Cypher query. What can you say about the complexity of answering Cypher queries?

¹<https://neo4j.com/docs/cypher-manual/current/functions/predicate/#functions-all>

²<https://neo4j.com/docs/cypher-manual/current/functions/predicate/#functions-any>

Exercise 11.4. Download and install Neo4j³, or use the Neo4j Sandbox⁴.

Use the `:play movies` command to load the movie example data set. Write Cypher queries that find

1. all actors who have co-starred in two movies,
2. for every actor, the length of the shortest path (along any **relationship type**) connecting this actor to Kevin Bacon,
3. pairs of persons and movies where the person has at least two **relationships** of distinct **relationship types** to the movie, and
4. the number of undirected triangles along any **relationship type**. How often is each triangle counted?
5. persons that have acted in or directed movies they wrote,
6. the top 5 movies by the number of actors, and
7. the top 5 actors that have co-starred most often with Keanu Reeves.

³<https://neo4j.com/download/>

⁴<https://neo4j.com/sandbox-v3/>