EXAMINER: Dr Valentina Tamma
DEPARTMENT: Computer Science
Tel. No. 795 4246

UNIVERSITY OF LIVERPOOL

SECOND SEMESTER EXAMINATIONS 2022/23

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.

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

A1 **(a)** Describe what a Knowledge Graph (KG) is, provide its formal definition and include at least one of its characteristics. (**5 marks**)

(b) KG schemas are often modelled as linked data (e.g. DBpedia). List and briefly explain the first 3 principles for publishing data as linked data. (**3 marks**)

(c) How can the use of RDF facilitate connecting resources in different knowledge graphs, thus creating a single global data space? (**2 marks**)

A2 **(a)** Explain why *reification* can be used as a way to represent n-ary relationships in RDF, and discuss the vocabulary used to define reified statements. (**4 marks**)

(b) What are the other ways to define n-ary relationships in RDF? Illustrate your answers by means of explaining how you can model the statement "*The University of Liverpool had 28,680 student in 2022*". (**6 marks**)

A3 In this question you will need to express the three phrases below as one or more OWL axioms (using its Manchester syntax). To do this, you may use the following class and property names, without namespaces:

- Classes: `Computer`, `HPComputer`, `CPU`, `Core`, `RAM`, `Motherboard`
- Properties: `hasPart`, `hasCore`, `isConnectedTo`
- Individuals: `hpCPU`

Note also the following constraints:

- A computer can have parts, e.g. a CPU, and a CPU can have cores;
- A CPU can be connected to, amongst other things, a motherboard. However, CPUs are not parts of motherboards, they are only connected to motherboards;

1. All computer parts are either motherboards or things connected to motherboards;

   (**3 marks**)
2. A HPComputer is something that either has more than 2 CPUs or a CPU with more than 8 cores; (**4 marks**)
3. Nothing is connected to itself. (**3 marks**)

A4 In ontology alignment a correspondence creates a relation between different entities in an ontology (classes, properties, or individuals). Describe the types of correspondence relations that can exist between classes belonging to different ontologies to be matched, and provide an example for each of these relations. (**10 marks**)

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

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 Two ontology alignment systems, $\mathcal{A}^1$ and $\mathcal{A}^2$ are being compared. Both are compared to a gold standard alignment handcrafted by an expert $\mathcal{R}$ composed of 215 correspondences. $\mathcal{A}^1$ returns 158 correspondences, of which 123 are present in $\mathcal{R}$. $\mathcal{A}^2$ returns 88 correspondences, of which 79 are contained in $\mathcal{R}$.

(a) Provide the formulation of *precision* and *recall* in the context of ontology alignment. Provide an explanation for the given formulae. (**5 marks**)

(b) Calculate *precision* and *recall* for both $\mathcal{A}^1$ and $\mathcal{A}^2$, showing the details of your calculations. (**10 marks**)

(c) Calculate the *F-measure* for each of the system. 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? (**10 marks**)

B2 Consider the following RDF graph G, expressed in Turtle, about companies, their suppliers and customers. Note that the triples have been numbered to improve readability, `rdf` and `rdfs` are the usual namespaces. The following namespaces are also used;

- @prefix r: <http://www.exam.org/business/>
- @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
- @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

```
1   r:Customer     rdfs:subClassOf     r:Company .
2   r:Supplier     rdfs:subClassOf     r:Company .
3   r:hasCustomer  rdfs:subPropertyOf  r:hasBusinessPartner.
4   r:hasSupplier  rdfs:subPropertyOf  r:hasBusinessPartner.
5   r:hasCustomer  rdfs:domain         r:Supplier .
6   r:hasCustomer  rdfs:range          r:Customer .
7   r:hasSupplier  rdfs:domain         r:Customer .
8   r:hasSupplier  rdfs:range          r:Supplier .
9   r:bestCakes    r:hasCustomer       r:myCakeShop .
10  r:bestCakes    r:hasSupplier       r:CakeSupplies .
11  r:myCakeShop   r:hasSupplier       _:n .
12  _:n            r:hasSupplier       r:CakeSupplies .
```

(a) Given the following triples in the list, determine whether these can be derived from the graph G. If they can be derived, then provide a derivation using the RDFS and simple entailment rules. If the triples cannot be derived, provide a short explanation of why such a derivation cannot exist.

i. `r:CakeSupplies a r:Company .` (**5 marks**)

ii. `r:hasBusinessPartner rdfs:Range r:Company .` (**5 marks**)

iii. `r:myCakeShop r:hasSupplier r:bestCakes .` (**5 marks**)

iv. `r:myCakeShop r:hasBusinessPartner _:y .`
`_:y a r:Customer .` (**5 marks**)

(b) Intuitively, if a company A is a customer of another company B, then company B is a supplier for company A, and vice versa. Is it possible to specify in RDFS such a relationship between `r:hasSupplier` and `r:hasCustomer`? Are there any other ways to achieve this? (**5 marks**)

UNIVERSITY OF LIVERPOOL

B3 Let us assume that $\mathcal{I}$ refers to the following interpretation:

- $\Delta^{\mathcal{I}} = \{charles, diana, william, harry\}$
- $Man^{\mathcal{I}} = \{charles, william, harry\}$
- $Woman^{\mathcal{I}} = \{diana\}$
- $King^{\mathcal{I}} = \{charles\}$
- $marriedTo^{\mathcal{I}} = \{< diana, charles >, < charles, diana >\}$
- $hasFather^{\mathcal{I}} = \{< william, charles >, < harry, charles >\}$
- $hasMother^{\mathcal{I}} = \{< william, diana >, < harry, diana >\}$

(a) Provide the interpretation of the following concepts expressed in OWL:

i. `owl:Thing` (**1 marks**)

ii. King **or** marriedTo **some** King (**5 marks**)

iii. King **or** marriedTo **only** King (**5 marks**)

(b) For each of the following axioms, determine what is their interpretation in $\mathcal{I}$. Are there any interpretations for which the axioms do not hold?

i. King **subClassOf** marriedTo **only** Woman (**7 marks**)

ii. King **subClassOf** marriedTo **only** owl:Thing (**7 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 .`