PAPER CODE NO. EXAMINER : Dr V. Tamma
COMP318 DEPARTMENT : Computer Science Tel. No. 54246

UNIVERSITY OF
LIVERPOOL

MAY 2007 EXAMINATIONS

Bachelor of Science : Year 3
Master of Engineering : Year 3
Master of Science : Year 1
No qualification aimed for : Year 1

Advanced Web Technologies

TIME ALLOWED : Two and a Half Hours

INSTRUCTIONS TO CANDIDATES

Attempt all questions in Section A.
Attempt **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).

UNIVERSITY OF LIVERPOOL

SECTION A

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

1. Many techniques exist for ranking the result set of search engines, most notably content based ranking (with different weightings such as TF/IDF), and ranking algorithms such as PageRank. Compare and contrast these two main approaches. **(10 marks)**

2. Describe the two ways for defining the structure of an XML document by comparing and contrasting their expressivity. Provide a short example for each of them. **(10 marks)**

3. Describe in terms of its components the protocol to access Web services and interchange information in a Web environment. **(10 marks)**

4. An ontology is a "*formal, explicit specification of a shared conceptualisation*". Discuss this definition of ontology, what it is meant here by *formal*, and the benefit of having formal ontologies. Discuss also the main benefit of shared ontologies. **(10 marks)**

5. Define the meaning of *reification* and explain what it is used for. Illustrate its use with an example. **(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.

1. Let us consider the RDF graph G:

```
:united-kingdom rdf:type :Country
:Country rdfs:subClassOf :GeographicalEntity
_:x :primeMinisterOf :united-kingdom
:primeMinisterOf rdfs:domain :primeMinister
:primeMinisterOf rdfs:range :Nation
:primeMinisterOf rdfs:subPropertyOf :leaderOf .
```

Determine if the following fragments are RDFS-entailed by G, and explain why. **(25 marks)**

a
```
:Nation rdf:type rdfs:Class
:leaderOf rdf:type rdf:Property
_:u :leaderOf :new-zealand
```

b
```
:united-kingdom rdf:type :GeographicalEntity
:leaderOf rdfs:range :Nation
```

c
```
_:x rdfs:subPropertyOf rdf:type
:PrimeMinister rdfs:subClassOf :civilServant
```

d
```
:primeMinisterOf rdfs:domain _:i
_:x rdfs:subPropertyOf :leaderOf
_:y rdf:type :GeographicalEntity
_:l :primeMinisterOf _:x
```

e
```
:primeMinisterOf rdfs:domain _:i
_:x rdfs:subPropertyOf :leaderOf
_:y rdf:type :GeographicalEntity
```

2. Define the four standard inferences in Description Logic and describe the types of checks they are used to perform in Knowledge Engineering. Specify if all the four inferences available are necessary, and motivate your answer with an explanation. **(25 marks)**

UNIVERSITY OF LIVERPOOL

3. Consider an XML document following the fragment of the XML schema below. Provide the XPath expression corresponding to each of the following statements **(25 marks)**:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<xsd: schema xmlns:xsd=:"http://www.w3.org/2001/XMLSchema">
<xsd: element name="company">
   <xsd:complexType>
     <xsd:sequence>
          <xsd:element name="department" type="Department" minOccurs="0"
               maxOccurs="unbounded" />
          <xsd:element name="employee" type="Employee" minOccurs="0"
               maxOccurs="unbounded">

            </xsd:element>
          <xsd:element name="project" type="Project" minOccurs="0">
                <maxOccurs="unbounded" />
      </xsd:sequence>
   </xsd:complexType>

   <xsd:complexType name="Department">
        <xsd:sequence>
             <xsd:element name="departmentName" type="xsd:string" />
             <xsd:element name="departmentLocation type="xsd:string minOccurs="0"
                  maxOccurs="unbounded" />
         </xsd:sequence>
   </xsd:complexType>
   <xsd:complexType name="Employee">
        <xsd:sequence>
             <xsd:element name="employeeName" type="Name" />
             <xsd:element name="employeeSSN" type="xsd:string" />
             <xsd:element name="employeeSalary" type="xsd:unsignedInt" />
             <xsd:element name="employeeDOB" type="xsd:Date" />

         </xsd:sequence>
   </xsd:complexType>
   <xsd:complexType name="Project">
        <xsd:sequence>
             <xsd:element name="projectName" type="xsd:string" />
             <xsd:element name="projectNumber" type="xsd:string" />
             <xsd:element name="projectWorker" type="Worker minOccurs="1"
                 maxOccurs="unbounded" />

         </xsd:sequence>
   </xsd:complexType>

   <xsd:complexType name="Worker">
        <xsd:sequence>
             <xsd:element name="SSN" type="xsd:string" />
             <xsd:element name="hours" type="xsd:float" />
        </xsd:sequence>
   </xsd:complexType>

<xsd:schema>
```

a Return the root node of the XML document and all its descendants.

b Return all department nodes (elements) and their descendant subtrees.

c Return, by specifying the full path, all the`employeeName` nodes that are direct children of an `employee` node having another child element, `employeeSalary` whose value is greater than 85000.

d Return, without specifying the full path, all the`employeeName` nodes that are direct children of an `employee` node having another child element, `employeeSalary` whose value is greater than 85000.

e Return all `projectWorker` nodes and their descendant that are children under a path `/company/project` and have a child node `hours` with a value less or equal to 20.0 hours.