PAPER CODE NO.	EXAMINER: Dr Valentina Tamma	Tel. No. 795 4246
COMP318	DEPARTMENT: Computer Science	

UNIVERSITY OF LIVERPOOL

SECOND SEMESTER EXAMINATIONS 2023/24

Ontologies and Semantic Web

TIME ALLOWED : TWO AND A HALF Hours

INSTRUCTIONS TO CANDIDATES

Answer **ALL** questions in Section A.
Answer **TWO** questions in Section B.

All questions in Section B have the same maximum mark. If you attempt to answer more questions than the required number of questions (in any section), the marks awarded for the excess questions answered will be discarded (starting with your lowest mark).

Calculators are permitted.

UNIVERSITY OF LIVERPOOL

SECTION A

Attempt ALL FIVE questions from this section. Section A is worth 50 marks.

A1 **(a)** Describe what a Knowledge Graph (KG) is, and provide two possible definitions of KGs. (**6 marks**)

(b) KG schemas are often modelled as linked data (e.g. DBpedia). Briefly explain the second principle for publishing data as linked data. (**4 marks**)

A2 Describe RDF simple entailment and define what does it mean for a graph G2 to be simply entailed from a graph G1 . (**10 marks**)

A3 Describe the meaning of the following restrictions in OWL and illustrate each of them with an example:

a. `owl:allValuesFrom X` (**3 marks**), **1** of which is for the example

b. `owl:someValuesFrom X` (**3 marks**), **1** of which is for the example

c. `owl:hasValue X` (**4 marks**), **1** of which is for the example

A4 Provide and explain the definition of *ontology*. Explain what *formal* means in the context of this definition, and the importance of a formal ontology. **(10 marks)**

A5 Describe the type of queries supported by SPARQL. (**10 marks**)

UNIVERSITY OF LIVERPOOL

SECTION B

Attempt TWO questions from this section. Each question is worth 25 marks. Credit will be given for the best 2 answers only.

B1 Consider the description, in terms of classes and properties, of the following two ontologies O1 and O2:

O1 : **Classes:** Institution, Process, GeographicalArea;
Properties: hasTitle, hasAction, locatedIn

O2 : **Classes:** Company, Operation, Location;
Properties: companyName, hasImplementation, isSituatedIn

(a) Discuss whether you would choose *syntactic matching* or *semantic matching with external dictionary* (e.g. WordNet) for identifying potential correspondences (matches) between entities, classes, or properties in O1 and O2.

(b) Identify potential correspondences (matches) between entities, classes, or properties in O1 and O2. Aim to generate five correct correspondences, i.e. those matches that are most similar using your chosen technique. (**5 marks**)

Assume that some expert has defined a gold standard made of 7 correspondences.

(c) Provide the formulation of *precision* and *recall* in the context of ontology alignment. Provide an explanation for the given formulae. (**6 marks**)

(d) Calculate *precision* and *recall* of the alignment you computed, showing the details of your calculations. (**4 marks**)

(e) Provide the formulation and calculate the *F-measure*. Why is the F-measure a more accurate measure of performance and why is it not sufficient to pick either precision or recall and use only that? (**6 marks**)

B2 Consider the following RDF graph G about hospitals and their departments. The graph is expressed in Turtle, with identifiers corresponding to URIs whilst `_:n` denotes a blank node.

```
a:SpecialistClinic  rdfs:subClassOf  a:Hospital .
b:ClatterbridgeCC   rdf:type         a:SpecialistClinic ;
                    rdf:label        "Clatterbridge Clinic" ;
                    a:hasService     b:Surgery ;
                    a:hasService     b:Radio .
b:AlderHayCH        rdf:type         a:Hospital ;
                    rdf:label        "Alder Hay Children Hospital ;
                    a:hasService     b:PS ;
                    a:hasService     b:Radio .
b:PS                rdf:type         a:PediatricSurgery .
a:PediatricSurgery  rdf:subClassOf   a:Surgery .
b:Radio             rdf:type         a:Radiology .
b:Surgery           rdf:type         a:Surgery .
```

(a) Identify the classes in the graph and explain why you think they are classes. (**5 marks**)

(b) Given the following RDF graph P, is P entailed by G? Justify your answer. (**10 marks**)

