1	2	3	4	5	6	7	8	total
20	10	10	10	20	20	20	20	130

Name:_____ UMBC ID: _____

UMBC CMSC 491/691 Final Exam, 18 December 2017

Write all of your answers on this exam, using the blank side of pages if you need more room. The exam is closed book but you are allowed one piece of letter-sized paper with notes. You have the two hours to work on this exam. Good luck.

1. True/False (20 points)

T F XML documents must include a link to a schema represented in DTD, XSD or some other format. False

T F XML is a metalanguage that allows one to define new tags. True

T F An XML document must have a single root node. True

T F While a DTD schema allows you to declare what are legal tags and their attributes, it does not let you constrain how tags must or can be nested. False

T F RDF's data model is based on a graph consisting of nodes and edges. True

T F While RDF has several standard serializations, we are free to use other ways to serialize an RDF graph. True

T F A shortcoming of RDF is that it is impossible to make a statement about an RDF statement. False

T F In RDFS, a property must have a declared domain and range. False

T F It is not possible to state or infer a logical contradiction in RDFS. True

T F Every graph that uses only RDF and RDFS properties can be rewritten as an equivalent finite graph using only RDF properties. True

T F A property cannot be an rdfs:subPropertyOf itself. False

T F OWL's data model is based on RDF, but introduces terms with new semantics. *True*

T F Reasoning in OWL-DL is decidable. True

T F OWL extends RDFS by allowing inherited properties to be overridden. False

T F The class owl:Nothing is owl:subClassOf every owl class, including itself. True

T F If :R is an owl:FuncionalProperty then no subject can have more than one value for :R. True

T F SWRL rules can infer facts from a graph that an OWL-DL reasoner cannot. True

T F An OWL-DL reasoner can infer facts from a graph that a SWRL reasoner cannot. True

T F The SPARQL language can be used to delete, add or modify triples in a triple store. True

T F Any information that can be embedded in an HTML document using RDFa can also be embedded using Microdata. False

T F RDFS is adequate to define the terms in the Schema.org schemata. True

2. Modeling in RDFS (10 points)

Suppose you are constrained to work only in RDFS and cannot use any OWL vocabulary. You are designing an ontology that has classes for a Person and three kinds of animals: Cat, Dog and Bird. Starting with the triples in the box to the right, you want to add triples so that the :hasPet predicate is assumed to hold between a person and either a dog, cat or bird.

```
:Person a rdf:Class.
:Dog a rdf:Class.
:Cat a rdf:Class.
:Bird a rdf:Class.
:hasPet a rdf:Property.
```

2.1 Is this possible in RDFS? Yes.

2.2 if so, show the triples you should add in Turtle, if not explain why it is not possible.

```
:Pet a rdf:Class.
:Dog rdfs:subClassOf :Pet.
:Cat  rdfs:subClassOf :Pet.
:Bird  rdfs:subClassOf Pet.
:hasPet rdfs:domain :Person.
:hasPet rdfs:range :Pet.
```

3. More modelling in RDFS (10 points)

Assume you are constrained to work only in RDFS and cannot use any OWL vocabulary. Using the initial triples in the box in problem 2, we would like to specify that the :Person class is equivalent to the foaf:Person class. If we were able to use owl vocabulary then we could just say

:Person owl:equivalentClass foaf:Person .

Is there a way to do this using the RDFS vocabulary? If so, show how to do it. If not, explain why it is not possible.

If classes C1 is owl:equivalentClass to class C2 it means that

- every member of C1 is necessarily a member of C2 and
- every member of C2 is necessarily a member of C1

We can also do this using two rdfs:subClassOf assertions:

```
:Person  rdfs:subClassOf  foaf:Person.
foaf:Person  rdf:subClassOf  :Person.
```

4. Doing without owl:SymmetricProperty (10 points)

The set of OWL primitives is not minimal in the sense that some can be defined using others. Take for example, owl:SymmetricPredicate. If :R is an owl:SymmetricPredicate, then if :R holds between O1 and O2, it also holds between O2 and O1. The spouse relation between people is typically modelled as a symmetric predicate. Show how the :spouse relation can be specified to be symmetric without using the owl:SymmetricProperty class.

