Universidade de Nova de Lisboa **MIEI**

Faculdade de Ciências e Tecnologia **November 10th, 2018**

Data Modelling
Modelação de Dados

Test
Duration: 2 hours

Group 1 [6]

Consider the Movies Database from Neo4J tutorial, that includes: node labels Movie, Person; relationships types ACTED_IN, DIRECTED, FOLLOWS, PRODUCED, REVIEWED, WROTE; and properties: **title, born**, **name**, **rating**, **released**, **roles**, **summary**, **tagline**.

a) **[2]** Write a cypher query to find all the actors that participated in the movies "Star Wars" and "Indiana Jones".
b) **[2]** Write a cypher query to present list all the ratings of all movies, sorted by movie name. All movies should be listed.
c) **[2]** Write a cypher query to list the names of persons who have written more than 3 movies.

Group 2 [7]

Consider the following N3 data of the RDF graph **O**:

```
@prefix : < http://foo.ex/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix org: <http://www.w3.org/ns/org#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

:c1 org:hasMember :c3; org:hasUnit :c2; skos:prefLabel "Company 1" .
:c3 org:hasMember :c1, :c2 .
[ org:headOf :c1; org:memberOf :c3 ] foaf:name "John" .
[ org:memberOf :c2 ] .
:p2 org:memberOf :c3 ; foaf:name "Mary" .
```

2a) [1] Draw the corresponding RDF graph.

2b) [3] Consider the next four RDF graphs where the default prefix `:` is associated to IRI `< http://foo.ex/>`

1. `_:x org:headOf _:y .`
`_:y org:hasMember _:y`
`_:y skos:prefLabel _:z .`
	2. `_:x org:memberOf _:z .`
	`_:y org:memberOf _:z .`
	`_:z org:hasMember _:w .`

3. `_:x org:headOf _:z .`
`_:y org:hasMember _:z .`
`_:z org:hasMember _:y .`
	4. `_:x org:memberOf _:y .`
	`_:x org:memberOf _:z .`
	`_:x foaf:name _:w .`

From the graphs specified previously indicate the ones which are simply entailed by graph **O**. Justify in detail by using any of the methods studied for RDF simple entailment.

2c) [3] The initial data in O is extended with the following schema information where the rdfs prefix is associated with the usual URI identifying RDF Schema:

```
1. org:memberOf rdfs:domain foaf:Agent .
2. org:memberOf rdfs:range org:Organization .
3. org:hasMember rdfs:domain org:Organization .
4. org:hasMember rdfs:range foaf:Agent.
5. org:headOf rdfs:subPropertyOf org:memberOf .
6. org:hasUnit rdfs:domain org:Organization .
7. org:hasUnit rdfs:range org:OrganizationalUnit .
8. org:OrganizationalUnit rdfs:subClassOf org:Organization .
9. org:Organization rdfs:subClassOf foaf:Agent .
```

Using the inference rules for RDF Schema entailment, check whether the following graph is entailed by the initial RDF graph **O** when extended by the schema information. Justify your answer.

```
[ a foaf:Agent ] org:memberOf [ a foaf:Agent; skos:prefLabel [] ].
```

Group 3 [7]

3a) [2] Present the solutions to the following SPARQL query when applied to graph **O of the previous group**, justifying the obtained solutions using the studied SPARQL algebra. Treat blank nodes in **O** as if they were distinct and different IRIs.

```
@prefix org: <http://www.w3.org/ns/org#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
SELECT ?x ?y ?z
WHERE { ?x foaf:name ?z
        OPTIONAL
        { { ?x org:memberOf ?y } UNION { ?y org:hasMember ?x } }
      } .
```

In the questions below you should produce queries that would produce results for RDF graphs of the form of the one in Group 2. Assume that entailment mode is simple but the graph has been previously closed with all the triples obtained with RDFS closure rules.

3b) [2] Construct a SPARQL query that retrieves all the organizations and their head's and/or members, and their names whenever they exist (organizations, head and members).

3c) [2] Present a SPARQL query to obtain the number of members who have a name, per organization, as well as the organization's preferred label.

3d) [1] Construct a SPARQL presenting for each organization their direct or indirect member organizations. It is enough to present the list of answers containing a column with the organization IRI and another column with the suborganization. It is suggested to user property paths to answer to this question.

THE END

Annex

Let **aaa, bbb,...** be IRIs, **uuu, vvv, ...** be blank nodes or IRIs, and **xxx, yyy, ...** be blank nodes, IRI references or literals.

Simple Entailment rules

Rule Name	If S contains	then add
se1	uuu aaa xxx .	uuu aaa _:nnn . where _:nnn designates a blank node allocated to xxx by rules se1 or se2.
se2	uuu aaa xxx .	_:nnn aaa xxx . where _:nnn designates a blank node allocated to uuu by rules se1 or se2.

RDF Entailment rules

Name	If S contains	then add
GrdfD1	xxx aaa "sss"^^ddd . for ddd in D	"sss"^^ddd rdf:type ddd .
rdfD2	uuu aaa yyy .	aaa rdf:type rdf:Property .

RDFS Entailment rules

	If S contains:	then S RDFS entails recognizing D:
rdfs1	any IRI aaa in D	aaa `rdf:type rdfs:Datatype .`
rdfs2	aaa `rdfs:domain` xxx . uuu aaa yyy .	uuu `rdf:type` xxx .
rdfs3	aaa `rdfs:range` xxx . uuu aaa vvv .	vvv `rdf:type` xxx .
rdfs4a	uuu aaa xxx .	uuu `rdf:type rdfs:Resource .`
rdfs4b	uuu aaa vvv.	vvv `rdf:type rdfs:Resource .`
rdfs5	uuu `rdfs:subPropertyOf` vvv . vvv `rdfs:subPropertyOf` xxx .	uuu `rdfs:subPropertyOf` xxx .
rdfs6	uuu `rdf:type rdf:Property .`	uuu `rdfs:subPropertyOf` uuu .
rdfs7	aaa `rdfs:subPropertyOf` bbb . uuu aaa xxx .	uuu bbb xxx .
rdfs8	uuu `rdf:type rdfs:Class .`	uuu `rdfs:subClassOf rdfs:Resource .`
rdfs9	uuu `rdfs:subClassOf` xxx . vvv `rdf:type` uuu .	vvv `rdf:type` xxx.
rdfs10	uuu `rdf:type rdfs:Class .`	uuu `rdfs:subClassOf` uuu .
rdfs11	uuu `rdfs:subClassOf` vvv . vvv `rdfs:subClassOf` xxx .	uuu `rdfs:subClassOf` xxx .
rdfs12	uuu `rdf:type` `rdfs:ContainerMembershipProperty .`	uuu `rdfs:subPropertyOf rdfs:member .`
rdfs13	uuu `rdf:type rdfs:Datatype .`	uuu `rdfs:subClassOf rdfs:Literal .`

If needed you can use the rdfs7x rule:

rdfs7x	aaa rdfs:subPropertyOf vvv . uuu aaa yyy .	uuu vvv yyy .