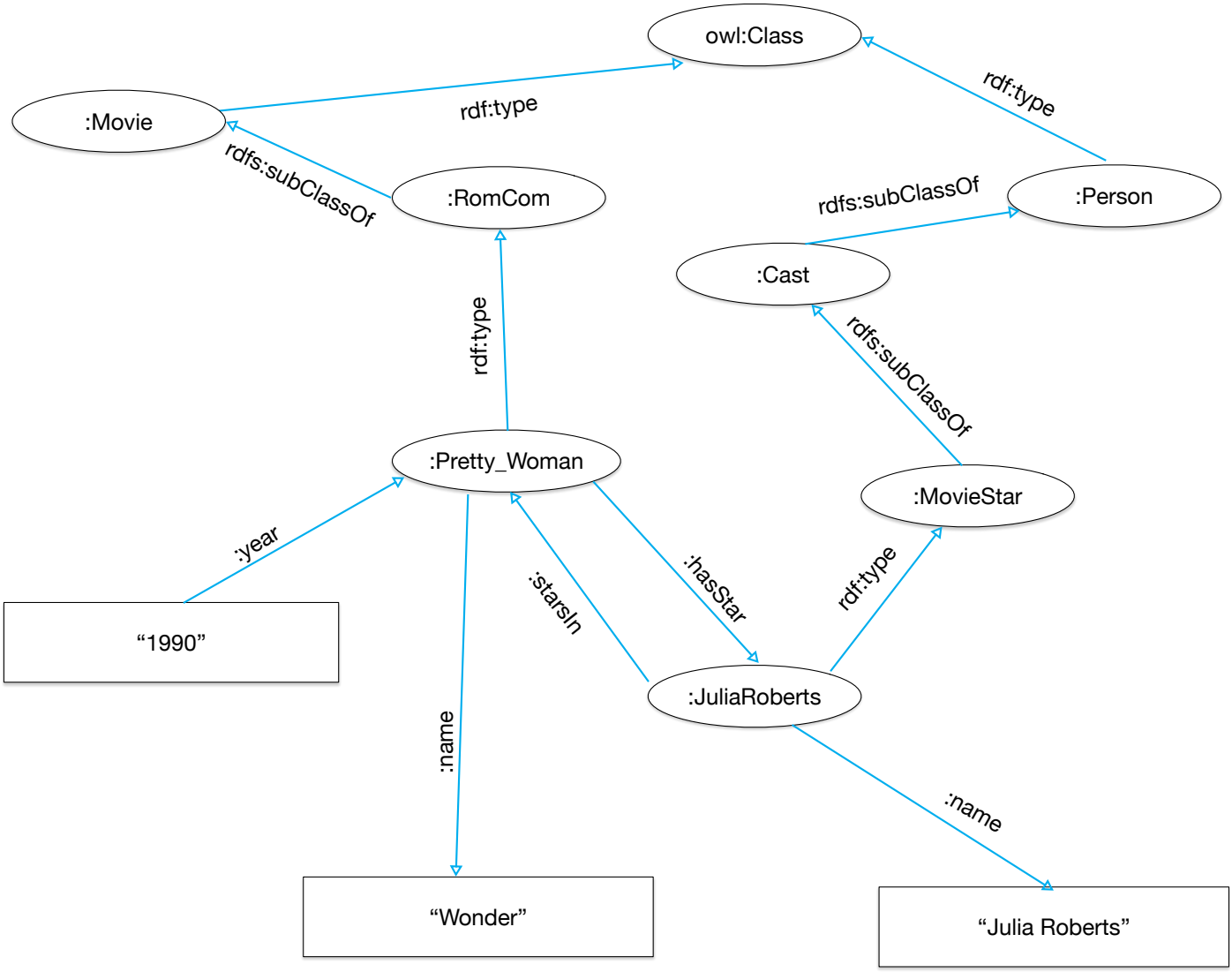
```
_:x  rdf:type      a:SpecialistClinic .
_:x  a:hasService  _:y .
_:y  rdf:type      a:Surgery .
```

(c) List the triples that will be returned by the query below. Is the query returning all the resources that are (either directly asserted or deduced to be) hospitals? Explain your answer. (**5 marks**)

```
SELECT ?hospital
WHERE {?hospital rdf:type a:Hospital .}
```

(d) If you answered no to the previous question write a SPARQL query that returns ALL hospitals. Otherwise modify the previous query to specify also the label of the hospital. (**5 marks**)

B3 Consider the following graph expressing some facts about movies and actors:



(a) Serialise the graph (without any inferred nodes or edges) using Turtle. You can omit prefix definitions. (**4 marks**)

(b) Add statements that can be inferred to the serialised document, and identify them by placing an asterisk or star next to them. Omit trivial statements (e.g. every class is an owl:Class). (**6 marks**)

(c) Underline, or clearly differentiate, the labels of the nodes that represent instances. (**4 marks**)

(d) Write appropriate domain and range assertions for the properties `:year`, `:name`, `:hasStar` and `:starsIn` in Turtle. (**5 marks**)

(e) Write assertion(s) that make the properties `:hasStar` and `:starsIn` be inverse of one another. (**6 marks**)

For your convenience here are the RDFS-entailment patterns

RDFS entailment patterns.

	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 . yyy aaa zzz .	yyy `rdf:type` xxx .
rdfs3	aaa `rdfs:range` xxx . yyy aaa zzz .	zzz `rdf:type` xxx .
rdfs4a	xxx aaa yyy .	xxx `rdf:type rdfs:Resource .`
rdfs4b	xxx aaa yyy.	yyy `rdf:type rdfs:Resource .`
rdfs5	xxx `rdfs:subPropertyOf` yyy . yyy `rdfs:subPropertyOf` zzz .	xxx `rdfs:subPropertyOf` zzz .
rdfs6	xxx `rdf:type rdf:Property .`	xxx `rdfs:subPropertyOf` xxx .
rdfs7	aaa `rdfs:subPropertyOf` bbb . xxx aaa yyy .	xxx bbb yyy .
rdfs8	xxx `rdf:type rdfs:Class .`	xxx `rdfs:subClassOf rdfs:Resource .`
rdfs9	xxx `rdfs:subClassOf` yyy . zzz `rdf:type` xxx .	zzz `rdf:type` yyy .
rdfs10	xxx `rdf:type rdfs:Class .`	xxx `rdfs:subClassOf` xxx .
rdfs11	xxx `rdfs:subClassOf` yyy . yyy `rdfs:subClassOf` zzz .	xxx `rdfs:subClassOf` zzz .
rdfs12	xxx `rdf:type rdfs:ContainerMembershipProperty .`	xxx `rdfs:subPropertyOf rdfs:member .`
rdfs13	xxx `rdf:type rdfs:Datatype .`	xxx `rdfs:subClassOf rdfs:Literal .`