:spouse owl:inverseOf :spouse

5. Functional and InverseFunctional Properties (20 points)

5.1 Explain the relationship between the concepts owl:FunctionalProperty and owl:InverseFunctionalProperty.

- If R is an owl:FunctionalProperty, then each instance in its domain can have at most one value
- If R is an owl:inverseFunctionalProperty, then for each instance in its domain, if that instance has a value, then no other instance can have the same value.

5.2 For the domain of people, give an example of properties that we might reasonably model as each of the following. Explain your answers.

5.2.a a FunctionalProperty but not an InverseFunctionalProerty

:hasMother is an owl:FunctionalProperty because every person has one and only one (biological) mother but several different people may share the same mother.

5.2.b an InverseFunctionalProperty but not a FunctionalProperty

:hasSSN is generally modeled as an owl:InverseFunctionalProperty since each SSN value uniquely identifies a living person but not every person has a SSN

6. Inferences in OWL (20 points)

Suppose we have the following knowledge graph.

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
:Person rdfs:subClassOf owl:Thing .
:hasParent a owl:ObjectProperty;
        rdfs:domain :Person;
        rdfs:range :Person.
:motherOf owl:inverseOf :hasMother
:hasMother a owl:FunctionalProperty;
        rdfs:subPropertyOf  :hasParent.
:john :hasMother :p1.
:p2 :motherOf :john.
```

Show the triples that can be inferred by a DL-reasoner from the graph above that have :john, :p1 or :p2 as a subject or object.

```
:john a :Person, owl:Thing;
  :hasParent :p1, p2;
  :hasMother :p2.
:p1 a :Person, owl:Thing;
  :motherOf :john;
  owl:sameAs :p2.
:p2 a :Person, owl:Thing;
  :motherOf :john;
  owl:sameAs :p1.
```

7. I'm My Own Grandpa (20 points)

I'm My Own Grandpa is an old novelty song about a man who, through an unlikely but legal combination of marriages, becomes his own grandfather. Complete the following SPARQL query to find examples of a person who is their own grandparent in DBpedia. Recall that the DBpedia relation from a person to one of their parent's is **dbo:parent**.

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>

SELECT  ?P

WHERE {

  # alternative: ?P dbo:parent / dbo:parent ?P
  ?P dbo:parent ?X. ?X dbo:parent ?P

}
```

In the U.S., two people are cousins if they share a grandparent. A person is not her own cousin. Complete the following SPARQL query to find pairs of people in DBpedia who have a cousin relationship. You will need to use the **dbo:parent** relation, of course.

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>

SELECT  ?P1 ?P2

WHERE {

  ?P1 dbo:parent ?X1. ?X1 dbo:parent ?GP.
  ?X2 dbo:parent ?GP. ?P2 dbo:parent ?X2.
  FILTER ( ?P1 < ?P2 )

}
```

8. Embedding semantic markup in HTML (20 points)

Both Microdata and RDFa can be used to embed semantic markup in HTML documents. Describe similarities and differences of the two approaches and identify the major advantages and disadvantages of both.

We didn't talk about **Microformats** this semester because while there are still a lot of web pages that have some microformat markup, it's not being used for new pages much.

Microdata is the native format developed for use with schema.org. It uses HTML tag attributes to encode markup, which is all assumed to refer to schema.org classes and properties. Its chief advantage is simplicity obtained by only allowing schema.org terms. One disadvantage is that it is not possible to use classes or properties from other namespaces/ontologies, like foaf or DBpedia. Another disadvantage is that schema.org is roughly like RDFS and does not have features that OWL adds that support more reasoning, such as being able to declare that a property is symmetric (e.g., spouse).

RDFa also uses HTML tag attributes to encode markup. Its big advantages are that the markup is not restricted to schma.org terms and can include terms from multiple namespaces/ontologies, including new ones that you invent. Since its based on RDF, all RDFS and OWL terms and features can be used. A disadvantage is that using it can be much more complicated than using Microdata, especially if you take advantage of the more complicated features or OWL. Using vocabulary other than schema.org will also not be understood by search engines.

JSON-LD is an approach in which the markup is not mixed in with the HTML, but segregated in (typically one) JSON object. An advantage of this is that it can be easier to generate the markup as a block and add it to the HTML content than it is to distribute the markup throughout the content. A disadvantage is that this introduces a possible problem of redundancy, where the same string (e.g., an address) needs to appear in both the html and JSON content. The burdon of ensuring that these are always the same may fall to the developer.